Running VAX/VMS Under Linux Using SIMH

Phillip Wherry
psw-at-wherry-dot-com
Revision 1.5
22 January 2004

Introduction

One of the first minicomputer systems that I worked with extensively was a Digital Equipment Corporation's VAX machine. The particular machine that I used, a VAX 11/730, was at the time (1983) an entry-level minicomputer that cost just under $40,000 (that's just the CPU!) and delivered real-world performance of about 0.15 MIPS. As a note of comparison, the flagship VAX of the time, the VAX 11/780, delivered about 0.5 MIPS for a little over $180,000. (VAX system benchmarking number are frequently stated in VAX Units of Performance [VUPS], which is a measure of the system's speed relative to the original 11/780; the 11/730 was about 0.33 VUPS compared to the 11/780's 1.0 VUPS. Thanks to Brian Foley for providing this clarification.)

At the time I first began using it, the 11/730 was running the VMS 3.3 operating system. VMS ("Virtual Memory System") was a multi-user timesharing system of extraordinary sophistication for the time. Dave Cutler, the operating system's designer, went on to Microsoft some years later to design Windows NT (and subsequent NT-derived operating systems). The first version of VMS, version 1.0, had shipped in 1978, so the operating system wasn't new at the time. The operating system was ported to the Alpha architecture in 1992. Compaq, a computer company that didn't exist until about 25 years after Digital was founded, purchased DEC in 1998. Compaq and HP merged in 2002. Though VAX hardware isn't sold anymore, OpenVMS is still supported. Amazingly enough, there's even development work going on: HP has announced that OpenVMS for Intel's 64-bit Itanium architecture will be commercially available in 2004!

In any case, I'm not the only person with an interest in this historic hardware. A simulator called SIMH, maintained by Bob Supnik (who had a pivotal role in the original development of a number of Digital's machines, serving at one time as technical director for the Alpha and VAX groups at DEC), emulates a wide range of old hardware. The VAX architecture is one of the many represented.

HP makes OpenVMS available for non-commercial use through its Hobbyist License program. In order to participate in this program, you must:

- Join Encompass US (or the appropriate user group for your country). There's a free membership available that will permit you to participate in the OpenVMS hobbyist license program, and some paid membership categories with additional benefits. It's interesting to note that this is a long-lived user group; the original Digital Equipment Corporation Users Society (DECUS) was formed in 1961! You'll receive a membership number by email (a process that takes several days based on my experience).
- Get a media kit. I ordered the OpenVMS VAX Hobbyist Kit 3.0 from Montagar Software and had it about a week later. This kit includes OpenVMS 7.3, the most recent (as of this writing) release for the

VAX architecture.

- Get software licenses for OpenVMS and the layered products using the license registration page. You'll be asked for a hardware serial number at one point in the process; since this really isn't applicable to SIMH, just enter the word NONE. You'll receive license registration information by email. Note that you have to register separately for the base operating system and the "layered products" that run on top of it. The license keys expire after a year, but you can simply request a new license at the appropriate time to extend the life of the system.

Once you have these items, you can get OpenVMS running atop a Linux platform fairly easily. Note that a dedicated platform is recommended for serious emulation, as the design of this (and most other) emulators ensures that the CPU will be busy essentially 100% of the time. Windows users should also be able to use most of these procedures, making appropriate file naming modifications here and there (you'll also need to install the winpcap drivers to get Ethernet working). Here's how I did it on a Linux box running a stock Debian 3.0 release.

But First, a Disclaimer, a Plea for Help, and a Few Acknowledgements...

I've been successful in making these procedures work in my environment. I hope you're able to duplicate my success, but make no guarantees.

I'd note that it's been over fifteen years since I used a VAX on a daily basis. There are plenty of things that I've forgotten in that time, and most likely even more things that I never knew. If anyone spots errors in this documentation or has suggestions for better ways to do things, I'd love to hear from you.

Since writing this document, I've gotten helpful feedback from a number of readers; for the most part, the changes they suggested are acknowledged within the text. I'd particularly like to thank Dave Hittner, of the SIMH Ethernet project, for an extensive and helpful set of suggestions that have been incorporated at numerous points within this document. While it wasn't practical to acknowledge each change individually, please know that his efforts have significantly improved the quality of this documentation.

