H3 Software System Tech Notes

H3 New Features Overview

The H3 software system provides a common bootable diskette for all Microengine product lines: SB1600, ME1600, and WD0900. The operating system does this by interrogating the hardware configuration it is running on and configures the software system accordingly. Note in order to achieve a common object the H3 bootstrap differs from previous boots. Therefore only H3 level bootstraps will work with the H3 release. See Section 6.2, 6.3 and Appendix G of the III.0 Operating System Reference Manual (III.0 Software Manual) for more details concerning the H3 bootstrap sequence.

Software support for SB1600 series.

a) Serial port B baud rate is software selectable using Setup to control the field 'BAUDRATE'.

b) Software floppy disk single-double density selection. This includes automatic boot up of either single or double density diskettes. In addition during normal system operation, diskettes with different densities may be freely interchanged.

c) Software support for hardware detection of memory parity and memory reply time-out.

Software support for Winchester disk on ME1600 series.

a) Sysgen capability to specify Winchester configuration for number of volumes. See Section 6.13 of III.0 Software Manual and Appendix A of this document for details.

b) Number of disk volumes increased from maximum of 8 to a maximum of 236.

c) Automatic incorporation of Winchester I/O driver if there is a Winchester drive on the system.

d) Software controlled single-double density selection. This feature requires the floppy disk controller card be upgraded to hardware level B8.

Enhanced console I/O capability. Output to a CRT is now twice as fast on 128K-byte systems as previous releases.

Number of directory accesses lowered by more efficient algorithm handling directory lookup.

System configurability for the operating system by means of 128 segment capability and segment processes which are used for system configurability.

The H3 system is configured for the class of terminals including the Soroc IQ120 and the Ampex Dialog 80. For users with Volker-Craig model terminals, switch #1 on the back of the terminal may be set to ADM-3 mode rather than VK404 mode. In this mode, the Volker-Craig terminal emulates a Soroc terminal. Note that the baudrate switch on the back of the Volker-Craig terminal should be set to 9600 baud.

Diagnostics run during boot up with output indicating any faults as a message to user. See Appendix G of the III.0 Software Manual for details.

Timekeeper process for ME1600 system to allow access to system clock. See Section 7.5.5 of the III.0 Software Manual for more details.

Protection during diskette removal so that the correct diskette is on line during loading of overlayable code segments. If the boot diskette is not on-line when required, the operating system will prompt with "Insert OSH3 in drive 4. Type <space> when ready or <esc> to abort". If the diskette with a user program is not on-line, the operating system will prompt with "Insert PROGRAM diskette in drive __. Type <space> when ready or <esc> to abort.", where the correct drive number will be displayed. Typing space with the correct diskette on line will cause the required code segment to be loaded. For example, removal of the system diskette during editing will not now cause a system hang.

The system is shipped with a SYSTEM.MISCINFO for disk drives that have at le least a 6 ms. seek rate. For systems with slower drives, a second file, SLOW.MISCINFO, is supplied. This has a 15 ms. seek rate. If your system has slow disk drives, change SLOW.MISCINFO to SYSTEM.MISCINFO on the H3 diskette and then boot.

New Feature Details

Operating System

128 Segments

The H3 software release extends the 128 segment capability to static (system segment vector) as well as vectored (user segment vector) code files. Prior to the H3 release, programs compiled with the $U- directive were restricted to 16 segments (0 to 15). With the H3 release, these programs may now contain up to 128 segments (0 to 127).

NOTE! Since $U- programs use the system segment vector at run time, they must coexist non-destructively with the operating system's segments. Thus to avoid replacing operating system segments by $U- program segments, all $U- programs should declare dummy segments for any segment in use by the operating system. The only segments guaranteed NOT to be operating system segments are segments 1, 8..15. Therefore, any $U- program which includes segments other than these may not be capable of coexisting non-destructively with the operating system.

This extension is upward compatible in that any code file generated by releases prior to H3 will run under H2.

The intrinsic FILLCHAR has been optimized. It is up to 15 times faster than in previous versions.

