```
Directory "Lattice_C_5.0.5:Assembler_Headers/intuition" on Saturday 29-Sep-90
intuition.i                38851_____rwed 07-Nov-88 14:58:00
intuitionbase.i             1426_____rwed 07-Nov-88 14:58:01
preferences.i               8024_____rwed 07-Nov-88 14:57:58
screens.i                   4804_____rwed 07-Nov-88 14:57:59
4 files - 114 blocks - 53105 bytes
```

```
        IFND    INTUITION_INTUITION_I
INTUITIONJNTUITIONJ    SET     1
xx
**      SFilename: intuition/intuition.i $
**      SRelease: 1.3 $
xx
xx      main intuition include
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
xx

        IFND EXEC_TYPES_I
        INCLUDE "exec/types.i"
        ENDC

        IFND    GRAPHICS_GFX_I
        include "graphics/gfx.i"
        ENDC

        IFND    GRAPHICS_CLIP_I
        include "graphics/clip.i"
        ENDC

        IFND    GRAPHICS_VIEW_I
        include "graphics/view.i"
        ENDC

        IFND    GRAPHICS_RASTPORT_I
        include "graphics/rastport.i"
        ENDC

        IFND    GRAPHICS_LAYERS_I
        include "graphics/layers.i"
        ENDC

        IFND    GRAPHICS_TEXT_I
        include "graphics/text.i"
        ENDC

        IFND EXEC_PORT5_I
        include "exec/ports.i"
        ENDC

        IFND    DEVICES_TIMER_I
        include "devices/timer.i"
        ENDC

        IFND    DEVICES_INPUTEVENT_I
        include "devices/inputevent. i"
        ENDC


; ============================================================================
; === Menü ===================================================================
; ============================================================================
 STRUCTURE Menu,0

    APTR  mu_NextMenu    ; menü pointer, same levél
    WORD mu_LeftEdge    J position of the select box
    WORD mu_TopEdge     5 position of the select box
    WORD mu_Width       ; dimensions of the select box
    WORD mu_Height      ; dimensions of the select box
```

```
    WORD mu_Flags       ; see flag definitions below
    APTR mu_MenuName    ; text far this Menü Header
    APTR  mu_FirstItem  ; pointer to first in chain

    ; these mysteriously-naraed variables are,for internál use only
    WORD rau_JazzX
    WORD mu_JazzY
    WORD mu_BeatX
    WORD mu_BeatY

    LABEL mu_SIZEOF

;*** FLAGS SET BY BOTH THE APPLIPROG AND INTUITION ***
MENUENABLED EQU $0001   ; whether dr not this menü is enabled

;*** FLAGS SET BY INTUITION ***
MIDRAWN EQU $0100      \ this menü' s items are currently drawn

; === Menultera ============================================================;
; ==========================================================================;
 STRUCTURE MenüItem,0

    APTR mi_NextItem    ; pointer to next in chained list
    WORD mi_LeftEdge    ; position of the select box
    WORD mi_TopEdge     ; position of the select box
    WORD mi_Width       ; dimensions of the select box
    WORD mi_Height      ; dimensions of the select box
    WORD mi_Flags       > see the defines below

    LONG mi_MutualExclude ; set bits mean this item excludes that item

    APTR mi_ItemFill    ; points to Image» IntuiText> or NULL

    ; when this item is pointed to by the cursor and the items highlight
    ; mode HIGHIMAGE is selected} this alternate image will be displayed
    APTR mi_SelectFill  ; points to Image> IntuiText9 or NULL

    BYTE mi_Command     ; only if appliprog sets the COMMSEQ flag

    BYTE mi_KludgeFill00 ; This is strictly for word-alignment

    APTR mi_SubItem     ; if non-zero} DrawMenu shows "->"

   ; The NextSelect field represents the menü number of next selected
   ; item (when user has drag-selected several items)
    WORD mi_NextSelect

    LABEL  mi_SIZEOF

i --- FLAGS SET BY THE APPLIPROG ----------------------------------------
CHECKIT         EQU $0001       ; whether to check this item if selected
ITEMTEXT        EQU $0002       ; set if textual, clear if graphical item
COMMSEQ         EQU $0004       ; set if there's an command sequence
MENUTOGGLE      EQU $0008       ; set to toggle the check of a menü item
ITEMENABLED     EQU $0010       ; set if this item is enabled

; these are the SPÉCIAL HIGHLIGHT FLAG state meanings
HIGHFLAGS       EQU $00C0       > see definitions below for these bits
HIGHIMAGE       EQU $0000       ; use the user1s "select image"
HIGHCOMP        EQU $0040       ; highlight by complementing the select box
```

```
HIGHBOX          EQU $0080        ; highlight by drawing a box around the image
HIGHNONE         EQU $00C0        ; don't highlight

; --- FLAGS SET BY BOTH APPLIPROG AND INTUITION ---------------------------------
CHECKED          EQU $0100        ; if CHECKIT, then set this when selected


; --- FLAGS SET BY INTUITION ---------------------------------------------------
ISDRAWN          EQU $1000        ; this item's subs are currently drawn
HIGHITEM         EQU $2000        ; this item is currently highlighted
MENUTOGGLED      EQU $4000        ; this item was already toggled




; ==============================================================================
; === Requester ================================================================
; ==============================================================================
 STRUCTURE Requester,0

    ; the ClipRect and BitMap and used for rendering the requester
    APTR  rq_OlderRequest
    WORD rq_LeftEdge             ; dimensions of the entire box
    WORD rq_TopEdge              ; dimensions of the entire box
    WORD rq_Width                ; dimensions of the entire box
    WORD rq_Height               ; dimensions of the entire box

    WORD rq_RelLeft              ; get POINTREL Pointer relativity offsets
    WORD rq_RelTop               ; get POINTREL Pointer relativity offsets

    APTR  rq_ReqGadget           ; pointer to the first of a list of gadgets
    APTR  rq_ReqBorder           ; the box's border
    APTR  rq_ReqText             ; the box's text

    WORD  rq_Flags               ; see definitions below

    UBYTE rq_BackFill            ; pen number for back-plane fill before draws

    BYTE rq_KludgeFill00         ; This is strictly for word-alignment

    APTR rq_ReqLayer             ; layer in which requester rendered
    STRUCT rq_ReqPad1,32         ; for backwards compatibility (reserved)

    ; If the BitMap plane pointers are non-zero, this tells the system
    ; that the image comes pre-drawn (if the appliprog wants to define
    ; it's own box, in any shape or size it wants!); this is OK by
    ; Intuition as long as there's a good correspondence between the image
    ; and the specified Gadgets
    APTR  rq_ImageBMap           ; points to the BitMap of PREDRAWN imagery

    APTR  rq_RWindow             ; points back to requester's window
    STRUCT rq_ReqPad2,36         ; for backwards compatibility (reserved)

    LABEL rq_SIZEOF

; FLAGS SET BY THE APPLIPROG
POINTREL        EQU $0001  ; if POINTREL set, TopLeft is relative to pointer
PREDRAWN        EQU $0002  ; if ReqBMap points to predrawn Requester imagery
NOISYREQ        EQU $0004  ; if you don't want requester to filter input
```

```
; FLAGS SET BY INTUITION;
REQOFFWINDOW    EQU $1000       j part of one of the Gadgets was offwindow
REQACTIVE       EQU $2000       > this requester is active
SYSREQUEST      EQU $4000       > this requester caused by system
DEFERREFRESH    EQU $8000       ; this Requester stops a Refresh broadcast



; ============================================================================
; === Gadget =================================================================
; ============================================================================
 STRUCTURE Gadget,0

    APTR gg_NextGadget          ; next gadget in the list

    WORD gg_LeftEdge            ; "hit box" of gadget
    WORD gg_TopEdge             ', "hit box" of gadget
  . WORD gg_Width               ', "hit box" of gadget
    WORD gg_Height              ; "hit box" of gadget

    WORD gg_Flags               ; see below for list of defines

    WORD gg_Activation          ; see below for list of defines

    WORD gg_GadgetType          ; see below for defines

    ; appliprog can specify that the Gadget be rendered as either as Bordér
    ; or an Image.  This variable points to which (or equals NULL if there's
    ; nothing to be rendered about this Gadget)
    APTR gg_GadgetRender

    ; appliprog can specify "highlighted" imagery rather than algorithmic
    ; this can point to either Bordér or Image data
    APTR gg_SelectRender

    APTR gg_GadgetText          ; text for this gadget',

    ; by using the MutualExclude word, the appliprog can describe
    ; which gadgets mutually-exclude which other ones.  The bits in
    ; MutualExclude correspond to the gadgets in object containing
    ; the gadget list.  If this gadget is selected and a bit is set
    ; in this gadget's MutualExclude and the gadget corresponding to
    ; that bit is currently selected (e.g. bit 2 set and gadget 2
    ', is currently selected) that gadget must be unselected.  Intuition
    » does the visual unselecting (with checkmarks) and leaves it up
    ; to the program to unselect internally
    LONG gg_MutualExclude       j set bits mean this gadget excludes that

    ; pointer to a structure of special data required by Proportional, String
    ; and Integer Gadgets
    APTR gg_SpecialInfo

    WORD gg_GadgetID    ; user-definable ID field
    APTR  ggJJserData   ; ptr to generál purpose User data (ignored by Pntuit)

    LABEL gg_SIZEOF

; --- FLAGS SET BY THE APPLIPROG ---------------------------------------------
; combinations in these bits describe the highlight technique to be used
```

```
GADGHIGHBITS    EQU $0003
GADGHCOMP       EQU $0000       > Complement the select box
GADGHBOX        EQU $0001       ; Draw a box around the image
GADGHIMAGE      EQU $0002       ; Blast in this alternate image
GADGHNONE       EQU $0003       ", don't highlight

; set this flag if the GadgetRender and SelectRender point to Image imagery5
; clear if it's a Bordér
GADGIMAGE       EQU $0004

j combinations in these next two bits specify to which corner the gadget's
j Left & Top coordinates are relative.  If relative to Top/Left,
; these are "normál" coordinates (everything is relative to something in
; this universe)
GRELBOTTOM      EQU $0008       ; set if rel to bottom, clear if rel top
GRELRIGHT       EQU $0010       ; set if rel to right, clear if to left
;, set the RELWIDTH bit to spec that Width is relative to width of screen
GRELWIDTH       EQU $0020
; set the RELHEIGHT bit to spec that Height is rel to height of screen
GRELHEIGHT      EQU $0040

; the SELECTED flag is initialized by you and set by Intuition.  It
; specifies whether or not this Gadget is currently selected/highlighted
SELECTED        EQU $0080


; the GADGDISABLED flag is initialized by you and later set by Intuition
i according to your calls to On/OffGadget().  It specifies whether or not
; this Gadget is currently disabled frora being selected
GADGDISABLED    EQU $0100


j — These are the Activation flag bits ————————————————————
i, RELVERIFY is set if you want to verify that the pointer was still over
; the gadget when the select button was released
RELVERIFY       EQU $0001

S the flag GADGIMMEDIATE> when set, informs the caller that the gadget
j was activated when it was activated.  this flag works in conjunction with
; the RELVERIFY flag
GADGIMMEDIATE   EQU $0002

; the flag ENDGADGET, when set, telis the system that this gadget, when
i selected, causes the Requester or AbsMessage to be ended.  Requesters or
j AbsMessages that are ended are erased and unlinked from the system
ENDGADGET       EQU $0004

j the FOLLOWMOUSE flag1, when set, specifies that you want to receive
> reports on mouse movements (ie, you want the REPORTMOUSE function for
; your Window).  When the Gadget is deselected (imraediately if you have
", no RELVERIFY) the previous state of the REPORTMOUSE flag is restored
; You probably want to set the GADGIMMEDIATE flag when using FOLLOWMOUSE,
; since that's the only reasonable way you have of learning why Intuition
; is suddenly sending you a stream of mouse movement events.  If you don't
't set RELVERIFY, you'11 get at least one Mouse Position event.
FOLLOWMOUSE     EQU $0008

; if any of the BORDÉR flags are set in a Gadget thaf's included in the
;, Gadget list when a Window is opened, the corresponding Bordér will
; be adjusted to make room for the Gadget
RIGHTBORDER     EQU $0010
```

```
LEFTBORDER      EQU $0020
TOPBORDER       EQU $0040
BOTTOMBORDER    EQU $0080

TOGGLESELECT    EQU $0100       > this bit for toggle-select mode

STR INGCENTER   EQU $0200       ; center the String
STRINGRIGHT     EQU $0400
                                ; right-justify the String
LONGINT         EQU $0800
                                5 This String Gadget is a Long Integer
ALTKEYMAP       EQU $1000       '
                                > This String has an alternate keymapping
BOOLEXTEND      EQU $2000       '
                                > This Boolean Gadget has a Boollnfo
i — GADGET TYPES _____----------------------------------------------
> These are the Gaget Type definitions for the variable GadgetType.
j Gadget nuraber type MUST start from one.  NO TYPES OF ZERO ALLOWED.
;, first comes the mask for Gadget flags reserved for Gadget typing
GADGETTYPE      EQU $FCOO       > all Gadget Global Type flags (padded)
SYSGADGET       EQU $8000       ; 1 = SysGadget, 0 = AppliGadget
SCRGADGET       EQU $4000       ; 1 = ScreenGadget> 0 = WindowGadget
GZZGADGET       EQU $2000       ; 1 = Gadget for GIMMEZEROZERO borders
REQGADGET       EQU $1000       ; 1 = this is a Requester Gadget
; system gadgets
SIZING-         EQU $0010
TORAGGING       EQU $0020
SDRAGGING       EQU $0030
WUPFRONT        EQU $0040
SUPFRONT        EQU $0050
WDOWNBACK       EQU $0060
SDOWNBACK       EQU $0070
CLOSE           EQU $0080
; application gadgets
BOOLGADGET      EQU $0001
GADGET0002      EQU $0002
PROPGADGET      EQU $0003
STRGADGET       EQU $0004



; ==========================================================================
; === Boollnfo=============================================================
; ==========================================================================
:1 This is the special data needed by an Extended Boolean Gadget
:
, Typically this structure will be pointed to by the Gadget field SpecialInfo

 STRUCTURE Boollnfo,0

    WORD    bi_Flags    ; defined below
    APTR    bi_Mask     ; bit mask for highlighting and selecting
                        ; mask must follow the same rules as an Image
                        ; pláne.  It's width and height are determined
                        ; by the width and height of the gadget's
                        ', select box. Ci.e. Gadget.Witlth and ..Height).
    LONG    bi_Reserved ; set to 0
    LABEL   bi_SIZEOF

:i set Boollnfo.Flags to this flag bit.
:, in the future> additional bits might mean more stuff hanging
```

```
; off of BoolInfo.Reserved.

BOOLMASK        EQU     $0001   >' extension is for masked gadget

; ==========================================================================
; === Propinfo ============================-=---=-=-=--=-==================
; ==========================================================================
; this is the special data required by the proportional Gadget
; typically, thís data will be pointed to by the Gadget variable SpecialInfo
 STRUCTURE Proplnfo,0

    WORD pi_Flags       ; generál purpose flag bits (see defines below)

    ; You initialize the Pot variables before the Gadget is added to
    ', the system.  Then you can look here for the current settings
    ; any time, evén while User is playing with this Gadget.  To
    ; adjust these after the Gadget is added to the System, use
    ; ModifyPropC)", The Pots are the actual proportional settings,
    ; where a value of zero means zero and a value of MAXPOT means
    ; that the Gadget is set to its maximum setting.
    WORD pi_HorizPot    5 16-bit FixedPoint horizontal quantity percentage;
    WORD pi_VertPot     » 16-bit FixedPoint verticai quantity percentagej

    ; the 15-bit FixedPoint Body variables describe what percentage
    ; of the entire body of stuff referred to by this Gadget is
    "> actually shown at one time.  This is used with the AUTOKNOB
    > routines, to adjust the size of the AUTOKNOB according to how
    ; much of the data can be seen.  This is alsó used to decidé how
    ; far to advance the Pots when User hits the Container of the Gadget.
    ; For instance, if you were controlling the display of a 5-line
    ; Window of text with this Gadget> and there was a totál of 15
    ;  lines that could be displayed} you would set the VertBody value to
    ;    (MAXBODY / (TotalLines / DisplayLines)) = MAXBODY / 3.
    ; Therefore, the AUTOKNOB would fill 1/3 of the container, and if
    ; User hits the Cotainer outside of the knob, the pot would advance
    ; 1/3 (plus or minus) If there[1]s no body to show> or the totál
    •> amaunt of displayable info is less than the display area, set the
    ; Body variables to the MAX.  To adjust these after the Gadget is
    ; added to the System» use ModifyPropC).
    WORD pi_HorizBody   ', horizontal Body
    WORD pi_VertBody    ; verticai Body >

    \ these are the variables that Intuition sets and maintains
    WORD pi_CWidth      ; Container width (with any relativity absoluted)
    WORD pi_CHeight     ; Container height (with any relativity absoluted)
    WORD pi_HPotRes     \ pot increments
    WORD pi_VPotRes     'J pot increments
    WORD pi_LeftBorder  > Container borders
    WORD pi_TopBorder   > Container borders
    LABEL  pi_SIZEOF

» --- FLAG BITS ----------------------------------------------------------
AUTOKNOB        EQU $0001       ; this flag sez:  gimme that old auto-knob
FREEHORIZ       EQU $0002       ; if set, the knob can move horizontally
FREEVERT        EQU $0004       ; if set, the knob can move vertically
PROPBORDERLESS  EQU $0008       ; if set, no bordér will be rendered
KNOBHIT         EQU $0100       ; set when this Knob is hit


KNOBHMIN        EQU 6           ; minimum horizontal size of the knob
KNOBVMIN        EQU 4           ; minimum verticai size of the knob
```

```
MAXBODY         EQU $FFFF       > maximum body value
MAXPOT          EQU $FFFF       ; maximum pot value


; ============================================================================
; === StringInfo ============================================================
; ============================================================================
; this is the special data required by the string Gadget
; typically, this data will be pointed to by the Gadget variable SpecialInfo
 STRUCTURE StringInfo,0

    ; you initialize these variables, and then Intuition maintains them
    APTR  si_Buffer     ; the buffer containing the start and final string
    APTR  si_UndoBuffer ; optional buffer for undoing current entry
    WORD si_BufferPos   ; character position in Buffer
    WORD si_MaxChars    ; max number of chars in Buffer (including NULL)
    WORD si_DispPos     ; Buffer position of first displayed character

    ; Intuition initializes and maintains these variables for you
    WORD siJJndoPos     ; character position in the undo buffer
    WORD si_NumChars    ; number of characters currently in Buffer
    WORD si_DispCount   ; number of whole characters visible in Container
    WORD si_CLeft       ; topleft offset of the container
    WORD si_CTop        ; topleft offset of the container
    APTR  siJLayerPtr   ; the RastPort containing this Gadget

    ; you can initialize this variable before the gadget is submitted to
    ; Intuition and then examine it later to discover what integer
    ; the user has entered (if the user never plays with the gadget,
    ; the value will be unchanged from your initial setting)
    LONG  si_LongInt    ; the LONG return value of a LONGINT String Gadget

    ; If you want this Gadget to use your own Console keymapping, you
    ; set the ALTKEYMAP bit in the Activation flags of the Gadget, and then
    ; set this variable to point to your keymap.  If you don't set the
    ; ALTKEYMAP, you'll get the standard ASCII keymapping.
    APTR si_AltKeyMap

    LABEL si_SIZEOF


; ============================================================================
; === IntuiText ==============================================================
; ============================================================================
; IntuiText is a series of strings that start with a screen location
; (always relative to the upper-left corner of something) and then the
; text of the string.  The text is null-terminated.
 STRUCTURE IntuiText,0

    BYTE it_FrontPen            ; the pens for rendering the text
    BYTE it_BackPen             ; the pens for rendering the text

    BYTE it_DrawMode            ; the mode for rendering the text

    BYTE itJQudgeFillOO         ; This is strictly for word-alignment

    WORD it_LeftEdge            ; relative start location for the text
    WORD it_TopEdge             ; relative start location for the text
```

```
    APTR  it_ITextFont          > if NULL> you accept the defaults
    APTR  itJText               ; pointer to null-terminated text
    APTR  it_NextText           ; continuation to TxWrite another text
    LABEL it_SIZEOF

;============================================================================
;=== Border ================================================================
;============================================================================
; Data type Bordér> used for drawing a series of lines which is intended for
; use as a bordér drawing> but which may, in fact, be used to render any
; arbitrary vector shape.
; The routine DrawBorder sets up the RastPort with the appropriate
; variables, then does a Move to the first coordinate, then does Draws
; to the subsequent coordinates.
; After all the Draws are done> if NextBorder is non-zero we call DrawBorder
; recursively
 STRUCTURE Bordér,0

    WORD  bd_LeftEdge           ; initial offsets from the origin
    WORD  bd_TopEdge            ; initial offsets from the origin
    BYTE  bd_FrontPen           ; pen number for rendering
    BYTE  bd_BackPen            J pen number for rendering
    BYTE  bd_DrawMode           j mode for rendering
    BYTE  bd_Count              ; number of XY pairs
    APTR  bd_XY                 ; vector coordinate pairs rel to LeftTop
    APTR  bd_NextBorder         ; pointer to any other Bordér too

    LABEL bd_SIZEOF

;============================================================================
; === Image =================================================================
;============================================================================
; This is a brief image structure for very simple transfers of
i image data to a RastPort
 STRUCTURE Image,0

    WORD ig..LeftEdge           j starting offset relative to something
    WORD ig"_TopEdge            ; starting offset relative to something
    WORD ig_"Width              ; pixel size (though data is word-aligned)
    WORD ig"..Height            ; pixel size
    WORD ig~JJepth              ; pixel size
    APTR ig._ImageData          J pointer to the actual image bits

    ; the PlanePick and PlaneOnOff variables work much the same way as the
    ; equivalent GELS Bob variables.  IVs a space-saving
    ; mechanism for image data.  Rather than defining the image data
    ; for every pláne of the RastPort> you need define data only for planes
    ; that are not entirely zero or one.  As you define your Imagery, you will
    ; often find that most of the planes ARE just as color selectors.  For
    ; instance> if you're designing a two-color Gadget to use colors two and
    ; three, and the Gadget will reside in a five-plane display, pláne zero
    ; of your imagery would be all ones, bit pláne one would have data that
    ; describes the imagery, and bit planes two through four would be
```

```
    ; all zeroes.  Using these flags allows you to avoid wasting all that
    i memory in this way:
    ; first> you specify which planes you want your data to appear
    ; in using the PlanePick variable.  For each bit set in the variable, the
    ; next "pláne" of your image data is blitted to the display.  For each bit
    ; clear in this variable, the corresponding bit in PlaneOnOff is examined.
    ; If that bit is clear, a "pláne" of zeroes will be used.  If the bit is
    ; set, ones will go out instead.  So, for our example:
    ;   Gadget.PlanePick = 0x02;
    ;   Gadget.PlaneOnOff = 0x01;
    ; Note that this alsó allows for generic Gadgets, liké the System Gadgets,
    ; which will work in any number of bit planes
    ; Note alsó that if you want an Image that is only a fiiled rectangle,
    ', you can get this by setting PlanePick to zero (pick no planes of data)
    ', and set PlaneOnOff to describe the pen color of the rectangle.
    BYTE ig_PlanePick•
    BYTE ig_PlaneOnOff

    ; if the Nextlmage variable is not NULL, Intuition presumes that
    ; it points to another Image structure with another Image to be
    j rendered
    APTR ig_NextImage


    LABEL ig_SIZEOF



; ============================================================================
; === IntuiMessage ===========================================================
; ============================================================================
 STRUCTURE IntuiMessage,0

    STRUCT im_ExecMessage,MN_SIZE

    ', the Class bits correspond directly with the IDCMP Flags, except for the
    ; special bit LONELYMESSAGE (defined below)
    LONG im_Class

    ; the Code field is for special values liké MENÜ number
    WORD im_Code

    ; the Qualifier field is a copy of the current InputEvent's Qualifier
    WORD im_Qualifier

    ; IAddress contains particular addresses for Intuition functions, liké
    ; the pointer to the Gadget or the Screen
    APTR im_IAddress

    ; when getting mouse mdvement reports, any event you get will have the
    ; the mouse coordinates in these variables.  the coordinates are relatíve
    ; to the upper-left corner of your Window CGIMMEZEROZERO notwithstanding)
    WORD im_MouseX
    WORD im_MouseY

    ; the time values are copies of the current system clock time.  Micros
    ; are in units of microseconds, Seconds in seconds.
    LONG im_Seconds
    LONG im_Micros
```

```
    ; the IDCMPWindow variable will always have the address of the Window of
    ; this IDCMP
    APTR  im_IDCMPWindow

    ', system-use variable
    APTR  ira_SpecialLink

    LABEL   im_SIZEOF


; --- IDCMP Classes ----------------------------------------------------
SIZEVERIFY      EQU     $00000001       } See the Programmer's Guide
NEWSIZE         EQU     $00000002       j See the Programmer's Guide
REFRESHWINDOW   EQU     $00000004       ; See the Programmer's Guide
MOUSEBUTTONS    EQU     $00000008       ; See the Programmer's Guide
MOUSEMOVE       EQU     $00000010       ; See the Programmer's Guide
GADGETDOWN      EQU     $00000020       } See the Programmer's Guide
GADGETUP        EQU     $00000040       ; See the Programmer's Guide
REQSET          EQU     $00000080       ; See the Programmer's Guide
MENUPICK        EQU     $00000100       ; See the Programmer's Guide
CLOSEWINDOW     EQU     $00000200       ; See the Programmer's Guide
RAWKEY          EQU     $00000400       > See the Programmer's Guide
REQVERIFY       EQU     $00000800       ; See the Programmer's Guide
REQCLEAR        EQU     $00001000       > See the Programmer's Guide
MENUVERIFY      EQU     $00002000       ; See the Programmer's Guide
NEWPREFS        EQU     $00004000       ; See the Programmer's Guide
DISKINSERTED    EQU     $00008000       ; See the Programmer's Guide
DISKREMOVED     EQU     $00010000       > See the Programraer's Guide
WBENCHMESSAGE   EQU     $00020000       > See the Programmer's Guide
ACTIVEWINDOW    EQU     $00040000       ; See the Programmer's Guide
INACTIVEWINDOW  EQU     $00080000       ; See the Programmer's Guide
DELTAMOVE       EQU     $00100000       » See the Programmer's Guide
VANILLAKEY      EQU     $00200000       ; See the Programmer's Guide
INTUITICKS      EQU     $00400000       \ See the Programmer's Guide
; NOTEZ-BIEN:           $80000000 is reserved for internál use by IDCMP

; the IDCMP Flags do not use this special bit, which is cleared when
"> Intuition sends its special message to the Task> and set when Intuition
; gets its Message back from the Task.  Therefore> I can check here to
; find out fást whether or not this Message is available for me to send
LONELYMESSAGE   EQU     $80000000


; --- IDCMP Codes ------------------------------------------------------
•> This group of codes is for the MENUVERIFY function
MENUHOT         EQU     $0001   ;  IntuiWants verification or MENUCANCEL
MENUCANCEL      EQU     $0002   ; HOT Reply of this cancels Menü operation
MENUWAITING     EQU     $0003   ;  Intuition simply wants a ReplyMsgC) ASAP

; These are internál tokens to represent state of verification attempts
; shown here as a clue.
OKOK            EQU     MENUHOT         ; guy didn't care
OKABORT         EQU     $0004           ; window rendered question moot
OKCANCEL        EQU     MENUCANCEL      ; window sent cancel reply

; This group of codes is for the WBENCHMESSAGE messages
WBF.NCHOPEN     EQU $0001
WBENCHCLOSE     EQU $0002
```

```
; ============================================================================
; ===Window======_============================-====-========================
; ============================================================================
 STRUCTURE Window,0

    APTR wd_NextWindow          ', for the linked list of a Screen

    WORD wd_LeftEdge            ; screen dimensions
    WORD wdJTopEdge             i screen dimensions
    WORD wd_Width               > screen dimensions
    WORD wdjteight              » screen dimensions

    WORD wd_MouseY              ; relative top top-left corner
    WORD wd_MouseX              ; relative top top-left corner

    WORD wd_MinWidth            ; minimum sizes
    WORD wd_MinHeight           ; minimum sizes
    WORD wd_MaxWidth            ; maximum sizes
    WORD wd_MaxHeight           ; maximum sizes

    LONG wd_Flags               ; see below for definitions

    APTR wd_MenuStrip           ; first in a list of menü headers

    APTR wd_Title               ; title text for the Window

    APTR wd_FirstRequest        ", first in linked list of active Requesters
    APTR wd_DMRequest           ; the double-menu Requester
    WORD wd_ReqCount            ; number of Requesters blocking this Window
    APTR wd_WScreen             ; this Window's Screen
    APTR wd_RPort               ; this Window's very own RastPort

    > the bordér variables describe the window bordér.  If you specify
    > GIMMEZEROZERO when you open the window, then the upper-left of the
    ; ClipRect for this window will be upper-left of the BitMap (with correct
    ; offsets when in SuperBitMap mode; you MUST select GIMMEZEROZERO when
    ; using SuperBitMap).  If you don't specify ZeroZero, then you savé
    ; memory (no allocation of RastPort, Layer, ClipRect and associated
    ; Bitmaps), but you alsó must offset all your writes by BorderTop,
    ; BorderLeft and do your own mini-clipping to prevent writing over the
    ; system gadgets
    BYTE wd_BorderLeft
    BYTE wd_BorderTop
    BYTE wd_BorderRight
    BYTE wd_BorderBottom
    APTR wd_BorderRPort

    ', You supply a linked-list of gadget that you want for your Window.
    ; This list DOES NOT include system Gadgets.  You get the standard
    ; window system Gadgets by setting flag-bits in the variable Flags.(see
    j the bit definitions below)
    APTR wd_FirstGadget

    ; these are for opening/closing the windows
    APTR wd_Parent
    APTR wd_Descendant

    ; sprite data information for your own Pointer
    ; set these AFTEJl you Open the Window by calling SetPointerO
    APTR wd_Pointer
    BYTE wd_PtrHeight
```

```
    BYTE wd_PtrWidth
    BYTE wd_XOffset
    BYTE wd_YOffset

    ; the IDCMP Flags and User's and Intuition's Message Ports
    ULONG wd_IDCMPFlags
    APTR wd_UserPort
    APTR wd_WindowPort
    APTR wd_MessageKey

    BYTE wd_DetailPen
    BYTE wd_BlockPen

    ; the CheckMark is a pointer to the imagery that will be used when
    ; rendering MenuItems of this Window that want to be checkmarked
    j if this is equal to NULL, you'll get the default imagery
    APTR wd_CheckMark

    j if non-null, Screen title when Window is active
    APTR wd_ScreenTitle

    ; These variables have the mouse coordinates relative to the
    ; inner-Window of GIMMEZEROZERO Windows.  This is compared with the
    ; MouseX and MouseY variables, which contain the mouse coordinates
    ; relative to the upper-left corner of the Window, GIMMEZEROZERO
    ', notwithstanding
    WORD wd_GZZMouseX
    WORD wd_GZZMouseY
    ', these variables contain the width and height af the inner-Window of
    ; GIMMEZEROZERO Windows
    WORD wd_GZZWidth
    WORD wd_GZZHeight

    APTR wd_ExtData

    ; generál-purpose pointer to User data extension
    APTR wd_UserData
    APTR wd_WLayer      ; stash of Window.RPort->Layer          *

    ; NEW 1.2: need to keep track of the font that OpenWindow opened,
    ; in case user SetFont's intő RastPort
    APTR IFont

    LABEL-wd_Size

', --- FLAGS REQUESTED (NOT DIRECTLY SET THOUGH) BY THE APPLIPROG ---
WINDOWSIZING    EQU $0001       ; include sizing system-gadget?
WINDOWDRAG      EQU $0002       ; include dragging system-gadget?
WINDOWDEPTH     EQU $0004       ; include depth arrangement gadget?
WINDOWCLOSE     EQU $0008       ; include close-box system-gadget?

SIZEBRIGHT      EQU $0010       > size gadget uses right bordér
SIZEBBOTTOM     EQU $0020       ; size gadget uses bottom bordér

; — refresh modes ——————————————————————————————
; combinations of the REFRESHBITS select the refresh type
REFRESHBITS     EQU $00C0
SMART_REFREáH   EQU $0000
SIMPLE_REFRESH  EQU $0040
SUPER_BITMAP    EQU $0080
OTHER_REFRESH   EQU $00C0
```

```
BACKDROP        EQU $0100       ; this is an ever-popular BACKDROP window

REPORTMOUSE     EQU $0200       \ set this to hear about every mouse move

GIMMEZEROZERO   EQU $0400       ', make extra bordér stuff

BORDERLESS      EQU $0800       j set this to get a Window sans bordér

ACTIVATE        EQU $1000       ; when Window opens, it's the Active one

; FLAGS 5ET BY INTUITION
WINDOWACTIVE    EQU $2000       ; this window is the active one
INREQUEST       EQU $4000       ", this window is in request mode
MENUSTATE       EQU $8000       ; this Window is active with its Menüs on

; — Other User Flags ----------------------------------------------
RMBTRAP         EQU $00010000   ; Catch RMB events for your own
NOCAREREFRESH   EQU $00020000   ; not to be bothered with REFRESH

; — Other Intuition Flags -----------------------------------------
WINDOWREFRESH   EQU $01000000   ; Window is currently refreshing
WBENCHWINDOW    EQU $02000000   ; WorkBench Window
WINDOWTICKED    EQU $04000000   ; only one timer tick at a tiræ

SUPERJJNUSED    EQU $FCFCOOOO   Jbits of Flag unused yet


', — see struct IntuiMessage for the IDCMP Flag definitions -----------------


; ================================================================================
:> = = = NewWindow ==========-----==========-=========--=======-===========
'; ================================================================================
 5TRUCTURE NewWindow,0

    WORD nw_LeftEdge            > initial Window dimensions
    WORD nw_TopEdge             ; initial Window dimensions
    WORD nw_Width               ; initial Window dimensions
    WORD nw_Height              ; initial Window dimensions

    BYTE nw_DetailPen           > for rendering the detail bits of the Window
    BYTE nw_BlockPen            ; for rendering the block-fill bits

    LONG nw_IDCMPFlags          ; initial IDCMP state

    LONG nw_Flags               ; see the Flag definition under Window

    ; You supply a linked-list of Gadgets for your Window.
    ; This list DOES NOT include system Gadgets.  You get the-standard
    ; system Window Gadgets by setting flag-bits in the variable Flags (see
    ; the bit definitions under the Window structure definition)
    APTR        nw_FirstGadget

    ; the CheckMark is a pointer to the imagery that will be used when
    ; rendering Menultems of this Window that want to be checkmarked
    ; if this is equal to NULL, you'11 get the default imagery
    APTR nw_CheckMark

    APTR nw_Title               ; title text for the Window

    ; the Screen pointer is used only if you've defined a CUSTOMSCREEN and
```

```
    ; want this Window to open in it.  If so, you pass the address of the
    ; Custom Screen structure in this variable.    Otherwise, this variable
    ; is ignored and doesn't have to be initialized.
    APTR nw_Screen

    "> SUPER BITMAP Window?  If so, put the address of your BitMap structure
    ; in this variable.    If not, this variable is ignored and doesn't have
    ; to be initialized
    APTR nw_BitMap

    ; the values describe the minimum and maximum sizes of your UiïdöWS.
    ; these matter only if you[1]ve chosen the WINDOWSIZING Gadget option,
    ; which means that you want to let the User to change the size of
    ; this Window.  You describe the minimum and maximum sizes that the
    ; Window can grow by setting these variables.  You can initialize
    ; any one these to zero, which will mean that you want to duplicate
    ; the setting for that dimension (if MinWidth == 0, MinWidth will be
    ; set to the opening Width of the Window).
    •> You can change these settings later using SetWindowLimits().
    ; If you haven't asked for a SIZING Gadget, you don't have to
    ; initialize any of these variables.
    WORD nw_MinWidth
    WORD nw_MinHeight
    WORD nw_MaxWidth
    WORD nw_MaxHeight

    i the type variable describes the Screen in which you want this Window to
    ; open.  The type value can either be CUSTOMSCREEN or one of the
    ; system standard Screen Types such as WBENCHSCREEN.  See the
    ; type definitions under the Screen structure
    WORD nw_Type

    LABEL nw_SIZE


   IFND INTUITION_SCREENS_I
   INCLUDE "intuition/screens.i"
   ENDC

   IFND INTUITION_PREFERENCES_I
   INCLUDE "intuition/preferences,i"
   ENDC

; ============================================================================
; === Remember ===============================================================
; ============================================================================
; this structure is used for remembering what memory has been allocated to
; date by a given routine, so that a premature abort or systematic exit
; can deallocate memory cleanly, easily, and completely
 STRUCTURE Remember,0

    APTR rm_NextRemember
    LONG rm_RememberSize
    APTR rm_Memory

 LABEL     r-«_SIZEOF



; ============================================================================
; === Miscellaneous ==========================================================
```

```
; ==========================================================

; = MACROS ================================================
;#define MENUNUM(n) (n & OxlF)
;#define ITEMNUM(n) (Cn >> 5) & 0x003F)
•,#define SUBNUM(n) ((n >> 11) & OxOOlF)
;
;#define SHIFTMENU(n) (n & OxlF)
;#define SHIFTITEM(n) ((n & 0x3F) << 5)
;#define SHIFTSUB(n) ((n & OxlF) << 11)

|#define SRBNUM(n)  (0x08 - (n >> 4))  /* SerRWBits -> read bits per char */
;#define SWBNUM(n)  (0x08 - (n & OxOF))/* SerRWBits -> write bits per chr */
•,#define SSBNUM(n)  (0x01 + (n >> 4))  /* SerStopBuf -> stop bits per chr */
;#define SPARNUM(n) (n >> 4)         /* SerParShk -> parity setting    */
;#define SHAKNUM(n) (n & OxOF)         /* SerParShk -> handshake mode    */
;
j = MENÜ STUFF ==========================================================
NOMENU EQU   $001F
NOITEM EQU   $003F
NOSUB  EQU   $001F
MENUNULL EQU   $FFFF


; = =RJ='s peculiarities ================================================
;#define FOREVER for(jj)
;#define SIGN(x) ( (Cx) > 0) - ((x) < 0) )


; these defines are for the COMMSEQ and CHECKIT menü stuff.  If CHECKIT,
; I1 11 use a generic Width (for all resolutions) for the CheckMark.
; If COMMSEQ, likewise I1 11 use this generic stuff
CHECKWIDTH   EQU   19
COMMWIDTH   EQU   27
LOWCHECKWIDTH   EQU   13
LOWCOMMWIDTH   EQU   16


; these are the AlertNuraber defines.  if you are calling DisplayAlert()
; the AlertNumber you supply must have the ALERT_TYPE bits set to one
; of these patterns
ALERT_TYPE   EQU   $80000000
RECOVERY_ALERT   EQU   $00000000   ; the system can recover from this
DEADEND_ALERT   EQU   $80000000   ; no recovery possible, this is it


; When you1re defining IntuiText for the Positive and Negative Gadgets
; created by a call to AutoRequest(), these defines will get you
\ reasonable-looking text,  The only field without a define is the IText
; field; you decidé what text goes with the Gadget
AUTOFRONTPEN   EQU   0
AUTOBACKPEN  EQU ' 1
AUTODRAWMODE   EQU   RP_JAM2
AUTOLEFTEDGE   EQU   6
AUTOTOPEDGE   EQU   3
AUTÓITEXTFONT   EQU   0
AUTONEXTTEXT   EQU   0


;* --- RAWMOUSE Codes and Qualifiers (Console OR IDCMP) ----------
```

```
SELECTUP    EQU    (IECODEJ-BUTTQN+IECODE_UP_PREFIX)
SELECTDOWN    EQU    (IECODE_LBUTTON)
MENUUP       EQU    (IECODE_RBUTTON+IECODE_UP_PREFIX)
MENUDOWN    EQU    (IECODE_RBUTTON)
ALTLEFT       EQU    (IEQUALIFIER_LALT)
ALTRIGHT    EQU    (IEQUALIFIER_RALT)
AMIGALEFT    EQU    (IEQUALIFIERJXOMMAND)
AMIGARIGHT    EQU    (IEQUALIFIER_RCOMMAND)
AMIGAKEY5    EQU    (AMIGALEFT+AMIGARIGHT)

CURSORUP    EQU    $4C
CUR50RLEFT    EQU    S4F
CUR50RRIGHT    EQU    $4E
CURSORDOWN    EQU    $4D
KEYCODE_Q    EQU    $10
KEYCODE_X    EQU    S32
KEYCODE_N    EQU    $36
KEYCODE_M    EQU    $37
KEYCODE_V    EQU    $34
KEYCODE_B    EQU    $35

    IFND    INTUITION_INTUITIONBASE_I
    include "intuition/intuitionbase.i"
    ENDC

    ENDC    ;  INTUITION_INTUITION_I
```

```
        IFND    INTUITIONJNTUITIONBASEJ
INTUITION_INTUITIONBASE_I       SET     1
*r
*x      SFilenarae:  intuition/intuitionbase.i $
**      SRelease:  1.3 $
XX
**   <   the IntuitionBase structure and supporting structures
XXI
**      (C) Copyright 1985,1986,1987,1988 Commodore-Araiga,  Inc.
*x          Ali Rights Reserved
XX

        IFND EXEC_TYPES_I
        INCLUDE "exec/types.i"
        ENDC

        IFND EXEC_LIBRARIES_I
        INCLUDE "exec/libraries.i"
        ENDC

        IFND    GRAPHICS_VIEW_I
        INCLUDE "graphics/view.i"
        ENDC

* Be sure to protect yourself against someone modifying these data as
* you look at them.  This is done by calling:
x
* lock = LocklBase(O), which returns a ULONG.  When done call
*  DO            DO
* UnlocklBase(lock) where lock is what LockIBaseC) returned.
x            AO
* NOTE: these library functions are simply stubs now, but should be called
* to be compatible with future releases.                          \

* =================================================================== *
* === IntuitionBase ================================================== x
K  ================================================================== 3(
 STRUCTURE IntuitionBase,0

        STRUCT  ib LibNode,LIB_SIZE
        STRUCT  ib!viewLord,v_SIZEOF
        APTR    ib_ActiveWindow
        APTR    ib_ActiveScreen

* the FirstScreen variable points to the frontmost Screen.  Screens are
* then maintained in a front to back order using Screen.NextScreen

        APTR    ib_FirstScreen

* there is not size here because...
x
*

        ENDC    ;  INTUITION_INTUITIONBASE_I
```

```
WORD pf_colorl9       ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
WORD pf_PointerTicks     ; Sensitivity of the pointer

 ; Workbench Screen colors
WORD  pf_^CO10rU       >*3R*aRRR*m**mJ!mmXMMX*»»RRmm*KXmm
WORD pf_colorl       ;  Standard default colours
WORD pf_color2       ;   Used in the Workbench
WORD pf_color3       1XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

 ; positioning data for the Intuition View
BYTE pf_ViewXOffset      > Offset for top lefthand corner
BYTE pf_ViewYOffset      > X and Y diraensions
WORD pf_ViewInitX        ; View initial offsets at startup
WORD pf_ViewInitY        > View initial offsets at startup

BOOL EnableCLI       ; CLI availability switch

 '•> printer configurations
WORD pf_Pr,interType      ; printer type
STRUCT pf_PrinterFilename>FILENAME_SIZE ; file for printer

 ; print formát and quality configurations
WORD pf_.PrintPitch       ; print pitch
WORD pf_„PrintQuality     í print quality
WORD pf*_.PrintSpacing    í number of lines per inch
WORD pf'_.PrintLeftMargin '; left margin in characters
WORD pf_.PrintRightMargin     ; right margin in characters
WORD pf_Pr  int  Image    ; positive or negative
WORD pf_.PrintAspect      ; horizontal or verticai
WORD pf_.PrintShade       ; b&w> half-tone» or color
WORD pf,_PrintThreshold   ; darkness ctrl for b/w dumps


 ; print paper description
WORD pf_PaperSize        » paper size
WORD pf_PaperLength      \ paper length in lines
WORD pf_PaperType        ; continuous or single sheet

 ; Serial device settings: These are six nibble-fields in three bytes
; (these look a little strange SQ the defaults will map out to zero)
BYTE pf_SerRWBits             > upper nibble = (8-number of read bits)
                              ; lower nibble = (8-number of write bits)
BYTE pf_SerStopBuf            ; upper nibble = (number of stop bits - 1)
                              ; lower nibble = (table value for BufSize)
BYTE pf_5erParShk             j upper nibble = (value for Parity setting)
                              ; lower nibble = (value for Handshake mode)

BYTE pfJLaceWB       ; if workbench is to be interlaced

STRUCT pf_WorkName,FILENAME_SIZE ; temp file for printer

BYTE pf_RowSizeChdnge         ;
BYTE pf_ColuranSizeChange        ;

UWORD   pf_PrintFlags         ; user preference flags
WORD    pf_PrintMaxWidth     -; max width of printed picture in lOths/inch
UWORD   pf_PrintMaxHeight     ; max height of printed picture in lOths/inch
UBYTE   pf_PrintDensity       ; print density
UBYTE   pf_PrintXOffset       5 offset of printed picture in lOths/inch

UWORD   pf_wb_Width           "> override default workbench width
```

```
    UWORD   pf_wb_Height        > override default workbench height
    UBYTE   pf_wb_Depth         > override default workbench depth

    UBYTE   pf_ext_size         » extension information -- do not touch!
