


```

0000 1: ;-----
0000 2: ;
0000 3: ;      Copyright 2010 Intergraded Logic      Sys
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: ;      oss utility for real.dos v1.0)
0000 10: ;
0000 11: ;-----
0000 12: ;
0000 13: ;      Dis_bat.com
0000 14: ;      This program disables batch files from running!
0000 15: ;
0000 16: ;      Code History.  Init design, 11/12/2005  sjc
0000 17: ;
0000 18: ;      This version will only turn off batch processing ew
0000 19: ;      to reboot to enable batch processings.
0000 20: ;
0000 21: ;      Updated the screen Print Utility so it will work wi
0000 22: ; This change will also help "Print P:"          6/1
0000 23: ;
0000 24: ;
0000 25: ;
0000 26: ;Updated the screen Print Utility so it will work with the
0000 27: ; This change will also help "Print P:"          8/1
0000 28: ;
0000 29: ;
0000 30: ;Also added code for Control+1 pause of screen to the high
0000 31: ;handling.          8/24/2012  sjc
0000 32: ;
0000 33: ;      Dis_bat.com turns on and off batch processing. 8/29
0000 34: ;
0000 35: ;      Still have some cleanup code to write
0000 36: ;
0000 37: ;
0000 38: ;
0000 39: ;
0000 40: ;
0000 41: ;
0000 42: ;
0000 43: ;int.dis_bat
0000 44: ;-----
0000 45: ; Notes: o This source code MAY NOT be placed for download
0000 46: ;      o "mea" is a macro that loads the address of the
0000 47: ;      into the pointer specified by the second field.
0000 48: ;-----
0000 49: ; Assembler: MADMAc (tm) ST Cross Assembler (Atari Corp
0000 50: ;      XASM st and IBM VERSIONS
0000 51: ;-----
0000 52: com_file_name:      .macro
0000 53: ;      dc.b 12,"filename.ext",$9b

```

```
0000 54:          dc.b 12,"Dis_bat.com ",9b
0000 55:          .endm
0000 56:
0000 57: ;~~~~~
0000 58: ; File revision history. Version Number in hex
0000 59: ;
0000 60: file_ver:      equ $12
0000 61: ;~~~~~
0000 62: ; the Month in dec   for the first time this revision
0000 63: ;was compiled
0000 64: ;
0000 65: c_month:      equ 8
0000 66: ;~~~~~
0000 67: ; the day in dec    for the first time this revision
0000 68: ;was compiled
0000 69: c_day:        equ 28
0000 70: ;~~~~~
0000 71: ; the year in dec   for the first time this revision
0000 72: ;was compiled
0000 73: c_year:       equ 2012
0000 74: ;~~~~~
0000 75: ; Type of compiler used in program
0000 76: ;
0000 77: ; 0 = unknown Compiler
0000 78: ; 1 = xasm
0000 79: ; 2 = mac_65
0000 80: ; 3 = basic
0000 81: ; 4 = compiled basic xl
0000 82: ; 5 = C65
0000 83: ; 6 = Action
0000 84: ;
0000 85: ; We can add more as time goes on!
0000 86: ;
0000 87: ;
0000 88: xasm:         equ 1
0000 89: mac_65:      equ 2
0000 90: basic:       equ 3
0000 91: basicxl:     equ 4
0000 92: c65:        equ 5
0000 93: action:      equ 6
0000 94:
0000 95: file_compiler: equ xasm
0000 96: ;~~~~~
0000 97: ; is this relocatable code ?
0000 98: ; anyother value other than 1 or 2 would be unknown
0000 99: r..yes:      equ 1
0000 100: r..no:       equ 2
0000 101: ;
0000 102: ;~~~~~
0000 103: ; Gotta   define if it can be relocated
0000 104: l_relocatable: equ r..no
0000 105: ;
0000 106: ;~~~~~
```

```

0000      107: ; Gotta define if it can be relocated
0000      108: r.crl:          equ   $7d
0000      109: r.crlf:         equ   $9b
0000      110: r.space:        equ   $20
0000      111:
0000      112: l_frstscreenbyte: equ   r..space
0000      113: ;~~~~~
0000      114: ; The language the output file is in.
0000      115: ;
0000      116: ;
0000      117: ; 0 =   Undefined
0000      118: ; 1 =   English
0000      119: ; 2 =   German
0000      120: ;
0000      121: r..language:     equ   1
0000      122:
0000      123: ;      System Equates
0000      124: ;-----
0000      125: ;
0000      126: ; $04
0000      127:
0000      128:
0000      129:
0000      130:          .include  equates
0000      131:          .include  globals
0000      132:          .include  macros
0000      133:
0000      134:          .org   $4000
0000      135: ;
0000      136: win_start:
0000      137: header_info:
0000      138:          .include  header
0000      139: ;
0000      140: start:
0000      141:          lda   #$60
0000      142:          sta   header_start
0000      143:
0000      144:          jsr   os_off
0000      145:          lda   int.dis_bat
0000      146:          sta   hold.data
0000      147:          jsr   os_on
0000      148:
0000      149:          lda   hold.data
0000      150:          cmp   #$60
0000      151:          beq   .aa
0000      152:
0000      153:          jsr   os_off
0000      154:          mea   tsr,s_ptr
0000      155:          lda   #dis_bat
0000      156:          jsr   seclvl
0000      157:          bmi   .init
0000      158: ;
0000      159: ;