The serial port control registers may now be altered by a user program without conflict with operating system operation. For example, parity may be enabled by altering the controll register in the WD1931 USART. Operating system operation sets DTR, normal mode, and receiver enable at initialization only. The RTS bit is altered during write operations, but all other control regi register bits may be altered without the operating system overriding changes.

Boot Sequence

The SB1600 and the ME1600 with Winchester support allow a bootable floppy disk to be inserted in any drive for boot-up. For a typical two drive system, booting form the left drive causes the boot unit to be #4: and booting from the right drive causes the boot unit to be #5:.

See Section 6.13 of the III.0 Software Manual for details.

Debugger

The Debugger has been enhanced as follows:

a) Memory output may be viewed as decimal, hex, binary, ascii, or unsigned decimal.

b) Memory output display may be targeted to any file or device.

c) Memory output may be stopped by hitting any character.

d) The C(hain command has been enhanced to automatically display all levels of call in the mark stack control word chain for either static or dynamic link mode..

e) The O(ffset command has been enhanced to display the address and contents of offsets from the current MP.

f) A new command T(asks has been added in X(amine mode. This allows the display of all user tasks in the system. The C(hain command can then be used to inspect each tasks mark stack control word chain.

g) The S(tatus command now displays the entire Task Information Block and the ready queue and system segment vector pointer.

h) The Debugger is a SEGMENT PROCESS which allows more flexibility in debugging programs with tasks as the Debugger no longer needs be loaded in the stack space of a task to be debugged. This frees the user from having to increase a task stack size for debug purposes.

See section 6.15 of the III.0 Software Manual for details.

Copy

The Copy program performs background file transfers, while the system may be used to do normal work in the foreground. The H2 version of the Copy program required a field manipulated by Setup for Copy usage. This is not required for H3 operation. Note due to the nature of asynchronous tasks, if a foreground program writes to the same destination file as the Copy program the output of the two tasks will be intermixed.

Pascal Compiler

128 static segments upgrade :

See Section 6.7.3 of the III.0 Software Manual.

Segment processes

The construct SEGMENT PROCESS is implemented on the H3 release. This capability allows processes to be started so that their code is resident only when the task is started. However when a segment process terminates its code is not automatically deallocated. The space can be deallocated by use of the MARK and RELEASE intrinsics. In order for these to work, no other tasks in the system may do a NEW to allocate from the heap. If no NEW is executed then by doing a MARK before the START of the SEGMENT PROCESS and a release as the last statement of the SEGMENT PROCESS, the memory space may be reclaimed. Note also that if this is to be done for two SEGMENT PROCESS's the order of MARK and RELEASE for each process must be symmetric.

Librarian

The librarian has been modified to support static 128 segment code files. Protocol for dealing with static 128 segment code files is identical to that pertaining to vectored 128 segment code files (see Section 6.5 of the III.0 Software Manual). The user interface has been modified slightly to allow <esc> alternatives at points where the user was previously committed to actions that may not have been desired.

Libmap

Libmap has been updated to support static 128 segment code files. The changes are functional only. The commands seen by the user are unchanged.

System.Library

A new unit, DELAYUNIT, has been added to the system library. Operating system support routines for this unit are loaded only for systems with 128 Kbyte memory space; systems with 64 Kbytes do not support this feature. See section 7.5.5 of the III.0 Software Manual for details.

Disasm

The Disassembler has been updated to support static 128 segment code files. The changes are functional only. The commands seen by the user are unchanged.

Setup

Two new fields have been added to Setup. The first is 'HAS NO INTERRUPTS'. This field replaces the field 'HAS 8510A' which had no meaning for the Microengine. The new field is normally false so the system runs interrupt driven. If this field is set true then non-interrupt I/O is performed. This can be useful in diagnosing problems in the interrupt subsystem, but should be set false otherwise.

