SIMH FAQ
07-Sep-2005

COPYRIGHT NOTICE

The following copyright notice applies to the SIMH source, binary, and documentation:

1 GENERAL QUESTIONS

1.1 WHAT IS SIMH?
1.2 WHY WAS SIMH WRITTEN?
1.3 WHAT IS THE HISTORY OF SIMH?
1.4 WHO WRITES AND MAINTAINS SIMH?
1.5 HOW IS SIMH LICENSED?
1.6 HOW IS SIMH DISTRIBUTED?
1.7 WHICH COMPUTER SYSTEMS DOES SIMH SIMULATE?
1.8 WHICH HOST SYSTEMS DOES SIMH RUN ON?
1.9 WHAT SOFTWARE PACKAGES ARE AVAILABLE TO RUN ON SIMH?
1.10 WHERE CAN I GET MORE INFORMATION ON SIMH?

2 OPERATIONAL QUESTIONS

2.1 HOW DO I INSTALL SIMH ON WINDOWS?
2.2 HOW DO I INSTALL SIMH WITH ETHERNET SUPPORT ON WINDOWS?
2.3 HOW DO I INSTALL SIMH ON UNIX?
2.4 HOW DO I INSTALL SIMH ON VMS?
2.5 HOW DO I TRANSCRIBE A REAL CD FOR USE WITH SIMH?
2.6 HOW DO I TRANSCRIBE OTHER ARCHIVAL MEDIA FOR USE WITH SIMH?
2.7 HOW CAN I GET TEXT FILES IN AND OUT OF SIMH?
2.8 HOW CAN I GET BINARY FILES IN AND OUT OF SIMH?
2.9 CAN I CONNECT REAL DEVICES ON THE HOST COMPUTER TO SIMH?
2.10 MY WINDOWS HOST CAN'T COMMUNICATE WITH THE PDP-11 OR VAX OVER ETHERNET; WHY?
2.11 HOW CAN I USE MY WIRELESS ETHERNET CARD WITH SIMH?

3 WRITING AND DEBUGGING NEW CODE

3.1 WHAT RESOURCES ARE AVAILABLE FOR WRITING NEW SIMULATORS?
3.2 WHAT DEBUGGING FACILITIES ARE AVAILABLE IN SIMH?
3.3 WHEN DO I NEED TO USE THE HOST DEBUGGER FOR DEBUGGING A SIMULATOR?
3.4 WHAT IS THE RELEASE PROCESS FOR SIMH?

4 VAX

4.1 WHERE CAN I GET SOFTWARE AND HOBBYIST LICENSES FOR THE VAX?
4.2 HOW DO I INSTALL VMS?
4.3 HOW DO I INSTALL NETBSD?
4.4 HOW DO I INSTALL ULTRIX?
4.5 WHAT'S THE CPU SERIAL NUMBER FOR MY HOBBYIST LICENSE PAK?

5 PDP-11

1 General Questions

1.1 What is SIMH?

SIMH is the Computer History Simulation system. It consists of simulators for approximately 20 different computers, all written around a common user interface package and set of supporting libraries. SIMH can be used to simulate any computer system for which sufficient detail is available, but the focus to date has been on simulating computer systems of historic interest.

1.2 Why was SIMH written?

Significant portions of the computing past are being irretrievably lost, as old systems are scrapped, documentation and software is thrown out, media become obsolete or unreadable, and inventors and pioneers die. SIMH was written as a vehicle to allow the computing past to be made accessible to a wider audience, for recreational and educational purposes. SIMH preserves historic computers as portable software that can be run on any modern system. SIMH also preserves representative software packages for these systems. With SIMH, anyone with a desktop computer can call up and run significant samples from the computing past, at any time.

1.3 What is the history of SIMH?

The SIMH project started in 1993, at the suggestion of Larry Stewart of DEC. Its immediate purpose was to preserve the fading hardware and software record of early minicomputers. Since then, the project has been expanded to include other important systems, spanning the history of computing from the late 50's to the late 80's.

