



Linux ブートアップ ソースコードの解析

カーネルのセットアップを調べる video.Sの解説

芳之内弘

前回と前々回では、Setup.S本体を解説しました。今回からは、Setup.Sにインクルードされている「video.S」の解説を始めます。このvideo.Sは、ビデオカードのハードウェアに依存する部分が多く、カーネルのバージョンアップによりサイズも大きくなっています。例えば、2.2.14のバージョンでは約2000行にもなります。しかし、基本的な動作は共通していますので、この基本部分について解説していきます。

今回調べるソースコードは、本誌2000年8月号の付録CD-ROM中の/Kernels/v2.2/linux-2.2.14を展開して得られる「linux/arch/i386/boot/video.S」です。

BOOT

ビデオBIOS

video.Sで使われているビデオBIOSサービスについて説明します。ビデオBIOSは、ビデオアダプタに実装されており、アダプタ固有のサービスもありますので、ここでは「VESA BIOS」を除いた、一般的なサービスについてのみ説明します。

ビデオBIOSのサービスはすべて「INT 0x10」の割り込みで提供されます。サービスの機能はAHレジスタに番号をセットしてサービスを要求します(表1)。

BOOT

VESA BIOS

オリジナルのビデオBIOSは、VGAまでがIBMによって定義されています。このVGAを拡張したものを、PC互換機市場ではスーパーVGAとして、さまざまなビデオモードがサポートされました。しかし、ビデオカード間の互換性に問題があったため、ビデオカードメーカーが集まって標準化作業がなされ

ました。この団体を「VESA(Video Equipment Standards Association)」といいます。最近のビデオカードはVESA規格をサポートしているものが多くなっていますので、基本的なVESA規格のディスプレイモードを表2に示します。

VESA BIOSのサービスは、AHレジスタに「0x4F」をセットして「INT 0x10」割り込みで要求します。ALレジスタには、要求する機能番号をセットし、補助機能番号がある場合は、BLレジスタにセットします(表3)。割り込み完了後のALの値の意味は、表4の通りです。

これらのVESA BIOSの詳細は、かなり膨大です。詳しい情報は、記事末RESOURCESの[1]を参照してください。

BOOT

video.Sの解説

video.Sは、Linuxが立ち上がったときの、コンソールのビデオモードを設定します。このvideo.Sは、かなり長いので(約2000行)、主な部分のみを解説します。フローチャートを見ながらソースを追いかけてください。

video.Sのルーチンは、メインが「video」です。このルーチンから呼ばれる主なものは、「basic_detect」、 「mode_set」、 「mode_menu」、 それに「mode_params」です。store_screenは、画面データをバッファに保存します。restore_screenは、このデータを再び画面に戻すルーチンです。

ビデオモードにより設定の仕方が異なりますし、ビデオBIOSの機能番号も異なりますので、ビデオモード個々のルーチンの解説は省略します。

それでは、videoルーチンからコールされるサブルーチンから、順を追って解説していきます。

表1

画面モードの設定	
設定:	AH=00 AL=画面モード
結果:	リターン値はなし
カーソルの大きさの設定	
設定:	AH=01 CX=カーソルの形状
結果:	リターン値はなし
カーソルの位置の設定	
設定:	AH=02 BH=ページ番号 DH=行位置 DL=桁位置
結果:	不定
カーソルの位置と大きさの取得	
設定:	AH=03 BH=ページ番号
結果:	CH=カーソルの上辺のライン番号 CL=カーソルの下辺のライン番号 DH=行 DL=桁
画面モードの取得	
設定:	AH=0x0F
結果:	AL=画面モード AH=表示桁数 BH=ページ番号
8x14ドットキャラクタセットの書き込み	
設定:	AH=0x1111(0x1101) BL=キャラクタジェネレータ番号
結果:	不定
8x8ドットキャラクタセットの書き込み	
設定:	AH=0x1112(0x1102) BL=キャラクタジェネレータ番号
結果:	不定
VGA情報の取得	
設定:	AH=0x12 BL=0x10
結果:	BH=常時モード 00: カラーモード 01: モノクロモード BL=ビデオメモリの大きさ(VGAメモリ。ビデオメモリ全体ではない) 00: 64Kbytes 01: 128Kbytes 02: 192Kbytes 03: 256Kbytes(通常はこの値になる)
ディスプレイの状態の取得	
設定:	AX=0x1A00
結果:	AL=0x1A BH=セカンダリビデオのコ-ド BL=プライマリビデオのコ-ド 0x00: ディスプレイなし 0x03: EGAと拡張カラーディスプレイ 0x07: VGAとアナログモノクロディスプレイ 0xFF: 不明なディスプレイ (他は省略)

