



# 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(2)

芳之内弘

前回は、Setup.Sにインクルードされているvideo.Sの解説を始めました。今回はこれの残りの主な部分を調べます。

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

video.Sで使われているビデオBIOSサービスについては、前号の記事を参考にしてください。

からコールされる「basic\_detect」と「mod\_set」について解説しました。今回は、videoがコールする「mode\_params」と「mode\_menu」について解説していきます。

なお、以下のリストの左側の数字は、Linuxのprコマンドを使い、

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

として、setup.Sのファイルに付けた行番号です。

### BOOT video.Sの解説(2)

前回は、video.Sのメインルーチンである「video」と、video

リスト

カーネルへ渡すビデオパラメータをセット(ページ番号、ビデオモード、行数、桁数、フォントサイズ、カーソル位置)

```
173
174 mode_params:
175 #ifdef CONFIG_VIDEO_SELECT
176     cmpb    [graphic_mode],#0           ! グラフィックモードか?
177     jnz     mopar_gr                    ! Yesでmopar_gr( 225 )へ
178 #endif                                  ! mopar_grは今回は省略する
179
180     mov     ah,#0x03                    ! カーソル位置の取得
181     xor     bh,bh                        ! DL = 行
182     int     0x10                         ! DL = 桁
183     seg     fs
184     mov     [PARAM_CURSOR_POS],dx       ! カーソル位置をセット
185
186     mov     ah,#0x0f                    ! 画面モードの取得
187     int     0x10                         ! AL = 画面モード
188     seg     fs
189     mov     [PARAM_VIDEO_PAGE],bx       ! BH = ページ番号、ページ番号セット
190     seg     fs
191     mov     [PARAM_VIDEO_MODE],ax       ! モードと桁数をセット
192     cmp     al,#7                        ! AL=7?(VGA?)
193     jnz     mopar0                       ! YesはHDA/MDAでmopar( 195 )へ
194     mov     [video_segment],#0xb000     ! NoはVGAなのでセグメント=0xb000
195 mopar0: seg     gs                       ! BIOSパラメータテーブルより
```

### リスト

```

196     mov     ax, [0x485]           ! フォントサイズを得る
197     seg     fs
198     mov     [PARAM_FONT_POINTS], ax           ! フォントサイズをセット
199                                     ! これはEGA / VGAのみ有効である
200     mov     ax, [force_size]       ! AX=[fprce_soze]
201     or      ax, ax                 ! AX=0でZF=1にセット
202     jz      mopar1                 ! [force_size]=0でmopar1( 209 )へ
203     seg     fs
204     mov     [PARAM_VIDEO_COLS], ah   ! [force_size]の桁数をセット
205     seg     fs
206     mov     [PARAM_VIDEO_LINES], al  ! [force_size]の行数をセット
207     ret
208
209 mopar1: mov     al, #25             ! [force_size]=0なのでAX( 行数 )= 25にする
210     cmpb   [adapter], #0          ! [adapter]=0はCGA / MDA / HGAを意味する
211     jz      mopar2                 ! そのときは行数は25でなければならない
212     seg     gs
213     mov     al, [0x484]           ! 表示桁数を使う
214     inc     al                     ! パラメータ = 桁数 - 1
215 mopar2: seg     fs
216     movb   [PARAM_VIDEO_LINES], al  ! 桁数をセット
217     ret
218
219 #ifdef CONFIG_VIDEO_SELECT
225 mopar_gr:
<今回は省略>

