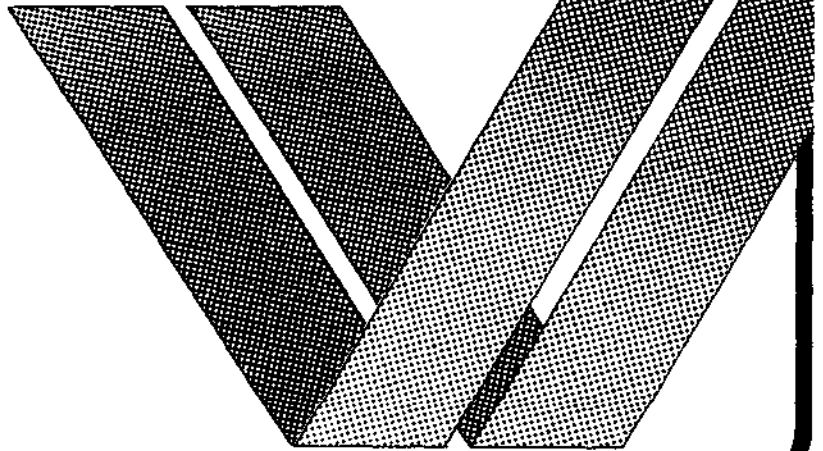


*NAGY
AMIGA-
KÖNYV*



Dr. MÉSZÁROS LÁSZLÓ

*NAGY
AMIGA-
KÖNYV*



Műszaki Könyvkiadó, Budapest, 1990

Lektorálta: Balogh Zsolt

© dr. Mészáros László, 1990

ETO: 681.31 Amiga: 519.682 BASIC

ISBN 963 10 8793 X

TARTALOM

BEVEZETÉS	9
1. AZ ELSŐ LÉPÉSEK	10
2. A WORKBENCH	16
2.1. Formázás és fájlmásolás.....	18
2.2. Workbench menük.....	19
2.3. Notepad.....	23
2.4. Preferences.....	25
2.5. Palette.....	27
2.6. PerfMon.....	27
2.7. Fed.....	27
2.8. NoFastMem.....	28
3. AZ AMIGA-BASIC	29
3.1. Az Edit menü.....	30
3.2. ARunmenü.....	32
3.3. A Windowmenü.....	32
3.4. Alaputasítások.....	32
3.5. Egy színes program.....	34
3.6. Egy kis matematika.....	38
3.7. Az IF...THEN és variációi.....	41
3.8. Grafika.....	42
3.9. Játék az adatokkal.....	44
3.10. Szekvenciális fájl.....	44
3.11. Közvetlen hozzáférésű fájl.....	46
3.12. IFF fájl.....	48
3.13. Menüprogramozás.....	49
3.14. Vezérlés megszakítással.....	50
3.15. Az Amiga beszél.....	52
3.16. Zene.....	53
3.17. Az OBJECTéstársai.....	54
3.18. Ami kimaradt.....	56
3.19. Library.....	61
4. AZ AMIGA-DOSÉSACLI	63
4.1. A DÍJ és társai.....	64
4.2. A CLI és a Shell.....	66
4.2.1. A dzsóker.....	67
4.2.2. A másolás módozatai.....	67
4.3. DOS parancsok egy lemez meghajtóval.....	69
4.4. Startup-sequence.....	70
4.5. További információk a rendszerről.....	72
4.6. Dátum.....	74
4.7. Különböző DOS parancsok vegyesen.....	75

4.8. Idegen nyelvek.....	76
4.9. A Basic programok profi indítása.....	77
4.10. A Basic lemez létrehozása.....	79
4.11. CLI programok.....	79
4.12. Rend a lelke mindennek.....	81
4.13. A gyorsítás egyik módszere.....	82
4.14. A beteghez doktort kell hívni.....	82
4.15. Az 1.3-as változat különlegességei.....	83
5. ANYOMTATÁS.....	85
5.1. Az egyes nyomtatótípusok.....	86
5.2. A Preferencias és a nyomtatás.....	87
5.3. Nyomtatás a CLI-ből.....	88
5.4. Nyomtatás Basicból.....	90
5.5. Hardcopy.....	91
5.6. Nyomtatás szoftverből.....	91
6. SZOFTVEREK.....	92
6.1. Programnyelvek.....	92
6.2. Grafika.....	93
6.3. RayTracing.....	94
6.4. Animáció és videó.....	95
6.5. CAD.....	95
6.6. Szövegszerkesztés.....	96
6.7. Kiadványszerkesztés(DTP).....	97
6.8. Adatbázis-kezelők.....	98
6.9. Könyvelés.....	99
6.10. MS-DOS.....	99
6.11. Segédprogramok (Utilities).....	99
6.12. Másolók.....	101
7. JÁTÉKOK.....	102
7.1. Autóverseny.....	103
7.2. Kalandjáték (Adventure).....	104
7.3. Fantázia (Fantasy).....	105
7.4. Kereskedelem.....	107
7.5. Harci játékok.....	108
7.6. Táblás játékok.....	109
7.7. Sport.....	109
7.8. Akciójátékok.....	110
7.9. Játéktípek.....	110
8. HARDVERKIFÉSZÍTŐK.....	112
8.1. Memóriabővítők.....	113
8.2. Külső lemez meghajtók.....	113
8.3. Monitorok.....	114
8.4. Merevlemez meghajtók.....	114
8.5. Egér.....	115
8.6. Videó.....	115
8.7. Processzorkártya.....	116
8.8. PC kártyák.....	116

9. TIPPEK ÉS TRÜKKÖK	118
9.1. Workbench lemez.....	118
9.2. Lapozás Basicben.....	119
9.3. Törlésmentés.....	120
9.4. A Workbench gyors elérése.....	120
9.5. Ha a CLI nem akar elindulni.....	120
9.6. DOS parancsok másképpen.....	120
9.7. Néhány magyar ékezetes betű egyszerűen.....	121
9.8. Egy kis elővigyázatosság.....	121
9.9. Memóriatakarékosság.....	121
9.10. A misztikus Guru.....	122
9.11. Hiányzó jelek.....	123
9.12. Egy Scribble trükk.....	123
9.13. A dátum beépítése.....	124
9.14. Az ablak aktiválása az egér nélkül.....	124
9.15. Egy gyors levél.....	124
9.16. A nyomtatás kellékei.....	125
9.17. Információ a betöltés közben.....	125
9.18. Átkapcsolás a Basic inzert üzemmódról.....	125
9.19. PD tipp.....	125
9.20. A Rename rejtett képessége.....	126
9.21. A nyíl színei.....	126
10. A VÍRUSOKRÓL	127
10.1. Kórkép.....	127
10.2. Bútblokkvírus.....	128
10.3. Trójai faló.....	128
10.4. Mechanizmus.....	128
10.5. SCA vírus.....	129
10.6. Byte Bandit.....	129
10.7. Byte Warrior.....	129
10.8. Exterminator.....	129
10.9. IRQ.....	130
10.10. Más vírusfajták.....	130
10.11. Védekezés.....	130
10.12. Megelőzés.....	131

BEVEZETÉS

A számítógépek korában élünk. Nap mint nap újabb és újabb konstrukciók jelennek meg, amelyek elkápráztatják a felhasználókat és a számítástechnika iránt érdeklődőket egyaránt. Némelyik gép csak rövid életű, néhánynak azonban megadatik a világsiker. Ez utóbbiak közül mutat be egy gépet, í 11 - gépcsáládot ez a könyv.

Mindazokhoz szól, akik szeretnének megismerkedni az Amigával. Szól tehát azokhoz, akik mielőtt megvennék, szeretnének bővebben tájékozódni. Elsősorban azonban azoknak íródott, akiknek már van gépük. Az egyes fejezetek és a könyv felépítése is olyan, hogy a kezdőknek éppúgy hasznos, mint a profi "amigásoknak". Az egyes részek ismertetésekor a szerző megpróbál azokhoz is szólni, akik a Commodore cég korábbi siker gépéről, a Commodore 64-ről álltak át (vagy állnak át rövidesen) erre a gépre. Számukra az adott helyeken összehasonlítjuk az Amigát a Commodore 64-gyel, rámutatva az egyezőségekre, amelyekkel a Commodore 64-es ismeretekkel rendelkezők elsajátíthatják a gép kezelését.

Néhány alapelv a könyv egészén végigvonul. Az Amigából négyféle modell kapható, az 500-as, az 1000-es, a 2000-es és a 2500-as. és hamarosan megjelenik a 3000-es is. Az egyes modellek egymással kompatibilisek, s főképp teljesítményükben, ill. kibővíthetőségükben térnek el egymástól. A könyvben általában az Amiga a legnépszerűbb — s valószínűleg Magyarországon is a legelterjedtebb — Amiga 500-ast jelenti. Az ismertetés során helyenként felhívjuk a figyelmet a teszvérmmodellek eltéréseire.

A könyvben jó néhány számítástechnikai kifejezés előfordul majd. Arra törekedtünk, hogy a megfelelő, már meghonosodott magyar kifejezést használjuk. Ahol erre nem volt lehetőség, megmaradt az angol kifejezés. A legtöbb számítástechnikai szakkifejezés első előfordulásakor — zárójelbe téve — egy kis segítséget kap az Olvasó. Helyenként az angol szavak kiejtését is közöljük, a magyar kifejezések mellé pedig ugyancsak zárójelben odaírtuk a szó angol megfelelőjét is. Ez utóbbi azért lehet hasznos, mert az Amiga nyelve az angol, s a gép által küldött, a képernyőn kiírt szövegek (pl. hibaüzenetek) angolul jelennek meg, és a szoftverek nagy része is angol nyelvű.

Jó munkát, ill. szórakozást kívánunk minden kedves Olvasónak, remélve, hogy könyvünk is segíti az Amiga kezelésének elsajátítását.

1. AZ ELSŐ LÉPÉSEK

Ez a fejezet az új, tehát kezdő Amiga-tulajdonosokhoz szól. Akinek még nincs gépe — vagy egy bizonyos szinten már kezelni tudja azt — nyugodtan átlapozhatja az első oldalakat. A fejezet végén azonban tanácsot szeretnénk adni a megfelelő Amiga modell kiválasztásához, egy kis összehasonlítást téve. Ezenkívül a gépek felépítését és műszaki adatait ismertetjük, ami nemcsak a kezdők érdeklődésére tarthat számot.

Az Amiga csomagjában kicsomagoláskor három fő egységet találunk. Magát a gépet, a súlyos tápegységet, valamint az ún. *egeret* (mouse). Még egy lényeges egység tartozik a géphez, a *lemez meghajtó egység* (floppy drive), ezt azonban beépítették a gépbe. Ezenkívül néhány fontos apróság van a dobozban; az idegen nyelvű használati utasítás, legalább két — esetleg három, kivételes esetben esetleg több — 3,5-es floppy programlemez, s végül egy kötetben (vagy külön-külön) az Amiga-Basic, ül. a DOS kézikönyv.

Helyezzük el a berendezést, és vegyük szemügyre a hátoldalán elhelyezkedő ki-, ül. bemeneteket! Jobbról balra haladva a következőket találjuk. Az első az egérbemenet, amely ritka esetben (játékban) a második *botkormány* bemenete (joystick port) is. Csomagoljuk ki az egeret, és mindjárt csatlakoztassuk is!

A következő bemenet az előzőhöz hasonló, s a fő botkormány csatlakoztatására való. Ha van ilyen eszközünk, akkor csatlakoztassuk azt is. (Meg kell jegyeznünk, hogy az Amiga ugyanazt a botkormányt használja, amit a Commodore 64-es.) Ez lesz tehát a 2-es joystick port, a legtöbb játék ezt a bemenetet használja.

Ezután a hifiberendezéseken vagy hordozható rádiómagnókon is megtalálható sztereó RCA kimenet következik, ez az Amiga hifi hangkimenete. Ide hangátjátszó kábelt csatlakoztatva összeköthetjük a gépet egy monitorral vagy egy hifizősítővel. Ha nincs még monitorunk, akkor ezt egyelőre hagyjuk üresen, mert ide mást fogunk csatlakoztatni.

Ezt követően egy lapos, széles, 23 pólusú be-, ül. kimenettel találkozunk, ide csatlakoztathatjuk a külső lemez meghajtó egységet; majd az ún. *soros kimenet* (serial port) következik. Ez különleges adatátviteli célokra használható, és némelyik nyomtatót is ide kell csatlakoztatni. Egyelőre ezt is hagyjuk üresen! A következő az ún. *párhuzamos kimenet* (parallel port), ez használatos a legtöbb nyomtató csatlakoztatására.

Az előző csatlakozási pontoknál jóval kisebb, majdnem négyzet alakú bemenet a tápegység csatlakoztatható. Csomagoljuk ki a tápegységet, s még mielőtt a 220 V-hoz csatlakoztatnánk, óvatosan (nem erőltetve) helyezzük be a megfelelő csatlakozót az Amiga említett bemenetébe!

Az utolsó előtti csatlakozási pont a *monitorkimenet*. Ha színes monitorunk van, akkor máris csatlakoztassuk a megfelelő kábellel. Ha nincs monitorunk, de a televízió keresztül színesben szeretnénk nézni az Amigát, akkor egy ún. *modulátorra* van szükségünk, amelyet ide tudunk csatlakoztatni. A modulátorhoz adott RF kábelt, népszerű nevén antennakábelt, az RF out jelzésnél először a modulátorba csatlakoztatjuk, a másik végét pedig a televízió antennabemenetébe helyezzük, hasonló módon, mintha egy videomagnót csatlakoztatnánk a tévéhez. Fontos tudni, hogy a modulátort nem adják az Amigához, ezért ha nincs monitorunk, de színes képet akarunk kapni, akkor ezt a boltban külön kell megvenni.

A modulátorhoz egy kisebb — egyik oldalán egy-, a másikon kétpólusú végződéssel ellátott — csatlakozó is tartozik. Ha a számítógép hangját is a tévén keresztül akarjuk hallgatni, akkor a

kétpólusú részt az előbb már említett hífi kimenethez kell csatlakoztatnunk, az egypólusú részt pedig a modulátor audio in jelzésű bemenetéhez.

Fontos tudni, hogyha valaki az Amigát valamilyen videoprogram feliratozására vagy videotrükkök készítésére is szeretné alkalmazni, a modulátorra akkor is szüksége lesz, ha van monitora.

Eljutottunk a legutolsó — mono feliratú — kimenethez. Ezen keresztül az Amigát a tv videobemenetére csatlakoztatva, fekete-fehérben nézhetjük a számítógépképet.

El kell azonban mondanunk, hogy az Amiga egyik nagy erőssége a grafika, ezért ha fekete-fehérben nézzük, akkor ez kicsit olyan, mintha egy ötsebességes versenyautót csak az egyes fokozatban használnánk. Sajnos a színes tv használata sem az igazi megoldás. A csodálatos színvilág ugyan meglesz, s a játékokhoz ez talán megfelelő is, azonban a tv felbontása az Amiga felbontásánál jóval kisebb, így a szövegek nehezen olvashatók, s a programozás — még jó színkombináció beállításakor is — szemet erőltető művelet. Bizonyos trükkök alkalmazhatók, de az igazi megoldás csak a monitor lehet. (A monitorral kapcsolatos tudnivalók a Hardverkiegészítők c. fejezetben olvashatók.)

Miután a csatlakozási pontokat megismertük, kapcsoljuk be a készüléket úgy, hogy az áramforrást a 220 V-os csatlakozóaljzatba dugjuk, és a tápegységen lévő kapcsolót ON állásba kapcsoljuk. Ekkor a készüléken lévő piros, POWER feliratú lámpa kigyullad, jelezve a bekapcsolt állapotot.

Evvel egy időben a képernyőn megjelenik egy floppyt tartó kéz, s felszólít, hogy helyezzük be a lemezmeghajtóba a *Workbench* (e. vörkbencs) lemezt. A Workbench helyett természetesen programot tartalmazó lemezt is behelyezhetünk. Ha nem helyezzük be semmilyen lemezt, akkor a géppel gyakorlatilag semmit sem tudunk kezdeni. Ha pl. Basicben akarunk programozni, akkor előbb a Basicet is be kell tölteni, ellentétben a Commodore 64-gyel, amely bekapcsolás után rögtön programozható.

Tegyük be tehát a Workbench lemezt a gép jobb oldalán elhelyezkedő lemezmeghajtó egységbe úgy, hogy a lemezt feliratos oldalával felfelé, s fém tolólapkájával előre könnyedén betoljuk. Mielőtt azonban ezt tennénk, nézzük meg, hogy a lemez a véletlen törlés ellen biztosítva van-e. A lemez jobb felső sarkában van egy kis lyuk, ami a lemez hátoldalán a lyuk mögött elhelyezett tolólapkával elzárható vagy üresen hagyható. A tolólapkát állítsuk úgy, hogy a lyukon át lehessen látni. Ilyenkor a lemezre nem lehet írni, tehát véletlenül sem tudunk beletörölni a lemezen lévő programokba. Ez a művelet a Commodore 64 esetében a lemezen lévő kivágás leragasztásával egyenértékű, csak egyszerűbb.

A gép a lemez betétele után automatikusan elkezdi a betöltést. Ezt hívják "*bútoldsnak*" (boot). Eközben eltűnik a képernyőről a lemezt tartó kéz, s kék alapon fehér felirattal bejelentkezik a rendszer. A bútolás 40-50 másodpercig tart, s a folyamatot a piros POWER lámpa alatti zöld. DRJVE feliratú, égló lámpa jelzi.

Amikor a lámpa kialszik, a képernyőn egy majdnem üres keretet látunk. A jobb felső sarokban két, lemezzel emlékeztető kis rajzolat jelenik meg. Ezt nevezik az Amiga esetében *ikonnak* (icon). A felső alatt a *RAM-DISK*, az alsó alatt pedig az *A500 WB 1.2D* vagy valami hasonló olvasható.

A képernyőn lévő keret felső szegélyén a Workbench felirat, egy hatjegyű szám, s az azt követő free memory felirat látható. Ezt a képernyőt nevezik *Workbench képernyőnek* (Workbench screen). A free memory előtt lévő szám a szabad memóriakapacitást jelzi bájtban. Az egészen kezdők kedvéért tennünk kell egy kis kitérőt, hogy ezt a fontos fogalmat — bájt (byte) — egyszerűen megmagyarázzuk.

A számítástechnikában az információ kódolása, továbbítása azon a két állapoton alapszik, hogy folyik-e áram, vagy nem. Ha folyik, akkor annak az állapotnak az értéke mondjuk 1, ha nem, akkor 0. Ez az információ alapegysége: a bit. Mivel a bit csak két értéket vehet fel, a folyamat matematikailag a kettes számrendszerrel írható le.

A bit önmagában nem sok információt tud hordozni, ezért a bitek egy céltudatosan megalkotott sorrendjét, a bájtot alkalmazzuk. 1 bájttal 8 bitet tartalmaz. Tehát pl. az 10010111 kettes számrendszer-beli szám. Az első, második, harmadik, ötödik és nyolcadik bit egyes értékű, a többi pedig ún. nullabit. A kettes számrendszer szabályainak megfelelően ez a bájttal a 151 tízesszámrendszer-beli számnak felel meg.

Ha pl. azt mondjuk hogy az *ú* betű kódja legyen a 151-es tízesszámrendszer-beli szám, akkor a számítógép ezt az előző bájttal képezi le. Leegyszerűsítve: ha egymást követő nyolc ütemben az áramimpulzusok úgy követik egymást, hogy az első három ütemben folyik áram, utána egy szünet következik, majd megint folyik az áram, s ezt két szünet követi, végül ismét folyik áram, akkor az előző bájtot kapjuk.

A bájttal tehát a legkisebb, értelmes információt hordozó egység. Ebből következik, hogy minél nagyobb egy gép bájttal meghatározott szabad memóriakapacitása, annál több információ hordozására, feldolgozására képes, sőt némely esetben az adatok feldolgozási sebességét is befolyásolja a szabad memóriahely.

Látni fogjuk a későbbiekben, hogy a Workbench képernyőn a szabad memóriát megadó szám a programok betöltésekor változni fog, jelezve, hogy a betöltött program (vagy az elvégzendő művelet) szabad memóriakapacitást köt le.

Mielőtt továbbmennénk, hajtsunk végre egy fontos biztonsági intézkedést: készítsünk másolatot a gépben lévő Workbench lemezről! Ezután csak a másolatot használjuk, az eredetit pedig biztos helyen őrizzük, s csak akkor vegyük elő, ha a másolat valamilyen oknál fogva tönkrement, vagy újabb másolatot akarunk készíteni. Ezt a módszert érdemes alkalmazni minden fontosabb szoftverlemez esetében is, így mindig lesz biztonsági tartalékunk.

A másoláshoz azonban üres lemezre is szükségünk van. Ha ezt elfelejtettük beszerezni, akkor minél előbb vegyünk legalább egy-két csomaggal. Mivel az Amiga a lemez (amit hivatalosan microfloppynak vagy 3,5-es disknek hívnak) mindkét oldalára ír, ezért *DS DD* (double sided — double density), azaz kétoldalas, dupla sűrűségű lemezt vegyünk. Anyagi okokból vehetünk *SS DD* (single sided — double density) jelű egyoldalas lemezt is, ennek azonban van bizonyos kockázata, mert az ilyen lemezen csak az egyik oldal hibamentességét ellenőrzik a gyártóhelyen. Azért általában az ilyen lemezek 95 %-ával nem lesz gondunk. Sokan használnak ún. *NO NAME*, azaz márkanév nélküli lemezt is. Ez ugyan jóval olcsóbb, mint a márkás, de hasonló kockázatot jelent, mint az egyoldalas lemez, mert ennek a lemeznek már egyik oldala sem ellenőrzött. Fontos adatainkat mindenesetre ne ilyen lemezre vigyük fel.

Most már tényleg készítsük el a Workbench lemez másolatát! Ehhez használatba kell vennünk az Amiga egyik kulcseszközét, az egeret. Az egeret a gép mellett helyezzük el úgy, hogy kb. 25x25 cm-es területen szabadon mozogjon. Ha az egeret az asztalon mozgatjuk, a mozgás irányának megfelelően a képernyőn lévő kis nyíl is mozogni fog. Az egeren van két gomb is, a jobb oldali (right button), ill. a bal oldali (left button).

Az eger működtetése igen egyszerű. Addig mozgatjuk, amíg a nyíl a kiválasztott helyre kerül a képernyőn, s ott az egeret a jobb vagy bal oldali gomb megnyomásával (click) működésbe hozzuk. Próbáljuk is ki mindjárt!

Mozgassuk úgy az egeret, hogy a nyíl az A500 WB 1.2 D feliratra (a Workbench lemez ikonjára) mutasson. Határozott kattintással nyomjuk le a bal oldali egérgombot. Az ikon ekkor színt vált. Ezután nyomjuk le a jobb oldali gombot, de ezt most tartsuk is lenyomva, és mozgassuk az egeret oly módon, hogy a nyíl a képernyő bal felső sarkában megjelenő Workbench feliratra mutasson. Majd továbbra is lenyomva tartva a jobb oldali gombot, húzzuk a nyílat a Workbench felirat alatt megjelenő függőleges téglalap Duplicate (másolás) feliratára, s amikor az fekete alapon sárga feliratnak látszik, engedjük el az eger gombját. Ezzel aktiváltuk a beépített másolóprogramot. Mi is történik ilyenkor?

A képernyő bal felső sarkában egy kis keret jelenik meg, amely angolul felszólít arra, hogy helyezzük el a Workbench lemezt — hiszen ezt akarjuk másolni — a DFO: jelzésű (azaz a belső) lemezmeghajtóba. Ezt a kis keretet kérdező ablaknak vagy egyszerűen *kérdezőnek* (Requester) nevezzük. Mivel a kívánt lemez már ott van, vigyünk az egerrel a nyílra a kis keretben található Continue (folytasd) felíratra, és nyomjuk le egyszer az eger bal oldali gombját.

A DRIVE feliratú lámpa újból kigyullad, s halljuk, ahogy a gép elkezd betölteni a Workbench lemezt. A bal felső keretben számok jelennek meg, amelyek folyamatosan jelzik, hány blokkot olvasott már be a gép, és mennyi van még hátra. Egy idő után a folyamat leáll, s a bal felső keretben megjelenik a felszólítás, hogy tegyük be az üres lemezt, amire a másolatot készíteni fogjuk.

A lemezmeghajtó egység nyílása alatt lévő gomb határozott megnyomásával vegyük ki a Workbench lemezt, majd a már ismert módon, tegyük be egy üres lemezt, ügyelve arra, hogy ez — ellentétben a kivett, ún. forráslemezzel — ne legyen írásvédett, tehát a kis tolóablak zárva legyen. Ezután ismét kattintsunk rá a Continue felíratra. A gép a már ismert módon most azt jelzi, hogy mennyit másoltunk át, s még mennyi hely van a lemezen.

Nemsokára jön az újbóli felszólítás, hogy tegyük vissza a Workbench lemezt, hiszen a gép, amelynek kapacitása 512 kb-át, a 880 kb-át kapacitású lemezt egy menetben nem tudja lemásolni. A lemezcserét mindaddig ismételjük, amíg a gép nem jelzi, hogy a másolás befejeződött.

Ha menet közben bármikor abba szeretnénk hagyni a másolást, akkor a Continue felirat helyett a Cancel felíratra kattintsunk rá. Természetesen ilyenkor az addig másolt részeket elveszítjük, s újbóli másolási szándék esetén az egészet előlről kell kezdeni.

Ezt a kicsit körülményes másolási módot valószínűleg csak kezdetben kell használnunk, hiszen igen jó másolóprogramok léteznek, amelyek gyorsaságban és kényelemben (pl. két vagy több lemezmeghajtó használata) messze felülmúlják az előbbit, nem beszélve arról, hogy bármilyen egyszerű védelem esetén a Workbench másolóprogramja leáll, s nem végzi el a másolást.

Ha másolás közben valamilyen komoly hibát vétettünk, és az Amiga esetleg lemerevedik, vagyis nem reagál semmire (ilyen a későbbiekben is előfordulhat), vagy a programból szeretnénk kilépni és visszajutni a Workbench-hez, de az valamilyen oknál fogva nem lehetséges; akkor nincs más hátra, mint a kikapcsolás vagy a sokkal egyszerűbb *reszetálás* (reset), azaz a gép alaphelyzetbe állítása. Ilyenkor a bent lévő lemezt előbb vegyük ki. A reszetálás a Ctrl, valamint a két A jelű, ún. *Amiga billentyű* (Amiga buttons) 1-2 másodpercig tartó együttes lenyomásával lehetséges. A képernyő elsötétedik, majd megjelenik a korábban ismertetett alapkép.

A reszetálás a megoldás akkor is, ha megjelenik a *Guru Meditation*. Mi is ez a Guru? A Guru szó Indiából származik. Azokat az embereket hívják így, akik a lélek mélyére látnak, a lélekkel foglalkoznak, s bajait gyógyítani tudják. A Guru egyik fő foglalatossága a meditáció.

Ha az Amigán megjelenik a Guru Meditation felirat, akkor nagy hiba történt, ugyanis a Guru Meditation az Amiga hibakijelzési módja. Ilyenkor a képernyő fekete lesz, és a képernyő felső negyedében egy piros villogó keretben, piros felírat jelzi a hibát, általában egy kicsit misztikusnak tűnő, számokból és betűkből álló 16 tagú kifejezéssel. A kifejezések jelentéséről a későbbiekben szólnunk még.

A legtöbb esetben a reszetálással azonos hatású a bal oldali egérgomb lenyomása is. A megjelenő angol szöveg is erre szólít fel.

Sajnos a korai Amiga szoftverekben elég gyakran megjelenik a Guru, de mint majd a későbbiekben látjuk, ha programozunk, mi is könnyen előidézhetjük.

Sok mindennel megismerkedtünk már, itt az ideje, hogy néhány pillantást vessünk a billentyűzetre is, amit az Amiga 500-asban a számítógépházba építettek, a többi modellben pedig egy külön csatlakoztatható egységbe. A billentyűzet jól áttekinthető, a számok és a betűk részben megfelelnek az írógép billentyűzetelhelyezésének, természetesen a magyar ékezetek nélkül. Akik német nyelvterületen vették a gépet, azoknak van *u,ó* betűjük. Egy kis trükkkel azonban a többi magyar ékezet is részben vagy egészben előcsalogatható. Erről később, az Amiga-Basic, ill. a Nyomtatás c. fejezetben lesz szó.

A betűk körül néhány fontos, szürke színű segédbillentyű helyezkedik el. A legfelső sorban találjuk az *Escape* (e. eszkép) — Esc jelű — billentyűt, valamint az ún. *funkcióbillentyűket* F1-től F10-ig jelölve. Ezeket jó néhány szoftver alkalmazza. A bal oldalon található a *Ctrl* (e. kontroll) billentyű, valamint az egyszerre jobbra és balra nyilat is tartalmazó *tabulátorbillentyű*. A Ctrl mellett van a *Caps Lock* (e. kepsz lok), lenyomásakor egy kis piros lámpa is kigyullad rajta. Hatására a betűk mindaddig, amíg újból megnyomjuk, nagybetűként jelennek meg. A Ctrl és a Caps Lock alatt van az egyik *Shift* (e. sif t) billentyű, amelynek hatása azonos a Caps Lockkal, de a Shift billentyű csak addig ad nagybetűt, amíg lenyomva tartjuk. Végül a bal oldalon alul található az *Alt* és az *A* jelű, ún. *Amiga billentyű*. A betűs billentyűzet bal oldalán a nagy, derékszögben meghajlított nyilat tartalmazó billentyű az *Enter* vagy *Return* (e. ritorn). Ez alatt van a második, az elsővel azonos hatású Shift billentyű. Legalul pedig a másik Alt és Amiga billentyű. A Return felett a bal felé mutató nyíl a torlés.

Ezzel megismertük a főblokkot. Látható, hogy jobb oldalon is van egy, főképp számokból álló, kisebb blokk. Ha elsősorban számokat akarunk betáplálni, akkor jól kézzerező helyen együtt vannak a számok. Ezek az Amiga 2000 IBM üzemmódjában IBM, a vele kompatibilis gépek esetében pedig a megszokott billentyűcsoportként használhatók.

A két ismertetett blokk között van egy harmadik kisebb blokk. A nyilat tartalmazó billentyűk a *kurzorbillentyűk* (cursor), amelyekkel a nyírnak megfelelően mozoghatunk a képernyőn. A *Del* és a *Help* billentyű pedig két, viszonylag ritkán használt segédbillentyű.

Miután tényleg megtettük az első lépéseket, ismerkedjünk meg jobban az Amigával és paramétereivel, s ekközben néhány fontos szakkifejezéssel is. Egy kis történeti áttekintéssel kell kezdenünk. Az első Amiga, az Amiga 1000, néhány éve jelent meg. Eredetileg játékgépnek szánták a tervezők, s habár a grafikában és a zenében mérőföldkövet jelentett, a számítástechnikában — viszonylag magas ára miatt — nem volt igazán keresett. A tapasztalatokon okulva a Commodore cég nem sokat késlekedett, s kidolgozta az 1000-es két követőjét. Az Amiga 500-at, amit kedvező árú házi számítógépnek szántak, és az Amiga 2000-t, amivel a profi kategóriát célozták meg a tervezők. A siker nem is maradt el.

Mindhárom modell alapját a Motorola 16/32 bites *68000-es mikroprocesszora* képezi. Az 1000-es 256 kb-át, az 500-as 512 kb-át, míg a 2000-es 1 Mb-át alapkonfigurációval — RAM memóriával — kerül forgalomba. Itt álljunk meg egy pillanatra.

A számítógépben általában kétféle tár, ill. memória van. Az egyik az ún. RAM (Random Access Memory), amelybe mi magunk tudjuk bevinni az információt, ide töltődnek be a programok is. A másik a ROM (Read Only Memory), ami gyárilag már betöltött memóriahely, itt vannak azok a rutinok, amelyek a számítógép beindulásához, alapműködéséhez szükségesek. Ide a felhasználó nem tud beírni, ezt csak olvasni tudja. Az alap/eto különbség tehát a kétféle tár között az, hogy a RAM-ot szabadon feltölthetjük és olvashatjuk, de a gép kikapcsolásakor a tartalma elvesz, míg a ROM-ból csak olvasni lehet, a benne levő adatok viszont a számítógép kikapcsolásakor is megmaradnak.

Az Amiga 1000-be a kétféle tár (RAM és ROM) különleges kombinációját építették be, az ún. WOM-ot (Write Only Memory). Ez a tár úgy működik, hogy a számítógép bekapcsolásakor a gép működéséhez szükséges *Kickstart* rendszerszoftvert lemezzről be kell tölteni, azután a tár már úgy viselkedik, mint a ROM, tehát felülírni nem tudjuk. (Ez a Kickstart 1.1-es változata.) A többi Amigában a Kickstart be van építve a ROM-ba, tehát a gép bekapcsolásakor már jelen van. Ez a Kickstart 1.2-es, a legújabb pedig a Kickstart 1.3-as.

Fontos adat a gépekben az *órajel*, ami — ha nem is közvetlenül, de összefüggésben a többi paraméterrel — a számítógép gyorsaságára utal. Az említett három modellben ez az órajel 7,14 MHz-es. Az Amiga tehát a mai leggyorsabb számítógépek közé tartozik.

A legújabb modell, az Amiga 2500 UX, további előrelépést jelent. Ez már a Motorola legújabb, 68020-as processzorával készül, amelynek órajele 14,3 MHz. Ennek a modellnek a különlegessége, hogy az Amiga rendszerszoftveren kívül az ún. Unix rendszert is beépítették, erre utal az UX jel. Ezáltal a Unix rendszerű szoftverek is futtathatók rajta. Az sem elhanyagolandó tény, hogy e modell már alapkonfigurációban is 3 Mbájtos RAM-mal dolgozik.

Annyit talán még illik tudni a 68000-es processzorról, hogy két másik, széles körben elterjedt számítógép — az Atari ST és az Apple — "lelkét" képezi.

Ha röviden tanácsot kellene adnunk, hogy mi szerint válasszuk ki a nekünk megfelelő modellt, akkor a következő néhány szempontot említenék meg. Az Amiga 1000-es ma már túlhaladott (kapni sem nagyon lehet), nem is ajánlatos megvenni a viszonylag kis RAM és a Kickstart 1.1 miatt. Egyedül az szól mellette, hogy elvétve igen nagy árengedménnyel kínálják. A Commodore 64 igazi utóda az Amiga 500. Ez a házszámítógépek kategóriájába tartozó gép, amely sok szempontból megközelíti az ún. PC-ket. Ma már igen kedvező áron kapható. Egyértelműen a PC kategóriába tartozó professzionális számítógép a 2000-es és a 2500 UX. Itt — ellentétben az Amiga 500-zal — a gép és a billentyűzet már külön egység. Az alapgépben 1 Mbájtos RAM van. Az igazi eltérés azonban az, hogy a PC kategóriának megfelelően négy kártyával bővíthető, s az egyik helyre PC/XT kártyát helyeztek el, ezért ezek az Amigák IBM kompatibilisen is tudnak működni. A gép egy 5¹/₄-es lemez meghajtó egységet is tartalmaz. Meg kell jegyezni, hogy a kisebb modellek is IBM kompatibilissé tehetőek, de sokkal körülményesebben. A 2000-es (de főleg a 2500 UX) árát tekintve is a professzionális kategóriába tartozik.

Sok szó esett már arról, hogy a nagyobb kompatibilitás érdekében az Amiga IBM PC-ként is alkalmazható. Ráadásul nem is akárhogyan. Párhuzamos üzemmódban (*multitasking*) egyszerre tud Amigaként és IBM-ként működni. A multitasking is az Amiga nagy erőssége. Ez azt jelenti, hogy a gép párhuzamosan több feladatot tud ellátni, a képernyőn egyszerre lehet jelen egy IBM és egy Amiga képernyő, s a gép egymás mellett tud Amiga és IBM programokat futtatni.

Miután sikerült gépünkkel az első alaplépéseket megtenni, és megismerkedtünk néhány kimagasló műszaki paraméterével, egy kis trükköt mutatunk, amelyből megtudhatjuk, kiknek is köszönhetjük ennek a nagyszerű gépnek a megalkotását. Tudomásunk szerint a konstruktőrök hivatalosan sehol nem írták le az eljárást, feltehetőleg csak saját szórakozásukra rejtették el a rendszerben. Tulajdonképpen az ilyen (és ehhez hasonló) fogásokra kísérletezéssel lehet rájönni. S mivel valószínűleg jó néhány ilyen még rejtve van, szeretném bátorítani az Olvasót, hogy kísérletezzon. Lehet, hogy maga is érdekes felfedezéseket fog tenni. Lássuk, mit is kell csinálni!

Először, a fejezet első részében ismertetett módon, töltsük be a Workbench lemezt! Ha ez megtörtént, akkor egy nehéz ujjgyakorlatot kell végrehajtanunk, de mindjárt segítségül is hívhatunk valakit, talán könnyebb lesz. Nyomjuk le a két Shift billentyűt, majd folyamatosan nyomva tartva őket, nyomjuk le a két Alt billentyűt is, s ezeket is nyomva tartva, egyenként nyomogassuk az F1 — F10 funkcióbillentyűket. A szöveg a Workbench képernyő felső keretében fog megjelenni.

2.AWORKBENCH

A Workbench szó szerinti fordításban: munkaasztal; s ez tulajdonképpen jól kifejezi funkcióját. A fogalmat legkönnyebben a Commodore 64 ismerői értik meg, mert annak mindenki által jól ismert Geos-ához hasonlítható, messze túltéve rajta. Egy másik megközelítéssel élve, azt is mondhatjuk, hogy a Workbench az Amiga legkülső héja, felszíne. Miért lehet ezt mondani?

Aki számítástechnikával egy kicsit már foglalkozott, jól tudja, hogy minden gépnek van saját nyelve, és ez a processzortól függ. Az előző fejezetben már említettük, hogy az Amigában ez a processzor a Motorola 68000-es. Ha ezen a nyelven akarunk "beszélni" a géppel, akkor mindent gépi kódban kell közölni. Ez a legközvetlenebb, a leggyorsabb, de egyben a legnehezebb, a legkevésbé felhasználóbarát módszer. E felett áll az ún. Command Line Interface (e. kommand lájñ interfész), a CLI, amelyről a 4. fejezetben részletesen szólunk. Ennél eggyel magasabb kommunikációs szintet képvisel a Workbench. Ez testesíti meg az Amigában létrehozott forradalmi újítást, az ún. intuíción.

Az intuíción elve nagyon egyszerű, és főképp a kezdő, ill. amatőr felhasználók számára kézenfekvő. Ha ugyanis a Workbench szintjén használjuk a számítógépet, szinte a billentyűzet érintése nélkül, kizárólag az egérrel végezhetjük a munkát. Különböféle önmagukat magyarázó szimbólumokat, ill. szimbólumrendszereket használhatunk, amelyek aktiválása egy-egy funkcion végrehajtását jelenti. Már szölvünk az ikonról, amely az intuíción egyik fontos eleme. Ez a kis ábra jelölheti a programjainkat, de a lemezmeghajtóban található lemezt is.

A Workbench azonban nemcsak a felhasználó és a számítógép közötti forradalmi kapcsolatteremtési módot jelenti, hanem a gyakorlatban is számos kisebb felhasználói programot és rutint tartalmaz, amelyek könnyítik a munkát, és amelyeket más számítógépekhez külön szoftverként kellene beszerezni. Tehát a Workbench kifejezés egyszerre jelöl egy kommunikációs szintet és egy szoftvergyűjteményt.

A Workbench egyik programját az előző fejezetben már használtuk, amikor másolatot készítettünk a Workbench-ről. Ez volt a DiskCopy program. Már régóta adnak a gyári csomagolásban a Workbench lemezhez egy kiegészítő lemezt is, az Extrás lemezt, amely a számítógép kiterjedt használatához nyújt további felhasználói programokat.

Idővel a Workbench is fejlődött, és a korábban említett Kickstarttal együtt, különböző változatok keletkeztek. Korábban az Amiga 500-as a Workbench 1.2-vel került forgalomba, ma pedig már a Workbench 1.3 jár a géphez. Léteznek azonban a két változat között átmenetet képező, nem végleges Workbench 1.3-as verziók is. Ha lehet hinni a híreknek, már készül az 1.4-es változat is. Ha valaki szeretné tudni, hogy az igazi 1.3-t használja-e vagy sem, akkor könnyen ellenőrizheti. A hivatalos 1.3-as változat a 34.20 verziószámot viseli. A következőkben mindkét változatot (1.2 és 1.3) ismertetjük. Néhány — ugyan igen fontos — eltéréstől eltekintve a két Workbench, ill. programjaik nagyrészt azonosak, működésük és logikai felépítésük hasonló.

Kezdjünk hozzá ! (Ha már betöltöttük a Workbench-et!) A Workbench ikonját és egy fejléctet látunk a képernyőn. Nézzük meg, mit is tartalmaz a lemez! Amiga módon kérjünk egy tartalomjegyzéket (directoryt)! A nyilat vigyük az ikonra, és a bal oldali egérgombbal kétszer, gyors egymásutánban kattintsunk rá! A lemezmeghajtó zümmögni kezd, és a képernyő közepén egy kis ablak nyílik, amely rövidesen megtelik különböző ikonokkal.

Mindjárt feltűnik, hogy az ikonok nagy része olyan, mint egy fiók (drawer). Ezek különleges ikonok, mert — a lemezikonhoz hasonlóan — kinyitva további ikonokat foglalnak magukba. Ezek a lemezen jól elkülönülő alegységek. A lemezen lévő programok jól rendezhetők velük, mert az azonos jellegű programok befoglalhatók egy ilyen fiókba, sőt a fiókok további fiókokat is tartalmazhatnak. Ha a lemezen lévő programjaink egymástól független témájúak, és számuk eléri vagy meghaladja a tizenötöt, akkor már nehezen, esetleg csak egymást átlapolva férnek el az ablakban. A fiókok mindenképpen javítják a rendezettséget és az áttekinthetőséget.

Itt láthatjuk még a szemétkosár formájú, Trashcan névre hallgató ikont, az 1.2-es változatban pedig egy Clock feliratú, óra formájú ikont stb. Nézzük sorjában ezeket! A lemezikon aktiválásakor megjelenő keret az ablak (window), amely önmagában is sok érdekes információt, ill. műveletet hordoz. A bal oldalon az *E* és *F* betű között egy sárga oszlop húzódik, ez a lemez telítettségét mutatja, azaz, hogy a rendelkezésre álló mintegy 880 kb-ájtól mennyit használtunk fel. Ha a sárga oszlop hiányzik, akkor a lemez teljesen üres. Ez az információ pontatlan (csak közelítés), de pontos (számszerű) információt is kaphatunk a telítettségéről.

Az *F* betű felett egy kis négyzet van, közepén egy ponttal. Ha erre a pontra helyezzük a nyilat, és a bal oldali egérgombot egyszer megnyomjuk, az ablak azonnal eltűnik. Így lehet a megnyitott (open) ablakot becsukni (close). (Most még ne zárjuk be a Workbench ablakát, hanem vizsgálódjunk tovább!)

Az ablakot a képernyő bármely részére egyszerűen eltolhatjuk. Ez akkor lesz fontos, ha majd több ilyen ablak lesz nyitva a képernyőn, és azt akarjuk, hogy lehetőleg ne (vagy csak a legszükségesebb mértékben) fedjék egymást. Vigyünk a nyilat az ablak felső keretéhez, nyomjuk le a bal oldali egérgombot, és folyamatosan nyomva tartva mozgassuk az asztalon az egeret! Az ablak az egerrel együtt fog mozogni.

Ha az egeret az ablak jobb alsó sarkába visszük, és itt lenyomjuk a bal oldali gombot, miközben mozgatjuk az egeret, akkor az ablak formáját és méretét tetszőlegesen változtathatjuk a képernyő adta méretek között.

Akik az 1.3-as változattal dolgoznak, kicsinyítsék le az ablakot! Látni fogják (az 1.2-es változatban már alaphelyzetben így van), hogy az ablak alsó és jobb oldali keretében egy-egy fehér — nevezzük így — lécs van, amit az előzőekhez hasonlóan (a nyállal, ill. az egerrel) mozgathatunk, és amely mozgatás hatására az ablak tartalmát képező ikonok is jobbra-balra, ill. fel-le mozognak. Ez azért igen hasznos, mert az ablakba sokszor nem fér be az összes ikon, és ilyenkor az ismertetett módszerrel tudjuk előcsalogatni őket.

Az ablak jobb felső sarkában látunk még két rajzot, egy fekete négyzetet fehér alapon és egy fehér négyzetet fekete alapon. Ezzel az egymást esetleg átlapoló, eltakaró ablakok közül a hátul lévő előre hozhatjuk, ill. az elől lévő a háttérbe vihetjük. Ezt csak akkor tudjuk kipróbálni, ha további ablakokat nyitunk ki, hiszen még csak egy ablakunk van.

Az itt elmondott ablakműveletek nemcsak a Workbench-re érvényesek, hanem bármelyik ablakra, amellyel a későbbiekben lesz dolgunk.

Nyissunk ki egy újabb ablakot! Kattintsunk rá gyorsan kétszer a Utilities feliratú fiókra! (Jegyezzük meg, hogy az ikonokat kinyitni, ill. a későbbiekben ismertetendő programikonokat elindítani is kétszeri rákattintással lehet.)

A rövid töltés után kinyíló ablak további ikonokat tartalmaz. Az 1.2-es változatban a Notepad és a Calculator feliratú, az 1.3-ban pedig e kettőn kívül további nyolcat. Mivel nem célunk a Workbench összes ikonjának ismertetése — ezt a felhasználói kézikönyv többé-kevésbé megteszi —, ezért a változat megnevezése nélkül sorra vesszük a fontosabb Workbench ikonokat, ill. programokat. Ez a módszer azért is célszerű, mert az egyes programok ugyan mindkét változatban megtalálhatók, de nem mindig ugyanabban a fiókban, sőt nem is mindig

ugyanazon a lemezen; hiszen a Workbench szerves kiegészítője az Extrás lemez is. Ebből következően az Extrás lemez néhány fontosabb alkotóját is nagyító alá vesszük. Az áttekintés megkönnyítésére egy-egy rövid táblázatban összefoglaljuk a két változat lemezeinek tartalmát (1. az 1. és 2. táblázatot). Mielőtt azonban az egyes programokat részletesebben is ismertetnénk, meg kell ismerkednünk néhány Workbench-csel kapcsolatos művelettel.

2.1. Formázás és fájlmásolás

Talán az egyik leglátványosabb művelet az egyes programok vagy adatok (az azokat jelentő fájlok) másolása. Ez azonban a Commodore 64-gyel még viszonylag jó fájlmásoló programmal is körülményes művelet, fájlmásoló program nélkül pedig rendkívül bonyolult. Az Amigával viszont gyerekjáték.

Először azonban készítsünk egy üres lemezt a másoláshoz! Az új lemez ugyanis még nem jó, más számítógép-rendszerekhez hasonlóan a lemezt először *formázni* (formát) kell. A formázás azt jelenti, hogy a számítógép a még érintetlen lemezt a számítógépre jellemző bizonyos egységekre, ill. alegységekre rendezi, amelyek a tulajdonképpeni lemezműveletekben segítik a lemezmeghajtó író-olvasó fejének eligazodását.

Képzeld el, hogy kiadjuk a parancsot a számítógépnek, hogy töltsön be egy programot. Ha a lemeznek nem volna meghatározott felépítése, akkor a fejnek végig kellene pásztáznia a lemezt, és még mindig kérdéses lenne, hogy miről ismeri meg a programunkat. Ezért az Amiga esetében a lemez *sávokra* (track), a sávok pedig *szektorokra* (sector) vannak felosztva. A lemez mindkét oldalán (az Amiga a lemez mindkét oldalára ír) 80 sáv van, tehát összesen 160. Egy sáv 11 szektort tartalmaz.

Vannak kitüntetett sávok is. Számunkra most a 40-es sáv a fontos, ugyanis formázáskor minden esetben itt jön létre a lemez tartalomjegyzéke (directory). Ha tehát valamilyen programot be kell tölteni, akkor a gép először a tartalomjegyzéket nézi meg, és ha ott fel van írva a keresett program neve, akkor mindjárt ki is tudja olvasni, hogy a lemezen mely sávokból, szektorokból kell betölteni. Természetesen a valóságban ez nem ilyen egyszerű, de a folyamatnak ez a lényege.

A formázás az 1. fejezetben bemutatott lemezmásoláshoz hasonlóan megy végbe. Helyezzük be a megformázatlan üres lemezt a Workbench lemez helyére! (Akiknek van külső lemezmeghajtójuk, azok egyszerűen a külső lemezmeghajtóba tegyék.) Kattintsunk rá a formázandó lemez ikonjára! Aktiváljuk — a jobb oldali egérgomb lenyomva tartásával — a Disk menüből az *Initialize* funkciót! Egyetlen lemezmeghajtó esetén a gép kéri, hogy helyezzük vissza a Workbench lemezt, majd rövid töltés után, újból tegyük be a formázandó lemezt. Külső meghajtó használatával a lemezváltás elmarad. Másfél perc múlva, a művelet befejezésekor, a kész lemez ikonja alatt az Empty (üres) felirat olvasható.

Ezután nyissuk ki az Empty lemezt! (Kattintsunk rá kétszer!) A képernyőn egy Empty feliratú, majdnem üres ablak jelenik meg, csupán a Trashcan ikon van benne. Az átmásoláshoz válasszuk a Calculator programot! Ehhez ki kell nyitni a Workbench lemezen található Utilities fiókot. Akinek csak egy meghajtója van, az tegye vissza a Workbench lemezt a lemezmeghajtóba, mert csak ezután lehet a fiókot kinyitni. Ha szükséges, úgy állítsuk be az ablakokat, hogy a Utilities fiók most megnyitott ablaka ne fedje az előzőleg formázott üres lemez ablakát!

Kezdd el a másolást. Vigyük a nyilat a Calculator ikonra, és a bal oldali egérgomb lenyomva tartásával mozgassuk úgy az egeret, hogy az ikon az üres lemez ablakába kerüljön. Ekkor engedjük el az egér gombját. A meghajtó zümmögni kezd, és végbemegy a másolás. Azonban álljunk meg egy pillanatra! Mit tegyen az, akinek csak egy lemezmeghajtója van? Amikor

ugyanis az ikont behelyezzük az üres lemez ablakába, a bal felső sarokban megjelenő kérdező (Requester) felszólít, hogy tegyük be az Empty lemezt, majd nemsokára azt kéri, hogy tegyük be újból a Workbench lemezt. S ez mindaddig tart, amíg végül végbemegy a másolás. Ez már nem tűnik olyan egyszerűnek, és ha sokszor másolunk, a gyakori lemezkivétel és -behelyezés valószínűleg a lemezmeghajtónak sem használ. Erre mondják, hogy szegény ember vízzel főz, de nem úgy az Amiga. Egy egyszerű trükkel segíthetünk a gondon.

A Workbench képernyő jobb felső sarkában, a Workbench lemezikon felett, van egy másik lemezikon is, a RAM DISK. Mondhatnánk, hogy a Workbench-en, ill. az üres új lemezen kívül nem tettünk a gépbe lemezt. Ez az ikon azonban már kezdettől fogva itt van. Amikor a bekapcsolás után betöltöttük a Workbench lemezt, annak ikonjával együtt jelent meg. Az Amiga ugyanis saját memóriájából lefoglal egy dinamikusan bővíthető kisebb területet, amely ettől kezdve úgy működik, mint egy külső lemezmeghajtó, amiben üres lemez van. Ide ugyanúgy tudunk programokat írni, s innét ugyanúgy tudunk programokat betölteni, mint a szokásos lemezről. Az alapvető különbség az, hogy a számítógép kikapcsolásakor az ide írt adatok, programok elvesznek, ill. ha ezekre később szükség lenne, akkor azokat a kikapcsolás előtt valódi lemezre kell kimenteni.

Megkönnyíti a helyzetünket, hogy a RAM DISK jóval gyorsabb, mint az igazi lemezmeghajtó, hiszen itt valójában nem lemez forog, hanem a memóriából kell kiolvasni az adatokat. A RAM DISK dinamikus bővíthetősége azt jelenti, hogy a rendszer folyamatosan figyeli, mennyi adatot kívánunk bevinni a RAM DISK-be, és éppen annyi memóriahelyet bocsát rendelkezésünkre. Ezáltal nem foglaljuk le feleslegesen a számítógép memóriáját. A RAM DISK kapacitásának határát természetesen a számítógép memóriakapacitása szabja meg.

De nézzük az ígért trükköt! Legyen a meghajtóban a Workbench lemez, természetesen a nyitott Utilities fiókkal. Nyissuk ki a szokott módon a RAM DISK ablakát is! Várhatóan egy üres ablakot kapunk. Ha az előbb a Calculator programot sikerült átmásolni, akkor másoljuk át a Notepad nevű programot is. Vigyük át a Notepad ikont a RAM DISK ablakába. Ha a lemezmeghajtó zümmögése megszűnik, és a DRIVE lámpa elalszik, akkor cseréljük ki a Workbench lemezt az üres lemezünkkel, és a RAM ablakból vigyük át a Notepad ikont az üres lemez ablakába! Rövid töltés után a másolás megtörténik. Így már jóval egyszerűbb volt a másolás. Ha nincs szükségünk az átmásolt programokra, akkor azokat könnyen ki is törölhetjük, ehhez azonban meg kell ismerkednünk a Workbench menürendszerével.

2.2. Workbench menük

Menünek nevezzük az olyan felsorolást, "étlapot", amelyről az egyes "fogásokat" kiválasztva bizonyos feladatokat elvégezhetünk. A menükkel jól áttekinthetjük a program funkcióit. Egy programban több (akár egymásba épülő) menürendszer is lehet. Aki már foglalkozott számítástechnikával, tudja, hogy a legegyszerűbb menüforma az, amikor a képernyőn — általában sorrendben következő — számok mellé vannak felírva az egyes funkciók, pl:

- 1.) Betöltés
- 2.) Törlés
- 3.) Vége

A szükséges funkciót a megfelelő szám beadásával választhatjuk ki.

Mivel az Amigát úgy szerkesztették meg, hogy a billentyűzethez ne nagyon kelljen nyúlni, ezért a menürendszer is a ma legkorszerűbbnek számító ún. *roló menü* (pulli down menü). Ennek az a lényege, hogy a jobb oldali egérgomb lenyomva tartásakor a Workbench felső keretén megjelennek az egyes menük megnevezései. Bármelyikre irányítjuk a nyilat, a menü, mint egy

roló, legördül, és az egyes választható funkciók láthatóvá válnak. A kiválasztás egyszerű, csak a megfelelő felíratra kell vinni a nyilat. (Természetesen a jobb oldali egérgombot még mindig lenyomva tartjuk, s csak választáskor engedjük el. Ezt a módszert alkalmaztuk, amikor a Workbench lemezről másolatot készítettünk, s a Duplicate funkciót választottuk ki a Workbench menüből.) Ha lenyomjuk a jobb oldali egérgombot, láthatjuk, hogy a Workbench három menüt tartalmaz, a Workbench-en kívül a Disk és a Special menüt. Vegyük sorra az egyes menüpontokat!

A Workbench menüben az *Open* ugyanazt a funkciót végzi, mint az ikonra való kétszeri rákattintás második kattintása. Egy ikont tehát úgy is aktiválhatunk, hogy egyszer rákattintunk, majd az Opent választjuk.

Ezzel ellentétes a *Close* (e. klóz), ami az ablak bal felső sarkában lévő pontra való rákattintással egyenértékű.

A *Rename-mel* (e. riném) az egyes ikonok alatti neveket változtathatjuk meg. Aktiválása előtt egyszer rá kell kattintani a kiválasztott ikonra; ennek színe ilyenkor inverzre vált (később látni fogjuk, hogy a beállítástól függően ez nem mindig inverz). A Rename menüpont kiválasztásakor egy szövegléc jelenik meg a képernyőn az ikon jelenlegi nevével, amit tetszés szerint átírhathunk. Ezután a Return lenyomásakor eltűnik a szövegléc, és az ikon alatt megjelenik az új név. Ily módon igen egyszerűen adhatunk új nevet a programoknak vagy magának a lemeznek is.

Az *info* igen hasznos funkció, amely az adott ikonról közöl információt. Ha ez a lemezikon, akkor magáról a lemezről is. Próbáljuk ki! Kattintsunk rá egyszer a Workbench lemez ikonjára, majd aktiváljuk a menüfunkciót! Ekkor egy új, fekete alapszínű ablak jelenik meg a képernyőn, Info release felirattal. Ezen láthatjuk a lemez nevét, az alatta sorrendben következő számok pedig

- a lemezen lévő összes blokk számát;
- a betöltött blokkok számát;
- a még üres blokkok számát;
- végül a blokkonkénti bájtok számát jelentik.

A *blokk* a lemezen lévő egységnyi hely megjelölése. Egy blokk tulajdonképpen 512 bájtot (azaz 0,5 kbájt) érne, de ez természetesen teljes egészében nem áll rendelkezésünkre, mert a lemez saját szervezéséhez is lefoglal bizonyos helyet, ezért blokkonként 488 bájt szabad hely van.

Ezenkívül a STATUS jelzésű kockában különféle feliratokat láthatunk, valószínűleg a Read Only feliratot is. Ez azt jelenti, hogy a lemez írásvédett, csak olvasni lehet róla. Ha az írásvédelmet kikapcsoljuk, és újból kérjük az Infot, a Read/Write szöveg jelenik meg, jelezve, hogy a lemezre írni és onnan olvasni is lehet. A DEFAULT TOOL, ill. a TOOL TYPES keretbe mi magunk is beírhatunk bizonyos utasításokat, majd a SAVE felíratra való rákattintással hagyhatjuk el az Info ablakot. Ha a változtatást figyelmen kívül hagyva (vagy változtatás nélkül) akarunk kilépni, akkor azt a QUIT-re való rákattintással, vagy a korábban már ismertetett bármelyik két ablaklezáró funkcióval megtehetjük.