以下のリストの左側の数字は、Linuxのprコマンドを使って、setup.Sのファイルに付けた行番号です。

```
$ pr -n setup.S > setup.S.n
```

とすれば作成されます。

表2

モード	解像度	色数
100h	640 × 400	256
101h	640 × 480	256
102h	800 × 600	16
103h	800 × 600	256
104h	1024 × 768	16
105h	1024 × 768	256
106h	1280 × 1024	16
107h	1280 × 1024	256

表3

VESAコントローラの情報を得る	
設定:	AX=0x4F00
ES:	DI=情報を格納するバッファアドレス
結果:	AX=VESAのステータス
VESAモード情報を得る	
設定:	AX=0x4F01
CX:	モード番号
ES:	DI=情報を格納するバッファアドレス
VESAモードの設定	
設定:	AX=0x4F02
BX:	モード番号
ES:	DI=CRTCのデータを格納するアドレス
VESAプロテクトモードのインターフェイスを得る	
設定:	AX=0x4F0A BL=00
結果:	AX=ステータス ES=テーブルのリアルモードのセグメント DI=テーブルのオフセット CX=プロテクトモードのコードを含むテーブルの長さ

表4

ALの値	意味
AL=0x4F	要求機能がサポートされている
AL != 0x4F	サポートされていない(AX=0x4F00で要求したとき)
AL=00	サービスは成功した
AL=01	サービスは失敗した
AL=02	機能はこのハードウェアではサポートされない
AL=03	現在のビデオモードではこの機能は無効

リスト

```

45                                     ! NORMAL_VGA (-1) = 0xFFFF.....ノーマルVGAモード(80×25)
46                                     ! EXTENDED_VGA (-2) = 0xFFFE.....拡張VGAモード(80×50)
47                                     ! ASK_VGA (-3) = 0xFFFD .....対話モードでユーザーが選択する

```

ビデオモードの4桁コード(対話選択のときに入力できる)

対話選択は、LILOのlilo.confに「vga=ask」を設定すると実行可能になります。

```

59 #define VIDEO_80x25  0x0f00          ! 「0f00」を入力すると80×25を選択する
60 #define VIDEO_8POINT 0x0f01
61 #define VIDEO_80x43  0x0f02
62 #define VIDEO_80x28  0x0f03
63 #define VIDEO_CURRENT_MODE 0x0f04    ! 「0f04」だと80×25を選択する
64 #define VIDEO_80x30  0x0f05
65 #define VIDEO_80x34  0x0f06
66 #define VIDEO_80x60  0x0f07          ! 「0f06」だと80×60を選択する

```

video.Sのメインルーチン

```

110 video: push  ds          ! DSを待避(DS=0x9000)
111         push  ds          ! セグメントレジスタの再設定を行う
112         pop   fs          ! FS=0x9000
113         push  cs
114         pop   ds          ! DS=0x9020
115         push  cs
116         pop   es          ! ES=0x9020
117         xor   ax,ax       ! AX=0
118         mov   gs,ax       ! GS=0
119         cld
120         call  basic_detect ! EGAまたはVGAであるかをチェックする
121 #ifdef CONFIG_VIDEO_SELECT
122         seg   fs          ! FS=x9000
123         mov   ax,[0x01fa] ! AX=[0x9000:0x1FA](すでに選択したVGAモードを読み込む)
124         cmp   ax,#ASK_VGA ! AX=0xFFFF(対話モードで選択?)
125         jz    vid2        ! Yesでvid2(130行目)へ
126         call  mode_set    ! 469行目のmode_setへ(VGAモードをセットする)
127         jc   vid1        ! CY=1、つまり設定が成功したので、vid1(131行目)へ
128         lea  si,badmdt    ! CY=0、つまり設定失敗したので、SI=#badmdt
129         call  prtstr      ! badmdtの文字表示する
130 vid2: call  mode_menu    ! ビデオモードをメニューから選択する
131 vid1:
132 #ifdef CONFIG_VIDEO_RETAIN
133         call  restore_screen ! 画面データをバッファにストアする
134 #endif /* CONFIG_VIDEO_RETAIN */
135 #endif /* CONFIG_VIDEO_SELECT */
136         call  mode_params   ! ビデオモードのパラメータをセットする
137         pop   ds          ! DSを戻す(DS=0x9000、行番号110でpush)
138         ret
139