```

```

42bf 160: ;          jsr  os_on
42bf 161:
42bf 162:
42bf 4c3f43 163:          jmp  turn_on_bat
42c2 4ccb42 164:          jmp  turnoff_dis_bat
42c5 165:
42c5 166: .init:
42c5 207d45 167:          jsr  os_on
42c8 4c3f43 168:          jmp  turn_on_bat
42cb 169: ;-----
42cb 170: ;this places the location of the E:handler that does
42cb 171: ;batch files to be enable and work
42cb 172: ;-----
42cb 173: .aa:
42cb 174: turnoff_dis_bat:
42cb 175:
42cb a000 176:          LDY  #0
42cd b91a03 177: .a:          LDA  $031a,Y
42d0 c945 178:          CMP  #'E'
42d2 f005 179:          BEQ  .b
42d4 c8 180:          INY
42d5 c8 181:          INY
42d6 c8 182:          INY
42d7 d0f4 183:          BNE  .a
42d9 184: .b:
42d9 8c3d43 185:          sty  .temp
42dc 207245 186:          jsr  os_off
42df ac3d43 187:          ldy  .temp
42e2 188:
42e2 b91b03 189:          lda  $031a+1,Y
42e5 8df1ff 190:          sta  int.dis_bat
42e8 b91c03 191:          lda  $031a+2,Y
42eb 8df2ff 192:          sta  int.dis_bat+1
42ee 193:
42ee a900 194:          LDA  # low $e400
42f0 991b03 195:          STA  $031a+1,Y
42f3 a9e4 196:          LDA  # high $E400
42f5 991c03 197:          STA  $031a+2,Y
42f8 198:
42f8 199:          mea  tsr,s_ptr
42f8 a9f5 ----:          lda  #low tsr
42fa 8582 ----:          sta  s_ptr
42fc a9ff ----:          lda  #high tsr
42fe 8583 ----:          sta  s_ptr + 1
4300 a917 200:          lda  #dis_bat
4302 20e945 201:          jsr  setlvl
4305 207d45 202:          jsr  os_on
4308 203:
4308 204:
4308 20d943 205:          jsr  printsi
430b 9b9b 206:          dc.b  $9b,$9b
430d 205265 207:          dc.b  " RealDos Batch Processor i
432e 205475 208:          dc.b  " Turned "