Nézzünk meg más ikonokat is! A programikonokban igen hasznos adat a blokkokban, ill. bájtokban megadott programméret.

A Workbench menü utolsó funkciója a *Discard* (e. diszkard), ezt használjuk törlésre. Az egyes ikonokkal szimbolizált programokat, de akár több programot tartalmazó fiókokat is törölhetünk vele. A módszert már bizonyára kitalálta az Olvasó. Rákattintunk a kiválasztott ikonra, majd a nyíllal aktiváljuk a funkciót. Még annyi teendőnk lesz, hogy a megjelenő

biztonsági kérdésre ok to discarddal, vagy tévedés esetén forget it! (felejtse el-lel) válaszoljunk.

Bizonyára feltűnt, hogy bizonyos funkciók időnként halványabban jelennek meg a menüben. Ilyenkor ez a funkció nem is aktiválható. Ha pl. nem kattintunk rá egyetlen ikonra sem, akkor az összes funkció ilyen halványan jelenik meg.

A mindössze két funkcióból álló másik menü: a Disk. Az *Empty Trash* (e. emti trash) működése jóval bonyolultabb, mint maga a funkció. Ez is egyfajta törlési módszer, amihez a Trashcan feliratú ikont kell használni. Először nyissuk ki a Trashcan! Egy viszonylag kisméretű üres ablak jelenik meg. A törölni kívánt ikont (vagy ikonokat) a másolásnál megismert módszerrel vigyük be ebbe az üres ablakba. *Figyelem!* Itt nincs szükség a RAM lemezes trükkre, hiszen egyazon lemezen belül másolunk. Figyeljük meg, hogy ez abban is eltér a másolástól, hogy a szemetekosárba bevitt ikonok nem maradnak meg az eredeti helyükön, úgy tűnik, mintha már le lennének törölve. Törlés azonban nincs, csupán a törlés előtti átmeneti állapotba kerülnek. Ha meggondoltuk magunkat, innét még mindig van visszaút, visszamásolhatjuk a programokat oda, ahova akarjuk. Ha mégis a törlés mellett döntünk, akkor az Empty Trash funkció aktiválásával a szemétkosárban lévő összes program letörlődik a lemezeről.

A harmadik menü a Special. Az itt található funkciók érdekesek, de csak ritkán használjuk őket. Ha egy ablakot kinyitottunk, akkor a *Clean Up* (e. clín áp, takarítás) funkcióval az ablakban lévő ikonokat rendezhetjük.

A *Last Error* funkcióval az ablak felső léccén korábban már ki jelzett hibaüzenet ismételtethető meg.

A *Redraw-vz!* (e. ridróu) a valamilyen okból véletlenül megváltoztatott képernyő képe rajzoltatható vissza.

A *Snapshot* (e. sznepsot) talán a leghasznosabb a Special menü funkciói közül. Bármely ablakban véglegesen megváltoztathatjuk vele egy ikon helyét. Ez akkor hasznos, ha nő a rendetlenség az ablakon belül (másolás, törlés vagy újabb fájlok létrehozatala miatt). Ekkor a korábban ismertetett módon az ikonokat célszerűen eltologathatjuk, és ha azt akarjuk, hogy a következő betöltéskor is megmaradjon ez a rend, akkor a Snapshottal rögzíthetjük a helyüket. A Snapshot alkalmazása nélkül ugyanis az eltolt ikonok az ablak becsukása, majd újbóli kinyitása után a korábbi helyükön jelennének meg. A Snapshottal véglegesen rögzíthetjük az ablak méretének megváltoztatását is.

A *Version* funkció csak a Kickstart és a Workbench verziószámát jelzi ki.

A menük megismerése után tekintsük most át az 1. és 2. táblázatban a Workbench és az Extrás lemez legfontosabb programjait! (A vastag betűs nevek a fiókokat jelölik.)

A programok nagy része az 1.2-es és az 1.3-as változatban egyaránt megtalálható, csak a megoszlás egy kicsit más a Workbench és az Extrás lemez között. Néhány program a Workbench-en és az Extráson is megtalálható. Ezenkívül mindkét lemezen számos további fájl van. Ezek részben programok, részben a rendszer működéséhez szükséges rutinok, de nincsenek bennük ikonok, ezért nem is indíthatók el a Workbench-ről. A DOS ismertetésekor majd kitérünk rájuk.

Most röviden sorra vesszük a táblázatokban felsorolt programokat. A géphez adott gépkönyv részben ismerteti ezeket, ezért csak a fontosabbakkal foglalkozunk. A gépkönyvben nem ismertetett programok közül a legfontosabbak használatát is bemutatjuk.

A Workbench 1.2 lemez

Trashcan
Clock
Preferences

Demos

Utilities

Expansioa

Empty

Utilities

Dots
Spots
Boxes
Lines

Notepad
Calculator

DiskCopy
SetMap
CLI
Say
IconEd
GraphicDump
NoFastMem
Formát
InitPrinter
SlowMemLast

AWorkbench 1.3 lemez

Trashcan
Hinweise
Shell

Prefs**Utilities****System****Expansion****Empty**

Pointer
Preferences
CopyPrefs
Printer
Serial

Notepad
More
Clock
ClockPtr
Say
Calculator
Cmd
GraphicDump
PrintFiles
InstallPrinter

CLI
DiskCopy
Formát
FastMemFirst
SetMap
InitPrinter
NoFastMem
MergeMem
FixFonts

1. táblázat

Nem árt tudni, ha néhány fiók kinyitására nem találunk semmit, akkor egy üres ablakot kapunk. Ennek kétféle oka lehet. Az egyik, hogy bizonyos programok vagy önálló programként nem működő fájlok (pl. adatfájlok) nem kaptak ikont, hiszen a Workbench-ről nem is indíthatók. Ilyen fiók a Fonts, a Devs, az FD1.2 és az FD1.3. A másik ok, hogy valóban üres a fiók. Az *Empty* azért üres, mert adott esetben ide tudjuk betölteni saját programjainkat. De ezt a fiókot a már ismertetett módon bármelyik lemezünkre átmásolva, s a Rename funkcióval átnevezve, saját programjainkhoz illő fiókok rendszerét hozhatjuk létre. Az *Expansion* fiók azért üres, mert ide tudjuk elhelyezni a későbbiekben alkalmazott bővítések ikonját, pl. a Winchestert (e. vincseszter).

Lesznek azonban olyan programikonok is, amelyek aktiválásakor a lemez meghajtó ugyan tölteni kezd, de valójában nem történik semmi. Legjobb esetben valamilyen üzenetet kapunk. Ez azért van, mert a program vagy nem indítható a Workbench-ről, vagy a Workbench menürendszeréből hívható. Az alkotók az ikonnal nyilván a program lemezen való jelenlétét akarták jelezni. Ne várjunk tehát semmit a következő ikonoktól: DiskCopy, Formát, Setmap.

Az Extrás 1.2 lemez

AmigaBASIC	Terminál	
Calculator	Hinweise	Trashcan

BasicDemos (számtalan Basic programot tartalmaz)

Demos (ugyanaz, mint a Workbench lemezen)

Tools	FD1.2
MicroEmacs	
More	
Fed	
PrintFUes	
IconMerge	
PM	
FreeMap	

Az Extrás 1.3 lemez

AmigaBASIC
Hinweise
ReadMe
Trashcan

BasicDemos (számtalan Basic programot tartalmaz)

Tools	PCUta	F D U
Fed	PCCopy	
FreeMap	PCFormat	
PerfMon	ToPCCopy	
IconEd		
KeyToy2000		
Palette		
IconMerge		

2. táblázat

2.3. Notepad

A Workbench-en található programok közül talán az egyik legfontosabb, leghasznosabb a Notepad (e. nótped), amely viszonylag egyszerű, de nagyon jól kezelhető szövegszerkesztő. Főképp a kezdő, programokkal még nem rendelkező felhasználó számára jelent igazi kincset. Ha már sikerült szert tennünk professzionális szövegszerkesztőre, akkor nemigen fogjuk használni, bár egyszerűbb feladatokra talán még a profik is célszerűen alkalmazhatják, hiszen nem mindig van szükség bonyolult és körülményes programra.

A Notepad egyszerű levelek megírásán kívül különféle feljegyzések elkészítésére is alkalmas. Akinek nincs nyomtatója, az nyilván levelet sem fog írni, saját magának azonban készíthet feljegyzéseket, vagy a saját Basic programjaihoz a lemezen mindjárt a program mellé csatolhatja a használati utasítást. S ha nyomtató nélkül levelet nem is tudunk írni, azért a Notepad segítségével tudunk levelezni, ha a partnerünknek is van Amigája, hiszen a lemeztől fog tudni olvasni.

Annak, aki most találkozik először szövegszerkesztővel, röviden elmondjuk, hogy ez olyan program, amellyel könnyen írhatunk rendezett szöveget (a hangsúly a rendezetten van), emellett különféle méretű, ül. típusú betűket alkalmazhatunk, és kiemelhetünk szavakat, aláhúzhatunk részeket stb.

Ismerkedjünk meg a Notepaddel! A szokásos módon, a megfelelő ikonra való kettős rákattintással indíthatunk. Egy viszonylag kis ablak nyílik a képernyő tetején, amit célszerű megnagyobbítani. A kurzor egy függőleges, kis fekete vonal. Mindjárt el is kezdhetjük begépelni a kívánt szöveget. Bárki leírhatja pl. az életrajzát. S íme itt az első nehézség!

Ha nem németül vagy angolul írunk, azonnal hiányozni fognak bizonyos ékezetes magánhangzók, hiszen csak a német nyelvben szokásos magánhangzók vannak meg. Ez a továbbiakban bármely más szövegszerkesztővel kapcsolatban is felmerül, amelyiket nem kimondottan a magyar nyelvre készítették. Az Amiga számos nemzeti nyelv egyedi karaktereit ismeri, s ezeket (pl. a svédet) viszonylag könnyen beállíthatjuk (l. a DOS-szal foglalkozó 4. fejezetben). A magyar nyelvre is lesz megoldás.

Életrajzunk megírásakor most ideiglenesen tekintsünk el a hiányzó ékezetektől. Ha elkészült, akkor a Notepad Project menüjéből válasszuk ki a *Savé As* (e. szév ez) funkciót. Azonnal megjelenik a kérdező, és felszólít, hogy adjunk címet, ül. nevet a szövegnek. Ha adtunk nevet, akkor az OK feliratra való rákattintással vagy egyszerűen a Return billentyű lenyomásával a szöveg letöltődik. Ha kicsit korai volt a letöltési szándék, akkor egyszerűen kattintsunk rá a Cancelre. Itt érdemes egy pillanatra megállni és megnézni az Amiga letöltési logikáját.

A legtöbb program ugyanazt a letöltési módot használja, mint a Notepad. A Project menüben kétféle Save-vel dolgozhatunk, a Savé As-en kívül a Save-vel is. Ha már van neve a szövegünknek (programunknak), és csak módosítottunk rajta, akkor a Save-vel felülírhatjuk az eredetit. Ilyenkor azonos néven csak az új változat marad meg. A Savé As-t akkor használhatjuk, ha az első letöltésnél vagyunk és nevet kell adnunk, vagy ha új változatot készítettünk, de nem akarjuk, hogy az eredeti elvessen. Ilyenkor ezzel a funkcióval új néven tudjuk elmenteni az új változatot.

Ha tovább vizsgáljuk a Project menüt, észrevehetjük a már ismert *Open* funkciót, ez indítja el az új szöveg betöltését. Ha az előző szöveget még nem töltöttük le, akkor az ilyenkor elvész. Ugyancsak elvész a szövegünk, ha a *New* funkciót indítjuk. Ilyenkor nem a már meglévő szöveget tölti be a gép, hanem kitörli a Notepad ablakot: indulhat az új szöveg szerkesztése.

A szövegszerkesztők fontos funkciója, a nyomtatás. Anélkül, hogy a nyomtatással itt részletesebben foglalkoznánk, megemlítjük, hogy a Notepad *Print* funkciójával a szöveget különböző arányban kicsinyíthetjük. Erre valók a Print funkció almenüpontjai: az *Autó size*, a *Small*, a *Médiüm* és a *Large*. Alapesetben az *Autó size*-t érdemes választani. Előbb azonban a *Print As* funkcióval be kell állítanunk, hogy szöveget (*Draft*) vagy grafikát — azaz grafikusán (*Graphic*) — akarunk-e nyomtatni. A megfelelő almenüfunkció kiválasztását a kis kampó jelzi a *Graphic* vagy a *Draft* felirat mellett. A nyomtatáshoz ne felejtsük el bekapcsolni és csatlakoztatni a nyomtatót, mert *Printer trouble* hibajelzést kapunk.

A szövegszerkesztőből vagy a Project menü *Quit* (ejtsd:kvit) funkciójával vagy az ablak bal felső sarkára való rákattintással tudunk kilépni.

Érdekes kísérletezésre ad módot a *Fonts* menü. Itt hétféle betűtípus közül választhatunk. Némelyik betűtípus több méretben is rendelkezésre áll. Alapesetben az Amiga mindig a topaz betűtípust használja. A *Style* menüvel azt határozhatjuk meg, hogy szövegünk dőlten (*Italic*), vastagon (**Bold**) vagy aláhúzva (Underline) jelenjen meg. A normál formát a *Plain*nel (sima) állíthatjuk vissza.

A *Formát* menüvel változtathatunk a háttér (*Paper Color*), ül. a szöveg színén (*Pen Color*). A *Word wrap* (e. vörd vrep) jellegzetes szövegszerkesztői funkció. Bekapcsolásakor ha egy adott

sorban az utolsó szó nem fér be a megadott sorhosszba, akkor az egész szó átkerül a következő sorba. Azért, hogy a sorok vége ne legyen cikcakkos, a gép üres karakterekkel tölti fel a sort, és így mindig egyforma hosszúak lesznek a sorok. Ezt a funkciót úgy tudjuk kikapcsolni, hogy a szokott módon kiválasztjuk a funkciót. Ilyenkor a Word wrap felirat mellől eltűnik a kis kampó. Újabb kiválasztáskor ismét aktiválódik.

A *Global font* funkció bekapcsolásakor a szöveg végig azonos betűtípussal (fonttal) íródik. Kikapcsolásakor akár szavanként, akár betűnként változtathatjuk a betűtípust. A *Remove fonts* és a *Remove style* funkcióval visszaállíthatjuk az alapbetűtípust és az alapbetűformát, azaz a simatopazt.

Az Edit (szerkesztés) menü két részből áll. Az első öt funkciót itt nem ismertetjük, mert a Basicről szóló 3. fejezetben ez sokkal érthetőbb lesz. A további négy funkcióval a szövegszerkesztőkre jellemző érdekes műveletet tudunk elvégezni. Ha pl. egy szövegben az és-eket vagy-okra akarjuk kicserélni, akkor a *Find* funkcióval kikereshetjük a szövegben előforduló és-eket, majd a *Replace* (helyettesít) funkcióval elvégezzük a cserét.

Tüzetesebben megvizsgálva a Notepad ablakot, új szimbólumokat fedezhetünk fel. A bal alsó sarokban egy számfülre emlékeztető jel van. Erre rákattintva tudunk lapozni a szövegben. Ha lapozunk, akkor a jobb felső sarokban lévő szám is változik, jelezvén az új oldalszámot. Az oldalszám alatt egy felfelé mutató nyíl, vele szemben pedig egy ellentétes irányú nyíl van. Az adott oldalon belül ezekkel tudjuk soronként felfelé, ül. lefelé eltolni a szöveget.

Ha a szöveget készítünk a Notepaddal, akkor látni fogjuk, hogy letöltés után a szöveg is kap egy különleges ikont, egy teleírt papírlapot. A szöveg elolvasásához nem kell betölteni a Notepad programot, elegendő kétszer rákattintani a szövegikonra, és a szöveggel együtt a Notepad is automatikusan betöltődik. Ezzel ízelítőt kaptunk az Amiga fantasztikus képességéről, a multitaskingról.

A *multitasking* azt jelenti, hogy a számítógép egyszerre több funkciót is képes ellátni, tehát párhuzamosan több program is futhat egymás mellett. Jól szemlélteti ezt a Demos fiókban lévő rákattintással egymás után indítsuk el a programokat.

2.4. Preferences

A *Preferences* a Workbench legfontosabb beállító funkciója. Ha a kettős rákattintással elindítjuk a programot, akkor egy új, a képernyőt teljesen kitöltő ablakot kapunk. A gépkönyvben minden funkciót leírtak, ezért itt csak a legfontosabbakra térünk ki.

A Preferences ablak alsó részét a színbeállító foglalja el. Már bizonyára mindenki észrevette, hogy az Amiga a Workbench lemez betöltésekor mindössze négy színt használ: feketét, fehéret, kéket és sárgát. Természetesen a gép ennél sokkal többet tud, több színt ismer, és a képernyőn egyszerre többet tud megjeleníteni. Aid egy kicsit foglalkozott tv- vagy videotechnikával, az jól tudja, hogy a képernyőn megjelenő bármely színárnyalat a három alapszín valamilyen arányú keveréke. A három alapszín: a piros (*R*), a zöld (*G*) és a kék (*B*). A zárójelben a színek angol nevének kezdőbetűi vannak. A Preferences ablak alsó részében is megtalálhatjuk az RGB színek beállítók.

Ha az egérrel az egyes alapszín-beállítóknak lévő fehér pontocskákat tologatni kezdjük, akkor megfigyelhetjük, hogy a Reset Colors felirat mellett látható négy szín közül az alapbeállításkor vastag sárgával bekeretezett szín változik. Hogy melyiket változtatjuk a négy szín közül, mi döntjük el, amikor az egérrel máshova visszük a vastag keretet. Akinek tehát a négy beállított Workbench szín nem tetszik, annak érdemes kísérleteznie. Különösen akkor

ajánlatos megváltoztatni a színeket, ha tv-n keresztül használjuk az Amigát, és szöveget vagy Basic programot szeretnénk írni. Ugyanis az alapbeállítás szövege tévén nehezen olvasható.

Próbálkozzunk meg a következő beállítással. A kék színt változtassuk inkább a barna felé hajló halványárgává. A fehérét és a feketét meghagyhatjuk, a sárgát pedig változtassuk lilává. Ezzel a beállítással sokkal jobban olvasható a szöveg. Ha nagyon összekevertük a színeket, és újból az eredeti gyári beállítást akarjuk visszaállítani, akkor kattintsunk rá a *Reset Colors* felíratra!

Növelhetjük a szöveg olvashatóságát, ha a bal oldalon, a *Text* felirat melletti számok közül a 60-at választjuk (a sárga szín jelzi a kiválasztottat). Igaz, ekkor csak 60 karakter fér majd egy sorba, de 8-as topaz betűtípus helyett a gép a nagyobb, 9-es topazzal dolgozik. A monitort használók maradjanak a 80-as beállításnál!

Ha a *Preferences* ablak közepén lévő derékszöveget az egérrel mozgatjuk, akkor megváltoztathatjuk a képernyőn lévő kép pozícióját. Jól beállított monitor vagy tv esetén erre nincs szükség, csak akkor nyúljunk hozzá, ha valami kilóg a képernyőről, s emiatt nem olvasható.

A jobb oldalon két függőleges vonal van, az egér képével. A jobb oldalival a kettős rákattintás közötti szünetet állíthatjuk be, azaz hogy a gép mit érzékeljen kettős rákattintásnak, és mit két egymás utáni külön kattintásnak. Ehhez ne nyúljunk. A bal oldali, az 1,2,4 számokat tartalmazó vonal az egér sebességét állítja. Valószínűleg az 1-es állásban kapjuk a legjobb eredményt.

Ha a változtatásokat ki akarjuk próbálni, és vissza akarunk térni a *Workbench* képernyőhöz, akkor kattintsunk rá a *Use* felíratra. Ha változtatás nélkül akarunk visszamenni, akkor a *Cancel* felíratra kattintsunk rá. Ha pedig a változtatásokat nemcsak kipróbálni akarjuk, hanem szeretnénk, ha a számítógép kikapcsolása után a legközelebbi *Workbench* betöltéskor a *Workbench* már az új paraméterekkel jelenjen meg, akkor a *Save* felíratra kattintsunk rá.

Ha átállítottuk a paramétereket, de valamiért mégsem tetszik, akkor a *Cancel* helyett két másik megoldás is van. A *Last Saved* segítségével a legutoljára kimentett beállítást hozhatjuk vissza, ha volt ilyen. Ha előzőleg nem mentettünk, akkor ez az alapbeállítással egyezik meg. A gyári alapbeállítást a *Reset Ali* aktiválásával érhetjük el.

A *Workbench Interlace On* helyzetbe állításával a képernyő vízszintes felbontása megduplázódik, azaz az adott képernyőre kétszer annyi sor fér el. Ez az Amiga *interlace* (e. interlész) üzemmódja. Csak azok próbálják ki, akiknek különlegesen jó minőségű — erre az üzemmódra alkalmas — monitoruk van, ugyanis más esetben a képernyő erősen vibrál. A kézikönyv azt ajánlja, hogy kísérletezzünk más színek beállításával, hogy a vibrálás megszűnjön. Ez azonban nem vezet eredményre. Fontos tudni, hogy ezt a változtatást a *Use* alkalmazásával nem tudjuk egyszerűen kipróbálni. Ha *Interlace* üzemmódba álltunk, akkor a *Save*-vel le kell tölteni, majd reszetálás után újból be kell tölteni a *Workbench*-et. A visszaállítás is hasonlóan körülményes.

A *Last Saved* felirat alatti megfelelő számra, ül. a fehér nyilakra való rákattintással a dátumot s az időt állíthatjuk. Ennek csak különleges esetekben, ill. akkor van jelentősége, ha a gépnek van belső órája. Ellenkező esetben ugyanis hiába állítunk a dátumon vagy az időn, a következő betöltéskor újból a gyárilag beállított érték jelenik meg. A DOS-szal foglalkozó 4. fejezetben azonban az idő beállítására még visszatérünk.

A *Change Serial* feliratú funkció állításának az elektronikus adatárvitelkor van jelentősége, de a megfelelő eszközökhöz ma már általában vezérlőszoftvert is adnak, így ez a beállítás itt most nem túl fontos.

Az *Edit Pointer* a nyíl formáját és színét változtathatjuk.

Meg kell említenünk a *Change Printer* funkciót, amely a nyomtatáskor fontos. A funkció aktiválásával megint a teljes képernyőt betöltő ablakot kapjuk, amelyben a nyomtatáshoz szükséges paramétereket tudjuk beállítani. Itt jelentős továbblépést jelentett az 1.3-as változat az 1.2-höz képest. A beállításokat részletesen a Nyomtatás című fejezetben ismertetjük.

Szorosan idetartozik a csak az 1.3-as változat *CopyPrefs* programja, mellyel az adott lemezen lévő jól bevált Preferences beállításokat átvihetjük a másik lemezre. Nem kell tehát az ottani Preferences behívásával és a beállítással próbálkoznunk. Főképp akkor hasznos a CopyPrefs, ha a másik lemezünkön nincs is Preferences, tehát nem is tudnánk beállítani. Ilyen lemez lehet pl. az Amiga-Basic munkalemez, ahol felmerülhet az igény a gyári alapszín-beállítás megváltoztatására. Akiknek nincs CopyPrefs programjuk, azok a Tippek és trükkök c. fejezetben találhatnak megoldást.

2.5. Palette

Ezzel a programmal a Preferences nélkül is azonnal elérhetjük a Preferences szíhbeállító funkcióját.

2.6. PerfMon

A PerfMon figyelő, monitorozó program, amely az Amiga pillanatnyi állapotáról tájékoztat.

2.7. Fed

A Fed, a Font Editor program különböző beállítási lehetőségeket kínál. A Notepadnél már esett szó arról, hogy az Amigának különféle méretű és formájú betűtípusai vannak. A Commodore jóvoltából jó néhány rajta van a Workbench lemezen. Akit ez a választék nem elégíti ki, az a Feddel újakat hozhat létre.

A program működése viszonylag egyszerű. Az ablak bal oldalán szürke alapon lila hálót látunk, ez a szerkesztési segédterület. A LoRes kapcsolóval tudjuk változtatni a vízintes felbontást, a Grid kapcsolóval lehet eltüntetni, ill. visszaállítani a segédhálót. A Zoom (e. zum) tolókar nagyítóként működik.

Indítsuk el a program menürendszerét a Project menü Open funkciójával! Ekkor kiválaszthatjuk, hogy melyik fonttípust akarjuk változtatni, ill. melyiket akarjuk új betűnk kialakításához alapul választani.

A betöltés után a jobb oldali 16 kék téglalapban megjelennek az adott betűtípus szerinti ábécé írásjelei. A jobb szélső fehér tolókéval áttekinthetjük az egész ábécét. A kék téglalap alatti zöld mezőben láthatjuk a betűformát, valamint a tizenhatos számrendszerbeli kódját. Ez utóbbira most nem lesz szükségünk. Bármelyik betűre rákattintva, az megjelenik a rácsos keretben. Ha a rácson belül bármelyik kis lapra rákattintunk, az az ellenkezőjére vált. A programmal akár kínai írásjeleket is alkothatunk.

Az Edit menüvel a betűket dőltté, vastaggá vagy aláhúzottá változtathatjuk, de az Attributes menü funkcióival is kísérletezhetünk. Két fogalmat meg kell magyarázni, ezekkel az Attributes

menü *Font type* funkciójában találkozhatunk. A **FIXWIDTH** olyan betűt jelent, amely a rajzolatától függetlenül azonos szélességű helyet foglal el. Így pl. (vastagságkülönbségük ellenére) az / és az *m* betű is azonos szélességű helyet foglal majd el a kinyomtatáskor. A **PROPORTIONAL** viszont azt jelenti, hogy a betű helyfoglalása annak méretével arányos.

Ha elkészültünk az új ábécével, akkor a **Save As** funkcióval letölthetjük, meghagyva a régi nevet vagy új nevet adva neki (ilyenkor nem írjuk át az eredetit).

2.8. NoFastMem

Általában ritkán használatos, de nagyon fontos program. Ha memóriabővítő Amigánk vagy 2000-es gépünk van, ahol 1 Mb-ot vagy annál nagyobb a RAM memória, szükség lehet az 512 kb-ot feletti rész — ezt hívjuk Fast Memory-nak — kikapcsolására. Vannak ugyanis olyan szoftverek, amelyek bővített memóriával nem működnek. Akkor adódhat probléma, ha olyan programunk van, amit nem a Workbench-ről indítunk, hanem a programlemez közvetlen betétele után a program automatikusan betöltődik. Ilyenkor mégis előbb a Workbench-et kell betenni, majd a NoFastMem aktiválása után a CLI-t, kell aktiválni, s a kérdéses szoftvert onnantól kell elindítani. Hogy hogyan, azt a 4. fejezetben írjuk le.

Még egy jellemzőre kell felhívni a figyelmet a NoFastMem használatával kapcsolatban. A memóriabővítés kikapcsolásakor a belső óra is kikapcsolódik, ezért ilyen beavatkozás után újból be kell állítani.

Ezzel végére értünk a Workbench fejezetnek, bár a Workbench, ül. az Extrás lemezen lévő jó néhány programról nem szóltunk. Ezek közül néhányat a későbbi fejezetekben bemutatunk.

3. AZ AMIGA-BASIC

Az előző fejezetekben megismerkedtünk az Amigával, megpróbálkozhatunk magunk is programokat, ill. egyszerűbb rutinokat írni: passzív felhasználókból aktív felhasználókká válva. Erre valók a programnyelvek, amelyek közül a Basic a legnépszerűbb, mert a legegyszerűbb és a legkönnyebben megtanulható. Ahány számítógéprendszer, annyi fajtája létezik, ezek azonban valójában csak apróságokban, ill. a felhasználható parancsok számában térnek el. Nyugodtan állíthatjuk, hogy aki egy Basic nyelvjárást elsajátított, az könnyedén át tud térni egy másikra is.

A könyv Amiga-Basic fejezete egy kicsit Basic tanfolyamként is felfogható, bár némi Basic előismeret hasznos lehet. A Basicben jártas Olvasó számára sem lesz azonban unalmas, mert — esetenként példákkal is alátámasztva — az Amiga nyelvjárásával ismerkedhet meg, ami eltérhet a korábban ismert Basic-től. A kis gépekljéz képest alapvető változás, hogy míg pl. a Commodore 64 esetében a Basicet beépítették a gép ROM-jába (és így már a bekapcsolás pillanatában rendelkezésünkre állt), az Amigában a Basicet szoftverként kell felfogni, amely ugyanolyan program, mint egy adatbázis-kezelő vagy grafikai program. A Basicet az Amiga esetében lemezepről kell betölteni, és a Basic programok is csak akkor futnak, ha a Basicet már betöltöttük. (Az Amiga-Basic 80 kb-ajtnyi helyet foglal el a memóriában.)

Hogyan tölthetjük be a Basicet? A számítógép bekapcsolása után a szokott módon tegyük be először a Workbench lemezt, majd a betöltés befejezése után helyezük be (két lemezmeghajtó esetén a külsőbe) az Extrás lemezt, majd nyissuk ki a meghajtót. Kattintsunk rá kétszer az AmigaBASIC ikonra.

A betöltés után egy kettéosztott képernyőt kapunk. Láthatjuk, hogy két egymásra helyezett ablak van rajta. Az egész képernyőt betöltő Basic, valamint az erre helyezett (a képernyő jobb felét betöltő) List ablak. A bal oldalon megjelenő szöveg valószínűleg nem okoz gondot. A két számadat fontos információt nyújt arról, hogy a Basic betöltése után mennyi szabad memóriahely maradt a gépben, valamint hány bájt áll rendelkezésünkre a Basic programozáshoz. Az előbbi szám változó, s annak függvénye, hogy mekkora a gép alapkapaaitása (ill. hány ablakot hagytunk nyitva a Basic betöltése előtt), a második szám viszont mindig 25 000. Ezen csak a Basic programon belül tudunk állítani, ha ennél nagyobb programot akarunk írni. A felső határt a számítógép kapaaaitása szabja meg.

A betöltés így kissé körülményes. Hogyan lehet egyszerűbben csinálni? Aki rendszeresen szeretne Basicben programozni, annak érdemes egy külön Basic lemezt létrehozni, ahova a Basic programokat lehet gyűjteni, s ez egyben munkalemezként is használható, nehogy az Extrás lemezen a Basic megsérüljön.

Formázzunk meg egy üres lemezt, s másoljuk rá az AmigaBASIC programot! Jegyezzük meg, hogy csak akkor tudjuk futtatni a Basicben írt programjainkat, ha az AmigaBASIC rajta van a lemezen, a futtatáshoz azonban nem kell külön betölteni.

A kezdők sokszor elkövetik azt a hibát, hogy megírt Basic programjukat átmásolják a Basic lemezepről egy másik lemezre, és meglepődve tapasztalják, hogy indításkor a program helyett hibaüzenetet kapnak. (A gép ugyanis nem találja a Basicet.)

Mivel a Basichez nincs feltétlenül szükségünk a Workbench-re, jó lenne megspórolni a Workbench betöltését is. Ahhoz azonban, hogy a bekapcsolás után azonnal a Basic lemezt tehesük be, és az automatikusan betöltődjön, néhány műveletet el kell végeznünk a lemezen (1. a Basic lemez létrehozása című 4.10. alfejezetet). Aki tehát rendszeresen Basicben akar

programozni (de nem a Workbench-en keresztül bonyolult betöltéssel), az lapozzon egy kicsit előre, mielőtt a Basic fejezetet folytatná.

Térjünk tehát vissza a Basic, ill. List ablakhoz. A tulajdonképpeni programozást — a programlista elkészítését — a List ablakban kell végezni. Ezt az ablakot az ismert módon akár a teljes képernyőméretre is megnövelhetjük. A Basic ablak a már futtatott programot jeleníti meg. Ilyenkor ez az ablak automatikusan előtérbe kerül, még akkor is, ha a List ablak teljesen eltakarta a programozáskor.

A Basic program utasításokból és a hozzájuk tartozó paramétereiből áll, amelyek írott formában sorokba vannak rendezve. A Commodore 64-en egy sorba 80 betű (pontosabban karakter) fér el, az Amigán pedig 256. Az Amiga-Basic érdekessége, hogy az egyes sorok nincsenek megszámozva. Természetesen megszámozhatjuk a sorokat, a gép azonban a sorszámok növekvő sorrendjét figyelmen kívül hagyja, és a sorokat leírásuk sorrendje szerint hajtja végre. Így összevissza is sorszámozhatunk, ami pl. a Commodore 64 esetében nem lehetséges, ugyanis a gép a sorszám szerint rendezi sorrendbe a Basic sorokat. A következő programot:

```
20 PRINT TS (1)
10 TS="HELLO" (2)
30 PRINT TS (3)
```

pl. az Amiga az (1),(2),(3) sorrend szerint hajtja végre, a Commodore 64 pedig a (2),(1),(3) sorrend szerint.

Hogyan tudjuk az Amiga-Basickezelővel megcímezni az egyes sorokat, pl. egy GOTO utasítással? A sorszámozás ugyanis erre is használható más Basic esetében. Az Amigán címkéket (label) alkalmazunk, s ez lehet egy sorszám is. Helyesebb számot mondani, ugyanis a gép nem rendezi a számok szerint sorba a sorokat. A címke azonban lehet tetszőleges szöveg is (a végén kettősponttal), pl. start:, hurok: stb.

A Basic utasításokat mindig nagybetűvel kell írni. Az Amiga-Basicben ezt megteszi helyettünk a gép, mert a sor lezárásakor, az Enter lenyomásával egy időben, a Basic utasítások átváltanak nagybetűre. Egyben ellenőriz is ez a módszer, mert csak a helyesen beírt utasításokat ismeri fel a gép.

Az Amiga-Basic erőssége a kitűnő szerkesztési képesség, ami részben a jól felépített menürendszerén keresztül valósul meg. A menürendszer négy rolómenüből áll. A Project menü a Notepadnál már megismert funkciókat tartalmazza, amelyek az indítáért, a letöltésért stb. felelősek.

3.1. Az Edit menü

A tulajdonképpeni szerkesztésért az Edit menü felel. Az érthetőség kedvéért az Extrás lemez BasicDemos fiókjából töltjük be a CLogo programot, s ezen próbáljuk ki a három Edit funkciót.

A Project menü *Open* funkciójával töltjük be a programot, és ne a kettős rákattintással. (Ez utóbbit már jól ismerjük, később majd kipróbálhatjuk.) A Name of program to load keretébe nem elég a CLogót beírni. Bonyolultabb a művelet, mint a Workbench-nél. A pontos szintaxis (syntax) a következő: BasicDemos/CLogo. A CLogo program ugyanis a BasicDemos fiókban van az Extrás lemezen.

A betöltést a Commodore 64 jól ismert LOAD parancsával is elvégezhetjük. Fontos, hogy ilyenkor a kurzor a Basic ablakban legyen, mert közvetlen utasítást csak itt tudunk adni. Ez a

szintaxis a következő: LOAD "BasicDemos/ciogo". Az Opennel ellentétben ilyenkor csak betöltődik a program, de nem indul el. Habár az előző szintaxist a nagy- és kisbetűk figyelembevételével (az eredetinek megfelelően) írtuk, ennek nincs jelentősége. Ha jól csináltuk, néhány másodperc múlva a List ablakban megjelenik a programlista.

Most próbáljuk ki a kivágást! Az első programsor "Commodore-Logo" részéből töröljük ki a Commodore szót! Vigyük a nyilat a C betűre, nyomjuk le a bal oldali egérgombot, s miközben nyomva tartjuk, a nyilat húzzuk a Commodore szó végéig. Látjuk, hogy a kiválasztott szó olyan, mintha kiemelő filctollal húztuk volna át, alapszín-beállítás esetén narancsszínű. Elengedve az egér gombját, válasszuk ki az Edit menüből a *Cut* funkciót! A Commodore szó eltűnik. Ha azonban nem szeretnénk ezt a részt végképp törölni, akkor bárhová pozicionálhatjuk a List ablakban a kurzort, pl. közvetlenül a Logo szó mögé. Válasszuk az Edit menüből a *Paste* funkciót! Ekkor közvetlenül a Logo mögött megjelenik a Commodore szó.

A következő példában kiemelés módszerrel jelöljük meg a "-Logo" szövegrészt, de most az Edit menüből a *Copy* (másolás) funkciót válasszuk ki! Ekkor a kiemelés után — mintha mi sem történt volna — a -Logo a helyén marad. Ha azonban a kurzort a Commodore szó mögé helyezzük, és aktiváljuk a *Paste* funkciót, akkor a kérdéses helyen a "-LogoCommodore-Logo" szöveget olvashatjuk. Tehát a *Cut* és a *Copy* is másolást jelent, de a *Copy* esetében az eredeti rész is megmarad a helyén. A két funkciónak akkor van jelentősége, amikor nagyobb részleteket, több sort kell másolni. A másolandó résznek ugyanis nincs korlátja.

A másolás működésének pontos ismeretéhez tudnunk kell, hogy a *Paste* funkcióval mindig csak a legutoljára megjelölt részt tudjuk az adott helyen (kurzorpozíció) megjeleníteni, az összes előző elveszik. Ugyanakkor a másolandó részt — akár több példányban is — több helyre is másolhatjuk. Mindaddig, míg nem aktiváljuk újra a *Cut*ot vagy a *Copy*t, az utoljára megjelölt rész a *Paste* segítségével másolható.

A bonyolultabb részeket is könnyedén törölhetjük a *Cut* utasítással. Ha csak törölni akarunk, és a kitörölt részt másutt nem akarjuk megjeleníteni, akkor sokkal kényelmesebb törlési módot is alkalmazhatunk. Ugyanúgy, mint a *Cut*nál, jelöljük meg a törlendő részt, és nyomjuk le a *Backspace* billentyűt (jobbra fent, a bal felé mutató nyilat).

A szerkesztés fontos része a programlistában való mozgás. Az Amiga-Basic erre sokféle lehetőséget kínál. A legegyszerűbb a kurzormozgató billentyűkkel végzett mozgás. Ennél sokkal gyorsabb és közvetlenebb a kurzor egérrel való pozicionálása.

Az ablak általában csak a program kiragadott részét mutatja, még ha a List ablakot a maximumra növeljük is. Az ablakoldalak között a *Shift* billentyűvel, ül. a megfelelő kurzormozgató billentyűvel lehet lapozni. Lapozás nélkül egyből a lista végére, ill. elejére tudunk ugrani az *Alt*tal és a megfelelő kurzormozgató billentyűvel. Hosszú Basic sorokban gyorsan a sor elejére vagy végére ugorhatunk az *Alt*tal és a kurzormozgató billentyűkkel.

Van egy különleges módja is az oldalak pörgetésének (*scroll*). Nyomjuk le a bal oldali egérgombot, és anélkül, hogy elengednénk, a kurzort húzzuk le a List ablak aljág! Ott megállva, amíg a gombot nyomva tartjuk, a lista görög lefelé. Természetesen ez fölfelé is hasonló módon megy végbe.

Ha nagy programlistában akarunk lapozgatni, akkor — főképpen az 1.2-es változatban — a *Shift* billentyűs módszer nagyon lassú, a *Tippek és trükkök* c. fejezetben bemutatandó megoldással azonban jelentősen meggyorsítható.

3.2. A Runmenü

Az egyik legfontosabb menü, vele tudjuk elindítani a megírt Basic programot. Az indítást természetesen a Basic programokra jellemző hagyományos módon is megtehetjük (pl. a Commodore 64 jól ismert **RUN** parancsával). Ilyenkor a Basic ablakba kell begépnünk a program nevét, a következő módon:

```
RUN "program"
```

Emellett az Amigás ikonról való indítást is alkalmazhatjuk, a Run menüből pedig a *Start* funkció aktiválásával indíthatunk.

A programot a *Stop* funkcióval lehet leállítani. Az ugyanebben a menüben található *Continue* a *Suspend* funkcióval felfüggesztett programot továbbfolytatja. A *Trace On* funkció pedig (a Commodore 64-es Simon's Basicjéből jól ismert) utasításonkénti programkövetést tesz lehetővé. Ez a program tesztelésekor jelentős. A Trace On még különlegesebb megoldása a *Step*.

3.3. A Window menü

Az utolsó fontos menü a Window (ablak) menü, ennek *Show List* funkciója megfelel a LIST Basic parancsnak; hatására a program listája megjelenik a List ablakban. A program leállításakor a program listája azonnal megjelenik a List ablakban, sokszor a program megjelenítésére használt Basic ablak eltakarja a List ablakot. A LIST parancs ilyenkor tulajdonképpen előrehozza a List ablakot. A Window Output viszont a Basic ablakot helyezi előre. Így adott esetben ide-oda váltogathatunk a Basic program képe és a program listája között.

A Basic programozással most ismerkedők számára megjegyezzük, hogy a Basic ismer ún. közvetlen parancsbevitelt is. Ilyen volt a Basic ablakban a RUN parancs. Ha ebbe az ablakba egy vagy több parancsot kettősponttal elválasztva beírunk, akkor az Enter lenyomása után a gép azonnal végrehajtja a parancso(ka)t. A List ablakba bevitt parancsok azonban nem hajtódnak végre azonnal a bevitel után, hanem csak külön utasításra.

3.4. Alaputasítások

Ismerkedjünk meg néhány alapvető Basic utasítással!

A PRENT "Szöveg" az idézőjelek közötti szöveget jeleníti meg az éppen következő sorban a képernyőn. A program elején vagy a képernyőt törölő **CLS** parancs hatására az éppen következő sor értelemszerűen a legelső sort jelenti. A **PRINT** Művelet a művelet eredményét adja a képernyőn, a PRINT 125+25 hatására 150 jelenik meg.

Több PRINT parancsot is írhatunk egymás után. Minden új PRINT eredménye új sorba kerül. Az egyes parancsokat a Basicben vagy új sorba írjuk, vagy kettősponttal választjuk el egymástól:

```
PRINT "Commodore":PRINT "AMIGA".
```

Ilyenkor a képernyőn a két szó egymás alá kerül. Ha azt akarjuk, hogy a szavak egymás mellé kerüljenek, akkor a kettőspont elé egy pontosvesszőt is tegyünk. Próbáljuk ki a következőt!

```
PRINT "A Commodore";:PRINT "AMIGA";  
PRINT "üdvözlí a kedves";:PRINT "Olvasót!"
```


A Commodore 64 Basicben programozók jól emlékeznek, hogy milyen körülményes volt a PRINT paranccsal a képernyő tetszőleges helyére szöveget elhelyezni, netán egy korábban használt sorba visszaírni. Erre való az Amiga LOCATE sor,oszlop parancsa. A sor és oszlop számának megadásával a megjelenítés helyére utalunk. Ez minden olyan parancs előtt alkalmazható, amely a képernyőn valamilyen megjelenítésért felelős.

Ha programozás közben valamilyen emlékeztetőt akarunk beírni a programsorok közé — természetesen úgy, hogy azt a program ne hajtsa végre, és a programot ne is zavarja meg —, akkor a szöveg elé tegyük a REM utasítást/Ehelyett azonban használhatjuk a ' jelet is.

A WRITE utasítás megegyezik a PRINT utasítással, mindössze annyiban térnek el egymástól, hogy nemcsak az idézőjelbe tett szöveg, hanem az idézőjelek is megjelennek a képernyőn. A másik igen fontos eltérés, hogy míg a PRINT utasítás a számokat az előjelet is figyelembe véve helyezi el a képernyőn — tehát a pozitív számok megjelenítésekor egy karakternyi helyet hagy a szám előtt —, addig a WRITE utasítás csak a negatív előjellel törődik. A PRINT "HELLO"; 100 (1) és a WRITE "HELLO"; 100 (2) között azonnal látjuk az eltérést:

```
HELLO 100 (1)
"HELLO",100 (2)
```

A helykihasználás szempontjából a WRITE sokszor igen hasznos.

Az Amigán a színeknek különösen fontos szerepük van. Az eddigiekben a gyárilag beállított négy alapszín használtuk, a képernyőn azonban egyszerre 32 színt tudunk egymás mellett megjeleníteni. Később látni fogjuk, hogy egy kis programozási trükkkel ennél jóval többet is. A színeket mi magunk keverhetjük ki, a három alapszín (a piros, a zöld és a kék) arányának megadásával. Ezt a PALETTE szám.piros.zöld.kék parancs teszi lehetővé, ahol a 0 és 31 közé eső számok az egyes színek sorszámai. A 0,1,2,3 már gyárilag megadott, de másképp is definiálhatjuk. A piros, zöld és kék arányát 0.00-tól 1.00-ig terjedő skálán adjuk meg. Az 1 a teljes intenzitás, a 0 a nulla intenzitás.

Eddig csak definiáltuk a színeket. Ha használni is akarjuk őket akkor a COLOR frásjiáttér utasításra lesz szükségünk. Az írás a szöveg PALETTE által meghatározott színszáma, míg a háttér a háttér színének száma. Futtassuk le a következő programot:

```
PALETTE 3,1,1,0 :COLOR 3
PRINT "HELLO"
```

A HELLO szöveg sárgán jelenik meg. Ha a COLOR után csak egy szám van, akkor az mindig a szöveg színét határozza meg.

A PALETTE utasítással 4096 féle színt keverhetünk ki. Miért pont ennyit? Anélkül, hogy részletesebben kifejténénk, fogadjuk el, hogy a 0.00 és az 1.00 között 16 eltérő értéket állíthatunk be. A háromszor 16-féle érték összes variációja $16 \times 16 \times 16 = 4096$. Mielőtt kísérletezni kezdenénk, és a programozást gyakran kísérő *Illegal function call* hibaüzenetbe ütköznénk, várjuk meg a SCREEN és a WINDOW utasításokkal foglalkozó részt. Ugyanis négy színnél több definiálása esetén további képernyőbeállításra is szükség van.

A PALETTE parancs kiegészítője a POINT (x,y), amely az x, y koordinátákkal meghatározott pont PALETTE parancs szerinti színkódszámát adja.

Előljáróban még egy alaputasítással érdemes megismerkednünk. Az y=FRE(x) a bájttban megadott szabad kapacitásról tájékoztat minket. Ha x=1, akkor a szabad Basic területet kapjuk meg, ez kezdetben mindig 25 000. Ha x=-1, akkor a számítógép teljes szabad kapacitását tudhatjuk meg. Ez az Amiga 500 alapgép esetén, ha programok nem futnak, 400 000 körül lehet. Ha x=-2, akkor a számítógép saját szervezéséhez szükséges elkülönített memóriát kapjuk meg. Ez alapesetben 4789 bájt. A FRE-től kapott értékeket a CLEAR.prg.belső utasítással megváltoztathatjuk. Ha nagyobb Basic programot akarunk írni, és legalább 50 kbájtra lenne szükség, akkor a program első sorába írjuk be: CLEAR ,51200

3.5. Egy színes program

Ezek után nézzünk egy példaprogramot, amelyben az új utasításokon kívül az előbb bemutatottak közül is felhasználtunk néhányat.

```
(1) SCREEN 1,320,256,5,1
(2) WINDOW 2,"Színes program",,17,1:CLS
(3) GOSUB col
(4) FOR k=0 TO 20
(5)   GOSUB türelem
(6)   FOR j=0 TO 2500
(7)     NEXT j
(8)   NEXT k
(9) LOCATE 16,17:PRINT - VÉGE ":END
(10)türelem:
(11) COLOR 0,1:tt=k
(12) FOR i=10 TO 100 STEP 10 :t=i/10+tt
(13)   LINE (i,i)-(310-i,246-i),t,bf
(14) NEXT i
(15) COLOR 1,t:tt=tt+1
(16) LOCATE 16,17:PRINT"TÜRELEM"
(17)RETURN
(18)col:
    PALETTE 4,1,.4,0      REM narancs
    PALETTE 5,1,.55,0     REM rozsdabarna
    PALETTE 6,1,.72,0     REM középsárga
    PALETTE 7,1,1,0       REM sárga
    PALETTE 8,.6,.1,13    REM fluórzöld
    PALETTE 9,.53,.8,.13  REM zöld
    PALETTE 10,.4,.6,.5   REM zöld
    PALETTE 11,.2,.4,0    REM sötétzöld
    PALETTE 12,0,.4,0     REM zöldesfekete
    PALETTE 13,0,.6,.67   REM türkiz
    PALETTE 14,0,.8,.9    REM kék
    PALETTE 15,0,1,-6     REM világoszöld
    PALETTE 16,.2,1,.93   REM világoskék
    PALETTE 17,.8,.8,.8   REM világosszürke
    PALETTE 18,.13,.4,1   REM sötétkék
    PALETTE 19,.4,0,1     REM sötétkék
    PALETTE 20,0,0,.6     REM sötétkék
    PALETTE 21,.33,.13,.87 REM liláskék
    PALETTE 22,.6,.2,1    REM lila
    PALETTE 23,1,0,1      REM pirosaslila
    PALETTE 24,.93,.53,.73 REM sötétlila
    PALETTE 25,1,.73,.73  REM lilásbarna
    PALETTE 26,.8,.53,.47 REM mályvabarna
    PALETTE 27,.6,.4,.33  REM barna
    PALETTE 28,.4,.2,0    REM sötétbarna
    PALETTE 29,.4,.4,.4   REM piszkosszürke
    PALETTE 30,.6,.6,.6   REM szürke
    PALETTE 31,.3,.3,.3   REM sötétszürke
(19)RETURN
```

A program begépelésekor a sorok elején lévő zárójeles sorszámokat hagyjuk figyelmen kívül, azok mindössze a szövegre való hivatkozást teszik majd egyértelművé.

Az (1)-es sor az Amiga-Basic egyik nagyon hasznos utasítását mutatja be:

SCREEN No .Pix.S.Szín.Modus

Az utasítás adott tulajdonságú képernyőt definiál és nyit meg. Mik ezek a tulajdonságok ?

A *No* 1 és 4 közötti szám, azaz egyszerre négy képernyőt definiálhatunk. Ennek a számnak szerepelnie kell a (2)-es sorban található WINDOW utasításban is.

A *Pix* az egy sorban elhelyezkedő pixelek (kocka alakú képelemek) száma. Ez határozza meg a képernyő vízszintes méretét.

Az *S* a képernyőn megjelenített sorok száma. Ez tehát a Pixhez hasonlóan szintén a képernyő méretét határozza meg, de függőleges irányban.

Az előző két paraméterrel szorosan összefügg a *Modus*, amely a képernyő vízszintes, 111. függőleges irányú felbontását, finomságát határozza meg. A *Modus* mondja meg, hogy a maximális nagyságú képernyőre összesen hány pixel, 111. sor fér. Ebből következik, hogy a *Pix*, az *S* és a *Modus* egymástól nem teljesen független. A *Modus* értékei a következők lehetnek:

Modus 1 320 pixel — 256 sor (200)

Modus 2 640 pixel — 256 sor (200)

Modus 3 320 pixel — 512 sor (400)

Modus 4 640 pixel — 512 sor (400)

A zárójelbe tett számok a *Workbench*, azaz a *Kickstart* 1.1-es változatában érvényes sorok számát megadó adatok.

A *Szín* paraméter a képernyőn megjeleníthető színek maximális számát adja.

Szín	A megjeleníthető színek maximális száma
1	2
2	4
3	8
4	16
5	32

A *SCREEN* utasítás használatakor mindig vegyük figyelembe, hogy ez az utasítás memóriaigényes, és a túl sok megnyitott képernyő vagy a nagy felbontást biztosító paraméterek használata könnyen a rendszer lefagyásához vezet, és olyankor jön a *Guru*, vagy az *Out of Memory* hibaüzenettel leáll a program. A legnagyobb felbontást a *Modus* 3, 111. 4 eredményezi. Az így definiált képernyő a *Workbench*-ben már megismert *Interlace* mód. A képernyőt a *SCREEN CLOSE No* utasítás zárja le.

A *SCREEN* utasítás elengedhetetlen párja a (2)-es sorban látható *WINDOW* utasítás: *WINDOW n,Cto,(x 1 ,yl Hx2,y2),típusjNío*

Az eddigi gyakorlatunkból már jól ismerjük az ablakok jelentőségét. Most bemutatjuk, hogyan definiálhatunk a *Basic*-ben kedvünk szerinti ablakokat. De nézzük, milyen tulajdonságú is lehet egy ablak!

Tekintettel arra, hogy több ablakot — elvileg akármennyit — használhatunk, az ablakokat is meg kell számozni. Az *n*-nek 0-nál nagyobb egész számnak kell lennie. Célszerű a számozást 2-vel kezdeni, mert az 1-es számú ablak az *Amiga-Basic* saját ablaka. Ezt a rendszer már automatikusan definiálta.

A *Cím* bármilyen idézőjelbe tett szöveg lehet, ami az ablak felső, üzenetet közvetítő keretében fog megjelenni (a példában "Színes program").

Ezután kell megadnunk az ablak méretét, ami természetesen nem lehet nagyobb a SCREEN-nel definiált méretnél. A méretet két pont koordinátájának megadásával határozzuk meg. Az $x1,y1$ az ablak bal felső sarkát, az $x2,y2$ pedig a jobb alsó sarkát határozza meg. Ha nem adunk meg értékeket — mint pl. a (2)-es sorban —, akkor az ablak automatikusan a SCREEN-nel meghatározott képernyő maximális méretét veszi fel.

A *típus* — amely 0 és 31 közötti egész szám — az ablak nem méret jellegű tulajdonságait adja meg. Az alapértékek a következők:

típus=1 Az ablak nagyságát az ismert módon, a jobb alsó sarokban lévő szimbólummal változtathatjuk.

típus=2 Az ablak a képernyőn eltolható.

típus=4 Az ablak a jobb felső sarokban lévő szimbólumokkal az előtérbe vagy a háttérbe helyezhető.

típus=8 Az ablak a bal felső sarokban lévő ponttal becsukható.

típus=16 Az ablak emlékszik, azaz a rajta lévő grafikát, adatokat, információt megőrzi akkor is, ha időközben egy másik ablak eltakarja.

A típusfunkciók a típusszámok összeadásával egyszerűen kombinálhatók. Ha pl. növelhető méretű (típus= 1) és bezárható (típus=8) ablakra van szükségünk, akkor a típust 9-cel adjuk meg. Itt is figyeljünk arra, hogy a növekvő típusszám egyre több memóriát igényel. Különösen "sokba kerül" a 16-os típus használata.

Az utolsó paraméter a *No*. Ez azonos a SCREEN utasítás paraméterével, tehát azt jelzi, hogy az ablak melyik képernyőre kerül.

Van még két, a WINDOW-hoz kapcsolódó utasítás. A **WINDOW CLOSE n** becsukja az *n* számmal jelölt ablakot. A **WINDOW OUTPUT n** pedig az *n* számmal jelölt ablakot aktiválja. Ez nem azt jelenti, hogy adott esetben az ablak a háttérből előre kerül, hanem azt, hogy az ablakban (azaz a képernyőn) valamilyen megjelenítést létrehozó utasítás az így aktivált ablakban hajtódik végre. (Pl. egy háttérben lévő, nem látható ablakban is létre tudunk hozni grafikát, és ha később az ablakot az előtérbe helyezzük, már csak a kész képet látjuk.)

A (3)-as sorban lévő utasítás egy ugrást jelent és egy alprogram végrehajtását. A **GOSUB** cím utasításban a *cím* lehet egy szám, ha a Basic programot hagyományosan sorszámoztuk, de leginkább egy címkét jelöl, ami bármilyen sor elejére írt, kettősponttal lezárt szó. (*Figyelem!* A címke nem lehet Basic utasítás!).

A programban két olyan cúnke is van, amelyre GOSUB utasítással ugunk. Az egyik a col (18), a másik a türelem (10). Nem véletlen, hogy nem türelem szót írtunk. Habár a német billentyűzet ismeri az *ü* betűt, a Basic editor (szerkesztőprogram) — rejtett hibája miatt — eltünteti ezt az *ü* betűt.

Ha valaki a címkében mégis ragaszkodik az *ü* betűhöz, akkor a program írásakor használjon u-t, majd közvetlenül a letöltés előtt írjon be helyette két ü-t (pl. türelem). A letöltéskor a gép csak az egyik **ü-t** tünteti el.

A GOSUB utasítás úgy működik, hogy a program a címkéhez ugrik, és az ott lévő utasításokat hajtja végre mindaddig, amíg egy RETURN-nel nem találkozik (mint pl. a (17)-es és (19)-es sorban). A RETURN hatására a program visszaugrik a GOSUB-hoz, és a program onnan folytatódik tovább. A RETURN a GOSUB elengedhetetlen tartozéka, nélküle a program nem

ugrik vissza. Ha viszont a kelleténél több RETURN-t helyeztünk el a programban, akkor a *RETURN without GOSUB* hibaüzenetet kapjuk.

Ugyancsak ugrást kiváltó parancs a **GOTO** cím. Habár a programban nem szerepel, a logika azt kívánja, hogy a GOSUB-bal együtt tárgyaljuk, ugyanis csak annyiban tér el tőle, hogy az ugrás után a program nem ugrik vissza. Ezért természetesen RETURN sincs.

A (4)-es sorban lévő ciklusutasítás csaknem minden Basic nyelvjárásban megtalálható. Az utasítás két — helyileg is elkülönülő — részből áll, a **FOR x = $n1$ TO $n2$ STEP $n3$** és a **NEXT x** részből. Az x tetszőleges alfanumerikus kifejezés. Lehet egy betű (pl. x), de lehet hosszabb, betűk és számok kombinációjából álló, de mindig betűvel kezdődő kifejezés is (pl. $zxc123alfa$). Az x kezdeti értékét az $n1$, végső értékét az $n2$, a léptetés mértékét pedig az $n3$ határozza meg. A STEP $n3$ elhagyható. Ekkor egyesével léptetünk. Mindhárom értéknek ($n1, n2, n3$) tetszőleges előjelű egész számnak kell lennie.