```

ビデオアダプタをチェックし、VGA/EGAあるいはCGA/MDAの区別をカーネルに知らせる

```

144 basic_detect:
145         seg   fs          ! FS=0x9000
146         movb  [PARAM_HAVE_VGA],#0 ! [PARAM_HAVE_VGA]=0(VGAなし)
147         ! 初期値はVGAなしでセットする
148         mov   ah,#0x12     ! VGA情報の取得機能
149         mov   bl,#0x10     ! AH=0x02、BL=0x10が機能呼出設定
150         int   0x10        ! 実行(結果はBX)
151         seg   fs          ! BL=ビデオメモリの大きさのコード

```

リスト (つづき)

```

152     mov     [PARAM_VIDEO_EGA_BX],bx      ! データ(BX)をストアする
153     cmp     bl,#0x10                      ! BL=0x10?(VGA or EGA?)
154     je      basret                        ! BL=0x10(EGA/VGAではない)ので165行目のbasretへ
155     incb   [adapter]                     ! [adapter]=1(EGAを意味する)
156
157     mov     ax,#0x1a00                    ! EGAとVGAの区別をする
158     int     0x10                          ! ディスプレイの状態の取得
159     cmp     al,#0x1a                      ! AL=0x1A?(0x1AはVGAを意味する)
160     jne    basret                        ! 「No」のときは、basret(165行目)へ
161     seg     fs                             ! VGAなので
162     incb   [PARAM_HAVE_VGA]              ! [PARAM_HAV_VGA]=1にセット
163     incb   [adapter]                     ! [adapter]=2(VGAを意味する)
164
165 basret: ret
166

```

173 mode_paramsは次回に解説します。

<226行目から292行目は省略>

<293行目からのmode_menu、306行目からのlistm、410行目からのmnuselは、次回に解説します。>

```

448 _setrec:   br     setrec                ! setrecにジャンプする
449 _set_80x25: br    set_80x25             ! set_80x25にジャンプする

```

setaliasは、旧バージョンとの互換性のためのルーチンです

```

455 setalias:                                     ! BXにはビデオモードがセットされています。
                                           ! この値は、「NORMAL_VGA=0xFFFF」か「EXTENDED_VGA=0xFFFF」です。
456     mov     ax,#VIDEO_80x25                ! AX=0x0F00(80x25モード)
457     inc     bx                             ! BX=BX+1
458     jz      mode_set                       ! VGAモード=0xFFFFならmode_set(469行目)へ
459     mov     al,#VIDEO_8POINT-VIDEO_FIRST_SPECIAL ! AL=0x0F01 - 0x0F00=0x01
460     inc     bx                             ! BX=BX+1
461     jnz    setbad                          ! VGAモード=0xFFFFでないときは、setbad(487行目)へ
462                                           ! VGAモード=0xFFFFのときは、次のmode_set(469行目)へ進む

