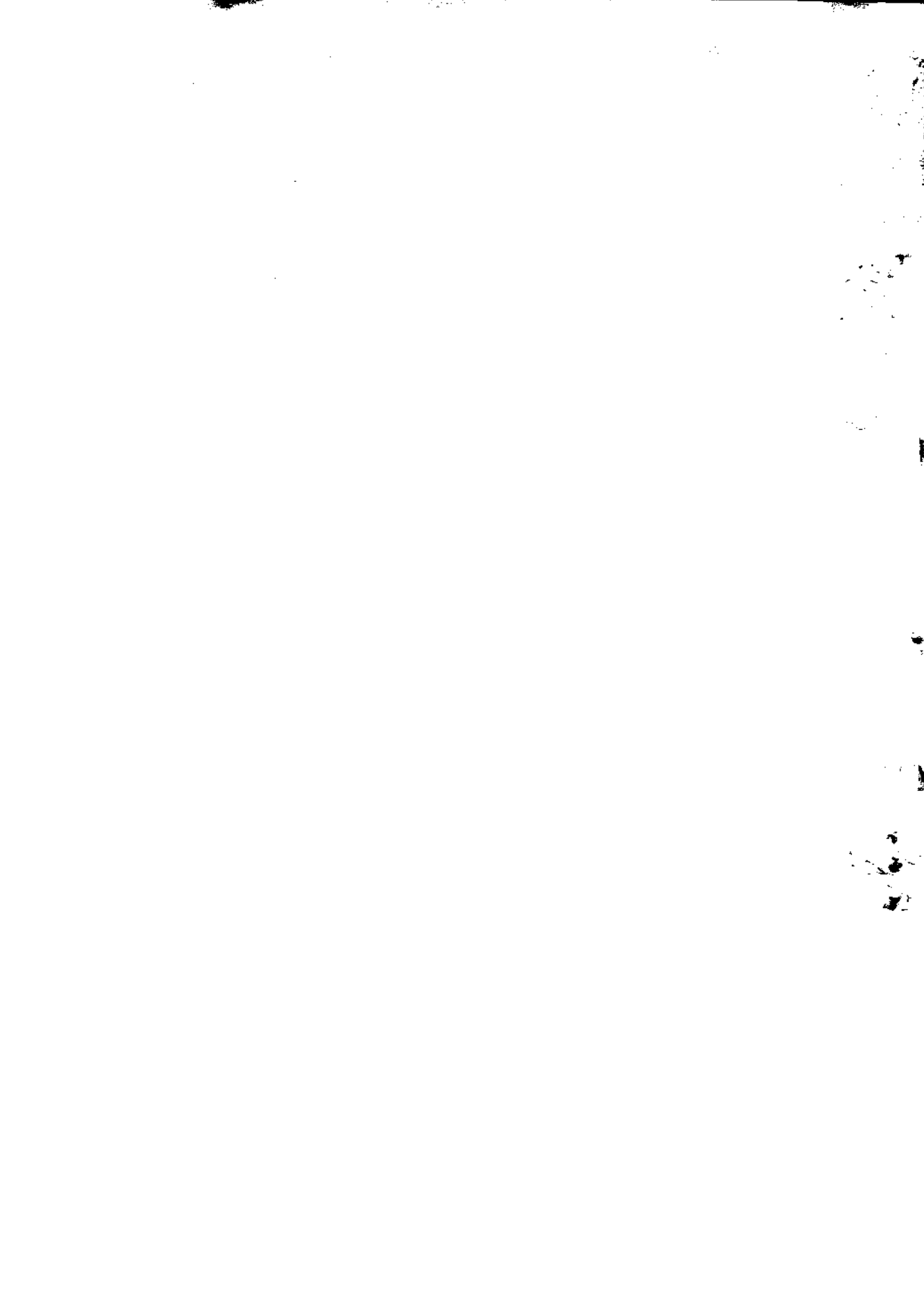


800
T430



Arany Sándor

Amiga felhasználók
kézikönyve

ÁMUVI

Könyvek #3

AURUM Könyvek sorozat #3

Sorozatszerkesztő:
Arany Sándor

Lektorálta:
Bálint József

Amiga felhasználók kézikönyve

(c) Arany Sándor 1993

ISBN 963 04 2977 2

Kiadja az AURUM DTP Stúdió
Tiszaföldvár Ószőlő Fő út 64.
Felelős kiadó: Arany Sándor

Borítóterv: Marinov Gábor

Nyomdai munkálatok:
Tisza Cípőgyár Rt. martfői nyomdája
Felelős vezető: Bélteki Tibor

Köszönetek

Kösznöm a könyv elkészítéséhez nyújtott segítségét a **GURU Szórakoztató Informatikai Magazin** szerkesztőgárdájának, külön is kiemelve **Berényi Zoltán (Bear)** főszerkesztőt, **Turcsán Tamás Péter (Shy)** lapmanagert, akiktől az ötletet és jó adag biztatást, **László József (Joco)** szerkesztőt, akitől rengeteg szakmai segítséget kaptam, valamint **Marinov Gábor (Gaborca)** szerkesztőt, aki a fedéltervet készítette.

- Köszönettel tartozom barátaimnak: **Bálint Józsefnek**, **Rontó Zoltánnak** hasznos tanácsaiért, valamint **Balázs Mártánnak**, a kézirat helyesírásának ellenőrzéséért. Ha maradt még hiba, az nem rajta múltott.

Végül de nem utolsó sorban köszönöm családom tagjainak, hogy megértésükkel türelmükkel és bátorításukkal segítettek munkámat. Külön köszönetet érdemel két éves **Ádám** fiam, aki társszerzőm volt, munkája a könyv néhány oldalán látható, valamint feleségem, **Böbe**, aki a szerkesztésben segített.

Bevezető

Ez a könyv azoknak szól, akik csak a közelmúltban ismerkedtek meg az Amiga számítógéppel, vagy éppen ezután lesznek bemutatva egymásnak. Természetesen azok is forgathatják - remélem haszonnal - akik már régebb óta használják gépüket, de még van mit felfedezniük rajta.

Könyvemet általános felhasználói útmutatónak szánom, amelyben megtalálható az összeszerelési útmutatótól kezdve a DOS leírásán keresztül a Workbench és a géphez mellékelt programok használatáig minden, amit az Amiga számítógépről tudni kell.

A könyv elsősorban az 1.3 Workbench-re, az Amiga 500-ra épül, de külön fejezetben ismertetésre kerül a 2.0-s rendszer, ami az Amiga 500+ és Amiga 3000 gépek sajátja.

Sajnos a könyv terjedelme véges, így nem tudtam kitérni olyan ismeretekre, amelyek a számítógépekkel először találkozóknak számára szükségesek. Nekik ajánlom az irodalomjegyzékben felsorolt könyveket, amelyekből a szükséges alapinformációk megszerezhetők.

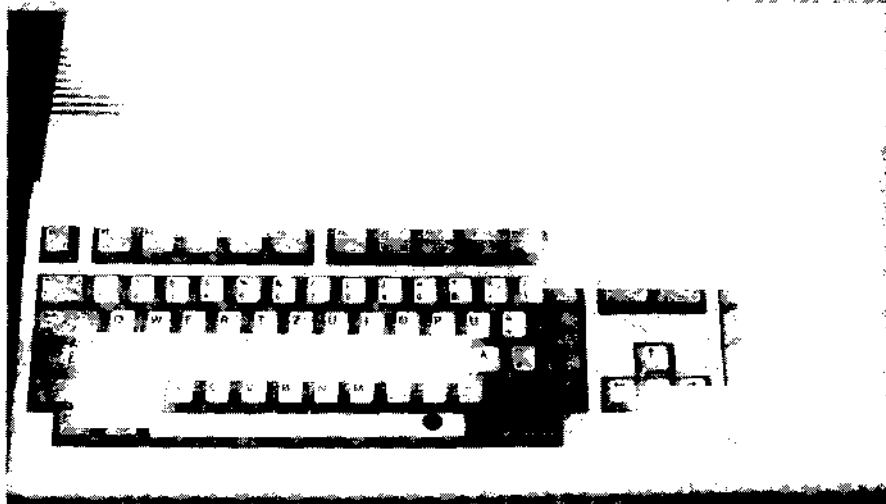
I. Általános leírás

A. A konfiguráció

Nézzük meg, miből áll egy Amiga minimálkonfiguráció:

1. Amiga 500
2. Tápegység
3. RF Modulátor
4. Egér
5. TV

Az Amiga 500 és 500 plusz ún. tasztatúra számítógép, ami azt jelenti, hogy a gép egybe van építve a billentyűzettel, **sőt ebben** az esetben még egy 3,5" floppy meghajtóval is.



Amiga 500

A számítógép lelkét a Motorola MC68000 típusú 16/32 bites processzor képezi, amelynek órajele 7.14 MHz. A 16/32 bit azt jelenti, hogy a processzor belül 32 bites adatokkal dolgozik de

A konfiguráció

kifelé csak 16 bites adatbusszal rendelkezik. Alap memóriája A500-nál fél megabájt, A500+ esetén egy mega. Ez a memória Chip RAM, ami azt jelenti, hogy a gépben rejtőző chipek, amelyek a zene és a kép elkészítésével foglalatkoskodnak, szintén ezt a memóriát használják. Másik memória fajta a Fást RAM (lásd a 44. oldalon), amit csak a processzor tud elérni, ezért gyorsabb. A Fást RAM-ot bővítésként tehetjük a gépbe (lásd a 11. oldalon).

Az Amiga nem kapcsolható közvetlenül a TV készülékre, ehhez szükség van a TV modulátorra"TV modulátor. Az általa nyújtott képminőség azonban jóval elmarad a tökéletesnek mondhatótól, ezért aki teheti, próbáljon meg egy monitor beszerzésével, öreg napjainak jó látásáról gondoskodni.



RF modulátor

Amigához a Commodore 1084 vagy ezzel kompatibilis analóg RGB monitor használható. Van mód a C64 1802 monitorának csatlakoztatására is, de a képminőség elmarad a 1084-estől (a TV-nél azért jobb).

Az egér széria tartozéka minden Amigának. Segítségével általában egy pointert mozgathatunk a képernyőn, amely pointerrel és az egér két gombjának a használatával a legkülönbféle dolgokat művelhetjük.

Általánosságban elmondható, hogy a bal oldali gomb kijelölésre, a jobb oldali pedig a menü kezelésére való, de ez bizonyos programoknál változhat.

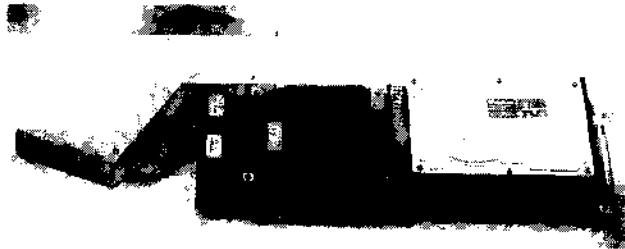
Mint már említettem, az Amiga tartalmaz egy beépített floppy meghajtót, amelynek kapacitása 880 kBájt. Ehhez DS/DD jelzésű floppy diszkeket szükséges használni.

Kívülről további három floppy drive csatlakoztatható, amelyek lehetnek 3.5 vagy 5.25 inch-esek. A látott minimál konfigurációhoz képest a gép igen rugalmasan tovább bővíthető. A leggyakoribb kiegészítés a hard diszk, azaz merev lemez. Ez a gép bal oldalán egy kis műanyag lap takarása alatt lévő csatlakozóra kapcsolható. Meg kell említeni, hogy vannak olyan merevlemezek is, amelyek az Amiga 500 és 500+ házában belülről szerelhetők, de ezek az alkalmazott 2,5"-os harddiszk magas ára miatt nem igazán terjedtek el.



Bal oldali bővítősín

A számítógép maga nem tartalmazza a merevlemez vezérlőt, azt kell a géphez kívülről (vagy belülről) illeszteni. A vezérlő háza általában tartalmaz egy merevlemezt, valamint egyes típusoknál 2-8 MBájt memóriabővítési lehetőséget is.



Külső merevlemez

A konfiguráció

A vezérlő típusa legtöbb esetben SCSI, de lehet IDE is, ami a PC-knél használatos. Az SCSI kontrollert intelligensebb és gyorsabb, ráadásul összesen 7 perifériát tud kezelni, amelyek nem csak Winchesterek, hanem például lézernyomtatók, scannerek, **cserélhető lemezes merev lemezek** is lehetnek.

Az IDE kontrollert előnye, hogy olcsón kapható hozzá a **merev-** lemez.

Az Amigához nyomtató is kapcsolható, akár a soros, akár a párhuzamos porton keresztül.

Egy időben két botkormány csatlakoztatására van lehetőség, de ekkor le kell mondanunk az egérről, mivel az szintén egy botkormány aljzatra van kapcsolva. Kaphatók azonban olyan átkapcsolók is, amelyek lehetővé teszik az egér és a két botkormány párhuzamos használatát, mindössze egy kapcsolót kell átbillenteni a megfelelő periféria kiválasztásához (sőt vannak automatikus átkapcsolók is).

1. A billentyűzet

Miután megismerkedtünk a konfiguráció elemeivel, nézzük meg közelebbről a billentyűzetet is.

Az Amiga billentyűzetén 97 gomb van 7 csoportra bontva.

A legkisebb csoport egy billentyűből áll, ez a bal felső sorokban álló Esc (escape) gomb.

A mellette lévő két ötös csoport a funkcióbillentyűk csoportja. Nevük F1-F5 és F6-F10.

Ezek alatt foglal helyet a legnagyobb csoport, a betűk, számok, írásjelek és kiegészítő billentyűk közös csoportja. Azok a billentyűk, amelyek leütése betűt, számot, írásjegyet nyomtat, hasonlóan az írógéphez, világos színűek. Az ABC betűit jelképező gombokon csak egy jel, egy betű látható. Ezekkel lehet kis betűt írni. Ha vele együtt, előtte lenyomod valamelyik Shift gombot is, vagy Caps Lock le lett ütve és a ledje világít (ez jelzi, hogy a Shift rögzítve van, most már nem kell folyamatosan

nyomni a nagy betű írásához). A Caps Lock újbóli leütésével a rögzítés feloldódik, a led kialszik, kis betűt tudsz írni.

Vannak olyan billentyűk is, amelyekén két jel van. Ezek önállóan az alsó, Shiftelve a felső jelet adják. Az alul-középen lévő hosszú billentyű a Space, amely szóközt ad.

A fehér billentyűkön kívül vannak szürke, vezérlő billentyűk is. Ezek általában nem egy karaktert nyomtatnak, hanem valamilyen műveletet végeznek. Kivétel a csoport bal felső sarkán lévő, amely önmagában aposztrófot, shiftelve hullámjelet ad.

A vezérlő billentyűk közé tartozik a már említett Shift és Caps Lock, valamint a CTRL, Alt és a két Amiga billentyű. Ezeket összefoglalóan emelő billentyűknek nevezzük, mert önmagukban lenyomva semmit nem eredményeznek, de más gombbal együtt használva megváltoztatják annak hatását.

A legjelentősebb az Amiga páros, amelyekkel különböző irányítási feladatok oldhatók meg. Például a bal Amiga + bal Alt az egér bal gombjának a szimulálására, a jobb Amiga + jobb Alt pedig a jobb gombéra való. A CTRL és mindkét Amiga egyidejű használata reszeteli, újra indítja a gépet. Ez a reszet csak ún. "soft reset", meleg indítás (kivéve az egy mega chip ramos gépeket, azoknál ez is hard reset). Ebben az esetben nem zajlik le minden olyan funkció, amely bekapcsolás után, vagy "hard reset" esetén megtörténik. A leglényegesebb különbség, hogy soft reset-nél a memória törlése elmarad, a vírusok túlélnek ezt. A Hard reset-et általában külön programmal válthatjuk ki, de a rendszer összeomlása után is ez zajlik le.

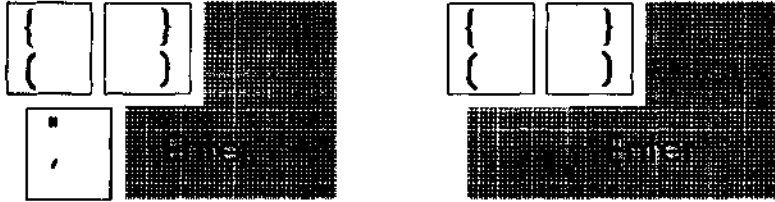
Ezen kívül van a Tab billentyű, amelyet legtöbbször szövegszerkesztőkben használunk, a következő tabulálási pontra vinni a kurzort.

A csoport másik oldalán van még két vezérlő billentyű. A felső a Backspace, amely egy hellyel balra viszi a kurzort és törli az útjába eső szöveget. A kurzortól jobbra eső szöveg is baka lép.

A konfiguráció

Az alatta lévő nagy billentyű a Return. Általában a sor végén használjuk, ezzel jelezve, hogy az a sor kész, a gép feldolgozhatja.

Az angol és német területen készült gépek között van egy kis különbség, ezt szemlélteti az alábbi ábra. Az angol gépek tasztatúráján tehát csak 96 billentyű van.



Német és Angol tasztatúra

Az ötödik billentyű csoport két gombból áll. Az egyik a Del, ami törli a kurzor alatti karaktert, de a kurzor nem mozdul, csak a tőle jobbra lévő szöveg lép balra. A másik a Help, amivel, ha az adott program kezeli, segítséget kaphatunk.

A hatodik csoport a kurzor nyilak csoportja. A négy billentyűvel a rajtuk jelzett irányba léptethető a kurzor.

Az utolsó, a hetedik csoport a numerikus billentyűzet nevet kapta. Bár ezeken olyan karakterek találhatók, amelyek a fő csoportban is megvannak, az általuk nyomtatott karakter is ugyanaz a legtöbb esetben, mégsem teljesen azonosak amazokkal. Sok program külön kezeli őket.

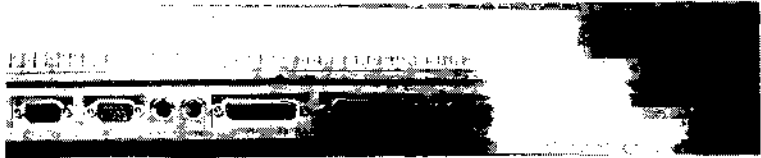
Az Enter feliratú billentyű általában azonos hatású a Returnnal.

2. A számítógép csatlakozói

A gép összeszerelése bizonyos fokú figyelmet igényel, helytelen összeszerelés esetén a gép tönkremehet. Ha betartod az alábbi útmutatót, nem lesz probléma.

A legfontosabb, hogy a bekapcsolt gépen semmiféle csatlakoztatást, dugasz kihúzást ne tegyünk, ugyan is az ekkor keletkező áramlökések egyik-másik érzékeny lelkű IC halálát okozhatják, ami a gép további működése szempontjából nem éppen biztató jel.

összeszerelés előtt ismerkedjünk meg a gépen található csatlakozókkal. Ehhez meg kell fordítani az Amigát, ugyanis az összes csatlakozó (kivéve a már említett bal oldali bővítősínt) hátul található.



Az Amiga hátsó csatlakozói

A csatlakozók bakói-jobbra:

1.joy1

Az egér, botkormány vagy fényceruza csatlakoztatására.

2.joy2

A második botkormány csatlakozója. Ha egy program kezeli, ide is kapcsolható újabb egér. Fényceruzát ez a bemenet nem tud kezelni.

3. Audio L-R

Az Amiga hangkimenete, külön jobb és bal oldal.

4. Extern Floppy

A külső floppy meghajtó csatlakozója.

A harmadik és negyedik drive a második és harmadik meghajtón lévő ugyanilyen aljzatba kapcsolható. Általában a tápfeszültséget is innen kapják a floppyk, de jobb az, ha a harmadik és negyedik drive-nak saját tápegysége van, az Amigáé ugyanis nem túl nagy teljesítményű. Különösen régebbi típusú, 5 1/4 colos meghajtóknál lehet gond, amelyek áramfelvétele igen nagy.

A konfiguráció

5. Serial port

RS232C csatlakozó telefon modem, null modem, midi vagy nyomtató kapcsolható a géphez ezen keresztül.

6. Paralel port

Párhuzamos port. Általában nyomtatót csatlakoztatunk rá.

7. Power

A hálózati tápegység kapcsolódik ide. Ezen keresztül kap az Amiga áramot.

8. Monitor

RGB monitor vagy RF Modulátor csatlakoztatására.

9. Mono Monitor

Monokrom videojel. Ritkán használatos.

Az itt ismertettekén kívül még két csatlakoztatási felület található a gépen. Az egyik a már említett, bal oldali bővítő csatlakozó. A másik szintén egy műanyag lap takarásában van, de a gép alján. Ide legtöbbször memória panelt csatlakoztatunk.

3. Az összeszerelés **művelete**

Miután a gépet kicsomagoltuk, a csatlakoztatandó kábeleket előkészítettük, elkezdhetjük az összeszerelést.

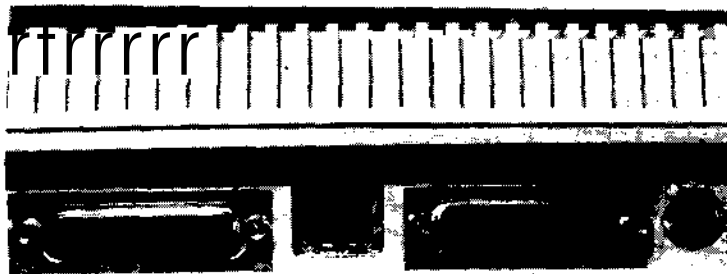
Az egér dugaszát nyomjuk be a JOY1 aljzatba. A csatlakozók csak egyféle képpen illeszthetők össze, a helyes pozícióra tehát nem lesz sok gond. Ha egyidejűleg botkormányt is szeretnénk használni, azt a Joy 2 aljzatba kell illeszteni. Ha két botkormányt szeretnénk, vásárolni kell egy közösítő-átkapcsolót a Joy 1 számára.

A külső floppy meghajtó vezetékét az Extern Drive feliratú aljzatba kell illeszteni. A második külső drive az első házában lévő aljzatba kapcsolódik. Egyszerre max négy lemez meghajtó alkalmazható, beleértve a belsőt is.

A du^sznak szilárdan kell illeszkedni, csak akkor lesz biztos a kontaktus, a nagyobb erőfeszítéseket viszont kerülni kell. Ekkor

inkább vizsgáljuk meg, megfelelő helyzetben van-e a dugasz, vagy egyáltalán jó helyre akarjuk-e csatlakoztatni. Ez általánosságban elmondható minden csatlakozóra.

Az aljzatok mellett két oldalról találunk egy-egy beépített csavaranyát. A csatlakozók végén legtöbbször megvan az ezekhez illeszkedő csavar, amely becsavarásával megóvhatjuk magunkat a szétcsúszó csatlakozók bosszantó hibáitól.



Rögzítő csavarok a csatlakozók mellett

A nyomtató csatlakoztatása attól függően történik, hogy soros vagy párhuzamos-e az illető printer. Ez a nyomtató dokumentációjából derül ki. Ha ezt eldöntöttük, a megfelelő ajzatba illesszük be a kábel végén lévő dugaszt és ha tehetjük, csavarozzuk is be oda.

A megjelenítő eszköz illesztésekor újabb választás elé kerülünk. Legegyszerűbb a dolgunk Commodore 1084 vagy ezzel kompatibilis RGB monitor esetén. A monitor csatlakozóját egyszerűen dugjuk be a számítógép Monitor jelű aljzatába.

A hangvezetékek RCA dugaszait az Audio feliratú két hüvelybe illesszük.

Ha televíziót használunk képmegjelenítőnek, be kell szerezni egy RF modulátort is. Ennek az egyik keskenyebb végén van egy olyan dugasz, ami illeszkedik a "Monitor" jelzésű aljzatba, dugjuk is be.

A konfiguráció

A modulátor rövidebb, Y alakú vezetékének az egyes vége a modulátor oldalán lévő Audio In jelzésű aljzatba kell, hogy kapcsolódjon, a dupla vége pedig a számítógép Audio L-R jelzésű két RCA hüvelyébe.

A hosszabbik vezeték megfelelő végét a modulátor másik rövidebb oldalán lévő RF Out jelzésű aljzatba dugjuk, majd a másik végét a TV antenna csatlakozójára, vagy ha rendelkezünk ilyenekkel, a közösítőbe.

Az RF modulátor használatakor lehetőség van színes videó jel előállítására is, amit felvehetünk. A videó jel kimenete a modulátor hangbemenete mellett van, Videó Out néven. Ekkor - a videó jel természeténél fogva - a hangot külön kell a videó magnóhoz vezetni.

Van még egy lehetőség a kép megjelenítésére, a monokrom videó jel, amely fekete-fehér videó monitoron való megjelenítést tesz lehetővé. Ritkán használatos, főként átalakított Junoszty TV-t használók körében. Meg kell mondani, hogy ez a lehető legrosszabb képminőséget adja. Ha valaki mégis e mellett dönt, használja a képen 9-es számmal jelölt RCA aljzatot. A hangot ebben az esetben is külön kell felhasználni.

Többször is szó esett már a hangról. Az Amiga külön sztereo jobb és bal kimenettel rendelkezik. Ezek igen jó minőségű jelek, érdemes Hi-Fi berendezésen keresztül hallgatni az Amiga hangját. A csatlakoztatáshoz két RCA dugaszban végződő vezeték szükséges.

Akik abban a szerencsés helyzetben vannak, hogy gépükhöz hard diszk is párosul, pattintsák ki a gép bal oldalán lévő műanyag lapot. Emögött található az Amiga bővítő csatlakozója, ami egy megfelelően kiképzett nyákvég. Erre kell rádugni a **hard** diszket. A csatlakoztatást különös figyelemmel végezzük, **nehogy** az áramköri lap megsérüljön.

Mindaddig az összeszerelés sorrendje tetszőleges **lehet** A tápfeszültség csatlakoztatását azonban hagyjuk utoljára. **Kihúzott** hálózati csatlakozó mellett dugjuk be a tápegység **négy**szögletű

dugaszát a "Power" feliratú aljzatba. Az illeszkedő horony miatt a csatlakoztatás csak egyféleképpen lehetséges, de azért figyeljünk oda. A dugacson lévő "Top" felírt ami a csatlakozó felső oldalát jelöli, a segítségünkre van.

Ha mindezekkel megvagyunk, ellenőrizzük, hogy minden kapcsoló, tápegység kikapcsolt helyzetben van-e. Ha igen, dugjuk be a tápegység hálózati dugaszát. Kapcsoljuk be a monitort, printert és egyéb perifériákat, majd legvégül magát "A" számítógépet. Erre a sorrendre mindig ügyeljünk, mert ez meghosszabbítja a gép életét. A kikapcsolás fordított sorrendben történjen.

A gép kikapcsolása után mindig várjunk 20-25 másodpercet a visszakapcsolásig.

Az itt felsorolt szabályok mind azért vannak, hogy megvédjék a gépet a csatlakozásokat követő áramlökések ártó hatásától. Hogy mennyire fontosak, arról Német Ákos barátom tudna pár szót szólni, Ő ugyanis a beteg Amigák doktora.

Az Amiga üzemképességét a Power Led fénye jelzi. Régebbi gépeken ez a fény működő gép esetén is kihunyhat. Ez nem hiba, a led a gép zenei szűrőjével van összhangban. A szűrő kikapcsolása esetén a led is kialszik. Az újabb gépeken ezt már csak a félfényerő mutatja.

Ha TV-t használunk monitorként, az első összeszerelés után rá is kell hangolni azt a modulátor frekvenciájára.

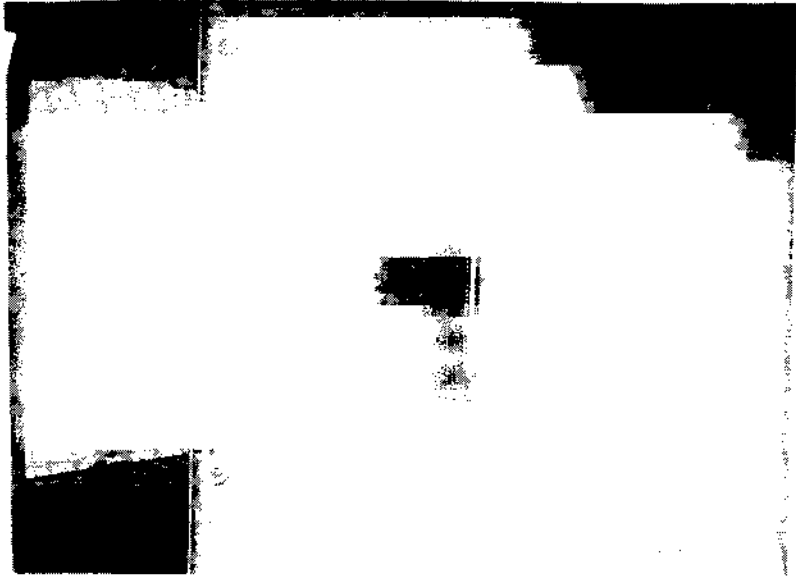
Bár nem tartozik szorosan ide, mégis megemlítem a memóriabővítő beszerelésének a módját.

A memória bővítésére több mód is van. Az egyik az alaplapon történik. Ezt 1.3 gépeknél lehetséges, úgy hogy az alaplapon meglévő helyre négy memória IC-t forrasztunk. Ezt a műveletet inkább bizzuk szakemberre.

A másik, hogy a gép aljában kiképzett csatlakozóra egy bővítő panelt illesztünk. Ehhez a gépet a tápfeszültség vezetékétől megszabadítva fordítsuk hanyatt. A gép házának közepén feltűnik egy kb. tenyérnyi műanyag lap. Ezt pattintsuk ki, ez alatt található a

A konfiguráció

belső memória bővítő csatlakozó sínje. Erre úgy kell feldugni a bővítőt, hogy az alkatrész oldala a gép belseje felé nézzen.

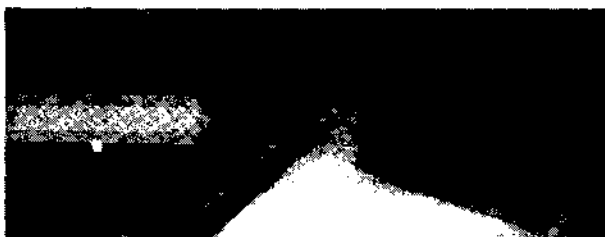


Beszerelt memóriabővítő

Amennyiben a bővítő kapcsolóját is szeretnénk használni, amivel a bővítő ki-be kapcsolható, sajnos meg kell fűrni a készülékházat. Ha erre vetemedünk, mindenképpen olyan helyen kell megtenni ahol:

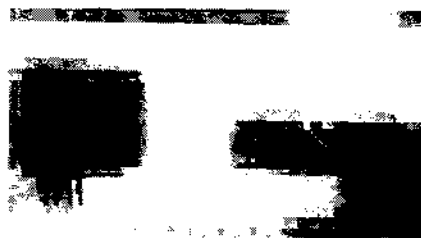
1. a kapcsoló semmihez nem ér *hozzá*, zavart nem okoz,
2. védve van attól, hogy véletlenül lekapcsoljuk, ez ugyanis a .rendszer összeomlásához vezet. Ilyenkor jön a Guru hibaüzenet és a gép újra indítása.

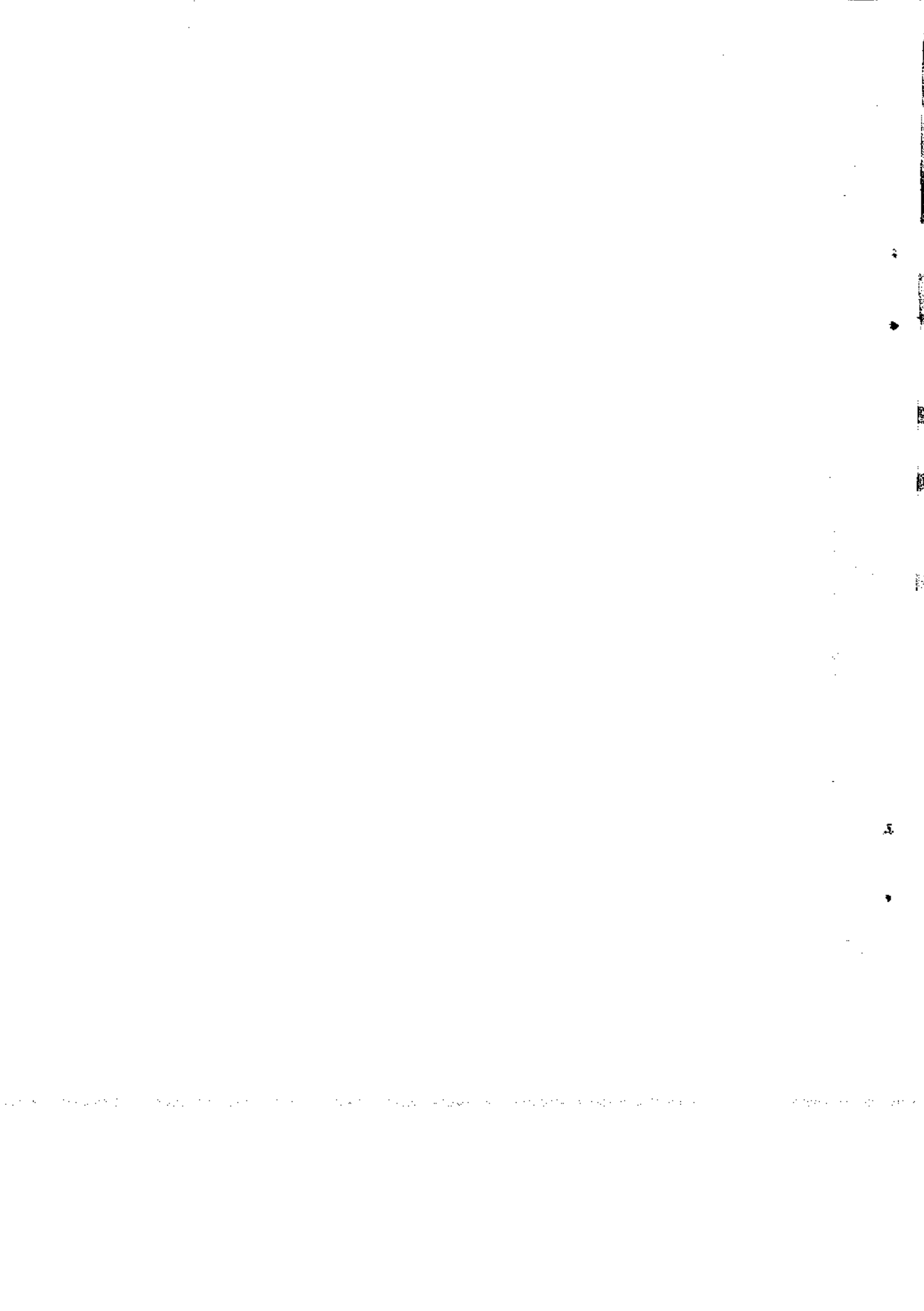
Van még egy harmadik mód is a memória bővítésére. A bal oldali bővítősínre maximum 8MB memóriapanelt kapcsolhatunk .



Fást RAM bővítő

Bizonyos esetekben ezt a panelt a merevlemez vezérlők tartalmazzák is, csak a szükséges memória IC-eket kell behelyezni. Aki HD-t vásárol, ajánlom neki, részesítse előnyben az ilyen típusokat.





B. Az első lépések

Ha sikeresen összeszereltük a gépet és megfelelően behangoltuk a TV-t vagy beállítottuk a monitort, az Amiga bejelentkező képét kell látnunk (kivéve, ha autobootos merevlemezünk is van a gépen). A bejelentkező kép 1.3-as kickstartnál egy kéz, 2.0-s rendszernél egy animáció. Ez a kép figyelmeztet bennünket arra, hogy a gép üzemkész, tegyünk be egy rendszerlemezt. Hogy ez miért kell, rögtön megmagyarázom.

A C64, Spectrum, stb típusú kisebb számítógépeken az operációs rendszer, amely a gép működtetését végző programok összessége, valamint a Basic nyelv a gép ROM-jába égetve volt megtalálható. Bekapcsolás után tehát rögtön rendelkezésre állt, a gép be is jelentkezett, általában a Basic-kel.

Nagyobb gépeken, mint például az Amigán, ezt a rendszert lemezzről kell betölteni. Látszólag ez rosszabb megoldás mint az előző, de ha belegondolunk, hogy a C64 operációs rendszere a gép élete során azonos, addig az Amigán egy újabb, korszerűbb rendszer használata bármikor megoldható, csak olyan lemezt kell betenni a meghajtóba bootoláskor (így nevezik a rendszer betöltődését és elindulását).

Persze azért nem ilyen szép az a menyasszony, hisz valami programnak csak kell lenni a ROM-ban, hiszen valakinek ki kellett tenni a "kezet" a képernyőre és a rendszert is be kell tölteni a lemezzről. Az ami ezt elvégzi, a KickStart. Ennek az új verzóra történő cseréje általában a ROM cseréjével jár. A kickstart verziója alapján különböztetjük meg az 1.2, 1.3 (A500) és a 2.0 (A50G+) rendszerű gépeket.

A ROM csere is elkerülhető, ha jó sok memória van a gépben. Ennek a módja az ún. Kickstart Disk. Az Amiga 1000 tulajdonosok ezt jól ismerik. Ez a lemez tartalmazza a kickstartot ami bootolás helyett betöltődik a memóriába, majd átveszi a végrehajtásjogát. Mint említettem, ehhez jó sok RAM kell, a 2.0 kickstart mérete 512 KB-át.

Az első lépések

Ha hiszed, ha nem, ez annyira jó megoldás, hogy például az Amiga 3000-ek szinte mind így működnek. Persze ott eleve több memóriát alkalmaznak és van merev lemez is, amiről a kickstart betöltése mindössze egy másodperc. Ja így könnyű!

Ha már a ROM-nál tartunk, összehasonlításképp megemlítem, hogy míg az előbbi példában említett gépek össz memóriája 64kB, addig az Amiga 500 és Amiga 2000 gépeknek csak a ROM-ja 256kB, az Amiga 500+, Amiga 600, Amiga 3000-nek pedig 512 kB!

1. Multitasking

A ROM nem véletlenül ilyen nagy, az Amiga rendszere igen bonyolult, rengeteg funkcióval rendelkezik. A legfontosabb közülük a "Multitasking". Ez "több feladat"-ot jelent, vagyis azt, hogy a gép egyszerre több programot is tud futtatni. Az egy időben futtatható programok számát csak a memória mérete korlátozza. A felhasználó számára teljesen úgy tűnik mintha a programok egyszerre működnének, valójában a gép szoftverrendszere szétosztja a processzor idejét és minden taszknak juttat belőle valamennyit. A taszk általában egy program. Hogy melyik taszk mennyi időt kap, az a fontosságától függ. A fontosságot a prioritásérték jelenti. Nézzünk a multitaskingra egy konkrét példát:

Tegyük fel, meg akarsz formázni egy doboz mágneslemezt - ennek mikéntjéről később lesz szó - mivel azonban ez hosszadalmas procedúra, közben a DPaint rajzprogrammal rajzolgatsz. A formázás is egy taszk, miközben rajzolsz, az is végzi a saját dolgát. Ha két floppy drive-od, vagy merevlemezed is van a géphez, miközben formázod egy másik lemezre nyugodtan ki is **mentheted** a rajzodat, az nem zavarja a formattálást.

Az Amiga multitaskingja olyan tökéletes, hogy akár ugyan arra a floppyra is képes több taszk egy időben írni, arról olvasni. **Ilyenkor** a floppy megosztva a működést, "egy kicsit ide, egy kicsit oda" elven szolgálja ki a lemezkezelést kérő taszkokat.

A személyi számítógépek multitaszkos operációs rendszerei közül az Amigáé messze a legjobb, egyedül az Apple Machintos veszi fel vele a versenyt, de az magasabb árkategóriájú gép.

Az Amiga 2.0-s rendszere magasan veri a PC-k Windows 3.1-ét is, annál könnyebben kezelhető, barátságosabb és azonos processzorteljesítményű gépeken összehasonlítva jóval gyorsabb. Ez nem az én saját véleményem, de alá tudom támasztani, mivel dolgozom mindkét géptípuson.

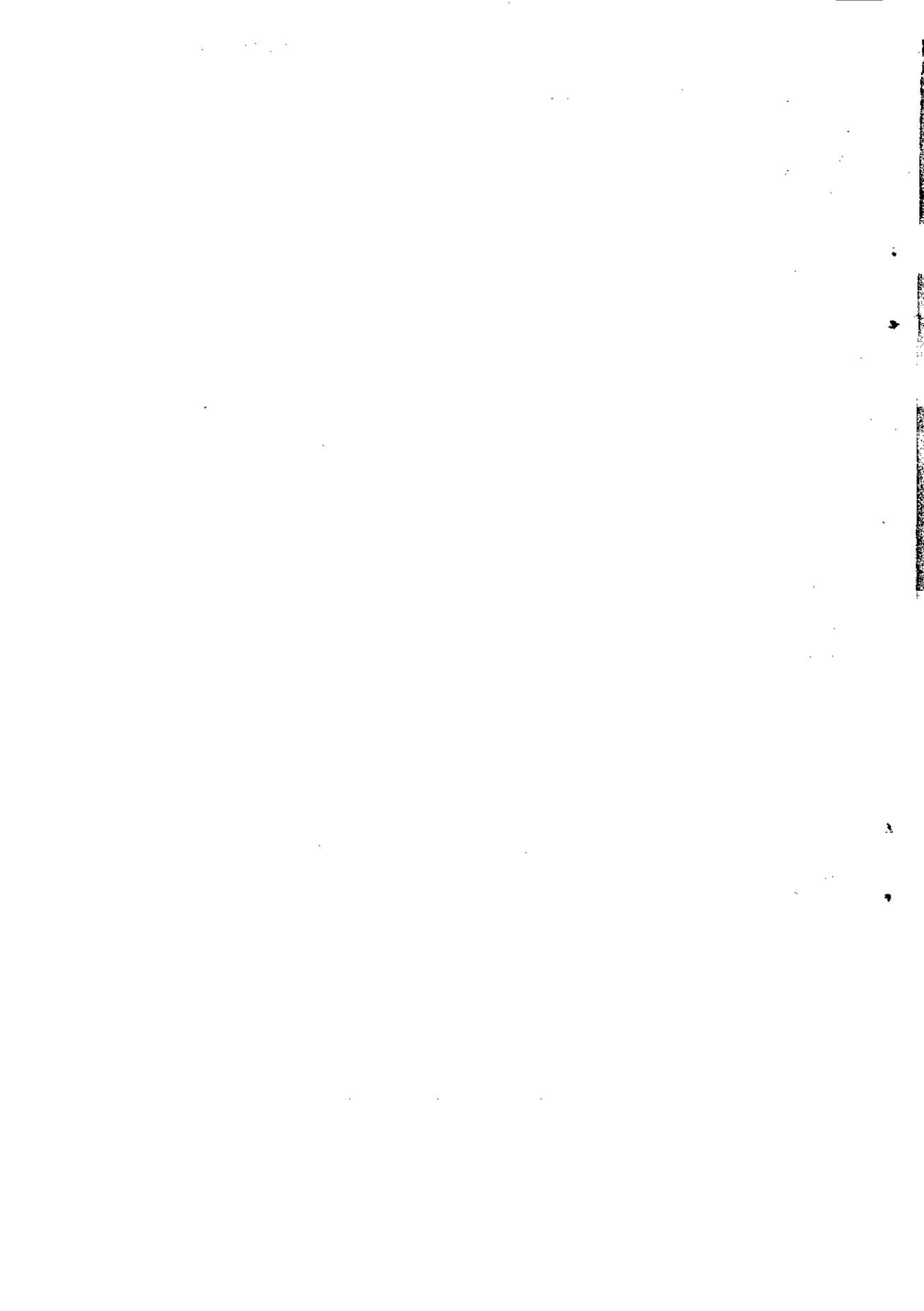
Az Amiga kiváló multitaszkja mellett felhasználóbarátságáról és magas szintű grafikus képességeiről ismert. Sok grafikus feladatra, mint például a metamorphosis, a 24 bites, 16.7 milliós színű képek feldolgozása, vagy a videoeffektek készítése, kategóriájában a legjobb.

Innentől kezdve szétválasztom az ismertetést. Először az A500 1.3-s rendszerének leírása következik. Természetesen ez az A500+ tulajdonosoknak és 2.0-s rendszer felhasználóknak sem lesz tanulság nélküli.

A könyv második részében az új rendszer újdonságai kerülnek nagyító alá, tehát csak azokról lesz szó, amik mások mint az 1.3-s Workbench és az A500. No itt előkerült egy új szó, a **Workbench**. Ez az Amiga grafikus kezelői felületének a neve, használata egyszerű, könnyen elsajátítható, vizuális. Ezen keresztül úgy vezérelhetjük a gépet, hogy szinte hozzá sem kell nyúlni a billentyűzethez.

A Workbenchen indított programok mind multitaszkban futnak.

Hát nem nagyszerű?



II. Az 1.2,1.3 kickstartú gépek használata

A. A bootolás

Eljutottunk tehát odáig, hogy betettünk egy rendszer lemezt a belső meghajtóba. A lemezt a kis fém tolokával előre, címkes oldalával felfelé kell betenni, a végén kissé megnyomni, amitől jól hallható kattanással beugrik. Kivenni a nyílás melletti gomb benyomásával lehet. Azért a belsőbe, mert az 1.3 rendszer (és persze az 1.2 is) csak a belső drive-ról tud bootolni. Kis hardver átalakítással rávehető, hogy kívülről töltsön, de ez más téma.



Lemez behejezése a belső drive-ba

A bootolás

Ennek akkor van jelentősége, ha a külső floppy 5 1/4 colos, ilyen lemezzel szeretnénk betölteni a rendszert.

Hogy a későbbiekben megtaláljuk a közös hangot, az ominózus lemez legyen mondjuk a géphez gyárilag adott **Workbench** nevű lemez. Amíg ez betöltődik, keressük meg, hová is tettük azt az egy-két doboz üres lemezt, amit külön vásároltunk, rövidesen szükség lesz rá!

Mivel látom, a lemezek elő voltak készítve és a gép még mindig töltöget, elmondom mit is csinál most.

Először is, mihelyt észrevette, hogy beraktál egy lemezt (igen, Ő észre veszi ezt, nem úgy mint más úgymond kiváló számítógépek, mint például a PC, ezt ne vegye senki személyeskedésnek, ez tény).

Erről jut eszembe, azért írtam, hogy "Ő", ugyanis az Amigának lelke van, ezt Te is fogod tapasztalni. Azért pedig hogy rossz példának mindig a sokat dicsért PC-t hozom fel, előre is elnézést kérek, de nekem kijutott mindkét géptípusból, a bőrömon érzem a különbséget. Majd igyekszem visszafogni magam.

Nos, ott tartottunk, hogy az Amiga észrevette az új lemez jelenlétét a meghajtóban. Mivel most egy rendszer lemezre várt, megvizsgálja a boot szektort. Na, ezt elmagyarázom.

1. A lemez sáv-szektor szerkezete

Az Amiga a floppy diszk mindkét oldalán 80-80 sávot tart fenn az adatok tárolására. Ezek a sávok tulajdonképpen koncentrikus gyűrűk, amelyeket a formázáskor jelölünk ki. Minden sáv fel van osztva 11 szeletre, ezek a szektorok. Egy szektor 512 bájttal tartalmaz információt. A szektorok helyét is kijelöli a formázás. A sávok számozása kívülről befelé 0-79, a szektoroké 0-10.

A szektorokat szokásos még blokkoknak is nevezni, azzal a kis különbséggel, hogy ha blokkról van szó, a számozás folytonos, például az 1. sáv 0. szektora a 12. blokk. Összesen 1760 blokk van, ezek számozása 0-1759 lehet.

A boot szektorok a 0. és 1. blokkokban foglalnak helyet. Az itt lévő 1024 bájt tartalmazhat egy kis programot, amit a kickstart bootoláskor megpróbál végrehajtani. Ha ez a program nincs meg, a lemez nem bootolható.

