


```

0000 1: ;-----
0000 2: ;
0000 3: ;      Copyright 2010 Intergraded Logic Systems
0000 4: ;      Source Code is Copyright Stephen J. Car
0000 5: ;
0000 6: ;
0000 7: ;      Please do not share this source!
0000 8: ;-----
0000 9: ;      os_menu.com used to program the Smartos for real.
0000 10: ;
0000 11: ;-----
0000 12: ;      Real.dos os_menu.COM
0000 13: ;      This program is to replace
0000 14: ;      os_menu.com
0000 15: ;
0000 16: ;      Code History.  Init design, 10/30/1998
0000 17: ;      last updated 11/05/2005 sjc
0000 18: ;      reformatted source 4/05/2010 sjc
0000 19: ;      update os detection 06/04/2010 sjc
0000 20: ;
0000 21: ;
0000 22: ;Updated the screen Print Utility so it will work with the
0000 23: ; This change will also help "Print P:"      8/1
0000 24: ;
0000 25: ;
0000 26: ;Also added code for Control+1 pause of screen to the high
0000 27: ;handling.      8/24/2012 sjc
0000 28: ;
0000 29: ;
0000 30: ;
0000 31: ;Added code so this file cannot run on the AtariWin800 Emu
0000 32: ;      8/24/2012 sjc
0000 33: ;;
0000 34: ;
0000 35: ;
0000 36: ;
0000 37: ;      ~~~~~
0000 38: ;-----
0000 39: ; Notes: o This source code MAY NOT be placed for download
0000 40: ;      o "mea" is a macro that loads the address of the
0000 41: ;      into the pointer specified by the second field.
0000 42: ;-----
0000 43: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 44: ;      XASM st and IBM VERSIONS
0000 45: ;-----
0000 46:
0000 47: com_file_name: .macro
0000 48: ;      dc.b 13,"filename.ext",$9b
0000 49: ;      dc.b 12,"Os_menu.com ", $9b
0000 50: ;      .endm
0000 51:
0000 52: ;
0000 53: ;~~~~~

```

```
0000 54: ; File revision history. Version Number in hex
0000 55: ;
0000 56: file_ver:      equ  $18
0000 57: ;~~~~~
0000 58: ; the Month in dec for the first time this revision
0000 59: ;was compiled
0000 60: ;
0000 61: c_month:      equ  10
0000 62: ;~~~~~
0000 63: ; the day in dec for the first time this revision
0000 64: ;was compiled
0000 65: c_day:      equ  15
0000 66: ;~~~~~
0000 67: ; the year in dec for the first time this revision
0000 68: ;was compiled
0000 69: c_year:      equ  2012
0000 70: ;~~~~~
0000 71: ; Type of compiler used in program
0000 72: ;
0000 73: ; 0 = unknown Compiler
0000 74: ; 1 = xasm
0000 75: ; 2 = mac_65
0000 76: ; 3 = basic
0000 77: ; 4 = compiled basic xl
0000 78: ; 5 = C65
0000 79: ; 6 = Action
0000 80: ;
0000 81: ; We can add more as time goes on!
0000 82: ;
0000 83: ;
0000 84: ;
0000 85: xasm:      equ  1
0000 86: mac_65:    equ  2
0000 87: basic:     equ  3
0000 88: basicxl:   equ  4
0000 89: c65:      equ  5
0000 90: action:    equ  6
0000 91:
0000 92: file_compiler:  equ  xasm
0000 93: ;~~~~~
0000 94: ; is this relocatable code ?
0000 95: ; anyother value other than 1 or 2 would be unknown
0000 96: r..yes:    equ  1
0000 97: r..no:     equ  2
0000 98: ;
0000 99: ;~~~~~
0000 100: ; Gotta define if it can be relocated
0000 101: l_relocatable:  equ  r..no
0000 102: ;
0000 103: ;
0000 104: ;~~~~~
0000 105: ; Gotta define if it can be relocated
0000 106: r..crl:     equ  $7d
```

```

0000    107: r.crlf:      equ   $9b
0000    108: r.space:      equ   $20
0000    109:
0000    110: l_frstscreenbyte: equ   r..space
0000    111: ;~~~~~
0000    112: ; The language the output file is in.
0000    113: ;
0000    114: ;
0000    115: ; 0 =   Undefined
0000    116: ; 1 =   English
0000    117: ; 2 =   German
0000    118: ;
0000    119: r..language:   equ   1
0000    120: ;~~~~~
0000    121:
0000    122:
0000    123:
0000    124: ;
0000    125: ;
0000    126: on:           equ   $01
0000    127: off:          equ   $00
0000    128: ;
0000    129: ;
0000    130: ;-----
0000    131: ;
0000    132: cart_a:      equ   $a000
0000    133: cart_l:      equ   $1fff
0000    134: os_l:        equ   $4000
0000    135: selftest:    equ   $e471
0000    136: reset:       equ   $e474
0000    137: cold_start:  equ   $e477
0000    138: memory:      equ   $4000
0000    139: os:          equ   $c000
0000    140: control:     equ   $d380
0000    141: boot.:       equ   $09
0000    142:
0000    143:
0000    144:
0000    145: kbd:         equ   $30
0000    146:
0000    147: i.in_ptr:    equ   $a0
0000    148: i.s_ptr:     equ   $a2
0000    149: i.d_ptr:     equ   $a4
0000    150: i.fn_ptr:    equ   $a6
0000    151: i.pr_ptr:    equ   $a8
0000    152:
0000    153: i.crc_ptr:   equ   $8b
0000    154: i.c_ptr:     equ   $9e
0000    155: i.st_ptr:    equ   $cc
0000    156: i.v_ptr:     equ   $ca
0000    157: ;
0000    158: ;
0000    159: os_rom:      equ   $40

```

```

0000    160: os_ram:          equ   $80
0000    161: os_ram1:             equ   $01
0000    162: os_write:            equ   $20
0000    163: ;
0000    164: ;-----Macros-----
0000    165: ;
0000    166: show_hex_word      .macro field
0000    167:                    lda    %1
0000    168:                    ldx    %1+1
0000    169:                    jsr    h_card
0000    170:                    .ENDM
0000    171: ;
0000    172: ;-----end of macros-----
0000    173: ;
0000    174:
0000    175:                    .include equates
0000    176:                    .include globals
0000    177:                    .include macros
0000    178:
2300    179:                    .org   $2300
2300    180: ;
2300    181: win_start:
2300    182: header_info:
2300    183:                    .include header
25a7    184:
25a7    185: A_start:
25a7 20b89f 186:                    jsr    mem_test
25aa 4c0029 187:                    jmp    start
25ad    188:
25ad    189:
25ad    190: parm_1:
25ad 000000 191:                    dc.b   0,0,0,0,0,0,0,0,0
25b7 000000 192:                    dc.b   0,0,0,0,0,0,0,0,0
25c1 000000 193:                    dc.b   0,0,0,0,0,0,0,0,0
25cb 000000 194:                    dc.b   0,0,0,0,0,0,0,0,0
25d5 000000 195:                    dc.b   0,0,0,0,0,0,0,0,0
25df 000000 196:                    dc.b   0,0,0,0,0,0,0,0,0
25e9 000000 197:                    dc.b   0,0,0,0,0,0,0,0,0
25f3 000000 198:                    dc.b   0,0,0,0,0,0,0,0,0
25fd    199:
25fd    200: parm_2:
25fd 000000 201:                    dc.b   0,0,0,0,0,0,0,0,0
2607 000000 202:                    dc.b   0,0,0,0,0,0,0,0,0
2611 000000 203:                    dc.b   0,0,0,0,0,0,0,0,0
261b 000000 204:                    dc.b   0,0,0,0,0,0,0,0,0
2625 000000 205:                    dc.b   0,0,0,0,0,0,0,0,0
262f 000000 206:                    dc.b   0,0,0,0,0,0,0,0,0
2639 000000 207:                    dc.b   0,0,0,0,0,0,0,0,0
2643 000000 208:                    dc.b   0,0,0,0,0,0,0,0,0
264d    209: instr:
264d 000000 210:                    dc.b   0,0,0,0,0,0,0,0,0
2657 000000 211:                    dc.b   0,0,0,0,0,0,0,0,0
2661 000000 212:                    dc.b   0,0,0,0,0,0,0,0,0

```

```

266b 000000 213:          dc.b  0,0,0,0,0,0,0,0,0
2675 000000 214:          dc.b  0,0,0,0,0,0,0,0,0
267f 000000 215:          dc.b  0,0,0,0,0,0,0,0,0
2689 000000 216:          dc.b  0,0,0,0,0,0,0,0,0
2693 000000 217:          dc.b  0,0,0,0,0,0,0,0,0
269d      218: in_str:
269d 000000 219:          dc.b  0,0,0,0,0,0,0,0,0
26a7 000000 220:          dc.b  0,0,0,0,0,0,0,0,0
26b1 000000 221:          dc.b  0,0,0,0,0,0,0,0,0
26bb 000000 222:          dc.b  0,0,0,0,0,0,0,0,0
26c5 000000 223:          dc.b  0,0,0,0,0,0,0,0,0
26cf 000000 224:          dc.b  0,0,0,0,0,0,0,0,0
26d9 000000 225:          dc.b  0,0,0,0,0,0,0,0,0
26e3 000000 226:          dc.b  0,0,0,0,0,0,0,0,0
26ed      227: ;
26ed      228: ;
26ed      229:
26ed      230: ;-----
26ed      231: ;  ERROR NUMBER
26ed      232: ;this take's the value of the a reg
26ed      233: ;and put a hex number to the screen
26ed      234: ;-----
26ed      235: ;
26ed 00      236: h_cardd:          dc.b  $00
26ee      237: ;
26ee      238: ; print a hex word value
26ee      239: ; low byte in a reg x is high byte
26ee      240: ;
26ee 8ded26 241: h_card:          sta  h_cardd
26f1 8a      242:          txa
26f2 20fc26 243:          jsr  drive_error
26f5 aded26 244:          lda  h_cardd
26f8 20fc26 245:          jsr  drive_error
26fb 60      246:          rts
26fc      247: ;
26fc      248: ;
26fc 48      249: drive_error:      pha
26fd 4a      250:          lsr
26fe 4a      251:          lsr
26ff 4a      252:          lsr
2700 4a      253:          lsr
2701 200727 254:          jsr  .a
2704 68      255:          pla
2705 290f     256:          and  #$0f
2707 c90a     257: .a:          cmp  #$0a
2709 b004     258:          bcs  .b
270b 0930     259:          ora  #$30
270d d002     260:          bne  .c
270f 6936     261: .b:          adc  #$36
2711 20568d 262: .c:          jsr  sprint
2714 60      263:          rts
2715      264: ;
2715      265: ;