Az utasítás úgy működik, hogy a FOR és a NEXT közötti utasításokat mindaddig (és annyiszor) végrehajtja a program, amíg az $n1$ el nem éri az $n2$ értékét. Ezután a program a NEXT utáni utasítástól folytatódik tovább.

A ciklusutasítás jól használható a program várakoztatására is. A (6)-os és (7)-es sorban látható összetartozó FOR és NEXT között nincs más utasítás, de a program mindaddig nem megy tovább, amíg $d \& j$ 0-tól egyesével lépkedve el nem éri a 2500-at. (A program az egyes lépések után mindig ellenőrzi, hogy elértük-e.) Belátható, hogy ez még gyors számítógépen is időt vesz igénybe. Az értékek megfelelő megválasztásával tetszőlegesen lassíthatjuk a programot. Például jól alkalmazható a módszer, ha egy felirat elolvasására kell időt hagynunk, mielőtt a képernyőt a program törölné és továbbmenne.

Változtassuk meg a (6)-os sorban a 2500-at 50-re, majd futtassuk le a programot. Ezután írjunk 10 000-et, és így is futtassuk le a programot.

Több FOR...NEXT ciklust egymásba ágyazhatunk, de ügyeljünk arra, hogy a NEXT-ek megfelelő sorrendben zárják le a ciklusokat. Ha egy NEXT utasítást elfelejtünk kitenni, akkor a Basic a *FOR without NEXT* hibaüzenetet küldi.

A (9)-es sorban lévő **END** utasítással leállíthatjuk a programot. Ez az utasítás megegyezik a menürendszer Stop utasításával.

Több helyen is látunk a programban *értékadó utasításokat*, mint pl. a (11)-es sorban a **tt=k**. A tt és a k a változók, és ez az egyszerű utasítás azt mondja, hogy ettől kezdve a tt vegye fel a k aktuális értékét. Természetesen az egyenlőség jobb oldalán hosszabb matematikai kifejezés is állhat. A (12)-es sor utolsó utasítása is ilyen. A / jel az osztást jelenti. *Fontos szabály*: ha valamilyen változónak nem adtunk értéket a programban, akkor azt a gép alapértelmezésben 0-val egyenlőnek tekinti.

A (13)-as sor nagyon hasznos Amiga-Basic utasítást mutat be:

LINESTEP (x1,y1)-STEP (x2,y2).szíh,bf

A parancs hatására a paraméterek függvényében vonalat vagy négyszöget kapunk. Az $x1, y1$ és az $x2, y2$ a WINDOW utasításban tárgyalt koordinátákat adja meg. A szín paraméterrel a négy alapszínből (0-3) vagy a PALETTE utasítással létrehozott színekből választhatunk. A STEP elhagyható rész. Ha mégis kitesszük, akkor az $xlylj.2\$2$ nem az abszolút koordinátákat jelenti, hanem a kurzor aktuális pozíciójához képest vett koordinátákat. A b és az / is elhagyható. Ebben az esetben az utasítás hatására egyenes vonalat kapunk. Ha a b betűt kifizük, akkor a megadott koordináták között felírható négyszög jelenik meg. Ha az / betűt is hozzátesszük, akkor a négyszög a megadott színnel lesz kifestve. (Az alapszínek: 0=kék; 1=fehér; 2=fekete; 3=narancs.)

A program hátralevő része az alapszíneken kívüli további 28 szint definiálja. A színek közül tetszés szerint bármelyik beépíthető a későbbi programokba.

Ez a program a színekkel játszik. Egy kis változtatással azonban jól használhatjuk alprogramként is a már megismert GOSUB utasítással.

Sokszor előfordul, hogy a programban időigényes lépések vannak (pl. lemezműveletek). Ilyenkor a profi programozó leköti a felhasználó figyelmét, s esetleg kéri a türelmét. Ellenkező esetben a felhasználó még azt hiheti, hogy azért nem történik semmi, mert a program hibás, leállt.

Ha a "Színes programból" az (1)-es, (2)-es, (4)-es, (5)-ös és (8)-as sort elhagyjuk, akkor a GOSUB türelem utasítással bárhol előhívhatjuk. Ezután kell következnie annak a programrésznek, amely a hosszabb időt igényli, majd az egészet lezáró (9)-es sornak (:END nélkül írjuk ide, mert így a művelet végét is jelezzük a felhasználónak). A továbbiakban tetszés szerint folytathatjuk a programot.

3.6. Egy kis matematika

A Basic programozásnak a matematikai műveletek is részei, ezért tekintsük át az ide kapcsolódó fontosabb utasításokat, alapfogalmakat. Kezdjük az ismerkedést a *változókkal*. Kétféle változótypust különböztet meg a Basic: numerikus és sztringváltozót. A sztringváltozó jele a \$, és ez mindig valamilyen szöveg. Formája: x\$="szöveg". A numerikus változó egy számérték. Formája: x=253. Az Amiga-Basic a numerikus változók négy fajtáját különbözteti meg:

1. DBL2.23*10⁴-308-tól 7.9*10³⁰⁸-ig (8bájt)

2. SNG 1.18* 10⁻³⁸-tól 3.4* 10³⁸-ig (4 bájt)

3. LNG -2 147 483 648-tól 2 147 483 647-ig (4 bájt) .i

4. INT -32 768-tól 32 767-ig (2 bájt)

(A ^ jel a hatványkitevő.) Jelölésük a következő: x#; x!; x&; x%.

A 3. és 4. típus csak egész számokat jelent, az 1. és 2. típus pedig különböző pontosságú tetszőleges tizedes szám lehet.

Mint látható, az egyes típusok nem egyforma memóriaigényűek. Ha előre tudjuk, hogy milyen típusú változókat fogunk használni, akkor érdemes a DEFTDPUS utasítással a program elején azt megadni. Ha pl. az ábécé első öt betűjét akarjuk viszonylag kis, egészszám-változóként használni, akkor a program elejére a DEFINT a-e írandó.

Az egyes változók indexszel együtt is használhatók (a(1), a(2), a(3), a(4) stb.). A számítógép alaphelyzetben egy tízmezős indexes változóra "számít". Ha az indexek száma meghaladja a tízet, vagy több indexe is van egy változónak, akkor a program elején ezt a DIM utasítással jelezniünk kell. Pl.:

```
DIM a(1000)
DIM x$(22)
DIM z(15,11)
```

A Basic a matematikai műveleteket — kis eltéréssel — a köznapi írásban megszokott módon jelzi. A + és a - az összeadás és a kivonás, a * a szorzás, a / pedig az osztás jele. A hatványkitevőt a ^ jelenti. Ha többlépéses műveletet írunk, akkor a gép előbb a zárójeles

részeket végzi el. Zárójel híján a műveletek sorrendje: hatványozás, szorzás-osztás, összeadás-kivonás. A PRINT $2*3-2*(10+5)$ hatására az eredmény-24.

Végül röviden szólni kell a *logikai műveletekről*. Ezek ugyanis szorosan kapcsolódnak a változókkal végzett műveletekhez. A logikai műveletek mindig valamilyen feltételt jelentenek, s ezt a Basicben az EF THEN vagy rokon utasításaival oldjuk meg. Az egyszerűség kedvéért most csak az IF THEN (HA AKKOR) kifejezéssel mutatjuk be a példákat.

IF X=5 AND y=6 THEN PRINT "RENDBEN"

Ez a kifejezés azt jelenti a számítógép számára, ha egyszerre mindkét feltétel teljesül, azaz x egyenlő öttel és y egyenlő hattal, akkor és csak akkor íródik ki a képernyőre, hogy RENDBEN. Természetesen a THEN után nemcsak a PRINT, hanem bármilyen más utasítás is állhat.

IF x=5 OR y=6 THEN PRINT "RENDBEN"

Ha a két egyenlőség közül az egyik már teljesül, akkor a RENDBEN megjelenik a képernyőn.

IF NOT x=5 THEN PRINT "RENDBEN"

Ebben az esetben a RENDBEN akkor jelenik meg, ha x nem egyenlő öttel.

IF x<5 THEN PRINT "RENDBEN"

A RENDBEN akkor jelenik meg, ha x kisebb ötnél.

Hasonló módon alkalmazhatók a következő operátorok is:

>= nagyobb egyenlő;

<= kisebb egyenlő;

<> kisebb vagy nagyobb.

Számos matematikai függvény is könnyen kiszámolható az Amiga-Basic parancsaival.

- **ABS(x)** az x abszolút értékét adja;
- **ATN(x)** az x arkusztangensét számítja ki;
- **COS(x)** az x koszinuszértékét adja;
- **EXP(x)** az e szám x hatványát számolja ki;
- **FIX(x)** az x értékét adja a tizedesek elhagyásával;
- **INT(x)** az x értékét felkerekíti a legközelebbi egész értékre;
- **LOG(x)** az x természetes alapú logaritmusát adja;
- **RND(x)** véletlen számot generáló függvény;
- **SGN(x)** a szignum függvény, amely az x előjelétől függően -1,0,1 értékeket vehet fel;
- **SIN(x)** az x szinuszerékét adja;
- **SQR(x)** az x -ből gyököt von;
- **TAN(x)** az x tangensértékét adja.

A változókkal, **III-** azok átalakításával kapcsolatosak a következő Basic utasítások is. Az **ASC(x\$)** az x \$ sztring első jelének (ez lehet betű vagy szám, sőt írásjel is) ASCII kódját adja. Az ASCII nemzetközileg elfogadott rendszer, az írásjelek számmal való azonosítására. A nagy A betű pl. a 65-ös kód, míg a kis a a 97. Ha x="alma"$, akkor az **ASC(x\$)** függvény eredménye 97. Ha nincs kéznél a Basic könyv, akkor a következő programmal egyszerűen lekérdezhethetjük akármelyik billentyű ASCII kódját:

```

WHILE 100
  X$ = " "
  WHILE x$ = " "
    x$=INKEY$
  WEND
  FOR i=1 TO LEN(x$)
    PRINT ASC(MID$(x$,i,1))
  NEXT
WEND

```

A programot lefuttatva üres képernyőt kapunk, ahol mindaddig nem történik semmi, amíg egy billentyűt le nem nyomunk. A billentyű lenyomása után azonnal megjelenik egy szám, amely a billentyűn lévő karakter ASCII kódját adja. A kódok felkutatását tetszőlegesen folytathatjuk. A programot a Stop menüutasítással tudjuk leállítani.

Néhány ismeretlen utasítás is van a programban, amelyek közül a WHILE...WEND kétszer is előfordul. Ez nagyon hasonlít a 3.5. alfejezetben megismert FOR...NEXT ciklusutasításra. Ugyanis mindaddig, amíg a WHILE és a WEND közötti állítás igaz, a ciklus folytatódik. Ez a magyarázata annak is, hogy a program magától nem áll le, hiszen az első sorban szereplő WHILE 100 folyamatosan igaz. Hogy megértsük miért, nézzünk egy ellenpéldát.

Ha az első sorban a WHILE $x < 100$ állna, és x értéke folyamatosan nőne a program valamelyik sorában elhelyezett utasítás hatására, akkor ha x elérné a 100-at, a ciklus megszakadna.

Fontos utasítás az x=INKEY$$ is. Ennek hatására a gép valamelyik billentyű lenyomását várja, azután az $x$$ -t a billentyű által képviselt karakterrel teszi egyenlővé. Ha a billentyűt nem nyomjuk le, az is egy bizonyos jel — az ún. üres jel —, amit két idézőjellel szoktunk jelölni (pl. x=""$). Azért, hogy a program ilyenkor ne fusson tovább és megvárja egy billentyű lenyomását — azaz egy ""-től különböző jel érkezését —, a második WHILE...WEND utasítást alkalmazzuk. Az x=""$ esetén ugyanis most nem megy tovább a program, hanem a WEND-től újból indul a ciklus mindaddig, amíg $x$$ értéke bármilyen, de ""-től eltérő lesz.

A LEN utasítás egy sztring karaktereinek számát, azaz a sztring hosszát adja meg. Ha az x="alma"$, akkor a LEN($x$$) értéke 4.

A MID\$ utasítás egy sztringből egy bizonyos részsstringet hoz létre. Ha x="utasításokat"$, akkor a MID\$(x,8,2$) eredménye "so". Az utasításban lévő első szám ugyanis azt határozza meg, hogy a sztringen belül, balról jobbra, hányadik karaktertől kezdődjön a kivágás. A második szám a kivágandó karakterek számát adja meg. A Basic könyv szerint mindkét szám 1 és 32 767 közötti egész szám lehet, de nyilvánvaló, hogy az $x$$ sztring hosszánál nagyobbak nem lesz értelme.

Itt szólunk a MID\$ utasítással rokon másik két utasításról is. Az y=LEFT(x, n) az $x$$ sztring bal oldali első n számú karakterét helyettesíti be $y$$ -ba., az y=RIGHT(x, n) pedig ugyanezt teszi a jobb oldalról kiindulva. A LEFT$(x,1) paranccsal pl. megadhatjuk egy szó kezdőbetűjét.$$$

A VAL($x$$) az $x$$ sztringben előforduló esetleges szám értékét adja. Ha x="alma24xc"$, akkor a függvény értéke 24. Ha a sztring nem tartalmaz számot, akkor a függvény értéke 0.

Végül néhány átalakító függvényt ismertetünk:

- CDBL(x) az x -et DBL típusú változóvá alakítja;
- CINT(x) az x -et INT típusú változóvá alakítja;
- CLNG(x) az x -et LNG típusú változóvá alakítja;
- CSNG(x) az x -et SNG típusú változóvá alakítja.

Nem tartozik szorosan ide, de itt említjük meg a `VS=DATE$` parancsot, amely az aktuális dátumot adja. Természetesen, ha a gépben nincs belső óra, akkor csak a Workbench-en beállított dátumot kaphatjuk meg.

3.7. Az IF...THEN és variációi

Már alkalmaztuk az IF...THEN feltételes utasítást. Mielőtt továbbmennénk, részletesebben is bemutatjuk ezt a fontos utasításrendszert, amely szinte minden Basic programban előfordul. Utasításrendszer, mert számos variációja van.

A legegyszerűbb az **IF...THEN** kifejezés, ahol ha az IF és a THEN közötti állítás igaz, akkor először a THEN utáni részt hajtja végre a program, ellenkező esetben azt figyelmen kívül hagyva, a következő sorra ugrik.

Ha azonban az utasítást **IF...THEN...ELSE** formában írjuk, akkor ha az IF és a THEN közötti rész igaz, akkor ismét a THEN mögötti részt hajtja végre a program, ellenkező esetben viszont — mielőtt a következő sorra ugrana — előbb az ELSE mögötti részt.

Gyakran előfordul, hogy a THEN utáni rész nem fér el egy sorban, sokszor akár hosszú programrészlet is lehet. Ilyenkor a következő formát alkalmazzuk:

IF...THEN

·
·
·

ENDIF

Ha az IF és a THEN közötti állítás igaz, akkor az IF sor és a THEN sor közötti rész végrehajtódik, és ezután az END IF után folytatódik a program. Ellenkező esetben a program mindjárt az END IF után folytatódik.

Ritkábban használatos, de jegyezzük meg, hogy a THEN helyett a GOTO-t is alkalmazhatjuk, a GOTO után azonban csak egy érvényes sorszám vagy címke állhat.

Előfordul az is, hogy több feltételt kellene egymásba ágyazni, több IF...THEN utasítás kombinációjára lenne szükség. Ezt megoldhatjuk több "IF...THEN...END IF"-fel, de lehetséges az ELSEIF rendszerbe való beépítése is.

IF...THEN

·
·

ELSEIF...THEN

ELSEIF...THEN

ELSE

·

ENDIF

Ebben az esetben, ha az IF...THEN közötti rész állítása igaz, akkor a program végrehajtja az IF...THEN és az első ELSEIF...THEN közötti részt, majd az END IF után folytatódik. Ha az IF...THEN közötti állítás nem igaz, akkor a program megvizsgálja az első ELSEIF...THEN közötti részt, s ha az igaz, akkor végrehajtja a két ELSEIF...THEN közötti részt, s csak utána ugrik az END IF mögé. Ha azonban ez az állítás sem igaz, akkor a következő ELSEIF...THEN közötti részt vizsgálja, és az előbbi logika szerint megy tovább.

Tetszőleges számú ELSEIF...THEN blokk építhető egybe. A blokk végén álld ELSE nem kötelező. Ha kitesszük, a program — mivel az összes előző állítást hamisnak találta — először az ELSE és az END IF közötti részt fogja végrehajtani.

Ha az ELSE hiányzik, akkor mindjárt az END IF után folytatódik a program.

3.8. Grafika

A grafikai lehetőségek kihasználása nem a Basicból származik, ennek ellenére a grafikával, a képernyőn való megjelenítéssel kapcsolatban jó néhány parancs áll rendelkezésünkre. Ezek a parancsok nem tartoznak bele a szokványos Basicbe. A Basic kiegészítéseként készült programok (pl. a Commodore 64 Simon's Basicje) tartalmaznak ilyen utasításokat.

A **CIRCLE STEP (x,y),r,Szín,Start.Stop,Arány** parancssal kört vagy ellipszist rajzolhatunk. Az xy a középpontot, az r pedig a sugarat jelöli. A STEP elhagyása vagy alkalmazása (ugyanúgy, mint a LINE-nál) az abszolút vagy relatív koordinátát jelenti. A *Szín* az alapszínnek vagy a PALETTE utasítással megadott színek egyike. Ha a körnek vagy az ellipszisnek csak egy szeletét akarjuk megrajzolni, akkor a *Start* és a *Stop* segítségével adhatjuk meg az indulási és a befejezési szöveget. Ellipszis esetén az *Arány* megadja a magasság és a szélesség arányát.

Az **ÁRKA STEP (x,y)** egy sokszög megrajzolásához nyújt segítséget. Az utasítás egyszeri kiadásával a sokszög egyik sarokpontját definiáljuk. Ezután a megfelelő paraméterekkel addig ismétljük a parancsot, amíg a szükséges sokszöget meg nem kapjuk. A STEP funkciója megegyezik a korábban leírtakkal.

Az **AREAFILL Modus** parancsot használhatjuk az előbbi sokszög kifestésére. Ha a *Modus* értéke 1, akkor a sokszög belseje inverz módon lesz kitöltve. Ha a *Modus* 0, vagy nem adunk neki értéket, akkor a PALETTE első színét kapja a sokszög.

A **PAINT STEP(x,y),Szfh,Keret** utasítás is a kifestéskor használható. Az x,y koordinátának azon a tárgyon belül kell lennie, amit ki akarunk festeni. Ugyanazt a hatást általában végtelenül sok $x,y-na$ érhetjük el. A kézikönyvvel ellentétben nemcsak a négy alapszín, hanem a PALETTE beállítása után 32 szín áll rendelkezésünkre. A *Kéret* a keret színét megadó szám. Vigyázzunk ezzel a parancssal! Ugyanis a *Szín* csak a kifestendő testet körülvevő rész színe lehet. Ha mégis más színnel próbálkozunk, akkor az "kifut" a kereten kívülre. Próbáljuk ki!

A **GET (x1,y1)-(x2,y2),Változd,Index** és a **PUT STEP (x,y),Változó,Index,Modus** két szorosan összekapcsolódó parancs. A GET parancssal a képernyő egy részletét tárolhatjuk a memóriában, majd a PUT-tal ismét elhelyezhetjük a képernyőn. Az $x1,y1$ és az $x2,y2$ a szóban forgó részlet bal felső, ül. jobb alsó csúcsának koordinátáját adja, az x,y pedig a bal felső koordinátát. A képernyőpontokról a tulajdonképpeni információt a *Változó-ban* tárolhatjuk. Az *Index-re* akkor van szükség, ha több grafikus információt tárolunk az *Indexszel* meghatározott többdimenziós *Változó-han*. A PUT parancs esetében a *Modus* a képernyőre vitel módját írja le.

A következő lehetőségek vannak:

PSET = a képernyőre vitel pontonként megy végbe, a színek változatlanok;

PRESET = ugyanaz, mint a PSET, de inverz színekkel;

AND = csak akkor helyezi el a képpontokat, ha az átvendő pontok alatt is van valamilyen grafika;

OR = átmásolja a már létező képet;

XOR = a kép mozgathatásakor nem törölünk bele a háttérbe.

Érdekes utasítás a **SCROLL (x1,y1)-(x2,y2),deltax,deltay**. A képernyőn levő kép görgetését, eltolását (szkrollozását) teszi lehetővé. Az x,y a szokásos képkoordinátákat, a $deltax$ és a $deltay$ pedig az x , ill. y irányba való képeltolás mértékét adja képpontokban számolva. Nézzünk erre egy példát!

```
LOCATE 8,1
```

```
PRINT " A COMMODORE AMIGA KÖSZÖNTI AZ ÖSSZES KEDVES OLVASÓT"
```

```
SCROLL (100,50)-(400,100),0,-5
```

```
WHILE 1:SLEEP:WEND
```

Az utolsó sorra azért van szükség, hogy a program ne álljon le a SCROLL végrehajtása után. Az utolsó sor nélkül ugyanis azonnal megjelenik a List ablak, és a képernyő felét eltakarja. A **SLEEP** (alvás) utasítás várakoztatja a programot.

A **PATTERN Vonal .Mező** paranccsal egy vagy két olyan vonal- vagy mezőmintázatot tudunk létrehozni, amelyet a korábban ismertetett parancsokkal vonalhúzásra, ill. adott felületek szín helyett mintával való feltöltésére alkalmazhatunk. A *Vonal-dX* a korábban megtanult INT számmal adhatjuk meg. Ez 2 bájt nyit foglal le a memóriából. A 2 bájt 16 bitből áll. Az egyes bitek határozzák meg, hogy a vonal — amely pontok összességéből tevődik össze — pontjai közül melyek vannak definiálva, s melyek nem. A létező pont bitértéke 1, a ki nem tett ponté 0. A mező összetételét az előzőek szerint felépülő vonalak összessége határozza meg. Ily módon a mező szélessége adott, 16 pontnyi, magassága az alkalmazott vonalak számától függ. A megengedett növekmények a kettő hatványai, azaz 1,2,4,8 stb.

A 16 bites leképezésnek megfelelően a számok megadásakor a *tizenhatos* (hexadecimális) *számrendszert* alkalmazzuk. Erről annyit kell tudni, hogy nem 9 után van a nagyságrendi váltás, azaz nem innen kezdve lesz az eddig egyjegyű szám kétjegyű, hanem 15 után. Mivel tíz különböző számjegyet ismerünk, ezért segédszámjegyeket kell bevezetni. A 10, 11, 12, 13, 14, 15 az A, B, C, D, E, F betűkkel képezhető. Segítségképpen közöljük néhány hexadecimális számrendszer-beli szám tízes számrendszer-beli megfelelőjét.

1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Érdemes megjegyezni ezt a számrendszert, mert a számítástechnikában legalább olyan gyakran alkalmazzák, mint a kettes számrendszert.

Az Amiga-Basic a kettes és a tizenhatos számrendszeren kívül a nyolcas számrendszert is ismeri. A tízestől való eltérést a szám elé tett jellel közölhetjük a géppel. Az &O jelzi a nyolcas (okta), az &H a tizenhatos (hexa) számrendszert. Az átszámolással nem kell bajlódunk, az Amiga-Basic megfelelő utasításait alkalmazva ez gyerekjáték.

PRINT HEX\$(tízes) a tízest alakítja hexába;

PRINT &H hexa a hexát alakítja tízesbe;

PRINT OCT\$(tízes) a tízest alakítja oktába;

PRINT &O okta az oktat alakítja tízesbe.

3.9. Játék az adatokkal

A számítógépek egyik legfontosabb feladata az adattömegek tárolása. A tárolás nem öntörvényű, hanem bizonyos **jól** átgondolt szempontok szerinti. A legfontosabb követelményekhez tartozik, hogy a lehető legkisebb helyen minél több adatot tudjunk tárolni, azokat gyorsan elő tudjuk keresni, s ha lehet, már a tároláskor bizonyos rendezettség valósuljon meg, amely további rendezések alapja lehet. Az adatok általában számok vagy szövegek (pl. könyvelési számadatok, névsorok).

Adattárolásra többé-kevésbé minden számítógép képes. Az Amiga viszonylag nagy tárolókapacitásával (880 kbájtos lemezkapacitás) és rendkívüli gyorsaságával tűnik ki. Emellett van egy különleges tulajdonsága: speciális formában grafikát és zenét tud tárolni. Ezt a formát interchange File Formainak (e.: interscendzs fájl formát), rövidítve IFF-nek hívják. A rendszert az amerikai Electronic Arts cég alkotta meg, amely számos szoftvert is forgalmaz.

3.10. Szekvenciális fájl

Az adatokat a legegyszerűbben soros (szekvenciális) módszerrel tárolhatjuk, azaz az egyes adatokat sorban egymás mögé írva a lemezen. A rendezettséget mindössze ez a sorrend jelenti. A visszaolvasás a betöltéssel pontosan ellentétes.

Ha **a** sok adat közül csak egyre van szükségünk, akkor is be kell olvasnunk az egész *adatbázist* (így hívják a valamilyen módon rendezett adatok halmazát). Hasonlóan, ha csupán egy adatot akarunk megváltoztatni, akkor is előbb be kell olvasni az egész adatbázist, el kell végezni a változtatást, majd az egészet újból le kell tölteni, felülírva ezzel a korábbi adatbázist. Nézzük, melyik parancsokkal tehetjük mindezt!

OPEN"O",#l,'név*

Az OPEN parancsral közöljük a számítógéppel, hogy a lemezen valamilyen néven adatbázist kívánunk megnyitni, és oda adatokat akarunk kivinni. Az OPEN után idézőjelben lévő *O* ugyanis azt jelenti, hogy kivitel (output) következik. A # jel után álló számnak túl sok jelentősége nincs, de jegyezzük meg, hogy ha az adatbázist egy adott számmal megnyitottuk, akkor az összes későbbi parancsban a # után ezt a számot kell írni, és két adatbázist egyszerre ugyanazzal a számmal nem nyithatunk meg.

WRITE#l,x\$,y\$,x,y...

Ha az adatbázist megnyitottuk, akkor a WRITE parancsral írhatunk bele. A WRITE (írni) parancsban az *x\$,y\$* tetszőleges sztring, az *xj* pedig tetszőleges szám, a sorrend nincs meghatározva. írhatunk csak számokat, csak szöveget és mindkettőt egyszerre. Van egy másik, különleges befrási mód, amely igen hasznos lehet. Itt is a WRITE parancsot alkalmazzuk, de másfajta megnyitással.

OPEN"A"#l,*név*

Ezt a megnyitási módot akkor használjuk, ha az adatbázis már létezik a lemezen, és egy vagy több új adatot akarunk hozzáfűzni (append). Hozzáfűzéskor az adatbázist nem kell előbb beolvasni, ezért a művelet gyorsan végbemegy.

Az eljárás korlátja, hogy az új adat csak a korábban már bevitt adatbázis adatai mögé kerülhet. Ha pl. neveket vittünk fel ábécé sorrendben, akkor az új nevek bevitelle tönkreteszi a kialakított rendezettséget. A legtöbbszor azonban ez nem okoz gondot, és a parancs jól alkalmazható.

írni már tudunk az adatbázisba, most nézzük meg, hogyan lehet az adatokat előhívni.

OPENT,#1,"név"

A különbség csupán az OPEN után álló / (az input rövidítése), amely az adatbevitelre utal. Ehhez társul a tulajdonképpeni beolvasó parancs.

INPUT* U\$,y\$,x,y....

Ez a parancs a **WRITE** parancsnak megfelelő.

A szekvenciális adatkivittel és -bevitellel kapcsolatos parancsok ismertetése után nézzünk egy másik igen fontos (ún. lezáró) parancsot. Ezt akkor kell alkalmazni, ha az adatmozgatást végleg befejeztük.

CLOSE#1

Egymás mellett több adatbázist is kezelhetünk, **III** megnyithatunk, s ha a munkát befejeztük, akkor a **CLOSE# 1,#2,#3,...** parancssal egyszerre lezárhatjuk őket.

Egy adatbázis felépítésekor ahhoz, hogy az adatokat letölthessük, először be kell olvasni azokat a billentyűzetről a számítógépbe. Erre való az **INPUT** parancs, amelynek #-kal és számmal kiegészített változatával már megismerkedtünk. Az **INPUT** parancs kétféle lehet, attól függően, hogy sztringet vagy számot akarunk-e bevinni.

INPUT *szöveg*pc\$ (sztringek bevitele)**INPUT "szöveg"pc (számok bevitele)**

Mindkét esetben a parancs hatására a képernyőn megjelenik egy kérdőjel, és a program addig vár, amíg egy megfelelő választ (azaz adatot) nem billentyűzünk be. Az idézőjelbe tett szöveg bármi lehet. A legtöbbször azonban kérdés, amely a szükséges adatra kérdez rá (pl. **INPUT "Mi az Ön neve"pc\$**). A szöveg el is maradhat, ilyenkor a pontosvessző sem kell.

Az adatbevitelt az Enter lenyomásával fejezzük be. A program ekkor folytatódik, a számítógép azonban először értékeli a választ. Ha nem megfelelő adatot (pl. szám helyett szöveget) adtunk be, akkor a gép hibaüzenetet küld, és újból kéri az adatot. Ez mindaddig folytatódik, amíg helyes (azaz megfelelő típusú) adatot nem szolgáltatunk. Ha a pontosvessző helyett vesszőt írunk, akkor a kérdőjel elmarad.

Minden **INPUT** utasítás a **PRINT**-hez hasonlóan új sort nyit meg. Figyeljünk, hogy az **INPUT**-ra adott válaszuk ne tartalmazzon ;: jeleket, mert hibaüzenetet kapunk.

Az **INPUT** különleges fajtája a **LINE INPUT**. A gép ilyenkor az előbbieken felsorolt írásjeleket is elfogadja. A **LINE INPUT**-tal számot csak sztringként vihetünk be. Ha azzal a továbbiakban számként akarunk dolgozni, akkor a **VAL** utasítást kell használnunk. Az érthetőség kedvéért nézzük a következő egyszerű programot, amely házi könyvtárunkról készít adatbázist.

```
REM Könyvtár
CLS:x=1
OPEN"0",#1,"könyv"
WHILE x=1
  LINE INPUT"Mi a könyv címe ";c$
  LINE INPUT"Ki írta a könyvet ";ir$
  INPUT"Mi a könyv sorszáma ";No
  PRINT"Van még könyv <I(gen)/N(em) > ?"
  ir$=ir$+": "
  GOSUB var
  IF x$="N" THEN x=2
  c$=OCASE$(c$)
  WRITE#1,No,ir$,c$
WEND
```

```

CLOSE#1
CLS:PRINT
PRINT"Kíváncsi a bevitt adatokra <I (gen)/N(em) > ?••
GOSUB var
IF X$="N" THEN END
CLS
OPEN"I",#1,"könyv"
WHILE NOT EOF(1)
    INPUT#1,No,ir$,c$
    PRINT No;ir?;c$
WEND
CLOSE#1
END
var:
x$=INKEY$:IF x$="" THEN var
x$=UCASE$(x$)
IF NOT x$="I" AND NOT x$="N" THEN var
RETURN

```

A programban szereplő utasítások nagy része már ismert, de vannak újak is. Az **UCASE\$(x\$)** utasítás az *x\$* sztringben lévő szöveget (függetlenül attól, hogyan írtuk) csupa nagybetűvé alakítja. Az **EOF(x)** a szekvenciális fájl végének elérését jelzi. Az *x*-nek a # utáni számmal kell megegyeznie. Bár a program könnyen követhető, röviden mégis elmagyarázzuk.

A "könyv" nevű szekvenciális fájl megnyitása után a program folyamatosan bekéri a könyvek adatait. Ez egy **WHILE...WEND** cikluson belül megy végbe, amely mindaddig megszakítatlanul megy tovább, amíg a feltett kérdésre nemmel válaszolunk. Nem válasz esetén ugyanis *x=2*, ami nem felel meg a **WHILE x= 1** feltételnek. Ezután a **CLOSE** utasítás lezárja az adatkivítelt, és újbdl egy kérdés következik. Igen válasz esetén a program az összes kivitt adatot visszatölti és folyamatosan megjeleníti a képernyőn.

A program legvégére került a "var" alprogram, amelyet későbbi programjainkba is beépíthetünk, ha egy eldöntendő kérdésre kell igennel vagy nemmel válaszolni. Vizsgáljuk meg a működését!

Az első sorban egy billentyű lenyomására várunk, de ha az a "" jel, ami tulajdonképpen nem billentyű jel, de azzal egyenértékű, akkor a program nem megy tovább, mert az **IF...THEN** újra az alprogram elejére küld. Ha lenyomunk egy billentyűt, akkor az előbb nagybetűre vált, így kiküszöbölhető, hogy a nagy- és kisbetű közötti különbség esetleg zavart okozzon. A harmadik sorban kiértékelődik a kapott jel. Ha nem i-t vagy rc-et nyomtunk le, akkor az egész előlről kezdődik.

A bemutatott program igen egyszerű, mégis jól alkalmazható, s könyvek helyett akármi mást is katalogizálhatunk. Egy kis átalakítással nemcsak egyszerre, hanem bármikor bevihetjük az adatokat, ha a harmadik sort kicseréljük az **OPEN"A", #1, "könyv"-re**. Ha nem akarunk adatot bevinni, csak a már meglévőket olvasni, akkor a beviteli részt egy megfelelően elhelyezett **GOTO** utasítással átugorhatjuk. Kísérletezzünk csak, s próbáljuk továbbfejleszteni a programot!

3.11. Közvetlen hozzáférésű fájl

A közvetlen hozzáférésű fájl magasabb szintű és rendezettségű, mint a szekvenciális. Az adatok tárolásakor a jól definiált *rekordok* összességét foglaljuk össze egy adatbázissá. A rekordok megfelelő sorrendben tartalmazzák a tulajdonképpeni adatokat. Egy könyvhöz

hasonlíthatjuk ezt az adatbázist, amelyben a sorok a rekordok, s ezek összessége adja a könyvet, azaz az adatbázist. A különbség csupán annyi, hogy az adatbázis rekordjainak hossza bizonyos keretek között szabadon választható, de az adatbázison belül minden rekordnak egyforma hosszúságúnak kell lennie. Az egyes rekordokat (tartalmukon kívül) a sorszámuk, azaz az adatbázison belül elfoglalt helyük különbözteti meg.

Mi az előnye ennek az adatkezelési módnak? Mindenekelőtt az, hogy egy adat bevitelekor nem kell beolvasni az egész adatbázist, elegendő ismerni a kiválasztott rekord sorszámát, hiszen csak az adatot majd tartalmazó rekordra lesz szükségünk. Ezenkívül jól megválasztott rekordösszetétel esetén — a rekordok viszonylagos függetlensége miatt — sok lehetőség kínálkozik különféle képpen kialakított rendezettségek megvalósítására. Nagyobb adatbázisok kezelése csak ezzel a módszerrel oldható meg, szekvenciálisán ugyanis nehéz.

Nézzük, hogyan valósíthatjuk meg a közvetlen hozzáférést!

```
OPEN "R", #1, "neV,x
```

A megnyitás hasonló a szekvenciális fájléhoz. Az *R* betű a relatív fájlra utal, gyakran így nevezik ezt a fájlt. A másik eltérés az *x*, amely egész szám, és a rekord hosszát adja meg. Azonban a rekord szerkezetét is meg kell határozni.

```
FIELD #1, y1 AS y1$, y2 AS y2$, ...
```

A FIELD utasítás a rekord felosztását adja meg, az előző hasonlittal élve meghatározza, hogy a könyv egy sorában hány betű jelöl egy-egy dolgot. Az *y1y2,...yn* egész számok, és összességük az OPEN-ben lévő *x*-szel egyenlő. (A betűk számának meg kell egyeznie a sor hosszával.) Az *y1y2\$,...yn\$* általunk választott tetszőleges, de *y1y2,...yn* hosszúságú sztringek. Az AS a FIELD része, jelentése mint.

Nézzünk egy egyszerű példát!

```
OPEN "R", #1, "könyv", 30
```

```
FIELD #1, 2 AS No$, 14 AS ir$, 14 AS c$
```

Mivel pontosan rögzítettük a rekordok — és azon belül az egyes sztringek — hosszát, kell egy utasítás, amellyel ezt a hosszúságot be is tudjuk tartani. Az egyes adatok ugyanis nem egyformák. A FIELD-ben a hosszakat úgy kell meghatározni, hogy a leghosszabb adat is beférjen. A rövidebb adatokat pedig nem zavaró, ún. üres jelekkel kell feltölteni. Ha pl. *c\$="alma"*, akkor — hogy a 14 AS *c\$* kifejezésnek megfeleljen — vagy *c\$="alma....."*, vagy *c\$=".....alma"* módon kell átalakítani. Az előbbire az LSET, az utóbbira pedig az RSET utasítást használhatjuk. (A pontok üres helyeket jelentenek.)

```
x$=MKI$(y%) (INT)
```

```
x$=MKL$(y&)(LNG)
```

```
x$=MKSS$(y) (SNG)
```

```
x$=MKD$(y#) (DBL)
```

A közvetlen hozzáférésű fájlba, a szekvenciálissal ellentétben, nem lehet számokat közvetlenül bevinni. Előbb sztringgé kell azokat alakítani. A különféle pontosságú numerikus karaktereknek megfelelően erre az előző négy függvény alkalmas.

```
x=CVI(x$) (INT)
```

```
x=CVL<x$) (LNG)
```

```
x=CVS(x$) (SNG)
```

```
x=CVD(x$) (DBL)
```

Szükségünk van az előző függvények ellentettjére is, ha az adatbeolvasás után a számoknak megfelelő sztringekből újból számokat akarunk kapni.

Közvetlen hozzáférésű fájlok esetén a WRITE# utasításnak a PUT#, az INPUT# utasításnak pedig a GET# felel meg. Van egy segédutasítás is, amely az EOF(x)-hez *hasonlít*: $y=LOC(x)$

Az y annak a rekordnak a számát adja meg, amellyel utoljára műveletet végeztünk. Az x itt is a # utáni számmal egyezik meg.

A FIELD utasításban szereplő sztringváltozókat azonban a programban másutt nem lehet használni. Az adatok program szerinti bevitelkor alkalmazott INPUT utasításban is egy másik változót kell használni. Csak az LSET, ill. RSET átalakítás során jelenhet meg a valódi változó. Ha ezt figyelmen kívül hagyjuk, az adatbázis nem működik.

y=LOF(x)

Az y az *-szel jelölt fájl hosszát adja meg bájtokban. A függvény kis kiegészítéssel könnyen felhasználható olyan rutin készítésére, amely lekérdezi, hogy a fájl vajon létezik-e a lemezünkön.

```
OPEN "A", #1, "Név"
IF LOF(1)=0 THEN a=1 ELSE a=2
CLOSE #1:PRINT a
```

A *Név* a keresett fájl neve. Ha a fájl rajta van a lemezen, akkor a értéke 2, ellenkező esetben 1. Természetesen az a értékének kiírása helyett másképpen is jelezhetjük az eredményt.

A közvetlen hozzáférésű és a szekvenciális módszer jól kiegészíti egymást, ha nagyobb adatbázist kell felépítenünk. A megfelelő szerkezetű adatbázist a közvetlen hozzáférésű módszerrel kell létrehozni. Ahhoz azonban, hogy az adatok mozgathatók legyenek, mindig tudjuk, hogy melyik (hányas számú) rekordot kell beolvasni vagy átírni, szükségünk lesz egy segédadatbázisra, amely lehet szekvenciális is. Ez az *index*.

Ha nagyobb könyvtárat akarunk bevinni az adatbázisba, amelyben a könyv címén és szerzőjén kívül még további adatokat is nyilvántartunk, akkor az indexfájl — ami lehet szekvenciális — a könyvek címéből állítjuk össze. Ez a fájl sorban tárolja a címeket, és minden cím mellett egy számot is tárolhatunk, a rekordsorszámot. A szekvenciális fájlban azonban a sorrend meg is egyezhet a közvetlen elérésű fájlban található rekordsorrenddel. Ha egy könyvet keresünk, az index alapján megállapítjuk a rekordsorszámot, majd beolvassuk a kérdéses rekordot.

A professzionális adatbázisok természetesen ennél sokkal bonyolultabb módszereket is alkalmaznak, hogy az adatbázis minél helytakarékosabb és gyorsabb legyen, de minden módszer többé-kevésbé erre az elvre vezethető vissza.

3.12. IFF fájl

Az IFF adatkezelést nem ismertetjük részletesen, csak a felhasználó számára legfontosabb alapokat és információkat. Az IFF fájl ún. Chunkokból (e. csánk) áll. Egy fájlban tetszőleges számú Chunk lehet. A Chunk első két adata mindig a Chunk azonosító neve és hossza, ezután következik a tulajdonképpeni tartalom. Bizonyos esetekben a Chunk befejezésekor egy töltelékbitra is szükség van, hogy a Chunk mindig páros memóriacímen kezdődjön. A legelső Chunk előtt 12 bájtszámú azonosító van, ebből ismeri fel a számítógép, hogy IFF adat következik. Ez az azonosító azt is jelzi, hogy grafikával (esetleg mással) dolgozunk. A Chunkok különböző funkciókat hajtanak végre, grafika esetén pl. a képernyőfelbontást, az elhelyezkedést, a színeket és a rajzolatot definiálják.

Ha IFF fájlokkal akarunk dolgozni, a kész programok megkímélnék a részletes programozási munkától. Vannak olyan adatbázisok is, amelyek a szöveges, ill. számszerű adatokon kívül IFF adatok bevitelére is alkalmasak. Az Extrás lemez LoadILBM-SaveACBM nevű programjával pl. a Deluxe Paint programmal készült IFF grafikát beolvashatjuk saját Basic programunkba, és a program futtatásakor megjeleníthetjük a képernyőn.

3.13. Menüprogramozás

A roló menürendszer igen célszerű és kényelmes, saját programjainkba beépíthető. A menürendszer fontos része egy jól megírt programnak, lehetővé teszi, hogy a felhasználó hosszabb keresgélés nélkül könnyen eligazodjon a programban kínált sokféle funkció között. Az Amiga-Basickel magunk is létrehozhatunk ilyen menürendszereket. Az ehhez szükséges parancsok viszonylag egyszerűek, használatuk könnyen elsajátítható.

Mielőtt elkezdenénk, ismerkedjünk meg egy érdekes vezérlőutasítással, amely nem kifejezetten ebbe a témakörbe tartozik, de működési mechanizmusának ismerete megkönnyíti a menüparancsok megértését.

ONx GOSUB

ONxGOTO

Mindkét paranccsal egy címkéhez (programrészhez) ugorhatunk az x értékétől függően. A jól felépített program néhány GOSUB utasítással elérhető alprogramot is tartalmaz, de a program egyes funkcióinak is egymást követő logikus szerkezetben kell lenniük.

Az előző parancsok különleges döntési helyzetet oldanak meg — hasonlóan az egyik korábbi mintaprogramban alkalmazott igen-nem választáshoz —, itt azonban kettőnél jóval több választási lehetőség van. Az egyes döntési lehetőségekhez ugyanis egy szám — az i — tartozik, és ennek függvényében a GOTO vagy GOSUB mögött felsorolt címke valamelyikéhez ugrik a program.

Ha pl. azt írjuk a programban, hogy `ON x GOTO név,cím,adat,200,bevitel,törlés,vége` majd i értékét hatnak választjuk, akkor a program a törlés címkétől folytatódik tovább. Mivel a hagyományos menürendszert ezzel a módszerrel lehet felépíteni, érdemes megtanulni, és néhány példán kipróbálni, mert nem célszerű mindig a rolómenüt alkalmazni.

Visszatérve a feladatunkhoz, a rolómenüben is hasonló jellegű utasítással érhetjük el az elágazást.

ON MENÜ GOSUB címke 1 ,cfinke2,...

A parancs hatására, attól függően, hogy melyik menüpontot választottuk ki az egérrel, a program a megfelelő alprogramhoz fog ugrani, de csak akkor, ha már felépítettük a menürendszert. Erre való a MENÜ utasítás.

MENÜx.y.z/'név

Az $xc?$, z pozitív egész számok. Az x megadja, hogy melyik menüről van szó. A számozás balról jobbra értendő, (az Amiga-Basic Project menüje pl. az 1-es számot viseli, az Edit menü a 2-es stb.). Az y az adott menün belüli menüpontot jelenti (a Project menü New menüpontja pl. az 1-es). Legfeljebb 19 menüpont lehet egy menüben. Az a menüpont státusát határozza meg. Ha $z=0$, akkor a menüpont csak halványan (szaggatottan) jelenik meg, és ilyenkor ezt a menüpontot nem is lehet aktiválni. Ha $z=1$, akkor a menüpont aktív. Akár az egész menüt inaktíválhatjuk akkor, ha y -nak 0 értéket adunk.

Az értéke lehet 2 is. Ilyenkor egy kis kampó jelenik meg a menüpont neve mellett, és ez jelzi, hogy a menüpont nem funkciót hajt végre, hanem két értéket vehet fel, és ezek közül az egyik be van kapcsolva. A Workbench Notepad nevű szövegszerkesztő programjában pl. a Wordwrap ugyanilyen menüpont. Ha a kis kampó megjelenik, akkor a Wordwrap be van kapcsolva, ellenkező esetben nincs. Ha rákattintunk a menüpontra, akkor a ki-, III. bekapcsolt állapotot az ellenkezőjére változtatjuk.

A MENÜ utasítás utolsó eleme a tetszőlegesen választott, idézőjelbe tett *név*. (Célszerű rövid és találó nevet választani.)

összefoglalásként nézzük meg az Amiga-Basic Project menüjének definícióját.

```
MENÜ 1,1,1,"New"  
MENÜ 1,2,1,"Open"  
MENÜ 1,3,1,"Savé"  
MENÜ 1,4,1,"Savé As "  
MENÜ 1,5,1,"Quif'
```

Az Amiga-Basicnek 4 menüje van, ezeket rendre az 1, 2, 3, 4 számok jelölik. Ha ezeket a menüsíkokat alkalmazzuk saját programunkban is, akkor felülírjuk az Amiga-Basic menüit, s ez bonyodalmakhoz vezethet. Ha pl. a programot csak a Stop paranccsal lehet majd leállítani, akkor a Run menü felülírása ezt lehetetlenné teszi. A MENÜ **RESET** paranccsal visszaállíthatjuk az eredeti állapotot.

Már tudjuk, hogyan kell a menürendszert létrehozni, most ismerkedjünk meg a menürendszer be- és kikapcsolásával. Erre valók a következő parancsok:

MENÜ ON aktiválja az ON MENÜ GOTO, III. GOSUB utasítást;

MENÜ OFF inaktíválja az ON MENÜ GOTO, ill. GOSUB utasítást;

MENÜ STOP időlegesen felfüggeszti az ON MENÜ utasítások hatását.

Nézzük meg, hogy működik a menüprogramozás a gyakorlatban! A program elején helyezzük el az ON MENÜ GOSUB (vagy GOTO) utasítást! Az ON x GOSUB utasítással ellentétben ezt csak egyszer kell kitenni. A gép tudja, hogy a menüpont kiválasztásakor először mindig ide kell ugrania.

Az egyes menüpontok programon belüli lekérdezéséhez meg kell ismernünk az **y=MENU(x)** függvényt. Az *y* az egérrel kiválasztott menü vagy menüpontot mutatja meg, *x* értékétől függően. Ha *x=0*, akkor a menü számát kapjuk meg, ha *x=1*, akkor a menüpontét. Ha a függvényt összekapcsoljuk az ON y GOSUB utasítással, akkor vezérli az egyes menüpontok kiválasztásával aktivizálandó alprogramokat.

3.14. Vezérlés megszakítással

Gondoljuk át a menüvezérlés általános működési mechanizmusát, hogy a további hasonló utasításrendszereket jobban megérthessük.

A menüpont kiválasztásakor az egérrel beavatkozunk a programba, megszakítjuk azt. A megszakítást a gép értékeli, s a program valahova elágazik. A kulcsszó a megszakítás. A külső beavatkozás hatására a program futása módosul. A menüvezérlés célja a többirányú választási és elágazási lehetőségek beépítése a programba. Milyen hasonló megszakítási lehetőségeket ismer még az Amiga-Basic? Az **ON BREAK GOSUB** címke utasítás működése megegyezik a már ismert ON MENÜ GOSUB címke utasításával. A megszakítást azonban nem a menürendszerből való valamelyik pont kiválasztása hozza létre, hanem a szó szerint vett

megszakítás, ez ugyanis a **BREAK** szó jelentése. Ezt a hatást a Run menüben lévő Stop aktiválása, üi. a CTRL-C vagy az Amiga billentyű lenyomása okozza. Ilyenkor a program a *Címke* alprogramhoz ágazik el.

A rendszer működéséhez tartoznak a MENU-nél megismert be- és kikapcsoló utasítások is, amelyek a következők: **BREAK ON, BREAK OFF, BREAK STOP**.

A megszakításvezérlés következő fajtája az időtől függ. Az **ON TIMER(x) GOSUB** Címke utasítás hatására a program minden x-edik másodpercben megszakad, és a *Címke* alprogramhoz ugrik. Ezt egészítik ki a **TIMER ON, TIMER OFF** és **TIMER STOP** utasítások. Van még egy különleges időmérő függvény is, az **x=TIMER**. Az x megadja a számítógép bekapcsolása óta eltelt másodpercek számát, ha pedig hosszabb munkáról van szó, akkor az éjfél óta eltelt időt.

Az Amiga esetében a megszakítás legjellegzetesebb módja az egérgomb lenyomása. Erre való a következő utasításkészlet: **ON MOUSE GOSUB** Címke, **MOUSE ON, MOUSE OFF, MOUSE STOP**.

A megszakítást a bal oldali egérgomb lenyomása hozza létre. Az egérről az **y=MOUSE(x)** függvény közöl információt. Ha $x=0$, akkor a bal oldali egérgombról van szó. A különböző y értékek a következőket jelentik:

y=0 A bal oldali gombot nem nyomtuk le a **MOUSE(x)** függvény legutolsó hívása óta.

y= 1 A bal oldali gombot egyszer nyomtuk le.

y=2 A bal oldali gombot kétszer nyomtuk le.

y=-1 A bal oldali gomb egyszeri lenyomás után lenyomva maradt.

y=-2 A bal oldali gomb kétszeri lenyomás után lenyomva maradt.

x=1 Az x irányú koordinátát adja meg.

A különböző x értékek jelentése pedig a következő:

x=2 Az y irányú koordinátát adja meg.

x=3 Az x irányú koordinátát adja meg a bal oldali gomb lenyomásakor.

x=4 Az y irányú koordinátát adja meg a bal oldali gomb lenyomásakor.

x=5 Az x irányú koordinátát adja meg a bal oldali gomb lenyomva tartásakor.

x=6 Az y irányú koordinátát adja meg a bal oldali gomb lenyomva tartásakor.

Van egy különleges, nem kívülről jövő megszakítás, amely mindig hibajelenségtől függ. Ha rosszul programoztunk, vagy valamilyen más hiba lép fel a program futása alatt, akkor a program leáll. Egy parancskészlettel azonban beavatkozhatunk, hogy a hibát kiküszöböljük. Ha az **ON ERROR GOTO** Címke parancsot alkalmazzuk, akkor az adott hiba fellépésekor, a hibát jelezve nem áll le a program, hanem a megadott Címfé-hez ugrik, és onnan folytatódik tovább. Az **ERROR** a fellépő hiba kódszámát jelenti. A hibakódszámok listáját és rövid értelmezését az Amiga-Basic kézikönyv függelékében találjuk meg. Nézzünk egy példát, hogyan tudjuk alkalmazni ezt a megszakításvezérlő parancsot! Képzeljük el, hogy készítettünk egy adatbázist. Az adatbázis adatfájlokkal dolgozik, amelyeket létre kell hozni. Ha olyan adatfájlba akarunk adatokat beírni, amelyek nem léteznek, akkor a program megszakad, és hibaüzenetet küld. Ezt azonban az **ON ERROR GOTO** utasítással kivédhetjük, mert a hiba fellépésekor a programot a megfelelő programrészlethez irányítva létrehozhatjuk a fájlt. Az olyan adatbázis-kezelő programokban, amelyekben nem hozzuk létre előre a fájlokat, az első beírási kísérlet során az említett eset mindig bekövetkezik.

Programírás előtt érdemes megvizsgálni, hogy milyen hibák lehetnek, és esetleg tanácsos ezek fellépését a programban figyelembe venni. Mesterségesen létrehozhatunk hibát az **ERROR No** paranccsal, ahol a *No* a hiba kódszáma. Ritkán van rá szükség, hogy valaki hibát idézzon elő saját programjában, ennek a parancsnak nem is ez az elsődleges célja. Segítségével azonban mi is definiálhatunk különleges hibaüzeneteket. Az Amiga-Basic kézikönyv szerint erre a 150 feletti hibakódokat használhatjuk. A példánál maradva, ha azt akarjuk, hogy a telefonszámokat katalogizáló adatbázis figyelmeztessen, ha véletlenül nem hétjegyű telefonszámot írtunk, akkor ezt a következő módon oldhatjuk meg:

```
IF telefonNo<1000000 OR telefonNO>9999999 THEN ERROR 180
IF ERR=180 THEN PRINT"Rossz szám 1!!"
```

Még egy függvény tartozik ide, amelyet bizonyos esetekben felhasználhatunk a hibakezeléskor, az **x=ERL**. Az *x* megadja, hogy az utoljára bekövetkezett hiba hányadik sorban van. Ennek természetesen előfeltétele, hogy a program hagyományosan sorszámozott Basic program legyen. Mikor használhatjuk a függvényt?

Előfordulhat, hogy ugyanazt a hibát a program több része is előidézi, a hibajelzés, ill. a hiba feldolgozása szempontjából azonban nem mindegy, hogy melyik. Ilyenkor az **x=ERL** segítségével tudunk különbséget tenni.

Az utolsó megszakításvezérlés a játékprogramozáshoz kapcsolódik. Az Amiga a Commodore 64-hez hasonlóan alkalmas *sprite-ck* (e. szprájt) létrehozására. A *sprite* olyan különleges kis kép a képernyőn, amely játék közben bizonyos módon mozogni tud. Lehet pl. egy figura, egy kis állat vagy egy úrhajó. A *sprite* speciális fajtája az Amigán használatos *bob*. A kettő közötti eltérésre még visszatérünk. Számunkra most az fontos, hogy az egyes *sprite*-ok, ill. *bobok* a képernyőn ütközhetnek egymással, és ennek a különböző játékokban más-más következménye lehet. Az ütközés következményeit az előbbi megszakításvezérlésekkel megegyezően működő parancsokkal programozhatjuk: **ON COLLISION GOSUB Címke, COLLISION ON, COLLISION OFF, COLLISION STOP**.

3.15. Az Amiga beszél

Rendkívül érdekes, amikor a számítógép az emberi hangot utánozva megszólal. A legtöbb számítógép esetében ez csak bizonyos kiegészítő tartozékokkal vagy bonyolult programozási műveletekkel érhető el. Az Amigát viszont a Basicel azonnal megszólaltathatjuk, és mindehhez nem kell parancsok tucatjait megismernünk.

Az alapparancs a következő:

SAYString,Modus

A *String* maga a kimondandó szöveg, amely közvetlenül vagy változóval egyaránt megadható. A **SAY "hello"** teljesen egyenértékű a **SAY z\$** paranccsal, ha **z\$="hello"**. A *Modus* megadása nem kötelező.

A különböző nyelveknek azonban eltérő a kiejtése. A magyar *e* betűt pl. az angolok a legtöbbször *i*-nek ejtik. Az Amiga ún. fonémakódrendszere egyértelművé teszi a kiejtést. A kódrendszer az Amiga-Basic kézikönyv függelékében található. Ha tehát azt szeretnénk, hogy a számítógép megközelítőleg helyesen ejtse a "jó napot" kifejezést, akkor a táblázatból kikeresve azt kell írunk, hogy **"YOWNAOPOHT"**. Ez ugyan nem túl egyszerű, azonban az emberek nem egy nyelven beszélnek.

Az angolul tudóknak az Amiga következő parancsa kicsit segít:

z\$=TRANSLATE\$Cangol szöveg*)

A paranccsal az idézőjelben lévő angol szöveg megfelelő fonémakódokkal megkonstruált z\$ sztringváltozóját kapjuk, amit a SAY parancs helyesen fog kiejteni. Ha magyar szöveget akarunk kimondatni, akkor a táblázatot felhasználva sajnos kísérletezni kell, vagy megegyező kiejtésű angol szavakat kell keresnünk és a TRANSLATES paranccsal lefordítanunk. Sokszor angol szöveg esetében sem azt kapjuk, amit várunk. A különleges kívánságok (pl. női hang) megvalósítását a *Modus* teszi lehetővé.

A *Modus*nak kilenc funkciója van, s a funkciók egész számokkal adják meg az utasításokat. A *Modus* tehát olyan kilencelemű mező, amelynek egyes elemei mindig egész számok. A példa kedvéért legyen ez z(y) — pontosabban z%(y) —, ahol y=0,1,2,...,8.

- z(0) az alulfrekvenciát határozza meg Hertzben, értéke 65 és 320 közötti;
- z(1) a modulációt határozza meg, értéke 0 vagy 1;
 - 0= normális, hangsúlyos
 - 1 = robotszerű, monoton
- z(2) a beszédsebességet határozza meg szó/perc-ben, értéke 40 és 400 közötti (az alapbeállítás 150);
- z(3) értéke 0 vagy 1;
 - 0 = férfihang
 - 1 = női hang
- z(4) a hang magasságát adja meg, értéke 5000 és 28 000 közötti;
- z(5) a hangerőt adja meg, értéke 0 és 65 közötti;
- z(6) a sztereó csatornák kiosztását adja meg, értéke 0 és 11 közötti (a Basic kézikönyvben részletezik);
- z(7) a szinkronizációt adja meg, értéke 0 vagy 1;
 - 0= a Basic program csak a szöveg után fut tovább
 - 1 = a Basic program azonnal folytatódik
- z(8) az egymás utáni SAY parancsok végrehajtását szabályozza, ha z(7)=1; értéke 0 és 2 közötti;
 - 0= egymás utáni szabályos feldolgozás
 - 1 = a szöveg megszakad
 - 2= az aktuális szöveg megszakad, és a következő jön

3.16. Zene

A beszéd sajátos formája az ének, s innen csak egy lépés a zene. Az Amiga egyik nagy erőssége a zenei képességeiben van. Ennek megfelelően már a Basicen belül is viszonylag könnyen tudunk zenei hatásokat, akár kisebb műveket is létrehozni. Mindössze négy utasítással kell megismerkednünk.