S. A normál boot mindössze annyit végez, hogy elindítja a ROM-ban lévő rendszert. Természetesen nem csak normál boot lehet ezen a helyen, hanem úgynevezett boot-loader is, ami a ROM-ban lévő program indítása helyett a lemezeiről másol a memóriába egy programot és azt indítja el. Ezen a módon működik a játék- és demo-programok nagy része. Előnye, hogy programozó kezét nem kötik a kickstart által támasztott feltételek, hátránya, hogy ezzel egyidejűleg nem használható a multitaszk (lásd a 16. oldalon), a program csak floppyról indítható, merev lemezeiről nem.

J f Maradjunk annál az egyszerű esetről, hogy a lemezen normál boot van, a ROM-ban lévő program indult el. Ez néhány apró inicializálási procedúra - ami tulajdonképpen a bootolás - után megkeresi a lemezen az "s" könyvtárban lévő "startup-sequence"-t, ami egy egyszerű szöveg fájl, ahol is le van írva, miket kell végrehajtani a gépnek. Erről egy későbbi fejezetben még közölni fogunk.

I No, éppen be is töltődött a lemez, nézzük tehát meg mit láthatunk a monitoron és mit végezhetünk a géppel.

NOVOCOMP

- Számítástechnika

Commodore

Amiga

Atari

PC

Kiegészítők, lemezek, monitorszűrők, egerek,
lemeztartó dobozok, szakkönyvek, leporellók
Amigára valamint PC-re felhasználói- és
játékprogramok előnyös áron

- Irodatechnika

Panasonic telefonok, faxok
írógépek, számológépek

- CD lemezek

Lemeztartók, boxok

- Makettek

Italeri, Esci, Monogram, Hasegawa, Tamiya,
Reveu, Airfix

Repülőgépek, harcjárművek, autók, motorok
Festékek, ragasztók, kiegészítők

5000, Szolnok Kossuth L. u. 12

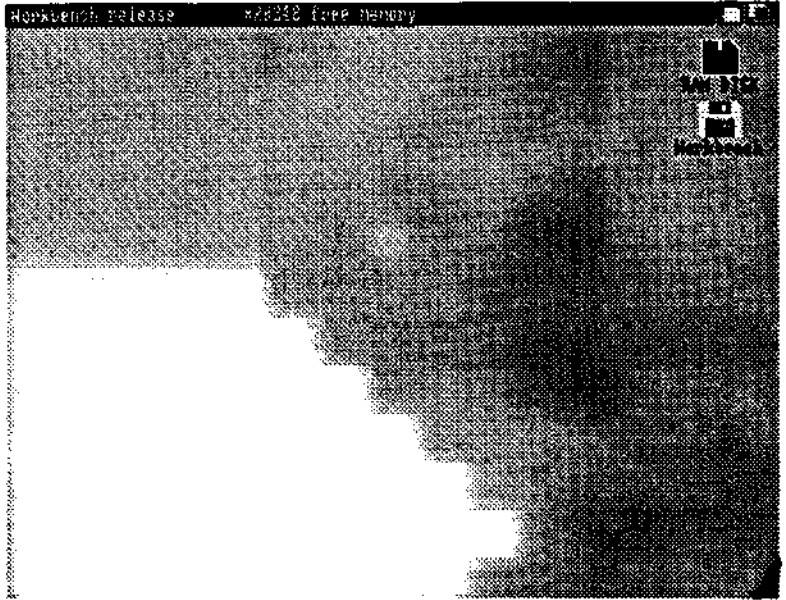
Tel.:56/344-605 Fax:56/370-179

NOVOCOMP

B. A Workbench

Mi is az a Workbench? Már említettem, hogy egy grafikus felhasználói felület. Segítségével úgy tudunk programokat indítani, másolni, a gépen dolgozni, hogy szinte nem is érintjük a billentyűzetet. Ebben két dolog van a segítségünkre: az ikonok és a menük.

A lemez betöltődése után egy ehhez hasonló képernyőt látsz:



A felső sorban a felirat jelzi, hogy a Workbench képernyőn, vagy a sor közepén a rendelkezésedre álló memória mennyiségéről kapsz tájékoztatást. A jobb szélén két kis jelet láthatsz. Ezek kapcsolók, úgynevezett gadgetek. Több fajtájuk is létezik. Ez a kettő a képernyő vagy ablak előtérbe hozását vagy háttérbe küldetését vezérli.

1. Képernyők és ablakok

Felmerült ka új kifejezés: "képernyők" és "ablakok". A képernyőn itt nem a monitor megjelenítő felületét értjük, most logikai értelemben használjuk. Amikor egy programot elindítasz, az kétféle módon teremthet vizuális kapcsolatot veled: vagy nyit egy új képernyőt, ami fizikailag ugyan úgy a monitorodon jelenik meg, de logikailag független a Workbench képernyőktől, vagy egy már megnyitott képernyőn nyit egy ablakot. Ezt rövidesen a gyakorlatban is kipróbálhatjuk.

A képernyő a monitoron mozgatható, ekkor az összes rajta nyitott ablak is vele mozdul. Ez tehát az egyik lényeges különbség az ablak és a képernyők között. A vízszintes elmozgatás az esetek nagy többségében nem lehetséges, mert a logikai képernyőt általában a maximális szélességgel nyitják a programok, így oldalra nem mozgathatók. A függőleges mozgatás viszont a képernyő magasságától függetlenül lehetséges. Hogy hogyan?

Kezdetben a bal felső sarokban látsz egy kis nyilat. Ha az egeret mozgatod, ez a nyíl is elmozdul. A nyilat pointernek, mutatónak nevezik. Ezzel tudunk a megfelelő dolgokra rámutatni, kiválasztani őket. A pointer többféle alakot felvehet, egy később ismertetendő módon magunk is megváltoztathatjuk, (lásd a 42. oldalon) Ha a gép el van foglalva, ezt általában egy "várakozási mutató"-val jelzi. Ilyenkor a szokásos alakja egy óra.

Vidd ezt a nyilat a képernyő legfelső sorába bárhová, a fejlécen belül. Most nyomd meg az egér bal, kijelölő gombját és tartsd is nyomva. Amíg a mutatóval "fogod" a fejléctet, azzal el is húzhatod, vele együtt az egész képernyőt. Próbáld ki.

Amikor a képernyőt lehúzó, az alja eltűnik a monitorról, de mivel ez csak logikai képernyő, továbbra is funkcionál, amikor visszahúzó, újból látható lesz minden része.

A képernyő mögött most nincs semmi, de ha olyan programot indítasz, amely saját "screen"-t (így nevezik a képernyőket) nyit,

az lenne a háttérben (vagy a Workbench, ha amaz volt elől és azt húztad le.)

A képernyők mérete, felbontása, színeinek száma különböző lehet, ismerkedjünk meg az Amiga képernyőparamétereivel, azok lehetséges értékeivel.

A látható méretű képernyők normál mérete 320 vagy **640 pont** széles és 200, 256, 400, vagy 512 pont magas lehet. A szélességgel összefüggésben áll a felbontás, a 320 pontos képernyő a LoRes, kislebontású, a 640-es a HiRes, nagyfelbontású. A Workbench screen mindig HiRes.

A magasság két dologtól is függ. Az egyik a videó norma, amely Pal, vagy Ntsc. Előbbi esetben - Európának ezen a részén ez az elterjedt - 256, vagy 512, utóbbi esetben 200, vagy 400 pont lehet a magasság. Az alap magasság mindig a kisebb, ezt lehet megkettőzni az ún. interlace móddal. Sajnos ebben az esetben a képernyő a váltottsoros megjelenítés miatt remegni fog, a remegést megszüntetni FlickerFliker áramkörrel és multisync monitorral lehet, de ezek egyiké sem olcsó dolog.

A képernyőn egyszerre 2, 4, 8, 16, 32 valódi, önállóan beállítható szín lehet. Minden szín a 4096 lehetséges egyike.

Az egyszerre megjeleníthető színek száma két trükkel megnövelhető. Az egyik trükk az Extra Half Bright (EHB). Ekkor a 32 valódi szín mellett mindegyik félfényerős változatát is használhatjuk, de ezek már nem független színek. A másik eljárás a Hold And Modify (HAM). Ilyenkor 16 valódi szín alkalmazható, emellett színkeveréssel mind a 4096. Hátránya ez utóbbi eljárásnak, hogy a 16 valódi szín kivételével, nem lehet minden pixel tetszőleges színű a színkeverés miatt. Legrosszabb esetben három pixelnyivel arrébb áll be a kívánt szín, addigra keveredik ki. Szerencsés esetben akár már a következő pixel felveheti a kívánt árnyalatot, ez az egymás melletti színektől függ. A HAM folyamatos színátmenetknél ideális, de a hirtelen színváltásokat nem jeleníti meg hűen.

A Workbench

A felbontások és színek száma nem kombinálható szabadon, a lehetséges párosításokat tartalmazza az alábbi táblázat:

<u>méret</u>	<u>színek</u>	<u>felbontás</u>	<u>memória igény</u>
320X256	2	LowRes	10240 bájt
320X256	4	LowRes	20480 bájt
320X256	8	LowRes	30720 bájt
320X256	16	LowRes	40960 bájt
320X256	32	LowRes	51200 bájt
320X256	64(EHB)	LowRes	61440 bájt
320X256	4096(HAM)		LowRes 61440 bájt
640X256	2	HiRes	20480 bájt
640X256	4	HiRes	40960 bájt
640X256	8	HiRes	61440 bájt
640X256	16	HiRes	81920 bájt

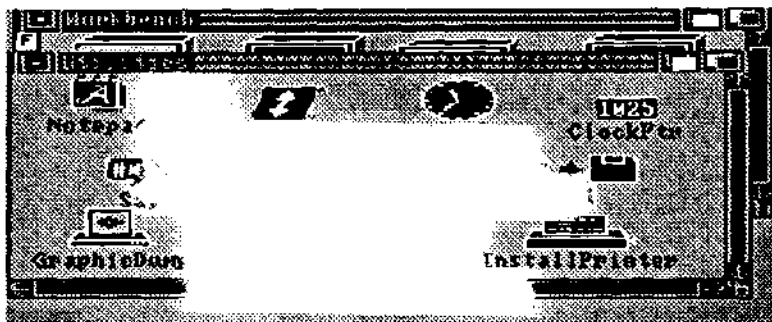
Az InterLace bekapcsolása nem befolyásolja a színek számát, de a memóriaszükségletet megduplázza.

Eddig csak a standard képernyőméretekről beszéltünk. Az ilyen méretű screen-ek hagynak maguk körül egy "overscan"keretet. Erre azért van szükség, mert a képernyő szélein az elektronsugár torzít. Az Amiga videó rendszere lehetővé teszi, hogy az overscan területet is használjuk megjelenítésre, ekkor nagyobb méretű képernyők nyithatók. Az overscant használó, látható méretű screen maximális mérete 368X290 (LoRes, NoLace), 368X580 (Lores, InterLace) 736X290 (HiRes, NoLace), vagy 736X580 (Hires, InterLace) lehet.

A látható méret azt jelenti, hogy a képernyő minden pontja egyszerre látható, de ez nem föltétlen van így. Nyithatók ennél nagyobb képernyők is, ekkor csak egy részük látható egyszerre, a teljes képernyő fölött a látható rész gördíthető.

A Workbench csak 640X256 vagy 640X512 (interlace) pixel méretű, hires képernyőt tud kezelni négy színben, de más programok - például a DPaint rajzprogram- használhatják a többi kombinációt is.

Az ablakok a képernyőn nyitott önálló területek, keretbe foglalva. Az ablakok a fejlécüknél fogva bárhová elmozgathatók a képernyőn belül.



Kinyitott ablakok

Az ablakokon és a képernyőkön különböző kapcsolókat találunk, amelyekkel azokat manipulálni tudjuk. Vegyük sorba a kapcsolókat - amelyeket gadgeteknek is nevezünk - és ismerkedjünk meg funkcióikkal.

A bal felső sarokban lévő gombbal az ablakot zárhatjuk be, ha ráklikkelünk.

A jobb felső kettővel az átfedésben lévő ablakok vagy képernyők előtérbe-háttérbe küldését végezhetjük.

A jobb alsó jelet megfogva és elmozgatva az ablak mérete szabályozható.

Ha az ablak tartalma nem fér el a látható méretben az alsó és a jobb oldali gördítő nyilakkal, vagy a vízszintes és függőleges csúszkák elmozgatásával állíthatjuk be a látható részt. A csúszkák mérete mindig arányban áll az ablak teljes méretéhez viszonyított látható méretével.

Nem biztos, hogy az ablak tartalmazza mindegyik most ismertett gadgetet.

2. Ikonok és menük

A Workbench screen-en két kis ábrát látsz (esetleg többet is többet is), ezek az ikonok. Feladatuk, hogy azonosítsanak egy lemezt, tartalomjegyzéket, programot, stb.

Ez a két ikon a "RAM Disk" és a "Workbench" lemez ikonja. A RAM Disk egy olyan memória rész, amit az operációs rendszer úgy kezel mint egy lemezt, programot tölthetünk bele, stb. A RAM Disk mérete azonban rugalmasan változik, mindig pont akkora, mint amit a benne lévő adatok igényelnek. A RAM Disk nem él automatikusan, azt előbb létre kell hozni és a gép reszetelésekor, kikapcsolásakor törlődik a tartalmával együtt.

Reszetelni a két "A" jelű (a space mellett jobbra és balra), valamint a CTRL billentyű lenyomásával lehet. Ekkor a gép majdnem ugyanazt végzi el, mint bekapcsoláskor, megpróbál bootolni és kiteszi a kezét, ha ez nem sikerül. Az "A" jelű billentyűknek Amiga a nevük és megkülönböztetjük őket a jobb-Amiga és bal-Amiga néven.

Felvetődik a kérdés, ha a RAM Disk nem él automatikusan, hogy-hogy mégis itt van? Nos, már említettem a "Startup-sequence" nevű fájlt, ő a felelős a dologért, de erről később.

Tekintsük most meg, mi van a Workbench lemezen. A neve nem biztos, hogy pontosan ez, de megtalálod az ikonját. Vidd tehát a mutatót az ikon fölé. A nyílnek a hegyén van a kijelölő pontja, ennek kell kép fölé lenni, a többi rész nem baj, ha lelóg. Most nyomd meg egy rövid időre a bal gombot.

Ezt a pillanatnyi gombnyomást nevezzük klikkelésnek. Megkülönböztetésül a jobb gomb lenyomásától, legyen ez a bal klikk, a másiké pedig a jobb klikk. Ha azt mondom nyomd le valamelyik egér gombot, az a továbbiakban azt jelenti, hogy miután lenyomtad, úgy is tartod azt.

A klikkelés hatására az ikon megváltozik, a legtöbb esetben ez invertálást jelent, de akár teljesen más kép is megjelenhet. Úgy mondjuk az ikon ifi van j*»1ft1v».

Egészen addig ez lesz a kijelölt tárgy, amíg máshol nem klikkelsz az egérrel. Ha az újabb klikkelés egy másik ikonon történik, az lesz kijelölve, ha a képernyő egy üres pontján, akkor a kijelölés megszűnik minden tárgyon.

Ha a Shift billentyűt nyomva tartod, miközben kiválasztod az ikont, az előzőleg kiválasztott is megtartja ezt az állapotát, tehát több ikon is kiválasztható egyszerre, amelyeken a műveletek ugyan- úgy hajthatók végre mint egy kijelölt ikonon.

Idáig nem használtuk az egér jobb gombját, de most eljött az ő ideje, nyomd hát meg. Ennek hatására a fejléc megváltozik, három felirat válik láthatóvá:

Workbench
Disk
Special

Ez a menü fejléce. A menükben parancsok vannak, amelyeket az egérrel lehet aktiválni, nem kell azokat begépelni. A parancsok csoportosítva vannak, a menü fejlécében az egyes csoportok neve látható. A Workbench menüben a munkaasztalt érintő parancsok, a másodikban a lemezekre vonatkozók vannak, a harmadik menü, a Special néhány egyéb művelet elvégzését teszi lehetővé.

Mielőtt tovább mennénk, meg kell tárgyalni a menük elemeit. A fejléccel már találkoztunk, ez jelenik meg a képernyő tetején. Erre rámutatva, miközben az egér jobb gombja folyamatosan nyomva van, legördül egy parancslista, innen kapta a menü a nevét: Roll Down (legördülő) menü. Ha lehúzzuk a pointert, az a menüpont, amelyekre az mutat, invertálódik. Ha most elengedjük a gombot, a menüparancs, amelyik inverz volt, végrehajródik. A menüpontoknak lehet más típusa is, van olyan, amelyből további alpontok nyílnak, ekkor a menü mellett jobbra, vagy balra egy almenü gördül le. A menük több mélységben is egymásba lehetnek ágyazva.

A másik típus a kapcsoló. Ha egy ilyet választunk ki, az nem műveletet hajt végre, hanem valamilyen állapotot kapcsol ki- és

A Workbench

be. Az állapot bekapcsoltságát egy pipa jel mutatja a menüpont előtt.

A harmadik típus neve után három pont áll. Ez jelzi, mielőtt a menüpont végrehajtná, egy ablak nyílik, ahol néhány kérdésre válaszolni kell. Ez a válasz lehet szöveges, vagy csak egy megerősítés valamilyen módon, például kapcsolóra klikkelve.

Némelyik menüpont után egy Amiga jel és egy karakter látható. Ekkor általában a jobb, de néha a bal Amiga gomb és a jelölt karakter együttes lenyomása azonos eredményű, mint a menüpont használata.

Van olyan menüpont is amely bizonyos szituációkban nem használható, ekkor annak neve halványan jelenik meg a listában, kiválasztásakor nem történik semmi.

A Workbench a menüiben nem használja az összes, itt felsorolt lehetőségeket.

Visszatérve a Workbench-hez, gördítsd le annak Workbench menüjét és válaszd ki belőle az Open pontot.

Az "Open" parancs azt jelenti, hogy kinyitni. Hatására a kijelölt lemez "kinyílik" azaz egy ablak keletkezik a képernyőn, amelynek fejlécében a lemez neve található, és ebben az ablakban megjelennek a lemezen lévő tartalomjegyzékek és fájlok ikonjai.

A kinyitott ablak automatikusan aktívvá válik, ezt a fejlécének színe jelzi. Az aktív ablak annyit jelent, hogy ezen az ablakon hajtnak végre a műveletek, például a Shell, vagy CLI ablakban csak akkor lehet valamit begépelni, ha az az aktív. Egyszerre mindig csak egy ablak lehet aktív.

3. A könyvtárstruktúra

A lemezen lévő adatok az Amigán struktúráíthatók, az összetartozó, hasonló célt szolgáló dolgokat csoportokba lehet foglalni, ez a csoport a tartalomjegyzék. Tudom sokan az Amigán keresztül először lépnek kapcsolatba a számítógéppel, így nekik egyszerűen kell elmagyaráznom ezt, elnézést kérek tehát azoktól, akiknek ez

már a könyökén jön ki. Akik C64-ről tértek át, azoknak is új lehet, hisz ott nincs könyvtárstruktúra.

Képzeld tehát el, hogy a lemez egy szekrény, a lemezen lévő fájlok pedig megfelelnek a szekrényben lévő holmiknak. Egy szekrényben lehet ezernyi apró tárgy, mondjuk egy gép, ami megfelel a program fájlak, vagy egy könyv, ami szöveg fájl, esetleg olyan dolog, ami ezekhez kell mint kiegészítő.

Ha sok holmi van egy szekrényben, egy bizonyos dolgot megtalálni igen nehéz, könnyen felborulhat a rend. Legyenek tehát a képzeletbeli szekrényünkben fiókok, és nevezzük ezeket tartalomjegyzékeknek. Ez még nem elég, a fiókon belül szükség lehet további rekeszekre, stb. A tartalomjegyzékek egymásba ágyazása bármekkora mélységű lehet.

Ezt a szerkezetet valahogy jelölni is kell, állapodjunk tehát meg egy egységes dologban. Nevezzük el a szekrényeinket. Ez két módon is lehetséges, vagy úgy hogy megmondjuk, melyik szoba melyik szekrényéről van szó, vagy saját nevet adunk a szekrénynek. Ugyan azt a szekrényt két módon is meghatározhatjuk. Legyen az első, amikor a helyét adjuk meg, a "Drives", meghajtók, az utóbbi a "Volumes", nevek jelölés mód. A szekrény neve után megkülönböztetésül tegyünk kettőspontot. A fiókot és az azon belüli rekeszeket sorrendben leírva, egymástól tört jellel válasszuk el.

A helyzet tehát a következő: Van egy előszobaszekrényünk, ez a "RAM:", amit név szerint "RAM Disk:"-nek mondunk. Ezen kívül van a "floppy" nevű szobában négy szekrénynek hely, most egy szekrény áll. Mondjuk, hogy ez a "DFO:", ahol D a drive, F a floppy rövidítése '0' pedig a szekrény sorszáma, ezen az alapon lehet még DF1:, DF2: és DF3: szekrényünk. A merev lemez jele általában DHO:, DH1: stb.

Kedvenc szekrényünk, a DFO: pozíciójában most a "Workbench:" nevű komód áll. Ha ebbe betesszük az ingünket, ezt így lehet leírni:

DF0:ing

A Workbench

vagy

Workbench:ing

Az ing mandzsettagomját pedig betehetjük az egyik fiók hátsó rekeszének bal oldalába:

DFO:felsőflók/hátsó rekesz/baloldal/mandzsettagomb

vagy

Workbenchifelsőflók/rekesz/baloldal/mandzsettagomb.

így már könnyű rendet tartani. De mi van, ha valamit sűrűn használunk? Például az étkezészet. Macerás mindig azt mondani, hogy:

Konyhaszekrényéből fiók/kanáltartó/villa

Egyezzünk meg a családdal, hogy a konyhaszekrény bal fiókjának kanáltartó rekeszét kanalasnak nevezzük, és ezt úgy vesszük, mintha önálló szekrény lenne, tehát:

kanalas:villa.

Igaz-e mennyivel egyszerűbb? Az ilyen egyezségeket nevezzük "Assigns", "hozzárendelés", vagy "logikai egység" néven. Ezekből a legfontosabbak:

C: az operációs rendszer parancsai

Devs: meghajtóprogramok

Fonts: betűkészletek adatai

L: kezelőprogramok

Libs: könyvtár rutinok

Sys: az a lemez, amelyen az operációs rendszer található.

S: programok indításának a segédeszközei, leírásai, többek között itt van a sokat emlegetett startup-sequence is, ami az egész gép indításának a leírása.

A lemez könyvtár szerkezetét fához szokás hasonlítani, ahol a gyökér maga a lemez, és ebből ágaznak szét a könyvtárak. Egy tárgy meghatározásához nem elég a neve, meg kell azt is adni, **hogymely útvonalon érhető el.**

Az egyes könyvtárakban lévő tárgyak száma nincs korlátozva, de két egyforma nevű nem lehet egy helyen. Természetesen egy másik útvonalon már lehet ugyanolyan nevű tárgy.

Ott tartottunk tehát, hogy kinyitottuk a Workbench lemezünket. Ez a művelet nyit egy ablakot a képernyőn és benne újabb ikonokat jelenít meg. Ezek az ikonok reprezentálják a lemez fő tartalomjegyzékében lévő dolgokat. Jegyezzük **meg**, hogy nincs minden tárgynak ikonja, ezek nem is érhetők el közvetlenül Workbench-ből.

Nézzük most meg, mik vannak a most kinyitott ablakban! Találsz néhány fiókra emlékeztető ikont, ezek tartalomjegyzéket jelölnek. A tartalomjegyzék neve az ikon alatt olvasható.

Vizsgáljuk meg, mit tartalmaznak az egyes fiókok. Nyisd ki először az **Empty** nevűt. Ennek a módja ugyan az, mint a lemez kinyitásának: kijelölöd, majd az Open paranccsal kinyitod. Elárulom, ennél van gyorsabb mód is, vidd a mutatót az ikon fölé és gyors egymás utánban klikkelj kettőt. Ennek a hatása teljesen azonos az Open-ével. Az ilyen klikkelést dupla klikknek nevezzük. Hogy mennyi idő telhet el a két klikk között, hogy azok dupla klikknek számítsanak, egy később bemutatásra kerülő módon beállítható, (lásd a 38. oldalon)

Kinyílt tehát az Empty fiók, ebben azonban semmi sincs, a fiók neve is erre utal. Azért van, hogy általa létre tudjunk hozni saját tartalomjegyzéket.

4. Tartalomjegyzék létrehozása, **másolása**

Hozzunk is létre egy tartalomjegyzéket mondjuk "Saját" néven. Ezt azonban ne a Workbench lemezen tegyük, nehogy tönkre menjen. Különbösen is, ez a lemez írásvédett. Ez onnan tudható, hogy a lemez jobb felső sarkában lévő kapcsoló fel van tolvá, a lyukon át lehet látni. Ahhoz, hogy egy lemezre írni lehessen, vagy törölni róla, írásgengedélyezetté kell tenni. Ez a kapcsolónak a lemez közepe felé tolásával érhető el. Ha a kapcsoló teljesen eltakarja a nyílást, csak akkor lehet a lemezre írni. Az ere-

A Workbench

deti Workbench-ről legtöbb esetben hiányzik is ez a kapcsoló, egyáltalán nem írható a lemez.

Akkor hol készítsünk tartalomjegyzéket? A RAM-ban! Ehhez másoljuk át az Empty ikont ide. A művelet nagyon egyszerű, fogd meg a mutatóval az ikont és húzd el pontosan a RAM Disk ikon fölé, majd engedd el. Az ikon "beleesik" a RAM-ba. Erről meggyőződhetesz, ha kinyitod a RAM-ot, abban is lesz Empty ikon.

Jelöld most ki a RAM-béli Empty-t, majd add ki a Workbench menü **Duplicate** parancsát. Megjelenik egy új ikon, amely azonos az Empty-vel, de a **"copy of Empty"** nevet viseli. A neve még nem az igazi, meg kell változtatni, de az új tartalomjegyzék már kész.

Ezzel a módszerrel nem csak tartalomjegyzékeket, hanem programokat, sőt egész lemezeket is megduplázhathatunk. Ezt rögtön ki is próbáljuk. Másolatot készítünk a Workbench lemezről és ezentúl azt használjuk, nehogy az eredeti megsérüljön.

Jelöld ki a Workbench ikonját, majd válaszd ki a Duplicate parancsot A másolás némiképp máshogy zajlik, mint az előbb, szükséges lesz egy üres, akár formázatlan új lemez. Rövid töltögetés után egy **"kérdő"** jelenik meg a bal felső sarokban.

Kérdővel gyakran fogunk találkozni, tisztázzunk tehát mi is az. A kérdőn keresztül az Amiga valamilyen választ, utasítást vár tőlünk. Több fajtája létezik, ez a legegyszerűbb, két részből áll.

Az egyik az üzenet, amely megkér, hogy tegyük be a másolandó, forrás lemezt a DFO: meghajtóba. A másik rész, az igen-nem válaszok. A bal oldali válasz az igen, jelen esetben a **"Continue"**, folytasd válasz, ezzel jelezzük, hogy eleget tettünk a kérésnek. A másik a nem válasz, most a **"Cancel"**, amely közli a géppel, hogy meggondoltuk magunkat, felejtse el amit csinált. Választani úgy lehet, hogy valamelyik válaszra ráklickelsz bal gombbal. A billentyűzetről a bal-Amiga, majd vele együtt a v

gomb lenyomása a bal oldali, a b gomb pedig a jobb oldali választ jelenti.

Mivel most a Workbench lemezt másoljuk, az már benne van a DF0:-ban, válaszoljunk tehát igent. A másolás megindult, folyamatosan látjuk hány sávot olvasott már be a gép és mennyi van még hátra. A memória méretétől függően esetleg több lépésben végzi el a másolást, mivel a teljes lemeztartalom egyszerre nem fér el a gép memóriájában.

Miután a lemezmásoló program beolvasott annyi sávot, amennyit tudott, újból egy kérdező jelenik meg, amely kéri, hogy tegyük be a cél lemezt a DF0:-ba. Ha ez megvan, újból adj igen választ a Contiune felirattal. A program most kiírja a memóriából a lemezre a sávokat. Ha a lemez nem volt formázva, azt megteszi. A másolás előrehaladtát most is folyamatosan látod. Ha egy menetből nem lehet átmásolni a lemezét, újból megjelenik a kérdező, amely a forráslemezt kéri. Addig kell a lemezeket cserélgetni, amíg az egész át nem másolódik.

Amíg a másolóprogram működik, megjelenik egy újabb ikon a "DF0:Busy" felirattal. Ez arra figyelmeztet, hogy a DFO: foglalt, most másra ne használj, mint a másolásra. Ha a másolás kész, a másoló ablaka és a lemezfoglaltságot jelző ikon eltűnik, utóbbi helyett pedig megjelenik egy "copy of Workbench" nevű ikon.

Nézzük most meg, hogyan változtathatjuk meg a neveket

S. Átnevezés

Nevezzük át a "copy of Workbench" nevű lemezt "Workbench"-re. Ehhez jelöld ki az ikont, majd válaszd ki a Workbench menü "Renatne" parancsát. Egy csík jelenik meg a képernyő közepén, benne a lemez jelenlegi neve. A képernyő fejlécén utasítást kapsz, hogy írd be az új nevet, majd nyomd meg a "return"-t, amikor kész. Előfordulhat, hogy hiába nyomod le a billentyűket, nem íródik semmi az ablakba. Ez azért van, mert valahová klikkeltél az ablakon kívül. Klickelj bele ebbe a csíkba, aminek hatására megjelenik abban a kurzor, egy kis téglalap.

A Workbench

Ahol a kurzor áll, ott fog megjelenni a leütött billentyűnek megfelelő karakter.

Vidd most a kurzort a nyilakkal a név elejére és a Del gomb többszöri lenyomásával töröld le a "copy of" feliratot, csak a "Workbench" maradjon, még szóköz se legyen előtte. Ha kész, nyomd le a Retura-t. Természetesen más nevet is adhatsz a lemeznek. A név legfeljebb 30 karakter lehet lemezeknél, 25 tartalomjegyzékeknél és más ikonoknál.

Ha elkészültél, reszeteld a gépet, és a Workbench másolatáról töldsd be a rendszert, ugyanis a gép általában azon a lemezen keresi a programokat, parancsokat és egyéb dolgokat, amelyről bebootolt. Hiába készítettünk egy másik Workbench lemezt, azt még nem tudjuk teljeskörűen használni, az Amiga minduntalan visszakeré a boot lemezt.

6. Az ablakok bezárása

Ha betöltődött a Workbench, nyissuk ki a lemezt. A képernyő fejléc sorában látható, a szabad memória méretét jelző szám valamennyivel csökkent. Megállapítható, hogy az ablakok memóriaterületet vonnak el a géptől. A már nem kellő, felesleges ablakokat tehát be kell zárni, hogy az általuk lefoglalt memória felszabaduljon. Ehhez jelöld ki a bezárandó lemezt, vagy fiókot és válaszd ki a workbench menü **Close** parancsát. Az ablak eltűnik és kis idő múlva felszabadul a lefoglalt memória. Van más módja is az ablak bezárásának, az ablak bal felső sarkán lévő ablakzáró gadget. Elég csak ide klikkelni a bal gombbal.

Miután megismerkedtél ezekkel az egyszerű műveletekkel, folytassuk a lemez tartalmának a vizsgálatát. Nyisd ki az **Expansion** fiókot. Ez szintén üres. Ide a géphez kapcsolt bővítő felszerelések - pl. merev lemez - kezelőprogramjai kerülnek a későbbiek folyamán, számunkra most nincs jelentősége, be is lehet zárni.

7. A Workbench beállításai

A következő fiók a Prefs. Itt olyan programok foglalnak helyet, amelyekkel be lehet állítani a Workbench munkaasztal jellemzőit. Ezeket a jellemzőket a devs tartalomjegyzékben lévő system-configuration nevű fájl tartalmazza, a programok ezt módosítják.

A legfontosabb a Preferences, indítsd el. Ez kétféle módon tehető meg, vagy kijelölöd és alkalmazod a Workbench menü Open parancsát, vagy duplán ráklickelsz.

A színbeállító részen a Workbench négy színét lehet meghatározni. Válaszd ki a beállítandó színt, majd a három pötty húzogatóásával beállíthatod az adott szín vörös, zöld és kék összetevőjét. A Reset Colors visszaállítja az eredeti színeket.

A Text kapcsolóval azt lehet beállítani, hogy a képernyő egy sorába hány karakter íródjon ki. Ha TV-t használsz monitorként, állítsd 60-ra, hogy a nagyobb karakterek miatt jobban olvasható legyen a szöveg. RGB monitornál nyugodtan kapcsolod be a 80-as opciót:

A középső, képernyő pozíció részben lehet a logikai képernyő helyzetét állítani a monitor fizikai képernyőjén. Fogd meg az egerrel a kis sarok jelet, majd mozgasd el. Hatása azonnal látható. Ha sikerül jól beállítani, engedd el a jelet.

A jobb oldali egerrel jelölt két oszlopban az egér paraméterei állíthatók. A bal szélsővel az, hogy az egér egységnyi elmozdítására a mutató mennyit mozduljon a képernyőn. Az 1-est választva mozdul legtöbbit, a 4-est választva az egeret széles mozdulattal lehet mozgatni, a mutató irányításához.

A jobb oldali oszlopban a pötty mozgatóásával azt lehet szabályozni, hogy két egymást követő klikkelés közt mennyi idő telhet el, amíg az dupla klikknek számít. Ha felhúrod a pöttyöt, igen gyorsan kell egymás után megnyomnod az egér gombját a dupla klikkhez. Ha lehúrod, a két gombnyomás közt elég sok idő lesz.

Ezek felett található az **Interlace** kapcsolója. Ha On állásba helyezed, a képernyő felbontása 256-ról 512 sorra emelkedik, de sajnos remegni fog. Ez a remegés egy **ficker-flixer** nevű áramkörrel és multisync monitor használatával küszöbölhető csak ki. Az Interlace úgy érvényesíthető, ha bekapcsolod, kimented a konfigurációt, majd újra bootolsz.

Fent középen a billentyűzetre vonatkozó két beállítási lehetőség csúszkája található. A felsővel azt lehet beállítani, hogy meddig kelljen lenyomva tartani egy billentyűt, hogy az ismételni kezdjen. A csúszkát balra húzva ez az idő csökken. Az alsó csúszkán az ismétlés sebessége állítható, balra húzva lassabban ismételik a billentyűk.

Bal szélen az időt és dátumot lehet állítani. Ez nem a belső óra ideje, az ugyanis nincs minden gépben, hanem egy lemezen tárolt idő. A bootoláskor a belső órából nem töltjük be a valós időt, ezt használja helyette a rendszer. Az állítás a két állító nyíllal lehetséges. Kiklikelj arra a számjegyre, amelyiket állítani akarsz, majd használd a nyilakat.

Az idő beállítása fölött van két kapcsoló. A Reset All minden beállítást az eredeti gyári értékre változtat. A Last Saved újra betölti a lemezeről a konfigurációs fájlt.

Van még néhány kapcsoló, amelyekről idáig nem szóltam. A jobb alsó sarokban a **Save** kimenti a beállításokat a Devs: könyvtárba a system-configuration néven, egyúttal használatba is veszi azt. A **Use** csak használatba veszi a beállításokat, de nem menti ki, a következő bootoláskor ismét a lemezen lévő beállítások lesznek érvényben. A **Cancel** kilép a programból, anélkül, hogy a beállításokat figyelembe venné.

A **Change Printer** a printer beállító programot indítja, az **Edit Pointer** a mutató-rajzoló, a **Change Serial** pedig a soros modem beállítóba lép át. Ezek önállóan, a Preferences nélkül is elindíthatók, a Prefs könyvtár Printer, Pointer és Serial programjaival.

Lépjünk először be a printer konfigurációs részbe.

8. A printer beállításai

A bal felső sarokban választhatunk, hogy a párhuzamos vagy a soros portot akarjuk-e használni. Ez attól függ, hová csatlakoztatuk a printert. Mellette egy listát láthatunk az érvényes printer meghajtókról, amelyek között a le-fel nyilakra klikkelve lehet választani. Most még csak két meghajtó van itt, a generic és a Custom. Ez utóbbi beépített meghajtó, a másik a Devs:printers könyvtárban található. Ebbe a könyvtárba kell lemásolni a megfelelő típusú meghajtót, hogy aktiválni tudjuk. A bemásolás módját is rövidesen ismertetem. Addig csak annyit, hogy a többi printer driver a géphez adott másik Extrás nevű lemez devs/printer könyvtárban van.

A Paper Size nevű ablakban választhatunk papír méretet,

US Letter - amerikai levélpapír
US Legal - amerikai szabvány
Narrow Tractor - keskeny papír
Wide Tractor - széles papír
Custom - szokásos méret

Legjobb ez utóbbi, mert akkor a printer által használt méretet használja a rendszer.

A Lenght az egy lapra nyomtatható sorok számát jelenti.

A Paper Type a papír típusát állítja be. Ez lehet "Fanfold" azaz leporelló és "Single", egyes lapok.

A Quality a nyomtatás minősége, amely "Draft", durva és "Letter" levél minőség lehet.

A Left Margin a bal margó pozíciója, a Right Margin a jobb margóé. Normál, 80 karakteres nyomtatónál az 5 és 75 érték azt jelenti, hogy mindkét oldalon 5-5 karakter széles margót hagyunk.

A Pitch a használt írássűrűség, karakter/inch-ben.

A Spacing az mutatja, hogy egy inch-ben (2,54 cm) hány sor legyen. Ez 6 vagy 8 lehet.

A Workbench

Maradt még egy kapcsoló a jobb szélén. A **Graphic 1** újabb beállítási lehetőségeket hoz elő, ahol a grafika nyomtatásának a paramétereit állítjuk be.

Találunk itt két ismerős kapcsolót, az **"OK"**-t amely elfogadja a beállításokat és visszalép az előző szintre, és a **"Cancel"**-t ami szintén visszalép, de nem veszi figyelembe a változtatásokat.

Az ablak felső részén, a **Threshold** címszó alatt azt állíthatjuk be, hogy fekete-fehér nyomtatáskor mi legyen a fekete szín küszöbértéke.

Az **Aspect** kapcsolóval a kinyomtatott grafika helyzetét határozhatjuk meg. A **Hozinttal** normál helyzetben, vízszintesen nyomtatja a grafikát, a **Vertikal** 90 fokkal elforgatva.

Az **Image** kapcsolóival lehet beállítani, hogy pozitív, vagy negatív képet kérünk. Negatív kép esetén a **Threshold** fordítva működik!

A **Shade** opciói a nyomtatott kép árnyalására vonatkoznak. A **"Black and White"** fekete-fehér képet eredményez. A **Threshold**nak ekkor van jelentősége.

A **"Gray Scale"** és a **"Gray Scale 2"** kétféle árnyalási algoritmus segítségével szürke árnyaltos képet állít elő.

A **"Color"** a színes nyomtatást kapcsolja be. Természetesen csak színes nyomtatónál van értelme.

Visszatérve az előbbi szintre, választhatjuk még a **"Graphic 2"** nyomógombot is, de csak az 1.3 Workbench-ben, az 1.2-ben nem. Itt fejlettebb grafikai beállításokra van lehetőség.

A **Smoothing** kapcsolóval a grafikát lágyítani lehet, az éles kontúrok elmosódnak.

A **Color Correct** kapcsolóval a három színösszetevőt külön-külön csökkenthetjük. Például egy erősen vörös árnyalatú **kép** nem biztos hogy jól mutat szürke árnyalatban. Ha bekapcsoljuk az **R** kapcsolót, akkor csökken vörös túlsúlya, javul a szürke kép minősége.

On van bekapcsolva, ez az érték nem számít, a grafika középre kerül.

A Dithering kapcsolóval a színátmenetek fokozatosságát szabályozhatjuk.

A Limits kapcsolóival különböző korlátozásokat róhatunk a grafikára. Az **Ignore** figyelmen kívül hagyja a korlátozásokat, a **Bounded** a lap keretéhez viszonyít, az Abszolúte a grafika méretét határozza meg inch-ben, a **Pixel** ugyanez, de pixelben, a **Multiply** pedig azt adja meg, vízszintesen és függőlegesen hányszor ismétlődjön meg a grafika.

Egyszerre csak egy lehet aktív a fenti kapcsolók közül. A szükséges számértékeket bal oldalt lehet megadni, felül a szélesség, alul a magasság értékét.

A Density a grafikát alkotó pontok sűrűségét állítja be.

A Scaling a kép méretarányára vonatkozik.

Természetesen minden grafikus beállításnak csak olyan nyomtató esetén van értelme, amely támogatja a grafika nyomtatását. Ajánlatos ezt ki is próbálni a különböző beállításokai, mert a kapcsolók nagy részének nyomtatótól függően más és más lehet a hatása.

9. A soros port konfigurálása

Visszatérve a Preferences Editor főkéjébe, a **Change Serial-t** választva a soros, RS232C port paramétereit állíthatjuk be.

A Baud Rate a port átviteli sebessége baud-ban. A le-fel nyilakra klikkelve állítható.

A **Buffer Size** az adatátviteli csatorna bufferének a mérete, szintén a le-fel nyilakkal állítható.

A Read Bits az adatbitek száma input esetén, a **Write Bits** ugyanez outputnál.

A Stop Bits a stop bitek száma.

A Parity a paritásvizsgálat módja. None-nincs, Even-páros paritás, Odd-páratlan paritás.

A Handshaking a csatlakozás kezelésének a módja.

10. A mutató szerkesztése

Ha a főképen az **Pointer Editor** kapcsolót nyomod meg, egy új képernyő nyílik, amelynek nincs se záró, se képernyőváltó gadgetje. Ha mégis vissza akarsz térni a Workbench screen-hez, nyomd meg a **bal-Amiga** és **"n"** billentyűket egyszerre. A Workbench ismételt háttérbe küldését vagy a megfelelő gadgetre klikkeléssel, vagy a **bal-Amiga+m** billentyű kombinációval érheted el.

A képernyő legnagyobb ablakában a mutató kinagyított képe látható, ez a szerkesztő terület. Mellette jobbra négy kis képen a mutató látható eredeti méretében, a Workbench négy színe előtt. A jobb oldali kapcsolók közül a **Restore** visszaállítja a mutatót, a **Clear** pedig törli azt a szerkesztőből. A **Reset Color** visszaállítja az eredeti színeket, a **Set Point** pedig beállítja a mutató érzékeny pontját

Ez az a pont, amelyen keresztül a mutató pozicionálva van, és amelyik a kijelölést végzi. Kapcsold be ezt, majd klikkelj a szerkesztő területen oda, ahová a pontot helyezni akarod.

Alul az **OK** és **Cancel** gombok mellett balra a mutató színeit lehet beállítani.

Ezzel be is fejeztem a Preferences Editor ismertetését.

11. System tartalomjegyzék

Nézzünk most be a System fiókba. Itt, ha minden jól megy, kilenc ikon található.

A CLI Dcon egy úgynevezett CLI promtot hoz elő a megnyitódó ablakba. Itt a billentyűzeten keresztül táplálhatunk be DOS parancsokat. Ismertetését majd később olvashatod.

A DiskCopy-val már találkozhattál. Ez a program végezte a Workbench lemez másolását. Innen is elindíthatod, a következő **módon: jelöld ki a másolandó lemezt, nyomd le a shift gombot,**

amivel multi select, azaz többes kiválasztás módba kerülsz. A

shift nyomva tartása mellett klikkelj kettőt a DiskCopy ikonon. Ezzel átadod paraméterként a másolandó lemezt a program számára. Ezt a műveletet jól jegyzed meg, sokszor kell hasonló módon eljárnod.

A következő ikon a Formát. Ezzel a lemezek formázását, adatok fogadására való előkészítését lehet elvégezni. Minden új lemezt használatba vétel előtt meg kell formázni, kivéve, ha DiskCopy-val, vagy más hasonló, egész lemezes másolóval másolunk rá, ekkor ugyanis automatikusan formázódik. Természetesen a már formázott, használatba vett lemez is formázható, de ekkor minden információ megsemmisül, ami eddig rajta volt.

A program használata az előzőéhez hasonló, jelöld ki a formázandó lemezt, majd multi select módban indítsd el a programot. A megjelenő kérdező figyelmeztet, hogy ez visszavonhatatlan adatvesztést okoz az adott lemezen. Ha igen választ adsz, a formázás megindul, melynek állapotáról folyamatos információt kapsz. Most is megjelenik a DF0:Busy icon. A megformázott lemez az Empty (üres) nevet kapja, de ez megváltoztatható.

A lemez formázása elvégezhető más módon is. Jelöld ki a lemezt, majd válaszd a Disk menü Initialize parancsát. Az eredmény ugyanaz, a menüparancs is a Formát programot használja.

A FastMemFirst utasítja a rendszert, hogy elsősorban a Fást Memóriát használja. Ez a programok gyorsulását idézi elő. No, most megint egy új dologgal találkoztunk.

Az Amiga a memóriát kétféle módon kezeli. A régi A500-ok a címtartomány első fél megabájtját, az újabb A500-asok az első egy megabájtot, az A500+ pedig az első két megabájtnyi területet kitüntetetten használja. Ez az a terület, amit a zenei és videó chippek is el tudnak érni, tehát zene és hang csak itt lehet. Ezt a memória területet Chip Memóriának nevezzük. Mivel a processzor ezt a részt közösen használja a már említett feladatokat végző "Custom Chip"-ekkel, a memória elérése több időt vesz igénybe, bizonyos esetekben a chipeknek előjoga van a processzorral szemben. Ez azt eredményezi, hogy a program futása lassabb mint az elméletileg lehetséges.

A memória másik típusa a **Slow-Fast Ram**. Ilyen a gép aljába helyezett memória bővítő. Ezt már nem tudják elérni a custom chippek, de bizonyos hardver okokból, még mindig nem használható teljes sebességgel.

A harmadik típus, a gép bal oldalán lévő bővítősín, ahová max. 8MB **Real-Fast Ram** helyezhető. Amelyik merevlemez memóriát tartalmaz, az is ilyen. Ezt a memóriaterületet már csak a CPU használja, az elérése tehát a leggyorsabb. Ha ide helyezünk egy programot, az gyorsabban fut, mint a chip memóriában.

A program arra utasítja a rendszert, hogy amit tud, tegyen be a gyorsabb RAM-ba.

A NoFastMem a FastMemFirst ellentéte. Lefoglalja az összes Fast memóriát, ezáltal az utána indított programok kénytelenek a Chip memóriát használni. Régebbi programok ugyanis hajlamosak Fast memória esetén le- "Guruzni". Ez gyakorlatilag a rendszer logikai összeomlását jelenti. Időnként fogsz vele találkozni, a fekete képernyőn villogó piros keretben piros felirat a Guru, minden Amigás réme. Az összes adat, ami a memóriában volt, elvész. A bal egér gomb megnyomásakor a gép újra bootol.

A Guru nem az Amigások bosszantására lett kitalálva, ez a gép hibakezelő rendszere, akkor merül fel, amikor valamelyik program súlyos hibát követ el, amit a rendszerprogram már nem bír kezelni.

A lefoglalt Fast memóriát a NoFastMem ismételt alkalmazásával nyered vissza.

A MergeMem a bővített memóriája gépeken megpróbálja a memóriaterületeket egybe fűzni, amennyiben lehetséges.

A következő program a Setmap. Ez egy billentyűzet kiosztási térképet helyez el a gépben, ami által tetszőleges karaktereket rendelhetünk az egyes billentyűkhöz. Megvalósítható a magyar ékezetes billentyűzet is. A Setmap nem tartalmazza a kódtáblát, azt el kell készíteni egy arra alkalmas programmal és a **devs:keymaps** tartalomjegyzékbe tölteni. Az eredeti Workbench

is tartalmaz jó- néhány billentyűzet kiosztást, ezek karakter hozzárendeléseinek ábrája az eredeti kézikönyvben található.

Felvetődik egy fogós, ravasz kérdés: Mivel a devs:-nek nincs fiók ikonja, nem lehet elérni a benne rejtőző kódtérképeket, hogyan használjuk a multi select-et a program elindításához? Nos, ezt máshogy kell megoldani. Jelöld ki az ikont és használd a Workbench menü **Info** parancsát. Egy új ablak jelenik meg, ahol a programról kapunk és adhatunk némi információt.