```

```

2715      266: ;-----
2715      267: ;time date off
2715      268: ;-----
2715      269: ;
2715      270: ;
2715      271: td:
2715 ad01d3 272:          lda    portb
2718 48      273:          pha
2719 29fe 274:          and    #$fe
271b 8d01d3 275:          sta    portb
271e 20c6ff 276:          jsr    vtdon
2721 68      277:          pla
2722 8d01d3 278:          sta    portb
2725 60      279:          rts
2726      280: ;
2726      281:
2726      282: ;
2726      283: pr_byte:
2726 a200 284:          ldx    #$00
2728      285: pr_card:
2728 85d4 286:          sta    fr0
272a 86d5 287:          stx    fr0+1
272c a900 288:          lda    #$00
272e 85f2 289:          sta    cix
2730 20aad9 290:          jsr    ifp
2733 20e6d8 291:          jsr    fasc
2736 a000 292:          ldy    #$00
2738      293: .a:
2738 b1f3 294:          lda    (inbuff),y
273a 08      295:          php
273b 297f 296:          and    #$7f
273d c8      297:          iny
273e 20568d 298:          jsr    echo
2741 28      299:          plp
2742 10f4 300:          bpl    .a
2744 60      301:          rts
2745      302: ;
2745      303: ;-----
2745      304: ;this one make's a nice tone
2745      305: ;-----
2745      306: bell:
2745 8d7127 307:          sta    bell_m3
2748 8c7227 308:          sty    bell_m3+1
274b 8e7327 309:          stx    bell_m3+2
274e a0af 310:          ldy    #$af
2750 a914 311:          lda    #$14
2752 8d00d2 312:          sta    $d200
2755 8c01d2 313: bell_m1:          sty    $d201
2758 a900 314:          lda    #$00
275a 8514 315:          sta    $14
275c a514 316: bell_m2:          lda    $14
275e c902 317:          cmp    #$02
2760 90fa 318:          bcc    bell_m2

```

```

2762 88 319:      dey
2763 c0a0 320:      cpy  #$a0
2765 b0ee 321:      bcs  bell_m1
2767 ad7127 322:      lda  bell_m3
276a ac7227 323:      ldy  bell_m3+1
276d ae7327 324:      ldx  bell_m3+2
2770 60 325:      rts
2771 000000 326: bell_m3:      dc.b  $00,$00,$00
2774      327: ;
2774      328: ;-----
2774      329: ; BIT LOGIC LOAD ACC
2774      330: ; LOAD Y REG WITH BIT #
2774      331: ;-----
2774      332: check_bit:
2774 8de227 333:      sta  check_hold
2777 8ce327 334:      sty  check_hold+1
277a a000 335:      ldy  #$00
277c a900 336:      lda  #$00
277e 99e427 337: check1:      sta  check_line,y
2781 c8 338:      iny
2782 c008 339:      cpy  #$08
2784 90f8 340:      bcc  check1
2786 ade227 341:      lda  check_hold
2789 18 342:      clc
278a 0a 343:      asl
278b 9006 344:      bcc  c_bit_1
278d a001 345:      ldy  #$01
278f 8ceb27 346:      sty  check_line+7
2792 18 347:      clc
2793      348: c_bit_1:
2793 0a 349:      asl
2794 9006 350:      bcc  c_bit_2
2796 a001 351:      ldy  #$01
2798 8cea27 352:      sty  check_line+6
279b 18 353:      clc
279c      354: c_bit_2:
279c 0a 355:      asl
279d 9006 356:      bcc  c_bit_3
279f a001 357:      ldy  #$01
27a1 8ce927 358:      sty  check_line+5
27a4 18 359:      clc
27a5      360: c_bit_3:
27a5 0a 361:      asl
27a6 9006 362:      bcc  c_bit_4
27a8 a001 363:      ldy  #$01
27aa 8ce827 364:      sty  check_line+4
27ad 18 365:      clc
27ae      366: c_bit_4:
27ae 0a 367:      asl
27af 9006 368:      bcc  c_bit_5
27b1 a001 369:      ldy  #$01
27b3 8ce727 370:      sty  check_line+3
27b6 18 371:      clc

```



```

27b7      372: c_bit_5:
27b7 0a    373:      asl
27b8 9006   374:      bcc  c_bit_6
27ba a001   375:      ldy  #$01
27bc 8ce627 376:      sty  check_line+2
27bf 18     377:      clc
27c0      378: c_bit_6:
27c0 0a     379:      asl
27c1 9006   380:      bcc  c_bit_7
27c3 a001   381:      ldy  #$01
27c5 8ce527 382:      sty  check_line+1
27c8 18     383:      clc
27c9      384: c_bit_7:
27c9 c900   385:      cmp  #$00
27cb f006   386:      beq  c_bit_8
27cd a001   387:      ldy  #$01
27cf 8ce427 388:      sty  check_line
27d2 18     389:      clc
27d3      390: c_bit_8:
27d3 18     391:      clc
27d4 ace327 392:      ldy  check_hold+1
27d7 b9e427 393:      lda  check_line,y
27da c901   394:      cmp  #$01
27dc d002   395:      bne  c_bit_9
27de 38     396:      sec
27df 60     397:      rts
27e0 18     398: c_bit_9:      clc
27e1 60     399:      rts
27e2 0000   400: check_hold:  dc.b  0,0
27e4 000000 401: check_line:  dc.b  0,0,0,0,0,0,0
27ec      402:
27ec      403:
27ec      404:
27ec      405: ;-----
27ec      406: install_dli:
27ec a9fc   407:      lda  # low dli_1
27ee 8d0002 408:      sta  $0200
27f1 a927   409:      lda  # high dli_1
27f3 8d0102 410:      sta  $0201
27f6 a9c0   411:      lda  #$c0
27f8 8d0ed4 412:      sta  $d40e
27fb 60     413:      rts
27fc      414:
27fc      415: ;dli_test:  dc.b  0,0,0,0
27fc      416:
27fc      417:
27fc      418: dli_1:
27fc 48     419:      pha
27fd 8a     420:      txa
27fe 48     421:      pha
27ff 98     422:      tya
2800 48     423:      pha
2801      424:

```

```

2801      425: correct_dli_1:
2801 c917 426:      cmp    #23
2803 f025 427:      beq    correct_dli_2
2805 c91c 428:      cmp    #28
2807 f04c 429:      beq    correct_dli_3
2809 c92d 430:      cmp    #45
280b f06e 431:      beq    correct_dli_4
280d a980 432:      lda    #$80
280f 8d0ad4 433:      sta    $d40a
2812 8d18d0 434:      sta    $d018
2815 a925 435:      lda    # low dli_2
2817 8d0002 436:      sta    $0200
281a a928 437:      lda    # high dli_2
281c 8d0102 438:      sta    $0201
281f 68 439:      pla
2820 98 440:      tya
2821 68 441:      pla
2822 aa 442:      tax
2823 68 443:      pla
2824 40 444:      rti
2825      445: ;
2825      446: dli_2:
2825 48 447:      pha
2826 8a 448:      txa
2827 48 449:      pha
2828 98 450:      tya
2829 48 451:      pha
282a      452:
282a      453: correct_dli_2:
282a      454: ;      lda    vcount
282a      455: ;      sta    dli_test+1
282a      456:
282a c913 457:      cmp    #19
282c f0d3 458:      beq    correct_dli_1
282e c91c 459:      cmp    #28
2830 f023 460:      beq    correct_dli_3
2832 c92d 461:      cmp    #45
2834 f045 462:      beq    correct_dli_4
2836 a940 463:      lda    #$40
2838 8d18d0 464:      sta    $d018
283b a950 465:      lda    # low dli_3
283d 8d0002 466:      sta    $0200
2840 a928 467:      lda    # high dli_3
2842 8d0102 468:      sta    $0201
2845 a900 469:      lda    #$00
2847 8ddf33 470:      sta    in_vbi
284a 68 471:      pla
284b a8 472:      tay
284c 68 473:      pla
284d aa 474:      tax
284e 68 475:      pla
284f 40 476:      rti
2850      477: ;

```

```

2850      478: dli_3:
2850 48   479:          pha
2851 8a   480:          txa
2852 48   481:          pha
2853 98   482:          tya
2854 48   483:          pha
2855      484:
2855      485: correct_dli_3:
2855      486:
2855 c913  487:          cmp    #19
2857 f0a8  488:          beq    correct_dli_1
2859 c917  489:          cmp    #23
285b f0cd  490:          beq    correct_dli_2
285d c92d  491:          cmp    #45
285f f01a  492:          beq    correct_dli_4
2861      493:
2861 a980  494:          lda    #$80
2863 8d18d0 495:          sta    $d018
2866 a976  496:          lda    # low dli_4
2868 8d0002 497:          sta    $0200
286b a928  498:          lda    # high dli_4
286d 8d0102 499:          sta    $0201
2870 68   500:          pla
2871 98   501:          tya
2872 68   502:          pla
2873 8a   503:          txa
2874 68   504:          pla
2875 40   505:          rti
2876      506: ;
2876      507: ;
2876      508: dli_4:
2876 48   509:          pha
2877 8a   510:          txa
2878 48   511:          pha
2879 98   512:          tya
287a 48   513:          pha
287b      514:
287b      515: correct_dli_4:
287b      516: ;          lda    vcount
287b      517: ;          sta    dli_test+3
287b      518:
287b c913  519:          cmp    #19
287d f082  520:          beq    correct_dli_1
287f c917  521:          cmp    #23
2881 f0a7  522:          beq    correct_dli_2
2883 c91c  523:          cmp    #28
2885 f0ce  524:          beq    correct_dli_3
2887      525:
2887 a940  526:          lda    #$40
2889 8d18d0 527:          sta    $d018
288c a9fc  528:          lda    # low dli_1
288e 8d0002 529:          sta    $0200
2891 a927  530:          lda    # high dli_1