SOUND Frekvencia, Tartam, (Hangerő), Csatorna
 SOUNDWAIT
 SOUND RESUME
 WAVE Csatorna JDefiníció

Az első utasítás a megjelölt paraméterekkel egy hangot hoz létre. A *Frekvenciának* a 20 és 15 000 Hertz közötti tartományba kell esnie. A kis értékek jelentik a mély, a nagyok a magas hangokat. A *Tartam* taktusban adja meg a hang hosszúságát. A megengedett értékek 0 és 77 közöttiek. Egy másodperc 18,2-es értéknek felel meg. A *Hangerő* értékét 0 és 255 között adhatjuk meg. Ez azonban nem kötelező, ilyenkor az érték automatikusan 127. Az Amiga négycsatornás, ezért a *Csatorna* lehetséges értékei; 0, 1, 2, 3. A 0 és a 3 a bal oldali

csatornákat, az 1 és a 2 pedig a jobb oldali csatornákat jelöli. Ha nem adunk értéket, akkor az alapbeállítás 0.

A SOUND WAIT utasítás mindaddig felfüggeszti a SOUND utasítás végrehajtását, amíg a SOUND RESUME utasítást ki nem adjuk.

Az első három utasítással az alapvezérlést tudjuk megvalósítani. A negyedik utasítással a hang hullámfarmáját változtathatjuk meg, alapesetben ugyanis a szinuszhullámot alkalmazza a gép. A WAVE Csatorna. Definíció utasítás *Csatorna* része megegyezik a SOUND utasításban bemutatottal. A *Definíció* vagy a SIN, vagy egy numerikus mező. Ha numerikus mező, akkor 256 elemet tartalmaz, ahol az elemek értékei -128 és 127 közötti egész számok, amelyek amplitúdókat határoznak meg. Ezeknek az összessége határozza meg a hullámfarmát. Kitarató kísérletezéssel érdekes eredményeket kaphatunk.

3.17. Az OBJECT és társai

Az Amiga erőssége a grafika és az animációs, megjelenítőképeség. Az *animáció* képek, figurák mozgatását jelenti. Mivel az Amigát eredetileg játékgépnek szánták, ezért ezt a képességet különösképpen átgondoltan fejlesztették ki. Ezeknek a képességeknek a maximumát azonban nem a Basicból lehet elérni, bár a Basic számtalan utasítása is jól támogatja az animációt. Ezen utasítások mindegyikében szerepel az OBJECT szó.

Az animációt a sprite-tal és a bobbal valósíthatjuk meg. A sprite max. 16 képpont szélességű, tetszőleges magasságú kép vagy figura. A sprite a hardverhez kötődő állandó elem. Max. 8 sprite hozható létre, és egy sprite 4 szint tud használni. Ha több szint szeretnénk, akkor a sprite-ok számának felére csökkentésével max. 16 szint érhetünk el. Hátrányain kívül viszont nagy előnye a sprite-nak (mint minden hardverelemnek) a rendkívüli gyorsasága.

A sprite-tal ellentétben a bob nem hardverhez kötődő, hanem programból vezérelt elem. Ennek megfelelően nincsenek korlátozások, a bob tetszőleges méretű lehet, és használhatja a képernyőn megengedett összes szint. Azonban a bobnak is van hátránya: minél nagyobb, annál lassabban mozgatható a képernyőn.

A következőkben nézzük sorra az egyes utasításokat!

OBJECT.JSHAPE Obj,Def

Az utasítás hatására új tárgy (objektum) jön létre, amely sprite vagy bob lehet. Az *Obj* nullánál nagyobb egész szám. Ez lesz a későbbiekben az adott tárgy jelzőszáma. A *Def* olyan sztring, amely az *Obj* számú tárgy nagyságát, formáját és színét határozza meg.

OBJECT.SHAPE Obj 1,Obj2

Az utasítás hatására új *Obj1* számú tárgy jön létre, amely az *Obj2-től* veszi át tulajdonságait.

OBJECT.XObj\X

OBJECT.YObj.y

A két utasítás a már létrehozott és *Obj* számmal jelzett tárgy *x* vagy *y* irányú mozgatását váltja ki. Az *x* és az *y* mindig a tárgy felső bal oldali koordinátáit jelöli.

OBJECT.VXObj.Seb

OBJECT.VYObj.Seb

Az OBJECT.SHAPE utasítással létrehozott *Obj* számú tárgy sebességét határozhatjuk meg, a *KX*-szel *x*, a *VY*-na pedig *y* irányban. A sebesség mértékegysége: képpont/s.

V=OBJECT.VX(Obj)

V=OBJECT.VY(Obj)

A két függvény az egyes tárgyakor, *ül.* y irányú sebességét adja. Az előző utasításban mi adtuk meg a sebességet, ebben a két utasításban viszont lekérdezzük azt.

X=OBJECT.X (Obj)

Y=OBJECT.Y(Obj)

Igen hasonló ez a két függvény is. Ezek a tárgy pillanatnyi koordinátáit (x, ill. y irányban) adják.

OBJECT.ON Obj 1 ,Obj2,...

A korábbiakban *Obj1*, *Obj2* stb. számmal meghatározott tárgyakat jeleníti meg a képernyőn. Vele ellentétes hatású az OBJECT.OFF Obj 1 ,Obj2,... utasítás.

OBJECT.START Obj 1 ,Obj2,...

A tárgyakat mozgatja. Az *Obj1,Obj2,...* elhagyható. Ebben az esetben az utasítás az összes tárgyra vonatkozik.

OBJECT.STOP Obj 1 ,Obj2,...

Megállítja az előző utasítással mozgatott tárgyakat. Az *Obj1,Obj2,...*-iz ugyanaz vonatkozik, mint az előbb.

OBJECT AX Obj.Gyorsulás

OBJECT AY Obj.Gyorsulás

A létrehozott tárgy x, ill. y irányú gyorsulását határozhatjuk meg. A *Gyorsulás* értékének 1 -nél nagyobbak kell lennie. A gyorsulás mértékegysége: képpont/s.

OBJECT JUT Obj,Saját,Idegen

Ez az utasítás szorosan összefügg az ON COLLISION megszakításvezérléssel. Ebben az esetben a tárgyak (sprite vagy bob) ütközése hozza létre a megszakítást. Az ismert *Obj* paraméter melletti két paraméter a következőket jelenti: a *Saját* egy 1 és 65 535 közötti egész szám, amely saját tárgyunk helyét, az *Idegen* olyan hasonlóan megadandó érték, amely az idegen — az ütközést létrehozó — tárgy helyét adja meg.

OBJECTJPRIORITY ObjJMor

Egymást átlapoló tárgyak esetében megadja, hogy melyik fedi át a másikat. A *Prior* egy -32 768 és 32 767 közötti egész szám, amely egyértelműen eldönti az elsőbbséget (a prioritást).

OBJECTPLANES Obj3bitszint,Szintbe-ki Ez az utasítás a bobokra vonatkozik. Adott Obj számú bob esetében megadja, hogy az melyik *Bitszinten* helyezkedik el. A bitszintek magyarázatául csak annyit, hogy a képernyőn lévő kép és színei egymás felett elhelyezkedő szintekből állnak. Aki már látta, hogyan készül a rajzfilmekben az egyes jelenetek rajza, az könnyen megérti ezt. Ott átlátszó celluloidra elkészítik a képet több részletben, majd azokat pontosan egymásra helyezve jön létre az elképzelt kép. Egészen leegyszerűsítve itt is erről van szó. A *Szintbe-ki* az OBJECT.PLANES utasítás harmadik paramétere, amely a *Bitszint* kiegészítője, és a *Bitszint* által meg nem határozott bitszintekről rendelkezik.

OBJECT.CLJP(x1,y1)-(x2,y2)

A tárgy mozgásterületét határozza meg. A mozgásterületet az *x1,y1* és *x2,y2* pontok által határolt négyszögre korlátozza.

OBJECT.CLOSE Obj 1 ,Obj2,...

Az adott tárgyakat megszünteti, szabaddá téve a definíciókor (OBJECT.SHAPE) lefoglalt memóriaterületet.

A sok utasítás eleinte bonyolultnak tűnik, kis gyakorlással azonban jól használhatók. A munkát jelentősen megkönnyíti az Extrás lemezen található ObjEdit nevű *object editor* program, amellyel sprite-ot és bobot egyaránt létrehozhatunk.

3.18. Ami kimaradt

A Basic hatalmas utasításkészletének nagy részét már átnéztük. Hogy az ismertetés teljes legyen, most azokat az utasításokat nézzük át, amelyek egyik részhez sem tartoznak.

Igen sok munkától mentesíthet minket a **READ** és a **DATA** utasításpáros. Ha bizonyos adatokat, számokat egy Basic programban többször is alkalmazni kell, vagy sok előre megadott azonos jellegű adat szükséges a program működéséhez, akkor ezeket az adatokat a programban a **DATA** utasítással elkülönítve összegyűjthetjük. A **DATA** utasítást mindig csak a sor elején kell kitenni, és az egyes adatokat vesszővel kell egymástól elválasztani. Az adatok vegyesek is lehetnek, mert csak a felhasználáskor kell figyelembe vennünk őket. **DATA** sor lehet pl. a következő: `DATA 100,45,fa,a1,fgxh100,2,3,4,5,7,7,a,f,0a`

A **DATA** sorokat célszerű a program utolsó soraiban elhelyezni, így nem zavarhatják a program futását, hiszen ezek a sorok végrehajtandó utasítást nem tartalmaznak.

Amikor szükségünk van az adatokra, akkor azokat előbb a **READ** (olvas) utasítással be kell olvasnunk. Az első **READ** utasítás megkeresi az első **DATA** sort, majd sorrendben elkezd beolvasni az adatokat úgy, ahogy ezt előírtuk. A **READ** *x, y, z \$* hatására pl. az előbbi **DATA** sorból a *100, a. 45 és az fa* kerül beolvasásra úgy, hogy a **READ** után megadott *xy,z\$* változókhoz rendeljük hozzá őket. A művelet ugyanaz, mintha a következőt írtuk volna:

```
x=100
y=45
z$="fa"
```

Ha a program további részeiben újból alkalmazzuk a **READ** utasítást, akkor az már nem az elején kezdi a beolvasást, hanem ott folytatja, ahol legutoljára abbahagytuk. Tehát esetünkben a következő adat az *al* lenne. Természetesen arra is van mód, hogy a **READ** újból a legelejéről vegye az adatokat, de ekkor előbb a **RESTORE** cím utasítást kell kiadnunk.

Az utasításban a *Cím* megadása nem kötelező, de ha megadjuk, akkor a gép a megfelelő *Címtől* kezdi a beolvasást. A *Cím* lehet sorszám vagy címke. Az adott címkének vagy sorszámnak természetesen **DATA** sort kell jelölnie.

Ügyeljünk arra, hogy mindig csak annyi adatot próbáljunk beolvasatni a **READ** utasítással, amennyi a **DATA** sorokban van. Ellenkező esetben a program megszakad, és az *Out of DATA* hibüzenetet kapjuk.

A Workbench bemutatásakor megismerkedtünk az Amiga rendezőképességével, amivel a programokat ún. *altartalomjegyzékek.be* (subdirectory) tudjuk összefoglalni. Ezek voltak a fiókok. A fiókok is tartalmazhatnak további fiókokat, sőt azok is továbbiakat, szinte a végtelenségig. Ha a Basic programokban ki akarjuk használni ezt a kitűnő lehetőséget (pl. egy könyvelőprogram adatait évenként elkülönítve akarjuk tárolni), akkor az adatbetöltéskor és a mentéskor közölni kell a géppel a pontos "letöltési útvonalat", azaz a munka előtt a gépet a megfelelő altartalomjegyzékhez kell irányítani. Erre való a **CHDIR** utasítás.

A CHDIR utasítás után idézőjelbe téve szerepel a fiók (vagy fiókok) megnevezése, ahol az utasítás kiadása után a munkát folytatni kell.

Tegyük fel, hogy a teljes könyvelőprogramot a KÖNYVELÉS című fiókban helyeztük el, és e fiókon belül külön fiókokban vannak az egyes évek adatai 1987, 1988, 1989 stb. megjelöléssel. Ha az 1988-as adatokra vagyunk kíváncsiak, akkor a betöltés előtt a következő parancsot kell kiadnunk: CHDIR "KÖNYVELÉS/1988"

Van néhány olyan utasítás, amely általában nem a Basic programban fordul elő, hanem ún. közvetlen utasításként használható. A következőkben ezeket soroljuk fel.

DELETEx-y

Az utasítással a Basic program bizonyos sorait törölhetjük. Ha a Basic programot sorszámoztuk, akkor az xy jelölhet sorszámot, címkét, sőt a kettőt egyszerre is. Az $x-y$ megadásakor az x -től y -ig terjedő sorokat törölhetjük. Ha csak az x -et adjuk meg, akkor csak az az egy sor törölődik. Ha DELETE $-x$ -ként adjuk meg, akkor a program elejétől x -ig az összes sor törölődik. Ha azonban fordítva írjuk, DELETE $X-$, akkor az x sortól kezdve a program végéig töröljük a sorokat. Ügyeljünk arra, hogy csak érvényes, létező sorokat jelöljünk meg, ellenkező esetben hibaüzenetet kapunk.

FILES

Az utasítás hatására a Basic ablakban megjelenik a lemez tartalomjegyzéke. Ez alapesetben a DFO, azaz a belső lemez meghajtóban levő lemezre vonatkozik. Ha a külső lemez meghajtóban levő lemezt akarjuk megnézni, akkor a FILES "DF1" ("DF2") ("DF3") . . . utasítást kell kiadnunk.

LIST

A LIST ablakban a programot nemcsak a menürendszer Show List funkciójával, hanem a LIST parancsral is kilistázhatjuk. Ebben az esetben arra is van mód, hogy ne a program elejétől, hanem egy tetszés szerinti helytől induljon a lista. Ilyenkor a LIST után egy sorszámot vagy címkét kell megadnunk.

KILL

A parancsral a lemezen levő valamelyik fájlt törölhetjük. A KILL után idézőjelben a fájl nevét kell megadni, ha nem a főtartalomjegyzékben van, akkor a pontos helyét is, pl. KILL "DF1:KÖNYVELÉS/1988"

MÉRGE

Ezt a parancsot, amellyel két Basic program összemácsolható, a Commodore 64 tulajdonosok jól ismerik a Simón's Basicból. Különösen hosszabb programok írásakor használható. A programot ugyanis különálló kisebb részek formájában írhatjuk meg, javíthatjuk, végül pedig egységes programmá olvaszthatjuk az egyes részeket.

Ha pl. a *program1* és a *program2* programot össze akarjuk másolni oly módon, hogy a *program1* utolsó sora után következzen *&program2* első sora, akkor előbb töltsük be a LOAD utasítással *aprogram1-t*, majd a MÉRGE utasítással másoljuk hozzá *z.program2-t*.

```
LOAD "program1"  
MÉRGE "program2"
```

Nagyon fontos előfeltétele a műveletnek, hogy a hozzámásolandó program, ami esetünkben a *program2*, ASCII formában legyen letöltve. Ez nem a szokványos Basic letöltési mód, ezért a menürendszer Savé vagy Savé As funkciójával ilyen formában nem tudjuk megoldani. A közvetlen letöltés a SAVÉ "program2", a begépelésével oldható meg.

A SAVÉ utasításnál meg kell említenünk annak opciós változatát. Ha programunkat a SAVÉ "progrató", p formában töltjük le, akkor a programot titkosítottuk, vagyis ettől kezdve a programot sem kilistázni, sem megváltoztatni nem tudjuk. Működni természetesen — ha más nehézség nincs — kifogástalanul fog. Ezzel megakadályozhatjuk, hogy illetéktelenek a program kilistázásával az általunk kidolgozott fogásokat ellessék, lemásolják. Tanácsos azonban a titkosítás előtt a magunk részére vagy egy biztonsági (nem titkosított) másolatot készíteni, vagy nyomtatóval a listát kinyomtatni, mert ha később hibát fedezünk fel, vagy újabb ötlet alapján a programot továbbfejlesztjük, csak így lesz rá módunk.

Ez a módszer alkalmas arra is, hogy a programot ne csak az illetéktelen lemásolástól, hanem az illetéktelen használatától is védjük. Ha pl. olyan személyes adatokat viszünk fel, amelyekről nem volna jó, ha illetéktelenek tudomást szereznének, akkor célszerű a program elindítását — ahogy ez a profi programokban is látható — egy *jelszó* ismeretéhez kötni. Nézzük, mit kell tennünk!

```
PRINT "Kérem a jelszót!":a=0
jel:
INPUT jel$
IF jel$="apokalipszis" THEN 110 ELSE a=a+1:GOTO 100
100 IF a>1 THEN END ELSE jel
110 PRINT-A JELSZÓ HELYES"
```

Ehhez hasonló részlettel kell indítanunk a Basic programot. Ha valaki, feltehetőleg tőlünk, tudja az apokalipszis jelszót, akkor minden nehézség nélkül a 110-es sorba kerül, és a program folytatódik. Ha azonban kétszeri próbálkozás után sem helyes a válasz, a program leáll, s mivel a listázás le van tiltva, nem lehet megnézni a jelszót. A kétszeri próbálkozás lehetőségét azért érdemes megadni, mert a jogosult felhasználóval is előfordulhat, hogy a jelszóban véletlenül elüt egy betűt. Túlzás lenne emiatt arra kényszeríteni, hogy újból töltsse be a programot. Aki pedig nem ismeri a jelszót, az biztosan nem jut előnyhöz a kétszeri próbálkozással sem.

SYSTEM

A parancs hatására — akár a programon belül kiadva — ugyanaz történik, mintha a záró pontocskára rákattintanánk, vagy a Quit funkciót választanánk. Ekkor a biztonsági kérdés elmarad, a Basic ablak azonnal bezárul, s egyből a Basic szintre kerülünk.

BEEP

Bizonyára sokszor megfigyeltük, ha valamilyen hiba lép fel, akkor a hibaüzenetet közvetlenül megelőzően a képernyő villan egyet, és egy rövid kis *bip hangot* is hallunk. Ezt a hangjelzést mi is előidézhetjük a BEEP utasítással.

A PRINT utasítás már sokszor előfordult, de egy különleges fajtájáról még nem szóltunk.

PRINT USING •Paraméter*

Az utasítás hatására a *Paraméterrel* meghatározott formában a szöveg vagy a szám kiíródik a képernyőre. Ha a *Paraméter* !, akkor az adott szöveg (sztring) első betűje íródik ki. Ha a *Paraméter* két \ között elhelyezkedő üres jel, pl. \ \, akkor *n* számú üres jel esetén a szöveg 2+H-nyi első betűje jelenik meg. Egyszerűen megoldható számok különféle tizedes formában való megjelenítése a # *Paraméterként* való alkalmazásával (l. a Basic kézikönyvben levő példákat).

A PRINT másik kiegészítője a SPACE\$(n) és az SPC(n). Nézzük a következő példát!

```
PRINT SPACE$(5)
PRINT SPC(5)
```

Mindkét utasítás hatása azonos, öt üres jeltől álló rész kerül a képernyőre. Az SPC-t csak a PRINT utasítással együtt alkalmazhatjuk. Ezzel ellentétben a SPACE\$(n) olyan függvény, amely n üres jeltől álló sztringet definiál, tehát a következőképpen is működik: $x\$=SPACE\(5)

Jól használható utasítás a változók értékének cseréje.

```
a=5:b=10
SWAP a,b
PRINT a
```

A programrészlet hatására a képernyőn a 10-es szám jelenik meg. Eredetileg az $a=5$ volt érvényben, de a SWAP utasítással a két változó értéket cserélt.

Főképp játékprogramokban, de esetleg máshol is használható az $x=RND(y)$ függvény. Az x 0 és 1 közötti véletlen szám lesz, az y elhagyható.

Mivel a számítógép bizonyos algoritmus szerint — tehát nem igazán véletlenszerűen — határozza meg a véletlen számot, ezért ha a programunkban többször előhívjuk a függvényt, akkor azonos véletlen számokat fogunk kapni. Ezen a RANDOMIZE n vagy a RANDOMIZE TIMER utasítás kiadásával segíthetünk.

Ugyancsak a játékprogramokban alkalmazható az $x=STICK(n)$ függvény. Az x a botkormány pozícióját adja meg, a következők szerint:

- $n=0$ az 1. botkormány x koordinátája;
- $n=1$ az 1. botkormány y koordinátája;
- $n=2$ a 2. botkormány x koordinátája;
- $n=3$ a 2. botkormány y koordinátája.

Az előző részekben már volt szó a WINDOW utasításról és a vele létrehozott ablakokról. A programozásban előfordulhat, hogy ismernünk kell néhány információt az ablakról. Ezt teszi lehetővé az $x=WINDOW(y)$ függvény. Az x az y függvényében a következő információkat nyújtja:

- $y=0$ a kiválasztott ablak száma;
- $y=1$ az aktuális ablak száma, ahol a Basic parancsokat végrehajtjuk;
- $y=2$ az aktuális ablak szélessége, képpontokban;
- $y=3$ az aktuális ablak magassága, képpontokban;
- $y=4$ az aktuális ablakon belül az x koordináta, amelynél a következő jel megjelenik;
- $y=5$ ugyanaz, mint az $y=4$, de az y koordinátával;
- $y=6$ a megengedett színek száma;
- $y=7$ az INTUITION ablak memóriacíme;
- $y=8$ a RASTPORT memóriacíme.

Az Amiga-Basic is ismeri a Commodore 64-en gyakori POKE utasítást, amellyel az adott memóriacímen levő értéket át lehet írni. Két különleges változata is van: a POKEL és a POKEW, amelyek egész számokra vonatkozó POKE utasítások. A POKE utasítás ellentettje a PEEK, a PEEKL és a PEEKW. Ezekkel tudjuk lekérdezni az adott memóriacímhez tartozó értéket.

Nézzünk egy gyakorlati alkalmazást, amelyben a POKE és a WINDOW parancsot is kipróbálhatjuk. Bizonyára sokakban felmerült, hogy a Notepadnél használt írásmódokat (dőlt, aláhúzott stb.) jó lenne saját Basic programokban is alkalmazni. Gépeljük be a következő programsort: POKE WINDOW(8)+56,n

Ha n=1, akkor a PRINT utasítás szövege aláhúzott lesz. Ha n=2, akkor vastagon szedett, ha pedig n=4, akkor dőltbetűs. Az n=0 beállítással térhetünk vissza a normál írásmódhoz.

Sokak szerint a Basic-kel csak igen egyszerű feladatok oldhatók meg, a profi felhasználót a Basic nem támogatja. Bár ebben van némi igazság, az Amiga-Basic azért ellenpéldát is szolgáltat. Nézzük meg egy profi rutin egyszerű Basic listáját, az előbbieik alátámasztására.

Az Amigán gyakran kérdezőt alkalmaznak a programozók. Ezt a Basicből is könnyen megvalósíthatjuk, s bármelyik programba beépíthetjük. A kérdező tulajdonképpen a képernyőn megjelenő kisebb-nagyobb ablak. Alapvető tartozéka egy szöveges rész, amely a tulajdonképpeni eldöntendő kérdést fogalmazza meg, ezenkívül két ablak, a lehetséges válaszokra (pl. igen, nem).

```
SCREEN 1,320,256,2,1
WINDOW 2,,,0,1
CLS
PRINT "A kérdező (Requester) bemutatkozik !!!"
FOR i=0 TO 10000 :NEXT i
CALL kérdez ("AKAR BASICBEN PROGRAMOZNI
?", "IGEN", "NEM")
IF ker=1 THEN CLS:PRINT:PRINT
PRINT "Akkor figyelmesen olvassa el ezt"
PRINT " a fejezetet !"
IF ker=2 THEN CLS:PRINT:PRINT
PRINT "Azért olvassa el ezt a fejezetet,"
PRINT "hátha kedvet kap hozzá !"
END

SUB kérdez (a$,al$,a2$) STATIC
SHARED ker
WINDOW 3,"Dönteni kell: ",(50,50)-(LEN(a$)*8+80, 100),0,1
CLS:COLOR 1,3:PAINT (4,4),3:PRINT
PRINT " a$:COLOR 1,2
LINE (10,30)-(50,30),1
LINE (10,45)-(50,45),1
LINE (10,30)-(10,45),1
LINE (50,45)-(50,30),1
PAINT (12,32),2,1
LOCATE 5,3:PRINT al$
LINE (-20+8*LEN(a$),30)-(20+8*LEN(a$),30),1
LINE (-20+8*LEN(a$),45)-(20+8*LEN(a$),45),1
LINE (-20+8*LEN(a$),30)-(-20+8*LEN(a$),45),1
LINE (20+8*LEN(a$),45)-(20+8*LEN(a$),30),1
PAINT (-10+8*LEN(a$),32),2,1
LOCATE 5,LEN(a$)-1:PRINT a2$
hurok:
WHILE MOUSE(0)<>1:WEND
x=MOUSE(1):y=MOUSE(2)
IF y<30 OR y>45 THEN hurok
IF x>10 AND x<50 THEN ker=1:WINDOW CLOSE 3
IF x>-20+8*LEN(a$) AND x<20+8*LEN(a$) THEN ker=2
:WINDOW CLOSE 3
END SUB
```

A programban a már ismert utasítások mellett újak is megjelentek. A profi program elengedhetetlen része az alprogram, amelyet a GOSUB utasítással már tudunk kezelni. Ennek egy sajátos formája a **SUB Név (Változók) STATIC** utasítás. Az adott nevű alprogramot **CALL Név (Változók)** formában bármikor egyszerűen meghívhatjuk. A CALL egyébként elhagyható, elég az alprogram nevét megadni. Egyetlen kivétel, ha a CALL-t az ELSE utasítás után akarjuk alkalmazni, ilyenkor ki kell írni.

A programban a SUB utasítás utáni programrészletet az **END SUB** utasítással zárhatjuk le. Ha az alprogramból annak vége előtt szeretnénk kilépni, azt az EXIT SUB utasítással tehetjük meg.

Az igazi eltérés a GOSUB-os és az előző alprogram között a változók ún. globális vagy lokális megkülönböztetése. A SUB... rendszerű alprogramban az egyes változók függetlenek a főprogramtól. Az alprogramban tehát nyugodtan használhatjuk ugyanazokat a változókat, nem fog zavart okozni. Ha azonban azt akarjuk, hogy egy változó globális legyen, azaz a főprogramban és a SUB... által létrehozott alprogramban is ugyanazt a változót jelentse, akkor ezt a **SHARED** utasítással kell közölnünk. A GOSUB megoldásban minden változó globális, az alprogramnak nincsenek lokális változói.

Nem tartozik sehova az érdekes **x=NSTR(n,x\$,y\$)** függvény sem, amely az x\$ sztringen belül megkeresi az y\$ sztringet, és megmondja, hogy azt az x\$ sztring hányadik eleménél fedezte fel először. Ha az n-et nem adjuk meg, akkor a keresés az x\$ sztring első eleménél kezdődik, ellenkező esetben az n-edik elemtől. Nézzünk egy egyszerű példát!

```
x$="Szeretem az almafa gyümölcsét "  
y$="alma"
```

Ha az n-et elhagyjuk, akkor az x értéke 13, hiszen ha megszámloljuk, akkor az x\$ sztringben a 13. betűtől kezdődik az alma szó. Ha adatbázist építünk fel, kereséskor nagyon jól alkalmazhatjuk a függvényt.

3.19. Library

A Library (e. lājbreri) (könyvtár) a magasabb szintű Basic programozáshoz vezető kulcsszó. E könyv keretei között részletesebben nem foglalkozunk a témával, a könyvtárak használatáról ugyanis bőségesen van irodalom. Az alapdefiníciókat azonban célszerű megismerni, ha ugyanis olyan Basicprogram-listákkal találkozunk, amelyek ilyen tartalmazznak, akkor azok nem lesznek ismeretlenek. Néhány program tüzetes áttanulmányozása után bárki kedvet kaphat, hogy a könyvtárak nyújtotta lehetőségeket maga is megpróbálja kihasználni. Miről is van tehát szó?

A könyvtár kisebb programok vagy inkább rutinok gyűjteménye. Ezek a rutinok azonban gépi kódban vannak, s ezért a Basicen belüli használatukhoz bizonyos előkészületekre van szükség. Egy-egy könyvtár több rutint is tartalmaz, amelyek igen hatékonyak, és bonyolult feladatok is könnyen megoldhatók velük.

A könyvtárakat az Amiga profi programozói készítették, és a Workbench lemezen található a libs nevű alkönyvtárban. Azért nem találkoztunk velük "A Workbench" c. fejezetben, mert nincs ikonjuk. A FILES paranccsal azonban könnyedén kilistáztathatjuk a Workbench tartalmát, s ha megtaláltuk rajta a [libs] jelzésű tartalomjegyzéket, akkor azt a FILES [libs] paranccsal kilistázva a következő könyvtárakat fedezhetjük fel:

- diskfont.library;
- icon.library;
- info.library;
- mathieedoubbas.library;
- mathieedoubtrans.library;
- mathtrans.library;
- translator.library;
- version.library.

A mathieedoubtrans.library csak az 13-as változatban található meg, az 1,2-ben még nem. Ezenkívül még három fontos könyvtár van, a dos.library, az intuition.library és a graphics.library. Ezeket azonban hiába keressük a lemezen, mert szerencsére nem onnan kell őket betölteni. Ezeket az Amiga ROM-jába hardveresen beépítették. Ezekre a könyvtárakra (ül. ezek egyes rutinjaira) a gépnek már a bú toláskor, ül. közvetlenül utána szüksége van.

A könyvtárrutinok gépi kódban írt programok, amelyek Basicben ugyanúgy használhatók, mintha mi magunk írtunk volna egy gépi kódú rutint, s a programban azt alkalmaznánk. A könyvtárakat azonban előbb át kell alakítani, hogy a megfelelő paramétereket a Basicből át tudják venni. Ugyanis nem elég csak aktiválni a könyvtárt, meg kell mondanunk azt is, hogy a könyvtáron belül melyik rutint akarjuk alkalmazni.

A rutin alkalmazása hasonló a nemrég megismert SUB... alprograméhoz. A program elején definiáljuk, hogy melyik könyvtár melyik rutinait akarjuk használni, és ezután már nincs más teendőnk, mint hogy megadjuk a rutin nevét, ill. zárójelben a hozzá tartozó paramétereket. Definiálni a következő Basic parancsokkal tudunk:

```
DECLARE FUNCTION Név (Paraméterek) LIBRARY  
LIBRARY "név.library"
```

Az egyes funkciók nevét és a hozzájuk tartozó paramétereket megtudhatjuk, ha megvizsgáljuk az EXTRÁS lemezen levő fdl.2 vagy fdl.3 tartalomjegyzék fájljait (pl. a DOS Ed editorral kifarjuk az fdl.2-ben levő valamelyik fájlt). A Cd paranccsal válasszuk aktuális tartalomjegyzéknek az fdl.2-t, mert akkor elég csak a fájl nevét (pl: icon_lib.fd) megadni. (Az Ed és a Cd parancsokról a 4. fejezetben szólunk, aki tehát a két parancsot még nem ismeri, egyelőre ne kísérletezzon.) Az fd végződés jelzi, hogy az egyes fájlok definíciói (file definition) ott találhatóak. A LIBRARY parancsban lévő név pedig a könyvtár neve (pl. icon).

Az fdl.2-ben levő fájlokkal az Amiga-Basic nem tud mit kezdeni, ezért különleges formába kell azokat átalakítani. Erre az Extrás lemezen a BasicDemos fiókban levő ConvertFD nevű programot használhatjuk. Eredményül ún. *bmap* fájlokat kapunk (pl. icon.bmap). Ha az így kapott bmap fájlokat használni akarjuk, akkor fontos tudni, hogy a számítógép azokat mindig az aktuális tartalomjegyzékben (vagy a libs tartalomjegyzékben) keresi.

A könyvtárt a **LIBRARY CLOSE** paranccsal zárhatjuk le.

Befejezésül — minden részletesebb magyarázat nélkül — egy egyszerű, a graphics.libraryt alkalmazó rutint ismertetünk, amit bárki könnyen beépíthet saját Basic programjába.

```
LIBRARY "graphics.library"  
CALL FreeSprite(0)
```

A program hatására a nyíl eltűnik a képernyőről, de bármelyik gomb lenyomására vagy az egér elmozdítására újból megjelenik.

4. AZ AMIGA-DOS ÉS A CLI

A DOS a Disk Operating System angol kifejezés rövidítése (magyarul lemezműködtető rendszer). Ilyen DOS rendszerük van más számítógépeknek is. A leghíresebb az IBM és a kompatibilis gépeken futó *MS-DOS*. A Commodore 64 használói talán emlékeznek rá, hogy a Commodore-nak is voltak olyan DOS parancsai, amelyekkel a lemezműveleteket végrehajthatták. A Commodore 64 esetében a DOS parancsok bevitele nem tért el a közvetlen Basic parancsok végrehajtásától, a Basic és a DOS között nem volt éles határ. Az Amiga esetében a Basic és a DOS két különböző rendszer, bár sok feladatot mindkettőből elvégezhetünk. Ebből következik, hogy ma már a DOS nem szigorúan a lemezműveletekhez kapcsolódik, s majd az Amiga-DOS esetében is látni fogjuk, sokkal szélesebb értelemben vett segédeszköz lett belőle.

A DOS közvetítésével sokkal közvetlenebb módon tudunk értekezni a számítógéppel, mint a Workbench-csel, cserében persze elveszítjük azt a kényelmet és eleganciát, amit a Workbench, az intuíció nyújt.

A PC-t használók igen könnyen eligazodnak majd az Amiga-DOS parancsai között, mert azok hasonlítanak az MS-DOS-hoz. Ennek az a magyarázata, hogy az Amiga-DOS nem kizárólag az Amigára kifejlesztett DOS. Hogy az Amiga is rendelkezzen DOS-szal, a Metacomco egy létező DOS rendszert írt át az Amigára. Ennek ellenére ez a DOS igen jól működik, sok parancsot ismer, amelyek között gyakran használt általános, valamint ritkán előforduló különleges, sőt kimondottan bonyolult parancsok is vannak.

A könyv e fejezete terjedelmi okok miatt nem mutatja be az összes parancsot, de megemlíti azokat, amelyek a hatékony DOS műveletekhez feltétlenül szükségesek. Ezeket a parancsokat esetenként részletesebben is megvizsgáljuk, mint ahogy a géphez kapott vékony DOS kézikönyv teszi.

A DOS szintre a *CLI-n* keresztül juthatunk. A CLI — Command Line Interface (parancsinterfész) — kapcsolatot teremt az általunk kiadott DOS parancsok és a számítógép között. Az új (1.3-as) Workbench-ben a CLI helyét átvette a több funkcióval ellátott *Shell*, azonban a Shell is csak a CLI átdolgozott, fejlettebb változata.

Az egyszerűség kedvéért a fejezetben mindig a CLI-t említjük (akinek az 1.3-as változat a rendelkezésére áll, az a Shellt is használhatja a CLI helyett). A két rendszer felfelé kompatibilis, azaz ami a CLI-vel elvégezhető, az a Shell-lel is az, ez természetesen fordítva nem igaz.

Két módon juthatunk a CLI-be. A Workbench-en keresztül kissé bonyolultan érhetjük el. A Workbench betöltése és a lemezablak kinyitása után nyissuk ki a System fiókot, majd aktiváljuk a CLI ikont. Rövid betöltés után új ablak nyílik meg a képernyőn, amely vízszintes irányban teljes egészében kitölti a képernyőt, függőlegesen azonban mindössze 10-15 sor. Természetesen az ismert módon a nekünk tetsző méretre állíthatjuk be az ablakot. Mindjárt feltűnik azonban, hogy a CLI ablaktól nehéz megszabadulni, mert a bal felső sarokban nincs ott a lezárást lehetővé tevő kis pont. Ha az ablakot be akarjuk zárni, hogy a DOS-ból kilépjünk, akkor az ablakba gépeljük be az *Endcli* parancsot, majd nyomjuk le az *Enter*t.

Egyszerűbb módszerrel is elérhetjük a CLI-t, ha a Workbench lemez betöltését a *Ctrl* és a *D* billentyű együttes lenyomásával megszakítjuk. A *BREAK* felirat megjelenése után azonnal a CLI-ben vagyunk. Ebben az esetben a CLI ablak az egész képernyőt betölti. A gyorsaságnak azonban van egy hátránya is, bizonyos betűk egyáltalán nem (vagy nem a helyükön) lesznek. Ha

lenyomjuk az *y*-t, helyette a *z* jelenik meg a képernyőn, a *z* helyén viszont az *y*-t találjuk. Ezen egy parancs begépelésével segíthetünk. Adjuk ki a `System/setmap d` parancsot! Figyelem! Ügyeljünk a felcserélt *z* és *y* betűre. A "/" jelet a "-" jel helyén találjuk. Az Enter lenyomása után rövid töltés következik, majd visszaáll a régi rend: a billentyűk a helyükön lesznek.

Eddig két CLI indítási módról beszéltünk, van azonban egy harmadik is, amely tulajdonképpen az első módszer hatalmas lehetőségeket magába rejtő változata. Gépeljük be a CLI ablakba a `Newcli` parancsot. Röviddel ezután az elsőhöz hasonló újabb CLI ablak nyílik. Folytassuk tovább a `Newcli` parancsok begépelését. Újabb és újabb ablakok nyílnak majd mindaddig, amíg a számítógép memóriája elfogy (a 3. fejezetben láttuk, hogy az ablakok sok memóriahelyet igényelnek). Az egyes ablakok a későbbiek során lehetővé teszik, hogy egymással párhuzamosan, de egymástól függetlenül több parancsot (vagy programot) is futtassunk az Amigán. (Ez `amultitasking`.)

Észre sem vettük, s máris alkalmaztunk három DOS parancsot (`Endcli`, `Newcli`, `Setmap`). A DOS működésekor az aktuális sorba begépelünk valamilyen parancsot, s az Enter lenyomása után a gép azt végrehajtja. Kiindulásként elfogadhatjuk, hogy a DOS-szal ugyanúgy dolgozhatunk, mint amikor Basicben közvetlenül adtunk ki egy parancsot. Természetesen itt is van rá mód, hogy több parancsot kisebb programmá foglaljunk össze, s azt folyamatosan hajtassuk végre a géppel. Az ilyen programot *batch* fájbak nevezzük.

A parancsok kiadása után a `DRTVE` feliratú lámpa kigyullad, s a parancsot a gép csak rövid betöltés után hajtja végre. Ez nem véletlen, ugyanis az egyes DOS parancsok tulajdonképpen kis programok, amelyek nem a számítógépben vannak, hanem a `Workbench` lemezen. Ha a `Workbench` lemez helyett másik lemezt teszünk be, hiába vagyunk már a CLI-ben, bármelyik DOS parancs kiadása után az *Unknown command* (ismeretlen parancs) hibaüzenetet kapjuk, a gép ugyanis nem találja a programot. Ha a DOS műveleteket nem a `Workbench` lemezen (ugye senki nem az eredeti lemezt használja!?) hajtjuk végre, s a gépen csak egy lemezmeghajtó van, akkor nehézségeink lehetnek. Ugyanis ha a `Workbench` lemez van a meghajtóban, akkor a parancs végrehajtásáért felelős program nem hiányzik majd, hibaüzenetet sem kapunk, de a parancs a `Workbench`-en hajtódik végTe. Ha viszont a parancs kiadásakor a munkalemez van a gépben, akkor az előbbieken bemutatott hibaüzenetet kapjuk. A feladat második lemezmeghajtó nélkül is megoldható, később még visszatérünk rá. Először azonban ismerkedjünk meg a DOS parancsokkal!

4. 1.ADiréstársai

Gyakran használt parancs a `Dir`, amellyel a lemez tartalomjegyzékét kapjuk meg. Adjuk ki a parancsot, s várjuk meg, amíg a `Workbench` tartalomjegyzéke megjelenik a képernyőn. Jó néhány ismerős névvel fogunk találkozni. A `Workbench`-ről ismert ikonok (programok és fiókok) neveit fogjuk felfedezni. A képernyőn új nevek is lesznek. Némelyik név mögött zárójelben a `Dir` felirat szerepel, jelezve, hogy az adott név nem fájl, hanem tartalomjegyzék. Ha kíváncsiak vagyunk, hogy mit tartalmaznak a további tartalomjegyzékek, akkor a `Dir` parancs után adjuk meg, melyik tartalomjegyzék érdekel bennünket. Ha semmit nem adunk meg, akkor mindig az aktuális tartalomjegyzék listázódik ki, ebben az esetben a lemez.

Adjuk ki a `Dir devs` parancsot! Olyan listát kapunk, amelyben fájlok és tartalomjegyzékek vannak. Ezután adjuk ki a `Dir keymaps` parancsot! A várt tartalomjegyzék helyett a *Keymaps notfound* (a `keymaps` nem található) hibaüzenetet kapjuk. Ezek után jobban érthető, hogy a `Dir` az aktuális tartalomjegyzéken belül is tud működni. Nézzük meg tehát, hogyan kellett volna helyesen megadni:

```
Dir devs/keymaps
```


A Dir parancs után először azt kell megadnunk, hogy a listázandó tartalomjegyzék (*keymaps*) a *devs* tartalomjegyzékben található. A keresett objektum akár több tartalomjegyzéknyi mélységben is lehet. Ilyenkor pontosan meg kell adni a teljes utat, az egyes tartalomjegyzékeket a / jellel elválasztva. A *devs* tartalomjegyzéken belül látunk más tartalomjegyzékeket is. Ha el akarjuk kerülni, hogy a pontos utat mindig meg kelljen adni, akkor a *keymapset* aktuális tartalomjegyzékké kell változtatni. Erre való a CD parancs.

A CD *devs/keymaps* parancs kiadása után már nem kell megadni a pontos utat, elegendő a DIRT kiadni. Ezt a parancsot érdemes megjegyezni, mert később is előfordulhat, hogy az a fájl, amivel a CLI-ben dolgozni akarunk, nem a lemez főtartalomjegyzékében, hanem mélyebben található. Ilyenkor körülményes lenne minden parancs előtt a pontos utat is megjelölni. Ha egy ilyen munka végeztével vissza akarunk térni az előző (eggyel feljebb elhelyezkedő) vagy a főtartalomjegyzékbe, akkor a következőképpen alkalmazzuk a parancsot:

```
CD /vagy
```

```
CD //
```

A Dir parancsnak is vannak opciói, ezekkel sok feladat megoldható. Ha a lemez teljes tartalomjegyzékére vagyunk kíváncsiak, de nem akarjuk az altartalomjegyzékeket egyenként (ahogy következnek) kilistáztatni, akkor alkalmazzuk a *Dir opt* a parancsot. A főtartalomjegyzéken kívül tartalmával együtt az összes altartalomjegyzék is megjelenik. Természetesen szűkítve is kiadhatjuk a parancsot:

```
Dir dev opt a
```

Ekkor csak a *devs* tartalomjegyzék és az azon belüli további tartalomjegyzékek listáznak ki. A főtartalomjegyzék System tartalomjegyzéke pl. itt már nem jelentkezik.

Hogyan nézhetjük meg egy másik lemez tartalomjegyzékét? Akinek van külső lemezmeghajtója, az tegye oda a lemezt. A Workbench lemezt természetesen hagyjuk a belső lemezmeghajtóban. A Dir parancs a következő:

```
Dir dfl:
```

A kettőspontot ne felejtjük el, mert különben hibaüzenetet kapunk.

Akinek nincs külső lemezmeghajtója, az egy kis trükkre kényszerül. Adjuk ki a

```
Dir df0:+
```

parancsot úgy, hogy a Workbench lemez legyen bent. Ilyenkor a Dir parancs csak betöltődik a Workbench lemezről, de a végrehajtás várat magára. Cseréljük most ki a Workbench lemezt a megvizsgálandó lemezre, és nyomjuk meg az Entert. A kért lemez tartalomjegyzékét kapjuk.

Van egy másik trükk is, a *Dir ?*. A kérdőjel hatására a Workbench-ről ugyancsak betöltődik a parancs anélkül, hogy azt a gép a bent lévő (azaz a Workbench) lemezen mindjárt végrehajtaná. Sőt, egy sor is megjelenik a képernyőn, a *DIR,OPT/K:.* (A különböző Workbench változatok miatt lehet eltérés.) A megjelent sorból megtudhatjuk, hogy a Dir parancs vagy önmagában áll, vagy az *OPT* (mindegy, hogy kis- vagy nagybetűvel írjuk) opciók valamelyikével kiegészítve. Ez azonban most nem fontos.

Cseréljük ki ismét a Workbench lemezt a kért lemezre, és írjuk be azt, hogy *df0:.* A kért tartalomjegyzéket kapjuk.

Térjünk vissza a *?-es* megoldásra. A kérdőjel nemcsak a Dir parancsban használható, hanem az összes ezután következőben. Szakkönyv nélkül is megtudhatjuk az adott parancs írásmódját (syntax) és kiegészítőit.

A kis kitérő után nézzük ismét a Dir opcióit, és próbáljuk ki a következőt: `Dir opt i`

A parancs hatására a Workbench (ha az van bent) tartalomjegyzékét kapjuk, de az első tétel megjelenése után leáll a gép, és a tétel után egy kérdőjelet is látunk: `c (dir) ?` Ekkor a következőket tehetjük.

Ha lenyomjuk az Entert, akkor a következő tétel jelenik meg, a mi esetünkben a Prefs (dir). Ha azonban eggyel mélyebbre akarunk menni, hogy a `c (dir)` után ne a Prefs (dir) következzen, hanem a `c` tartalomjegyzék tételei, akkor írjuk be az `e` betűt. Ha később vissza akarunk térni a felsőbb szintre, akkor a `b` betűt kell kiadni. Így mozoghatunk a tartalomjegyzékben le és fel.

Ha nem a tartalomjegyzéknél, azaz a (dir) kiegészítést tartalmazó tételnél vagyunk, akkor a kérdőjelre a `d e l` választ adva, az adott fájlt töröljük.

Az interaktív kérdőjeles eljárásból a `q` betű beadásával bármikor kiléphetünk.

Az opciók kombinálhatók is: `Dir opt ai`. Ekkor a részletes interaktív tartalomjegyzék következik. A `b` és `e` utasításnak nincs értelme, mert minden megjelenik.

A `Dir` parancs következő opciójával csak a tartalomjegyzékeket jelezhetjük ki: `Dir opt d` vagy `Dir opt davag` `Dir opt di`.

A `Dir` parancs úgy is kiadható, hogy csak az aktuális tartalomjegyzékben található fájlokat jelzi ki, és kihagyja a tartalomjegyzékeket (ez az opció csak az 1.3-as változatban van meg):

`Dir files`

Ha nem nyitottuk elég nagyra a CLI ablakot, akkor a listázást nem tudjuk leállítani, mielőtt kifutna az ablakból. Erre később is szükség lesz, pl. a szövegfájlok kiírásakor, ha az nem fér el a képernyőn. Ilyenkor nyomjuk le a Space billentyűt. A listázás megszakad (természetesen nem végleg), a Backspace billentyű lenyomásával minden folytatódik tovább. A jobb oldali egérgomb lenyomásával ugyanezt a hatást érhetjük el. Amíg a gombot nyomva tartjuk, szünetel a listázás. A gomb elengedésekor minden folytatódik tovább.

A `Dir` parancssal való ismerkedést fejezzük be a `c (dir)` tartalomjegyzék kilistázásával, itt vannak ugyanis a DOS parancsok. Figyelmesen olvasva könnyen felfedezhetjük a `Dir` parancsot is.

4.2. A CLI és a Shell

Mielőtt részletesebben ismertetnénk a DOS parancsokat, azoknak a kedvéért, akik az 1.3-as Workbench változattal dolgoznak, röviden felsoroljuk a Shell előnyeit, hogy azt használhassák a CLI helyett.

Az első különbség a promptnál mutatkozik, amely a CLI betöltésekor a CLI ablakban megjelenő `1>` jel. Ez a Shell esetében módosul. Az `1.SYS:>` már nemcsak azt jelzi, hogy hányadik CLI (Shell) ablakról van szó (erre utal a szám), hanem azt is, hogy az aktuális tartalomjegyzék (amelyben tartózkodunk) a `SYS` (a rendszer), hiszen a rendszerlemez (a Workbench-et) töltöttük be. Ha kiadjuk a `CD df0:` parancsot, akkor a CLI-ben a prompt változatlan marad, a Shellben viszont a lemez neve is megjelenik a promptban: `1.Workbench 1.3:>`

Sőt, ha mondjuk a `CD devs/keymaps` parancsot adjuk ki, akkor a Shellben megjelenik az aktuális tartalomjegyzék, de a hozzá vezető teljes úttal: `1 .Workbench 1.3 :devs/keymaps>`

Ez a szolgáltatás hasznos lehet, ha összevissza kalandozunk a lemezen.

A Shell másik előnye szerkesztési (editálási) képességében van. A Shellbe egy *soreditort* is beépítettek. A CLI-ben ha egy betűt elírtunk, akkor a kijavításához a betű mögött álló összes többi szöveget a Backspace-szel törölni kell. A Shellben használhatjuk a kurzorbillentyűket, csak a hibás betűt kell törölni, javítani.

Ha egy sort hibásan leírva már leütöttük az Entert, és megjelent a hibaiüzenet, akkor a CLI-ben sajnos a hiba kijavítása után az egész sort újból le kell írunk. A Shellben viszont: ha lenyomjuk a kurzor fel billentyűt, akkor a hibás sor azonnal megjelenik, és az editorral könnyen kijavíthatjuk a hibás részt.

A Shellben a tévesen bevitt sort a Ctrl+B billentyűk lenyomásával törölhetjük. A Ctrl+A billentyűkombinációval a sor elejére ugorhatunk. A Shellt az Endcli paranccsal zárhatjuk le, de új parancs is létezik, az Endshell.

4.2.1. A dzsóker

A Commodore programozói jól ismerik a *-ot és a ?-et, a dzsókereket. Az Amigának is van ilyen segédeszköze, a # és a ?. A ? bármilyen betűt helyettesíthet, a # egy egész, tetszőleges hosszúságú karaktersort. Nézzünk néhány példát!

```
Dir s#?
Dir ??a#?
```

Az első esetben a tartalomjegyzékből azokat a tételeket jelezzük ki, amelyek s betűvel kezdődnek. A második esetben pedig minden olyan tétel megjelenik, amelynek harmadik betűje a.

4.2.2. A másolás módozatai

E könyvben első műveletként másolatot készítettünk a Workbench lemezről. Ezt a feladatot a CLI-ben a Diskcopy paranccsal oldhatjuk meg. A parancsot a következőképpen kell kiadni:
Diskcopy df0: to df1: (Név)

Ebben az esetben a belső meghajtóban levő Workbench a külső meghajtóban levő lemezre másolódik, és a másolat az általunk megadott nevet viseli. A névmegadást elhagyhatjuk, ilyenkor a másolat neve megegyezik az eredetivel. Ha később mégis úgy döntenénk, hogy más néven szerepeljen a másolat, akkor a Relabel df1: Név paranccsal átnevezhetjük.

A Diskcopy parancsot egy lemezmeghajtó esetén alkalmazva a df1: helyett is df0:-t kell írunk, s legalább háromszori lemezcserével kell számolnunk.

A Diskcopy parancsnál sokkal érdekesebb a Copy parancs, amellyel fájlokat tudunk másolni. Ehhez azonban üres lemez vagy üres helyvel rendelkező lemez kell. Az üres lemez nem egyenlő a gyári új lemezzel, mert azt előbb a Formát paranccsal meg kell formázni.

A Formát DRIVE df0: NAME Név (NOICONS) egy lemezmeghajtóval is egyszerűen működik. A parancs először betöltődik a Workbench lemezről. Ha nem az van a meghajtóban, akkor a gép a kérdezővel felszólít rá, hogy tegyük be. A formázás nem indul be azonnal, lesz időnk lemezt cserélni. Erre külön felszólít a gép. Az előző parancs hatására ezután a belső lemezmeghajtóban levő lemezt formázza meg a gép. Ennek a lemeznek Név lesz a neve, de a megszokott szemétkosár nem lesz a lemezen, a CLI-ben nincs rá szükség. Ha mégis szeretnénk egy szemétkosarat, akkor hagyjuk el a NOICONS utasítást.

Ha nem a belső meghajtóban levő lemezt formázzuk, akkor a df0: helyett a kért lemezmeghajtó címét adjuk meg.

Előfordulhat, hogy nem egyszótagú szöveget választunk lemeznévnek. Ilyenkor, ha azt akarjuk, hogy ne egy kioktató *Usage: formát DRIVE <disk> NAME <name> [NOICONS]* felirat jelenjen meg, akkor a névnek szánt szöveget tegyük idézőjelbe:

```
formát DRIVE dfO: NAME "Ez itt a lemez neve"
```

Az idézőjeles megoldásra a későbbiekben is mindig szükség lesz, ha egy parancsban használt név nem egy szótagból áll. Az idézőjel elmulasztása téves parancsvégrehajtást okoz, vagy hibüzenetet (pontosabban *Usage: instrúció-t*) kapunk. (A Formát és a Diskcopy parancs egyébként nem a c, hanem a System tartalomjegyzékben van.)

Nézzük meg a fájlmásolás műveletét. A szükséges parancs a Copy. Természetesen itt is több variáció létezik, és a legkönnyebben példákkal ismerkedhetünk meg velük.

Copy devs/keymaps/d to c

A parancs két részből áll. Az egyik megmondja, hogy honnan és mit másolunk, a másik pedig megadja, hogy hova. Ebben az esetben a *devs* tartalomjegyzékben levő *keymaps* tartalomjegyzékből a *d* nevű fájlt másoljuk a *c* tartalomjegyzékbe. A *to* szócska jelzi, hogy hova, de ez elhagyható. A parancs egyik érdekessége, hogy lemezen belül is lehet fájlokat másolni, a feltétel, hogy a honnan és a hova más-más tartalomjegyzék legyen.

Copyc/Dirto dfO:

A Dir parancsot másoljuk a *c* tartalomjegyzékből, egy szinttel feljebb, azaz a lemez főtartalomjegyzékébe, amit a meghajtó nevével (dfO:) jelölünk.

Copy c/Dir to df 1:Tartalom

Ez a művelet két lemez között megy végbe. A Dir parancsot megtestesítő fájlt másoljuk a belső lemezmeghajtóban levő lemezről. Kitehettük volna itt is a dfO: megjelölést, de a honnan megadásakor — ha az a dfO: — nem szükséges. A másolás célpontja a *df1*: meghajtóban levő lemez. Az előzőekkel ellentétben — amikor az átmásolt fájl neve megegyezett az eredetivel — itt a másolás után a név is megváltozik, és a fájl neve Tartalom lesz. Ha azonban a df 1: lemezen a másolás előtt létrehoztunk volna egy *Tartalom* nevű tartalomjegyzéket, akkor más eredménye lett volna a másolásnak. A Dir parancs — nevét megtartva — bemásolódott volna ebbe a Tartalom nevű tartalomjegyzékbe. A gép ugyanis a másolás előtt pontosan megvizsgálja a célt, és aszerint dönt, hogy mit talál. Ha a végcél egy létező tartalomjegyzék, akkor oda bemásolja az adott fájlt.

Copy s to ram:

Nemcsak fájlokat tudunk a Copy parancssal másolni, hanem teljes tartalomjegyzékeket is. A példában a Workbench tartalomjegyzékét átmásoljuk. Szokatlan, hogy itt nem lemezre másolunk, a célállomás a számítógép memóriája, a RAM, amely azonban pontosan úgy működik, mint egy lemez.

A próba kedvéért hajtsuk végre a parancsot. Észre fogjuk venni, hogy az *s* tartalomjegyzéket nem fájlként másolja át, és erről a számítógép tájékoztat minket. Folyamatosan kiírja a képernyőre az éppen átmásolt fájl nevét. Ez persze kicsit lassítja a másolást, ezért nagyobb, sok fájlt tartalmazó tartalomjegyzékek másolásakor célszerű egy kiegészítővel ellátni a Copy parancsot, s máris kijelzés nélkül dolgozhatunk. Próbáljuk ki ezt is!

Copy l to ram: quiet

Ha ezek után megvizsgáljuk a *ram*: tartalomjegyzéket, akkor megtaláljuk az általunk átmásolt fájlokat, azonban nem tartalomjegyzékbe összefoglalva, mint ahogy az eredeti Workbench lemezen voltak. Ha azt akarjuk, hogy a parancs az azonos (vagy tetszés szerint valamilyen más)

nevű tartalomjegyzékbe összefoglalva másoljon, akkor a másolás előtt létre kell hozni a tartalomjegyzéket. Nézzük a következő példát!

Másoljunk át a *c* tartalomjegyzékből két fájlt a *ram* lemeze, de ott is legyen *c* tartalomjegyzék.

```
Makedir ram:c  
Copy c/dir ram:c  
Copy c/copy ram:c
```

A *ram* lemezen a *Makedir* parancs hozza létre a *c* tartalomjegyzéket.

4.3. DOS parancsok egy lemezmeghajtóval

Ha kísérleteztünk már DOS parancsokkal — de nem a *Workbench*-en —, akkor bizonyára problémát okozott, hogy a gép állandóan bekéri a *Workbench*-et. Ezt az előző két parancsral (Copy, ill. *Makedir*) és a *ram* lemezzel tudjuk kiküszöbölni.