SIMH's core design is based on an earlier simulation system called MIMIC. MIMIC was written in the late 1960's at Applied Data Research, by Mike McCarthy, Len Feshkens, and Bob Supnik. MIMIC was a mini-computer simulator that ran on the PDP-10. Its purpose was to facilitate the development and debugging of real-time embedded systems by using the PDP-10 timesharing environment for program development, instead of the limited facilities of the native minicomputer environments. Ironically, given SIMH's mission to preserve the computing record, all machine-readable copies of MIMIC have been lost.

1.4 Who writes and maintains SIMH?

Many people have contributed, and continue to contribute, to SIMH. The full list of contributors can be found on the SIMH web site. Bob Supnik coordinates SIMH development.

1.5 How is SIMH licensed?

SIMH is licensed under a modified X-Windows license. This license allows more or less unrestricted use of the sources and binaries. The license is included with the documentation and is also included in every source module. The software packages are available under various terms and conditions; see the documentation included with each software package.

1.6 How is SIMH distributed?

SIMH is distributed in source form from the SIMH web site, in the form of a Zip archive. For Windows users, pre-compiled binaries are also available.

1.7 Which computer systems does SIMH simulate?

SIMH simulates the following computer systems:

```
Manufacturer                   Model

Digital Equipment Corporation  PDP-1, PDP-4, PDP-7, PDP-8, PDP-9,
                               PDP-10, PDP-11, PDP-15, VAX-11/780,
                               MicroVAX 3900
Data General Corporation       Nova, Eclipse
IBM Corporation                1130, 1401, 1620, System 3, 7094
GRI Corporation                GRI-909
Honeywell Corporation          H316/516
Hewlett Packard Corporation    HP2116, HP2100, HP21MX
Interdata Corporation          16b systems, 7/32, 8/32
Scientific Data Systems        SDS-940
MITS                           Altair 8080, Altair Z80
Royal-Mcbee                    LGP-30, LGP-21
```

The documentation contains more details on supported models and peripherals.

1.8 Which host systems does SIMH run on?

```
Host System      Compiler         comments

OpenVMS/VAX      DEC C            no 64b support; no Ethernet support
OpenVMS/Alpha    DEC C            Ethernet support provided in pcap-vms

Windows 9x or    Mingw/gcc or     requires WinPcap for Ethernet support
Windows 2000 or  Visual C++ or
Windows XP       Borland C++

Mac OS/X                          requires libpcap for Ethernet support

Linux            gcc              requires libpcap for Ethernet support
Tru64 UNIX       DEC C            no Ethernet support
AIX                              no Ethernet support
Solaris                          requires libpcap for Ethernet support
HP/UX                            no Ethernet support
NetBSD           gcc              requires libpcap for Ethernet support
OpenBSD          gcc              requires libpcap for Ethernet support
FreeBSD          gcc              requires libpcap for Ethernet support

OS/2             EMX              no Ethernet support
```

1.9 What software packages are available to run on SIMH?

The list of available software packages can be found on the SIMH web site.

1.10 Where can I get more information on SIMH?

The SIMH web site is http://simh.trailing-edge.com.

2 Operational Questions

2.1 How do I install SIMH on Windows?

The simplest way is to download the pre-compiled binaries. Unzip these into the directory where you want to run SIMH. You can then run whichever binary that you want.

2.2 How do I install SIMH with Ethernet support on Windows?

Separate pre-compiled binaries contain Ethernet support. Before running these binaries, you must download and install the WinPCAP AutoInstaller from

```
http://www.winpcap.org
```

This creates a network packet driver in Windows for SIMH to attach to.

To use network support, you must either be an administrator on the Windows machine (implied in Windows 9X), or you must set the windows packet driver to autostart when the system boots; see the WinPCAP FAQ page for detailed information on how to do this.