```

```

2893 8d0102 531:          sta  $0201
2896 68    532:          pla
2897 98    533:          tya
2898 68    534:          pla
2899 aa    535:          tax
289a 68    536:          pla
289b 40    537:          rti
289c      538: ;
289c      539: ;
289c 707070 540: d_list:          dc.b  $70,$70,$70
289f c2    541:          dc.b  $42+$80
28a0 ea34  542:          dc.w  my_sjc          ;
28a2 82    543:          dc.b  $82
28a3 42    544:          dc.b  $42
28a4 3c8c  545:          dc.w  program_name
28a6 80    546:          dc.b  $80
28a7 42    547:          dc.b  $42
28a8 648c  548:          dc.w  rw_screen
28aa 020202 549:          dc.b  2,2,2
28ad 0080  550:          dc.b  0,$80
28af 42    551:          dc.b  $42
28b0      552: fixup:
28b0 0000  553: d_l_mem:          dc.w  0
28b2 020202 554:          dc.b  2,2,2,2,2,2,2
28ba 020202 555:          dc.b  2,2,2,2,2,2,2
28c2 41    556:          dc.b  $41
28c3 9c28  557:          dc.w  d_list
28c5      558: ;
28c5      559: ;
28c5      560: break_int:
28c5      561:          mea  break_it,$0206
28c5 a9d0  ----:          lda  #low break_it
28c7 8d0602 ----:          sta  $0206
28ca a928  ----:          lda  #high break_it
28cc 8d0702 ----:          sta  $0206 + 1
28cf 60    562:          rts
28d0      563: ;
28d0      564: break_it:
28d0 20a38a 565:          jsr  printsi
28d3 9b5468 566:          dc.b  $9b,"This is ODD. a BRK ins
28fa 20bb3b 567:          jsr  more
28fd 4cc329 568:          jmp  main_menu
2900      569: ;
2900      570: start:
2900 20c528 571:          jsr  break_int
2903 20ec27 572:          jsr  install_dli
2906 a902  573:          lda  #$02
2908 8552  574:          sta  $52
290a a000  575:          ldy  #off
290c 201527 576:          jsr  td
290f a900  577:          lda  #$00
2911 8db602 578:          sta  invflg
2914 8dbe02 579:          sta  shflok

```

```

2917 201796 580:      jsr  test_os
291a 208195 581:      jsr  test_smartos
291d 20fa88 582:      jsr  init_screen
2920      583:  ;
2920 208195 584:      jsr  test_smartos
2923 205e90 585:      jsr  do_all_crc
2926 20d292 586:      jsr  pause
2929 205138 587:      jsr  statbox
292c      588:  ;
292c ad1fd0 589:      lda  consol
292f c905  590:      cmp  #$05          ; S
2931 f003  591:      beq  lisa
2933 4cc329 592:      jmp  main_menu
2936      593:  ;
2936 20a38a 594: lisa:      jsr  printsi
2939 9b9b20 595:      dc.b  $9b,$9b," Start $",-1
2944      596:  show_hex_word prg_start
2944 ad8797 ----:      lda  prg_start
2947 ae8897 ----:      ldx  prg_start+1
294a 20ee26 ----:      jsr  h_card
294d 20a38a 597:      jsr  printsi
2950 202020 598:      dc.b  " End $",-1
2959      599:  show_hex_word prg_end
2959 ad8997 ----:      lda  prg_end
295c ae8a97 ----:      ldx  prg_end+1
295f 20ee26 ----:      jsr  h_card
2962 20a38a 600:      jsr  printsi
2965 204c65 601:      dc.b  " Length $",-1
296f      602:  show_hex_word prg_length
296f ad8b97 ----:      lda  prg_length
2972 ae8c97 ----:      ldx  prg_length+1
2975 20ee26 ----:      jsr  h_card
2978 20a38a 603:      jsr  printsi
297b 9b9b20 604:      dc.b  $9b,$9b," Start $",-1
2986      605:  show_hex_word prg_s_start
2986 ad8d97 ----:      lda  prg_s_start
2989 ae8e97 ----:      ldx  prg_s_start+1
298c 20ee26 ----:      jsr  h_card
298f 20a38a 606:      jsr  printsi
2992 202020 607:      dc.b  " End $",-1
299b      608:  show_hex_word prg_s_end
299b ad8f97 ----:      lda  prg_s_end
299e ae9097 ----:      ldx  prg_s_end+1
29a1 20ee26 ----:      jsr  h_card
29a4 20a38a 609:      jsr  printsi
29a7 204c65 610:      dc.b  " Length $",-1
29b1      611:  show_hex_word prg_s_length
29b1 ad9197 ----:      lda  prg_s_length
29b4 ae9297 ----:      ldx  prg_s_length+1
29b7 20ee26 ----:      jsr  h_card
29ba 20a38a 612:      jsr  printsi
29bd 9b9bff 613:      dc.b  $9b,$9b,-1
29c0      614:  ;

```

```

29c0 20d292 615:          jsr  pause
29c3      616: ;
29c3      617: main_menu:
29c3      618: ;
29c3      619:          mea  os_name,i.s_ptr
29c3 a9ba ----:          lda  #low os_name
29c5 85a2 ----:          sta  i.s_ptr
29c7 a987 ----:          lda  #high os_name
29c9 85a3 ----:          sta  i.s_ptr + 1
29cb 620:          mea  parm_1,i.d_ptr
29cb a9ad ----:          lda  #low parm_1
29cd 85a4 ----:          sta  i.d_ptr
29cf a925 ----:          lda  #high parm_1
29d1 85a5 ----:          sta  i.d_ptr + 1
29d3 20d581 621:          jsr  move_str
29d6 622:          mea  parm_1,i.fn_ptr
29d6 a9ad ----:          lda  #low parm_1
29d8 85a6 ----:          sta  i.fn_ptr
29da a925 ----:          lda  #high parm_1
29dc 85a7 ----:          sta  i.fn_ptr + 1
29de 208e38 623:          jsr  file_stat
29e1      624: ;
29e1 205138 625:          jsr  statbox
29e4 20a38a 626:          jsr  printsi
29e7 7dff 627:          dc.b  $7d,-1
29e9 ad152c 628:          lda  which_menu
29ec c900 629:          cmp  #$00
29ee f003 630:          beq  A_
29f0 4cf629 631:          jmp  screen_2
29f3 4cf52a 632: A_:          jmp  screen_1
29f6      633: screen_2:
29f6 20a38a 634:          jsr  printsi
29f9      635: ;
29f9 9b 636:          dc.b  $9b
29fa dbc8dd 637:          dc.b  <+128>,"[H]"
29fd 20496e 638:          dc.b  " Install Hex String into m
2a1d dbc4dd 639:          dc.b  <+128>,"[D]"
2a20 65706f 640:          dc.b  "eposit a Single Hex byte t
2a43 dbd2dd 641:          dc.b  <+128>,"[R]"
2a46 205075 642:          dc.b  " Purge all of the ramdisk"
2a60 dbd0dd 643:          dc.b  <+128>,"[P]"
2a63 726f67 644:          dc.b  "rogram SMART OS",$9B
2a73 dbd7dd 645:          dc.b  <+128>,"[W]"
2a76 20546f 646:          dc.b  " Toggle SMART OS write mod
2a92 dbcfdd 647:          dc.b  <+128>,"[O]"
2a95 20444f 648:          dc.b  " DOS Functions!",$9B
2aa5 dbd1dd 649:          dc.b  <+128>,"[Q]"
2aa8 20496e 650:          dc.b  " Inspect a bank of Ram!",$
2ac0 dbcddd 651:          dc.b  <+128>,"[M]"
2ac3 616b65 652:          dc.b  "ake Cartridge in com file!
2ade dbdadd 653:          dc.b  <+128>,"[Z]"
2ae1 204261 654:          dc.b  " Bank switch OS",$9B
2af1      655: ;

```

```

2af1 ff 656:          dc.b -1
2af2 4ce12b 657:      jmp  work_it
2af5 658: ;
2af5 659: screen_1:
2af5 660: ;
2af5 201d32 661:      jsr  frankie
2af8 20a38a 662:      jsr  printsi
2afb dbb4dd 663:      dc.b <+128>,"[4]"
2afe 20456e 664:      dc.b " Enter comment line(S)!", $
2b16 dbb5dd 665:      dc.b <+128>,"[5]"
2b19 204372 666:      dc.b " Create Smart OS Loadable
2b39 dbb6dd 667:      dc.b <+128>,"[6]"
2b3c 204368 668:      dc.b " Change Control ddress!", $
2b54 dbc9dd 669:      dc.b <+128>,"[I]"
2b57 6e7370 670:      dc.b "nspect HEX address", $9B
2b6a dbccdd 671:      dc.b <+128>,"[L]"
2b6d 6f6164 672:      dc.b "oad E-Prom File!", $9B
2b7e dbd3dd 673:      dc.b <+128>,"[S]"
2b81 617665 674:      dc.b "ave E-Prom File!", $9B
2b92 dbc3dd 675:      dc.b <+128>,"[C]"
2b95 686563 676:      dc.b "hecksum Test on OS Ram!", $
2bad dbd8dd 677:      dc.b <+128>,"[X]"
2bb0 204578 678:      dc.b " Exit program", $9B
2bbe dbc6dd 679:      dc.b <+128>,"[F]"
2bc1 696c65 680:      dc.b "ile name Change", $9B
2bd1 dbc5dd 681:      dc.b <+128>,"[E]"
2bd4 726173 682:      dc.b "rase buffer", $9B
2be0 ff 683:          dc.b -1
2be1 684: ;
2be1 685: work_it:
2be1 20a38a 686:      jsr  printsi
2be4 dbd3d0 687:      dc.b <+128>,"[SPACE]"
2beb 20546f 688:      dc.b " Toggles Menu", $9B
2bf9 9b2043 689:      dc.b $9b," Command "
2c03 ff 690:          dc.b -1
2c04 691:      mea  key_table,i.s_ptr
2c04 a916 ----:      lda  #low key_table
2c06 85a2 ----:      sta  i.s_ptr
2c08 a92c ----:      lda  #high key_table
2c0a 85a3 ----:      sta  i.s_ptr + 1
2c0c 20d430 692:      jsr  my_key
2c0f 20bc85 693:      jsr  execute
2c12 4cc329 694:      jmp  main_menu
2c15 695: ;
2c15 00 696: which_menu:      dc.b 0
2c16 697: ;
2c16 698: key_table:
2c16 7d 699:          dc.b $7d
2c17 74e4 700:          dc.w reset
2c19 45 701:          dc.b "E"
2c1a 1e95 702:          dc.w bank_purge
2c1c 58 703:          dc.b "X"
2c1d 5c2d 704:          dc.w s_exit