```

```

4336 cfc6c6 209:      dc.b  <+128>,"OFF"
4339 9b9b  210:      dc.b  $9b,$9b
433b ff    211:      dc.b  -1
433c 60    212:      RTS
433d      213:
433d 00    214: .temp:      dc.b  0
433e 00    215: hold.data:  dc.b  0
433f      216: ;
433f      217:
433f      218:
433f      219: turn_on_bat:
433f a000  220:      LDY  #0
4341 b91a03 221: .a:      LDA  $031a,Y
4344 c945  222:      CMP  #'E'
4346 f005  223:      BEQ  .b
4348 c8    224:      INY
4349 c8    225:      INY
434a c8    226:      INY
434b d0f4  227:      BNE  .a
434d      228: .b:
434d 8cb043 229:      sty  .temp
4350 207245 230:      jsr  os_off
4353 acb043 231:      ldy  .temp
4356 adf1ff 232:      lda  int.dis_bat
4359 991b03 233:      sta  $031a+1,Y
435c adf2ff 234:      lda  int.dis_bat+1
435f 991c03 235:      sta  $031a+2,Y
4362      236:
4362 a960  237:      lda  #$60
4364 8df1ff 238:      sta  int.dis_bat
4367 a900  239:      lda  #$00
4369 8df2ff 240:      sta  int.dis_bat+1
436c      241:
436c      242:      mea  tsr,s_ptr
436c a9f5  ----:      lda  #low tsr
436e 8582  ----:      sta  s_ptr
4370 a9ff  ----:      lda  #high tsr
4372 8583  ----:      sta  s_ptr + 1
4374 a917  243:      lda  #dis_bat
4376 202b46 244:      jsr  clrlvl
4379 207d45 245:      jsr  os_on
437c      246:
437c 20d943 247:      jsr  printsi
437f 9b9b  248:      dc.b  $9b,$9b
4381 205265 249:      dc.b  " RealDos Batch Processor i
43a2 205475 250:      dc.b  " Turned "
43aa cfce  251:      dc.b  <+128>,"ON"
43ac 9b9b  252:      dc.b  $9b,$9b
43ae ff    253:      dc.b  -1
43af 60    254:      RTS
43b0      255:
43b0 00    256: .temp:      dc.b  0
43b1      257:

```

```

43b1 258:
43b1 259:
43b1 260:
43b1 261:
43b1 262: ; ~~~~~
43b1 263: set_handler:
43b1 207245 264:      jsr  os_off
43b4 265:      mea  tsr,s_ptr
43b4 a9f5 ----:      lda  #low tsr
43b6 8582 ----:      sta  s_ptr
43b8 a9ff ----:      lda  #high tsr
43ba 8583 ----:      sta  s_ptr + 1
43bc a917 266:      lda  #dis_bat
43be 20e945 267:      jsr  setlvl
43c1 207d45 268:      jsr  os_on
43c4 60 269:      rts
43c5 270:
43c5 271: ; ~~~~~
43c5 272: clear_handler:
43c5 207245 273:      jsr  os_off
43c8 274:      mea  tsr,s_ptr
43c8 a9f5 ----:      lda  #low tsr
43ca 8582 ----:      sta  s_ptr
43cc a9ff ----:      lda  #high tsr
43ce 8583 ----:      sta  s_ptr + 1
43d0 a917 275:      lda  #dis_bat
43d2 202b46 276:      jsr  clrlvl
43d5 207d45 277:      jsr  os_on
43d8 60 278:      rts
43d9 279:
43d9 280:
43d9 281:
43d9 282:
43d9 283:
43d9 284:
43d9 285: ;-----
43d9 286: ;prints Routeen macro
43d9 287: ;this is a slow print to the screen
43d9 288: ;usage jsr  prints
43d9 289: ; .byte $9b," string to be printed",$ff
43d9 290: ;-----
43d9 291: ;
43d9 292: prints:
43d9 68 293:      pla
43da 8dea43 294:      sta  .pstr+1
43dd 68 295:      pla
43de 8deb43 296:      sta  .pstr+2
43e1 eeea43 297: .prsl:      inc  .pstr+1
43e4 d003 298:      bne  .pstr
43e6 eeeb43 299:      inc  .pstr+2
43e9 adffff 300: .pstr:      lda  $ffff
43ec c9ff 301:      cmp  #$ff
43ee f006 302:      beq  .estri