```

### 対話にてビデオモードを選択する

ビデオモードをユーザーがメニューから選択する。Enterキーでメニュー表示。Spaceキーかタイムアウト( 30秒)でデフォルトのビデオモードになる。

```

293 mode_menu:
294     lea     si, keymsg             ! SI="Return/Space/Timeout"メッセージのアドレス
295     call   prtstr                 ! メッセージを表示する
296     call   flush                 ! キーボードバッファをクリアする
297 nokey:  call   getkt              ! キー入力またはタイムアウトを待つ
298     cmp     al, #0x0d             ! Enterキー?
299     je      listm                 ! Yesで対話モードの選択listm( 306 )へ
300     cmp     al, #0x20             ! Spaceキー?
301     je      defmd1               ! Yesでフォルトモードになる( ret )
302     call   beep                 ! BEEP音出力( 無効入力警告音 )
303     jmp     nokey                ! nokey( 297 )から繰り返す
304 defmd1: ret
305
306 listm:  call   mode_table         ! アドレスmodelistにビデオモードテーブルを作る
307 listm0: lea     si, name_bann     ! SI="Video adapter "メッセージのアドレス
308     call   prtstr                 ! メッセージを表示する
309     mov     si, [card_name]       ! SI=[card_name]
310     or      si, si                 ! SI=0でZF=1にセットされる
311     jnz     an2                  ! SI !=0 のときはan2( 321 )へ
312     mov     al, [adapter]         ! AL=[adapter]
313     lea     si, old_name          ! SI="CGA/MDA/HGA"メッセージのアドレス
314     or      al, al                 ! AL=[adapter]=0でZF=1にセットされる
315     jz      an1                  ! AL=[adapter]=0でan1( 323 )へ
316     lea     si, ega_name         ! SI="EGA"メッセージのアドレス
317     dec     al                     ! AL=[adapter]=1?( AL=1でZF=1にセットされる )
318     jz      an1                  ! Yesでan1( 323 )へ
319     lea     si, vga_name         ! SI="VGA"メッセージのアドレス
320     jmp     an1                  ! an1( 323 )へ
321 an2:    call   prtstr             ! SIで示すメッセージを表示する
322     lea     si, svga_name        ! SI=" VGA"メッセージのアドレス
323 an1:    call   prtstr             ! SIでポイントするメッセージを表示する
324     lea     si, listhdr          ! SI="改行MODE: COLSxROWS: 改行"メッセージのアドレス
325     call   prtstr               ! SIでポイントするメッセージを表示する
326     mov     dl, #0x30            ! DL=0x30( モード番号="0" )
327     lea     si, modelist         ! SI=モードテーブルの先頭アドレス
328 lm1:    cmp     (si), #ASK_VGA   ! (SI)=0xFFFF( 終了マーク )?
329     jz      lm2                  ! Yesでlm2( 352 )へ
330     mov     al, dl               ! AL=DL( メニュー選択番号 )
331     call   prtchr                ! ALの文字( 選択番号 )を表示する
332     call   prtsp2               ! スペースを2個表示する