Download and Build the Emulator

First, download a copy of SIMH. The version I used for this documentation is version 2.10-4. Find a location for this software (I used /home/software) and make a directory for it.

```
emulator:/home/software$ mkdir simh
emulator:/home/software$ cd simh
```

Next, unpack the software. Be careful; unlike most Unix tarballs, this .zip file expands in the context of the *current* directory rather than creating a named subdirectory underneath. So, be sure you're currently in the location where the software should be unpacked; otherwise, if you're not careful, you can wind up with a mess.

```
emulator:/home/software/simh$ unzip -a ../simhv210-4.zip
Archive: ../simhv210-4.zip
inflating: 0readme_210.txt
inflating: 0readme_ethernet.txt
.
.  (lengthy output omitted)
.
inflating: VAX/vax_sys.c
inflating: VAX/vax_sysdev.c
```

Now, build the VAX portion of the emulator. The build process puts things in a directory called **BIN**, and will fail if it doesn't already exist. In order to use Ethernet support, you also need to include **USE_NETWORK** in the build command, as follows:

```
emulator:/home/software/simh$ mkdir BIN
emulator:/home/software/simh$ make USE_NETWORK=1 BIN/vax
gcc -O2 -lm -I . VAX/vax_cpu1.c VAX/vax_cpu.c VAX/vax_fpa.c VAX/vax_io.c VAX/vax_mmu.c VAX/vax_stddev.c
VAX/vax_sys.c VAX/vax_sysdev.c PDP11/pdp11_rl.c PDP11/pdp11_rq.c PDP11/pdp11_ts.c PDP11/pdp11_dz.c
PDP11/pdp11_lp.c PDP11/pdp11_tq.c PDP11/pdp11_pt.c PDP11/pdp11_xq.c scp.c scp_tty.c sim_sock.c sim_tmxr.c
sim_ether.c sim_tape.c -I VAX/ -I PDP11/ -DUSE_INT64 -DUSE_NETWORK -lpcap -o BIN/vax
```

At this point, you've got the basic VAX emulator constructed. The application and its data need a home; I chose **/usr/local/vax/bin** and **/usr/local/vax/data**.

```
emulator:/home/software/simh# mkdir /usr/local/vax
emulator:/home/software/simh# mkdir /usr/local/vax/bin
emulator:/home/software/simh# mkdir /usr/local/vax/data
```

At this point, copy the VAX emulator binary into its new location:

```
emulator:/home/software/simh# cp BIN/vax /usr/local/vax/bin
```

The processor boot ROM (from the KA655 processor found in a MicroVAX 3900) is included with the SIMH package. Copy this into the data directory.

```
emulator:/home/software/simh# cp VAX/ka655.bin /usr/local/vax/data
```

In addition to the boot ROM, you'll need a couple of other files to get started. The file **cd.iso** contains an image of the OpenVMS Hobbyist CD-ROM. Create this using **dd** or your favorite ISO image extraction application. If you're using the **dd** command under Linux, the following command should create the required image (presuming your CD-ROM is named **/dev/cdrom**; substitute a different name if needed):

```
emulator:/home/software/simh-data# dd if=/dev/cdrom of=cd.iso
```

In addition, you'll need an initialization file to set up the emulator unless you like typing the commands yourself. The commands below assume that these two files reside in a directory called **/home/software/simh-data**.

```
emulator:/home/software/simh-data# cp cd.iso /usr/local/vax/data
emulator:/home/software/simh-data# cp vax.ini/usr/local/vax/data
```

The contents of vax.ini should initially be as follows. Edit file paths as appropriate, of course.

```
;
; Load CPU microcode
load -r /usr/local/vax/data/ka655.bin
;
; Attach non-volatile RAM to a file
attach nvr /usr/local/vax/data/nvram.bin
;
; This virtual machine has 64M memory
set cpu 64m
;
; Define disk drive types. RA92 is largest-supported VAX drive.
set rq0 ra92
set rq1 ra92
set rq2 ra92
set rq3 cdrom
;
; Attach defined drives to local files
attach rq0 /usr/local/vax/data/d0.dsk
attach rq1 /usr/local/vax/data/d1.dsk
attach rq2 /usr/local/vax/data/d2.dsk
;
; Attach the CD-ROM to its file (read-only)
```

```
attach -r rq3 /usr/local/vax/data/cd.iso
;
; Disable unused devices. It's also possible to disable individual devices,
; using a construction like "set rq2 disable" if desired.
;
set rl disable
set ts disable
;
; Attach Ethernet to a network interface
set xq mac=08-00-2B-AA-BB-CC
attach xq eth0
;
; Now start the emulator
boot cpu
```

The emulator looks in its startup directory for the initialization file by default. I don't like having binaries and data in the same place, so I added a symbolic link:

```
emulator# ln -s /usr/local/vax/data/vax.ini /usr/local/vax/bin
```

Install the OpenVMS Operating System

At this point, the emulator can be started! You do need to do this as superuser (root), since the Ethernet emulation depends on it. More on this detail later.

```
emulator# /usr/local/vax/bin/vax
VAX simulator V2.10-4
NVR: creating new file
NVR: buffering file in memory
RQ: creating new file
RQ: creating new file
RQ: creating new file
RQ: unit is read only
KA655-B V5.3, VMB 2.7
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..09..
08..07..06..05..04..03..
Tests completed.
```