```

mode_setは、AX=ビデオモードID番号がセットされてコールされます

設定に成功するとCY=1にセットされます。

```

469 mode_set:
470     mov     bx,ax                          ! BX=AXモード番号をBXにストア
471     cmp     ah,#0xff                       ! AH=0xFF?
472     jz      setalias                       ! Yesでsetalias(455行目)へ
473     test    ah,#VIDEO_RECALC>>8          ! No。AHのbit-7=1?
474     jnz    _setrec                         ! Yesで_setrec(448行目)へ
475     cmp     ah,#VIDEO_FIRST_RESOLUTION>>8 ! AH >= 0x10?
476     jnc    setres                          ! Yesでsetres(532行目)へ
477     cmp     ah,#VIDEO_FIRST_SPECIAL>>8   ! AH=0x0F?
478     jz      setspc                         ! Yesでsetspc(513行目)へ
479     cmp     ah,#VIDEO_FIRST_V7>>8        ! AH=0x09?
480     jz      setv7                          ! Yesでsetv7(440行目)へ
481     cmp     ah,#VIDEO_FIRST_VESA>>8      ! AH >= 0x02
482     jnc    check_vesa                     ! Yesでcheck_vesa(547行目)へ
483     or     ah,ah                          ! AH=0?
484     jz      setmenu                       ! Yesでsetmenu(519行目)へ
485     dec    ah                             ! AH=1?
486     jz      setbios                       ! Yesでsetbios(501行目)へ

```

リスト (つづき)

```

487 setbad:      clc                                ! CY=0
488      movb    [do_restore],#0          ! 画面ストアフラグをクリアする
489      ret

```

< 途中省略 >

スペシャルモードの設定

```

513 setspc: xor   bh,bh                                ! sepec_initsのアドレスリストからBXのモードに対応したアドレスへ
514      cmp    bl,#VIDEO_LAST_SPECIAL-VIDEO_FIRST_SPECIAL ! BH=0。BL=ビデオモードの下位バイト
515      jnc   setbad                                ! Yesでエラー-setbad(487行目)へ
516      add   bx,bx                                ! BX=BX+BX(2倍する)
517      .word 0xa7ff, spec_inits                    ! JMP [BX+spec_inits]
518
! sepec_initsテーブル(621行目)からBXのオフセットを足した位置のアドレスを読み取り、そのアドレスへジャンプする命令です。

```

メニューからビデオモードを選択して、モードテーブルを作成します

```

519 setmenu:
520      or     al,al                                ! AL=0?(ここへ来るときはAH=0)
521      jz    _set_80x25                            ! Yesで、636行目のset_80x25へ(449行目から636行目へ)
522      push  bx                                    ! BX(選択したビデオモード)を待避
523      call mode_table                            ! modelistにモードリストを作る
! SIにはモードリストの最初のアドレスが、DIにはエンドマークのアドレスが、それぞれセットされる
524      pop   ax                                    ! AX=選択したビデオモード
525      shl  ax,#2                                  ! AH=0なのでAX=AL x 4
526      add  si,ax                                  ! SI=SI+AX
! モードリスト内で選択したモードのアドレスをセット
527      cmp  si,di                                  ! SI < エンドマークのアドレス
528      jnc  setbad                                ! Noでエラー、setbad(487行目)へ
529      mov  ax,(si)                                ! AX=選択したモードID
530 _m_s:  jmp  mode_set                              ! 選択したモードのパラメータをセットする

```

スペシャルモードのジャンプアドレステーブル(スペシャルモード設定時に、設定するモードに対応したジャンプアドレスを与えます)

```

621 spec_inits:
622      .word  set_80x25
623      .word  set_8pixel
624      .word  set_80x43
625      .word  set_80x28
626      .word  set_current
627      .word  set_80x30
628      .word  set_80x34
629      .word  set_80x60
630      .word  set_gfx

```

80 x 25モードをセット(すでにセット済の場合は何もしません。ただし、最後にCY=1にセットします)

```

636 set_80x25:
637      mov   [force_size],#0x5019                ! [force_size]=0x5019(80 x 25モード)
638 use_80x25:
639 #ifdef CONFIG_VIDEO_400_HACK
640      mov   ax,#0x1202                            ! AX=0x1202(垂直解像度の設定)
641      mov   bl,#0x30                              ! AL=0x(400ライン)
642      int   0x10                                  ! 実行する
643 #else
644      mov   ah,#0x0f                              ! 現在の画面モードを得る
645      int   0x10                                  ! AL=画面モード(Mode7: 80 x 25)、AH=表示桁数