                                ; extension size in blocks of 64 bytes
    LABEL pf_SIZEOF


i == Preferences definitions ==========================================

; Workbench Interlace (use one bit)
LACEWB      EQU SOI

*, PrinterPort
PARALLEL_PRINTER EQU    $00
SERIAL_PRINTER  EQU $01

", BaudRate
BAUD_110    EQU $00
BAUD_300    EQU $01
BAUD_1200   EQU $02
BAUD_2400   EQU $03
BAUD_4800   EQU $04
BAUD_9600   EQU $05
BAUD_19200  EQU $06
BAUD_MIDI   EQU $07

\, PaperType
FANFOLD     EQU $00
SIKGLE      EQU $80

; PrintPitch
PICA        EQU $000
ELITÉ       EQU $400
FINE        EQU $800

l PrintQuality
DRAFT       EQU $000
LETTER      EQU $100

; PrintSpacing
SIX_LPI     EQU $000
EIGHT_LPI   EQU $200

; Print Image
1MAGE_POSITIVE  EQU $00
IMAGE_NEGATIVE  EQU $01

; PrintAspect
ASPECT_HORIZ    EQU $00
ASPECT_VERT EQU $01

i PrintShade
SHADE_BW    EQU $00
SHADE_GREYSCALE EQU $01
SHADE_COLOR EQU $02

; PaperSize
USJJETTER   EQU $00
US_LEGAL    EQU $10
N_TRACTOR   EQU $20
```

```
W_TRACTOR    EQU $30
CUSTOM       EQU $40

; PrinterType
CUSTOM_NAME EQU $00
ALPHA_P_101 EQU $01
BR0THER_15XL    EQU $02
CBM_MP51000 EQU $03
DIAB_630    EQU $04
DIAB_ADV_D25    EQU $05
DIAB_C_150  EQU $06
EPSON       EQU $07
EP50N_JX_80 EQU $08
OKIMATE_20  EQU $09
QUME_LP_20  EQU $0A
j new printer entries, 3 October 1985
HP_LASERJET EQU $0B
HP_LASERJET_PLUS EQU    $0C


; Serial Input Buffer Sizes
SBUF_512        EQU     $00
SBUF_1024       EQU     $01
SBUF_2048       EQU     $02
SBUF_4096       EQU     $03
SBUF_8000       EQU     $04
SBUF_16000      EQU     $05

; Serial Bit Masks
SREAD_BITS  EQU $F0 ; pf_SerRWBits
SWRITE_BITS EQU $0F

SSTOP_BITS  EQU $F0 ; pf_SerStopBuf
SBUFSIZE_BITS   EQU $0F

SPARITY_BITS    EQU $F0 ; pf_SerParShk
SHSHAKE_BITS    EQU $0F

', Serial Parity (high nibble, but here sh'ifted right> as by C-macro SPARNUM)
SPARITY_NONE    EQU     $00
SPARITY_EVEN    EQU     $01
SPARITY_ODD     EQU     $02

; Serial Handshake Mode (low nibble, mask by SHSHAKE_BITS)
SHSHAKE_XON     EQU     $00
SHSHAKE_RTS     EQU     $01
SHSHAKE_NONE    EQU     $02

; new defines for PrintFlags
CORRECT_RED         EQU $0001    > color correct red shades
CORRECT_GREEN       EQU $0002    ; color correct green shades
CORRECT_BLUE        EQU $0004    5 color correct blue shades

CENTER_IMAGE        EQU $0008      > center image on paper

IGNORE_DIMENSIONS   EQU $0000    ; ignore max width/height settings
BOUNDED_DIMENSIONS  EQU $0010    ; use max width/height as boundaries
ABSOLUTE_DIMENSIONS EQU $0020    ; use max width/height as absolutes
PIXEL_DIMENSIONS    EQU $0040    ; use max width/height as prt pixels
MULTIPLY_DIMENSIONS EQU $0080    ; use max width/height as multipliers
```

```
        IFND    INTUITION_SCREENS_I
INTUITION_SCREENS_I     SET     1
x*
**      ÍFilename: intuition/screens.i $
**      SRelease: 1.3 $
xx
XX

*x '    (C) Copyright 1987>1988 Commodore-Amiga, Inc.
*x          Ali Rights Reserved
xx

        IFND EXEC_TYPES_I
        INCLUDE "éxec/types.i"
        ENDC

        IFND GRAPHICS_GFX_I
        INCLUDE "graphics/gfx.i"
        ENDC

        IFND GRAPHICS_CLIP_I
        INCLUDE "graphi cs/c1i p.i"
        ENDC

        IFND GRAPHICS_VIEW_I
        INCLUDE "graphics/view.i"
        ENDC

        IFND GRAPHICS_RASTPORT_I
        INCLUDE "graphics/rastport.i"
        ENDC

        IFND GRAPHICS_LAYERS_I
        INCLUDE "graphics/layers.i"
        ENDC

; ============================================================================
; === Screen =================================================================
; ============================================================================
 STRUCTURE Screen,Q

    APTR sc_NextScreen          J linked list of sereens
    APTR sc_FirstWindow         ; linked list Sereen's Windows  .

    WORD sc_LeftEdge            ; pararaeters of the sereen
    WORD sc_TopEdge             ; parameters of the sereen

    WORD sc_Width               ; null-terminated Title text
    WORD sc_Height              ', for Windows without ScreenTitle

    WORD sc_MouseY              ; position relative to upper-left
    WORD sc_MouseX              ; position relative to upper-left

    WORD sc_Flags               ; see definitions below

    APTR sc_Title
    APTR se_DefaultTitle

    ; Bar sizes for this Sereen and all Window's in this Sereen
    BYTE sc_BarHeight
    BYTE sc_BarVBorder
    BYTE sc_BarHBorder
    BYTE se_MenuVBorder
```

```
Directory "Lattice_C_5.0.5:Assembler..Headers/devices" on Saturday 29-Sep-90
audio.i                    1144 ----rwed 07-Nov-88 14:58:29
bootblock.i                 727 ----rwed 07-Nov-88 14:58:30
clipboard.i                1728 ----rwed 07-Nov-88 14:58:30
console.i                  1857 ----rwed 07-Nov-88 14:58:29
conunit.i                  2523 ----rwed 07-Nov-88 14:58:27
gameport.i                 1157 ----rwed 07-Nov-88 14:58:29
hardblocks.i               8571 ----rwed 07-Nov-88 14:58:30
input.i                     518 ----rwed 07-Nov-88 14:58:30
inputevent.i               4492 ----rwed 07-Nov-88 14:58:30
keyboard.i                  481 ----rwed 07-Nov-88 14:58:29
keyraap.i                  1635 ----rwed 07-Nov-88 14:58:28
narrátor.i                 2727 ----rwed 07-Nov-88 14:58:28
parallel.i                 3179 ----rwed 07-Nov-88 14:58:30
printer.i                  8158 ----rwed 07-Nov-88 14:58:28
prtbase.i                  6152 ----rwed 07-Nov-88 14:58:30
prtgfx.i                   2512 ----rwed 07-Nov-88 14:58:31
scsidisk.i                 3371 ----rwed 07-Nov-88 14:58:30
serial.i                   5358 ----rwed 07-Nov-88 14:58:29
timer.i                     649 ----rwed 07-Nov-88 14:58:28
trackdisk.i                5256 ----rwed 07-Nov-88 14:58:30
20 files - 157 blocks - 62195 bytes
```

```
        IFND    DEVICES_AUDIO_I
DEVICES_AUDIO_I SET     1
**
*X      SFilename: devices/audio.i $
        SRelease: 1.3 $
XX
XX
XX
rx      (C) Copyright 1985,1986,1987,1988 Commodore-Araiga, Inc.
XX          Ali Rights Reserved
XX

                        IFND    EXEC_IO_I
                        INCLUDE "exec/io.i"
                        ENDC

AUDIONAME               MACRO
                        DC.B    'audio.device',0
                        ENDM

ADHARD_CHANNELS         EQU     4

ADALLOC_MINPREC         EQU     -128
ADALLOC_MAXPREC         EQU     127

ADCMD_FREE              EQU     CMD_NONSTD+0»
ADCMD_SETPREC           EQU     CMD_NONSTD+1
ADCMD_FINISH            EQU     CMD~N0NSTD+2
ADCMD_PERVOL            EQU     CMD~N0NSTD+3
ADCMD_LOCK              EQU     CMD_NONSTD+4
ADCMD_WAITCYCLE         EQU     CMDJÍ0NSTD+5
ADCMDB_NOUNIT           EQU     5
ADCMDF_NOUNIT           EQU     1<<5
ADCMD_ALLOCATE          EQU     ADCMDF_NOUNIT+0

ADIOB_PERVOL            EQU     4
ADIOF_PERVOL            EQU     1<<4
ADIOB_SYNCCYCLE         EQU     5
ADIOF_SYNCCYCLE         EQU     1<<5
ADIOB_NOWAIT            EQU     6
ADIOF_NOWAIT            EQU     1<<6
ADIOB_WRITEMESSAGE      EQU     7
ADIOF_WRITEMESSAGE      EQU     1<<7-

ADIOERR_NOALLOCATION    EQU     -10
ADIOERR_ALLOCFAILED     EQU     -11
ADIOERR_CHANNELSTOLEN   EQU     -12

                        STRUCTURE       IOAudio,IO_SIZE
                        WORD    ioa_AllocKey
                        APTR    ioa_Data
                        ULONG   ioa_Length
                        UWORD   ioa_Period
                        UWORD   ioa_Volume
                        UWORD   ioa_Cycles
                        STRUCT  ioa_WriteMsg,MN_SIZE
                        LABEL   ioa_SIZEOF

        ENDC    ; DEVICES_AUDIO_I
```

```
        IFND    DEVICES_BOOTBLOCK_I
DEVICES_BOOTBLOCK_I    "SET     1
*X
**      SFilename: devices/bootblock.i $
*X      SRelease: 1.3 $
XX
XX      BootBlock definition:
X*
i*      (C) Copyright 1985,1986,1987?1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

 STRUCTURE BB,0
        STRUCT  BB_ID,4         * 4 character identifier
        LONG    BB_CHKSUM       * boot block checksum (balance)
        LONG    BB DOSBLOCK     * reserved for DOS patch
        LABEL   BBIENTRY        * bootstrap entry point
        LABEL   BB_SIZE

BOOTSECTS       equ     2       * 1K bootstrap

BBID_DOS        macro           * something that is bootable
                dc.b    'DOS',0
                endm

BBID_KICK       macro           * firmware image disk
                dc.b    'KICK'
                endm


BBNAME_DOS      EQU     (('D'<<24)!('O'<<16)!('S'<<8))
BBNAME_KICK     EQU     (CK[1]<<24)!Cr<<16)!([1]C[1] <<8)!('K'))

        ENDC    5 DEVICES_BOOTBLOCK_I
```

```
        IFND    DEVICES_CLIPBOARD_I
DEVICES_CLIPBOARD_I     SET     1
XX
XX      SFilename: devices/clipboard.i $
XX      $Release: 1.3 $
XX
XX      clipboard device coramand definitions
XX
XX      (C) Copyright 1985,1986>1987,1988 Commodore-Araiga, Inc.
XX          Ali Rights Reserved
XX

   IFND     EXEC_NODES_I
   INCLUDE     "exec/nodes.i"
   ENDC
   IFND     EXEC_LI5T5_I
   INCLUDE     "exec/lists.i"
   ENDC
   IFND     EXEC_PORTS_I
   INCLUDE     "exec/ports.i"
   ENDC
   IFND     EXEC_IO_I
   INCLUDE     "exec/io.i"
   ENDC

   DEVINIT

   DEVCMD   CBD_POST
   DEVCMD   CBD_CURRENTREADID
   DEVCMD   CBD_CURRENTWRITEID

CBERR_OBSOLETEID  EQU   1


 STRUCTURE  ClipboardUnitPartial,0
    STRUCT  cu_Node,LN_SIZE;     ; list of units
    ULONG   cu_UnitNum>       »' unit number for this unit
    ; the remaining unit data is priváté to the device


 STRUCTURE  I0ClipReq,0
    STRUCT  ÍQ_Message,MN_SIZE
    APTR    io_Device         ; device node pointer
    APTR    ioJJnit           ; unit (driver priváté)
    UWORD   io_Command        ; device command
    UBYTE   io_Flags          ; including QUICK and SATISFY
    BYTE    io_Error          ; error or warning num
    ULONG   io_Actual         ; number of bytes transferred
    ULONG   io_Length         ; number of bytes requested
    APTR    io_Data           ; either clip stream or post port
    ULONG   io_Offset         ; offset in clip stream
    LONG    io_ClipID         ; ordinal clip identifier
    LABEL   iocr_SIZEOF



PRIMARY_CLIP   EQU   0        ; primary clip unit

 STRUCTURE  SatisfyMsg,0
    STRUCT  sm_Msg,MN_SIZE    ; the length will be 6
    UWORD   sm_Unit           ; which clip unit this is
    LONG    sm_ClipID         ; the clip identifier of the post
    LABEL   satisfyMsg_SIZEOF
```

```
ENDC    ; DEVICES_CLIPBOARD_I
```

```
        IFND    DEVICES_CONSOLE_I
DEVICES_CONSOLE_I       SET     1

XX      SFilename: devices/console.i $
XX      SRelease: 1.3 $
XX
IX      Console device command definitions
SX
XX      (C) Copyright 1985jl986,1987>1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
iX

   IFND  EXECJO_I
   INCLUDE  "exec/io.i"
   ENDC

******* Console commands *******
   DEVINIT

   DEVCMD         CD_ASKKEYMAP
   DEVCMD         CD_SETKEYMAP
   DEVCMD         CD_ASKDEFAULTKEYMAP
   DEVCMD         CD_SETDEFAULTKEYMAP


******* SGR paraméters

SGR_PRIMARY       EQU   0
SGR~BOLD          EQU   1
SGR_ITALIC        EQU   3
SGR_UNDERSCORE    EQU   4
SGR_NEGATIVE      EQU   7

*  these names refer to the ANSI standard, not the implementation
5GR_BLACK         EQU   30
SGR_RED           EQU   31
SGR_GREEN         EQU   32
SGR_YELLOW        EQU   33
SGR_BLUE          EQU   34
SGR_MAGENTA       EQU   35
SGR_CYAN          EQU   36
SGR~WHITE         EQU   37
SGR_DEFAULT       EQU   39

SGR_BLACKBG       EQU   40
SGR_REDBG         EQU   41
SGR_GREENBG       EQU   42
SGR_YELLOWBG      EQU   43
SGR_BLUEBG        EQU   44
SGR_MAGENTABG     EQU   45
SGR_CYANBG        EQU   46
SGR_WHITEBG       EQU   47
SGR_DEFAULTBG     EQU   49

*  these names refer to the implementationj they are the preferred
*  names for use with the Amiga console device.
SGR CLRO          EQU   30
SGíIcLRl          EQU   31
SGR_CLR2          EQU   32
SGR_CLR3          EQU   33
SGR_CLR4          EQU   34
SGR_CLR5          EQU   35
SGR_CLR6          EQU   36
```

```
SGR_CLR7          EQU   37

SGR_CLR0BG        EQU   40
SGRICLRIBG        EQU   41
SGR_CLR2BG        EQU   42
SGR_CLR3BG        EQU   43
SGR_CLR4BG        EQU   44
SGR_CLR5BG        EQU   45
SGR_CLR6BG        EQU   46
SGR_CLR7BG        EQU   47


******  DSR parameters

DSR_CPR           EQU   6

******  CTC parameters
CTC_HSETTAB       EQU   0
CTC_HCLRTAB       EQU   2
CTCJÍCLRTABSALL   EQU   5

******   TBC parameters
TBC_HCLRTAB       EQU   0
TBCJCLRTABSALL    EQU   3

******  SM and RM parameters
M_LNM             EQU   20    > linefeed newline mode
M_ASM    MACRO
      DC.B  '>!'              > autó scroll mode
      ENDM
M_AWM    MACRO
      DC.B  '?7'              i autó wrap mode
      ENDM

        ENDC    ', DEVICES_CONSOLE_I
```

```
        IFND    DEVICES_CONUNIT_I
DEVICES_CONUNIT_I       5ET     1
XX
XX      $Filename: devices/conunit.i $
XX      SRelease: 1.3 $
XX
XX      Console device unit definitions
XX
XX      (C) Copyright 1986*1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

   IFND  EXEC_PORTS_I
   INCLUDE  "exec/ports.i"
   ENDC

   IFND  DEVICES_CONSOLE_I
   INCLUDE  "devices/console.i"
   ENDC

   IFND  DEVICES_KEYMAF_I
   INCLUDE  "devices/keyraap.i"
   ENDC

   IFND  DEVICES_INPUTEVENT_I
   INCLUDE  "devices/inputevent.i"
   ENDC

PMB_ASM     EQU   MJ_NM+1     ; internál storage bit for AS flag
PMB_AWM     EQU   PMB_ASM+1   ; internál storage bit for AW flag
MAXTABS     EQU   80


 STRUCTURE  ConUnit,MP_SIZE
    ;———————read only variables
   APTR  cu_Window            > intuition window bound to this unit
   WORD  cu_XCP               ; character position
   WORD  cu_YCP
   WORD  cu_XMax              i max character position
   WORD  cu_YMax
   WORD  cu_XRSize           ', character raster size
   WORD  cu_YRSize
   WORD  cu_XR0rigin          ; raster origin
   WORD  cu_YR0rigin
   WORD  cu_XRExtant          ; raster maxima
   WORD  cu_YRExtant
   WORD  cu_XMinShrink        ; smallest area intact from resize process
   WORD  cu_YMinShrink
   WORD  cu_XCCP              ; cursor position
   WORD  cu_YCCP

   J———————read/write variables (writes must must be protected)
   j———————storage for AskKeyMap and SetKeyMap
   STRUCT   cu_KeyMapStruct,km_SIZEOF
   ;——————tab stops
   STRUCT   cu_TabStops,2*MAXTABS   ; 0 at start, Oxffff at end of list

   ;———————console rastport attributes
   BYTE  cu_Mask              ; these must appear as in RastPort
   BYTE  cu_FgPen             ;    {
   BYTE  cu_BgPen             ;    !
   BYTE  cu_A0LPen            ;    +
   BYTE  cu_DrawMode          ; these must appear as in RastPort
   BYTE  f>u_AroaPt£3          1    +
```

```
    APTR  cu_AreaPtrn          ; cursor area pattern
    STRUCT   cu_Minterms,8     ; console minterms
    APTR  cu_Font              ;
    UBYTE cu_AlgoStyle         ; these must appear as in RastPort
    UBYTE cu_TxFlags           ;   +
    UWORD cu_TxHeight          ; these must appear as in RastPort
    UWORD cu_TxWidth           ;   !
    UWORD cu_TxBaseline        ;   !
    UWORD cu_TxSpacing         ;   +

    ;-------console MODES and RAW EVENTS switches
    STRUCT   cu_Modes><(PMB_AWM+7)/8>    ; one bit per mode
    STRUCT  cu_RawEvents,<(IECLASS_MAX+7)/8 >

    ;-------ensure the ConsUnit structure is evén
ODDEVEN  EQU    ((PMB_AWM+7)/8)+((IECLASS_MAX+7)/8)
    IFNE ODDEVEN- UODDE VEN/2 )*2)
       UBYTE cu_pad
    ENDC

    LABEL ConUnit_SIZEOF

         ENDC    ; DEVICES_CONUNIT_I
```

```
        IFND    DEVICES_GAMEPORT_I
DEVICES_GAMEPORT_I      SET     1
xx
xx      SFilename: devices/gameport.i $
xx      SRelease: 1.3 $
xx
»x      Game Port device command definitions
xx
*x      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
xx          Ali Rights Reserved
xx

   IFND  EXEC_IO_I
   INCLUDE  "exec/io.i"
   ENDC


кxxxxxx GamePort commands xxxxxxx
   DEVINIT

   DEVCMD   GPD_READEVENT
   DEVCMD   GPD_ASKCTYPE
   DEVCMD   GPD_SETCTYPE
   DEVCMD   GPD_ASKTRIGGER
   DEVCMD   GPD_SETTRIGGER

xxxxxxx GamePort structures'xxxxxx*

*  gpt_Keys
   BITDEF       GPT,D0WNKEYS,0
   BITDEF       GPT,UPKEYS,1

 STRUCTURE  GamePortTrigger,0
       UWORD   gpt_Keys         ;key transition triggers
       UWORD   gpt_Timeout      ;time trigger (verticai blank units)
       UWORD   gpt_XDelta       jX distance trigger
       UWORD   gpt_YDelta       ;Y distance trigger
       LABEL   gpCsiZEOF

xxxxxxx Controller Types xxxxxx
GPCT_ALLOCATED    EQU   -1     ; allocated by another user
GPCTJÍOCONTROLLER EQU   0

GPCT_MOUSE        EQU   1
GPCT_RELJOYSTICK  EQU   2
GPCT_ABSJOYSTICK  EQU   3


xxxxxxx Errors xxxxxx
GPDERR_SETCTYPE   EQU   1      ; this controller not valid at this time

        ENDC    ; DEVICES_GAMEPORT_I
```

```
        IFND    DEVICES_HARDBLOCKSJ
DEVICES_HARDBLOCKS_I    SET     1
*x
**      SFilename: devices/hardblocks.i $
xx      SRevision: 1.0$
xx      SDate: 88/07/11 15:32:58 $
xx
xx      Fi le System identifier blocks for hard disks
xx
xx      (C) Copyright 1988 Commodore-Amiga) Inc.
xx          Ali Rights Reserved
xx

;-----------------------------------------------------------------------
;
;       This fi le describes blocks of data that exist on a hard disk
;       to describe that disk.  They are not generically accessable to
;       the user as they do not appear on any DOS drive.  The blocks
;       are tagged with a unique identifier) checksummed) and linked
;       together.  The root of these blocks is the RigidDiskBlock.
;
;       The RigidDiskBlock must exist on the disk within the first
;       RDB_LOCATION_LIMIT blocks.  This inhibits the use of the zero
;       cylinder in an AmigaDOS partition: although it is strictly
;       possible to store the RigidDiskBlock data in the reserved
;       area of a partition) this practice is discouraged since the
;       reserved blocks of a partition are overwritten by "Formát")
;       "Install") "DiskCopy") etc.  The recommended disk layout,
;       then> is to use tfie first cylinder(s) to store all the drive
;       data specified by these blocks: i.e. partition descriptions,
;       file system load imageS) drive bad block mapS) spare blocks,
;       etc.
;
;       Though only 512 byte blocks are currently supported by the
;       file system) this proposal tries to be forward-looking by
;       making the block size explicit) and by using only the first
;       256 bytes for all blocks but the LoadSeg data.
;
;-----------------------------------------------------------------------

;
;   NOTE
;       optional block addresses below contain Sffffffff to indicate
;       a NULL address
;
 STRUCTURE      RigidDiskBlock)0
    ULONG   rdbJD               ; 4 character identifier
    ULONG   rdb_SummedLongs     ; size of this checksummed structure
    LONG    rdb_ChkSum          ; block checksum (longword sum to zero)
    ULONG   rdbJtostID          ; SCSI Target ID of hőst
    ULONG   rdb_BlockBytes      ; size of disk blocks
    ULONG   rdb_Flags           ; see below for defines
    ; block list heads
    ULONG   rdb_BadBlockList    ; optional bad block list
    ULONG   rdb_PartitionList   ; optional first partition block
    ULONG   rdb_FileSysHeaderList ; optional füle system header block
    ULONG   rdb_Drivelnit       ; optional drive-specific init code
                                ; Drivelnit(lun,rdb,ior): "C" stk & dO/aO/al
    STRUCT  rdb_Reservedl,6*4   ; set to Sffffffff
    ") physical drive characteristics
    ULONG   rdb_Cylinders       ; number of drive cylinders
    ULONG   rdb_Sectors         ; sectors per track
    ULONG   rdb_Heads           ; number of drive heads
    ULONG   rdb_Interleave      ; interleave
```

```
    ULONG   rdbJPark            ; landing zone cylinder
    STRUCT  rdb_Reserved2,3*4
    ULONG   rdb_writePreComp    ; starting cylinder: write precompensation
    ULONG   rdb_ReducedWrite    ; starting cylinder: reduced write current
    ULONG   rdb_StepRate        ; drive step rate
    STRUCT  rdbjteserved3,5*4
    ; logical drive characteristics
    ULONG   rdb_RDBBlocksLo     ; low block of rangé reserved for hardblocks
    ULONG   rdb_RDBBlocksHi     ; high block of rangé for these hardblocks
    ULONG   rdbJ_oCylinder      ; low cylinder of partitionable disk area
    ULONG   rdbJüCylinder       ; high cylinder of partitionable data area
    ULONG   rdb_CylBlocks       ; number of blocks available per cylinder
    ULONG   rdb_AutoParkSeconds ; zero for no autó park
    STRUCT  rdb_Reserved4,2*4
    J drive identification
    STRUCT  rdb_DiskVendor,8
    STRUCT rdb_Di skProduct,16
    STRUCT  rdb_DiskRevision,4
    STRUCT  rdb_ControllerVendor,8
    STRUCT  rdb_ControllerProduct,16
    STRUCT  rdb_ControllerRevision,4

    STRUCT  rdb_Reserved5,10*4

    LABEL   RigidDiskBlock_SI2E0F

IDNAME_RIGIDDISK     EQU    (('R'<<24)!('D'<<16)!CS'<<8)!('K'))
RDB_LOCATION_LIMIT      EQU     16

    BITDEF  RDBF,LAST,0         ; no disks exist to be configured after
                                ;   this one on this controller
    BITDEF  RDBF,LASTLUN,1      ; no LUNs exist to be configured greater
                                ;   than this one at this SCSI Target ID
    BITDEF  RDBF,LASTTID,2      ; no Target IDs exist to be configured
                                ;   greater than this one on this SCSI bus
    BITDEF  RDBF,N0RESELECT,3   ; don't bother trying to perform reselection
                                ;   when talking to this drive
    BITDEF  RDBF,DISKID,4       ; rdb_Disk... identification valid
    BITDEF  RDBF,CTRLRID,5      ; rdb_Controller... identification valid


;---------------------------------------------------------------------
 STRUCTURE      BadBlockEntry,0
    ULONG   bbe_BadBlock        ; block number of bad block
    ULONG   bbe_GoodBlock       ; block number of replacement block
    LABEL   BadBlockEntry_SIZEOF

 STRUCTURE      BadBlockBlock,0
    ULONG   bbbJD               ; 4 character identifier
    ULONG   bbb_SummedLongs     ; size of this checksumraed structure
    LONG    bbb_ChkSum          ; block checksum (longword sum to zero)
    ULONG   bbbJtostlD          ; SCSI Target ID of hőst
    ULONG   bbb_Next            ; block number of the next BadBlockBlock
    ULONG   bbb_Reserved
    STRUCT  bbb_BlockPairs,61*BadBlockEntry_SIZE0F ; bad block entry pairs
    ; note 61 assumes 512 byte blocks
    ; there is no BadBlockBlock_SIZEOF: try rdb_BlockBytes

IDNAME_BADBLOCK         EQU     (('B'<<24)!('A'<<16)!('D'<<8)!('B'))

;---------------------------------------------------------------------
```

```
 STRUCTURE      PartitionBlockjO
    ULONG   pb_ID               ; 4 character identifier
    ULONG   pb_SummedLongs      ; size of this checksuramed structure
    LONG    pb_ChkSum           ; block checksum (longword sum to zero)
    ULONG   pb_HostID           ; SCSI Target ID of hőst
    ULONG   pb_Next             ; block number of the next PartitionBlock
   •ULONG   pb_Flags            ; see below for defines
    STRUCT  pb_Reserved1,2*4
    ULONG   pb_DevFlags         ; preferred flags For OpenDevice
    STRUCT  pb_Dr ive Name >32  ; preferred DOS device name: BSTR form
                                ; (not used if this name is in use)
    STRUCT  pb_Reserved2,15*4   ; fillér to 32 longwords
    STRUCT  pb_Environment,17*4 ; environment vector for this partition
    STRUCT  pb_EReserved,15*4   ; reserved for future environment vector
    LABEL   PartitionBlock_SIZEOF

IDNAME_PARTITION        EQU     ('P'<<24)!('A'<<16)!('R'<<8)!('T')

    BITDEF  PBF,BOOTABLE,0      ; this partition is intended to be bootable
                                ;   (expected directories and files exist)
    BITDEF  PBF,NOMOUNT,1       ; do not mount this partition (e.g. manually
                                ;   mountedj but space reserved here)

;---------------------------------------------------------------------
 STRUCTURE      FileSysHeaderBlock,0
    ULONG   fhb_ID              ; 4 character identifier
    ULONG   fhb_SummedLongs     ; size of this checksummed structure
    LONG    fhb_ChkSum          ; block checksum (longword sum to zero)
    ULONG   fhb_HostID          ; SCSI Target ID of hőst
    ULONG   fhb_Next            ; block number of the next FileSysHeaderBlock
    ULONG   fhb_Flags           ; see below for defines
    STRUCT  fhb_Reserved1,2*4
    ULONG   fhb_DosType         ; file system description: match this with
                                ; partition environment's DE_DOSTYPE entry
    ULONG   fhb_Version         ; release version of this code
    ULONG   fhb_PatchFlags      ; bits set for those of the following that
                                ;   need to be substituted intő a standard
                                ;   device node for this file system: e.g.
                                ;   $180 to substitute SegList & GlobalVec
    ULONG   fhb_Type            ; device node type: zero
    ULONG   fhb_Task            ; standard dos "task" field: zero
    ULONG   fhb_Lock            ; not used for devices: zero
   'ULONG   fhb_Handler         ; filename to loadseg: zero placeholder
    ULONG   fhb_StackSize       ; stacksize to use when starting task
    LONG    fhb_Priority        ; task priority when starting task
    LONG    fhb_Startup         ; startup msg: zero placeholder
    LONG    fhb_SegListBlocks   ; first of linked list of LoadSegBlocks:
                                ;   note that this entry requires somé
                                ;   processing before substitution
    LONG    fhb_GlobalVec       ; BCPL global vector when starting task
    STRUCT  fhb_Reserved2,23*4  ; (those reserved by PatchFlags)
    STRUCT  fhb_Reserved3,21*4
    LABEL   FileSysHeader_SI2E0F

IDNAME_FILESYSHEADER    EQU     ('F'<<24)!('S'<<16)!('H'<<8)!('D')

;---------------------------------------------------------------------
 STRUCTURE      LoadSegBlockjO
    ULONG   lsb_ID              ; 4 character identifier
    ULONG   lsb_SummedLongs     ; size of this checksummed structure
    LONG    lsb_ChkSum          ; block checksum (longword sum to zero)
```

```
ULONG   lsbJtostID          5 SCSI Target ID of hőst
ULONG   lsb_Next            ; block number of the next FileSysBlock
STRUCT  lsb_LoadData,123*4  \ data for "loadseg"
; note 123 assumes 512 byte blocks
; there is no LoadSegBlock_SIZEOF: try rdb_BlockBytes

IDNAME_LOADSEG          EQU     (('L'<<24)!('S'<<16)!('E'<<8)!('G'))

        ENDC
```

```
        IFND    DEVICESJNPUTJ
DEVICES_INPUT_I SET     1
**
**      ÍFilename: devices/input.i $
**      SRelease: 1.3 $
**
**      input device command definitions
XX
*       (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
        Ali Rights Reserved
XX
**

   IFND     EXEC_IO_I
   INCLUDE     "exec/io.i"
   ENDC

   DEVINIT

   DEVCMD       IND_ADDHANDLER
   DEVCMD       IND_REMHANDLER
   DEVCMD       IND_TOITEEVENT
   DEVCMD       IND_SETTHRESH
   DEVCMD       IND_SETPERIOD
   DEVCMD       IND_SETMPORT
   DEVCMD       IND_SETMTYPE
   DEVCMD       IND_SETMTRIG

        ENDC    ; DEVICES_INPUT_I
```

```
        IFND    DEVICES_INPUTEVENT_I
DEVICES_INPUTEVENT_I    SET     1
XX
**      $Filename: devices/inputevent.i $
**      SRelease: 1.3 S
XX
XX      input event definitions
XX
XX      (C) Copyright 1985>1986,1987>1988 Commodore-Amiga> Inc.
**          Ali Rights Reserved
XX

   IFND  DEVICES_TIMER_I
   INCLUDE  "devices/timer.i"
   ENDC

*------ constants ------------------------------------------------------

*     — InputEvent.ie_Class —
* A NOP input event
IECLASS_NULL            EQU   $00
* A raw keycode froin the keyboard device
IECLASS_RAWKEY          EQU   $01
* A raw mouse report from the garae port device
IECLASS_RAWMOUSE        EQU   $02
* A priváté console event
IECLASS_EVENT           EQU   $03
* A Pointer Position report
1ECLASS_POINTERPOS      EQU   $04
* A tiraer event
IECLASS_TIMER           EQU   $06
* select button pressed down over a Gadget Caddress in ie_EventAddress)
IECLASS_GADGETDOWN      EQU $07
* select button released over the same Gadget (address in ie_EventAddress)
IECLASS_GADGETUP        EQU   $08
* somé Requester activity has taken place.  See Codes REQCLEAR and REQSET
IECLASS_REQUESTER       EQU   $09
* this is a Menü Nuraber transmission (Menü number is in ie_Code)
IECLASS_MENULIST        EQU   $0A
x User has selected the active Window' s Close Gadget
IECLASS_CLOSEWINDOW     EQU   $0B
* this Window has a new size
IECLASS_SIZEWINDOW      EQU   $0C
x the Window pointed to by ie_EventAddress needs to be refreshed
IECLASS_REFRESHWINDOW   EQU   $0D
* new preferences are available
IECLASS_NEWPREFS        EQU   $0E
* the disk has been removed
IECLASS_DISKREMOVED     EQU   $0F
* the disk has been inserted
IECLASS_DISKINSERTED    EQU   $10
* the window is about to be been made active
IECLASS_ACTIVEWINDOW    EQU   $11
* the window is about to be made inactive
IECLASS_INACTIVEWINDOW  EQU   $12

* the last class
IECLASS_MAX             EQU   $12

x————InputEvent.ie_Code ——
x  IECLASS_RAWKEY
IECODE_UP_PREFIX        EQU   $80
IECODEB_UP_PREFIX       EQU   7
```

```
IECODE_KEY_CODE_FIRST   EQU   $00
IECODE_KEY_CODE_LAST    EQU   $77
IECODE_COMM_CODE_FIRST  EQU   $78
IECODE_COMM_CODEJLAST   EQU   $7F

*   IECLASS_ANSI
IECODE_C0_FIRST         EQU   $00
IECODE_CO_LAST          EQU   $1F
IECODE ASCII_FIRST      EQU   $20
IECODE!ASCII_LAST       EQU   $7E
IECODE_ASCII_DEL        EQU   $7F
ÍECODE_Cl_FIRST         EQU   $80
IECODE_C1_LAST          EQU   $9F
IECODE_LATIN1_FIRST     EQU   $A0
1ECODE_LATIN1_LAST      EQU  -$FF

*   IECLASS_RAWMOUSE
IECODE_LBUTTON          EQU   $68  j alsó uses IECODE_UP_PREFIX
1ECODE_RBUTTON          EQU   $69  ;
1ECODE_MBUTTON          EQU   $6A  ;
IECODE_NOBUTTON         EQU   $FF

*   IECLASS_EVENT
IECODE_NEWACTIVE        EQU   $01  > active input window changed

*   IECLASS_REQUESTER Codes
* REQSET is broadcast when the first Requester (not subsequent ones) opens
* in the Window
IECODE_REQSET           EQU   $01
* REQCLEAR is broadcast when the last Requester clears out of the Window
IECODE_REQCLEAR         EQU   $00


*   —   InputEvent.ie_Qualifier   —
IEQUALIFIER_LSHIFT      EQU   $0001
IEQUALIFIERB_LSHIFT     EQU   0
IEQUALIFIER_RSHIFT      EQU   $0002
IEQUALIFIERB_RSHIFT     EQU   1
IEQUALIFIER_CAPSLOCK    EQU   $0004
IEQUALIFIERB_CAPSLOCK   EQU   2
IEQUALIFIER_CONTROL     EQU   $0008
IEQUALIFIERB_CONTROL    EQU   3
IEQUALIFIER_LALT        EQU   $0010
IEQUALIFIERB_LAL'T      EQU   4
IEQUALIFIER_RALT        EQU   $0020
IEQUALIFIERB_RALT       EQU   5
IEQUALIFIER_LCOMMAND    EQU   $0040
IEQUALIFIERB_LCOMMAND   EQU   6
IEQUALIFIER_RCOMMAND    EQU   $0080
IEQUALIFIERB_RCOMMAND   EQU   7
IEQUALIF'IER_NUMERICPAD EQU   $0100
IEQUALIFIERB_NUMERICPAD EQU   8
IEQUALIFIER_REPEAT      EQU   $0200
IEQUALIFIERB_REPEAT     EQU   9
IEQUALIFIER_INTERRUPT   EQU   $0400
IEQUALIFIERB_INTERRUPT  EQU   10
IEQUALIFIER_MULTIBROADCAST    EQU   $0800
IEQUALIFIERB_MULTIBROADCAST   EQU   11
IEQUALIFIER_MIDBUTTON   EQU   $1000
IEQUALIFIERB_MIDBUTTON  EQU   12
IEQUALIFIER_RBUTTON     EQU   $2000
```

```
IEQUALIFIERB_RBUTTON     EQU   13
IEQUALIFIER_LEFTBUTTON   EQU   $4000
IEQUALIFIERB_LEFTBUTTON  EQU   14
IEQUALIFIER_RELATIVEMOUSE      EQU   $8000
IEQUALIFIERB_RELATIVEMOUSE     EQU   15

*------ InputEvent ------------------------------------------------------

 STRUCTURE   InputEvent,0
   APTR  ie_NextEvent          ; the chronologically next event
   UBYTE    ie_Class           ; the input event class
   UBYTE    ie_SubClass        ; optional subclass of the class
   UWORD    ie_Code            ; the input event code
   UWORD   ie_Qualifier "      ; qualifiers in effect for the event
   LABEL  ie_EventAddress      ; a pointer"paraméter for an event
   WORD     ie_X               ; the pointer position for the event
   WORD     ie_Y               ;   usually in canvas relative coords
   STRUCT  ie_TimeStamp,TV_SIZE  ; the system tick at the event
   LABEL    ie_SIZEOF

        ENDC    ; DEVICES_INPUTEVENT_I
```

```
        IFND    DEVICES KEYBOARDJ
DEVICESJCEYBOARDJ     ~SET     1
i*
**      SFilename: devices/keyboard.i S
xx      SRelease: 1.3$
XX
**      Keyboard device coramand definitions
XX
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
XX
   IFND     EXEC_IO_I
   INCLUDE     "exec/io.i"
   ENDC

   DEVINIT

   DEVCMD       KBD_READEVENT
   DEVCMD       KBD_READMATRIX
   DEVCMD       KBD_ADDRESETHANDLER
   DEVCMD       KBD_REMRESETHANDLER
   DEVCMD       KBD_RESETHANDLERDONE

        ENDC    j DEVICES_KEYBOARD_I .
```

```
        IFND    DEVICES_KEYMAP_I
DEVICES_KEYMAP_I        SET     1
XX
XX      SFilenarae: devices/keymap.i $
XX      SRelease: 1.3 $
XX
XX      keymap.resource definitions and console.device key map defiriitions
XX
XX      (C) Copyright 1985>1986>1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

        IFND     EXEC_NODES_I
        INCLUDE "exec/nodes.i"
        ENDC
        IFND     EXEC_LISTS_I
        INCLUDE "exec/lists.i"
        ENDC

 STRUCTURE  KeyMap,0
   APTR  km_LoKeyMapTypes
   APTR  km_LoKeyMap
   APTR  km_LoCapsable
   APTR  km_LoRepeatable
   APTR  km_HiKeyMapTypes
   APTR  km_HiKeyMap
   APTR  km_HiCapsable
   APTR  km HiRepeatable
   LABEL km"^SIZEOF

 STRUCTURE      KeyMapNode,0
    STRUCT  kn_Node,LN_SIZE     ; including name of keymap
    STRUCT  knJCeyMap> km_S IZEOF
    LABEL   kn_SIZEOF

;_______the structure of keymap.resource
 STRUCTURE      KeyMapResource,0
    STRUCT  kr_Node,LN SIZE
    STRUCT  kr_List,LfIsiZE     ; a list of KeyMapNodes
    LABEL   kr_SIZEOF


KCB-NOP     EQU   7
KCF_NOP     EQU   $80

KC_NOQUAL    EQU   0
KC_VANILLA   EQU   7          ; note that SHIFT+ALT+CTRL is VANILLA
KCB_SHIFT    EQU   0
KCF"SHIFT    EQU  ,$01
KCB_ALT      EQU   1
KCF_ALT      EQU   $02
KCB_CONTROL  EQU   2
KCF_CONTROL  EQU   $04
KCB_DOWNUP   EQU   3
KCFJXJWNUP   EQU   $08
KCB_DEAD     EQU   5            ; may be dead or módified by dead key:
KCF_DEAD     EQU   $20          ;   use dead prefix bytes

KCB_STRING   EQU   6
KCF_STRING   EQU   $40

;------ Dead Prefix Bytes
DPB_MOD          EQU     0
```

```
DPF_MOD          EQU     $01
DPB_DEAD         EQU     3
DPF_DEAD         EQU     $08

DP_2DINDEXMASK   EQU     $0F     ; mask for index for lst of two dead keys
DP_2DFACSHIFT    EQU  .  4       ; shift for factor for lst of two dead keys

        ENDC    ; DEVICES_KEYMAP_I
```

```
        IFND    DEVICES_NARRATORJ
DEVICES_NARRATOR_I      ~SET     1
**
**      SFilename: devices/narrátor.i $
**      SRelease: 1.3 $
x*
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

        IFND    EXEC_IO_I
        INCLUDE "exec/io.i"
        ENDC

x_________DEFAULT VALUES, USER PARMS, AND GENERAL CONSTANTS

DEFPITCH  EQU       110             ;DEFAULT PITCH
DEFRATE   EQU       150             ;DEFAULT RATE
DEFVOL    EQU       64              ;DEFAULT VOLUME (FULL)
DEFFREQ   EQU       22200           ;DEFAULT SAMPLING FREQUENCY
NATURALFO EQU       0               ;NATURAL FO CONTOURS
ROBOTICFO EQU       1               ;M0N0T0NE FO
MALE      EQU       0               ;MALE SPEAKER
FEMALE    EQU       1               ;FEMALE SPEAKER
DEFSEX    EQU       MALE            ;DEFAULT SEX
DEFMODE   EQU       NATURALFO       ;DEFAULT MODE

*     Paraméter bounds

MINRATE   EQU       40              •,MINIMUM SPEAKING RATE
MAXRATE   EQU       400             ;MAXIMUM SPEAKING RATE
MINPITCH  EQU       65              5MINIMUM PITCH
MAXPITCH  EQU       320             5MAXIMUM PITCH
MINFREQ   EQU       5000            5MINIMUM SAMPLING FREQUENCY
MAXFREQ   EQU       28000           jMAXIMUM SAMPLING FREQUENCY
MINVOL    EQU       0               ;MINIMUM VOLUME
MAXVOL    EQU       64              >MAXIMUM VOLUME

*     Driver error codes

ND_NotUsed      EQU     -1              ;
ND_NoMem        EQU     -2              ;Can1 allocate memory
ND_NoAudLib     EQU     -3              ;Can't open audio device
ND_MakeBad      EQU     -4              ;Error in MakeLibrary call
NDJJnitErr      EQU     -5              ;Unit other than 0
ND_CantAlloc    EQU     -6              ;Can1t allocate the audio channel
NDJJnimpl       EQU     -7              ;Unimplemented command
ND_NoWrite      EQU     -8              ;Read for mouth shape without write
ND_Expunged     EQU     -9              ;Can1t open, deferred expunge bit set
ND_PhonErr      EQU     -20             ;Phoneme code spelling error
ND_RateErr      EQU     -21             ;Rate out of bounds
ND_PitchErr     EQU     -22             ;Pitch out of bounds
ND_SexErr       EQU     -23             ;Sex not valid
ND_ModeErr      EQU     -24             ;Mode not valid
ND_FreqErr      EQU     -25             ;Sampling freq out of bounds
ND_VolErr       EQU     -26             ;Volume out of bounds


't              ;_______Write IORequest block
 STRUCTURE NDI,IOSTD_SIZE
```

```
        UWORD   NDI_RATE                ;Speaking rate in words/minute
        UWORD   NDI_PITCH               ;Baseline pitch in Hertz
        UWORD   NDI_MODE                ;FO mode
        UWORD   NDI_SEX                 ;Speaker sex
        APTR    NDI_CHMASKS             ;Pointer to audio channel masks
        UWORD   NDI_NUMMASKS            ;Size of channel masks array
        UWORD   NDI_VOLUME              ;Channel volume
        UWORD   NDI_SAMPFREQ            ;Sampling frequency
        UBYTE   NDI_MOUTHS              ;Generate mouths? (Boolean value)
        UBYTE   NDI_CHANMASK            ;Actual channel mask used (internál use)
        UBYTE   NDI_NUMCHAN             ;Nuraber of channels used (internál use)
        UBYTE   NDI_PAD                 ;For alignment
        LABEL   NDI_SIZE                ;Size of Narrátor IORequest block

*               ;_______Mouth read IORB
 STRUCTURE MRB,NDI_SIZE
        UBYTE   MRB_WIDTH               ;Mouth width
        UBYTE   MRB_HEIGHT              ;Mouth height
        UBYTE   MRB_SHAPE               >Compressed shape (height/width)
        UBYTE   MRB_PAD                 >Alignment
        LABEL   MRB_SIZE

        ENDC    ; DEVICES_NARRÁTOR_I
```

```
        IFND    DEVICES_PARALLEL_I
DEVICES_PARALLEL_I      SET     T
XX
**      SFilename: devices/parallel.i $
**      SRelease: 1.3 $
XX
»*      external declarations for Serial Port Driver
XX
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
*X

    IFND     EXEC_IO_I
    include "exec/io.i"
    ENDC        j EXEC_IO_I

*_____----------------------------------------
X
* Driver error definitions
X
*----------------------------------------------------------------------

ParErr_DevBusy          EQU     1
ParErr_BufTooBig        EQU     2
ParErr_InvParam         EQU     3
ParErr_LineErr          EQU     4
ParErr_NotOpen          EQU     5
ParErr_PortReset        EQU     6
ParErr_InitErr          EQU     7

*----------------------------------------------------------------------
*
* Useful constants
*
*----------------------------------------------------------------------
*
PDCMD_QUERY         EQU     CMD_NONSTD
PDCMD_SETPARAMS     EQU     CMD_NONSTD+1
Par_DEVFINISH       EQU     10          ; number of device comands
*
*----------------------------------------------------------------------
*
* Driver Spécific Commands
*
*----------------------------------------------------------------------

PARALLELNAME:   MACRO
                dc.b    'parallel.device',0
                ds.w    0
                ENDM

        BITDEF  PAR,SHARED,5        ; PARFLAGS non-exclusive access
        BITDEF  PAR,RAD_B00GIE,3    ;    "     (not yet implemented)
        BITDEF  PAR,EOFMODE,1       ;    "     EOF mode enabled bit
        BITDEF  I0PAR,QUEUED,6      ; IO_FLAGS rqst-queued bit
        BITDEF  I0PAR,AB0RT,5       ;    "     rqst-aborted bit
        BITDEF  IOPAR,ACTIVE,4      ;    "     rqst-qued-or-current bit
        BITDEF  IOPT,RWDIR,3        ; I0_STATUS read=0,write=l
        BITDEF  I0PT,PARSEL,2       ;    "     printer selected on the A1000
                                    ; printer selected & serial "Ring Indicator"
                                    ; on the A500/A2000.  Be careful when making
                                    ; cables.
        BITDEF  IOPT,PAPEROUT,1     ;    "     paper out
```

```
        BITDEF  I0PT>PARBUSY,0    ;     "     printer in busy toggle
jNote: Previous versions of this include file had bits 0 and 2 swapped
*
*****************************************************************************

 STRUCTURE  PTERMARRAY,0
        ULONG     PTERMARRAY_0
        ULONG     PTERMARRAY~1
        LABEL     PTERMARRAY^SIZE

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
*  CAUTION !!!  IF YOU ACCESS the parallel.device, you MUST (!!!!) use an
*   IOEXTPAR-sized structure or you may overlay innocent memoryj okay ?!