A bal-felső sarokban a program neve található. Nem csak programról, hanem bármi másról is lehet Info-t kérni, aminek van ikonja, ugyanis abban tárolódnak az adott tárgyról a kiegészítő információk. Az ikon és az információ egy külön fájlban van tárolva a "**tárgynevcinfo**" fájlban, a lemezé a **Disk.info-ban**. A most látható információk tehát a **Setmap.info** fájlból kerültek elő.

A tárgy neve alatt annak típusa van feltüntetve. Ez lehet **Tool**-program, **Project**-terv, **Drawer**-tartalomjegyzék, **Disk**-lemez és **Garbage**-szemétkosár. Ez utóbbi egy speciális tartalomjegyzék.

A típus alatt néhány méretre vonatkozó jellemző található. Tool típusnál ez a program mérete, és az, hogy hány szektort foglal el. Drawer és Garbage típusnál itt nincs semmi, Disk típusnál pedig a lemezen összesen lévő, a használt, majd az üres blokkok, végül a blokkonkénti bájtok száma. Ez a legtöbb esetben 488.

A Tool és Project típus esetén van még egy **Stack** bejegyzés is, az utána álló szám a program által használt memóriapuffer mérete.

A jobb oldalon a tárgy státusza található. Ez a kapcsolókra klikkelve megváltoztatható. A **Not Archived** azt jelenti, hogy a fájl még nincs archiválva, azaz még nem készült róla biztonsági másolat. Ez csak egy emlékeztető jellegű kapcsoló. A **Readable** azt jelenti, hogy a fájl olvasható, tartalma például egy szövegszerkesztővel megtekinthető. A **Writeable** jelentése, hogy a fájlt módosítani is lehet. Ha ez ki van kapcsolva, nem lehet felülírni egy azonos nevű fájlal. A **Deletable** engedélyezi, vagy

A Workbench

le- tiltja a fájl törölhetőségét. Ha itt **Not Deletable** áll, a fájlt nem lehet letörölni a lemezről.

Kissé lentebb, típustól függően két vagy három vízszintes csík áll. Tool, Drawer és Garbage típusoknál a felső a **Comment**. Ha ide belekattintasz, megjelenik benne a kurzor és 78 karakernyi megjegyzést fűzhetsz a fájlhoz.

A **Tool Types** mezőben paramétert lehet adni a programnak, pont erre van most szükségünk, de előbb nézzük végig az egész Info ablakot.

A **Disk.info** ablakában a Comment helyett a **Default Tool** áll, amely a lemez legfontosabb programját tartalmazza, de nem kötelező megadni. Ez általánosságban is elmondható, az info csak tájékoztató, semmi sem kötelező benne.

A Project típusú infóban a Comment és a Tool Types között van egy **Default Tool** nevű sáv is. Ennek megértéséhez előbb tisztázni kell a Project fogalmát. A project egy olyan fájl, amit a felhasználó hozhat létre valamilyen program segítségével, például egy szöveg, zene, stb. A project tehát nem program; ahhoz, hogy használni tudjuk, szükség van általában az őt létrehozó programra, vagy egy megjelenítőre. Ebben a sorban annak a programnak a megnevezése áll, amely a tervet meg kell hogy jelenítse. Az ikonra történő dupla klikkeléskor ez a program indul el, majd ebbe töltődik be az ikonhoz tartozó terv.

Térjünk most vissza a Setmap infojához. A Comment sorába írd le: Német billentyűzet, majd klikkelj a Tool Types sor Add kapcsolójára. Ez létrehoz egy új bejegyzést ebben a sorban. Több ilyen bejegyzés is lehet, ezek között a le-fel nyilakkal lehet választani. A már létező módosításához elég beleklikkelni a sorába. A törléshez a Del kapcsoló használható.

Add most a Tool Types-nek azt, hogy **KEYMAP=d**, az idézőjeleket nem kell beírni, de a keymap-ot nagy betűvel írd. Mentsd ki a beállítást a Savé kapcsolóval. Ha ezután elindítod a **programot, az betölti a német ssab-v-ányú billentyűdet térképet**. Természetesen csak akkor, ha ez megvan a devs:keymaps-ban.

Ha felváltva többféle meghajtót akarsz használni, duplázd meg a Setmap-ot, mindegyikbe írd be egyet a kívánt billentyűzetek közül és nevezd át őket olyan névre, ami emlékeztet, például "Setnap Deutsch".

A következő program a **Init-Printer**. Ezt akkor kell használni, ha a nyomtatót később kapcsoltad be, mint az Amigát. A program megpróbálja a Preferencesben beállított paraméterekkel a rendszerbe illeszteni a nyomtatót és betölti a beállított nyomtató driwert.

12. Utilities tartalomjegyzék



Ha idáig eljutottál, nyisd ki az **Utilities** fiókot. Ebben található a legtöbb ikon. Ezek mind hasznos segédprogramokat jelképeznek. Az igazsághoz tartozik, hogy ezek csak kezdetben lesznek fontosak, később mindegyik helyett beszerezhetsz jobb, nagyobb tudású, használhatóbb változatot. A géppel való megismerkedéshez viszont elegendők.

A **Notepad** egy egyszerű szövegszerkesztő program. Segítségével szövegfájlok készíthetők. A program a szöveget úgynevezett **ASCII** formátumban menti ki, tehát minden olyan programmal olvasható, amely ezt ismeri. A legtöbb szövegszerkesztő ilyen. Sőt ha file konverterrel átalakítod a fájlt és kimented egy PC - lemezre, még IBM kompatibilis gépen is feldolgozható a szöveg. A legismertebb ilyen fájl konverter a **DOS2DOS**, ami a fájl átalakítását és PC-s lemezre írását egyaránt elvégzni, akár oda-vissza is.

Az átalakítás nagyon egyszerű. Az Amiga a sor végének jelzésére egy ASCII 10-es és egy 13-as kódú karaktert használ, a PC csak a 13-ast. Ezt a jelet kell tehát minden sor végéről törölnie, vagy fordított esetben be kell szúrnia a konverternek.

Igaz, hogy ez a szövegszerkesztő nem tartozik a világ nagyjai közé, mégis érdemes kicsit részletesebben is foglalkozni vele, mert a funkciói a komolyabb szerkesztő programokban is megta-

A Workbench

lálhatóak, ezek ismeretével bármely szövegszerkesztővel könnyebben elboldogulhatsz.

A program betöltődése után az ablak bal felső sarkában egy függőleges vonalat találsz. Ez a kurzor, ennek a helyén jelenik meg az éppen lenyomott billentyűhöz tartozó karakter. A karakter megjelenése után a kurzor egy hellyel jobbra lép. Ha kiért az ablak jobb szélére, a következő sor elejére áll. Amikor a Formát menü **Word Wrap** pontja be van kapcsolva, azaz előtte megjelenik egy kis pipa jel, a sor végére ért kurzor átvizsi a be nem fejezett szót a következő sor elejére. Ha ez a pont ki van kapcsolva, a szó kettészakadhat.

A begépelte szövegben a kurzor nyilakkal lehet mozogni. Ha a kurzor után is áll karakter és ekkor gépelsz be valamit, az beszűrődik, maga előtt tolja az egész szöveget.

Ha a kurzor nyilakkal együtt lenyomod a Shift billentyűt is, a kurzor a sor elejére, végére, lap tetejére, aljára ugrik. Ha szövegben valahová klikkelsz a bal gombbal, a kurzor oda ugrik.

Az ablak bal alsó sarkában lévő számárfüllel lehet előre lapozni. Az aktuális lapszám az ablak jobb felső sarkában található. Erre kattintva visszafelé lapozhatsz.

Karaktereket törölni a **Backspace** és **Del** gombokkal lehet.

Ennyi előzetes után nézzük át a Notepad menüpontját. Természetesen ezeket a menüket csak akkor tudod elérni, ha a Notepad az aktív. Némely menüpont a billentyűzetről is elérhető a jobb Amiga gomb és a menüben megadott billentyű együttes lenyomásával.

Project menü

New - Új szöveg létrehozása. Az előzőleg szerkesztett törlődik a programból, tehát előtte mentsd ki!

Open - Már meglévő text fájl megnyitása, további szerkesztése. A feltűnő kis ablakba írd be a fájl elérési útvonalát és nevét, majd klikkelj az OK-ra. A Cancel nem nyitja meg a fájlt, vissztérsz a szerkesztőbe. Ha megnyitsz egy fájlt, az előzőleg szerkesztett törlődik az editorból.

Savé - Ha a fájlnek már van neve, például létezőt nyitottál meg, vagy már kimentted, a fájl nevének és elérési útvonalának megkérdezése nélkül, a már ismert néven menti ki a szerkesztőben lévő szöveget.

Savé As - Annyiban különbözik az előzőtől, hogy mielőtt kimentené a szöveget, megkérdezi hová és milyen néven írja a fájlt. A text első létrehozásakor mindenképpen ezt kell használni sőt ilyenkor az egyszerűbb Savé hatására is ez hajtódik végre.

Print - Kinyomatja a szerkesztőben lévő dokumentumot. A menünek négy alpontja van, ezekkel választhatjuk meg, hogy a nyomtatás hogyan történjen.

Autó Size - körülbelül a Notepad ablakának a szélességével nyomtat

Small - a papír negyed szélességében nyomtat

Médium - a papír fél szélességében nyomtat

Large - a nyomtató a papír teljes szélességét kihasználja.

Utóbbi három opció csak akkor hatásos, ha a Print As-ról a Draft van bekapcsolva.

Print As - A menüpontnak három alpontja van, amikkel a nyomtatás paramétereit lehet beállítani. A beállítás nem nyomtat automatikusan, erre a Print-et kell használni.

Graphic - A jegyzetömb tartalmát grafikusán nyomtatja ki.

Minden oldal úgy fog kinézni, ahogy az a monitoron megjelent, sőt, még a Notepad ablaka is kinyomtatódik.

Draft - Csak a szöveg nyomtatódik ki a printer karakterkészletével. Nem veszi figyelembe a Font és Style beállításait.

E két alpontból csak az egyik lehet egyszerre bekapcsolva.

A következő pont kombinálható az előzőekkel.

From Feed - hatására a nyomtató minden lap elején kinyomtat néhány üres sort.

Read Fonts - Hatására a Notepad listát készít a boot lemezen lévő fontokról, ezek neveit pedig megjeleníti a Fonts menüben. Csak akkor kell használni, ha a szöveget előzőleg fontok nélkül mentetted el.

Többször is említettem a Font-okat. Ezek a betűkészletek adatait tartalmazzák. Minden font tartalmaz egy komplett betű és írásjel készletet, egy méretben. Mivel az Amiga mindig grafikus módban működik, a fontkészlet definíciója nélkülözhetetlen a képernyőre történő íráshoz. A fájl minden jelhez külön tartalmaz egy leírást, amely meghatározza az adott jel pontjait. A font magasságát **pixelben** szokás megadni. Ezek **bitmap fontok**, ami azt jelenti, hogy a karaktereket alkotó minden pixel helyzete pontosan meg van határozva.

Az Amiga a fontfájlokat általában a boot lemez fonts könyvtárában tárolja, amit hozzárendel a Fonts: logikai egységnevéhez. Két fontkészlet a ROM-ban is található, a **Topaz/9** és **Topaz/8**, alaphelyzetben ezeket használja a gép. A szám a font méretét jelenti. A Topaz/9 a 60 karakteres módban, a Topaz/8-at pedig 80 karakteres módban használja az Amiga (lásd a Preferences-nél a 37. oldalon). A Notepad Fonts menüjében a Fonts tartalomjegyzékben lévő fontkészletek nevei jelentik az egyes menüpontokat. Az alpontokat a választható méretek teszik ki. Ezekkel a fontokkal lehet előállítani a dokumentumot.

Fontos megjegyezni, hogy a választott fontnak megfelelő íráskép csak **Graphic** módban (lásd Print As) jelenik meg nyomtatón. **Draft** módban a nyomtató a saját fontkészletét használja.

Ha a Fonts menüből kiválasztasz egy új fonttípust, az az egész dokumentumra érvényes lesz.

A **Style** menüben meghatározhatod a szöveg stílusát. Ez lehet:

Plain - normál

Italic - dőlt betűs

Bold - vastag betűs

Underline - aláhúzott

Utóbbi három kombinálható.

A Nőne megszünteti a stílusjelzéseket.

Az ismertetésből kimaradt egy menü, az **Edit**. Ebben szerkesztő parancsok kaptak helyet. Nézzük át őket:

Mark Place - Bár nem ez az első menüpont, mégis elsőnek kell megemlíteni. Hatására a kurzor jelenlegi pozíciójába egy láthatatlan jelet tesz. Ha ezután elmozgatód a kurzort egy másik pozícióba, majd ismét használod a parancsot, a két jel közötti terület "kijelölődik". Ezt a háttérszínének megváltozása jelzi. A kijelölt terület blokk-nak nevezzük. A blokk az alapja a menü néhány pontjának.

Cancel - Az éppen futó művelet elvetése, hatástalanítása.

Cut - Kivágja a blokkot és egy átmeneti tárolóba helyezi. Ezt az átmeneti tárolót Clipboard-nak nevezzük. Itt egyszerre csak egy kivágott blokk lehet, tehát ha a következőben használod a parancsot, az törli a Clipboard előző tartalmát.

Paste - A Clipboard-ban lévő blokkot a kurzortól kezdve bemásolja a szövegbe. A blokk a Clipboard-ban is megmarad.

Copy - A kijelölt blokkot úgy másolja a szövegből a Clipboard-ba, hogy közben az editorban is megmarad.

Az Edit menü további pontjai más jellegű feladatokhoz valók ezt jelzi az is, hogy kissé távolabb helyezkednek el az előző csoporttól.

Find - A kérdezőjébe a Find után beírt karaktersorozat próbálja megkeresni a szövegben. A Next-re klikkelve a kurzortól előre, a Prev-re klikkelve pedig a kurzortól visszafelé. Ha nem talált ilyen szöveget, a kurzor nem mozdul. Ha talált, a kurzor oda ugrik.

Find Next - Tovább folytatja az előző pontban megadott szöveg keresését, előre.

Find Prev - Tovább folytatja a keresést visszafelé.

Replace - Ha a Find kérdezőjében a Replace után is adtál meg szöveget, e parancs hatására a megtalált szövegrészt erre az új szövegre cserélheted ki.

Még egy menü maradt a Notepadban, ez a Formát, amelyben a dokumentum néhány formai jellemzőjét adhatod meg.

A Workbench

Paper Color - A szöveg háttér színe. A négy Workbench alapszínből választhatsz. Ha nem teszik egyik szín sem, a Preferences editorban változtathatod meg azokat (lásd a 37. oldalon).

Pen Color - A karakterek színe. Az előbb elmondottak érvényesek itt is.

Word Wrap - Volt már róla szó. Ha be van kapcsolva, a sorba be nem férő szót átviszi a következő sor elejére.

Global Fonts - Engedélyezi a különböző fontok használatát.

Remove Fonts - eltávolítja a szövegből a font attribútumokat.

Remove Styles - eltávolítja a szövegből a stílus attribútumokat.

A fiók következő programja a **More**. Ez egy szövegfájl megjelenítő program. Megnézhetjük vele például a Notepad-dal készült dokumentumainkat. Ha önmagában indítod el, egy rövid infót ad, majd feltesz egy kérdést, hogy melyik fájlt akarod megnézni. Ha erre egy létező text fájl elérési útvonalát és nevét adod meg, megjeleníti azt. Ha csak egy enter-t nyomsz, kilép a programból.

Használatának másik módja, hogy a megjelenítendő fájl Infójában a már ismertetett módon a **Default Tool** sorában javítsd ki a Notepad-ot More-ra. A többit hagyd változatlanul, majd mentsd ki az infót. Ha most duplán kattintasz a dokumentum fájl ikonján, akkor előbb elindul a More, majd megjeleníti a kívánt szöveget.

Harmadik lehetősége az indításnak a multi szelekt, a már ismertetett módon.

A program elindulása és a szöveg megjelenése után a dokumentumban a **Space** megnyomásával képernyőoldalanként, a Return-nal soronként léphetünk előre. A **Back space** billentyű hatására egy képernyő oldallal visszafelé lehet lépni.

A programból a q billentyűvel, vagy a **CTRL+C** kombinációval lehet kilépni.

A Kézikönyv *xyí lőü*éé<5 *&y Help (e<sgí*eég) aWalcot koz &IS. Itt megtalálhatod a More irányító billentyűjének a leírását. Ezeket most röviden felsorolom: (a már említettek kivételével)

/szöveg - A megadott szöveg keresése a nagy betűk figyelembevételével.

. szöveg - A megadott szöveg keresése, a nagy betűk egyenértékűek a kicsikkel.

n - Tovább keres
CTRL+L - Újrarajzolja az ablakot.
< - Az első oldalra áll.
> - Az utolsó oldalra áll.
%n - A szöveg n%-ára áll.

A következő program a Clock. Ez egy órát rajzol a Workbench képernyőre, ami ugyan olyan ablakban jelenik meg, mint a többi program, tehát velük azonos módon elmozgatható, mérete megváltoztatható. Ha ez az ablak az aktív, használhatod az óra menüparancsait. Ezek:

Type menü:

Analóg - Mutatós óra
 Digital 1 - Digitális óra
 Digital 2 - Digitális óra

Mode menü

12 Hour - 12 órás kijelzés
 24 Hour - 24 órás kijelzés

Seconds menü

Seconds On - másodperckijelzés be
 Seconds Off - másodperckijelzés ki

Date menü

Date On - dátumkijelzés be
 Date Off - dátumkijelzés ki

Alarm menü

Set - Riasztás beállítása
 Alarm On - Riasztás engedélyezése
 Alarm Off - Riasztás kikapcsolása

A ClockPtr hasonló az előzőhöz, szintén egy óra, de ez a Workbench mutatójában jelenik meg. A nyíl egy vízszintesen

A Workbench

fekvő téglalapra változik, amiben egy digitális óra mutatja az időt.

A mutató óra csak a **Workbench screen-en** működik, ugyanis minden programnak saját mutatója lehet, annak ablakába klikkelve a mutató felveszi a program által meghatározott alakot.

A **Say** egy beszédprogram. Két ablakot nyit. Az egyik a **Phoneme Window**, itt jelenik meg a kimondott szöveg fonéma kódja. A másik az **Input Window**, ahol a kimondandó szöveget tápláld be, return-nal lezárva.

A **Calculator** egy egyszerű, alapl műveletes számológép. A betáplálást a gombjaira klikkelve végezheted el.

A **CMD** egy olyan program, amely átírányítja a nyomtató sorozatot egy fájlba. Akkor hasznos, ha nincs nyomtató, ahol pedig van nyomtató és ki akarod printelni a fájlt, nincs meg a nyomtatást végző program. Ebben az esetben elég az átírányítás után keletkező fájlt átvinni, majd azt a **Shell Copy** parancsát használva kimásolni a **PRT:** portra (lásd a 90. oldalon). Ha a nyomtatáshoz **ANSI** printer meghajtó van installálva, olyan programmal is tudsz text fájlt nyomtatni, amely nem **ASCII** formában tárolja a szöveget. Ezzel a driverrel fájlba nyomtatva **ASCII** fájlt kapsz.

A **Graphic Dump** kinyomtatja a képernyő tartalmát a géphez kapcsolt nyomtatón, vagy az előző prancsral meghatározott fájlba.

A **Printfiles** a **multi sselect-tel** kijelölt fájlt nyomtatja ki. Egyszerre több ikont is ki lehet jelölni és együtt átadni őket a programnak, ekkor az mindegyiket kinyomtatja.

Az **Install Printerrel** az **Extrás** lemezről másolhatsz át nyomtató meghajtókat a **Devs:printers** könyvtárba. Indítás után számba veszi az átmásolható drivereket majd megkérdezi melyiket tegye át Egyszerűbb azonban, ha erre a célra valamilyen fájl-másoló programot (Pl. Disk-Master, File Master, Directori Opus stb...) használsz.

Ezzel végére értünk a Workbench lemezen található programokkáival való ismerkedésünknek. Van azonban még néhány érdekesség. Ebből az egyik, a lemez ablakában található kuka formájú, **Traschan** nevű ikon. Mint már említettem, ez egy különleges tartalomjegyzék. Ugyanúgy helyezhetsz bele fájlokat, mint más fiókba, de ide általában azokat a dolgokat tesszük, amikre már nincs szükség, de még nem akarjuk letörölni őket a lemezről (a törlésről még egyébként sem volt szó), hátha még jók lesznek valamire. Innen bármikor egyszerűen vissza lehet venni a fájlokat. Ha végleg nincs rájuk szükség, a Workbench Disk menüjének Emty Trashcan pontját kiválasztva, az összes Trashcan-ban lévő fájl törlődik.

Ha már a törlésnél tartunk, nézzük meg hogyan lehet azonnal letörölni egy fájlt. Jelöld ki a törlendő tárgy ikonját vagy a Shift gomb lenyomásával egyszerre többet is, majd a Workbench menü Discard parancsát válaszd ki. Ez jelenti a törlést. Mivel ez a művelet véglegesen törli a fájlt, óvatosan kell vele bánni, rá is kérdez a művelet végrehajtása előtt. Ha a bal oldali "Ok to Discard" választ adod, a törlés megtörténik. A jobb oldali gombbal visszavonhatod a parancsot.

A Workbench és Disk menün kívül van még egy harmadik menüje is a Workbench screen-nek, a **Special**. Ebben néhány különleges parancs található. Nézzük sorba őket:

Clean Up - A kiválasztott ablak ikonjait elrendezi.

Last Error - Megismétli az utolsó hibaüzenetet. A Workbench hibaüzenetei a fejléc sorban jelennek meg, az "üzenet" a hiba rövid leírása, a "kód" pedig a hiba kódszáma. Az egyes

A Workbench

kódszámokhoz tartozó hibák és azok részletes leírása a függelékben található.

Redraw - Néha előfordul, hogy egy kérdező, vagy ablak elrontja a képernyő rendjét. Ez a parancs megismétli a kirajzolást

Snapshot - Rögzíti a kijelölt ikon helyzetét. Következőleg újból itt jelenik meg az ikon. A Shift használatával egyszerre több ikon kijelölése esetén is hatásos. Nem csak az ikonok, hanem az ablakok helyzete, mérete is beállítható.

Version - A fejléc sorba kiírja a használt Kickstart és Workbench verziószámát

Miután a Workbench-csel való ismerkedést befejeztük, nézzük meg, mi található a másik, az **Extrás** lemezen. Vedd ki a Workbench lemezt, és tedd be helyette az Extras-t. Rövid idő múlva megjelenik ennek is az ikonja, de megmaradt a Workbench-é is. A boot lemez képe soha nem tűnik el, ellenben ha például az Extras-t kiveszed és nem marad róla kinyitott ablak vagy futó program, az ikonja is eltűnik.

Nyisd ki az **Extrást**. Benne néhány ismerős és új ikon tűnik fel. Az ismerősök közül való az **FD1.3** fiók alakú ikonja. Ez most is tartalomjegyzéket reprezentál, kinyitva azonban csak egy feiratot találsz, miszerint ebben a tartalomjegyzékben FD fájlok vannak.

Valójában itt a Workbench lemez libs tartalomjegyzékében lévő library kiterjesztésű fájlok leírása található. Most nem ismertetem őket, ez csak a programozók számára fontos. Számukra meg fog jelenni egy kézikönyv, amely részletesen ismerteti az Amiga könyvtárrutinjait és azok használatát.

A **Devs** tartalomjegyzékben látszólag semmi sincs, de ez nem így van, csak nincs ikonjuk. Itt a **Workbench** alá installálható printermeghajtók és billentyűzetkiosztások találhatóak. A megfelelőt át kell másolni a boot lemez megfelelő alkönyvtárába ahhoz, hogy használhatóak legyenek.

A Fonts-ban újabb fontok találhatóak, ezeket is át lehet másolni a boot lemez fonts könyvtárába. **Hogy** miért nem ott vannak ere-

detileg? Nos, azért mert a Workbench lemez így is kis híján tele van, nem férnének rá még ezek is. Mindenkinek el kell döntenie, mely billentyűzetkiosztást, nyomtatómeghajtót, stb. akarja használni, és azt kell átmásolnia a Workbench lemezre (természetesen a munkamásolatra!).

Itt is megtalálható a Trashcan ikon, használata a már ismertezett.

Nézzük meg mit találunk a Tools fiókban, nyisd ki. Várakozásunknak megfelelően, segédprogramok töltik ki ezt a könyvtárat. Nézzük át őket. A részletes ismertetéstől hely hiányában le kell mondani.

A Memacs egy szövegszerkesztő program hasonlóan a Notepad-hoz, de ez annál jóval többre képes, több funkciója van. A használatának elve viszont azonos a Notepadéval, kis gyakorlással, kísérletezéssel kezelése elsajátítható.

A Fed egy fontszerkesztő. Segítségével fontkészleteket hozhatunk létre, vagy meglévőket módosíthatunk. Jó szolgálatot tesz a kétvesszős hosszú magánhangzók ékezeeteinek a kijavításánál. A program kezelése igen egyszerű.

A project menü

- New - új fontkészlet létrehozása
- Open - Meglévő font módosítása. A módosítandó fontnak a fonts:-ban kell lenni.
- Savé - Fontkészlet kimentése eredeti néven.
- Savé As - Fontkészlet kimentése más néven
- About - Információ a programról.
- Quit -Kilépés.

Edit menü

- Make Italic- Dőlt fontok készítése
- Make Bold - Vastagított fontok készítése
- Make Underlined - Aláhúzott fontok készítése
- Copy to - Karakter másolása
- Erase - Karakter törlése
- AllRight - Minden karakter jobbra

A Workbench

Ali Left - Minden karakter balra

Ali Up - Minden karakter fel

Ali Down - Minden karakter le

Attributes menü

Font Type

Fixewidth - Rögzített szélességű karakterek

Proportional - Minden betűnek saját szélességértéke lehet

Font Style - Az éppen szerkesztett fontok stílusa.

A képernyő bal oldalán van a szerkesztőterület, a jobb oldalon a karaktertáblázat egy része látható. A táblázatban látható a karakter jelenlegi alakja, standard alakja és **ASCII** kódja.

E két rész közötti jelek körül a felső két nyíl jobbra-balra lépteti az adott karaktert a szerkesztőben, az alatta lévő két nyíl ugyan ezt teszi függőlegesen. Az "R" invertálja a karaktert, az alsó két nyíl pedig tükrözi.

A képernyő alján bal oldalt lévő kapcsolóval, aminek LoRes a felirata, azt tudjuk váltani, hogy a karakter **Lores** vagy **Hires** képernyőn látható formájában jelenjen-e meg. A Grid a szerkesztési segédrácsot kapcsolja ki-be.

A Zoom-mal közelebb-távolabb lehet vinni a karakter képét. A **Scroll** akkor használható, ha olyan közelebb nézzük a fontot, hogy egyszerre már nem látható minden pontja. Ekkor a Scroll nyilaival gördíteni lehet a látható részt

A Lo megjelenítendő első, a Hi pedig az utolsó karakter kódja. A **Kern, Space, Width** a karakter alávágási, betűköz, szélesség adatai. A **Bline** a karakter bázissora, ezt egy zöld vonal jelzi a szerkesztőben. Ez alá nyúlnak le az olyan betűk, mint például a kis p. Az utolsó két érték a karakter szélessége és magassága pixelben.

A következő program a FreeMap, amely a gép memóriakihasználtságáról ad grafikus tájékoztatást.

A PerlMon folyamatosan kijelzi a processzor kihasználtságát

Az IconEd programmal ikonokat lehet rajzolni és a fájlokhoz rendelni.

Az IkonMerge két ikont fűz össze harmadikká.

A Palette segítségével a képernyő színeit lehet beállítani.

A KeyTo2000 bemutatja a jelenleg érvényes billentyűzet-kiosztást.

A Terminál a telefonmodemet kezeli, ha van ilyen a géphez csatlakoztatva. A modem segítségével két gép köthető össze telefonvonalon, vagy nullmodemen keresztül.

Nézzük most meg, mit találunk még az **Extrás** lemezen.

A Hinweise egy text fájl, a **Notepad**dal is meg lehet nézni, információát ad a lemezen lévő tárgyakról.

Az AmigaBasic az Amiga **Basic interpretere**. Odáig, hogy Basic, gondolom mindenki értette, de mit jelent az interpreter? Magyarul értelmező. Ez egy program, amely beolvassa a Basic programot, ami tulajdonképpen egy text fájl, az utasítások sorozata, majd soronként értelmezi a program utasításait és meghatározza a gép számára a teendőket. Az értelmezést a Basic program futása alatt végzi. Belátható tehát, ez a folyamatos értelmezés a gép kapacitásának nagy részét leköti. A Basic program a nyelv interpretere nélkül nem is futtatható. Előnye viszont az egyszerű kezelhetőség, a programlista mindig rendelkezésre áll, könnyű javítani, fejleszteni a programot, a hiba helye gyorsan behatárolható. Főleg kezdők használják ezt a módszert.

Ha ez az egyik módszer, természetes, hogy van másik is. Ez a fordító (Compiler). Ez úgy működik, hogy a programlistát lefordítja a gép nyelvére, gépi kódra. A lefordított programot ezután önállóan futtathatod. Előnye tehát a gyorsaság és a kisebb terjedelem, nem kell ugyanis futtatás közben az értelmező. Hátránya ennek a módszernek, hogy a program mellett nem szerepel a prog-

A Workbench

ramlista, ezért nem lehet egyszerűen fejleszteni. Ha megvan a lista, bármilyen kis módosítás után újra le kell azt fordítani. Mivel nem közvetlenül a programlista fűt, nehéz a hibák behatárolása.

A fordítós nyelvek csoportjába tartozik például az Assembler, C, Pascal. Megjegyzendő, hogy némelyik Basic interpreterhez is kapható Compiler. Ilyen a Bliíz Basic vagy a GFA Basic.

A lemez Basic Demos fiókjában Basic mintaprogramok találhatóak. Kétféle módon használhatod őket. Az egyik, hogy duplán klikkelsz rájuk, ennek hatására elindul az AmigaBasic interpreter, majd betöltődik a választott program és elindul.

A másik eljárás, hogy elindítod az AmigaBasic-et és betöltőd, majd elindítod a programot.

A Basic nyelv kifejezetten kezdők számára van kitalálva, nem túl hatékony, de egyszerű és könnyen elsajátítható programnyelv. Hogy mégsem foglalkozunk vele részletesebben, ennek három oka van. Egyik, hogy a téma megtöltene egy egész könyvet. A másik, és ez a lényegesebb, hogy az AmigaBasic egy elavult, alacsony színvonalú programnyelv, bár megjelenésekor, a hobbi számítógépek legjobb Basic-je volt. Azok számára, akik még nem járatosak a programozásban és egy Basic-szerű nyelvre van szükségük, ajánlom az AMOS-t. Ez egy olyan programnyelv, amely hasonlít a Basic-re (nevezik AMOS Basic-nek is), de jóval hatékonyabb annál. Akár összetett felhasználói programok írására is alkalmas. Az AMOS programok interpreterben fejleszthetők, és ha készek, az AMOS Compiler segítségével gépi kódra fordíthatók.

Az AMOS programozási nyelvről egyébként már jelent meg egy könyv, az Aurum DTP Stúdió gondozásában. Címe: AMOS a Kreátor.

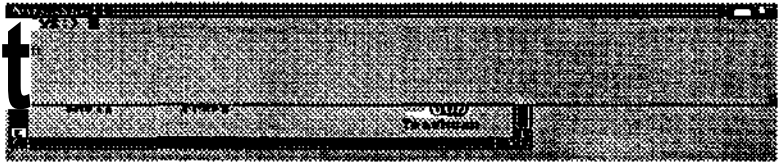
A harmadik ok, hogy mellőzzük a leírást, az, hogy maga a Commodore is belátta, mára elavult a Basic, az újabb gépekhez már nem is mellékelik.

C.ADOS

Maradt még a **Workbench** ablakában egy ikon, a **Shell**. Ez egy **DOS promptot** hoz elő, ahol **DOS** parancsokat lehet kiadni. Hasonló *hozzá*, de kisebb kényelmi szolgáltatásokat nyújt a System fiókban lévő **CLI**.

Mi is az a **DOS**? A **Disk Operating System**, lemezműködtető rendszer rövidítése. Ez különböző parancsokból, azaz önálló kis programokból áll, amelyek általában a lemezen, a C tartalomjegyzékben található. A parancsokkal fájlok másolása, törlése, programok indítása, dátum beállítása és még rengeteg dolog végezhető el. Ebben a fejezetben áttekintjük a **Shell** működését és az **AmigaDOS** parancsait. A **DOS** alatt most az az 1.3 verziószámú operációs rendszert értjük. A 2.0-s rendszer külön fejezetben kerül ismertetésre.

Hogyan adhatunk ki egy DOS parancsot? Először is el kell indítani a Shell-t, vagy CLI-t, az ikonjára történő dupla klikkeléssel. Megjegyzendő, hogy más úton is el lehet érni a CLI-t. Az egyik, hogy a Workbench lemez bootolásakor lenyomod a CTRL+D billentyűket. Erre megáll a töltés egy ún. "**Break CLI**"-t kapsz, ahonnan már kiadhatod a DOS parancsokat. A Break CLI felirat után egy ">" jelet láatsz, ez a **prompt**. Ide várja a rendszer a parancs betáplálását. Ha nincs prompt, abban a DOS ablakban nem lehet parancsot kiadni. A Shell-el akartunk megismerkedni, maradjunk tehát az első megoldásnál.



Shell ablak

Mivel a Shell-ben minden elvégezhető, ami a CLI-ben, sőt több is, inkább ezt részesítsük előnyben. Mit tegyen az, aki az 1.2 verziójú Workbenchet használja, ott ugyanis nincs Shell? Használjon Ő is 1.3-as Workbenchet, még ha 1.2-es kickstartú gépe van is.

Amint beléptünk a Shell-be, ami egy szokásos ablak, kivéve, hogy nincs ablakzáró gadgetje, feltűnik egy rövid felirat, végén a ">" jellel. Ez a prompt, ide kell begépelni a parancsokat, majd a Return lenyomásával végrehajtatni. Ha elrontottad a betáplálást, helytelenül írod be a parancsot, a gép nem érti meg mit akarsz, ezt az Unknown Command üzenettel jelzi, újra be kell gépelned a prompt után azt.

Közben máris választ adtam arra a kérdésre, hogy miként kell kiadni egy DOS parancsot, egyszerűen be kell gépelni a nevét, majd szóközzel elválasztva a paramétereit, ezután az egészet Return-nal lezárni. Egy sorban csak egy parancsot lehet kiadni. A kisbetűk-nagybetűk között a DOS nem tesz különbséget. Amíg ennek a feldolgozása tart, eltűnik a prompt, a gép nem fogadja a további parancsaidat. Ennek ellenére begépelheted azokat, mert egy közbelső tárolóban megőrződnek, az előző parancs feldolgozása és a prompt visszatérte után az így betápláltak is végrehajthatódnak.

Az így bevitt parancsok nem multitaszkban futnak, erre utal, hogy eltűnik a prompt (kivéve, ha a program automatikusan áttér multitaszkra, ekkor visszaadja a promptot). Másik programot úgy indíthatsz, hogy nyitsz neki egy új Shell-t, vagy eleve közlöd, hogy multitaszkos indítást kérsz. Ezt úgy éred el, hogy a parancs elé beírod:

Run

Hatására a program automatikusan nyit magának egy ablakot, abban hajtódik végre. Az elindított programot befejezve, bezáródik annak ablaka.

Mi van ha rosszul írsz be valamit? A Shell előnye a CLI -vei szemben (többek között) a parancssor-szerkesztő. Az éppen beírás

alatt lévő sorban a kurzor nyilakkal mozogni is tudsz, szemben a CLI-vel, ahol visszafelé lépni csak törléssel lehet. Ráadásul a beírt parancsokat meg is jegyzi a Shell, a le-fel nyilakkal választhatsz közülük.

Ha a Shift-el együtt használod a kurzor nyilakat, a sor elejére - végére, vagy a parancslista elejére és végére ugorhatsz.

Mivel a Shell-lel gyakran fogunk találkozni meg kell ismerkedni néhány vele kapcsolatos dologgal. A Shell-hez szükség van az L: logikai egységben a Newcon-handler és Shell-seg fájlokra, ezek közül a Shell-seg-et rezidenssé kell tenni a

Resident L:Shell-seg CLI

parancssal, amely leírását később olvashatod, (lásd a 71. oldalon)

A Shell-nek van egy belső parancsa, amely nem a C:-ben van, hanem a Shell-seg ismeri fel. Ez a parancs az Alias, amellyel makrót hozhatunk létre, vagy kérdezhetjük le a meglévőket.

A makró egy parancssorozat rövidítése, sokszor ismétlődő, hosszú parancsokat célszerű rövidíteni vele. A rövidítés módja:

Alias rövidítés parancs

Az Alias-t magában használva a már létrehozott makrók listáját kapod meg. A Shell elindítása után érdemes megnézni az eredeti makrókat, ezeket az S: logikai egységben lévő Shell-startup nevű batch fájl hozza létre, amely a Shell minden egyes indításakor végrehajtódik.

A felvetődött ismeretlen dolgokat, mint például a logikai egység, nemsokára megmagyarázzuk, addig is fogadjuk el így ahogy vannak.

Van a parancsadásnak más módja is. Egy szövegszerkesztővel, például a Notepad-dal sorban leírhatod a végrehajtandó parancsokat, majd ezt kimentve, át lehet adni végrehajtásra. Ebben az esetben a Shell sorba veszi a fájlban leírt parancsokat és egyenént

végrehajtja azokat. Az ilyen fájlokat batch fájloknak nevezzük, az előbb említettük meg az egyiket.

Legnevezetesebb képviselőjük a sokat emlegetett Startup-sequence, amely a gép bootolásakor hajtódik végre. A Batch fájlokkal később még foglalkozunk, most azonban nézzünk szét a lemezen, add ki a következő parancsot:

`Dir`

Hatására megjelenik az aktuális tartalomjegyzék, jelen esetben a Workbench diszk tartalomjegyzéke. Jóval több bejegyzéssel találkozunk, mint a Workbench felületen lévő ikonok száma. Ennek oka, mint már említettem az, hogy csak azoknak a fájloknak van ikonja, amelyekhez tartozik egy "fájlnév.info" fájl is, ami nem kötelező.

A listában a fájlokat a tartalomjegyzéktől a nevük után álló **(dir)** felirat jelzi.

Mielőtt tovább mennénk, tegyünk egy kis kitérőt és tisztázzuk, mi az: aktuális tartalomjegyzék. Mindig ki van jelölve egy lemez vagy egy tartalomjegyzék, amely az aktuális, ha mást nem adunk meg, ebben hajtódnak végre a parancsok, mint jelen esetben a Dir. A Shell promptjában általában látható is az aktuális tartalomjegyzék neve és útvonala.

Bootolás során a boot lemez lesz az aktuális, erre utal a Sys: prompt. A Sys: egy logikai egység, amely automatikusan hozzárendelődik a boot lemezhez, de ez megváltoztatható.

Az aktuális tartalomjegyzéket könnyedén meg lehet változtatni a CD parancs segítségével, amely paramétere az új elérési út lehet. A CD parancs paraméter nélkül használva kiírja a jelenleg aktuális tartalomjegyzék nevét és útvonalát. Ez CLI-ben hasznos, vagy ha megváltoztattuk a Shell promptot.

A kijelölendő tartalomjegyzék útvonalát megadhatjuk abszolút vagy relatív módon. Az előbbi azt jelenti, hogy a teljes elérési útvonalat definiálni kell. Például:

```
CD DFO:Devs/keymaps
```

A másik, a relatív, amikor a jelenlegi tartalomjegyzékből származtatjuk az újat. Például ha a DFO: az aktuális, elég ennyi:

CD Devs/keymaps „_ „

Ha az új tartalomjegyzék nem a jelenlegiből származik, akkor azt abszolút módon kell megadni.

A könyvtárstruktúrán felfelé lépni két módon lehet. Vagy abszolút módon megadod a feljebb lévő tartalomjegyzéket, vagy használsz a "Parent" jelet, ami a szülő, eggyel fentebbi könyvtárra lép, így:

CD /

Ha több tartalomjegyzék mélységben vagy és a gyökérbe akarsz

fellépni, ehhez ennyi is elég:

CD //

A parent és a gyökér jel nem csak önmagában használható, hanem útmeghatározásban is. Például:

CD DFO:C

majd

CD /devs/keymaps

Innen újból a C-be:

CD//C

Térjünk vissza a **Dir** parancshoz, nézzük meg mi van a C-ben. Ehhez nem muszáj a C-t aktuálissá tenni (bár megteheted), elég ennyi:

DirC

Mint már említettem, a C tartalomjegyzékben a DOS parancsai találhatóak, köztük a CD és a **Dir**. Itt sok fájl van, előfordul, hogy nem is férnek ki egyszerre a képernyőre. A Space lenyomásával szerencsére bármikor meg lehet állítani a listát. A folytatáshoz a

Backspace billentyűt kell használni. Ha úgy akarsz megállítani a listázást, hogy nem is vagy kíváncsi a többire, nyomd meg a CTRL és C billentyűket egyszerre.

A **Dir** parancsnak van néhány opciója, amivel befolyásolni lehet a működést. Az egyik opció az "**opt a**", minek hatására nemcsak a listázott tartalomjegyzékben lévő altartalomjegyzékek nevei, hanem kicsit beljebb kezdve az azokban lévő fájlok és újabb altartalomjegyzékek (és azok tartalma!) is megjelennek. Például:

Dir devs opt a

A következő opció az "opt i". Ez interaktív tartalomjegyzék listát ad. Minden fájl vagy directory név megjelenítése után egy kérdőjel jelenik meg és a listázás addig nem folytatódik, amíg valamilyen választ nem adsz, Returnnal lezárva. Ha csak egy sima Returnt nyomsz, kiíródik a következő név. Ezen kívül a következő lehetséges válaszok vannak:

E - Ha a kiírt név tartalomjegyzéké, ezzel beléphetsz abba, a lista ennek a tartalmával folytatódik.

B - Ha belptél egy tartalomjegyzékbe, ezzel léphetsz vissza az előző szintre, az összes bejegyzés kiírása előtt.

T - A fájl tartalma szövegesen megjelenik a képernyőn. Csak text fájlnál van igazán értelme, bár minden kiírható.

Del - Töröld vele az éppen megjelenített fájlt, vagy tartalomjegyzéket (utóbbit csak akkor, ha az üres).

Cvagy

Com - A listázás abbamarad, megjelenik a "Command?" felirat, ahol egy tetszőleges DOS parancsot adhatsz ki. Ennek végrehajtódása után folytatódik a Dir parancs.

Q - Kilépés a parancsból, újból visszakapod a promptot.

Az egyes opciók kombinálhatók is. Például:

Dir opt ad

Dir opt di

A DOS parancsokban használni lehet **dzsóker** jeleket is. Két ilyen van, a ?, ami egy konkrét karaktert helyettesít és a #? ami egy tetszőleges részét a névnek. A dzsókereket tartalmazó meghatározásokat mintáknak, vagy patterneknek nevezzük.

* A Dir parancsnál ezeknek az elérési út és a fájlnev megadásánál van jelentősége. Például kíváncsi vagy a C könyvtár A betűvel kezdődő parancsaira:

* DirC/A#?

Ha azt akarod mely tárgyaknak van Workbench ikonja, írd be:

Dir #?. info

A következő parancs

Dir ??e??. info

Azokat az öt karakterből + .info-ból álló neveket listázza ki, amelyek harmadik betűje e. Ilyen a Prefs. info, és a Shell.info.

Van az Amiga DOS -nak egy másik parancsa is, amellyel a tartalomjegyzéket nézhetjük meg. Ez a List. Haználatának módja, a tartalomjegyzék megadása hasonló a Dir-nél megismertekhez, opciói azonban mások. A List jóval több adatot szolgáltat a fájlokról, mint a Dir, próbáld is ki mindjárt

List

A különbség szembetűnő. Alapesetben a következő információkat kapod a fájlokról és tartalomjegyzékekről:

méret (vagy dir jelzés) attribútumok dátum, idő megjegyzés.

A **méret** a fájl mérete bájtokban. Ha a bejegyzés egy tartalomjegyzéké, a **dir** felirat olvasható a helyén.

** Az Attribútumok oszlopban a fájlok és tartalomjegyzékek attribútumai találhatóak. Ezek egy részéről már volt szó, azonban itt szerepel négy újabb is (csak Shell-ben van értelmük, a CLI-ben nem használtak)

a- archív, a fájl már archiválva lett.

s - script fájl, shell-ben execute nélkül is végrehajtható batch fájl. Lásd később.

p - a fájl rezidenssé tehető, egyszerre többször is futtatható. Lásd később.

h - rejtett fájl (az 1.3-ban még nem működik)

r- a fájl olvasható

w - a fájl felülírható

e - a fájl végrehajtható (program)

d - a fájl törölhető

A dátum és az idő a fájl keletkezésére utal. Csak akkor van igazi jelentése, ha a gépben van belső óra, vagy ha minden bootolás után beállítod a rendszeridőt.

A megjegyzés egy max 78 karakteres szöveg, amelyet a **Filenote** paranccsal fűzhetünk a fájlhoz, (lásd a 129. oldalon).

Tanács a List-hez. Ha a listázandó elérési útvonal tartalomjegyzékkel végződik, tegyél a végére egy / jelet, hogy egyértelműsítsd a parancs számára, ez directory. Ha ezt elhagyod, könnyen összekeveri a nagyszámú opciók valamelyikével az útmeghatározást. Például:

UstC/

Ezek után lássuk a List opciót:

Dates - A dátum konkrétan kiíródik. Egyébként használja a parancs a Today és Friday neveket is. PL.: List DFO: dates

Upto - Csak a megadott dátumig készült fájlok és tartalomjegyzékek listázódnak ki. List Upto 18-jan-93

Since - A megadott dátum óta létrehozott fájlok és tartalomjegyzékek listái. List C/ Since 18-jan-93

Nodates - Nem jelenik meg sem a létrehozás dátuma, sem az időpontja. List devs/nodates

Quick - Csak a nevek jelennek meg. Gyakorlatilag azonos a Dir-rel.

To - **A lista nem a képernyőn jelenik meg, hanem a To után megadott fájlba íródik.** Ezután azt akár a Notepad-

dal, vagy a More-ral is bármikor meg lehet nézni. Pl.: List C/ To RAM:Lista

Sub - Csak azokat a neveket listázza, amelyek tartalmazzák a Sub után álló karaktersorozatot

Pat - Mintákat adhatunk meg, amelyek szerint listázni kell. Például: List DF0:P +? (.iff:.pic) Ez az iff vagy pic kiterjesztésű fájlok listáját adja.

P - Ugyan az, mint a Pat

Files - Csak a fájlok jelennek meg a listában

Dirs - Csak a tartalomjegyzékek jelennek meg a listában.

Nohead - Nincs fejléc és összegzés

Keys - A kezdőblokk száma is megjelenik. A fájl ebben a számú blokkban van bejegyezve a lemez tartalomjegyzékébe.

Block- A méret nem bájtokban, hanem blokkokban íródik ki.

Lformat - Lehetővé teszi a List eredményének formába öntését. Utána idézőjelbe szöveget írhatunk, amit a parancs ki-nyomtatott. A szövegbe a %S helyettesíti be a listázott fájl vagy tartalomjegyzék nevét, a %S%S pedig ugyanez, de elérési úttal együtt. Például:

```
List DFO: Lformat 'A%S teljes neve %S%S'
```

vagy

```
List DFO: RAM: lista Dirs Lformat "LIST %S%S"
```

majd

```
Execunte RAM: lista
```

A RAM:lista utasítja a parancsot, hogy ne a képernyőre, hanem a RAM:lista fájlba írja az eredményt, ebben az esetben batch fájlt hozva létre. Ezt akár a More-val is megnézheted ezután.