```

リスト (つづき)

```

646     cmp     ax,#0x5007           ! AX=Mode7( 80×25、白黒 )?
647     jz      st80                ! Yesで662行目へ( CGA / MDA / HGA、EGAMで有効 )
648     cmp     ax,#0x5003           ! AX=Mode3?( 80×25、カラー? )
649     jnz     force3              ! Noのときはforce3( 659行目 )へ
650 st80: cmpb   [adapter],#0       ! Mode3とMode7は80×25モード
651     jz      set80                ! [adapter]=0は、GA / MDA / HGAで80×25モード
652     seg     gs
653     mov     al,[0x0484]          ! AL=( 桁数- 1 ): BIOSパラメータテーブルの値
654     or      al,al                ! AL=0?
655     jz      set80                ! Yesのときは何もせずリターン
656     cmp     al,#24               ! AL=24のときは正しい値( 24+1=25行 )
657     jz      set80                ! 正しいのでset80へ
658 #endif /* CONFIG_VIDEO_400_HACK */
659 force3: DO_STORE
660     mov     ax,#0x0003           ! 画面データをバッファにストアする
661     int     0x10                 ! 80×25、カラーモードで強制セット
662 set80: stc                      ! 設定に成功したのでCY=1にセット
663     ret

```

<途中省略>

setup.Sコードの直後(modelist:)にビデオモードテーブルを作ります

各ビデオモードレコードは以下のように、4bytesで構成されます。

```

.word  MODE-ID      モードIDコード( ライン59のようなコード )
.byte  rows         桁数
.byte  columns      行数

```

このルーチンを終了すると、DIはこのテーブルの終了マークのアドレスを指します。終了マークは「ASK_VGA I=0xFFFD」です。

```

935 mode_table:
936     mov     di,[mt_end]          ! DI=[mt_end]( すでにテーブルがあるときは0以外の値になる )
937     or      di,di                ! DI=0?
938     jnz     mtab1x              ! Noでmtab1x( 951行目 )へ
939     lea     di,modelist         ! DI=modelistのアドレス
940
941     mov     eax,#VIDEO_80x25 + 0x50190000 ! これは80×25モード( ALL )のデータ
942     stosd
943     mov     al,[adapter]        ! CGA / MDA / HGAのときはAL=0
944     or      al,al                ! AL=0?( AL=0でZF=1にセット )
945     jz      mtabe               ! Yesでmtabe( 976行目 )へ
946     dec     al                  ! [adapter]=1?( EGA? )
947     jnz     mtabv              ! No( VGA )でmtabv( 953行目 )へ
948     mov     eax,#VIDEO_8POINT + 0x502b0000 ! EAX= 80×43 EGAモード
949     stosd
950     jmp     mtabe               ! mtabe( 976行目 )へ
951 mtab1x: jmp     mtab1          ! mtab1( 1004行目 )へ

```

VGAモードのテーブル作成

```

953 mtabv: lea     si,vga_modes      ! 標準VGAモードのテーブル
954     mov     cx,#vga_modes_end-vga_modes ! CX=VGAテーブルのバイト数
955     rep
956     movsb
957
958     cmpb   [scanning],#0        ! モードスキャンを要求しているか?
959     jz      mscan1              ! Noでmscan1()へ
960     call   mode_scan            ! Yesでmode_scan( 1125行目 )を実行する
961 mscan1:

```

リスト (つづき)

```

962                                     ! 今回はCONFIG_VIDEO_LOCAL、VESA、SVGA、COMPACT=0と仮定し省略し
                                     ! ます。

< 963行目 ~ 975行目、979行目 ~ 999行目は省略 >

976 mtabe:
977
978 #ifdef CONFIG_VIDEO_COMPACT
1000 #endif /* CONFIG_VIDEO_COMPACT */
1001
1002     mov     (di),#ASK_VGA           ! modelistに終了マーク(0xFFFFD)を書き込む
1003     mov     [mt_end],di           ! [mt_end]=終了マークのアドレス
1004 mtab1: lea     si,modelist       ! SI=modelistテーブルの頭のアドレス
1005 ret0:  ret                       ! DI=modelistテーブルの最後のアドレス

1006

```

標準VGAのビデオモードテーブル(modelistテーブルを作成するのに使われるテーブル)