At this point, boot from the virtual CD-ROM, entering the date and time when asked.

```
>>>boot dua3
(BOOT/R5:0 DUA3)
2..
-DUA3
1..0..
%SYSBOOT-I-SYSBOOT Mapping the SYSDUMP.DMP on the System Disk
%SYSBOOT-W-SYSBOOT Can not map SYSDUMP.DMP on the System Disk
%SYSBOOT-W-SYSBOOT Can not map PAGEFILE.SYS on the System Disk
OpenVMS (TM) VAX Version X7G7 Major version id = 1 Minor version id = 0
%WBM-I-WBMINFO Write Bitmap has successfully completed initialization.
PLEASE ENTER DATE AND TIME (DD-MMM-YYYY HH:MM) 19-APR-2003 15:46
Configuring devices . . .

Now configuring HSC, RF, and MSCP-served devices . . .

Please check the names of the devices which have been configured,
to make sure that ALL remote devices which you intend to use have
been configured.

If any device does not show up, please take action now to make it
available.

Available device DUA0: device type RA92
Available device DUA1: device type RA92
Available device DUA2: device type RA92
Available device DUA3: device type RRD40

Enter "YES" when all needed devices are available: y
%BACKUP-I-IDENT, Stand-alone BACKUP T7.2; the date is 19-APR-2003 15:48:04.20
```

At this point, you're running a small "standalone" operating system that's used for backing up and restoring data. Use the command that follows to restore the VMS "save set" from CD-ROM to the first hard disk.

```
$ backup dua3:vms073.b/save_set dua0:
%BACKUP-I-PROCDONE, operation completed. Processing finished at 19-APR-2003 15:51:51.01
If you do not want to perform another standalone BACKUP operation,
use the console to halt the system.

If you do want to perform another standalone BACKUP operation,
ensure the standalone application volume is online and ready.
Enter "YES" to continue:
```

At this point, VMS has been copied to your virtual hard disk. Since you have no more save sets to restore at this time, stop the simulation by pressing control-E. Then reboot the virtual machine to continue the installation process.

```
Simulation stopped, PC: 839ABD40 (MOVL (R5),R4)
sim> boot cpu
```

Once again, the virtual VAX boots:

```
KA655-B V5.3, VMB 2.7
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..09..
08..07..06..05..04..03..
Tests completed.
```