```

```

43f0 202f44 303:          jsr  fast_output
43f3 4ce143 304:          jmp   .prsl
43f6 adeb43 305: .estri:      lda   .pstr+2
43f9 48      306:          pha
43fa adea43 307:          lda   .pstr+1
43fd 48      308:          pha
43fe 60      309:          rts
43ff      310: ;
43ff      311: ;
43ff      312: ;
43ff      313: ;
43ff      314: ;-----
43ff      315: ;actual screen handler
43ff      316: ;screen offset definitions
43ff      317: ;-----
43ff      318: scr_offset:
43ff 0000 319:          dc.w  0
4401 2800 320:          dc.w  40
4403 5000 321:          dc.w  80
4405 7800 322:          dc.w 120
4407 a000 323:          dc.w 160
4409 c800 324:          dc.w 200
440b f000 325:          dc.w 240
440d 1801 326:          dc.w 280
440f 4001 327:          dc.w 320
4411 6801 328:          dc.w 360
4413 9001 329:          dc.w 400
4415 b801 330:          dc.w 440
4417 e001 331:          dc.w 480
4419 0802 332:          dc.w 520
441b 3002 333:          dc.w 560
441d 5802 334:          dc.w 600
441f 8002 335:          dc.w 640
4421 a802 336:          dc.w 680
4423 d002 337:          dc.w 720
4425 f802 338:          dc.w 760
4427 2003 339:          dc.w 800
4429 4803 340:          dc.w 840
442b 7003 341:          dc.w 880
442d 9803 342:          dc.w 920
442f      343: ;
442f      344: fast_output:
442f 855a 345:          sta  $5a
4431 ad0407 346:          lda  sc.redirect
4434 c902 347:          cmp  #print_p
4436 f01f 348:          beq  .go_cio
4438      349: .leo:
4438 adff02 350:          lda  $02ff      ; CONTROL 1
443b c900 351:          cmp  #$00
443d d0f9 352:          bne  .leo
443f      353:
443f a55a 354:          lda  $5a
4441 c920 355:          cmp  #32

```



```

4443 9004 356:      bcc .do_look
4445 c97d 357:      cmp #125
4447 9013 358:      bcc .do_here
4449 a000 359: .do_look:      ldy #0
444b b95245 360: .lookup:      lda .char_table,y
444e f00c 361:      beq .do_here
4450 c55a 362:      cmp $5a
4452 f003 363:      beq .go_cio
4454 c8 364:      iny
4455 d0f4 365:      bne .lookup
4457      366:
4457      367:
4457 a55a 368: .go_cio:      lda $5a
4459 4c6245 369:      jmp putlocal
445c      370:
445c      371:
445c a000 372: .do_here:      ldy #0
445e a55d 373:      lda $5d
4460 915e 374:      sta ($5e),y
4462 a55a 375:      lda $5a
4464 c99b 376:      cmp #$9b
4466 f044 377:      beq .docr
4468 203145 378: .notcr:      jsr .get_adr
446b a55a 379:      lda $5a
446d 297f 380:      and #$7f
446f c920 381:      cmp #32
4471 9007 382:      bcc .add64
4473 c960 383:      cmp #96
4475 9009 384:      bcc .sub32
4477 4c8344 385:      jmp .asis
447a 18 386: .add64:      clc
447b 6940 387:      adc #64
447d 4c8344 388:      jmp .asis
4480 38 389: .sub32:      sec
4481 e920 390:      sbc #32
4483 245a 391: .asis:      bit $5a
4485 1002 392:      bpl .xxlate
4487 0980 393:      ora #$80
4489 a000 394: .xxlate:      ldy #0
448b 915e 395:      sta ($5e),y
448d e655 396:      inc $55
448f e663 397:      inc $63
4491 a555 398:      lda $55
4493 c928 399:      cmp #40
4495 b019 400:      bcs .next_row
4497 c8 401:      iny
4498 b15e 402:      lda ($5e),y
449a 855d 403:      sta $5d
449c 0980 404:      ora #$80
449e aef002 405:      ldx 752
44a1 d002 406:      bne .nocurs1
44a3 915e 407:      sta ($5e),y
44a5 e65e 408: .nocurs1:      inc $5e