A *Copy c ram:c quiet* parancs az összes DOS parancsot átmásolja a *ram* lemeze, ugyanis ha kiadunk egy DOS parancsot, akkor a gép nemcsak a belső lemezmeghajtóban levő lemezen keresi, hanem megvizsgálja a *ram* lemezt is. Ha tehát itt megtalálja a keresett DOS utasítást, akkor azt végre is hajtja anélkül, hogy szükség lenne a *Workbench* lemeze.

Ha még elegánsabban akarjuk megoldani a feladatot, akkor két sort hozzá kell írunk a programhoz, így a *ram* lemezen is a *c* tartalomjegyzékben lesznek a parancsok.

```
Makedir ram:c  
Copy c ram:c  
Path ram:c Add
```

Az első két sor ismerős, a harmadik viszont új parancs. Erre azért van szükség, hogy a gép ne csak a *ram* lemez főtartalomjegyzékében keressen, hanem vizsgálja meg a *c* tartalomjegyzéket is. A DOS parancsokat külön utasítás nélkül ugyanis csak a rendszerlemez esetében keresi a főtartalomjegyzéken kívül a *c*-ben is. A *Path* (e. pász) parancs megmutatja, hol kell keresni.

Mivel a bútoláskor néhány *Path* parancsot már kiadtunk, ezért újabb utat jelölünk meg. Ez a parancs végén kötelezően megjelenőidig (hozzáadás).

A *Path Show* parancsral megnézhetjük, hogy milyen utakat jelöltünk ki. Ha valamelyik utat meg akarjuk szüntetni, akkor a *Path névReset* parancsot kell kiadni.

Térjünk vissza a DOS parancsok *ram* lemeze másolásához. A feladatot megoldottuk, de nagyon sok parancsot kellett átmásolni. Ez sokáig tart, és mintegy 200—250 kb-át foglal. Ezért csak a legfontosabb parancsokatl kell átmásolni, s ha később egyszer-egyszer olyanra lesz szükség, amit nem másoltunk át, akkor azt bármikor átmásolhatjuk.

Nehéz tanácsot adni, melyek a legfontosabb parancsok, a következő lista azonban segít eligazodni. *CD*, *Copy*, *Delete*, *Dir*, *Echo*, *Ed*, *EndCLI*, *Execute*, *Info*, *List*, *Makedir*, *Mount*, *NewCLI*, *Path*, *Relabel*, *Rename*, *Resident*, *Run*, *Type*, *Why*. Így egyenként kell a parancsokatl átmásolni, nem alkalmazhatjuk az előbbi kényelmes megoldást, a *c* tartalomjegyzék teljes másolását. Aki rendszeresen akar a *CLI*-ben dolgozni, annak szüksége van ezekre a parancsokra, ezért előbb utóbb saját *CLI* lemezt hoz létre magának. Ennek egyik jellemzője, hogy a lemez — ellentétben a *Workbench* lemezzel — egyből a *CLI*-be tölt be, s a *Workbench*-et egyáltalán nem (vagy csak különleges betöltéssel) használja. Ilyenkor egyszerűen megoldhatjuk az általunk fontosnak tartott DOS parancsok *ram* lemeze töltését. A DOS parancsokatl automatikusan a rendszer fogja a *ram* lemezeze másolni.

4.4. Startup-sequence

A megoldáshoz a kulcsszó a *Startup-sequence* (e. sztártáp szikvensz), az a különleges DOS program, amely a betöltést vezérli, vagyis meghatározza, hogy a betöltéskor mi minden történjen. A Startup-sequence a batch fájlok sajátos alfaja. Ahhoz, hogy ilyen programot írjunk (ill. a már megírt programot megvizsgáljuk és esetleg módosítsuk), szükségünk van egy editorra. A DOS-ban kettő is van belőle, az egyik az Edit, a másik az Ed.

Az Ed használata viszonylag egyszerű, a következőképpen aktiválható: Ed név. A *név* a szerkesztendő programfájl neve. Ha a lemezen nem létező nevet adunk meg, akkor a *Creating new file* (új fájl létrehozása) felirat jelenik meg, s egy teljes, üres ablak nyílik, ahol azonnal elkezdhetjük az új fájl szerkesztését.

A mi esetünkben azonban nem egy új, hanem a már meglévő fájlról van szó. Ennek a mindenkori helye az s tartalomjegyzékben van, ezért az editort a következőképpen indítjuk:

```
Ed s/startup-sequence
```

A képernyőn egy hosszabb-rövidebb program listáját kapjuk, attól függően, hogy a Workbench 1.2-es vagy 1.3-as változatával dolgozunk. Anélkül, hogy a program egyes utasításait most részletesen elemeznénk, vegyük észre, hogy itt minden sor DOS utasítás, a pontosvessző utáni kommentárral. A pontosvesszőnek itt ugyanaz a szerepe, mint a Basic programokban a REM utasításnak.

A kurzor a bal felső sarokban található, és a kurzormozgató billentyűkkel mozgatható. Ha a szerkesztésből bármikor ki akarunk lépni anélkül, hogy a létrehozott változásokat letöltenénk (azaz meghagyjuk az eredetit), akkor az Esc billentyű lenyomása után — ilyenkor a bal alsó sarokban megjelenik egy * — a *q-i* kell lenyomni, majd az Entert. Még egy biztonsági kérdésre kell válaszolnunk:

```
Edits will be lost - type Y to confirm :
```

Nyomjuk le az y-t, de ne hagyjuk el a Startup-sequence fájlt, hiszen még nem építettük be a DOS parancsokat másoló fájlt!

Menjünk a kurzorral (az 1.3-as verzióban) a failat 11 sorig úgy, hogy a kurzor a 11 mögött álljon. Nyomjuk le az Entert! Egy új üres sort kapunk, ide kezdjük begépelni a programrészletet:

```
Makedir ram:c
Copy c/CD ram:c
Copy c/Copy ramtc
-
-
Path ram:c Add
```

Ha egy sort véletlenül elrontottunk, akkor az adott sorba bárhova visszaállítva a kurzort, a Ctrl és a B billentyű együttes lenyomásával a sort törölhetjük.

Ha sikeresen beírtuk a programrészletet, akkor nyomjuk le az Esc billentyűt, majd a bal alsó sarokban megjelenő * után nyomjuk le az x-et és végül az Entert. Ezzel most már letöltöttük a módosított Startup-sequencet.

Ha a Ctrl+A+A billentyűkkel reszetálunk, akkor a bútolás után egy meghajtó esetében sem lesz több problémánk a DOS parancsok kiadásakor. Később még az így létrehozott Startup-sequencen is finomítunk majd.

Az Ed editornak van még egy opcionális kiegészítője, amely az editorral lefoglalt memóriaterületre vonatkozik. Akkor kell használnunk, ha különösen hosszú szöveget vagy programot akarunk szerkeszteni, azaz ha a szerkesztendő fájl mérete 40 000 bájtól nagyobb. Ha kb. 50 000 bájt méretű fájlban szeretnénk dolgozni, akkor az editort a következőképpen kell indítani:

```
Ed fájl név SIZE 50000
```

A DOS parancsok átmásolását másképpen is megoldhatjuk. Írhatunk külön programot, batch fájlt, amelyet tetszés szerint aktiválhatunk, lefuttathatunk. Tehát a Startup-sequence-be nem avatkozunk be, a parancsokat nem a bútoláskor, hanem a batch fájl lefuttatásakor visszük fel a jam lemezre. Ezt a batch fájlt is az Ed editorral írhatjuk meg, s tartalma azonos avval, amit az előbb beszerkesztettünk a Startup-sequence-be. Tegyük fel, hogy a batch fájlunk a *dosram* nevet adtuk. A fájlt a következőképpen futtathatjuk le:

```
Execute dosram
```

Ha a *dosram* név később nem tetszik, akkor a Rename paranccsal bármikor átnevezhetjük ezt a fájlt (vagy bármelyik másikat). A Rename *dosram To dosparancsok* kiadása után az új név *dosparancsok* lesz.

Ha már nincs szükség valamelyik fájlra, akkor letörölhetjük a lemezről. A *dosparancsok* fájlt a Delete *dosparancsok* beírásával törölhetjük. Egyszerre több fájlt is törölhetünk. Ha egy tartalomjegyzék teljes tartalmát törölni akarjuk, akkor írjuk az *AH* kiegészítőt a parancshoz:

```
Delete c/#? Ali
```

Hatására a *c* tartalomjegyzékben található összes DOS parancs törlődik. Ha ezt végrehajtjuk, akkor a Copy paranccshoz hasonlóan, a számítógép itt is minden fájl törlése után kiírja a fájl nevét, s azt, hogy törölve van (deleted). Ha ezt nem kérjük, a végrehajtást is gyorsítva, a Delete *c/#? Ali Quiet* sort írjuk be. A *Quiet* helyett *Q* betűt is írhatunk.

A Delete parancs fordítottja az Undelete, ez nem a Workbench lemezen, hanem az RPD-103 Public Domain lemezen van. Erről részletesebben a 6. fejezetben szólnunk. Az utasítás egy véletlenül törölt fájlt meg tud menteni, ha a lemezt még nem formáztuk meg, és nem vettünk fel rá más fájlt.

A Delete parancs nemcsak fájlokat, hanem a tartalomjegyzékeket is törli, de csak akkor, ha a tartalomjegyzékben már nincs egy fájl sem. (Tehát az ott lévő fájlokat előbb törölni kell.) Megeshet, hogy ennek ellenére sem sikerül a tartalomjegyzéket törölni, és a *File in use* (a fájlal dolgoznak) üzenetet kapjuk. Ennek az az oka, hogy korábban esetleg a Path paranccsal az adott tartalomjegyzéket is keresési útnak neveztük ki. Ekkor a törlés előtt a Path Reset alkalmazásával ezt meg kell szüntetni.

Jó néhány műveletet megismertünk, de szinte mindegyik a másolással vagy a törléssel kapcsolatos. Később is így lesz, azaz a lemezen lévő fájlok rendszeresen változnak majd. Aki sokat töröl és sokat másol, az egy idő után észreveszi, hogy az egyes fájlok betöltéséhez és a bútoláshoz is több időre lesz szüksége, mint azt korábban megszokta. A sok másolás és törlés következtében ugyanis a lemezen korábban meglévő rend felborul, és a DOS az egyes programokat már talán nem egy helyre, hanem a lemezen több részre elosztva helyezi el. Emiatt csökken a sebesség.

Mit lehet ez ellen tenni? A közmondás szerint kutyaharapást szőrével. Először tehát formáznunk kell egy üres lemezt, majd a régít fájlunként a Copy paranccsal át kell másolnunk. Az üres lemezen ugyanis a DOS ismét a lehető legtömörebben fogja elhelyezni a fájlokat.

Ha a lemezünk nem tartalmaz altartalomjegyzékeket, akkor egyszerű dolgunk van, a másolás egy sorral megoldható:

```
Copy df0: to df1: Ali
```

Ha az altartalomjegyzékeket is át akarjuk menteni, akkor előbb a Makedir paranccsal létrehozzuk azokat, majd az egyes tartalomjegyzékek tartalmát tartalomjegyzékeként külön másoljuk.

4.5. További információk a rendszerről

A Dir paranccsal megtudhattuk, hogy melyik fájlok milyen szerkezetben helyezkednek el a lemezen. A Dir parancs párja a List, amellyel — a Dir parancshoz hasonlóan — a lemez tartalomjegyzéke jeleníthető meg, azonban jóval több információt látatva a képernyőn.

Minden fájl bájtokban mért nagyságát is feltünteti a fájl neve után. Ha a tétel másik tartalomjegyzéket jelöl, akkor a méret helyén a Dir szócska áll. Ezenkívül egy dátum is megjelenik minden tétel után. A dátumból másodpercre pontosan megtudhatjuk, hogy az adott fájl mikor került fel a lemezre, ill. utoljára mikor írtuk át. Mindig a legutolsó letöltés dátuma marad meg. A tartalomjegyzék mellett is létrehozatalának dátuma jelenik meg. Néha a dátum mellett vagy helyett valamilyen szöveget látunk. A List paranccsal kapott tartalomjegyzék egy összefoglaló sorral végződik, amely összegzi, hogy az adott tartalomjegyzék — a lemez főtartalomjegyzéke — hány fájl és hány altartalomjegyzéket tartalmaz, ill. a felsoroltak hány blokkot foglalnak el.

Nem szóltunk még egy érdekes funkcióról. A méret és a dátum között több-kevesebb betű található, ezek az adott fájlról közölnek információt. Nézzük, mit jelentenek!

- r* a fájl olvasható, azaz betölthető;
- w* a fájl átírható, nem írásvédett;
- e* a fájl valamilyen módon használható, futtatható;
- d* a fájl törölhető.

Aki a Shellt használja, az egy//űg-nek (e.fleg) nevezett betűvel is találkozhat.

- a* a fájl az eredeti (archív) formában van jelen;
- s* batchfájl;
- p* többszörösen futtatható program.

A betűk közül számunkra csak néhány érdekes. A *d* egy további paranccsal van összefüggésben. A *d* flag a Protect paranccsal törölhető, a fájl a Delete paranccsal nem tudjuk törölni. Erre a legfontosabb fájlok biztonsága miatt lehet szükség. (Az eredeti állapot a Protect paranccsal visszaállítható.) A Dir parancsot pl. a következőképpen védhetjük, ill. szüntethetjük meg a védelmét:

```
Protect df0:c/Dir
Protect df0:c/Dir rwed
```

Az *s* a batch fájlokat jelöli, ezek a Shellben az Execute parancs nélkül, a név beadásával futtathatók.

A *p-Ti* jó példa a *More* szöveg megjelenítő program, amely multitasking üzemmódban, egyszerre akár több CLI ablakban is futtatható. A DOS parancsként is felfogható programmal a

szövegfájlokat egyszerűen megjeleníthetjük a képernyőn. Egyszerre több szöveget is a képernyőre vihetünk. A More az 1.3-as változatban a Workbench lemez Utilities fiókjában található, az 1.2-es változatban pedig az Extrás lemezen. Érdemes megismerkedni a használatával.

More név

A *név* nevű szövegfájl megjelenik a képernyőn, de csak annyi, amennyi a képernyőre ráfér. A képernyő alján a More feliratot láthatjuk, ami azt jelenti, hogy a szöveg folytatódik. A Space billentyűvel lapozhatunk mindaddig, amíg a szöveg tart, azaz amíg meg nem jelenik az *End of file* felirat. Ha ezután lenyomjuk a Space-t, akkor visszajutunk a CLI-be. A fájlok megjelenítésére ez a módszer alkalmasabb, mint a Type parancs.

Ha a szöveg közben lenyomjuk a *h* betűt, akkor a Help (segítség) képernyőt kapjuk. A Return billentyűvel soronként is előre vihető a szöveg. A *q*-val vagy a *Ctrl+C*-vel bármikor kiléphetünk. (Ez hosszú szövegben fontos, mert nem kell a végére mennünk.) Hasznos funkció a visszafelé lapozás, ezt a Backspace billentyű végzi. A *<* lenyomásával azonnal az első oldalra ugorhatunk, a *>* pedig az utolsó oldalra visz. Ha hosszabb szövegben keresünk valamit, nem kell végigolvasni a teljes szöveget. A */* vagy a *.* billentyűvel megkereshetjük az adott sztringet. (A */* esetén figyelni kell a kis- és nagybetűkre.)

A More programnak sok változata van, ezek egyes segédparancsai eltérnek. Most a 3.27-es változatot mutattuk be. Aki egyszer kipróbálta, s gyakran készít, ill. olvas szövegfájlokat, az nagyon hamar megkedveli a More-t, ezért röviden ismertetjük, hogyan tudjuk használni a Workbench ikonrendszerével.

Ha egy szövegfájl őrünk, s ellátjuk ikonnal (a legtöbb szövegszerkesztő ezt automatikusan megteszi), akkor a Workbench menü Info funkciójával nyissuk ki az Info ablakot, és a TOOL TYPES feliratú részbe írjuk be a More-t. Ezután kattintsunk rá a *SAVE* felírra! Ettől kezdve, ha rákattintunk a szövegfájl ikonjára, akkor a szövegfájl és a More egyszerre betöltődik, és a szöveget a More-ral olvashatjuk. A More-t külön nem kell betölteni!

E kis kitérő után folytassuk a List parancs ismertetését. A parancsnak ugyanis nemcsak a megjelenítéskor van sok opciója, hanem az aktiváláskor is.

List Quick

Csak a fájlok neve jelenik meg, az összes többi adat hiányzik. A felsorolás végén azonban itt is látható az összesítő kimutatás.

List Nodates

A dátum nem jelenik meg.

List Keys

A fájl neve után kapcsos zárójelben megjelenik annak a lemezblokknak a száma, ahol a fájl a lemezen kezdődik. Ez a profi programozó számára nagyon hasznos lehet.

List Since dátum1 Upto dátum2

Mivel a List a dátumokat is nyilvántartja, ezért olyan tartalomjegyzéket is kérhetünk, amelyben csak a megadott két dátum közé eső fájlokat jelezzük ki. A dátumban olyan kifejezések is megengedettek, mint *yesterday* (tegnap), *today* (ma), sőt a hét egyes napjai is megadhatók (természetesen angolul). A Type szöveg parancs az egyes fájlok tartalmáról informál. A parancs hatására a *szöveg* nevű fájl megjelenik a képernyőn, de ha nem fér el a teljes szöveg, akkor sem áll meg a program úgy, mint a More esetén, hanem szalad tovább a szöveg végéig. Ha azonban bármelyik billentyűt lenyomjuk, akkor bárhol megszakíthatjuk a

programot, és a szöveg megáll a képernyőn. A Backspace billentyű lenyomására a program folytatódik.

A Type parancs akkor a leghasznosabb, ha nem a képernyőre, hanem mondjuk a nyomtatóra irányítjuk a műveletet. A nyomtatást itt nem részletezzük, de meg kell említeni a Type szöveg prt: nyomtatási lehetőségét. Ekkor a szöveg nem a képernyőn jelenik meg, hanem — ha a nyomtatónkat megfelelően csatlakoztattuk —, nyomtatott formában jutunk hozzá.

Érdekes lehetőséget kínál a Type hello hahó megoldás. Ebben az esetben hahó néven hozunk létre másolatot a hello nevű fájlról. Ez a fájlba másolás. Ha pl. hello néven új fájlt akarunk létrehozni, s azt akarjuk, hogy az új fájl ne írja felül a régit, akkor az ismertett megoldás jól használható.

A profi programozó számára két további opciót is tartogat a Type, az egyikkel az adott szövegfájlt vagy programot hexadecimális formában is megjeleníthetjük. Ha pl. a Dir parancsot meg akarjuk nézni, akkor a Type c/Dir opt h sort kell begépelnünk.

A másik opcióval a Basic programot automatikusan átszámozhatjuk, ha korábban ASCII formában mentettük ki: Type Basicprogram opt n.

A teljes rendszerről közöl adatokat az info parancs. Adjuk ki a parancsot, és várjuk meg, mi történik! Attól függően, hogy hány lemez meghajtónk van, azokban milyen lemezek vannak stb., egy információ-összeállítást kapunk. Mivel mindenki más konfigurációval dolgozik, ezért egy képzeletbeli helyzet felvázolásával megismerhetjük az egyes információk jelentését.

Unit	Size	Used	Free	Full	Errs	Status	Name	Volumes available:
DF2:	No disk present							Empty
DF1:	880K	1006	752	57%	0	Read/Write	AMIGA	AMIGA [Mounted]
DFO:	880K	1726	32	98%	0	Read Only	WB1.3	WB1.3 [Mounted]
RAM:	2K	4	0	100%	0	Read/Write	RAM DISK	RAM DISK [Mounted]

A *Unit* (egység) a rendelkezésre álló lemez meghajtókat és a RAM lemezt sorolja fel. A *No disk present* azt jelenti, hogy a kettes külső lemez meghajtóban pillanatnyilag nincs lemez. A *Size* (méret) megadja a lemez kapacitását. Az Amiga 880 kb-át kapacitású mikrofloppy lemezt használ, ha viszont 5¼-es lemez meghajtót kapcsolunk rá, akkor az érték már nem 880K, hanem 360K (ez már nem mikrofloppy). A *Used* (felhasznált) oszlopban levő szám a lefoglalt blokkok számát jelzi, a *Free* (szabad) pedig a szabad blokkokét. A *Full* (tele) azt jelzi, hogy a lemez hány %-a telített, az *Errs* az esetleges hibákról tájékoztat. A *Status* (állapot) azt jelzi, hogy a lemezről csak olvasni tudunk-e (Read Only) vagy írni is (Read/Write). A *Name* (név) oszlopban jelenik meg a lemez neve. A *Volumes available* (hozzáférhető egységek) oszlopban összefoglalva láthatjuk, hogy mely lemezek állnak rendelkezésünkre (mounted). A példában az Empty lemeznév mellett nem áll mounted felirat, vagyis a kettes meghajtóban nincs lemez, korábban viszont az Empty nevű volt benne.

4.6. Dátum

Az itt következő rész főképp azok számára fontos, akiknek a gépében van belső óra, mert így megfelelő előkészítés után mindig tájékozódhatnak a napi dátumról. Adjuk ki a Date parancsot! Válaszul azonnal megjelenik, hogy milyen nap van, és pontosan mennyi az idő (ha a belső óra megfelelően be van állítva).

Ha nincs belső óra, akkor is kapunk egy dátumot és időt, ez azonban hamis lesz. Az így kapott dátum ugyanis úgy jön létre, hogy a gép bekapcsolása pillanatától kezdve méri az eltelt időt, s azt hozzáadja a Workbench-en gyárilag beállított időhöz. (Az időt a Preferencesen keresztül lehet megnézni és beállítani.)

Két olyan parancs van, amely a belső órával kapcsolatos. Az egyikkel pontosan be tudjuk állítani a belső órát. A másikat a Workbench Startup-sequence-ben helyezhetjük el, és a Preferencesben beállított időt minden bútoláskor a belső óra szerint aktualizálhatjuk vele (így mindig rendelkezésünkre áll majd a pontos idő és a dátum).

Első lépésként be kell tölteni a Preferencest. Ehhez nem kell kilépni a CLI-ből, hiszen onnan is elindítható minden program. Amelyik program a Workbench szintről elindítható, az általában elindítható a CLI-ből is, fordítva azonban ez nem mindig igaz. Billentyűzzük be tehát a Preferences sort, majd állítsuk be a pontos dátumot és az időt. A SAVÉ aktiválásával lépünk ki, majd a belső órát állítsuk a Preferencesben beállított időre:

```
Setclock opt Savé
```

Ettől kezdve a belső óra a gép kikapcsolása után is pontosan mutatja az időt.

A Preferencest[^] természetesen nem szükséges bevonnunk, a műveletet a CLI-ből is elvégezhetjük. Új parancsot sem kell megtanulnunk, a Date ugyanis nemcsak kijelzésre, hanem beállításra is alkalmas:

```
Date 12-aug-1989  
Date 12:44
```

Az első sor a dátumot, a második az időt állítja be.

Mivel a Date parancs (és minden más időt lekérdező rendszer) nem a belső órát, hanem a Preferencest figyeli, s mivel a Preferences a kikapcsolás után "elfelejti" a beállított időt, bútoláskor a belső óra által megtartott pontos időt rá kell másolnunk a Preferencesre. Ezt a Startup-sequence-ben elhelyezett Setclock opt Load parancs végzi. Előfordulhat, hogy a géphez adott német DOS kézikönyv némelyik példánya — tévesen — összekeveri az opt Load és az opt Savé opció jelentését.

A List parancs tárgyalásakor említettük, hogy a List a dátumokat is nyilvántartja. Ha valamilyen oknál fogva az egyes fájlokhoz bejegyzett dátumot és/vagy időt meg szeretnénk változtatni, akkor azt a Setdate parancssal tehetjük. A Setdate df1:prog/hello 12-aug-89 11:33 programsor azt eredményezi, hogy a külső meghajtóban levő lemez prog nevű fiókjában elhelyezkedő hello fájl mellé egy augusztusi dátum és idő kerül.

4.7. Különböző DOS parancsok

Eddig már jó néhány parancssal megismerkedtünk. Most olyan parancsok következnek, amelyek önmagukban is használhatók, és néhány esetben jól kiegészítik a korábban bemutatottakat.

Ha az l > vagy az l .SYS:> helyett a promptban egyedi szöveget akarunk megjeleníteni, akkor a megoldás a következő Prompt parancs:

```
Prompt "Kiss János>"
```

Ettől kezdve a parancssorok elején a *Kiss János* felirat jelenik meg. Ha vissza akarunk térni a megszokott számhoz (amely a CLI ablak sorszáma), akkor adjuk ki a Prompt "%n"> parancsot.

Az Echo paranccsal különféle üzeneteket vihetünk fel a képernyőre. A gyártók általában a Startup-sequence-be beépítve alkalmazzák, hogy a lemez betöltésekor a felhasználót néhány dologról tájékoztassák. A parancs igen egyszerű:

```
Echo "szöveg"
```

A kezelő felhasználók gyakran kapnak hibaüzenetet. Ez lehet szöveg, vagy szöveg és hibakód. Ilyenkor meg kell nézni, hogy az adott hibáról (ül. a hiba okáról) mit ír a DOS kézikönyv. A CLI két parancsa (ha tudunk angolul) megkönnyíti a dolgunkat. Az egyik a Why (miért), amit így, önmagában kell kiadnunk. Válaszul egy magyarázó szöveget kapunk, amely általában utal a hibakódra is. A Fault paranccsal pedig egy adott hibakód jelentésére kérdezhetünk rá. Pl. `fault 221`,aválasz: `Fault 221: disk full`.

A List paranccsal részletes tartalomjegyzéket kaptunk. A DOS lehetővé teszi, hogy a List tartalomjegyzékben megjelenő adatok mellé (akár minden fájlhoz) max. 80 karakterből álló szöveget írjunk. A `Filenote dfl:prog/hello "Üdvözléssel"` begépelése után a *hello* programban a dátum mellett megjelenik az *"Üdvözléssel"* felirat.

Több fájl összemásolására a `Jóin` parancsot használhatjuk. A `Jóin prog1 progr2 AS` program hatására a *prog1* és *prog2* létrehozza *program* nevű fájlt. Ezt a megoldást jól használhatjuk, ha egy szöveget külön-külön megírt részekből akarunk összeállítani. A paranccsal max. 15 fájlt tudunk összemásolni.

A szövegmásoláson kívül megemlíjtük a `Jóin` paranccsal végzett nyomtatást is. A szövegfájlokat ugyanis nemcsak egy újabb szöveggé másolhatjuk össze, hanem összemásoltan mindjárt ki is nyomtathatjuk:

```
Jóin szövega szövegb AS prt:
```

A *szövega* és a *szövegb* egymás után folyamatosan kinyomtatva jelenik meg a papíron.

Az Amiga leghatásosabb alkotóeleme a Workbench. A gép bekapcsolásakor azonban a Workbench sem áll rendelkezésre, azt is be kell tölteni. Ez a bűtöláskor általában végbemegy, mert a Startup-sequence tartalmazza a `Loadwb` parancsot. Ezért a CLI-ben végzett munka során általában nincs rá szükség, mert ha egyszer betöltöttük, akkor a gép kikapcsolásáig bent marad a rendszerben. Csak akkor kell alkalmaznunk, ha nem a Workbench-en keresztül jutottunk a CLI-be. Ilyenkor az `Endcli` parancs kiadása előtt — ha már csak egy CLI ablak van nyitva — feltétlenül alkalmazni kell a `Loadwb` parancsot, mert egyébként a CLI ablak bezárul, és egy üres képernyőhöz jutunk, amivel semmit nem lehet kezdeni, csak kikapcsolni.

Amikor a Workbench menüit próbálgattuk, a Special menüben volt egy funkció, a `Version`. `Version` nevű DOS parancs is van, azonos funkcióval.

4.8. Idegen nyelvek

A legtöbb Amiga német nyelvterületről került Magyarországra. A billentyűzet ennek megfelelően a német helyesíráshoz szükséges betűket tartalmazza. A magyar ékezetes betűk nagy részét ezért ne keressük, ugyanúgy, mint pl. a francia abécé egyes betűit sem. A gép tervezői azonban számoltak azzal, hogy az Amiga a világ sok tájára eljut, és különböző nyelveken fognak vele dolgozni. Ezért kialakítottak egy viszonylag rugalmas rendszert, amellyel az egyik nyelvről a másikra — pontosabban az egyik betűkészletről a másikra — lehet váltani. Ennek a rendszernek az egyik korlátja az, hogy a billentyűzet (azaz a billentyűkön levő felirat) nem, vagy csak átragasztással változtatható, ezért előfordulhat, hogy betűkészletváltás után a billentyű lenyomásakor nem az a betű jelenik majd meg a képernyőn, mint ami a

billentyűn van. A másik korlátja a rendszernek, hogy csak azok a betűk, ill. nemzeti karakterek jeleníthetők meg, amelyekre a devs/keymaps tartalomjegyzékben elhelyezték a szükséges programot. Az amerikai és a kanadai angolon kívül megtalálható itt a svéd, a norvég, az olasz, a francia, a dán, s még jónéhány ábécé.

Az egyes karakterkészleteket a Setmap paranccsal aktiválhatjuk. Érdemes megjegyezni, hogy ez a parancs — a többi DOS paranccsal ellentétben — nem a *c* tartalomjegyzékben, hanem a Systemben található. A német karakterkészlet pl. igen egyszerűen aktiválható a Setmap *d* sor begépelésével. Feltéve, ha a Path System add paranccsal a *System-ei* már kijelöltük.

A Path és a Setmap parancsot általában már a bútoláskor kiadjuk, mert a Startup-quence-be vannak építve. A Setmap parancsra csak akkor van szükség, ha a karakterkészletet a munka közben akarjuk megváltoztatni. Előtte azonban tanulmányozzuk a gépkönyv végén található táblázatokat, a különböző ábécék billentyűzetéről.

Lássuk, hogy juthatunk a magyar ábécéhez! Több megoldás van, aszerint, hogy csak a képernyőn (pl. a Basic programokban) jelenjen meg a magyar ábécé, vagy esetleg nyomtatásban is. (A nyomtatásról az 5. fejezetben lesz majd szó.)

Az igazi megoldás az lenne, ha létre tudnánk hozni a devs/keymaps tartalomjegyzékben található programokhoz hasonló programot, amit *h* betűvel jelölnénk. Ezután csak a Setmap *h* parancsot kellene kiadni.

Egy ilyen program létrehozásához alapos programozói ismeretekre van szükség. Megoldást jelenthet a forgalomban lévő Setkey szoftver, amely ezt elvégzi helyettünk. A szoftvert elindítva a képernyőn megjelenik az Amiga billentyűzetének képe, s bármelyik billentyű átdefiniálhatjuk. Az így kapott változatot akár *h* jelzéssel is kimenthetjük a lemezre.

Az egyes billentyűk definiálásához a teljes ASCII-kddkészletet alkalmazhatjuk. A teljes készlet sem tartalmazza azonban a magyar ábécé két ékezetes magánhangzóját, az *ó*-t és az *ü*-t. A létrehozott karakterkészlet csonka lesz. Egy másik megoldás ezt is kiküszöböli. A 2. fejezetben bemutatott font editor használatával akár kínai frásjeleket is létrehozhatunk.

4.9. A Basic programok profi indítása

A CLI-ből is indíthatunk Basic programokat. Tudjuk azonban, hogy a Basic programok futásához az Amiga-Basicet is be kell tölteni. Természetesen betölthetjük először az Amiga-Basicet, majd a Basicből hagyományos módon indíthatjuk a szükséges programot. Van azonban ennél sokkal elegánsabb megoldás is. Ehhez felhasználjuk az Amiga multitasking-képességét, a **Run** parancs segítségével. A Run parancs — a Basic Run utasításával ellentétben — itt nemcsak arra való, hogy a programot elindítsuk, hanem a programot egy másik (vagy akár több) programmal párhuzamosan indítja el.

Tegyük fel, hogy az elindítandó Basic program neve demonstráció. A Run amigabasic "demonstráció" programsor lehetővé teszi, hogy a programot közvetlenül a CLI-ből indítsuk. Ekkor az Amiga-Basic és a program egyszerre töltődik be. A CLI ablakban megjelenő [CLI 2] jelzi, hogy két program fut egymás mellett. Ha három programot kellene egyszerre futtatni, akkor a hasonló eljárás eredményeképpen a [CLI 3] is megjelenne. Rövid betöltés után fut a *demonstráció* nevű program.

De lépünk tovább, hiszen az igazán jó megoldás az lenne, ha sem a Workbench-et, sem a CLI-t nem kellene betölteni, hanem ahogy a gyári szoftvereken megszoktuk, a lemezt behelyezve a program automatikusan betöltődik, s természetesen el is indul. Az előzőek ismeretében ez sem túl bonyolult. Azt kell tudnunk, hogyha egy formázott üres lemezre a Copy paranccsal teljes egészében átmásoljuk a Workbench lemezt (tehát nem a Diskcopyt alkalmazzuk), akkor hiába

rendelkezőnk a bútolást vezérlő Startup-sequence-szel, nem fog történni semmi. A lemezt az erre a célra való paranccsal előbb még bútolhatóvá kell tenni. Ez a parancs átírja a lemez nulladik blokkját, az ún. bútblokkot. Ezután a megfelelő Startup-sequence-szel már elvégezhetjük a betöltést.

Az Install DRTVE df0: paranccsal a belső lemezmeghajtóban levő lemezt alakíthatjuk bútolhatóvá. így kell eljárunk a Basic programot tartalmazó lemezzel is. Ezután az Eddel kicsit alakítunk a Startup-sequence-en. A legfontosabb, hogy a Run amigabasic "programnév" sort beépítsük, de a legegyszerűbb, ha magunk írunk egy Startup-sequence-et, mert alig van valamire szükség a Workbench Startup-sequence-éhez képest. Később ezt igényeink szerint bővíthetjük.

A következő egyszerű példa a legtöbb esetben tökéletesen megfelel.

```
Echo "Ide tetszőleges szöveget írhatunk !"
System/Setmap d
Setclock opt load
Run amigabasic "program"
Loadwb
Endcli.
```

A program lelke a negyedik sor. Az első sor akkor fontos, ha nemcsak egyedül használjuk a Basic programot, és a többi felhasználónak jelezni akarunk valamit (pl., hogy a programot ki és mikor írta). A második sor a német karakterkészletet definiálja. A harmadik sor csak belső óra használatukra célszerű. Az utolsó két sorra akkor van szükség, ha a programot befejezve, vissza akarunk térni a Workbench szintre.

Lépjünk azonban eggyel tovább. Sok professzionális szoftver csak kis lemezrészre foglal el. Bizonyára sokan találkoztak már azzal, hogy valaki lemeztakarékosságból azonos jellegű programokat — általában felhasználói programokat — összemásolt egy lemezre. A lemez behelyezésekor az a menü jelentkezik, amelyből a kérdéses program könnyen kiválasztható. Ilyet mi is készíthetünk.

Először tegyük fel, hogy nem Basic programokról van szó, tehát az Amiga-Basic betöltésére nem lesz szükség. Készítsük el a programok listáját, a menüt, amiből majd választani kell. Ezt bármelyik szövegszerkesztővel (pl. az Eddel) megtehetjük. A menü legyen pl. a következő:

```
SZUPERSZOFTVER-VÁLASZTÉK
1. Első program
2. Második program
3. Harmadik program
4. Negyedik program
```

Ezután helyezzük el ezt a szövegfájlt egy megfelelő helyen. Legyen ez az *s* tartalomjegyzék, és a fájlunk adjuk *zprog* nevet.

```
Copy df0:prog to df0:s
Delete df0:prog
```

Ezután az Eddel írjunk egy egyszerű Startup-sequence-et:

```
Type s/prog
Prompt " Válasszon:"
```

Már csak az van hátra, hogy az egyes programokat a Rename paranccsal átnevezzük úgy, hogy a *prog* menüben alkalmazott számot kapják. Ne felejtjük el a lemezt az Install paranccsal bútolhatóvá tenni! Ha ezek után az Amiga bekapcsolásakor behelyezzük a lemezt, néhány

másodperc múlva megjelenik *zprog* szöveg, s alatta középen a "Válasszon:" felirat. Nincs más dolgunk, csak a megfelelő számot megadni, és az Entert lenyomni.

4.10. A Basic lemez létrehozása

A 3. fejezetből tudjuk, hogy a programozáskor célszerű egy Basic lemezt létrehozni. A Workbench lemez alapesetben annyira telített, hogy az Amiga-Basic program ugyanúgy nem fér rá, mint az általunk elkészített programok. A 4. fejezet eddigi részei tulajdonképpen majdnem minden információt megadtak ahhoz, ami egy Basic lemez elkészítéséhez szükséges. Most azonban a konkrét feladatra összpontosítva nézzük meg az egyes lépéseket.

1. Formázzunk meg egy üres lemezt!
2. Másoljuk rá az Extrás lemezeztől a Basicet!
3. Másoljuk rá a Workbench lemezeztől a teljes *t*, *l* és *libs* tartalomjegyzéket!
4. Hozzunk létre egy üres *s* tartalomjegyzéket a Makedir paranccsal!
5. Hozzunk létre hasonló módon egy üres *c* tartalomjegyzéket is.
6. Másoljuk át a *c* tartalomjegyzékbe a Workbench-ről a következő parancsokat: CD, Date, Echo, Ed, Endcli, FF (csak az 1.3-as változatban), Loadwb, Run, Setclock (csak belső óra esetén).
7. Másoljuk át a System tartalomjegyzékből a Setmap parancsot!
8. Hozzunk létre a *devs* — azon belül a *keymaps* — tartalomjegyzéket, majd másoljuk ide a szükséges karakterkészletet (feltehetőleg a *d-i*)\
9. Tegyük bútolhatóvá az Install paranccsal a lemezt!
10. Készítsük el az Eddel a Startup-sequence-et, és másoljuk be az *s* tartalomjegyzékbe! (A legjobb, ha mindjárt ott készítjük el.)

```
CD df0(vagy df1:):s
Ed Startup-sequence
```

Az üres ablakba a következőket írjuk:

```
Echo "Basic munkalemez"
Setclock opt load
Date
Setmap d
FF >NIL: -0
amigabasic
Loadwb
Endcli
```

Ezután ki-ki finomíthat rajta.

4.11. CLI programok

Nézzünk egy megoldást arra, amikor két Basic program van a lemezen, s azok közül kell választanunk. Ebben az esetben a Startup-sequence-be építjük be a következő részt:

```
Echo "1. Első program"
Echo "2. Második program"
Ask "Az első programot kéri ? (y/n) "
If warn
    Run amigabasic "Első program"
else
    Run amigabasic "Második program"
endif
```

Ez a programrészlet több új DOS parancsot is tartalmaz. Mindjárt feltűnik a Basichez hasonló `If warn...else...endif`, a működésük is nagyon hasonlít. Ha az `Ask` parancs után álló kérdésre igennel (yes) válaszolunk, akkor a `warn` után álló rész hajtódik végre. Ha a válasz nem (no), akkor az `else` után álló rész fut le. Az `Ask` parancs után csak eldöntendő kérdés állhat, mert a válasz csak igen vagy nem lehet.

Az `If` szerkezetnek más variációi is vannak, ezek alkalmazásával érdekesebbé tehetjük a CLI programokat. Nézzünk néhány példát!

```
If exists c/Dir
Dir
else
Echo "A Dir parancs nincs a lemezen"
```

A program megvizsgálja, hogy a lemezen a `c` tartalomjegyzékben van-e (exists) a `Dir` parancs. Ha igen, akkor végrehajtja, ha nem, akkor ezt szövegesen jelzi.

Az `ON ERROR GOSUB Basic` megoldáshoz hasonlóan működik az `If error` feltétel.

Térjünk vissza az eredeti programhoz, amely a Basic programok közül segít választani. A programot nemcsak a Startup-sequence-be építhetjük be, hanem működhet önálló batch programként is. Ilyenkor valamilyen néven elhelyezzük a lemezen, és az `Execute` utasítással (az 1.3-as változatban csak a név megadásával) aktiválhatjuk. Akár több ilyen CLI programot is elhelyezhetünk a lemezen, és az igénynek megfelelően lefuttathatjuk azokat.

Utolsó CLI programként nézzünk meg egy olyat, amely nem két, hanem akár több Basic program közül segít kiválasztani a megfelelőt. Tegyük fel, hogy a Basic programok megegyeznek az előzőekben említett szuperszoftver-választékkal. Ekkor használhatjuk az ott megírt `prog` nevű szövegfájl is:

```
Type s/prog
Skip ?
Lab 1
    Run amigabasic "Első program"
    Skip stop
Lab 2
    Run amigabasic "Második program"
    Skip stop
Lab 3
    Run amigabasic "Harmadik program"
    Skip stop
Lab 4
    Run amigabasic "Negyedik program"
    Skip stop
Lab stop
```

A program lényege a `Skip` és a `Lab` utasítás. A `Lab` nem is igazán utasítás, inkább címke. Ugyanolyan címke, mint amelyeneket a Basic programozáskor megismertünk, azaz a

tulajdonképpen címketartalom a Lab szó után következik. Kettőspont itt nem kell. A Skip utasítás a GOTO utasításhoz hasonlítható. Ha pl. azt szeretnénk, hogy a program a 3-as címkéhez ugorjon, akkor Skip 3-t kellene írunk. Az előző példában azonban a felhasználó választása dönti el, hogy a program melyik Lab címkéhez ugorjon, ezért a Dir parancsnál bemutatott megoldást alkalmaztuk. A kérdőjel hatására ugyanis a program nem hajtódik végre, hanem várja, hogy mi adjuk meg a paramétereket.

Ha a programot lefuttatjuk, akkor a sor elején a LABEL: (az 1.3-as változatban a LABEL.BACK/S:) felirat jelenik meg. Ezután csak a kért Basic program sorszámát kell megadni és az Entert lenyomni.

Csak akkor tudjuk elkerülni, hogy a futtatás után a program az esetleg hátramaradt címkék mögötti részt is sorban végrehajtsa, ha mindenhova elhelyezzük a Skip stop utasítást (ekkor a stop egyszerű címke). Jegyezzük meg, hogy a Skip utasítással az 1.2-es változatban csak előre lehet ugrani, tehát pl. a tizedik sorban kiadott Skip utasítással akkor sem lehet a hatodik sorba ugrani, ha a Skip után írt címke a hatodik sorban van. Az 1.3-as változat már kiküszöböli ezt a hiányosságot.

CLI programjainkba az Echo (vagy más) alkalmazásával szövegeket építettünk be. Ha rögtön a szöveg után olyan programrészlet következne, amely miatt a szöveg elolvasására nem lenne időnk, akkor alkalmazzuk a Wait utasítást:

```
Wait 10
Wait 2 min
Wait until 22:10
```

Az első utasítás hatására a program 10 másodpercet vár, a másodiknál két percet. A harmadik utasítás hatására mindaddig vár, amíg a rendszer este tíz dra tíz percet nem jelez.

Főképp több hosszú program vagy nagyobb mennyiségű szövegfájl esetén gyakori igénye a felhasználónak, hogy egy bizonyos szót, szövegrészt hamar megtaláljon. A nagyobb tartalomjegyzékek, ill. a bennük levő fájlok átböngészése nem jó megoldás. Erre a feladatra alkalmas a Search parancs.

Ha a feladat pl. a kulcs szó megtalálása a dfl: meghajtóban lévő lemezen, akkor a következőt kell írunk:

```
Search dfl: search kulcs
```

A parancs hatására megkapjuk annak a fájlnak (vagy fájloknak) a nevét, amely(ek)ben a kulcs szó előfordul. Egyetlen gond van csak, hogy a lemezen levő esetleges altartalomjegyzékeket nem vizsgálja a gép. Ezen egy rövid kiegészítéssel segíthetünk:

```
Search dfl: search kulcs all
```

4.12. Rend a lelke mindennek

A közmondás a fájlokra is igaz. Képzeljük el, hogy olyan adatbázist készítettünk, amellyel barátaink nevét és címét gyűjthetjük össze (vagy pl. videofilmek címét egy videotár számára). Tegyük fel, hogy az adatbázist a 3. fejezetben megismert szekvenciális fájlal hozzuk létre. A megfelelő címetek naponta folyamatosan visszük be, és az újabbakat hozzáírjuk.

Természetesen elkészítettük a Basic programnak azt a részét is, amellyel rendezett formában jeleníthetjük meg az adatokat. Célszerű lenne az ábécé szerinti rendezettség. Ezt egy Basic

programmal is megoldhatjuk, de nehézkes. Minél több az adat, annál lassúbb lesz a Basic program.

Ekkor kell betölteni a CLI-t, és elővenni a Sort parancsot. Ez a parancs bármelyik fájl tartalmát gyorsan ábécé sorrendbe rendezi, majd másik fájlként kimentí a lemezre. Már csak a régi fájl kell letörölni, az újat pedig a régi nevére átnevezni.

Nézzük meg, mi a teendő, ha a rendezésre váró fájl neve *címek*:

```
Sort df0:címek dfl:újcímek
Delete címek
Copy dfl:újcímek df0:címek
```

Ha nem akarjuk az egész fájlt rendezni (mert pl. az első része már rendezett), akkor a parancsot azzal az opcióval bővítjük, amelyik megmondja, hogy hányadik karaktertől végezze a gép a rendezést. Ha pl. az összes név *p* betűvel kezdődik, akkor a rendezést elég a második betűtől kezdeni:

```
Sort címek újcímek Colstart 2
```

A bemutatott példa csak egy lehetőség a sok közül a DOS parancs alkalmazására. Mielőtt azonban elkezdenénk a rendezést, vizsgáljuk meg, hogy mekkora a rendezendő fájl. Ha túl nagy, akkor gondunk lesz a memória miatt, és megjelenik a Guru. Alaphelyzetben a rendezésre 4000 bájtot foglal le a gép, ez a DOS könyv szerint kb. 200 soros fájl rendezésére elegendő. Ha ennél nagyobb mérettel dolgozunk, akkor a Stackparanccsal megnövelhetjük a memóriát:

```
Stack 7000
```

A lefoglalt memória most 7000 bájt.

4.13. A gyorsítás egyik módszere

Ugyancsak a memóriával függ össze a következő parancs, amellyel bizonyos esetekben gyorsíthatjuk a CLI-ben való műveletvégzést. Amikor a gépre egy lemez meghajtót kapcsolunk, akkor az Amiga bizonyos mennyiségű memóriát leköti a lemez műveletek számára, azaz *puffért* képez. Ez 17 kbájt, amit azonban növelni tudunk. A növelés egysége a blokk, ebből azonban egy rész bizonyos belső műveletekhez kell, tehát egy egységnyi növekedés nem több 500 bájtnál. A növeléshez az Addbuffers df0: 15 parancsra van szükség.

A belső lemez meghajtóhoz rendelt 17 kbájtot tehát 15 egységnyivel — azaz mintegy 15x500kbájjal — növeltük meg. Milyen hatással lesz ez pl. a DOS parancsokkal végzett műveletekre? Tudjuk, hogy a parancsokat először mindig lemezzől kell betölteni, ilyenkor ezek bekerülnek a pufferbe. A további műveletek azonban (hely hiányában) a legutolsó művelettel mindig felülírják a memóriát. Ha a puffert megnöveltük, akkor az egyszer már betöltött parancs legközelebb nem a lemezzől, hanem a puff érből töltődik be. A memóriából való betöltés — a RAM lemezhez hasonlóan — nagyságrendekkel gyorsabb, mint a mechanikus működésű lemez meghajtó.

4.14. A beteghez doktort kell hívni

Mindenkivel előfordulhat, hogy egy floppyja megsérül. Ez a sérülés lehet mechanikus, de keletkezhet szoftveresen is. A mechanikus sérülés legtöbbször egy porszem bejutásakor keletkezik, de a túlzott hősugárzás vagy a szakszerűtlen tárolás is okozhatja. A szoftver

eredetű sérüléseket később részletezzük, most csak a vírust említjük. Az eredmény mindkét esetben az, hogy a rendszer hibajelzéssel leáll, és az adatokat, í ll. a programot sem betölteni, sem letölteni nem tudjuk. Ha ilyen hiba jelentkezik, maga a rendszer javasolja a hiba kijavítását, a CLI-ből hívható Diskdoctor paranccsal:

```
Diskdoctor df0:
```

A parancs kiadása után a gép az egész lemezt (ebben az esetben a df 0:-t) átvizsgálja, és jelzi a hibás részeket. Ha a hiba nem súlyos, akkor a parancs kijavítja a lemezt. A legtöbbször azonban a rendszer megszünteti a hibás részeket, majd a többi újraformázza. Ezzel sajnos fontos részek semmisülhetnek meg a lemezen. Mivel a művelet hatására a lemezen helyreáll a rend, a nem sérült részeket újból olvashatjuk, ill. másolhatjuk.

Szerencsés esetben az adatok nagy részét meg tudjuk menteni ezzel a módszerrel. Ha a megmaradt sértetlen adatokat átmásoltuk, a sérült lemezt érdemes többször egymás után formázni és ellenőrizni.

Az ellenőrzés legkiválóbb módszere a Public Domain program, a TurboCopy. Tetszőleges lemezt kell átmásolni a többször formázott "meggyógyított" lemeze. Ha bármilyen rejtett hiba maradt, a TurboCopy ezt kijelzi. Ha a többszöri formázás ellenére sem sikerül kifogástalan lemezhez jutni, akkor feltételezhetően súlyosabb mechanikus sérülésről van szó. Ne kíséreltezzünk tovább, a lemezt ne használjuk többet.

Ha a hiba egy ismeretlen forrásból származó program használatakor (pl. kalózmásolatban) merült fel, akkor legyünk nagyon óvatosak a további munkával. Ilyenkor a kérdéses programot — akár játék, akár felhasználói program — provokálni kell. A provokáció azt jelenti, hogy számunkra nem fontos — de garantáltan hibátlan — lemezzel és adatokkal hosszú ideig összevissza használjuk a programot. Ha egy bizonyos idő után újból jelentkezik a hiba, akkor alapos lehet a gyanú, hogy valamilyen vírussal állunk szemben. A témát a 10. fejezet tárgyalja.

A Diskdoctor használatakor gondoljunk arra, hogy a parancs memóriagényes.

4.15. Az 1.3-as változat különlegességei

Már többször említettük az 1.3-as változat többletszolgáltatásait, az adott paranccsnál megemlítve az eltéréseket, főképp az új **CLI**, a Shell használatát illetően. Most az 1.3-as változatba beépített néhány további eszközt mutatjuk be.

Először csak a hatása miatt vesszük észre az új Startup-sequence-ben elhelyezett FF parancsot, a **FastFonts (FF)** ugyanis rendkívüli módon meggyorsítja a szöveg képernyőn való megjelenítését. Ez jól észrevehető, ha egy teljesen teleírt oldalakkal rendelkező Basic programban lapozunk. Alapesetben elég lassan haladunk. Az FF alkalmazása után az oldalak szinte leomlanak. A program nagyon hasonlít a Blitzfonts programhoz, amely számos Public Domain lemezen előfordul. A parancs a következő módon aktiválható:

```
FF >NIL: -0
```

A gyártók az 1.3-as változatban bevezették a Fast-Filing-System-et, amely a merevlemezrel rendelkezők munkáját könnyíti meg, mert gyorsabb hozzáférést tesz lehetővé. A programmal egyszerűen felépíthetjük a fájlokat és a tartalomjegyzékeket.

A **Pipe** (cső) alkalmazásával két program között teremthetünk kapcsolatot, hasonlóan a Copy parancshoz, amikor egy fájlt egy másikba másoltunk. Ebben az esetben ez programok között megy végbe.

A Copy parancs kiegészült a Buf opcióval, amely az egy lemezmeghajtóval rendelkezők munkáját könnyíti meg. A Buf után megadandó egész számok az Addbuffershez hasonlóan plusz memóriát különítenek el, így hosszabb programok másolásakor elkerülhető a gyakori lemezcsere, az egylemezes másolás szokásos kíséréje.

A Copy parancs új opciója a Date is, amellyel elérhetjük, hogy a másolat az eredetivel azonos dátummal töltsön le. A Com opcióval pedig azt érhetjük el, hogy a fájlhoz (a Filnote paranccsal) hozzáfűzött szöveg is megjelenik a másolatban. A Date és a Com a Clone opció megadásával egyszerűen kombinálható.

A CLI gyakori használóinak kézenfekvő az új parancs, az Alias alkalmazása. A hosszabb parancsokban munkát és időt takaríthatunk meg, ha begépeljük az Alias dis Diskdoctor programsort. Ettől kezdve, ha a Diskdoctor parancsot akarjuk alkalmazni, akkor elég csak dis-t gépelni.

De nemcsak a parancsok, hanem a hosszabb parancssorok is rövidíthetők. Az egyik leggyakrabban előforduló parancs a Startup-sequence editálása:

```
Ed dfl:s/Startup-sequence
```

Ha a munka elején azt írjuk, hogy Alias el Ed dfl:s/Startup-sequence, akkor ettől kezdve elég, ha e l-et írunk csak.

A végére maradt az egyik legérdekesebb új parancs, amellyel az egy lemezmeghajtóval rendelkezők más módszerrel is előhívhatják a DOS parancsokat. Eddig a megoldást a gyakran szükséges parancsok RAM lemezre másolásával oldottuk meg. A következő paranccsal a szükséges DOS parancsot *rezidensen* be tudjuk tölteni a memóriába. Ez azt jelenti, hogy a programozáskor más programok nem írják felül, a háttérben féke van téve. A parancs igen egyszerű:

```
Resident parancs pure
```

Aparancs az általunk tetszőlegesen kiválasztott, a memóriába rezidensen beépítendő parancs. Érdemes megvizsgálni az 1.3-as változat Startup-sequence-ét (vagy a StartupII-t), mert ott további példákat láthatunk a Resident használatára.

5. NYOMTATÁS

A nyomtatás a számítógép használatának fontos része. A nagy cégek gondot fordítanak arra, hogy az általuk kifejlesztett számítógép rendelkezzen azokkal a feltételekkel, amelyekkel a nyomtatás könnyen és gyorsan megvalósítható. Becslések szerint a számítógépeknek — és azon belül is a személyi számítógépeknek — legalább a fele kizárólag (vagy főképp) szövegszerkesztési feladatokat old meg. A szövegszerkesztésnek pedig az a célja, hogy a szöveg előbb vagy utóbb nyomtatásban is megjelenjen.

A szövegszerkesztés már a számítógépek alkalmazásának korai szakaszában is az egyik legkedveltebb felhasználási mód volt, napjainkban pedig fantasztikus gyorsasággal tör előre. Ennek fő oka, hogy a nyomdák egyre inkább áttérnek az elektronikus szedésre, és a kisebb-nagyobb cégek is használják levelezésre, hiszen a hagyományos írógéphez képest rengeteg időt lehet megtakarítani. (Természetesen így kevesebb dolgozóra van szükség.)

A nyomtatás azonban nemcsak a szövegek megjelenítésekor szükséges. A programozónak — akár Basicben, akár más, magasabb szintű nyelven programoz — időnként ki kell nyomtatnia a programlistáját, mert még a néhány képernyőoldali programok is sokkal áttekinthetőbbek, egyszerűbben javíthatók, ha az egészet magunk előtt látjuk. Kinyomtathatunk különböző grafikákat is, amelyek a gép kikapcsolása után elvesznének.

Egy hasonlattal élve a számítógép nyomtatási lehetőség nélkül olyan, mint egy autó" hátrameneti fokozat nélkül. Közlekedni ugyan lehet vele, de számos dolgot csak körülményesen vagy nem lehet megoldani.

Különböző márkájú és típusú nyomtatók kaphatók. Nagyon fontos, hogy a cémak megfelelő nyomtatót szerezzük be, elkerülve az "ágyúval lövünk verébre" vagy a "Trabanttal a Forma-1 -ben" tipikus eseteit.

Komolyra fordítva a szót, ismerkedjünk meg a főbb nyomtatótípusokkal és azok alapvető tulajdonságaival. Azt is tudnunk kell, hogy a nyomtató a soros vagy a párhuzamos porton keresztül kapcsolódik-e a számítógéphez. A nyomtatók többsége a párhuzamos portot használja.

Nézzük meg, hogy a nyomtatóhoz milyen kiegészítő egységek kapcsolhatók, mert ezek megkönnyíthetik a munkát. A legtöbb nyomtatóhoz kapható pl. *lapadagoló*. Ha nem *leporellóra* (olyan papírtekercs, ahol az oldalak megszakítás nélkül, folyamatosan következnek, s az oldalán levő perforálás segítségével halad előre a papír a nyomtatóban), hanem külön lapokra dolgozunk, akkor a több tíz vagy több száz oldalnyi nyomtatni való esetén a papírlapok betétele fárasztó és unalmas munka, ráadásul közben végig a gép mellett kell állni.

A lapadagoló a nyomtatóra ráhelyezhető kis szerkezet, amelyben 100—150 papírlapot is elhelyezhetünk, majd nyugodtan elmehetünk a géptől, mert a lapok egyenkénti adagolását megoldottuk.

Néhány nyomtató *betűmodullal* is rendelkezik. A legtöbb nyomtatót gyárilag programozott betűtípusokkal, ún. fontokkal látták el. Némelyekhez behelyezhető modulként más fontkészletek is kaphatók, ezek szélesítik a választékot. Ezekkel a modulokkal megoldhatjuk nehézségeinket (pl. a cirill betűk használatát).

5.1. Az egyes nyomtatótípusok

Ismerkedjünk meg a ma kapható nyomtatók három fő típusával, amelyek közül az elvégzendő feladatoknak megfelelően kell kiválasztanunk a legalkalmasabbat. Nem csak a típus, hanem az ár szerint is tájékozódjunk. Sok nyomtató jóval többbe kerül, mint az Amiga 500-as, de olyanok is vannak, amelyek árából több Amiga 2000-t is vásárolhatunk.