At this point, we'll be booting from the **DUA0** device in the future, so we'll set up the default boot device. If this isn't done, the system attempts to boot from device XQA0, which isn't the behavior we're after. The **boot** command that follows tells the system to boot from the default device we just set. Installation proceeds from this point; my answers to the installation questions are highlighted below.

```
>>>set boot dua0
>>>boot
(BOOT/R5:0 DUA0)
2..
-DUA0
1..0..

%SYSBOOT-I-SYSBOOT Mapping the SYSDUMP.DMP on the System Disk
%SYSBOOT-W-SYSBOOT Can not map SYSDUMP.DMP on the System Disk
%SYSBOOT-I-SYSBOOT Mapping PAGEFILE.SYS on the System Disk
%SYSBOOT-I-SYSBOOT SAVEDUMP parameter not set to protect the PAGEFILE.SYS
OpenVMS (TM) VAX Version BI73-7G7 Major version id = 1 Minor version id = 0
%WBM-I-WBMINFO Write Bitmap has successfully completed initialization.

OpenVMS VAX V7.3 Installation Procedure

Model: VAXserver 3900 Series
System device: RA92 - _DUA0:
Free Blocks: 2854566
CPU type: 10-01

* Please enter the date and time (DD-MMM-YYYY HH:MM) 19-apr-2003 15:58
******************************************************
%SYSTEM-W-TZGMT, your local timezone has defaulted to GMT
%SYSTEM-I-SETTZ, to set your local timezone use:

$ @SYS$MANAGER:UTC$TIME_SETUP.COM

******************************************************
On MIN or UPGRADE system startup - CLUE is not run.
%%%%%%%%%% OPCOM 19-APR-2003 15:58:21.21 %%%%%%%%%%
Operator _OPA0: has been enabled, username SYSTEM

%%%%%%%%%% OPCOM 19-APR-2003 15:58:21.26 %%%%%%%%%%
Operator status for operator _OPA0:
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, CLUSTER, SECURITY,
LICENSE, OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10,
OPER11, OPER12
```

```
%%%%%%%%%%  OPCOM 19-APR-2003 15:58:21.56 %%%%%%%%%%
Logfile has been initialized by operator _OPA0:
Logfile is SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;1

%%%%%%%%%%  OPCOM 19-APR-2003 15:58:21.59 %%%%%%%%%%
Operator status for operator SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;1
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, CLUSTER, SECURITY,
LICENSE, OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10,
OPER11, OPER12

%SYSTEM-I-BOOTUPGRADE, security auditing disabled
%%%%%%%%%%  OPCOM 19-APR-2003 15:58:25.31 %%%%%%%%%%
Message from user JOB_CONTROL
%JBC-E-OPENERR, error opening SYS$COMMON:[SYSEXE]QMAN$MASTER.DAT;

%%%%%%%%%%  OPCOM 19-APR-2003 15:58:25.34 %%%%%%%%%%
Message from user JOB_CONTROL
-RMS-E-FNF, file not found

%LICENSE-F-EMTLDB, license database contains no license records
%SYSTEM-I-BOOTUPGRADE, security server not started
%%%%%%%%%%  OPCOM 19-APR-2003 15:58:34.07 %%%%%%%%%%
Message from user SYSTEM
%LICENSE-E-NOAUTH, DEC VAX-VMS use is not authorized on this node
-LICENSE-F-NOLICENSE, no license is active for this software product
-LICENSE-I-SYSMGR, please see your system manager

%LICENSE-E-NOAUTH, DEC VAX-VMS use is not authorized on this node
-LICENSE-F-NOLICENSE, no license is active for this software product
-LICENSE-I-SYSMGR, please see your system manager
Startup processing continuing...

%SET-I-INTSET, login interactive limit = 1, current interactive value = 0
%SET-I-INTSET, login interactive limit = 0, current interactive value = 0

If this system disk is to be used in an OpenVMS Cluster with multiple
system disks, then each system disk must have a unique volume label.
Any nodes having system disks with duplicate volume labels will fail
to boot into the cluster.

You can indicate a volume label of 1 to 12 characters in length. If you
want to use the default name of OVMSVAXSYS, press RETURN in response
to the next question.

* Enter the volume label for this system disk [OVMSVAXSYS]: <cr>

* Enter name of drive holding the OpenVMS distribution media: DUA3
* Is the OpenVMS media ready to be mounted? [N] y
%MOUNT-I-MOUNTED, VAXVMS073 mounted on _DUA3:

Select optional software you want to install. You can install one
or more of the following OpenVMS or DECwindows components:

o OpenVMS library - 52200 blocks
o OpenVMS optional - 19000 blocks
o OpenVMS Help Message - 10400 blocks
o OpenVMS Management Station - 20000 blocks
o DECwindows base support - 4400 blocks
o DECwindows workstation support - 23800 blocks
- 75 dots per inch video fonts - (included)
- 100 dots per inch video fonts - 6200 blocks
o DECnet-Plus networking - 80000 blocks
o DECnet Phase IV networking - 800 blocks

Space remaining on system disk: 2854377 blocks

* Do you want to install the OpenVMS library files? (Y/N) y

Space remaining on system disk: 2802177 blocks

* Do you want to install the OpenVMS optional files? (Y/N) y

Space remaining on system disk: 2783177 blocks

The Help Message utility (MSGHLP) provides online explanations
and user actions for OpenVMS messages in place of the hardcopy
OpenVMS System Messages and Recovery Procedures Reference Manual,
which is now separately orderable.

The MSGHLP database file, MSGHLP$LIBRARY.MSGHLP$DATA,
consumes approximately 10400 blocks and will be
```

```
placed by default on your system disk in SYS$COMMON:[SYSHLP]
unless you specify an alternate device when prompted.

* Do you want to install the MSGHLP database? (Y/N) y

You can install this database on your system disk in SYS$COMMON:[SYSHLP]
or on an alternate device. If you specify an alternate device, but no
directory, MSGHLP$LIBRARY.MSGHLP$DATA is placed in [HELP_MESSAGE]. When
prompted, take the default of the system disk or specify an alternate
device using this format:

device:[directory]

* Where do you want to install the MSGHLP database?
[SYS$COMMON:[SYSHLP]] <cr>

Space remaining on system disk: 2772777 blocks

The OpenVMS Management Station is a client-server application that
provides OpenVMS system management capabilities through a client
application on a personal computer (PC) running Microsoft Windows.

The server application runs on OpenVMS systems and is automatically
installed as part of the OpenVMS operating system.

This option provides the files used to install the PC client software.
If you want to use the OpenVMS Management Station, you must install
these optional files on at least one OpenVMS system and then use one or
both of them to install the PC client on one or more PCs. There are two
files: TNT030_I.EXE for Intel systems (Windows 95 and Windows NT), and
TNT030_A.EXE for Alpha Windows NT systems.

The OpenVMS Management Station optional files consume approximately 20000
blocks and will be placed on your system disk in SYS$COMMON:[TNT.CLIENT].

* Do you want to install the optional OpenVMS Management Station files? (Y/N) y

Space remaining on system disk: 2752777 blocks

You can select DECwindows now, or you can use the DECW$TAILOR utility
to provide or remove DECwindows support after the installation.