Az Execute magyarázatát lásd az oldalon (lásd a 108. oldalon)

Az opciók egymással kombinálhatók is. Például:

Ust DFO.C/ Nodates To RAM:lista

Miután felfegyverkeztünk e két tartalomjegyzék megtekintésére alkalmas paranccsal, kukkantsunk bele egy másik lemezbe is, mondjuk az Extras-ba. Tedd be ezt a lemezt a belső meghajtóba. Hogy megtaláljuk a közös hangot, még ha van se használd most a külső drive-ot. Amint megvan a lemezcseré, írd be:

Dir DFO:

Megjelenik egy kérdező, amely kéri, hogy tedd be a Workbench-et Tegyéél neki eleget Az igen választ képviselő **"Retry"-t** nem szükséges megnyomni, általában felismeri a gép a szükséges lemez jelenlétét. Miután ez megtörtént, rövid töltögetés után megjelenik a lemez tartalomjegyzéke.

Nem ezt vártad? Az bizony meglehet Nézzük végig a dolgot lépésről lépésre, onnan hogy kiadtad a parancsot. Először a gép megnézi a parancsot az aktuális tartalomjegyzékben. Mivel ez most a Workbench lemez, bekéri azt. Ott nem találja, ezért megnézi a C: logikai egységben, amely rendszeren a boot lemez C tartalomjegyzéke. Ebben meg is találja a parancsot tehát végrehajtja a DF0:-n, amiben éppen a Workbench tartózkodik.

Mi hát a megoldás, ha nem a boot lemez tartalomjegyzékére vagyunk kíváncsiak? Erre több mód is kínálkozik:

- **Winchester.** Mivel ekkor általában ezen van a C: könyvtár.
- Két **floppy** drive, az egyikben a boot lemez, így nem kell visszakérni azt. A rendszer végignézi minden meghajtót és ha egyikben sincs meg amit keres, csak akkor adja ki a kérdezőt. Ebből következik, hogy bármelyik meghajtóban lehet a boot lemez (kivéve bootoláskor, akkor csak a DF0:-ban)
- így add ki a parancsot:

Dir +

Ekkor a Workbench-ről beolvasódik a parancs, de nem hajtódik végre. A töltés befejeződése után cserélj lemezt majd írd be:

DFO:

- **Használd abszolút útmeghatározást** a Dir parancsban, a lemez nevének megadásával.

Dir Extrás:

A parancs beolvasása után a gép visszakeri az Extrás lemezt, és megjeleníti a tartalomjegyzékét.

- Tedd aktuálissá az Extras-t a CD parancssal, így nem kell mindig beírnod a lemez nevét, elég elérési út nélkül kiadni a Dir-t, azaz aktuális tartalomjegyzéket listázza. Tehát Extrás be, majd:

CD DFO:

Ha minden rendben ment, elcsodálkozhatasz, nem kérte vissza a Workbenchet, pedig a CD parancs is a C:-ben van, erről már meggyőződhetél. Akkor mi történt? Hogy tisztázzuk a dolgot, írd be:

Resident

A return lenyomása után egy rövid kis lista jelenik meg (ha a megállapodásunknak megfelelően az 1.3-s Workbenchet használod). Ezek az ún. rezidens parancsok. Ezeket előre betöltöttük a memóriába, nincs szükség az újbóli beolvasásukra, mindig rendelkezésre állnak. Hogy mikor töltöttük be? Gyanítom, ezért is a Startup-sequence a felelős (igen sok van már a rovásán, és lesz is még).

Felvetődik a kérdés, nem lehetne a Dir-t is rezidenssé tenni? De, lehetne, tedd be a Workbenchet és írd be:

Resident C:Dir Add Pure

A rezidenssé teendő parancsot teljes elérési útvonalával nem kell megadni, *ha az az* aktuális tartalomjegyzékben van. Az Add opció jelenti, hogy a parancsot hozzáadjuk a többi rezidenshez, elhagyható.

Egy parancs akkor tehető rezidenssé, ha az attribútumai közt a p flag is be van kapcsolva (lásd a List parancsnál, a 67. oldalon).

Ha nincs, akkor kell használni a Pure opciót, de egyébként nem okoz zavart.

Mivel a rezidens parancsok foglalják a memóriát, miután feleslegessé váltak, törölni kell őket. Erre való a parancs Remove opciója:

Resident Dir Remove

Csak akkor lehet törölni egy parancsot, ha azt egyik taszk sem használja (például egy másik Shell ablak). Hogy egy rezidens parancsot hány taszk használ, nyilván van tartva, amikor listát kérsz róluk, ezt a nyilvántartást látod az Usage felirat alatt. A listát a Replace opcióval is lekérheted:

Resident Replace

Ha egy parancsot úgy teszel rezidenssé, hogy használod a System opciót, azt a parancsot nem lehet később törölni, a következő bootolásig megmarad:

Resident C:Dir System

Ha valaki CLI-ben próbálkozik a Resident-el, ne az én felmerítésem emlegesse, amikor nem lesz hatása, ez csak Shell-ben működik!

Említettem az előbb, hogy a p flag-ot (kapcsolónak nevezik magyarul, de jó ha tudod az eredeti nevét is) be kell kapcsolni. Ebből következik, hogy kell rá valamilyen módszer, mondjuk egy parancs. Ez a Protect:

```
Protect C:Dir p+
```

vagy

```
Protect C:Dir p Add
```

A parancsal mind a nyolc flag beállítható (lásd a List leírásánál, a 67. oldalon), kezdőbetűjének megadásával és a + vagy az Add opcióval. Beállított flagot törölni a -, vagy Sub opcióval lehet. Egyszerre több kapcsoló is megadható. Ha nem használod a

fenti opciók valamelyikét, a megadott flag-ok kivételével mind törlődik.

Kétségtelen, hogy a Resident jó megoldás, ha egy floppy meghajtód van, hogy ne kelljen folyton lemezt cserélni, de a teljesség kedvéért meg kell még említeni egy módszert (annál is inkább, mivel a Resident-nek nincs hatása a CLI-ben).

Először tervezzük meg, mit fogunk csinálni. Létrehozunk a RAM:-ban egy C könyvtárat, abba bemásoljuk a Dirt, majd utasítjuk a rendszert, hogy ott is nézze meg a keresett parancsokat. Miután kész vagyunk a nagyszabású tervvel, lássunk hozzá!

El kell készíteni az új tartalomjegyzéket. Erre a **Makedir** parancs szolgál. Használata egyszerű, csak meg kell adni az új tartalomjegyzék relatív vagy abszolút elérési útjait és nevét:

Makedir RAM:C

Ezzel kész is. Ellenőrizheted a Dir-rel. Itt jegyzem meg, hogy nem csak fájloknak, hanem tartalomjegyzékeknek is vannak attribútumai. Néhány flag-nak, mint például az e, persze nincs értelme, de kapcsolható.

Ezután másoljuk be ebbe az új tartalomjegyzékbe a Dir parancsot.

Copy From C:DirTo RAM:C Clone

A forrás és a cél megadásánál mind az abszolút, mind a relatív útmeghatározást használhatjuk, sőt, ha a forrás az aktuális tartalomjegyzék, azt meg sem kell adni. A **From** és a **To** nem kötelező. PL:

CD C:

Copy Dir RAM:C

A parancs **Clone** opciója hatására az átmásolt tárggyal együtt annak dátuma, attribútumai és a hozzá tartozó megjegyzés is átmásolódik. Ezenkívül a következő opciók vannak:

Ali - Ha a forrás nem fájl, hanem tartalomjegyzék, annak teljes tartalma átmásolódik, létrehozva a célon ugyanazt a könyvtárstruktúrát.

Nopro - Nem engedi az Attribútumok másolását.

Quiet - Nem íródik ki a képernyőre a másolás folyamatának állapota.

A cöpy-val másolás közben meg lehet változtatni a fájl nevét. Például:

Copy C: Dir RAM:C/listtart

A Dir parancs a RAM:-ban a listtart nevet fogja viselni. Ebből a tulajdonságból következik, hogy a Copy-val másolatot lehet készíteni egy fájlról vagy tartalomjegyzékről, mert a forrás és a cél lehet azonos:

CopyC:DirC:Lista

Ezután a C:-ben lesz Dir is és Lista is (vigyázz, ha már létező nevet adsz meg, az a fájl felülíródik az újjal!).

Mostmár minden kész, csak rá kell venni a rendszert, hogy itt is *nézzen* szét, ha egy parancsot keres:

Path RAM:C Add Quiet

Az **Add** opció elhagyható, az eredmény ugyanaz, a RAM:C felkerül a keresési útvonalak listájára. Tisztázzuk, mi ez. Amikor útvonalmeghatározás nélkül kiadsz egy parancsot, azt a rendszer először az aktuális tartalomjegyzékben keresi, ha ott nincs meg, megnézi a **Path-al** definiált tartalomjegyzékekben, majd eredménytelen keresés esetén a C:-ben. Ha sehol sem akad rá, kiadja az Unknown kommand (ismeretlen parancs) üzenetet. Megjegyzendő, hogyha egy parancs rezidens, a Shell azt hajtja végre, még ha az utak valamelyikén az meg is található, elérhető, kivéve, ha elérési útjával együtt adod meg.

Honnan tudhatod meg, mik a kijelölt keresési utak? Add ki a Fatn parancsot önmagában, vagy a show opcióval

Path

> PathShow

Mi van akkor, ha ugyanolyan néven több helyen is létezik a parancs? Az hajtódik végre, amelyik elérési útvonala a listában előbb szerepel, mivel az utakat ebben a sorrendben nézi végig, és ha talált, nem is megy tovább, hanem végrehajtja azt.

Ha ez így van, miért nem kéri be a copy parancs használatakor a Workbenchet, hisz annak a neve előbb áll, mint a RAM:C ? Ezt a Quiet opció intézte el. Hatására a nem elérhető utakat nem vizsgálja. Ennek igazában csak CLI-ben van haszna, a Shell-ben automatikusan így tesz, még ha el is hagyod a Quiet-et.

Miután végignézte az elérhető utakat, de nem találta amit keresett, tesz még egy kísérletet a C:-vel. Ha ott sincs, vagy a Workbench iránti kérelmét el "cancelezed", feladja a kutatást. Van még egy opció, a Reset. Ez törli a kijelölt utakat. Ha azonban vele együtt utakat is adsz meg, a törlés után azok beállítódnak. Például:

```
Path  RAM  :C  Reset  #  ^
```

Ezután a Path listájában csak ennyi lesz:

Current dir- Itt keres először, ez nem a path érdeme, hanem alapértelmezés

RAM:C - A kijelölt path

C: - Ez sem a Path hatása, ha eredménytelenül keresett, ide mindig benéz

Egy Path paranccsal több utat is lehet definiálni, szóközzel elválasztva kell őket megadni. Például:

```
Path RAM: RAM:C Add
```

Ennyi ismeret birtokában már bármelyik lemez tartalomjegyzékét megvizsgálhatjuk, de mi van akkor, ha el akarod távolítani a feleslegessé vált programokat, fájlokat a memóriából? Ki kell őket törölni, erre való a Delete:

```
Delete RAM:C/Dir
```

Ez egy veszélyes parancs, könnyen letörölhetünk olyan dolgot is, ami még fontos. Ezek védelmére szolgál a D flag. Ha ezt kikapcsolod a **Protect** paranccsal (lásd a 173. oldalon), a fájlt nem lehet törölni.

A törlendő fájlokat meg lehet adni a joker jelekkel is, például:

Delete RAM:C/#?

Ez a lehető legveszélyesebb módszer, csak akkor használd, ha biztos vagy benne, hogy abból a tartalomjegyzékből már nincs szükség semmire.

Egyszerre legfeljebb tíz fájl nevét adhatod meg a paraméterek között. Például:

Delete RAM:C/Dir RAM:C/#?.info

Tartalomjegyzék is törölhető, ha már üres. Az AH opció használatával akkor is lehet törölni egy tartalomjegyzéket, ha még van benne valami:

Delete RAM:C AH

A törlés előrehaladtáról folyamatosan értesítést kapsz a képernyőn, de ez letiltható a Q vagy a **Quiet** opcióval.

Ha idáig eljutottál és önállóan is próbálkoztál a DOS-szal, biztosan volt már olyan eseted, hogy parancs ismételt kiadása után nem kellett azt a lemeztől betölteni, holott a parancs nem volt sem rezidens, sem a RAM:-ban Path-ba téve. Ha még nem volt ilyen eseted, most csinálunk. Töröld a RAM:-ból a Dir-t, majd add ki többször is egymás után a Dir RAM:-ot. Ha minden rendben van, csak először indul el a lemezmeghajtó, utána már nem.

Ennek a jelenségnek a **lemez puffer** az oka. Ez egy átmeneti tároló a memóriában (nem a RAM:-ban ez csak a memória egy lemezmeghajtóként viselkedő része!), ahová a lemeztől beolvasódnak a dolgok további feldolgozás végett. Ha elég nagy ez a puffer, megmaradnak benne a betöltött adatok mint jelen esetben a Dir parancs, nem kell újra a lemeztől beolvasni azt.

A puffer méretét mi is megváltoztathatjuk. A lemezpuffer növelése egy bizonyos határig gyorsítja a floppy és Winchester kezelését, de ugyanakkor memóriát foglal. A kisebb pufferrel memória takarítható meg, de egy bizonyos méret alatt nagyon lelassulnak a lemezműveletek. Az ésszerű tartomány floppy esetén 5-15 hard diszknél 25-50 kb-ot.

AddBuffers dfO:30

Hatása a DFO: lemezpuffere 30 X 512 bájtt (nem kilobájtt!) lesz.

Az eddigi ténykedésünk mind a memória rovására ment (gyanítom ezután is ez lesz a helyzet, ugyanis bármekkora memóriája is van egy számítógépnek, az mindig kevés), meg kellene valahogy tudnunk, hányadán állunk a gép eszével. A Workbench screen fejlécezen megtalálható ez a mennyiség bájtokban mérve, de kell hogy legyen valami megoldás DOS-ban is. Fejtegetésünk helyes, van olyan parancs, amely kijelzi nekünk a memória mennyiségét, ez az

Avail

Opciók nélkül használva egy kis táblázatot ad, amelyben közli a Chip és Fást memóriát külön-külön, és együtt is. Megtudhatjuk, miből mennyi van, mennyi szabad, mekkora rész van használat alatt és mekkora a legnagyobb egybefüggő terület. Ez utóbbi fontos érték, mert hiába van sok szabad memóriád, ha szét van tördelve apró darabokra, könnyen kaphatsz "Out of memory" (kifogyott a memória) üzenetet.

Az Avail-nak van három opciója, a Chip, Fást és a Totál. Ekor csak a kért területéről ad információt. PL.:

Avail Chip

Ha már a kérdezősködéskor tartunk, arról is felvilágosítást kellene kérnünk, milyen lemezkapacitások állnak rendelkezésünkre, mert ez is fontos dolog.

Info

ADÓS

A parancs az összes rendszerben lévő, működő lemezmeghajtóról, köztük a RAM:-ról, RAD:-ról (lásd a 96. oldalon) és merev lemezeiről ad információt a következő formában:

```
Drives Size Used Free Full Errs Status Name
RAM: 20k 40 0 100% ORead-Write enableRam
Disk
DFO: 880k 1600 156 90% 0 Read Only Workbench
DF1 :No disk present

Volumes available:
Ram Disk /Mounted/
Workbench /Mounted/
Extrás
```

A **Drives** a meghajtók fizikai neve, a **Size** méretük bájtokban. Az **Used** a használt, a **Free** az üres blokkok száma a lemezen, a **Full** a kihasználtság százaléka. Az **Errs** a hibák száma, a **Status** a lemez státusa. Ez utóbbi lehet **"Read-Write"** azaz olvasható - írható vagy **"Read Only"** csak olvasható, azaz írásvédett. A name értelemszerűen a lemez neve.

A **"Volumes Available"** az érvényes, használatban lévő lemezek felsorolása. Nem csak a meghajtókban lévő, hanem azok a lemezeké is, amelyeket valamely program használ. Például kinyitottad az ikonját, majd kivetted. A **"[Mounted]"** jelzi, hogy a lemez rendelkezésre is áll valamelyik meghajtóban.

Ha a parancs után valamelyik fizikai egység nevét megadod, csak arról kapsz info-t.

Ha már az informálódásnál tartunk: ha valamilyen hiba következik be a DOS parancs végrehajtásakor, a hiba típusát általában csak egy háromjegyű kódszám jelzi. Ebből nem mindenki tudja mi a probléma. Ha a hiba fellépése után kiadod a **Why** parancsot, a felmerült probléma szöveges meghatározását kapod meg.

Van az Amiga DOS-nak egy olyan parancsa is, amellyel bármelyik hiba kódról megmondja, az mit jelent. Ez a parancs a **Fault**. Használata egyszerű:

Fault kód

A Kickstart és a Workbench verziószámát is megtudhatjuk egy parancs által. Ez a

Version

Ha a parancsnak paraméterként egy library vagy device nevét megadod, akkor annak a verziószámát adja meg. Az elérési út ebben az esetben nem kell, azt a parancs automatikusan ismeri. Nem csak kérhetünk, adhatunk is információt a DOS számára. Beállíthatjuk például az időt. Ha van a gépben belső óra, ez a gép kikapcsolása után is működik.

Date

Önmagában alkalmazva kijelzi a dátumot és az időt. Ez nem feltétlen azonos a belső órában lévő dátummal és idővel! A parancs után megadva az időt és a dátumot, be is állítja azokat. Az idő formátuma: **ÓÓ:PP:MM**, óra, perc, másodperc. Utóbbi megadása nem kötelező. A dátumot a NN-**HHH**-**EÉ** formában kell megadni, a hónapot nevének három betűs rövidítésével. Például:

Date 25-jan-93 16:43

Ez a parancs csak a rendszeridőt állítja be, a belső órát nem, ahhoz külön parancsot kell kiadni.

Setclock

Ez a parancs szolgál a rendszeridőnek az órából való beállítására és viszont. Három opciója van. Ezek:

Load- Betölti az órából az időt. Ha ez elmarad, a rendszeridőt nem a belső óra határozza meg, hanem a Preferences beállítása.

Savé- A rendszeridőnek megfelelően állítja be a belső órát.

Reset- A belső órát a Preferencesben beállított időhöz igazítja.

A parancs használatára egy példa a **Startup-sequence** fájlban található, ahol a

Setclock Load

betölti a belső órából az időt a rendszerbe.

Ha nincs a gépben óra, vagy az nem működik, hibaüzenetet kapsz, "**Battery backed clock not found**" felirattal.

Bár *Murphy* szerint:

"A számítógép kiválóan alkalmas arra, hogy megkönnyítse azon dolgok elvégzését, amelyek számítógép nélkül nem léteznének"

a DOS mégsem öncélú, mégha eddig úgy nézett is ki. A CLI vagy Shell ablakból programokat is tudunk indítani ahogy azt a Workbenchen tettük.

DOS szintről programot indítani úgy lehet, mintha parancsot adnánk ki, azzal a különbséggel, hogy a programok általában nem a C:-ben vannak (de lehetnek), vagy a könyvtárukat kell aktuálisá tenni a CD-vel, vagy teljes elérési útjukat kell megadni. Persze ha az elérési útjuk Path-ban van, az is jó.

A programok indításának gyakorlását kezdjük olyan programokon, amiket már Workbench-ből ismerünk. A legfontosabb és legveszélyesebb közülük a **Formát**, amivel az új lemezeket vehetjük használatba. Ha már egyszer formázott lemezt formázol újra, annak előző tartalma elvész!

Mivel a Startup-sequence jóvoltából a Formát tartalomjegyzéke a System, már a keresési utak között szerepel, elég a program nevét beírni. Ha ez nem így lenne, meg kell adni az utat is. Tehát:

Formát Drive DFO: Name lemeznév

A Formát a parancs neve, a "Drive" jelöli, hogy utána következik a formázandó meghajtó neve. A "Name" után a formázáskor a lemeznek adandó név áll. A név több szóból is állhat, ekkor idézőjelbe kell tenni. Maximális hossza 30 karakter. Mind a Drive, mind a Name kulcsszó kötelező. Ha valamelyiket nem jól írod be, kioktatásban lesz részed.

Ha sikerült a betáplálás, rövid töltés után a parancs megkér, hogy tedd be a formázandó lemezt a DF0:-ba, majd nyomd meg a Return-t. Tégy eleget a kérésnek, és máris elidul a formázás, amelynek állapotáról folyamatos visszajelzést kapsz.

A programnak van néhány opciója, amelyet a név után kell megadni. Egyszerre többet is lehet, szóközzel elválasztva.

Az opciók:

Noicons - A formázás után a Trashcant automatikusan létrehozza a lemezen a parancs, ez az opció ezt tiltja le.

Quick - Ha a lemez már Amigával volt formázva, tehát a szerkezete ki lett alakítva, nem kell azt újra megtenni, elég csak a tartalomjegyzéket törölni és a lemez máris üresnek számít. Az opció ezt a célt szolgálja, csak a boot szektorok "sávja, a 0-s és a tartalomjegyzéké, a 40-es formázódik meg. Formázatlan, vagy más formátumú, például Atari, PC lemezeknél nem használható! Régebben formázott lemezek újraformázásánál se használd, ha lehet, mert a teljes formázás amikor újra kialakítja a szerkezetet, frissíti a lemezt. Ha nem hallgatsz rám, az így formázott lemezen a jel gyengülése miatt hiba léphet fel.

FFS - Csak az 1.3 verziótól él. Ez egy új, gyorsabb fájlkezelési rendszer, a FastFileSystem mintájára alakítja ki a lemezt. Az 1.3-rendszerben csak merevlemezeken használható eredményesen. A 2.0 kickstart már floppyra is automatikusan a FastFileSystem-et alkalmazza.

NoFFS - A formázás a régi, úgynevezett OldFileSystem, OFS szerint készíti el a lemezt. Ha ezt az opciót elhagyod, az is ugyan ezt jelenti.

Inhbit - Az utolsó opció. Megakadályozza, hogy a Disk-Validator hibája esetén a formázás megszakadjon.

Az így megformázott lemez még nem bootolható, ugyanúgy mint a Workbench esetében. A bootolhatóvá tételhez való az Install parancs.

Install Drive meghajtó

A "meghajtó" helyére az installálni kívánt lemezt tartalmazó drive neve kerül. A "Drive" nem kötelező. A parancsnak két opciója van:

Noboot - Megfordítja a parancs hatását, olyan bootblokkot ír fel, amellyel nem bootol be a rendszer.

Check - Megvizsgálja a bootblokkot, hogy az standard bootblokk-e?

Ha nem vagy megelégedve a formázáskor adott névvel, hát változtasd meg:

Relabel Drive meghajtó Name név

Sem a "Drive", sem a "Name" kulcsszó nem kötelező. A lemez neve, amennyiben a parancs sikeresen végrehajtott, "név" lesz.

Ha már az átnevezéseknél tartunk, fájlok, vagy tartalomjegyzékek nevét is meg lehet változtatni, de ehhez egy másik parancs szükséges.

Rename From régi To új

A "From" a "To" kulcsszó nem kötelező. A "régi" az átnevezendő fájl neve, amely az "új" nevet veszi fel.

Nézzük meg, hogyan lehet a lemezt lemásolni a DOS-ban:

Diskcopy From drive 1 To drive2

A "From" elhagyható. A "drive1" a forrás, a "drive2" a cél lemez meghajtójának a neve. A cél és a forrás lehet ugyanaz is, de ekkor a memóriától függően esetleg többszöri lemezcserével kell számolni. A programnak három opciója van.

Noverify - A másolatot a program automatikusan ellenőrzi, ami lassítja a folyamatot. Az opció hatására ez elmarad, de a **másolat nem lesz biztonságos**

Multi - Több másolatot készít a program.

Name név - A másolat nem az eredeti nevét veszi fel ebben az esetben, hanem a "név"-et. Például:

Diskcopy DFO: To DF1: Name Másolat

Most nézzük meg, hogyan használhatjuk a **Preferences-t** DOS-ból. Ez a program a **Prefs** tartalomjegyzékben van, de ez is a keresési utak között szerepel a Workbench lemez esetén, tehát nem kell azt külön megadni.

A programhoz három opció valamelyike is kapcsolható.

Pointer - A pointer editor hívása.

Printer - A nyomtató beállítások ablaka.

Serial - A soros pont beállítása. Például:

Preferences Serial

Mindhárom opcióra feltűnő ablakok a Preferences részei, a Workbench leírásából már ismerősek.

Printfiles

A már ismert program, amely az utilites tartalomjegyzékben található, DOS-ból használva egyszerre több fájl kinyomtatását is elvégezheti, a fájlok nevét a szükséges sorrendben vesszővel elválasztva kell beírni.

A programnak egy opciója van, a -F, amely hatására a fájl kinyomtatása után a printer az utolsó lapot is kidobja, a következő nyomtatás mindig új lapon kezdődik. Például:

Printfiles.-F, S:Startup-sequence, -F, RAM:Read-me

Van az Amiga DOS-nak egy parancsa, amely szintén több fájl együttes használatát segíti. Ez a

Jóin

amellyel több, maximum 15 fájlt fűzhetünk egybe. Természetesen, csak text fájlknál van értelme. Az összefűzött fájlok egy új fájlba kerülnek. Például:

Jóin fájl fájl2 fájl3 RAM:fájlhTo Ram;printfájl

Ha a fájlok nem az aktuális tartalomjegyzékben vannak, az elérési útvjakat is meg kell adni. A To kulcsszó helyett használható az "As" is.

Az eredmény fájl nem csak nyomtatásra használható, például szövegek összefűzése esetén az maga is szöveg fájl.

1. Logikai egységek

A könyv elején már esett szó a logikai egységekről (lásd a 32. oldalon), de csak futólag. Itt az ideje, hogy részletesen is megismerkedjünk a létrehozásuk mikéntjével.

Mivel a könyv elején egy közérthető, de bugyuta példát alkalmaztam, most, hogy sokmindent megtanultunk már az Amiga operációs rendszeréről, itt az ideje, hogy konkrétan is megismerkedjünk a lokikai egységekkel.

A logikai egységek azért vannak, hogy mind a felhasználó, mind az opreációs rendszer számára könnyebben kezelhetővé váljanak azok a fájlok, programok, amelyek a gép működéséhez szükségesek. Ez egységes szerkezetet ad, miáltal a programok együttműködése is jobban megvalósítható.

A logikai egységet az Amiga ugyanúgy használja, mintha azok fizikailag is léteznének. Ezt a tulajdonságát kihasználhatjuk, például a **DPaint** rajzoló program fioppy-ról való használata esetén, amikor is a lemez már annyira tele van a programmal és a működéséhez szükséges fájlokkal, hogy a betűkészletek számára nem marad sok hely. Ha ennek ellenére mégis több fontot akarsz használni, formázz meg egy lemezt, és add neki a Fonts nevet. Mivel így a lemezre a Fonts:-al lehet hivatkozni, a rajta lévő fontokat használni tudja a rendszer. A fonts directoryt már nem kell elkészíteni, a fájlokat a gyökérbe kell tenni!

Nézzük most sorban a bootoláskor létrejö-vő logikai egységeket. Ezeket automatikusan *hozza* létre az operációs rendszer, nem kell értük semmit sem tenni. Először is a boot le-

mez gyökerkönyvtára hozzárendelődik a Sys: logikai egységhez, majd létrejönnek a következő megfeleltetések is:

Sys:C -+ C: A DOS parancsok tartalomjegyzéke. Alaphelyzetben, ha a Path-szal nem adsz meg keresési utakat, az aktuális könyvtár után itt keres a rendszer.

SysrDevs —> Devs: Az eszközök (decives) (lásd a 88. oldalon) tartalomjegyzéke. Itt található még a Keymaps, a Printers és a Clipboards altartalomjegyzék. A Keymaps a billentyűzet-térképeket, a Printers a nyomtatómeghajtókat tartalmazza. A Clipboards-ról később lesz szó(lásd a 86. oldalon).

SysrFonts —> Fonts: A font fájlok tartalomjegyzéke. Ezen belül van minden fonttípushoz egy "fontnév" tartalomjegyzék, amiben az adott fontkészlet különböző méretekhez szükséges fájlok vannak és egy "fontnév.font" fájl, ami az egyes fontokhoz tartozó, használható méreteket tartalmazza. Ennek összhangban kell lenni a hozzá tartozó tartalomjegyzékekben lévő fájlokkal, ezt az összehangolást végzi el a Fixfonts menü program, amelyet a Workbench-et leíró fejezetben nem tárgyaltunk meg.

Ez a program átnézi az egyes fontok tartalomjegyzékét, megvizsgálja, hogy a betűkészletek milyen méretekben állnak rendelkezésre és ennek megfelelően aktualizálja a .font kiterjesztésű fájlokat.

Sys:L -> L: Az eszközezőrlő programok találhatóak itt, ezeket Handlereknek nevezzük. Rajtuk kívül itt van még a Disk-Validator, amely feladata a lemez érvényesítése és a FastFileSystem, amely az új fájlrendszert teszi érthetővé a gép számára. A régi fájlrendszer kezelője már a ROM-ban van.

Sys:Libs -> Libs: Az Amiga külső könyvtárai. Ezek programrutin-gyűjtemények, feladatok szerint csoportosítva. Később még lesz róluk szó.(lásd a 101. oldaltól).

Sys:S —> S: Batch fájlok, amelyek főleg valamely program indításakor jutnak szerephez. Itt található a nevezetes Startup-Segquence is.

Ezeket a hozzárendeléseket megváltoztathatjuk, sőt, újakat is hozhatunk létre. A hozzárendelések létrehozásához egy parancsra van szükségünk (az előbb felsoroltakhoz nem kell, azokat a rendszer önmaga hozza létre).

Assign log: tartalomjegyzék.

Az "**assign**" a parancs, a "**log:**" a létrehozandó logikai egység neve, végén kettősponttal, a "tartalomjegyzék" pedig az egységhez rendelendő tartalomjegyzék elérési útvonala. Ennek természetesen létezni kell.

Az

Assign log: "

forma, a jelenlegi tartalomjegyzéket rendeli hozzá a log: logikai egységhez.

A parancsra példát találsz a Startup-sequence-ben, amely a RAM:-ban létrehozza a **Clipboard**, **Env** és **T** tartalomjegyzékeket, amelyeket sorban hozzárendel a **Clips:**, **Env:** és **T:** logikai egységekhez. Ha már itt tartunk, elmagyarázom, mik ezek.

A **Clips:** egy olyan egység, amelyen keresztül adatokat adhatnak át programok egymásnak, például ha egy szövegszerkesztő szöveget helyez el a Clips:-ben, akkor azt egy másik taszk is fel tudja használni.

Az Env:-ben környezeti változókat, a T:-ben pedig átmeneti dolgokat tárolhatnak a futó programok.

A parancsot önmagában vagy a **List** opcióval együtt használva, a már érvényes logikai egységek és készülékek felsorolását kapod. A Remove opció eltávolítja a megadott logikai egységet a nyilvántartásból. Például:

Assign T: Remove

Egy logikai egység úgy is törölhető, hogy nem adsz meg hozzá tartalomjegyzéket. Például:

Assign log:

Az Exist opcióval lekérdezhető, hogy egy egység létezik-e. Ha az egység létezik, a parancs kiírja a nevét, ha nem létezik, a **"not recognized"** (nincs nyilvántartva) hibaüzenetet adja. Ha nem egységnevet (Device) adsz meg, hanem saját nevet (Volume), mint például a Workbench, háromféle választ kaphatsz:

- Ha arról a lemezeről nyitva van egy ablak, fut egy program, vagy valamelyik fájl használatban van róla, esetleg csak be van téve a meghajtóba, kiíródik a neve.
- Ha ezen felül a lemez elérhető, azaz benne van a meghajtóban, szögletes zárójelben utána íródik, hogy Mounted.
- Ha egyik előző feltétel sem adott, most is "not recognized" üzenet a válasz.

A választ természetesen átirányíthatjuk, például:

Assign NILRAD: Exists

A vizsgálat eredménye a NIL:re küldődik, azaz belevész a semmibe. Mi az értelme akkor a vizsgálatnak? A return kód. Általában mindegyik parancs ad egy kódot, amikor végrehajtott. Ha minden rendben volt, zéró, de ha hiba merült fel, akkor a hiba szintje, ami 1-5-ig **Warn**, figyelmeztetés; 6-10-ig **Error**, hiba; és 11-20-ig **Fail**, fagyás, azaz végzetes hiba. Az Assign Exists zéró kóddal tér vissza, ha az egység létezik, 5-tel ha nem. A parancsot követően a return kódot feldolgozhatjuk. Ezt rendszerint batch fájlban végezzük, ott használt az Assign ezen formája (lásd a batch programozásról szóló részt a 107. oldaltól).

Nézzük most meg újból az egységek listáját. Legfelül áll a Volumes, majd utána a **Workbench** lemez és a **RAM Disk**. Itt az éppen elérhető lemezek nevei vannak felsorolva, lehet tehát több is. Következő kulcsszó a **"Directories"**. Ezután áll a logikai egységek listája. Bakói az egységnevé, jobbról a hozzá tartozó tartalomjegyzék, elérési útjával együtt.

Ezek alatt a "Devices" kulcsszót követően a rendszerben élő készülékek nevei találhatóak. Van köztük ismert is, mint például a DFO: vagy a RAM:, de többségük nevével még nem találkoztunk.

Itt az ideje hogy bemutassalak benneteket egymásnak, előtte azonban el kell mondanom egy kis háttérsztorit.

2. Az Amiga egységei, készülékei

Az Amiga rendszerében az átlagos felhasználó sohasem lép kapcsolatba közvetlenül a perifériális, és perifériaként viselkedő egységekkel, még ha úgy is tűnik számára. Ez kissé érthetetlenül hangzik, de egy példa után sokkal zavarosabb lesz. Nos, amikor például egy fájlt másolsz a floppyra, tulajdonképpen te nem is a drive-vel kerülsz kapcsolatba, nem arra írsz. (Mondtam ugye, hogy zavaros lesz az egész?) A DF0:-ra amikor hivatkozol, az csak egy programnak szól. Az ilyen programokat nevezzük "Device"-eknek. Ezek végzik az adott egység feladatainak besorolását, kezelését. Ebben a munkában "Handler"-ek lehetnek a segítségükre. A devices-t úgy képzeld el, mint egy fejes titkárnő-jét. Legyen ez a fejes például a floppy meghajtó. Azért kell neki egy titkárnő, mert a multitaskban futó programok közül egyszerre többnek is az eszébe juthat a floppy-t dolgoztatni. A floppy kezelése felelősségteljes beosztás, nagy odafigyeléssel kell csinálni. Mivel csak egy író-olvasó fejpár van a meghajtóba, az csak egy dolgot tud végezni egy időben. Az Amigán mégis megengedett, hogy ugyanazt a lemezt egyszerre több taszk is használja. Akkor hogy van ez? A helyzet a következő: Minden taszk a titkárnővel érintkezik, a DF0:-hoz intézett telefonokat ő veszi fel. Ez a titkárnő rendszerezi a feladatokat, nyilvántartásba veszi a kérelmeket és beosztja főnökének az idejét. Ha egyszerre több anyag is érkezik, akkor azt "ebből is egy kicsit, abból is egy kicsit" elven adagolja be az öregnek. Ennek a titkárnőnek a segítségére van az OldFileSystem (OFS) vagy a FastFileSystem, amelyek megmondják, hogy a főnök milyen rendszer szerint tudja értelmezni az anyagokat.

Igaz, hogy a sok alkalmazott miatt lelassul valamelyest az ügyintézés, iroda is nagyobb kell a cégnek, de mindenki kapcsolatba tud lépni az a «lott göré»-l, még íia «sol& most került is a ~rál-lalathoz. A rendszer előnye tehát a kompatibilitás és a biztonságos együttműködés.

Természetesen programozói oldalról el lehet érni mindenkit közvetlenül is. Ekkor elveszítjük a kompatibilitást, de sebességet nyerhetünk. Ilyen hardver közeli kód azonban csak demóknál, esetleg játékprogramoknál megengedett, ott ugyanis nem olyan borzasztó nagy hátrány, ha a program egy-egy géptípuson nem fut. Az ilyen programok rovására írható még az is, hogy elfoglalják a gépet a többi program elől, nem használható a multitaszk.

Mint már említettem a titkárnőket "Device"-oknak nevezik. Ezek önálló programok, rendszerint a Devs:-ben tartózkodnak, várva, hogy szükség legyen valamelyikükre. A **Handlerek**, azok a kezelő programok, amelyek az adatfolyamatok kezelését, és a device-k indítását végzik, vagy nem létező egységeket helyettesítenek. Bevert tartózkodási helyük az L:-ben van.

A **Handler** egy kezelő program, amelyre akkor van leginkább szükség, ha az egység fizikailag nem létezik, mint ahogy például a **DFO:**. Ilyen elméleti készülék a **CON:**, **NEWCOW:**, **PIPE:**, **RAM:**, stb. Rövidesen mindegyikről lesz szó bővebben is. Az elméleti készülékeknél a **Handler** kezeli a készülék adatforgalmát, emiatt a rendszer és a felhasználó felé úgy tűnik, mintha ez is valódi, fizikailag is létező lenne.

A lemez kezelését végző **Trackdisk.device** kivételt képez, ő a ROM-ban található, hiszen nélküle semmit sem lehet kezdeni a floppy-val. A régebbi fájlrendszert értelmező OldFileSystem szintén a ROM-ban ütött tanyát, de az új, az 1.3-as kickstart alatt a merevlemezt kezelő FastFileSystem rendes helyén, az L:-ben van.

Az Amigán nem csak a floppy, hanem minden egység kezelése fent vázolt módon megy végbe. Ezáltal elérhető, hogy egy újabb egység rendszerbe kapcsolása egyszerű feladat, csak az őt felügyelő device-t kell létrehozni.

A merevlemez kezelését a **hdisk.device** végzi, ő is a ROM-ban található.

Ezek után nézzük, milyen egységet használhatunk még a Floppy-n és merevlemezen kívül.

A RAM: mint már említettem, a floppy-hoz hasonló módon van kezelve. Létrehozásához szükség van az L: logikai egységben a Ram-Handler nevű kezelőre. Ehhez az egységhez nem tartozik device program.

A lemezek után leggyakrabban használt fizikai egység a nyomtató. Kezelése szintén a fent vázolt módon megy végbe. Titkárnyókének Printer.device a neve, őrá a PRT: névvel lehet hivatkozni. Ezen kívül a használatához szükség van még a DevsrPrinters könyvtárban a Preferencesben beállított nyomtató meghajtóra, amely a printer által érthetővé teszi az anyagot, az L:Port-handler-re, amely a soros és párhuzamos port kezeléséhez szükséges, valamint attól függően, hogy a printer melyik portra van kötve, a Devs:Serial.device vagy Devs:Parallel.device fájlra.

Azok a DOS parancsok, amiket eddig megismertünk, de csak a DF0:-val vagy RAM:-mai kapcsolatban használtunk, ugyanúgy használhatók a PRT:-vei és minden ezután ismertetésre kerülő egységgel is. Egy kis megkötés van azonban a PRT:-re, természetéből adódóan csak írni lehet rá, adatok forrásaként nem jöhet szóba.

A PRT: használatára egy jellemző példa, a következő:

Copy S: Startup-sequence To PRT:

Ez kinyomtatja a Startup-sequence nevű szövegfájlt a nyomtatón. Ugyanezzel a módszerrel bármilyen text fájl a printerre küldhető.

A következő ilyen egység a soros port, amire telefonmodemet vagy nullmodemet köthetünk. A port egységneve a SER:, ennek a Serial.device a készülék meghajtója, a Port-Handler a kezelője.

A soros port egy olyan adatkapu, amelyen az adat bitjei sorban egymás után küldődnek ki. Mivel az adat így egy vezetéken is átvihető, kitűnően alkalmas arra, hogy telefonvonalon összekapcsoljunk két gépet és azok kétirányú adatforgalmat bonyolítsanak le. Mivel a soros átvitel szabványos, különböző típusú gépek is

kommunikálhatnak ezen a módon egymással. DOS-ból kissé kényelmetlen lenne az átvitel irányítása, szerencsére többféle program is létezik, amely ezt a feladatot hivatott segíteni.

A soros telekommunikációra legjobb példák a BBS-ek, amik telefonon elérhető számítógépes adatbázisok.

A soros porthoz hasonló adatátviteli lehetőség a párhuzamos port, ennek egységneve a PAR:. Működéséhez szükség van az L:Port-Handler és Devs:Parallel.device fájlokra.

A párhuzamos port többek között abban különbözik a sorostól, hogy az adat bájtokat egyszerre viszi át nyolc vezetéken. Ez okból kifolyólag az átvitel sebessége jóval nagyobb lehet.

A SER: és a PAR: forgalma, ha arra nem nyomtató van kapcsolva kétirányú is lehet.

Mind a párhuzamos, mind a soros portra kapcsolható nyomtató, ekkor a Printer.device mellett az adott portot kezelő device és handler programokra is szükség van.

Eddig csak olyan készülékekkel találkoztunk, amelyek a valóságban is léteznek, azaz hardverileg is jelen vannak. Az Amiga azonban fizikailag nem létező dolgokat is tud úgy kezelni, mintha ezek valóban léteznének. Ezek egyik képviselője a CON:, ami a Consol rövidítéséből származik és egy konzol megjelenítőt, jelesül egy CLI- hez hasonló ablakot jelent. A hasonlóság oka, hogy a CLI-is a CON-t használja az ablak kezeléséhez. Próbáld ki például a következőt:

Copy S:Startup-sequence CON:0/0/640/200/Név

Egy ablak nyílik a képernyőn, fejlécében a "Név"-vel, amiben megjelenik a Startup-sequence nevű fájl tartalma, majd az ablak bezáródik.

A CON: használatakor meg kell adni a nyitandó ablak elhelyezkedését és méretét, valamint a nevét. Ezt a célt szolgálja, az elérési út kinézetű számcsoport. Az első két szám az ablak bal felső sarkának koordinátái pixelemben (a Workbench mindig 640X256. vagy 640X512 (interlace) képpont méretű, PAL

képernyőn). A második két szám az ablak szélessége és magassága, szintén pixelben. A "Név" az ablak fejlécében is megjelenő név.

Az Amiga DOS az aktuális képernyő azonosítására külön jelet is használ. Ez hasonlóan működik, mint a Con:, de nem nyit új ablakot. Például:

Copy S:Startup-sequence *

A csillag ki- és bementi egységként egyaránt használható.

A **CON:** egységhez hasonló a **RAW:**, szintén a konzolon való megjelenítést teszi lehetővé, jelentősége azonban jóval kisebb. Mind a **CON:**, mind a **RAW:** csak **Output**, azaz kimeneti egység, olvasni nem lehet róluk. Közös még bennük az is, hogy a kezelő programjaik a ROM-ban találhatóak.

Az idáig felsorolt egységek, név szerint a **DFx:**, **DHx:** **RAM:** **PRT:**, **SER:**, **PAR:**, **CON:**, és **RAW:** közös jellemzője, hogy beépített egységek. Ez annyit jelent, hogy működési adataikat, jellemzőiket, mint például a kezelőprogram neve, a kickstart tartalmazza, az egységre történő első hivatkozáskor automatikusan megpróbálja azt létrehozni.

Újabb egységeket is létrehozhatunk, ha valahol meghatározzuk a szükséges paramétereiket. Ez a paraméterdefiníciós fájl a Devs:-ben helyezkedik el és **Monutlist** névre hallgat. Ez az egy fájl tartalmazza minden új egység adatát. Az, hogy egy készüléket ide bejegyeztünk, még nem jelenti azt, hogy működik is. Ehhez előbb azt a rendszerbe kell illeszteni egy paranccsal. Például a Newcon: egység a következő módon hozható létre, miután a Mountlistben be van jegyezve:

Mount Newcon:

A lényeg tehát a Montlistben van, nézzük meg azt közelebbről:

More Devs:Monuntlist

A fájl általában a megértését elősegítő megjegyzéssel kezdődik. Megjegyzés bárhol lehet a fájlban belül. A kezdetét a /*, a végét a */ jelöli.

Egy egység leírása annak kettősponttal lezárt nevével kell, hogy kezdődjön, például Newcon:. Utána különböző kulcsszavak állnak, amelyekhez egyenlőségjellel csatlakoznak a paraméterek.

Minden kulcszót külön sorba kell írni, vagy egy sorba írás esetén pontosvesszővel elválasztani. A legfontosabb kulcsszavak a teljesség igénye nélkül a következők:

Handler = A Handler neve, például L:Newcon-handler

Fylesystem = Az adatrendszer típusa, például L:FastFileSystem

Device = A készülékvezérlő program neve, az elérési út nem szükséges, Például: Clipboard.device

Priority = Az egység prioritása, azaz elsőbbsége.

Unit = A készülék egység száma, például DF0:nál 0, DF1 :-nél 1, stb.

Surfaces = A felületek, azaz lemezoldalak száma: Floppy diszknél ez kettő, de merevlemeznél több is lehet.

BlockPerTrack = A blokkok száma sávonként, floppynál 11, a merevlemeznél több.

Resrved = A boot blokkok száma. Rendszerint 2.

LowCyl = Az első sáv száma.

HighCyl = Az utolsó sáv száma.

StackSize = Az egység pufférének mérete.

BuffMemType = A buffer memória típusa, 0,1 - mindegy, 2,3-Chip 4,5 Fást Ram.

Mount = Ha ez az érték pozitív, a Mount utasítás be is tölti a kezelő és vezérlő programokat, ha negatív, ezt csak az első hivatkozás alkalmával teszi meg.

BootPri = Az egység boot prioritása. Amelyik egységnél ez magasabb, arról bootol a rendszer. Értéke -129-től 127-ig terjedhet. Az az egység, amelynek bootprioritása -129, nem bootolható.

DosType = Az adatrendszer fajtája. Ha nincs megadva, vagy 0x4445300, normál rendszer. A 0X444F5301 az FFS-t jelöli

A meghatározást végül dupla kereszttel (#) kell zárni.

Ha egy új egységet illesztés be, az egység kezelési útmutatójában le vannak írva a kulcsszavak után megadandó értékek. Mielőtt azonban ezt megtennénk, ismerkedjünk meg a Mountlist-ben már meglévő, definiált elemekkel.

Az első, amivel találkozunk, a **NEWCON:**. Ez hasonló célt szolgál, mint a **CON:**, de annál kényelmesebb, fejlettebb ablakkezelést nyújt. A **Shell** is ezen keresztül működik, ahhoz tehát elengedhetetlenül szükséges.

A **NEWCON:** kezelését az **L:** logikai egységben lévő **Newcon-Hardler** végzi, ennek meg kell lenni.

A következő a **SPEAK:**. Ez egy olyan egység, amely a rá kiküldött adatokat elmondja emberi hangon. Kezelője a **Speak-Handler**. Szüksége van ezen kívül a **Narrator-device** fájlra is, amelynek természetesen a **Devs:**-ben van a helye.

Próbáld ki a következőt:

Copy S:Startup-sequence SPEAK:

A kiejtés az angol fonetikának felel meg, de ez nem a handler-en vagy a device-en múlik, hanem egy könyvtárprogramon, a **Libs:Translator.library-n**. Ezt használja a **Narrator.device** a szöveg fonéma kódokká alakítására, amelyet már meg tud szólaltatni. A rendes magyar kiejtéshez tehát egy, a magyar kiejtési szabályokat figyelembe vevő **Translator.library** szükséges. Sajnos mindeztidáig nem tudok ilyenről, csak az eredeti, angol library-ról.

A **SPEAK:** által kiadott hang többféle lehet, ezt paraméterekkel megváltoztathatjuk:

SPEAK:OPT /páram 1/páram 2 ...

Az egyes paraméterek a következők lehetnek:

pxxx- a hang frekvenciája, ahol xxx értéke 65-320 lehet.

sxxx- a beszéd sebessége, ahol xxx értéke 30-400 szó/perc lehet.

m - Férfi hang.

f - Női hang.

r - Robot, hangsúlytalan hang.

n - Természetes hang.

aO - A fonetikát a Translator.library állítja elő.

al - Közvetlen fonetikai mód, a betáplált szöveget nem alakítja át a Translator.library, hanem azt veszi fonetikus írásnak.

dO - A mondatot a mondatvége jelnél fejezi be a hanglejtsében.

dl - A mondatot hangtanilag a sorvége jelnél fejezi be.