```

```

44a7 d002 409:      bne  .nooav
44a9 e65f 410:      inc  $5e+1
44ab 60   411: .nooav:      rts
44ac a900 412: .docr:      lda  #0
44ae 8563 413:      sta  $63
44b0 a552 414: .next_row:    lda  $52
44b2 8555 415:      sta  $55
44b4 e654 416:      inc  $54
44b6 a454 417:      ldy  $54
44b8 c018 418:      cpy  #24
44ba b013 419:      bcs  .scroll
44bc 203145 420:      jsr  .get_adr
44bf a000 421:      ldy  #0
44c1 b15e 422:      lda  ($5e),y
44c3 855d 423:      sta  $5d
44c5 aef002 424:      ldx  752
44c8 d004 425:      bne  .nocurs2
44ca 0980 426:      ora  #$80
44cc 915e 427:      sta  ($5e),y
44ce 60   428: .nocurs2:      rts
44cf      429:
44cf c654 430: .scroll:      dec  $54
44d1 a558 431:      lda  $58
44d3 8568 432:      sta  $68
44d5 18   433:      clc
44d6 6928 434:      adc  #40
44d8 855e 435:      sta  $5e
44da a559 436:      lda  $58+1
44dc 8569 437:      sta  $68+1
44de 6900 438:      adc  #0
44e0 855f 439:      sta  $5e+1
44e2 a000 440:      ldy  #0
44e4 a203 441:      ldx  #3
44e6 b15e 442: .sloop:      lda  ($5e),y
44e8 9168 443:      sta  ($68),y
44ea c8   444:      iny
44eb d0f9 445:      bne  .sloop
44ed e65f 446:      inc  $5e+1
44ef e669 447:      inc  $68+1
44f1 ca   448:      dex
44f2 d0f2 449:      bne  .sloop
44f4 b15e 450: .sloop2:      lda  ($5e),y
44f6 9168 451:      sta  ($68),y
44f8 c8   452:      iny
44f9 c098 453:      cpy  #152
44fb 90f7 454:      bcc  .sloop2
44fd a558 455:      lda  $58
44ff 18   456:      clc
4500 6998 457:      adc  #920&$ff
4502 855e 458:      sta  $5e
4504 a559 459:      lda  $58+1
4506 6903 460:      adc  #920/256
4508 855f 461:      sta  $5e+1

```

```

450a a000 462:          ldy  #0
450c 98   463:          tya
450d 915e 464: .clear:      sta  ($5e),y
450f c8   465:          iny
4510 c028 466:          cpy  #40
4512 90f9 467:          bcc  .clear
4514 aef002 468:          ldx  752
4517 d006 469:          bne  .nocurs3
4519 a980 470:          lda  #$80
451b a455 471:          ldy  $55
451d 915e 472:          sta  ($5e),y
451f a900 473: .nocurs3:      lda  #0
4521 855d 474:          sta  $5d
4523 a55e 475:          lda  $5e
4525 18   476:          clc
4526 6555 477:          adc  $55
4528 855e 478:          sta  $5e
452a a55f 479:          lda  $5e+1
452c 6900 480:          adc  #0
452e 855f 481:          sta  $5e+1
4530 60   482:          rts
4531 a554 483: .get_adr:      lda  $54
4533 0a   484:          asl
4534 a8   485:          tay
4535 a558 486:          lda  $58
4537 18   487:          clc
4538 79ff43 488:          adc  scr_offset,y
453b 855e 489:          sta  $5e
453d a559 490:          lda  $58+1
453f 790044 491:          adc  scr_offset+1,y
4542 855f 492:          sta  $5e+1
4544 a55e 493:          lda  $5e
4546 18   494:          clc
4547 6555 495:          adc  $55
4549 855e 496:          sta  $5e
454b a55f 497:          lda  $5e+1
454d 6900 498:          adc  #0
454f 855f 499:          sta  $5e+1
4551 60   500:          rts
4552      501: .char_table:
4552 1b1c1d 502:          dc.b  27,28,29,30,31,125,126,127
455a 9c9d9e 503:          dc.b  156,157,158,159,253,254,255
4562      504:
4562      505:
4562      506: putlocal:
4562 a20b 507:          ldx  #11
4564 8e4203 508:          stx  $0342
4567 a200 509:          ldx  #0
4569 8e4803 510:          stx  $0348
456c 8e4903 511:          stx  $0349
456f 4c56e4 512:          jmp  $e456
4572      513: ;
4572      514: ;~~~~~

