

```

*****
*
*           FinderLaunch - An AppleWorks Init
*
*   - Prevents Crash to Monitor when AppleWorks
*     is Re-launched from the GS/OS Finder after
*       previously launching and quitting in
*         a single session.
*
*   - Init must be present in the AppleWorks
*     inits directory
*       (e.g. /HD/AppleWorks/AW.Inits)
*   - Init Manager must be activated in
*     AppleWorks 'Standard Settings' Menu
*
*           Version 1.0 (for AppleWorks Version 5.1)
*           (c) 2015 Hugh Hood
*
*****

```

```

        TR      ADR          ; truncate bank address

```

```

        XC                      ; enable 65C02 code

```

```

* Equates *

```

```

EMULSTACK    EQU    $100      ; in AuxMEM - $01/0100
Seg00Type    EQU    $FE0      ; memory manager type
                                   ; $41 = SEG.AM / $52 = SEG.RM (IIGS) /
                                   ; $58 = SEG.XM / $4D = Deja IIX (Mac)
InitAdr      EQU    $4000     ; load address for Init files
MainZP       EQU    $C008     ; Main Zero Page (AppleWorks)
AuxZP        EQU    $C009     ; Aux Zero Page (UltraMacros)

```

```

        ORG      InitAdr     ; ($4000)

```

```

*****
*           Init Header
*
*****
START

```

```

        JMP      IStart      ; skip over header

```

```

**-----

```

```

        ASC      'mb'        ; Init ID Bytes (AW 5.1)
        DB        $0A         ; Init Version - programmer assigned
                                   ; e.g. - $0A/1.0 $0B/1.1 $1A/2.6
        STR      'FinderLaunch'
        ; Init Screen Name (max 18 character)
        HEX      0000         ; Header End Bytes

```

```

**-----

```

```

IStart      LDA      Seg00Type ; Memory Manager Type
            CMP      #$52      ; Is it Apple IIGS? (SEG.RM)
            BNE      Done      ; disregard - IIGS only

```

```

SEI                ; disable interrupts
LDA    #$7F        ; change value to start halfway
STA    AuxZP        ; switch in AuxZP and LC
                    ; (UltraMacros runs in AuxMEM)
STA    EMULSTACK    ; store MainZP Stack pointer here
                    ; when AuxZP and LC are in use

STA    MainZP        ; switch in MainZP and LC
                    ; (AppleWorks runs in MainMEM)
CLI                ; re-enable interrupts

```

*-----

*

* NOTE: For background information on this issue, see page 152-153 of the
 * Apple IIe Technical Reference Manual, page 13 of the Apple IIgs
 * Firmware Reference Manual, IIgs Tech Note #88, GS/OS source code
 * for 'GQuit', and GSbug revision notes for Version 1.5b15.

*

* After quitting AppleWorks and returning to the GS/OS Finder, the location,
 * \$01/100, also called EMULSTACK, [and 'mnemstkptr' in the GS/OS GQuit/P8Quit
 * source code] is set to the value of the stack pointer, which will contain
 * either a \$B2 (AW 5.1) or a \$B4 (AW 3.0 with Ultra4).

*

* This value is NOT placed there by AppleWorks, but by GS/OS, which is using
 * a routine located at \$E0/D8AD in language card Bank 0 memory. This routine
 * reads the value of the page 1 stack pointer and saves it at EMULSTACK.

*

* By convention, EMULSTACK is SUPPOSED to receive the value of MAIN zero page
 * stack pointer prior to a program's switching in the AUXMEM zero page and
 * stack. This allows interrupting programs (and certain IIGS tools), when
 * called while the AUXMEM zero page and stack are active, to set the correct
 * stack pointer prior to switching back to the MAIN zero page stack to
 * accomplish their task(s). AppleWorks, however, does not follow this
 * convention when switching between the MAIN and AUXMEM zero page and stack,
 * even though UltraMacros, when enabled, makes use of the AUXMEM zero page
 * and stack, and obviously must have some other method of preserving the
 * respective stack pointers prior to switching back and forth.

*

(Probably MainStackPtr @ \$B4FC in Main Memory)

*

* Consequently, upon a SECOND launch of AppleWorks an interrupting program or
 * tool will find an INCORRECT value (\$B2 or \$B4) in EMULSTACK, but will assume
 * that it is in fact a valid stack pointer for the MAIN stack, and will make
 * use of the stack from \$01B2/B4 - \$0100 for servicing the interrupt.

*

* Since it is not valid, and very likely to be too 'high' on the stack, the
 * end result is that the MAIN zero page stack is corrupted when AppleWorks
 * regains control. AppleWorks generally crashes at this point.

*

* By setting EMULSTACK to \$7F prior to starting AppleWorks, we can ensure
 * that any interrupting routines will ONLY use stack space between \$0100-\$017F,
 * and leaving \$0180-\$01FF for AppleWorks' exclusive use.

*

* Credit for this 'technique' belongs to Glen Bredon, who implemented it in
 * his PS.16.TO.8 from ProSEL 16 as a pre-launch to ANY ProDOS 8 program
 * launched from GS/OS.

*

*-----

Done RTS ; back to Init Manager

PatchEnd	EQU	*
	SAV	I.FINDERLAUNCH
	LST	OFF
	END	