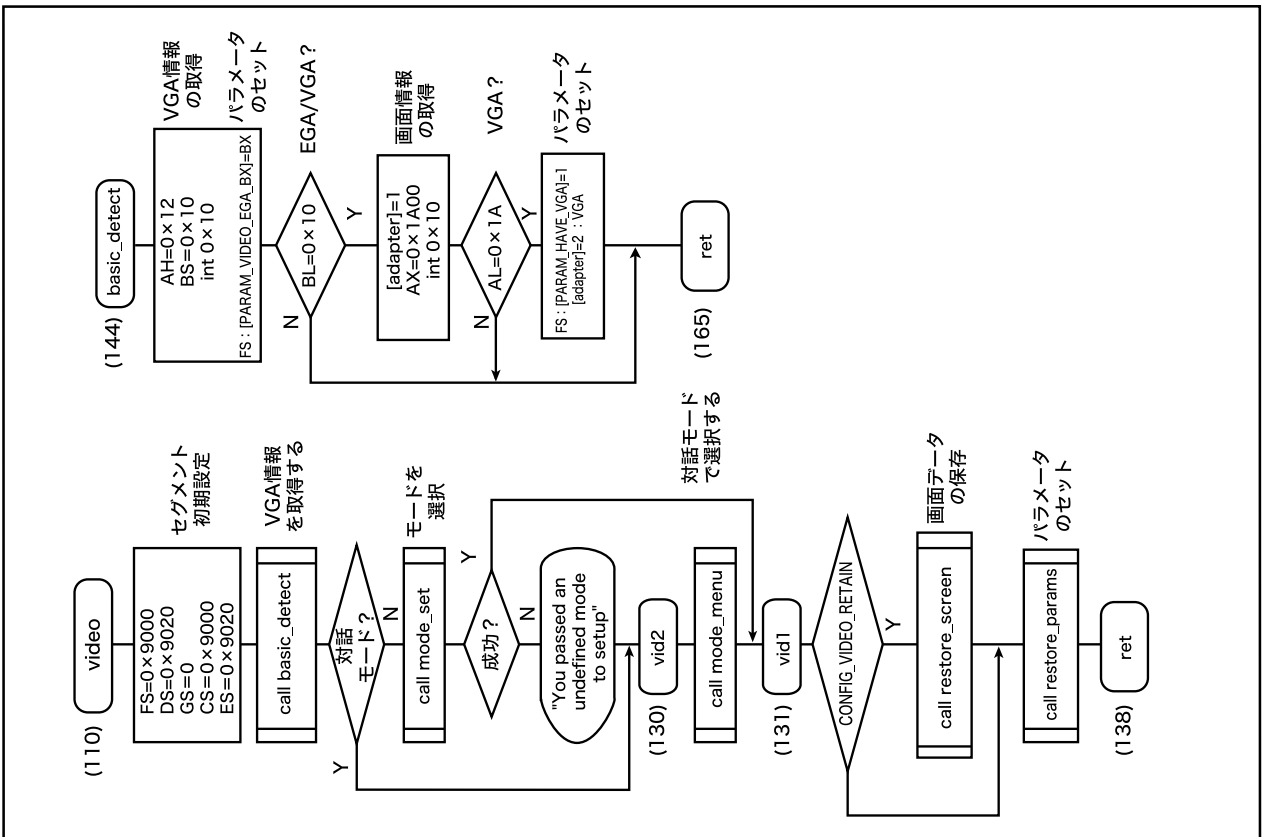
```

1009 vga_modes:
1010     .word  VIDEO_8POINT
1011     .word  0x5032                ! 80 x 50
1012     .word  VIDEO_80x43
1013     .word  0x502b                ! 80 x 43
1014     .word  VIDEO_80x28
1015     .word  0x501c                ! 80 x 28
1016     .word  VIDEO_80x30
1017     .word  0x501e                ! 80 x 30
1018     .word  VIDEO_80x34
1019     .word  0x5022                ! 80 x 34
1020     .word  VIDEO_80x60
1021     .word  0x503c                ! 80 x 60
1022 #ifdef CONFIG_VIDEO_GFX_HACK
1023     .word  VIDEO_GFX_HACK
1024     .word  VIDEO_GFX_DUMMY_RESOLUTION
1025 #endif
1026 vga_modes_end:

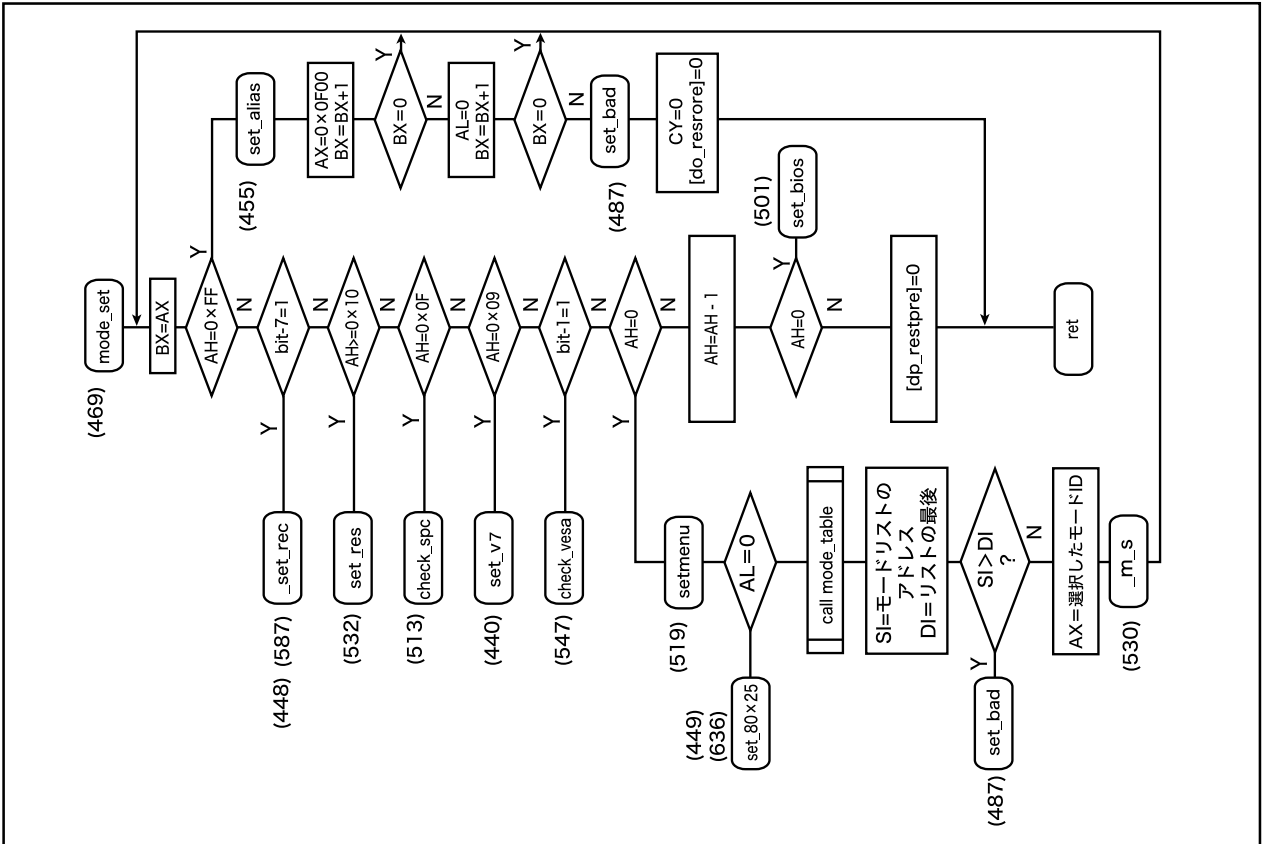
```