Some media, TK50s in particular, can be very slow when tailoring on files.
You might want to select DECwindows now and tailor off unwanted files later.

NOTE: This kit does NOT contain full DECwindows.
To obtain full DECwindows, you must also install the separate
layered product, DECwindows Motif for OpenVMS VAX.
V1.2-3 is the minimum version of DECwindows Motif for OpenVMS VAX
that can be used with OpenVMS VAX V7.3.

The DECwindows components provided in this kit requires approximately
34400 blocks, broken down as follows:

o DECwindows base support - 4400 blocks
o DECwindows workstation support - 23800 blocks
- 75 dots per inch video fonts - (included)
- 100 dots per inch video fonts (optional) - 6200 blocks

You must select the DECwindows base support option if
- you plan to run DECwindows software, or
- you are installing this kit on
* a workstation or
* an OpenVMS Cluster that contains workstations, or
- you want to provide font files for Xterminals.

If you are installing this kit on a system that includes Xterminals
and you do NOT select DECwindows base support, then you will have to use
the DECW$TAILOR utility to provide font files.

* Do you want the DECwindows base support? (Y/N) n

Beginning with OpenVMS V7.1, the DECnet-Plus kit is provided with
the OpenVMS operating system kit. Compaq strongly recommends that
DECnet users install DECnet-Plus. DECnet Phase IV applications are
supported by DECnet-Plus.

DECnet Phase IV is also provided as an option. Support for DECnet
Phase IV is available through a Prior Version Support Contract.

If you install DECnet-Plus and TCP/IP you can run DECnet
```

```
applications over a TCP/IP network. Please see the OpenVMS
Management Guide for information on running DECnet over TCI/IP.

If you plan to install DECnet Phase IV do NOT select DECnet-Plus.

* Do you want to install DECnet-Plus? (Y/N) n

* Do you want to install DECnet Phase IV? (Y/N) n

The following options will be provided:

OpenVMS library
OpenVMS optional
OpenVMS Help Message
OpenVMS Management Station Software -- PC files

Space remaining on system disk: 2752777 blocks

* Is this correct? (Y/N) y

Restoring OpenVMS library save set ...
%BACKUP-I-STARTVERIFY, starting verification pass

Restoring OpenVMS optional save set ...
%BACKUP-I-STARTVERIFY, starting verification pass

Restoring OpenVMS Help Message save set ...
%BACKUP-I-STARTVERIFY, starting verification pass

Restoring OpenVMS Management Station Software -- PC files
%BACKUP-I-STARTVERIFY, starting verification pass

Now registering the OpenVMS operating system in the
POLYCENTER Software Installation product database

The following product will be registered:
DEC VAXVMS VMS V7.3 DISK$VAXVMSV73:[VMS$COMMON.]

The following product has been registered:
DEC VAXVMS VMS V7.3 Transition (registration)

You can now remove the distribution kit from DUA3:.

In an OpenVMS Cluster, you can run multiple systems sharing all files
except PAGEFILE.SYS, SWAPFILE.SYS, SYSDUMP.DMP, and VAXVMSSYS.PAR.

Cluster configuration cannot be done at this time because no network
is present. In order to configure a cluster you must FIRST do one
or both of the following:

o Install DECnet-Plus (or DECnet Phase IV), or
o Execute SYS$STARTUP:LAN$STARTUP.COM by removing the
comment delimiter ("!") from the line

$! @SYS$STARTUP:LAN$STARTUP

in SYS$MANAGER:SYSTARTUP_VMS.COM.

Then configure the cluster by executing the following command:

@ @SYS$MANAGER:CLUSTER_CONFIG

See the OpenVMS System Manager's Manual: Essentials for more information.


Now we will ask you for new passwords for the following accounts:

SYSTEM, SYSTEST, FIELD

Passwords must be a minimum of 8 characters in length. All passwords
will be checked and verified. Any passwords that can be guessed easily
will not be accepted.


* Enter password for SYSTEM: <password here>
* Re-enter for verification: <password here>
%UAF-I-MDFYMSG, user record(s) updated
%VMS-I-PWD_OKAY, account password for SYSTEM verified


* Enter password for SYSTEST: <password here>
```

```
* Re-enter for verification: <password here>
%UAF-I-MDFYMSG, user record(s) updated
%VMS-I-PWD_OKAY, account password for SYSTEST verified


The SYSTEST_CLIG account will be disabled. You must re-enable
it before running UETP but do not assign a password.

%UAF-I-PWDLESSMIN, new password is shorter than minimum password length
%UAF-I-MDFYMSG, user record(s) updated

* Enter password for FIELD: <password here>
* Re-enter for verification: <password here>
%UAF-I-MDFYMSG, user record(s) updated
%VMS-I-PWD_OKAY, account password for FIELD verified


Creating RIGHTS database file, SYS$SYSTEM:RIGHTSLIST.DAT
Ignore any "-SYSTEM-F-DUPIDENT, duplicate identifier" errors.

%UAF-I-RDBCREMSG, rights database created
%UAF-I-RDBADDMSGU, identifier DEFAULT value [000200,000200] added to rights database
%UAF-I-RDBADDMSGU, identifier FIELD value [000001,000010] added to rights database
%UAF-I-RDBADDMSGU, identifier SYSTEM value [000001,000004] added to rights database
%UAF-I-RDBADDMSGU, identifier SYSTEST value [000001,000007] added to rights database
%UAF-E-RDBADDERRU, unable to add SYSTEST_CLIG value [000001,000007] to rights database
-SYSTEM-F-DUPIDENT, duplicate identifier
%UAF-I-NOMODS, no modifications made to system authorization file
%UAF-I-RDBDONEMSG, rights database modified

Creating MODPARAMS.DAT database file, SYS$SYSTEM:MODPARAMS.DAT

* Please enter the SCSNODE name: PSWVAX  (this should be the name of your VAX)

* Please enter the SCSSYSTEMID: 1025

After the installation finishes, you might want to do one or more of the
following tasks:

o DECOMPRESS THE SYSTEM LIBRARIES - To save space, many of the system
libraries are shipped in a data-compressed format. If you have
enough disk space, you can decompress the libraries for faster access.
To data expand the libraries, type:

$ @SYS$UPDATE:LIBDECOMP.COM

If you do not decompress these libraries, you will experience
slower response to the HELP and LINK commands.

o BUILD A STANDALONE BACKUP KIT - You can build a standalone backup kit
using the procedure described in the "Backup Procedures" chapter of
tye upgrade and installation supplement provided for your VAX computer.

o TAILOR THE SYSTEM DISK - You might want to review the files provided or
not provided during this installation. If you find there are files
you want to remove from the system disk (TAILOR OFF) or files you want
to add (TAILOR ON), use the following utilities to perform the
desired tailoring.

OpenVMS tailoring: $ RUN SYS$UPDATE:VMSTAILOR

DECwindows tailoring: $ RUN SYS$UPDATE:DECW$TAILOR

NOTE: The tailor procedure cannot be used to TAILOR ON or TAILOR OFF
files located on an alternate disk.

================================================================
Continuing with OpenVMS VAX V7.3 Installation Procedure.

Configuring all devices on the system ...
```