A három típus közül a *mátrixnyomtatók* a legnépszerűbbek. (Az eladott nyomtatók legnagyobb részét ezek a nyomtatók teszik ki.) A mátrixnyomtatók — másféle csoportosítás szerint — a tús nyomtatók közé tartoznak. A nyomtató legfontosabb része néhány (9–24) darab kis tű, amelyek a nyomtatófejben helyezkednek el. A tűk és a papír között van a nyomtatószalag. A nyomtatási kép úgy jön létre, hogy bizonyos vezérlés hatására a tűk egy része megnyomja a nyomtatószalagot, s ezáltal pontok keletkeznek a papíron. Azért nevezik mátrixnyomtatónak, mert a csoportot képező tűk a jeleket, betűket mátrixként hozzák létre. Aszerint alakul ki a betűkép, hogy bizonyos impulzus hatására melyik tű lendül ki (pontot hozva létre), s melyik marad mozdulatlanul. Minél több tű van, annál szebb a nyomtató rajzolata.

Gondoljunk csak a tv-re! A kép ott is képpontokból jön létre, s minél több ponttal tudjuk leképezni az adott képet, annál jobb lesz a felbontás, annál élethűbb a kép.

Az átlagos mátrixnyomtató 9 tűt használ, a színvonalasabbak 24 tűvel dolgoznak, de az Epson már a 48 tűs változatot (Epson TLQ 4800) is a piacra dobta. Az ár a tűk számától függ.

A mátrixnyomtatók jól alkalmazhatók grafika kinyomtatására is. Különleges karbantartást nem igényelnek, csupán a nyomtatószalagot kell időnként felújítani vagy kicserélni.

Némelyik típus különleges nyomtatószalag alkalmazásával színes nyomtatásra is képes. Ez 6-7 színt jelent. Ilyen pl. az igen közkedvelt Star LC 10 színes változata, a Star LC 10 Color. A 24 tűs nyomtatók talán legjobb változata a kedvező árú Star LC 24-10.

Sokkal kisebb a másik csoport, a *festékes nyomtatók*. Ezek is a tús nyomtatók típusába tartoznak, de itt nem nyomtatószalag hozza létre a nyomot a papíron, hanem a tű festéket lövell a papírra. A mátrixelv itt is érvényesül.

Ezek a nyomtatók csak egyenletes papíron adnak kifogástalan képet. A tintát felszívó és elkenő papíron a kép zavaró lehet. (A magyar papírminőség nem előnyös az ilyen nyomtatók számára.) Ha azonban jó papírt alkalmazunk, akkor a nyomtatási kép határozottan jobb lesz, mint mátrixnyomtatóval (ezért drágábbak ezek a nyomtatók).

A harmadik csoportba a jelenleg legkorszerűbbnek számító *lézernyomtatók* tartoznak. Kiváló minőségük és az előzőekhez képest nagyságrendekkel nagyobb sebességük miatt a professzionális irodai munkára is kiválóan felhasználhatók. Viszonylag magas áruk ellenére is egyre jobban terjednek.

Működési elvük hasonló a fénymásoló berendezésekéhez. A központi egység (ugyanúgy, mint a fénymásolókon) egy szeléndob, amelyre egy lézersugár leképezi a nyomtatandó képet. A nagy sebesség ellenére a felbontás messze meghaladja a mátrixnyomtatókét.

A nyomtatók felbontását *dpi* mértékegységben mérik. A dpi az angol dots per inch (incsenkénti pontok száma) rövidítése. Ez az adat vízszintes és függőleges irányban egyaránt megadja a nyomtatási kép finomságát. Annál jobb a felbontás, minél nagyobb ez a szám, azaz minél nagyobb a pontok száma.

A másik fontos adat (a felbontáson kívül) a sebesség. Ennek a mértékegysége a *cps*. A cps az angol characters per second (másodpercenként kinyomtatott karakterek száma) rövidítése. Ez a sebesség nemcsak a gép műszaki megoldásától függ, hanem az alkalmazott nyomtatási módtól is, ezért legalább kétféle adatot szoktak megadni.

Az egyik a *draft*, amely a gép maximális sebességét jelenti (itt pl. a 9 tús mátrixnyomtatókon a nyomtatási képen jól kivehetők a betűket alkotó pontok). Éppen ezért kitalálták az *NLQ* (Near Letter Quality) nyomtatási módot, amely levélminőséget megközelítő nyomtatási képet ad. A megoldás lényege, hogy a gép többször átír egy betűt, s ezért összefüggőbb, vastagabb lesz a kép. Ez azonban nagymértékben csökkenti a sebességet. Amíg egy átlagos 9 tús mátrixnyomtató *draft* üzemmódban 100–120 cps sebességgel ír, addig ez a szám *NLQ* módban legfeljebb 25–30. A 24 tús gépeken az *NLQ* helyett *LQ*-ről (Letter Quality), levélminőségről beszélünk, ez a több tível elérhető jobb minőségre utal.

A nyomtatók a betűket többféleképpen tudják megjeleníteni, a betű vízszintes kiterjedésétől és a betű rajzolatától (formájától) függően. A vízszintes kiterjedés szempontjából a legáltalánosabban alkalmazott megjelölések a *Pica*, az *Elité*, a *Condensed* és a *Proportional*. A *Pica* és az *Elité* fantázianév, a *Pica* 10 cpi-s, míg az *Elité* 12 cpi-s leképezést jelent. A *cpi* az angol characters pro inch (incsenkénti karakterek száma) rövidítése. A *Condensed* (sűrített) azt jelenti, hogy az adott típusból még sűrűbben kerülnek egymás mellé a jelek (pl. a *Star LC 10* esetében a *Condensed Pica* 17 cpi-t, a *Condensed Elité* 20 cpi-t jelent). A *Proportional* (arányos) leképezés azt jelenti, hogy a normál esettől eltérően a betűk nem ugyanakkora helyet foglalnak el, hanem méretarányosat. Tehát az *i* betű jóval kevesebb helyet igényel majd a nyomtatáskor, mint pl. az *o* betű. Itt cpi-t nem szükséges megadni.

A betű rajzolat szerinti megjelenítése az *ún.fonttípus*. A legelterjedtebb típusok a *Courier*, a *Román*, a *Sanserif*, az *Orator* és a *Prestige*. Igen érdekes a kézírásat utánozó *Script*.

A felsorolt tulajdonságokon kívül számos extra-, ill. luxusszolgáltatás van, pl. a *Paper-park* funkció vagy a fontkazetták használata. Az előbbi azt jelenti, hogy amíg külön lapokra nyomtatunk, addig a perforált papírt nem kell kivenni, hanem "parkoló helyzetbe" lehet állítani.

A nyomtatókkal, ill. a hozzájuk kapcsolódó alapfogalmakkal foglalkozó alfejezetet rövid típusbemutató zárja, ennek alapján bárki kiválaszthatja az Amigához legmegfelelőbb nyomtatót. A fel nem soroltakon kívül is számos kitűnő nyomtató van, a táblázat alapján azonban képet kaphatunk arról, mely adatokra kell figyelni a vásárláskor. A megvétel előtt feltétlenül készítsünk nyomtatási próbát!

Típus	Tű	cps(draft)	cps(NLQ)	A fontok száma
StarLC10	9	120	30	3
Epson LX 800	9	150	25	2
OKIMikroline 320	9	250	62	1
Star LC 24-10	24	142	47	4
Epson LQ 500	24	150	50	2
Epson LQ 850	24	220	73	2
NECP6	24	225	75	6

5.2. A Preferencés és a nyomtatás

A Workbench lemezen számos fájl található, amely a nyomtatáshoz elengedhetetlenül szükséges. Ezekkel azonban eleinte nem kell törődnünk, mert a Preferencessel a nyomtatáshoz szükséges összes paramétert egyszerűen beállíthatjuk. Aktiváljuk tehát a Preferencés ikont, majd az 1.2-es változatban kattintsunk rá a Change Printer feliratú kockára! Az 1.3-as változatban a Printer ikont aktiváljuk! Mindkét esetben ugyanaz az eredmény, a Printer-beállító ablakba jutunk.

Először döntsük el, hogy a nyomtató a soros vagy a párhuzamos porton keresztül működik-e. Az alapbeállítás a Preferencesben a párhuzamos (Parallel), ha a sorosra van szükségünk, kattintsunk rá a Serial felírra. Ezután a jobb oldali, két nyílal jelzett keretben állítsuk be a nyomtató" típusát! A nyilakkal választhatjuk ki a nekünk megfelelőt. Előfordulhat, hogy a felkínált lehetőségek között nincs ott az általunk használt típus. A nyomtatóhoz adott gépkönyvben a gyártók általában több típust is megadnak, ha ezekre állítjuk a rendszert, akkor nem lesz gond a nyomtatással. A Star nyomtatók helyett pl. választhatjuk az Epsont.

Ezután még jó néhány alapbeállítást megváltoztathatunk, ezeket azonban a mai modern nyomtatókon legtöbbször be lehet állítani, vagy a nyomtatáshoz használt szoftverből (pl. szövegszerkesztő) végezhetjük a beállítást. Aki mégis kísérletezni szeretne, annak felsoroljuk az egyes funkciók jelentését.

A Paper Size az alkalmazott papír méretét adja meg. A Paper Type-pal a perforált (fanfold) és a külön lapok (single) közül választhatunk. A betű vízszintes méretét a Pitch határozza meg, a sorok közötti távolságot pedig a Spacinggel állíthatjuk be. A margókat a Left (bal) és a Right (jobb) Margin margóállítóval szabályozhatjuk. Az egy oldalra írható sorok számát a Length adja meg. Mindezek elsősorban a szövegnyomtatás beállítására használhatók.

A grafikus paraméterek beállításához tovább kell lépnünk, és a Graphic Select ablakot kell aktiválnunk. Egy újabb beállító képernyőhöz jutunk, ahol a színes, ill. fekete-fehér nyomtatás beállításán kívül még néhány paraméteren változtathatunk.

Az Aspecttel választhatunk a normál vagy a 90 fokkal elforgatott leképezés között. Az Image Negative beállítása inverz nyomtatási képet eredményez. Ha fekete-fehér nyomtatóval nyomtatunk, akkor a finomság beállítását (amely rendkívül lényeges) a Thresholddal végezhetjük. Ahhoz ugyanis, hogy a képernyőn színesben megjelenő kép fekete-fehérben jelenjen meg, a számítógépnek be kell sorolnia az aktuális színeket vagy a fekete, vagy a fehér közé. A nyomtatási kép felismerhetőségét az határozza meg, hogy melyik színeknél húzzuk meg a határt.

Az 13-as változatban még több lehetőség van. A grafikát a Graphic 1 és Graphic 2 jelzésű ablakkal állíthatjuk be. A Graphic 1 majdnem megfelel az 1.2-es változat állítási lehetőségeinek, a Graphic 2-vel tovább finomíthatunk a grafikán.

Akinek még többre van szüksége, az használja a Turboprint program — pontosabban a legújabb változat, a Turboprint II — variációs lehetőséget. Ez többek között számos nyomtató specifikációját (azaz meghajtóját) tartalmazza, pl. a Star LC 10-ét vagy a Hewlett—Packard lézernyomtatóét, de említhetnénk még a Star 8 lézernyomtatót is. A programmal a grafika gyorsabban kinyomtatható.

Mielőtt e viszonylag hosszú elméleti bevezetés után rátérnénk a nyomtatás gyakorlására, még egy fogalommal meg kell ismerkednünk, a *hardcopyval*. A nyomtatandó anyag ugyanis két forrásból származhat. Nyomtathatunk a fájlból (ez lehet szövegfájl vagy olyan Basic program, amely valamilyen grafikát jelenít meg). A másik módszerrel — ez a hardcopy — a képernyőn levő képet vagy szöveget nyomtatjuk ki, a képernyőképpel megegyezően. Ez a módszer sokszor hasznos lehet, ha egy adott képernyőképet (pl. egy program menürendszerét) szeretnénk a gép kikapcsolása után tüzetesebben megvizsgálni.

5.3. Nyomtatás a CLI-ből

A legegyszerűbb nyomtatási módszer, ha az Extrás lemezen (az 1.3-as változatban a Workbench lemezen) található Printf iles nevű programot alkalmazzuk:

Printfiles fájlnev

Hasonló eredményt érhetünk el az ismert Type parancs alkalmazásával, ha a parancsot a nyomtató felé irányítjuk:

```
Type >prt: név.
```

A *név* a kinyomtatandó szövegfájl neve.

A Printfiles és a Type esetén (a dfO-t kivéve) is pontosan meg kell adni a nyomtatandó fájl helyét.

Hasonló megoldással a tartalomjegyzéket is kinyomtathatjuk:

```
Dir >prt: dfl:c
```

Ezzel az utasítással a papíron a *dfl* lemez meghajtóban levő lemez *c* tartalomjegyzékét kapjuk meg.

Logikusan következik, hogy a List parancssal is megtehetjük ugyanezt, sőt számos más parancs is a nyomtató felé irányítható, ha megjelenítenek valamit a képernyőn.

A papír tetején a dátumot a Date >prt: utasítás írja ki.

Ismerkedjünk meg egy bonyolultabb, de nagyon jól alkalmazható eljárással! Ehhez a Copy parancsot hívjuk segítségül:

```
Copy * to prt:
```

Ezzel a sorral a képernyőn megjelenő valamennyi szöveget kinyomtathatjuk a nyomtatóval. A szöveget soronként kell írni, ha a sor végére érünk és megnyomjuk az Entert, akkor az adott sort a nyomtató kinyomtatja. Tehát soronként nyomtathatunk, és mindaddig, amíg az Entert le nem nyomjuk, javíthatunk az adott soron.

A szöveg pozicionálása is pontosan meghatározható. Tegyük be egy üres papírlapot a nyomtatóba, majd egytől kilencig nyomjuk le a számokat, végül az Enter lenyomásával nyomtassuk ki azokat. Látni fogjuk, hogy a számok hova kerültek. Ezek után ezzel összehasonlítva mindig be tudjuk állítani a szöveget. Ha azt akarjuk, hogy a sor a 4-es számnak megfelelő helyen kezdődjön, akkor a szöveg begépelése előtt nyomjuk le háromszor a Space billentyűt. A módszer nagyon jól alkalmazható olyan egyedi feladatokban, ahol viszonylag rövid szöveget (esetleg csak néhány szót) kell írni, viszont a pozicionálás igen fontos.

Előfordulhat az is, hogy a CLI-ben dolgozva néhány soros üzenetet akarunk írni. Ekkor túl bonyolult lenne szövegszerkesztőt alkalmazni. Módszerünk ekkor is jó, sőt bizonyos egyedi elvárásoknak is eleget tud tenni.

A nyomtatandó betű típusát vezérlőjelekkel megváltoztathatjuk. A vezérlőjel az Esc billentyű lenyomásakor indul, innen tudja a gép, hogy a következő néhány jel nem kinyomtatásra vár, hanem vezérlőkód. Nézzünk néhány példát!

```
Esc[4m      aláhúzott szöveget kapunk;
```

```
Esc[ 1 m    vastag, ún. boldface szöveget kapunk;
```

```
Esc[3m      dőlt betűs, ún. italics szöveget kapunk;
```

```
Esc[6w      kétszeres nagyságú betűket kapunk;
```

```
Esc[Om      visszatérés a normál betűkhöz.
```

Az 1.2-es változatban figyeljünk a vezérlőkódok alkalmazásakor, mert sem az Esc, sem a vezérlőkódok nem jelennek meg a képernyőn — tehát vakon kell bebillentyűzni —, és csak az

eredményből derül ki, hogy valamit elrontottunk. Az 1.3-as változatban mindent látunk. Ha egy sort elrontottunk, akkor azt a Ctrl+X billentyűvel törölhetjük.

A nyomtatás befejezésekor a Ctrl és a \ billentyű lenyomásával visszatérhetünk az alap CLI üzemmódba.

5.4. Nyomtatás Basicból

Ha a Basicból nyomtatunk, akkor két feladatot kell megoldanunk. Az egyik, hogy a Basic program futása közben bizonyos szövegeket, esetleg táblázatokat kinyomtathassunk. A másik feladat a Basic program listájának kinyomtatása. Ez utóbbi az egyszerűbb, kezdjük ezzel! Adjuk ki az **LLIST** parancsot! Ugyanúgy, mint a LIST parancs esetében, ha nem az egész listára vagyunk kíváncsiak, akkor megfelelő tól-ig kiegészítéssel kell a parancsot alkalmazni:

```
LLIST 500-2000
LLIST adat-
```

Az első utasítás az 500-as és a 2000-es sorszámú sor közötti programrészt jeleníti meg a papíron, a másik az *adat* nevű címkétől kezdődően a program többi részét, egészen a program végéig. Ehhez hasonlóan kapjuk az *L* betű hozzáadásával a PRINT parancsból a futás közbeni nyomtatáshoz szükséges **LPRINT** parancsot. Ez a PRINT-hez hasonlóan működik, de a szöveg a papíron jelenik meg. A szövegnek vagy egy sztringváltozóban kell lennie, vagy idézőjelben (haazLPRINTutánáll).

Bizonyos formátumot is meghatározhatunk, mert a **WIDTH** parancssal megadhatjuk a nyomtatási sor szélességét. Az alapbeállítás 80 karakter. Hiába férne a papírra több, a nyomtatás új sorban folytatódik. Ezenkívül a nyomtatási tabulálást is megoldhatjuk a **WTDTH** parancssal, mert a szélesség mellett egy másik paramétert, az ún. nyomtatási zónát is meg lehet adni:

WIDTH szélesség, nyomtatási zóna

A nyomtatást az **x=LPOS(n)** függvénnyel vezérelhetjük. Az *x* értéke az utoljára kinyomtatott jel pozícióját adja meg. Az **IF LPOS(n)=60 THEN LPRINT CHR\$(13)** Basic sor azt jelenti, hogy ha nyomtatás közben a sorban elértük a 60. karaktert, akkor egy Enter következik, amely a sor vége (új sor). Az Enter ASCII kódja a **CHR\$(13)**.

Itt említjük meg azt a lehetőséget, amellyel a mátrixnyomtatókon megoldhatjuk valamennyi magyar ékezet használatát. Az a nehézség, hogy sajnos a Setmap parancssal nem tudunk magyar ékezetes billentyűkhöz jutni. A Setkey program alkalmazásával azonban létrehozhatjuk a *h* betűvel jelölt keymaps fájlt. Csak a nemzetközileg elfogadott ASCII karakterek közül válogathatunk, ami viszont nem teszi lehetővé a hosszú *ö*, *ül*. *ű* megjelenítését. Ezen fáradtságos munkával és egy font editorral segíthetünk. Így már megkaphatjuk a teljes magyar ábécét, de a nyomtatás még mindig nem lesz tökéletes.

Sajnos a **CHR\$(128)** feletti ASCII karaktereket a nyomtatók nem egyformán értékelik, bizonyos eltérések lehetnek. Előfordulhat, hogy a nyomtató a képernyőn megjelenő *á* betű helyett más jelet, vagy semmit sem nyomtat ki. A font editorral létrehozott új karakter esetében pedig annak a billentyűnek a jele (azaz betűképe) fog megjelenni, amelyiket a font editorral átalakítottunk. Aki azonban nem sajnálja a fáradtságot, az a nyomtatóhoz adott kézikönyvvel és egy Basic programmal megoldhatja a feladatot. Természetesen a Basic programot mindenkinek saját magának kell megírnia.

Egy karaktergenerátor Basic programot (download character generátor) kell írunk. Ehhez azt kell tudni, hogy a nyomtatónak is van ROM memóriája, amelyben az egyes karakterkészleteit tárolja. Amikor nyomtatunk, akkor a megfelelő vezérlőjelek innen jönnek. A nyomtatónak azonban van egy RAM memóriája is. Bizonyos parancsokkal elérhetjük, hogy a nyomtató a ROM-ból egy karakterkészletet (pl. a draftot) letöltsön (download) a RAM-ba, majd a nyomtatáskor a karaktereket innen, és ne a ROM-ból vegye. Ha ezt megoldottuk, akkor csak annyi a feladat, hogy az egyes karaktereket úgy, mint a font editor esetében, átváltoztassuk a nekünk megfelelő betűvé (pl. hosszú ü-vé). Ezt azért tehetjük meg, mert az egyes betűk tulajdonképpen pontmátrixok. A nyomtatóhoz kapott kézikönyvben megtalálhatjuk, hogy az adott nyomtató hogyan hozza létre a pontmátrixot, és hogyan tudjuk azt megváltoztatni. Az elmondottakat egy Basic programmal megvalósítva, és a nyomtatás előtt a Basic programot lefuttatva a kérdést megoldottuk.

Van egy radikálisabb, de szakembert igénylő módszer is, a ROM csere. A nyomtató eredeti ROM-ját egy már átalakított, magyar ékezeteket tudó ROM-ra cseréljük.

5.5. Hardcopy

Sajnos az Amiga esetében a hardcopy nem oldható meg olyan elegánsan, mint az IBM PC-n, ahol két billentyű lenyomása után már kezdődik is a képernyő tartalmának kinyomtatása. A hardcopyhoz a Workbench lemezen található GraphicDumpot kell alkalmazni. A program aktiválása után mintegy 10 másodpercünk van arra, hogy a kinyomtatandó ablakot (vagy ablakokat) kinyissuk, **III** az előtérbe hozzuk.

Ha a Basic programból szeretnénk megvalósítani a hardcopyt, akkor egy könnyen beépíthető hardcopy modult találhatunk az Extrás lemez BasicDemos fiókjában, ScreenPrint néven.

5.6. Nyomtatás szoftverből

Talán ez a leggyakoribb nyomtatás. Ilyenkor a beállítással nem nagyon kell törődnünk, mert a programkészítők erről gondoskodtak. Csupán azt kell megnéznünk, hogy a mi nyomtatónk megfelel-e a gyári beállításnak. A profi szoftver azonban ezt az igazítást is megoldja.

A szövegszerkesztő programokkal általában nem lesznek nehézségeink. A grafikus programok — ha színesben akarunk nyomtatni — mindenképp több figyelmet igényelnek majd.

6. SZOFTVEREK

A szoftver a számítógép alapja. A legkorszerűbb számítógép sem ér semmit a szoftver, a futtatható program nélkül. A szoftvert magunk is készíthetjük. Egyszerűbb feladatok megoldására nem is nagyon találunk gyári programokat. A szoftverek nagy része azonban szoftvergyártóknál készül, és forgalmazókon, eladókon keresztül jut el a felhasználóhoz.

Az Amiga a szoftverellátottság szempontjából kezdetben mostohagyereknek számított, ma azonban hatalmas választék áll rendelkezésünkre. Ez a nagy választék egyrészt rendkívül hasznos, hiszen a legkülönlegesebb igények kielégítésére is alkalmas. Másrészt viszont megnehezíti a választást, nehéz eldönteni, hogy az azonos jellegű (és sokszor közel azonos árú) termékek közül melyiket válasszuk, melyik fog az igényeinknek legjobban megfelelni. A legnehezebb a Public Domain (e. páblik domen) szoftverdsungelében eligazodni.

A Public Domain (PD) (más néven Free Soft) a szoftver különleges területe. Az alapvető különbség az, hogy a Public Domain programok szerzői jogvédelem nélkül kerülnek forgalomba, tehát bárki korlátozás nélkül másolhatja és továbbadhatja azokat. (Pénzért forgalmazni tilos.) Természetesen valamennyit ezekért a szoftverekért is fizetni kell, mert ezek általában a PD-terjesztőktől szerezhetők be, akik több ezer katalogizált PD lemezt tartanak raktáron. Becslések szerint ma már több mint 2000 PD lemezt forgalmaznak. Ez mindenképpen indokolja, hogy a könyv foglalkozzon a fontosabb programokkal.

Még annyit kell tudni, hogy a lemezek nem a rajtuk lévő programok valamelyikének címével futnak, hanem a lemez összeállítójának nevét viselik. Ilyen összeállító bárki lehet, hiszen jogi korlátozás nincs. Ez azonban mégis profi munka, hiszen nagyon sok amatőr programozóval kell kapcsolatban lenni ahhoz, hogy komoly összeállítást készíthessünk. Ezért általában klubokban alakultak PD-gyűjtő helyek, és a lemezek is rendszerint a klubok nevét viselik. A név mellett egy sorszám is található, hiszen ezek a klubok több tucat lemezt is összeállítottak, és így jelentős sorozatok jöttek létre.

Az első összeállítók egyike az amerikai Fred Fish, neki van ma a legnagyobb sorozata. Sorozatának a Fish nevet adta.

Ha be akarunk szerezni egy programot, akkor a sorozat nevét és sorszámát kell tudnunk (pl. Fish 112.). A szoftverek ismertetésekor így jelöljük majd a PD programokat.

A szoftverek bizonyos területek szerint kategorizálhatók, a jobb eligazodás kedvéért a számítógépes üzletek is ilyen kategóriákba sorolva hirdetik, árulják a programokat. Mi is ezt a csoportosítást próbáljuk meg áttekinteni.

6.1. Programnyelvek

Talán az lesz a legkézenfekvőbb, ha először a géphez legközelebb álló programokról szólunk, nevezetesen a programnyelvekről. Az Amiga-Basic a gép tartozéka. Számos Basic változat létezik, amelyeknek mind az a céljuk, hogy az Amiga-Basicnél gyorsabbak, hatékonyabbak — tehát még kényelmesebbek — legyenek. Ilyenek a GFA Basic, a SAM Basic, a True Basic és a Pecan Basic.

Megjelentek a piacon a Basic *compiler*ek (ejtsd: kompájler) is. A compiler a Basic programot lefordítja gépi kódba (vagy ahhoz közeli formába), ezzel jelentősen megnöveli a program futási sebességét. A kapható két fontosabb compiler az A/C Basic és a Hisoft Basic-Compiler.

A géphez legközelebb a gépi kód áll. A gépi kódban írt programok a leggyorsabbak. Miért programoznak mégis csak kevesen gépi kódban? A válasz egyszerű; nagyon bonyolult és egyhangú. Ehelyett az *Assembler* programozás terjed. Ezeket a programokat az Assembler compilerével könnyen gépi kóddá alakíthatjuk. Aki a programozással mélyrehatóbban akar foglalkozni, az a Basicen kívül szerezzen be Assemblert is. A következők kaphatók: Seka Assembler, Devpac Assembler, Profimat. Aki nem akar sok pénzt áldozni, és megelégszik szerényebb szolgáltatásokkal, annak a Fish 110 lemezen található PD programot, az A68k-t ajánljuk.

Az Amigára legjellemzőbb — és a programozók között is népszerű — magasabb szintű nyelv, a C. Az Amiga rendszerszoftverének jó része is ezen a nyelven íródott. A piacot két nagy márka, a Lattice C és az Azték C versenye uralja. (Egymás után rukkolnak ki az újabb és újabb változatokkal.) A szerényebb lehetőségekkel rendelkezők ebben az esetben is találnak egy PD programot a Fish 110-en, a Pdc-t.

Gyorsan terjed egy viszonylag új programozási nyelv, a Modula—2. A nyelv a Pascalból fejlődött ki, s mint ahogy neve is mutatja, a lényege a moduláris — tehát különálló egységekből (modulokból) felépülő — program. A leggyakrabban használt szoftver az M2 Modula—2, amelynek ma már számos segédprogramja is van. Az M2 Modula PD verziója a Fish 113 lemezen található.

A Pascal ugyancsak magas szintű programozási nyelv, két változatát említjük: az UCSD Pascalt és az ISO Pascalt.

Ezenkívül vannak az Amiga szempontjából kisebb jelentőségű nyelvek is: a Prolog, a Forth, és a Fortran.

Nem programozási nyelv, de itt kell megemlíteni az Assemblerben programozók egyik fontos szoftverét, a DSM Disassemblert. Ez az Assemblerrel ellentétesen működik. Eddig ugyanis az assemblált programot, ha az *ún. forráskód* (a program még nem lefordított szöveges formája) nem állt rendelkezésünkre, nem tudtuk megnézni. A disassembler sok programozónak nyújthat segítséget.

6.2. Grafika

Az Amigára készült szoftverek valószínűleg legnagyobb csoportját a grafikát megvalósító programok képviselik. Az Amiga megjelenése óta kiállításokon, bemutatókon mindig a legfőbb látványosság valamilyen szép grafika, ill. animáció bemutatása. Nem hisszük, hogy akad olyan Amiga-tulajdonos, akinek nincs ilyen programja.

A gép alkalmas a képi megjelenítésre, gépi kódban és Basicben azonban kevesen készítenek grafikát. A profi programozók is legtöbbször saját készítésű, ún. fejlesztő segédprogramokat használnak.

A grafikus programoknak is az a feladatuk, hogy az Amiga nyújtotta lehetőségekkel — az intuícióval — a lehető legegyszerűbbé tegyék a grafika megrajzolását. A programok számos eszközzel — előre elkészített elemekkel, ill. mintázatokkal — könnyítik meg a rajzolást. A részletek megrajolásakor ezeket ki tudjuk nagyítani. A szint különböző segédeszközökkel állítjuk be. A legtöbb programban természetesen a hardcopy lehetőség is adott.

Az első — és a mai napig legnépszerűbb — grafikus program az Electronic Árts Deluxe Paint (DPaint) nevű szoftvere, amely már a IH-as változatban kapható. Ez a program tulajdonképpen szabványnak is tekinthető. Tud mindent, ami a rajzoláshoz elengedhetetlen, és a kezdő felhasználó is könnyen megtanulhatja a kezelését. A munkát kitűnő, rolós menürendszer, valamint a képernyő jobb oldalán elhelyezkedő szimbólum- menürendszer könnyíti. Ez utóbbi

a grafikus programok igen kedvelt eszköze, hiszen ha az egyes funkciókat szövegesen megnevezve kizárólag csak rolómenüként próbálnák beépíteni a program írói, akkor gondjaik lennének a helyyel. Ezért — kihasználva az intuáció adta lehetőségeket — az egyes funkciókat találó ábrával jelölik. Ha pl. egy kézinagyítót látunk, akkor arra az egérrel rákattintva, a zoom funkciót, egy tárgy részletének kinagyítását érhetjük el.

A DPaint II a DPaint I javított változata, a DPaint III azonban — megtartva az elődök kitűnő tulajdonságait — jelentős újításokat tartalmaz. Vele már animációt is készíthetünk, sőt az animáció sebességét is szabályozhatjuk.

A DPaint változatok 4096 színt alkalmaznak, de a 4096 kikeverhető színből egyszerre csak 32-t tudnak használni. A DPaint III ebben is újítást hozott. Egy ügyes trükk — az Extra Half Bright módszer — segítségével a felhasználható színek megduplázzhatók, azaz a 32 színen kívül azok fele olyan világos változatai is alkalmazhatók. (Ha meggondoljuk, hogy a 32 szín sem 32 különböző szín, mert közülük több csak árnyalatban különbözik egymástól, akkor ez a módszer valóban 64 színt megjelenítését teszi lehetővé.) Érdekes újdonság, hogy a DPaint III-mal a lemezre való letöltéskor kérhetünk ikon nélküli (.info fájl nélküli) letöltést. Ezzel helyet takarítunk meg, és azt tapasztaljuk, hogy a letöltés és a betöltés is gyorsabb lesz.

Már klasszikusnak számító program a Photon Paint. A program érdekessége, hogy a képernyőn egyszerre megjeleníthető a lehetséges összes szín, tehát 4096-féle. A programban van zoom funkció és több más beépített segédeszköz.

A Digi Paint programmal mind a 4096-féle színt alkalmazhatjuk. Számos elegáns funkciójával lehetővé teszi digitalizált képek továbbalakítását, de önálló grafikus programként is megállja a helyét.

Az Electronic Arts programja a Photolab is, mely három különálló programból áll. Az első egy festőprogram, amely mind a 4096-féle színt használja, a második az utómunkálatok elvégzését segíti, a harmadik pedig a grafika kinyomtatását vezérli.

Viszonylag egyszerűbb program az Accolade (a Test Drive játékprogram készítője) Graphics Stúdiója. Sajnos a program nem használja ki a PAL rendszer felbontási lehetőségét.

Az Express Paint maradéktalanul kielégíti a professzionális igényeket. A működéséhez legalább 1 Mb-ajt memóriára van szükség. A program érdekessége, hogy alkalmas színes fontok fogadására, és használja a virtuális képernyőt, amihez a RAM-ot veszi igénybe.

A grafikus programok közül érdemes megjegyezni néhány hasznosat a régebbiekből (ezek feltehetőleg már olcsóbbak): a Prismot, a Butchert és a Pixmate-et.

6.3. RayTracing

A grafika témaköréhez tartozik a *Ray Tracing* (szabad fordításban: fénysugárkövetés). A Ray Tracing program a grafikus programoknak az a sajátos fajtája, amely a kép élethűségét úgy próbálja megvalósítani, hogy szimulálja a valóságban is meglévő fénytükrozódéseket. így válik térszerűvé, plasztikussá a megformált tárgy. Igazi számítógépes feladat!

Ennek a kategóriának egyik legjobb képviselője a Sculpt/Animate 3D, és annak legújabb változata, a Sculpt/Animate 4D. Hasonló program a háromdimenziós animációra is képes Forms in Flight és a DBW Render (PD). De itt már átléptünk egy másik határterületre, a videó, ül. az animáció területére. (Természetesen éles határokat húzni nem lehet.)

6.4. Animáció és videó

Ezek a programok a grafikából nőttek ki, de ma már önállóan is használhatók. Számos videostúdió a trükkök megalkotását az Amigára és egy-egy jó szoftverre bízta. (Azért egy- két hozzáértő szakember is bizonyára elkel.)

Az animáció lényege a mozgókép létrehozása. A legegyszerűbb animáció, amikor néhány összefüggő képet diashowként levetítünk. Ez volt az animáció első lépése, az elv azonban ma sem változott. Az első ilyen programok egyike a Videoscape 3D volt.

Az animáció rendkívül memóriaiigényes, és a sebesség is fontos. Ezért aki az animációval professzionális módon akar foglalkozni, annak mindenképpen memóriabővítővel kell ellátni az Amigát, és egy különleges sebességnövelő processzorkártyát is ajánlatos beépíteni (csak az Amiga 2000 esetében).

A videoprogramok másik fontos feladata a feliratok készítése (pl. filmek főcíme vagy emblémák.) A feliratkészítésre vannak megfelelő programok. Az egyik ilyen — igen sok fontot és ügyes lehetőségeket kínáló (mégsem bonyolult) — program a TV Text. A Videotitler programmal már futófeliratot is készíthetünk. A két lemez meghajtót és legalább 1 Mbájt memóriát igénylő programmal a hatalmas fontkészletnek köszönhetően jó minőségű feliratok készíthetők.

Az Aegisnek a grafikát, animációt és videózást is magában foglaló programválasztéka van. A programok a következők: Animagic, Animator, Art Pak, Draw, Draw Plus, Images, Impact, Lights Camera Action, Modeler 3D, Videoscape 3D, Videotitler. Ha valaki ezeket a programokat (vagy legalább egy részüket) megveszi, akkor mindent megvalósíthat az ismertetett területen, azzal az előnnyel, hogy az Aegis programok képesek fogadni azokat az anyagokat, amelyeket egy másik Aegisszel hoztunk létre.

Kizárólag a billentyűzetet használva dolgozik a Pro Videó Plus feliratozó program.

Különleges a Videó Wipe Master program, amellyel különféle videotrükköket hozhatunk létre, két videokamera képét keverve. Ehhez a programhoz és a legegyszerűbb feliratozóhoz is be kell szerezni azonban egy drága készüléket, a genlockot.

6.5. CAD

A grafikus programok különleges részterületét alkotják a CAD programok. A CAD az angol computer aided design (e. kompjuter édid dizájn) rövidítése, és számítógépes tervezést jelent. Rövidesen nem lehet majd elképzelni ilyen program nélkül egyetlen profi tervezőirodát sem. Egy jól kifejlesztett CAD program ugyanis megkönnyítheti a mérnöki tervező munkát, és eddig nem ismert lehetőségeket lehet vele kiaknázni.

A CAD programok nem olcsók, kifejlesztésük komoly munka. Az Amigára készült szoftverek közül kiemelkedik az X-CAD. A programot a tervezői az MS-DOS alatt futó híres AutoCAD konkurenciájának szánták. (Ismerve az Amiga képességeit, erre jó esélyük volt.) Nem kezdőknek készült, több száz oldalas kézikönyv írja le a használatát. A program sokrétűségét jellemzi, hogy kifogástalan működéséhez legalább 2 Mbájt memória szükséges és egynél több lemez meghajtó. Az X-CAD Interlace módban működik, ezért megfelelő monitorra van szükség, nehogy a vibrálástól megfájduljon a szemünk.

Hasonlóan igényes program a Dynamic-CAD is. Kifogástalan működéséhez az alapkonfigurációnál több memória szükséges.

Kezdőknek is ajánlható a viszonylag olcsó Intro-CAD.

A grafikával foglalkozó alfejezetek befejezéséül még egy érdekes programról szeretnénk szólni, amely nem sorolható be egyik kategóriába sem. A programmal nem lehet létrehozni grafikát, azonban egy másik programból (pl. játékprogramból) kimenthetjük a nekünk tetsző képet. A Supervisor programmal átpáztázhatjuk a számítógép memóriáját is, és ha megfelelő képre bukkanunk, akkor azt is kimenthetjük.

6.6. Szövegszerkesztés

Nemcsak az Amigának, hanem valószínűleg minden számítógéptípusnak fontos felhasználási területe a szövegszerkesztés. Mit csinál egy szövegszerkesztő? Lehetővé teszi, hogy a képernyőn írott szöveget hozzunk létre, és azt addig javítsuk, amíg nyomtatható lesz. A nyomtatás helyett azonban a szöveget lemezre is vihetjük.

Ha a hagyományos írógéppel hibát követünk el (vagy később szeretnénk átírni a szöveget), akkor fáradtságos munkával újból le kell gépelnünk az egészet. A szövegszerkesztővel ezt elkerülhetjük. A jó szövegszerkesztőnek számos olyan funkciója is van, amely a szöveg megtervezésétől a kinyomtatásig segítségünkre lehet.

Mielőtt áttekintenénk a fontosabb programokat, nézzük meg, melyek azok a fontosabb funkciók, amelyekkel egy jó szövegszerkesztőnek rendelkeznie kell.

Legelőször a *WYSIWYG elvet* említenénk, amely az angol what you see is what you get (amit látsz, azt kapod) rövidítése. Ez azt jelenti, hogy a képernyőn megjelenő szövegformátum lehetőség szerint egyezzen meg a kinyomtatandó formátummal.

A formátum kialakításához kapcsolódik a Notepadnél már megismert word wrap. Számos program kezeli az *elválasztást* (hyphenation), azaz a sor végére érve nem kell törődnünk az elválasztással, mert a program elvégzi helyettünk. Szorosan ehhez kapcsolódik a program azon képessége, hogy ha pl. beszúrunk egy szót, akkor képes az egész szöveget úgy átrendezni, hogy a *beszúrás* miatt egyik sor se lesz sem rövidebb vagy hosszabb.

A jó szövegszerkesztő megengedi a választást, hogy inzert üzemmódban dolgozzunk-e vagy sem. Ha *inzert üzemmód* esetén a szövegbe bárhol be akarunk még írni, akkor az ott lévő szöveget nem törli a program, hanem tolja maga előtt.

A szövegformátumhoz tartozik, hogy szabadon beállíthassuk az egy sorba írható karakterek számát, a sorhosszt. Ezenkívül az egy oldalra írható sorok számát, a lap tetejétől, ±11- aljától számított margót, és mindent, ami a nyomtatási képet kialakítja, szabadon kell tudnunk állítani, lehetőség szerint nemcsak közvetlenül, hanem a szövegen belül is, az erre a célra kitalált *pontparancsokkal*.

A szerkesztői munkában nélkülözhetetlen, hogy két vagy több szövegből egymásba átvigyünk részeket.

Nagyon hasznos az ún. *keres-cserél* (find and replace) funkció, amellyel az általunk meghatározott szövegrészletet kikereshetjük, és az ugyancsak általunk meghatározható másikkra cserélhetjük. Előfordulhat pl., hogy egy szövegben az összes *vagy* szót *és-it* kell cserélni. Egy 100 oldalas szöveggel jó ideig kellene bajlódni, és figyelmetlenségünkben könnyen kihagyhatunk egy-két szót.

A komolyabb szövegszerkesztőtől elvárjuk, hogy sorozatlevelek írására (*mail mérge*) is alkalmas legyen. Ha ugyanazt a levelet akarjuk elküldeni több vállalatnak, akkor nem kell mindegyiket külön-külön megcímezni. Ezt címlistánk alapján a mail mérge funkció elvégzi.

Számos olyan funkció is van, amely nem feltétlenül szükséges, de bizonyos feladatok megoldásakor kapóra jön. Amikor szövegszerkesztőt választunk magunknak, akkor a

szövegszerkesztő képessége mellett mérjük fel azt is, hogy milyen feladatra fogjuk alkalmazni. A professzionális szövegszerkesztőket elég bonyolult kezelni, s ha csupán magánlevelezésre használjuk, akkor biztos nem ez az igazi megoldás. Mindig a feladatnak megfelelő bonyolultságú programot válasszunk!

A legkorábbi szövegszerkesztők egyike a Micro-System Scribble! nevű programja. Ez viszonylag egyszerű, de **jól** használható program. Nagy előnye a gyors elsajátíthatóság és a könnyű kezelhetőség. Kiválóan alkalmas levélírára és tanulmányok, szövegek elkészítésére is. A program a Works! programcsomag része, amelybe egy adatbázis-kezelő (Organize!) és egy táblázatszerkesztő (Analyze!) is tartozik. A programcsomagot eredetileg IBM számítógépen használták, később pedig az Amigára is lefordították. A Scribblek a 2.10-es változattól felfelé érdemes beszerezniük.

A német nyelvterületen a két legelterjedtebb program a Data Becker Beckertext programja, **III** annak egyszerűbb változata, a Textomat. A Beckertext a sokat író profi felhasználóknak is ajánlható, rendelkezik a mail mérge funkcióval és egy szótárral. Mindkét programba beépítették az automatikus elválasztást (természetesen csak német nyelven). A Beckertext használatához 1 Mbájt szükséges.

A professzionális kategóriát közelíti a Micro-System másik szövegszerkesztője, az Excellence!, valamint a New Horizons Prowrite programja. Mindkettő támogatja a színes nyomtatást, és számos luxusfunkciója van. Az Excellence! esetében maradéktalanul érvényesül a WYSIWYG elv.

A ma kapható szövegszerkesztők közül a legprofesszionálisabb a szinte minden igényt kielégítő Wordperfect, amely angol és német változatban is kapható, de rendkívül drága. A német változatba mintegy 415 000 szót tartalmazó szótárt építettek be. A program legalább 1 Mbájt memóriát igényel.

Rétegigényt elégít ki a TEX program, amelynek különlegessége a különféle számok, képletek és egyedi írásjelek megjelenítése. Ebből már kiderül, hogy a természettudományok területén tanulmányokat, cikkeket készítőik számára nyújt különleges szolgáltatást. Természetesen benne van a hagyományos szövegszerkesztőtől elvárható összes alapfunkció is.

Végül megemlítenek egy egyszerűbb, főképp a kezdők számára alkalmas programot, a Markt & Technik Documentum nevű termékét. Ezt könyvként árulják, programlemezrel együtt. Ez az egyik képviselője az egyre terjedő ún. bookware (e. bukver) termékeknek. A Documentum programot azonban nemcsak kezdők használhatják sikerrel.

6.7. Kiadványszerkesztés (PTP)

A DTP az angol *Desktop Publishing* (e. deszktop páblising) rövidítése. A DTP tulajdonképpen a szövegszerkesztőből fejlődött ki, és magyarul íróasztal mellőli publikálást, kiadványszerkesztést jelent. A jó DTP programmal nyomdakész újságot, könyvet vagy bármely más kiadványt lehet szerkeszteni.

A nem túl jelentős luxusfunkciókon kívül a DTP programmal grafikákat, ábrákat építhetünk be a szövegbe. Az Amigára készült DTP-k közül a legkomolyabb a Professional Page, amely legalább 1 Mbájt memóriát igényel. A Professional Page kiadója — a Gold Disk — hasonló minőségű, de jóval olcsóbb és egyszerűbb programmal is képviselteti magát a piacon, ez a PageSetter. Ugyancsak a legjobbak közé tartozik a Publisher Plus, a Shakespeare és a Page Stream.

6.8. Adatbázis-kezelők

Az adatbázis-kezelő programokat elsősorban nem személyes célokra szánták, hanem inkább bizonyos irodai munka elvégzésre. (Természetesen személyes célra is kiválóan használhatók.) Így az adatbázis-kezelőkkel nemcsak ismerőseink címlistáját, hanem videokazettáinkat vagy kedvenc újságunk régebben megjelent cikkeinek címét is nyilvántarthatjuk. Csak a fantázia szabhat határt a felhasználásnak.

Az adatbázis-kezelők alapfeladata adattömegek tárolása, különféle szempontok szerinti csoportosítása, ill. a képernyőn (vagy nyomtaton) való megjelenítése. Mindezt még sok adat esetén is a lehető legnagyobb sebességgel kell végezni, mert a felhasználó nem szeret várni pl. egy rendezésre.

Az adatbázis-kezelők vagy soros, vagy közvetlen hozzáféréstű fájlt használnak. Ez utóbbi esetben igen nagy jelentősége van a jó indexelésnek. Bizonyos követelmények szerint (amelyeket a felhasználó jelöl ki) az adatokról listát kell készíteni, ennek alapján a gép a teljes adatsort könnyen megtalálhatja.

Az Amigára sok adatbázist készítettek, de két programcsalád különösen kiemelkedik a többiek közül. Azért kell programcsaládról beszélni, mert nem egy programról, hanem sorozatról van szó, amelynek első tagja az amatőr felhasználóknak készült, míg a legdrágább változat a professzionális igényeket is kielégíti. A programok természetesen lefelé kompatibilisek, azaz a drágább program mindig tudja fogadni az olcsóbb (tehát egyszerűbb) programmal készített adatbázisokat. Így ha valaki kinövi a programot, és váltani szeretne, nem kell az adatait újból bevinni.

Az egyik ilyen család a Superbase sorozat: Superbase, Superbase 2, Superbase Professional, Superbase Professional Developers Package. A legegyszerűbb változatot bookware-ként forgalmazza a cég. Ezzel a közvetlen hozzáféréstű fájlokkal dolgozó programmal hatalmas adatbázisok is létrehozhatók, csak a lemez határozza meg az adatok mennyiségét. Az adatok számtalan módon megjeleníthetők. Figyelemre méltó a program ún. szűrési technikája, amellyel az adatokat bizonyos szempontok szerint csoportosítani lehet. A 2-es változat érdekessége, hogy megtartva az első összes tulajdonságát, képes digitalizált képeket is fogadni és összekapcsolt adatbázisokat létrehozni. A Professional változat kiegészül egy saját, Basichez hasonló programnyelvvvel. A negyedik változatot profi fejlesztőknek ajánlják. Ez a program tartalmaz egy *runtime* (e. rántájm) modult is, amellyel a létrehozott adatbázis akkor is futtatható, ha a szoftver nincs a gépben. Ez azért fontos, mert ha valaki megbízás alapján egy vállalatnak készít programot, akkor a felek sem technikai, sem szerzői jogi okokból nem kényszerülnek a drága szoftver külön beszerzésére.

A másik nagy család a Data Becker Datamat programja: a Datamat, a Datamat Plus és a Datamat Professional. Már a címek is jelzik, hogy ez is hasonló elveken felépülő sorozat. A Superbase-zel ellentétben ez a program soros fájlokat használ. A Professional változatnak ugyanúgy saját programozási nyelve van. A Datamat kezelése talán egyszerűbb, és az amatőr felhasználó számára előnyösebb. A Superbase-zel ellentétben a Datamat számos betűtípusot használ, és maszkeditorja támogatja a színes maszkok létrehozását. (A maszk az a képernyő-elrendezés, amelyben az adatok megjelennek, ill. amelyen keresztül az adatokat bevihetjük. Ennek alapján megkülönböztetünk beviteli, képernyő- és nyomtatási maszkot.) Mindkét programcsalád sebessége kielégítő, még nagyobb adatbázisok esetében is.

Aki csak kisebb listák elkészítésére akar egyszerű, jól kezelhető és könnyen megtanulható adatbázis-kezelő programot beszerezni, annak a GoAmiga! ajánlható. Ez a program is képes IFF képeket fogadni. Mivel az adatokat mindig betölti a lemezzel a memóriába, ezért 512 kb-át alapkonfiguráció esetén a lemez 880 kb-át kapacitása nem használható ki.

Egyedi megoldása miatt említjük meg még az Aquisition és a Microfiche Filer programot. Az Organize! adatbázis-kezelőről már volt szd a Scribble! szövegszerkesztőnél. Aki a szövegszerkesztő miatt már megvette a Works! lemezt, az mindenképpen próbálja meg használni az Organize!-t.

6.9. Könyvelés

A háztartási könyvelésben vagy egy üzlet forgalmának nyilvántartásában jó szolgálatot tesz az AmigaCalc. Az adatokat táblázatokba gyűjthetjük, és ezekkel a táblázatokkal különféle számításokat is végezhetünk. A programot számos előre beépített számítási mód teszi alkalmassá a legkülönbözőbb könyvelési feladatok ellátására.

A meglehetősen drága Maxiplan Plus csak üzleti célokra ajánlható, a program szolgáltatásai minden igényt kielégítenek.

Aki nem akar sokat költeni, de egy számlavezető programra szeretne szert tenni, annak érdemes beszereznie a Taifun 79 lemezen található GiroMan (PD) programot.

A német nyelvű Amiga-Buch, amely a vállalat teljes áruforgalmát és vevőkörét nyilvántarja, raktárkészletező program is. A program különleges változata már adó-tanácsadói feladatokat is ellát. A programmal max. 32 000 vevő és szállító tartható nyilván.

6.10. MS-DOS

Már szoltunk róla, hogy az Amiga IBM kompatibilissé tehető. Erre több mód is kínálkozik. A legtökéletesebb megoldást az Amiga 2000-en alkalmazhatjuk PC-kártyák betételével. Az Amiga 1000-hez Sidecar (e. szájdkár) fejlesztettek ki, ezt hozzákapcsolva az előzőhöz hasonló jó eredményt kaphatunk. Kis átalakítással a Sidecar az Amiga 500-hoz is csatlakoztatható. Az eredmény jobb a szoftverrel — az IBM emulátorral — elérhetőnél.

Bármelyik módszert is alkalmazzuk, a világ legnagyobb szoftverkínálata, az MS-DOS alatt futó szoftverek teljes választéka rendelkezésünkre áll. Ezek sok esetben jól kiegészíthetik az Amiga főképp grafikaorientált szoftverkínálatát. Irodai használat esetén érdemes ezt a területet tanulmányozni.

Azt is megtehetjük, hogy az Amigán, Amiga üzemmódban írt programot az MS-DOS-ba transzformáljuk, vagy fordítva. Erre használható a PC-Bridge szoftver. A program tulajdonképpen két programból áll. Az egyik a PC oldalon, a másik az Amiga oldalon fut, de a szöveget az egyik részből a másikba egyszerűen átvihetjük. A szoftver (vagy inkább szoftvercsomag) egy további programot is tartalmaz. Ez egy merevlemez másolóprogram. A másoló az adatokat sűrített formában viszi a merevlemezre. A PC-Bridge jól alkalmazható az Amiga 2000-en és a Sidecarral ellátott Amiga 500-on, ill. 1000-en.

6.11. Segédprogramok (utilities)

A Workbench lemezen is van egy ilyen feliratú fiók, amely hasznos segédprogramokat tartalmaz. Ezeket a programokat azért nevezzük segédprogramoknak, mert általában nem az előzőekben bemutatott kategóriákhoz hasonló, önálló feladatot végeznek, hanem valamilyen számítógépes kiegészítő feladatot. Természetesen nem lehet éles határt húzni. A Workbench lemezen is található segédprogramok, de ezek legtöbbször már foglalkoztunk. Az ilyen

programok és azok továbbfejlesztése iránti igényt mutatja a WorkbenchExtras programlemez megjelenése, amely további Workbench-szerű segédprogramok gyűjteménye. A lemezen új fontok, IFF betöltő program és számos új menüpont ((XI), ViewText stb.) található.

Az egyik legnépszerűbb segédprogram a disk monitor. A lemezt nézhetjük meg vele, és alkalmasint bizonyos adatokat megváltoztathatunk, ill. átírhatunk. Ehhez a művelethez azonban — akármilyen korszerű disk monitorunk is van — megfelelő szakismeret szükséges. Aki azonban ezt elsajátította, az előtt semmilyen titkos jelszó vagy védelem nem marad rejtve (akár kalandjátékok is megf ejthetők egy jó disk monitorral). Ha rangsorolni kéne, akkor az élre kerülne az a lehetőség, hogy megfelelő ismeret birtokában sérült program kijavítására is vállalkozhatunk.

Az Amigára készült disk monitorok közül a legjobb a DTM Werbung & EDV DISCoverry programja, amely modulokból épül fel és négy lemez meghajtó (df0, df 1, df2, df3) kezelésére alkalmas. A programindításkor bekért jelszó beadása után a modulokat jelképező menürendszer jelenik meg. A felső sorból mindjárt megtudhatjuk, hogy a rendszerre hány lemez meghajtó kapcsolódik, milyen ezek státusa és ki készítette a programot. A Quit DISCoverry-vel kiléphetünk a programból.

Az egyes modulok: Bitmap Modulé, Directory Modulé, Files Modulé, Sectors Modulé, Tracks Modulé, Trackdisk Errors, AmigaDOS Errors. Közülük talán a Sectors Modulé a legfontosabb, a kiválasztott szektorok betölthetők és átvizsgálhatók vele. A legkülönbözőbb kijelzési módok közül választhatunk: nyolcas, tízes, tizenhatos számrendszer vagy bájt, word (esetleg longword). Azonnal betölthetjük a bútblokkot, de bármelyik más blokkba is írhatunk, és így akár az egész programot megváltoztathatjuk. Ne felejtjük el a checksumot is megváltoztatni.

Igen érdekes és hasznos áttekintést adó modul a Bitmap, amely nagyon hasonlít a Commodore 64 tulajdonosok által jól ismert BAM-hoz. A Bitmapból megtudhatjuk, hogy a lemezen vannak-e szabad blokkok, vagy nincsenek, ill., hogy Data, Header, NoDos, Root, Extension, UserDir vagy Short blokk van-e rajta. A törölt blokkok is felismerhetők, mert azokat *D* helyett *d*-vel jelöli a Bitmap tábla.

A kezdő felhasználó számára hasznos lehet még az AmigaDOS Errors modul, amellyel végigpásztázva az egész lemezt, megtalálhatók és kijelvezhetők a lemezhibák.

A DISCoveryn kívül egyszerűbb és olcsóbb monitorok is vannak. Kimondottan jól használhatókat találunk a PD-k között is. Ilyen pl. a DiskX (Fish 158), a NewZap (Fish 164) vagy a Sektorama (Fish 108).

A segédprogramok egyik kedvelt, ám a bolti szoftverek között nemigen található kategóriáját képviselik a bútblokkot manipuláló, a bútblokkba kisebb programot, ill. programindítást elhelyező programok (pl. a Bootgirl vagy a Bootboy).

Nem szeretnénk a CLI jelentőségét és alkalmazhatóságát lebecsülni, de el kell mondani, hogy az AmigaDOS parancsok némelyik csoportját nem a CLI-ben, hanem inkább a klasszikusnak számító CLI-mate, ill. ahhoz hasonló programokban tanácsos alkalmazni. A CLI-mate 1.2-es változatával a képernyőn egyszerre két lemez tartalomjegyzéke, de akár egy lemez két tartalomjegyzéke is megjeleníthető. Az így megjelenített egységek között számtalan egyszerű funkcióval fájlok másolását (file copy) végezhetjük. A program alkalmas fájlok törlésére, szövegfájlok megjelenítésére, lemezek formázására, fájlok átnevezésére, IFF képek megjelenítésére stb.

A CLI-mate-hez hasonló a PD CLI-Wizard (RPD 67), a Directory Utility (RPD 38) és a Disk Master.

Igen hasznosak azok a segédprogramok, amelyek az egyes funkcióbillentyűk működtetését teszik lehetővé. Ilyen program a Zing Keys vagy a PD Funkey (Fish 106).

6.12. Másolók

A segédprogramok sajátos csoportját alkotják a másolóprogramok. A kalózmásolatok készítését leszámítva, ezek a programok a vásárolt szoftverek védelmét biztosítják. Ugyanis — mint már többször említettük — számítógépes alapszabály, hogy sohase használjuk az eredeti lemezt, mert egy véletlen sérülés pótolhatatlan kárt okozhat.

A másolat készítésekor az okozhat gondot, hogy a szoftvergyártók legtöbbje különféle másolásvédelmi eljárásokat alkalmaz, ezért a Workbench-en található DiskCopy programmal nem sokra megyünk. Az Amiga megjelenése után rövidesen kaphatók voltak a piacon olyan hivatalos, de magánkézből származó másolóprogramok is, amelyek nem törődnek a védelemmel, de nagyon gyorsan egyszerre akár több másolatot is készítenek. (Ha van elég lemezmeghajtó.)

Némelyik másolóprogram különleges védelemmel ellátott programok (főképp játékok) másolására alkalmas modulokkal is rendelkezik, pl. a Hercules Copy.

A forgalomban levő másolók közül az egyik legkorszerűbb a méltán népszerű Xcopy, de más hasonlóan jó és megbízható programok is vannak: a Project D, a Marauder II, a Fást Lightning, a True Copy és a Raw Copy.

Az egy lemezmeghajtóval rendelkezők számára külön kiemelendő a White Lightning, amely 1 Mb-át memória esetén egyszerre lemásolja az egész lemezt. Az A-Copy program pedig az IBM üzemmódot alkalmazóknak hasznos, mert idegen formátumban is képes másolni. A DFC másoló érdekessége, hogy multitasking üzemmódban működtethető, tehát másolás közben a képernyőn bármit elvégezhetünk.