R	E	S	O	U	R	C	E
[1]	VESA BIOSに関する資料 「VESA BIOS EXTENSION (VBE) Core Functions Standard Ver.3.0 (VESA)			[4]	PCの割り込み処理LS(8259)に関する資料 三菱半導体データブック(三菱電機)		
[2]	PC互換機のビデオシステムに関する参考文献 「Programer's Guide To PC & PS/2 VIDEO SYSTEMS」 (Richard Wilton著、SE訳、Microsoft Press) 「DOS/Vプログラミングガイド(最上晃 著、アスキー出版)			[5]	PC互換機のアーキテクチャに関する参考文献 「The Programmer's PC Sourcebook」 (Thom Hogan著、SE編集部訳、Microsoft Press)		
[3]	x86 CPUのプロテクトモードに関する参考文献 「初めて読む486(蒲池輝尚著 / アスキー出版) 「80486の使い方(W.B.スルヤント著 / オーム社) 「Inter Architecture Software Developer's Manual Volume 3(Intel)			[6]	PC互換機のシステムBIOSに関する参考文献 「Phenix BIOS 4.0 Revision 6 User's Manual (Phoenix Technologies LTD) 「BIOS Enhanced Disk Driver Specification version.3.0 Rev.0.8 (Phoenix Technolginies LTD)		