The second field 'MAX SERIAL PORTS' is normally set to 0. This field controls the maximum number of serial ports supported by I/O driver processes. A 0 means use the default number for the machine. This number is 2 for the SB1600 and the WD0900 and 4 or 8 for the ME1600 depending on the number of serial cards installed. By setting 'MAX SERIAL PORTS' to non-zero and less than the maximum, the I/O drivers for these ports are not loaded. This can save about 170 words for port driver not loaded. Only when memory space is limited is it recommended that this field be non-zero.

Editor

With the H3 system, the normal screen editor is still supported. However a new editor, ADV.EDITOR, is also included. For details of its use see Section 3.2 of the III.0 Software Manual.

Filer

The Filer now determines the size of a volume for the Z(ero and B(ad block scan commands. This allows the filer to indicate to the user the maximum number of blocks a diskette may have and the number of blocks a bad block scan will cover. The B(ad block scan command now allows a bad block scan for a block range as a user option. In addition if a bad block is found, the Filer will ask if an E(xamine operation should be performed, in order to attempt to cure the bad block.

The V(olumes command now indicates whether serial and parallel I/O devices are on-line. A unit number in parentheses indicates the unit is not on-line.

A volume to volume transfer on an ME1600 system between two volumes on a Winchester disk unit will now ask for a new name for the destination volume. This is done as duplicate volume names cause file access ambiguity.

Bug fixes

Operating System

When reading an integer, READLN no longer echoes when it is waiting for end of line. Only the EOL character will be echoed.

Memory management routines now use the directory lock to assure one task does not destroy another task's directory copy.

Pascal compiler

a) The compiler now generates error #191 if a file is declared in the implementation part of a unit. This error is necessary as file initialization code may only be generated if the host program knows at compile time of the existence of a variable. Files declared in the interface are thus allowed.

b) The compiler now generates correct code for the expression ... 'a' <relop> ch ... . For example , 'x' <= ch where ch is of type char works correctly.

c) The compiler now generates correct code for the expressions 'a' in ['a'..'z'] ... and ... 'a' in chset .

d) The compiler no longer erroneously generates error #117 for the following forward type reference :

```
     type a = ^x;
     b = ^x;
     c = ^z;
     x = .....;
     z = .....;
     begin ......
```

e) The compiler no longer suffers a value range error when the following is encounterd in a Pascal program:

```
      const n1 = -1;
      one = 1;
      .......
      if i in [n1..one] then ......
```

note: The correct code causes a value range error at run time.

f) The compiler generates error #101 for the construct:

```
     procedure x; forward;
     segment procedure x; begin ...
```

g) Fixed a "bad list traversal" problem in {$u-} uses situations.

h) Fixed PMACHINE so that the construct (<variable>) now works correctly for all sizes of <variable>. previously, sizes > 1 loaded only the first word.

i) Fixed PMACHINE so that error 203 is generated when constants greater than 255 are generated. This is done as PMACHINE emits opcodes which are

Winchester Disk Support on the ME1600 Series

Introduction

The H3 release supports Winchester disk drives on the ME1600 product line. Note that H3 requires the latest version of boot PROM's as these PROM's have a boot from floppy and Winchester capability. Since a Winchester disk has so much more space than a floppy disk, a Winchester disk may be partitioned into multiple volumes. In order to flexibly preform this partitioning, each Winchester disk has a configuration record on track 0 that describes the volume partitioning. Floppy disks may also have a configuration record, but a configuration record would be put on a floppy only for some special and unusual purpose. If a floppy disk has no configuration record then the configuration defaults to drives 4,5,9-14 as floppy disk volumes and no Winchester volumes.

This document will cover configuration of a new disk, WFORMAT, CONFIGURE, and BOOTMAKE programs and migration of files from H2 based Winchester systems. A detailed description of the use of these program to configure a system is then shown.

Overview

At boot the ME1600 will first attempt to boot from any floppy disks on line. If no floppy disks are on line, the system will boot from the Winchester disk. If a floppy disk without a boot is on line then the system will attempt to boot from it. However the lack of a boot will cause booting to fail. Therefore care must be taken to assure that floppy disks are not on line when booting from Winchester is desired.