```

## リスト (つづき)

```

333 lodsw
334 call prthw
335 call prtsp2
336 mov al,(si+1)
337 call prtdec
338 mov al,#0x78
339 call prtchr
340 lodsw
341 call prtdec
342 mov al,#0x0d
343 call prtchr
344 mov al,#0x0a
345 call prtchr
346 inc dl
347 cmp dl,#0x3a
348 jnz lm1
349 mov dl,#0x61
350 jmp lm1
351
352 lm2: lea si,prompt
353 call prtstr
354 lea di,edit_buf
355 lm3: call getkey
356 cmp al,#0x0d
357 jz lment
358 cmp al,#0x08
359 jz lmbs
360 cmp al,#0x20
361 jc lm3
362 cmp di,#edit_buf+4
363 jz lm3
364 stosb
365 call prtchr
366 jmp lm3
! AX=(SI)=リストのビデオモード(SI=SI+2)
! モード記号(4桁)を表示する
! スペースを2個表示する
! AL=リストの次の文字
! 文字を表示する(桁数)
! AL='x'
! 「x」を表示
! AX=(SI) 次のワード(SI=SI+2)
! 文字を表示(行数)
! AL=改行コード
! 改行する
! AL=復帰コード
! 復帰する
! DL=DL+1(つぎのモード)
! DL=0x3A?( "0" - "9"の範囲か?)
! "0" - "9"ならlm1(328)へ
! 0x3Aのときは0x61("a")へ変更する
! lm1(328)へ戻る
! SI="改行Enter mode number"メッセージのアドレス
! メッセージを表示する
! DI=エディットバッファのアドレス
! キー入力待
! Enterキー?
! Yesでlment(378)へ
! BackSpace?
! Yesでlmbs(368)へ
! AL>=0x20 有効なキー?
! No(無効)でlm3(355)へ。再入力する。
! 入力は5文字目?
! Yesでlm3(355)へ。再入力。
! Noで(DI)=AL、DI=DI+1、入力文字をエディットバッファにストアする
! 入力文字(ALの文字)を表示する
! lm3(355)へ。繰り返す。
キーがBackSpaceのとき、カーソルを戻し1文字消す
368 lmbs: cmp di,#edit_buf
369 jz lm3
370 dec di
371 mov al,#0x08
372 call prtchr
373 call prtspc
374 mov al,#0x08
375 call prtchr
376 jmp lm3
377
378 lment: movb (di),#0
379 lea si,crlft
380 call prtstr
381 lea si,edit_buf
382 cmpb (si),#0
383 jz lmdef
384 cmpb (si+1),#0
385 jz mnusel
386 cmp (si),#0x6373
387 jnz lmhx
388 cmp (si+2),#0x6e61
389 jz lmscan
390 lmhx: xor bx,bx
391 lmhex: lodsb
392 or al,al
393 jz lmuse1
394 sub al,#0x30
395 jc lmbad
396 cmp al,#10
397 jc lmhx1
398 sub al,#7
! 最初の文字?
! Yesでlm3(355)へ(再入力へ)
! NoでDI=DI - 1(1つ戻る)
! AL=BS(バックスペース)コード
! カーソルを1つ戻す
! スペースを1つ出力する
! AL=BS(バックスペースコード)
! カーソルを1つ戻す
! lm3(355)へ
! (DI)=0、DI=エディットバッファの現在位置
! SI=改行復帰メッセージのアドレス
! 改行復帰する
! SI=#edit_buf(エディットバッファ先頭アドレス)
! バッファは空?
! 空だとlmdef(434)へ(retを実行する)
! 2文字目は0?(1文字入力か?)
! Yesでmnusel(410)へ
! (SI)="scan"の2文字?
! Noならlmhx(391)
! (SI)="scan"の2文字?
! Yes("scan")でlmscan(427)へ
! No scan BX,bx
! AX=(SI)=入力文字、SI=SI+1
! AL=0でZF=1にセット
! AL=入力文字=0でlmuse1(407)
! AL=AL-0x30(asciiを数字に変換)
! 入力が"0"以下は不可につきlmbad(423)へ
! AL<10? 数字?
! Yesでlmhx1(404)へ
! No: 入力は英文字、AL=AL-7、入力値-0x37
! "A" - "F"を0xA - 0xFに変換する
! 小文字を大文字へ変換する
! AL<0x0A?
! Yesでエラーlmbad(423)へ
399 and al,#0xdf
400 cmp al,#10
401 jc lmbad

```

### リスト (つづき)

```

402      cmp     al,#16          ! AL > 0x0F ? ( 0x0A<=AL<=0x0FがOK )
403      jnc     lmbad          ! Yesでエラー-lmbad( 423 )へ
404 lmhx1: shl     bx,#4        ! BXの下位デジットを上位へ1桁分移動する
405      or      bl,al          ! BL=BL or AL=( 前回のデジット + 今回のデジット )
                                ! BXには4桁の16進コードがセットされる
406      jmp     lmhex         ! lmhex( 391 )へ戻る
407 lmuse1: mov   ax,bx        ! AX=BX( 4桁コード )
408      jmp     lmuse         ! lmuse( 421 )へ

1文字(メニュー番号)入力でメニューを選択するルーチン( 選択文字は0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,...z )
410 mnusel: lodsb          ! AL=(SI)エディットバッファよりSI=SI+1
411      xor     ah,ah         ! AH=0
412      sub     al,#0x30      ! AL < 0x30( "0" )?
413      jc      lmbad        ! Yesでlmbad( 423 )
414      cmp     al,#10       ! 数字?
415      jc      lmuse        ! Yesでlmuse( 421 )へ
416      cmp     al,#0x61-0x30 ! NoでALは英字。ではAL < "a" ?
417      jc      lmbad        ! Yesでエラー。lmbad( 423 )へ
418      sub     al,#0x61-0x30-10 ! AL=AL-0x26。入力文字 - 0x56
419      cmp     al,#36       ! AL>0x24、入力 > "z" ?
420      jnc     lmbad        ! Yesでエラー-lmbad( 423 )へ
421 lmuse:  call   mode_set    ! 選択したモードの設定
422      jc      lmdef        ! 設定に成功したのでlmdef( 434 )へ。ret実行。
423 lmbad:  lea   si,unknt     ! SI="Unknown menu ID. Try again."メッセージのアドレス
424      call   prtstr        ! メッセージの表示
425      br     lm2           ! lm( 352 )から繰り返す