2.3 How do I install SIMH on Unix?

Ethernet support is available only on Linux, NetBSD, OpenBSD, and FreeBSD.

- Unzip the archive of sources to a new directory. You must specify the   -a switch to unzip for proper conversion of Windows cr-lf sequences to UNIX newline sequences.

- If your system supports gmake, you can compile the simulators with the command:

```
% gmake all
```

- If you want Ethernet support in the PDP-11 and VAX, you should compile   the simulators with the command:

```
% gmake USE_NETWORK=1 all
```

2.4 How do I install SIMH on VMS?

Download the SIMH source kit, and UNZIP it using the /TEXT=AUTO qualifier to the directory that you want SIMH to reside in. Use MMK or MMS and the descrip.mms file to build the binaries.

On a VAX use:

```
$ MMx
```

On a Alpha use:

```
$ MMx/MACRO=("__ALPHA__=1")              !Without ethernet support
$ MMx/MACRO=("__ALPHA__=1","__PCAP__=1") !With    ethernet support

UNZIP can be found on the VMS freeware CDs, or from www.info-zip.org
MMK   can be found on the VMS freeware CDs, or from www.madgoat.com
MMS   can be licensed from HP/Compaq/Digital.
```

Note that the PDP-10 emulator cannot be built and used on VAX/VMS, because the DEC C compiler for VAX/VMS does not support 64-bit integers. DEC C on on Alpha VMS has the required 64-bit capability to build and run all of the emulators. Ethernet support is only available on Alpha VMS 7.3-1 and above.

2.5 How do I transcribe a real CD for use with SIMH?

- On UNIX, you can copy a CD to an ISO file with the dd command:

```
% dd /if=/dev/raw_cd_device /out=/path/cdimage.iso
```

- Linux, and many Unix variants, support direct access to the CD ROM from SIMH:

```
sim> set rq1 cdrom
sim> att rq1 /dev/cdrom_drive
```

- On Windows, there are quite a few products that can do this. The two most common products are detailed below. Make sure to disable any antivirus software before proceeding. Antivirus software tends to interfere with the smooth flow of data from the CD and will occasionally transform the data in strange and unexpected ways to 'protect' you. You may also need to limit the read speed. Some burnt CD-Rs do not read correctly at the highest rate of speed, depending on the accuracy of the burner; pressed CD-ROMs should not have this problem.

   1) Roxio
     A) EZ-CD Creator 5.x
       Go to the the Disc menu and select Disc Info (there will be a delay).
       Select the track shown, then click the Read Track button.
       Enter the Save file name, then OK.
     B) Easy Media Creator 7.x
       Go to Creator Classic
       Select Other Tasks | Disc and Device Utility
       Drill down on the device until you find the data track, then select it
       Click the 'Read Track...' button
       Enter the save file name, then OK.

   2) Nero 5.5
     Select Recorder|Save Track
     Select the track, set the output filename
     Click GO

2.6 How do I transcribe other archival media for use with SIMH?

You must have access to a real system that can read the media to be transcribed (e.g., a system with a working DECtape drive to read a DECtape). Most systems have utilities to copy raw data to a disk file; that file can then be transferred over the console serial line to a system with an Internet link. Utility programs are available to convert raw data streams to SIMH format.

2.7 How can I get text files in and out of SIMH?

Since SIMH supports the universal serial interface using TELNET, text can be transferred using one of the serial line transfer protocols (X/Y/Zmodem, Kermit) or using standard cut and paste techniques, if the host's TELNET program supports it.

To use the TELNET feature, connect to the SIMH machine using TELNET, and set the target environment into a 'receive' mode. This is usually something like running a text editor. Then tell the TELNET program to 'send', 'transfer', or 'paste' the text that you want sent into the SIMH system.

To get text out of the system, have the TELNET program either log the output, or if the TELNET program supports a backscroll region you can use that. Tell the SIMH system to 'type' or 'cat' the text file, sending the output to the TELNET device, where you can edit it into a text file.