```

```

2c1f 1b 705:      dc.b  $1b
2c20 5c2d 706:      dc.w  s_exit
2c22 43 707:      dc.b  "C"
2c23 8c2c 708:      dc.w  s_csum
2c25 53 709:      dc.b  "S"
2c26 2d98 710:      dc.w  steve_os
2c28 4c 711:      dc.b  "L"
2c29 812d 712:      dc.w  s_load
2c2b 35 713:      dc.b  "5"
2c2c 292e 714:      dc.w  s_create
2c2e 31 715:      dc.b  "I"
2c2f 662e 716:      dc.w  s_rom
2c31 32 717:      dc.b  "2"
2c32 372e 718:      dc.w  s_ram
2c34 33 719:      dc.b  "3"
2c35 2b31 720:      dc.w  s_ram1
2c37 46 721:      dc.b  "F"
2c38 b22e 722:      dc.w  s_file
2c3a 4d 723:      dc.b  "M"
2c3b 0080 724:      dc.w  read_cart
2c3d 35 725:      dc.b  "5" ;
2c3e f22e 726:      dc.w  s_address
2c40 49 727:      dc.b  "I"
2c41 552f 728:      dc.w  s_inspect
2c43 2a 729:      dc.b  "*"
2c44 552f 730:      dc.w  s_inspect
2c46 44 731:      dc.b  "D"
2c47 a32f 732:      dc.w  s_put
2c49 34 733:      dc.b  "4" ;
2c4a fb2f 734:      dc.w  s_memo
2c4c 48 735:      dc.b  "H"
2c4d 8330 736:      dc.w  my_put
2c4f 52 737:      dc.b  "R"
2c50 4e39 738:      dc.w  bank_mem
2c52 50 739:      dc.b  "P"
2c53 5a31 740:      dc.w  s_program
2c55 4f 741:      dc.b  "O"
2c56 892c 742:      dc.w  t_dos
2c58 20 743:      dc.b  " "
2c59 762c 744:      dc.w  s_page
2c5b 51 745:      dc.b  "Q"
2c5c 992d 746:      dc.w  s_ram_look
2c5e 56 747:      dc.b  "V"
2c5f 6a2c 748:      dc.w  s_brk
2c61 57 749:      dc.b  "W"
2c62 6b2c 750:      dc.w  s_write_m
2c64 5a 751:      dc.b  "Z"
2c65 7c32 752:      dc.w  program_rom
2c67 00 753:      dc.b  0
2c68 952e 754:      dc.w  s_error
2c6a      755: ;
2c6a      756: s_brk:
2c6a 00 757:      brk

```



```
2c6b      758: ;
2c6b      759: s_write_m:
2c6b ad8a8a 760:      lda  rw_control
2c6e 4920 761:      eor  #os_write
2c70 8d8a8a 762:      sta  rw_control
2c73 4c8f8a 763:      jmp  control_line
2c76      764: ;
2c76      765: s_page:
2c76 ad152c 766:      lda  which_menu
2c79 c900 767:      cmp  #$00
2c7b f006 768:      beq  l_gh
2c7d a900 769:      lda  #$00
2c7f 8d152c 770:      sta  which_menu
2c82 60 771:      rts
2c83      772: l_gh:
2c83 a901 773:      lda  #$01
2c85 8d152c 774:      sta  which_menu
2c88 60 775:      rts
2c89      776: ;
2c89      777:
2c89      778:
2c89      779: t_dos:
2c89      780: ;      pop_word i_sc_hold_mem,savmsec
2c89      781: ;      pop_word i_sc_hold_dlist,sdlstl
2c89 4c6d3a 782:      jmp  s_dos
2c8c      783:
2c8c      784:
2c8c      785:
2c8c      786: s_csum:
2c8c 205138 787:      jsr  statbox
2c8f 20a38a 788:      jsr  printsi
2c92 7d9bff 789:      dc.b $7d,$9b,-1
2c95      790:
2c95      791: ;debug code
2c95      792: ;      lda  dli_test
2c95      793: ;      jsr  pr_byte
2c95      794: ;      lda  #$20
2c95      795: ;      jsr  echo
2c95      796: ;      lda  dli_test+1
2c95      797: ;      jsr  pr_byte
2c95      798: ;      lda  #$20
2c95      799: ;      jsr  echo
2c95      800: ;      lda  dli_test+2
2c95      801: ;      jsr  pr_byte
2c95      802: ;      lda  #$20
2c95      803: ;      jsr  echo
2c95      804: ;      lda  dli_test+3
2c95      805: ;      jsr  pr_byte
2c95      806: ;      lda  #$20
2c95      807: ;      jsr  echo
2c95      808: ;end of debug
2c95      809:
2c95      810:
```

```

2c95 20a38a 811:      jsr  printsi
2c98 9b9b 812:      dc.b  $9b,$9b
2c9a 204368 813:      dc.b  "Checksum menu!",$9B,$9B
2cab ff 814:      dc.b  -1
2cac 201d32 815:      jsr  frankie
2caf 20a38a 816:      jsr  printsi
2cb2      817:      ;
2cb2 dbb4dd 818:      dc.b  <+128>,"[4]"
2cb5 204368 819:      dc.b  "Checksum test RAM OS",$9B
2ccb dbb5dd 820:      dc.b  <+128>,"[5]"
2cce 204368 821:      dc.b  "Checksum test Memory",$9B
2ce4 dbb6dd 822:      dc.b  <+128>,"[6]"
2ce7 204368 823:      dc.b  "Checksum test Rom OS",$9B
2cf4 dbcd4d 824:      dc.b  <+128>,"[M]"
2d00 61696e 825:      dc.b  "ain menu!",$9B
2d0a 9b9b20 826:      dc.b  $9b,$9b,"Selection",-1
2d18      827:      mea  m_crc_table,i.s_ptr
2d18 a929 ----:      lda  #low m_crc_table
2d1a 85a2 ----:      sta  i.s_ptr
2d1c a92d ----:      lda  #high m_crc_table
2d1e 85a3 ----:      sta  i.s_ptr + 1
2d20 20d430 828:      jsr  my_key
2d23 20bc85 829:      jsr  execute
2d26 4c8c2c 830:      jmp  s_csum
2d29      831:      ;
2d29      832:      ;
2d29      833:  m_crc_table:
2d29 34 834:      dc.b  "4"
2d2a 808d 835:      dc.w  crc_rom
2d2c 35 836:      dc.b  "5"
2d2d be8d 837:      dc.w  crc_ram
2d2f 36 838:      dc.b  "6"
2d30 562d 839:      dc.w  crc_rom_os
2d32 31 840:      dc.b  "1"
2d33 662e 841:      dc.w  s_rom
2d35 32 842:      dc.b  "2"
2d36 372e 843:      dc.w  s_ram
2d38 33 844:      dc.b  "3"
2d39 2b31 845:      dc.w  s_ram1
2d3b 4d 846:      dc.b  "M"
2d3c c329 847:      dc.w  main_menu
2d3e 58 848:      dc.b  "X"
2d3f 5c2d 849:      dc.w  s_exit
2d41 1b 850:      dc.b  $1b
2d42 5c2d 851:      dc.w  s_exit
2d44 2a 852:      dc.b  "*"
2d45 552f 853:      dc.w  s_inspect
2d47 49 854:      dc.b  "I"
2d48 552f 855:      dc.w  s_inspect
2d4a 48 856:      dc.b  "H"
2d4b 8330 857:      dc.w  my_put
2d4d 44 858:      dc.b  "D"
2d4e a32f 859:      dc.w  s_put