```

### "scan"が入力されたので[scanning]=1にセットする

```

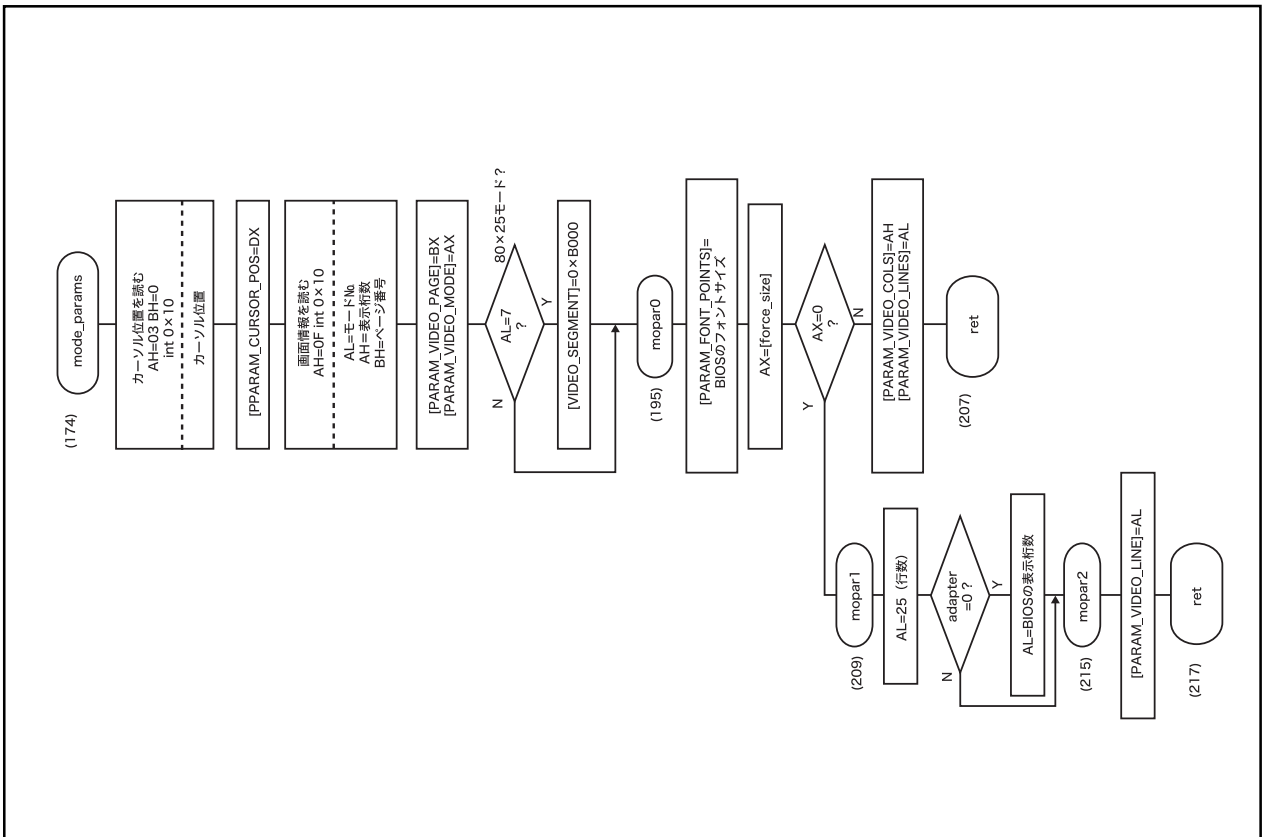
426
427 lmscan: cmpb    [adapter],#0          ! EGA or VGAはサポートされているか?
428      jz      lmbad          ! Noならscanはできないのでlmbad( 423 )へ
429      mov     [mt_end],#0         ! モードテーブルクリアする
430      movb   [scanning],#1       ! モードスキャンフラグをセットする
431      call   mode_table         ! モードテーブルを作る
432      br     listm0            ! listm( 307 )へ戻る
433
434 lmdef:  ret
439
440 setv7:
441      DO_STORE
442      sub     bh,#VIDEO_FIRST_V7>>8
443      mov     ax,#0x6f05
444      int    0x10
445      stc
446      ret
447