At this point, it's time to register a license key for VMS. Refer to the email you received in response to your license request to obtain the information that's entered below.

```
If you have Product Authorization Keys (PAKs) to register, you can
register them now.

* Do you want to register any Product Authorization Keys? (Y/N): y
```

```
VMS License Management Utility Options:

1. REGISTER a Product Authorization Key
2. AMEND an existing Product Authorization Key
3. CANCEL an existing Product Authorization Key
4. LIST the Product Authorization Keys
5. MODIFY an existing Product Authorization Key
6. DISABLE an existing Product Authorization Key
7. DELETE an existing Product Authorization Key
8. COPY an existing Product Authorization Key
9. MOVE an existing Product Authorization Key
10. ENABLE an existing Product Authorization Key
11. SHOW the licenses loaded on this node
12. SHOW the unit requirements for this node

99. EXIT this procedure

Type '?' at any prompt for a description of the information
requested. Press Ctrl/Z at any prompt to return to this menu.

Enter one of the above choices [1]:1
Do you have your Product Authorization Key? [YES]: <cr>

Use the REGISTER option to add a new license to a license
database. A Product Authorization Key (PAK) provides the product
name and information you need to register the license. You must
enter all the information provided by your PAK exactly as it
appears.

Type '?' at any prompt for a description of the information
requested. Press Ctrl/Z at any prompt to return to the main menu.

Issuer [DEC]: DECUS
Authorization Number []: DECUS-USA-nnnnnn-nnnnnn
Product Name []: VAX-VMS
Producer [DEC]: DEC
Number of Units [1]: 0
Version []: <cr>
Product Release Date []: <date from email license info> This date may be in the future; don't worry about it.
Key Termination Date []: <date from email license info>
Availability Table Code []: <cr>
Activity Table Code []: a
Key Options []: NO_SHARE
Include Node []: <cr>
Product Token []: <cr>
Hardware-Id []: NONE
Checksum []: n-cccc-cccc-cccc-cccc

Here is a list of the license information just entered:

Issuer: DECUS
Authorization: DECUS-USA-NNNNNN-NNNNNN
Product Name: VAX-VMS
Producer: DEC
Units: 0
Release Date: 5-APR-2004
Version:
Termination Date: 5-APR-2004
Availability:
Activity: A
Options: NO_SHARE
Token:
Hardware ID: NONE
Checksum: N-CCCC-CCCC-CCCC-CCCC

Is that correct? [YES]: <cr>
Registering VAX-VMS license in SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB...

Do you want to LOAD this license on this system? [YES]: <cr>
%LICENSE-I-LOADED, DEC VAX-VMS was successfully loaded with 0 units
```