```

```

2d50 7d 860:          dc.b  $7d
2d51 74e4 861:        dc.w  reset
2d53 00 862:          dc.b  0
2d54 982d 863:        dc.w  s_done
2d56      864: ;
2d56      865: ;
2d56      866: crc_rom_os:
2d56 20662e 867:        jsr   s_rom
2d59 4cbe8d 868:        jmp   crc_ram
2d5c      869: ;
2d5c      870: ;
2d5c      871: s_exit:
2d5c 68 872:          pla
2d5d 68 873:          pla
2d5e a97d 874:          lda   #$7d
2d60 20568d 875:        jsr   echo
2d63 a9ff 876:          lda   #$ff
2d65 8dfc02 877:        sta   ch
2d68 203389 878:        jsr   exit_screen
2d6b ad0007 879:        lda   $0700
2d6e c952 880:        cmp   #'R' ; SpartaDos
2d70 f00c 881:        beq   ss_exit
2d72 ad0107 882:        lda   $0701
2d75 c940 883:        cmp   #$40
2d77 9005 884:        bcc   ss_exit
2d79 a9ff 885:          lda   #$ff
2d7b 8d4402 886:        sta   $0244
2d7e 4c74e4 887: ss_exit: jmp   reset
2d81      888: ;
2d81      889: ;
2d81      890: s_load:
2d81 a220 891:          ldx   #disk_2
2d83 20ee88 892:        jsr   close
2d86      893:        mea   os_name,i.fn_ptr
2d86 a9ba ----:        lda   #low os_name
2d88 85a6 ----:        sta   i.fn_ptr
2d8a a987 ----:        lda   #high os_name
2d8c 85a7 ----:        sta   i.fn_ptr + 1
2d8e a220 894:          ldx   #disk_2
2d90 20d49a 895:        jsr   bload
2d93 a220 896:          ldx   #disk_2
2d95 20ee88 897:        jsr   close
2d98      898: ;
2d98 60 899: s_done:    rts
2d99      900: ;
2d99      901: s_ram_look:
2d99 20a38a 902:        jsr   printsi
2d9c 7d 903:          dc.b  $7d
2d9d 9b9b52 904:        dc.b  $9b,$9b,"Read Which Bank Of
2dbc adac9b 905:        lda   total_banks
2dbf 202627 906:        jsr   pr_byte
2dc2 20a38a 907:        jsr   printsi
2dc5 9b456e 908:        dc.b  $9b,"Enter The Bank You wis

```

```

2dea 203020 909:      dc.b  " 0 = Main Memory", $9B, $9B
2dfc 42616e 910:      dc.b  "Bank ?", -1
2e03 204c85 911:      jsr   get_byte
2e06 cdac9b 912:      cmp   total_banks
2e09 f006 913:      beq   kio
2e0b 108c 914:      bpl   s_ram_look
2e0d c900 915:      cmp   #$00
2e0f f00e 916:      beq   hyu
2e11 a8 917: kio:      tay
2e12 8dfe37 918:      sta   current_ram_bank
2e15 b9ac9b 919:      lda   total_banks, y
2e18 4d2e9c 920:      eor   xor_byte
2e1b 8d01d3 921:      sta   portb
2e1e 60 922:      rts
2e1f 8dfe37 923: hyu:      sta   current_ram_bank
2e22 ad3c9c 924:      lda   ils_hold_d301
2e25 8d01d3 925:      sta   portb
2e28 60 926:      rts
2e29 927: ;
2e29 928: s_create:
2e29 a901 929:      lda   #$01
2e2b 8d8e8a 930:      sta   rw_mode
2e2e 208796 931:      jsr   make_os
2e31 a900 932:      lda   #$00
2e33 8d8e8a 933:      sta   rw_mode
2e36 60 934:      rts
2e37 935: ;
2e37 936: s_ram:
2e37 201e95 937:      jsr   bank_purge
2e3a 207a95 938:      jsr   my_wait
2e3d 08 939:      php
2e3e ad0ed4 940:      lda   nmien
2e41 48 941:      pha
2e42 a900 942:      lda   #$00
2e44 8d0ed4 943:      sta   nmien
2e47 78 944:      sei
2e48 20fe97 945:      jsr   load_2
2e4b a980 946:      lda   #os_ram
2e4d 208f8a 947:      jsr   control_line
2e50 205489 948:      jsr   rom_to_ram
2e53 ad9397 949:      lda   smos_data
2e56 208f8a 950:      jsr   control_line
2e59 68 951:      pla
2e5a 8d0ed4 952:      sta   nmien
2e5d 28 953:      plp
2e5e 201298 954:      jsr   restore_2
2e61 58 955:      cli
2e62 207a95 956:      jsr   my_wait
2e65 60 957:      rts
2e66 958: ;
2e66 959: s_rom:
2e66 201e95 960:      jsr   bank_purge
2e69 207a95 961:      jsr   my_wait

```

```

2e6c 08 962:      php
2e6d ad0ed4 963:    lda  nmien
2e70 48 964:      pha
2e71 a900 965:    lda  #$00
2e73 8d0ed4 966:    sta  nmien
2e76 78 967:      sei
2e77 20fe97 968:    jsr  load_2
2e7a a940 969:    lda  #os_rom
2e7c 208f8a 970:    jsr  control_line
2e7f 205489 971:    jsr  rom_to_ram
2e82 ad9397 972:    lda  smos_data
2e85 208f8a 973:    jsr  control_line
2e88 68 974:      pla
2e89 8d0ed4 975:    sta  nmien
2e8c 28 976:      plp
2e8d 201298 977:    jsr  restore_2
2e90 58 978:      cli
2e91 207a95 979:    jsr  my_wait
2e94 60 980:      rts
2e95      981: ;
2e95      982: ;
2e95      983: s_error:
2e95 20a38a 984:    jsr  printsi
2e98 9b9b20 985:    dc.b  $9b,$9b," That Key not used
2eaf 4cd292 986:    jmp  pause
2eb2      987: ;
2eb2      988: s_file:
2eb2 20a38a 989:    jsr  printsi
2eb5 9b9b20 990:    dc.b  $9b,$9b," Enter Filename >"
2ec9 209883 991:    jsr  get_str
2ecc ad25 992:    dc.w  parm_1
2ece 260001 993:    dc.b  38,0,1
2ed1      994:    mea  parm_1,i.s_ptr
2ed1 a9ad ----:    lda  #low parm_1
2ed3 85a2 ----:    sta  i.s_ptr
2ed5 a925 ----:    lda  #high parm_1
2ed7 85a3 ----:    sta  i.s_ptr + 1
2ed9 20aa86 995:    jsr  upstring
2edc      996:    mea  parm_1,i.s_ptr
2edc a9ad ----:    lda  #low parm_1
2ede 85a2 ----:    sta  i.s_ptr
2ee0 a925 ----:    lda  #high parm_1
2ee2 85a3 ----:    sta  i.s_ptr + 1
2ee4      997:    mea  os_name,i.d_ptr
2ee4 a9ba ----:    lda  #low os_name
2ee6 85a4 ----:    sta  i.d_ptr
2ee8 a987 ----:    lda  #high os_name
2eea 85a5 ----:    sta  i.d_ptr + 1
2eec 206682 998:    jsr  toascii
2eef 4c548d 999:    jmp  cr_lf
2ef2      1000: ;
2ef2      1001: s_address:
2ef2 a930 1002:    lda  #'0'

```

```

2ef4 8dfc85 1003:      sta    sss
2ef7 8dfd85 1004:      sta    sss+1
2efa 8dfe85 1005:      sta    sss+2
2efd 8dff85 1006:      sta    sss+3
2f00 20a38a 1007:      jsr    printsi
2f03 9b9b20 1008:      dc.b   $9b,$9b," Enter Hex $",-1
2f12 209883 1009:      jsr    get_str
2f15 dd37   1010:      dc.w   a.cnt_point
2f17 040001 1011:      dc.b   4,0,1
2f1a      1012:      mea    a.cnt_point,i.s_ptr
2f1a a9dd   ----:      lda    #low a.cnt_point
2f1c 85a2   ----:      sta    i.s_ptr
2f1e a937   ----:      lda    #high a.cnt_point
2f20 85a3   ----:      sta    i.s_ptr + 1
2f22 20aa86 1013:      jsr    upstring
2f25      1014:      mea    sss,i.d_ptr
2f25 a9fc   ----:      lda    #low sss
2f27 85a4   ----:      sta    i.d_ptr
2f29 a985   ----:      lda    #high sss
2f2b 85a5   ----:      sta    i.d_ptr + 1
2f2d 206682 1015:      jsr    toascii
2f30 200586 1016:      jsr    steve_pack
2f33      1017:      pop_word    sss_data,store2
2f33 ad0386 ----:      lda    sss_data
2f36 8dd433 ----:      sta    store2
2f39 ad0486 ----:      lda    sss_data+1
2f3c 8dd533 ----:      sta    store2+1
2f3f      1018:      pop_word    sss_data,rw_line
2f3f ad0386 ----:      lda    sss_data
2f42 8d8c8a ----:      sta    rw_line
2f45 ad0486 ----:      lda    sss_data+1
2f48 8d8d8a ----:      sta    rw_line+1
2f4b 201796 1019:      jsr    test_os
2f4e 208195 1020:      jsr    test_smartos
2f51 20fa88 1021:      jsr    init_screen
2f54 60     1022:      rts
2f55      1023:      ;
2f55      1024: s_inspect:
2f55 a930 1025:      lda    #'0'
2f57 8dfc85 1026:      sta    sss
2f5a 8dfd85 1027:      sta    sss+1
2f5d 8dfe85 1028:      sta    sss+2
2f60 8dff85 1029:      sta    sss+3
2f63 20a38a 1030:      jsr    printsi
2f66 9b9b20 1031:      dc.b   $9b,$9b," Enter Hex $",-1
2f75 209883 1032:      jsr    get_str
2f78 ad25   1033:      dc.w   parm_1
2f7a 040001 1034:      dc.b   4,0,1
2f7d      1035:      mea    parm_1,i.s_ptr
2f7d a9ad   ----:      lda    #low parm_1
2f7f 85a2   ----:      sta    i.s_ptr
2f81 a925   ----:      lda    #high parm_1
2f83 85a3   ----:      sta    i.s_ptr + 1