Many TELNET programs also support transferring large files via X/Y/ZModem or Kermit, which you can use as long as the SIMH system has the appropriate matching program.

C-Kermit from Columbia University (http://www.columbia.edu/kermit) is probably the most universal way to transfer files in and out of SIMH systems.

If the SIMH system supports Ethernet connectivity (PDP-11, VAX), you can also use the various network copy programs (FTP, DECNET) to transfer files.

Finally, you can "print" text files to the simulated line printer. Printer output is automatically formatted as an ASCII text file.

2.8 How can I get binary files in and out of SIMH?

Since SIMH supports the universal serial interface using TELNET, binary files can be transferred using one of the serial line transfer protocols (X/Y/ZModem, Kermit) or by converting the binary to a text-encoded file (HEXify, UUENCODE, VMShare, etc.) and transferred in text mode (see section 2.7).

Many TELNET programs also support transferring large files via X/Y/ZModem or Kermit, which you can use as long as the SIMH system has the appropriate matching program.

C-Kermit from Columbia University (http://www.columbia.edu/kermit) is probably the most universal way to transfer files in and out of SIMH systems.

If the SIMH system supports Ethernet connectivity (PDP-11, VAX), you can also use the various network copy programs (FTP, DECNET) to transfer files.

2.9 Can I connect real devices on the host computer to SIMH?

At the moment, Ethernet is the only supported real device.

2.10 My Windows host can't communicate with the PDP-11 or VAX over Ethernet; why?

Due to the inherent limitations of WinPCAP, the SIMH system **CANNOT** communicate with the host on the primary interface. To establish communications between SIMH and a PC host, add a second Ethernet controller, attach both controllers to the same hub, and attach SIMH to the second controller. The host and SIMH will now be able to communicate across the physical network connection.

Attaching the Ethernet device to a TAP-Win32 pseudo-controller can also solve this problem, since you would normally bridge the pseudo-controller to the main ethernet controller; the simulated system and the host will communicate over the bridge. TAP-Win32 is a component of OpenVPN software: http://openvpn.net.

2.11 How can I use my wireless Ethernet card with SIMH?

Wireless Ethernet is something of a misnomer - it "works like" Ethernet. Wireless cards behave differently than real Ethernet cards in promiscuous mode but can be successfully used with existing SIMH code.

The caveat is that the simulated machine cannot run any software which changes the simulated MAC address, or the network connection will stop working. For example, DECNET Phase IV (or Phase V in compatibility mode) tries to change the MAC of the network card to AA-00-04-xx-xx-xx. Nor can you preset the wireless MAC address to the anticipated target DECNET address using something like SMAC to get DECNET to work - DECNET will see the MAC already preset to the required DECNET address and generate an invalid media (duplicate address) fault.

Otherwise, TCP/IP, LAT, VMS Clustering, and DECNET Phase V in non-compatibility mode work fine.

To get wireless cards to work with SIMH, set the simulated MAC to be the same as the MAC of the wireless card. An example:

```
c:\> IPCONFIG/ALL

Windows 2000 IP Configuration

   Host Name . . . . . . . . . . . . : LLOH3-EXP29189
   Primary DNS Suffix  . . . . . . . : ad.tasc.com
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : ad.tasc.com

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : D-Link DWL-650+ Wireless Cardbus
   Physical Address. . . . . . . . . : 00-80-C8-08-CE-DB   <-- MAC address
   DHCP Enabled. . . . . . . . . . . : No
   IP Address. . . . . . . . . . . . : 192.168.0.5
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DNS Servers . . . . . . . . . . . :
   Primary WINS Server . . . . . . . : 132.228.188.100
   Secondary WINS Server . . . . . . : 132.228.196.98

c:\> VAX
VAX simulator V3.2-1
sim> DO VAX_CONFIG.DO              <-- setup VAX as normal
sim> SET XQ MAC=00-80-C8-08-CE-DB  <-- set XQ MAC to wireless MAC address
sim> B CPU                         <-- and continue...
```

3   Writing and Debugging New Code

3.1 What resources are available for writing new simulators?

The SIMH web site contains documentation on the internals of SIMH, as well as specific help for writing new peripherals for several of the popular simulators.