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 STRUCTURE  IOEXTPAR,IOSTD_SIZE

X       STRUCT    MsgNode
»     0   APTR      Succ
»     4   APTR      Pred
X     8   UBYTE     Type
X     9   UBYTE     Pri
X     A   APTR      Name
X     E   APTR      ReplyPort
X    12   UWORD     MNLength
X       STRUCT    IOExt
X    14   APTR      10_DEVICE
X    18   APTR      IOJJNIT
X    IC   UWORD     10_COMMAND
X    1E   UBYTE     IO~FLAGS
X    IF   UBYTE     10_ERROR
X       STRUCT    IOStdExt
X    20   ULONG     10_ACTUAL
X    24   ULONG     10_LENGTH
X    28   APTR      10_DATA
 *   2C   ULONG     IO_OFFSET
 *


m    30
        ULONG    IO_PEXTFLAGS     ; (not used) flag extension area
        UBYTE    IO_PARSTATUS     ; device status (see bit defs above)
        UBYTE    IolPARFLAGS      ; see PARFLAGS bit definitions above
        STRUCT   10_PTERMARRAY,PTERMARRAY_SIZE ; terrainationchar array
        LABEL    IOEXTPar_SIZE

*****************************************************************************

        ENDC     ; DEVICES_PARALLEL_I
```

```
        IFND    DEVICES_PRINTER_I
DEVICES_PRINTER_I       SET     ~1
XX
IX      SFilename: devices/printer.i $
XX      SRelease: 1.3 $
XX
XX      printer device command definitions
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

   IFND  EXEC_NODES_I
   INCLUDE  "exec/nodes.i"
   ENDC

   IFND  EXEC_LISTS_I
   INCLUDE  "exec/lists.i"
   ENDC

   IFND  EXEC_P0RT5_I
   INCLUDE  "exec/ports.i"
   ENDC

   IFND  EXEC_IO_I
   INCLUDE  "exec/io.i"
   ENDC

   DEVINIT

   DEVCMD   PRD_RAWWRITE
   DEVCMD   PRD_PRTCOMMAND
   DEVCMD   PRD_DUMPRPORT
   DEVCMD   PRD_QUERY

;xxxxxx printer definitions
aRIS     EQU    0 ; ESCc  reset                          ISO
aRIN     EQU    1 ; ESC#1 initialize                     +++
aIND     EQU    2 ; ESCD  lf                             ISO
aNEL     EQU    3 ; ESCE  return,lf                      ISO
aRI      EQU    4 ; ESCM  reverse lf                     ISO

aSGRO    EQU    5 ; ESCfOm normál char set               ISO
aSGR3    EQU    6 ; ESC[3m italics on                    ISO
aSGR23   EQU    7 ; ESCf23m italics off                  ISO
aSGR4    EQU    8 ; ESCUm underline on                   ISO
aSGR24   EQU    9 ; ESC[24m underline off                ISO
aSGRl    EQU   10 ; ESCÍlm boldface on                   ISO
aSGR22   EQU   ηΣ ESCE22m boldface off                  ISO
aSFC     EQU   12 ; SGR30-39  set foreground color      ISO
aSBC     EQU   13 ; SGR40-49  set background color      ISO

aSHORPO  EQU   14 ; ESCtOw normál pitch                  DEC
aSHORP2  EQU   15 ; ESCt2w elité on                      DEC
aSHORPl  EQU   16 ; ESCHw elité off                      DEC
aSHORP4  EQU   17 ; ESC[4w condensed fine on             DEC
aSH0RP3  EQU   18 ; ESCC3w condensed off                 DEC
aSH0RP6  EQU   19 ; ESC[6w enlarged on                   DEC
aSHORP5  EQU   20 ; ESC[5w enlarged off                  DEC

aDEN6    EQU   21 j ESCC6"z shadow print on              DEC (sort of)
aDEN5    EQU   22 i ESC[5"z shadow print off             DEC
aDEN4    EQU   23 i ESCC4"z doublestrike on              DEC
```

```
aDEN3    EQU   24 ; ESC[3"z doublestrike off          DEC
aDEN2    EQU   25 ; ESCC2"z  NLQ on                   DEC
aDENl    EQU   26 5 ESCtT'z  NLQ off                  DEC

aSUS2    EQU   27 ; ESC[2v superscript on             +++
aSUSl    EQU   28 J ESCÍlv superscript off            +++
aSUS4    EQU   29 ; ESCC4v subscript on               +++
aSUS3    EQU   30 ; ESC[3v subscript off              +++
aSUSO    EQU   31 ; ESCfOv normalize the line         +++
aPLU     EQU   32 ; ESCL  partial line up             ISO
aPLD     EQU   33 ; ESCK  partial line down           ISO

aFNTO    EQU   34 ; ESC(B US char set        or Typeface  0 (default)
aFNTl    EQU   35 ; ESCCR French char set    or Typeface  1
aFNT2    EQU   36 ; ESC(K Germán char set    or Typeface  2
aFNT3    EQU   37 ; ESC(A UK char set        or Typeface  3
aFNT4    EQU   38 ; ESC(E Danish I char set  or Typeface  4
aFNT5    EQU   39 ; ESCCH Sweden char set    or Typeface  5
aFNT6    EQU   40 J ESCCY Italian char set   or Typeface  6
aFNT7    EQU   41 ; ESCCZ Spanish char set   or Typeface  7
aFNT8    EQU   42 ; ESC(J Japanese char set  or Typeface  8
aFNT9    EQU   43 ; ESC(6 Norweign char set  or Typeface  9
aFNTl0   EQU   44 ; ESC(C Danish II char set or Typeface 10

5        Suggested typefaces are:

          0 - default typeface.
J         1 - Line Printer or equiv.
\         2 - Pica or equiv.
          3 - Elité or equiv.
          4 - Helvetica or equiv.
          5 - Times Román or equiv.
          6 - Gothic or equiv.
j         7 - Script or equiv.
5         8 - Prestige or equiv.
          9 - Caslon or equiv.
         10 - Orator or equiv.

aPROP2   EQU   45 i ESC[2p  proportional on           +++
aPROPl   EQU   46 5 ESCHp  proportional off          +++
aPROPO   EQU   47 ; ESCtOp  proportional clear       +++
aTSS     EQU   48 ; ESCtn E set proportional offset  ISO
aJFY5    EQU   49 ; ESCt5 F autó left justify        ISO
a.JFY7   EQU   50 ; ESCC7 F autó right justiy        ISO
a.JFY6   EQU   51 ; ESCC6 F autó full justify        ISO
aJFYO    EQU   52 ; ESCfO F autó justify off         ISO
a.JFY2   EQU   53 ; ESCC2 F  word space(auto center) ISO (special)
aJFY3    EQU   54 ; ESCE3 F letter space (justify)   ISO (special)

aVERPO   EQU   55 ; ESCCOz  1/8" line spacing        +++
aVERPl   EQU   56 ; ESCHz  1/6" line spacing         +++
aSLPP    EQU   57 ; ESCtnt  set form length n        DEC
aPERF    EQU   58 ; ESCCnq  perf skip n (n>0)        +++
aPERFO   EQU   59 J ESCEOq  perf skip off            +++

aLMS     EQU   60 ; ESC#9  Left margin set           +++
aRMS     EQU   61 ; ESC#O  Right margin set          +++
aTMS     EQU   62 ; ESC#8  Top margin set            +++
aBMS     EQU   63 ; ESC#2  Bottom marg set           +++
aSTBM    EQU   64 ; ESC[Pnl»Pn2r  T&B margins        DEC
```

```
aSLRM     EQU   65 ; ESC[Pn;Pn2s  L&R margin          DEC
aCAM      EQU   66 ; ESC#3  Clear margins             +++

aHTS      EQU   67 ; ESCH    Set horiz tab            ISO
aVTS      EQU   68 ; ESCJ    Set verticai tabs        ISO
aTBCO     EQU^  69 ; ESC[0g  Clr horiz tab            ISO
aTBC3     EQU   70 ; ESC[3g  Clear all h tab          ISO
aTBCl     EQU   71 ; ESC[1g  Clr verticai tabs        ISO
aTBC4     EQU   72 ; ESC[4g  Clr all v tabs           ISO
aTBCALL   EQU   73 ; ESC#4   Clr all h & v tabs       +++
aTBSALL   EQU   74 ; ESC#5   Set default tabs         +++
aEXTEND   EQU   75 ; ESC[Pn"x extended commands       +++

aRAW      EQU   76 ; ESC[Pn"r Next 'Pn' chars are raw  +++


 STRUCTURE  IOPrtCradReq,IO_SIZE
    UWORD   io_PrtCommand  ; printer command
    UBYTE   io_ParmO    ; first command paraméter
    UBYTE   io_Parml    ; second command paraméter
    UBYTE   io_Parm2    ; third command paraméter
    UBYTE   io_Parm3    ; fourth command paraméter
    LABEL   iopcr_SIZEOF

 STRUCTURE  IODRPReq,IO_SIZE
    APTR    io_RastPort     ; raster port
    APTR    io_ColorMap     ; color map
    ULONG   io_Modes        ; graphics viewport modes
    UWORD   io_SrcX         ; source x origin
    UWORD   io_SrcY         ; source y origin
    UWORD   io_SrcWidth     ; source x width
    UWORD   io_SrcHeight    ; source x height
    LONG    io_DestCols     ; destination x width
    LONG    io_DestRows     ; destination y height
    UWORD   io_Special      ; option flags
    LABEL   iodrpr_SIZEOF

SPÉCIAL_MILCOLS         EQU     $0001   ; DestCols specified in 1/1000"
SPÉCIAL_MILROWS         EQU     $0002   ; DestRows specified in 1/1000"
SPECIAL_FULLCOLS        EQU     $0004   ; make DestCols maximum possible
SPÉCIAL_"FULLROWS       EQU     $0008   ; make DestRows maximum possible
SPECIAL_FRACCOLS        EQU     $0010   ; DestCols is fraction of FULLCOLS
SPECIAL_FRACROWS        EQU     $0020   ; DestRows is fraction of FULLROWS
SPECIAL_CENTER          EQU     $0040   ; center image on paper
SPECIAL_ASPECT          EQU     $0080   ; ensure correct aspect ratio
SPECIAL.DENSITY1        EQU     $0100   ; lowest resolution (dpi)
SPECIAL_DENSITY2        EQU     $0200   ; next res
SPECIAL_DENSITY3        EQU     $0300   ; next res
SPECIAL.DENSITY4        EQU     $0400   ; next res
SPÉCIAL"DENSITY5        EQU     $0500   ; next res
SPECIAL_DENSITY6        EQU     $0600   ; next res
SPECIAL_DENSITY7        EQU     $0700   ; highest res
SPÉCIAL]NOFORMFEED      EQU     $0800   ; don't eject paper after gfx prints
SPECIAL.TRUSTME         EQU     $1000   ; don't reset on gfx prints
;
;       Compute print size, set 'io_DestCols' and ' io_DestRows' in the calling
;       program' s 'IODRPReq' structure and exit> don't print.  This allows the
;       calling program to see what the final print size would be in printer
;       pixels.  Note that it modifies the ' io_DestCols' and ' io_DestRows'
;       fields of your 'IODRPReq' structure.  Also, set the print density and
;       update the ' MaxXDots', ' MaxYDots', 'XDotsInch', and 'YDotsInch' fields
```

```
;       of the 'PrinterExtendedData' structure.
;
SPECIAL_NOPRINT         EQU     $2000   ; see above

PDERR_NOERR             EQU     0       ; clean exit> no errors
PDERR_CANCEL            EQU     1       ; user cancelled print
PDERR_NOTGRAPHICS       EQU     2       ; printer cannot output graphics
PDERR_INVERTHAM         EQU     3       ; OBSOLETE
PDERR_BADDIMENSION      EQU     4       ; print dimensions illegal
PDERR_DIMENSIONOVFLOW   EQU     5       ; OBSOLETE
PDERR_INTERNALMEMORY    EQU     6       ; no memory for internal variables
PDERR_BUFFERMEMORY      EQU     7       ; no memory for print buffer
;
;       Note ; this is an internal error that can be returned from the render
;       function to the printer device.  It is NEVER returned to the user.
;       If the printer device sees this error it converts it ' PDERR_NOERR'
;       and exits gracefully.  Refer to the document on
;       'How to Write a Graphics Printer Driver' for more info.
;
PDERR_TOOKCONTROL       EQU     8       ; I took control in case 0 of render

; internal use
SPECIAL_DENSITYMASK     EQU $0700       ',  masks out density values
SPECIAL_DIMENSIONSMASK  EQU SPECIAL_MILCOLS!SPECIAL_MILROWS!SPECIAL_FULLCOLS"SPEC
IAL_FULLROWS!SPECIAL_FRACCOLS!SPECIAL_FRACROWS!SPECIAL_ASPECT

        ENDC    ; DEVICES_PRINTER_I
```

```
        IFND    DEVICES_PRTBASE_I
DEVICES_PRTBASE_I       SET     1
*X
**      SFilename: devices/prtbase.i $
f X     iRelease: 1.3 $
IX      printer device data definition
»*
IX      (C) Copyright 1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
IX

   IFND  EXEC_NODES_I
   INCLUDE  "exec/nodes.i"
   ENDC
   IFND  EXEC_LIST5_I
   INCLUDE  "exec/llsts.i"
   ENDC
   IFND  EXEC_PORTS_I
   INCLUDE  "exec/ports.i"
   ENDC
   IFND  EXEC_LIBRARIES_I
   INCLUDE  "exec/libraries.i"
   ENDC
   IFND  EXEC_TASKS_I
   INCLUDE  "exec/tasks.i"
   ENDC

   IFND  DEVICES_PARALLEL_I
   INCLUDE  "devices/parallel.i"
   ENDC
   IFND  DEVICES_SERIALJ
   INCLUDE  "devices/serial.i"
   ENDC
   IFND  DEVICES_TIMER_I
   INCLUDE  "devices/timer.i"
   ENDC
   IFND  LIBRARIES_DOSEXTENS_I
   INCLUDE  "libraries/dosextens.i"
   ENDC
   IFND  INTUITION_INTUITION_I
   INCLUDE  "intuition/intuition.i"
   ENDC


 STRUCTURE  DeviceData,LIB_SIZE
    APTR dd_Segment           ; AO when initialized
    APTR dd_ExecBase          ; A6 for exec
    APTR dd_CmdVectors        ; command table for device commands
    APTR dd_CradBytes         ; bytes describing which command queue
    UWORD   dd_NumCoramands   ; the number of commands supported
    LABEL   DeviceData_SIZEOF ; (was dd_SIZEOF)


X_______
x_______device driver priváté variables ____________________________---
X_______
du_Flags EQU   LN_PRI        ; various unit flags

;_______IO_FLAGS
    BITDEF  IO,QUEUED,4       ; command is queued to be performed
    BITDEF  I0,CURRENT,5      5 command is being performed
    BITDEF  I0,SERVICING,6    ; command is being actively performed
```

```
    BITDEF  10,DONÉ,7          » command is done

;------ du Flags
    BITDEF" DU,ST0PPED>0       > commands are not to be performed


«_______Constants _____________----------------------------------------
P PRIORITY      EQU     0
P'STKSIZE       EQU     $0800   ; stack size for child task
P_BUFSIZE       EQU     256     ; size of internál buffers for text i/o
P_SAFESIZE      EQU     128     > safety margin for text output buffer

*-_____pd_Flags _____-
   BITDEF   P,I0R0,0           5 I0R0 is in use
   BITDEF   P,IOR1,1           > I0R1 is in use
   BITDEF   P,EXPUNGED,7       ; device to be expunged when all closed

 STRUCTURE  PrinterData,DeviceData_SIZEOF
    STRUCT  pd_Unit,MP_SIZE    > the one and only unit
    BPTR pd_PrinterSegment     > the printer specific segment
    UWORD   pd_PrinterType     ; the segment printer type
    APTR pd_SegmentData        ; the segment data structure
    APTR pd_PrintBuf           ", the raster print buffer
    APTR pd..PWrite            ; the parallel write function
    APTR pd..PBothReady        ; the parallel write function's done

    IFGT IOEXTPar_SIZE-I0EXT5ER_SIZE
    STRUCT  pd_IORO,IOEXTPar SIZE   j port 1/0 request 0
    STRUCT  pd_IORl,IOEXTParIsiZE   j   and 1 for double buffering
    ENDC

    IFLE IOEXTPar_SIZE-IOEXTSER_SIZE
    STRUCT  pd_I0R0,IOEXTSER_SIZE   ; port 1/0 request 0
    STRUCT  pd_I0Rl,IOEXTSER_SIZE   ;   and 1 for double buffering
    ENDC

    STRUCT  pd_TIOR,IOTV_SIZE       j timer 1/0 request
    STRUCT  pd_IORPort,MP_SIZE      ;   and message reply port
    STRUCT  pd_TC,TC_SIZE           ! write task
    STRUCT  pd_Stk,P_STKSIZE        \   and stack space
    UBYTE   pd_Flags                ; device flags
    UBYTE   pd_pad
    STRUCT  pd_Preferences,pf_SIZEOF i the latest preferences
    UBYTE      pd_PWaitEnabled      ; wait function switch
    LABEL   pd_SIZEOF            '  ; warning! this may be odd

    BITDEF  PPC,GFX,0           jgraphics (bit position)
    BITDEF  PPC,COLOR,1         >color (bit position)

PPC BWALPHA     EQU     $00     >black&white alphanumerics
PPClBWGFX       EQU     $01     Jblack&white graphics
PPCICOLORALPHA  EQU     $02     jcolor alphanumerics
PPC_COLORGFX    EQU     $03     '>color graphics

PCC_BW          EQU     1       Jblack&white only
PCC_YMC         EQU     2       »yellow/magenta/cyan only
PCC_YMC_BW      EQU     3       »yellow/magenta/cyan or black&white
PCC_YMCB        EQU     4       ;yellow/magenta/cyan/black

PCC_4C0L0R      EQU     $4      ;a flag for YMCB and BGRW
PCC_ADDITI VE   EQU     $8      Jnot ymcb but blue/green/red/white
```

```
PCC_WB          EQU     $9      Jblack&white only, 0 == BLACK
PCC_BGR         EQU     Ja      >blue/green/red
PCC_BGR_WB      EQU     Sb      ;blue/green/red or black&white
PCC_BGRW        EQU     Se      ;blue/green/red/white
;       The picture must be scanned once for each color component> as the
;       printer can only define one color at a time.  ie. If ' PCC_YMC then
;       first pass sends all ' Y1 info to printer> second pass sends all 'M'
;       inf0) and third pass sends all C info to printer.  The CalComp
;       PlotMaster is an example of this type of printer.
PCC_MULTI_PASS  EQU     $10     ;see explanation above

 STRUCTURE  PrinterExtendedData>0
    APTR    ped_PrinterName   > printer name> null terminated
    APTR    ped_Init          ; called after LoadSeg
    APTR    ped_Expunge       ; called before UnLoadSeg
    APTR    ped_0pen          > called at OpenDevice
    APTR    ped_Close         ; called at CloseDevice
    UBYTE   ped_PrinterClass  ; printer class
    UBYTE   ped_ColorClass.   > color class
    UBYTE   ped_MaxColumns    ; number of print columns available
    UBYTE   ped_NumCharSets   > number of character sets
    UWORD   ped_NuraRows      ; number of 'pins1 in print head
    ULONG   ped_MaxXDots      ; number of dots maximum in a raster dump
    ULONG   ped_MaxYDots      j number of dots maximum in a raster dump
    UWORD   ped_XDotsInch     ; horizontal dot density
    UWORD   ped_YDotsInch     > verticai dot density
    APTR    ped_Commands      5 printer text command table
    APTR    ped_DoSpecial     ; spécial command handler
    APTR    ped_Render        ; raster render function
    LONG    ped_TimeoutSecs   ; good write timeout
;------ the following only exists if the segment version is 33 or greater
        APTR    ped_8BitChars  Jconversion strings for the extended font
        LONG    ped_PrintMode  jset if text printed5 otherwise 0
;------ the following only exists if the segment version is 34 or greater'
        APTR   ped_ConvFunv    ; ptr to conversion function for all chars
        LABEL  ped_SIZEOF

 STRUCTURE  PrinterSegment,0
    ULONG   ps_NextSegment    ; (actually a BPTR)
    ULONÓ   ps_runAlert       ; MOVEQ #0,D0 : RTS
    UWORD   ps_Version        ; segment version
    UWORD   ps_Revision       ; segment revision
    LABEL   ps_PED            ; printer extended data

        ENDC    ; DEVICES_PRTBASE_I
```

```
        IFND    DEVICES_PRTGFX_I
DEVICES_PRTGFX_I        SET     1
XX
XX      SFilename: devices/prtgfx.i $
XX      SRelease: 1.3 $
XX
XX
*X
XX      (C) Copyright 1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
**

PCMYELLOW       EQU     0               ; byte index for yellow
PCMMAGENTA      EQU     1               ; byte index for magenta
PCMCYAN         EQU     2               ; byte index for cyan
PCMBLACK        EQU     3               ; byte index for black
PCMBLUE         EQU     PCMYELLOW       ; byte index for blue
PCMGREEN        EQU     PCMMAGENTA      ; byte index for green
PCMRED          EQU     PCMCYAN         ; byte index for red
PCMWHITE        EQU     PCMBLACK        ; byte index for white

        STRUCTURE       colorEntry,0
                LABEL   colorLong       ; quick access to all of YMCB
                LABEL   colorSByte      ; 1 entry for each of YMCB
                STRUCT  colorByte,4     ; ditto (except signed)
                LABEL   ce_SIZEOF

        STRUCTURE       PrtInfo,0
                APTR    pi_render       ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_rp           ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_temprp       ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_RowBuf       ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_HamBuf       ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_ColorMap     ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_ColorInt     ; color intensities for entire row
                APTR    pi_HamInt       ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_DestlInt     ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_Dest2Int     ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_ScaleX       ; array of scale values for X
                APTR    pi_ScaleXAlt    ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_dmatrix      ; pointer to dither mátrix
                APTR    pi_TopBuf       ; PRIVÁTÉ - DO NOT USE!
                APTR    pi_BotBuf       ; PRIVÁTÉ - DO NOT USE!

                UWORD   pi_RowBufSize   ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_HamBufSize   ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_CoíorMapSize ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_ColorIntSize ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_HamIntSize   ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_DestlIntSize ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_Dest2IntSize ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_ScaleXSize   ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_ScaleXAltSize 5 PRIVÁTÉ - DO NOT USE!

                UWORD   pi_PrefsFlags   ; PRIVÁTÉ - DO NOT USE!
                ULONG   pi_special      ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_xstart       ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_ystart       i PRIVÁTÉ - DO NOT USE!
                UWORD   pi_width        \ source width (in pixels)
                UWORD   pi_height       ; PRIVÁTÉ - DO NOT USE!
                ULONG   pi_pc           ; PRIVÁTÉ - DO NOT USE!
                ULONG   Pi_pr           ; PRIVÁTÉ - DO NOT USE!
                UWORD   pi_ymult        ; PRIVÁTÉ - DO NOT USE!
```

```
        UWORD   pi_ymod         ; PRIVÁTÉ - DO NOT USE!
        UWORD   pi_ety          ; PRIVÁTÉ - DO NOT USE!
        UWORD   pi_xpos         ; offset to start printing from
        UWORD   pi_threshold    ; copy of threshold value (from prefs)
        UWORD   pi_tempwidth    ; PRIVÁTÉ - DO NOT USE!
        UWORD   pi_flags        ; PRIVÁTÉ - DO NOT USE!
        LABEL   prtinfo_SIZEOF

ENDC    i DEVICES_PRTGFX_I
```

```
        IFND    DEVICES_SCSIDISK_I
DEVICES_SCSIDISK_I      EQU     1
xx
**      $Filename: devices/scsidisk.i $
xx      $Revision: 1.0$
xx      $Date: 88/07/11 15:33:14 $
xx
**      SCSI exec-level device command
x*
**      (C) Copyright 1988 Commodore-Amiga> Inc.
xx          All Rights Reserved
xx
;-------------------------------------------------------------------------
;
;   SCSI Command
;       Several Amiga SCSI controller manufacturers are converging on
;       standard, ways to talk to their controllers.  This include
;       file describes an exec-device command (e.g. for hddisk.device)
;       that can be used to issue SCSI commands
;
;   UNIT NUMBERS
j       Unit numbers to the OpenDevice call have encoded in them which
;       SCSI device is being referred to.  The three decimai digits of
;       the unit number refer to the SCSI Target ID (bus address) in
;       the l's digit, the SCSI logical unit (LUN) in the 10's digit,
;       and the controller board in the 100's digit.
5                                                                   v
;       Examples:
;                 0     drive at address 0
;                12     LUN 1 on multiple drive controller at address 2
;               104     second controller board, address 4
;                88     not valid: both logical units and addresses
;                       rangé from 0..7.
;
;   CAVEATS
;       Original 2090 code did not support this command.
5
;       Commodore 2090/2090A unit numbers are different.  The SCSI
;       logical unit is the 100's digit, and the SCSI Target ID
;       is a permuted l's digit: Target ID 0..6 maps to unit 3..9
;       (7 is reserved for the controller).
9
;           Examples:
;                 3     drive at address 0
;               109     drive at address 6, logical unit 1
;                 1     not valid: this is not a SCSI unit.  Perhaps
;                       it's an ST506 unit.
•
>
j       Somé controller boards generate a unique name (e.g. 2090A's
;       iddisk.device) for the second controller boardj instead of
,       implementing the 100's digit.
>
;       There are optional restrictions on the alignment, bus
;       accessability, and size of the data for the data phase.
;       Be conservative to work with all manufacturer's controllers.
;-------------------------------------------------------------------------

HD_SCSICMD      EQU     28      ; issue a SCSI command to the unit
                                ; io_Data points to a SCSICmd
                                ; io_Length is sizeof(struct SCSICmd)
                               > io_Actual and io_Offset are not used
```

```
 STRUCTURE      SCSICmd,0
    APTR    scsi_Data           ; word aligned data for SCSI Data Phase
                                ; (optional) data need not be byte aligned
                                ; (optional) data need not be bus accessable
    ULONG   scsi_Length         ; even length of Data area
                                ; '(optional) data can have odd length
                                ; (optional) data length can be > 2**24
    ULONG   scsi_Actual         ; actual Data used
    APTR    scsi_Command        ; SCSI Command (same options as scsi_Data)
    UWORD   scsi_CmdLength      ; length of Command
    UWORD   scsi_CmdActual      ; actual Command used
    UBYTE   scsi_Flags          ; includes intended data direction
    UBYTE   scsi_Status         ; SCSI status of command
    LABEL   scsi_SIZEOF


;-------scsi Flags ------
SCSIF_WRITE ~           EQU     0       ; intended data direction is out
SCSIF_READ              EQU     1       ; intended data direction is in

;-------SCSI io_Error values ------
HFERR_SelfUnit          EQU     40      ; cannot issue SCSI command to self
HFERR_DMA               EQU     41      ; DMA error
HFERR_Phase             EQU     42      ; illegal or unexpected SCSI phase
HFERR>arity             EQU     43      ; SCSI parity error
HFERR SelTimeout        EQU     44      ; Select timed out
HFERR!BadStatus         EQU     45      ; status and/or sense error

;-------OpenDevice io_Error values ------
HFERR_NoBoard       ~   EQU     50      ; Open failed for non-existant board

        ENDC    ; DEVICES_SCSIDISK_I
```

```
        IFND DEVICESJSERIALJ
DEVICES_SERIAL_I SET 1

XX      IFilename: devices/serial.i $
XX      tRelease: 1.3 $
XX
XX      external declarations for the serial device
•X
*X      (C) Copyright 1985>1986>1987,1988 Commodore-Amiga> Inc.
IX          Ali Rights Reserved
XX
    IFND     EXEC_IO_I
    include "exec/io.i"
    ENDC        ; EXEC_IO_I

*-------------------------------------------------------------------
*
* Useful constants
*
*-------------------------------------------------------------------
*
SER_DEFAULT_CTLCHAR EQU $11130000  ; default chars for xON,xOFF
; You may change these via SETPARAMS.  At this time, parity is not
\ calculated for xON/xOFF characters.  You must supply them with the
; desired parity.

X
X_____________________________________________________________________.
*
* Driver Specific Commands

SDCMD_QUERY     EQU     CMD_NONSTD
SDCMD_BREAK     EQU     CMD_NONSTD+1
SDCMDISETPARAMS EQU     CMDJíONSTD+2

SER_DEVFINISH   EQU     CMD_NONSTD+2 > number of device comands

*-------------------------------------------------------------------

SERIALNAME:     MACRO
                dc.b    'serial.device',0
                dc.w    0
                ENDM

        BITDEF  SER,XDISABLED>7  ', SERFLAGS xOn-xOff feature disabled bit
        BITDEF  SER,E0FM0DE,6    >     5>     EOF mode enabled bit
        BITDEF  SER,SHARED,5      ;    >1     non-exclusive access
        BITDEF  SER,RAD_B00GIE,4  ;    11     high-speed mode active
        BITDEF  SER,QUEUEDBRK>3   \    11     queue this Break ioRqst
        BITDEF  SER,7WIRE,2       ;    11     RS232 7-wire protocol
        BITDEF  SER,PARTY_ODD.l   ',   11     use-odd-parity bit
        BITDEF  SER,PARTY_ON,0    ;    11     parity-enabled bit
;
JWARNING: The next series of BITDEFs refer to the HIGH order BYTE of
;IO_STATUS.  Example usage: "BTST.B #IOST_XOFFWRITE,IO_STATUS+1(AX)"
;
        BITDEF  IOST,XOFFREAD,4   ; IOST_HOB receive currently xOFF'ed
        BITDEF  IOST,XOFFWRITE,3  ;     "    transmit currently xOFF'ed
        BITDEF  IOST,READBREAK,2  ;     "    break was latest input
        BITDEF  IOST,WROTEBREAK,1 ;     "    break was latest output
        BITDEF  IOST,OVERRUN5O    ;     "    status word RBF overrun
;
;       BITDEF's in a longword field)
```

```
;       Example usage:  BSET.B #SEXTB_MSPON>I0_EXTFLAGS+3(AX)
                                ;IO_EXTFLAGS (extended flag longword)
        BITDEF  SEXT»MSPON,1    ;     "    use mark-space parity,not odd-even
        BITDEF  SEXT,MARK,0     >     "    if mark-space, use mark
*
*****************************************************************************
 STRUCTURE TERMARRAY,0
        ULONG    TERMARRAY_O
        ULONG    TERMARRAYJ
        LABEL    TERMARRAY_SIZE

*****************************************************************
* CAUTION !!  IF YOU ACCESS the serial.device, you MUST (!!!!) use an
*  IOEXTSER-sized structure or you may overlay innocent memory, okay ?!
*****************************************************************

 STRUCTURE  IOEXTSER,IOSTD_SIZE

*      STRUCT   MsgNode
*    0   APTR     Succ
*    4   APTR     Pred
*    8   UBYTE    Type
*    9   UBYTE    Pri
*    A   APTR     Name
*    E   APTR     ReplyPort
*   12   UWORD    MNLength
*      STRUCT   IOExt
*   14   APTR     IO_DEVICE
*   18   APTR     IOJJNIT
*   IC   UWORD    IOICOMMAND
*   1E   UBYTE    IO_FLAGS
*   IF   UBYTE    IO_ERROR
*      STRUCT   IOStdExt
*   20   ULONG    I0_ACTUAL
*   24   ULONG    10_LENGTH
*   28   APTR     IO_DATA
*   2C   ULONG    10_OFFSET
*
*   30
        ULONG   IO_CTLCHAR      ; control char's Corder = xON}x0FF,rsvd}rsvd)
        ULONG   IO_RBUFLEN      ; length in bytes of serial port's read buffer
        ULONG   IO_EXTFLAGS     ; additional serial flags (see bitdefs above)
        ULONG   IO_BAUD         ; baud rate requested (true baud)
        ULONG   10_BRKTIME      ; duration of break signal in MICROseconds
        STRUCT  IOJTERMARRAY,TERMARRAY_SIZE ; termination character array
        UBYTE   IO_READLEN      ; bits per read char (bit count)
        UBYTE   IO_WRITELÉN     ; bits per write char (bit count)
        UBYTE   IO_STOPBITS     ; stopbits for read (count)
        UBYTE   IO_SERFLAGS     > see SERFLAGS bit definitions above
        UWORD   10_STATUS       ; status of serial port, as follows:
*
x                  BIT  ACTIVE  FUNCTION
x                   0           reserved
x                   1           reserved
x                   2    high   Connected to parallel "select" on the A1000.
x                               Connected to both the parallel "select" and
x                               serial "ring indicator" pins on the A500 &
x                               A2000.  Take care when making cables.
x                   3    low    Data Set Ready
x                   4    low    Clear To Send
x                   5    low    Carrier Detect
```

```
x                    6     low     Ready To Send
x                    7     low     Data Terminál Ready
x                    8    high     read overrun
x                    9    high     break sent
x                   10    high     break received
x                   11    high     transmit x-OFF'ed
x                   12    high     receive x-OFF'ed
x                13-15             reserved
x
        LABEL    IOEXTSER_SIZE

*****************************************************************************

*---------------------------------------------------------------------------
x
* Driver error definitions
x
*---------------------------------------------------------------------------

SerErr_DevBusy          EQU      1
SerErr_BufErr           EQU      4       ;Fai led to allocate new read buffer
SerErr_InvParam         EQU      5
SerErr_LineErr          EQU      6
SerErr_ParityErr        EQU      9
SerErr_TimerErr         EQU     11       ;(See the serial/OpenDevice autodoc)
SerErr_BufOverflow      EQU     12
SerErr_NoDSR            EQU     13
SerErr_DetectedBreak    EQU     15


 IFD    DEVICES_SERIAL_I_OBSOLETE
SER_DBAUD               EQU     9600     >unused
SerErr_BaudMismatch     EQU      2       5unused
SerErr_InvBaud          EQU      3       >unused
SerErr_NotOpen          EQU      7       ;unused
SerErr_PortReset        EQU      8       ;unused
SerErr_InitErr          EQU     10       ;unused
SerErr_NoCTS            EQU     14       ">unused
        BITDEF   IOSER.,QUEUED,6     ; io_FLAGS rqst-queued bit
        BITDEF   IOSER.,AB0RT,5      ;      "    rqst-aborted bit
        BITDEF   IOSER.,ACTIVE,4     ;      "    rqst-qued-or-current bit
 ENDC

        ENDC     ; DEVICES_SERIAL_I
```

```
        IFND    DEVICES_TIMER_I
DEVICES_TIMER_I SET     1
XX
**      SFilename: devices/timer.i $
xx      $Release: 1.3 $
XX
XX
XX
i*      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
XX

        IFND    EXEC_IO_I
        INCLUDE "exec/io.i"
        ENDC    ; EXEC_10_I

* unit defintions
UNIT_MICROHZ    EQU     0
UNIT_VBLANK     EQU     1

TIMERNAME       MACRO
                DC.B    'timer.device',0
                DS.W    0
                ENDM

 STRUCTURE TIMEVAL,0
        ULONG   TV_SECS
        ULONG   TV_MICRO
        LABEL   TV_SIZE

 STRUCTURE TIMEREQUEST,I0_SIZE
        STRUCT  IOTV_TIME,TV_SIZE
        LABEL   IOTV_SIZE

* IO_COMMAND to, use for adding a timer
        DEVINIT
        DEVCMD  TR_ADDREQUEST
        DEVCMD  TR_GETSYSTJME
        DEVCMD  TR_SETSYSTIME

        ENDC    ; DEVICES_TIMER_I
```

```
        IFND    DEVICES_TRACKDISK_I
DEVICES_TRACKDISK_I     SET     1
XX
*X      SFilename: devices/trackdisk.i $
**      SRelease: 1.3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

        IFND    EXEC_IO_I
        INCLUDE "exec/io.i"
        ENDC    5 EXEC_IO_I

        IFND    EXEC_DEVICES_I
        INCLUDE "exec/devices.i"
        ENDC    j EXEC_DEVICES_I

*---------------------------------------------------------------------
*
* Physical drive constants
*
*---------------------------------------------------------------------


* OBSOLETE -- only valid for 3 1/4" drives.  Use the TD_GETNUMTRACKS command!
X
*NUMCYLS                EQU     80              ; normál # of cylinders
iMAXCYLS                EQU     NUMCYLS+20      ; max # of cyls to look for
m                                               ;       during a calibrate
•NUMHEADS       EQU     2
*NUMTRACKS      EQU     NUMCYLS*NUMHEADS

NUMSECS         EQU     11
NUMUNITS        EQU     4

X---------------------------------------------------------------------
X
* Useful constants
X
K---------------------------------------------------------------------


*-- sizes before mfm encoding
TD_SECT0R       EQU     512
TD_SECSHIFT     EQU     9                       •, log TD_SECTOR
X                                               ;    2


*---------------------------------------------------------------------
*
* Driver Specific Commands

X---------------------------------------------------------------------

*-- TD_NAME is a generic macro to get the name of the driver.  This
«-- way if the name is ever changed you will pick up the change
*-- automatically.
X--
*-- Normál usage would be:
X--
```

```
X-- internalName:       TD_NAME
*--

TD_NAME:        MACRO
                DC.B    'trackdisk.device' >0
                DS.W    0
                ENDM

        BITDEF  TD,EXTC0M,15

        DEVINIT
        DEVCMD  TD_MOTOR                5 control the disk' s motor
        DEVCMD  TD_"SEEK                5 explicit seek (for testing)
        DEVCMD  TD_FORMÁT               > formát disk
        DEVCMD  TD_REMOVE               > notify when disk changes
        DEVCMD  TD_CHANGENUM            > number of disk changes
        DEVCMD  TD_CHANGESTATE          5 is there a disk in the drive?
        DEVCMD  TD_"PROTSTATUS          > is the disk write protected?
        DEVCMD  TD_RAWREAD              > read raw bits from the disk
        DEVCMD  TD_RAWWRITE             ; write raw bits to the disk
        DEVCMD  TD_GETDRIVETYPE         > get the type of the disk drive
        DEVCMD  TD_GETNUMTRACKS         ; get the # of tracks on this disk
        DEVCMD  TD..ADDCHANGEINT        > TD_REMOVE done right
        DEVCMD  TD_"REMCHANGEINT        5 removes softint set by ADDCHANGEINT
        DEVCMD  TD_LASTCOMM             5 dummy placeholder for end of list


*
*
* The disk driver has an "extended command" facility.  These commands
* take a superset of the normál 10 Request block.
*
ETD_WRITE       EQU     (CMD_WRITE!TDF_EXTCOM)
ETD_READ        EQU     (CMD_READ!TDF_EXTCOM)
ETD_MOTOR       EQU     (TD_M0T0R!TDF_EXTCOM)
ETD_SEEK        EQU     (TD_SEEK!TDF_EXTCOM)
ETD_FORMÁT      EQU     (TD_F0RMAT!TDF_EXTCOM)
ETD_UPDATE      EQU     (CMDJJPDATE !TDF_EXTCOM)
ETD_CLEAR       EQU     (CMD_CLEAR!TDF_EXTCOM)
ETD_RAWREAD     EQU     (TD_RAWREAD!TDF_EXTCOM)
ETD_RAWWRITE    EQU     (TD_RAWWRITE!TDF_EXTCOM)


*
* extended 10 has a larger than normál io request blóck.
X

 STRUCTURE  IOEXTTD,IOSTD_SIZE
        ULONG   IOTD_COUNT      ; removal/insertion count
        ULONG   IOTD_SECLABEL   ; sector label data region
        LABEL   IOTD_SI2E

*
* raw read and write can be synced with the index pulse.  This flag
* in io^equest's IO_FLAGS field telis the driver that you want this.
X

        BITDEF  I0TD>INDEXSYNC,4

* labels are TD_LABELSIZE bytes per sector

TD_LABELSIZE    EQU     16
```

```
*
* This is a bit in the FLAGS field of OpenDevice.  If it is set, then
* the driver will allow you to open all the disks that the trackdisk
* driver understands.  Otherwise only 3.5" disks will succeed.
*
*
        BITDEF  TD,ALL0W_N0N_3_5,0


X
X  If you set the TDB_ALL0W_N0N_3_5 bit in OpenDevice, then you don't
*  know what type of disk you really got.  These defines are for the
X  TD_GETDRIVETYPE command.  In addition, you can find out how many
X  tracks are supported via the TD_GETNUMTRACKS command.
X
DRIVE3_5        EQU     1
DRIVE5_25       EQU     2

X------------------------------------------------
X
* Driver error defines
X
X------------------------------------------------

TDERR_NotSpecified      EQU     20      ;» generál catchall
TDERR_NoSecHdr          EQU     21      5 couldn't evén find a sector
TDERR_BadSecPreamble    EQU     22      ;» sector looked wrong
TDERR_BadSecID          EQU     23      > ditto
TDERR_BadHdrSum         EQU     24      \ header had incorrect checksum
TDERR_BadSecSum         EQU     25      5 data had incorrect checksum
TDERR_TooFewSecs        EQU     26      5 couldn[1]t find enough sectors
TDERR_BadSecHdr         EQU     27      ; another "sector looked wrong"
TDERR_WriteProt         EQU     28      ;» can't write to a protected disk
TDERR_DiskChanged       EQU     29      ;» no disk in the drive
TDERR_SeekError         EQU     30      ;» couldn't find track 0
TDERR_NoMem             EQU     31      > ran out of memory
TDERR_BadUnitNura       EQU     32      5 asked for a unit > NUMUNITS
TDERR_BadDriveType      EQU     33      > not a drive that trackdisk groks
TDERR_DriveInUse        EQU     34      > someone else allocated the drive
TDERR_PostReset         EQU     35      ;» user hit reset; awaiting doom

*------------------------------------------------------------------------
*
* Public portion of unit structure
*
*------------------------------------------------------------------------

 STRUCTURE TDU_PUBLICUNIT,UNIT_SIZE
        UWORD   TDU_C0MP01TRACK         ; track for first precomp
        UWORD   TDU_C0MP10TRACK         ; track for second precomp
        UWORD   TDU_COMP11TRACK         ; track for third precomp
        ULONG   TDU_STEPDEI_AY          ; time to wait after stepping
        ULONG   TDU_SETTLEDELAY         ; time to wait after seeking
        UBYTE   TDU_RETRYCNT            ; # of times to retry
        LABEL   TDU_PUBLICUNITSIZE

        ENDC    ; DEVICES_TRACKDISK_I
```

```
Directory "Lattice_C_5,0.5:Assembler_Headers/exec" on Saturday 29-Sep-90
ables.i                    1886 ---rwed 07-Nov-88  14:57:33
alerts.i                   6777 ---rwed 07-Nov-88  14:57:32
devices.i                  1052 ---rwed 07-Nov-88  14:57:32
errors.i                    466 ---rwed 07-Nov-88  14:57:32
exec.i                      432 -r--rwed 07-Nov-88  14:57:31
execbase.i                 4770 ---rwed 07-Nov-88  14:57:31
execname.i                  278 ---rwed 07-Nov-88  14:57:32
exec_lib.i                 2153 ---rwed 07-Nov-88  14:57:31
funcdef.i                   194 ---rwed 07-Nov-88  14:57:31
initializers.i              880 ---rwed 07-Nov-88  14:57:32
interrupts.i               1488 ---rwed 07-Nov-88  14:57:32
io.i                       2773 ---rwed 07-Nov-88  14:57:33
libraries.i                2903 ---rwed 07-Nov-88  14:57:31
lists.i                    2985 ---rwed 07-Nov-88  14:57:33
memory.i                   2338 ---rwed 07-Nov-88  14:57:29
nodes.i                    1049 ---rwed 07-Nov-88  14:57:32
ports.i                    1214 ---rwed 07-Nov-88  14:57:33
resident.i                 1217 ---rwed 07-Nov-88  14:57:33
semaphores.i               1350 ---rwed 07-Nov-88  14:57:30
strings.i                   849 ---rwed 07-Nov-88  14:57:33
tasks.i                    2230 ---rwed 07-Nov-88  14:57:32
types.i                    1938 ---rwed 07-Nov-88  14:57:33
22 files - 115 blocks - 41222 bytes
```

```
        IFND    EXEC_ABLES_I
EXEC.ABLES^I    SET     1
XX
XX      SFilename: exec/ables.i $
XX      SRelease: 1.3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX        Ali Rights Reserved
XX

    IFND EXECJTYPES_I
    INCLUDE "exec/types.i"
    ENDC        ; EXEC_TYPES_I

    IFND EXEC_EXECBASE_I
    INCLUDE "exec/execbase.i"
    ENDC        ; EXEC_EXECBASE_I


X-------------------------------------------------------------------
X
X   Interrupt Exclusion Macros
X
X-------------------------------------------------------------------

INT_ABLES   MACRO                       * externals for dis/enable
            XREF    _intena
            ENDM


DISABLE     MACRO   x [scratchRegl
            IFC
            MOVE.W  #S04000,_intena     x(NOT IF_SETCLR)+IF_INTEN
            ADDQ.B  #1,1DNestCnt(A6)
            ENDC
            IFNC    '\1',''
            MOVE.L  4,\1
            MOVE.W  #$04000,_intena     *(NOT IF_SETCLR)+IF_INTEN
            ADDQ.B  #1,IDNestCnt(\1>
            ENDC
            ENDM


ENABLE      MACRO   * [scratchReg]
            IFC
            SUBQ.B  #1,1DNestCnt(A6)
            BGE.S   ENABLE\3
            MOVE.W  #I0C000,_intena     *IF_SETCLR+IF_INTEN
ENABLEX3:
            ENDC
            IFNC    '\1',''
            MOVE.L  4,\1
            SUBQ.B  #1,IDNestCnt(\1)
            BGE.S   ENABLEX3
            MOVE.W  #$0C000,_intena
ENABLE\3:
            ENDC
            ENDM


*-------------------------------------------------------------------
*
```

```
*    Tasking Exclusion Macros
*-------------------------------------------------------------------

TASK_ABLES  MACRO
*           INCLUDE "execbase.i" for TDNestCnt offset
            XREF    _LVOPermit
            ENDM

FORBID      MACRO
            IFC     '\1'V
            ADDQ.B  #1,TDNestCnt(A6)
            ENDC
            IFNC    '\1',"
            MOVE.L  4,\1
            ADDQ.B  #1,TDNestCnt(\1)
            ENDC
            ENDM

PERMIT      MACRO
            IFC       '\1V
            JSR     _LVOPermit(A6)
            ENDC
            IFNC    '\1'»"
            MOVE.L  A6,-(SP)
            MOVE.L  4,A6
            JSR     _LVOPermit(A6)
            MOVE.L  (SP)+,A6
            ENDC
            ENDM

        ENDC    5 EXEC_ABLES_I
```

```
        IFND    EXEC_ALERTS_I
EXEC_ALERTS_I   SET     1
XX
**      SFilename: exec/alerts.i $
**      SRelease: 1.3 $
XX

XX
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
XX

    BITDEF  S,ALERTWACK,1       * in ExecBase.SysFlags


XXXXXXXXXXXXXXXXXXKXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X
*  Formát of the alert error number:
X
*    +---------------+---------------+-------------------------------+
*    !D!  SubSysId   i  General Error i    SubSystem Specific Error    !