```

```

2f85 20aa86 1036:      jsr  upstring
2f88      1037:      mea  sss,i.d_ptr
2f88 a9fc ----:      lda  #low sss
2f8a 85a4 ----:      sta  i.d_ptr
2f8c a985 ----:      lda  #high sss
2f8e 85a5 ----:      sta  i.d_ptr + 1
2f90 206682 1038:      jsr  toascii
2f93 200586 1039:      jsr  steve_pack
2f96      1040:      pop_word  sss_data,store2
2f96 ad0386 ----:      lda  sss_data
2f99 8dd433 ----:      sta  store2
2f9c ad0486 ----:      lda  sss_data+1
2f9f 8dd533 ----:      sta  store2+1
2fa2 60      1041:      rts
2fa3      1042: ;
2fa3      1043: s_put:
2fa3 a930 1044:      lda  #'0'
2fa5 8dfd85 1045:      sta  sss+1
2fa8 8dfe85 1046:      sta  sss+2
2fab 8dff85 1047:      sta  sss+3
2fae 20a38a 1048:      jsr  printsi
2fb1 9b9b20 1049:      dc.b  $9b,$9b," Enter Hex Byte $"
2fc5 209883 1050:      jsr  get_str
2fc8 ad25 1051:      dc.w  parm_1
2fca 020001 1052:      dc.b  2,0,1
2fcd      1053:      mea  parm_1,i.s_ptr
2fcd a9ad ----:      lda  #low parm_1
2fcf 85a2 ----:      sta  i.s_ptr
2fd1 a925 ----:      lda  #high parm_1
2fd3 85a3 ----:      sta  i.s_ptr + 1
2fd5 20aa86 1054:      jsr  upstring
2fd8      1055:      mea  sss,i.d_ptr
2fd8 a9fc ----:      lda  #low sss
2fda 85a4 ----:      sta  i.d_ptr
2fdc a985 ----:      lda  #high sss
2fde 85a5 ----:      sta  i.d_ptr + 1
2fe0 206682 1056:      jsr  toascii
2fe3 200586 1057:      jsr  steve_pack
2fe6 add433 1058:      lda  store2
2fe9 8df82f 1059:      sta  dun+1
2fec add533 1060:      lda  store2+1
2fef 8df92f 1061:      sta  dun+2
2ff2 a000 1062:      ldy  #$00
2ff4 ad0486 1063:      lda  sss_data+1
2ff7 99ffff 1064: dun:      sta  $ffff,y
2ffa 60      1065:      rts
2ffb      1066: ;
2ffb      1067: s_memo:
2ffb 20a38a 1068:      jsr  printsi
2ffe 7d9b9b 1069:      dc.b  $7d,$9b,$9b,$9b
3002 9b9b20 1070:      dc.b  $9b,$9b," Enter your two li
3021 202020 1071:      dc.b  " 111111111122222222
3046 313233 1072:      dc.b  "12345678901234567890123456

```

```

306c 209883 1073:      jsr  get_str
306f ff37  1074:      dc.w  memo1
3071 240001 1075:      dc.b  36,0,1
3074 20548d 1076:      jsr  cr_lf
3077 209883 1077:      jsr  get_str
307a 2838   1078:      dc.w  memo2
307c 240001 1079:      dc.b  36,0,1
307f 20548d 1080:      jsr  cr_lf
3082 60     1081:      rts
3083       1082: ;
3083       1083: my_put:
3083 20a38a 1084:      jsr  printsi
3086 9b9b20 1085:      dc.b  $9b,$9b," Enter Hex Byte st
30a3 209883 1086:      jsr  get_str
30a6 ad25   1087:      dc.w  parm_1
30a8 100001 1088:      dc.b  16,0,1
30ab       1089:      mea  parm_1,i.s_ptr
30ab a9ad   ----:      lda  #low parm_1
30ad 85a2   ----:      sta  i.s_ptr
30af a925   ----:      lda  #high parm_1
30b1 85a3   ----:      sta  i.s_ptr + 1
30b3 20aa86 1090:      jsr  upstring
30b6 20f486 1091:      jsr  steve_byte
30b9 add433 1092:      lda  store2
30bc 8dc30  1093:      sta  pun+1
30bf add533 1094:      lda  store2+1
30c2 8dcc30 1095:      sta  pun+2
30c5 a000   1096:      ldy  #$00
30c7       1097: xyz:
30c7 b9e886 1098:      lda  zzz_data,y
30ca 99ffff 1099: pun:      sta  $ffff,y
30cd c8     1100:      iny
30ce ccf386 1101:      cpy  zzz_x
30d1 d0f4   1102:      bne  xyz
30d3 60     1103:      rts
30d4       1104: ;
30d4       1105: my_key:
30d4 a511   1106:      lda  brkkey
30d6 c900   1107:      cmp  #$00
30d8 d00c   1108:      bne  _.a
30da a980   1109:      lda  #$80
30dc 8511   1110:      sta  brkkey
30de 203a95 1111:      jsr  bbmenu
30e1 a9ff   1112:      lda  #$ff
30e3 8dfc02 1113:      sta  ch
30e6       1114: _.a:
30e6 ad1fd0 1115:      lda  consol
30e9 c905   1116:      cmp  #$05
30eb f01e   1117:      beq  ..select
30ed c906   1118:      cmp  #$06
30ef f01e   1119:      beq  ..start
30f1 c903   1120:      cmp  #$03
30f3 f01e   1121:      beq  ..option

```



```

30f5 c900 1122:      cmp    #$00
30f7 f01e 1123:      beq    ..restart
30f9 addc02 1124:      lda    helpfg
30fc c900 1125:      cmp    #$00
30fe d01f 1126:      bne    ..help
3100 adfc02 1127:      lda    ch
3103 c9ff 1128:      cmp    #$ff
3105 f0cd 1129:      beq    my_key
3107 4c598d 1130:     jmp    get_key
310a 60 1131:      rts
310b 1132: ;
310b 202531 1133: ..select:      jsr    ..pugkey
310e 60 1134:      rts
310f 202531 1135: ..start:      jsr    ..pugkey
3112 60 1136:      rts
3113 202531 1137: ..option:     jsr    ..pugkey
3116 60 1138:      rts
3117 a9ff 1139: ..restart:     lda    #$ff
3119 8d4402 1140:      sta    $0244
311c 4c74e4 1141:      jmp    $e474
311f a900 1142: ..help:      lda    #$00
3121 8ddc02 1143:      sta    helpfg
3124 60 1144:      rts
3125 1145: ..pugkey:
3125 a9ff 1146:      lda    #$ff
3127 8dfc02 1147:      sta    ch
312a 60 1148:      rts
312b 1149: ;
312b 1150: s_ram1:
312b 201e95 1151:      jsr    bank_purge
312e 207a95 1152:      jsr    my_wait
3131 08 1153:      php
3132 ad0ed4 1154:      lda    nmien
3135 48 1155:      pha
3136 a900 1156:      lda    #$00
3138 8d0ed4 1157:      sta    nmien
313b 78 1158:      sei
313c 20fe97 1159:      jsr    load_2
313f a901 1160:      lda    #os_ram1
3141 208f8a 1161:      jsr    control_line
3144 205489 1162:      jsr    rom_to_ram
3147 ad9397 1163:      lda    smos_data
314a 208f8a 1164:      jsr    control_line
314d 68 1165:      pla
314e 8d0ed4 1166:      sta    nmien
3151 28 1167:      plp
3152 201298 1168:      jsr    restore_2
3155 58 1169:      cli
3156 207a95 1170:      jsr    my_wait
3159 60 1171:      rts
315a 1172: ;
315a 1173: s_program:
315a 20a38a 1174:      jsr    printsi

```

```

315d 7d9b9b 1175:      dc.b  $7d,$9b,$9b,$9b
3161 20534d 1176:      dc.b  " SMART OS program menu!",$
317a 20 1177:          dc.b  " "
317b dbb1dd 1178:      dc.b  <+128>,"[1]"
317e 205072 1179:      dc.b  " Program boot ram bank",$9
3195 20 1180:          dc.b  " "
3196 dbb2dd 1181:      dc.b  <+128>,"[2]"
3199 205072 1182:      dc.b  " Program Aux Ram Bank",$9B
31af 20 1183:          dc.b  " "
31b0 dbcd dd 1184:      dc.b  <+128>,"[M]"
31b3 61696e 1185:      dc.b  "ain menu!",$9B
31bd 9b9b20 1186:      dc.b  $9b,$9b," Selection",-1
31cb 1187:          mea  m_pgm_table,i.s_ptr
31cb a9dc ----:      lda  #low m_pgm_table
31cd 85a2 ----:      sta  i.s_ptr
31cf a931 ----:      lda  #high m_pgm_table
31d1 85a3 ----:      sta  i.s_ptr + 1
31d3 20d430 1188:      jsr  my_key
31d6 20bc85 1189:      jsr  execute
31d9 4c5a31 1190:      jmp  s_program
31dc 1191: ;
31dc 1192: m_pgm_table:
31dc 31 1193:          dc.b  "1"
31dd fd31 1194:          dc.w  pgm_ram
31df 32 1195:          dc.b  "2"
31e0 0532 1196:          dc.w  pgm_ram1
31e2 4d 1197:          dc.b  "M"
31e3 c329 1198:          dc.w  main_menu
31e5 58 1199:          dc.b  "X"
31e6 5c2d 1200:          dc.w  s_exit
31e8 1b 1201:          dc.b  $1b
31e9 5c2d 1202:          dc.w  s_exit
31eb 2a 1203:          dc.b  "*"
31ec 552f 1204:          dc.w  s_inspect
31ee 49 1205:          dc.b  "I"
31ef 552f 1206:          dc.w  s_inspect
31f1 48 1207:          dc.b  "H"
31f2 8330 1208:          dc.w  my_put
31f4 44 1209:          dc.b  "D"
31f5 a32f 1210:          dc.w  s_put
31f7 7d 1211:          dc.b  $7d
31f8 74e4 1212:          dc.w  reset
31fa 00 1213:          dc.b  0
31fb 982d 1214:          dc.w  s_done
31fd 1215: ;
31fd 1216: pgm_ram:
31fd a980 1217:          lda  #os_ram
31ff 8d8a8a 1218:          sta  rw_control
3202 4cf089 1219:          jmp  ram_to_rom
3205 1220: ;
3205 1221: pgm_ram1:
3205 ad8a8a 1222:          lda  rw_control
3208 8d1c32 1223:          sta  pgm_ram1_d