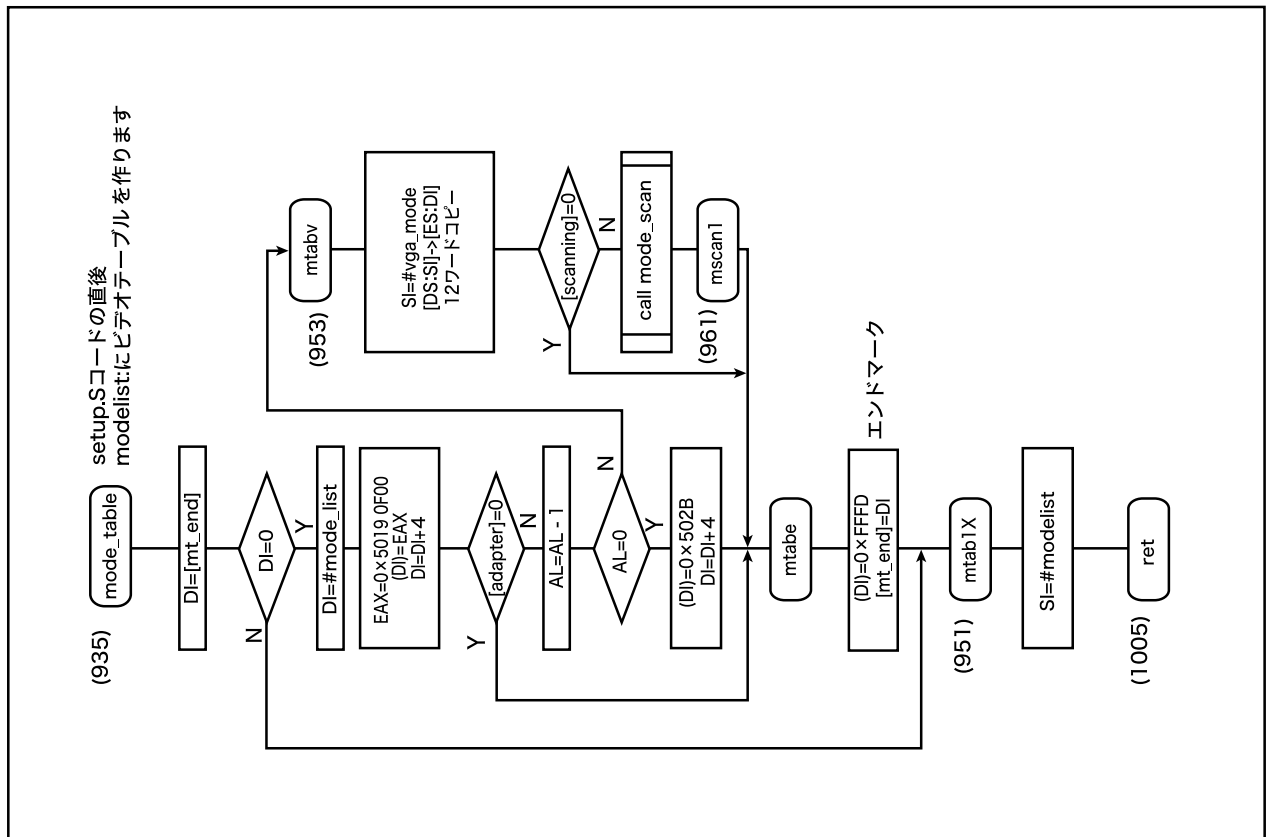
フローチャート1



フローチャート2



フローチャート3



フローチャート4