Ezek alapján egy kissé gyorsan beszélő, női hangot például a következő módon érhetünk el:

```
Copy S:Shell-Startup SPEAK:OPT/p200/s200/f
```

A mountlistet tovább olvasva, a következő bejegyzés az AUX:. Ez szintén egy soros port kezelő egység, mint a **SER:**, de nincs pufferelev, az adatok nem a memória egy elkülönített részébe kerülnek a portra való kiküldésük előtt, hanem azonnal oda irányulnak. A kezelését az **Aux-Handler** végzi.

A következő egység a Mountlistben, a **PIPE:**. Ez egy olyan egységet kezel, ami fizikailag nem is létezik. A **PIPE:** egy memóriapuffer, amin keresztül a programok adatokat cserélhetnek egymással. Nézzünk erre egy példát!

Nyiss két Shell ablakot. Az egyikbe írd be:

```
Copy S:Startup-Sequence PIPE:
```

Miután ezt a Return megnyomásával elküldted, kattints át a másik ablakba, és ott ezt írd be:

```
Copy PIPE: NEWCON: 0/0/640/200/Akármi
```

Természetesen a PIPE:-ről jövő adatokat nem csak a képernyőre lehet kiküldeni, hanem akár egy fájlba valamelyik lemezen, vagy nyomtatóra is.

A Montlistben tovább nézelődve, ráakadhatunk a RAD: leírására is. Ennél már igen sok kulcszót találunk megadva, amelyeket figyelmesen átnézve, az az érzésünk támad, mintha ez egy lemezegység lenne, ugyan is felületei, szektorai és blokkjai vannak. A megérzés nem csal, ez tényleg egy lemez meghajtó, azonban fizikailag nem létező. Elindítása után (Mount RAD:) a memóriában alakít ki egy lemezként viselkedő területet, a RAM:-hoz hasonlóan. Van azonban néhány lényeges különbség. Az egyik, hogy ez teljesen úgy viselkedik mint például a floppy, előre kialakított sávjai és szektorai vannak, sőt még boot szektorokkal is rendelkezik (lásd: Reserved=2)! Tehát míg a RAM: rugalmasan változtatja méretét a benne lévő adatok függvényében, addig a RAD: mérete mindig állandó, ha ez a hely betelt, akkor sem tudsz a RAD:-ra több adatot felvinni, ha még lenne is memória a gépben.

A másik különbséget már megpedzettem, a boot blokk említésével: A RAD: nem törlődik a reset hatására (általában). Ha nincs más bootolható egység (nincs a DF0:-ban floppy és nincs merevlemez sem), a RAD:-ról bebootol a gép.

Van néhány eset, amikor a RAD: is elvész:

- Ha a reset a gép kikapcsolásával jár.
- Némely Guru hibaüzenet után.
- Hard resetkor. Ilyet főleg a vírusirtó, víruskereső boot programok tudnak csinálni, mint például a Scoopex Utility Boot. A hard reset mindent kitakarít a memóriából, még a vírusokat is.
- Vírusmentesítéskor. A vírusok ugyanazt az eljárást használják a resetelés túlélésére, mint a RAD:, ezért a vírusirtók többsége a RAD:-ot is vírusnak nézi, törli azt a memóriából.
- Egy megabájt Chip intmúriáviil rGudclkez.0 gépek&a a három gombos reset is hard resetet okoz. Ez azonban kiküszöbölhető a Setpatch R utasítással, minek hatására a következő

három gombos reset csak soft reset lesz, míg az említett gépeken is. A parancs paraméter nélkül használva kiküszöböli a Kickstart néhány hibáját.

Mivel a RAD: igen szívós egy valami, a kiirtásához és az általa lefoglalt memóriaterület felszabadításához külön DOS parancs kell.

Remrad

A parancsnak van azonban egy apró hibája. Törli ugyan a RAD: tartalmát, és felszabadítja a lefoglalt területet, de nem törli *azt* az érvényes egységek listájából. Ennek ellenére, a RAD: már nem használható, "Not a DOS Disk" nem DOS lemez megjegyzéssel szerepel.

Ez akkor jelent problémát, ha újból létre akarod hozni a RAD:-ot, mivel a létező egységek nevei között szerepel. A Device RAD: already mounted (az egység már be van építve) hibaüzenetet kapod és a RAD: nem jön létre. Más megoldás nincs, ha a RAD:-ot ismételten létre akarod hozni, újra kell bootolni a rendszert.

Érdemes megismerkedni a RAD: Montlisbeli paramétereivel közelebbről is.

Device = Ramdrive.device

Az egységet a Devs:-ben lévő Ramdrive.device nevű program vezérli, működéséhez elengedhetetlen.

Unit = 0

A RAD: egység száma. Elvileg egyszerre több RAD: is lehet a rendszerben, gyakorlatilag azonban az 1.3 kickstart alatt ez nem működik jól.

Flags = 0

Ez egy kapcsoló, értéke szinte mindig 0 kell, hogy legyen.

Sarfaces = 2

Ez a felületek, azaz író-olvasó fejek száma. Annak ellenére be kell állítani, hogy a valóságban ez nem is létezik. Ennek az oka,

ADÓS

hogy teljesen úgy kezelődik a RAD:, mint egy létező lemezmeghajtó. A felületek szokásos száma 2, de ettől el lehet térni. Jelentősége a lemezkapacitás kiszámításánál van.

BlocksPerTrack = 11

A blokkok sávonkénti száma, szintén fontos, de elméleti érték.

Reserved = 2

A boot blokkok száma. Ez Amigán mindig 2.

Interleave = 0

Ez egy korrekciós tényező a lemez beolvasásához. A legtöbb esetben 0.

LowCyl = 0; HighCyl = 21

A RAD: szerkezete 22 sávból áll, amelyek számozása 0-tól 21-ig terjed.

Buffer = 5

A RAD: kezeléséhez felhasznált buffer mérete, 5X512 bájttal, azaz 2,5 kB.

BuffMemType = 1

A buffer mind a Chip, mind a Fást memória területén lefoglalható.

A paraméterek ismeretében már magunk is bejegyezhetünk egy rezidens ram diszket, de a következőket kell szem előtt tartanunk:

- A Montlistben ugyanazon néven nem lehet két egység, az új RAD:-ot más néven, például RAD1:, kell bejegyezni.
- Másik RAD: bejegyzésére akkor van általában szükség, ha annak más a mérete, ezért tudni kell, a méret hogyan határozható meg. Erre való a következő formula, ahol C a kapacitás:

$$C = \text{Surfaces} * \text{BlocksPerTracks} * (\text{HighCyl} - \text{LowCyl}) * 512$$

Mivel a Surfaces, BlocksPerTrack és LowCyl értékeit általában 2, 11 és 0-nak vesszük, ezek változtatása nem indokolt, így a méret egyedül a HighCyl-től függ:


```
HighCyl = ----- -1
          Surfaces * BlackPerTrack * 512
```

```
azaz HighCy=-----1
          11264
```

így már mindenki maga döntheti el, hogy mennyi memóriát szán resetálló memórialemezre.

Továbbmenve a Mountlistben, találunk egy DF2: bejegyzést. Ez kisebb kapacitású, 440 kBájt-os lemezegységet definiál, gyakorlatilag azonban nincs jelentősége.

A következő egység a **FÁST:**. Ez egy **FFS** alatt működő merevlemez határoz meg. Jelentősége csak 1.2-es kickstarú gépeken van, amelyek nem ismerik fel automatikusan az **FFS**.-es merevlemezeket. Ekkor a **Startup-sequence-ben** mountálni kell ezeket a partíciókat.

Egy új kifejezés a **partíció**, lássuk mit jelent. A merevlemez logikailag szét lehet osztani részekre, amelyek úgy működnek mintha önálló lemezek lennének, egymástól függetlenül formázhatók, különböző filesystem-eik lehetnek, stb. A partícionálást formázás előtt kell végrehajtani. Az erre szolgáló programot általában a merevlemezrel együtt adják. A partíciók kijelölése után az érintett részeket meg kell formázni. A partíciók folyamatosan helyezkednek el a lemezen. Például egy 400 sávú merevlemez szétpartícionálhatunk úgy, hogy az első partíció a 0-200, a második a 201-399 sávokon lesz. Természetesen több partíciót is létrehozhatunk és ezeknek nem is kell azonos méretűeknek lenni.

A több részre partícionálás nem kötelező, a merevlemez állhat egyetlen partícióból is.

Az 1.3 kickstartú gépeken már nincs szükség az **FFS** mountálására, ezek automatikusan ismerik és használják azt merevlemezen. Elvileg az **FFS** alkalmazható lenne floppy-ra is,

ADÓS

de ez csak elv, a gyakorlatban nem igazán működik. A 2.0 kick-startban ez a hiba már ki van javítva, sőt, az alpból FFS-el használja a floppy-kat, de ismeri a régi fájlrendszert is.

Az utolsó bejegyzés a DF1:-é, erre azonban soha nem lesz szükség, a külső floppy-kat is automatikusan ismeri fel és használja a gép.

Mint látható, minden, a lemez működtetéséhez szükséges adat megtalálható a Mountlistben, ilyen alapon létrehozhatunk olyan lemezegységet is, amelyik nem Amiga lemezeket kezel. Az egyik ilyen a sokak által ismert MSH:, amely 720kBájtos PC formátumú lemezeket tud kezelni.

A Mountlistbe történő bejegyzés mellett szükség van azonban az MSH:-hoz két fájlra is. Az egyik a **MessyFileSystem**, amit az L:-be, a másik a Messydisk.device, amit a Devs:-be kell másolni. Az MSH: bejegyzése a Mountlistben így néz ki:

```
MSH: FileSystem = L:MessyFileSystem
Device= Messydisk.device
Unit = 0
Flags = 0
LowCyl= 0 ; HighCyl = 79
Surfaces = 2
BlocksPerTrack = 9
Buffers = 5
BufMemType = 1
BootPri = 0
StackSize = 4096
Priority = 5
Mount = 1
GlobVec = -1
```

#

Ha rendelkezünk a fent nevezett két programmal és a Mountlist is el van készítve az előző listának megfelelően, a

Mount MSH:

parancsot követően a PC 720 kB formátumú lemezeket is ugyanolyan módon kezelhetjük, mintha Amiga formátumúak volnának, akár programokat is futtathatunk róluk (persze csak Amiga programokat, hisz a gép továbbra is Amiga).

Mountálás után a belső floppy-t kétféle módon is elérhetjük, az két lemez meghajtónak látszik. Elérhető egyszer a hagyományos módon, DF0:-ként, ha Amiga, vagy MSH:-ként, ha PC-s lemezt akarsz használni.

Mivel mind a DFO:, mind az MSH: automatikusan észreveszi a lemezcserét, elő fordul, hogy a kettő közül az egyik, a használt lemezzel ellentétes, lemezhibát jelez. Például ha Amiga lemezt teszel be, "**MessyDisk Error**" üzenetet kaphatsz, mert a Messydisk.device-nek nem felel meg a lemez. Ez azonban nem igazi hiba, Cancel-t válaszolva minden megy tovább.

Ha nem a belső, hanem a külső, a DF1: meghajtóban akarod kezelni a PC-s lemezeket, az Unit = 0 sort meg kell változtatni Unit = 1-re, DF2:-nél Unit = 2-re, stb. Abban az esetben, ha több meghajtót is akarsz így használni, mindegyiknek el kell készíteni egy bejegyzést, különböző névvel és egység számmal. Az alkalmazott névre nincs megkötés.

3. A könyvtárak

Többször esett már szó az Amiga könyvtáiról, de részletesen még nem beszéltük meg őket. Ezeknek a könyvtáraknak mindössze annyi közük van a bibliotékákhoz, hogy itt is különböző szerzőktől összegyűjtött műveket őriznek, téma szerint csoportosítva. A könyvtárak, angolul library-k tehát olyan programrutinok gyűjteményei, amely hasonló feladatokat látnak el. A programoknak már nem kell ezeket a rutinokat tartalmazniuk, helyette csak megnyitják a szükséges library-t vagy library-kat. Az első megnyitáskor a library betöltődik a Libs: logikai egységből (néhány fontosabb könyvtár a ROM-ban van, nem kell betölteni) a memóriába. Ha ezután egy másik program is használni szeretné ezt a könyvtárat, azt már nem kell ismételt

betölteni, az előzővel közösen használhatja azt. A könyvtár felépítése a hétköznapi felhasználó számára lényegtelen, csak programozók számára fontos. Annyit azonban tudni kell róla, hogy minden könyvtár két fő részből áll. Az egyik a belépési táblázat, amely az egyes funkciók kezdetére mutat, a másik pedig a funkciók csoportja. Az egyes rutinokat mindig a belépési táblázaton keresztül hívják meg a programok.

Láthatók tehát a könyvtárak előnyei:

- Programozói munkát takarítanak meg, rutin feladatokat nem kell minden programban megírni, elég azokat a könyvtárból felhasználni.
- Memóriát takarítanak meg, mivel egy megnyitott könyvtárat számtalan program használhat.
- Szabványossá válnak a programok, új rendszereken is képesek futni.

Akiket a könyvtárak programozói szempontból is érdekelnek, bővebb információkat szerezhetnek az angol és német nyelven hozzáférhető ROM Kernel Manual-ból, vagy az Aurum könyvek előkészületben lévő, könyvtárprogramozással foglalkozó kötetéből.

Nézzük most meg, mely könyvtárakkal találkozunk leggyakrabban és ezek milyen célt szolgálnak:

A legfontosabb az Execlibrary, amely a gép ROM-jában található. Ez a könyvtár tartalmazza a rendszer működéséhez szükséges funkciókat, például más könyvtár megnyitása is ennek egy funkciójával végezhető. Ha egy program rendben véget ért, bezárja az általa kinyitott könyvtárakat, amelyhez szintén Execlibrary rutint használ fel. Ez nyilvántartja, hogy egy könyvtárat hányan használnak. Ha már senki sem használ egy könyvtárat, az akár töölhető is a memóriából.

A következő, nagy fontosságú könyvtár a DOS.library, amely szintén a ROM-ban van. Ez a lemezek és egységek működéséhez nyújt funkciókat.

Az **Intuition.library**, amely szintén ROM könyvtár, a multi-taszak kezelését végzi. Ennek a segítségével működnek az ablakváltó, záró, mértetváltó, stb, gadgetek is.

A **Graphich.library** is a ROM-ba fészkelte be magát. Ennek a feladata a grafika kezelése.

A **Rombootlibrary** egy kevésbé ismert library a ROM-ból. Ennek a funkciói végzik a bootolást.

A **Ramlib.library** szintén a ROM-ban található, ez a Ram diszkek kezeléséhez tartalmaz funkciókat.

A többi könyvtár, amivel most megismerkedünk, a **Libs**: logikai egységben található, kérjük először egy tartalomjegyzéket róluk. A következő könyvtár-fájlokat találjuk itt:

diskfontlibrary - A fontkészletek kezelését végző rutinokat tartalmaz. Nélküle csak a két ROM-fontot (Topaz/8 és Topaz/9) lehet használni.

icon.library - Mint nevében is benne van, a Workbench ikonjainak kezelését végzi, nélküle az nem is működik.

info.library - Akkor jut szerephez, amikor a Workbench menü Info parancsát használjuk valamely fájl info adatainak a beállítására. Ha ez nem elérhető, az említett menüpont sem működik.

mathieeedonbbas.library

mathieeedoubtrans.library

tnathtrans.library - Ez a három könyvtár, azt hiszem nem volt nehéz rájönni, matematikai funkciókat végez. Az 1.3 rendszerrel szállítottak már át vannak dolgozva úgy, hogy képesek az aritmetikai kooprocesszor használatára is.

Aritmetikai kooprocesszor? Az meg milyen jószág? Nos az tulajdonképpen egy mikroprocesszor, amely matematikai műveletek nagy sebességű elvégzésére szakosodott. Bár a gép lelkét képező processzor is képes ilyen dolgokra, de mivel neki sokminden mást is tudni kell, így nem túl jó matematikus. A matemati kooprocesszor, mint nevében is bennne van, együttműködik, kooperál a főprocesszorral, elvégzi helyette a számításigényes feladatokat, nagy mértékben gyorsítva az alkalmazások sebességét.

Jelenleg két típusa van az Amigában használható kooprocesszoroknak, az MC 68881 és az MC 68882. Lényegi különbség nincs közöttük, azonban mindegyik csak az MC 68020 vagy MC68030-as mikroprocesszorokkal képes együttműködni. Ilyen processzorok találhatóak az Amiga 1200 és Amiga 3000 típusú gépekben. Elvileg van mód, hogy a 68000-es mellé kooprocesszort építsünk, de a 16 bites adatbusz és a kooprocesszorkezelő utasítások hiánya miatt kicsi a jelentősége.

Amiga 500, 500+ és 2000 típusú gépekben csak turbókártyákra szerelve, az említett processzorokkal együtt használható.

Meg kell említeni a Motorola processzorcsalád MC 68040 típusú tagját, amely már önmagába integrálva tartalmazza a kooprocesszort, ahhoz már nem kell külön beépíteni.

Talán nem tűnt ki az előbb leírtakból, hogy a kooprocesszor nem kötelező, csak opcionális. Az Amiga 1200-ban például csak a helye van meg, oda külön kell beszerezni. Az A 3000-be gyárilag van beépítve, az A 4000 pedig az MC 68040-el van szerelve.

A következő könyvtárral, a `translator.library`-val már találkoztunk a `SPEAK`: készülék ismertetésével. Ez csak egy funkciót tartalmaz, amely a feldolgozásra átadott stringet az angol kiejtési szabályoknak megfelelően átalakítja fonéma kódokká, amit a `Speak-Handler` már helyesen tud kiejteni.

A `Workbench` könyvtárai közül az utolsó a `Version.library`. Ez a `kickstart`, a `Workbench`, valamely `library` vagy `device` verziószámának lekérdezését hivatott segíteni. A `DOS Version` parancsa, vagy a `Workbench Special` menüjének `Version` menüpontja használja elsősorban.

Az eredeti `Workbench` lemezen található könyvtárakon kívül még jónéhány létezik. Ezek általában valamely programhoz kapcsolódnak, azzal együtt juthat hozzá a felhasználó. Érdemes a leggyakoribb könyvtárakkal legalább annyira megismerkedni, hogy **tudjuk, melyik mire való.**

Arp.library - Fejlettebb kérdezőket kezelő funkciókat tartalmaz elsősorban. Egy ilyen kérdezővel ha fájlnévet kér a program, nem kell emlékezetből megadni az elérési utat^és a fájl nevét, mint például a Notepad-nál, hanem egy ablakban szabadon választhatunk az egységekből, az azokon lévő könyvtárakból, megtekinthetjük az abban lévő fájlokat stb. Ilyen kérdezőben úgy is megadhatjuk egy fájl nevét, hogy a nevére klikkelünk.

Reqtools.library - Hasonló az előzőhöz, szintén fejlett kérdezőkezelési funkciókat tartalmaz.

Req.library- mint az előző kettő.

Explode.library - A **Turbo Imploder** nevű tömörítőprogrammal tömörített fájlok kitömörítéséhez szükséges rutinok gyűjteménye

Powerpacker.library - Szintén egy tömörítő program segédeszköze, de ez a **Powerpacker-é**. A **PP** az egyik legelterjedtebb tömörítő program az Amigán.

Mi is az a tömörítés? A fájlok általában sok ismétlődő, azonos bájtokból álló részeket tartalmaznak, amelyeket egy ügyes algoritmus segítségével kisebb méretűre is össze lehet nyomni. Például ha öt nulla bájtot követi egymást, nem kell mindet eltárolni, elég csak utalni rá, hogy öt nulla következik. Az így betömörített fájl közvetlenül nem használható fel, azt előbb vissza kell alakítani eredeti formájába.

A tömörítés szempontjából megkülönböztethetünk adat és programfájl tömörítést. A betömörített adatfájlok csak kitömörítés után használhatók újra, ellenben a program fájlok betömörítve is futtathatók, ugyanis tömörítéskor elébük kerül egy kis programocska, amely képes a memóriában futás közben kitömöríteni az utána következő programot és lefuttatni azt. Az ilyen tömörítést "Runtime", futás közbeni tömörítésnek is nevezik.

A tömörítéssel jelentős helyet takaríthatunk meg, ami nem elhanyagolható szempont, ha floppyról használod a gépet. Hátrá-

ADÓS

nya, hogy a kitömörítés időbe kerül, de ennek egy része megtérül, mert a kisebb fájl betöltése kevesebb időt vesz igénybe.

Mint már említettem, a **Powerpacker** a legelterjedtebb Amiga tömörítő, angolul **Cruncher** program. Ehhez léteznek olyan programok, amelyekkel a tömörített adatfájlok tartalma is megnézhető. Az eljárásban itt is szerepel a kitömörítés, de azt a program maga végzi. Például a **PPMore** és **PPType** a tömörített szövegfájlok tartalmát jeleníti meg, a **PPAnim**-mal animációkat nézhetünk meg, a **PPShow**-val pedig Powerpackerrelt IFF képeket nézhetünk meg.

Nem volt még szó az IFF-ről. Ez egy szabvány fájlformátum, nem csak az Amigán létezik. Leginkább grafikák tárolására szolgál, de lehet **IFF** fájl hangminta, szöveg stb. is.

A legtöbb rajzolóprogram, amely Amigán fut, például a **DPaint**, az **BFF-ILBM** nevű grafikus fájlformátumot használja. Az ilyen fájlok kezelésére szolgáló programrutinok gyűjteménye az **Iff.library**.

Van a fájlok tömörítésének más módja is, az **archiválás**. Ez csak adatfájlkénti tömörítésre alkalmas, programokat is csak ebben a formában tud sűríteni. A leglényegesebb különbség, hogy egy sűrített fájlba több fájlt is betömöríthet, sőt akár egy egész lemezt is, az egyes fájlok tartalomjegyzékeivel, elérési útjaival együtt. Kitömörítéskor újra helyreáll az eredeti könyvtárstruktúra.

Ennek a módszernek, mint az a nevében is benne van, az archiválásban van jelentősége, a keletkező egyetlen fájl tartalmazza az összesét, könnyebb kezelni, nem kell arra ügyelni, hogy nehogy lemaradjon valami másolás közben.

Az archiváló programok általában jobb hatásfokkal működnek, mint az előbb ismertetett tömörítők, mivel azoknál a sebesség is befolyásoló tényező, a gyakoribb kicsomagolás miatt.

A legelterjedtebb archiválók az **LHA** és az **Lharc**. Utóbbi **PC-5 változatban is létezik, elvileg kompatibilisek is egymással, az Amigával becsomagolt fájlokat PC-n is ki lehet bontani. Azért írom, hogy elvileg, mivel sajnos rossz tapasztalataim vannak. Jól**

meg kell választani, hogy melyik verziószámú Amigás Lharc-al készített archív fájlhoz melyik PC-s verziót használjuk.

Idáig olyan archiváló programokról volt szó, amelyek fájlként sűrítenek. Ezek azonban nem jók bootloderes, nem Amiga DOS lemezszerkezetet használó programoknál. Ide olyan archiváló kell, amely nem fájlokat, hanem lemez szektorokat sűrít. Van ilyen is, például a **Disk Imploder**, vagy a **DMS**. Ezek a programok egymás után letömörítik a lemez blokkjait. Hátrányukra írhatók, hogy minden szektor, a fájl adatain kívül tartalmaz a szektorra és a fájlra vonatkozó plussz információkat is, amik fájl tömörítésénél nem kerülnek be, mert nincs rájuk szükség. A másik ok, hogy az egész lemezes tömörítéskor a nem használt blokkokat is betömörítjük, ami nem okozna gondot, ha teljesen üresek lennének, de általában törölt fájl adataiból képződött szeméttel vannak tele. Utóbit át lehet hidalni, ha a program képes arra, hogy csak a foglaltsági térképen bejelölt blokkokat tömörítse.

4. A batch programozás

Ha eddig eljutottál és érted is amit olvastál, belevághatunk a DOS legnehezebb részébe, a **batch programozásnak**. Ehhez mindenképpen konyítani kell valamit, mert nélküle az összeállított saját felhasználói lemezünkhöz nem tudnánk startup-sequence-t sem írni.

A batch programokról már volt szó, tudjuk, hogy ezek szöveg-fájlok, amelyekben DOS parancsok és paramétereik vannak. Ezek a parancsok sorban egymás után, vagy bizonyos parancsok hatására, esetleg feltételesen valamilyen más sorrendben hajtódnak végre. Előfordulhat, hogy valamilyen feltétel hatására egy részük nem, vagy ellenkezőleg, csak egy részük kerül végrehajtásra. Az hogy melyik az egyik rész és melyik a másik, természetesen relatív dolog.

A batch, azaz köteg program végrehajtását az **Execute** utasítással lehet kezdeményezni:

Execute név (páram 1) (param2...

A "név" a batch fájl neve, szükség szerint elérési útjával együtt. A "**param1**" és társai, az átadott paraméterek. Ezeket a programban fel lehet dolgozni, értéküktől függően döntéseket lehet hozni.

Shell-ben, ha a **Shell-Seg** rezidens és a batch fájl s flagja be van kapcsolva, az **Execute** nélkül is futtatható a program. Ekkor elegendő csak a program nevét és a paramétereket beírni. A batch fájlok között kitüntetett jelentőséggel bír a **Startup-sequenc**. Ez a bootolás után automatikusan lefut.

Sokan helytelenül a Startup-sequence végrehajtódását is bootolásnak mondják, valójában az csak a boot blokkokban lévő program és a ROM boot rutinjának a lefutását jelenti. Ezek létrehozzák a logikai egységeket, letesztelik a hardvert, beillesztik a létező egységeket, mint pl. külső floppy. Ha ez lefutott, megpróbálják lefuttatni a Startup-sequence-t. Ha ez nincs meg, a rendszer akkor is bebootolt.

Legjobb lesz tehát, ha ezzel a különleges jelentőségű program megismerésével kezdjük batchprogramozási tanulmányainkat.

A program írásához bármelyik, **ASCII** formátumban is kimenteni képes szövegszerkesztőt használhatjuk. Ilyen például a Notepad, de az AmigaDOS-nak is van erre a célra szövegszerkesztője, az Ed. Ennek használata:

Ed név (**SIZE** méret)

A "név" a fájl neve, szükség esetén elérési útvonalával együtt. Alaphelyzetben az Ed-dal 40 kB hosszú szöveget szerkeszthetünk, de ha ez nem elég, a **SIZE** kulcszót követő méret megadásával növelhetjük.

Az Ed új ablakot nyit, amiben közvetlenül szerkeszthetjük a szövegünket. Ha létező fájl adunk meg, ez bekerül a szerkesztőbe, folytathatjuk rajta a munkát. Nem létező fájl név megadásakor az Ed létrehozza azt.

Anélkül, hogy mélyebben belemerülnénk az Ed használatába, néhány funkcióját meg kell ismernünk. Azért nem foglalkozunk vele részletesen, mert sokkal jobb szövegszerkesztők is beszerezhetők nála. Az Ed-et legfeljebb néhány soros batch fájlok megírására szokás használni, ahhoz a program minimális ismerete is elegendő.

A szerkesztő ablaka logikailag két részből áll, az egyik maga a szerkesztő terület, a másik az alsó ún. kibővített parancssor. Itt a programot vezérelhetjük, például a szöveget menthetjük ki.

A parancs sorba nem lehet a kurzor gombokkal bejutni. A parancs adását az Esc billentyűvel kell kezdeni. Ekkor megjelenik a parancs sorban egy csillag, lehet kezdeni a betáplálást. Nézzük át a legfontosabb parancsokat.

X - kimentí a szerkesztett szöveget az Ed indításakor megadott néven, majd befejezi a szerkesztést.

Q - A szerkesztett szöveg kimentése nélkül lép ki a programból. Ha változtatás volt a fájlban, a parancs végrehajtása előtt figyelmeztet rá, hogy ez elveszhet. Ha Y-nal válaszolsz, csak akkor hajtja végre.

SA - Kimentí a szöveget, de nem lép ki a szerkesztőből. Hosszabb fájlok szerkesztésénél ajánlatos időnként használni, nehogy egy váratlan áramszünet tönkretegy az egész munkánkat. Ez a kibővített parancs lehetővé teszi, hogy a fájlt ne csak a megnyitásakor használt néven mentse ki, hanem akár más tartalomjegyzékbe, más néven is. Ehhez a parancs után idézőjelben meg kell adni az új elérési utat és fájl nevet.

SH - Információkat ad a szerkesztett fájlról, úgymint a neve, tabulátor és margó méretek, blokk kijelölés, valamint a buffer telítettség. A tabulátor méretét és a margókat beállíthatjuk az ST, SL és SR kibővített parancsokkal.

U - Ezzel a parancssal vissza lehet állítani a legutolsó szerkesztő műveletet, például sortörlést.

Az Ed-ben való szerkesztést szerkesztő billentyűk segítik. Ezek a CTRL és egy betű kombinációi. Ízelítőül néhány:

- CTRL+B** - Törli a kurzor sorát.
- CTRL+Y** - Törli a sorból a kurzor utáni karaktereket.
- CTRL+O** - Törli azt a szót, amelyben a kurzor áll.
- CTRL+V** - Újrarajzolja a képernyőt, ha azt valami elrontotta.
- CTRL+A** - A jelenlegi sor után új sort szúr be és az elejére áll a kurzor.
- CTRL+J** - A kurzort a sor végére viszi. Ha az már ott áll, akkor a sor elejére.
- CTRL+E** - A kurzort a szöveg elejére viszi, ha az már ott áll, akkor a szöveg végére.

Ezekon kívül van még néhány szerkesztőbillentyű-kombináció, próbálgatással meg lehet találni őket.

Már van annyi ismeretünk az Ed-ről, hogy elkezdhetjük vele a munkát, töltsd is be minjárt a Startup-sequence-t:

Run Ed S:Startup-sequence

A Run nem szükséges, de így a program multitaszkban indul el, megmarad a Shell promptja.

Nézzük végig, miből áll ez a fájl. Az első sorában ismerős parancsot találunk, (lásd a 96. oldalon)

C:SetPach >NIL:

A >NIL: azt jelenti, hogy a parancs esetleges eredményére nincs szükségünk, azt küldje a NIL: látszólagos egységre. Ez az egység olyan mint a szakadék a világ végén, bele lehet dobni bármit, és az örökre el is tűnik (lásd az egyszeri Bushman esetét a kólás üveggel).

Addbuffers DFO: 10

Beállítjuk a DFO: bufferét 5 kB-ra. A parancsot már leírtuk (lásd a 77. oldalon).

CDC:

A C: logikai egységet tesszük aktuálissá, ami gyorsít valamit a parancsok elérésén, mert egyébként is megnézi a C:-t, de előtte

az aktuális könyvtárat is át kell vizsgálnia a parancs után. Így rögtön, az aktuális könyvtár átnézése során ráakadhat a parancsra.

Echo "A500/A2000..."

Ez egy eddig nem említett parancs. Az utána következő szöveget kiírja alapesetben a képernyőre. Ha a kiírandó szöveg nem tartalmaz szóközt, nincs szükség az idézőjelre. A parancsnak van néhány opciója és ún. vezérlő karaktere, amivel irányítani lehet a kiírást. Az opciókat mindig a szöveg után kell írni.

Noline - A szöveg kiírása után nem jön létre soremelés, a következő Echo innen folytatja.

First n - A szöveg az n. karaktertől kezdődően íródik ki.

Len n - Az előző fordítottja, csak az első n karakter kerül kiírásra.

Az idézőjelek közé tett szövegben vezérlő karaktereket lehet alkalmazni, amivel a kiírt szöveg formázható. A vezérlő karakterek általában a *E[karakterekkel kezdődnek, de van kivétel. A "*E["-t Escape jelnek nevezzük. Megjegyzendő, hogy ezek némelyike a nyomtató vezérlésére vonatkozik, képernyőn nem látható hatásuk. Ebből következik, hogy az Echo átirányítható a képernyőről a printerre. Ez teljesen logikus, hiszen az Amiga szempontjából az ablak, amit a **CON:** vagy **NEWCON:** egységen keresztül ér el, ugyanolyan mint bármelyik más egység, például a **PRT:** vagy **DFO:**. Ha az átirányítás fájlba történik meg, kell adni a hozzá tartozó elérési utat és nevet is. Az átirányítást a > jellel kell jelezni. Például:

Echo >PRT: "Most nyomtatunk." NOLINE

Néhány példa a vezérlőkódokra: (A teljes lista a függelékben található).

*E[0m	Normál írás.
*E[1m	Kövér betűs írás
*E[3m	Dőlt betűs írás.
*E[4m	Aláhúzás bekapcsolása.
*E[24m	Aláhúzás kikapcsolása.

*N Soremelés, a szöveg új sorban folytatódik.

Mivel a csillag karakter ilyen kitüntetett jelentőséggel bír, ha a kinyomtatandó szövegben is alkalmazni akarod, duplán kell beírni, például:

Echo "Ez egy csillag: **"

A szövegen belül többször is lehet vezérlő karaktereket alkalmazni, mindegyik hatása az őt követő szövegen érvényesül. Például

Echo ""E(0m Normál szövegben "E^m aláhúzott
"E(24m rész."

Sys: System/FastMemFirst

A Startup-sequence következő sora a már ismert programot indítja. Ez utasítja a rendszert, hogy ha csak teheti, a Fást memóriát használja.

BindDrivers

A parancs bővítő egységeket, mint például merevlemez, stb, csatol a rendszerhez. Ezeket az egységeket kezelő programok az Expansion tartalomjegyzékben vannak. Akkor van rá szükség, ha nem autokonfigurációs hardvereket használunk, amelyek önmaguktól kerülnek használatba.

SetClock Load

A rendszeridőt átvesszük a belső órából. Ha nincs belső óra, a **"Battery backed Clock not found"** üzenetet adja.

FF> NIL: -0

A parancs egy gyorsabb szöveg megjelenítést tesz lehetővé, ebben a formájában. A gyorsított megjelenítés kikapcsolható az

FF-N

paranccsal. Van azonban egy másik alkalmazási területe is, amikor a Fonts tartalomjegyzékből egy új betűkészletet töltünk vele a memóriába:

FF fontnév.font

Resident CLI L:Shell-Seg System Pure Add

A parancs már ismert, viszont a CLI paraméterével először találkozunk. Ez figyelmezteti a rendszert, hogy az új CLI vagy Shell ablak megnyitásakor ezt a rezidens utasítást használni kell. Részletesen olvashatsz róla a Shell-ről szóló fejezetben, a 62. oldalon.

Resident C:Execute Pure

Az Execute parancsot is rezidenssé tesszük.

Mount Newcon:

A parancs beépíti a rendszerbe a Newcon: egységet, ami a Shell ablak kezeléséhez szükséges.

Failat 11

Beállítja a hibahatárt, ami alatt még nem számít a hiba végzetesnek, nem kell miatta megszakítani a feldolgozást. Az Amiga DOS-ban minden hibának van egy kódja. Minél magasabb ez a kód, annál súlyosabb a hiba.

A paraméter nélkül alkalmazott Failat lekérdezi a jelenleg érvényes hibahatárt.

Run Execute S:StartupII

Ezzel a sorral a StartupII batchfájl párhuzamos feldolgozását indítjuk el. A feldolgozás multitasztkban folyik a Run miatt, tehát a Startup-sequence nem függesztődik fel.

Wait >NIL: 5 MINS

Ez a parancs várakozásra utasítja a batch programot a megadott ideig. Az időt háromféleképpen lehet megadni.

Másodpercben, jelölő kulcsszó ekkor SEC vagy SECS

Percben. A kulcsszó MIN vagy MBVS

Megadott időpontig, A kulcsszó UNTEL és az időpont előtt áll.
Például:

WaitUntil 12:42

A multitaskos programindítás és az indító taszk várakoztatása itt csak a demonstrációt szolgálja. Mielőtt továbbmennénk, nézzünk bele a párhuzamosan futó másik batch fájlba is:

Az első hat sorban rezidenssé teszünk néhány fontos utasítás, néhányat csak azért, mert ez a program többször használja őket, így időt takaríthatunk meg. A következő hat sorban létrehozuk a RAM:-ban a T, ENV és Clipboards tartalomjegyzékeket, majd hozzá is rendeljük őket a T:,ENV: és CLIPS: logikai egységekhez.

A következő három sorban mountáljuk a SPEAK:, AUX: és PIPE: készülékeket. Ezt akkor is megéri megtenni, ha kicsi az esély használatukra, mivel a Mountlistben nincs benne a Mount= 1 sor, a kezelőprogramok nem töltődnek be, a mountálás alig foglal memóriát.

Az utolsóelőtti három Resident eltávolítja azokat az utasításokat, amelyekre csak ennek a fájlnak volt szüksége.

A program utolsó parancsa

Break 1 C

még ismeretlen, itt az ideje, hogy szóljak róla. Már találkoztunk a programmegszakítással, amikor a Startup-sequence végrehajtását megszakítottuk a CTRL+C billentyűk együttes lenyomásával. Általában így lehet megszakítani a programokat, de létezik még CTRL+D, CTRL+E és CTRL+F billentyűkkel kiváltott megszakítás is, ezek mind különböző szintűek. A billentyűzetről kiadott megszakítások akkor működnek, ha a program ablaka az aktuális. A Break egy másik taszk programját szakítja meg, úgy, mintha annak ablakában használtad volna a fent említ-

tett billentyűkombinációk valamelyikét A dolog nagyon egyszerű, csak meg kell adni a taszk számát (a promptban benne van) és hogy C, D, E, vagy F megszakítást akarunk-e kiváltani.

Ebben az esetben tehát az 1-es taszk felé adunk CTRL+C megszakításkérést, ami a Wait utasítás megszakítását okozza. Ez az oka, hogy hosszú, öt perces időt adtak meg ott a várakozásnak. Mint említettem, ebben a formában csak demonstrációs célt szolgál ez a megoldás.

Most, hogy a Wait megszakadt, tovább folytatódik a Startup-sequence. A következő utasítás, a

Sys:System/Setmap d

sintén ismerős, a német billentyűzetkiosztást tölti be.

Path RAM: C: Sys:Utilities Sys:System S: Sys:Prefs Add

Meghatározzuk a keresési utakat. Egy parancs kiadása után az aktuális tartalomjegyzéket, majd sorban ezeket a tartalomjegyzéket nézi át a rendszer, és legvégül a C:-t is. Az, hogy a C: a parancs paraméterei közt is szerepel, annak az az oka, hogy általában a parancsok a C:-ben vannak, de mivel alapesetben ezt csak utoljára nézi át a rendszer, a többi tartalomjegyzék vizsgálata lassítja a keresését. A módszer megszívlelendő, ha saját Startup-ot írsz és abban alkalmazod a Path-ot, a keresési útvonalakat a gyakoriságuk sorrendjében add meg.

Loadwb Delay

Ez a parancs indítja el a Workbench kezelését, nélküle az nem működik. Rendszerint már a startup-ban kiadjuk, de elképzelhető egy olyan felállítás is, hogy egyáltalán nem működik a Workbench, csak a Shell, vagy CLI ablakok.

A Delay hatására a parancs vár kb. 3 másodpercet, mielőtt tovább fut a batchfájl.

Van a LoadWB-nek egy másik opciója is, a -Debug. Ennek hatására egy új, fejléc nélküli menüpont jelenik meg a Workbenchen, amelynek két pontja van, a Debug és a Flushlibs. Ezek a

felmerült hibák (nem DOS, hanem rendszer szintűek) kezelését lennének hivatottak szolgálni. Gyakorlatilag azonban az 1.3 kicstart alatt még nem tökéletesen működnek.

Azok a batch fájlok, amiket eddig láttunk, egyszerű, szekvenciális fájlok, azaz nem tartalmaznak elágazást, vagy feltételvizsgálatot, a parancsok egymás után sorban hajtódnak végre. Ismerkedjünk meg azokkal a módszerekkel, amivel a fentieket is létrehozhatjuk. A feltételvizsgálatok egy részének az átadott paraméter az alapja, kell tehát valami módszer, amivel azt átvehetjük. Ilyen és ehhez hasonló célt szolgálnak a pont parancsok, más néven a direktívák. Ezek nem valódi parancsok, csak az Execute ismeri őket. Nevüket onnan kapták, hogy mindig ponttal kezdődnek.

Ha egy fájl tartalmaz pont parancsot, akkor első sorában is annak kell állni. Lássuk milyen pont parancsok vannak:

.Key vagy **.K** - Az utána felsorolt szavak változóként viselkednek és átveszik a paramétereket. Ha kevesebb paraméter van, mint ahány változó, lesznek inicializálatlan változók, ezek olyanok, mintha nem is léteznének. Ha valamelyik változónak föltétlen szükséges paramétert átvennie tegyél utána egy /A jelzést, például: `.Key óra/A`. Ekkor, ha neki nem jutna paraméter, a program megáll és "elégtelen paraméter" hibaüzenetet ad. A Key direktíva csak egyszer fordulhat elő egy programban.

.Def - Változót definiál. Az a változó, amely a .Key direktívában paramétert vett át, így nem kaphat új értéket. Csak olyan változó definiálható, amely az előtte lévő .Key direktívában fel lett sorolva. Tipikus alkalmazása, amikor a .Key inicializálatlanul maradt változóinak alapértéket adunk.

Az előbbi két paranccsal létrehozott változókat később a kisebb-nagyobb jelek közé téve használhatjuk fel. Például:

Echo "A változó értéke:" <változó>

Van egy speciális változó, amely a programot futtató taszk számát tartalmazza. Ez a <\$\$>, de egy paranccsal megváltoztatható:

.dol - Megváltoztatja a dollárjelet, helyette ezentúl más használhatunk. Van még pár, a használt jeleket megváltoztató parancs.

.dot - A pontot cseréli ki megadott **jelre**:

•bra - A < jelet cseréli fel.

.ket - A > jelet cseréli fel.

Mind a négy megváltoztató parancs, csak az utána következő programrészre hatásos. Például:

```
Echo <változó>
```

```
.bra{
```

```
Echo {változó>
```

A zárójeleknek használt relációs jelek megváltoztatására akkor lehet szükség, ha a programban átirányítást is alkalmazunk, hisz annak ugyanez a jele. A batch fájlba üres sort is szúrhatunk a sor elejére tett ponttal, de ennek nincs sok jelentősége, hisz üres sort nélküle is lehet hagyni. Ha a sor elején álló pontot egy szóköz, majd tetszőleges karaktersorozat követi, azt figyelmen kívül hagyja az Execute, megjegyzésnek tekinti.

Megjegyzést más módon, a pontosvessző alkalmazásával is vihetünk a parancsok közé. Ebben a formában az bárhol állhat, akár egy parancs után is. Bármelyik módszert alkalmazzuk, a pont vagy a pontosvessző után következő összes karakter abban a sorban, megjegyzésnek számít, a következő parancsot új sorban kell kezdeni.

A dollár jelnek van egy különleges alkalmazása is, amikor egy változó helyett alternatívát jelöl. Ha a változó inicializálatlan, még sem a .Key, sem a .Def parancsokkal nem kapott értéket, helyette az alternatíváját használjuk. Például:

```
.Key port
```

```
Echo "Az TEXT fájlok kiírása a < port $ PRT:> egységre"
```

```
Copy #? .TEXT < port $ PRT:>
```

Ez a kis program kimásolja az aktuális tartalomjegyzékből a TEXT kiterjesztésű fájlokat a nyomtatóra, vagy a paraméterként megadott egységre: Például, ha a batch fájl neve kiír:

Execute kiír

Kiírás a nyomtatóra.

Execute kiír RAM: Kiírás a RAM:-ba

Az előbbi példa még nem igazi feltételvizsgálat, de azt is lehet csinálni. Tegyük fel, hogy olyan Startupot akarsz írni, amelyik létrehozza a RAD:-ot, majd a legfontosabb parancsokat oda másolja. A következő bootoláskor viszont már nem kell a RAD:-ot mountálni, sőt az akkor "az egység már létezik" hibajelzést váltaná ki. Először tehát meg kell vizsgálni, hogy a RAD: létezik-e, ha nem, akkor létre kell hozni.

A feltételvizsgálat parancsa az if, amihez a feltételel csatolni kell a következő módon:

```
If feltétel
    feltételes parancsok
Endif
```

vagy

```
If feltétel
    feltételes parancsok
Else
    ellenkező feltételű parancsok
Endif
```

Ha a feltétel megfelel az elvárásoknak, az If-et követő parancsok hajtódnak végre. Az első esetben az Endif zárja le a feltételes parancsok listáját. Ha a feltételvizsgálat nem az elvárt eredményt hozza, a program az Endif-et követő sorral folytatódik.

A második esetben, ha a feltétel igaz eredményt hoz, az If-Else parancsok közötti lesz, haOUih végre, cllcuXczú cselben az Else-Endif közötti.

A feltételek a következők lehetnek:

Exists név - A vizsgálat akkor ad igaz eredményt és hajtódik végre a feltételes ág, ha a megadott nevű fájl létezik. A fájlhoz megadható az elérési útvonala is, ha az nem az aktuális tartalomjegyzékben van, sőt, akár egységek léte is vizsgálható.

Például:

```
If Exists RAD:
    Echo "A RAD: létezik"
Else
    Echo "A RAD: nem létezik"
Endif
```

szövegl EQ szöveg2 - Akkor ad igaz eredményt, ha a két szöveg azonos.

szövegl GT szöveg2 - Akkor igaz, ha a szövegl nagyobb mint a szöveg2. Az összehasonlítás az ASCII kódok alapján történik. Például b nagyobb, mint a. A függelékben találász ASCII táblázatot.

szövegl GE szöveg2 - Nagyobb vagy egyenlő feltétel, az előző kettő kombinációja.

Az EQ, GT és GE feltételvizsgálatokban nem csak szövegeket lehet összehasonlítani, hanem a Val kulcszóval számokat, vagy a dollár jellel környezeti változókat is. Például:

.Key érték/A; muszáj megadni valamit

```
If Val <érték> EQ 0
    Echo "Nullát adtál meg"
Else
    Echo "Nem nulla"
Endif
```

vagy

.Key érték/A
Echo >ENV:ert <érték>. Az ert környezeti változó létrehozása. A

ADÓS

környezeti váltzók az ENV:-ben lévő szöveg fájlok.

```
If Sert EQ Text
```

```
    Echo "Azt írtad, hogy Text"
```

```
Endif
```

A Val és a \$ kombinálható is:

```
IfVal Svait EQ 15
```

```
    Echo "A vált környezeti változó értéke 15"
```

```
Endif
```

Sőt, a feltétel negálható is:

```
IfNotValSertEQIő
```

```
    Echo "Nem azt írtad, hogy 15"
```

```
Endif
```

Warn - Akkor igaz, ha az előző parancs által adott visszatérési kód legalább 5

Error - Hasonló az előzőhöz, de a returnkódnak legalább 10-nek kell lenni.

Fail - Mint az előző kettő, de az igaz értékhez a returnkódnak minimum 20-nak kell lenni.

Az idáig felsorolt feltételek mindegyike megfordítható, negálható az elébük írt Not kulcsszóval. Például:

```
If Not Exists RAD:
```

```
    Echo "A RAD: nincs meg"
```

```
Endif
```

Még egy lényeges dolog, az If szerkezetek egymásba ágyazhatók, minden ilyen If saját szintet képvisel a saját Else és Endif parancsával együtt.

Ezek ismeretében megírhatjuk azt a programrészt, amely megvizsgálja a RAD: létezését, és ennek megfelelően dönt:

```
If Not Exists RAD:
```

```
    Mount RAD:
```

```
Endif
```

Próbáld is ki a műveletet. Egy kellemetlen dolog fog bekövetkezni, a program leáll és egy kérdező felszólít, hogy tedd be a RAD nevű lemezt valamelyik meghajtóba. Ez nem nagy probléma, válaszolj Cancel-t és minden úgy működik tovább, mint azt elképzeltük. Hogyan lehetne azonban elkerülni ezt az incidenst? Van rá mód, az Assign parancs és az Exists opciója. A parancs 0 kóddal tér vissza, ha a specifikált egység létezik, S-tel ha nem. Ha már úgylis átdolgozzuk a batch programot, tegyünk bele egy részt, ami megvizsgálja, hogy a legfontosabb parancsok meg vannak-e a RAD:C-ben, ugyanis hiába másoltuk oda őket az előző bootolás után, később le is törölhattük őket

```

Failat 10; Beállítjuk a hibahatárt 5 fölé
Assign >NIL: RAD: Exists; A RAD: létezik-e?
If Warn; Ha nem
    Mount RAD;; megcsináljuk
    Skip RADC; ugrás subrutinra, a program a LAB
RADC sortól folytatódik.
    Copy C: parancs RAD:C; A fontos parancsok
felmásolása
    Copy...
Else; Ha a RAD: már él
    Skip RADC; Subrutin
    Copy C:If RAD:C
    Copy C: Endif RAD:C
    If Not Exists RAD:C/parancs; Ha a parancs még
nincs meg
        Copy C: parancs RAD:C; most felmásoljuk
a RAD:C-be
    Endif
    If Not Exists...
        Copy...
    Endif
Endif
Quit; Kilépés a batch programból

```

A DOS

Lab RADC; RADC címke definiálása, ide ugrunk a Skip-pel.