```

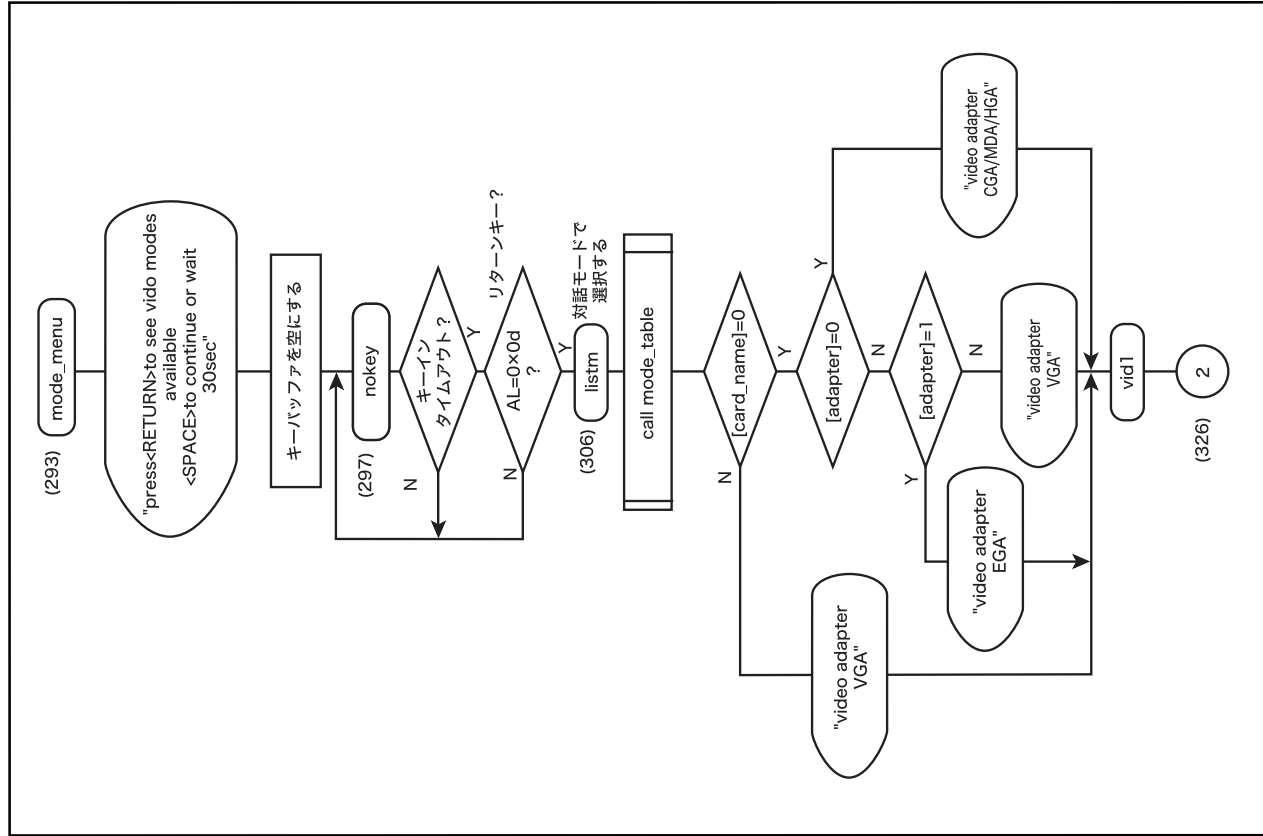
今回で、setup.( videoe.Sを含む )の解説は終了します。次回からは、head.Sに  
よりカーネルを展開して、初期化するルーチンに入ります。