With H3 a Winchester disk's volumes or a floppy disk may be any unit number. This allows the boot unit to be any unit number, not just unit #4. It is recommended that when configuring a Winchester disk with the CONFIGURE program that the Winchester disk volumes be in the range 28..255. This is recommended as floppy disks which have no configuration records (almost all floppy disks will be in this state) are unit #4. This causes a conflict if the Winchester is also unit #4 causing the system to be unable to read the floppy disk. For flexibility in reading floppy disks, it is recommended that the Winchester units not be unit #4 or unit #5.

Another recommendation is to create some volumes on the Winchester that have the same number of blocks as a floppy disk. This allows a volume to volume transfer between floppies and a Winchester. Note that volume to volume transfer between Winchester and floppies does not transfer the bootstrap.

To configure a new unformatted Winchester drive the following steps must be followed:

1. Boot up the H3 system floppy disk.

2. X(ecute WFORMAT to format the Winchester disk. After the Winchester disk is formatted the WFORMAT program writes a configuration record to track 0 of the Winchester disk being formatted. The configuration record produced by WFORMAT describes the number of heads and cylinders the Winchester disk contains. A 10 MB Winchester disk has 2 heads, 512 cylinders, and (2 * 512) = 1024 tracks each holding 16 blocks. The 10 MB disk has (16 * 1024) = 16K blocks or 8 MB of useable data space. A 40 MB Winchester disk has 8 heads, 512 cylinders, and (8 * 512) = 4096 tracks each holding 16 blocks. The 40 MB disk has (16 * 4096) = 64K blocks or 32 MB of useable data space.

3. X(ecute CONFIGURE to partition the Winchester disk into multiple volumes. The CONFIGURE program adds this volume configuration information to the configure record initially created by WFORMAT. Note the CONFIGURE program will not run on systems with less than 128 Kbytes of memory.

4. Reboot the system so that the newly created configuration record can be read in by the operating system.

5. X(ecute BOOTMAKE to install a boot on the Winchester disk. The codefile BOOT is the bootstrap to be placed on the Winchester disk by BOOTMAKE.

6. Go to the filer and Z(ero the lowest unit number on the Winchester. Then transfer all files from the H3 floppy to the Winchester volume nearest the start of the Winchester disk.

7. Reboot the system without a floppy disk on line.

Migration from H2w to H3 on Winchester systems

Note in order to use the H3 system which does a complete boot from Winchester, new PROM's must be installed in the terminator card. These PROM's [illegible] included with the H3 release for H2w customers. Users who have used the H2w system on Winchester disks may convert to H3 by either of the following procedures:

A. Copy all files to be saved from Winchester to floppy disk. Now that all files are backed up follow the directions for initializing a Winchester disk. This entails reformatting the disk, thus requiring all important files to be saved.

B. If it is inconvenient to save all files on Winchester to floppy then use of the following procedure will preserve existing files.

1. Boot up the H3 floppy disk.

2. X(ecute Configure.

3. Type 'A' and Configure will display:

```
Convert H2 configuration record to a H3 record ?
```

Type 'y' and Configure will display:

```
Unit to convert is F(loppy or W(inchester ?
```

The Winchester is to be configured so type 'W'.

```
Convert Winchester drive [0..3] ?
```

This prompt asks which drive on line to configure. For a single drive system the drive on line is drive 0. Enter 0.

4. The H2 configuration record on the Winchester is then read and the Configure program displays the H2 configuration. Winchester units will be unit numbers in the range 4..14. In order to avoid conflict in H3 these unit numbers should be changed to unit numbers in the range 28..255 (see next step).

5. Type 'E' for E(dit then 'P' for P(rint record to display the H2 configuration.

   Type 'W' for W(inchester to modify Winchester units. Use the C(hange command to change unit numbers 4..14 to unit numbers in the range 28..255. Then W(rite out the new configuration record.

6. Reboot from the H3 floppy, go to F(iler and observe that the H2 volumes are on line.

7. X(ecute Bootmake to place a H3 boot on the Winchester.

8. Reboot the system without a floppy on line.