Végül egy csupán másolásra alkalmazható másolóprogramot kell megismernünk, ez a Turbo Backup (PD). A program nem másol védett lemezt, azonban kettős verify (e. verifaj) funkciója van, és nemcsak a másolatot, de a forráslemezt is ellenőrzi. Arra is alkalmas, hogy új lemezeinket ellenőrizzük. Nem kell mást tenni, mint egy ismert lemezt erre átmásolni. A Turbo Backup jelzi a lemez legkisebb rejtett hibáját is, de nem biztos, hogy a másolás leáll (durva hiba esetén a másolás nem folytatható!). Az ilyen lemezre ne vigyünk fontos adatokat, mert könnyen lehet, hogy később pórul járunk. A Turbo Backup egyébként igen gyorsan — kb. 105 másodperc alatt — másol le egy lemezt. Ha egyszerre több másolatot készítünk, akkor természetesen tovább fog tartani.

Végeláthatatlanul lehetne folytatni az Amigára írt szoftverek ismertetését, a legfontosabb kategóriákat azonban már bemutatottuk. Befejezésül megemlítjük még az Amiga zenei képességeit kihasználó programokat, amelyek a következők: Deluxe Sound, Aegis Audiomaster II, Dynamic Studio, Dynamic Drums, Hotlicks, Synthia stb. Érdekességképpen néhány különleges alkalmazási területre szánt szoftverről is szólnunk. A horoszkópkészítők bizonyára lelkesen használják majd a német nyelvű Programm des Lebens programot. Az elektronika kedvelőinek ajánljuk a nyáktervező Newio, ProNet vagy ProBoard programot, az adatátvitel iránt érdeklődőknek pedig az Amiga Call bookware-t (inkább kezdőknek) vagy a profiknak készült A-Talk IH-at.

7. JÁTÉKOK

Ha a személyi és házi számítógépek játékprogramjai között világbajnokságot rendeznének, a zsűri bizonyára az Amigának, Ul. az Amigán futó játékoknak adná az első díjat. Nem csoda, hiszen az Amiga két erőssége — a grafika és a zene — a játékok legfőbb alkotóeleme. A szaklapok véleménye szerint némelyik játék minden vonatkozásban eléri a játékkermi játékok színvonalát. Ez azért is nagy dicséret, mert a játékkermi berendezések kimondottan az adott játéokra létrehozott számítógépek, az Amigával pedig — csak úgy "mellékesen" — játszani is lehet. Szinte biztos, hogy nincs olyan Amiga-tulajdonos, akit ne büvöltek volna el a játékok.

A könyvnek nem a legfontosabb fejezete a Játékok, mégis sokan szívesen olvassák majd az itt közölteket. Nem különféle játékokhoz adunk használati utasítást, bár jó néhány játékkal kapcsolatban lesznek ilyen elemek is. Megpróbáljuk áttekinteni a játékválasztékot, felhívva a figyelmet a legszínvonalasabbakra, segítve a válogatást, hiszen egy-egy játék beszerzése nem olcsó. Ezenkívül a szerző által is kipróbált játékokról közlünk útmutatást és néhány tippet.

Kezdjük egy kis történeti áttekintéssel! Az Amiga 1000 megjelenésekor az Amiga ellen felhozott kritikák egyik sarkalatos pontja a szoftverhiány volt, természetesen ez a játékokra is vonatkozott. Ez persze minden új számítógép esetében így van, és éppen az dönti el, hogy a gép milyen népszerű lesz, hogy a szoftvergyártók megfelelő mennyiségű és minőségű szoftvert írnak-e a gépre.

A házi számítógépek esetében különösen jelentősek a játékok. Sőt a számítógépek sikerét alapvetően befolyásolja a szoftverellátottság. Gondoljunk a jó öreg Commodore 64-re, amit már többször eltemettek, de továbbra is ott van a piacon. Ezt kedvező árán kívül főképp szoftverellátottságának köszönheti.

Az Amiga 1000 megjelenésekor a piac nem volt bővebben a játékprogramoknak, ezért a Commodore cég az Amigával egy szuper játékgepet akart a piacra dobni. Szerencsére időközben megváltozott a stratégia, az eredeti elképzelés jegyeit mégis magán viseli a gép, kiváló zenei és grafikus képességei révén.

A kezdetben megjelent játékok nagy része átírat volt, azaz a már létező Commodore 64-es, Sinclair stb. játékokat ültették át az Amigára. Sajnos a legtöbb esetben igen gyenge minőségben. Pontosabban az átültetés tökéletes volt, mert az így átírt játékok nagy része semmiben sem különbözött a Commodore 64-estől vagy más változattól, egyáltalán nem használta ki az Amiga adottságait. Ez mindössze arra volt jó, hogy az Amigára átallok a régi számítógépükön megszeretett játékaik egy részét ezentúl az Amigán is tudták játszani.

Elég sokáig ez volt a helyzet, ugyanis az átírás jóval olcsóbb, mint az új játék megalkotása (tulajdonképpen a magas színvonalú átírás szinte egyenlő lett volna egy új játék megírásával). Kezdetben ugyanis a szoftvergyártók nem akartak kockáztatni, a várhatóan alacsony példányszám miatt túl nagy befektetést vállalni.

A helyzet igazán akkor változott meg, amikor megjelent az Amiga 500-as. Evvel ugyanis sokak számára elérhető árkategóriába került az Amiga, és a korábban elég drága Amiga 1000 ára is meredeken zuhanni kezdett. A gép egyre erőteljesebb elterjedése felkeltette a szoftvergyártók érdeklődését, s tulajdonképpen ekkor indult meg az Amiga játékok áradata. Ma már a hivatalosan forgalmazott játékok száma meghaladja az ezret, és ez a szám meredeken nő.

A kizárólag játékszoftverek bemutatásával, tesztelésével foglalkozó folyóiratok egyre nagyobb helyet engednek az Amiga játékok bemutatásának. Ma már egy játék gyakran egyszerre több változatban jelenik meg a piacon, azaz a forgalmazd egyszerre dobja piacra az Amiga, a Commodore 64, az Atari stb. változatot. A folyóiratokban szívesen elemzik az Amiga változatot, és képeket is közölnek róla. Ezek már nem szolgai módon másolt átiratok. Az Amiga változat általában sokkal többet tud a többieknél.

Van azonban néhány kivétel, amelyek ugyan a "hőskorban" kerültek piacra, de már felcsillantották az Amigában rejlő lehetőségeket. Ilyen volt a Defender of the Crown, a Gee Bee Air Rally, a Marble Madness, az Emerald Mine, a Guild of the Thieves és az Uninvited.

Itt szólunk egy új — természetesen nem csak a játékszoftverekre jellemző — jelenségről; a programok feltöréséről, III. illetéktelen másolásáról. A programok feltörése szinte iparággá alakult. A feltörők, crackerek (e. krekker) általában kisebb társaságokba, klubokba tömörülve tevékenykednek. Szeretik otthagyni kézjegyüket a feltört programon. Kezdetben a program nyitó képeinek valamelyikébe — ott ahol általában a program címe és a hivatalos szerzők neve szerepelt — beírták, hogy "Cracked: ez és ez". Ma már az Intro használják — az angol Introduction (bevezetés, bemutatás) rövidítése —, amely szinte külön program, sokszor fantasztikus grafikával és zenével. Ha betesszük a lemezt, akkor néhány másodperc múlva elindul az Intro és szinte elkápráztatja a nézőt. Az Introban természetesen közlik az Intro készítőjének vagy készítőinek nevét, esetleg a klub nevét, sokszor még a postafiókcímet is megadják.

A crackerek igen büszkék munkájukra, és ezt sokszor az Introban hosszú futófeliratként megjelenő szövegben ki is fejtik. Ugyanebben a futófeliratban üzeneteket is küldenek egymásnak és a felhasználóknak. Gyakran itt lehet a betöltés előtt a játékon módosítani (a crackerek munkájának köszönhetően), így pl. a játékban örök életünk lehet, vagy kezdetünk akár a tizedik szinten is.

A programfeltörést ugyan sokféle szempontból elítélhetjük, azt azonban feltétlenül el kell ismerni, hogy az Intro jelentős része önmagában is nagyszerű teljesítmény. Gyakran az Intro színvonalasabb, mint a játék eredeti bevezetője. Kétségtelen, hogy az Intro külön műfajjá nőtte ki magát, ez más számítógépek esetében is így van, azonban a gépi adottságok miatt jellegzetesen amigás tevékenység.

Ezek után ismerkedjünk meg az egyes játékokkal. A játékokat megpróbáltuk fajta szerint csoportosítani, és a csoportra jellemző tulajdonságok ismertetésén kívül azok legnevesebb képviselőit is röviden bemutatjuk. Néhány játékhoz tippet és trükköt is közlünk.

7.1. Autóverseny

Klasszikus, ősi játékkategória, a bemutatására nem is kell sok időt szánni. Elsőnek a *Test Drive-ot*, ill. ennek bővített változatát, a Test Drive H-t említjük. Tulajdonképpen nem igazi verseny ez. Szuper sportkocsikat kell tesztpilótaként kipróbálnunk, egyre nehezedő körülmények között. Időnként megjelenik mögöttünk a rendőrség vijjogó járőr-kocsija, és ilyenkor vagy rálépünk a gázra, vagy megadjuk magunkat. Az utóbbi esetben azonban megbüntetnek, és ez jelentős idővesztéséget okoz.

A játék időre megy, a gép az adott táv megtételének idejét értékelve pontoz, figyelembe véve az útközben okozott baleseteket is. Ha túl gyorsan vagy figyelmetlenül megyünk, akkor sok mindennek nekimehetünk; akár a magasabb szinteken egyre sűrűbben szembejövő autóknak is. A teszt ugyanis a közúton folyik.

Jól sikerült a játék bevezetője, az elején megszólaló hang tisztán érthető. Fontos, hogy jó autót válasszunk. Az egyes modellek nem egyformán tudnak száguldani, az egyikben kézi- a másikban pedig automata váltó van. A H-es változatban több az átmenet. Talán a Porsche és a Lamborghini képes a legtöbbre. Akinek nem elég izgalmas az alkalmankénti versenyfutás a rendőrséggel, a Test Drive II-ben versenytársat választhat magának, ilyenkor nemcsak az idő, hanem a másik autó ellen is folyik a küzdelem.

Az *Out Run* sem igazi autóverseny, itt feltehetőleg Kalifornia útjain száguldhatunk sportkocsinkkal. Nagyon jól sikerült a közlekedő autók grafikája, különösen a bogárhátú Volkswagené. Kiemelkedő az *Out Run* zenei aláfestése, a bevezető képek is kitűnőek.

A *Turbo Cup* már valódi autóverseny. Egy 924-es Porsche volánjánál ülve próbára tehetjük képességeinket, a profi versenyek szabályainak megfelelően. Először egyedül kell kvalifikáló edzőfutamot menni. Ennek alapján kapunk starthelyet, ami jelentősen befolyásolhatja későbbi eredményünket. Ha nem jól választjuk meg a sebességet, látványosan megpördülhetünk.

A rally versenyek világát hozza a képernyőre a *Lombard Rac Rally*. Az igen változatosan megírt játékban van éjszakai vezetés, adhatunk interjút stb. Egy különlegesen felkészített Ford Sierrát kell vezetnünk. A versenyt az autó belsejéből látjuk, nagyon jó a műszerfal és a sebességváltó megjelenítése.

Technikailag igen jól sikerült játék a *Crazy Cars II*. Ebben a jelenleg kapható egyik legdrágább és leggyorsabb sportkocsival, a Ferrari F 40-nel kell egy megadott térkép szerint az amerikai autópályákon az egyik helyről a másikra száguldani. Itt is csak az idővel kell versenyeznünk. Néha azonban meg kell küzdenünk az állandőrökkel, akiket onnan ismerhetünk meg, hogy autójukkal le akarnak szorítani az útról.

Ebbe a kategóriába tartozó játék a *Super Hang On*, habár itt nem autókkal, hanem superbike-ekkel (szupermotorokkal) versenyzünk. A játék kitűnő, érdemes kipróbálni.

7.2. Kalandjáték (Adventure)

A kalandjátékokat főképp az angol nyelvet bizonyos szinten már ismerők számára ajánljuk, de szótárral is megpróbálkozhatunk az ilyen típusú játékkal. Napokra képes lebilincselni az embert.

Egy történet játszódik a képernyőn (sokszor krimi vagy sci-fi), amelynek mi vagyunk a főszereplői, és a játékot magunk irányítjuk különféle utasításokkal. Ezeket a játékokat általában kitűnő grafikai megoldások és humoros szövegek jellemzik. Ez a szöveg az utasításunknak megfelelően gyakran a kép változásával egyszerre jelenik meg, válaszként és utasításként. Minél színvonalasabb a játék, annál összetettebb szöveget képes megérteni. Előfordul, hogy a káromkodásra is válaszol, gyakran igen kemény, rendreutasító hangon.

Az Amigának itt is megvannak a sajátosságai, és a szoftvergyártók igyekeznek kihasználni a gép lehetőségeit. Kitűnően alkalmazza a több ablakot és a multitaskingot a *Shadowgate*. Ezt a játékot az angollal most ismerkedőknek is ajánljuk, ugyanis a szövegbeírással nincs gond, mert mindössze nyolc utasítást tudunk adni. Ezeket nem kell begépelni, mert a képernyő jobb felső sarkában külön keretben van a nyolc utasítás, és az egénél kell rákattintani arra, amelyiket végre akarjuk hajtani. A szavak jelentése a következő:

Examine = Vizsgáld meg!
 Open = Nyisd ki!
 Close = Csukd be!
 Speak = Beszélj!
 Operate = Működtesd, használd!
 Go = Menj!
 Hit = Üsd meg!
 Consume = Fogyaszd el!

Az utasításokat a legtöbbször a történet különböző helyszíneinek tárgyaival kapcsolatban kell használni. Ha pl. egy ajtót látunk, akkor először az ajtóra kattintsunk rá, hogy az ikonokhoz hasonlóan inverz színre váltsan, majd az Openre kattintva kinyíljon. Előfordulhat, hogy kulcs kell hozzá, s ha azt a történetben megtaláltuk, akkor a következő rákattintási sorrenddel használni is tudjuk: kulcs, Operate, ajtó. A játékban sok egyszerűsítő trükk van, az ajtót kinyithatjuk az ajtóra való kétszeri rákattintással is.

Bizonyos tárgyakat mozgatni is tudunk az egérrel. Minél több dolgot találunk és gyűjtünk össze, mert a játék folyamán fel tudjuk használni azokat, sőt jó részük nélkülözhetetlen a továbbjutáshoz. A tárgyakat az Inventory feliratú ablakba gyűjtjük.

A történet egy elvarázsolt, föld alatti barlangrendszerben játszódik, ahol az előrehaladásért mindenféle gonosz lényel kell megküzdeni, s közben számtalan rejtelet megoldani. A szótárra csak akkor lesz szükségünk, ha a válaszul kapott utasításokat pontosan meg akarjuk érteni, de ez a többlet megéri a fáradságot.

Ugyanennek a cégnek azonos technikával készült, hasonlóan magas színvonalú játéka a *Deja Vu*. Itt egy bűneset felderítésében veszünk részt, de valamilyen beavatkozás következtében emlékezetkihagyásunk van. Tulajdonképpen arra kell rájönnie a játékosnak, hogy ki is ő valójában. Már a sikeres játék folytatása is kapható Oeja Vu II címen.

Kimondottan profi angolnyelvtudásúaknak ajánlható a *Guild of Thieves*. Ebben a játékban is van néhány szép technikai megoldás, de itt minden utasítást nekünk kell kitalálni, és angolul helyesen megfogalmazni. Az egyik legfontosabb utasítás a haladás, amit a go ige és az égtáj — pl. észak (north) — megadásával definiálunk. A legtöbb kalandjáték megelégszik a megfelelő égtáj kezdőbetűjének megadásával, pl. s (south=dél), vagy w (west=nyugat). Érdeemes kísérletezni, mert majdnem minden játék a fontosabb utasítások kezdőbetűjét is hasonló módon megérti, azaz elfogadja. Úgyis jelzi a gép, ha valamit rosszul írtunk.

A Guild of Thieves egyik különlegessége a menürendszer, ami szokatlan a kalandjátékokban. A másik különlegesség a mintegy 30, finoman kidolgozott grafika, amit angol festőművészek készítettek. A grafikák rolóként a képernyőn felfelé, ül. lefelé húzhatók. Ez a hosszabb szövegek nyugodt tanulmányozásához szükséges.

A játékban egy sziget teljes kifosztása a feladatunk, hogy a tolvajklub tagjai lehessünk. Az indulás megkönnyítése érdekében az első utasításra adunk egy tippet: puli rope; get on jetty.

Az előzőhöz hasonló játék a *Fish*, amelyben halként, egy akvárium lakójaként kezdjük a játékot, hogy kiderítsük, mi is volt az a varázslat, amely ezt a fura helyzetet okozta. Az első lépésre itt is adunk tippet: turn over; enter castle; enter warp.

Más megoldású a *Zak McCracken*. A két lemezt elfoglald, igen változatos, közepesen bonyolult játék mentesít minket a szöveg begépelésétől. Itt is előre megadott szavakból, az egérrel állíthatjuk össze az utasításokat. Számos megoldás forog közkezen, ül. jelent meg folyóiratokban. Ezekkel könnyedén végigjátszható a kaland, de ha ilyenhez hozzájutunk, csak akkor vegyük elő, ha végleg elakadtunk. A játék egyébként a Commodore 64-en hasonló nagy sikert megért Maniac Manson folytatásának tekinthető. A módszer, a főszereplő ugyanaz.

7.3. Fantázia (Fantasy)

A kalandjátékok különleges fajtája a fantáziajáték. Ezt az eredetileg nem számítógépre készült társasjátékot ma sok helyütt klubokba tömörült rajongók tömege űzi. A fantáziajátékban egy elvarázsolt mesévilágban kell nekünk vagy alkalmi csapatunknak megküzdeni. Sokszor igen bonyolult szabályok vannak, amelyeknek szigorúan érvényt szerez a

játékvezető. A számítógép a gépi megjelenítéssel (zene, grafika) a játékvezető szerepét veszi át, növelve a játék színvonalát.

Az egyik legsikerültebb ilyen játék a *Dungeon Master*, amit eredetileg Atarira írtak. Később az Amigára is elkészült. A játék magas színvonalá miatt részletesebben szeretnénk bemutatni.

A játék kezdetekor ki kell választanunk négy társat. Erejük, felszerelésük, ül. lelki tulajdonságaik lehetőleg különbözzenek. Mivel a játékban előrehaladva fontos szerepe van a varázslatnak, ezért célszerű olyan társaságot választani, ahol már kezdetben is elég magas a manapontok száma, ez határozza meg ugyanis a varázslási képességet. Lehetőség szerint erős társat is válasszunk, mert számos küzdelmen keresztül vezet az út előre. A négy társat az első szinten lévő labirintusban kell kiválasztani a 24 lehetséges közül. A kiválasztáskor a falon lévő képeket szemből megközelítjük, majd rákattintunk. Ekkor nézhetjük meg, hogy ki lesz a társunk, és milyen tulajdonságai vannak. Ha nem tetszik, akkor a Cancellal visszaléphetünk. Ha a jelölt a játszámához megfelelő, akkor a Reincarnate-re kattintsunk rá, majd új nevet adva az illetőnek, megvan az emberünk. Ha mind a négy társunk készen áll, indulhat a kaland.

Az első szinten nem történik semmi, gyakorolhatjuk a mozgást, a fordulást stb. A játék tizenhét egyre bonyolultabb labirintusból áll. Az egyik szintről a másikra lépcső visz lefelé. Tulajdonképpen ezt a lépcsőt kell megtalálnunk, ha nincs térképünk. Térképet folyóiratokból vagy kluboktól szerezhetünk.

Az első szintről azonban akkor se siessünk lefelé, ha megtaláltuk a lépcsőt, kicsit menjünk tovább. Egy zsákutcába érve néhány hasznos tárgyat fogunk találni. Ha megvannak, akkor mehetünk lefelé.

A játékot a megfelelő nyilakra való rákattintással vezérelhetjük, de egyszerűbb a vezérlés a négy kurzorbillentyűvel, valamint a Del és a Help billentyűvel. A csapat párosával, szinte kézen fogva megy előre. A pontos helyzetről a játszámában részt vevő társakra jellemző színű négy koszorú tájékoztat a jobb felső sarokban. A koszorúk helyének megváltoztatásával megmászhatjuk a sorrendet. Ez azért fontos, mert ha a labirintusban valaki szemből támad, akkor mindig az elöl állók sérülnek. Ide tehát a legerősebb társakat, a legjobb küzdőket kell helyezni, ha pedig valaki megsérült, és nincs mód a pihenésre, akkor hátra kell vinnük, nehogy meghaljon.

Nagyon kell vigyázni társaink életére, a játék ugyanis addig tart, amíg legalább egy társ él, viszont egy társ elvesztése jelentősen csökkenti a csapat ütőképességét. Néhány helyen van mód az újjáélesztésre az "Altar of Rebirth" segítségével. Ez azonban csak ritkán fordul elő, addig viszont a halott társ tárgyait és tetemét vinni kell.

Az út során sokféle tárgyat találunk, amelyeket fel kell venni, és valamelyik társunknak — a terhelést lehetőleg egyenletesen elosztva — oda kell adnunk. A tárgyat — a kiválasztott képére rákattintva — a társunk adatlapján található kis négyzetek valamelyikébe tehetjük. Miket fogunk találni? Nagyon fontosak a különféle kulcsok, mert sok ajtót csak ezekkel tudunk kinyitni. Találunk különféle élelmiszereket is. Ezek is lényegesek, mert látni fogjuk, hogy a Food (étel) és a Water (víz) melletti jelzés csökkenni fog. Ilyenkor enni és inni kell adni a társunknak, az élelmiszert a szájukhoz illesztve. Jó ízű moraj nyugtázza az ételt s az italt. Ezenkívül fegyvereket is találunk, amelyeket amíg a kézben van hely, oda kell helyezni, mert csak így tudunk velük harcolni. Jó fegyver a balta is, a bunkót pedig jól lehet hajítani. Sorozatfegyverként kiválóan alkalmazhatjuk a nyílveszőket, amelyből egyszerre többet is kézbe tudunk venni a harcosok keze melletti kockákkal. A nyíl kilövésekor automatikusan az újabb kerül a kézbe.

Mint minden jó kalandjátékot, tetszőleges helyen ezt is letölthetjük, s ha valahol a csapat csúfosan kimúlna, nem kell előlről kezdeni. A letöltés bármelyik társunk lapján található lemezszimbólumra való rákattintással indítható.

Ugyancsak itt van egy nagyon fontos eszköz, a z betűvel jelölt "altató". Ezzel tudjuk pihentetni a csapatot, és ez a társak neve melletti szimbólumokból is láthatóan az erő, az egészség és a varázslási képesség növekedését vonja maga után. Vigyázni kell, hogy biztos helyen pihenjünk, nehogy közben megtámadjanak.

A játékban érdekesebbnél érdekesebb szörnyekkel kell megküzdenünk, a múmiáktól a repülő sárkányig vagy a kőszörnyig. Ezenkívül számos rejtélyt, logikai feladványt kell megoldani. Útközben pergamentekercseket is találunk, ha ezeket bármelyik társunk szeméhez illesztjük, akkor az ott lévő igen hasznos információkat elolvashatjuk.

Szóljunk még a varázslatról is! A képernyő jobb oldalán különféle hieroglifákat láthatunk. Ezek a varázsszavak, amelyek kombinációja a legkülönbélebb akciókat váltja ki. Tudunk • mágikus fényt gyűjtani, gyógyított készíteni stb. A varázsige legalább két szóképből áll. Az első ugyanis csak a varázserőt határozza meg. Mindjárt el kell kezdeni varázsolni, eleinte ugyan nem fog sikerülni, a számítógép azt fogja válaszolni, hogy kevés a gyakorlatunk. Minél többször próbálkozunk, annál nagyobb lesz a varázserőnk, és ez a harmadik-negyedik szinttől igen fontos lesz. Varázslat nélkül szinte lehetetlen továbbjutni.

Reméljük sikerült kedvet csinálni ehhez a kiváló, hetekig játszható játékhoz. S hogy ne legyen nagyon nehéz a kezdet, az első néhány szinthez a 7.9. alfejezetben közlünk néhány tippet.

Jó fantáziajáték a filmmel azonos című *Willow* is. A kétlemezes *Galdregon's Domain* is érdekes kalandokat kínál. Nem lehet kihagyni a *Bard's Tale* nevű fantáziajátékot sem, aminek már a negyedik része is kapható.

7.4. Kereskedelem

Az egyéb kategóriába kellene sorolni az ún. kereskedelmi játékokat, azonban lassan már külön kategóriát képviselnek. Eredetileg csak az úrkereskedelem volt ismert. Ilyen a híres *Elité*, amelynek Amiga változata kitűnően sikerült, megtartva a játék eredeti, igen ötletes elemeit.

A kereskedelmi játékok közül az egyik legjobb a *Ports of Call*. A játék egyedül is játszható, de igazi élvezetet kettő, három, esetleg négy játékosal jelent. A játék lényege, hogy milyen jól tudunk vezetni egy hajózási vállalatot. A szimulációs játékkategóriába is besorolhatnánk, mert olyan tökéletesen imitálja a valóságot.

A játék elején minden játékos választ magának egy nevet és egy hazai kikötőt. (New-York és Szingapúr talán a legalkalmasabb.) Ezután ötmillió dollárral kezdődhet a vásárlás. A létrejött hajótársaságnak a fuvarozáshoz hajókra van szüksége.

A játék az Amigától megszokott menürendszerrel működik. A menü egyes pontjai szinte magukat magyarázzák. A hajóvásárláshoz a Ship bróker feliratú lapra kell rákattintanunk, s máris egy hatalmas irodaépület liftjében találjuk magunkat. A grafika és a hanghatás az egész játék alatt kitűnő. Az épület 3., 4. és 5. emeletén tudunk vásárolni. A játék kezdetén célszerű a 4. emeletre menni, mert ott jó állapotú hajókat kapunk elfogadható áron. A 3. emeleten ugyan kedvezményesebb áron vásárolhatunk, a hajók rossz állapota miatt azonban hamar elveszítjük a nyereséget a gyakori javításokkal. Az 5. emeleten levő hajókat pedig nem tudjuk megfizetni, legalábbis kezdetben.

A 2. emeleten adhatjuk el a hajóinkat. Így tudunk spekulálni, ugyanis a hajók ára a játékban rendszeresen, de kiszámíthatatlanul változik. A legjobb időpontban lebonyolított vétel és eladás óriási nyereséget hozhat. Gyakorlatilag többet (és főképp gyorsabban) lehet keresni ezzel, mint a játék céljával, a fuvarozással.

Ha megvettük álmaink hajóját, még nézzünk be az irodába, az Office feliratú lapra való rákattintással. Ide a játék folyamán akkor is érdemes időnként benézni, ha semmilyen akciót nem akarunk végrehajtani, mert ha ezt elfelejtjük, akkor könnyen kirabolják az irodát. Az irodában információt kaphatunk a játékban elfoglalt pillanatnyi helyzetünkről, pénzünkről, hajóinkról stb.

Nagyon fontos adat a kezdetben 0 értéket mutató Status. Ez jó üzleti akciókkal vagy jói végrehajtott hajózási manőverekkel növelhető, és egy bizonyos érték elérésekor bankkölcsön és jelzálog felvételét teszi lehetővé, amiből újabb hajókat vehetünk. Mindezt az Action funkcióval tehetjük.

Általános jó tanács, ha nem feltétlenül szükséges, a kikötőkre ne álljunk be kézi vezérléssel. Természetesen sztrájk esetén ez is megtörténhet. A kikötők nagy részéből elvállalhatjuk viszont a kiállást, pénzt takarítunk meg, és néhány ügyes kiállás növeli a Statust. A játék elején nem érdemes csempészfüvart vállalni, mert könnyen rajtaveszthetünk, és behozhatatlan hátrányba kerülünk a többiekhez képest. Figyeljük rendszeresen a hajó műszaki állapotát, és 60% alatt ne induljunk el, mert ha egy hosszú út közben a műszaki állapot 50% alá csökken, akkor a hajónk elsüllyed. Hosszadalmas javítgatás helyett inkább adjuk el a használt hajót, és vegyünk újat. Jobban megéri.

Még egy jó tanács: az 5. szinten levő legnagyobb hajót csak spekulációs célból vegyük meg (ha van rá pénzünk). Ezzel a nagy hajóval elég nehéz manőverezni, s mivel gyorsabban is megy a többiekénél, ez ugyancsak a manőverezéskor jelenthet nehézséget.

Ebbe a kategóriába sorolható még a remek börzejáték, a *Wallstreet Wizard* is.

7.5. Harci játékok

Az Amiga akciójátékok különleges, népszerű ágát képviselik a kommandós vagy általánosan harcinak nevezett játékok.

Az elsők egyike volt a színvonalasan elkészített *Capone*, amelyben egy amerikai kisváros utcáin kell végigküzdeni magunkat, miközben mindenfelől (még az ablakokból is) lőnek ránk. Természetesen mi is tudunk lőni. Egy nagyszerű célzókereszt segít a pontos "munkában". (A játékhoz fénypisztolyt is árulnak a tökéletes hatás kedvéért.) Különösen jól sikerült elem a csörömpölvé betörő ablak.

A Capone sikerén felbuzdulva, a gyártó cég elkészítette a kommandós változatot, a *Prisoner of War* (röviden P.O.W.) játékot. Itt hasonló taktikával — egy rövid előzetes kiképzés után — a dzsungelben kell különféle csapatokkal megküzdenünk. Az eseményeket szenzációs hanghatások festik alá.

A túszkiszabadításos filmekből jól ismert kommandósokkal találkozhatunk a *Hostages* játékban. Már az induló képek és hangok jelzik, hogy gondosan megírt játékkal van dolgunk. Jó botkormány-technika kell hozzá.

A kétlemezes *Operádon Wolf* dzsungelkommandót a P.O.W.-hez lehetne hasonlítani, de csak türelmes profi játékosoknak ajánljuk, mert nagyon nehéz.

7.6. Táblás játékok

A táblás játék valamelyik nem számítógépes játék számítógépes változata. A legkézenfekvőbb példa a sakkjáték. Míg az eredeti sakk partner nélkül általában nem játszható, a számítógépes változatban a számítógép személyében sokszor kemény ellenfélre találunk.

A piacon kapható sakkváltozatok közül csak egyet említünk, amely némileg többet nyújt, mint a sakk szolgái lemásolása. A *Battle Chess* harcosokként jeleníti meg a figurákat, és az ellenfél figurájának leütésekor igazi lovagi csata dül. Nagyszerű hanghatások kísérik a játékot, és a legtöbb amatőr számára valószínűleg elég nehéz.

Nagyon érdekes az ősi kínai játék, a *Kikugi* megvalósítása. Számptalan változatban (azaz táblán) tehetjük próbára logikánkat. A játék lényege, hogy a malomhoz hasonló táblán levő figurákat el kell tüntetni. Ez úgy lehetséges, ha az egyik figurát egy vele szomszédos figurán úgy tudjuk átemelni, hogy az átemelt mellett levő üres helyre kerüljön. Természetesen az üres helyre szükség van és a szomszédos figurára is. A játék elején az induláshoz mindössze egy üres hely van.

Ebbe a kategóriába kell sorolni a kártya- és más kaszinójátékok számítógépes változatát is. A kártyajátékok nagy része azonban póker, s annak is egy sajátos változata, a vetkőzős póker. A játékban egy csinos hölgy az ellenfelünk, aki kezdőtőkéje elvesztése után egyes ruhadarabjait ajánlja fel a további játékhoz. Az egyik legsikerültebb változat a kétlemezes *Hollywood Poker Pro*, amelyben a viszonylag jó minőségű digitalizált képeken kívül a négy hölgyet nagytón keresztül is megtekinthetjük. Ez a típusú játék ízlés kérdése.

7.7. Sport

Jelentős hányadot képviselnek a különféle sportokat megjelenítő játékok. Két fő csoportot lehetne megkülönböztetni aszerint, hogy egy sportág jelenik-e meg, vagy a sportágak olimpiaszerű gyűjteményét látjuk. De akár külön kategóriákat is lehetne a sporton belül felállítani, gondoljunk csak a futballjátékok tömegtelen mennyiségére.

A futballmegjelenítések közül két kiemelkedőről szólunk. A *Kick Off* élvezetes, sok beállítást tesz lehetővé. A futballpályát felülről látjuk. A másik játék nem a mi fogalmaink szerint értendő, ugyanis az amerikai futballról, a rögbi közeli rokonáról van szó. A játék neve *TV Sports Football*. A kétlemezes program megjelenítésekor úgy érezzük magunkat, mintha egy igazi, amerikai tv-közvetítést néznénk, sőt ha valamilyen oknál fogva nem folytatjuk (vagy nem tudjuk folytatni) a játékot, akkor a számítógép automatikusan átveszi a vezénylest. A jd szabályismeret mindenesetre elengedhetetlen a játék hosszabb élvezetéhez.

A futballhoz hasonló játékot elevenít meg a *Speedball*. A jövőben játszódó meccsen páncélba öltözött fickók küzdenek egy acélgolyó kapuba juttatásáért.

A több sportot is magában foglaló játékok egyike a *Summer Olympiad*. A nyári olimpiai játékok elevenednek meg, annak minden látványos kellékével.

Hasonló, de nem az olimpiai sportokat összefoglaló játék a *California Games*.

A téli sportok kedvelőinek ajánljuk a sportjátékokban leghíresebb cég, az Epyx *The Games-Winter Edition* című játékát. A sí, a korcsolya és a bob különféle formáit úzhatjuk.

Nagyszerű golf szimuláció a *Leader Board Golf*.

Az ökölvívás kedvelőinek pedig a *Ringside*-ot ajánljuk.

7.8. Akciójátékok

A legtöbb játék ebbe a kategóriába sorolható. Az alapséma szerint eszeveszetten kell lőni valamire (vagy valakire), miközben esetleg különleges módon menekülünk (vagy egy objektumot üldözünk), de szörnyekkel is birokra kelhetünk.

Igazi mestermunka a *Hybris*. A Földön túlról érkezett űrhajó szállít minket egy fantasztikus védelmi hálózattal kiépített birodalomba. Hogy küldetésünk teljesítsük, három szintet kell végigküzdenünk. Ez nem tűnik soknak, de egy-egy szint több alszintből áll.

Célszerű mindjárt az elején auto-fire (gyorstüzéző) üzemmódba kapcsolni, mert az ujjunk hamar görcsöt kap. Próbáljuk meg sztereó fejhallgatóval hallgatni az aláfestő zenét és zajokat, mert különleges élményt jelent. Csodálatos a grafika és kítűnő a függőleges scroll.

A játék érdekessége, hogy három életünk elvesztésekor nem a lelegejére kerülünk vissza, feltéve ha sikerült már jókora távolságot megtennünk. A szintek ugyanis alszintekre vannak osztva, és ha legalább két alszintet jutunk, előre a három életünkkel, akkor a következő menetben csak egy szinttel kerülünk lejjebb. Így folyamatosan mehetünk előre. Persze ha nem jól haladunk, akkor akármilyen jó pozíciónk is van, szép lassan visszacsúszhatunk.

Minden szinten a ránk lövő objektumok mellett találkozunk két tűzköpő szörnyel, ezeket nem lehet azonnal kilőni. Úgy kell manővereznünk, hogy a mi lövésünk elérje őket, de ők ne találhassanak el minket. Ezért néha a jó tüzelőpozíciót is fel kell adni, és inkább oldalirányba kell menekülni. Minél magasabb szinten járunk, annál nehezebb lesz ez a feladat.

Időnként a hátunk mögött egy anyahajó jelenik meg, amit ha sikerül megérintenünk, akkor szárnyunk nő, és ez meghosszabbítja életünket. A pörgő kerék megérintésével bizonyos időre fantasztikusan megnövelhetjük a tűzgyorsaságot. Ha végképp vész helyzetbe kerültünk, akkor a Space billentyű érintésével a képernyőn lévő összes ellenséget elpusztíthatjuk, egy nagy villanás kíséretében. Egy élet folyamán azonban csak háromszor alkalmazhatjuk.

A játékot akkor tudjuk megoldani, ha megtaláljuk az ideális arányt az ellenség minél tökéletesebb elpusztítása és az előrehaladás között.

A *Sword of Sodan* nem túl magas színvonalú, négylemezes játék. A gyári Intro különleges, egy kis előleget kapunk a fantasztikus grafikából. Ezután egy elvárásolt várat kell bevennünk, különböző emberekkel és szörnyekkel harcolva. A beszéd szintetizátortól kapjuk az utasításokat. A harci taktikára hamar rá lehet jönni, a játék nem igazán nehéz. Tulajdonképpen nem a feladat motiválja a játékost, hanem a látvány.

Hasonló játék a hat lemezes *Dragon's Lair* is, amelyről jellemzőképpen szinte ugyanazt lehetne elmondani, mint a *Sword of Sodan*-ról. Nagyobb játékgyűjteményből azonban ez sem hiányozhat.

Az akciójátékok egyik nagy családját képezik a *Hybris*-hez hasonló, de vízszintesen előrehaladó lövöldözős játékok. A kétlemezes *BloodMoney* nagyszerűen sikerült.

Nagyon sok játékról lehetne még írni. Bizonyára mindenki tudna még említeni tucatnyi színvonalas játékot, amelynek itt lenne a helye az ismertetésben. Reméljük, mindenkinek elég ennyi információ az eligazodáshoz.

7.9. Játéktípek

Rockét Ranger: A térképteremben ügyködve minél előbb küldjünk ügynököt Dél-Amerikába, mert ott legtöbbször üzemanyagraktár van. Ha üzemanyagot találunk, akkor azonnal szervezzünk ellenállást.

Shadowgate: Ahol az átjáróban a gyertyák vannak, ott van egy könyv is, amit el kell olvasni. A koporsók közül a bal oldalon a leghátsó vezet tovább, a bal középsőt pedig nem szabad kinyitni. A folyóban lévő csontváz kezéből úgy vehetjük ki a kulcsot, ha befagyasztjuk a tó vizét. A fagyasztáshoz a jégkristályt a megfelelő gem megfelelő nyílásba való behelyezésével kapjuk. A nyílás az egyik ajtó mellett látható. Ha a kulcs megvan, ne felejtünk el visszamenni az első terembe!

Dungeon Master: A szörnyek megölésének igen hatásos módja, ha a legközelebbi — nem kézi vezérléssel működő — ajtóhoz csaljuk őket, s amikor éppen az ajtó alatt vannak, az ajtót működésbe hozzuk. A következő négy harcossal érhetjük el a legjobb eredményt: Wuuf, Darouu, Boris, Gothmog. A terheket egyenletesen osszuk el közöttük. Egy idő után nem érdemes mindent felvenni, kivéve az ételt, mert az később gyorsan fog fogyni. A sikoltozó virág megölésekor négy darabra esik szét, és kitűnő eledel lesz belőle.

A 2. szinten látszólag nem nyílik az ajtó, de a mellékfolyosón néhány bunkó- vagy fejszecsapással mégis ki tudjuk nyitni.

A 3. szintnek mindjárt az elején egy rejtett gombbal kinyithatunk egy oldalfolyosót, ahol némi ügyességgel egy iránytűt találunk, amely később még fontos lesz. A ládában talált Mirror of Dawn vegyük a kezünkbe, álljunk vele a szem elé, és egy titkos ajtó fog kinyílni. A szint végén bőven tankoljunk vizet!

A 4. szinten a szellemszörnyeket nem lehet lebunkózni, csak a des ew varázsszó segít. Vizsgáljuk meg tüzetesen a zöld folyadékot tartalmazó nyílásokat, mert az egyikben egy kulcs is lesz.

Az 5. szinten a transzportmezőn az iránytűvel a következő irányokba menve jutunk át: észak, kelet, nyugat, dél, észak, kelet.

A 6. szinten a Riddle teremben akkor tudjuk az ajtót kinyitni, ha a következő négy tárgyat elhelyezzük valamelyik nyílásban: íj, Mirror of Dawn, Blue Gem, arany. Egy kulcshoz is hozzájutunk. Aki idáig eljutott, annak már nem nagyon lesz szüksége további tippekre, és talán rossz néven is venné, ha a többi rejtélyt előre elárulnánk.

Ütravalóul foglaljuk össze a legfontosabb varázsszavakat:

ful	fénybe borít;
vi bro	antiméreg (legyen a kézben lombik);
ful ir	tűzgolyó (nagyon jó fegyver a szörnyek ellen);
ya ir	mágikus pajzs;
ya bro ros	lábnyomokat hagyunk hátra (könnyebb visszatálni).

Varázsláskor ne felejtjük, hogy az első szimbólum nem varázsszó, csak a varázserőt határozza meg, balról jobbra. Kezdetben csak a legkisebb varázserőt próbálgassuk. Jó varázslást!

8. HARDVERKIEGÉSZÍTŐK

Kezdjük ezt a fejezetet rendhagyó módon! Közelítsük meg a témát egy sajátos, de mindjárt felhasználható gyakorlati oldalról, hogy a kezdőknek és a haladóknak egyaránt hasznos legyen (feltéve, hogy még nem vásárolták meg az összes hardverkiegészítőt).

Következzen tehát egy vásárlási útmutató! Sokan bizonyára azért vették meg az Amigát, mert sok jó tulajdonságán kívül az alapgéphez adott kiegészítések is igen vonzóknak tűntek, összevetve az árral. Hiszen nemcsak az egér, hanem a lemezmeghajtó is be van építve a gépbe. Hiába volt olcsó a Commodore 64, ugyanannyiért (vagy drágábban) meg kellett vásárolni a lemezmeghajtót vagy legalább egy kazettás magnót.

Az Amigával sokkal egyszerűbb a kezdet. Szinte mindent megkapunk a géphez. Aztán kiderül, hogy mennyi minden kellene még, és rájövünk, hogy nemcsak az Amiga zseniális konstrukció, hanem az az üzletpolitika is, amivel a piacra kerül. Az alapkonzfiguráció — akármilyen sokrétű is az Amiga — bizony sok esetben csak körülményesen vagy sehogy sem tudja majd kielégíteni az igényeinket. Ezért vásárolni kényszerülünk, és mire észbe kapunk, az alapgép áránál jóval többet költöttünk a kiegészítőkre. Ezért most, mielőtt az egyes kiegészítőkkal részletesebben is megismerkednénk, a saját tapasztalatokat felhasználva, hadd adjunk segítséget az egyes kiegészítők beszerzésének sorrendjéhez.

Az alapkiegészítők megvásárlásának sorrendje azok nélkülözhetőségétől függ. Természetesen a felállított sorrend, ül. szempontrendszer nem általános érvényű. Lehet, hogy valaki más szempontokat tart fontosnak, ezért más sorrendet fog magának felállítani. Az alapkiegészítők megítélésében is eltérhetnek a vélemények.

Legyen az első beruházás az 512 kb-ajos memóriabővítő, beépített órával! Általában így árulják, az órát tehát nem kell külön kérni, néha azonban (minimális árkülönbséggel) óra nélkül is kínálják. Ezt viszont ne vegyük meg, ragaszkodjunk az órás változathoz. Azért javasoljuk első lépésnek a memóriabővítőt, mert manapság jó néhány program (ül. gyári szoftver) csak 1 Mb-ajt memóriakapacitás esetén működik, vagy a kifogástalan működéshez az 1 Mb-ajt ajánlott. Emellett megfelelő memóriahellyel (RAM lemez) a második helyen ajánlott kiegészítőt — a külső lemezmeghajtót — részben pótolni tudjuk. A belső óra nem nélkülözhetetlen, de hasznos kiegészítő. S van még egy azonnal nem érzékelhető, de a profi számára nem elhanyagolandó szempont. Bizonyos programok futása meggyorsul a nagyobb memória eredményeképpen.

Második lépésként vegyük meg a külső lemezmeghajtót! Azon szerencséseknek, akik az Amiga 2000-rel kezdenek, természetesen nem kell külső meghajtót venni, hiszen beépíthető a második meghajtó is. Akik eddig figyelmesen végigolvasták a könyv előző fejezeteit, azoknak nem kell bizonygatni a második meghajtó jelentőségét, hiszen a CLI műveletektől kezdve számtalan esetben teszi kényelmessé a munkát, megkímélve a felhasználót sok kényszerű trükk és lemezcsere alkalmazásától. Ezenkívül van még egy nem elhanyagolható szempont, amely minden kételkedőt meggyőzhet a beruházás hasznáról. A munkák jellegétől függően ugyanis a belső meghajtót 10–40%-ban mentesíteni tudjuk. Ez pedig meghosszabbíthatja a belső meghajtó élettartamát.

Arra is van mód, hogy a főmeghajtó — a DFO — ne a belső legyen, s így a 10–40% arány megfordulhasson. Ehhez bootselectort (e. bútszelektort) kell vennünk, amellyel tetszőlegesen kijelölhetjük, hogy melyik meghajtó legyen a DFO, és melyik a DF1. Szoftverrel is megoldható a feladat, de ekkor a gép bekapcsolásakor az első bútolást a DFO-ról kell végrehajtani.

Harmadik kiegészítőként vegyük meg a monitort. Aki használta már monitorral az Amigát, annak biztosan nem kell különösebb indoklás. Annyit azonban hadd említsünk meg, hogy a legjobb tv-készülékkel is megerőltető az ilyen munka, aki tehát sokat programoz vagy gyakran szerkeszt szöveget, szerezzen be egy monitort, hogy óvja a szemét. Még a monokróm monitor is jobb a programozáshoz vagy a szövegszerkesztéshez, a többi programot pedig nézzük a tévén színesben. Előbb-utóbb azonban a színes monitor lehet a jd megoldás.

Ez a három kiegészítő alapvetően fontos, hogy az Amigával megfelelően dolgozhassunk. Aki teheti, az mindjárt vegye meg ezeket is. Számos üzlet ugyanis ezekkel a kiegészítőkkal együtt kihálja az Amigát, mégpedig jelentős árkedvezményrel.

Ezután vegyük sorra az egyes hardverkiegészítőket.

8. 1. Memóriabővítők

A memóriabővítő az alapgép memóriakapacitását növeli. Erre több szempontból is nagy szükség lehet. A bővítő egy integrált áramkört tartalmaz, amit valamilyen módon a géphez kell csatlakoztatni.

A legegyszerűbb bővítő a Commodore A501-es kártyája. Ezt az Amiga 500-as alján lévő lap levétele után szabaddá válni nyílásba helyezve kell csatlakoztatni. A kapacitást 512 kb-áttal növeli meg.

A Combitec DRAM bővítővel az Amiga 500-as RAM-ja 2, 4 vagy 8 Mb-áttal bővíthető. Ezt nem a gép alján kell csatlakoztatni, hanem a gép bal oldalán található Expansion portnál.

Az ismertetett bővítők a dinamikus memóriabővítők családjába tartoznak. A gép kikapcsolása után a bennük tárolt információ elvész. Ezzel ellentétben a statikus memóriabővítőkben a modul a kikapcsolás után akár hónapokig képes a benne taroltakat megőrizni, majd az újból bekapcsoláskor felhasználni. A Combitec S-RAM 500/1M típusú 512 kb-áttos bővítője ilyen.

A Combitec dinamikus bővítői az Amiga 2000-hez is megvásárolhatók. Ugyancsak az Amiga 2000-hez fejlesztette ki a Commodore az A2058 jelzésű, 2, 4 és 8 Mb-átt kapacitású dinamikus bővítőket. Ezek a bővítők nagyon csábítók, de drágák, így ha különösebb célunk nincs, az A501-es az ideális bővítő.

8.2. Külső lemezmeghajtók

Zavarba ejtően nagy a választék. Igazán tanácsot sem lehet adni, mert az egyes készülékek között nincs minőségileg nagy különbség, legalábbis egyetlen kipróbálással nem lehet eldönteni, öt-tíz éves használat tapasztalataival pedig még nem rendelkezik senki.

Ha valamilyen szempont szerint mégis dönteni kellene, akkor az egyik szempont a zajtalanság legyen. A beépített lemezmeghajtó ugyanis elég zajos. Sajnos a Commodore külső meghajtója is meglehetősen zajos, ráadásul elég nagy méretű, holott az összes többi márka szinte kizárólag slimline típusú, azaz lapos, kicsi.

A biztonság kedvéért érdemes a legjobb márkákat venni, ilyen a Combitec, az Alcomp, a Profex, a Message, a Kupke, a NEC. (A márka nélküliek sem kerülnek sokkal kevesebbe.)

Arra is figyeljünk a vásárláskor, hogy az összekötő kábel legalább 60 cm hosszú (vagy hosszabb) legyen, és a meghajtó hátlapján legyen egy kapcsoló, amellyel az összekötő kábel kihúzása nélkül is le tudjuk kapcsolni az Amigáról. Erre a memóriának és a szoftvernek is szüksége lehet.

Felárért olyan meghajtót is kaphatunk, amely digitális kijelzőjével folyamatosan mutatja, hogy a fej melyik sávon van. Az így kapott információ igen hasznos lehet a fejlesztőknek.

Szándékosan nem hangsúlyoztuk, hogy 31/2-es vagy PC formátumú meghajtóról van-e szó, mert az elmondott elvek többé-kevésbé mindkét fajtára érvényesek. Nagy meghajtót azonban csak azután érdemes venni, ha már van egy 31/2-es külső meghajtónk. A nagy meghajtó használatával viszont jelentős lemezköltséget takaríthatunk meg, mert a hozzá való lemez ára kb. a fele a mikroflopplémezőnek.

Érdemes megjegyezni, hogy két lemezmeghajtó összeszereléséből és közös házba való beépítéséből származó ikermeghajtó is kapható. A legérdekesebb a MAX 1, amely tetszés szerinti elosztásban három meghajtót tud befogadni és kijelezni.

8.3. Monitorok

Habár a számítógéppiacon nagy választék van monitorokból, az Amigásoknak csak néhány közül kell választani, mert a drága monitorok nagy része arra való, hogy más — sokkal kisebb grafikai képességekkel rendelkező — számítógépek grafikai képét kiválóvá varázsolják.

Nézzük a választékot! Valószínűleg a legtöbb felhasználó számára a legkézenfekvőbb megoldás, ha vesz egy Amiga monitort. Jelenleg négy típusa van, amelyek közül az első, a Commodore 1081-es már csak elvétve kapható. A legelterjedtebb típus az 1084-es, amelynek Scart, RGB és FBAS bemenete is van. Ennek sztereó változata az 1084 S jelzésű készülék. A felsorolt monitorok közös hibája, hogy Interlace üzemmódban remeg a kép. Ezt kiküszöböli a nagyfelbontású (1008x800 pont) 2024-es típus. Az 1084 S modellel szinte teljesen azonos tulajdonságú a Philips CM 8833-as típusa, amely szintén sztereó. A 1084-es modellel ellentétben ennek tükröz a képernyője, tehát munka közben kerülni kell az erős hátsó megvilágításokat, viszont rendkívül éles képet ad.

Aki ezeknél többre képes monitort szeretne, az bátran választhatja (háromszoros árért) a NEC Multisync II monitort. Az Amiga programokban nem veszünk észre különbséget a képben, aki azonban PC üzemmódban is használja az Amigát, annak kiváló PC képminőséget nyújt ez a monitor.

Az Amiga monitorok kategóriájába tartozik a viszonylag olcsó sztereó Highscreen KP 748-as monitor is.

8.4. Merevlemez meghajtók

A merevlemez meghajtó, amit a köznyelv Winchesternek (e. vincseszter) nevez, olyan különleges beépített (Amiga 2000) vagy külső csatlakoztatású (Amiga 500) lemezmeghajtó, amelyben a lemezt nem lehet cserélni, nem hajlékonylemez (floppy) van benne, hanem egy ún. merevlemez forog légmentesen zárt térben. A megoldás előnye, hogy jóval (átlagosan 20—40-szer) sűrűbben lehet a lemezre írni, tehát a kapacitása rendkívül nagy. A merevlemez mérete lehet 51/4-es vagy 31/2-es.

A különféle márkák választéka gazdag, kezdve a beépíthető Seagate 225 (20), NEC D5126 (20), Seagate 138 (32) modellektől a külső Supra-Drive (20), Message AHD (20), X-tension(40) modellekig. A zárójelben levő szám az egyes modellek kapacitását adja meg Mbájtban. (A floppy lemez kapacitása 880 kbájt, tehát még az 1 Mbájtot sem éri el.)

Amellett, hogy a merevlemezen nagyon kényelmesen tárolhatunk, a hozzáférési idő is jóval kisebb a hagyományos lemezhez képest. Ez azt jelenti, hogy a gép a lemezen a keresett

programot igen hamar megtalálja, és rövid idő alatt képes be is tölteni. Természetesen a jdl szervezett tartalomjegyzéknek itt jelentősen megnő a szerepe. Külön DOS parancsok is vannak, amelyek támogatják a merevlemez-zel való munkát.

Az Amiga hardverpiacán az 1989-es év egyik slágere a Commodore cég A590 jelzésű Hard Drive Plus készüléke volt, amely az Amiga 500-hoz csatlakoztatható, és a 20 Mbájtos merevlemez-en kívül egy 2 Mbájtos memóriabővítőt is tartalmaz. Ez utóbbit csak megrendelés-re, külön opcióként kaphatjuk meg. Igazán kihasználni azonban csak a Kickstart 1.3-as változatával lehet. Ebben az esetben ugyanis autobútolható a merevlemez, vagyis ha a Workbench-et ide átmásoljuk, akkor a gép bekapcsolásakor a merevlemez-re jellemző sebességgel innen bútolhatunk. Természetesen egy kapcsolóval átkapcsolhatjuk a berendezést az.1.2-es változatra. Az A590-nel rokon berendezés a szinte azonos szolgáltatásokat nyújtó DoubleDisk20/2A.

Kuriózumként érdemes megemlíteni a K&S Subsystem-Megafile nevű merevlemezét, amelynek kapacitása 250 Mbajt.

8.5. Az egér

Az egér nem igazi kiegészítő, hiszen a készülékkel együtt kapjuk, azonban előfordulhat, hogy egyszer elhaszná-lódik, és újat kell vennünk. Ilyenkor választhatunk, hogy veszünk egy hagyományosat, vagy a három gombos DMouse egér mellett döntünk. A harmadik gombot a mellékelt szoftverrel bármilyen funkció-ra beállíthatjuk.

Érdekes a DMouse egér működési elve. Nem a hagyományos, mechanikusan guruló golyó elvén működik. A pontos helymegállapítást az optika végzi. Az egér a talpán fénysugarat bocsát ki, amely a tartozékként hozzá adott asztallapról visszaverődik, és a lapon levő vonalhálóval megállapítható az adott pozíció. Mivel forgó alkatrészt nem tartalmaz, élettartama jelentősen megnő a hagyományos egérhez képest. A hagyományos egér élettartamát is megnövelhetjük, ha ügyelünk rá, hogy lehetőleg tiszta asztalon használjuk. A legjobb, ha veszünk egy *Mousepad*nek (e. mauszped) nevezett tartozékot, amely a DMouse-nál említett asztallaphoz hasonló, kb. 30x40 cm-es érdes felületű lapocska.

8.6. Videó

Az Amiga egyre több profi és amatőr videózónak lett elválaszthatatlan társa. Köszönhető ez a gép kiváló grafikus képességének, valamint az egyre jobb és több, a videózást és a grafikát támogató szoftvernek. Nézzük, milyen különleges hardvereszközökre lesz szükség a programok hatékony kihasználásához. Természetesen nem a profi stúdiók milliókat érő felszerelését szeretnénk bemutatni, hanem azokat az (esetleg amatőr számára is elérhető) eszközöket, amelyek feltétlenül szükségesek.

Ahhoz, hogy a létrehozott grafikát vagy videofelíratot a már kész videofilmre másolhassuk, *genlockra* (e. dzsenlok) lesz szükségünk. A műszaki megoldás részletezése nélkül csupán annyit jegyzünk meg, hogy a genlock teremti meg a szinkront a videokészülék képei, valamint a számítógép által szolgáltatott kép között. A berendezés nélkül a videofelíratot csak a videofilm elé vagy mögé tudnánk felvenni, rákeverni semmiképp sem lehetne. A piacon gazdag választékban kaphatók, hiszen nem csak az Amigához vásárolják a videózok.

Jól sikerült konstrukció a Magni 4005 jelzésű berendezés, képalkotása és jelerőssége a legmagasabb követelményeket is kielégíti. A Commodore is forgalmaz genlockot, ennek típusjelzése A2301.

A másik eszköz a *képdigitalizáló*, amellyel egy videokamera által továbbított képet tudunk bevinni az Amigába, majd megfelelő átalakítás után azt a lemezre is le tudjuk tölteni. Ily módon megfelelő adatbázis-kezelő programmal \s persze az érintettek hozzájárulásával) ismerőseink címjegyzékéhez akár arcképeket is csatolhatjuk, és a lemeztől betöltve a képernyőn megjeleníthetjük. Kezdetben a digitalizálók csak fekete-fehér képet tudtak létrehozni, a mai korszerűbb berendezéseknek a színes kép előállítására sem gond. A Snapshot Professional és a Superpic két korszerű, j5 minőségű képdigitalizáló.

Nem a videózás eszköze a *hangdigitalizáló*. Ugyanazon az elven működik, mint a képdigitalizáló, de vele szöveget, zenét vagy zajokat tudunk a számítógép által tárolható formába alakítani és lemezre letölteni.