Then exit the license management tool by using option 99 on the menu. You'll be prompted for information about your time zone next; enter the data requested.

Following that, the system will do some autoconfiguration and will then shut down. The messages tell you that the system will automatically reboot, but this isn't the case due to the way the simulator and virtual CPU are

currently configured.

```
Running AUTOGEN to compute the new SYSTEM parameters ...

%AUTOGEN-I-BEGIN, GETDATA phase is beginning.
%AUTOGEN-I-NEWFILE, A new version of SYS$SYSTEM:PARAMS.DAT has been created.
You may wish to purge this file.
%AUTOGEN-I-END, GETDATA phase has successfully completed.
%AUTOGEN-I-BEGIN, GENPARAMS phase is beginning.
%AUTOGEN-I-NEWFILE, A new version of SYS$MANAGER:VMSIMAGES.DAT has been created.
You may wish to purge this file.
%AUTOGEN-I-NEWFILE, A new version of SYS$SYSTEM:SETPARAMS.DAT has been created.
You may wish to purge this file.
%AUTOGEN-I-END, GENPARAMS phase has successfully completed.
%AUTOGEN-I-BEGIN, GENFILES phase is beginning.
%SYSGEN-I-EXTENDED, SYS$SYSROOT:[SYSEXE]PAGEFILE.SYS;1 extended
%SYSGEN-I-EXTENDED, SYS$SYSROOT:[SYSEXE]SWAPFILE.SYS;1 extended
%SYSGEN-I-CREATED, SYS$SPECIFIC:[SYSEXE]SYSDUMP.DMP;1 created
%SYSGEN-I-CREATED, DUA0:[SYS0.SYSEXE]ERRORLOG.DMP;1 created

%AUTOGEN-I-REPORT, AUTOGEN has produced some informational messages which
have been stored in the file SYS$SYSTEM:AGEN$PARAMS.REPORT. You may
wish to review the information in that file.

%AUTOGEN-I-END, GENFILES phase has successfully completed.
%AUTOGEN-I-BEGIN, SETPARAMS phase is beginning.
%AUTOGEN-I-END, SETPARAMS phase has successfully completed.
%AUTOGEN-I-BEGIN, REBOOT phase is beginning.

The system is shutting down to allow the system to boot with the
generated site-specific parameters and installed images.

The system will automatically reboot after the shutdown and the
installation will be complete.


SHUTDOWN -- Perform an Orderly System Shutdown

%SHUTDOWN-I-BOOTCHECK, performing reboot consistency check...
%SHUTDOWN-I-CHECKOK, basic reboot consistency check completed

%SHUTDOWN-I-OPERATOR, this terminal is now an operator's console
%OPCOM-W-NOOPCOM, the request was not sent, the OPCOM process is not running
%SHUTDOWN-I-DISLOGINS, interactive logins will now be disabled
%SET-I-INTSET, login interactive limit = 0, current interactive value = 0
%SHUTDOWN-I-STOPQUEUES, the queues on this node will now be stopped
%JBC-E-OPENERR, error opening SYS$COMMON:[SYSEXE]QMAN$MASTER.DAT;
-RMS-E-FNF, file not found

SHUTDOWN message from user SYSTEM at Batch 16:32:58
The system will shut down in 0 minutes; back up SOON. Please log off.
Reboot system with AUTOGENerated parameters

%SHUTDOWN-I-STOPUSER, all user processes will now be stopped
%SHUTDOWN-I-REMOVE, all installed images will now be removed
%SHUTDOWN-I-DISMOUNT, all volumes will now be dismounted
%OPCOM-W-NOOPCOM, the request was not sent, the OPCOM process is not running
%OPCOM-W-NOOPCOM, the request was not sent, the OPCOM process is not running
HALT instruction, PC: 8443B709 (MOVB 400(R1),R0)
```