```

```

4572      515: ; OS on and off ROUTEEN
4572      516: ;~~~~~
4572      517: ;
4572      518:
4572      519: os_off:
4572 78    520:          SEI
4573 ad01d3 521:          LDA  PORTB
4576 29fe  522:          AND  #$FE
4578 8d01d3 523:          STA  PORTB
457b 58    524:          CLI
457c 60    525:          RTS
457d      526: ;
457d      527: os_on:
457d 78    528:          SEI
457e ad01d3 529:          LDA  PORTB
4581 0901  530:          ORA  #1
4583 8d01d3 531:          STA  PORTB
4586 58    532:          CLI
4587 60    533:          RTS
4588      534:
4588      535: ;~~~~~
4588      536: ;checks user security level pointed to by S_PTR to see if
4588      537: ;bit specified in the A register is ON. Returns MINus if
4588      538: ;user doesn't have the specified security level.
4588      539: ;~~~~~
4588      540:
4588 8ce345 541: seclvl:          sty  .tempy
458b 8ee445 542:          stx  .tempx
458e c900  543:          cmp  #0          ;se
4590 f042  544:          beq  .passed      ;ye
4592 a8     545:          tay          ;:-1
4593 a980  546:          lda  #128
4595 8de545 547:          sta  .temp          ;in
4598 a900  548:          lda  #0
459a 8de645 549:          sta  .temp+1
459d 8de745 550:          sta  .temp+2
45a0 8de845 551:          sta  .temp+3
45a3 88    552:          dey
45a4 f011  553:          beq  .mask          ;if
45a6 4ee545 554: .shloop:        lsr  .temp          ;sh
45a9 6ee645 555:          ror  .temp+1
45ac 6ee745 556:          ror  .temp+2
45af 6ee845 557:          ror  .temp+3
45b2 88    558:          dey
45b3 d0f1  559:          bne  .shloop
45b5 a000  560:          ldy  #0
45b7 b9e545 561: .mask:          lda  .temp,y          ;no
45ba 3182  562:          and  (s_ptr),y
45bc 99e545 563:          sta  .temp,y
45bf c8     564:          iny
45c0 c004  565:          cpy  #4
45c2 d0f3  566:          bne  .mask
45c4 ade545 567: .check:        lda  .temp