X    +_______________+_______________+_______________________________-+
X
*                   D:  DeadEnd alert
*             SubSysId:  indicates ROM subsystem number.
*        General Error:  roughly indicates what the error was
*       Specific Error:  indicates more detail
*************************************************************************


X
*  Use this macro for causing an alert.  THIS MACRO MAY CHANGE!
*  It is very sensitive to memory corruption.... üke stepping on
*  location 4'  But it should work for now.          >
X
ALERT           macro   (alertNumber, paramArray, scratch)
                movem.l d7/a5/a6,-(sp)
                move.l  #\l,d7
                IFNC    '\2' ,"
                lea     \2,a5
                ENDC
                move.1  4,a6            ; (use proper name!!!)
                jsr     _LV0Alert(a6)
                movem.1 7sp)+,d7/a5/a6
                endm


*************************************************************************
X
*  General Dead-End Alerts
X
*  For example:  timer.device cannot open math.library:
X
*  ALERT  (AN_Ti  merDev!AG_OpenLi  b!AO_MathLi  b),(AO),Al
X
*************************************************************************

;_______alert types
AT_DeadEnd      equ $80000000
AT_Recovery     equ $00000000

;_______generál purpose alert codes
AG_NoMemory     equ $00010000
AG_MakeLib      equ $00020000
AG_Open.Lib     equ $00030000
```

```
AG_OpenDev      equ $00040000
AG_OpenRes      equ $00050000
AG_IOError      equ $00060000
AG_NoSignal     equ $00070000

;-------alert objects:
A0_ExecLib      equ $00008001
AO_GraphicsLib  equ $00008002
A0_LayersLib    equ $00008003
ACMntuition     equ $00008004
AOJ-lathLib     equ $00008005
AO~CListLib     equ $00008006
AO~DOSLib       equ $00008007
AO_RAMLib       equ $00008008
AO_IconLib      equ $00008009
AO_ExpansionLib equ $0000800A
AO_AudioDev     equ $00008010
AO_ConsoleDev   equ $00008011
AO_GamePortDev  equ $00008012
AO_KeyboardDev  equ $00008013
AO_TrackDiskDev equ $00008014
AO_TimerDev     equ $00008015
AO_CIARsrc      equ $00008020
AO_DiskRsrc     equ $00008021
AO_MiscRsrc     equ $00008022
AO_BootStrap    equ $00008030
AO_Workbench    equ $00008031


*****************************************************************************
x
x   Specific Dead-End Alerts:
x
x   For example:   exec.library -- corrupted memory list
*
x           ALERT AN_MemCorrupt,(A0),Al
x
*****************************************************************************

;------ exec.library
AN_ExecLib      equ $01000000
AN_ExcptVect    equ $81000001   ; 68000 exception vector checksum
AN_BaseChkSum   equ $81000002   ; execbase checksum
AN_LibChkSum    equ $81000003   ; library checksum failure
AN_LibMem       equ $81000004   ; no memory to make library
AN_MemCorrupt   equ $81000005   ; corrupted memory list
AN_IntrMem      equ $81000006   ; no memory for interrupt servers
AN_InitAPtr     equ $81000007   ; InitStructO of an APTR source
AN_SemCorrupt   equ $81000008   ; a semaphore is in illegal state
AN_FreeTwice    equ $81000009   ; freeing memory that is already free
AN_BogusExcpt   equ $8100000A   ; illegal 68k exception taken

;-------graphics.library
AN_GraphicsLib  equ $02000000
AN_GfxNoMem     equ $82010000   ; graphics out of memory
AN_LongFrame    equ $82010006   ; long frame> no memory
AN_ShortFrame   equ $82010007   ; short framej no memory
AN_Te xt TmpRas equ $02010009   ; texts no memory for TmpRas
AN_BltBitMap    equ $8201000A   ; BltBitMap, no memory
AN_RegionMemory equ $8201000B   ; regions> memory not available
ANJfekeVPort    equ $82010030   ; bfakeVPort, no memory
```

```
AN_GfxNoLCM     equ $82011234   ; emergency memory not available

;-------layers.library
AN_LayersLib    equ $03000000
AN_LayersNoMem  equ $83010000   ; layers out of memory

;------- intuition.library
AN_Intuition    equ $04000000
AN_GadgetType   equ $84000001   ; unknown gadet type
AN_BadGadget    equ $04000001   ; Recovery form of AN_GadgetType
AN_CreatePort   equ $84010002   ; create port, no memory
AN_ItemAlloc    equ $04010003   ; item pláne alloc, no memory
AN_SubAlloc     equ $04010004   ; sub allocj no memory
AN_PlaneAlloc   equ $84010005   ; pláne alloc> no memory
AN_ItemBoxTop   equ $84000006   ; item box top' < RelZero
AN_OpenScreen   equ $84010007   ; open screen» no memory
AN_OpenScrnRast equ $84010008   ; open screen} raster alloc$ no memory
AN_SysScrnType  equ $84000009   ; open sys screen» unknown type
AN_AddSWGadget  equ $8401000A   ; add SW gadgets, no memory
AN_OpenWindow   equ $8401000B   ; open window, no memory
AN_BadState     equ $8400000C   ; Bad State Return entering Intuition
AN_BadMessage   equ $8400000D   ; Bad Message réceived by IDCMP
AN_WeirdEcho    equ $8400000E   ; Weird echo causing incomprehension
AN_NoConsole    equ $8400000F   ; couldn't open the Console Device

;-------math.library
AN_MathLib      equ $05000000

;-------clist.library
AN_CListLib     equ $06000000

;-------dos.library
AN_DOSLib       equ $07000000
AN StartMem     equ $07010001   j no memory at startup
AlTEndTask      equ $07000002   ; EndTask didn't
AfToPktFail     equ $07000003   > Qpkt failure
AN_AsyncPkt     equ $07000004   ; Unexpected packet received
AN_FreeVec      equ $07000005   ', Freevec fai led
AN_DiskBlkSeq   equ $07000006   j Disk block sequence error
AN_BitMap       equ $07000007   ; Bitmap corrupt
AN~KeyFree      equ $07000008   > Key already free
AN_BadChkSum    equ $07000009   > Invalid checksum
AN~DiskError    equ $0700000A   ; Disk Error
ANJÍeyRange     equ $0700000B   > Key out of rangé
AN_Bad0verlay   equ $07000000   ; Bad overlay

;-------ramiib.library
AN_RAMLib       equ $08000000
AN_BadSegList   equ $08000001   j overlays are illegal for library segments

j-------icon.library
AN_IconLib      equ $09000000

;-------expansion.library
AN_ExpansionI_ib equ $0A000000
AN_BadExpansionFree     equ S0A000001

;-------audio.device
AN_AudioDev     equ $10000000

;-------console.device
```

```
AN_ConsoleDev   equ $11000000

;-------gameport.device
AN_GamePortDev  equ $12000000

;-------keyboard.device
AN_KeyboardDev  equ $13000000

;-------trackdisk.device
AN_TrackDiskDev equ $14000000
ANlTDCalibSeek  equ $14000001   ; calibrate: seek error
AN_TDDelay      equ $14000002   ; delay: error on timer wait

;-------timer.device
AN_TimerDev     equ $15000000
ANJTMBadReq     equ $15000001   ; bad request
AlITMBadSupply  equ $15000002   ; power supply does not supply ticks

;-------cia.resource
AN_CIARsrc      equ $20000000

;-------disk.resource
AN_DiskRsrc     equ $21000000
AN_DRHasDisk    equ $21000001   ; get unit: already has disk
AN_DRIntNoAct   equ $21000002   ; interrupt: no active unit

;-------misc.resource
ANJüscRsrc      equ $22000000

;-------bootstrap
AN_BootStrap    equ $30000000
AN_BootError    equ $30000001   ; boot code returned an error

;-------workbench
ANJttorkbench   equ $31000000

;-------DiskCopy
AN_DiskCopy     equ $32000000

        ENDC    ; EXEC_ALERTS_I
```

```
        IFND    EXEC_DEVICES_I
EXEC_DEVICES_I  SET     1
XX
**      SFilename:  exec/devices:i $
**      SRelease:  1.3 $
XX
XX
XX
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga,  Inc.
**          Ali Rights Reserved
XX

    IFND EXEC_LIBRARIES_I
    INCLUDE "exec/libraries.i"
    ENDC        ; EXEC_LIBRARIES_I

    IFND EXEC_PORTS_I
    INCLUDE "exec/ports.i"
    ENDC        ; EXEC_PORTS_I


*-------------------------------------------------------------------------
X
X   Device Data Structure
X
*-------------------------------------------------------------------------

 STRUCTURE  DD,LIB_SIZE
    LABEL   DD_SIZE                     * identical to library


*-------------------------------------------------------------------------
X
*   Suggested Unit Structure
X
X-------------------------------------------------------------------------

 STRUCTURE  UNIT,MP_SIZE                * queue for requests
    UBYTE   UNIT_FLAGS
    UBYTE   UNIT_pad
    UWORD   UNIT_OPENCNT
    LABEL   UNIT_SIZE


«_______UNIT_FLAG definitions:

    BITDEF    UNIT,ACTIVE,0             x driver is active
    BITDEF    UNIT,INTASK,1             x running in driver's task

        ENDC    J EXEC_DEVICES_I
```

```
        IFND    EXEC_ERRORS_I
EXEC_ERRORSJ    SET     1
XI
XX      JFilename: exec/errors.i $
XX      SReleasíí: 1.3 $
XX
XX      Standard 10 Errors:
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rightf> Reserved
XX

IOERR_OPENFAIL  EQU     -1              * device/unit failed to open
IOERR_ABORTED   EQU     -2              x request aborted
IOERR_NOCMD     EQU     -3              * command not supported
IOERR_BADLENGTH EQU     -4              x not a valid length


ERR_OPENDEVICE  EQU  IOERR_OPENFAIL     * REMOVE !!!

        ENDC    j EXEC_ERRORS_I
```

```
        IFND    EXEC EXEC I
EXEC_EXEC_I     SET "   l"
*X
**      SFilename: exec/exec.i $
**      SRelease: 1.3 $

f*
**
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
**

        INCLUDE "exec/nodes.i"
        INCLUDE "exec/lists.i"
        INCLUDE "exec/interrupts.i"
        INCLUDE "exec/memory.i"
        INCLUDE "exec/ports.i"
        INCLUDE "exec/tasks.i"
        INCLUDE "exec/libraries.i"
        INCLUDE "exec/devices.i"
        INCLUDE "exec/io.i"

        ENDC    ; EXEC_EXEC_I
```

```
        IFND    EXEC EXECBASE_I
EXEC_EXECBASE_I SET "   1
XX
XX      JFilename: exec/execbase.i $
KK      iRelease: 1.3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

    IFND EXECJTPES_I
    INCLUDE "exec/types.i"
    ENDC        ; EXEC_TYPES_I

    IFND EXEC_LISTS_I
    INCLUDE "exec/lists.i"
    ENDC        J EXEC_LIST5_I

    IFND EXEC_INTERRUPTS_I
    INCLUDE "exec/interrupts.i"
    ENDC        ; EXEC_INTERRUPTS_I

    IFND EXEC_LIBRARIES_I
    INCLUDE"exec/libraries.i"
    ENDC        ; EXEC_LIBRARIES_I


xxxxxxx  Static  System  Variables  ************************************

 STRUCTURE  ExecBase,LIB_SIZE          ; Standard library node

            UWORD       SoftVer        » kickstart release number
            WORD        LowMemChkSum   ; checksum of 68000 trap vectors
            ULONG       ChkBase        i system base pointer complement
            APTR        ColdCapture    ; cold soft capture vector
            APTR        CoolCapture    "> cool soft capture vector
            APTR        WarmCapture    ; warm soft capture vector
            APTR        SysStkUpper    > system stack base   (upper bound)
            APTR        SysStkLower    j top of system stack (lower bound)
            ULONG       MaxLocMem      ; last calculated local memory max
            APTR        DebugEntry     ; global debugger entry point
            APTR        DebugData      ; global debugger data segment
            APTR        AlertData      J alert data segment
            APTR        MaxExtMem      ; top of extended mem, or null if nőne

            WORD        ChkSum         > for all of the above


xxxxxxx  Interrupt  Related  ****************************************

            LABEL       IntVects
            STRUCT      IVTBE,IV_SIZE
            STRUCT      IVDSKBLK,IV_SIZE
            STRUCT      IVSOFTINT,IV_SIZE
            STRUCT      IVPORTS,IV_SIZE
            STRUCT      IVCOPER,IV^SIZE
            STRUCT IWERTB, I V_S I ZE
            STRUCT      IVBLIT,IV_SIZE
            STRUCT      IVAUDO,IV_SIZE
            STRUCT      IVAUD1,IV_SIZE
            STRUCT      IVAUD2,IV_SIZE
            STRUCT      IVAUD3,IvlsIZE
```

```
    STRUCT      IVRBF,IV_SIZE
    STRUCT      IVDSKSYNC,IV SIZE
    STRUCT      IVEXTER,IV_s7zE
    STRUCT      IVINTEN,IV SIZE
    STRUCT      IVNMI.IV_SÍZE

xxxxxxx Dynamic System Variables xxxxx*xxxx*x*x*xxxxxxxxxxx*xxxxx*****

    APTR        ThisTask        ; pointer to current task
    ULONG       IdleCount       ; idle counter
    ULONG       DispCount       ; dispatch counter
    UWORD       Quantum         ; time slice quantum
    UWORD       Elapsed         ; current quantum ticks
    UWORD       SysFlags        ; mise system flags
    BYTE        IDNestCnt       ; interrupt disable nesting count
    BYTE        TDNestCnt       j task disable nesting count

    UWORD       AttnFlags       ; spécial attention flags
    UWORD       AttnResched     ; rescheduling attention
    APTR        ResModules      ; pointer to resident modulé array

    APTR        TaskTrapCode    ; default task trap routine
    APTR        TaskExceptCode  ; default task exception code
    APTR        TaskExitCode    ; default task exit code
    ULONG       TaskSigAlloc    ; preallocated signal mask
    UWORD       TaskTrapAlloc   ; preallocated trap mask

******* System List Headers ****************************************

    STRUCT      MemList,LH_SIZE
    STRUCT      ResourceList,LH_SIZE
    STRUCT      DeviceList,LH_SÍZE
    STRUCT      IntrList,LH_sIZE
    STRUCT      LibList,LH_SIZE
    STRUCT      PortList,LH_SIZE
    STRUCT      TaskReady,LH_SIZE
    STRUCT      TaskWait,LH_SIZE

    STRUCT      SoftInts,SH_SIZE*5

    STRUCT      LastAlert,4*4


    ;------ these next two variables are provided to allow
    ;------ system developers to have a rough idea of the
    ;------ period of two externally controlled signals --
    ;------ the time between verticai blank interrupts and the
    ;------ external line rate (which is counted by CIA A's
    ;------ "time of day" clock).  In generál these values
    ;------ will be 50 or 60) and may or may not track each
    ;------ other.  These values replace the obsolete AFB_PAL
    ;------ anü AFB_50HZ flags.
    UBYTE       VBlankFrequency
    UBYTE       PowerSuppiyFrequency

    STRUCT      SemaphoreList,LH_S I ZE

    ;------ these next two are to be able to kickstart intő user ram.
    ;------ KickMemPtr holds a singly linked list of MemLists which
```

```
            ;--------will be reraoved from the memory üst via AllocAbs.  If
            ;--------aii the AllocAbs's succeeded, then the KickTagPtr will
            ;--------5e added to the rom tag list.
            APTR        KickMemPtr      ; ptr to queue of mem lists
            APTR        KickTagPtr      ; ptr to rom tag queue
            APTR        KickCheckSum    j checksum for mem and tags

            STRUCT      ExecBaseReserved>10
            STRUCT      ExecBaseNewReserved,20

            LABEL       SYSBASESIZE

*******AttnFlags
*  Processors and Co-processors:
        BITDEF  AF,68010,0      ; alsó set for 68020
        BITDEF  AF,68020,1
        BITDEF  AF,68881,4

» These two bits used to be AFB_PAL and AFB_50HZ.  After somé sóul
> searching we reálized that they were misnomers, and the information
» is now kept in VBlankFrequency and PowerSupplyFrequency above.
; To find out what sort of videó conversion is done, look in the
> graphics subsytem.
        BITDEF  AF,RESERVED8,8
        BITDEF  AF,RESERVED9,9

        ENDC    ; EXEC_EXECBASE_I
```

```
        IFND    EXEC_EXECNAME_I
EXEC_EXECNAME_I SET     1
x*
**      SFilename:  exec/execname.i $
**      SRelease:  1.3 $
K*

»X
*x      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga,  Inc.
**          Ali Rights Reserved
XX
EXECNAME        macro
                de.b    'exec.1ibrary',0
                ds.w    0
                endm

        ENDC    \ EXEC_EXECNAME_I
```

```
        IFND    EXEC_EXEC_LIB_I
EXEC_EXEC_LIBJ SET     1
*X
**      SFilename: exec/exec_lib,i $
**      SRelease: 1.3 $
**
»*
**
**      (C) Copyright 1985,1986,1987,1988 Commodore-Araiga,  Inc.
**          Ali Rights Reserved
**


        FUNCDEF Supervisor
        FUNCDEF Exitlntr
        FUNCDEF Schedule
        FUNCDEF Reschedule
        FUNCDEF Switch
        FUNCDEF Dispatch
        FUNCDEF Exception
        FUNCDEF  InitCode
        FUNCDEF  InitStruct
        FUNCDEF MakeLibrary
        FUNCDEF MakeFunctions
        FUNCDEF FindResident
        FUNCDEF  InitResident
        FUNCDEF Alert
        FUNCDEF Debug
        FUNCDEF Disable
        FUNCDEF Enable
        FUNCDEF Forbid
        FUNCDEF Permit
        FUNCDEF SetSR
        FUNCDEF SuperState
        FUNCDEF UserState
        FUNCDEF SetlntVector
        FUNCDEF AddlntServer
        FUNCDEF RemlntServer
        FUNCDEF Cause
        FUNCDEF Allocate
        FUNCDEF Deallocate
        FUNCDEF AllocMem
        FUNCDEF AllocAbs
        FUNCDEF FreeMem
        FUNCDEF AvailMem
        FUNCDEF AllocEntry
        FUNCDEF FreeEntry
        FUNCDEF  Insert
        FUNCDEF AddHead
        FUNCDEF AddTail
        FUNCDEF Reraove
        FUNCDEF RemHead
        FUNCDEF RemTail
        FUNCDEF Enqueue
        FUNCDEF FindName
        FUNCDEF AddTask
        FUNCDEF RemTask
        FUNCDEF FindTask
        FUNCDEF SetTaskPri
        FUNCDEF SetSignal
        FUNCDEF SetExcept
        FUNCDEF Wait
        FUNCDEF Signal
        FUNCDEF AllocSignal
        FUNCDEF FreeSignal
```

```
    FUNCDEF AllocTrap
    FUNCDEF FreeTrap
    FUNCDEF AddPort
    FUNCDEF RemPort
    FUNCDEF PutMsg
    FUNCDEF GetMsg
    FUNCDEF ReplyMsg
    FUNCDEF WaitPort
    FUNCDEF FindPort
    FUNCDEF AddLibrary
    FUNCDEF RemLibrary
    FUNCDEF OldOpenLibrary
    FUNCDEF CloseLibrary
    FUNCDEF SetFunction
    FUNCDEF SumLibrary
    FUNCDEF AddDevice
    FUNCDEF RemDevice
    FUNCDEF OpenDevice
    FUNCDEF CloseDevice
    FUNCDEF DoIO
    FUNCDEF SendIO
    FUNCDEF CheckIO
    FUNCDEF WaitIO
    FUNCDEF AbortIO
    FUNCDEF AddResource
    FUNCDEF RemResource
    FUNCDEF OpenResource
    FUNCDEF RawIOInit
    FUNCDEF RawMayGetChar
    FUNCDEF RawPutChar
    FUNCDEF RawDoFmt
    FUNCDEF GetCC
    FUNCDEF TypeOfMem
    FUNCDEF Procure
    FUNCDEF Vacate
    FUNCDEF OpenLibrary
    FUNCDEF InitSemaphore
    FUNCDEF ObtainSemaphore
    FUNCDEF ReleaseSemaphore
    FUNCDEF AttemptSemaphore
    FUNCDEF ObtainSemaphoreList
    FUNCDEF ReleaseSemaphoreList
    FUNCDEF FindSemaphore
    FUNCDEF AddSemaphore
    FUNCDEF RemSemaphore
    FUNCDEF SumKickData
    FUNCDEF AddMemList
    FUNCDEF CopyMem
    FUNCDEF CopyMemQuick

    ENDC    ; EXEC_EXEC_LIB_I
```

```
* FUNCDEF macro definition for 'exec/exec_lib.i1

FUNCDEF     MACRO     *function
_LVO\1      EQU       FUNC_CNT
FUNC_CNT    SET       FUNC_CNT-6
            ENDM
FUNC_CNT    SET       5*-6
```

```
        IFND    EXEC_INITIALIZERS_I
EXECJNITIALIZERS_I      SET     1~
XX
XX      JFilename: exec/initializers.i $
XX      SRelease: 1.3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          All Rights Reserved
XX

INITBYTE        MACRO   * &offset,&value
                DC.B    $e0
                DC.B    0
                DC.W    \1
                DC.B    \2
                DC.B    0
                ENDM

INITWORD        MACRO   * &offset,&value
                DC.B    $d0
                DC.B    0
                DC.W    \1
                DC.W    \2
                ENDM

INITLONG        MACRO   x &offset,&value
                DC.B    $c0
                DC.B    0
                DC.W    \1
                DC.L    \2
                ENDM

INITSTRUCT  MACRO   * &size,&offset,&value,&count
            DS.W    0
            IFC     '\4' ,"
C0UNTX3     SET     0
            ENDC
            IFNC    '\4' ,"
C0UNT\3     SET     \4
            ENDC
CMD\3       SET     (((\1)<<4)!COUNT\3)
            IFLE    (\2)-255
            DC.B    (CMD\3)!$80
            DC.B    \2
            MEXIT
            ENDC
            DC.B    CMD\3!$0C0
            DC.B    (((\2)>>16)&$0FF)
            DC.W    ((\2 )&$OFFFF)
            ENDM

        ENDC    ; EXEC_INITIALIZERS_I
```

```
        IFND    EXEC_INTERRUPT5_I
EXECJNTERRUPTS_I        SET     ~1

**      SFilename: exec/interrupts.i $
**      ÍRelease: 1.3 S
**
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga,  Inc.
**          Ali Rights Reserved
**

    IFND EXEC_NODES_I
    INCLUDE "exec/nodes.i"
    ENDC        ; EXEC_NODES_I

    IFND EXEC_LISTS_I
    INCLUDE "exec/lists.i"
    ENDC        J EXEC_LISTS_I


*
*   Interrqpt Structure
*

 STRUCTURE  IS,LN_SIZE
    APTR    IS_DATA
    APTR    IS_CODE
    LABEL   IS_SIZE


*
*   Exec Internál Interrupt Vectors
*

 STRUCTURE  IV,0
    APTR    IVJ3ATA
    APTR    IV CODE
    APTR    IV~NODE
    LABEL   IV~SIZE


*------ System Flag bits (in SysBase.SysFlags )

    BITDEF  S,SAR,15        * scheduling attention required
    BITDEF  S,TQE,14        * time quantum expended -- time to resched
    BITDEF  S,SINT,13


*
*   Software Interrupt List Headers
*

 STRUCTURE  SH,LH_SIZE
    UWORD   SH PAD
    LABEL   SlTsiZE

SIH_PRIMASK  EQU        SOFO
```

```
SIH_QUEUES    EQU        5

** this is a fake INT definition> used only for AddIntServer and the üke
        BITDEF   INT,NMI,15

        ENDC     ; EXEC_INTERRUPTS_I
```

```
        IFND    EXEC 10 I
EXEC_10_I       SET "  "1
**
        SFilename: exec/io.i $
XX      SRelease: 1.3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX        Ali Rights Reserved
XX

    IFND EXEC_PORTS I
    INCLUDE "exec/pőrts.i"
    ENDC        5  EXEC_PORTS_I

    IFND EXEC_LIBRARIES_I
    INCLUDE "exec/libraries.i"
    ENDC        ;  EXEC_LIBRARIESJ


*-------------------------------------------------------------------
x
*   10 Request Structures
x
x———————————————————————————,———————————————————————————.-

x———————Required portion of 10 request:

 STRUCTURE  IO,MN_SIZE
    APTR    IO_DEVICE                   * device node pointer
    APTR    IO_UNIT                     * unit (driver privátě)
    UWORD   IO~COMMAND                  * device command
    UBYTE   IOJFLAGS                    * spécial flags
    BYTE    IO_ERROR                    * error or warning code
    LABEL   10_SIZE
          |

*------ Standard 10 request extension:

    ULONG   IO_ACTUAL                   * actual # of bytes transfered
    ULONG   IO_LENGTH                   * requested # of bytes transfered
    APTR    IO_DATA                     * pointer to data area
    ULONG   IO_OFFSET                   x offset for seeking devices
    LABEL   IOSTD_SIZE


x———————IO_FLAGS bit definitions:

    BITDEF  I0,QUICK,0                * complete 10 quickly


*-------------------------------------------------------------------
x
*   Standard Device Library Functions
x
*-------------------------------------------------------------------

            LIBINIT

            LIBDEF  DEV_BEGINIO         x process 10 request
            LIBDEF  DEV_ABORTIO         x abort 10 request
```

```
X-------------------------------------------------------------
X
X    10 Function Macros
X
X-------------------------------------------------------------

BEGINIO     MACRO
            LINKLIB DEV_BEGINIO,IO_DEVICE(A1)
            ENDM

ABORTIO     MACRO
            LINKLIB DEV_ABORTIO,10_DEVICE(A1)
            ENDM


i-------------------------------------------------------------
X
*    Standard Device Command Definitions
X
*-------------------------------------------------------------

x_____Command definition macro:
DEVINIT     MACRO   • [baseOffset]
            IFC      '\1',"
CMD_COUNT   SET     CMD_NONSTD
            ENDC
            IFNC    '\1',"
CMD_COUNT   SET     \1
            ENDC
            ENDM

DEVCMD      MACRO   * cmdname
\1          EQU     CMD_COUNT
CMD_COUNT   SET     CMD_COUNT+1
            ENDM
            \

*_____Standard device coraraands:

            DEVINIT 0

            DEVCMD  CMD_INVALID         X invalid command
            DEVCMD  CMD_RESET           X reset as if just inited
            DEVCMD  CMD_READ            X standard read
            DEVCMD  CMD_WRITE           X standard write
            DEVCMD  CMD.JJPDATE         X write out all buffers
            DEVCMD  CMD""CLEAR          X clear all buffers
            DEVCMD  CMD""STOP           X hold current and queued
            DEVCMD  CMD_START           X restart after stop
            DEVCMD  CMD]JLUSH           X abort entire queue


*_____- Firet non-e4:flnHs. . . . . irnmanrl 179 1 1.10.

            DEVCMD  CMD_NONSTD

        ENDC    ; EXEC_IO_I
```

```
        IFND    EXEC_LIBRARIES_I
EXEC_LIBRARIES_I        SET     1
XX
**      SFilename: exec/libraries.i $
»*      JRelease: 1.3 1
**
XX
XX
*x      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
**

    IFND EXEC_NODES_I
    INCLUDE "exec/nodes.i"
    ENDC        ; EXEC_NODESJ


*------ Special Constants ------------------------------------

LIB_VECTSI2E    EQU     6
LIB_RESERVED    EQU     4
LIB_BASE        EQU     SFFFFFFFA       * (-LIB_VECTSIZE)
LIB_USERDEF     EQU     LIB_BASE-(LIB_RESERVED*LIB_VECTSIZE)
LIB_NONSTD      EQU     LIB_USERDEF

*------------------------------------------------------------------
X
X   Library Definition Macros
*
*------------------------------------------------------------------

*_____LIBINIT sets base offset for library function definitions:
m_____1.11,

LIBINIT     MACRO   * [baseOffsetl
            IFC     ' W ,''
COUNT_LIB   SET     LIBJJSERDEF
            ENDC
            IFNC    '\r,"
COUNT_LIB   SET     \1
            ENDC
            ENDM


x_____LIBDEF is used to define each library function entry:

LIBDEF      MACRO   * libraryFunctionSymbol
\1          EQU     COUNTJ-IB
COUNT_LIB   SET     COUNT_LIB-LIB_VECTSIZE
            ENDM


i-------------------------------------------------------------
X
x   Standard Library Functions
X
                                                    ------------

    LIBINIT LIB_BASE

    LIBDEF  LIB_OPEN
    LIBDEF  LIB_CLOSE
    LIBDEF  LIB_EXPUNGE
    LIBDEF  LIB_EXTFUNC                 * reserved *
```

```
*------------------------------------------------------------------
*   Standard Library Data Structure
*
*------------------------------------------------------------------

 STRUCTURE LIB,LN_SIZE
    UBYTE   LIBJ1AGS
    UBYTE   LIB^pad
    UWORD   LIB_NEGSIZE                 * number of bytes before LIB
    UWORD   LIB_POSSIZE                 * number of bytes after LIB
    UWORD   LIB_VERSION                 * major
    UWORD   LIB_REVISION                * minor
    APTR    LIB_IDSTRING                * identification
    ULONG   LIB_SUM                     * the checksum itself
    UWORD   LIB_OPENCNT                 * number of current opens
    LABEL   LIB_SIZE


*------   LIB_FLAGS bit definitions:

    BITDEF  LIB,SUMMING,0               * we are currently checksumming
    BITDEF  LIB,CHANGED,1               * we have just changed the lib
    BITDEF  LIB,SUMUSED,2               * set if we should bother to sum
    BITDEF  LIB,DELEXP,3                * delayed expunge


*------------------------------------------------------------------
*
*   Function Invocation Macros
*
*------------------------------------------------------------------

*------   CALLLIB for calling functions where A6 is already correct:

CALLLIB     MACRO   * functionOffset
        IFGT NARG-1
            FAIL    !!! CALLLIB MACRO - too many arguments !!!
        ENDC
            JSR     \1(A6)
            ENDM


*------ LINKLIB for calling functions where A6 is incorrect:

LINKLIB     MACRO   * functionOffset,libraryBase
        IFGT NARG-2
            FAIL    !!! LINKLIB MACRO - too many arguments !!!
        ENDC
            MOVE.L  A6,-(SP)
            MOVE.L  \2,A6
            CALLLIB \1
            MOVE.L  (SP)+,A6
            ENDM

        ENDC    5 EXEC_LIBRARIES_I
```

```
        IFND    EXECJJSTS_I
EXEC_LISTS_I    SET     1
i*
**      SFilename: exec/lists.i $
**      SRelease: 1.3 $
x*
**
xx
xx      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc,.
xx          Ali Rights Reserved
xx

    IFND EXEC_NODES I
    INCLUDE "exec/nödes.i"
    ENDC        ; EXEC_NODES_I


x-------------------------------------------------------------------
x
x   List Structures
x
x-------------------------------------------------------------------

; normál, full featured üst
 STRUCTURE  LH,0
    APTR    LH_HEAD
    APTR    LH_TAIL
    APTR    LH_TAILPRED
    UBYTE   LH_TYPE
    UBYTE   LH_pad
    LABEL   LH_SI2E

; minimai list, no type checking possible
 STRUCTURE  MLH,0
    APTR    MLH_HEAD
    APTR    MLH_TAIL
    APTR    MLH_TAILPRED
    LABEL   MLH_SI2E


NEWLIST     MACRO   * list
            MOVE.L  \1,(\1)
            ADDQ.L  #LH_TAIL,(\1)
            CLR.L   LH_TAIL(\1)
            MOVE.L  \1,(LH_TAIL+LN_PRED)(\1)
            ENDM

TSTLIST     MACRO   * [list]
            IFC     '\1',''
            CMP.L   LH_TAIL+LN_PRED(AO),AO
            ENDC
            IFNC    '\1',''
            CMP.L   LH_TAIL+LN_PREDC\1),\1
            ENDC
            ENDM

5UCC        MACRO   * node,succ
            MOVE.L  (\1),\2
            ENDM

PRED        MACRO   * node,pred
            MOVE.L  LN_PRED(\1),\2
            ENDM
```

```
IFEMPTY     MACRO   * list>label
            CMP.L   LH_TAIL+LN_PRED(\1),\1
            BEQ     \2
            ENDM

IFNOTEMPTY  MACRO   * list>label
            CMP.L   LH_TAIL+LN_PRED(\1),\1
            BNE     \2
            ENDM

TSTNODE     MACRO   * node,next
            MOVE.L  (\1),\2
            TST.L   (\2)
            ENDM

NEXTNODE'   MACRO   * next,current,exit_label (DX,AX,DISP16)
            MOVE.L  \1,\2
            MOVE.L  (\2),\1
            IFC     '\0',"
            BEQ     \3
            ENDC
            IFNC    '\0',"
            BEQ.S   \3
            ENDC
            ENDM

ADDHEAD     MACRO
            MOVE.L  (A0),D0
            MOVE.L  A1,(AO)
            MOVEM.L DO/AO,(A1)
            MOVE.L  DO,AO
            MOVE.L  AljLN.PREDÍAO)
            ENDM

ADDTAIL     MACRO
            LEA     LH_TAILCAO),AO
            MOVE.L  LN_PRED(AO),DO
            MOVE.L  A1,LN_PRED(AO)
            MOVE.L  AO,(A1)
            MOVE.L  DO,LN_PRED(Al)
            MOVE.L  D0,A0
            MOVE.L  A1,(AO)
            ENDM

REMOVE      MACRO
            MOVE.L  (A1),AO
            MOVE.L  LN_PRED(A1),A1
            MOVE.L  A07(A1)
            MOVE.L  A1,LN_PRED(AO)
            ENDM

REMHEAD     MACRO
            MOVE.L  (AO),A1
            MOVE.L  (Al),DO
            BEQ.S   REMHEAD\a
            MOVE.L  DO,(AO)
            EXG.L   DO,A1
            MOVE.L  AO,LN_PRED(A1)
REMHEADX3
            ENDM
```

```
x-------------------------------------------------------------------
x
x  REMHEADQ -- remove-head quickly
x
x       Useful when a scratch register is available> and
x       list is known to contain at least one node.
x
x-------------------------------------------------------------------

REMHEADQ    MACRO   * head,nodejscratchReg
            MOVE.L  (\1),\2
            MOVE.L  (\2),\3
            MOVE.L  \3,(\1)
            MOVE.L  \1,LN_PRED(\3)
            ENDM

REMTAIL     MACRO
            MOVE.L  LH_TAIL+LN_PRED(AO),A1
            MOVE.L  LN_PRED(A1),DO
            BEQ.S   REMTAILXa
            MOVE.L  DO,LH TAIL+LN_PRED(AO)
            EXG.L   DO,A1~
            MOVE.L  AO,(A1)
            ADDQ.L  #4,(A1)
REMTAILX3
            ENDM

        ENDC    ;  EXEC_LISTS_I
```

```
        IFND    EXEC_MEMORY_I
EXEC_MEMORYJ    SET     1
x*
xx      SFilename: exec/memory.i $
xx      $Release: 1.3 $
x*
xx      definitions for use with the memory allocator
xx
xx      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
xx          All Rights Reserved
xx

    IFND        EXEC_NODES_I
    INCLUDE     "exec/nodes.i"
    ENDC        5 EXEC_NODES_I


x-------------------------------------------------------------------------
x
x   Memory List Structures
x
x-------------------------------------------------------------------------
x
x   A memory list appears in two forms:  One is a requirements listx
x   the other is a list of already allocated memory.  The formát is
x   the same, with the reqirements/address field occupying the same
x   position.
x
x   The formát is a linked list of ML structures each of which has
x   an array of ME entries.
x
x-------------------------------------------------------------------------

 STRUCTURE ML,LN_SIZE
    UWORD   ML_NÜMENTRIES           x The number of ME structures that follow
    LABEL   ML_ME                   * where the ME structures begin
    LABEL   ML_SIZE


 STRUCTURE ME,0
    LABEL   ME_REQS                 x the AllocMem requirements
    APTR    ME_ADDR                 * the address of this block (an alias
*                                   *   for the same location as ME_REQS)
    ULONG   ME_LENGTH               * the length of this region
    LABEL   ME_SIZE


*------ memory options:

    BITDEF  MEM,PUBLIC,0
    BITDEF  MEM,CHIP,1
    BITDEF  MEM,FÁST,2
    BITDEF  MEM,CLEAR,16
    BITDEF  MEM,LARGEST,17 /


*------ alignment rules for a memory block:

MEM_BLOCKSIZE   EQU 8
MEM_BLOCKMASK   EQU (MEM_BLOCKSIZE-1)


*-------------------------------------------------------------------------
```

```
*
*    Memory Region Header
*
*-------------------------------------------------------------------------

 STRUCTURE   MH,LN_SIZE
    UWORD    MH_ATTRIBUTES          * characteristics of this region
    APTR     MH_FIRST               * first free region
    APTR     MH_LOWER               * lower memory bound
    APTR     MHJJPPER               * upper memory bound+1
    ULONG    MH_FREE                * number of free bytes
    LABEL    MH_SI2E


X-------------------------------------------------------------------------
X
i    Memory Chunk
*
i -------------------------------------------------------------------------

 STRUCTURE  MC,0
    APTR    MC NEXT                 * ptr to next chunk
    ULONG   MCIBYTES                * chunk byte size
    LABEL   MC-SI2E

        ENDC    ; EXEC_MEMORY_I
```

```
        IFND    EXEC_NODES_I
EXEC_NODES_I,   SET     1
x*
**      SFilename: exec/nodes.i S
**      SRelease: 1.3 $
xx
xx
xx
xx      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
xx          Ali Rights Reserved
xx

x-------------------------------------------------------------------
x
*   List Node Structure
x
x-------------------------------------------------------------------

 STRUCTURE  LN,0
    APTR    LN_SUCC
    APTR    LN"PRED
    UBYTE   LNJTPE
    BYTE    LN_PRI
    APTR    LN_NAME
    LABEL   LN_SIZE

; min node -- only has minimum necessary, no type checking possible
 STRUCTURE  MLN,0
    APTR    MLN_SUCC
    APTR    MLNJPRED
    LABEL   MLN_SIZE

*------ Node Types:

NT_UNKNOWN      EQU     0
NTJTASK         EQU     1
NT_INTERRUPT    EQU     2       ; alsó for software interrupt node
NT_DEVICE       EQU     3
NT_MSGPORT      EQU     4
NTJ4ESSAGE      EQU     5
NT_FREEMSG      EQU     6
NT_REPLYMSG     EQU     7
NT_RESOURCE     EQU     8
NT_LIBRARY      EQU     9
NT_MEMORY       EQU     10
NT_SOFTINT      EQU     11      ; exec priváté
NT_FONT         EQU     12
NT_PROCESS      EQU     13
NT_SEMAPHORE    EQU     14
NT_SIGNALSEM    EQU     15      ; signal semaphores
NT_BOOTNODE     EQU     16

        ENDC    ; EXEC_NODES_I
```

```
        IFND    EXEC_PORTS_I
EXEC_PORTS_I    SET     1
XX
**      ÍFilename: exec/ports.i $
**      SRelease: 1.3 $
XX
XX
XX
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga,  Inc.