If Not Exists RAD:C; Ha nincs a RAD:-on C könyvtár

Makedir RAD:D; létrehozuk azt

Endif

Path RAD:C Add; A RAD:C-t kijelöljük keresési útnak

Copy C: Copy RAD:C; Ezután már a

CD RAD:C; RAD:C-ből használjuk a Copy-t.

Endskip; Vissza oda, ahonnet a RADC-re ugrottunk.

Az előző program parancsai jórészt ismerősek, de találhatók közöttük újak is. A program a megértését elősegítendő, bőségesen el van látva megjegyzésekkel, ezeket nem szükséges begépelni. Vegyük sorba az új parancsokat:

Skip címke (Back)

A parancs hatására a program nem a következő, hanem a Lab címke utáni sorral folytatódik. A Back opciót nem szükséges megadni, ha a megadott címke a listában később következik. Ennek az a hatása, hogy visszafelé is keresi a címkét, de csak a legközelebbi Execute parancsig.

A címkében a Lab-ot nem kell megadni. Ha nincs ilyen címke, az hibaüzenetet vált ki.

Lab

A Skip-pel összefüggésben alkalmazható, címkét definiál, ahová a Skip ugrani fog.

Quit (kód)

Kilépés a batch programból. Ha megadod a "kód"-ot, ilyen hibakóddal tér vissza, ha elmarad, nullával. A példában azért kellett használni, nehogy a végrehajtás hívás nélkül a Lab RADC-re kerüljön.

Most nézzünk egy olyan programot, ami ciklikusan ismételi valamit, annyiszor, amennyit paraméterenként megadsz:

.; Ha direktíva van, mindig ponttal kezdődik az első sor

```
Failat 11
.Key Loop/A; Valamit meg kell adni
If Val 0 GE <Loop>; Ellenőrzés
    Echo "Nullánál nagyobb számot adj meg
ismétlésszámnak!"
    Quit 20
Endif
.Bra{
.Ket}
Echo >ENV: Loop {Loop}; Loop környezeti változó
létrehozása
Lab Start
Echo "számláló állás:" Noline
Type Env: Loop; Loop környezeti változó kiírása
Eval <ENV:Loop >NIL: To=T:ABC{$$} Value 2=1 Op=-?
Loop csökkentése 1-el.
Copy T: Abc {$$} ENV:Loop frissítése
If Val $ Loop GT 0; Ha nem nulla
    Skip Start Back; Vissza, folytatni
Endif
Echo "Kész"; Vége
```

A programban most is találunk új parancsokat, de nézzük sorban az egészet.

Az első sorban azért kell a pont, mert ha a batch fájl pont parancsot tartalmaz, az első sorban is állni kell egynek. Miután leellenőrizte a program, hogy a paraméter nagyobb mint 0, megváltoztatja a < és > jeleket. Erre azért van szükség, hogy a változó jele ne tévesztődjön össze az átirányítás jelével. Az **Echo>ENV:Loop {Loop}** sor a Loop-ban lévő paramétert az ENV: logikai egységbe írja, létrehozva ezzel a Loop környezeti változót. Ezek tulajdonképpen egyszerű szöveg fájlok, amiket \$

ADÓS

jellel fel lehet használni a programban. Mivel ezeket az ENV:-en keresztül más program is elérheti, akár üzeneteket is küldhetnek futó batch programok egymásnak.

A környezeti változók kezeléséhez tartozik két parancs, amelyek ebben a példában nem szerepelnek, de érdemes megismerkedni velük.

Getenv változó

Kiírja a "változó" környezeti változó tartalmát. Az elérési utat nem kell megadni, mert a változók mindig az ENV:-ben vannak. Tulajdonképpen azonos a Type Env: változó paranccsal.

Setenv változó érték

A "**változó**" környezeti változónak az "**érték**" értéket adja. Ha az "érték"-et nem adod meg, a változó tartalma törlődik (A változó nem!).

A **Lab Start** címkétől kezdődik maga a számláló ciklus. Először az Echo-val kiírjuk a kommentárt, majd a kurzort a sor végén hagyva, a Type paranccsal utána írjuk a környezeti változót tartalmazó fájlt.

A Type egy új parancs, ismerkedjünk meg vele. Lényegét tekintve hasonló az Echo-hoz, de nem közvetlenül megadott szöveget, hanem egy fájl tartalmát írja ki a képernyőre, vagy a megadott egységre. Általános alakja a következő:

Type (From) név ((To)cél) (Opt opció)

A "**név**" a kiírandó fájl neve, a "**cél**" azé az egysége és névé, amibe írni kell. Ha ez elmarad, a kiírás az elsődleges output egységre, általában a képernyőre megy.

A parancshoz két opció használható, az N, minek hatására a sorok sorszámozva íródnak ki, és a H, ami hexadecimális kiírást okoz. A következő parancs szintén olyan, amivel még nem volt szerencsénk találkozni.

Az Eval matematikai műveleteket végez el, egész számokon. Általános alakja:

Eval op1 művelet op2

Az "op1" és "op2" a két operandust jelöli, a művelet pedig a köztük végrehajtandó matematikai művelet jele. A következő műveletek használhatók:

- + összeadás
- kivonás
- / osztás
- * szorzás
- & logikai ÉS
- | logikai VAGY
- ~ logikai NEM
- mod egész osztás
- Xor kizáró vagy
- Eqv egyenlőség
- « eltolás balra

A To opcióval az eredmény nem a konzolon, azaz a képernyőn jelenik meg, hanem átirányul a megadott egységre.

Az Lformat opcióval a kiadást formázhatjuk, sőt, a szabvány decimális megjelenítést is megváltoztathatjuk oktálisra vagy hexadecimálisra. Például:

Eval 3*7 Lformat "Eredmény: %N *N"

A %N a művelet eredményét a *N pedig a soremelést jelenti. Az operandusok nem csak decimálisak lehetnek, hanem oktálisak és hexadecimálisak is. Az oktális (nyolcas számrendszerbeli) számok elé O betűt, a hexadecimális (tizenhatos számrendszerbeli) elé pedig # jelet kell írni. Például:

```
O1472
#FA
```

Ezekén kívül van három opció, amelyek lényegi változást nem okoznak. A Valuel és a Value2 a két operandust jelenti, az Op pedig az operandust, ezek értékadásban szerepelhetnek. Használatuk akkor válik szükségessé, amikor a példában is szereplő

bonyolult formátumot alkalmazzuk, ekkor egyértelműsíteni kell, hogy melyik adat micsoda.

Nézzük végig ennek a formának az elemeit A parancs után az első operandus helyén mindjárt egy fűrcsaság áll. A "<" jel az átirányítás jelének a ">"-nek a fordítottja, hatása is ilyen, az utána megadott fájlból kell venni az adatokat. Az első operandus tehát a Loop környezeti változó. Jogos a felvetés, a környezeti változó tényleg text fájl és nem matematikai érték, ellenben ha a szöveg egy számnak vehető karaktersorral kezdődik, azt felismeri a parancs, de csak az egész részét

A következő szokatlan dolog, hogy átirányítunk valamit a NIL:-re. Akkor szükséges az Eval-ban, ha mint az előbb, valahonnan olvastuk az operandust, mert ekkor az Eval küld egy figyelmeztetést, hogy nem igazán ezt várta, de azért megérti. Hogy a figyelmeztetés, kioktatás ne csúfoskodjon a képernyőn, átirányítjuk a falrahányt-borsó művekbe.

A következő a To kulcszó már ismerős, itt éppen a T:-be küldi a számítás eredményét A fájl nevében a {\$\$} jel a programot futtató ablak számát helyettesíti be, nehogy egy másik ablakban futó program azonos nevű környezeti változójával összekeveredjen. A Valne2 és az Op használata egyértelmű, a kérdőjel pedig a művelet megkezdését indítja, gyakorlatilag idáig tart az Eval sora. Akkor kell használni, ha nem az alapformát alkalmazzuk.

A következő sorban a T: átmeneti egységből átmásoljuk az ideiglenesen létrehozott fájlt az Env:-be. Loop néven, ezzel frissítve a környezeti változót.

Miért nem lehet rögtön az ENV:Loop-ba írni az eredményt, miért kell a kerülőút? A Loop fájlt a parancs során megnyitottuk, használatban van, tehát megváltoztatni, törölni nem lehet. A parancs végrehajtódása után a fájl bezárul, már nincs adadálya a felülírásnak.

Ezután leellenőrizzük, hogy a számláló értéke nullára csökkent-e. Ha még nem, visszaugrunk a Start címkére. Ha a ciklus a szükséges számban lefutott, kiírjuk hogy kész. Idáig csak a pa-

rancsadáskor adtunk át paramétert a batch programoknak, de van lehetőség a program futása közben is. Például:

```
Failat 10
If Not Exists RAD:
    Echo "A RAD: nincs Mountálva."
    Ask "Akarod megtenni? Y/N"
    IfWarn
    Mount RAD:
Endif
Endif
```

Az Ask parancs kiírja az idézőjelbe tett szöveget és válaszra vár. A válasz vagy az Y vagy az N betű lehet Ha Y(yes) a válasz, a return kódja 5, ha N (No) akkor 0.

A következő If-fel megvizsgálhatjuk a választ Fontos, hogy az Ask-et közvetlenül kövesse az If, mert egyébként nem a válasz returnkódját fogja vizsgálni.

5. Batch programok a Workbench lemezen

Miután a batch programok írásához elegendő alapismereteket szereztünk, nézzünk meg néhány, a Workbench-en lévő ilyen programot amelyek az S: tartalomjegyzékben vannak:

PCD

A CD-hez hasonló hatású program, de megjegyzi az előző könyvtárat amibe visszalép, ha paraméter nélkül használod. Például:

```
PCD RAM
PCD
```

```
SPAT
```

A More vagy Type parancsok argumentumaiban nem alkalmazható joker, a fájl teljes nevét ki kell írni, ezt hidalja át a program:

SPATTypeS:#?

DPAT

Hasonló az előzőhöz, de két argumentumos utasításokhoz, mint például a Copy vagy Rename:

DPAT Rename RAM:#? RAD:C

CLI-Startup

Shell-Startup

A NewCli és Newshell utasítások startprogramjai. Ezek az említett parancsok kiadásakor mindig végrehajtnak, érdemes átnézni őket és igény szerint alakítani rajtuk.

Startup-sequence.HD

A nem autobootos merevlemez startup-ja. Nem magától indul, az Execute paranccsal kell indítani. Az autobootos merevlemezek szintén az S könyvtárunkban lévő Startup-sequence fájlt indítják bootolás után.

6. Az AmigaDOS egyéb parancsai

Az Amiga-DOS-szal való ismerkedésünk végén röviden összefoglalom az előző fejezetek során nem említett parancsokat.

Changetaskpri

Minden taszknak van egy prioritása, amely meghatározza feladatának fontosságát. Minél nagyobb valamelyik prioritás a többiéhez képest, annál több időt kap a futtatáshoz. Ha nem arra az ablakra akarod vonatkoztatni, amelyekben használod, az ablak számát is meg kell adni. Használata:

Changetaskpi prioritás (taszk)

A prioritás a -128 +127 tartományban lehet, de 5-nél nagyobbat ne alkalmazz ha lehet, nehogy valami fontos rendszerfunkció emmiatt kevesebb időt kapjon.

Diskchange

Ha régebbi típusú 5 1/4 collos floppy meghajtót használsz, abban nem veszi észre a gép a lemezcserét, hanem azt ezzel a paranccsal kell tudtára hozni. Például:

Diskchange DF2:

Diskdoctor

Fogod tapasztalni, a lemezen az adatok megsérülhetnek, ilyenkor a Read-Write Error hibaiüzenetet kapod, azaz a gép nem bírja elolvasni a lemezen az adatokat. A lemezhiba nem minden esetben jár együtt az adataink elvesztésével, bizonyos esetekben kijavítható. A javítást végzi a Diskdoctor. Sajnos a parancs tartalmaz egy kis bug-ot, néha nagyobb kárt csinál, mint hasznot. Javaslom szerezd be helyette a Disk Repair-t, vagy egy hasonló lemezjavító programot.

Filenote

A Workbench ismertetésénél, az Info menüpontnál már volt róla szó, hogy a fájlokhoz megjegyzések fűzhetők. A Filenote is ezt teszi a DOS szintjéből. A kommentár legfeljebb 78 karakter lehet, ha szóközt is tartalmaz, akkor idézőjelbe kell tenni. A fájlhoz fűzött megjegyzést a List paranccsal lehet elolvasni. Alkalmazása:

Filenote fájl COMMENT megjegyzés

A COMMENT kulcsszó kötelező.

Lock

A merevlemez védhetjük meg illetéktelen felülírásoktól és törlésektől, ezeket csak az tudja a bekapcsolás (mármint a Lock bekapcsolása) után végrehajtani, aki ismeri a legfeljebb négy betűs jelszót. Használata:

Lock lemez ON/OFF jelszó

Az ON és OFF kulcsszavak közül értelemszerűen csak az egyiket kell használni. A védelem kizárólag FFS alatt él.

Prompt

Megváltoztatja a CLI vagy a Shell ablak promptját, ami alapesetben CLI vagy a Shell-ben ezen felül az aktuális könyvtár. Például:

Prompt "Saját Prompt>"

Két különleges karakter is használható, az egyik a %N, amely az ablak számát, a másik a %S, amely az aktuális könyvtárat jelenti. Ez utóbbi csak Shell-ben használható. Például a Shell eredeti promptjának a visszaállítása:

Prompt "%N.%S>"¹

Search

Egy megadott string előfordulását keresi a fájlokban. Ha megtalálta a szöveget, kiírja annak a sornak a számát, majd az egész sort Általános alakja:

Search ((FROM)fájl) SEARCH string

Sem a FROM kulcsszó, sem a fájlnev nem kötelező a SEARCH viszont igen. Ha nincs fájlnev, az aktuális tartalomjegyzékben mindent átnéz. A szöveget idézőjelbe kell tenni. A parancshoz használható néhány opció is.

Ali - A keresést kiterjeszti a tartalomjegyzékben lévő altartalomjegyzékekre is.

File - Nem stringet, hanem fájlt keres.

Nonum - Nem írja ki a sorszámot

Quick - Tömörebb, gyorsabb listát készít

Quiet - Csak annak a fájlnek a nevét írja ki, amelyben megtalálta a szöveget

A keresés a CTRL-D-vel megszakítható az adott fájlban, CTRLK-C-vel az egész parancs megszakad.

Setdate

Az Amiga nyilvántartja minden tartalomjegyzék és fájl létrehozásának idejét és dátumát, amit ezzel a paranccsal megváltoztathatunk. Használata:

Setdate név idő

A név a kívánt fájl vagy tartalomjegyzék neve, az idő pedig az új dátum és időbejegyzése. Ha csak az időt adod meg, a dátum változatlan marad. Fordított esetben az idő az éppen aktuális értéket veszi fel.

Sort

Egy szövegfájl sorait ABC-be rendezi, a rendezett fájlt ki is menti. Általános alakja:

Sort (From) név (To) rendezett

A rendezett fájl nem lehet közvetlenül az eredetibe írni, annak más néven vagy más úton kell szerepelnie. A From és a To kulcsszó elhagyható. A parancsnak van egy opciója, amivel elérhető, hogy a rendezés ne az első betűtől kezdődjön. Ebben az esetben a COLSTART után meg kell adni az első figyelembe veendő oszlop számát.

Sort DFO:nevek PRT: COLSTART 4

Stack

Lekérdezi vagy beállítja a számítógép munkamemóriáját, ami alapesetben 4kB. A lekérdezéshez a parancsot paraméter nélkül

kell használni, beállításához meg kell adni az új méretet bájtban.
Például

Stack 8000

Status

A Shell vagy CLI ablakokban futó programokról nyújt információt. Ha megadjuk az ablak számát, csak arról ad infót, ha nem, akkor mindenről.

A FULL vagy a TCB opció hatására az ablak(ok) stack méretéről, globális vektoráról és prioritásáról is tájékoztatva leszünk.

Which

Hasonló, mint a Search File Ali, megadja egy fájl elérési útvonalát. Két opciója közül a RES hatására csak a rezidens parancsok listáját nézi át. A NORES ennek fordítottja, megakadályozza a rezidensek közt a keresést

7. Tippek és Trükkök

Elérkeztünk az 1.3 Workbench és DOS leírásának a végére. Befejezésül néhány tanács:

Bizonyára észrevetted, hogy a nyitott ablakok memóriát igényelnek. Ráadásul a sok nyitott ablak között sokszor nem találja az ember, amit keres. Mit tegyünk, ha mégis különböző tartalomjegyzékekben lévő programokat szeretnénk használni? A megoldás nagyon egyszerű, a használni kívánt ikonokat fogd meg és húzd ki az ablakból, olyan helyre, amely egyik ablakba sem esik. Miután a kívánt ikonokat kiraktad, bezárhatod az ablakokat, az ikonok ezután is a Workbench screen-en maradnak, minden korlátozástól mentesen használhatod őket. Amellett, hogy ez a megoldás kényelmessé teszi a munkát, memóriát is takarít meg. **Nemcsak program, hanem bármilyen ikon kiteheti!, például a fiókoké is.**

Sajnos ezt a műveletet minden bootolás után meg kell ismételni, mert az ikon nem jegyzi meg, hogy ki lett rakva.

Ha sok ablak van nyitva, amelyek átfedik egymást, némelyik nem is látszik, nehéz megtalálni a keresettet. Ilyenkor klikkelj duplán a kívánt ablak ikonján, az előtérbe kerül. Ugyanaz az ablak ikon nem lehet egy időben kétszer nyitva, viszont program bármennyi példányban futhat, eltekintve néhány kivételtől.

Tapasztalhattuk, az ikonok milyen fontos dolgok, a fájlok Workbenchen történő kezeléséhez egyenesen nélkülözhetetlenek, mégsem rendelkezik minden fájl vele. Hogyan adhatunk egy olyan fájlnak ikont, amelynek eddig nem volt? Erre több mód is kínálkozik. A legegyszerűbb, ha a projectet létrehozó program ikonnal együtt menti ki azt.

A másik, hogy az IkonEd-del tervezünk neki egyet. Ez kissé nehézkes és nem is minden esetben szükséges eljárás.

A harmadik módszer egy trükk, másik, ikonnal rendelkező program info-ját rendeljük hozzá. Ehhez először azt az info-t át kell nevezni, de úgy, hogy az eredeti is megmaradjon, majd az így létrehozott info-t megfelelően beállítjuk.

Nyissunk egy Shell, vagy CLI ablakot, majd a kiválasztott info-t, amelynek típusa azonos az ikonnal ellátandó fájllel (tartalomjegyzéknek Drawer, programnak Tool, stb ikon dukál), be kell másolni mondjuk a RAM:-ba. Itt át kell nevezni a "Célfájl.info" névre - "Célfájl" természetesen az ikonozandó fájl neve -, majd bemásolni abba a tartalomjegyzékbe, ahol az a fájl van. Az info tartalma ezután az ismert módon beállítható.

A Workbench lemezen sok olyan fájl van, amelyre nem biztos, hogy szükség lesz. Hozz létre egy saját munkalemezt, másold át azokat a fájlokat, DOS parancsokat, amelyeket használni akarsz, a lemezen fennmaradó helyre pedig másolj felhasználói programokat.

Ha nem vagy biztos egy parancs paramétereinek a szintaxisával, megkérdezheted tőle azt, így:

Parancs ?

A parancs ennek hatására infot ad önmagáról, megjeleníti az egyes opcióit, amik használatát /x-el jelöli. Az x helyén valamilyen betű áll, amely az adott paraméterről informál. Ez a következők valamelyike lehet:

- /A - Az így jelölt argumentumot kötelező megadni, nélküle a parancs nem működik.
- /K - Kulcsszó, amennyiben szükség van rá, szó szerint kell megadni.
- /M - Sablon, ide a felhasználó adhat paramétert, például egy fájl nevét.
- /N - Numerikus paraméter.
- /S - Kapcsoló, az opciót szó szerint megadva valamilyen igen-nem állapotot kapcsol be.

Ha két floppy meghajtód van, a boot lemezre vedd fel az összes elérhető library-t, device-t handlert, DOS parancsokat, az általad használhatónak tartott fontokat, stb, és ezt a lemezt, ha csak lehet, tartsd mindig a gépben. A másik drive-be tedd a felhasználói programokat tartalmazó lemezt. Az ezekről indított programok mindig a boot lemezeden keresik majd a működésükhöz szükséges dolgokat, nem kell cserélni a lemezeket.

Ha merevlemez vásárlására szánod rá magadat, mindig részesítsd előnyben az autobootos típusokat. Lehetőleg SCSI egységet vegyél, vagy ha kevesebb a pénzed, AT kontrolerest, de kerülj az XT buszosokat. Ezek már elavult, lassú konstrukciók. Amennyiben lehet memóriabővítést, vagy legalábbis annak opcióját tartalmazó vezérlőt válassz!

A merevlemezen hozz létre egy kb. 5 MB méretű boot partíciót, ezen elfér a rendszer. A fennmaradó részt annyi darabra

particionálhatod, amennyire akarsz, de a nagyon kis partícióknak nincs értelme.

A rendszer külön partícióba tételének több előnye is van. A legfontosabb, hogy adathiba vagy más katasztrófa esetén a másik partíción lévő adatok még akkor is sértetlenül megmaradnak, ha a boot részt újra kell formázni.

Ha van nyomtató, egyszerűen használhatod írógépnek. írd be egy Shell, vagy CLI prompt után:

Copy" To PRT:

Ezután amit gépelsz, az a Return lenyomása után kiíródik a printeren. Az írógép módból a CTRL+X billentyűk együttes lenyomásával léphetsz ki. Az Amiga rendszerfelépítéséből adódóan természetesen nem csak a nyomtatóra, hanem fájlba, vagy valamelyik portra is küldhető az output.

Az egér alá tegyél mous pad-ot, ez megvédi az idő előtti elkoszolódástól. Hiába vagy azonban elővigyázatos, a használat során az egér is szennyeződik, ezt abból veszed észre, hogy a mutató nem mindig követi az elképzelt mozgáspályát. Ez gyorsan megszüntethető, fordítsd hanyatt a kis jószágot, a hasán lévő korongot forgasd el az óramutató járásával ellenkezőleg, hogy a nyíl az Ó-ra mutasson, majd emeld fel a fedelet.

Mostmár kiveheted a golyót, amit langyos, szappanos vízben moss meg. Az egér belsejét alkoholos ruhával töröld ki, végül szereld össze. A hatás azonnali.

Az AURUM DTP Stúdió

vállalja

- reklámnyomtatványok
- dokumentációk, termékismertető
- alkalmi és rendszeres kiadványok
- újságok, folyóiratok
- könyvek

szerkesztését a szedéstől a nyomdai kivitelezésig,
gyorsan, rugalmasan.

Kérjen árajánlatot, garantáltan mi vagyunk a
legolcsóbbak.

AURUM DTP Stúdió

5430, Tiszaföldvár

Ószőlő Fő út 64.

Telefon : 307

III. A 2.0-s kickstartú gépek kezelése

A 2.0-s és magasabb verziójú (például 2.1) Workbench működéséhez szükség van az új, 2.0 kickstartra. Ez az 500+, 600 és 3000 típusokba gyárilag be van építve, de az 500 és 2000 modellekbe is beszerelhető utólag. Az új ROM mindössze néhány ezer forintba kerül, ennyit érdemes rá áldozni.

A könyv elején említettem, hogy az új kickstart RAM-ba is tölthető, ha az nincs a gép ROM-jában, vagy csak egyszerűen gyorsítani akarod a gépet. A RAM elérése ugyanis jóval kevesebb időt vesz igénybe mint a ROM-é.

Az A3000-hez mellékelnek kickstart lemezt arról az betölthető, de ez nem jó az A500-hoz, mivel az nincs felkészülve a kezeléséhez. Léteznek olyan kickstart lemezek, amelyek direkt 500-as és 2000-es gépek tulajdonosai számára készültek. Ezek tartalmazzak egy betöltő programot is.

Mivel a kickstart mérete fél megabájt, az alap A500-on ez nem járható út, de még egy megán sem rejt sok élvezetet. A másik bökkenő, hogy ilyen típusú kickstart lemezen én még csak béta tesztos kick verzióval találkoztam, de nincs kizárva, hogy a végleges tesztelt ROM is megjelent ebben a formában.

Az új rendszer rengeteg pluszt nyújt az 1.3-hoz képest, amelyek jóval felhasználóbarátabb géppé teszik az Amigát, mint egy PC, a Windows 3.1-el.

A könyv 2.0 kickstartról és Workbenchről szóló része az előző leírásokra épül, azoknak is érdemes tehát átolvasni azokat, akik már az új ROM-ot tartalmazó gépen kezdik Amigával való ismerkedésüket.



III. A 2.0-s kickstartú gépek kezelése

A 2.0-s és magasabb verziószámú (például 2.1) Workbench működéséhez szükség van az új, 2.0 kickstartra. Ez az 500+, 600 és 3000 típusokba gyárilag be van építve, de az 500 és 2000 modellekbe is beszerelhető utólag. Az új ROM mindössze néhány ezer forintba kerül, ennyit érdemes rá áldozni.

A könyv elején említettem, hogy az új kickstart RAM-ba is tölthető, ha az nincs a gép ROM-jában, vagy csak egyszerűen gyorsítani akarod a gépet. A RAM elérése ugyanis jóval kevesebb időt vesz igénybe mint a ROM-é.

Az A3000-hez mellékelnek kickstart lemezt arról az betölthető, de ez nem jó az A500-hoz, mivel az nincs felkészülve a kezeléséhez. Léteznek olyan kickstart lemezek, amelyek direkt 500-as és 2000-es gépek tulajdonosai számára készültek. Ezek tartalmaznak egy betöltő programot is.

Mivel a kickstart mérete fél megabájt, az alap A500-on ez nem járható út, de még egy megán sem rejt sok élvezetet. A másik bökkenő, hogy ilyen típusú kickstart lemezen én még csak béta tesztes kick verzióval találkoztam, de nincs kizárva, hogy a végleges tesztelt ROM is megjelent ebben a formában.

Az új rendszer rengeteg pluszt nyújt az 1.3-hoz képest, amelyek jóval felhasználóbarátabb géppé teszik az Amigát, mint egy PC, a Windows 3.1-el.

A könyv 2.0 kickstartról és Workbenchről szóló része az előző leírásokra épül, azoknak is érdemes tehát átolvasni azokat, akik már az új ROM-ot tartalmazó gépen kezdik Amigával való ismerkedésüket.



A. A Workbench2.0

Az első különbség már a bekapcsolás után rögtön látszik (hacsak nincs autobootos winchester a gépben), a Workbench disk behelyezését kérő kéz helyett egy színes animációt láthatunk. Ez a képernyő egy felirattal tájékoztat a gépben lévő kickstart verziószámáról.

Tegyük be a **Workbench2.0** lemezt, minek hatására a gép szabályosan bootolni kezd. Rövid idő múltán megjelenik a Workbench képernyője, amely már szemre is kellemesebb az előzőeknél. Előfordulhat, hogy a Workbench betöltődése egy **Keymap Selection** kezdetű szöveg kiírása után megáll. Ez abban az esetben történik meg, ha a **Devs/Keymaps** könyvtárba még nincs betöltve billentyűzetmeghajtó program. A szöveg arra kérdez rá, hogy akarsz-e betölteni egy ilyen programot az Extrás lemezről. Ha 1-el válaszolsz, választhatsz az Extráson lévő keymapokból, amit a program rámásol a Workbenchre, vagy 0-t nyomhatsz, minek hatására nem foglalkozik tovább a kérdéssel, folytatja a betöltést.

Ha jó az időérzéked, vagy mérted a betöltődéshez szükséges időt, észreveheted a következő különbséget, a Workbench2.0 felállításához kevesebb idő kellett mint az 1.3 esetében.

A gyorsaságnak két oka is van. Az egyik, hogy a 2.0 kickstart már a floppy diszkeken is az FFS (FastFileSystem) adatrendszert használja, ami a kezelés gyorsulása mellett kis mértékű kapacitás növekedéssel jár.

A régi rendszer, az **OFS**, ugyanis minden szektor 512 bájtból 6 lonword-öt, azaz 24 bájtot a lemezkezelés céljaira lefoglalt, így a hasznos blokkméret 488 bájtra csökkent. Az FFS ezt már csak a 880-as sávon, a tartalomjegyzéknél teszi, az adattárolásra használt többi szektor mind az 512 bájta a felhasználóé. A régi fájlrendszer alatt formázott lemezek is használhatók, ezeket természetesen OFS lemezként kezeli a gép.

A gyorsulás másik oka, hogy sok DOS parancs a ROM-ba van építve, rezidens parancsként áll rendelkezésre. Erről meggyőződhetsz, ha egy shell ablakot nyitsz és kiadod benne a Resident parancsot. A ROM rezidens parancsok az **INTERNÁL** státuszt viselik.

Említettem, hogy a Workbench ablaka szemre is kellemesebb. Ezt a 3 dimenziós kinézetű ablakoknak és gadgeteknek köszönheti elsősorban. Az alapszínek is megváltoztak, a képernyőn egy kellemes szürke dominál.

A menüpontok száma is öröndetesen megszorodott. Újdonság, hogy a fontosabb menüpontokat már jobb Amiga billentyű + egy betű kombinációjával is elérhetjük.

A **Workbench** menü a helyén maradt, de a Disk és Special menüket felváltották a **Window**, **Icons** és **Tools** menük.

A **Workbench** menü a Workbench általános kezeléséhez szükséges parancsokat rejti. Jóval több nevet találunk a legördülő listában mint az elődjénél. Nézzük most sorba ezeket.

Backdrop. A jobb Amiga+B-vel rövidíthető. Ha a menüpont ki van kapcsolva, a Workbench munkaterülete is ablakba kerül, gördítő nyilai, méretváltoztató és átkapcsoló gadgete van, stb. Ha ezt a pontot bekapcsolod, egy pipa jelenik meg előtte és eltűnik a Workbench kerete, de elveszíted a logikailag nagyobb képernyő nyújtotta előnyöket.

Egy később ismertetésre kerülő módon a Workbench screent nagyobbra is állíthatjuk, mint az egyszerre látható méret (lásd a Screen programnál a 160. oldalon) Csak nem? De igen, a 2.0-s kickstartól kezdődően már nagyobb lehet a Workbench mérete, mint a látható képernyő, nem kell tehát nyomorogni a kinyitott ablakokkal.

Kapcsoljuk most be a keretet és nézzük meg figyelmesen azt. Szembetűnő, hogy a Workbenchnek is van záró gadgetje, amivel Ke lek<<4 ceuleni a munWafia4al^ helyet enV>>Ewif4>> fel ajz önálló képernyőn futó programok számára. Ezzel kapcsolatban van egy

megkötés. A Workbenchet csak akkor lehet bezárni, ha nincs rajta futó program vagy nyitott képernyő. A bezárás végrehajtása előtt egy biztonsági kérdésre OK-val kell válaszolni. Az igen-nem válaszok itt is helyettesíthetők a bal Amiga+V vagy B kombinációkkal.

Tovább szemlélve az ablakot, észrevehetjük, hogy a gördítő jelek is megváltoztak kissé, a nyilak lekerültek mind a jobb alsó sarokba, ezáltal gyorsabban elérhető, nem kell sokat mozgatni az egeret egy finomabb ablakbeállításhoz. A görgetés egyébként teljesen azonos módon megy végbe, mint az 1.3 Workbenchnél.

A méretváltó jel a jobb alsó sarokban maradt, de formája kissé megváltozott.

Lényeges változás történt a jobb felső sarokban, a két előtérbe-háttérbe helyező gadget helyett csak egyet találunk. Ez úgy működik, hogy az előtérben lévő ablakot a háttérbe teszi, az ott lévő előtérbe hozza.

A mellette balról lévő jel új dolog. Rákattintva az ablak minimálisra zsugorodik. Ismételt klikkelésre visszaáll eredeti méretűre. Ha a minimalizált ablakot átméretezed, pozícióját megváltoztatod, azt megjegyzi, következő használatkor ezt a méretet veszi fel az ablak, erre a helyre ugrik el. Az átkapcsoló jelhez tartozik, hogy a bal Amiga+N és M kombinációkkal nem csak az előtérben lévő képernyő és a Workbench között lehet váltani, hanem sorban minden képernyő elérhető.

Van az ablakoknak egy jó tulajdonsága az új rendszer alatt: emlékeznek a tartalmukra. Ez annyit jelent, hogy ha az ablakot valamilyen módon összezsugorítod, majd ismét széthúzod, tartalma megmarad, nem törlődik. Hasonló dolog, hogy a méretükre és pozíciójukra is emlékeznek, ha elmozgatsz, majd bezársz egy ablakot, következőben ott nyílik ki, a becsukáskori méretben. Ez nem helyettesíti a Snapshot-ot, az új méret és pozíció nem íródik ki a lemezre!

Ennek demonstrálására nyisd ki a Workbench ikont és indítsd el a Shell-t. Kérj néhány tartalomjegyzéket a Dir vagy List pa-

A Workbench2.0

ranccsal. Zsugorítsd össze, majd húzd szét a Shell ablakot, az eredmény rögtön látható. Még az is megjelenik az ablakban, ha befér, amely már az összezsugorítás előtt sem volt látható.

A következő menüpont a **Workbench** menüben az **Execute Command**, amely a jobb Amiga+E billentyűzetkombinációval rövidíthető. Hatására megjelenik egy kérdező, ahol beírhatunk egy DOS parancssort, ami az OK megnyomása után végrehajódik, miközben a kérdező is eltűnik.

Azok a parancsok, amelyeknek képernyőre kerülő outputjuk van, mint például a **Dir**, **List**, **Type** stb, az outputot egy "**Workbench Output Window**" nevű ablakban jelenítik meg. Az ablakban nincs prompt, tehát parancs adására nem alkalmas, bezárni a bal felső sarokban lévő szokásos gadgettel lehet

A menüpont rendkívül hasznos pl. ikon nélküli programok indításakor, nem kell ugyanis elindítani egy **Shell-t**.

Tovább böngészve a menüpontok között, ráakadunk a **Redraw AU-ra**. Akkor kell használni, amikor valamely kérdező vagy ablak elrontja a képet. A menüpont hatására újrajzolóódik a képernyő.

Az **Update Ali** újraolvassa az ikonokat és fájlokat, visszaállítva azokat a **Snapshot-tal** rögzített pozíciójukba.

A **Last Error** ismerős az 1.3 Workbenchből, az utójára felmerült hiba megnevezését és kódját jeleníti meg.

A **Version** szintén nem új menüpont, a kickstart és a Workbench verziószámát jeleníti meg a fejléc sorban.

A Workbench menü utolsó pontja a **Quit**, amely a jobb Amiga+Q billentyűk kombinációjával is rövidíthető. Ennek hatása azonos a Workbench záró ikonjára történő klikkeléssel. A biztonsági kérdés itt is megjelenik, mielőtt kilépnénk.

A következő menü a **Window**. Ennek pontjai a kinyitott ablakok kezelését hivatottak elősegíteni. Hatásuk mindig az aktív ablakra vonatkozik.

Az első pont a New Drawer, amely a jobb Amiga +N billentyűk megnyomásával rövidíthető. Ez Unnamed néven új tartalomjegyzéket hoz létre. A nevet kiegészíti egy sorszámmal is. Az újonnan létrehozott fiók átnevezhető, más ablakba átrakható. Nincs tehát szükség az Empty fiókra és annak duplikálására (lásd a 34. oldalon).

A második menüpont, az Open Parent, akkor hasznos, ha egy több szint mélységben lévő ablak felmenőjét, az őt tartalmazó fiókot szeretnénk kinyitni. Abban az esetben is használható, ha a szülő már nyitva van, ekkor az előtérbe kerül hasonlóan ahhoz, mikor egy már megnyitott fiók vagy lemez ikonján ismételtent kattintasz (lásd a 133. oldalon).

A következő menüpont ismét egy ismerős. A Close, amely a jobb Amiga+K-val rövidíthető, az aktuális ablakot bezárja.

Az Update hasonló a Workbench menü Update Ali pontjához, a különbség, hogy csak a kijelölt ablakon hatásos.

A következő menüpont a Select Contents, amely a jobb Amiga+A billentyűk együttes lenyomásával rövidíthető, az aktuális ablakban lévő összes ikont kijelöli.

A Clean Up szintén egy ismerős menüpont (lásd a 55. oldalon), az aktuális ablak ikonjait rendezi el. Ha a Workbench ablaka az aktuális, akkor is használható, ebben az esetben a lemez meghajtók és kirakott programok ikonjait szedi sorba.

A következő pont a Snapshot, erről is volt már szó (lásd a 56. oldalon), de itt két alpontja van. Az egyik a Window, amely a kijelölt ablak pozícióját és méretét menti ki az ablakhoz tartozó infófájlba. A másik alpont az Ali, amely az ablak pozíciója és mérete mellett, az ablakban található ikonok elhelyezkedését is kimentí. Ez csak nyitott, kijelölt ablakokra hatásos. Ha ez az ablak maga a Workbench, akkor a rajta lévő ikonok mellett a nyitott ablakok mérete és pozíciója is elmentésre kerül.

A Show menüpont az új rendszer egyik fontos újításához kapcsolódik, nevezetesen, hogy nemcsak azokat a fájlokat tudja meg-

jeleníteni a Workbench, amelyeknek ikonjuk van, hanem azokat is, amelyeknek nincs.

A menüpontnak két alpontja van, az egyik az **Only Icons**, amely az alapesetet kapcsolja be, vagyis csak azok a tárgyak jelennek meg, amelyeknek ikonjuk van. A másik pont az **Ali Files** aminek bekapcsolásával az ikonnal nem rendelkező fájlok is megjelennek az ablakban.

Ezzel a menüponttal összefüggésben áll a következő, a View By, aminek négy alpontja közül választhatunk egyet. Az első alpont az **Icon**, ami ha be van kapcsolva, az ablakban az ikonok jelennek meg. Ha az előző pontnál az **Ali Files**-t választottuk, akkor az ikonnal nem rendelkező tartalomjegyzékek és fájlok, nekik megfelelő, uniformizált ikont kapnak. Ezek az ikonok nem hoznak létre info fájlokat, csak a megjelenítéshez vannak használva. Ennek ellenére ugyanúgy használhatók, mint a valódi ikonok, sőt az Information kérdező segítségével akár megjegyzést is fűzhetünk hozzájuk, ezt megjegyzi, még reszet vagy kikapcsolás után is. Persze ez nem nagy dolog, hisz a 78 karakter kommentet maga a fájl tárolja és nem az info (lásd a Filenote parancsnál, a 129. oldalon).

A második alpont a Name. Ha ez van bekapcsolva, az ablakban nem ikonok jelennek meg, hanem egy olyan lista, mint amit a List parancs is eredményez (lásd a 67. oldalon). A lista a fájlnevek alapján van rendezve. Ha nem férnek el egyszerre az ablakban, a gördítő nyilakkal és a gördítő sávval a kívánt részre állhatsz.

A további két alpont ugyanilyen módon jeleníti meg a fájlokat, de a rendezés alapja a **Date** esetében a létrehozási idő, a Size-nál pedig a méret.

Bármelyik listás megjelenítést is alkalmazzuk, a fájlok nevei, amelyek az első oszlopban találhatók, úgy viselkednek, mintha ikonok lennének, kijelölhetők, áthelyezhetők másik ablakba és **dupla klikkel, vagy az open-nel megnyithatók, sőt elindíthatók, ha programról van szó.**

Az ikonnal nem rendelkező programok Workbencből való indítása némileg máshogy megy, mint ikonos társaiké. Az ideiglenes ikonon, vagy lista esetén a néven történt dupla klikk hatására a már ismertetett Execute Command kérdezője jelenik meg, benne a fájl nevével. A billentyűzetről beírhatjuk az esetleg szükséges argumentumokat, majd a Return billentyű, vagy az OK kapcsoló lenyomásával hajthatjuk végre a parancsot vagy programot.

A Show menüpont beállítása a View By minden módjában hatásos, ha Only Icons van bekapcsolva a listában sem jelenik meg olyan fájl, amelynek nincs ikonja.

Az info fájlok egyik módban sem jeleníthetők meg.

A lista bármelyik elemét a névre történő klikkeléssel lehet kijelölni. A Shift billentyű használatával multiszelekt is megvalósítható. Sőt, ez a rendszer a többes kijelölésre bevezet egy új módszert is, a gumikeretet. Ez úgy működik, hogy az eger bal gombjának lenyomásával, annak folyamatos nyomvatartása mellett az egeret elmozgatva egy szaggatott keretű téglalapot rajzolunk. A bal gomb elengedése után a kereten belül lévő összes ikon vagy név kijelölődik, még akkor is, ha csak egy része lóg be a keretbe. Az így kijelölt ikonok, a Shift használatával mozgathatók el egyszerre. A Shift akkor is alkalmazandó, ha több, nem egymás mellett lévő ikoncsoportot akarsz kijelölni gumikerettel, úgy, hogy a köztük lévő ikonok ne választódjanak ki.

Hasonlóan az 1.3-as Workbenchez, bármennyi ikon kijelölhető egyszerre, de minig csak egy ablak lehet aktív. A Workbench következő menüje az Icons. Ebben az ikonok kezeléséhez találunk menüpontokat. Alapvető szabály, hogy mindig csak a kijelölt ikonra vagy ikonokra vonatkoznak. Nézzük sorba ezeket is.

Az első pont az Open, rövidíteni a jobb Amiga+O billentyűkombinációval lehet. Hatására a kijelölt lemez vagy fiók jellegű

ikonok ablaka kinyílik, a programok elindulnak. Egyszerre több ikonon is végrehajtható a művelet

A menü nem tartalmaz parancsot az ablak bezárására, azt a **Window** menüből tehetjük meg, hiszen a kinyitott ikon már ablak.

A következő menüpont a Copy, amely a jobb Amiga+C-vel érhető el a billentyűzetről. Ezzel lemezeket másolhatunk, hasonlóan, mint az 1.3 **Workbench**nél tettük (lásd a 34. oldalon).

A **Rename** az ikonhoz tartozó fájl átnevezésére szolgál. Az utána álló három pont jelenti, hogy egy kérdező fog feltűnni, ahol az új nevet be lehet táplálni.

Az **Information** szintén ismerős menüpont, azonos az 1.3 rendszer **Workbench** menüjének Info pontjával, (lásd a 45. oldalon) A különbség, hogy ehhez már nincs szükség a Libs: könyvtárban az info.library-ra, az a ROM-ban van. Az Info rövidítése a jobb Amiga+I.

A **Snapshot** hasonlóan működik, mint a **Window** menü azonos nevű pontja, de csak a kijelölt ikonokra van hatással. Billentyűzetről a jobb Amiga+S váltja ki. Mivel a **Snapshot** információ az info fájlokban tárolódik, ezért csak a valódi ikonnal rendelkező fájlokon van hatása. Azokon az ikonokon, amiket a rendszer ad a **Show/Alt** Files bekapcsolásakor, hatástalan. Ez a megkötés természetesen érvényes a **Window** menüben lévő **Snapshot-ra** is.

Az **UnSnapshot** az előző ellenkezője, leveszi a kijelölt ikon **Snapshot** információját, és annak az alapinformációként tárolt helyzete lesz az érvényes. Gyors elérése a jobb Amiga+U kombinációval lehetséges.

A **Leave Out** ismét egy hasznos szolgáltatás, amely a jobb Amiga+L billentyűk együttes lenyomásával közvetlenül is elérhető. Hatására a kijelölt ikon vagy ikonok "kiugranak" az ablakból a Workbench képernyőre. Emlékszel, ennek a **hasznosságát már megbeszéltük az előző fejezetben (lásd a 132. oldalon)**. Van azonban a két eljárás közt egy hatalmas különbség, a 2.0-s rendszertől kezdődően a **Leave Out**-tal kirakott ikon

megjegyzni, hogy kirakták, következő bootolás után már eleve ott lesz. Hasznossága tehát rögtön látható, véget ért az ezernyi kinyitandó ablak korszaka, mindenki a legfontosabb sűrűn használt programjának ikonját kirakhatja a munkaasztalra, ahonnan azok az ablakok nyitogatása nélkül közvetlenül indíthatók. Nem csak programok, hanem fiókok ikonjai is kirakhatók.

A következő menüpont a **Put Away**, amivel a jobb Amiga+P billentyűk együttes használata egyenértékű, az előző fordítottja, a kihelyezett ikont visszateszi saját ablakába. Ehhez a művelethez nem kell nyitva lenni az ablaknak, az ikon "tudja" hol a helye.

Az Icons menü utolsó három pontja el van választva a többitől. Ezek mind a "veszélyes" kategóriába tartoznak, használatukkal óvatosan kell bánni.

Az első közülük a Delete, amely törli a kijelölt ikonokat és a hozzájuk tartozó fájlokat. A parancs veszélyességét csökkentendő, egy requester-ben OK-val kell válaszolni a biztonsági kérdésre, a tényleges végrehajtáshoz.

A következő a Formát Disk, amely a kijelölt lemez formázását végzi. Az eljárás hasonló a már megismerthez. A formázás megkezdése előtt most is megjelenik a biztonsági kérdést tartalmazó requester, de három választási lehetőséget kínál. Az OK elkezd a lemez teljes formázását, az OK-Quick csak gyors formázást végez, (lásd a 80. oldalon) a Cancel pedig visszavonja a parancsot. A bal Amiga+V és B billentyűkombináció használható az OK és a Cancel válasz adásához, de az OK-Quick csak egérrel adható ki.

A menü utolsó pontja, az Empty Trash szintén nem új, a kijelölt szemétkosár tartalmát törli.

A munkaasztal utolsó menüje a Tools, amely jelen esetben csak egy pontot, a ResetWB-t tartalmazza. Ez újraindítja a Workbenchet. A menü pontjainak száma bővíthető segédprogramokkal. Ide a felhasználó saját menüparancsokat definiálhat. Ennek módja az alkalmazott programtól függ.

1. A Workbench2.0 lemez tartalma

A **Workbench** menüinek ismertetése után ismerkedjünk meg a géphez mellékelt lemezeken található dolgokkal. Az ismerkedést kezdjük magán a **Workbench** lemezen, nyissuk tehát ki. Egyenlőre csak a saját ikonnal rendelkező fiókokat és programokat nézzük át, ezért az ablak kijelölése után a **Window** menüben kapcsoljuk be a **Show/Only Icons-t** és a **View By/Icon-t**.

A lemez fő tartalomjegyzékében mindössze egy programikont találunk, a **Shell-t**, a többi mind fiókot jelképez. A Shell az előző fejezetekben megismert módon használható, itt nem részletezzük.

A már ismerős fiókokon kívül - **System, Utilities, Expansion, Trashcan, Prefs** - találunk két újat is. Ezek a **WBStartup** és a **Monitors**.

2. System fiók

Kukkantsunk bele először a System-be. Ebben a fiókban rendszerint kilenc ikonnal találkozhatunk, amiből a **Setmap, NoFastMem, Formát, CLI, DiskCopy** és **FixFonts** programokat már az 1.3-as **Workbench** leírásánál megismerhettük.

A CLI működése kissé megváltozott, már ez is a Newcon:-t használja, teljesen azonos módon viselkedik a Shell-lel, nincs köztük különbség.

a) Arexx

Az új ikonok közül való a RexxMast Ez az Arexx programnyelv interpretere, amely nyelvet együtt szállítják a 2.0-s Workbenchel, de minden ellenkező híresztelés dacára az Arexx az 1.3 és 1.2-s kickstartok alatt is fut, használható. Az ilyen gépek tulajdonosai az Arexx-et külön vásárolhatják meg.