```

```

320b a901 1224:      lda  #os_ram1
320d 8d8a8a 1225:      sta  rw_control
3210 20f089 1226:      jsr  ram_to_rom
3213 ad1c32 1227:      lda  pgm_ram1_d
3216 8d8a8a 1228:      sta  rw_control
3219 4c8f8a 1229:      jmp  control_line
321c      1230: ;
321c 00      1231: pgm_ram1_d:      dc.b  0
321d      1232: ;
321d 20a38a 1233: frankie:      jsr  printsi
3220 9b      1234:      dc.b  $9b
3221 dbb1dd 1235:      dc.b  <+128>,"[1]"
3224 205265 1236:      dc.b  " Read OS ROM into Memory!"
322e dbb2dd 1237:      dc.b  <+128>,"[2]"
3241 205265 1238:      dc.b  " Read OS Ram into Memory!"
325b dbb3dd 1239:      dc.b  <+128>,"[3]"
325e 205265 1240:      dc.b  " Read OS Ram 1 into Memory"
327a ff      1241:      dc.b  -1
327b 60      1242:      rts
327c      1243: ;
327c      1244: ;
327c      1245: program_rom:
327c 205138 1246:      jsr  statbox
327f 20a38a 1247:      jsr  printsi
3282 7d9b      1248:      dc.b  $7d,$9b
3284 204f53 1249:      dc.b  " OS SWAP Menu",$9B,$9B,$9B
3294 dbb1dd 1250:      dc.b  <+128>,"[1]"
3297 20476f 1251:      dc.b  " Goto Rom OS",$9B
32a4 dbb2dd 1252:      dc.b  <+128>,"[2]"
32a7 20476f 1253:      dc.b  " Goto Boot Ram OS",$9B
32b9 dbb3dd 1254:      dc.b  <+128>,"[3]"
32bc 20476f 1255:      dc.b  " Goto aux Ram OS",$9B
32cd dbcd dd 1256:      dc.b  <+128>,"[M]"
32d0 204d61 1257:      dc.b  " Main Menu",$9B
32db 9b9b20 1258:      dc.b  $9b,$9b," Enter Your Select
32f5      1259: ;
32f5      1260: ;
32f5      1261:      mea  m_rom_table,i.s_ptr
32f5 a906 ----:      lda  #low m_rom_table
32f7 85a2 ----:      sta  i.s_ptr
32f9 a933 ----:      lda  #high m_rom_table
32fb 85a3 ----:      sta  i.s_ptr + 1
32fd 20d430 1262:      jsr  my_key
3300 20bc85 1263:      jsr  execute
3303 4c7c32 1264:      jmp  program_rom
3306      1265: ;
3306      1266: m_rom_table:
3306 31      1267:      dc.b  "1"
3307 2a33      1268:      dc.w  do_rom
3309 32      1269:      dc.b  "2"
330a 7433      1270:      dc.w  do_ram
330c 33      1271:      dc.b  "3"
330d 4f33      1272:      dc.w  in_ram

```

```

330f 4d 1273:      dc.b  "M"
3310 c329 1274:      dc.w  main_menu
3312 58 1275:      dc.b  "X"
3313 5c2d 1276:      dc.w  s_exit
3315 1b 1277:      dc.b  $1b
3316 5c2d 1278:      dc.w  s_exit
3318 2a 1279:      dc.b  "*"
3319 552f 1280:      dc.w  s_inspect
331b 49 1281:      dc.b  "I"
331c 552f 1282:      dc.w  s_inspect
331e 48 1283:      dc.b  "H"
331f 8330 1284:      dc.w  my_put
3321 44 1285:      dc.b  "D"
3322 a32f 1286:      dc.w  s_put
3324 7d 1287:      dc.b  $7d
3325 74e4 1288:      dc.w  reset
3327 00 1289:      dc.b  0
3328 982d 1290:      dc.w  s_done
332a      1291: ;
332a      1292: ;
332a      1293: do_rom:
332a a000 1294:      ldy  #0
332c b99697 1295: in_rom:      lda  current_os,y
332f d99a97 1296:      cmp  rom_os,y
3332 d065 1297:      bne  cannot
3334 c8 1298:      iny
3335 c004 1299:      cpy  #4
3337 d0f3 1300:      bne  in_rom
3339 a940 1301:      lda  #os_rom
333b 8d8a8a 1302:      sta  rw_control
333e 8d9397 1303:      sta  smos_data ; ADDED
3341 208f8a 1304:      jsr  control_line
3344      1305:      mea  smosrom,sm_cfg
3344 a9a4 ----:      lda  #low smosrom
3346 8dec88 ----:      sta  sm_cfg
3349 a988 ----:      lda  #high smosrom
334b 8ded88 ----:      sta  sm_cfg + 1
334e 60 1306: not_rom:      rts
334f      1307: ;
334f      1308: in_ram:
334f a000 1309:      ldy  #$00
3351 b99697 1310: in_ram1:      lda  current_os,y
3354 d9a297 1311:      cmp  ram1_os,y
3357 d040 1312:      bne  cannot
3359 c8 1313:      iny
335a c004 1314:      cpy  #4
335c d0f3 1315:      bne  in_ram1
335e a901 1316:      lda  #os_ram1
3360 8d8a8a 1317:      sta  rw_control
3363 8d9397 1318:      sta  smos_data
3366 208f8a 1319:      jsr  control_line
3369      1320:      mea  smosram1,sm_cfg
3369 a9c8 ----:      lda  #low smosram1

```

```

336b 8dec88 ----:      sta  sm_cfg
336e a988 ----:      lda  #high smosram1
3370 8ded88 ----:      sta  sm_cfg + 1
3373 60   1321: not_ram:      rts
3374      1322: ;
3374      1323: do_ram:
3374 a000 1324:      ldy  #$00
3376 b99697 1325: do_ram1:    lda  current_os,y
3379 d99e97 1326:      cmp  ram_os,y
337c d01b 1327:      bne  cannot
337e c8   1328:      iny
337f c004 1329:      cpy  #4
3381 d0f3 1330:      bne  do_ram1
3383 a980 1331:      lda  #os_ram
3385 8d8a8a 1332:      sta  rw_control
3388 8d9397 1333:      sta  smos_data
338b 208f8a 1334:      jsr  control_line
338e      1335:      mea  smosram,sm_cfg
338e a9b6 ----:      lda  #low smosram
3390 8dec88 ----:      sta  sm_cfg
3393 a988 ----:      lda  #high smosram
3395 8ded88 ----:      sta  sm_cfg + 1
3398 60   1336: do_not_ram: rts
3399      1337: cannot:
3399 20a38a 1338:      jsr  printsi
339c 9b2053 1339:      dc.b  $9b," Sorry CRC's Did not M
33b8 205265 1340:      dc.b  " Request aborted!",$9B,-1
33cb 4cd292 1341:      jmp  pause
33ce      1342: ;
33ce 9b9b 1343:      dc.b  $9b,$9b
33d0      1344:
33d0      1345:
33d0      1346: ;-----
33d0 0404 1347: store:      dc.b  $04,$04
33d2 0000 1348: store1:     dc.b  $00,$00
33d4 0040 1349: store2:     dc.w  memory
33d6 1335 1350: rik:      dc.w  ric
33d8 1735 1351: line:     dc.w  line1
33da 7f35 1352: v_bootup:  dc.w  bootup
33dc 0029 1353: v_trap:   dc.w  start
33de 00   1354: recurse:  dc.b  0
33df      1355: ;
33df 00   1356: in_vbi:   dc.b  0
33e0      1357: trap:
33e0 adda33 1358:      lda  v_bootup
33e3 8de702 1359:      sta  $02e7
33e6 addb33 1360:      lda  v_bootup+1
33e9 8de802 1361:      sta  $02e7+1
33ec ad2402 1362:      lda  $0224
33ef 8de534 1363:      sta  exit+1
33f2 ad2502 1364:      lda  $0225
33f5 8de634 1365:      sta  exit+2
33f8 a00a 1366:      ldy  #top_init&$ff

```

```

33fa a234 1367:      ldx  #top_init/256
33fc a907 1368:      lda  #7
33fe 205ce4 1369:     jsr  setvbw
3401 a900 1370:      lda  #$00
3403 8dbe02 1371:     sta  shflok
3406 8db602 1372:     sta  invflg
3409 60 1373:      rts
340a 1374: ;
340a 1375: ;
340a 1376: ;
340a 1377: top_init:
340a 1378:      save
340a 48 ----:      pha
340b 98 ----:      tya
340c 48 ----:      pha
340d 8a ----:      txa
340e 48 ----:      pha
340f a511 1379:      lda  brkkey
3411 c900 1380:      cmp  #$00
3413 d007 1381:      bne  at_shi
3415 a980 1382:      lda  #$80
3417 8511 1383:      sta  brkkey
3419 203a95 1384:     jsr  bbmenu
341c 1385: at_shi:
341c ad1fd0 1386:     lda  consol
341f 1387: ;
341f 1388: ;
341f c905 1389:     cmp  #$05      ;Se
3421 d003 1390:     bne  smith
3423 4cda34 1391:     jmp  rob
3426 1392: smith:
3426 1393: ;      LDA  $14
3426 1394: ;      CMP  RECURSE
3426 1395: ;      BEQ  AT_IT
3426 1396: at_tt:
3426 a901 1397:      lda  #$01
3428 8ddf33 1398:     sta  in_vbi
342b 1399: ;
342b 1400: ;
342b 1401: ;      LDA  $02FC
342b 1402: ;      CMP  #$8F      ; r
342b 1403: ;      BEQ  AT_SS
342b 1404: ;      CMP  #$0E      ; r
342b 1405: ;      BEQ  AT_SS
342b ad1fd0 1406:     lda  consol
342e c903 1407:     cmp  #$03      ; o
3430 d00d 1408:     bne  at_itt
3432 1409: at_ss: ;      lda  #$FF
3432 1410: ;      STA  $02FC
3432 eed433 1411:     inc  store2
3435 add433 1412:     lda  store2
3438 c900 1413:     cmp  #$00
343a d003 1414:     bne  at_itt