R	E	S	O	U	R	C	E
[ 1 ]	PC互換機のアーキテクチャに関する参考文献 「プログラマーズPCソースブック」 ( Thom Hogan 著 / インプレス )			[ 4 ]	PCの割り込み処理LSK( 8259 )に関する資料 三菱半導体データブック( 三菱電機 )		
[ 2 ]	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 )			[ 5 ]	PC互換機のシステムBIOSに関する参考文献 「Phenix BIOS 4.0 Rivision 6 User's Manual」 ( Phoenix Technologies LTD ) 「BIOS Enhanced Disk Driver Specification version.3.0 Rev.0.8」 ( Phoenix Technollogines LTD )		
[ 3 ]	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 )			[ 6 ]	VESA BIOSに関する資料 「VESA BIOS EXTENSION (VBE) Core Functions Standard Ver..3.0( VESA )		

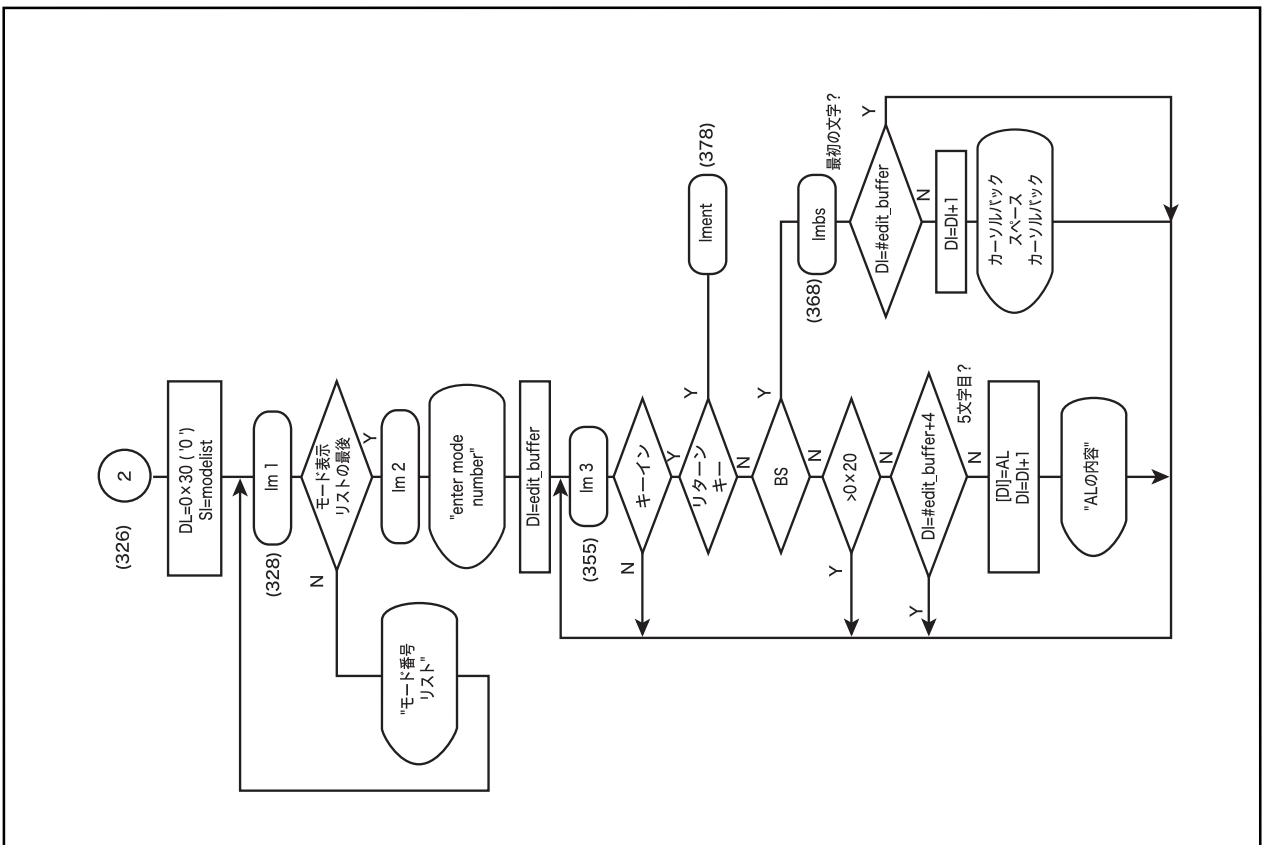
フローチャート1



フローチャート2



フローチャート3



フローチャート4