A program rezidens, elindítása után beül a memóriába és ott is marad a következő resetig, vagy míg a Sys:Rexxc könyvtárban

meghúzódo RXC nevet viselő ikonnal nem rendelkező programmai el nem távolítjuk.

Mire való az Arexx? Ez egy olyan programnyelv, amelyben írt programokkal vezérelni tudunk futó, Arexx kompatibilis felhasználói programokat, egyszerre akár többet is. Olyankor hasznos, amikor az adott felhasználói programmal többször, közel azonos módon ismétlődő feladatokat kell végrehajtanunk, ilyenkor az Arexx program leveszi a vállunkról a rabszolgamunka terhét. Belátom ez kissé még zavaros, de egy példán keresztül rövidesen tisztázom.

Nos, képzelj el, hogy adva van egy számítógépes animáció 1000 képkockája, mind 24 bites nagyfelbontású kép. Ez az animáció mérete és képfarmátuma miatt mezei Amigákon nem tekinthető meg, át kell konvertálni például HAM-be és 'kisebb, mondjuk 320X256 pixeles felbontásba. A konvertáláshoz természetesen nagy teljesítményű Amiga szükséges, például egy A 3000-es, valamint egy képkonverter program, legyen ez az ADPro, amely Arexx kompatibilis.

A helyzet bonyolításához tegyük fel továbbá, hogy minden egyes képkocka össze van sűrítve a PowerPacker tömörítő programmal, a kitömörítéshez ennek legalább 4.0-s változatát használjuk, mert ettől kezdve Arexx kompatibilis.

Ha minden adott, a művelet manuálisan a következő: Betöltjük az első képkockát a Powerpacker-be, majd kitömörítjük és kimentjük egy átmeneti fájlba. Ezt az átmeneti fájlt betöltjük az ADPro-ba, megfelelő formátumúra alakítjuk és kimentjük a végleges nevéen. Mivel az átmeneti képfájltra már nincs szükségünk, azt a Delete paranccsal letöröljük.

Ha ezzel megvagyunk a művelet sor már csak 999-szer kell megismételni. Mire ezzel a módszerrel végeznénk, a nyugdíjkorhatárt is elérnénk, kell tehát egy kényelmesebb módszer.

Ezt a kényelmesebb módszert nyújtja az Arexx. írni kell egy ciklust, ami pont annyiszor hajtódik végre, ahány képkocka van. A program a ciklusban utasítja a Powerpackert a soron következő

fájl betöltésére és kitömörítésére, majd az átmeneti néven történő kimentésre. Ezután ugyanazon a cikluson belülről a program parancsot küld az ADPro-nak, hogy töltsse be az átmeneti fájlt, konvertálja és mentse ki. Végül a programból kiadjuk a DOS Delete parancsát az átmeneti fájl törlésére. Mindezek után a ciklusszámlálót eggyel csökkentve vagy növelve, megismételjük a műveletet. Mire a ciklus 1000-szer lefut, el is készül a konvertálás, úgy, hogy közben nem kell a gép mellett tartózkodni, például alhatsz egyet.

Legjobb, ha a képek nevei PicOO1, PicOO2... stb, vagy hasonló, de ez általában így is van, az animációkészítő programok eleve így mentik ki a képkockákat. Ebben az esetben nem kell betáplálni mind az ezer nevet, elég csak a név állandó részéhez hozzáadni a ciklusszámláló értékét.

Ennyiből is látható az Arexx haszna, pedig lehetséges alkalmazásának ez csak piciny része. Sajnos a könyv keretei nem teszik lehetővé, hogy bővebben tárgyaljuk a nyelv lehetőségeit, de a téma iránt érdeklődők figyelmébe ajánlom a Guru Szórakoztató Informatikai Magazint, amelyben többek között egy Arexx rovat is olvasható hónapról-hónapra. Amennyiben elegendő igény mutatkozik rá, nem elképzelhetetlen, hogy könyvet is jelentetünk meg ebben a témában.

b) Monitorok

A következő új ikon a System-ben, az AddMonitor programé. Ezzel szoros összefüggésben áll a Monitors nevű fiók. A program működésének megértéséhez elevenítsük fel a monitorokról szóló ismeretanyagunkat. A könyv elején megtárgyaltuk, hogy az Amigánkhoz RGB monitort kapcsolhatunk. Az Amiga 500+ , 600, 1200, 3000 és 4000 típusú gépeken azonban másfajta monitor csatlakoztatására is van lehetőség, de ekkor a megfelelő működés érdekében fel kell hívni a gép figyelmét erre. Ezután a gép videó rendszere már az új monitornak megfelelő formátumban fog dolgozni. Az AddMonitor az, amely tájékoztatja a gépet a monitor típusáról.

Az AddMonitort Workbenchből kétféle módon használhatod. Az egyik, hogy a program ikonjában a Tool Types-be beírod a szükséges paramétereket, a másik, hogy szöveg fájlba írod a program nevét és paramétereit, beállítod az S flagot a Protect paranccsal, majd ikonnal látod el az így létrehozott egysoros batchprogramot. A megfelelő monitor beállításához a szokásos módon dupla klikkel indíthatod el a szükséges batchfájlt.

Szerencsére a Workbench írói gondoltak erre, és elkészítették a legfontosabb, Amigához is használt monitorok használatához szükséges batch fájlokat, amelyeket az Extrás lemez Monitor-Store fiókjában helyeztek el. A következő négy program található itt

A2024 - Amiga 2024 típusú nagyfelbontású monokrom monitorhoz.

Multiscan - Multiscan monitorokhoz, mint például a Commodore 1960-as.

NTSC - NTSC Videó monitorokhoz

PAL - PAL Videó monitorokhoz.

A szükséges programot másold át a Workbench Monitora könyvtárába.

A másik hasonló program, a Bindmonitors. Ez nem monitorokat, hanem azokhoz tartozó képernyő módokat épít be a rendszerbe. A videó módok megnevezését és kódját a WB Startup tartalomjegyzék Mode-Names fájlja tartalmazza. Ez hasonló felépítésű, mint az előzőekben ismertetett monitor fájlok.

Utóbbi két program általában a Startup-scquence része, használni már nem kell őket.

3. Utilities fiók

Nézzünk át a másik ismerős fiókba, az Utilities-be. A Clock, Say és a More mellett két új ikon is található itt, a Display és az Exchange. Mielőtt azonban ezekkel részletesebben megismerkednénk, tekintsük át, mit fejlődtek a régi programok.

A **Clock** nem változott lényegesen, de a **Say** tökéletesítve lett. Használata megegyezik az 1.3 verziójú Workbenchbeli társáéval, (lásd a 54. oldalon) de sokkal jobb minőségű beszédhangot állít elő.

A legnagyobb változáson a **More** ment keresztül. Ha nem adunk át neki paramétert, akkor egy kérdezőt fog megjelentetni, ahol kényelmes módon választhatjuk ki a megtekinteni kívánt szöveg fájlt.

A kérdező három részből áll. Tetején egy ablakban listát találunk az aktuális könyvtárban lévő fájlokról és tartalomjegyzékekről. Ha a lista hosszabb, mint ami az ablakban elfér, a jobb oldali gördítő nyilakkal mozoghatunk benne a már ismert módon. Ha egy könyvtár nevére ráklickelek, akkor belépünk abba, az ablakban az új könyvtár tartalma jelenik meg.

A kérdező középső részén három sort találunk. Az első a **Pattern**. Ide a megjelenítéshez vagy megjelenítés letiltásához szükséges mintákat írhatjuk be. A mintákról már szoltam a könyv egy korábbi részében (lásd a 67. oldalon). Kezdetben a minta a ~(#?.info), amely hatására az infofájlok nem jelennek meg a listában. (A ~ a letiltás jele.) Ha például ide beírjuk, hogy #?.text, akkor csak a .text kiterjesztésű fájlok és a tartalomjegyzékek neveit láthatjuk a listaablakban.

A második sorban, amely a **Drawer** nevet viseli az aktuális könyvtár elérési útvonalát és nevét láthatjuk. Ha belépünk egy könyvtárba, annak a nevével bővül az út.

A harmadik sor a **File**, amelyben a kiválasztott fájl neve látható. Kiválasztani úgy lehet egy fájlt, hogy kattintasz a bal gombbal egyet a nevére.

Az alsó sorban négy kapcsoló van. Az OK-val beolvassuk a fájlt a More-ba. Csak akkor működik, ha van kiválasztott fájl. Ha nincs ilyen, hatására kilépünk a kérdezőből és a More-ból. A fájl megnyitása úgy is elvégezhető, hogy duplán klickelek: a nevére a listában.

A második gomb a Disks. Erre klikkelve a rendszerben lévő fizikai és logikai egységek listáját kapjuk meg. A logikai egységek előtt kiemelt háttérű <ASN> kulcsszó áll. A tartalomjegyzékeket ugyanígy, de <Dir>-rel jelzi. Az aktuális meghajtó változtatásához elég egyet klikkelni annak a nevének. Ez megjelenik a Drawer sorában.

A Parent kapcsolóval az eggyel magasabb szintű tartalomjegyzéket tesszük aktuálissá. Ha a használatkor valamely fizikai vagy logikai egység gyökérkönyvtárában voltunk, akkor a Disks listáját kapjuk eredményül.

Az utolsó kapcsoló a Cancel, hatására kilépünk a More-ból, függetlenül attól, hogy volt-e kijelölt fájl vagy sem.

A kérdező egy normál ablak, a szokásos gadgetekkel. Az ablakzáró kapcsolóval nem csak a kérdezőt zárjuk be, hanem a More-ból is kilépünk.

Ha a More-nak indításakor paramétert adunk át, a kérdező nem jelenik meg, hanem megpróbálja megjeleníteni a paraméterben megadott szöveg fájlt.

Az egyik új program az Exchange, összefüggésben van az Extras2.0 lemez Tools/Commodities fiókjában lévő programokkal, ezért majd azokkal együtt ismertetem.

A másik új program a Display. Ennek részletes ismertetését szintén elhalasztjuk, mivel sokféle paramétert adhatunk át neki, ami Workbench-ből az ikonjának Tool Types mezőjén keresztül lehetséges. Célszerűbb a programot DOS szintjén használni, ezért ott térünk vissza rá. Egy Workbench alkalmazását azért megemlítem, ha egy *Ttt* képet muhi szelekttel átadsz neki, azt megjeleníti. A megjelenítésből az Esc billentyű, vagy a bal gomb használatával lehet kilépni.

4. A Workbench beállításai

A Workbench2.0 következő tartalomjegyzéke, amellyel megismerkedünk, a Prefs. A preferences beállító programok az előző

A Workbench2.0

verzió óta igen nagy változáson mentek keresztül. A változás a fiók kinyitása után azonnal szembe ötlik, ugyanis az eddigi négy ikon helyett 12 program és egy fiók ikonnal találkozhatunk. A Presets fiókban különböző beállítási értékeket tárolnak a Prefs programjai.

Mint a programok neveiből látszik, jóval több mindent állíthatunk be, mint az előző vátozatban, ráadásul ezeken a beállításcsoportok teljesen szét vannak bontva önálló beállító-programokra, nincs egy közös, amelyikben mind benne van. Nézzük, melyik programmal mit állíthatunk be.

IControl

A Workbench ikon és kérdező kezelésével kapcsolatos paraméterek beállítására nyílik lehetőség. A program képernyőjén több csoportra vannak bontva a beállítható dolgok. Az első csoport a Verify Timeout-é. Itt a rendszer reakcióidejét állíthatjuk be. Ez a reakcióidő határozza meg, hogy milyen gyorsan reagál az egyes ikonműveletekre. Ha nagyobb időt állítunk be, valamelyest gyorsul a Workbench, de az ikonkezelés lassul. Az optimális érték 1 másodperc.

A második csoport a Command Keys. Azokat a billentyű kombinációkat állíthatjuk be, amelyekkel a képernyőt háttérbe vagy előtérbe küldhetjük, valamint a kérdezők OK illetve Cancel választ adhatjuk. Ezek a műveletek mindig a bal Amiga és egy karakter billentyű egyidejű lenyomásával válthatóak ki. Az alap beállítás az N-előtérbe, M-háttérbe küldés, V-Ok és B-Cancel.

A harmadik csoport a Mouse Drag. Ezt beállítva nemcsak a fejlécénél fogva húzhatjuk le a képernyőt, hanem az itt jelzett emelőbillentyű lenyomásával bármelyik pontján keresztül. Az alap beállítás a bal Amiga gomb. Egyszerre több emelő billentyűt is beállíthat, ekkor azokat mind le kell nyomni és úgy tartani a hatás eléréséhez.

A negyedik csoport a Coercion. Itt két kapcsolót találunk, amelyeknek csak a productivity képernyő üzemmódban van hatásuk (lásd később, a 160. oldalon).

Az Avoid Flicker a nagyfelbontású productivity módot használó multiscan monitorral csökkenti a képernyő remegését.

A Miscellaneous Flags csoportjában a menük megjelenését a Shell szerkesztő billentyűinek (pl CTRL+A, CTRL+Z) használatát engedélyezhetjük.

Az utolsó csoport, a kapcsolók sora már ismerős, a Savé kimenteti a beállításokat, a következő bootlás után már eleve ezek lesznek érvényesek, a Use csak használatba veszi, de nem menti, a Cancel pedig figyelmen kívül hagyja a beállításokat.

A preferences programokban van még egy közös újdonság, saját menüik vannak, amelyek mindben azonosak, ismerkedjünk meg velük.

Az első menü a Project, ez a szerkesztett beállításokra vonatkozik. Az első pontja az Open, amellyel már elkészített beállításokat tölthetünk be a programba. Ezeknek a beállításoknak a tárolására szolgál a Presets fiók a Prefs tartalomjegyzékében.

A második menüpont a Savé As. Ezzel menthetjük ki az elkészített beállításokat más néven, mint az alapértelmezés szerinti Devs:system-configuration. Az így kimentett fájlokat általában a Presets fiókba tesszük, a menüpont is ezt kínálja fel alaptól, de megváltoztathatjuk, hasonlóan a felkínált fájlnevhez.

A Project menü utolsó pontja a Quit, amely a Cancel kapcsolóval egyenértékű.

A második menü az Edit. Ebben három menüpontot találunk, amelyekből az első, a Reset to defaults, az alapértelmezés szerinti beállításokat *hozza*, vissza az editorba. A második pont, a Last saved, az utolsó mentés szerinti értékeket olvassa be, a Restoré pedig a jelenleg érvényes beállítást adja.

Az utolsó menü, az Options, mindössze egy pontot tartalmaz, a Savé Icons?-t. Ha ez a pont ki van pipálva, a Savé As a kimentett prefs-fájlnak egy ikont is készít. Ezt az ikont azután multiszelekt módszerrel átadva az őt létrehozó program

indításakor, mint paramétert, nem törlődik be az editor, hanem az ikonhoz tartozó fájlban tárolt beállításokat veszi fel a rendszer.

Az IControl Shell-ből vagy CLI-ből is használható:

Icontrol

Opciók és paraméter is adhatók meg használatakor. A "**FROM név**" opcióval meglévő prefs-fájlt adhatunk át neki feldolgozásra. A feldolgozás mikéntjét a második opció határozza meg. Ez a következő három valamelyike lehet:

Edit - betölti az editort és abba beolvassa a megadott fájlban tárolt paramétereket. Ha nem adsz meg második paramétert, ez az alapértelmezés.

Use - Az editor betöltése nélkül használatba veszi a megadott fájlban lévő paramétereket.

Savé - Mint az előző, de a beállításokat ki is menti a system-configuration-ba, következőekben ez lesz az induló beállítás.

A preferences editorok hasonló felépítése miatt nemcsak menük, hanem a programparamétereik és azok használata is azonos, a fent leírtak szerinti.

A következő preferences editor program az **Input**. Ebben az egérre és a billentyűzetre vonatkozó beállításokat eszközölhetünk, amelyek már ismerősek lehetnek (lásd a 37. oldalon).

A Mouse **Speed** az egér sebessége, a **Double Click** pedig a kettős klikkelés ideje. Újdonság az Acceleration kapcsoló, amellyel a Mouse Speed értékét duplázzhatjuk meg. Hogy megfelelő-e a sebesség, azt rögtön kipróbálhatjuk, ugyanis a tolóka vagy kapcsoló használata után az egér rögtön használja a beállított sebességet.

A dupla klikk kipróbálására két módszer is van. Az egyik a Show kapcsoló, amelyre ráklikkelve a mellette lévő négyzet annyi időre színesedik el, mint amennyi a dupla klikk időintervalluma. A másik lehetőség a Test kapcsoló. Ha erre kettőt klikkelünk, Yes vagy No kiírásával megmondja, hogy az dupla klikk volt-e vagy sem.

Az editor közepén a **Keyboard** nevű részben a billentyűzet paramétereinek beállítását szolgáló tolokák vannak. Használatuk hasonló, mint az 1.3 Workbench Preferences-ében, (lásd a 38. oldalon) de itt numerikusán is megjelenik a beállított érték. A **Key Repeat Test** sorába klikkelve kipróbálhatjuk a beállítást.

Az ablak alján az ismerős három kapcsoló található.

Az Input menüinek és **DOS** szintű használatok opcióinak alkalmazása azonos az IPrefs-nél elmondottakkal.

A harmadik prefs program, a **Font**. Ebben a képernyőn használt fontok típusát és méretét, valamint az ikonok feliratának színét állíthatjuk be.

Eltérően az 1.3-as rendszertől, amely csak a **Topaz/8** vagy **Topaz/9** ROM font használatát tette lehetővé a Workbenchben (text 80 vagy text 60 kapcsoló, (lásd a 37. oldalon), az új rendszerben már bármilyen fontot használhatunk, nem csak a ROM-ban lévő Topaz/8,9,11-et, hanem a **Fonts:** tartalomjegyzék akármely fontját is. Ráadásul a fontokat külön beállíthatjuk a Workbench ikonokra, a képernyő felirataira és a rendszer alapszövegeire.

Az ablak tetején három kapcsolóval választhatjuk ki, hogy a lehetőségek közül, amelyet most soroltam fel, melyiket állítjuk be. Az aktuális font a kapcsoló után olvasható. Választani az ablak közepén lévő listából lehet, ahol a ROM és a Fonts: direktori fontjai és azok méretei jelennek meg. A jobb oldali négyzetben a kiválasztott font néhány eleme jelenik meg mintaként.

Az eredeti Workbench használatok mindössze három font közül választhatunk, ugyanis a lemez Fonts könyvtára üres, csak a ROM fontok állnak rendelkezésünkre. A fontokat külön lemezen, a Fonts nevűn kapjuk meg.

Érdeemes egy kis kitérőt tenni erre a lemezre, hisz tartalmaz néhány újdonságot az előző rendszer fontjaihoz képest. A lemezen a fontok a Fonts tartalomjegyzékben vannak. Nézzük meg ennek a listáját. Az egyik, ami szembetűnik, az az, hogy több fontkészlet szerepel rajta, mint az 1.3 Workbenchben. A másik,

lényegesebb különbség a _Bullet-Outlines könyvtár és a furcsa, CG kezdetű és ottag kiterjesztésű fájlok.

Ha belenézünk ebbe az új könyvtárba, találunk benne négy type kiterjesztésű fájlt. Ezek a 2.0-s rendszer fontkezelés terén történt hatalmas előrelépésével vannak kapcsolatban. Mi ez az újdonság? A 2.0-s kickstarttól kezdődően lehetőség van az úgynevezett **vektorfontok** használatára. Ennek megértéséhez fel kell eleveníteni a fontokról szóló tudásanyagunkat.

Az eddigi fontok mind ún. bitmap fontok voltak, vagyis olyanok, amelyek előre meg voltak rajzolva pontokból, különböző méretekben. Hátrányuk, hogy csak a rendelkezésre álló méretekben használhatóak. Ha valamilyen módszerrel, például egy grafikán kinagyítjuk őket, a betűk "kiszőrösödnek" mivel a nagyítás a karaktert alkotó pontok méretének növelésével megy végbe.

Ezeken a korlátokon lép túl a **vektorfont**, ahol nem a betűk pontjai, hanem az őket alkotó görbék és egyenesek vannak megadva vektoros formában, matematikai módon. így elegendő a fontot egyszer, méretfüggetlenül meghatározni, utána az bármekkora méretben felhasználható, minőségének romlása nélkül.

Az Amiga vektorfontkezelő rendszere, amelynek fő része az új **diskfontlibrary**, az **AGFA Computer Graphics** fontokat alkalmazza, erre utal a fontok nevében a CG. Megfelelő fontkonverterrel más formátumú vektorfontokat, például a PC-n elterjedt **True Type** és a Machintosokról elszármazott **Adobe Type 1** fontokat is átkonvertálhatjuk erre a formátumra.

Ennyi kitérő után térjünk vissza a Font beállításához. Az ablak alsó részén lévő két kapcsolóval az ikonok feliratának színét tudjuk beállítani. A színt a jobb oldali skáláról vehetjük le.

Ha a fontokat változtattuk, a Savé és Use használata után figyelmeztetést kapunk, hogy a fiókok kivételével minden ablakot csukjunk be, mert a beállítás érvényesítéséhez a Workbenchet **resztelni kell, az pedig csak akkor bajtható végre, ha nincs futó program rajta.**

A következő prefs-program az Overscan. Itt a képernyőt állíthatjuk be úgy, hogy az használja az Overscan területet is. Ez az a rész, amelyet a fizikai képernyő szélein hagyunk a logikai képernyő közül. Azért van rá szükség, mivel az overscan területen az elektronsugár torzítva rajzolja a képet. Bizonyos esetekben, például ha videóra vesszük a képet, a jobb esztétikai hatás eléréséhez szükség lehet arra, hogy az overscan területet is használjuk.

A beállítást kétféle módon végezhetjük el. A standard beállításokat az ablak tetején lévő monitor kiterjesztésű fájlok listájából választhatjuk ki. Ha itt nem találunk megfelelőt, az Edit Text Overscan és Edit standard Overscan kapcsolóra klikkelve az egérrel megváltoztathatjuk a használt képméretet. Ha megfelelőnek találjuk a beállítást, nyomjuk meg a Returnt, ha meggondoltuk magunkat az Esc billentyűt. Ez a két válasz menüből is kiadható.

A kapcsolók alatt láthatjuk a beállított értékeket.

Az ötödik prefs editorban a Palette-ben újból olyan értékeket állíthatunk be, amelyekkel már találkoztunk. A színek beállítása a megismert mód szerint történik (lásd a 37. oldalon).

A következő program, a Pointer szintén ismerős feladatot végez, a mutatót szerkeszthetjük vele. Ez teljesen azonos a 42. oldalakon leírtakkal.

A Printer és a Printer GFX használata is hasonló a 39. oldalon leírtakkal, mindössze a program ablakának kinézetében van különbség, itt nem részletezem azokat.

A Screenmode viszont egy teljesen új program. Segítségével a Workbench által használt képernyő paramétereit állíthatjuk be.

A bal felső Choose Display Mode nevű listából választhatjuk ki az alkalmazandó képernyő típusát. Itt olyan értékek jelennek

A Wokbench2.0

meg, amelyekre a gép képes. Ezek a következők lehetnek egy A500+, A600 vagy A3000 típusú gép esetében:

<u>Mód</u>	<u>monitortípus</u>	<u>felbontásmax, szín</u>	
PAL: Hires	PAL-Monitor	640x256	16
PAL: Hores			
Interlace	PAL-Monitor	640x512	16
PAL: SuperHires	PAL-Monitor	1280x2564	
PAL: SuperHires			
Interlace	PAL-Monitor	1280x512	4
NTSC: Hires	NTSC-Monitor	640x200	16
NTSC: Hires			
Interlace	NTSC-Monitor	640x400	16
NTSC: SuperHires	NTSC-Monitor	1280x200	4
NTSC: SuperHires			
Interlace	NTSC-Monitor	1280x400	4
Productivity	Multiscan-Monitor	640x480	4
Productivity			
Interlace	Multiscan-Monitor	640x960	4
A 2024-1 OHz	A2024-Monitor	1008x800	4 szürke
A2024-15Hz	A2024-Monitor	1008x800	4 szürke

Az aktuális kiválasztott mód neve szerepel a lista alatti sorban. A módhoz tartozó lehetőségeket, úgy mint PAL vagy NTSC, Supports Genlok (genlokkolható) stb, a jobb oldali listában olvashatjuk.

A bal oldali lista ablak és az aktuális beállítás alatt látjuk a Visible Size után, mekkora a képernyő látható mérete. A Min Size a minimális, a Max Size a maximális méretet mutatja, a a Max Colors pedig a lehetséges színek számát.

A maximális méret értékeként igen nagy számok láthatók. A 2.0-s Workbenchen a képernyő nincs korlátozva a látható méretre, ennél nagyobbakat is létrehozhatunk. Itt valódi képernyőméretről van szó, és nem a Backdrop menüpontnál említett (lásd a 140. oldalon) ablak méretéről.

A képernyő méretét jobb oldalt, a Width-szélesség és Height-magasság rovatban adhatjuk meg. A Default kapcsolókkal az

alapértékre hozhatjuk a méretet. A Colors tolokájával a Workbench színeinek számát állítjuk be. Ebből adódik, hogy nemcsak négy színű képernyő használatára van lehetőség, hanem akár 16-ra is.

Az Autoscroll kapcsolónak akkor van jelentősége, ha a látható méretnél nagyobb képernyőt állítottunk be. Ebben az esetben bekapcsolva az Autoscrollt, amint az egér mutatója kikerül valamelyik oldalon a képernyő szélére, az automatikusan gördülni fog.

A másik megoldás a képernyő mozgatására, hogy vagy a fejlécénél, vagy az IControlban beállított emelő billentyűk) lenyomásával valamely pontjánál fogva a megfelelő irányba elhúzzod.

Mielőtt még óriási képernyők létrehozásába kezdenél, felhívom a figyelmed arra, hogy ez jelentős mennyiségű Chip memóriát költ le.

A Serial programmal a soros port átviteli jellemzőit lehet beállítani a már ismertetett módon (íásd a 41. oldalon).

A Time-mal a dátum és az idő állítható be, kissé kényelmesebb módon, mint eddig. A program használata egyszerű, a naptárból könnyedén választható ki a dátum. A hónap kiválasztása a visszaforduló nyílra klikkeléssel lehetséges, ekkor a következő hónapra ugrunk. Visszafelé is lehet lépni, ekkor a bal klikk ideje alatt lenyomva kell tartani a Shift billentyűt.

Megjegyzendő, hogy ez a típusú kapcsoló mindenütt így működik (eltekintve egy-két kivételtől).

A Time többi beállító funkciója egyértelmű, kezelése az eddig elmondottak alapján nem okoz gondot.

Az utolsó preferencias program a WBPatten. Ezzel a Workbench képernyő és az ablakok háttérmintázatát tervezhetjük meg és állíthatjuk be. Hogy mire vonatkozzon (Workbench, Windons) a bal felső sarok két kapcsolójával állíthatjuk be. Kissé lentebb a

A Wokbench2.0

Test kapcsolójával megtekinthetjük művünket, a **Clear**-ral törölhetjük a szerkesztő területet a kiválasztott színnel, az **Undo** pedig a legutolsó szerkesztő műveletet hatástalanítja. Hogy^ mi fog visszaállítani, azt a kapcsoló melletti kis négyzetben megtekinthetjük.

A képernyő közepén van a szerkesztő terület, itt rajzolhatjuk meg a háttér mintázatot, amely csempészerűen rakódik ki a képernyőre. A rajzolás hasonló módon megy végbe, mint a mutató tervezése. A szint a szerkesztő terület mellett jobbra lévő színszalpról választhatjuk meg. Természetesen csak az aktuális Workbench színek használatára van lehetőség.

Az ablak jobb felső sarkában lévő kis területen láthatjuk, hogyan néz ki majd a kész mű, 3x4 csempényi területen.

E mintaablak alatt találunk nyole előre elkészített csempét, amelyet felhasználhatunk, esetleg tovább alakíthatunk. Valamely csempére ráklikkelve, az megjelenik a szerkesztő területen.

Ezzel befejeztük a Workbench ikonnal ellátott programjainak ismertetését, nézzünk át az Extrás lemeze.

A lemezt kinyitva három fiókot és egy szemétkosarat találunk. A MonitorStore fiókkal és annak tartalmával már megismerkedhettünk az **AddMonitor** kapcsán (lásd a 151. oldalon).

Nézzünk be a **Tools** fiókba. Itt különböző segédeszközöket találunk, amelyek többsége az előző Extrás verzió leírásából ismerős lehet, a **CMD**, **MEMacs**, **Calculattor**, **KeyShow**, **Graphicdump**, **Initprinter** és **Printfiles** programokkal már megismerkedtünk.

Tulajdonképpen az IconEdit is régi program, hogy mégis terítékre vesszük, annak az az oka, hogy rengeteget fejlődött, sok új funkciója van, amelyekkel érdemes megismerkedni. A program ablakában bal oldalt találjuk a szerkesztő területet, középen a színválasztó részt és a szerszámos ládát. A szerszámos ládában a következő rajzeszközöket találjuk:

Szabadkézi szaggatott vonal. Billentyűzetről az S billentyűvel váltható ki, egérrel úgy, hogy ráklikkelünk.

Szabadkézi vonal, folytonos szabadkézi vonal rajzolásához. Billentyűzetről a D gomb lenyomásával érhető el.

Körző, körök és ellipszisek rajzolásához. Bal oldalára klikkelve csak a kör körvonalát rajzolhatjuk meg, jobb oldalára klikkelve a kör ki is színeződik. A rajzeszköz billentyűzetről az E lenyomásával kapcsolható be. A fillezett kör rajzolásához kétszer egymás után kell lenyomni a billentyűt.

Négyszet. Hasonló módon működik, mint az előző, fillezetlen vagy fillezett négyszögek rajzolását végezhetjük vele. Billentyűzetről az R betű váltja ki.

Vonalzó, egyenesek rajzolásához. Az L billentyűvel is kiváltható.

Kitöltés. Ezzel az eszközzel zárt területek szinkitöltését végezhetjük el.

A jobb felső sarokban látjuk az éppen szerkesztett ikon eredeti méretű képet, amikor az ikon nincs kijelölve. Kissé lentebb egy ugyan ilyen négyzetben az ikon kijelölt állapoti képét szemlélhetjük, ha már el van készítve. Az hogy éppen melyik képet szerkesztjük, azt a két kép közti gombokkal választhatjuk ki.

A jobb alsó sarokban lévő négy nyíllal a szerkesztett képet tolnak el a szerkesztőben. Vigyázzunk, mert ha a kép valamely részlete ily módon kicsúszik a szerkesztő területéről, az törlődik!

Az ikon megrajzolását a felsorolt eszközökön kívül menük is segítik. Nézzük meg, melyek ezek és mire valók:

Project menü

- New - új ikon létrehozása
- Open - már meglévő ikon tovább szerkesztése
- Savé - ikon kimentése
- Savé As - ikon kimentése névadással
- Savé As Default Icon - ikon kimentése alap ikonként
- Quit - kilépés a programból.

Edit menü

A Wokbench2.0

Cut - kivágás, megőrzéssel
Copy - hasonló mint az előző, de az eredeti nem törlődik
Paste - az előző két mód valamelyikével eltárolt kép visszahelyezése.
Open Clip - egy Clipboardon lévő kép beolvasása
Save.Clip As- kimentés a Clipboardra
Show Clip - megmutatja a Clipboard tartalmát.

Type menü - az ikon típusának kiválasztása
Disk - lemez ikon
Drawer - fiók ikon
Tool - program ikon
Project - adat, terv ikonja, például szövegfájl, kép, stb
Garbage - szemétkosár ikon

Highlight menü - a kiválasztás jelölése
Complement - invertálás
Backfill - háttér kitöltése
Image - másik kép

Images menü - a kép manipulálása
Exchange - felcseréli a normál és a kiválasztott ikon képét.
Copy - a jelenlegi kép átmásolása a másik állapotra, Például, ha a normál képet szerkesztjük, ezt átmásolja a kiválasztott állapotnak.
Template - az ikon képe a választott típusnak megfelelő alap ikon lesz.
Load IftBrush - betölt egy IFF képet a szerkesztőbe
Normál Image - betölt egy IFF képet a normál ikonnak, függetlenül attól, hogy azt szerkesztjük-e.
Selected Image - betölt egy képet a kiválasztott ikonnak
Both Images - betölt egy képet és azt mindkét állapothoz hozzárendeli.
Savé IFF Brush - kimenteti a szerkesztett képet IFF Brush-ként, később az visszaolvasható.
Restore - visszaállítja az ikon eredeti képét.

Extrás menü
Recolor - új színeket rendel az ikonhoz

- Autó Top Left** - automatikusan a bal felső sarokba igazítja az ikon képét.
Color Palette - a szinek beállítására van lehetőség

Settings menü

- Use Grid** - megjelenít egy szerkesztést segítő rácsot
Savelcons? - bekapcsolásával a Savé IFF Brush menüpont ikont is ad a kimentett képnek
Savé Settings - elmenti az ikon editor beállításait, következő indításkor eleve ezeket fogja használni.

A Tools egyik új programja a **PrepCard**. Ezzel a **PCMCIA** memóriakártyák formázása valósítható meg. Ez a kártya egy elemes, a gépből való eltávolításkor sem törlődő memóriakártya, amely rendszerfüggetlen, azaz használható PC-hez Atarihoz, Macintoshhoz is, nem csak Amigához. Ez a függetlenség csak hardverileg igaz, a tárolt adatok formátuma minden gépen más. Természetesen megfelelő meghajtóprogrammal mindenféle adatformátumot értelmezni lehet.

PCMCIA csatlakozóval az Amiga 600-as és 1200-as model rendelkezik.

A programmal formázhatjuk a kártyákat és beállíthatjuk, hogy memória diszk-ként vagy rendszer memóriaként kívánjuk-e használni. Utóbbi esetben a kártya memóriabővítésként funkcionál, de ekkor nem lehet eltávolítani a bekapcsolt gépből a rendszer összeomlása nélkül.

A Tools fiók utolsó programja a **HDBackup**. Ez egy merevlemez archiváló program. Segítségével a teljes tartalmát biztonságba helyezhetjük a Winchesterünknek, vagy csak egy részének. Az összetömörített adatokat mágneslemezen vagy szalagos háttértárolón, ún. Streameren helyezhetjük el.

Ha az eredeti adatok megsérülnének, például egy véletlen törlés által, azok a másolatról helyreállíthatóak.

A program ismertetésétől terjedelmi okok miatt el kell tekintenünk. Egyébként is, nagyon sokféle, ennél jobb, kényelmesebben használható tömörítő és archiváló program van forgalomban,

A Wokbench2.0

hogy a legnépszerűbbek ismertetése is megtöltene egy egész könyvet. ígérem azonban, hogy a sorozat egy későbbi kötetében megpróbálunk részletes információkat adni a tömörítésről-archiválásról. Addig is az olvasó figyelmébe ajánlom a **Guru** magazint, ahol a téma iránt érdeklődők rendszeresen kaphatnak információkat.

A Tools-on belül találunk egy fiókot, a Commodies-t. Ebben hat program található, amelyek közös jellemzője, hogy elindításuk után rezidenssé válnak és különböző hasznos dolgokat végeznek. A korábban említett **ExChange** nevű program (lásd a 153. oldalon) ezen rezidens programok keretétül szolgál, de használata nem kötelező.

Autopoint - Elindítása után, mindig automatikusan az az ablak lesz aktív, amelyikben a pointer tartózkodik, anélkül, hogy ebbe az ablakba kellene kattintani

Blanker - A beállított, másodpercben megadott idő elteltével kikapcsolja a képernyőkezelést - ha közben nem nyúlnak a géphez - kímélve a monitort. Akár az egér megmozdítása, akár valamelyik billentyű lenyomása visszaállítja a képernyőt.

Az időintervallum beállítása után a Hide gombbal aktiválhatjuk a programot, vagy a Quit-tal léphetünk ki belőle. A háttérben működő program beállító képernyője vagy az Exchange-n keresztül, vagy a Shift+F1 lenyomásával érhető el.

ClickToFront - Automatikusan előtérbe kerül, alkalmazása után, az az ablak, amelyikben duplán kattintunk.

FKey - Különböző karaktorsorozatokat rendelhetünk a 20 funkcióbillentyűhöz (tíz normál + tíz a Shift-tel együtt). Szövegbevitelnél, például egy Shell ablakban, a funkcióbillentyű lenyomása egyenértékű a hozzárendelt karaktorsorozat begépelésével. Például a legsűrűbben használt DOS parancsokat hozzárendelhetjük egy-egy funkcióbillentyűhöz, így azok egy gombnyomásra elérhetőek, ha a Shell ablak az aktuális. A sorvégi return-t a \n beírása jelenti. Ha ezzel zárjuk a hozzárendelést, még a Returnt sem kell használni a parancs

érvényesítéséhez. A Shift használatát a Modifier kapcsolóval állíthatjuk be.

IHelp Az ötödik Commodities program, ezzel néhány ablakkezelő funkciót elérhetünk a billentyűzetről is, mintha a nekik megfelelő gadgeteket használnánk. A rövidítések a funkció-billentyűkhöz vannak rendelve.

F1 - képernyő váltása

F2 - maximalizálja az aktuális ablakot

F3 - minimalizálja az ablakot

F4 - előtérbe hozza az aktuális ablakot

F5 - az ablak zoom gadgetjét használja

Ha együtt használjuk az FKey programmal, ne állítsunk be értékeket ezekhez a billentyűkhöz, mert megzavarhatják egymást. Ennek a programnak is előhívható a beállítóképernyője. Ehhez az Alt és az FI billentyűket kell egyszerre lenyomni.

NoCapsLock- -Az utolsó Commodities program, mindössze annyit végez, hogy kikapcsolja a CapsLock ledjét.

Szó volt már arról, hogy az **Exchange** egy keretprogram a Commodities programok számára, ismerkedjünk most meg vele. Elindítása után egy ablakkal jelentkezik be, amelyen kapcsolók, egy listaablak és néhány információs sor van.

Bal oldalon két kapcsoló található. A **Hide** elrejtí az **Exchange** programot, ablaka eltűnik, de az **Alt+Help** billentyűkombinációval bármikor előhívható. A másik gombbal, a **Quit-tal** kilépünk a programból. Ezzel egyenértékű az ablakzáró gadget használata.

Az Available Commodities nevű listában a futó commodities programok listáját látjuk. Ha nem fér el minden név egyszerre, lapozni lehet. Amikor kiválasztunk egy programot a listából, az ablak alján lévő három információs sorban adatokat kapunk arról.

A Title után a program megnevezése áll. A **Description** mögött a program funkciójának leírása, a **Status** után pedig a jelenlegi állapota, amely lehet **Enabled** (engedélyezett) vagy **Disabled** (letiltott).

A Wokbench2.0

A jobb oldali öt kapcsoló a kiválasztott programra vonatkozik. A **Show** előhossa az adott program beállító képernyőjét, a **Hide** pedig eltűnteti azt a program háttérbe küldésével. Ennek a két kapcsolónak természetesen csak akkor van értelme, illetve akkor kínálja fel az **Exchange** a használatát, ha a kiválasztott programnak van beállító ablaka.

A következő két kapcsolóval engedélyezhető (**Enable**) vagy letiltható (**Disable**) annak a programnak a működése. Ez az állapot jelenik meg a Status sorban.

A **Kill** kapcsolóval "kíértható" a commodities program, a memóriából.

Maga az Exchange is commoditiesként működik, annak neve szintén szerepel a listában, a jobb oldali kapcsolók alkalmazhatók rá.

B. DOS 2.0

Az Amiga DOS 2.0 sok változtatást tartalmaz az 1.3 verziójú elődjéhez képest. Lépjünk be egy Shell ablakba, hogy megismerkedhessünk a változásokkal.

Mivel már volt rá eset, hogy használtuk a Shell-t, feltűnhetett, ennek az ablaknak is van bezáró kapcsolója, nem kell ezentúl minden esetben az EndCli vagy EndShell parancsot alkalmazni. Természetesen ennek ellenére megmaradt az EndCli, hiszen a batch programok nem tudják használni a gadgeteket, sőt ezentúl az EndShell is önálló néven szerepel. Eddig csak a Shell-Startup hozta létre Alias-szal.

Említettem már, az Amiga DOS 2.0 parancsainak nagy része ROM rezidens, ezek nem is találhatóak még a C: könyvtárban. Az DOS 2.0 tartalmaz néhány parancsot, amelyek csak ettől a verziótól léteznek, ezen kívül a régi parancsok némelyike új opciókkal bővült, hibáit kijavították.

1. A régi parancsok

Mielőtt megismerkednénk a még nem ismert parancsokkal, nézzük meg, a régiak miként változtak. Csak azokat sorolom fel, amelyekben tényleges változtatást eszközöltek. A felsorolás ABC szerinti.

AddBuffers

A BUFFERS opció, a bufferméret megadása nem kötelező elhagyásával az aktuális bufferméretet kérdezhetjük le.

Assign

A parancs jónéhány opcióval bővült, ezek:

Dismount - eltávolítja a megadott egységet a listából. Az egységkezelő programok által lefoglalt hely felszabadul. Minden egység, például a RAM: vagy a DFO: is törölhető.

Gyakorlatilag azonos az 1.3 verziójú parancs Remove opciójával, de megbízhatóbban működik annál.

Defer - Hatására az Assign nem Volume-hoz, azaz lemeznévhez köti a logikai egységet, hanem fizikai meghajtóhoz. Ez azt jelenti például, hogy az Assign C: DF0:C Defer parancs kiadása után a rendszer mindig az éppen a DF0:-ban lévő lemez C könyvtárát tekinti C:-nek. Igen hasznos egy floppy meghajtóval, merevlemez nélkül használt gépek esetén mert így a parancsot, a szükséges library-t, stb, mindig a meghajtóban keresheti a rendszer. Ha egy logikai egység a Defer opcióval lett létrehozva az Assign listájában, ezt relációjelek közé tett elérési útvonal jelzi, például <DF0:O.

Path - Ezzel az opcióval úgy is definiálhatunk egy logikai egységet, hogy a hozzá rendelendő elérési út nem elérhető. Például, ha a fontok egy külön lemezen vannak és a Startup-sequence-ben hozzá akarjuk ehhez a lemezhez rendelni a Fonts: logikai egységet, a parancs végrehajtásakor mindig be kellene tenni a fontlemez. A Path opció megadásával a parancs elhiszi, hogy van ilyen egység, nem kéri be azt. A listában az így létrehozott logikai egység szögletes zárójelbe téve jelenik meg. Például

Assign Fonts: Fontlemez: Fontok Path

Vols

Dirs

A három kulcsszóval az Assign listájából csak a kulcsszónak megfelelő rész megjelenítését engedélyezzük. Emlékezzünk rá, hogy ez a lista három részből áll:

Volumes - A rendszerben használt lemezek nevei

Directories - A logikai egységek és a hozzájuk tartozó tartalomjegyzékek

Devices - A fizikai egységek. Például:

Assign Volumes

vagy

Assign List Volumes

Az új kulcsszavakon kívül az Assign-nak van még egy lényeges újítása, egy logikai egységhez több tartalomjegyzék tartozhat. Például:

Assign C: DF0:C RAM: RAD:

A tartalomjegyzékek felsorolása prioritási sorrendet is jelent, először az előbb állóban keres a rendszer. A többes tartalomjegyzékhez tartozik az Assign utolsó új kulcsszava az Add - Ha ezzel az opcióval adunk meg egy már létező logikai egységhez új könyvtárat, akkor megmarad a jelelegi, és hozzáfűződik az új is. Például

Assign C: RAM: Add

Avail

Egy új opciója van, a Flush - Ennek az opciónak a hatására az Avail egy "nagytakarítást" is végez a memóriában, ami abból áll, hogy a nem használt adatterületeket felszabadítja, a használaton kívüli library-kat, device-eket, handler-eket törli.

Delete

Egy új opcióval gazdagodott, amellyel a törlés ellen védett fájlokat és tartalomjegyzékeket is törölhetjük. Ez az opció a Force. Például:

Delete RAM:SSS Force

Filenote

Két új opcióval gazdagodott a program.

Ali - Ha egy könyvtár nevére kérdezzük a filenote paranccsal, az Ali opció hatására nem csak a könyvtár, hanem a benne lévő fájlok, alkönyvtárak és az azokban lévő tárgyakhoz fűzött megjegyzést is megjeleníti a parancs.

Quiet - Gyorsított megjelenítés, azok a fájlok nem jelennek meg, amelyekhez nincs megjegyzés fűzve.

Getenv

A környezeti változók kezelése a 2.0-s DOS-ban kissé megváltozott, bevezetésre került a lokális-globális változók megkülönböztetése. Erről részletesen később, a **Get** és a **Set** parancsok ismertetésénél (lásd a 176. oldalon)

A változók megkülönböztetéséhez két új opciót kínál fel a parancs:

Local - A megadott változót a lokális környezeti változók között keresi.

Global - A megadott változót a globális környezeti változók között kerese.

Ha nem adunk meg a parancsban kulcsszót, ez az alapértelmezés szerinti.

Install

Mindössze egy új opcióval gazdagodott.

FFS - FastFileSystem-et használó lemezek megfelelő bootblokkot helyez el a lemezen.

Ust

Az **ALL** opcióval bővült, amely a **Dir**-nél már megismert módon a listázott tartalomjegyzékből származó altartalomjegyzékeket is kilistázza.

Az **Lformat** opció néhány új dzsóker jellel bővült. Eddig csak a **%S** volt használható, mostmár a következők is:

%A - Protection flagok állapota

%B - A fájl mérete blokkokban, vagy a **Dir** kulcsszó

%C - A fájlok vagy tartalomjegyzékek megjegyzései

%D - Dátuminformáció

%F - Az elérési utak

%K - A kezdőblokkok sorszáma, mint a Keys opciónál (lásd a 69. oldalon).

%L - A hossz bájttban, vagy a Dir kulcsszó

%N - A fájlok és tartalomjegyzékek neve

%P - A relatív elérési útvonal

%T - A létrehozási idők

LoadWB

Két opcióval bővült a parancs.

Clean Up - A Workbench a betöltődése után azonnal végrehajt egy Clean Up műveletet, azaz rendezzi az ikonokat.

NewPath - Az újonnan meghatározott keresési útvonalak figyelembe vételével indítja a Workbenchet.

Path

A parancs a Remove opcióval bővült, amely opció használatával a megadott keresési út törölhető.

Protect

Két új opciója van a parancsnak:

Ali - A tartalomjegyzékben található összes fájl elvégzi a beállítást.

Quiet - A képernyőn nem jelenik meg a beállítás állapota, de gyorsul a művelet. Ennek elhagyásakor minden, a műveletnek alávetett fájl neve, mérete, stb kiíródik.

Remrad

Az új kickstart lehetővé teszi, hogy egyszerre több RAD: is a rendszerben legyen, természetesen különböző egységsszámokkal. Elvileg erre már az 1.3-as rendszerben is lett volna lehetőség, de nem igazán működött.

Mivel több RAD:-unk is lehet, az eltávolításkor egyértelműen meg kell adni, melyik a törlendő. Erre a célra az egységszámot kell használni a parancs argumentumaként.

A RAD:-is lehet törlésvédett, de ennek figyelmen kívül hagyására használható a Force kulcsszó.

Rename

A Quiet kulcsszóval lett kibővítvé, amely a már ismert módon elnyomja az információk kiírását.

Resident

Maga a parancs nem változott, de az általa generált listában megjelent az INTERNÁL kulcsszó a ROM rezidens parancsok megkülönböztetésére.

Search

A parancsban a Pattéra kulcsszó után keresési mintákat határozhatunk meg a már ismert dzsóker jelek használatával.

SetDate

A parancs az AH opcióval bővült, amely használatával a tartalomjegyzékben lévő összes fájl időbejegyzése beállítódik.

SetPatch

Két opcióval bővült a parancs.

Quiet - Hatása a szokásos, elnyomja a kijelzést.