**          Ali  Rights Reserved
XX
    IFND EXEC_NODES_I
    INCLUDE "exec/nodes.i"
    ENDC        ; EXECJIODESJ

    IFND EXEC_LISTS_I
    INCLUDE "exec/lists.i"
    ENDC        ; EXECJ.ISTS_I

*------------------------------------------------------------------
X
*   Message Port Structure
*
*------------------------------------------------------------------

 STRUCTURE  MP,LN_SIZE
    UBYTE   MP_FLAGS
    UBYTE   MP~SIGBIT               * signal bit number
    APTR    MP_SIGTASK              * task to be signalled
    STRUCT  MP_MSGLIST,LH_SIZE      * message linked list
    LABEL   MP_SIZE


*_______unions:

MP_SOFTINT      EQU  MP_SIGTASK


x_______flags fields:

PF_ACTION       EQU  3


x_______PutMsg actions:

PA_SIGNAL       EQU  0
PA_SOFTINT      EQU  1
PA_IGNORE       EQU  2


X_____________________-----------------------------------------
X
*   Message Structure
*
*------------------------------------------------------------------

 STRUCTURE  MN,LN_SIZE
    APTR    MN_REPLYPORT            * message reply port
    UWORD   MN_LENGTH               * message len in bytes
    LABEL   MN_SIZE

        ENDC    ; EXEC_PORTS_I
```

```
        IFND    EXEC_RESIDENT_I
EXEC_RESIDENT_I SET     1
S*
**      SFilename: exec/resident.i $
**      SRelease: 1.3 $
XX
XX
IX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

*------------------------------------------------------------------
X
X   Resident Modulé Tag
X
*------------------------------------------------------------------

 STRUCTURE RT,0
    UWORD RT_MATCHWORD                  * word to match
    APTR  RT_MATCHTAG                   * pointer to structure base
    APTR  RTlENDSKIP                    * address to continue scan
    UBYTE RT_FLAGS                      * various tag flags
    UBYTE RT_VERSION                    * release version number
    UBYTE RT_TYPE                       * type of modulé
    BYTE  RT_PRI                        * initialization priority
    APTR  RT_NAME                       * pointer to node narae
    APTR  RT~IDSTRING                   * pointer to id string
    APTR  RT_INIT                       * pointer to init code
    LABEL RT_SIZE


*_______Match word definition:

RTC_MATCHWORD   EQU     S4AFC           * (ILLEGAL instruction)


«_______RT_FLAGS bit and field definitions:

    BITDEF RT,C0LDSTART,0
    BITDEF RT,AUTOINIT,7                * RT_INIT points to data

* Compatibility:
RTM_WHEN        EQU     1               * field position in RT_FLAGS
RTW_NEVER       EQU     0               * never ever init
RTW_COLDSTART   EQU     1               * init at coldstart time

        ENDC    5 EXEC_RESIDENT_I
```

```
        IFND    EXEC SEMAPHORESJ
EXEC_SEMAPHORES_I   ~   SET     1
xx
**      SFilename: exec/semaphores. i $
**      SRelease: 1.3 $
XX
XX
XX
*x      (C) Copyright 1986)1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
XX
    IFND EXEC_NODES_I
    INCLUDE "exec/nodes.i"
    ENDC        j EXEC_NODES_I

    IFND EXEC_LISTS_I
    INCLUDE "exec/lists.i"
    ENDC        5 EXEC_LISTS_I

    IFND EXEC_PORTS_I
    INCLUDE "exec/ports.i"
    ENDC        ; EXEC_PORTS_I

*-------------------------------------------
X
x   Semaphore Structure
X
X-------------------------------------------

 STRUCTURE  SM,MP_SIZE
    WORD    SM_BIDS          x number of bids for lock
    LABEL   SM-SIZE

*_____unions:

SM_LOCKMSG    EQU  MP_SIGTASK


*-------------------------------------------------------------------s
X
*   Signal Semaphore Structure
X
X-------------------------------------------------------------
* this is the structure used to request a signal semaphore -- allocated
* on the fly by ObtainSemaphoreC)
 STRUCTURE  SSR,MLN_SI2E
    APTR    SSR_WAITER
    LABEL   SSR_SIZE


* this is the actual semaphore itself -- allocated statically
 STRUCTURE  SS,LN_SIZE
    SHORT   SS_NESTCOUNT
    STRUCT  SS_WAITQUEUE,MLH_SIZE
    STRUCT  SS_MULTIPLELINK,SSR_SIZE
    APTR    SS_OWNER
    SHORT   SS_QUEUECOUNT
    LABEL   SS_SIZE
```

```
        ENDC    ; EXEC_5EMAPH0RES_I
```

```
        IFND    EXEC_STRINGS_I
EXEC_STRINGS_I  SET     1
XX
XX      SFilename: exec/strings.i $
X       SRelease: 1.3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

*_______Terminál Control:

EOS             EQU     0
BELL            EQU     7
LF              EQU     10
CR              EQU     13
BS              EQU     8
DEL             EQU     $7F
NL              EQU     LF


X---------------------------------------------------------------------
X
X   String Support Macros
X
X---------------------------------------------------------------------

STRING      MACRO
            DC.B    \1
            DC.B    0
            CNOP    0,2
            ENDM


STRINGL     MACRO
            DC.B    13,10
            DC.B    \1
            DC.B    0
            CNOP    0,2
            ENDM


STRINGR     MACRO
            DC.B    \1
            DC.B    13,10,0
            CNOP    0,2
            ENDM


STRINGLR    MACRO
            DC.B    13,10
            DC.B    \1
            DC.B    13,10,0
            CNOP    0,2
            ENDM

        ENDC    ; EXEC_STRINGS_I
```

```
        IFND    EXEC_TASKS_I
EXEC_TASKS_I    SET   1
x*
xx      SFilename: exec/tasks.i $
xx      SRelease: 1.3 $
xx
KK
xx
xx      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
xx          Ali Rights Reserved
xx

    IFND EXEC_NODES_I
    INCLUDE "exec/nodes.i"
    ENDC        5 EXEC_NODES_I

    IFND EXEC_LISTS_I
    INCLUDE "exec/lists.i"
    ENDC        ; EXEC_LISTS_I


x-------------------------------------------------------------------
x
x   Task Control Structure
x
x-------------------------------------------------------------------

 STRUCTURE  TC,LN_SIZE
    UBYTE   TC FLAGS
    UBYTE   TC STATE
    BYTE    TC_IDNESTCNT            * intr disabled nesting
    BYTE    TCJTDNESTCNT            * task disabled nesting
    ULONG   TC_SIGALLOC             x sigs allocated
    ULONG   TC_SIGWAIT              * sigs we are waiting for
    ULONG   TC_SIGRECVD             * sigs we have réceived
    ULONG   TC_SIGEXCEPT            x sigs we take as exceptions
    UWORD   TC_TRAPALLOC            * traps allocated
    UWORD   TC_TRAPABLE             * traps enabled
    APTR    TC_EXCEPTDATA           x data for except proc
    APTR    TC_EXCEPTCODE           * exception procedure
    APTR    TCJTRAPDATA             * data for proc trap proc
    APTR    TC_TRAPCODE             * proc trap procedure
    APTR    TC_SPREG                * stack pointer
    APTR    TC_SPLOWER              * stack lower bound
    APTR    TC_SPUPPER              * stack upper bound + 2
    APTR    TC_SWITCH               * task 1osing CPU
    APTR    TC_LAUNCH               x task getting CPU
    STRUCT  TCJ4EMENTRY,LH_SIZE     x allocated memory
    APTR    TC_Userdata
    LABEL   TC_SIZE


*------- Flag Bits:

    BITDEF  T,PROCTIME,0
    BITDEF  T,STACKCHK,4
    BITDEF  T,EXCEPT,5
    BITDEF  T,SWITCH,6
    BITDEF  T,LAUNCH,7


x_____-_ Task States:
TS_INVALID  EQU     0
```

```
TS_ADDED    EQU     TS_INVALID+1
TS_RUN      EQU     TS_ADDED+1
TS_READY    EQU     TS_RUN+1
TS_WAIT     EQU     TS_READY+1
TS_EXCEPT   EQU     TS_WAIT+1
TS_REMOVED  EQU     TS_EXCEPT+1


*       System Task Signals:

SIGF_ABORT      EQU     $0001
SIGF_CHILD      EQU     $0002
SIGF_BLIT       EQU     $0010
SIGF_SINGLE     EQU     $0010
SIGF_DOS        EQU     $0100

SIGB_ABORT      EQU     0
SIGB_CHILD      EQU     1
SIGB_BLIT       EQU     4
SIGB_SINGLE     EQU     4
SIGB_DOS        EQU     8


SYS_SIGALLOC    EQU     $OFFFF          > pre-allocated signals
SYSJTRAPALLOC   EQU     $08000          ; pre-allocated traps

        ENDC    ; EXEC_TASKS_I
```

```
        IFND    EXEC_TYPES_I
EXEC_TYPES_I    SET     1
XX
**      SFilename: exec/types.i $
**      SRelease: 11.3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          All Rights Reserved
XX

EXTERN_J.IB MACRO
            XREF    _LVO\1
            ENDM

STRUCTURE   MACRO
\1          EQU     0                   * far assembler's sake
SOFFSET     SET     \2
            ENDM

BOOL        MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+2
            ENDM

BYTE        MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+1
            ENDM

UBYTE       MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+1
            ENDM

WORD        MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+2
            ENDM

UWORD       MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+2
            ENDM

SHORT        MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+2
            ENDM

USHORT      MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+2
            ENDM

LONG        MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+4
            ENDM

ULONG       MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+4
```

```
            ENDM

FLOAT       MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+4
            ENDM

APTR        MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+4
            ENDM

CPTR        MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+4
            ENDM

RPTR        MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+2
            ENDM

STRUCT      MACRO
\1          EQU     SOFFSET
SOFFSET     SET     SOFFSET+X2
            ENDM

LABEL       MACRO
\1          EQU     SOFFSET
            ENDM

*------ bit definition macro ---------------------------------
*
*   Given:
*
*       BITDEF  MEM>CLEAR,16
*
*  Yields:
*
*       MEMB_CLEAR  EQU 16
*       MEMF_CLEAR  EQU (1.SL.MEMB_CLEAR)
*

BITDEF      MACRO   * prefix>&name>&bitnum
            BITDEFO \1,\2,B_,\3
\3BITDEF    SET     1<<\3
            BITDEFO \1\2>F_,\3BITDEF
            ENDM

BITDEFO     MACRO   * prefix>&name>&type,&value
\1\3\2      EQU     \4
            ENDM

LIBRARY_VERSION EQU     34

        ENDC    J EXEC_TYPES_I
```

```
Directory "Lattice C_5.0.5:Asserabler_Headers/graphics" on Saturday 29-Sep-90
clip.i              1706_____rwed 07-Nov-88 14:57:43
copper.i            1975_____rwed 07-Nov-88 14:57:44
display.i           1071_____rwed 07-Nov-88 14:57:44
gels.i              8905_____rwed 07-Nov-88 14:57:45
gfx.i                600_____rwed 07-Nov-88 14:57:45
gfxbase.i           2001_____rwed 07-Nov-88 14:57:43
layers.i            1112_____rwed 07-Nov-88 14:57:45
rastport.i          2958_____rwed 07-Nov-88 14:57:44
regions.i            539_____rwed 07-Nov-88 14:57:43
sprite.i             402_____rwed 07-Nov-88 14:57:44
text.i              2670_____rwed 07-Nov-88 14:57:42
view.i              1403_____rwed 07-Nov-88 14:57:44
12 files - 72 blocks - 25342 bytes
```

```
        IFND    GRAPHICS_CLIP_I
GRAPHICS_CLIP_I SET     1
**
**      SFilename: graphics/clip.i $
**      SRelease: 1.3 $
xx
xx
xx
xx      (C) Copyright 1985,1986,1987)1988 Commodore-Amiga,  Inc,
x*          Ali Rights Reserved
xx

    IFND    GRAPHICS_GFX_I
    include "graphics/gfx.i"
    ENDC
        IFND    EXEC_SEMAPHORES_I
        include "exec/semaphores.i"
        ENDC

NEWLOCKS        equ     1

 STRUCTURE  Layer,0
    LONG    lr_front
    LONG    lr_back
    LONG    lr_ClipRect
    LONG    Ír_rp
    WORD    Ír_MinX
    WORD    Ír_MinY
    WORD    Ír_MaxX
    WORD    Ír_MaxY       ;
        STRUCT  lr_reserved,4
        WORD    lr_priority
    WORD    lr_Flags
    LONG    1r_SuperBi tMap
    LONG    lr_SuperClipRect
    APTR    Ír_Window
    WORD    Ír_ScrollJC
    WORD    lr_Scroll_Y
    APTR    Ír_cr
    APTR    Ir_cr2
    APTR    lr_crnew
    APTR    1r_SuperSaverCli pRects
    APTR    Ír__cliprects
    APTR    lr_LayerInfí)
x                       just by lucky coincidence
x                       this is not confused with simplesprites
        STRUCT  lr_Lock,SS_SIZE
        STRUCT  Ir_reserved3,8
        APTR    lr_ClipRegion
        APTR    Ír_saveClipRects
        STRUCT  Ír_reserved2,22
    APTR    Ír_DamageList
    LABEL   lr_SIZEOF

 STRUCTURE  ClipRect,0
    LONG    cr_Next
    LONG    cr_prev
    LONG    cr_lobs
    LONG    cr_BitMap
    WORD    cr_MinX
    WORD    cr_MinY
    WORD    cr_MaxX
    WORD    cr_MaxY
```

```
    APTR    cr__pl
    APTR    cr_p2
    LONG    cr_reserved
    LONG    cr_Flags
 LABEL      cr_SIZEOF

* internál cliprect flags
CR_NEEDS_NO_CONCEALED_RASTERS   equ     1

* defines for clipping
ISLESSX equ 1
ISLESSY equ 2
ISGRTRX equ 4
ISGRTRY equ 8

* for ancient history reasons
        IFND    lr_Front
lr_Front        equ lr_front
lr_Back         equ lr_back
lr_RastPort     equ     lr_rp
cr_Prev     •   equ cr_prev
cr_LObs         equ     cr_lobs
        ENDC

        ENDC    j GRAPHICS_CLIP_I
```

```
        IFND    GRAPHICS_DISPLAY_I
GRAPHICS_DISPLAY_I      SET     í
XX
**      SFilename: graphics/display.i $
**      SRelease: 1.3 £
XX
**      include define file for display control registers
XX
**      (C) Copyright 1985>1986>1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
XX

* bplconO defines
M0DE_640    equ   $8000
PLNCNTMSK   equ   $7              * how many bit planes?
*                                 x 0 = none> l->6 = l->6} 7 = reserved
PLNCNTSHFT  equ   12              * bits to shift for bplconO
PF2PRI      equ   $40             * bplcon2 bit
COLORON     equ   $0200           * disable color burst
DBLPF       equ   $400
HOLDNMODIFY equ   $800
INTERLACE   equ   4               * interlace raode for 400

* bplconl defines
PFA_FINE_SCROLL       equ   $F
PFB_FINE_SCROLL_SHIFT equ   4
PF_FINE_SCROLL_MASK   equ   $F

* display window start and stop defines
DIW_HORIZ_POS         equ   $7F     * horizontal start/stop
DIW_VRTCL_POS         equ   '$1FF   * verticai start/stop
DIW_VRTCL_POS_SHIFT   equ   7

* Data fetch start/stop horizontal position
DFTCH_MASK            equ   $FF

* vposr bits
VPOSRLOF              equ   $8000

        ENDC    > GRAPHICS_DISPLAY_I
```

```
        IFND    GRAPHICS_GELS_I
GRAPHICS_GELS_I SET     1
**
**      JFilenarae: graphics/gels.i S
**      SRelease: 1.3 $
SX
**      include file for AMIGA GELS (Graphics Elements)
X*
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga} Inc.
**          Ali Rights Reserved
KX

*------ VS_vSflags ------------------------------------------------

*    ;-- user-set vSprite flags --
SUSERFLAGS  EQU $OOFF         ; mask of all user-settable vSprite-flags
    BITDEF  VS,VSPRITE,0      ; set if vSprite, clear if bob
    BITDEF  VS,SAVEBACK,1     ; set if background is to be saved/restored
    BITDEF  VS,0VERLAY,2      ; set to mask image of bob ontó background
    BITDEF  VS,MUSTDRAW,3     ; set if vSprite absolutely must be drawn
*    >-- system-set vSprite flags --
    BITDEF  VS,BACKSAVED>8    ; this bob's background has been saved
    BITDEF  VS,B0BUPDATE,9    ; temporary flag, useless to outside world
    BITDEF  VS,GELGONE,10     ; set if gel is completely clipped (offscreen)
    BITDEF  VS,VSOVERFLOw",ll  ; vSprite overflow (if MUSTDRAW set we draw!)


X_______B_f lags _____________________________________________---
*    ;-- these are the user flag bits --
BUSERFLAGS  EQU $OOFF        j mask of all user-settable bob-flags
    BITDEF  B,SAVEBOB,0      ; set to not erase bob
    BITDEF  B,BOBISCOMP,1    ; set to identify bob as animComp
*    ;-- these are the system flag bits --
    BITDEF   B,BWAITING,8     ', set while bob is waiting on 'after[1]
    BITDEF   B,BDRAWN,9       ; set when bob is drawn this DrawG pass
    BITDEF   B,B0BSAWAY,10    ; set to initiate removal of bob
    BITDEF   B,BOBNIX,11      ; set when bob is completely removed
    BITDEF   B,SAVEPRESERVE,12 ; for back-restore during double-buffer
    BITDEF   B,0UTSTEP,13     ; for double-clearing if double-buffer


*------ defines for the animation procedures ----------------------

ANFRACSIZE  EQU 6
ANIMHALF    EQU $0020
RINGTRIGGER EQU $0001

X_______macros -•_____________________________________________
* these are GEL functions that are currently simple enough to exist as a
* definition.  It should not be assumed that this will always be the case

InitAnimate MACR0   * SanimKey
       CLR.L   \1
       ENDM

RemBob      MACRO   * &b
       0R.W    #BF_B0BSAWAY,b_BobFlags+\l
       ENDM

X_______VS : vSprite_________.----------------------------------------
 STRUCTURE  VS,0    ; vSprite
*    -- SYSTEM VARIABLES --
```

```
    APTR    bob_Before          ; struct *bob: draw this bob before bob pointed
                                ; to by before
    APTR    bob_After           ; struct *bob: draw this bob after bob pointed
                                ; to by after
    APTR    bob_BabVSprite      ; struct svSprite; this bob's VSprite definition
    APTR    bob_BobComp         ; struct *animComp: pointer to this bob's
                                ; animComp def
    APTR    bob_DBuffer         ; struct dBufPacket: pointer to this bob's
                                ; dBuf packet
    LABEL   bob_BUserExt        ; bob user extension
    LABEL   bob_SIZEOF

*------ AC ; animComp ----------------------------------------------------

 STRUCTURE  AC>0    ; animComp
*    -- COMMON VARIABLES --
    WORD    ac_CompFlags        ; animComp flags for system & user
*     timer defines how long to keep this component active:
*       if set non-zero> timer decrements to zero then switches to nextSeq
*       if set to zero, animComp never switches
    WORD    ac_Timer
*    -- USER VARIABLES --
*     initial value for timer when the animComp is activated by the system
    WORD    ac_TimeSet
*     pointer to next and previous components of animation object
    APTR    ac_NextComp         i struct *animComp
    APTR    ac_PrevComp         ', struct *animComp
*     pointer to component component definition of next image in sequence
    APTR    ac_NextSeq          \ struct *animComp
    APTR    ac._PrevSeq         ', struct *animComp
    APTR    ac._AnimCRoutine    ; address of special animation procedure
    WORD    ac._YTrans          ; initial y translation (if this is a component)
    WORD    ac._XTrans          ; initial x translation (if this is a component)
    APTR    ac."HeadOb          ; struct *animOb
    APTR    ac._AnimBob         ; struct *bob
    LABEL   ac.IS IZE

*------ AO ; animOb ------------------------------------------------------

 STRUCTURE  AO:,0   ; animOb
*    -- SYSTEM VARIABLES --
    APTR    ao._NextOb          ; struct *anim0b
    APTR    ao_Prev0b           ; struct *anim0b
*     nuraber of" calls to Animate this animOb has endured
    LONG    ao._Clock
    WORD    ao._AnOldY          ; old y,x coordinates
    WORD    ao_AnOldX           ;
*    -- COMMON VARIABLES --
    WORD    ao..AnY             j y,x coordinates of the animOb
    WORD    ao_AnX              ;
*    -- USER VARIABLES --
    WORD    ao_.YVel            ; velocities of this object
    WORD    ao..XVel            ;
    WORD    ao._XAccel          ; accelerations of this object
    WORD    ao._YAccel          ;   !!! backwards !!!
    WORD    ao_RingYTrans       ; ring translation values
    WORD    ao_RingXTrans       ;
    APTR    ao_AniniORoutine    ; address of special animation procedure
    APTR    ao_HeadComp         ; struct *animComp: pointer to first component
    LABEL   ao._AUserExt        ; animOb user extension
    LABEL   ao."SIZEOF
```

```
*------DBP ; dBufPacket ----------------------------------------------
* dBufPacket defines the values needed to be saved across buffer to buffer
*   when irt double-buffer mode

 STRUCTURE  DBP,0             ; dBufPacket
    WORD    dbp_BufY          > savé the other buffers screen coordinates
    WORD    dbp_BufX          ;
    APTR    dbp_BufPath       > struct *vSprite: carry the draw path over
                               ; the gap
*    these pointers must be fiiled in by the user          v
*    pointer to other buffer's background savé buffer
    APTR    dbp_BufBuffer     •> *W0RD
*    pointer to other buffer's background pláne pointers
    APTR    dbp_BufPlanes     ; *«WORD
    LABEL   dbp_SIZEOF

        ENDC    ; GRAPHICS_GELS_I
```

```
        IFND    GRAPHICS_GFX_I
GRAPHICS_GFX_I  SET     T
XX
**      SFilename: graphics/gfx.i $
**      SRelease: 1.3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

BITSET      equ $8000
BITCLR      equ 0
AGNUS       equ 1
DENISE      equ 1

    STRUCTURE   BitMap,0
    WORD    bra_BytesPerRow
    WORD    bm_Rows
    BYTE    bm_Flags
    BYTE    bm_Depth
    WORD    bm_Pad
    STRUCT  bm_Planes,8*4
    LABEL   bm_SIZEOF

   STRUCTURE   Rectangle,0
      WORD  ra_MinX
      WORD  raJlinY
      WORD  ra_MaxX
      WORD  ra_MaxY
   LABEL    ra^SIZEOF

        ENDC    ; GRAPHICS_GFX_I
```

```
        IFND    GRAPHICS_GFXBASE_I
GRAPHICS_GFXBASE_I      SET     1
XX
XX      SFilename: graphics/gfxbase.i $
**      SRelease: 1.3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

    IFND    EXEC_LISTS_I
    include "exec/lists.i"
    ENDC
    IFND    EXEC_LIBRARIES_I
    include "exec/libraries.i"
    ENDC
    IFND    EXEC_INTERRUPTS_I
    include "exec/interrupts.i"
    ENDC

 STRUCTURE  GfxBase,LIB_SIZE
    APTR    gb_ActiView     ; struct *View
    APTR    gb_copinit      ; struct xcopinitj ptr to copper start up list
    APTR    gb_cia      ; for 6526 resource use
    APTR    gb_blitter      ; for blitter resource use
    APTR    gb_LOFlist      ", current copper list being run
    APTR    gb_SHFlist      ; cürrent copper list being run
    APTR    gbjalthd        i struct *bltnode
    APTR    gb_blttl        ;
    APTR    gblbsblthd      ;
    APTR    gb_bsblttl      ;
    STRUCT  gb_vbsrv,IS_SIZE
    STRUCT  gb_timsrv,IS_SIZE
    STRUCT  gb_bltsrv,IS_SIZE
    STRUCT  gb_TextFonts,LH_SIZE
    APTR    gbJDefaultFont
    UWORD   gb_Modes        ; copy of bltconO
    BYTE    gb_VBlank
    BYTE    gb_Debug
    UWORD   gb_BeamSync
    WORD    gb_system_bplconO
    BYTE    gb_SpriteReserved
    BYTE    gb_bytereserved

    WORD    gbJFlags
    WORD    gb_BlitLock'
        WORD    gb_BlitNest
        STRUCT  gb_BlitWaitQ,LH_SIZE
        APTR    gb_BlitOwner
        STRUCT  gb_TOF_WaitQ,LH_SIZE

        WORD    gb_DisplayFlags
        APTR    gb_SimpleSprites
        WORD    gb_MaxDisplayRow
        WORD    gb_MaxDisplayColumn
        WORD    gb_NormalDisplayRows
        WORD    gbJtormalDisplayColumns
        WORD    gb_NormalDPMX
        WORD    gb_NormalDPMY

        APTR    gb_LastChanceMemory
```

```
        APTR    gbJLCMptr

        WORD    gb_MicrosPerLine        > usecs per line times 256
        WORD    gb_MinDisplayColumn

    STRUCT  gb_reserved,92   ; bytes reserved for future use
    LABEL   gb_SIZE

* bits for dalestuff> which may go away when blitter becomes a resource
OWNBLITTERn equ 0   * blitter owned bit
QBOWNERn    equ 1   * blitter owned by blit queuer

QBOWNER     equ l<<QBOWNERn

        ENDC    ; GRAPHIC5_GFXBASE_I
```

```
        IFND    GRAPHICS_LAYERS_I
GRAPHICS_LAYERS_I       SET     1
**
**      SFilename: graphics/layers.i $
**      SRelease: 1.3 S
**
**
x*
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
xx

    IFND    EXEC_SEMAPHORES_I
       include "exec/semaphores.i"
    ENDC

    IFND  EXEC_LISTS_I
       include  "exec/lists.i"
    ENDC

* these should be clip.i/h but you know backwards compatibility etc.
LAYERSIMPLE             equ   1
LAYERSMART              equ   2
LAYERSUPER              equ   4
LAYERUPDATING           equ   $10
LAYERBACKDROP           equ   $40
LAYERREFRESH            equ   $80
LAYER_CLIPRECTS_LOST    equ   $100

LMN_REGION  equ -1

    STRUCTURE  Layer_lnfo,0
    APTR       li_top_layer
    APTR       li_check_lp
    APTR       li_obs
    STRUCT     li_FreeClipRects,MLH_SIZE
    STRUCT     li_Lock,SS_SIZE
        STRUCT     li_gs_Head,LH_SIZE
        LONG           li_long_reserved
        WORD       li_Flags
    BYTE       1i_fatten_count
        BYTE       1i_LockLayersCount
    WORD       li_Layerlnfo_extra_size
        APTR           li_blitbuff
    APTR       li_Layerlnfo_extra
    LABEL      li_SIZEOF

NEWLAYERINFO_CALLÉD     equ 1
ALERTLAYERSNOMEM        equ $83010000

        ENDC    ; GRAPHICS_LAYERS_I
```

```
    LONG    rp_BitMap
    LONG    rp_AreaPtrn
    LONG    rp_TmpRas
    LONG    rp_AreaInfo
    LONG    rp_GelsInfo
    BYTE    rp_Mask
    BYTE    rp_FgPen
    BYTE    rp_BgPen
    BYTE    rp_A0LPen
    BYTE    rp_DrawMode
    BYTE    rp_AreaPtSz
    BYTE    rp_Dummy
    BYTE    rp_linpatcnt
    WORD    rp_Flags
    WORD    rp_LinePtrn
    WORD    rp_cp_x
    WORD    rp_cp_y
    STRUCT  rp_mintermsj8
    WORD •  rp_PenWidth
    WORD    rp_PenHeight
    LONG    rp_Font
    BYTE    rp_AlgoStyle
    BYTE    rp_TxFlags
    WORD    rp_TxHeight
    WORD    rpJTxWidth
    WORD    rp_TxBaseline
    WORD    rp_TxSpacing
    APTR    rp_RP_User
    STRUCT  rp_longreserved)8
        ifnd    GFX_RASTPORT_1_2  .
    STRUCT • rp_wordreserved,14
    STRUCT  rp_reserved»8
        endc
    LABEL   rp_SIZEOF

 STRUCTURE  ArealnfOjO
    LONG    ai_VctrTbl
    LONG    ai_VctrPtr
    LONG    ai_FlagTbl
    LONG    ai_FlagPtr
    WORD    ai_Count
    WORD    ai_MaxCount
    WORD    ai_FirstX
    WORD    ai_FirstY
    LABEL   ai_SIZEOF

ONE_DOTn    equ     1
ONE_DOT     equ     $2      * l<<ONE_DOTn
FRST_DOTn   equ     0
FRST_DOT    equ     1       * l<<FRST_DOTn

        ENDC    ; GRAPHICS_RASTPORT_I
```

```
        IFND    GRAPHICS_REGIONS_I
GRAPHICS_REGIONS_I      SET     T
XX
XX      SFilename: graphics/regions.i S
XX      SRelease: 1,3 $
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Araiga,  Inc.
XX          Ali Rights Reserved
XX

   IFND  GRAPHICS_GFX_I
   include  "graphics/gfx.i"
   ENDC

    STRUCTURE   Region,0
      STRUCT   rg_bounds,ra_SIZEÓF
      APTR  rg_RegionRectangle
   LABEL    rg_SIZEOF

   STRUCTURE   RegionRectangle,0
      APTR  rr_Next
      APTR  rr_Prev
      STRUCT   rr_bounds,ra_SIZEOF
   LABEL    rr^SIZEOF

        ENDC    ; GRAPHICS_REGIONS_I
```

```
        IFND    GRAPHICS_SPRITE_I
GRAPHICS_SPRITE_I       SET     1
**
**      SFilename: graphics/sprite.i $
**      SRelease: 1.3 $                                         i
XX
XX
XX
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          All Rights Reserved
XX
   STRUCTURE    SimpleSprite,0
   APTR         ssjposctldata
   WORD  '      ss_height
   WORD         ss_x
   WORD         ss_y
   WORD         ss_num
   LABEL        ss_SIZEOF

        ENDC    ; GRAPHICS_SPRITE_I
```

```
        IFND    GRAPHICS_TEXT_I
GRAPHICS_TEXT_I SET     1
XX
XX      SFilename: graphics/text.i $>,
XX      SRelease: 1.3 $
XX
XX      graphics library text structures
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

    IFND     EXEC_PORT5_I
    INCLUDE  "exec/ports.i"
    ENDC

*------ Font Styles ----------------------------------------------------
FS_NORMAL   EQU 0               ;normal text (no style attributesset)
    BITDEF   FS,EXTENDED,3      jextended face (must be designed)
    BITDEF   FS,ITALIC,2        jitalic (slanted 1:2 right)
   •BITDEF   FS,BOLD,1          Sbold face text (ORed w/ shifted right 1)
    BITDEF   FS,UNDERLINED,0    junderlined (under baseline)

*------ Font Flags -----------------------------------------------------
    BITDEF   FP,ROMFONT,0       ;font is in rom
    BITDEF   FP,DISKFONT,1      »font is from diskfont.library
    BITDEF   FP,REVPATH,2       ;designed path is reversed (e.g. left)
    BITDEF   FP,TALLDOT,3       Sdesigned for hires non-interlaced
    BITDEF   FP,WIDED0T,4       ;designed for lores interlaced
    BITDEF   FP,PR0P0RTI0NALs5  ;character sizes can vary from riominal
    BITDEF   FP,DESIGNED,6      ;size is "designed", not constructed
    BITDEF   FP,REMOVED,7       ; the font has been removed


xxxxxxx TextAttr node *********************************************
 STRUCTURE  TextAttr,0
    APTR     ta_Name            ',name of the desired font
    UWORD    ta_YSize           Jsize of the desired font
    UBYTE    ta_Style           ;desired font style
    UBYTE    ta_Flags           Jfont preferences
    LABEL    ta_SIZEOF


xxxxxxx TextFont node *********************************************
 STRUCTURE Te xtFont,MN_SIZE
*                               jfont name in LN        \ used in this
    UWORD    tf_YSize           jfont height             ! order to best
    UBYTE    tflstyle           jfont style              í match a font
    UBYTE    tf_Flags           jpreference attributes   / request.
    UWORD    tf_XSize           jnominal font width
    UWORD    tf_Baseline        jdistance from the top of char to baseline
                                ;smear to affect a bold enhancement
    UWORD    tf_BoldSmear
                                jaccess count
    UWORD    tf_Accessors
                                jthe first character described here
    UBYTE    tf_LoChar          jthe last characte,r described here
    UBYTE    tf_HiChar          jthe bit character data
    APTR     tf_CharData
    UWORD    tf_Modulo          Jthe row modulo for the strike font data
    APTR     tf~CharLoc         5ptr to location data for the strike font
```

```
Directory "Lattice_C_5.0.5:Assembler_Headers/resources" on Saturday 29-Sep-90
cia.i                          318 ____rwed 0?-Nov-88 14:58:10
ciabase.i                      758 ____rwed 07-Nov-88 14:58:10
disk.i                        2521 ____rwed 07-Nov-88 14:58:10
filesysres.i                  1746 ____rwed 07-Nov-88 14:58:10
mathresource.i                1445 ____rwed 07-Nov-88 14:58:10
misei                          900 ____rwed 07-Nov-88 14:58:10
potgo.i                        317 ____rwed 07-Nov-88 14:58:09
7 files - 26 blocks - 8005 bytes
```

```
          IFND     RFSOURCES C1A J
RESOURCES_CIA_I SET       l"
xx
**      SFilename: resources/cia.i $
**      SRelease: 1.3$
XX
XX
XX
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
XX
CIAANAME        MACRO
                DC.B    'ciaa.resource'}0
                ENDM

CIABNAME        MACRO
                DC.B    'ciab.resource[1],0
                ENDM

        ENDC ; RESOURCES Cl A _I
```

```
*****************************************************************************
*               Commodore-Amiga, Inc.                                       *
*               ciabase.i                                                   *
*****************************************************************************

*----------------------------------------------------------------------
X
X  CIA Resource Data Definition
*
X----------------------------------------------------------------------


 STRUCTURE  CIAR,LIB_SIZE
    APTR    CR_HWADDR
    UWORD   CR_IntMask
    UBYTE   CR_IEnable
    UBYTE   CR_IActive
    STRUCT  CR_INTNODE,IS_SIZE
    STRUCT  CR_IVTA,IV_SIZE
    STRUCT  CR_IVTB,IV_SIZE
    STRUCT  CR_IVALRM,IV_SIZE
    STRUCT  CR_IVSP,IV_SIZE
    STRUCT  CR_IVFLG,IV_SIZE
    LABEL   CR_SIZE
```

```
        IFND    RESOURCES_DISK_I
RESOURCES_DISK_I        SET     1
XX
**      SFilename: resources/disk.i $
** ^    SRelease: 1.3 S
XX
**      external declarations for disc resources
XX
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga,  Inc.
**          Ali Rights Reserved
XX

        IFND    EXEC_TYPES_I
        INCLUDE "exec/types.i"
        ENDC    ; EXECJTPESJ

        IFND    EXEC_LIST5_I
        INCLUDE "exec/lists.i"
        ENDC    ; EXEC_LISTS_I

        IFND    EXEC_PORTS_I
        INCLUDE "exec/ports.i"
        ENDC    ; EXEC_PORTS_I

        IFND    EXEC_INTERRUPTS_I
        INCLUDE "exec/interrupts.i"
        ENDC    ; EXEC_INTERRUPTS_I

        IFND    EXEC_LIBRARIES_I
        INCLUDE "exec/libraries.i"
        ENDC    ; EXEC_LIBRARIES_I


*****************************************************************************

* Resource structures
*
*****************************************************************************

    STRUCTURE DISCRESOURCEUNIT,MN_SIZE
        STRUCT  DRU_DISCBLOCK,IS_SIZE
        STRUCT  DRU_DISCSYNC,IS_SIZE
        STRUCT  DRU_INDEX,IS_SIZE
        LABEL   DRU_SIZE



    STRUCTURE DISCRESOURCE,LIB_SIZE
        APTR    DR_CURRENT      ; pointer to current unit structure
        UBYTE   DR_FLAGS
        UBYTE   DR_pad
        APTR    DR_SYSLIB
        APTR    DR_CIARESOURCE
        STRUCT  DR_UNITID,4*4
        STRUCT  DR_WAITING,LH_SIZE
        STRUCT  DR_DISCBLOCK,IS_SIZE
        STRUCT  DR_DISCSYNC,IS_SIZE
        STRUCT  DR_INDEX,IS_SIZE
        LABEL   DR_SIZE

        BITDEF  DR,ALL0C0,0     ; unit zero is allocated
        BITDEF  DR,ALLOC1,1     ; unit one is allocated
        BITDEF  DR,ALLOC2,2     ; unit two is allocated
```

```
        BITDEF  DR,ALL0C3,3     ; unit three is allocated
        BITDEF  DR>ACTIVE,7     >' is the disc currently busy?


xxxxxxxxxxxxxxxxxx*xxxxxx*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
x
* Hardware Magic
*
*****************************************************************************


DSKDMAOFF       EQU     $4000   •> idle command for dsklen register


*****************************************************************************
x
x Resource spécific commands
x
*****************************************************************************

*-- DR_NAME is a generic macro to get the name of the resource.  This
*-- way if the name is ever changed you will pick up the change
x-- automatically.
x--
x-- Normál usage would be:
x--
*-- internálName:       DISKNAME
x--

DISKNAME:       MACRO
                DC.B    'disk.resource' ,0
                DS,.W   0
                ENDM

        LIBINIT LIB_BASE
        LIBDEF  DR_ALLOCUNIT
        LIBDEF  DR_FREEUNIT
        LIBDEF  DR_GETUNIT
        LIBDEF  DR_GIVEUNIT
        LIBDEF  DR..GETUNITID

DR_LASTCOMM     EQlJ    DR_GIVEUNIT


XXXXXXXXX*XXXX*«XKXXXXXXXXXX*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*XX
*
* drive types
x
XXXXXXXXXXXKXXXXKXXXXXXXXXXXXXXXXXXXXXXXXXXXXKXXXXXXXXXXXXKXXXXXXXXXXKX

DRT_AMIGA       EQU     $00000000
DRT_37422D2S    EQU     $55555555
DRT_EMPTY       EQU     $FFFFFFFF

        ENDC    ', RESOURCES_DISK_I
```

```
        IFND    RESOURCES_FILESYSRES_I
RESOURCES_FILESYSRES_I  SET     1
**
xx      $Filename: resources/filesysres.i $
xx      JRevision: 1.0$
xx      $Date: 88/07/11 15:32:39 $
xx
xx      FileSystem.resource deseription
xx
xx      (C) Copyright 1988 Commodore-Amiga^ Inc,
xx          Ali Rights Reserved
xx

        IFND    EXEC_NODES_I
        INCLUDE "exec/nodes.i"
        ENDC
        IFND    EXEC_LISTS_I
        INCLUDE "exec/lists.i"
        ENDC
        IFND    LIBRARIES_DOS_I
        INCLUDE "libraries/dos.i"
        ENDC

FSRNAME MACRO
                dc.b ' FileSystem. resource^ 0
        ENDM

 STRUCTURE  FileSysResource,LN_SIZE     ; on resource list
    CPTR    fsr_Creator                 ; name of creator of this resource
    STRUCT  fsr_FileSysEntries,LH_SIZE  ; list of FileSysEntry structs
    LABEL   FileSysResource_SIZEOF

 STRUCTURE  FileSysEntry,LN_SIZE        ; on fsr_FileSysEntries list
                                        ; LN_NAME is of creator of this entry
    ULONG   fse_DosType         ; DosType of this FileSys
    ULONG   fse_Version         ; Version of this FileSys
    ULONG   fse_PatchFlags      ; bits set for those of the following that need
                                ;   to be substituted intő a standard device
                                ;   node for this file system: e.g. $180
                                ;   for substitute SegList & GlobalVec
    ULONG   fse_Type            ; device node type: zero
    CPTR    fse_Task            ; standard dos "task" field
    BPTR    fse_Lock            ; not used for devices: zero
    BSTR    fse_Handler         ; filename to loadseg (if SegList is null)
    ULONG   fse_StackSize       ; stacksize to use when starting task
    LONG    fse_Priority        ; task priority when starting task
    BPTR    fse_Startup         ; startup msg: FileSysStartupMsg for disks
    BPTR    fse_SegList         ; code to run to start new task
    BPTR    fse_GlobalVec       ; BCPL global vector when starting task
    ; no more entries need exist than those implied by fse_PatchFlags

        ENDC    ; RESOURCES_FILESYSRES_I
```

```
            IFND    RESOURCES_MATHRESOURCE_I
RESOURCES_MATHRESOURCE_I "      SET     I
XX
xx      SFilename: resources/mathresource.i $
x*      iRelease: 1.3 $
XX
XX
XX
xx      (C) Copyright 1987,1988 Commodore-Amiga> Inc.
xx          Ali Rights Reserved
xx


        IFND    EXEC_TYPES_I
        include "exec/types.i"
        ENDC

        IFND    EXEC_NODES_I
        include "exec/nodes.i"
        ENDC

x                                       "                       '
x       The 'Init' entries are only used if the corresponding
*       bit is set in the Flags field.
x
*       So if you are just a 68881> you do not need the Init stuff
x       just make sure you have cleared the Flags field.
x
*       This should allow us to add Extended Precision later.
x
*       For Init usersj if you need to be called whenever a task
x       opens this library for use5 you need to change the appropriate
*       entries in MathlEEELibrary.
x


        STRUCTURE MathIEEEResourceResource,0
                STRUCT  MathIEEEResource_Node,LN_SIZE   ,
                USHORT  MathIEEEResource_Flags
                APTR    MathIEEEResource_BaseAddr       x ptr to 881 if exists *
                APTR    MathIEEEResource~DblBasInit
                APTR    MathIEEEResource_DblTransInit
                APTR    MáthIEEEResource_SglBasInit
                APTR    MathIEEEResource_SglTransInit
                APTR    MathIEEEResource_ExtBasInit
                APTR    MathIEEEResource_ExtTransInit
        LABEL   MathIEEEResourceResource_SIZE

* definations for MathIEEERESOURCE_FLAGS x
        BITDEF  MATHIEEERES0URCE>DBLBAS,0
        BITDEF  MATHIEEERESOURCE,DBLTRANS,1
        BITDEF  MATHIEEERESOURCE,SGLBAS,2
        BITDEF  MATHIEEERESOURCE5SGLTRANS,3
        BITDEF  MATHIEEERESOURCE5EXTBAS,4
        BITDEF  MATHIEEERESOURCE,EXTTRANS,5

        ENDC    ; RESOURCES_MATHRESOURCE_I
```

```
        IFND    RESOURCES_MISC_I
RESOURCES_MISC_I        SET     1
XX
**      SFilename: resources/misc.i $
**      SRelease: 1.3 $
XX
**      external declarations for mise system resources
XX
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
XX
        IFND    EXEC_TYPES_I
        INCLUDE "exec/types.i"
        ENDC    ; EXEC_TYPES_I

        IFND    EXEC_LIBRARIES_I
        INCLUDE "exec/libraries.i"
        ENDC    ; EXEC_LIBRARIES_I

*****************************************************************************
*
* Resource structures
*
*****************************************************************************

MR_SERIALPORT   EQU     0
MR_SERIALBITS   EQU     1
MR_PARALLELPORT EQU     2
MR_PARALLELBITS EQU     3

NUMMRTYPES      EQU     4

    STRUCTURE MiscResource,LIB_SIZE
        STRUCT  mr_AllocArray,4*NUMMRTYPES
        LABEL   mr_Sizeof

        LIBINIT LIB_BASE
        LIBDEF  MR_ALLOCMISCRESOURCE
        LIBDEF  MR_FREEMISCRESOURCE


MISCNAME        MACRO
                DC.B    'mise.resource',0
                ENDM

        ENDC    ; RESOURCES_MISC_I
```

```
Directory "Lattice_C_5.0.5:Assembler_Headers/workbench" on Saturday 29-Sep-90
icon.i              ~        492_____rwed 07-Nov-88 14:57:10
startup.i                    965_____rwed 07-Nov-88 14:57:09
workbench.i                 2706_____rwed 07-Nov-88 14:57:10
3 files - 13 blocks - 4163 bytes
```

```
        IFND    WORKBENCH_STARTUP_I
WORKBENCH_STARTUP_I     SET     1~

XI      JFilename: workbench/startup.i $
**      SRelease: 1.3 $
XX
XX      Workbench startup definitions
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          All Rights Reserved
XX


        IFND    EXEC_TYPES_I
        INCLUDE "exec/types.i"
        ENDC    ; EXEC_TYPES_I

        IFND    EXEC_PORTS_I
        INCLUDE "exec/ports.i"
        ENDC    ; EXEC_PORTS_I

        IFND    LIBRARIES_DOS_I
        INCLUDE "libraries/dos.i"
        ENDC    ; LIBRARIES_DOS_I

STRUCTURE WBStartup,0
    STRUCT      sm_Message,MN_SIZE      ; a standard message structure
    APTR        sm_Process              ; the process descriptor for you
    BPTR        sm_Segment              ; a descriptor for your code
    LONG        sm_NumArgs              ; the number of elements in ArgList
    APTR        sm_ToolWindow           ; description of window
    APTR        sm_ArgList              ; the arguments themselves
    LABEL       sm_SIZEOF

STRUCTURE WBArg,0
    BPTR        wa_Lock                 ; a lock descriptor
    APTR        wa_Name                 ; a string relative to that lock
    LABEL       wa_SIZEOF

        ENDC    ; WORKBENCH_STARTUP_I
```

```
    APTR        do_ToolWindow           ; only applies to tools
    LONG        do_StackSize            ; only applies to tools
    LABEL       do_SIZEOF

WB_DISKMAGIC    EQU     $e310   ; a magic number, not easily impersonated
WB_DISKVERSION  EQU     1       ; our current version number

 STRUCTURE FreeList,0
    WORD        fi_NumFree
    STRUCT      fl_MemList,LH_SIZE
    ; weird name to avoid conflicts with FileLocks
    LABEL       FreeList_SIZEOF


* each message that comes into the WorkBenchPort must have a type field
* in the preceeding short.  These are the defines for this type
*

MTYPE_PSTD              EQU     1       ; a "standard Potion" message
MTYPE_TOOLEXIT          EQU     2       ; exit message from our tools
MTYPE_DISKCHANGE        EQU     3       ; dos teli ing us of a disk change
MTYPE_TIMER             EQU     4       ; we got a timer tick
MTYPE_CLOSEDOWN         EQU     5       ; <unimplemented>
MTYPE_IOPROC            EQU     6       ; <unimplemented>


* workbench does different complement modes for its gadgets.
* It supports separate imagesj complement mode> and backfill mode.
* The first two are identical to intuitions GADGIMAGE and GADGHCOMP.
* backfill is similar to GADGHCOMP, but the region outside of the
* image (which normally would be color three when complemented)
* is flood-filled to color zero.
*
GADGBACKFILL            EQU     $0001

* if an icon does not really live anywhere, set its current position
* to here
*
NO_ICON_POSITION        EQU     ($80000000)

        ENDC    J WORKBENCH_WORKBENCH_I
```

```
Directory "Lattice_C_5.0.5:Assemblerjteaders/hardware" on Saturday 29-Sep-90
adkbits.i                      1912 ----rwed 07-Nov-88 14:58:20
blit.i                         1723 ----rwed 07-Nov-88 14:58:20
cia.i                          4526 ----rwed 07-Nov-88 14:58:19
custom.i                       2383 ----rwed 07-Nov-88 14:58:18
dmabits.i                      1180 ----rwed 07-Nov-88 14:58:19
intbits.i                      1722 ----rwed 07-Nov-88 14:58:20
6 files - 36 blocks - 13446 bytes
```

```
        IFND    HARDWARE_ADKBITS_I
HARDWARE_ADKBITS_I      SET     1
xx
x*      SFilenarae: hardware/adkbits.i $
xx      SRelease: 1.3 $
xx
xx      bit definitions for adkcon register
xx
xx      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
xx          Ali Rights Reserved
xx

ADKB_SETCLR     EQU   15 ; standard set/clear bit
ADKB_PRECOMP1   EQU   14 ; two bits of precompensation
ADKB_PRECOMPO   EQU   13
ADKB_MFMPREC    EQU   12 ; use mfm style precompensation
ADKB_UARTBRK    EQU   11 ; force uart output to zero
ADKB_WORDSYNC   EQU   10 ; enable DSKSYNC register matching
ADKB_MSBSYNC    EQU   9  ; (Apple GCR Only) sync on MSB for reading
ADKB_FÁST       EQU   8  ; 1 -> 2 us/bit (mfm), 2 -> 4 us/bit (gcr)
ADKB_USE3PN     EQU   7  ; use aud chan 3 to modulate period of ??
ADKB_USE2P3     EQU   6  ; use aud chan 2 to modulate period of 3
ADKB_USE1P2     EQU   5  ; use aud chan 1 to modulate period of 2
ADKB_USEOP1     EQU   4  ; use aud chan 0 to modulate period of 1
ADKB_USE3VN     EQU   3  ; use aud chan 3 to modulate volume of ??