```

```

45c7 0de645 568:      ora  .temp+1      ; i
45ca 0de745 569:      ora  .temp+2
45cd 0de845 570:      ora  .temp+3
45d0 c900  571:      cmp  #0
45d2 f004  572:      beq  .failed
45d4 a900  573: .passed:      lda  #0
45d6 f002  574:      beq  .return
45d8 a980  575: .failed:      lda  #$80
45da 08    576: .return:      php
45db ace345 577:      ldy  .tempy
45de aee445 578:      ldx  .tempx
45e1 28    579:      plp
45e2 60    580:      rts
45e3      581:
45e3 00    582: .tempy:      dc.b  0
45e4 00    583: .tempx:      dc.b  0
45e5 000000 584: .temp:      dc.b  0,0,0,0
45e9      585:
45e9      586:
45e9      587: ; sets the bit specified (in the A register) of the se
45e9      588: ; level pointed to by S_PTR to a TRUE value.
45e9      589: ; ~~~~~
45e9      590:
45e9 8c2546 591: setlvl:      sty  .tempy
45ec 8e2646 592:      stx  .tempx
45ef c900  593:      cmp  #0      ;se
45f1 f031  594:      beq  .return  ;ye
45f3 a8    595:      tay      ;:-
45f4 a980  596:      lda  #128
45f6 8d2746 597:      sta  .temp      ;in
45f9 a900  598:      lda  #0
45fb 8d2846 599:      sta  .temp+1
45fe 8d2946 600:      sta  .temp+2
4601 8d2a46 601:      sta  .temp+3
4604 88    602:      dey
4605 f011  603:      beq  .mask      ;if
4607 4e2746 604: .shloop:      lsr  .temp      ;sh
460a 6e2846 605:      ror  .temp+1
460d 6e2946 606:      ror  .temp+2
4610 6e2a46 607:      ror  .temp+3
4613 88    608:      dey
4614 d0f1  609:      bne  .shloop
4616 a000  610:      ldy  #0
4618 b92746 611: .mask:      lda  .temp,y      ;no
461b 1182  612:      ora  (s_ptr),y
461d 9182  613:      sta  (s_ptr),y
461f c8    614:      iny
4620 c004  615:      cpy  #4
4622 d0f4  616:      bne  .mask
4624 60    617: .return:      rts
4625      618:
4625 00    619: .tempy:      dc.b  0
4626 00    620: .tempx:      dc.b  0

```

```

4627 000000 621: .temp:      dc.b  0,0,0,0
462b      622:
462b      623:
462b      624:
462b      625:
462b      626: ;   CLEARS the bit specified (in the A register) of the
462b      627: ; security level pointed to by S_PTR to a FALSE value.
462b      628: ; ~~~~~
462b      629:
462b 8c7646 630: clrlvl:  sty  .tempy
462e 8e7746 631:          stx  .tempx
4631 c900   632:          cmp  #0          ;sec lvl of zero?
4633 f040   633:          beq  .return      ;yes..just exit (s
4635 a8     634:          tay           ;,-1 for shifting
4636 a980   635:          lda  #128
4638 8d7846 636:          sta  .temp        ;initialize mask
463b a900   637:          lda  #0
463d 8d7946 638:          sta  .temp+1
4640 8d7a46 639:          sta  .temp+2
4643 8d7b46 640:          sta  .temp+3
4646 88     641:          dey
4647 f011   642:          beq  .mask        ;if it was one, no
4649 4e7846 643: .shloop: lsr  .temp        ;shift bits down t
464c 6e7946 644:          ror  .temp+1
464f 6e7a46 645:          ror  .temp+2
4652 6e7b46 646:          ror  .temp+3
4655 88     647:          dey
4656 d0f1   648:          bne  .shloop
4658 a000   649:          ldy  #0
465a b97846 650: .mask:   lda  .temp,y      ;now, XOR the bit
465d 49ff   651:          eor  #$ff
465f 997846 652:          sta  .temp,y
4662 c8     653:          iny
4663 c004   654:          cpy  #4
4665 d0f3   655:          bne  .mask
4667 a000   656:          ldy  #0
4669 b97846 657: .low:   lda  .temp,y      ;now, AND the bit
466c 3182   658:          and  (s_ptr),y
466e 9182   659:          sta  (s_ptr),y
4670 c8     660:          iny
4671 c004   661:          cpy  #4
4673 d0f4   662:          bne  .low
4675 60     663: .return: rts
4676      664:
4676 00     665: .tempy:  dc.b  0
4677 00     666: .tempx:  dc.b  0
4678 000000 667: .temp:   dc.b  0,0,0,0
467c      668:
467c      669:
467c      670:
467c      671:
467c      672:
467c      673: ;

```

```
467c    674: win_end:      ds.b  0
467c    675: ;
467c    676: ;
467c    677:      ds.b  0
467c    678: ;
467c    679: ;
```

End assembly: no errors