NoCache, amely utasítja a parancsot, hogy ne használjon Cache-t, amely egy adat terület változó adatok számára.

Sort

Két új opcióval bővült.

Case - Hatására a parancs rendezéskor már figyelembe veszi a kis és nagy betűk közötti különbségeket.

Numeric - A számmal kezdődő sorokat rendezi el a fájl végén.

Version

A parancs négy kulcsszóval egészült ki, ezek:

File - Fájl verziószámát keresi.

Internál - ROM rezidens parancs, library, device, stb verziószámát keresi.

Res - A rezidens fájlok között keres.

Full - A teljes verzió bejegyzést megjeleníti. A verzió ugyanis nem csak a verzió számból állhat, hanem tartalmazhat dátumot, copyright bejegyzést, stb.

Az egyes kulcsszavak kombinálhatók is.

Ezzel gyakorlatilag befejeztük a már ismerős parancsok fejlődésének az ismertetését, jöhetnek az új a 2.0-s DOS-tól kezdve meglévő parancsok.

2. Az új parancsok

A könyv során többször is volt már szó a környezeti változók-ról, sőt, néhány oldallal ezelőtt felvetődött a lokális és globális környezeti változó fogalma, tisztáznunk kell a különbségeket.

A globális környezeti változó, az amit eddig is ismertünk (lásd a 124. oldalon). Ez mindig az Env: logikai egységben tároljuk, ott bármelyik taszk elérheti, innen a globális elnevezés.

A lokális környezeti változó új fogalom. Olyan változótípus, amit minden taszk a saját változóterületén, egy memóriarészen tárol. A lokális változókat csak saját taszkjuk érheti el, a taszk megszűntekor a változói törlődnek.

A lokális kezelésből adódóan két vagy több taszknak is lehet azonos nevű lokális változója, sőt ugyanilyen néven lehet egy globális környezeti változó is.

A lokális környezeti változók kezelését két új parancs szolgálja-

Set

Lokális környezeti változó beállítása. Ugyanazok a paraméterei, mint a Setenv-nek, tehát a változó neve és a hozzárendelendő érték. Az érték állhat több szóból is, nem szükséges idézőjelbe tenni. Paraméter nélkül alkalmazva az összes lokális változót felsorolja.

Get

Környezeti változó lekérdezése. Ugyanúgy használható, mint a Getenv, formátuma is azonos. Nem csak a lokális, hanem globális változó is lekérdezhető vele, a GLOBAL kulcsszó megadásával. A LOCAL kulcsszóval, vagy kulcsszó nélkül alkalmazva a lokális környezeti változókra vonatkozik.

A Shell vagy CLI ablak nyitáskor automatikusan létrejön egy process nevű lokális környezeti változó, amely az illető taszk számát tartalmazza.

A lokális változók kezeléséhez tartozik még egy dolog. A Shell vagy CLI (az azonosság miatt ezentúl csak a Shell nevet használom) ablakból NewShell, New CLI paranccsal nyitott új Shell örökli az őt létrehozó taszk lokális környezeti változóit, de a továbbiakban függetlenek egymástól. Ez az átöröklés nem jön létre, ha Workbenchből a Shellt ikonnal indítod el.

A környezeti változókhoz tartozik még két új parancs. Ezek a változók törlését végzik. Idáig nem lehetett törölni a környezeti változókat, csak null stringet tudtunk hozzá rendelni. Az új parancsokkal a tényleges törlés is megvalósítható.

Unset

Törli a parancsot használó taszk megadott nevű lokális változóját. Például:

```
Unset Process
```

Unsetenv

Törli a megadott nevű globális változót.

Ha már itt tartunk, van az Amiga DOS 2.0-nak egy harmadik új parancsa is, amely valamilyen belső DOS munkaparaméter törlésére szolgál.

Unalias

Törli az Alias-szal létrehozott makrót.

Az 1.3 operációs rendszerben nem volt lehetőség, de igény sem, hogy DOS szintjén megállapíthassuk milyen processzorú, milyen paraméterekkel rendelkező gépen dolgozunk. Azért nem volt rá szükség, mert ezzel az operációs rendszerrel egyidőben jószerivel csak 68000-as processzorú, MMU (Memory Management Unit, memória kezelő egység) nélküli gépei voltak az Amiga családnak. Az új rendszerrel új hardverek jelentek meg, amelyek már 68020, 68030 sőt 68040 processzorra is épültek, szükségessé vált tehát megtudni, milyen gépen is dolgozunk. Erre egy új parancs szolgál.

CPU

önmagában használva kiírja a processzor típusát és az ahhoz kapcsolódó értékeket, úgymint Cache állapot, MMU, FPU léte, stb. A parancsnak rengeteg opciója lehet, amelyekkel a CPU tesztet tudjuk szabályozni, vagy a processzorhoz kapcsolódó beállításokat végeztethetjük el.

Cache - Bekapcsolja a processzor általános adat és utasítás Cache-jét. Ez egy belső memória a processzorban, amely tárol bizonyos mennyiségű adatot és néhányat a soron következő utasításból. A belső adatok használatával csökken a processzor külső RAM elérésére fordítandó ideje, ami a programfűtést gyorsítja. Az utasítás cache használatára van egy megkötés, miszerint csak a 68030 processzortól fölfelé működik, mivel csak ezek tartalmazzak cache-t.

NoCache - Az előző opció ellenkezője, kikapcsolja a cache-t.

Burst - Szintén egy gyorsulást eredményező kapcsoló, de csak 68030-as processzortól felfelé működik, SCRAM-okkal.

NoBurst - Lekapcsolja az utasítás és adat burst-öt.

DataCache - Csak az adat cache-t aktiválja

NoDataCache - Kikapcsolja az adat cache-t.

DataBurst - Bekapcsolja az adat burst-öt.

NoDataBurst - Kikapcsolja az adat burst-öt

InitCache - Bekapcsolja az utasítás cache-t.

NoInitCache - Kikapcsolja az utasítás cache-t.

InitBurst - Bekapcsolja az utasítás burst-öt.

NoInitBurst - Kikapcsolja az utasítás burst-öt.

FastROM - A ROM tartalmát lemásolja a RAM-ba és ezután ónná használja. Ha a RAM 32 bites, (68020, 68030, 68040 processzorok esetén, például egy turbókártyán) ez jelentősen felgyorsítja a rendszer futását, ugyanis a RAM-ok elérése lényegesen kevesebb időt vesz igénybe, mint a ROM-oké. Az elérési időt Wait States-ekben szokás megadni. Ez azt mutatja, hogy a processzornak hány időciklust kell várnia az adat-olvasási kérelme teljesítéséhez. Megfelelően gyors RAM-oknál ez 0, vagyis ugyanabban a ciklusban megkapja az adatot, amelyikben kérte. A ROM-ok Wait States-e akár 3-5

is lehet, érezhető tehát a különbség. A 32 bites RAM-ra azért van szükség, mert maga a processzor is 32 bites. Ha a RAM 16 bites, hosszú szó olvasásához kétszer kell a memóriához fordulni, ami igencsak lelassítja a dolgot.

NoFastROM - Kikapcsolja a ROM-gyorsítást, azaz visszatér a ROM-ba, és törli a másolatot a RAM-ból, felszabadítva a lefoglalt helyet. Emlékezzünk rá, hogy a 2.0-s ROM mérete 512 kBájt, a FastROM ennyi helyet foglal el.

Trap - Nyomkövetési módba kapcsolja a processzort.

NoTrap - Visszakapcsolja a processzort a nyomkövetésből. A nyomkövetési módról részletesen a processzorról szóló könyvekben olvashatsz, lásd az irodalomjegyzéket.

NoMMUTest - A parancs ezen opció hatására nem végez MMU tesztet.

Check - A kulcsszót követően megadhatjuk, hogy a CPU parancs minek a létét vizsgálja. Ha a keresett egység megvan, a return kód 0, egyébként 5. Batch programokban használatos. A vizsgálható tárgyak:

68010

68020

68030 - A megadott processzorok létét vizsgálja.

68881

68882 - Kooprocesszor teszt.

MMU

68851 - MMU teszt.

A parancs és a Check opciója felhasználható például egy batch fájlban, amely megvizsgálja, hogy van-e a gépben matematikai kooprocesszor és ennek megfelelően indítja egy program normál vagy kooprocesszoros változatát:

```
Failat 11
```

```
CPU >NIL: Check 68881
```

```
IfWarn
```

```
    Scenery
```

```
Else
```

```
    Scenery.881
```

```
Endif
```

3. A Workbench2.0 fájljai

A 2.0 DOS ismertetése után nézzük át azokat a programokat, fájlokat, amelyeket a Workbench2.0 és Extras2.0 lemezeken találunk. Kezdjük a vizsgálódást az előbbivel.

A C tartalomjegyzék tartalmát tulajdonképpen már megismertettük, ugyanis ebben találhatóak a DOS parancsai.

A Devs tartalomjegyzékben megfigyelhetünk némi változást, a Workbench 1.3- hoz képest, nevezetesen, hogy hiányzik a Clipboard tartalomjegyzék (ezt most a startup hozza létre a RAM:-ban), és nincs Ramdrive.device (ezt a ROM tartalmazza).

Az **Expansion** szokás szerint üres, ha csak valamilyen eszközezőrlőt nem tettünk bele (lásd a Binddrivers parancsot a 112. oldalon).

A **Fonts** tartalomjegyzék most üres, a fontok egy külön lemezen lettek elhelyezve, helyhiány miatt.

Az L tartalomjegyzékből több dolog is hiányzik, ezek mindegyike be lett építve a ROM-ba. Legfontosabb közülük a **Disk-Validator**, aminek a ROM-ba költözése véget vet a Validator-típusú vírusok, mint például a Saddam, ténykedésének. Ezek a Validator-t támadták meg, cserélték le önmagukra.

A **Libs** könyvtárban is változások történtek. Az info- és icon.library-k beépültek a ROM-ba, ezek már nem találhatóak itt.

Néhány új library került be, nagyvonalakban ismerkedjünk meg ezekkel:

Asl.library - Rendszerkérdőző kezelési funkciói.

Commodities.library - Az Exchange és a commodities programok kezeléséhez szükséges funkciók.

Iffparse.library - Az Iff formátumú fájlok kezelőrutinjai.

Mathieeesingtrans.library - Matematikai rutinok

Powerpackcr.library - A népszerű tömörítőprogram ki- és becsomagoló rutinjai.

Reqtools.library - Kérdezőkezelési funkciók.

Rexxsupportlibrary

Rexxsyslib.library - Az Arexx nyelvnek és interpreterének run-tingyűjteményei.

A Monitors directory-ról már volt szó, (lásd a 151. oldalon) ez a Monitorprogramokat tárolja.

A Prefs a már ismertetett preferences editorokon és a Presets fiókon kívül, tartalmaz még egy **Env-Archive** tartalomjegyzéket is, amelyek különböző környezeti paramétereket tartalmaznak. Mellesleg ez a könyvtár az **Envarh**: logikai egységhez van rendelve.

A **Rexxc** könyvtár az Arexx-hez kapcsolódó parancsokat valamint a Rexx:-ként az Arexx programokat tartalmazza. Részletes leírásukat a **Guru** újság e témában futó sorozatában olvashatjuk. Elegendő igény esetén elképzelhető, hogy külön könyvet is megjelenetünk ebben a témában.

Az S tartalomjegyzékben a szokásos startszekvenciák vannak.

A System-ben csak azok a fájlok találhatóak, amelyeket a Workbench-nél már ismertettünk (lásd a 148. oldaltól).

A T tartalomjegyzék üres, ez csak szimbolikus jelentőségű, hisz a T:-t a RAM-ban hozzuk létre.

A Trashcan szintén üres, feladatát már ismerjük.

Az Utilities csak olyan programokat tartalmaz, amelyeket a WB-nél tárgyaltunk.

A **WBStartup** tartalomjegyzékről már volt szó, a Workbench indulásakor felhasználandó programokat rakjuk ide. Ha ez a tartalomjegyzék tartalmaz egy vagy több programot, azok a LoadWB hatására, a Workbench betöltődése után automatikusan elindulnak.

Az Extras2.0 lemez tartalmának nagy részével megismerkedtünk. Ezeken kívül van még egy Devs könyvtár is a lemezen,

DOS 2.0

amelyben a **Keymaps** és a **Printers** alkönyvtárak vannak. Ezekben találjuk meg a billentyűzet és printer meghajtó programokat, amelyekről a megfelelőt átmásolhatjuk a boot lemezünkre.

C Végző

Ezzel sajnos végére értünk a könyvnek. Tisztában vagyok vele, hogy bizonyos részeket bővebben ki kellett volna fejteni, sőt vannak olyan dolgok is, amelyekről egyáltalán nem írtam, de a könyv keretei korlátozottak, ennyi fért el. Összehasonlításképpen elmondom, hogy az Amiga 3000-s gép felhasználói kézikönyve, amely a 2.0-s kickstart, Amiga DOS és Workbench 2.0 leírását tartalmazza, több mint 1200 oldalas, B/5 formátumú könyvecske, 2.5 kg-ot meghaladó súllyal.

Kérlek kedves olvasó, ha bármilyen kritikád, észrevételed, megjegyzésed van a könyvvel kapcsolatban, ragadj tollat és írd meg, megfontoljuk észrevételedet, és felhasználjuk a sorozat további köteteinek elkészítésében.

Örömmel fogadnánk témajavaslatokat is, hisz célunk, hogy minél több Amigásnak adjunk értékes információkat gépével, annak használatával, programozásával kapcsolatban.

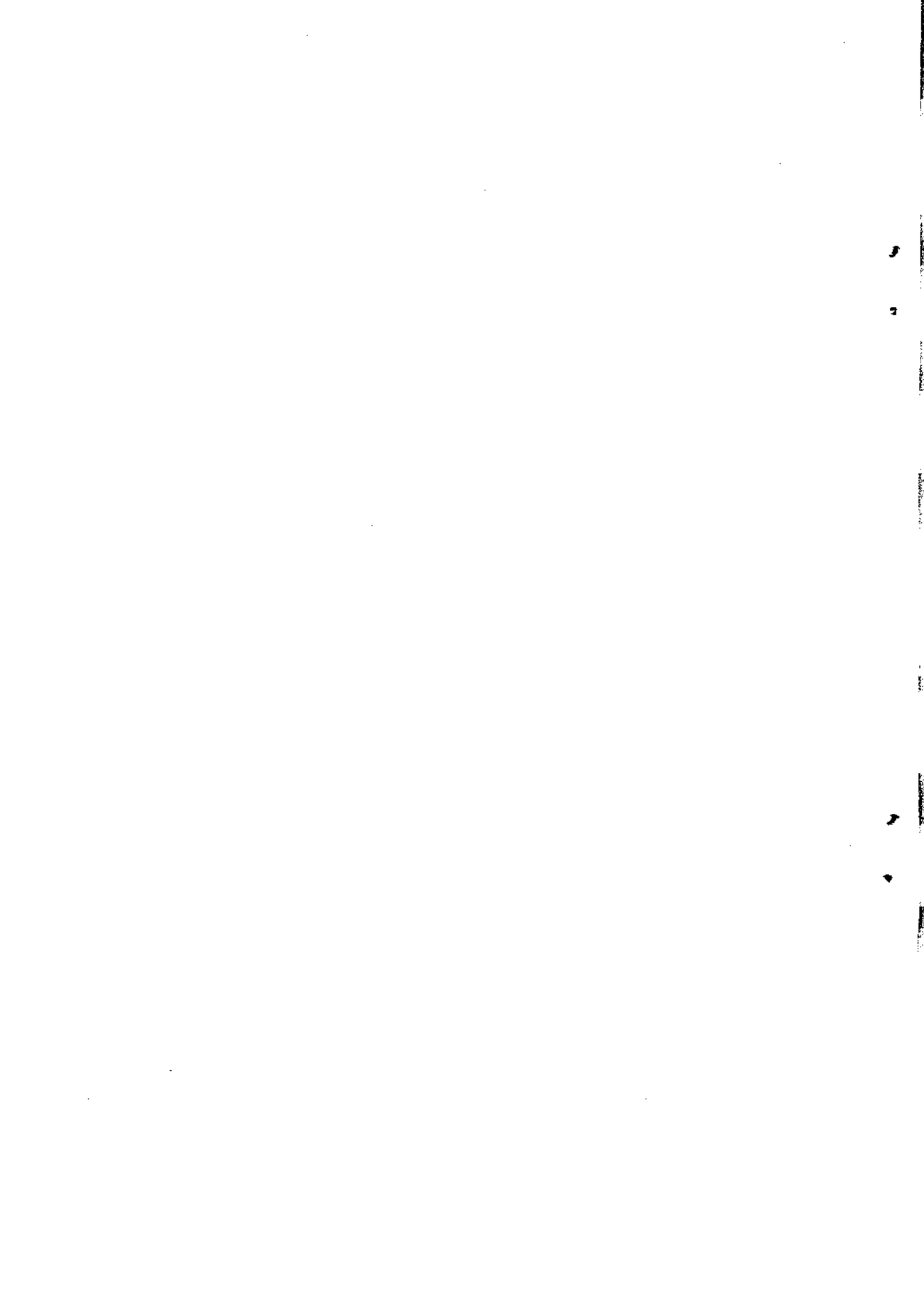
Ha nem csak témajavaslatod, de konkrét elképzelésed is van, valamint úgy érzed, elég tapasztalattal bírsz az adott tárgyban, színvonalas írás esetén akár szerzője is lehetsz az Aurum könyvek sorozat valamely kötetének, ami a jogdíjon keresztül még agyagi hasznot is jelenthet számodra.

A cím, ahová a leveleket várjuk:

AURUM DTP Stúdió

5430 Tiszaföldvár

Ószőlő Fő út 64.



IV. Függetelék**DOS hibäüzenetek**

<u>Kód</u>	<u>Megnevezés</u>
103	Insufficient free storage
105	Task table full
120	argument line invalid or too long
121	file is not an object modulé
122	invalid resident library during load
201	not default directory
202	object in use
203	object already exists
204	directory not found
205	object not found
206	bad stream name
207	object too large
209	action not known
210	invalid stream component name
211	invalid object lock
212	object not of required type
213	disk not validated
214	disk write protected
215	rename across devices
216	directory not empty
217	too many levels
218	device not mounted
219	seek error
220	comment too long
221	disk full
222	file delete protected
223	file write protected
224	file read protected
225	not a DOS disk
226	no disk

ANSI x3.64-ESC-szekvenciák jegyzéke

ANSI x3.64-ESC-szekvendák jegyzéke

<u>Megnevezés</u>	<u>ESC-szekvencia</u>	<u>funkció</u>
aRIS	ESCo	reset
aRIN	ESC	initalize
aLND	ESCD	line feed
aNEL	ESCE	return line feed
aRI	ESCM	reverse line feed
aSRGO	ESC(0m	normál character set
aSRG3	ESC(3m	italics on
aSRG23	ESC(23m	italics off
aSR4	ESC(4m	underline on
aSRG2	ESC(24	underline off
aSRG1	ESC(1m	boldface on
aSRG22	ESC(22m	boldface off
aSFC	ESC(30m-(39m	set foreground color
aBBC	ESC(40m-(49m	set background color
aSHORPO	ESC(0w	normál pitch
aSHORP	ESC(2w	elité on
aSHORP1	ESC(1w	elité off
aSHORP4	ESC(4w	condensed fine on
aSHORP3	ESC(3w	condensed off
aSHORP6	ESC(6w	enlarged on
aSHORP5	ESC(5w	enlarged off
aDEN6	ESC(6"z	shadow print on
aDEN5	ESC(5"z	shadow print off
aDEN4	ESC(4"z	doublestrike on
aDEN3	ESC(3"z	doublestrike off
aDEN2	ESC(2"z	NLQon
aDEN1	ESC(1"z	NLQoff
aSUS2	ESC(2v	superscript on

n

ANSI x3.64-ESC-szekvenciák jegyzéke

aSUS1	ESC(1v	superscript off
aSUS4	ESC(4v	subscript on
aSUS3	ESC(3v	subscript off
aSUS0	ESC(0v	nonnalize the line
aPLU	ESCL	partial line up
aPLD	ESCK	partial down
aFNTO	ESC(B	US character set
aFNT1	ESC(Rr	French character set
aFNT2	ESC(K	Germán character set
aFNT3	ESC(A	UK character set
aFNT4	ESC(E	Danish I character set
aFNT5	ESC(H	Swedish character set
aFNT6	ESC(Y	Italian character set
aFNT7	ESC(Z	Spanish character set
aFNT8	ESC(J	Japaese character set
aFNT9	ESC(G	Norwegian character set
aFNT10	ESC(C	Danish II character set
aPROP2	ESC(2p	proportional spacing on
aPROPI	ESC(1p	proportional spacing off
aPROPO	ESC(Op	proportional spacing clear
aTSS	ESC(n E	set proportional offset
aJFY5	ESC(5 F	autoleftjustify
aJFY7	ESC(7 F	autorightjustify
aJFY6	ESC(6 F	autofulljustify
aJFY0	ESC(OF	ató justify off
aJFY3	ESC(3 F	letter space /justify/
aJFYL	ESC(1 F	word fill /ató center/
aVERPO	ESC(Oz	1/8" line spacing
aVERPI	ESC(lz	1/6" line spacing
aSLPP	ESC(nt	set form length n
aPERF	ESC(nq	perf skipn(n)0
aPRFO	ESC(Oq	perf skipoff

ASCII kódtáblázat

Hsa	0	1	2	3	4	5	6	7
Q								
	a	i	£	3	l	E	t	I
1								
2	16	17	18	ú	21	22	23	
	t	"	ü		y	ÿ	i	
	32	33	31	3J	31	31	3i	32
3	0	1	2	3	4	5	6	7
	44	42	49	51	52	53	54	55
4	C	A	B	C	D	E	F	G
	64	65	66	67	68	69	70	71
5	p	Q	R	S	T	U	V	H
	82	81	12	12	»	85	86	!7
fi	'	a	b	c	d	e	f	g
	?<	97	98	99	in	101	102	103
1	p	q	r	S	t	u	v	W
	112	m	ÜL	US	IUL	m	US	112

Hesa	Q	I	2	3	i	5	6	7
'	DDDDDDDD							
Q	¹¹⁸ □	¹²⁹ □	¹³⁰ D	¹³¹ D	^{UD} ¹³³ D	^{IP} ¹³³ D	^{13*} D	¹³⁶ □
	₁₄₄	₁₄₆	₁₄₆	₁₄₇	₁₄₈	₁₄₉	_{igp}	₁₅₁
Δ	•	«	£	*	¥	!	5	
	₁₆₀	_{KI}	₁₆₂	₁₆₃	₁₈₄	_{US}	₁₆₆	₁₆₇
B	◊	±	2	3	'	μ	¶	•
	₁₇₆	₁₇₇	₁₇₈	₁₇₉	₁₈₀	_{UI}	₁₈₂	₁₈₃
C	À	Á	Ä	Å	Æ	Ç		
	₁₉₂	₁₉₃	_{»4}	₁₉₆	₁₉₆	₁₈₇	₁₉₈	_{»»}
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Ø
	_{£.0.8}	_{2.0.9}	_{2.1.0}	₂₁₁	₂₁₂	₂₁₃	₂₁₁	₂₁₅
E	à	á	ä	å	æ	ç		
	_{2.2.4}	_{2.2.5}	_{2.2.6}	₂₂₇	₂₂₈	₂₂₉	₂₃₀	₂₃₁
E	S r í ö ö í i ö ü D							
	₂₄₀	₂₄₁	₂₄₂	₂₄₃	₂₄₄	₂₄₅	₂₄₆	₂₄₇

Hexa	8	9	A	B	C	D	E	F
Q	BS s	TAB 2	U ifi	VR 11	U 12	3i 12	S£ 11	ii li
1	24	ig	25	ESC 27	28	29	30	31
2	Ü	Ü	«_»	Í	1	4	E	«_Ü
3	8	9	:	;	<	=	>	?
4	H	I	J	K	L	M	N	O
5	X	Y	Z	E	I	A		
6	h	i	J	k	i	n	n	o
7	x	y	z	í	l	}	"	■
	120	121	122	123	124	125	126	127

Hexa	S	I	A	B	C	D	E
'	DDDDDDDD						
	12í	137	13J	139	14J	141	Uí142
9	DDDDDDDD						
	152	153	154	155	156	157	158 159
A	"	©	®	™	-	®	-
	161	162	170	171	172	173	174 175
B	,	ı	o	»	¼	½	¾
	1M	UI	156	187	188	LOT	1901S1
C	<i>ttztii</i>						
	200	201	202	203	204	205	206 207
D	ø	ù	ó	ü	ý	þ	ß
	216	217	218	219	220	221	222 223
E	éééëííí						
	232	233	234	23S	23ı	23T	238 239
F	ø	ù	ó	ü	ý	þ	ß
	248	í ü	25B	25ı	2g	2fi	2g4

V. Tárgymutató

- A**
- ablak, 24
 - AddBuffers, 79, 171
 - AddMonitor, 152
 - aktuális tartalomjegyzék, 66
 - Alias, 65
 - AmigaBasic, 59
 - archiválás, 108
 - Arexx, 150
 - Aritmetikai kooprocesszor, 105
 - Arp.library, 107
 - Ask, 129
 - Asl.library, 182
 - Assign, 88, 171
 - Assigns, 32
 - Autopoint, 168
 - Aux-Handler, 97
 - AUX:, 97
 - Avail, 79, 173
- B**
- Backdrop, 142
 - Basic Demos, 60
 - batch program, 109
 - BindDrivers. 114
 - Bindmonitors, 153
 - Blanker, 168
 - blokk, 20
 - boot szektor, 20
 - boot-loader, 21
 - bootolás, 15, 19
 - botkormány, 4
 - Break, 116
- C**
- Calculator, 54
 - Cancel, 34
 - CD, 66
 - Changetaskpri, 130
 - Chip Memória, 43
 - Chip RAM, 2
 - Clean Up, 55, 145
 - CLI-Startup, 130
 - ClickToFrorit, 168
 - Clips:, 88
 - Clock, 53
 - ClockPtr, 54
 - Close, 36, 145
 - CMD, 54
 - Commodities.library, 182
 - CON:, 93
 - Continue, 34

- Copy, 148
- CPU, 180
- csatlakozók, 7
- Custom Chip, 43
- D**
- Date, 81
- Delete, 77, 173
- Dir, 66
- Discard, 55
- Diskchan^e, 131
- DiskCopyt, *íl*
- Diskcopy,'#4
- Diskdoctor, 131
- diskfont.library, 105
- DOS.library, 104
- DOS2DOS, 47
- DPAT, 130
- Drives, 31
- dupla klikk, 33
- Duplicate, 34
- dzsóker, 68
- E**
- Echo, 113
- Ed, 110
- egér, 2
- Else, 121
- Empty Trash, 149
- Emty Traschcan, 55
- Endif, 120
- Env:, 88
- Eval, 127
- Exec.library, 104
- Execute, 109
- Execute Command, 144
- Expansion, 36
- Explode.library, 107
- Extern Floppy, 7
- Extra HalfBright, 25
- F**
- Failat, 115
- fájl, 31
- Fást Memória, 43
- Fást RAM, 2
- FÁST:, 101
- FastMemFirst, 43
- Fault, 80
- Fed, 57
- FF, 114
- Filenote, 131, 173
- Fixfonts, 87
- FKey, 169
- FlickerFlixer, 25
- Font, 159
- Fontok, 50
- Formát, 43, 82
- Formát Disk, 149
- FreeMap, 59

G

gadget, 27
Getenv, 126, 174
Graphic Dump, 54
Graphich.library, 105
Guru, 44

H

hangkimenet, 7
hard diszk, 3
HDBackup, 167
HiRes, 25
Hold And Modify, 25

I

icon.library, 105
IconEd, 59
IconEdit, 165
IControl, 156
IDE, 4
if,, 120
IFF, 108
If.f.library, 108
Iffparse.library, 182
IHelp, 169
IkonMerge, 59
Info, 45, 79
info.library, 105
Information. 148
Init-Printer, 47

Input, 158

Install, 83, 174

Install Printer, 55

interlace, 25

Intuition.library, 105

Í

írásvédelem, 33

J

Jóin, 85

K

képernyő, 24

kérdező, 34

KeyTo2000, 59

KickStart, 15

Kickstart Disk, 15

kijelölés, 28

klikkelés, 28

külső floppy, 7

L

Lab, 124

Last Error, 56, 144

LeaveOut, 148

lemez puffer, 78

Libs:, 105

List, 69, 174

LoadWB, 175

Loadwb, 117

Lock, 132

- logikai egység, 32
 LoRes, 25
- M**
- Makedir, 75
 makró, 65
 mathieeedonbbas.library, 105
 mathieeedoubtrans.library, 105
 Mathieeesingtrans.library, 182
 mathtrans.library, 105
 MC 68881, 106
 MC 68882, 106
 Memacs, 57
 memória bővítés, 11
 menü, 29
 MergeMem, 44
 Messydisk.device, 102
 MessyFileSystem, 102
 minta, 69
 Monutlist, 94
 More, 52, 154
 MSH:, 102
 multi select, 42
 Multitaszk, 16
 mutató, 24
- N**
- Narrator-device, 96
 NewDrawer, 145
 NEWCON, 96
 Newcon-Hardler, 96
 NIL:, 89
 NoCapsLock, 169
 NoFastMem, 44
 normál boot, 21
 Notepad, 47
 nyomtató, 4
- O**
- Open, 30, 148
 OpenParent, 145
 Overscan, 161
 overscan, 26
- P**
- Palette, 59, 161
 PAR:, 93
 Paralel port, 8
 Parallel.device, 92, 93
 Párhuzamos port, 8
 partíció, 101
 Path, 76, 175
 pattéra, 69
 PCD, 129
 PCMCIA, 167
 PerfMon, 59
 PIPE:, 97
 Pointer, 161
 pointer, 24
 Port-Handler, 92, 93

- Port-handler, 92
- Powerpacker.library, 107, 182
- Preferences, 37, 85
- Prefs, 37
- PrepCard, 167
- Printer, 162
- Printer GFX, 162
- Printer.device, 92
- Printfiles, 55, 85
- prioritás, 16
- processzor, 1
- Prompt, 132
- Protect, 74, 175
- PRT:, 92
- Put Away, 149
- Q**
- Quit, 124, 144
- R**
- RAD:, 98
- RAM Disk, 28
- Ram-Handler, 92
- Ramlib.library, 105
- RAW:, 94
- Real-Fast Ram, 44
- Redraw, 56
- Redraw Ali, 144
- Relabe, 84
- Remrad, 99, 175
- Rename, 35, 84, 148, 176
- rendszer lemez, 15
- Req.library, 107
- Reqtools.library, 107, 182
- ResetWB, 150
- Resident, 73, 115, 176
- return kód., 89
- Rexxc, 183
- Rexxsupport.library, 183
- Rexxsyslib.library, 183
- RGB monitor, 2
- Roll Down menü, 29
- Romboot.library, 105
- Run, 64
- S**
- sáv, 20
- Say, 54
- screen, 24
- Screenmode, 162
- SCSII, 4
- Search, 132, 176
- Select Contents, 145
- SER:, 92
- Serial, 163
- Serial port, 8
- Serial.device, 92
- Set, 178
- SetClock, 114
- Setclock, 81

- SetDate, 176
 Setdate, 133
 Setenv, 126
 Setmap, 44
 SetPatch, 176
 Setpatch, 98
 Shell, 65
 Shell-seg, 65
 Shell-Startup, 130
 Shell-startup, 65
 Show, 146
 Skip, 124
 Slow-Fast Ram, 44
 Snapshot, 56, 145, 148
 soft reset, 5
 Sort, 133, 176
 SPAT, 130
 Speak-Hainder, 96
 SPEAK:, 96
 Stack, 133
 Startup-sequence, 66
 Startup-sequence.HD, 130
 Status, 134
 Sys:, 66
 szektor, 20
T
 T., 88
 tápegység, 8
 tartalomjegyzék, 30
 tasztatúra számítógép, 1
 Terminál, 59
 többes kijelölés, 29
 tömörítés, 107
 Translator.library, 96
 translatos.library, 106
 Traschan, 55
 TV modulátor, 2
 Type, 126
U
 Unalias, 179
 Unset, 179
 Unsetenv, 179
 UnSnapshot, 148
 Update, 145
 Update AU, 144
V
 várakozási mutató, 24
 vektor font, 160
 Version, 56, 81, 144
 Version, library, 106
 Videó norma
 Ntsc, 25
 Pal, 25
 View By, 146
 Volumes, 31
W
 Wait, 115

Tárgymutató

WBPatten, 164

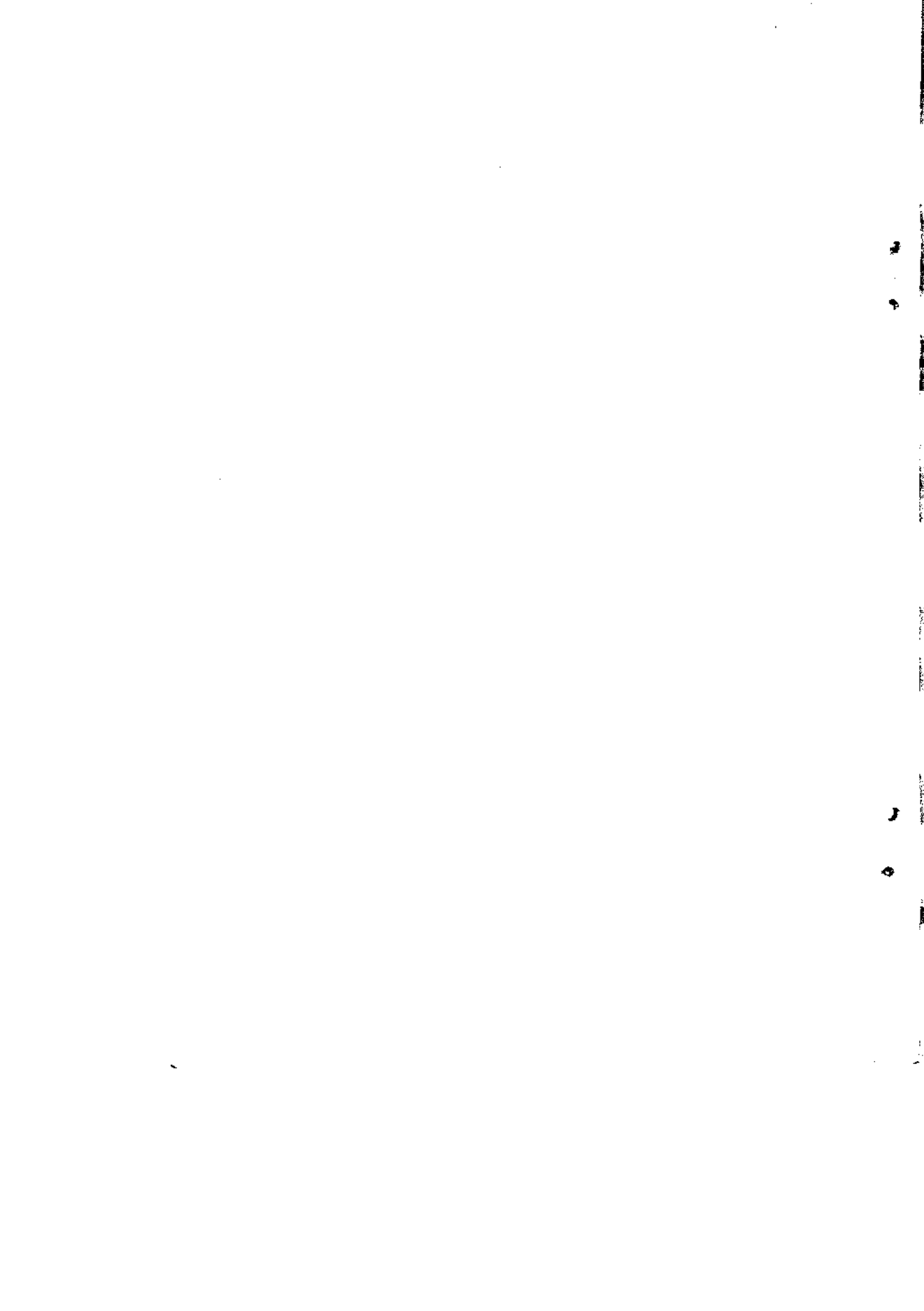
Which, 134

Why, 80

Workbench, 23

VI. Irodalomjegyzék

- A500 Benutzerhandbuch
Commodore Electronics Limited, 1988
- Handbuch zur Systemsoftware
Commodore Electronics Limited, 1991
- Amiga DOS Inside & Out
Abacus-Data Becker GmbH, 1989
- Amiga DOS Quick Reference Guide
Abacus-Data Becker GmbH, 1989
- Amiga for Beginners
Abacus-Data Becker GmbH, 1989
- Amiga Machine Language
Abacus-Data Becker GmbH, 1989
- Amiga Tricks & Tips
Abacus-Data Becker GmbH, 1989
- Arany Sándor
AMOS a Kreátor
AURUM DTP Stúdió, 1992
- Dr. Mészáros László
Nagy Amiga könyv
Műszaki Könyvkiadó, 1990
- Mészáros László
Amiga DOS, Amiga BASIC
Műszaki Könyvkiadó, 1991
- More Tricks & Tips for the Amiga
Abacus-Data Becker GmbH, 1989



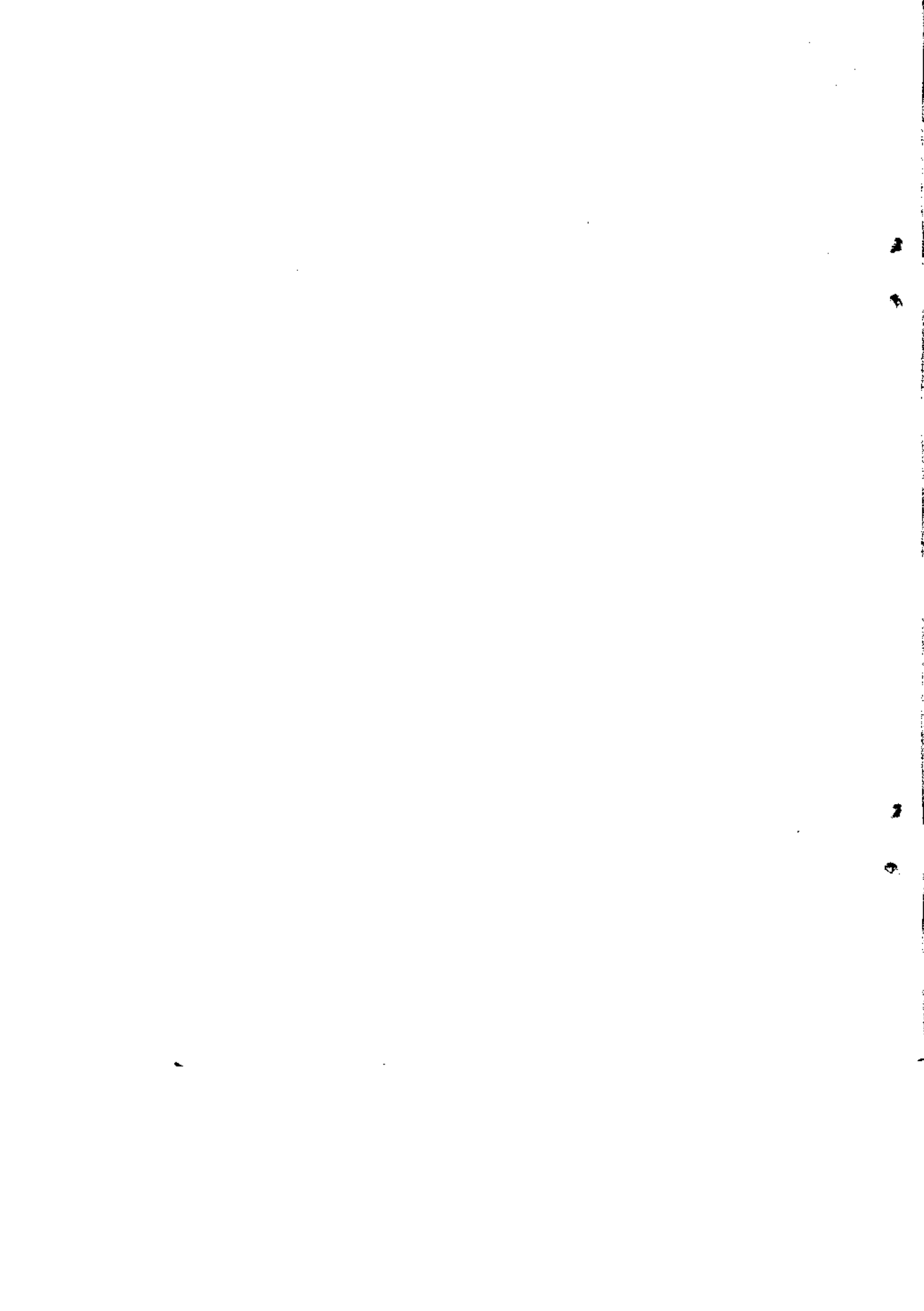
VII. Tartalomjegyzék

I. Általános leírás.....	1
A. A konfiguráció.....	1
1. A billentyűzet.....	4
2. A számítógép csatlakozói.....	6
3. Az összeszerelés művelete.....	8
B. Az első lépések.....	15
1. Multitaszk.....	16
II. Az 1.2, 1.3 kickstartú gépek használata.....	19
A. Abootolás.....	19
1. A lemez sáv-szektor szerkezete.....	20
B. AWorkbench.....	23
1. Képernyők és ablakok.....	24
2. Ikonok és menük.....	28
3. A könyvtárstruktúra.....	30
4. Tartalomjegyzék létrehozása, másolása.....	33
5. Átnevezés.....	35
6. Az ablakok bezárása.....	36
7. A Workbench beállításai.....	37
8. A printer beállításai.....	39
9. A soros port konfigurálása.....	41
10. A mutató szerkesztése.....	42
11. System tartalomjegyzék.....	42
12. Utilities tartalomjegyzék.....	47

Tartalomjegyzék

C. A DOS.....	63
1. Logikai egységek.....	86
2. Az Amiga egységei,készülékei.....	90
3. A könyvtárak.....	103
4. A batch programozás.....	109
5. Batch programok a Workbench lemezen	129
6. Az AmigaDOS egyéb parancsai.....	130
7. Tippek és Trükkök.....	134
III. A 2.0-s kickstartú gépek kezelése.....	139
A. A Wokbench2.0.....	141
1. A Workbench2.0 lemez tartalma.....	150
2. System fiók.....	150
a) Arexx.....	150
b) Monitorok.....	152
3. Utilities fiók.....	154
4. A Workbench beállításai.....	156
B. DOS 2.0.....	171
1. A régi parancsok.....	171
2. Az új parancsok.....	178
3. A Workbench2.0 fájljai.....	182
C. Végszó.....	185
IV. Függelék.....	I
DOS hibaüzenetek.....	I
ANSI x3.64-ESC-szekvenciák jegyzéke.....	II
ASCII kódtáblázat.....	IV

V. Tárgymutató.....	VIII
VI. Irodalomjegyzék.....	XV
VII. Tartalomjegyzék.....	XVII



M e g j e l e n t

Arany Sándor:
AMOS a Kreátor

Az Amos programozási nyelvet 1990 óta fejleszti a Mandarin Software. Ez egy Basic szerű programozási nyelv, amely azonban szinte teljesen kihasználja az Amiga hardverében rejlő adottságokat, gyors és könnyen kezelhető. Kezdő és középfeladók számára ideális nyelv, hisz már kis tanulás, gyakorlás után is szép kinézetű játékprogramok írására ad lehetőséget.

A nyelv mind népszerűbb, ezt bizonyítja, hogy egyre több BBS-en, PD könyvtárban, sőt kereskedelmi forgalomban jelennek meg Amos nyelven írt játékprogramok.

Az Amoshoz kapható 3D-s vektorgrafikus kiegészítő és Compiler is. A Compiler képes igen gyors, gépi kódú programok előállítására az Amos programból, a 3D kiegészítő pedig lehetővé teszi a vektoros, három dimenziós ábrázolást.

A könyv 220 oldalon ismerteti az Amos programozási nyelvet, az 1.34 verzióig. Feladatcsoportonként tagolva leírja a több mint 500 utasítás jelentését, szintaxisát és példákkal illusztrálja azok használatát.

A könyv hasznos lehet kezdőknek a nyelvvel való megismerkedéshez, haladóknak az utasítások részletekbe menő megismeréséhez. Alapos tartalomjegyzéke és tárgymutatója segítségével referencia kézikönyvként szolgálhat az Amos nyelvű programok megírásánál.

A könyv kapható a számítástechnikai könyvesboltokban, szaküzletekben, vagy megrendelhető a kiadó címén:

AURUM DTP Stúdió
5430 Tiszaföldvár
Ószőlő Fő út 64.

Ára a kiadónál 330 Ft + postaköltség.

É l ő k é s z ü l e t b e n

Arany Sándor:

Imagine 2.0

Az Imagine egy fénysugárkövetés elvén működő, ún. ray-tracer program, amellyel meglepően élethű, 3D-s képeket, animációkat lehet készíteni.

A program az egyik legnépszerűbb Amiga ray-tracer. Népszerűségét nagyfokú áttekinthetőségnek, felhasználóbarátságának, könnyű kezelhetőségének és egyedülálló animációs képességének köszönheti.

A magyar Amigások körében is igen elterjedt, már a régebbi 1.1-es verziószámú elődje sok hívet szerzett magának. Erről a programról bőszégesen jelentek meg újságcikkek, rövid ismertetőik, sőt az AURUM DTP Stúdió is jelentett meg Rick Rodriguez - Arany Sándor tollából egy részletes, teljes leírást.

Az Imagine új, 2.0-s verziószámot viselő verziójáról nagyon kevés információ jelent meg, pedig a program nem a legfrissebbek közül való.

Ez a változat lényegesen több funkcióval bír, mint az elődje, erre utal, hogy új verziószámmal látták el, nem 1.5-ös, hanem 2.0-s. Igaz, hogy tulajdonságainak egy része az előző változat ismeretében, annak leírása alapján megismerhető, de részletes, minden részletre kiterjedő használatához átfogó szakkönyvre van szükség.

Ezt a célt szolgálja az Imagine 2.0 című könyv, amely részletesen ismerteti a program használatát az alapfogalmaktól a különleges trükkökig. Példákon keresztül megismerkedhetünk minden egyes menüpont, funkció jelentésével, a kérdézőkkel, az azokban elvégezhető beállításokkal, stb.

A könyv nem száraz referencia kézikönyv, de nem is "step by step" útmutató, így jól használhatják kezdők és haladók egyaránt.

Ízelítő a tartalomjegyzékből:

- A 3 dimenziós tárgyábrázolás alapjai az Imagine-ben.
- Az editorok
- Tárgyszerkesztés
- Textúrák használata
- Animációszerkesztés
- Speciális animációs effektusok
- Az Imagine konfigurálása
- Billentyűzetfunkciók

A könyvet az Imagine-nel készült színes képek melléklete zárja, leírva az egyes képek elkészítésének műveletét, az alkalmazott fogásokat.

Megjelenésének várható ideje 1993. július.

Előkészületben

Az AURUM DTP Stúdió a következő könyvek megjelentetését tervezi:

- Imagine 2.0
- Amos kiegészítő programok
- DPaint
- Az Amiga könyvtárprogramozása

Az egyes könyvek megjelenésének pontos ideje még nem ismert (kivéve az Imagine 2.0, amely várhatóan 93. júliustól lesz kapható).

NOVOCOMP

- Számítástechnika

Commodore

Amiga

Atari

PC

Kiegészítők, lemezek, monitorszűrők, egerek,
lemeztartó dobozok, szakkönyvek, leporellók
Amigára valamint PC-re felhasználói- és
játékprogramok előnyös áron

- Irodatechnika

Panasonic telefonok, faxok
írógépek, számológépek

- CD lemezek

Lemeztartók, boxok

- Makettek

Italeri, Esci, Monogram, Hasegawa, Tamiya,
Reveu, Airfix

Repülőgépek, harcjárművek, autók, motorok
Festékek, ragasztók, kiegészítők

5000, Szolnok Kossuth L. u. 12

Tel.:56/344-605 Fax:56/370-179

NOVOCOMP