ADKB_USE2V3     EQU   2  ; use aud chan 2 to modulate volume of 3
ADKB_USE1V2     EQU   1  ; use aud chan 1 to modulate volume of 2
ADKB_USEOV1     EQU   0  ; use aud chan 0 to modulate volume of 1

ADKF_SETCLR     EQU   (1<<15)
ADKF_PRECOMP1   EQU   (1<<14)
ADKF_PRECOMPO   EQU   (1<<13)
ADKF_MFMPREC    EQU   (1<<12)
ADKF_UARTBRK    EQU   (1<<11)
ADKF_WORDSYNC   EQU   (1<<10)
ADKF_MSBSYNC    EQU   (1<<9)
ADKF_FÁST       EQU   (1<<8)
ADKF_USE3PN     EQU   (1<<7)
ADKF_USE2P3     EQU   (1<<6)
ADKF_USE1P2     EQU   (1<<5)
ADKF_USEOP1     EQU   (1<<4)
ADKF_USE3VN     EQU   (1<<3)
ADKF_USE2V3     EQU   (1<<2)
ADKF_USE1V2     EQU   (1<<1)
ADKF_USEOV1     EQU   (1<<0)

ADKF_PREOOONS   EQU   0                       ; 000 ns of precomp
ADKF_PRE140NS   EQU   (ADKF_PRECOMP0)         ; 140 ns of precomp
ADKF_PRE280NS   EQU   (ADKF_PRECOMP1)         ; 280 ns of precomp
ADKF_PRE560NS   EQU   (ADKF_PRECOMPO!ADKF_PRECOMP1) ; 560 ns of precomp

        ENDC    ; HARDWARE_ADKBITS_I
```

```
        IFND    HARDWARE_BLIT_I
HARDWARE_BLIT_I SET     1
XI
**      JFilename: hardware/blit.i $
**      SRelease: 1.3 S
XX
XX
XX
XX      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

   STRUCTURE bltnode,0
   LONG  bn_n
   LONG  bn_function
   BYTE  bn_stat
   BYTE  bn_dummy
   WORD  bn_blitsize
   WORD  bn_beamsync
   LONG  bn_cleanup
   LABEL bn_SIZEOF

* bit defines used by blit queuer
CLEANMEn    equ 6
CLEANME     equ K<CLEANMEn

* include file for blitter */
HSIZEBITS   equ   6
VSIZEBITS   equ   1S-HSIZEBITS
HSIZEMASK   equ   $3f         /* 2"6 -- 1 */
VSIZEMASK   equ   $3FF        /* 2" 10 - 1 */

MAXBYTESPERROW EQU   128

x definitions for blitter control register 0 */

ABC         equ    $80
ABNC        equ    $40
ANBC        equ    $20
ANBNC       equ    $10
NABC        equ    $8
NABNC       equ    $4
NANBC       equ    $2
NANBNC      equ '  $1

BCOB_DEST   equ      8
J3C0B_SRCC  equ      9
BCOB_SRCB   equ      10
BCOB_SRCA   equ      11
BCOF_DEST   equ    $100
BCOF_SRCC   equ    $200
BCOF_SRCB   equ    $400
BCOF_SRCA   equ    $800

BC1F_DESC   equ 2

DEST        equ    $100
SRCC        equ    $200
SRCB        equ    $400
SRCA        equ    $800

ASHIFTSHIFT equ   12 /* bits to right align ashift value */
BSHIFTSHIFT equ   12 /* bits to right align bshift value */
```

```
* definations for blitter control register 1 */
LINEMODE    equ   $1
FILL_OR     equ   S8
FILLJCOR    equ   $10
FILLICARRYIN    equ    $4
ONEDOT      equ   $2
OVFLAG      equ   $20
5IGNFLAG    equ   $40
8LITREVERSE equ   $2

SUD         equ   $10
5UL         equ   $8
AUL         equ   $4

0CTANT8     equ   24
0CTANT7     equ   4
0CTANT6     equ   12
0CTANT5     equ   28
0CTANT4     equ   20
OCTANT3     equ   8
0CTANT2     equ   0
0CTANT1     equ   16

        ENDC    \ HARDWARE_BLIT_I
```

```
        IFND    HARDWARE_CIA_I
HARDWARE_CIA_I  SET     1
«i
**      JFilename: hardware/cia.i $
**      SRelease: 1.3 $

**      registers and bits in the Complex Interface Adapter (CIA) chip
**


«*

*
* _ciaa is on an ODD address (e.g. the low byte) -- IbfeOOl
* _ciab is on an EVÉN address (e.g. the high byte) -- SbfdOOO
*
* do this to get the definitions:
*    XREF _ciaa
*    XREF _ciab
*


* cia register offsets
ciapra          EQU   $0000
ciaprb          EQU   $0100
ciaddra         EQU   $0200
ciaddrb         EQU   $0300
ciatalo         EQU   $0400
ciatahi         EQU   $0500
ciatblo         EQU   $0600
ciatbhi         EQU   $0700
oiatodlow       EQU   $0800
ciatodraid      EQU   $0900
oiatodhi        EQU   $0A00
oiasdr          EQU   $0C00
ciaicr          EQU   $0D00
oiacra          EQU   $OE0O
ciacrb          EQU   $0F00

* interrupt control register bit numbers
CIAICRBJFA      EQU   0
CIAICRB_TB      EQU   1
CIAICRB_ALRM    EQU   2
CIAICRB_SP      EQU   3
CIAICRB_FLG     EQU   4
CIAICRB_IR      EQU   7
CIAICRB_SETCLR  EQU   7

* control register A bit numbers
CIACRAB_START   EQU   0
CIACRAB_PBON    EQU   1
CIACRAB_OUTMODE EQU   2
CIACRAB_RUNMODE EQU   3
CIACRAB_LOAD    EQU   4
CIACRAB_INMODE  EQU   5
CIACRAB_SPMODE  EQU   6
CIACRAB_TODIN   EQU   7

* control register B bit numbers
CIACRBB_START   EQU   0
CIACRBB_PBON    EQU   1
CIACRBB_OUTMODE EQU   2
CIACRBB_RUNMODE EQU   3
```

```
CIACRBB_LOAD        EQU    4
CIACRBBJNMODEO      EQU    5
CIACRBB_INMODE1     EQU    6
CIACRBB_ALARM       EQU    7

* interrupt control register bit masks
CIAICRF_TA          EQU    (l<<0)
CIAICRF_TB          EQU    (1<<1)
CIAICRF_ALRM        EQU    (1<<2)
CIAICRF_SP          EQU    (1<<3)
CIAICRF_FLG         EQU    (1«4)
CIAICRF_IR          EQU    (1<<7)
CIAICRF_SETCLR      EQU    (1<<7)

* control register A bit masks
CIACRAF_START       EQU    (l<<0)
CIACRAF_PBON        EQU  (1< <1)
CIACRAF_OUTMODE     EQU  (1< < 2)
CIACRAF_RUNMODE     EQU  (1< < 3 )
CIACRAF_LOAD        EQU  (1< < 4)
CIACRAF_INMODE      EQU    (1<<5)
ClACRAF_SPMODE      EQU    (1<<6)
CIACRAF_TODIN       EQU    (1<<7)

* control register B bit masks
ClACRBF_START       EQU    (1<<0)
CIACRBF_PBON        EQU    (1<<1)
CIACRBF_OUTMODE     EQU  (1< < 2)
CIACRBF_RUNMQDE     EQU    Cl<<3)
CIACRBF_LOAD        EQU  (1< < 4 )
CIACRBF_INMODE0     EQU    (1<<5)
CIACRBF_INMODE1     EQU    (1<<6)
CIACRBF_ALARM       EQU    (1<<7)

* control register B INMODE masks
CIACRBF_IN_PHI2     EQU    0
CIACRBF_IN_CNT      EQU    (CIACRBFJNMODEO)
CIACRBF_IN_TA       EQU    (CIACRBFJNMODE1 )
CIACRBF_IN_CNT_TA   EQU    (CIACRBF_INMODEO!CIACRBF_INMODE1)


i
* Port definitions -- what each bit in a cia peripheral register is tied to
*

* ciaa port A (0xbfe001)
CIAB_GAMEPORT1      EQU    (7)    * gameport 1, pin 6 (fire button*)
CIAB_GAMEPORTO      EQU    (6)    * gameport 0, pin 6 (fire button*)
CIAB_DSKRDY         EQU    (5)    * disk ready*
CIAB_DSKTRACKO      EQU    (4)    * disk on track 00*
CIAB_DSKPROT        EQU    (3)    * disk write protect*
CIAB_DSKCHANGE      EQU    (2)    * disk change*
CIABJ.ED            EQU    (1)    * led light control (0==>bright)
CIAB_OVERLAY        EQU    (0)    * memory overlay bit

* ciaa port B (0xbfe101) -- parallel port

* ciab port A (0xbfd000) -- serial and printer control
CIAB_COMDTR         EQU    (7)    * serial Data Terminál Ready*
CIAB_COMRTS         EQU    (6)    * serial Request to Send*
CIAB_COMCD          EQU    (5)    * serial Carrier Detect*
```

```
CIAB_COMCTS       EQU    (4)    * serial Clear to Send*
CIAB_COMDSR       EQU    (3)    * serial Data Set Ready*
CIAB_PRTRSEL      EQU    (2)    * printer SELECT
CIAB_PRTRPOUT     EQU    (1)    * printer paper out
CIAB_PRTRBUSY     EQU    (0)    * printer busy

* ciab port B (0xbfdl00) -- disk control
CIAB_DSKMOTOR     EQU    (7)    * disk motorr*
CIAB_DSKSEL3      EQU    (6)    * disk select unit 3*
CIAB_DSKSEL2      EQU    (5)    * disk select unit 2*
CIAB_DSKSEL1      EQU    (4)    * disk select unit 1*
CIAB_DSKSELO      EQU    (3)    * disk select unit 0*
CIAB_DSKSIDE      EQU    (2)    * disk side select*
CIAB_DSKDIREC     EQU    (1)    * disk direction of seek*
CIAB_DSKSTEP      EQU    (0)    * disk step heads*

* ciaa port A (0xbfe001)
CIAF_GAMEPORT1    EQU    (1<<C7)
CIAF_GAMEPORTO    EQU    (1<<C6)
CIAF_DSKRDY       EQU    (1<<;5)
CIAF_DSKTRACKO    EQU    (1<<;4)
CIAF_DSKPROT      EQU    Cl<<:3)
CIAF_DSKCHANGE    EQU    (1<<:2)
CIAF_LED          EQU    (1<<;l)
CIAF_OVERLAY      EQU    (1<«:0)

* ciaa port B (0xbfel01) -- parallel port

* ciab port A (0xbfd000) -- serial and printer control
CIAF_COMDTR       EQU    (1<<í?)
CIAF_COMRTS       EQU    (1<<C6)
CIAF_COMCD        EQU    (1<<í5)
CIAF_COMCTS       EQU    (1<<C4)
CIAF_COMDSR       EQU    (1<<C3)
CIAF_PRTRSEL      EQU    (1<<:2)
CIAF_PRTRPOUT     EQU    (1<<:l)
CIAF_PRTRBUSY     EQU    (1<<C0)

* ciab port B (0xbfdl00) -- disk control
CIAF_DSKMOTOR     EQU    (1<<;7)
CIAF_DSKSEL3      EQU    (1<<;6)
CIAF_DSKSEL2      EQU    (1<<C5)
CIAF_DSKSEL1      EQU    (1<<:4)
CIAF_DSKSELO      EQU    (1<<;3)
CIAF_DSKSIDE      EQU    (1<<C2)
CIAF_DSKDIREC     EQU    (1<<Cl)
CIAF_DSKSTEP      EQU    (1<«C0)

        ENDC    ; HARDWARE_Cl A_I
```

```
        IFND    HARDWARE_CUSTOM_I
HARDWARE_CUSTOM_I       SET     1
t*
**      $Filename: hardware/custom.i $
**      SRelease: 1.3 $
**
**
**      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
**          Ali Rights Reserved
**

*
* do this to get base of custom registers;
*  XREF _custom;
*

bltddat     EQU   $000
dmaconr     EQU   $002
vposr       EQU   $004
vhposr      EQU   $006
dskdatr     EQU   $008
joyOdat     EQU   $00A
joyldat     EQU   $00C
clxdat      EQU   $00E

adkconr     EQU   $010
potOdat     EQU   $012
potldat     EQU   $014
potinp      EQU   $oia
serdatr     EQU   $018
dskbytr     EQU   $01A
intenar     EQU   $01C
intreqr     EQU   $01E

dskpt       EQU   $020
dsklen      EQU   $024
dskdat      EQU   $026
refptr      EQU   $028
vposw       EQU   $02A
vhposw      EQU   $02C
oopcon      EQU   $02E
serdat      EQU   $030
serper      EQU   $032
potgo       EQU   $034
joytest     EQU   $036
strequ      EQU   $038
strvbl      EQU   $03A
strhor      EQU   $03C
strlong     EQU   $03E

bltconO     EQU   $040
bltconl     EQU   $042
bltafwm     EQU   $044
bltalwm     EQU   $046
bltcpt      EQU   $048
bltbpt      EQU   $04C
bltapt      EQU   $050
bltdpt      EQU   $054
bltsize     EQU   $058

bltcmod     EQU   $060
bltbmod     EQU   $062
```

```
bltamod     EQU   $064
bltdmod     EQU   S066

bltcdat     EQU   $070
bltbdat     EQU   $072
bitadat     EQU   $074

dsksync     EQU   $07E

coplle      EQU   $080
cop21c      EQU   $084
copjrapl    EQU   $088
copjmp2     EQU   $08A
copins      EQU   $08C
diwstrt     EQU   $08E
diwstop     EQU   $090
ddfstrt     EQU   $092
ddfstop     EQU   $094
dmacon      EQU   $096
clxcon      EQU   $098
intena      EQU   $09A
jntreq      EQU   $09C
adkeon      EQU   $09E

aud         EQU   $0A0
audO        EQU   $0A0
audl        EQU   $0B0
aud2        EQU   $0C0
aud3        EQU   $0D0

* STRUCTURE AudChannel>0
ac_ptr      EQU   $00   ; ptr to start of waveform data
ac_len      EQU   $04   ; length of waveform in words
ac_per      EQU   $06   ; sample period
ac_vol      EQU   $08   ; volume
ac_dat      EQU   $0A   ; sample pair
acIsiZEOF   EQU   $10

bplpt       EQU   $0E0

bplconO     EQU   $100
bplconl     EQU   $102
bplcon2     EQU   $104
bpl1raod    EQU   $108
bpl2mod     EQU   $10A

bpldat      EQU   $110

sprpt       EQU   $120

spr         EQU   $140
* STRUCTURE SpriteDef
sd_pos      EQU   $00
sd_ctl      EQU   $02
sd_dataa    EQU   $04
sd_datab    EQU   $08

color       EQU   $180

        ENDC    \ HARDWARE_CUSTOM_I
```

```
        IFND    HARDWARE_DMABITS_I
HARDWARE_DMABITS_I      SET     1
*x
**      SFilename: hardware/dmabits.i $
**      ÍRelease: 1.3 $
xx
*x      include file for defining dma control stuff
*x
*x      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
(x          Ali Rights Reserved
**

* write definitions for dmaconw
DMAF_SETCLR     EQU   $8000
DMAF_AUDIO      EQU   SOOOF   * 4 bit mask
DMAF_AUD0       EQU   $0001
I)MAF_AUD1      EQU   $0002
DMAF_AUD2       EQU   $0004
DMAF_AUD3       EQU   $0008
DMAF_DISK       EQU   $0010
DMAF_SPRITE     EQU   $0020
DMAF_BLITTER    EQU   $0040
DMAF_COPPER     EQU   $0080
DMAFJRASTER     EQU   $0100
f)MAF_MASTER    EQU   $0200
DMAF_BLITHOG    EQU   $0400
DMAF_ALL        EQU  $01  FF   * all dma channels

* read definitions for dmaconr
* bits 0-8 correspnd to dmaconw definitions
DMAF_BLTDONE    EQU   $4000
DMAF_BLTNZERO   EQU   $2000

DMAB_SETCLR     EQU   15
DMAB_AUDO       EQU   0
DMAB_AUDI       EQU   1
DMAB_AUD2       EQU   2
DMAB_AUD3       EQU   3
DMAB_DI5K       EQU   4
DMAB_SPRITE     EQU   5
DMAB_BLITTER    EQU   6
DMAB_COPPER     EQU   7
DMAB_RASTER     EQU   8
DMAB_MASTER     EQU   9
DMAB_BLITHOG    EQU   10
DMAB~BLTDONE    EQU   14
DMAB_BLTNZERO   EQU   13

        ENDC    ; HARDWARE_DMABITS_I
```

```
Directory "Lattice_C_5.0.5:Assembler_Headers/libraries" on Saturday 29-Sep-90
configregs.i                5922  ---rwed 07-Nov-88  15:16:07
configvars.i                1329  ---rwed 07-Nov-88  15:16:08
diskfont.i                  2026  ----rwed 07-Nov-88  15:16:08
dos.i                       5709  ---rwed 07-Nov-88  15:16:07
dosextens.i                 9717  ----rwed 07-Nov-88  15:16:08
dos_lib.i                   1073  ----rwed 07-Nov-88  15:16:06
expansion.i                  405  ----rwed 07-Nov-88  15:16:08
expansionbase.i             1711  ----rwed 07-Nov-88  15:16:09
filehandler.i               4850  ----rwed 07-Nov-88  15:16:07
mathlibrary.i               1246  ----rwed 07-Nov-88  15:16:07
roraboot_base.i              939  ----rwed 07-Nov-88  15:16:09
translator.i                 412  ---rwed 07-Nov-88  15:16:09
12 files - 89 blocks - 35339 bytes
```

```
        IFND    LIBRARIES_CONFIGREGS_I
LIBRARIES_CONFIGREGS_I  SET     1
**
**      SFilename: libraries/configregs.i $
**      SRelease: 1.3 $
**
**      register and bit definitions for expansion boards
**
**      (C) Copyright 1986,1987,1988 Commodore-Amiga, Inc.
**          All Rights Reserved
**

** Expansion boards are actually organized such that only one nibble per
** word (16 bits) are valid information.  This table is structured
** as LOGICAL information.  This means that it never corresponds
** exactly with a physical implementation.
**
** The expansion space is logically split into two regions:
** a rom portion and a control portion.  The rom portion is
** actually stored in one's complement form (except for the
** er_type field).


 STRUCTURE  ExpansionRom,0
    UBYTE       er_Type
    UBYTE       er_Product
    UBYTE       er_Flags
    UBYTE       er_Reserved03
    UWORD       er_Manufacturer
    ULONG       er_SerialNumber
    UWORD       er_InitDiagVec
    UBYTE       er_Reserved0c
    UBYTE       er_Reserved0d
    UBYTE       er_Reserved0e
    UBYTE       er_Reserved0f
    LABEL       ExpansionRom_SIZEOF

 STRUCTURE ExpansionControl,0
    UBYTE       ec_Interrupt            ; interrupt control register
    UBYTE       ec_Reserved11
    UBYTE       ec_BaseAddress          ; set new config address
    UBYTE       ec_Shutup               ; don't respond, pass config out
    UBYTE       ec_Reserved14
    UBYTE       ec_Reserved15
    UBYTE       ec_Reserved16
    UBYTE       ec_Reserved17
    UBYTE       ec_Reserved18
    UBYTE       ec_Reserved19
    UBYTE       ec_Reserved1a
    UBYTE       ec_Reserved1b
    UBYTE       ec_Reserved1c
    UBYTE       ec_Reserved1d
    UBYTE       ec_Reserved1e
    UBYTE       ec_Reserved1f
    LABEL       ExpansionControl_SIZEOF

**
** many of the constants below consist of a triplet of equivalent
** definitions: xxMASK is a bit mask of those bits that matter.
** xxBIT is the starting bit number of the field.  xxSIZE is the
** number of bits that make up the definition.  This method is
** used when the field is larger than one bit.

** If the field is only one bit wide then the xxB_xx and xxF_xx convention
```

```
** is used CxxB_xx is the bit number, and xxF_xx is mask of the bit).
IX

** manifest constants */
E_5LOTSIZE              EQU     $10000
EJiLOTMASK              EQU     Sffff
E_SLOT5HIFT             EQU     16

** these define the two free regions of Zorro raemory space.
** THESE MAY WELL CHANGE FOR FUTURE PRODUCTS!
EJiXPANSIONBASE         EQU     $e80000
EJIXPANSIONSIZE         EQU     $080000
E_F.XPANSIONSLOTS       EQU     8

E_MEMORYBASE            EQU     $200000
E_MEMORYSIZE            EQU     $800000
E_MEMORYSLOTS           EQU     128


XXKX*** ec_Type definitions */

** board type -- ignore "old style" boards */
ERT_TYPEMASK          . EQU     $cO
ERTJTYPEBIT             EQU     6
ERT TYPESIZE            EQU     2
ERT~NEWBOARD            EQU     $cO


** type field memory size */
ERTJCMMASK              EQU     $07
ERT_MEMBIT              EQU     0
ERT_MEMSIZE             EQU     3


** other bits defined in type field */
        BITDEF  ERT,CHAINEDCONFIG,3
        BITDEF  ERT,DIAGVALID,4
        BITDEF  ERT,MEMLIST,5


** er_Flags byte -- for those things that didn't fit intő the type byte */
        BITDEF  ERF,MEMSPACE,7          ; wants to be in 8 meg space.  Alsó
                                        ;     implies that board is moveable
        BITDEF  ERF,N0SHUTUP,6          ; board can't be shut up.  Must not
                                        ;     be a board,  Must be a box that
                                        ;     does not pass on the bus.


** interrupt control register */
        BITDEF  ECI,INTENA,1
        BITDEF  ECI,RESET,3
        BITDEF  ECI,INT2PEND,4
        BITDEF  ECI,INT6PEND,5
        BITDEF  ECI,INT7PEND,6
        BITDEF  ECI,INTERRUPTING,7

**********************************************************************
**
** these are the spécifications for the diagnostic area.  If the Diagnostic
```

```
Mft Address Valid bit is set in the Board Type byte (the first byte in
XX expansion space) then the Diag Init vector contains a valid offset.
*»
XX The Diag Init vector is actually a word offset from the base of the
XX board.  The resulting address points to the base of the DiagArea
XX structure.  The structure may be physically implemented either four>
XX eight, or sixteen bits wide.  The code will be copied out intő
XX ram first before being called.   .
XX
XX The da_Size fieldj and both code offsets (da_DiagPoint and da_BootPoint)
XX are offsets from the diag area AFTER it has been copied intő ram> and
*X "de-nibbleized" (if needed).  Inotherwords> the size is the size of
XX the actual information, not how much address space is required to
*X store it.
XX
*X Ali bits are encoded with uninverted logic (e.g. 5 volts on the bus
XX is a logic one).
IX
IX If your board is to make use of the boot facility then it must leave
*X its config area available evén after it has been configured.  Your
XX boot vector will be called AFTER your board's final address has been
XX set.
»X
*************************************************************************

 STRUCTURE DiagArea,0
    UBYTE       da_Config       > see below for definitions
    UBYTE       da_Flags        ; see below for definitions
    UWORD       da_Size         ; the size (in bytes) of the totál diag area
    UWORD       da_DiagPoint    > where to start for diagnostics> or zero
    UWORD       da_BootPoint    > where to start for booting
    UWORD       da_Name         ; offset in diag area where a string
                                j   identifier can be found (or zero if no
                                >   identifier is present).

    UWORD       da_Reserved01   ; two words of reserved data.  must be zero.
    UWORD       da_ReservedO2
    LABEL       DiagArea_SIZEOF

; da_Config definitions
CAC_BUSWIDTH    EQU     SCO     ; two bits for bus width
DAC_NIBBLEWIDE  EQU     $00
CAC_BYTEWIDE    EQU     $40
DAC_WORDWIDE    EQU     $80

DAC_BOOTTIME    EQU     $30     i two bits for when to boot
CAC_NEVER       EQU     $00     » obvious
DAC_CONFIGTIME  EQU     $10     » call da_BootPoint when first configing the
                                ;   the device
DAC_BINDTIME    EQU     $20     i run when binding drivers to boards

XX
xx These are the calling conventions for Diag or Boot area
XX
** A7 -- points to at least 2K of stack
xx A6 -- ExecBase
xx A5 -- ExpansionBase
** A3 -- your board's ConfigDev structure
xx A2 -- Base of diag/init area that was copied
** A0 -- Base of your board
XX
```

```
** Your board should return a value in DO.  If this value is NULLj then
** the diag/init area that was copied in will be returned to the free
** memory pool.
XX

        ENDC    ; LIBRARIES_CONFIGREGS_I
```

```
        ÍFND    LIBRARIES_CONFIGVARS_I
LIBRARIES_CONFIGVARS_I  SET     1   .
IX
XX      SFilename: libraries/configvars.i $
XX      SRelease: 1.3 $
XX
XX      software structures for configuration subsystem
XX
XX      (C) Copyright 1986,1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX


        IFND    EXEC_NODES_I
        INCLUDE "exec/nodes.i"
        ENDC    J EXEC_NODES_I

        IFND    LIBRARIE5_C0NFIGREGS_I
        INCLUDE "libraries/configregs.i"
        ENDC    > LIBRARIES_CONFIGREGS_I


 STRÜCTURE ConfigDev,0
    STRUCT      cd_Node,LN_SIZE
    UBYTE       cd_Flags
    UBYTE       cd_Pad
    STRUCT      cd_Rom,ExpansionRom_SIZEOF ; copy of boards config rom
    APTR        cd_BoardAddr    ; where in memory the board is
    APTR        cd_BoardSize    ; size in bytes
    UWORD       cd_SlotAddr     i which slot nuraber
    UWORD       cd_SlotSize     ; number of slots the board takes
    APTR        cd_Driver       ; pointer to node of driver
    APTR        cd_NextCD       ; linked list of drivers to config
    STRUCT      cd_Unused,4*4   ; for whatever the driver whats
    LABEL       ConfigDev_SIZEOF

; cd_Flags
        BITDEF  CD,SHUTUP,0     ; this board has been shut up
        BITDEF  CD,CONFIGME,1   ; this board needs a driver to claim it

; this structure is used by GetCurrentBindingC) and SetCurrentBinding()
 STRÜCTURE CurrentBinding,0
    APTR        cb_ConfigDev
    APTR        cb_FileName
    APTR        cb_ProductString
    APTR        cbJToolTypes
    LABEL       CurrentBinding_SIZEOF

        ENDC    ; LIBRARIES_CONFIGVARS_I
```

```
        IFND    LIBRARIES_DISKFONTJ
LIBRARIES_DISKFONT_I    SET     1
XX
/**      SFilename:  libraries/diskfont.i $
**       JRelease:  1.3 $
XX
**       diskfont  library definitions
**
**       (C) Copyright 1985,1986,1987,1988 Commodore-Amiga,  Inc„
**           Ali Rights Reserved
XX

   IFND     EXEC_NODES_I
   INCLUDE     "exec/nodes.i"
   ENDC
   IFND     EXECJ.ISTS_I
   INCLUDE     "exec/lists.i"
   ENDC
   IFND     GRAPHICS_TEXT_I
   INCLUDE     "graphics/text.i"
   ENDC

MAXFONTPATH EQU   256   ', including null terminator

 STRUCTURE  FC,0
   STRUCT   fc_FileName,MAXFONTPATH
   UWORD fc_YSize
   UBYTE fc_Style
   UBYTE fc_Flags
   LABEL fc_SIZEOF

FCH_ID      EQU   ÍOfOO

 SrRUCTURE  FCH,0
   UWORD fch_FileID  1 FCH_ID
   UWORD fch_NumEntries ; the number of FontContents elements
   LABEL fch_FC     ; the FontContents elements


DFH_ID      EQU   S0f80
MAXFONTNAME EQU   32 J font name including ".font\0"

 STRUCTURE  DiskFontHeader,0
    ; the following 8 bytes are not actually considered a part of the
    > DiskFontHeader, but immediately preceed it.  The NextSegment is supplied
    ; by the linker/loader, and the ReturnCode is the code at the beginning
    ; of the font in case someone runs it...
    ; ULONG dfh_NextSegment   > actually a BPTR
    ; ULONG dfh_ReturnCode    ; MOVEQ #0,D0 : RTS
    ; here then is the official start of the DiskFontHeader...
   STRUCT   dfh_DF,LN_SIZE    ; node to link disk fonts
   UWORD dfh_FifeID           ; DFH_ID
   UWORD dfh_Revision         ; the font revision in this version
   LONG  dfh_Segment          ; the segment address when loaded
   STRUCT   dfh_Name,MAXFONTNAME ; the font name (null terminated)
   STRUCT   dfh_TF,tf_SIZEOF  J loaded TextFont structure
   LABEL dfh_SIZEOF


   BITDEF   AF,MEMORY,0
   BITDEF   AF,DISK,1

 STRUCTURE  AF,0
```

```
    UWORD af_Type        ; MEMORY or DISK
    STRUCT   af_Attr,ta_SIZEOF ; text attributes for font
    LABEL af_SIZEOF

STRUCTURE   AFH,0
    UWORD afh_NumEntries    » number of AvailFonts elements
    LABEL afh_AF         > the AvailFonts elements

        ENDC    ; LIBRARIES_DISKFONT_I
```

```
        IFND    LIBRARIES_DOS_I
LIBRARIES_DOS_I SET     1
**
*s      SFilename: libraries/dos.i S
XX      iRelease: 1.3 £
XX
XI      Standard assembler header for AmigaDOS
XX
XI      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
ti          Ali Rights Reserved
XX

*       IFND EXEC_TYPES_I
x       INCLUDE "exec/types.i"
*       ENDC


DOSNAME     MACRO
      DC.B  'dos.library',0
      ENDM

x Predefined Amiga DOS global constants

DOSTRUE       EQU     -1
DOSFALSE      EQU      0

* Mode paraméter to OpenC)
M0DE_OLDFILE         EQU   1005   * Open existing fi le read/write
x                                 x positioned at beginning of file.
MODE_NEWFILE         EQU   1006   * Open freshly created file (delete
*                                 * old file) read/write
MODE_READWRITE       EQU   1004   * Open old file w/exclusive lock
x Relative position to Seekí)
OFFSET_BEGINNING     EQU   -1     * relative to Beginning Of File
OFI SET_CURRENT      EQU    0     * relative to Current file position
OFFSET_END           EQU    1     * relative to End Of File

OFFSET_BEGINING     EQU   OFFSET_BEGINNING     x Ancient compatibility

BI15PERBYTE          EQU   8
BYTESPERLONG         EQU   4
BITSPERLONG          EQU   32
MAXINT               EQU   S7FFFFFF
MI NINT              EQU   $80000000

x Passed as type to LockC)
SHARED_LOCK          EQU   -2   ; File is readable by others
ACCESS_READ          EQU   -2   ; Synonym
EXCLUSIVE_LOCK       EQU   -1   ; No other access allowed
ACCESS_WRITE         EQU   -1   j Synonym


 STRUCTURE DateStamp,0
   LONG  ds_Days               ; Number of days since Jan. 1, 1978
   LONG  ds_Minute             ; Number of minutes pást midnight
   LONG  ds_Tick               ; Number of ticks pást minute
   LABEL ds_SIZE0F             ; DateStamp
TICKS_PER_SECOND EQU 50        ; Number of, ticks in one second

* Returned by ExamineC) and ExInfoC)
 STRUCTURE FileInfoBlock,0
   LONG   fib_DiskKey
   LONG   fib_DirEntryType     >'Type of Directory.  If < 0, then a plain file.
```

```
                                    ; If > 0 a directory
   STRUCT fib_FileName,108          ; Null terminated. Max 30 chars used for now •
   LONG   fib_Protection            ; bit mask of protection> rwxd are 3-0.
   LONG   fib_EntryType
   LONG   fib_Size                  ; Number of bytes in file
   LONG   fib~NumBlocks             ; Number of blocks in fi le
   STRUCT fibJDateStamp,ds_SIZEOF   > Date file last changed.
   STRUCT fib_Conunent,80           ; Null terminated. Comment associated with file
   STRUCT fib_Reserved,36
   LABEL  fib_SIZEOF                > FileInfoBlock

* FIB stands for FileInfoBlock
* flBB are bit definitions, FIBF are field definitions
   BITDEF   FIB,SCRIPT,6            ; program is an execute script
   BITDEF   FIB,PURE,5              ; program is reentrant and reexecutable
   BITDEF   FIB,ARCHIVE,4           ; cleared whenever file is changed
   BITDEF   FIB,READ,3              ; ignored by the systera
   BITDEF   FIB,VRITE,2             ; ignored by the system
   BITDEF   FIB,EXECUTE,1           ; ignored by the system
   BITDEF   FIB,DELETE,0            ; prevent file from being deleted


* Ali BCPL data must be long word aligned.  BCPL pointers are the long word
* address (i.e byte address divided by 4 (>>2))

* Macro to indicate BCPL pointers
BPTR     MACRO                       * Long word pointer
         LONG     \1
         ENDM
BSTR     MACRO                       * Long word pointer to BCPL string.
         LONG     \1
         ENDM

« #define BADDR( bptr ) (bptr << 2) * Convert BPTR to byte addressed pointer

* BCPL strings have a length in the first byte and then the characters.
* For example:  síOJ=3 stl]=S st2]=Y s[31=S

* teturned by InfoC)
 STRUCTURE InfoData,0
   LONG id_NumSoftErrors            * number of soft errors on disk
   LONG id_UnitNumber               * Which unit disk is (was) mounted on
   LONG idJ)iskState                * See defines below
   LONG id_NumBlocks                * Number of blocks on disk
   LONG id_NumBlocksUsed            * Number of block in use
   LONG id^BytesPerBlock
   LONG id_DiskType                 * Disk Type code
   BPTR id_VolumeNode               * BCPL pointer to volume node
   LONG id_InUse                    * Flagj zero if not in use
   LABEL id_SIZEOF                  * InfoData

« ID stands for InfoData
i            Disk states
ID_WRITE_PROTECTED      EQU     80      * Disk is write protected
ID_VALIDATING           EQU     81      * Disk is currently being validated
ID_VALIDATED            EQU     82      * Disk is consistent and writeable
i          Disk types
ID_NO_DISK_PRESENT      EQU -1
ID_UNREADABLEJ3ISK      EQU  ('B'<<24)!C'A'<<16)!CD'<<8)
IDINOT_REALLY_DOS       EQU  'N'<<24)!('D'<<16)!('O'<<8)!('S')
ID_DOS_DISK             EQU  CD'<<24)!C0'<<16)!('S'<<8)
```

```
IDJ(ICKSTART_DISK     EQU  ('K'<<24)!('T<<16)!('C<<8)!('K')

* Errors from IoErrO, etc.
ERROR_NO_FREE_STORE               EQU  103
ERROR_TASK_TABLE_FULL             EQU  105
ERROR_LINE_TOO_LONG               EQU  120
ERROR_FILE_NOTJ3BJECT             EQU  121
ERROR_INVALID_RESIDENT_LIBRARY    EQU  122
ERROR_OBJECT_IN_USE               EQU  202
ERROR OBJECT_EXISTS               EQU  203
ERRORlOBJECTJlOT_FOUND            EQU  205
ERROR_ACTION_NOT_KNOWN            EQU  209
ERRORllNVALID_COMPONENT_NAME      EQU  210
ERROR_INVALID_LOCK                EQU  211
ERROR_OBJECT_WRONG_TYPE           EQU  212
ERROR_DISK_NOT_VALIDATED          EQU  213
ERROR_DISK_WRITE_PROTECTED        EQU  214
ERROR_RENAME_ACROSS_DEVICES       EQU  215
ERROR_DIRECTORY_NOT_EMPTY         EQU  216
ERROR_DEVICE_NOT_MOUNTED          EQU  218
ERROR_SEEK_ERROR                  EQU  219
ERRORj:OMMENT_TOO_BIG             EQU  220
ERROR_DISK_FULL                   EQU  221
ERROR_DELETE_PROTECTED            EQU  222
ERROR_WRITE_PROTECTED             EQU  223
ERROR_READ_PROTECTED              EQU  224
ERROR_NOT_A_DOS_DISK              EQU  225
ERROR_NO_DISK                     EQU  226
ERROR_NO_MORE_ENTRIES             EQU  232

* These are the return codes used by convention by AmigaDOS comraands
* See FAILAT and IF for relvance to EXECUTE files
RETURN_OK                         EQU    0  * No problems, success
RETURN_WARN                       EQU    5  * A warning only
RETURN_ERROR                      EQU   10  * Something wrong
RETURN_FAIL                       EQU   20  * Complete or severe failure

* Bit numbers that signal you that a user has issued a break
        BITDEF   SIGBREAK,CTRL_C,12
        BITDEF   SIGBREAK,CTRL_D,13
        BITDEF   SIGBREAK,CTRL_E,14
        BITDEF   SIGBREAK,CTRL_F,15

        ENDC    J LIBRARIES_DOS_I
```

```
        IFND    LIBRARIES_DOSEXTENS_I
LIBRARIES_DOSEXTENS_I   SET     1
**
xx      SFilename: libraries/dosextens.i $
x*      SRelease: 1.3 $
**
xx      DOS structures not needed for the casual AmigaDOS user
xx
xx      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga, Inc.
xx          Ali Rights Reserved
xx

     IFND  EXEC_TYPES_I
     INCLUDE "e xec/types.i"
     ENDC
     IFND  EXEC_TASKS_I
     INCLUDE "exec/tasks.i"
     ENDC
     IFND  EXEC_PORTS_I
     INCLUDE "exec/ports.i"
     ENDC
     IFND  EXEC_LIBRARIES_I
     INCLUDE "exec/libraries.i"
     ENDC

     IFND  LIBRARIES_DOS_I
     INCLUDE "libraries/dos.i"
     ENDC


* Ali DOS processes have this STRUCTure
* Create and DeviceProc returns pointer to the MsgPort in this STRUCTure
x Process_addr = DeviceProcC.. ) - TC_SIZE

 STRUCTURE Process,0
    STRUCT  pr_Task,TC_SI2E
    STRUCT  pr_MsgPort,MP_SIZE  * This is BPTR address from DOS functions
    WORD    pr_Pad              x Remaining variables on 4 byte boundaries
    BPTR    pr_SegList          x Array of seg lists used by this process
    LONG    pr_StackSize        x Size of process stack in bytes
    APTR    pr_GlobVec          x Global vector for this process CBCPL) .
    LONG    pr_TaskNum          x CLI task number of zero if not a CLI
    BPTR    pr_StackBase        * Ptr to high memory end of process stack
    LONG    pr_Result2          * Value of secondary result from last call
    BPTR    pr_CurrentDir       * Lock associated with current directory
    BPTR    pr_CIS              * Current CLI Input Stream
    BPTR    pr_COS              x Current CLI Output Stream
    APTR    pr_ConsoleTask      * Console handler process for current window
    APTR    pr_FileSysteraTask  * File handler process for current drive
    BPTR    pr_CLI              * pointer to Conso1eLineInterpreter
    APTR    pr_ReturnAddr       * pointer to previous stack frame
    APTR    pr_PktWait          x Function to be called when awaiting msg
    APTR    pr_WindowPtr        * Window pointer for errors
    LABEL   pr_SIZEOF           * Process

x The long word address (BPTR) of this STRUCTure is returned by
x Open() and other routines that return a file.  You need only worry
x about this STRUCT to do async io's via PutMsg() instead of
« standard file system calls

 STRUCTURE FileHandle.O
   APTR   fh.Link               x pointer to EXEC message
   APTR   fh_Interactive        * Boolean; TRUE if interactive handle
```

```
   APTR   fh_Type              * Port to do PutMsg() to
   LONG   fh_Buf
   LONG   fh_Pos
   LONG   fh_End
   LONG   fh_Funcs
fh_Funcl EQU fh_Funcs
   LONG   fh_Func2
   LONG   fh_Func3
   LONG   fh_Args
fh_Argl EQU fh_Args
  ~LONG   fh_Arg2
   LABEL  fh_SIZEOF * FileHandle

* This is the extension to EXEC Messages used by DOS
 STRUCTURE DosPacket,0
   APTR   dp_Link         * pointer to EXEC message
   APTR   dp_Port         * pointer to Reply port for the packet
*                         * Must be fiiled in each send.
   LONG   dp_Type         * See ACTION_... below and
*                        * 'R' means Read, 'W' means Write to the file system
   LONG   dp_Resl         * For file system calls this is the result
*                         * that would have been returned by the
*                         * function, e.g. Write CW ) returns actual
*                          * length written
   LONG   dp_Res2         * For file system calls this is what would
*                          * have been returned by IoErrC)
   LONG   dp_Argl
*  Device packets common ecjuivalents
dp_Action  EQU  dp_Type
dp_Status  EQU  dp_Resl
dp_Status2 EQU  dp_Res2
dp_BufAddr EQU  dp_Argl
   LONG   dp_Arg2
   LONG   dp_Arg3
   LONG   dp_Arg4
   LONG   dp_Arg5
   LONG   dp_Arg6
   LONG   dp_Arg7
   LABEL  dp_SIZEOF * DosPacket

* A Packet does not require the Message to before it in memory> but
* for convenience it is useful to associate the two.
* Alsó see the function init_std_pkt for Ínitializing this STRUCTure

 STRUCTURE StandardPacket,0
   STRUCT sp_Msg,MN_SIZE
   STRUCT sp_Pkt,dp_SIZEOF
   LABEL  sp_SIZEOF * StandardPacket


* Packet types
ACTION_NIL              EQU     0
ACTION_GET_BLOCK        EQU     2       50BS0LETE
ACTION_SET_MAP          EQU     4
ACTION_DIE~             EQU     5
ACTION_EVENT            EQU     6
ACTION_CURRENT_VOLUME   EQU     7
ACTION_LOCATE_OBJECT    EQU     8
ACTION_RENAME_DISK      EQU     9
ACTION_WRITE            EQU     'W'
ACTION_READ             EQU     'R'
```

```
ACTION_FREE_LOCK        EQU     15
ACTION_DELETE_OBJECT    EQU     16
ACTION_RENAME_OBJECT    EQU     17
ACTIONJIORE_CACHE       EQU     18
ACTION_COPY_DIR         EQU     19
ACTION_WAIT_CHAR        EQU     20
ACTION~SET_PROTECT      EQU     21
ACTION_CREATE_DIR       EQU     22
ACTION_EXAMINE_OBJECT   EQU     23
ACTION_EXAMINE_NEXT     EQU     24
ACTION_DISK_INFO        EQU     25
ACTION_INFO~            EQU     26
ACTION_FLUSH            EQU     27
ACTION_SET_COMMENT      EQU     28
ACTION_PARENT           EQU     29
ACTION_TIMER            EQU     30
ACTIONJNHIBIT           EQU     31
ACTION_DISK_TYPE        EQU     32
ACTION_DISK_CHANGE      EQU     33
ACTION_SET_DATE         EQU     34

ACTION_SCREEN_MODE      EQU     994

ACTION_READ_RETURN      EQU     1001
ACTION_WRITE_RETURN     EQU     1002
ACTION_SEEK             EQU     1008
ACTION_FINDUPDATE       EQU     1004
ACTION_FINDINPUT        EQU     1005
ACTION_FINDOUTPUT       EQU     1006
ACTION_END              EQU     1007
ACTION~TRUNCATE         EQU     1022    /* fást file system only */
ACTION_WRITE_PROTECT    EQU     1023    /* fást file system only */

* DOS library node structure.
* This is the data at positive offsets from the library node.
* Negative offsets from the node is the jump table to DOS functions
* node = (STRUCT DosLibrary *) OpenLibraryC "dos.library" ..  )

 STRUCTURE DosLibrary>0
    STRUCT  dl_lib,LIB_SIZE
    APTR    dl_Root          * Pointer to RootNode} described below
    APTR    dl~GV            * Pointer to BCPL global vector
    LONG    dl~A2            * Priváté register dump of DOS
    LONG    dl_A5
    LONG    dl~A6
    LABEL   dl_SIZEOF *  DosLibrary

*

 STRUCTURE RootNode,0
    BPTR    rn_TaskArray       * [0] is max number of CLI's
*                              * [1] is APTR to process id of CLI 1
*                              * [n] is APTR to process id of CLI n
    BPTR    rn_ConsoleSegment  * SegList for the CLI
    STRUCT  rn JTime,ds_SIZEOF * Current time
    LONG    rn_RestartSeg      * SegList for the disk validator process
    BPTR    rn_Info            * Pointer ot the Info structure
    BPTR    rn_FileHandlerSegment * code for file handler
    LABEL   rn_SIZEOF * RootNode

 STRUCTURE Doslnfo,0
```

```
    BPTR    di_McNarae          * Network name of this machine currently 0
    BPTR    di_DevInfo          * Device List
    BPTR    di_Devices          * Currently zero
    BPTR    di_Handlers         * Currently zero
    APTR    di_NetHand          * Network handler processid currently zero
    LABEL   di_SIZEOF * Doslnfo

* DOS Processes started frora the CLI via RUN or NEWCLI have this additional
* set to data associated with them

 STRUCTURE CommandLinelnterface,0
    LONG    cli_Result2         * Value of IoErr from last command
    BSTR    cli_SetName         * Name of current directory
    BPTR    cli_CoramandDir     * Lock associated with command directory
    LONG    cli_ReturnCode      * Return code from last command
    BSTR    c1i_CommandName     * Name of current command
    LONG    cli_FailLevel       * Fail levél (set by FAILAT)
    BSTR    cli_Prompt          * Current prompt (set by PROMPT)
    BPTR    cli_StandardInput   * Default (terminál) CLI input
    BPTR    cli_CurrentInput    * Current CLI input
    BSTR    cli_CommandFile     * Name of EXECUTE command file
    LONG    cli_Interactive     * Boolean True if prompts required
    LONG    cli_Background      * Boolean True if CLI created by RUN
    BPTR    cli_CurrentOutput   * Current CLI output
    LONG    cli_DefaultStack    * Stack size to be obtained in long words
    BPTR    cli_StandardOutput  * Default (terminál) CLI output
    BPTR    cli_Modulé          * SegList of currently loaded command
    LABEL   cli_SIZEOF          * CommandLinelnterface

* This structure can take on different values depending on whether it is
* a device, an assigned directory, or a volume.  Below is the structure
* reflecting volumes only.  Following that is the structure representing
* only devices. Following that is the unioned structure representing all
* the values

* structure representing a volume

 STRUCTURE DevList,0
    BPTR        dl_Next                 ; bptr to next device list
    LONG        dl_Type                 ; see DLT below
    APTR        dl_Task                 ; ptr to handler task
    BPTR        dl_Lock                 ; not for volumes
    STRUCT      dl_VolumeDate,ds_SIZEOF ; creation date
    BPTR        dl_LockList             ; outstanding locks
    LONG        dl_DiskType             ; ' DOS', etc
    LONG        dl_unused
    BSTR        dl_Name                 ; bptr to bcpl name
    LABEL       DevList_SIZEOF

* device structure (same as the DeviceNode structure in filehandler.i

 STRUCTURE       Devlnfo,0
    BPTR    dvi_Next
    LONG    dvi_Type
    APTR    dvi_Task
    BPTR    dvi_Lock
    BSTR    dvi_Handler
    LONG    dvi_Stacksize
    LONG    dvi_Priority
    LONG    dvi_Startup
    BPTR    dvi_SegList
```

```
        IFND    LIBRARIES_DOS_LIB_I
LIBRARIES_DOS_LIB_I     SET     1
xx
**      SFilename:  libraries/dos_lib.i $
xx      SRelease:  1.3 $
*ft
xx      Library interface offsets for DOS library
xx
x*      (C) Copyright 1985,1986,1987,1988 Commodore-Amiga,  Inc.
xx          Ali Rights Reserved
xx

reserve EQU     4
vsize   EQU     6
count   SET     -vsize*(reserve+1)
LIBENT  MACRO
_LVO\1  EQU     count
count   SET     count-vsize
        ENDM
x
*
*
   LIBENT   Open
   LIBENT   Close
   LIBENT   Read
   LIBENT   Write
   LIBENT   Input
   LIBENT   Output
   LIBENT   Seek
   LIBENT   DeleteFile
   LIBENT   Rename
   LIBENT   Lock
   LIBENT   UnLock
   LIBENT   DupLock
   LIBENT   Examine
   LIBENT   ExNext
   LIBENT   Info
   LIBENT   CreateDir
   LIBENT   CurrentDir
   LIBENT   IoErr
   LIBENT   CreateProc
   LIBENT   Exit
   LIBENT   LoadSeg
   LIBENT   UnLoadSeg
   LIBENT   GetPacket
   LIBENT   QueuePacket
   LIBENT   DeviceProc
   LIBENT   SetComraent
   LIBENT   SetProtection
   LIBENT   DateStamp
   LIBENT   Delay
   LIBENT   WaitForChar
   LIBENT   ParentDir
   LIBENT   Islnteractive
   LIBENT   Execute

        ENDC    ; LIBRARIES_DOS_LIB_I
```

```
        IFND    LIBRARIES_EXPANSION_I
LIBRARIES_EXPANSION_I   SET     1
XX
**      ÍFilename:  libraries/expansion.i $
**      iRelease:  1.3 $
XX
**      external definitions for expansion.library
x*
**     .(C) Copyright 1986,1987,1988 Commodore-Amiga,  Inc.
**          Ali Rights Reserved
XX
EXPANSIONNAME   MACRO
                dc.b    'expansion.library1 ,0
                ENDM


;* flags for the AddDosNode0 call */
        BITDEF  ADN,STARTPR0C,0

        ENDC    j LIBRARIES_EXPANSION_I
```

```
        IFND    LIBRARIES_EXPANSIONBASE_I
LIBRARIES_EXPANSIONBASE_I       SET     1
XX
**      SFilename:  libraries/expansionbase.i $
XX      SRelease: 1.3 $
XX
**      library structure for expansion library
*X
XX      (C) Copyright 1987,1988 Commodore-Amiga, Inc.
XX          Ali Rights Reserved
XX

        IFND    EXEC_TYPES_I
        INCLUDE "exec/types.i"
        ENDC    j EXEC_TYPES_I

        IFND    EXEC_LIBRARIES_I
        INCLUDE "exec/libraries.i"
        ENDC    ; EXEC_LIBRARIES_I

        IFND    EXEC_INTERRUPTS_I
        INCLUDE "exec/interrupts. i"
        ENDC    ; EXEC_INTERRUPT5_I

        IFND    EXEC_SEMAPHORES_I
        INCLUDE "exec/semaphores.i"
        ENDC    ; EXEC_SEMAPHORES_I

        IFND    LIBRARIES_CONFIGVARS_I
        INCLUDE "J.ibraries/conf igvars. i"
        ENDC    ; LIBRARIES_CONFIGVARS_I

TOTALSLOTS      EQU     256

 STRUCTURE      ExpansionInt,0
    UWORD               ei_IntMask      ; mask for this list
    UWORD               ei_ArrayMax     ; current max valid index
    UWORD               ei_ArraySize    ; allocated size
    LABEL               ei_Array        ; actual data is after this
    LABEL               ExpansionInt_SIZEOF

 STRUCTURE      ExpansionBase,LIB_SIZE
    UBYTE               eb_Flags
    UBYTE               eb_pad
    ULONG               eb_ExecBase
    ULONG               eb_SegList
    STRUCT              eb_CurrentBi ndi ng,CurrentBi ndi ng_SIZEOF
    STRUCT              eb_BoardList,LH_SIZE
    STRUCT              eb_MountLi st,LH_SIZE
    STRUCT              eb_AllocTable,TOTALSLOTS
    STRUCT              eb_BindSemaphore,SS_SIZE
    STRUCT              eb_Int2List,IS SIZE
    STRUCT              eb_Int6List,IsIsiZE
    STRUCT              eb_Int7List,IS_SIZE
    LABEL               ExpansionBase_SIZEOF


; error codes
EE_LASTBOARD    EQU     40      ; could not shut him up
EE_NOEXPANSION  EQU     41      ; not enough expansion mem; board shut up
EE_NOBOARD      EQU     42      ; no board at that address
EE_NOMEMORY     EQU     42      ; not enough normál memory
```

```
; flags
        BITDEF  EB,CL0GGED>0    > someone could not be shutup
        BITDEF  EB,SHORTMEM,1   ; ran out of expansion mem

        ENDC    ; LIBRARIES_EXPANSIONBASE_I
```

```
DE_LOWCYL       EQU     9       ; starting cylinder. typically 0
DE JJPPERCYL    EQU     10      ; max cylinder.  drive specific
DE_NUMBUFFERS   EQU     11      ; starting # of buffers.  typically 5
DE"MEMBUFTYPE   EQU     12      ; type of mem to allocate for buffers.