A következő eszköz sem kapcsolódik szorosan a videózáshoz, azonban az általa végrehajtott feladat a képdigitalizálók feladatának rokona. A *scannert* (e. szkennert) a szövegszerkesztéssel rendszeresen foglalkozók értékelik igazán. Segítségével ugyanis — a billentyűzet érintése nélkül — több száz oldalnyi már leírt szöveget bevihetünk a számítógépbe és azt a lemezre is letölthetjük. Ha a scannert a szöveg felett mozgatjuk, akkor az emberi szemhez hasonlóan beolvassa azt. Természetesen nemcsak szöveget, de grafikát, sőt akár fényképet is be lehet vele másolni. Igen kedvező áron megvehetjük a Reisware cég Handy Scanner márkanevű, különböző képességű készülékeit.

8.7. Processzorkártya

Minden elfogultság nélkül mondhatjuk, hogy az Amiga ma a legkorszerűbb személyi számítógépek egyike, ezt kitűnő paraméterei is bizonyítják. Alig elképzelhető, hogy ezt még fokozni lehet, a Commodore cég mégis töretlenül fejleszt tovább. Ennek a munkának az eredménye az Amiga 2000-be helyezhető A2620-as processzorkártya. Az A2620-as kártya legfontosabb alkotórésze a Motorola 68020-as 14,3 MHz-es órajelű mikroprocesszora. Ehhez tartozik az ugyancsak 14,3 MHz-es órajelű MC 68881-es matematikai koprocesszor. A kvarc kicserélésével ez utóbbi órajele 20 vagy 25 MHz-es is lehet. A RAM 2 Mbájtos, de 4 Mbájtra bővíthető. A processzorkártya alkalmazásával — a programtól függően — az Amiga sebessége a többszörösére növekszik.

De nincs vége a képességek fejlesztésének, mert megjelent az amerikai Great Valley Products Impact A2000-030 jelzésű processzorkártyája. A kártya legfontosabb alkotóeleme a Motorola 68030-as mikroprocesszora, amely 16, Hl. 25 MHz-es órajelű 32 bites processzor. Matematikai koprocesszorként a 25 MHz-es órajelű MC 68882-es processzort használja. Az alapkonfiguráció 8 Mbájtra bővíthető 4 Mbájtos RAM. Az alapgép sebessége akár tízszeresére is növekedhet.

8.8. PC-kártyák

Ugyancsak az Amiga 2000 tulajdonosok figyelmére tarthatnak számot a különböző PC-kártyák. Az Amiga többféleképpen is IBM kompatibilissé tehető. Ennek egyik módja, hogy az Amiga 2000-ben kialakított részbe PC kártyát helyezünk.

A PC-ismerők jól tudják, hogy az IBM PC-k alapvetően két csoportba sorolhatók. Az egyik az XT, a másik a nagyobb teljesítményű AT. Ennek megfelelően kétféle kártya közül választhatunk. A 2088-as az XT-, a 2286-os pedig az AT-kártya. Behelyezésükkel az Amigán

is kihasználhatjuk az MS-DOS alatt futó óriási mennyiségű szoftvert, és az Amiga kompatibilissé válik a világ legtöbb irodájában használt IBM kompatibilis gépekkel, ill. a rajtuk futó programokkal.

De nézzük az egyes kártyák főbb adatait! Az XT-kártya legfontosabb alkotórésze egy Intel 8088-as mikroprocesszor, 4,77 MHz-es órajellel. A RAM 512 kb-átos, amely 640-re bővíthető. Az AT-kártya alapja az Intel 80286-as mikroprocesszor. Az órajel frekvenciája 8 MHz. A RAM 1 Mb-át kapacitású. Ily módon közvetlen adatforgalom hozható létre az MS-DOS és az Amiga-DOS között.

9. TIPPEK ÉS TRÜKKÖK

A figyelmes olvasó bizonyára már számos tippet és trükköt fedezett fel az előző fejezetekben. Azokat célszerű volt ott közölni, mert a munkához feltétlenül szükségesek voltak. Ebben a fejezetben további trükköket szeretnénk ismertetni, amelyekkel a munka könnyebbé vagy érdekesebbé tehető. Ezek a trükkök általában egyszerűek, aki sokat ül az Amiga előtt, maga is felfedezhet hasonlókat. Természetesen akadnak bonyolultabbak is, amelyek nem kapcsolódnak szorosan a napi munkához. A tippek és trükkök ismertetésekor nem a logikai sorrendet követtük, mert sok trükk egyszerre több témához is kapcsolódik.

9.1. Workbench lemez

Az első tipp a Workbench-hez kapcsolódik. Nézzük a következő esetet! A Workbench lemezt gyakran használjuk, és sok lemezt, ill. az ezeken levő programokat csak a Workbench betöltése után indítunk. A Workbench lemez — és így a róla készült biztonsági másolat is — tele van. Több programot tehát nem vihetünk fel rá.

Ha pl. a Notepaddal néhány szöveget akarunk írni, akkor a letöltéskor a *Diskfull* hibaüzenetet kapjuk. Ha azonban részletesebben megvizsgáljuk a Workbench tartalmát, akkor kiderül, hogy számos fájl nem, vagy csak nagyon ritkán használunk. Ezekre tehát nincs igazán szükség.

Természetesen a Workbench készítői nem felesleges programokkal töltötték tele a lemezt, ellenkezőleg, a lehető legnagyobb variációs lehetőséget akarták nyújtani a különféle beállításokhoz. Azonban kézenfekvő, hogy csak a számunkra hasznos programokat tartsuk a lemezen. Nézzük meg azt, hogy mit törölhetünk le róla! A letörlendő fájlok nyilván eltérhetnek az általunk ajánlottaktól, attól függően, hogy kinek mi a fontos. Az elv azonban azonos.

A CLI vagy a Shell megnyitása után a `Dir opt` ai paranccsal, ill. a `del` kiegészítéssel törölhetünk a legegyszerűbben. A `C` tartalomjegyzékből a következőkre van ritkán szükség (tehát ezeket törölhetjük):

- Break;
- Diskchange;
- Edit (vagy Ed, attól függően, hogy melyiket használjuk);
- Fault;
- Info;
- Prompt;
- Protect;
- Relabel;
- Setdate;
- Sort;
- Status;
- Version.

A fonts tartalomjegyzékből — hacsak valamelyik különleges fontot nem akarjuk használni — a topaz font kivételével az összes többit kitörölhetjük. A Utilitiesből kitörölhetjük a Clockot és a Calculator-t. A libs tartalomjegyzékben viszont hagyjuk meg a következőket:

- diskfont.library;
- icon.library;
- info.library.

Nagyon fontos megjegyezni, hogy az icon.library a Loadwb parancshoz feltétlenül szükséges. Ha tehát nem az itt említett mddon (a Workbench-ről törölve), hanem fordított módszerrel, az üres lemez Workbench programokkal való feltöltésével kísérletezünk, akkor az icon.library-ről feledkezzünk meg.

A devs tartalomjegyzékből sok fájlt törölhetünk:

- mountlist;
- narrátor device;
- parallel device;
- printer device (ha nincs nyomtatónk);
- serial device (ha nincs nyomtatónk).

A devs tartalomjegyzékben további tartalomjegyzékek is vannak. Ha nincs nyomtatónk, akkor a printers tartalomjegyzéket teljes tartalmával együtt törölhetjük. Ha van, akkor is csak a nyomtatónkra vonatkozó egyetlen fájlt (pl. az Epsont) kell megtartanunk. A keymaps tartalomjegyzékből a nemzeti karaktértípust előállító /ájlok közül csak azt tartjuk meg, amelyiket ténylegesen használunk, a többit törölhetjük. (Általában a *d-t* kell megtartanunk.)

Ha nincs merevlemezünk, akkor az s tartalomjegyzékből töröljük ki a Startup-Sequence.HD fájlt.

Végül a System tartalomjegyzékből kitörölhetjük a következőket:

- InitPrinter;
- InitPrinter.info;
- MergeMem;
- MergeMem.info.

Ha a felsorolt törléseket — igényeinket is figyelembe véve — elvégezzük, tapasztalni fogjuk, hogy a Workbench lemez továbbra is betölti majd funkcióját, de rengeteg hely felszabadul, amelyet esetleg más programoknak tarthatunk fenn. Több olyan PD program van, amely — rövidegsége és hatékonysága miatt — szintén a Workbench-re kívánczik. Ilyen a képernyőt egy bizonyos idő után kikapcsoló PopCLI vagy a gyors szöveg megjelenítést segítő Blitzfont stb.

9.2. Lapozás Basicben

A Basic programozásban gyakran előfordul, hogy hosszú programot írunk, majd a program elejétől a közepére vagy a végére akarunk lapozni. Bizony néha türelemre van szükség, még a viszonylag gyors képernyőkezelést nyújtó Workbench 1.3-as változatban is.

Ha tudjuk (pontosan vagy megközelítőleg), hogy hányat akarunk lapozni, akkor először vízszintes irányban lapozunk jobbra úgy, hogy az egyes Basic soroknak csak a végét lássuk, és

ezután végezzük el a lapozást. Meglepően gyorsan fog menni, mert a képernyőn viszonylag kevés a szöveg, és ez jelentősen növeli a sebességet.

9.3. Törlésmentés

A Basic programozásban előfordulhat, hogy egy Basic sort átírunk vagy törölünk, s amikor rájövünk, hogy semmi értelme sem volt, akkor szeretnénk visszaállítani az eredeti állapotot. Ha azonban nem emlékszünk már pontosan, akkor az adott Basic sort nem tudnánk újból begépelni. Ha viszont még nem léptünk ki abból a sorból, amelyben a hibás lépést elköveztük, akkor a jobb oldali Amiga billentyű és az *L* lenyomásával visszaállíthatjuk az eredeti sort.

9.4. A Workbench gyors elérése

Ha a Workbench szintről szeretnénk dolgozni, de nem akarunk a Workbench lemez betöltésére akár másfél percig is várni, akkor tegyük a következőt.

A Workbench lemez behelyezése után, amint megjelenik az AmigaDos ablak, nyomjuk le a Ctrl és a D billentyűt. A betöltés ekkor megszakad, és a BREAK szó jelenik meg a képernyőn. Ezután egymás után hajtsuk végre a Loadwb és az Endcli parancsot! Máris a Workbench szinten vagyunk.

Természetesen a Startup-sequence-be épített rutinok ilyenkor nem futnak le, tehát bizonyos dolgok hiányoznak majd. Például az eredeti angol billentyűzetkiosztás jelenik meg, mert a Setmap d parancs sem tudott lefutni.

9.5. Ha a CLI nem akar elindulni

Előfordulhat, hogy az új Amiga-tulajdonosoknak (vagy az új, ül. idegen Workbench-et használóknak) nem sikerül aktiválni a CLI-t. A bűnös a Preferences, ül. az ott elhelyezett "központi kapcsoló".

A Preferences szokásos aktiválása után a bal oldalon elhelyezkedő CLI kapcsolót *on* állásba kell állítani.

9.6. DOS parancsok kézközeiben

Számos szoftverhez nem kell előzetesen betölteni a Workbench lemezt, a bútoláshoz szükséges fájlokat ugyanis a programozók már felvitték a lemezre. Előfordulhat azonban, hogy — a programból kilépve, vagy a programmal a CLI-> aktiválva — olyan DOS parancsra lenne szükségünk, amelyik véletlenül sincs az adott lemezen. Ilyenkor hiába tesszük be a Workbench lemezt, mert a DOS parancsokat a gép mindig a rendszerlemez (system) c tartalomjegyzékében keresi. Ebben az esetben a rendszerlemez az adott szoftverlemez.

Ha viszont abból indulunk ki, hogy a DOS parancsok ugyanolyan programok, mint bármely más szoftver, akkor a CLI-ből ugyanúgy el kell tudni indítani azokat. (A pontos utat — path — azonban meg kell adni.) Ha tehát a Workbench lemezt (vagy azt a tetszőleges lemezt, amelyen a szükséges CLI parancsokat tartjuk) a belső lemezmeghajtóba tesszük, s mondjuk a Dir parancsra van szükségünk, akkor a következőt kell begépelnünk:

```
dfO:c/Dir
```

Ha az a feladat, hogy a külső meghajtóban levő szövegfájl jelenítsük meg a képernyőn — ráadásul ez a fájl a "téma" nevű tartalomjegyzékben van —, akkor az előző logikát folytatva, a következőt írjuk:

```
dfO:c/Type dfl:téma/szöveg
```

9.7. Néhány magyar ékezetes betű egyszerűen

A magyar programozók bánatára sajnos az Amigába nem építették be a magyar karakterkészletet. Ezért különböző módszereket, szoftvereket vagy trükköket kell alkalmazni, hogy részben vagy egészben helyesen írassunk magyarul is. Az itt következő tipp talán a legegyszerűbb, bár minden ékezetet nem lehet vele megoldani, csupán az é, É, á, Á, ó, O, ú, Ú, í, I betűket (vagyis az egyvesszős, hosszú ékezeteket).

A módszernek az a lényege, hogy az Amiga a német betűkiosztás (Setmap d) esetén tartalmaz egy olyan billentyűt, amelynek megnyomásakor nem történik semmi, még a kurzor sem mozdul. Ez a Backspace billentyűtől balra a másodjé billentyű. Ha e billentyű lenyomása után a megfelelő magánhangzót is lenyomjuk, akkor az ékezzel jelenik meg a képernyőn. Ha a szöveg írásakor előfordul egy francia szó — de nem a francia billentyűzettel (Setmap f) dolgozunk —, akkor a franciák sajátos (a mi ékezetünkkel ellentétes irányba dőlő) ékezetét ezzel a módszerrel elő lehet állítani. Ekkor a magánhangzó előtt az önmagában jelet nem adó billentyűt a Shift billentyűvel együtt kell lenyomni.

9.8. Egy kis elővigyázatosság

Az Amigával végzett munka közben az egyik legkellemetlenebb jelenség, ha valamilyen okból íráshiba (read/write error) lép fel. Ez általában több-kevesebb adat (fájl) megsemmisülését jelenti. A legnagyobb baj pedig az, ha a hiba a 40-es sávban következik be, mert itt van a tartalomjegyzék, és a sérülés következtében a teljes lemez veszendőbe megy, annak ellenére, hogy sem program, sem adat nem sérült meg.

A Diskdoctor vagy a sokkal hatékonyabb Disk Repair program sokat segíthet, de jobb megelőzni a bajokat. A lehetséges okok számbavétele helyett (ezt számos, a floppy lemezek kezelését ismertető könyv vagy cikk már részletesen megtette), most egy olyan okot említünk meg, amelyet valószínűleg csak kevesen vesznek észre. Az Amiga multitasking-képességéről van szó. Ha ugyanis a géphez több lemez meghajtó kapcsolódik, akkor lehetséges, hogy egyszerre hozzuk őket működésbe. Az egyikkel pl. egy programot töltünk be, a másikkal pedig egy ablakot nyitunk ki. Ha jobban odafigyelünk, tapasztalhatjuk, hogy ilyenkor néha elég furcsa hangot hallatnak a meghajtók. Ha ez sűrűn előfordul, akkor az említett végzetes hiba is felléphet.

Ezért tehát ne aktiváljuk egyszerre a meghajtókat, kivéve a másolást, amikor a meghajtók szinkronban működnek. A párhuzamos működtetés egyébként nem jár időnyereséggel, mert ilyenkor a meghajtók jóval lassabban működnek. Érdemes figyelni rá!

9.9. Memóriatarakékoság

Ha sok ablakot nyitunk ki, azok jelentősen fogyasztják a memóriát, főképp ha elég nagyok is. Ha egy ablak kinyitására csak azért van szükség, hogy a benne levő valamelyik ikont aktiváljuk.

akkor az aktiválás előtt az adott ikont vigyük az ablakon kívültre. (De nehogy egy másik ablakba tegyük!) Ezután, még mindig az aktiválás előtt, csukjuk be az ablakot, és indítsuk el a programot. Ez ugyanúgy zajlik le, mintha az ablakból indítottuk volna, viszont az ablak nem foglal el memóriaterületet.

A módszerrel különösen sok memóriahelyet takaríthatunk meg, ha a keresett programhoz esetleg több ablakot is ki kell nyitni, mert a program az egymásba ágyazott fiókok mélyén lapul. Ilyenkor előbb az összes ablakot csukjuk be, és azután indítsuk el a programot! Ha a programot befejeztük, az ikon nem tűnik el a Workbench-ről, akárhányszor újból indíthatjuk. Időt is megtakaríthatunk, ha az egyes ablakokat nem kell nyitogatni.

9.10. A misztikus Guru

Aki nemcsak egyszerű felhasználóként alkalmazza az Amigát, hanem maga is programoz, az előbb-utóbb találkozik az egyik legbosszantóbb jelenséggel, a Guru hibajelzéssel. Ez akkor jelentkezik, ha olyan hiba lépett fel, amelyet a gép már nem tud feldolgozni. Azonkívül, hogy a gép ezt ilyenkor a Guruval bejelenti, további együttműködést nem várhatunk.

Nincs más hátra, újból bútolni kell. Ráadásul a hibajelzésből nem sokat értünk, így nem tudjuk a hibát kijavítani, mert még azt sem tudjuk, hol keressük. Pedig ha megfigyeljük a Guru hibajelzést, akkor minden esetben egy 16 jegyű szám is megjelenik. Sajnos a számítógéphez kapott használati utasítás semmit nem árul el ezeknek a misztikus számoknak a jelentéséből. Pedig a számok bizonyára nem misztikusak, és mindegyiknek megvan a pontos jelentése, amelynek ismeretében a programozó már legtöbbször tudná, hova kell nyúlni. Amíg a teljes dokumentáció nem jelenik meg a Guru számokról — vagy az Amiga nem oldja meg másképp ezt a kérdést —, felsoroljuk néhány hibakód jelentését. (Jó hír, hogy a készülő legújabb 1.4-es Kickstart verzióban a Guru helyett már csak System Message — rendszerüzenet — lesz, és ez nem fog együttjárni a számítógép lemerevedésével, tehát elmaradhat az idegesítő újrabútolás.)

A kód második nyolc számjegye a hiba helyét jelzi, számunkra tehát csupán a hiba mibenlétét jelentő első nyolc számjegy jelentése az érdekes.

00000004	Nem megengedett utasítás
00000005	Osztás nullával
81000003	Hibás, sérült az exec.library
84010007	Nincs elegendő memória az új Screenhez
8401000B	Nincs elegendő memória az új Window-hoz
3 0000001	A gép nem találja a DOS libraryt
0700000A	Lemezhiba (diskerror)

Aki ennél jóval többet szeretne tudni, azt a Gomf 2.0 program felvilágosítja a fellépett Guru hibajelzésről. Amikor a Guru jelentkezne, akkor a program — a Gurunál informatívabb — hibajelzését látjuk, s a program az esetek jelentős részében kiküszöböli a hibát, valamint visszaállítja az eredeti helyzetet, elkerülve így a kényszerű leállást.

A Guru jelentkezése mindig bosszantó, különösen akkor, ha egy hosszabb programot írtunk, amelyet elmentés nélkül próbálunk ki, vagy adatokkal dolgozunk, és esetleg már régen mentettük el az azokat. Ilyenkor rendkívül kellemetlen, hogy a program vagy az adatok kárba vesznek. Egy kis előrelátással és néhány intézkedéssel azonban megmenthetjük a helyzetet.

A Guru a legtöbbször úgy jelentkezik, hogy a képernyő bal felső sarkában egy kérdező felszólít minden tevékenység befejezésére, tehát már előre figyelmeztet. Igaz, ekkor már nem tudjuk kijavítani a programot, a Gurut sem tudjuk leállítani, de az adatainkat megmenthetjük.

Mivel az Amiga multitasking-képessége lehetővé teszi, hogy egymástól függetlenül fussanak a programok, ha van egy független CLI ablakunk, akkor arra rákattintva a kérdező ablaktól független helyzetbe jutunk, amelyből nyugodtan dolgozhatunk tovább. Tehát csak azt kell megszervezni, hogy legyen egy kinyitható CLI ablakunk. A legjobb, ha még a munka kezdetekor nyitunk egy CLI ablakot, és azt a lehető legkisebbre kicsinyítve eltesszük a háttérbe. Ha megjelenik a Guru, akkor csak az előtérbe kell hozni és a szükséges méretre növelni.

9. ILHiányzd jelek

A Basic programlistákat tanulmányozva gyakran találkozhatunk a REM utasítás különös megjelenítésével, a ' jellel, amely a REM-et helyettesíti. A kezdő programozó ehelyett általában a hasonló — a billentyűzet bal felső részén található — ' jelet alkalmazza, és nem érti, hogy a program futtatásakor miért kap hibajelzést. Amikor pedig rájön a tévedésre, akkor azt konstatálja, hogy a' nem található.

A Setmap d billentyűzetkiosztásban a' az Alt és az umlaut a lenyomásakor jelenik meg.

Ezenkívül még számos ilyen rejtett jel van, amelyek a mindennapi munkához általában nem kellenek, de különleges esetben jó hasznukat vehetjük. Következzék most egy csokor ezekből a speciális jelekből! A jel után zárójelben megadjuk a jelet megjelenítő billentyűkombinációt.

ı	(Alt+1)
2	(Alt+Shift+2)
3	(Alt+3)
°	(Alt+4)vagy(Alt+w)
1/4	(Alt+5)
1/2	(Alt+6)
3/4	(Alt+7)
á	(Alt+q)
®	(Alt+r)
£	(Alt+1)
©	(Alt+b)

9.12. Egy Scribble trükk

A 6. fejezetben már szóltunk erről a viszonylag egyszerű, de jól használható szövegszerkesztőről. Nyomtatáskor az alkalmazott papírfajta szerint két beállítás közül választhatunk, az egyik a leporelló (fanfold), a másik a külön lap (single). Ha egy kis munkát szeretnénk magunknak megtakarítani, vagy a nyomtató nincs túl közel a billentyűzethez, akkor a külön lap esetén is érdemes a fanfold beállítást választani, ugyanis így az újabb papírlapok behelyezése és a nyomtató online kapcsolójának aktiválása után azonnal folytatódik a

nyomtatás. Single beállításkor az előzők elvégzésén kívül a Space billentyűt is meg kell nyomni, jelezve, hogy a papírt behelyeztük.

9.13. A dátum beépítése

Már megemlítettük, hogy a belső órával miképpen állíthatjuk be a pontos időt, arról is gondoskodva, hogy a Workbench minden bútoláskor átvegye a pontos időt a gépben lévő órától. Mit tegyünk, ha nincs belső óránk?

Csupán a Startup-sequence-et kell egy sorral megtoldani, így a bútoláskor a gép mindig megkérdezi tőlünk a pontos időt, és a kikapcsolásig pontosan jelzi is. Mindössze a Date ? sort kell beírunk. Ennek hatására bútoláskor a TIME,DATE,TO=VER/K: sor jelenik meg a képernyőn, a bútolás megáll, és várja tőlünk az adatot. A dátumot és a pontos időt angolul kell beírunk. Pl. 22-jan-90 12:33:00. Ezután a Startup-sequence szokásosan folytatódik tovább.

9.14. Ablak aktiválása az egér nélkül

Aki szép Basic programot ír, gyakran alkalmaz egynél több ablakot. Ha az ablakot nemcsak kijelzésre használjuk, hanem valamelyiken keresztül adatokat is beviszünk, akkor előtte az adott ablakot aktiválni kell. (Rá kell kattintanunk az egérnél.)

Ha sűrűn váltogatjuk az ablakot, akkor az egeret is sűrűn kell alkalmazni. Ezt elkerülhetjük az intuition.libraryvel, a Basic programban kiadott ablakaktiváló utasítással:

```
CALL activatewindow(WINDOW(7))
```

Ne felejtjük el a program elején megnyitni az intuáció könyvtárat:

```
LIBRARY "intuition.library"
```

9.15. Egy gyors levél

Ha egy egyszerű levelet vagy üzenetet gyorsan kell megírunk, akkor nem szükséges betölteni egy szövegszerkesztőt, mert a CLI-ből az Ed paranccsal ezt könnyen megoldhatjuk.

Először helyezzük be a Workbench lemezt (írásvédett állapotban) valamelyik meghajtóba! Ezután aktiváljuk a CLI-t, és gépeljük be a következőt:

```
Ed prt:név
```

A *név* akármi lehet, nincs jelentősége, de valamit mindenképpen meg kell adni, különben hibaüzenetet kapunk. Ezután a megjelenő Ed ablakban írjuk meg a szöveget. Ha kész, akkor nyomjuk le az Esc+x billentyűkombinációt. A kérdező jelentkezik, a következő szöveggel:

```
Volume
Workbench 1.3 is write protected
Retry Cancel
```

Ekkor kattintsunk rá a Cancel feliratra! A képernyő alján megjelenik a writing to file prt:név, és elindul a levél kinyomtatása. A nyomtatás befejezése után az Ed ablak automatikusan bezárul.

9.16. A nyomtatás kellékei

Ha saját célra készítünk lemezt — és előbb-utóbb nyomtatási feladatokat kell ellátni —, akkor a következő fájlokat kell a Wbrkbench-ről átmásolni:

- devs/printers/N (az Ni nyomtatónkhoz megfelelő meghajtó);
- devs/printer device;
- devs/parallel device.

A másolás előtt a fájlokhoz létre kell hoznunk a devs, **III**, a devs/printers tartalomjegyzéket.

9.17. Információ a betöltés közben

A Type parancs nyújtotta lehetőségeket már ismertettük a 4. fejezetben. Bemutattuk, hogy a Startup-sequence-be hogyan lehet beolvasni egy különálló fájlt, és hogyan lehet felhasználni menüként. Arra is mód van, hogy ezt a fájlt a Startup-sequence futtatásakor úgy olvassuk be, hogy közben ne álljon le a program.

Ha pl. a betöltés alatt (anélkül, hogy a betöltést ezzel leállítanánk) egy információt vagy utasítást akarunk közölni a felhasználóval, akkor ezt is könnyen megoldhatjuk. Ebben az esetben a Type parancsot multitasking módban kell alkalmazni.

Ha az információt tartalmazó fájlt *szöveg-nck* nevezzük, akkor a Startup-sequence-be beépítendő sor a következő:

```
Run Type szöveg
```

9.18. Átkapcsolás a Basic inzert üzemmódról

Az Amiga-Basicben való programozáskor is egy editort használunk, a Basic editort. Az editor — a szövegszerkesztőkhöz hasonlóan — kétféle üzemmódban működhet. Az egyik az inzert üzemmód, amelyben ha a szöveg közé írunk, akkor az új szöveg nem törli a mögötte álló részt, hanem maga előtt tolja. A Basic editor is így működik. A másik a felülíró (overwrite) üzemmód, amely az előző ellentéte. A legtöbb szövegszerkesztőben a két üzemmód között át lehet kapcsolni, a Basic editorban azonban nem.

Vannak esetek, amikor a felülíró üzemmód jobb lenne. Ez is megoldható, ha a felülírandó betűt, szót vagy szövegrészt az egérrel — mint a Copy vagy a Cut funkcionál — megjelöljük. Ilyenkor az új szöveg a megjelölt részt felülírja.

9.19. PD tipp

Sokak számára kiváló programforrás a Public Domain lemezek beszerzése. Annál is inkább, mert a professzionális szoftverekkel ellentétben ezek nagyon olcsók. Mindenesetre egy kis elővigyázatosság nem árt, mert számos PD program több lemezen is megjelenik, sőt az egyes programoknak több javított változata is van. Ezért mielőtt nagyobb mennyiséget vásárolnánk, célszerű az ún. katalóguslemezeket beszerezni.

A nagyobb PD-árúsító cégek sorozatok (Fish, RPD stb.) szerint rendszeresen katalogizálják az egyes lemezeket. Ma kb. 3 teljes lemezt tesz ki egy nagyobb PD cég teljes katalógusa. A

katalógusban nemcsak az egyes lemezeken található programok címét és verziószámát tüntetik fel, hanem sok programról rövid ismertetőt is közöttek. Néha témák szerinti csoportosítást is találunk a katalóguslemezeken.

9.20. A Rename rejtett képessége

A Rename paranccsal a Copy parancshoz hasonlóan tudunk másolni, igaz, csak a lemezen belül. Másolni természetesen csak az egyik tartalomjegyzékből a másikba lehet. Ha ezt a Rename paranccsal végezzük, akkor a másolandó fájl el is tűnik, tehát a Copy paranccsal végzett műveletekkel ellentétben a kérdéses fájl a másolás után is csak egy példányban lesz a lemezen. Nézzünk egy példát!

```
Rename Utilities/Notepad devs/Notepad
```

A Notepad programot a Utilities tartalomjegyzékből áthelyeztük a devsbe. A Copy és a Rename kapcsolatát talán az Amiga-Basic Copy-Paste és Cut-Paste parancsaival lehetne párhuzamba állítani.

9.21. A nyíl színei

A Workbench-en kívül más programlemezeket is vizsgálva azt tapasztaljuk, hogy az egeret a képernyőn jelző nyíl nemcsak nyíl formájában, és nemcsak a Workbench-en megjelenő színben látható. A Basic programban megjelenő többi színhez szükség szerint mi magunk is hozzáigazíthatjuk a nyíl színét. Pl. akkor, ha azt szeretnénk, hogy a háttérből jobban kiemelkedjen.

A nyíl színe tulajdonképpen három színből tevődik össze, ezt a képernyőn is észrevehetjük. A három színt a Basic PALETTE parancsával tetszés szerint tudjuk állítani, a 17-es, 18-as és 19-es PALETTE színekkel (1. a 3. fejezetet).

10. VÍRUSOK

10.1. Kórkép

Egy igen érdekes, de a legtöbbször aggasztó jelenségről szólnunk ebben a fejezetben. A számítógépvírus az utóbbi időben egyre inkább előtérbe kerülő probléma a programozók és a felhasználók körében. Gyakorlatilag minden számítógéptípust érint, természetesen az Amigát is. A vírus szót nem véletlenül vették át a biológiából, tulajdonságai ugyanis megegyeznek a biológiából ismert vírus két fő tulajdonságával: a fertőzőképességgel és a károkozással. Itt azonban nincs semmilyen biológiai lény, a vírus is program. Ugyanolyan program, mint pl. a szövegszerkesztő. A programra a kis mérete jellemző.

A számítógépvírust az ember hozza létre, rendszerint ártó szándékkal. A vírusfáz első alkalmat (kivéve) egy már fertőzött lemezzel kerülhet hozzánk. Ha ugyanis vírussal fertőzött lemez kerül a számítógépbe, akkor a vírus első dolga, hogy minél előbb bejusson a RAM-ba. Ha pedig másik lemezt (vagy lemezeket) teszünk a gépbe, akkor az első adandó alkalommal a vírus azokat is megfertőzi, azokra is felíródik. Mivel a vírus kikapcsolásig a számítógép memóriájában van, akár több tucat lemez is észrevétlenül megfertőződhet. Ezután a legkülönbözőbb hatásokra számíthatunk.

Pl. egyszer csak elsötétül a képernyő, vagy mindenféle zavaró jelek akadályozzák meg a munka folytatását. Az is előfordul, hogy valamilyen hibajelzést kapunk, és a hiba kijavításáig nem tudunk továbbmenni, mivel azonban valódi hiba nem volt, ezért csak a kikapcsolás segít.

A vírushatások változatossága kimeríthetetlen, hiszen az emberi leleményesség is határtalan. A vírus miatt nem tudjuk a programot folytatni, amely egy játékprogram esetében csupán bosszantó, de egy szövegszerkesztő vagy egy adatbázis-kezelő működtetésében már gondot okoz, hiszen több órai munkánk veszhet kárba, ha az eredményt nem tudjuk letölteni.

A legdurvább vírusprogram mesterségesen hibát hoz létre a lemezen, vagy beletöröl a programokba, és tönkreteszi az egész lemezállományt. Igen nagy veszélyt jelenthet a vírus a merevlemez kiegészítővel rendelkezők számára, mert a vírus azt is megtámadja, és a tönkretétel 20–40 Mbájtos pusztítást jelenthet.

Vannak ártalmatlan vírusok is, amelyek gyakran megtréfálják a felhasználót. Ilyenkor nem károsodik semmi, csupán riasztó felirat jelenik meg, vagy kis'figurák kezdenek összevissza mászkálni a képernyőn. Sajnos a felhasználó sokszor nem tudja eldönteni, hogy ártalmatlan vírussal áll-e szemben.

A másik csoport az illetéktelen másolás elleni védelemre létrehozott vírus. A baj csak ott van, hogy a vírus — terjedési mechanizmusa miatt — nemcsak azokat érinti, akik kalózmásolatokkal foglalkoznak, hanem szinte bárkit. Ugyanis a vírusfertőzéshez az is elegendő, ha a még nem fertőzött lemezt olyan gépbe tesszük be — akárcsak egy pillanatra is —, amelyben előzőleg vírussal fertőzött lemez volt.

10.2. Bútblokkvírus

Mivel a vírus rövid program, könnyen el tud rejtőzni, hogy a megfelelő pillanatban aktiválódjon és lefusson. Az Amigán erre a bútblokk a legalkalmasabb hely. Itt általában nincs helyszűke, és programozás közben a rendszer ide nem ír. Ezért az Amiga vírusainak nagy része ún. bútblokk vírus. Ha viszont valamelyik program használja a bútblokkot, akkor még a legártalmatlanabb vírus is végzetes lehet, hiszen az a bútblokkba beírva magát, felülírja az eredeti programot. (Néhány játék pl. használja a bútblokkot!)

A profi vírusnak a két fő tulajdonságán kívül van egy másik fontos jellemzője is. A reszetálással nem tüntethető el, bent marad a számítógépben. Sajnos már az egyszerűbb vírusok nagy része is ilyen (reszetbiztos).

10.3. Trójai fal

Van egy sokkal rafináltabb vírusfajta is, amelyet trójai falóként emlegetnek. Mindjárt megértjük, miért. Ez a típus nem a bútblokkot használja, hanem be van építve a lemezen lévő (a felhasználó által futtatandó) programba. Az a célja, hogy az illetéktelenül másolt kalózprogramok felhasználását lehetővé tegye. Nagyon nehezen fedezhető fel, hiszen a program szerves részeként töltjük be a gépbe. Innen ered a neve: csak akkor veszünk róla tudomást, amikor már megtörtént a baj. És akkor programozó legyen a talpán, aki megtalálja és megsemmisíti! Hiszen adott esetben akár több száz kb-ait gépi kódú programot kellene tüzetesen átböngészni, megérteni ahhoz, hogy felfedezzük azt a programrészletet, amely a hibát okozta.

A trójai faló működési elve igen egyszerű. Az illetéktelen másolás után a másolaton egy kis részlet nem egyezik meg az eredetivel. A vírusprogram ezt a kis részletet megvizsgálja, és ha tényleg nem egyezik, akkor aktiválódik. Természetesen ez nem ilyen egyszerű, hiszen így nem lenne nagy a kár. A kritikus részlet eltérésekor általában bekapcsol egy késleltető mechanizmus, és nem engedí azonnal aktiválni a programot. Ez a késleltetés egy további feltételhez köti a program működését. Ilyen lehet pl. meghatározott számú letöltés vagy bizonyos idő kivárása, de akár egy dátumot is figyelhet a rendszer (pl. a tizenharmadikát). A késleltetés miatt a felhasználó nem vesz észre semmit, és adott esetben rengeteg munkát elvégez, amit azután a vírusprogram többé-kevésbé tönkretesz.

Ez a vírustípus ugyancsak illetéktelen másolással terjed, mégis megkérdőjelezhető etikai álláspontot tükröz. Jelentős szoftvergyártók ezért ma már nem is alkalmazzák.

A trójai falónak több fajtája is van, az említett nehézségek miatt azonban a megsemmisítésük szinte megoldhatatlan. A következőkben inkább a sokkal "népszerűbb" — és ami lényeges —, sokkal könnyebben leküzdhető bútblokk vírusokról lesz szó.

10.4. Mechanizmus

Tudjuk, ha egy lemezt beteszünk az Amiga lemezmeghajtójába, az automatikusan működésbe lép, és a lemez ismertetőit beolvassa magának. Ily módon egy automatikusan töltődő normál lemezen — legyen az a Workbench vagy más — a gép megkeresi a DOS jelzést, és ha megtalálja, akkor elindítja az alap DOS rutint. Ellenkező esetben megjelenik a felirat: *This is not a DOS disk* (nem DOS lemez).

Ezután a start rutint hajtja végre a gép. Ha ide vírus kerül, az érthető módon minden lemez behelyezésekor kapcsolatba kerülhet a rendszerrel. A vírus beavatkozik a start rutinba, és azt

finoman módosítja. A módosított rutin helyet foglal le magának a memóriában, és ezzel egyidejűleg módosítja a reszet rutint, amely a tulajdonképpeni reszetálást hajtja végre (azaz eltéríti, hogy a vírus rutint hajtsa végre).

A bútbláskor természetesen semmit sem veszünk észre, hiszen a megszokott rutinok is rendben végrehajtnak. Mivel az Amiga jellemzője, hogy a fentieket automatikusan végrehajtja anélkül, hogy a gép bármit is kérne vagy kérdezne tőlünk, ezért ha nem írásvédett lemezt teszünk be a vírussal fertőzött gépbe, az azonnal "megkapja" a vírust.

A bútblakk vírus felkutatásában egy egyszerű disk monitor jó szolgálatot tehet. Nézzük meg a monitorral a Workbench lemez — remélhetőleg még nem fertőzött — első két blokkját, majd ehhez hasonlítsuk a gyanús lemezeket.

10.5.SCA vírus

Az SCA vírus Svájcban származik, létrehozója a Swiss Computer Association klub. Ez volt az első széles körben elterjedt Amiga vírus. Kárt csak a bútblakkot is használó programokban okoz (pl. Terropods, Barbarian, Obliterator, Menace, Katakis, Super Hang On). Persze az is elég bosszúság, hogy a számítógép időnként "lefagy", és csak a reszetálás segít. Sokszori reszetálás (10-15) után a program a képernyőn is bejelentkezik, futófelirat formájában.

A bútblakkot monitorral vizsgálva a *Sométhing wonderful has happened. Your Amiga is alive* (Valami csodálatos történt. Az Amigád életben van.) felirat árulja el a jelenlétét.

A vírusnak ma már számos fajtája vándorol a lemezek bútblakkjai között (pl. az AEK, az LSD, az OBELISK).

10.6. Byte Bandit

Svájci ez a vírus is. Fő jellegzetessége, hogy időnként elsötétül a képernyő, és a program lefagy. A munkát vagy a játékot ilyenkor be kell fejezni. Van azonban egy trükk, amely segít.

Nyomjuk le egyszerre a két Alt, a két Amiga, valamint a Space billentyűt. Máris folytathajuk a programot.

Sajnos nem tökéletes megoldás ez a trükk, mert a viszonylag sűrűn jelentkező sötét képernyő egy idő után kezd idegesítővé válni. A bútblakkot monitorral vizsgálva a *Time Bandit in 9/87. Number ofcopies* (Időbandita a 9/87-ben. Másolatok száma) szöveget találjuk benne.

10.7.ByteWarrior

A Byte Warrior vírus a Byte Bandit egyik korszerűbb fajtája. Ha disk monitorral megvizsgáljuk a fertőzött lemezt, akkor azonkívül, hogy tapasztaljuk a normálistól való eltérést, felfedezhetjük a *DASA0.2* részletet. Ezért DASA vírusnak is szokták hívni.

10.8. Exterminator

Az Exterminator — más néven LAMER — a lemezen lévő sávokat pusztító módon a LAMER!LAMER!LAMER! felirattal írja tele.

10.9. IRQ

Ez ún. link vírus. Megkeresi a Startup-sequence második programját, és megpróbálja megfertőzni. Ha ez valamiért nem sikerül, akkor hozzákapcsolja magát (link) a DIR parancshoz. Ha írásvédett lemezt teszünk be, akkor azonnal megjelenik a kérdező, és felszólít, hogy távolítsuk el a védelmet (*DISK XY is write protected, please remove...*). Ez természetes is, hiszen másképp nem tud fertőzni. Ez tehát nem bútblukk vírus.

10.10. Más vírusfajták

Valószínűleg lehetne még a sort folytatni, mert sok tréfás kedvű programozó van. Végül megemlítnék néhány vírusnevet, amelyekkel a bútblukk vizsgálatoknál találkozhatunk: (Crackright by Disk-Doctors, GADAFFI, TIME BOMB, REVENGE, NORTH STAR.

10.11. Védekezés

A vírusokkal szembeni legjobb védekezés a megelőzés, de ha a fertőzés megtörtént, akkor a bútblukk vírusok ellen — azok megsemmisítésével — elég könnyen tudunk védekezni. Az Amiga-DOS Install parancsával a bútblokkot felülírjuk, és a vírust tulajdonképpen kitöröljük a bútblokkból.

Tegyük a fertőzött lemezt a df 1: meghajtóba, mialatt a Workbench lemez a dfO: meghajtóban van. Ezután írjuk be a következőt:

```
install dfl:
```

Ha csak egy meghajtónk van, akkor kerülőúton kell eljárni. Először tegyük be a Workbench-et: copy dfO: install ram:

Ezután tegyük be a fertőzött lemezt:

```
ram: install dfO:
```

Sokkal jobb módszereket kínálnak az egyre nagyobb számban megjelenő antivírus programok. Az egyszerűbbek csak figyelmeztetnek arra, hogy fertőzött lemezt helyeztünk a gépbe. A bonyolultabbak a megfelelő egérgomb lenyomására (vagy esetleg más utasítás hatására) kitörlik a vírust a fertőzött lemezről. Itt azonban legyünk óvatosak, mert ezek a programok általában csak azt érzékelik, hogy a bútblokkban valami nincs rendben, a vírust azonban nem azonosítják.

A legtöbbször tényleg vírussal van dolgunk, de vannak olyan programok is, amelyek használják a bútblokkot, és egy elhamarkodott mozdulattal könnyen tönkretelhetjük a szoftvert.

Az igazán profi vírusellenőrző program a vírust azonosítani is tudja. A System Z Virus Killer mellett, hogy a vírust azonosítja — legyen a lemez bármelyik meghajtóban — utasításunkra felülírja a bútblokkot egy védelmi programmal. Ennek hatására a lemez védetté válik. A legtöbb bútblukk vírus ezentúl nem tudja felülírni a védett lemez bútblukkját. Ha ezt a lemezt használjuk, majd kivesszük, és a meghajtóba egy fertőzött lemezt teszünk, akkor azonnal figyelmeztető jelzést kapunk. A védő program ugyanis rezidensen a tárban marad. Ha a lemezállomány nagy részét, vagy legalább a sűrűn használt lemezeket ellátjuk ezzel a védelemmel, akkor gyakorlatilag állandó ellenőrzéssel dolgozhatunk.

A profi vírusellenes program végig rezidensen a tárban marad, és jelenlétét minden reszetálás után jelzi. Sőt, ezzel egy időben némelyik egy menüt is felkínál, ezáltal számtalan lehetőség közül választhatunk.

Ilyen program a The Amiga Tool (PD). A program elindítása után kiváltott reszetálás után, ha nem teszünk lemezt a lemez meghajtóba, akkor a fekete képernyő közepén egy vastag kék csík jelenik meg, amelyben a szöveg a tulajdonképpeni menü. Többek között a Showboot opció aktiválásával (mint egy monitorral) nézhetjük meg a bútblokkot. És hogy legyen összehasonlítási alapunk, a Standard funkció aktiválásával a program megmutatja, milyen a normális bútblokk. Ha fertőzést találtunk, akkor az Install funkcióval intézkedhetünk.

A programnak a vírushoz szorosan nem kapcsolódó további hasznos szolgáltatásai is vannak.

10.12. Megelőzés

A betegség elleni védekezés legjobb módja a megelőzés. A számítógépvírus ellen is az a leghatásosabb, ha mindig figyelünk a következő szempontokra:

1. Azokat a lemezeket, amelyekről csak betöltjük az adatokat vagy programokat, mindig írásvédett állapotban használjuk. Így a gépben kószáló vírus ellen is védve vagyunk.
2. Ha nem gyári lemezekről másolunk (vagy ismeretlen forrásból származik a lemez), akkor a másolás előtt — de legkésőbb a másolás után — ellenőrizzük, hogy nem volt-e jelen vírus. Így még időben intézkedhetünk.
3. Ha lehet, a vásárolt vagy általunk írt fontosabb programokról még a legelső felhasználás előtt készítsünk biztonsági másolatot garantáltan vírusmentes másolóval. Érdemes a sűrűn használt adatlemezről is másolatot készíteni, mert ha esetleg károsodnak, akkor az utolsó alkalommal letöltött állapot a rendelkezésünkre áll, és csak a sérüléskor elvégzett munka vész el.
4. A merevlemez tulajdonosa az új programok vizsgálatakor célszerű, ha lekapcsolja a merevlemezt, és csak akkor kapcsolja vissza, ha már meggyőződött a vírusmentességről.
5. Néha az is segít, ha figyeljük, hogyan tölti be az adott programot a gép. Ha valami furcsát, eddig nem tapasztaltat látunk, akkor azonnal reszetáljunk, és vegyük ki a lemezt. Csak alapos vizsgálat után folytassuk a munkát.
6. Ha az idegen lemezekkel végzett hosszabb munka után újból saját lemezeinket vesszük elő, akkor ne reszetálással töröljük ki a memóriát, hanem kapcsoljuk ki a számítógépet. A kikapcsolással ugyanis garantáltan kiirtjuk a vírust, hiszen az is csak egy program.
7. Az előző intézkedést ki lehet egészíteni. Ha a gépben van belső óra, akkor a kikapcsolás után a szokásosnál néhány másodperccel többet várjunk. A belső órát ugyanis a beépített tölthető akkumulátor működteti, és a kikapcsolás után elképzelhető egy rövid ideig tartó kondenzátorhatás, azaz a gép néhány másodpercig még megtarthat a memóriában bizonyos dolgokat. A nagyon gyors visszakapcsolás amúgy sem használ a gépnek.
8. Végül a sűrűn használt lemezekhez nem árt időnként lekérni (a **DIR** vagy a **LIST** paranccsal) a lemez tartalomjegyzékét. Ilyenkor két dolgot figyeljünk meg. Van-e esetleg olyan új fájl a lemezen, amelynek eredetéről nem tudunk? Általában rövid (néhány blokknyi) programokra gyanakodjunk. A másik, hogy nem változott-e meg valamelyik állandó méretű program nagysága. Főképp a c tartalomjegyzékben lévő programokat érdemes megnézni.

TÁRGYMUTATÓ

A

ablak 17
A-Copy 101
adatbázis 46
adatbázis-kezelő program 98
Aegis 95
Alias 84
alprogram 61
altartalomjegyzék 56
Amiga-Basic 92
Amiga billentyű 13
Amiga-Buch 99
AmigaCalc 99
animáció 54,95
Aquisition 99
AREAFILL 42
AREASTEP 42
ASC 39
ASCII 39
Ask80
Assembler 93
átalakító függvény 40
Attributesmenü 27

B

bájt 11, 12
Bard'sTale 107
Basic 29
Basic ablak 30
batch fájl 64
BattleChess 109
Beckertext 97
BEEP 58
belső óra 74
beszúrás 96
betűmodul 85
billentyűzet 14
bit 12
bitszint 55
blokk 20
BloodMoney 110

bob 52,54
bootselector 112
botkormány 10
BREAKOFF51
BREAKON51
BREAKSTOP 51
bűtblakk vírus 128
Butcher 94
bútolás 11
Byte Bandit 129
ByteWarrior 129

C

C93
CAD95
Calif ornia Games 109
CALL61
Capone 108
CD 65
Change Printer 27
Change Serial 26
CHDIR56
Chunk48
ciklusutasítás 37
CIRCLESTEP 42
CleanUp 21
CLEAR 33
CLI 63
CLI lemez 69
CLI-mate 100
CLI-Wizard 100
Close 20
CLOSE 45
COLLISIONOFF52
COLLISIONON 52
COLLISIONSTOP 52
COLOR 33
Compiler93
Continue 32
Copy 31,67
cpi87
CopyPrefs 27
cps 86
CrazyCarsII 104
Cut 31

D

DASA vírus 129

DATA 56
Datamat 98
Date 74
DATES 41
DBWRender 94
DECLAREFUNCTIONLIBRARY 62
DEFTIPUS 38
DejaVu 105
Delete 71
DELETE 57
DeJuxPaint 93
DesktopPublishing 97
DFC 101
DigiPaint 94
DIM 38
Dir 64
DirectoryUtility 100
Discard 20
DISCOVERY 100
Diskmenü 21
DiskMaster 100
disk monitor 100
Diskcopy 67
DiskCopy program 16
Diskdoctor 83
DiskX 100
Documentum 97
DOS 63
DPaint 93
dpi 86
draft 87
Dragon's Lair 110
DSMDisassembler 93
DTP 1. Desktop Publishing
Dungeon Master 106,111
Dynamic-CAD 95
dzsóker 67

E

Echo 76
Ed 70,71
Edit 70
Edit menü 25,30
Edit Pointer 26
editor 70
egér 10,12
Elité 107
elválasztás 96
EmptyTrash 21
END 37

Endcli 63
Endshell 67
ENDSUB61
EOF 46
ERL52
ERROR 52
értékadó utasítás 37
Excellence! 97
EXITSUB 61
Express Paint 94
Exterminator 129
Extrás lemez 16,21

F

fájlmásolás 68
Fást-Filing-System 83
FastFonts 83
Fást Lightning 101
Fault 76
Fed 27
felbontás 86
festékes nyomtató 86
FIELD 47
FILES 57,61
Find 25
fiók 17
Fish 105
FIXWIDTH 28
flag 72
Font Editor 27,90
Fontsmenü 24
fonttípus 87
Font type 28
Formát 67
Formát menü 24
formázás 18
Forms in Flight 94
forráskód 93
FRE 33
funkcióbillentyű 14
Funkey 100

G

Galdregon's Domain 107
genlock 115
gépi kód 93
GET 42,48
GiroMan 99

Global font 25
globális változó 61
GoAmiga 98
GOSUB 36
GOTO 37
gráfikus program 93
Graphics Stúdió 94
GuildofThieves 105
Guru 13,82,122,123

H

hangdigitalizáló 116
hardcopy 88,91
Hercules Copy 101
Hollywood Poker Pro 109
Hostages 108
Hybris 110

I

IFFfájl 48
If szerkezet 80
IFTHEN 39
ikon 11,16,17
index 48
Info 20,74
INKEY40
INPUT 45
Install 78
INSTR 61
InterchangeFile Formát (IFF) 44
Interlace üzemmód 26,35
Intro 103
Intro-CAD 95
intuíció 16
IRQ 130

J

jelszó 58
Jóin 76

K

képdigitalizáló 116
kérdező 13,60
keres-cserél (find and replace) funkció 96

kiadványszerkesztés 97
KickOff 109
Kickstart 14
Kikugi 109
KILL57
könyvtár 61
könyvtárutin 62
közvetlen hozzáférésű fájl 46
külső lemezmeghajtó 112,113

L

Lab 80
LAMER 129
lapadagoló 85
LastError 21
Last Saved 26
LeaderBoardGolf 109
LEFT\$ 40
lemezmeghajtóegység 10
LEN 40
lézernyomtató 86
LIBRARY 62
LIBRARYCLOSE 62
LINE INPUT 45
LINESTEP 37
link vírus 130
List 72,73
LIST 32,57,
List ablak 30
LLIST ablak 90
LOAD 30
Loadwb 76
LOCATE 33
LOF48
logikai művelet 39
lokális változó 61
Lombard Rac Rally 104
LPOS 90
LPRINT 90
LQ 87
LSET 47

M

mail mérge 96
Makedir 69
MarauderII 101
másolás 12,18,31,67
másoldprogram 101

maszk 98
matematikai függvény 39
mátrixnyomtató 86
MaxiplanPlus 99
megszakítás 50
memóriabővítő 112,113
MENÜ 49,50
MENUOFF 50
MENUON 50
MENURESET 50
MENÜSTOP 50
menü 21
menüprogramozás 50
merevlemez meghajtó 114
MÉRGE 57
Microfiche Filer 99
MIDS 40
mikroprocesszor
— 68000-es 14
modulátor 10
Modula—2 93
monitor 113,114
monitorkimenet 10
More 72
mouse 1. egér
MOUSE 51
MOUSE OFF 51
MOUSE ON 51
MOUSE STOP 51
MS-DOS 63
multitasking 15, 25,121

N

New 24
Newcli 64
NewZap 100
NLQ87
NoFastMem 28
Notepad 23
numerikus változó 38
nyomtató 24
nyomtatói mód 87

O

object editor program 56
OBJECTAX 55
OBJECTAY 55
OBJECT.CLIP 55

OBJECT.CLOSE 56
OBJECT.HIT 55
OBJECT.OFF 55
OBJECT.ON 55
OBJECT.PLANES 55
OBJECT.PRIORITY 55
OBJECT.SHAPE 54
OBJECT.START 55
OBJECT.STOP 55
OBJECT.VX 54,55
OBJECT.VY 54,55
OBJECT.X 54,55
OBJECT.Y 54,55
ONBREAKGOSUB 50
ONCOLLISIONGOSUB 52
ONERRORGOTO 51
ON MOUSE GOSUB 51
ONTIMERGOSUB 51
Open 20,24,30
OPEN44
Operádon Wolf 108
órajel 15
Organize! 99
OutRun 104

P

PageSetter 97
Page Stream 97
PAINT STEP 42
Palette 27
PALETTE 33
parancs 84
párhuzamos kimenet 10
párhuzamos üzemmód 15
Pascal 93
Paste 31
Path 69,77
PathReset 69,71
Path Show 69
PATTERN 43
PC-Bridge 99
PC-kártya 99
PEEK 59
PEEKL 59
PEEKW 59
PerfMon 27
Photolab 94
PhotonPaint 94
Pipe 83
Pixmate 94

POINT 33
POKE59
POKEL 59
POKEW 59
pontparancs 96
PortsofCall 107
Preferences 25,75
Print 24,58
PRINT 32
Prismo 94
PrisonerofWar 108
Professional Page 97
programok feltörése 103
Project D 101
Project menü 24,30
prompt 75
PROPORTIONAL 28
Pro Videó Plus 95
provokáció 83
Prowrite 97
Public Domain 92
PublisherPlus 97
puffer 82
PUT48
PUTSTEP42

R

RAM 14
RAMDISK 19
RANDOMIZE 59
RANDOMIZETIMER 59
RawCopy 101
RayTracing 94
READ 56
Redraw 21
rekord 46
REM 33
Removefonts 25
Remove style 25
Rename 20
rendezés 82
Replace 25
reset 13
Resident 84
RESTORE 56
RETURN 36
RIGHTS 40
Ringside 109
RND59
Rockét Ranger 110

roldmenü 19
ROM 14,91
RSET47
Run 77
RUN 32
Run menü 32
runtime modul 98

S

sáv 18
SAVÉ 58
SaveAs 24
SAY52
SCA vírus 129
scanner 116
SCREEN 35
SCREENCLOSE 35
Scribble! 97
scroll 31
SCROLL 43
Sculpt/Animate 3D 94
Sculpt/Animate 4D 94
Search 81
sebesség 86
segédbillentyű 14
segédprogram 100
Sektorama 100
Setclock 75
Setdate 75
Setmap 77

Shadowgate 104,111
Shakespeare 97
SHARED 61
Shell 63
Show List 32
Sidecar 99
Skip 80
SLEEP 43
Snapshot 21
soreditor 67
soros kimenet 10
Sort 82
SOUND 53
SOUNDRESUME 53
SOUND WAIT 53
Specialmenü 21
Speedball 109
sprite 52,54
Stack 82
Start 32

Startup-sequence 70
STATIC 61
Step 32
STEP 37
STICK 59
Stop 32
Style menü 24
SUB61
Summer Olympiad 109
Superbase 98
SuperHangOn 104
Supervisor 96
Suspend 32
SWAP 59
SwordofSodan 110
SYSTEM 58
System/setmap 64
System ZVirusKiller 130
szektor 18
szekvenciális fájl 44
színbeállító 25
szövegszerkesztés 96
szövegszerkesztő 23,24
sztringváltozó 38

T

tápegység 10
tárolás 44
tartalomjegyzék 18
TestDrive 103
TEX 97
Textomat 97
The Amiga Tool 131
The Games-Winter Edition 109
TIMER51
TIMEROFF51
TIMER0N51
TIMERSTOP 51
tizenhatos számrendszer 43
topaz betűtípus 24
TraceOn 32
TRANSLATE 53
trójai faló 128
TrueCopy 101
Turbo Backup 101
Turbo Cup 104
TurboCopy 83
TVSportsFootball 109
TVText 95
Type 73

U

UCASES 46
Undelete 71
Unix 15

V

VAL 40,45
változó 38,61
Version 21,76
vezérlőutasítás 49
videó 95
Videoscape 3D 95
Videotitler 95
Videó WipeMaster 95
vírus 127

W

Wait 81
WallstreetWizard 108
WAVE53
WHILE...WEND 40
WhiteLightning 101
Why 76
WIDTH 90
Willow 107
Winchester 114
windowl. ablak
WINDOW 35,59
Windowmenü 32
WINDOW CLOSE 36
WINDOW OUTPUT 36
WOM 14
Wordwrap 24
Wordperfect 97
Workbench 11,16,21
WorkbenchExtras 100
Workbench képernyő 11
Workbench menü 20
WRITE 33,44
WYSIWYG elv 96

X

X-CAD 95
Xcopy 101

Z

ZakMcCracken 105
zenei programok 101
ZingKeys 100

Kiadja a Műszaki Könyvkiadó
Felelős kiadó: Szűcs Péter igazgató

A szedést a CompAlmanach Kft. készítette
Zrínyi Nyomda, 90.1411/66-14
Felelős vezető: Grasselly István vezérigazgató

Felelős szerkesztő: Györké Tiborné oki. villamosmérnök
Műszaki vezető: Bereczki Gábor
Műszaki szerkesztő: Bagi Miklós
A borítót tervezte: Kováts Tibor
A könyv formátuma: B5
ívetjedelme: 12,5 (A5)
Azonossági szám: 190 511
MŰ: 4432-h-9093
Készült az MSZ 5601-1983 és 5602-1983 szerint
A kézirat lejárva 1990. februári

1