3.2 What debugging facilities are available in SIMH?

Most simulators provide the following debugging capabilities:

- Symbolic assembly and disassembly of memory contents.
- Numeric examination and modification of the data store of any simulated device.
- Numeric search on both memory and device data.
- Visibility to simulator internal structures, such as the event queue.
- An unlimited number of instruction breakpoints.
- Proceed counts on breakpoints.
- Automatic execution of simulator commands on a breakpoint.
- Stepped execution (from single step to 'n' steps).
- A PC change queue, usually 64 instructions deep.

Specific simulators may provide additional features, such as an instruction history buffer, CPU and/or device logging, and breakpoints on memory reads and writes.

3.3 When do I need to use the host debugger for debugging a simulator?

While a simulator is being debugged, its execution of instructions or debugging support code may be unreliable. During this process, the programmer may need to use the host debugger to stop in the middle of an instruction execution, or to trap an error condition. Host debugger breakpoints should be invisible to the simulator; with the exception of clock calibration, all simulator events are driven off the event queue rather than real-world events.

If the programmer needs to force a simulator stop from the host debugger, most simulators provide an "address stop" global variable. Setting this variable to 1 will cause the simulator to stop after completing the current instruction.

3.4 What is the release process for SIMH?

SIMH is released whenever a significant number of new features, or important bug fixes, has accumulated. This has averaged every 8-16 weeks. The major version number only changes when there is a major restructuring of SIMH's internal structures. The minor version number is changed when the format of the save/restore file must be updated.

4   VAX

4.1 Where can I get software and hobbyist licenses for the VAX?

HP (formerly Compaq formerly DEC) provides licenses to OpenVMS for hobbyist use. A description of the hobbyist license program can be found on http://www.openvmshobbyist.com.

4.2 How do I install VMS?

To install VMS, you will need a distribution CD ROM. Any version after VMS 5.5-2 should run on the MicroVAX 3900 simulator.

- Transcribe the distribution CD ROM to an ISO-format CD image file. (See question 2.5 for information on how to do this.)
- Set drive RQ1 to be a CD ROM.
- Attach the CD ROM image file to simulated drive RQ1.
- Set drive RQ0 to be the type of disk you want. Be sure that the disk is large enough to hold VMS.
- Attach a blank disk image file to simulated drive RQ0.
- Boot the CPU.
- When the self-test code completes, boot the CD ROM.
- Use standalone backup to restore the CD ROM contents to the simulated disk.

```
sim> set rq0 rd54
sim> set rq1 cdrom
sim> att rq0 new_vms.dsk
sim> att rq1 cd_rom_image.iso
sim> boot cpu
:
>>> boot rq1

$ (prompt from standalone backup)
```

A writeup on the procedure can be found on the VMS hobbyist site.

4.3 How do I install NetBSD?

Directions for installing NetBSD on the NetBSD web site, at http://www.netbsd.org/Ports/vax/emulator-howto.html.

4.4 How do I install Ultrix?

Ultrix is not presently licensed for hobbyist use. If you have a valid license for Ultrix, and distribution tapes for a version that supports the MicroVAX 3900 series (V4 or later), then you should be able to install Ultrix on the simulator.

- Transcribe the distribution tapes to SIMH-format tape image files. (See question 2.6 for information on how to do this.)
- Mount the installation tape image on simulated drive TQ0.
- Set drive RQ0 to be the type of disk you want. Be sure that the disk is large enough to hold Ultrix.
- Mount a blank disk image file on simulated drive RQ0.
- Boot the CPU.
- When the self-test code completes, boot the installation tape.
- The installation tape will guide you through the installation of Ultrix.

```
sim> set rq0 rd54
sim> att rq0 new_vms.dsk
sim> att tq0 ultrix_install.tap
sim> boot cpu
    :
>>> boot mua0

(Ultrix installation dialog)
```

4.5 What's the CPU serial number for my hobbyist license PAK?

On a MicroVAX 3900, the CPU serial number is not readable and can be an arbitrary value. 12345 will work fine.