DE_BUFMEMTYPE   EQU     12      ; same as abovej better name
                                ; 1 is public, 3 is chip, 5 is fást
DE_MAXTRANSFER  EQU     13      ; Maximum number of bytes to transfer at a time
DE_MASK         EQU     14      ; Address Mask to block out certain memory
DE_BOOTPRI      EQU     15      ; Boot priority for autoboot
DE_DOSTYPE      EQU     16      ; ASCII (HEX) string showing filesystem type
                                ; 0X444F5300 is old filesystem,
                                ; 0X444F5301 is fást file system


*
* The file system startup message is linked intő a device node's startup
* field.  It contains a pointer to the above environment, plus the
* information needed to do an exec OpenDeviceC).
*

 STRUCTURE FileSysStartupMsg,0
    ULONG       fssm_Unit       ; exec unit number for this device
    "BSTR       fssm_Device     > null terminated bstring to the device name
    BPTR        fssm_Environ    > ptr to environment table (see above)
    ULONG       fssm_Flags      ; flags for OpenDeviceC)
    LABEL       FileSysStartupMsg_SIZEOF


* The include file "libraries/dosextens.h" has a DeviceList structure.
* The "device list" can have one of three different things linked ontó
* it.  Dosextens defines the structure for a volume.  DLT_DIRECTORY
* is for an assigned directory.  The following structure is for
* a dos "device" (DLT_DEVICE).

 STRUCTURE DeviceNode>0
    BPTR        dn_Next         ; singly linked list
    ULONG       dn_Type         ; always 0 for dos "devices"
    CPTR        dn_Task         ; standard dos "task" field.  If this is
                                ;     null when the node is aceesses, a task
                                ;     will be started up
    BPTR        dnJLock         ; not used for devices -- leave null
    BSTR        dn_Handler      ; filename to loadseg (if seglist is null)
    ULONG       dn_StackSize    ; stacksize to use when starting task
    LONG        dn_Priority     ; task priority when starting task
    BPTR        dn_Startup      ; startup msg: FileSysStartupMsg for disks
    BPTR        dn_SegList      ; code to run to start new task (if necessary).
                                ;     if null then dn_Handler will be loaded.
    BPTR        dn_GlobalVec    ; BCPL global vector to use when starting
                                ;     a task.  -1 means that dn_SegList is not
                                ;     for a bcpl program, so the dos won't
                                ;     try and construct one.  0 teli the
                                ;     dos that you obey BCPL linkage rules,
                                ;     and that it should construct a global
                                ;     vector for you.
    BSTR        dn_Name         ; the node name, e.g. '\3','D' ,'F','3'
    LABEL       DeviceNode_SIZEOF

        ENDC    ; LIBRARIES_FILEHANDLER_I
```

```
        IFND    LIBRARIES_ROMBOOT_BASE_I
LIBRARIES_ROMBOOT_BASE_I        SET     1
x*
**      SFilename:  libraries/romboot_base.i $
**      SRelease:  1.3$
xx
xx
**      (C) Copyright 1987,1988 Commodore-Amiga,  Inc.
**          Ali Rights Reserved
xx

    IFND    EXEC_TYPES_I
    include "exec/types.i"
    ENDC
    IFND    EXEC_NODES_I
    include "exec/nodes.i"
    ENDC
    IFND    EXEC_LISTS_I
    include "exec/lists.i"
    ENDC
    IFND    EXEC_LIBRARIES_I
    include "exec/libraries.i"
    ENDC
    IFND    EXEC_EXECBASE_I
    include "exec/execbase.i"
    ENDC '
    IFND    EXEC_EXECNAME_I
    include "exec/execname.i"
    ENDC

 STRUCTURE  RomBootBase,LIB_SIZE
        APTR    rbb_ExecBase
        STRUCT  rbb_BootList,LH_SIZE
        STRUCT  rbb_Reserved,16                 ; for future expansion
    LABEL   rbb_SIZEOF

 STRUCTURE BootNode,LN_SIZE
        UWORD   bn_Flags
        CPTR    bn_DeviceNode
        LABEL   BootNode_SIZEOF

ROMBOOT_NAME:  MACRO
        DC.B    'romboot.library',0
        DS.W    0
        ENDM

        ENDC    > LIBRARIES_ROMBOOT_BASE_I
```

```
Directory "Lattice_C_5.0.4:source/*.a" on Sunday 30-Sep-90
c.a                          10501 ----rwed 07-Nov-88 14:53:14
catch.a                      24062 ----rwed 07-Nov-88 14:53:24
cback.a                      13665 ----rwed 07-Nov-88 14:53:22
cxffp.a                       2912 ----rwed 07-Nov-88 14:53:17
ffptran.a                     3169 ----rwed 07-Nov-88 14:53:22
QVS.a                        11935 ----rwed 07-Nov-88 14:53:21
ucxovf.a                       857 ----rwed 07-Nov-88 14:53:24
7 files - 148 blocks - 67101 bytes
```

```
Directory "Lattice_C_5.0.4:source" on Sunday 30-Sep-90
c.a                          10501_____rwed 07-Nov-88 14:53:14
l files - 23 blocks - 10501 bytes
*
* C initial startup procedure under AraigaDOS
*
* Use the following coramand line to raake c.o
* asm -u -iINCLUDE: c.a
*
* Use the following command line to make cres.o
* asm -u -dRESIDENT -iINCLUDE: -ocres.o c.a
*
        INCLUDE "exec/types.i"
        INCLUDE "exec/alerts.i"
        INCLUDE "exec/nodes.i"
        INCLUDE "exec/lists.i"
        INCLUDE "exec/ports.i"
        INCLUDE "exec/libraries.i"
        INCLUDE "exec/tasks.i"
        INCLUDE "exec/memory.i"
        INCLUDE "exec/execbase.i"
        INCLUDE "libraries/dos.i"
        INCLUDE "1ibrari es/dose xtens.i"
        INCLUDE "workbench/startup.i"
        INCLUDE "exec/funcdef.i"
        INCLUDE "exec/exec_lib.i"
        INCLUDE "libraries/dos_lib.i"

MEMFLAGS EQU    MEMF_CLEAR+MEMF_PUBLIC
AbsExecBase EQU 4

; somé usefull macros:

callsys macro
        CALLLIB _LVO\1
        endm

        xdef    XCEXIT                  * exit(code) is standard way to leave C.
        xdef    3XCEXIT

        xref    LinkerDB                * linker defined base value
        xref    _BSSBAS                 * linker defined base of BSS
        xref    _BSSLEN                 * linker defined length of BSS
        IFD     RESIDENT
        xref        RESLEN
        xref        RESBASE
        xref    NEWDATAL
        ENDC

*       library references

        section textjcode

        xref    _main                    * Name of C program to start with.
        xref    MemCleanup               * Free all allocated memory
        xref    __fpinit                 * initialize floating point
        xref    __fpterm                 * terrainate floating point

í.tart:
        move.1  aO,a2                   * savé command pointer
        move.1  dO,d2                   * and command length
```

```
        lea     LinkerDB,a4             * load base register

        IFND    RESIDENT
        lea     _BSSBAS,a3              * get base of BSS
        moveq   #0,dl
        move.l  #_BSSLEN>dO             * get length of BSS in longwords
        bra.s   clr_lp                  * and clear for length given
clr_bss raove.1 dl,(a3)+
clr_lp  dbf     dO>clr_bss
        ENDC

        IFD     RESIDENT
        move.1  AbsExecBase.W,a6

        movem.1     dO-dl/aO-a2,-(a7)
        sub.l         #RESBASE,a4
        move.1        #RESLEN,dO
        move•1        #MEMFLAGS,dl
        callsys       AllocMem
        tst.l         dO
        beq. w        abort
        move.1        dO,aO
        move.1        dO,a2
;a2 now has difference
        move.1        dO,al
        move.1        #NEWDATAL,dO
;copy data over
c;py:   move.1        (a4)+,(aO)+
        subq.1        #l>dO
        bne           cpy
;a4 now points at number of relocs
        move.1        Ca4)+,dO
reloc:  beq.s         nreloc
        move.1        al,aO
        add.l         Ca4)+>aO                ;aO now has add of reloc
        add.l         (aO),a2
        move.l        a2,(aO)
        move.1        al,a2                 Jrestore offset
        subq.1        #l,dO
        bra.s         reloc

nreloc: move.1        al,a4                   > set up new base register
        add.l         #RESBASE,a4
        movem.1       (A7)+,dO-dl/aO-a2
        ENDC

        move.l  AbsExecBase.W}a6
        move.l  a6)SysBase(A4)
        move.l  a?,_StackPtr(A4)        * Savé stack ptr
        clr.l   WBenchMsg(A4)

*------ get the address of our task
        move.l  ThisTaskCaG)>A3

*-----  clear any pending signals
        moveq   #0sd0
        move.1  #$00003000,dl
        callsys SetSignal

*------ are we running as a son of Workbench?
        move.1  pr_CurrentDir(A3),curdir(A4)
```

```
        tst.l   pr_CLI(A3)
        beq     fromWorkbench

*=======================================================================
*====== CLI Startup Code ==============================================
*=======================================================================
*
* Entry: D2 = coramand length
*        A2 = Command pointer
fromCLI:
        move.1  a7>DO           * get top of stack
        sub.l   4(a7),D0        * corapute bottom
        add.l   #128,DO         * allow for parms overflow
        move.1  DO,_base(A4)    * savé for stack checking
*------ attempt to open DOS library:
        bsr.w     openDOS

*------ find command name:
        move.l  pr_CLI(a3),aO
        add.l   aO,aO           * bcpl pointer conversion
        add.l   aO,aO
        move.l  cli_CommandName(aO)ɔal
        add.l   alsal           * bcpl pointer conversion
        add.l   al,al

*------ collect parameters:
        move.l  d2,dO                   * get command line length
        moveq.l #0,dl
        move.b  (al)+,dl
        move.1  al>_ProgramName(A4)
        add.l   dl»dO                   * add length of command name
        addq.l  #l,dO                   * allow for space after command

        clr. w  -(A7)                   * set null terminator for command line
        addq.1  #1,DO                   * force to evén number of bytes
        andi.w  #$fffe,DO               *Cround up)
        sub.l   DO,A7                   * make room on stack for command line
        subq.1  #2,D0
        clr. w  0(A7,D0)

*------ copy command line ontó stack
        move.1  d2,dO                   * get command line length
        subq.l  #l,dO
        add.l   dl,d2

copy_line:
        move.b  0(A2,D0.W),0(A7,D2.W)   * copy command line to stack
        subq.l  #l,d2
        dbf     dO,copy_line
        move.b  r ',0(a7,d2.w)          * add space between command and parms
        subq.1
                #l,d2
copy_cmd:
        move.b  0(al>d2.w)50Ca7,d2.w)   * copy command name to stack
        dbf     d2,copy_cmd
        move.1  A7,A1
        move.1  A1,-CA7)                * push command line address
        bra     main                    * call C entrypoint

*=======================================================================
ÍK======Workbench Startup Code=====
```

```
*=============================================================================

fromWorkbench:

        move.l  TC_5PL0WER(a3),_base(A4) * set base of stack
        moveq   #127,dO
        addq.l  #l,dO                    * Efficient way of getting in 128
        add.l   dO5_base(A4)             * allow for parms overflow
*------ open the DOS library:
        bsr.w     openDOS

t_______we are now set up.  wait for a raessage from our starter
        lea     pr_MsgPort(A3),aO        * our process base
        callsys WaitPort
        lea     pr_MsgPort(A3),aO        * our process base
        callsys GetMsg
        move.1  dO,WBenchMsg(a4)
        move.1  dO,-(SP)
*
        move.1  dO,a2                    * get first argument
        move.1  sm_ArgList(a2),dO
        beq.s   do_cons
        move.1  D0SBase(a4)>a6
        move.1  dO,aO
        move.1  wa_Lock(aO))dl
        move.1  dl5curdir(A4)
        callsys CurrentDir
do_cons:
        move.l   sm_ToolWindow(a2)>dl    * get the window argument
        beq.s   do_main
        move.1  #MODE_OLDFI LE,d2
        callsys Open
        move.l  dO>stdin(a4)
        beq.s   do_main
        lsl.l   #2,dO
        move.1 dO j aO
        move.1   fh_Type(aO)> pr_ConsoleTask(A3)
do_main:
        move.l  WBenchMsg(A4)5a0         * get address of workbench message
        move.l  aO,-(a7)                 * push argv
        pea     NULL(a4)                 * push argc
        move.l  sm_ArgList(aO),aO        * get address of arguments
        mo ve.1 wa_Name(aO) > _ProgramName(A4)       * get name of program

*=============================================
*_______common code_________
*=============================================

main    jsr_______fpinit(PC)             * Initialize floating point
        jsr     _main(PC)                * call C entrypoint
        moveq.1 #0,d0                    * set successful status
        bra.s   exit2
*

XCEXIT:
        move.l  4(SP),dO                 * extract return code
aXCEXIT:
exit2:
        move.1  dO,-C a7)
        move.l  _ONEXIT(A4),dO           * exit trap function?
        beq.s   exit3
```

```
        move.l  dO,aO
        jsr     CaO)
exit3   jsr     MemCleanup(PC)          * cleanup leftover memory alloc.
        move.l  AbsExecBase.W,a6
        move.l  DOSBase(A4)>al
        callsys CloseLibrary            * close Dos library

        jsr     __fpterm(PC)            * clean up any floating point

done_lc:
t_____if ye ran from CLI, skip workbench cleanup:
        tst.l   WBenchMsg(A4)
        beq.s   exitToDOS
        move.l  stdin(a4),dl
        beq.s   done_4
        callsys Close
done_4:

i_____return the startup message to our parent
*       we forbid so workbench can't UnLoadSeg() us
*       before we are done:
        move.l  AbsExecBase.W>A6
        callsys Forbid
        move.l  VBenchMsg(a4)>al
        callsys ReplyMsg

*_____this rts sends us back to DOS:
exitToDOS:
        IFD     RESIDENT
        move.l  #RESLEN,dO
        move.l  a4,al
        sub.l   #RESBASE,al
        raove.1 AbsExecBase.W>a6
        callsys FreeMem
        ENDC

        MOVE.L  (A7)+,D0
        movea.l _StackPtr(a4)»SP        * restore stack ptr
        rts

        IFD     RESIDENT
abort:
        movem.1     Ca7)+>dO-dl/aO-a2
        rts
        ENDC

*-------------------------------------------------------------------------
noDOS:
                moveq.1 #100,dO
                bra     exit2



*-------------------------------------------------------------------------
»  Open the DOS library:

openDOS
                lea     DOSName(PC),Al
                raoveq.l#0,D0
                callsys OpenLibrary
                raove.1 D0,D0SBase(A4)
```

```
                beq     noDOS
                rts

DOSName         de.b    'dos.1ibrary',0

        section MERGED,BSS

        xref    DOSBase

        xdef    NULL,SysBase,WBenchMsg
        xdef    curdir,jnbase,_mnext,jnsize,_tsize
        xdef    _oserr,_OSERR,_FPERR,_SIGFPE,_SIGINT,_ONERR,_ONEXIT,_ONBREAK
        xdef    _SIGINT
        xdef    _ProgramName,_StackPtr,_base

*
NULL            ds.b    4               *
_base           ds.b    4               * base of stack
jnbase          ds.b    4               * base of memory pool
_mnext          ds.b    4               * next available memory location
_jnsize         ds.b    4               * size of memory pool
_tsize          ds.b    4               * totál size?
__oserr         equ     *
_OSERR          ds.b    4
_FPERR          ds.b    4
_SIGFPE         ds.b    4
_SIGINT         ds.b    4
_ONERR          ds.b    4
_ONEXIT         ds.b    4
_ONBREAK        ds.b    4
rurdir          ds.b    4
SysBase         ds.b    4
WBenchMsg       ds.b    4
_StackPtr       ds.b    4
stdin           ds.b    4
_ProgramName    ds.b    4
                END
```

```
Directory "Lattice_C_5.0.4:source" on Sunday 30-Sep-90
catch.a                    24062_____rwed 07-Nov-88 14:53:24
1 fi les - 51 blocks - 24062 bytes
*
* C initial startup procedurp under AraigaDOS
*
* Use the following command line to make catch.o
* asm -u -iINCLUDE: catch.a
*
* Use the following command line to make catchres.o
* asm -u -iINCLUDE:  -dRESIDENT -ocatchres.o catch.a
*
* Use the following command line to make catchresnr.o
* asm -u -iINCLUDE:  -dRESIDENT -dNOREQ -ocatchresnr.o catch.a
i

        INCLUDE "exec/types.i"
        INCLUDE "exec/alerts.i"
        INCLUDE "exec/nodes.i"
        INCLUDE "exec/lists.i"
        INCLUDE "exec/ports.i"
        INCLUDE "e xec/1i brar i es.i"
        INCLUDE "exec/tasks.i"

        INCLUDE "exec/memory.i"
        INCLUDE "exec/execbase.i"
        INCLUDE "libraries/dos.i"
        INCLUDE "libraries/dosextens.i"
        INCLUDE "workbench/startup.i"
        INCLUDE "intuition/intuition.i"
        INCLUDE "e xec/f uncdef.i"
        INCLUDE "exec/exec_li b.i"
        INCLUDE "libraries/dos_lib.i"

VERSION         equ     1
REVISION        equ     0

CATCH           set     1
        IFND NOREQ
AUTOREQ         set     1
        ENDC


MEMFLAGS EQU    MEMF_CLEAR+MEMF_PUBLIC
AbsExecBase EQU 4

callsys macro
        CALLLIB _LVO\1
        endm

        xdef    XCEXIT                          ; exit(code) is standard way to 1
eave C.
        xdef    3XCEXIT

        xref    LinkerDB                        5 linker defined base value
        xref    _BSSBAS                         ; linker defined base of BSS
        xref    _BSSLEN                         ; linker defined length of BSS

        IFD     RESIDENT
        xref    RESLEN
        xref    RESBASE
```

```
        xref    NEWDATAL
        ENDC

*       library references

        section   text,code

        xref    _main                   * Name of C program to start with.
        xref    MemCleanup
        xref    __fpinit                * initialize floating point
        xref    __fpterm                * terrainate floating point

start:
        movem.1 dl-d6/aO-a6,-(a7)
REGSIZE EQU     (6+7)*4
        lea     REGSIZE(a7),A5                  > determine old stack pointer
        raove.1 aO,a2                           ; savé command pointer
        move.1  dO5d2                           ; and command length
        lea     LinkerDB,a4                     ; load base register

        IFND    RESIDENT
        lea <   _BSSBAS,a3                      * get base of BSS
        raoveq  #O,dl
        move.1  #_BSSLEN>dO             * get length of BSS in longwords
        bra.s   clr_lp                  * and clear for length given
clr_bss move.1  dl»(a3)+
clr_lp  dbf     dO,clr_bss
        ENDC

        IFD '   RESIDENT
        move.1  AbsExecBase.W>a6
        movem.1     dO-dl/aO-a2,-(a7)
        sub.l           #RESBASE,a4
        move.1          #RESLEN,dO
        move.1          #MEMFLAGS,dl
        callsys         AllocMem
        tst.l           dO
        beq.w           abort
        move.1          dO,aO
        move.1          dO,a2
;a2 now has difference
        move.1          dO,al
        move.1          #NEWDATAL,dO
Jcopy data over
rpy:    move.1          Ca4)+,(aO)+
        subq.l          #l,dO
        bne             cpy
;a4 now points at number of relocs
        move.1          (a4)+>dO
reloc:' beq.s           nreloc
        move.1          al,aO
        add.l           (a4)+,aO                        ; aO now has add of reloc
        add.l           (aO),a2
        move.1          a2,(aO)
        move.1          al,a2                   > restore offset
        subq.1          #l,dO
        bra.s           reloc

nreloc: move.1          al,a4                   ; set up new base register
        add.l           #RESBASE,a4
        movem.1       (A7)+5d0-dl/a0-a2
```

```
        ENDC

        move.1  AbsExecBase.W> a6
        move.1  a6 >SysBase(A4)

        ifd     CATCH
        move.w  AttnFlags(a6)»Environment+2(a4) j savé copy for dump
        endc

        move.l  a7>_StackPtr(A4)              ; Savé stack ptr
        clr.l   WBenchMsg(A4>

*_____attempt to open DOS library:
        bsr.w   openDOS

        ifd     CATCH
         ifd     AUTOREQ
*_____attempt to open Intuition library:
                bsr.w           openlntui
         endc
        endc

*_____clear any pending signals
        raoveq #0 > dO
        move.l  #£00003000,dl
        callsys SetSignal

«_____get the address of our task
        raove.1 Thi sTask(a6)s A3

        ifd     CATCH
        move.l  A3,TaskID(a4)

s_____initialize exception handler
x_____Remember to preserve the old handler first
        move.1  TC_TRAPDATA(a3)>oldtrapdata(A4)
        move.1  TC..TRAPCODE(a3),dO             ; check current exception
        move.1  dO,ioldtrapcode(A4)
        swap    dO
        cmpi.w  #ifb,dO                         ", see if points to ROM
        ble.s   1$                              ; somebody else (debugger?) has v
ector
        move.l  #Exceptions TC_TRAPC0DE(a3)     ; install pointers to code
        move.1 a4 > TC_TRAPDATA(a3)             ;  ...and data
1$:
        endc

*_____are we running as a son of Workbench?
        move.1 pr_CurrentDi r (A3),curdi r(A4)
        tst.l   pr_CLI(A3)
        beq     fromWorkbench

*==========================================================================
*====== CLI Startup Code ==================================================
*==========================================================================
x
* Entry: D2 = command length
*        A2 = Command pointer
fromCLI:
        ifd     CATCH
        moveq   #-l,d0
```

```
        move.1   dO,Starter(a4)          ; non-zero means CLI
        raove.l  a5,StackTop(a4)
        endc

        move.1  a5jDO                    > get top of stack
        sub.l    4(a5),D0                > cotnpute bottom
        add.l    #128,DO                 ; allow for parras overflow
        move.l   DO5_base(A4)            j savé for stack checking

*------ fjncj comraand name:
        move.l   pr_CLI(a3),aO
        add.l    aO,aO           * bcpl pointer eonversion
        add.l    aO,aO
        move.l   cli_ComraandName(aO),al
        IFND     AUTOREQ
        move.1   cli_Standard0utput(a0))GConsole(a4)     * savé output fh
        ENDC
        add.l    al>al           * bcpl pointer eonversion
        add.1 al, al

*------ collect parameters:
        move.1   d2,dO                   * get command line length
        moveq.l #O,dl
        move.b   (al)+>dl
        move.1   al>_ProgramName(A4)
        add.l    dl,dO                   * add length of command name
        addq.l   #l,dO                   * allow for space after command

        clr.w    -(A7)                   * set null terminator for command line
        addq.l   #1,DO                   * force to evén number of bytes
        andi.w   #$fffe,DO               *(round up)
        sub.l    D0,A7                   * make room on stack for command line
        subq.l'  #2,D0

        clr.w    O(A7»DO)

*------ copy command line ontó stack
        move.1  d2,dO                    * get command line length
        subq.l   #l,dO
        add.l    dl,d2

copy_line:
        move.b   0(A2jD0.W),0(A7,D2.W)   * copy command line to stack
        subq.l   #l,d2
        dbf      dO,copy_line
        move.b   #'  ',0(a7,d2.w)   * add space between command and parms
        subq.l   #l,d2

copy_cmd:
        move.b   OCal,d2.w),0(a?,d2.w)   * copy command name to stack
        dbf      d2}copy_cmd
        move.1  A?>Al
        move.l   A1,-CA7)                * push command line address
        bra.w    main                    * call C entrypoint

*=================================================================
JK======WorkbenchStartupCode=============================
*=================================================================

fromWorkbench:
        move.1   TC_SPLOWERCa3),_baseCA4)        * set base of stack
```

```
        add.l   #128j_base(A4)            * allow for parms overflow

        ifd     CATCH
        move.l  TC_SPUPPER(a3),StackTop(a4)     ; set top of stack
        endc

*_______we are now set up.  wait for a message from our starter
        lea     pr_MsgPort(A3),aO       * our process base
        callsys WaitPort
        lea     pr_MsgPort(A3)>aO       * our process base
        callsys GetMsg
        move.1 dO > WBenchMsg(a4)
        move.l  dO,-CSP)
*
        move.l  dO,a2                   * get first argument
        raove.l sm_ArgList(a2),dO
        beq.s   do_cons
        move.1  DOSBase(a4)> a6
        move.1  dO,aO
        move.l  wa_Lock(aO)jdl
        move.l  dl>curdir(A4)
        callsys CurrentDir
do_cons:
        move.l  sm_ToolWindow(a2)>dl    * get the window argument
        beq.s   do_main
        move.1  #MODE_OLDFI LE,d2
        callsys Open
        move.l  dO,stdin(a4)
        beq.s   do_main
        lsl.l   #2,dO
        move.1  dO,aO
        move.1  fh_Type(aO)>pr_ConsoleTask(A3)
do_main:
        move.l  WBenchMsg(A4),aO        * get address of workbench message
        move.l  aO,-(a7) -              * push argv
        pea     NULL(a4)                * push argc
        move.l  sm_ArgList(aO)>aO       * get address of arguments
        move.l  wa_Name(aO),_ProgramName(A4)    * get name of program

*=======================================
*_______common code_________
*=======================================

main    jsr_______fpinit(PC)            * Initialize floating point
        jsr     _main(PC)               * call C entrypoint
        moveq.l #0,d0                   * set successful status
        bra.s   exit2
*

XCEXIT:
        move.l  4(SP)>dO        * extract return code
3XCEXIT
exit2:
        move.l  dO,-(a?)

        move.l  _ONEXIT(A4),dO  * exit trap function?
        beq.s   exit3
        move.l  dO>aO
        jsr     (aO)
exit3:
        jsr     MemCleanup(PC)          ; cleanup leftover memory alloc.
```

```
        move.1  AbsExecBase.W>a6

*------ Restore the original exception handler
        move.l  ThisTask(a6)>A3
        raove.1 oldtrapdata(A4),TC_TRAPDATA(a3)
        raove.1 oldtrapcode(A4),TC_TRAPC0DE(a3) » check current exception

        move.1  DOSBase(A4)»al
        callsys CloseLibrary            ; close Dos library

        ifd     CATCH
         ifd     AUTOREQ
                move.l  IntuiBase(a4)>al
                callsys         CloseLibrary            s close Intuition library
         endc
        endc

        jsr     __fpterm(PC)            * clean up any floating point

done_le:
*------ if we ran from CLI, skip workbench cleanup:
        tst.l   WBenchMsg(A4)
        beq.s   exitToDOS
        move.l  stdin(A4),dl
        beq.s   done_4
        callsys Close
done_4:

*------ return the startup message to our parent
*       we forbid so workbench can't UnLoadSegC) us
*       before we are done:
        raove.1 AbsExecBase. W> A6
        callsys Forbid
        move.1 WBenchMsg(a4 )} al
        callsys ReplyMsg

*       this rts sends us back to DOS:
exitToDOS:
        IFD     RESIDENT
        move.1  #RESLEN,dO
        move.1  a4,al
        sub.l   #RESBASE,al
        move.1  AbsExecBase.W»a6
        callsys FreeMem
        ENDC



        MOVE.L  (A?)+5DO
        movea.l _StackPtr(a4),SP        * restore stack ptr
        movem.1 (a7)+,dl-d6/aO-a6
        rts

        IFD     RESIDENT
abort:
        movem.1     <a7)+sd0-dl/aQ-a2
        rts
        ENDC

a_____----------------------------------------------------------------
noDOS:
```

```
                moveq.l  #100,d0
                bra.w    exit2


*-------------------------------------------------------------------------
*       Open the DOS library:

openDOS
                lea      DOSName(PC),Al
                raoveq.l#0,D0
                callsys  OpenLibrary
                move.l   D0,DOSBase(A4)
                beq.s    noDOS
                rts
POSNarae        dc.b     'dos.library'>0

        ifd     CATCH
*-------------------------------------------------------------------------
*       Open the Intuition library:

openlntui:
                lea      IntuiName(PC),Al
                moveq.l  #0,D0
                caJlsys  OpenLibrary
                move.l   D0,IntuiBase(A4)
                beq.s    noDOS
                rts
IntuiName       dc.b     'intuition.library',0

*-------------------------------------------------------------------------
*       The Exception Handler - catches GURUs and exits Csemi)cleanly
Exception:
        move.l  AbsExecBase.W,aO
        move.l  ThisTask(a0)5a0
        move.l  TC_TRAPDATA(aO),aO              ;  ...anddata

        move.1  (a7)+,d0                ; get exception # from stack
        move.1  dO,GURUNum(aO)          ; and savé it
        cmpi.1  #3,dO                   j ADDRESS or BUS error?
        bgt.s   2Í                      ; no, skip adjustment
        btst    #0,Environment+3(aO)    ; is it 68010 or 68020?
        beq.s   1$                      ", 0 means NO
        bset    #7,8(a7)                •» set Rerun flag
        bra.s   2$
1$:
        addq.1  #8,a7                   5 adjust for 68000
21:
        move,1  2(a7),dO                > get PC at crash •
        move.1  dO,GURUAddr(aO)         ; and savé it
        move.1  #GURUExit,2(a7)         ; use our own exit point
        rte

*-------------------------------------------------------------------------
*       The Exception exit routine - write ' PGTB[1] IFF chunk to file
*       [1] SnapShot.TB[1]  in current directory, then exit to system.

GURUExit:
        movem.l dO-d7/aO-a?>-(sp)       ; savé all registers
        move.1 AbsExecBase.W> a6        ; make sure we are working with Exec
        callsys GetCC                   5 safe way - works with all CPUs
        move.l  ThisTask(a6),a3
```

```
        mo ve.1 TC_TRAPDATA(a3)»a4      ; make sure we have a valid # in a4
        move.l  d(^Flags(a4)
        raovem.1(sp)+jdO-d7             ', savé area
        movem.1 dO-d7)DDump(a4)
        movem.1 (sp)+,dO-d7             •> savé data reg contents
        movera.1 dO-d7>ADump(a4)
        tst.l   StackPtr(a4)            ; savé address reg contents
        bne     GExitl                  i if there's something there
        lea     TempStore(a4)>aO        í ...we' ve been here before!
        move.1  aO>TempAddr(a4)         j calculate addr of TempStore
        move.l  A7Store(a4)>dO          > ...and savé for later access
        move.1  dO,StackPtr(a4)         '> make sure we have proper TOS
        moveq #0 > dO                   ; ... and savé i t
        mo ve.1 _ProgramName(a4),aO     s find length of program name

        ifd     AUTOREQ
        mo ve.1 aO > PName(a4)
        endc

        subq.1  #l,aO
        move.b  (a0)5d0
        addq.1  #4,d0                   ; adjust for shift
        lsr.l   #2,dO
        move.1  dO,NameLen(a4)          ; store length
        add.l   dO,FAILlen(a4)          ; and sub-chunk totál

        moveq   #0,d0                  ; clear dO for use
        lea     VBlankFrequency(a6)>aO  ; set yp aO to find correct data
        move.b  (a0)+5d0               ; get just in case
        move.1  d0>VBlankFreq(a4)       ; ...so we can figure what
        move.b  (a0),d0          ; ...type of machine
        move.1  d0,PowerSupFreq(a4)     ; ...we're working on

        lea     start-4(pc),aO          ; get seglist ptr
        moveq   #-l,dO                  ; always at least 1
2$:
        addq.1  #l,dO
        move.1  (aO),dl          ; find end of list
        beq.s   3S
        lsl.l   #2,dl                   5 BPTR!!!!!
        move.1  dl,aO
        bra.s   2$
3$:
        add.1   dO,SegCount(a4)  ; store # of seglist pointers
        lsl.l   #l,dO                   ; multiply by 2 for longword count
        add.l   dÖ,FAILlen(a4)          ; and sub-chunk length

        move.l  StackTop(a4)jd0  ; get top of stack
        sub.l   StackPtr(a4)>dO  ; find number of bytes used
        addq.l  #4,dO                   ; adjust for longword conversion
        lsr.l   #2,dO                   ; convert from bytes to long
        raove.l dO»Stackl_en(a4)  ; and savé
        add.l   dO,s21en(a4)            ; and sub-chunk totál

        move.l  a5,-(sp)                ; savé a5 for later
        callsys Forbid                  ; don't let ' em change while we ask
        move.l  MemList+LH_HEAD(a6),dO  ; first node in MemList
checkchip:
        move.1  dO,a5                   ; move node address to address reg
        move.w  MH_ATTRIBUTES(a5),d4    > get node attributes
        btst    #MEMB_CHIP,d4           ; is it chip"?
```

```
        beq.s   checkfast               ; no, go on
        lea     chipAvail(a4),a3
        bsr.w   Addlt
checkfast:
        btst    #MEMB_FAST,d4           ; is it fast?
        beq.s   next                    ; no, go on
        lea     fastAvail(a4),a3
        bsr.w   Addlt
next;
        move.1  LN_SUCC(a5),dO          ; get address of next node
        bne.s   checkchip               ; ...and loop back if valid
        callsys Permit                  ; allow others access again
        raove.1 #MEMF_CHIP+MEMF_LARGEST,dl ; to find largest hunk in chip ram
        callsys AvailMem
        raove.1 dO>chipLargest(a4)      ; store
        move.1  #MEMF_FAST+MEMF_LARGEST,dl ; to find largest hunk in fast ram
        callsys AvailMem
        move.1  dO,fastLargest(a4)      j store
        move.1  (sp)+»a5                ; and restore a5

        ifd     AUTOREQ
        moveq   #0,d0                   ; PosFlag
        move.1  dO,dl                   > NegFlag
        move.1  dO,aO                   > 0 means use current window
        lea     ITextl(a4),al           ; Body Text
        lea     IText5(a4),a2           ; Positive Gadget Text
        lea     IText6(a4),a3           ; Negative Gadget Text
        moveq   #l,d2
        lsl.l   #8,d2                   ; quick way to set Width
        moveq   #76,d3                  ; Height
        move.1  IntuiBase(a4),a6        ; get intuition library pointer
        jsr     -$15c(A6)               ; callsys       AutoRequest
        move.1  AbsExecBase.W,a6
        tst.l   dO                      ; save SnapShot?
        beq.s   GExit2                  ; no> just exit
        endc

        move.1  D0SBase(a4),a6
        lea     DumpName(a4),aO  ; get name of output file
        move.1  aO,dl
        move.1  #MODE_NEWFILE,d2        ; create new file
        callsys Open
        bne.s   4$
        lea     DumpPath(a4),aO  ; if error in current dir, try DF1:
        mdve.1  aO,dl
        move.1  #M0DE_NEWFILE,d2
        callsys Open
        bne.s   4$
        move.b  #'O',DumpPath+3(a4)     ; still error?  Try DFO:
        lea     DumpPath(a4),aO
        move.1  aO,dl
        move.1  #M0DE_NEWFILE,d2
        callsys Open

        ifnd    AUTOREQ
        bne.s   4$                      ; if no error» continue (finally!)
        move.1  GConsole(a4),dl
        beq.w   GExit2
        lea     failmsg(a4),aO
        move.1  aO,d2
        move.1  #2.3,d3
```

```
        callsys Write
        endc

        bra.w   GExit2                  ; else, print msg & DIE gracefully
4$:
        move.1  dO,d5                   > savé file handle for Write
        move.1  dO,fp(a4)               j ...and in a safe place for later
        move.1  d5,dl                   ; get file handle
        lea     PGTB(a4),aO             ; first part of fixed
        move.1  aO,d2
        move.1  #chunk_len_l,d3         j length of first
        callsys Write                   ; ...since it gets written over

        move.1  d5,dl                   > get file handle
        move.1  _ProgramName(a4)s d2    > get address of program name
        move.1  NameLen(a4)}d3          > get # longs in program name
        lsl.l   #2,d3                   ; ..and convert to bytes
        callsys Write

        move.1  d5 dl                   ; get file handle
        lea     Environment(a4)5a0      ; second part of fixed
        move.1  aO,d2
        move.1  #chunk_len_2>d3         ; length of second part
        callsys Write

        lea     start-8(pc),aO          \ address of seglist (size of seg)
        move.1  (aOH,dO                 ; segsize
        move.1  dO>TempStore+4(a4)      j savé it
        move.1  aO5TempStore(a4)        > store first number
        move.1  SegCount(a4))d4
51:
        move.1  d5,dl                   ; get file handle
        move.1  TempAddr(a4)5d2  ; address of write buffer
        moveq   #TempSize}d3            ; size of segment pointer
        callsys Write
        move.1  TempStore(a4)>aO        ; retrieve pointer
        move.1  (aO),dO          \ get next seg pointer
        lsl.l   #2,dO                   ; adjust
        move.1  dO,TempStore(a4)        ; ..and savé
        move.1  dO,aO
        move.1  -4(aO),dO               J get segsize
        move.1  dO,TempStore+4(a4)      ; ...and savé it
        subq.1  #l,d4                   ; done yet?
        bne.s   5$                      ; no, do next

        tst.l   _FMEM(a4)               ; do they want memory reported?
        beq.s   55$                     ; no> forget it
        move.1  d5,dl
        lea     subFMEM(a4),aO
        move.1  aO,d2
        move.1  #FMEMlen,d3
        callsys Write

55$:
        move.1 d5 > dl                  ; (get the idea?)
        lea     subREGS(a4)}a0          ; third part of fixed
        move.1 aO > d2
        move.1  #chunk_len_3>d3         ; length of third
        callsys Write

        move.l  StackLen(a4)»dO  ; get length of stack used
```

```
        cmpi.1  #2048,dO                ', > 8k ?
        bgt.s   6$
        move.1  d5,dl                   ; yes, dump two chunks
        lea     STAK2(a4),aO
        move.1  aO,d2                   ; whole stack chunk
        moveq   #STAK21en,d3
        callsys Write                   ; length of fixed part

        move.1  d5,dl
        move.1  StackPtr(a4),d2  ; address of stack
        move.1  StackLen(a4),d3  ; get # longwords on stack
        lsl.l   #2,d3                   ; ..and convert to bytes
        callsys Write
        bra.s   7$
6$:
        move.1  d5,dl
        lea     STAK3(a4),aO            ; top4k chunk
        move.l  aO,d2
        moveq   #STAK31en,d3            ; length of fixed part
        callsys Write

        move.1 d5, dl
        move.1  StackTopC a4)> d2  ; find top of stack
        sub.l   #4096,d2                ; find top-4k
        move.l  #4096,d3                \ # bytes to write
        callsys Write

        move.1  d5,dl
        lea     STAK4(a4),aO            ; bottom4k chunk
        move.1  aO,d2
        moveq   #STAK41en,d3            ; length of fixed part
        callsys Write

        move.1  d5,dl
        move.1  StackPtr(a4),d2  ; current stack address
        move.1  #4096,d3                ; # bytes to write
        callsys Write  -
7$:
        move.1  _STAKOffset(a4),d3
        beq.s   8$
        lsr.l   #2,d3
        addq.1  #l,d3
        move.1  d3,_5TAKOffset(a4)
        addq.l  #l,d3
        move.1  d5,dl
        lea     STAK5(a4),aO
        move.1  aO,d2
        raoveq  #STAK51en,d3
        callsys Write

        move.1  d5,dl
        move.1  StackPtr(a4),d2
        move.1  _STAKOffset(a4),d3
        subq.1  #l,d3
        move.1  StackLen(a4),d4
        cmp. 1  d3,d4
        bge.s   75$
        move.1  StackLen(a4),d3
75$:
        lsl.l   #2₅d3
        callsys Write
```

```
8$:
        tst.l   _0NGURIKA4)             ; user GURU function?
        beq.s   9$
        move.1  d5,-(sp)
        move.1  d5,dl
        lea     UDAT(a4),aO
        move.1  aO,d2
        move.1  #UDATlen,d3
        callsys Write
        move.1  d5,dl
        moveq   #0,d2                   > zero offset
        moveq   #l,d3                   ; ...from EOF
        callsys Seek
        move.1  dO,SeekStore(a4)
        move.1  _ONGURU(a4),aO
        jsr     (aO)
        addq.l  #4>sp
9$:
        move.1  fp(a4),d5
        move.1  d5,dl
        moveq   #0sd2                   ; offset from EOF
        moveq   #l,d3                   ; OFFSET_END
        callsys Seek                    ; Seek returns OLD position
        move.1  dO,dl
        andi.1  #3,dl                   ; did user write evén longwords?
        beq.s   10$                     ; Yep!  Nice Humán.
        move.1  dl,d6                   ; Nope, savé for later.
        clr.l   TempStore(a4)           ; clear temp storage
        move.1  d5,dl
        move.1  TempAddr(a4),d2
        moveq   #4,d3
        sub.l   d6,d3                   ; find how many NULLs to pad
        callsys Write
        bra.s   9$
10$:
        tst.l   SeekStore(a4)           ; did we write UDAT?
        beq.s   11$                     ; nope!
        sub.l   SeekStore(a4),dO        ; find length of UDAT section
        lsr.l   #2,dO                   ; adjust to longwords
        move.1  dO>TempStore(a4)        ; savé UDAT length for write
        move.1  d5,dl
        move.1  SeekStore(a4),d2        ; find where to write it
        subq.1  #4,d2
        moveq   #-l,d3                  J OFFSET_BEGINNING
        callsys Seek
        move.1  d5,dl
        move.1  TempAddr(a4))d2
        move.1  #4,d3
        callsys Write                   ; write length of UDAT field to file
11$:
        move.1  d5,dl
        moveq   #0,d2                   ; offset to 'Length' field
        moveq   #l,d3                   ; OFFSET_END
        callsys Seek                    ; make sure we are at end of file
        raove.1 d5,dl
        moveq   #4,d2                   ; offset to 'Length1 field
        moveq   #-l,d3                  ; OFFSET_BEGINNING
        callsys Seek
        subq.1  #8,dO                   ; adjust totál length
        lsr.l   #2,dO                   ; adjust to longwords
        move.1  dO,TempStore Ca4)       ; savé for write
```

```
        move.1 d5 > dl
        move.l  TempAddr(a4)}d2
        raove.1 #4,d3
        callsys Write                   > write 'Length1 field
GExitl:
        move.l  fp(a4),dl
        beq.s   GExit2
        move.1  DOSBase(a4)> a6
        callsys Close

        ifnd    AUTOREQ
        move.1  GConsole(a4)5dl
        beq.s   GExit2
        lea     success(a4),aO
        move.1  aO,d2
        move.1  #32,d3
        callsys Write
        endc
GExit2:
        move.1  TaskID(a4),a6
        move.1  AbsExecBase. W,a6
        moveq   #$47,dO
        bra. w  exit2

*-------------------------------------------------------------------------
* Addlt:        routine to add memory parts to variables

Addlt:
        move.1  MH_FREE(a5),dO
        add.l   dO,(a3)          ; add to available
        move.1  MH_UPPER(a5),dO
        sub.l   MH_L0WER(a5)5d0
        add.l   dO,4(a3)                ; add to Max section
        rts
        endc


        sect i on _MERGED, DATA
*
        xref    DOSBase

        xdef    NULL,SysBase,WBenchMsg
        xdef    curdi r,_mbase,_mnext,_msize,_tsize
        xdef    _oserr,_OSERR,_FPERR,_SIGFPE,_ONERR,_ONEXIT,_ONBREAK
        xdef    _SIGINT
        xdef    _ProgramName,_StackPtr,_base

        ifd     CATCH
        xdef    _ONGURU,_FMEM,_STAKOffset
        endc
*
NULL            dc.l    0               *
_base           dc.l    0               * base of stack
_mbase          dc.l    0               * base of memory pool
_mnext          dc.l    0               * next available memory location
_msize          dc.l    0               * size of memory pool
_tsize          dc.l    0               * totál size?
_pserr:
_OSERR          dc.l    0
„FPERR          dc.l    0
_SIGFPE         dc.l    0
```

```
_SIGINT         dc.l    0
"ONERR          dc.l    0
_ONEXIT         dc.l    0
j)NBREAK        dc.l    0
curdir          dc.l    0
SysBase         dc.l    0
WBenchMsg       dc.l    0
oldtrapcode     dc.l    0
oldtrapdata     dc.l    0
_StackPtr       dc.l    0
^tőin           dc.l    0
_ProgramName    dc.l    0

        ifd     CATCH
_ONGURU         dc.l    0
 IntuiBase      dc.l    0
TaskID          dc.l    0
        ifnd    AUTOREQ
GConsole        dc.l    0
failmsg         dc.b    7,'Can"t write SnapShot!',10
success         dc.b    7,'GURU caught; SnapShot written!' ,10
        endc
        cnop    0,4
_FMEM           dc.l    0
fp              dc.l    0               5 savé SnapShot file pointer
DumpPath        dc.b    1DF1:'
DumpName        dc.b    1SnapShot,,TF ,0
SeekStore       dc.l    0
TerapAddr       dc.l    0               ', Storage for &TerapStore
TerapStore      dc.l    0,0             ; Temporary storage for BPTR -> APTR
TempSize        equ     *-TerapStore

        cnop    0,4
        ifd     AUTOREQ
TAttr:                                  5 Text attributes for font
        dc.l    TName                   ; name of font
        de. w   TOPAZ_I:IGHTY           ; font size
        dc.b    FS_NORMÁL               ; font style
        dc.b    FPF_ROMFONT             j font preferences
TName:
        dc.b    'topaz1',0
        cnop    0,4

ITextl:                                 ', Text definitions for AutoReq call
        dc.b    3,0,RP_JAM1,O           ; front & back pens, drawmode and fillér byte
        de.w    6,4                     5 XY origin relative to container TopLeft
        dc.l    TAttr                   5 font pointer or NULL for default
        dc.l    ITextTextl              ', pointer to text
        dc.l    IText2                  ', next IntuiText structure
ITextTextl:
        dc.b    'Program:' ,0
        cnop 0,4
IText2:
        dc.b    3,0,RP_JAM1,O
        de. w   78,4
        dc.l    TAttr
IPName  de.1    0
        dc.l    IText3
        cnop    0,4
IText3:
        dc.b    3,0,RP_JAM1,O
```

```
        de.w    55,16
        dc.l    TAttr
        dc.l    ITextText3
        dc.l    IText4
ITextText3:
        dc.b    'I caught a GURU" ,0
        cnop 0,4
lText4:
        dc.b    3,0,RP_JAMl,0
        de.w    20,28
        dc.l    TAttr
        dc.l    ITextText4
        dc.l    NULL
ITextText4:
        dc.b    'Should I make a SnapShot?',0
        cnop 0,4
ITextö:
        dc.b    3,0,RP_JAMl,0
        de.w    6,4
        dc.l    TAttr
        dc.l    ITextText5
        dc.l    NULL
ITextText5:
        dc.b    'YES' ,0
        cnop 0,4
IText6:
        dc.b    3,0,RP_JAMl,0
        de.w    7,4
        dc.l    TAttr
        dc.l    ITextText6
        dc.l    NULL
ITextText6:
        dc.b    'NO' ,0
        endc

        cnop 0,4

;-------------------------------------------------------------------------
* New IFF chunk formát -
*       PGTB = Program Traceback, header for chunk
*       FAIL = reason for and environment of crash
*       REGS = registers at time of crash, including PC and CCR
*       VERS = version, revision, name of this program
*       STAK = ENTIRE stack at time of crash or, alternately,
*              the top and bottom 4k if the stack used is > 8k
*       UDAT = optional user data dump (if _ONGURU is set to a
*              function pointer in the user's program)
;-------------------------------------------------------------------------

PGTB            dc.b    'PGTB'
Length          dc.l    0               ; length of chunk (in longwords)

subFAIL    '    dc.b    'FAIL'
FAILlen         dc.l    9
NameLen         de. 1   0               ; length of program name
chunk_len_l     equ     *-PGTB
Environment     dc.l    0               ; CPU (, Math)
VBlankFreq      dc.l    0               ;       PAL = 50, NTSC = 60 (approx.)