```

```

343c eed533 1415:          inc   store2+1
343f      1416: at_itt:
343f      1417: ;          CMP   #$8E          ; r
343f      1418: ;          BEQ   AT_ST
343f      1419: ;          CMP   #$06          ; r
343f      1420: ;          BEQ   AT_ST
343f ad1fd0 1421:          lda   consol          ; S
3442 c906  1422:          cmp   #$06
3444 d003  1423:          bne   at_it
3446      1424: at_st:
3446      1425: ;          lda   #$FF
3446      1426: ;          STA   $02FC
3446 eed533 1427:          inc   store2+1
3449      1428: at_it:
3449 add633 1429:          lda   rik
344c 8d7935 1430:          sta   fixt+1
344f add733 1431:          lda   rik+1
3452 8d7a35 1432:          sta   fixt+2
3455 a900  1433:          lda   #$00
3457 8dd033 1434:          sta   store
345a add533 1435:          lda   store2+1
345d 8d8434 1436:          sta   ken+2
3460 203a35 1437:          jsr   error
3463 add433 1438:          lda   store2
3466 8d8334 1439:          sta   ken+1
3469 203a35 1440:          jsr   error
346c      1441: ;
346c a000  1442:          ldy   #$00
346e add833 1443:          lda   line
3471 8d7935 1444:          sta   fixt+1
3474 add933 1445:          lda   line+1
3477 8d7a35 1446:          sta   fixt+2
347a a900  1447:          lda   #$00
347c 8dd033 1448:          sta   store
347f      1449: loop_1:
347f eed033 1450:          inc   store
3482 b9ffff 1451: ken:          lda   $ffff,y
3485 203a35 1452:          jsr   error
3488 c8     1453:          iny
3489 c00b   1454:          cpy   #11
348b d0f2   1455:          bne   loop_1
348d      1456: ;
348d      1457: popper:
348d      1458:          pop_word   i.pr_ptr,mhd
348d a5a8   ----:          lda   i.pr_ptr
348f 8de834 ----:          sta   mhd
3492 a5a9   ----:          lda   i.pr_ptr+1
3494 8de934 ----:          sta   mhd+1
3497 ad01d3 1459:          lda   portb
349a 200f39 1460:          jsr   s_hex
349d      1461:          mea   s_hex_st,i.pr_ptr
349d a923   ----:          lda   #low s_hex_st
349f 85a8   ----:          sta   i.pr_ptr

```

```

34a1 a939 ----:      lda  #high s_hex_st
34a3 85a9 ----:      sta  i.pr_ptr + 1
34a5 a24d 1462:      ldx  #77
34a7 20c738 1463:      jsr  writes
34aa      1464: ;
34aa ad8a8a 1465:      lda  rw_control
34ad 200f39 1466:      jsr  s_hex
34b0      1467:      mea  s_hex_st,i.pr_ptr
34b0 a923 ----:      lda  #low s_hex_st
34b2 85a8 ----:      sta  i.pr_ptr
34b4 a939 ----:      lda  #high s_hex_st
34b6 85a9 ----:      sta  i.pr_ptr + 1
34b8 a230 1468:      ldx  #48
34ba 20c738 1469:      jsr  writes
34bd      1470: ;
34bd adfe37 1471:      lda  current_ram_bank
34c0 201b37 1472:      jsr  spr_byte
34c3      1473:      mea  spr_data,i.pr_ptr
34c3 a914 ----:      lda  #low spr_data
34c5 85a8 ----:      sta  i.pr_ptr
34c7 a937 ----:      lda  #high spr_data
34c9 85a9 ----:      sta  i.pr_ptr + 1
34cb a237 1474:      ldx  #55
34cd 20c738 1475:      jsr  writes
34d0      1476: ;
34d0      1477:      pop_word  mhd,i.pr_ptr
34d0 ade834 ----:      lda  mhd
34d3 85a8 ----:      sta  i.pr_ptr
34d5 ade934 ----:      lda  mhd+1
34d8 85a9 ----:      sta  i.pr_ptr+1
34da      1478: ;
34da      1479: rob:
34da a900 1480:      lda  #$00
34dc 8ddf33 1481:      sta  in_vbi
34df      1482:      restore
34df 68 ----:      pla
34e0 aa ----:      tax
34e1 68 ----:      pla
34e2 a8 ----:      tay
34e3 68 ----:      pla
34e4 4cffff 1483: exit:      jmp  $ffff
34e7 60 1484:      rts
34e8      1485: ;
34e8 0000 1486: mhd:      dc.w  0
34ea      1487: my_sjc:
34ea      1488: ;      dc.b  <internal>,<+128>,"OS_MENU
34ea b2e5e1 1489:      dc.b  <internal>,<+128>,"RealDos
3512 84 1490:      dc.b  <internal>,<+128>,"$
3513 a6a6a6 1491: ric:      dc.b  <internal>,<+128>,"FFFF"
3517 808080 1492: line1:      dc.b  <internal>,<+128>,"
353a      1493: ;
353a      1494: ;
353a 48 1495: error:      pha

```



```

353b 4a 1496:      lsr
353c 4a 1497:      lsr
353d 4a 1498:      lsr
353e 4a 1499:      lsr
353f 204535 1500:   jsr   dd_8913
3542 68 1501:      pla
3543 290f 1502:      and   #$0f
3545 c90a 1503: dd_8913:  cmp   #$0a
3547 b004 1504:      bcs   dd_891b
3549 0930 1505:      ora   #$30
354b d002 1506:      bne   dd_891d
354d 6936 1507: dd_891b:  adc   #$36
354f 205335 1508: dd_891d:  jsr   fxch
3552 60 1509:      rts
3553      1510: fxch:
3553 c920 1511:      cmp   #32
3555 9013 1512:      bcc   a_dd64
3557 c960 1513:      cmp   #96
3559 9015 1514:      bcc   s_ub32
355b c980 1515:      cmp   #128
355d 9014 1516:      bcc   fixt1
355f c9a0 1517:      cmp   #160
3561 9007 1518:      bcc   a_dd64
3563 c9f4 1519:      cmp   #244
3565 9009 1520:      bcc   s_ub32
3567 4c7335 1521:   jmp   fixt1
356a      1522: a_dd64:
356a 18 1523:      clc
356b 6940 1524:      adc   #64
356d 4c7335 1525:   jmp   fixt1
3570      1526: s_ub32:
3570 38 1527:      sec
3571 e920 1528:      sbc   #32
3573      1529: fixt1:
3573 aed033 1530:      ldx   store
3576 4980 1531:      eor   #$80
3578 9dffff 1532: fixt:  sta   $ffff,x
357b eed033 1533:      inc   store
357e 60 1534:      rts
357f      1535: my_end_mon:
357f      1536: ;
357f      1537: ;-----
357f      1538: ; END OF MEMORY RESIDENT PORTION
357f      1539: ;-----
357f      1540: ;
357f      1541: ;
357f      1542: bootup:
357f      1543: ;
357f adda33 1544:      lda   v_bootup
3582 8de702 1545:      sta   $02e7
3585 addb33 1546:      lda   v_bootup+1
3588 8de802 1547:      sta   $02e7+1
358b ad2402 1548: top:   lda   $0224

```

```

358e 8de534 1549:          sta  exit+1
3591 ad2502 1550:          lda  $0225
3594 8de634 1551:          sta  exit+2
3597      1552: ; LDA TOP_HAT
3597      1553: ; STA $0224
3597      1554: ; LDA TOP_HAT+1
3597      1555: ; STA $0225
3597 a00a 1556:          ldy  #top_init&$ff
3599 a234 1557:          ldx  #top_init/256
359b a907 1558:          lda  #7
359d 205ce4 1559:          jsr  setvbw
35a0      1560: shit:
35a0 a50c 1561:          lda  $0c
35a2 8de133 1562:          sta  trap+1
35a5 a50d 1563:          lda  $0c+1
35a7 8de233 1564:          sta  trap+2
35aa addc33 1565:          lda  v_trap
35ad 850c 1566:          sta  $0c
35af addd33 1567:          lda  v_trap+1
35b2 850d 1568:          sta  $0c+1
35b4 a94c 1569:          lda  #$4c
35b6 8da035 1570:          sta  shit
35b9      1571:          mea  my_mon,shit+1
35b9 a9c3 ----:          lda  #low my_mon
35bb 8da135 ----:          sta  shit+1
35be a935 ----:          lda  #high my_mon
35c0 8da235 ----:          sta  shit+1 + 1
35c3      1572: ;
35c3      1573: my_mon:
35c3 20a38a 1574:          jsr  printsi
35c6 7d9b9b 1575:          dc.b  $7d,$9b,$9b,$9b
35ca 202020 1576:          dc.b  " "
35cf 919292 1577:          dc.b  <+128>,"
35e9 9b 1578:          dc.b  $9B
35ea 202020 1579:          dc.b  " "
35ef fca0a0 1580:          dc.b  <+128>,"| Super Memory Mon
3609 9b 1581:          dc.b  $9B
360a 202020 1582:          dc.b  " "
360f fca0a0 1583:          dc.b  <+128>,"|   Written by
3629 9b 1584:          dc.b  $9B
362a 202020 1585:          dc.b  " "
362f fca0a0 1586:          dc.b  <+128>,"| Stephen J. Car
3649 9b 1587:          dc.b  $9B
364a 202020 1588:          dc.b  " "
364f fcd3ed 1589:          dc.b  <+128>,"|Smartos hardware D
3669 9b 1590:          dc.b  $9B
366a 202020 1591:          dc.b  " "
366f fca0a0 1592:          dc.b  <+128>,"| By Bob Woolley
3689 9b 1593:          dc.b  $9B
368a 202020 1594:          dc.b  " "
368f fca0a0 1595:          dc.b  <+128>,"| Stephen J. Car
36a9 9b 1596:          dc.b  $9B
36aa 202020 1597:          dc.b  " "

```

```
36af fca0d0 1598:      dc.b  <+128>,"| Press Start or op
36c9 9b 1599:      dc.b  $9B
36ca 202020 1600:      dc.b  "  "
36cf fca0a0 1601:      dc.b  <+128>,"| Select to freez
36e9 9b 1602:      dc.b  $9B
36ea 202020 1603:      dc.b  "  "
36ef 9a9292 1604:      dc.b  <+128>,"
```