4.6 How do I change the simulator from a VAXserver 3900 to a MicroVAX 3900?

The system type is controlled by a "magic byte" in the CPU's boot ROM. By default, the system type is a VAXserver 3900. To change the type to a MicroVAX 3900, patch the boot ROM as follows:

```
sim> set ptr ena
sim> att ptr ka655x.bin
sim> ie ptr 4
4:    2     1
sim> det ptr
```

and reboot the simulated VAX.

4.7 Is there an example of the simulator running VMS?

This example assumes you are trying to emulate a MicroVAX 3900 with 64MB of memory, with a single 1GB disk drive, a CDROM, and an Ethernet controller.

The host OS is Windows NT/2000/XP, and you have previously dumped the contents of the VMS Hobbyist CD to a disk file as detailed in 2.5, and have loaded WinPCAP on the system for Ethernet support. Other host OS's will look similar but will have different file name syntax.

```
c:\simh> vax                           ; run VAX emulator
sim> set cpu 64m                       ; set memory size to 64MB
sim> load -r vax\ka655x.bin            ; load the MicroVAX 3900 console ROM
sim> attach NVR vax\ka655.nvr          ; create/load a Non-Volatile RAM file
sim> set LPT disable                   ; disable devices we don't want/need
sim> set TQ disable                    ;    "
sim> set rq0 ra90                      ; set disk 0 to 1GB (RA90 size)
sim> attach rq0 vax\vaxsys.dsk         ; create/use disk file
sim> set rq1 rrd40                     ; set disk 1 as a cdrom
sim> attach -r rq1 vax\hobbyist.dsk    ; attach cdrom dump file as read-only
sim> set rq2 offline                   ; turn off disk rq2
sim> set rq3 offline                   ; turn off disk rq3
sim> attach xq eth0                    ; attach to host ethernet controller
sim> b cpu                             ; start (boot) VAX console

KA655-B V5.3, VMB 2.7
 1) Dansk                              ; will not appear if the controlling
    ..                                 ; keyboard doesn't support multi-
15) Svenska                            ; national characters!
 (1..15): 5
Performing normal system tests.
40..39..38..37..36..35..34..33..32..31..30..29..28..27..26..25..
24..23..22..21..20..19..18..17..16..15..14..13..12..11..10..9..
8..7..6..5..4..3..
Tests completed.
>>> show device                       ; tell console to show all devices
UQSSP Disk Controller 0 (772150)
-DUA0 (RA90)
-DUA1 (RRD40)
```

```
   Ethernet Adapter 0 (774440)
   -XQA0 (08-00-2B-AA-BB-CC)
   >>> b dua1                              ; tell console to boot cdrom
   (BOOT/R5:1 DUA1)

   2..1..0
```

4.8 How can I import files to a simulated VMS environment?

- Use a CD burner program, like Easy CD Creator or Nero, to create an ISO 9660 CD image containing the files you want to import. Note that file names are limited to DOS '8.3' conventions.
- Attach the simulated CD image to a simulated CD drive.
- Mount the simulated CD as an ISO 9660 file system under VMS.
- Copy the files you need from the simulated CD to the simulated disk.

(Thanks to Tim Stark for this suggestion.)

4.9 How can I export files from a simulated VMS environment?

- Utility ODS2 (available on the Web) can read an ODS-2 disk image and copy files from that image to the host file system.
- Text files can be printed to the simulated line printer, as described above.

5 PDP-11

5.1 When installing RSTS/E from simulated magnetic tape, the installation process hangs with no error message; why?

RSTS/E installation from magnetic tape requires that the tape be write locked.