PowerSupFreq    dc.l    0               5 Europe = 50,  USA = 60 (approx.)
Starter       • dc.l    0               5 0 = WB, -1 = CLI
GURUNum         dc.l    0               ; cause of crash (GURU #)
```

```
SegCount        dc.l    1               ; # hunks in seglist
chunk_len_2     equ     *-Environment

subFMEM         dc.b    'FMEM'          ; FMEM - free memory at crash
                dc.l    6
chipAvail       dc.l    0               ; available chip memory
chipMax         dc.l    0               ;       maximum chip memory
chipLargest     dc.l    0               ;       largest chip memory
fastAvail       dc.l    0               ; available fast memory
fastMax         dc.l    0               ;       maximum fast memory
fastLargest     dc.l    0               ;       largest fast memory
rMEMlen         equ     *-subFMEM

subREGS         dc.b    'REGS'          ; REGS - register storage field
REGSlen         dc.l    18
GURUAddr        dc.l    0               ; PC at time of crash
Flags           dc.l    0               ; Condition Code Register (CCR)
DDump           dc.l    0,0,0,0,0,0,0,0 ; data registers
ADump           dc.l    0,0,0,0,0,0,0   ; address registers
A7Store         dc.l    0

*=ubVERS        dc.b    'VERS'          ; VERS - program version field
                dc.l    6
                dc.l    VERSION         ; version #
                dc.l    REVISION        ; revision #
                dc.l    3               ; length of name of program
                IFD     RESIDENT
                dc.b    'catchres.o',0,0 ; name
                ENDC
                IFND    RESIDENT
                dc.b    'catch.o  ',0,0 ; name
                ENDC

subSTAK         dc.b    'STAK'          ; STAK - stack field
STAKlen         dc.l.   4
Type            dc.l    0               ; 0 = Info
StackTop        dc.l    0               ; top of stack pointer
StackPtr        dc.l    0               ; current Stack Pointer
StackLen        dc.l    0               ; # bytes used on stack
chunk_len_3     equ     *-subREGS

STAK2           dc.b    'STAK'
s21en           dc.l    1               ; length of subtype
                dc.l    1               ; 1 = whole stack
STAK21en        equ     *-STAK2

STAK3           dc.b    'STAK'
                dc.l    1025
                dc.l    2               ; 2 = top 4k of stack
STAK31en        equ     *-STAK3

STAK4           dc.b    'STAK'
                dc.l    1025
                dc.l    3               ; 3 = bottom 4k of stack
STAK41en        equ     *-STAK4

STAK5           dc.b    'STAK'
STAKOffset      dc.l    0
                dc.l    4               ; 4 = user defined amount
STAKSlen        equ     *-STAK5
```

```
UDAT            de.b    'UDAT'
                de.l    0
ÜDATlen         equ     *-UDAT
        endc

        END
```

```
Directory "Lattice C__5.t"i.4;saurctí" on Sunday 30~Sep-90
cbacl.a           ""        13665_____rwed 07-Nov-88 14:53:21
1 fi les - 30 blocls - 13665 bytes

•* C ínitial startup procedure under AmigaDOS
*

        INCLUDE "exec/types.i"
        INCI.UDE "exec/alerts.i"
        INCLUDE "exec/nodes.i"
        INCLUDE "eüííc/lists.i"
        INCLUDE "exec/ports.i"
        INCLUDE Vsec/libraries.i"
        INCLUDE "eí-ec/tasKs.i"
        INCLUDF "e::ec/memory.i"
        INCLUDE "exec/eüecbase.i"
        INCLUDE "librcu-xes/das.i"
        INCLUDE "libraries/dQse;'tens.i"
        INCLUDE "e;;ec/íuncdef„i"
        INCLUDE "e:>ec/e:;ec__lib.i"
        INCLUDE"11brarí esTdoşJíb.í"
        INCLUDE "wortbench/startup.í"

AbsExecBase EQU 4

TINY    EQU     0
rallsys macro
        CALLLIB _LVO\1
        endm

        xdef    XCEXIT                  ; exit(code) is standard ,way to leave C.
        xdef    5XCEXIT

        xref    LinterDB                ; linter defined base value
        xref    _BSSBAS         ; linter defined base of BSS
        :,re-f  JBSSLEN         ; linter defined length of BSS

        ;rref   _main                   ; Name of C program to start with.
        ;:ref   __f pimt                * jnitialize floating pomt
        'ref    __f pterm               * terminate floating point

*       library references

        csect   _N0MER6E,0,0,l,2        ; any 'cref's after this are 16-bit reloc

fitart:
        movem.1 dl-dó/aO-a6,~(a7)
PE6SIZE EQU     (6+7)*4
        lea     REGSIZE(a7),A5          * detetmine old stacl- pointer
        move.1  aO,a2                   * savé command pointer
        mo\fe.l dO,d2                   * and command length
        lea     Lirú erDB,a4            * load base registe^r

        move. 1 AbsEs^ecBase. W,a6
        movt>»l  a6,SysBase(A4)
        movp, 3 'a7,_Stacl-Ptr (A4 )    • Savé stact ptr
        clr.l   WBenchMsg(A4)

* ------ Clesr out out BSS sert ion
        lea     _.BSSBAS,a3             * .jet bast! of BSS
        moveq   #O,dl
```

```
        move,1   #_BSSLbN,df»              * gef" J ength of BSS in longwords
        bra. s   círjp                     * and clear for 1ength given
clr_b5Svmove.1   dl,7a3)+
clrj.p  dbf      dO,clr_bss

+________get the address of our tast
        move. 1  ThisTasI-. (ab) ,A3

*_________clear any pending signsH
        moveq    #0,d0
        move.1   #í00003000,d1
        callsys SetSignal

-. . . . . .are we rumung as a son oí Warl-beneh'^
        move. 1  pr__CurrentDir (A3) ,curdir(A4)
        t=st.l   pr_CLKA3)
        beq     -f romWort bench

*==========================================================================
*====== CLI Startup Code ===================================================
*==========================================================================

+ramCLI:
*------- attempt to open DOS library:
        bsr.w    openDOS

*------- Duplicate i-he current djrectory for the ralled process
        move.l   curdir(A45,D1
        mave.l   DQSB3se(A4),A6
        callsys  DupLocl-
        move.l   D0,curdir(A4)

4_________find coinm*nd n-ame:
        move.l   pr_CLI(A3),aO
        add.l    aO,aO                      ; bcpl pointer conversion
        add.1    aO,aO
        move.1   c11 JDommandName<sQ),a1
        add.l    al,al                      ; bcpl pointer ronversion
        sdd.l    al,al

*-------- collett paramétere:
        move.l   d2,dO             ; get command line ]ength
        moveq.l #0,dl
        move.b   (al)+,dl
        move.l   A1,D4                      ;save our pointer
        mave.l   dl,d3
        add.l    dl,dl        .             ; double length sjnce we copy it twi ce
        add.l    dl,dO                      ; add length of command name
        addq.l   #5,dO                      ; allow for space afier command
*                             .             ; As wel1 as for the roundup
*-----— We have the length of the command string to be copied.
4 ---— Allocate space far ít.
  ---— Pound up to the nearest 4 byte=>
        lsl.l    #2,dO
        add.l    memsjze(A4),dO
        move.l   dO,memsize(A4)

**************************** *#*X*#-*7í #****#**#*##*###+•)(#•**•< x x-**-*#****#*#*#*
*****************************************************************************
 ******** Allocate and copy ovtár ^he codp to du the magic stuff *********
*****************************************************************************
```

```
*****************************************************************

        lea     merniist(A4),A0
        move.l  Ab5E:-ecBa<5s. W,aó
        callsys AllocEntry              ;get a place to put the code
*       btst.l  #31,dO                  ;did we get the memory we as\ ed -for"
*       beq     ellenit                 ;no, can't do an/thing then
        move.l  DO,A1
        move.l  D0,-(A7)                ;remember the chuní we ?lloc;>ted

*-------- |N|QW we copy the code övet
        movee.l ML. _SIZE+ME_ADDR(A1! ,A1  jlocalo o\x\ allocated block

*-------- Copy over the command line to the allocated area
        lea     commandbuf-copybeg(Al),A3

*-------- At this point, we have:
*               AO - Wort pointer
'               Al - Pointer to Allocated memory blorl-
.               A? - POJnter to program name
r               A3   Pointer to target buf-fer
*               A3 - Pointer to our process structure
*               DO - Wort register
                Dl - Worl- register
*               D2 - length oí command name
*               D3 -- length of program name
*               D4 - Pointer to program name

        movea.1 D4,A0                   ; Load program, nante
        move.1  D3,D0                   ;       and length
        bra.s   cpyí
( pycmd: move.b (A0)+,(A3)+             ; copy övet thp command name
cpy1:   db-f    dO,cpyrmcl
        move.b  #' ', (A35-+            ; add the space delimiter

        move.1  A2,A0                   ; Load command te~;t
        move.1  D2,D0                   ;       ;md length
        bra.5   cpy?
':pylm: move.b  (A0)+,(A3)+
' py2:  dbf     dO,cpylin
        clr.b   (A3)+

*-------- Now copy over the program name
        move.l  A3,_ProgramName(A4)
        rnove.1 D4,A0
        iTiove.1 D3,DO
        bra.5   cpy3
' pypgm: move.b (A0)+, (A3)-t
' p/3;  dbf     D0,cpypgm
        clr.b   (A3)                    ;don't -forget trailing null on name

*-------- First we want tc> put it j ntu uur oeglist
        lea     sege:<it-copybeg (Al) ,A2   \
        move.l  A2,XCEXIT+2<A4)         jpatch the jump instruction
        addq    #4,a2
        move.l  a2,.5XCEXIT+2(A4)
        lea     ---,tart-4(PC) ,A2

i       Now we have:    Al to out  new fate segment
*4                      A2 to the start oí the reál seglist
        mij«e.l #(cüpvt?nd-copvbeg)/4, (Al)-t
```

```
        move.l   A1,D3
        lsr.1    #2,D3                    ;convert it to a bptr to the list
        move.l   (A2),(A1)+               ;save our ne:<t s.egment pointer
        clr.l    <A2>                     ;and lil] it -írom the chain
*------ We have the seglit->t

        lea      AutoLoader(PC),A2
        move. 1  #(copyend~Autol.oader>-1 ,D0
• opyit move.b   (A2)+,(A1)+
        dbf      dO,copyit

•*_____Bee í-f they want to do snme I '0
        tst.l    _BadGroundI0(A4>
        beq      noio
        lea      current_windüw(A4),Afi
        move.l   AO,D1
        move.l   #1005,D2
        move.l   D0SBase(A4),A6
        callsys  Open
        rnove.1 d0,. Bact stdout 'A4)
noio:
        move.l   AbsFuecBctse. w,Aó
        callsys  Forbi d

*------ Attach the tasl to da the dirt/ wori-
        move.l   j>rocname(A4),D1         j Name of í?sí to start
        move.l   jDnonty(A4) ,D2          ; Priority o-f bactground tasf
        move.l   _stacMA4),D4             ; StarJ si;e o-f created taal-

        move. 1  D05Bí>se(A4) ,Aé
        callsys  CreateProc
        ínove.l  A6,A1           ; restorc- closb- ' fn rlose 11
        tst.l    DO
        bnp      ol-

*------- Ser íous problems.  For somé reason the CreateProc -fai led.
*------- We need to free the AlocEntry memory and go home
        move.l   (A?)+,A2
        movea.l  ML_SIZE+ME_ADDR(A2),AÍ   ;]ocate our allocated bloct
*------- Alsó must get rid of current directory
        move.l   curdirfA4),D1
        callsys  lJnLocl-

        move.l   A2,A0
        move.l   AbsEííecBase. w,A6
        callsys  FreeEntry
        move.l   #104,D0
        bra      clie;<it

ul:
*------- The tasl started o\, altach the memory we allocated to
*------- íts tasl- contro] blocl-
        move.l   D0,A2

        move.l   (a7) + ,A3               ;ílocate the memory we allocated
        lea      - pr_.MsgPort íA2) ,A2
        1ea      TC JIFMENTRY íA2),A0


        ADDTAIL
        IEA      LH_TAIL(AO) ,A0
```

```
        MOVE.L  LN_PPED(AO),DO
        MOVE.L  AJ,LN_PREDíAO^
        MOVE.L  AO,(A1)
        MOVE.L  DO,LN_PRED(AÍ)
        MOVE.L  DO,AO
        MOVE.L  A1,(AO)

        move.l  #0,D0

' lxei'it:
        move.]  AbsExecBase.w,A6
        callsys Permit
        move.l  A3,A1
        move.l  AbsExecBase.w,A6
        callsys CloseL]brary
        movem.l  <a7)+,dl-dó/aO~aó
        rts

#====== Worl-bench Startup Cnde ==========

immWnrl-bench:

        move.l  TC_SFLOWER<A3> ,_baseíA4>       ; set base of stact-
        add.l   #l?S,_base(A^I)          i^llow far panns overflow
--------opeh t-he DOS library:
        bsr.w   openDOS

        lea     .cent(PC),A2
        move.l  A2,XCEXIT+2(A4)          ;patch the .uimp in=>trurtion
        addq    #4,a2
        mave.l  A2,5XCEXIT+2(A4)        ;pst(.h tht jump instruction
        subq    #4,a2

•*--------we are now set up.  wait íot a me=.sage from our starter
        bsr.w   waitmsg
        move.l  dO,WBenchMsgíA4)
        move.l  dO,-(SP)

        move.l  d("),a2                  ; get -first argument
        move.l  sm_ArgList(a2),dO
        beq.s   do_con=i
        move.l  D0SBaso<A4),a6
        move.l  dO,aO
        move.l  wa .Lort (aQ) ,dl
        move.l  dl ,curdirfA4)
        ral]sys CurrentDif
!Ío_cans:
        move.l  sm_ToolWindowfa2),dl     ; get the window argument
        beq.'a  do__main
        move.l  #HODE _OLDf11 F,d?
        callsysQpíin
        mave.l  dO,stdin<A4)
        beq.s   do__main
        lsl.l '  #2,dO
        move.l  dO,aO
        move.l i h_Type <sO'> ,pr_C,unsol eTa^t (A3^
'1D_main:
        move.l  WBenrhMsglA4) ,aO                ; get address of worl-bench mesaag
```

```
        move.l  a0,-(a7)                ; push
        pea     NULL(A4)                ; push
        move.l  sm_ArgList(a0),a0       ; address of arguments
        move.l  wa_Name(a0),_ProgramName(A4)    ; name of program
* ------ for startup from workbench, there is no reason to go through
* ------ The fancy stuff to make the startup work.  Just run the
* ------ Code directly
        jsr     __fpinit(PC)

        moveq   #0,d0
        bra.l   exit2


copybeg:
        dc.l    (copyend-copybeg)/4
next:   DC.L    0
AutoLoader:
        move.l  a7,d0
        movem.l d1-d6/a0-a6,-(a7)
        move.l  d0,a5
        move.l  AbsExecBase.w,a6
        lea     LinkerDB,A4             ; load base register
        move.l  a7,_StackPtr(A4)

        bsr.w   _fpinit

        suba.l  a1,a1
        move.l  AbsExecBase.w,A6
        callsys FindTask
        move.l  d0,a3

        move.l  ...,_base(A4)           ; set base of stack
        add.l   #128,_base(A4)          ;allow for parms overflow

        move.l  pr_SegList(A3),a0       ;locate the segment list
        add.l   a0,a0
        add.l   a0,a0                   ;and convert to APTR
        move.l  12(A0),A1               ;so we can access the segments
        add.l   A1,A1
        add.l   A1,A1

        move.l  (A1),seglist(A4)        ;save for unloading later
        move.l  (A1),12(A0)             ;cut off the segment

        move.l  curdir(A4),D1
        move.l  DOSBase(A4),a6
        callsys CurrentDir

        pea     ...(a3)
        jsr     ...
        addq    #4,A7                   ; call ...

* ------ They are done so unload them
```

```
segexit:
        move.l   seglistíA4),03
        move.l   D0SBase(A4),a6
        callsys  Unt oadSeg

*-------Unlocl- the djrectory we 3t?t up -far them
        move.l   curdiríA4),D1
        callsys  UnLoct

        moveq.l          ttO,do           ; -et successful status
        bra.s    e;dt?

: cenit:
        move.l   4(SP),dO                 ; e.,tr*ct reLui-n code
p-it2;
        move.l   dO,-(a7)
        IFEQ     TINY
        move.l   _0NExIT(A4),dO           ; e;at trap function^
        beq.s    e;:it3
        move.1   dO,aO
        jsr <3.0)
        ENDC

e: it3  jsr      MemCleanup               ; rleanup leftover memory alloc.
        move.l   AbsE"ecBase.w,a6
        move.1   DOSBase(A4),a1
        callsys  CloseLíbrary             ; close Dos librgry

        jsr      _fptermíPC)

dontí_le:
*-------]-f we ran -from CLI, skip workbench cleanup:
        tst.l    WBenchMsg(A4)
        beq.s    exitToDOS
        move.1   stdin(A4),dl
        beq.5    done_4
      ' callsys  Close
done_4:

*-------return the startup message to our parent
<       we forbid ?o worlbench ran't UnLoadSegO us
•*      be-fore we =ire done:
        move.1   AbsExecBsse.w,A6
        calIsys  Forb3 d
        move.1   WBenchMsg(A4),al
        ca J1sys Pep1yMsg

*-------this rts sendb us bacl- tu DOS:
e;itToDOS:
        MOVE.L   fA7)+,D0
        movea.l  „Staeí<Ptr (A4) ,SP            ; rettore stact ptr
        •nuvem.l <37)+,dl-dó/sO a6
        rts

*-------------------------------------------------------------------------
*  Open the DOS 1íbrary:

o'-enDOS
                 lea      D0SNameíA4),A1
                 moveq.l  #0,DO
                 callsys  Oppr\L 3 br#t y
```

```
             move.l   D0,DOSBase(A4)
             beq      noDOS
             rts
---------------------------------------------------------------------------
noDOS:
             moveq    #100,d0
             bra      exit2

copyend:
commandbuf:

**************************************************************************
**************************************************************************
**************                                              **************
************** End of code that is copied to fake segment **************
**************                                              **************
**************************************************************************
**************************************************************************

*-------------------------------------------------------------------------
* This routine gets the message that workbench will send to us
* called with task in [illegible]

waitmsg:
             lea      pr_MsgPort(A3),a0        ; our process base
             callsys  WaitPort
             lea      pr_MsgPort(A3),a0        ; our process base
             callsys  GetMsg
             rts

        csect  __MERGED,1,,2,2

*       These externals control the task that is started
        xref    _BackGroundIO                  ; flag to indicate background task
        xref    _stack
        xref    _procname                      ; name of process to be created
        xref    _priority

        xref    MemCleanup
        xref    FindTask
        xref    DOSBase
        xref    MathBase
        xref    MathTransBase
*
        xdef    NULL,SysBase,WBenchMsg
        xdef    curdir,_mbase,_mnext,_msize,_tsize
        xdef    _oserr,_OSERR,_FPERR,_SIGFPE,_ONERR,_ONEXIT,_ONBREAK
        xdef    _SIGINT,_Backstdout
        xdef    _ProgramName,_StackPtr,_base
*
             [illegible]
initial ds.b     MN_SIZE
             [illegible]
             dc.l     [illegible]
             dc.l     0
             dc.w     $4ef9                    ; patch jump for [illegible]
             dc.w     $4ef9
             dc.l     0
```

```
neglist         dc.l    0
tJULL           dc.l    0               ;
 ha.se          dr.1    0               ; base of sta.ck
 mbase          dc.l    0               ; base of memory pool
jnnext          dc.l    0               ; next available mentory location
_msize          dc.l    0               ; size of mentory pool
_.tsize         dc.l    0               ; totál size?
jiserr          equ     *
"OSERR          dc.l    0
"FPERR          dc.l    0
"SIGFPE         dc.l    0
„SIBINT         dc.l    0
"DNERR          dc.l    0
"ONEXIT         dc.l    0
".QNBREAK       dc.l    0
curdir          dc.l    0
HysBase         dc.l    0
WBenchMsg       dc.l    0
„StackPtr       dc.l    0
íiosCmdLen      dc.l    0
ilosCmdBuf      dc.l    0
íitdin          dc.l    0
_Backstdout     dc.l    0               ;filehandle for 1/0
_ProgramName    dc.l    0
DOSName         dc.b    'dos.library',0
current_window  dc.b    '*',0
                END
```

```
Directory "L3t-tice_C_5.0.4:sourctí" an Sunday T0-Sep-90
c«ffp.a                    2912———rwed 07-Nov-88 14:53:17
1 fj.le3 - 7 bluds - 2912 bytes
        IDNT    "a-Hp.o"
*****************************************************************************
*
+ na,me          cu-f-fp.a
#
* descnptjon   provide CX function^ -for Affliga's Mcitornls F sst Floeting
-*               Poi nt formát
*
*
*****************************************************************************
*
        section  te^t,C0DE

        ,-def    nFSPAdd
        í'reí    _LVOBPAdd

MFSPAdd movem.l  dl/d7/d6, (a7)
        move.l   #_LV0SPAdd,D7
        tst.l    „MathBase
        bne.s    callfunc
        br^.s    open_lib

        :-def    HFSPSub
        í-ref    _LV0SPSub

fiFSPSub movem.l dl/d7/a6,--(a7)
        move.l   #_LVOSPSub,D7
        tst.l    _MathBase
        bne.s    callfunc
        bra.s    open_1íb

        ,;def    MFSPNeg
        :;ref    _LVOSPNeg

MFSPNeg movem.l  dl/d7/a6, fa7)
        move. 1  #_LVOSPNeg, ü''
        tst.l    JiathBase
        bne.s    callfunc
        bra.s    open_]th

        xdef     MFSPMul
        ;;ref     _LVDSPMul

HFSPMul movem.l  dl/d7/a6,-fa7)
        mnve.1   #_LV0SPMnl,D7
        tst.1    _MathBase
        bne      rallfunc
        bra      open_1ib

                 , c ef     ¸H<ái:hBase
                 , re-f     J.eüit
                 ííre-f     _OpenLibrary¸

                 nnp
cnen__lib        bsr        open__call
call-func        move. 1    __MathBase,a6
                 j^r        Ofa6,ri7.])
```

```
                movem.l {é7)+,dJ'' d7/a6
                rts

of..en_cal3     movem.1 dO»dl,-<a/)             savé paraméter5 for call
                clr.l   -(a7)
                pea     FFPLibName
                jsr     _QpenLibrary
                lea     Sía?),a7
                move.1  DO,_MathBase
                movem.1 <a7M,dO-dl              restore paméteri5
                beq.3   er ror
                rts

ti ror          moveq.1 #100,d0
                j Bt"   __ex11


        ,ıJei   MFSPDiv
        :,re-í  _LVOSPDiv

fi"SF'Div movem.1 dl/d7/a6,-U7)
        move.1  #_LVOSPDJV,D7
        tst.l   _MsthBase
        bne     rallfunc
        bra     open_Jib

        xdef    MFSPFi;;
        „ref    J-VQSPFi!'

HfSF-'F) • mavem.1 dl/d7/a6,-(a7)
        move.1  tt_J..VOSPFi,D/
        tst.l   _MathBase
        bne     calIfunc
        bra     open__lib

        üdef    MFSPFIt
        ;;ref   _LVOSPF1t

flFSPFlt müvem.1 dl/d7/d6,-(a7)
        move.1  *_LVOSPFlt,D7
        t-st.3  _MathBa^e
        bne     call-func
        bra     open_lib

        ;def    MFSPCmp
        ,:reí   J VOSPCmp

MFSPCmp movem.3 dO-d7/s6,-(a7)
        move.1  tt^VOSPCmp,D7
        tst.l   „MathBase
        bne.5   '-mp_2
        bsr.B   open_cal1               open the lxbrary
<mp_2   move.1  _MathBase,Ab
        jsr     0(a6,d7.1)
'-     1-.c,1   dO                      correct for 1.2
        neg.1   dO                      \luge for 1.1 bug
        movem.1 fa7)*,dO-d7/*b
        rts

fFPLibName      dc.b    'm3thf-fp.llbrary',0
```

end

```
Director^ "tjttice_C_5.0.4:saurcp" un Sunday 3Q-Sep-?Q
•ffptran.a           "       3169    i-wed 07-Nov-88 14:53:22
  •files - 8 blocl-s • 3169 bytes
         IDNT    "ffptran.o"
++*++*** Copyright- 1986, Laitxce, Inc.   *•«*•+**+•**•#*•**#**#•*+•***+####**

* name           mfíptrari.a
*
* description   This file contains the interface stub routinen between tht[1]
t               Idttice i mplementation o-f HFFP and th-? Amiga MFFP Library.
*
*
        ac os
        jiin
        3 tan
        cos
        cosh
        e:-p
        -fabs
        loq
        loglO
        paw
        sí n
        sinh
        sqrt
        tan
        tanh
*
-'+*•*+*+•*•*#***#*•«•*******••*•****#***** ****** x- **###«**#*#»- )(-**+**#+*•**** *#***##**i(

        ;ref    MdthBase
        síref   MathTransBase
*
        csert   te/t,0
        ^ection   te;-t,CnDE

        xdeí    acos
        ;-reí   SPAcob

acos    tst.l   MathTransBase
        bne. 5  + :;
        bsr.s   open_1íb
f(      jmp     SPAcos


        : deí   asin
        nrei    SPAsm

asin    tst.l   MathTransBase
        bne.s   fO
        bsr. 5  open__l3b
i'j     jmp     SF'ASJ n

        :'def   atan
        ;;ref   SPAtan

í<tan   t st. 1 Ma t h Fr ansBase
        bne.s   fi
        bsr. 5  opp>n_ ] íb
 1      jmp     SPAtan
```

```
            xdef    cos
            xref    SPCos-

cos         tst.l   MathTransBase
            bne.s   f2
            bsr.s   open_lib
f2          jmp     SPCos

            xdef    cosh
            xref    SPCosh

cosh        tst.l   MathTransBase
            bne.s   f3
            bsr.s   open_lib
f3          jmp     SPCosh

            xdef    exp
            xref    SPExp

exp         tst.l   MathTransBase
            bne.s   f4
            bsr.s   open_lib
f4          jmp     SPExp

            xdef    abs
            xref    SPAbs

abs         tst.l   MathTransBase
            bne.s   f5
            bsr.s   open_lib
f5          jmp     SPAbs

            xdef    log
            xref    SPLog

log         tst.l   MathTransBase
            bne.s   f6
            bsr.s   open_lib
f6          jmp     SPLog

                    xref    OpenLibrary

open_lib            clr.l   -(a7)
                    pea     FFPLibName
                    jsr     OpenLibrary
                    lea     8(a7),a7
                    move.l  D0,MathTransBase
                    beq.s   nd_Lib
                    rts

                    xref    _exit

nd_Lib      moveq   #100,d0
            jsr     _exit

            xdef    pow
            xref    SPPow

pow         tst.l   MathTransBase
            bne.s   f7
```

```
          b'ai'.b   apen_lib
f7        move.1    4(a7),dO        ; swap arguments (sheesh!)
          move.1    8(a7),4(a7)
          move.1    dO,8(a7)
          jmp       SPPow

          xdef      sin
          xref      BPSin

s.in      tst.l     MathTransBase
          bne.s     fa
          bsr.s     open_jlib
f8        jmp       SPSin

          xdef      sinh
          xref,     SPSinh

f.ính     tst.l     MathTransBase
          bne.s     f9
          bsr.s     open,_lib
f7        jmp       SPSinh

          xdef      sqrt
          xref      SPSqrt

r-qrt     tst.l     MathTransBase
          bne.s     f10
          bsr.s     open„lib
'10       jmp       SPSqrt


          xdef      tan
          xref      SPTan

'ui       tst.l     MathTransBase
          bne.s     f11
          bsr.1     open__lib
'11       jmp       SPTan


          xdef      tanh
          xref      SPTanh

t-anh     tst.l     MathTransBase
          bne.s     f12
          bsr.l     open__lib
f12       jmp       SPTanh

FTPLibName        dc.b      'mathtrans.1ibrary',0

                  end
```

[•irector/ "Latti.-* _C_L"i.".4:saurLe" on Sunday 30-Sep-90
o-s.a 11935——**rwed 07-Nov-88** 14:53:2'1
1 files - 26 blocts - 11935 bytes
TTL AmigaDOS 68000 overlay supervisor
* 4₄ₕₕₜ₊₄#.«.₊#.jн[..rt* t*%4#^4^^^##########*+**» 4-44-s •* * * + * * * i ' •* -i #•****

```
* This version of the overlay supervisor is callable     *
* from C or Assembler; it will not wort for MCC Pascal   *
•* or BCPL which use a calling sequence whery the called  *
* routine's address is saved on the stact. A different   *
* version of the overlay supervisor is available for     *
* case.                                                  •*
*                                                        *
* The main overlay algnrithm used here wac origjnal]y    *
> described by Pochard Evans *t Cambridge Umversity     *
* in about 1*979. This code imp] ementati on was done by  +
* Tim King of Mptarciini.o.                              *
>                                                        *
•* Two main data structures are used;                    *
                                                         *
•*   1. The hunt table                                   *
                                                         *
t-       This table, giving the base addresseá of all    *
           hunts currentl/ loaded, is manufactured by    *
           the loader.  Its addre-ss is planted in HTA8, *
•*         and its fjrst word is the address of hunt 0.  #
•*         Note that the addresses it contains are those *
•*        of the lint word oi each hunt -- not the first #
           data word.                                    *
                                                         *
•4   2. The overl •>.- I -tit^                           *
                                                         *
*       Thi -5 table i<-, manufacturyd by the linkage    *
*         editor, and íls address is planted in QVTAB    *
*         by the loader.   It cansiBts of twn mun       •»
-»        parts, the ordinate table, and the overlay     *
*         data table.  The farmer is a tahi*3 which is   *
*         used to remember the ordinate number of each   *
3        segment that is currently loadtíd. If n LS the  #
          maítjmum levél number, then the table contains *
1        n+1 entries, and the f-jrst word is n+2. The    *
*         ertra entry is required becausp a scan         *
•4        through the table below is terminated by a     *
+         zero entry.  AH the? entries are ínitially     *
*         cleared by the lintage editor.  The address    *
—         of the overlay table, plus the contents of     4
1        its firit word (n+2), gives the address of      *
*         the overlay data table, which 33 indexed by    •*
-*        overlay number.  Ea>"h entry in this table     *
•4 contains the following inf ormát ion, not             *
-*        necessarily in this order:                     #
*                                                        *
*  1. A fiié mart-, genurated by "not--", ÍJxv 1 ny 4    *
*              the position of the segment to be         -4
•*            loaded.                                    *
*          2. The overlay levél of the segment.          *
*          3. The overlay ordinate of the segment.       *
*          4. The initial hunt number of the segment.    *
•*         5. The number of the huni containing the      *
*             Symbül.                                    *
4          6. The ofis^t nf the <3>mbo1 in the hunt.     #
```

```
*                                                                    *
•f If an error is detecteci dunng n¥H l*ying, there is a  *
  limited choxce of sensible actions, ÍJ nc.ci the overJfty  *
* supervisor can mai e no assumptions concernxng the       *
* enviranment ít wáá eritered iram.  The ar.tion on art     *
+• error is to call <»n ihlett with the rocle for invalid  *
* averla/ request.                                         *

<« Note that the value irt HTAB is a loivj ward address,   *
* whilst that in OVTAR is a machine address.  Each entry *
*• in the table is faúr bytes long.                        •*
*                                                                    *
¥*###•*!***#*+**+#iH#*-ií+++*#####****#*#+#1f***#***##í#*•*•*#*•)!*
*•  Change History:

+       CAP1  05SEP85
-            1> Changes made (suggested by Paul) to the handling
*               of s>9el. and loadseg errors.  When an error
-               occurs, the program w} ] 1 re?tf y until it is
-               successful.
*            2) Changed bra.s to just bra in OVENT and 0VEN72
•*              iiijcro def initiona.
*            3) Added the OVFPLANS equate and thp necessary OVENTI'
+               table entries í':undi tionally assembled).
*            4) EJiminatecI unrieceasary inclucle -files.
+       CAPL 06SEP85
*            15 CAl L macro modi-fied.
-            ?) CAl I. UÍIP madif xed.
-    "   DTM  15JUN87
-            1) modified to either deíine or use static OCVs, or
•r              Overlay Call Vectors, as wp now rali them.
·            2) changed entry name to "_pvlyMgr" (note that the
<*              undor ijcort[1] is adiJed autómat i cally, and case IS
·               ver y jmportant
*            1) added instructiun Lu m^i-e %ury that onl/ lower
*               16 bit=; of dO are used irt inde: ing intő table;
•*              the itew dyrtamic OCVn only pas-í a word in dO,
*               slthaugh ít's still multiplied by 8 and ' 0.
*            4) Basically al 1 of thi^ xniirease the tata] numbpr
'               of OCVs to Bl , and furthermare maies xt
*               im , n,i . in put thxá rodtí mtu a library



        section NTRYHUNJ ,C0DE

        INCI UDE "e:>ec/typps.i"
        INCI.ÜDE "e:'er/tasls.x"
        INCLUDE "e:;ec/3 ibranes.i"
        INCLUDE "es'ec/alerts.i"
        INCLUDE "lxbranes/dos.x"

        ;'de-f    ovlytigr

rüNCDEF         MACRO   *• function
_1 V0\I         EQU     FUNC_CNT
r!JNC_CNT       SET     FUNC_CNr-6
                ENDM

ríJNC_CNT       SET     I.ID _NONSTD
```

```
        INCLUDE "e;-íec/e;;tíc„lib.3 "
        INCLUDE "libranes/dos_lib.i"


-• Q-ffsets within t-he nver 3 ay table.
SvsBase EQU          4

OrMAF-t  EQU          0                    Fi le position
OTLEVL   EQU         12                    Ievei
PTOPD    EQU         16                    Or dinate
UTIHM    EQU         20                    Initial huni- for load
DTPHNK   EQU         24                    H"nl- tontaxning svmbo]
OTQFF    EQU         28                    Of-fset of symhnl
iTr.wnpD EQU         ?"45A


4 A m'irro to ...ll a LIBRARY routine:
TALLB    MACRO
         MOVEA.L     Sy=.B35e.w,A6
         JSR         _LV0\I(A6)
         E^4DH

-f Call a ljbrafy, sa-vt and restore A6.
CALL     MACRO       ; CALL  3 ibrary_vectnr_ üfffiet , 11 brjf /_pointet
         MOVE.L      A6,-(SP)
         MOVE.L      \2,A6
         JSP         _LV0M(Aó)
         MQVE.L  '   (SP)+,A6
         FNDH


- ttow for the first word oi the program.

rif-'ST    BPfi.L Ne: tModulé

' Thi  ne + word serves in íd^ntiíy the oví'rlay
•' uupervisor to 'unloader'".

         DC.L        ÍABCD                Spco £)] v=il ue


• The loader plants values in the ne::t \m ...ons.

 TREAM   DC.L    0                        Overlay input strc.am
CVTAB    DC.L    Ö                        Qvsrlay table fMachine address)
HTAB     DC.L    0                        Hunt table    (BCPL address)
"1BVEC   DC.L    !.)                      61obs3 ve>rt'»r (Mji:hintj address)


•* The[1] nmn code of the program.

         CNOP        0,4
         DC.L        L1BWQRD
         DC.B        7,'0verlay'

nvlyMgr
rvi      MOVEM .L    A2-A7/D2- D7,-'SP)
         MOVE. L     QVTAB(PC),A3         Addres^ HÍ iverl iy t.ble
         MOVEA .1    A3, A4
         AND.L       ttíOFFFF.DO          Mai- e SUNr Lh 1 upper word is ZERO
```

```
        ADD.L     (AD ,DO              Add value in fu%L wurd of table
        LSL.L     #2,D0
        ADDA.L    DO,A3                Address of overlay entry
        MOVE.l.   OTLFVL <A3),DU       Bet required leve!
        LSL.L     #2,D0
        ADDA.L    DO,A4                Pointer to entry in ordinate table
        MOVE.l    HTORDÍAJ),DO         Get required ordinate
        CMP.L     <A4),DO              Compare with adua]
        BEQ       ROI strn             Branch if no load required

* Now clear other entriPíi in the orrii nate table

        MOVE.L    DÜ,<A4) +
f J?    TST.L     ÍA4)                 Tablc ternuriaten' by ?erro
        BEQ.S     nvr
        CLR.L     (A4H
        PPA.S     0V2

i)V3    MOVE.L    OTIHNf (A3),D0       First huni- number of overlay segmení
        ADD.L     HTAB<Pr),D0
        LSL.L     #2,D0                Address of entry in huni table
        HOVEA.L   Dn,A4
        MOVE.L    -4ÍA4),D0            F'revi fiis huni-
        LSI..L    #2, PU
        MOVEA.L   D0,A6
        CLP.L     (A6>                 Cle.ir linl -field in case of error

* Now free all the unwanted hunts ...

ñ "4    TST.L     f.A4)                Scsn is terminálért by zero
        BEQ       0V5
        MOVE.L    (A4M,D1              Address as aryument far 'freever'
I˙RFEVTC  TST.L     DI
          BEQ.S     pvri             Return if V = 0
          ASL.L     #2,1)3           Bacl to mc address
          MOVEA.L   D1,A1
          MHVF.L      (AD ,D0        DO = sine
          MOVE.L    DO,D1            Ta^e copy
          ANDI.L    #ÍFF000003,D1    Top byte "ind boltom 2 bit*}.-
          BNE       OVERR            .. must be clear to be \3l1d length
          CALL      Ft eeMem,GysBr
F VC1     BRA.S     QV4              and again

•* Now savé CIS and pcnnf the file

ÍlV5
        LF.A.L    LIBNAMEfPC),AI
        MOVEQ     #ö,DO
        CALL      OpenLibrary,SysBase.w
        MOVE.L    D0,D7
        BEQ.S     OVERR

        ADDA.L    #OTMAR* ,A3          í-id>1ress c>f file mart

' Now spet to the segment and load ít.  £f w« fail (|jrohibl> beiause
* the user removed the di?l>, then retry until successfu3 ,  This has
* the ef-fect of contxnual 1 / reprompting the user to ínsert í-he diai-
-» unt 13 he obeys.

ÜJ_RETRY:
        MOVE.L    STPEAM(Prr. ,D1          New stream
```

```
        MOVE.L      (A3),D2
        MOVEQ       #0FFSET_BF6INNING,D'J Offset to be from beginnmcj
        CALI        Seel ,D7               Cdii 'paint (scb,m^rl- ,u)'
        T^T.L       Dl                     Checl result
        BMI.S       0V_RFTRY               Branch if errnr

        ; Now í~al 1 the-1 1 oadet ay:-in:
        MOVE.L      HTAB(PC),D2            Huni- Uble <serond paraméter)
        MOVEQ       #O,D1                   7ero paraméter far overlay
        MO   VEJ    STREAM(PC),D3
        CALL        LoadSeg,D7
        TST.L       ül                     r.hecl result
        BMI.S       OV_RFTh1Y              Branch íf error

* Add new 'J J =st to chain a-fter pr^v1ous one

        MnvE.t      Dl,ÍAó)                Add new chain

* Here when aegmenL is in store: A3 hnldL tnble entry, Al and Du dre hee

G0TSE6  MOVE.L      OTRHM (A35,D0          Numbor oí hunt coniíumncj symbol
        ADD.L       1ITAB(PH),Dü
        LSL.L       tr,DO                  Mac hl ne addresb of rntry in hunt tatür
        MOVEA.Í     Dn,A4
        MOVE.L      (A4) ,DO               BCPL addresa of hunt
        LSL.L       #2, l)Ci
        ADD.L       OTOfF- (A3),DO         Add in offset oí symbol

* The address to jump tn 13 now in DO. P3 ar.a iL in A1 and jump to it.
        MOVEA.L     DO,A1
        MOVEM.L     (SP)+,A2-A7/D2-D7      resLore regs
        JMP         (Al)                   Jump to new code

•* Come iv^re on an ert or (Juring tlie ovprlay

C'/EPP
        MOVE•L  #AN_BadOvtr la/,D 7
        CALI S  AJ er t
        RTS

•* The? codf he-re will raus>e UÍ> to jumfí tü the ne:-'t secjment in the list,
* whirh should be the intend^d entry pmnt.

N• , tModule
        LEA.L       rlPfíT (PC),A3         Pointer to start of modulé
        MOVE.L      -4(A3) ,D7             Ne: t BHPI word ptr
        ASL.L       #2,D7                  MC ptr to ne;'t modulé
        MOVE.L      D7,A3                  Tnto iddres"3 register
        JMP         4(A3)                  Tall/ ho1

* Data Area

L IBNAME DD.B       'dos.l ibrary',C
        CNOP        0,4


        EfJD
```

```
Direr.tory "Láttice._C_5.0.4:saurce" on Sunday 30-Sep-90
ucxovf.a                  •   857_____rwed 07-Nov-88 14:53:24
f -files - 3 blocks - 857 bytes
***     Copyright 1986, Lattice,  Inc.
#
* name           xcovf
*
•* description   this function is called when a stack overflow is
•*               detected.  It resets the stack pointer and calls
*•              the overflow handler
*
#**

         section te;<t,CODE

         xref     _StackPtr
         xdef     _xcovf
       IFD REGARGS
         xref     Scxovf
         xref     Sexit
       ELSE
         xref     cxovf
         xref     exit
       ENDC
_xcovf   equ      #
       IFD NOBASER
         move.1   _StackPtr,A7          ; reset stack pointer
       ELSE
         move-1   „StackPtr(A4),A7          ; reset stack pointer
       ENDC
       IFD REGARGS
         jsr      ;ícxovf(PC)
         moveq.1  #20,DO
         jsr      ó)exit(PC)
       ELSE
         jsr      cxovf(PC)                 ; display requestor
         pea      20
         jsr      exit(PC)                  ; call exit
       ENDC
         end
```

```
Directory "Lattice_C_5.0.4:example5/*.a" on Sunday 30-Sep-90
emsup.a                    1237 ------rwed 07-Nov-88 15:18:37
popsup.a                    723 ------rwed 07-Nov-88 15:13:35
2 files - 7 blocks - 1960 bytes
```

```
nrertary "Lattice f, _5.f',4:e!'amplps" an Sunday I0-Sep-90
i^msup.a                        1237_____rwed Ö7-Nov-G8 Í5íl8s37
  fi les - 4 blocls - 1237 bytes
        csect    te;'t,0,,l,2

        i'def     _SaveMem
        ;: d pf __V a 11 d 91 eMem

*** this is the number of L0NGWGBP5 al the start o+ memory to he» rhecled
MEMSAVESIZE     equ     64
*****************************************************************
* Validat"-? thf; current memar> ínfnrmation dnd put up ari :Oert íf it is
* in sad shape
*****************************************************************
..'Val ídateMem:
        movem.L  AO/A1,--(A7)
        mrweq.1  #MEMSAVESTZE,DO
        1EA      0,A0
        l.EA      -\eareaíPC),À1
( hec kit:
        cmp.1    ''AO)H , (Al ) +
        dbne     DO,checl it
        addq     #1,DO
      ,. beq     allol
-i restore what got ser ad
*       first -formát the message to be printed
»       we get a pointer to the address and thp data portion
        move.l   -íA05,D0        get the value that was put in
        move.l   -fAl),fA'"')    and replace wit^h what should heve been +1
        move.l   J2i'A7),Al      point to where we should savé thp resulí
        move.l   D0,fAl)         savé thtí value
        move.l   AO,4CA1)        and tho atldress -for the rontino i-o munge
        moveq.l  #1,D0             ^ A re%ult value to indicate failure
\.flat:
        ma sem.1   (a7) +,Au/fú
        i' ts

      Men.:
        movem.L DO/AO/A1,- >A7)
        ma_vpq,1 ttMEMSAVESIZE,DO
        l.EA     0,A0
        LEA      saveareafPC),A1
- ive!p:
        move.1 Í'AO) >,fAl) +
        DBF      D0,savelp
        movem.l  fA7)+,DO/AO/A1
        rts
s avearea       ds.l     MEMSAVESIZE+2    jfor %avp and rompar^ of memory
        END
```

Directory "l --^-+i •••• '" ^ i">„4:example5" on Sunday 30-Sep-90
j.-opsup.a 723 -.....-rwed 07-Nov-88 15:18:35-
1 files - 3 blocks - 723 byt.es

```
•*******•*********************•»*****************•************************•***
•*HandlerInterfac.e()

•*   This code is needed to tunv^tt the calling sequence performed by
*    the input.task for the input stream management intő something
•    that a C program can understand.

•*   This routine expects a pointer to an InputEvent in AO, a pointer
•*   to a data area, in Al.  These values are transferred to the stack
•*   in the order that. a C program would need to find them. • Since the
•*   actual handler is written in C, this works out fine.

•<   Author: Rob Peck, 12/1/85

        ;;def     _HandlerInterface
        ;; ref _myhand 1 er
        section "text",code
_HandlerInterface:
        movem.L AO/Aí. ,-<A7)'
        jsr     _myhandler'
        addq.L  #8,A7
        rts

        END
```