At this point, the virtual CPU is halted and the simulator is back in control. For now, tell it to boot once again (we'll set the system up for automatic boot later). You'll get a whole bunch of messages in response as the final phase of the system installation/initial boot runs.

```
sim> boot cpu
KA655-B V5.3, VMB 2.7
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..09..
08..07..06..05..04..03..
Tests completed.
>>>boot
(BOOT/R5:0 DUA0)

2..
-DUA0
1..0..
```

```
%SYSBOOT-I-SYSBOOT Mapping the SYSDUMP.DMP on the System Disk
%SYSBOOT-I-SYSBOOT SYSDUMP.DMP on System Disk successfully mapped
%SYSBOOT-I-SYSBOOT Mapping PAGEFILE.SYS on the System Disk
%SYSBOOT-I-SYSBOOT SAVEDUMP parameter not set to protect the PAGEFILE.SYS
OpenVMS (TM) VAX Version V7.3 Major version id = 1 Minor version id = 0
%WBM-I-WBMINFO Write Bitmap has successfully completed initialization.

*****************************************************************

OpenVMS VAX V7.3

You have SUCCESSFULLY installed the OpenVMS VAX Operating System.

The system is now executing the STARTUP procedure. Please
wait for the completion of STARTUP before logging in to the
system.

*****************************************************************

%STDRV-I-STARTUP, OpenVMS startup begun at 19-APR-2003 16:36:29.70
%RUN-S-PROC_ID, identification of created process is 00000206
%DCL-S-SPAWNED, process SYSTEM_1 spawned
%%%%%%%%%%% OPCOM 19-APR-2003 16:37:25.92 %%%%%%%%%%%
Operator _PSWVAX$OPA0: has been enabled, username SYSTEM

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:25.96 %%%%%%%%%%%
Operator status for operator _PSWVAX$OPA0:
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, CLUSTER, SECURITY,
LICENSE, OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10,
OPER11, OPER12

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:26.09 %%%%%%%%%%%
Logfile has been initialized by operator _PSWVAX$OPA0:
Logfile is PSWVAX::SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;1

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:26.12 %%%%%%%%%%%
Operator status for operator PSWVAX::SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;1
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, CLUSTER, SECURITY,
LICENSE, OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10,
OPER11, OPER12

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:28.00 %%%%%%%%%%%
Message from user AUDIT$SERVER on PSWVAX
%AUDSRV-I-NEWSERVERDB, new audit server database created

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:28.53 %%%%%%%%%%%
Message from user AUDIT$SERVER on PSWVAX
%AUDSRV-I-REMENABLED, resource monitoring enabled for journal SECURITY

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:28.84 %%%%%%%%%%%
Message from user AUDIT$SERVER on PSWVAX
%AUDSRV-I-NEWOBJECTDB, new object database created

%SET-I-NEWAUDSRV, identification of new audit server process is 0000020C
%%%%%%%%%%% OPCOM 19-APR-2003 16:37:30.91 %%%%%%%%%%%
Message from user JOB_CONTROL on PSWVAX
%JBC-E-OPENERR, error opening SYS$COMMON:[SYSEXE]QMAN$MASTER.DAT;

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:30.93 %%%%%%%%%%%
Message from user JOB_CONTROL on PSWVAX
-RMS-E-FNF, file not found

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:36.07 %%%%%%%%%%%
Message from user SYSTEM on PSWVAX
%SECSRV-E-NOPROXYDB, cannot find proxy database file NET$PROXY.DAT
%RMS-E-FNF, file not found

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:36.49 %%%%%%%%%%%
Message from user SYSTEM on PSWVAX
%SECSRV-E-NOPROXYDB, cannot find proxy database file NET$PROXY.DAT
%RMS-E-FNF, file not found

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:36.52 %%%%%%%%%%%
Message from user SYSTEM on PSWVAX
%SECSRV-I-CIACRECLUDB, security server created cluster intrusion database

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:36.54 %%%%%%%%%%%
Message from user SYSTEM on PSWVAX
%SECSRV-I-SERVERSTARTINGU, security server starting up
```

```
%%%%%%%%%%% OPCOM 19-APR-2003 16:37:36.57 %%%%%%%%%%%
Message from user SYSTEM on PSWVAX
%SECSRV-I-CIASTARTINGUP, breakin detection and evasion processing now starting up

%SMG-W-GBLNOTCRE, global section not created
-SYSTEM-F-GPTFULL, global page table is full
%%%%%%%%%%% OPCOM 19-APR-2003 16:37:43.71 %%%%%%%%%%%
Message from user SYSTEM on PSWVAX
Warning: DECdtm log file not found (SYS$JOURNAL:SYSTEM$PSWVAX.LM$JOURNAL)
%RMS-E-FNF, file not found
TP server process waiting

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:46.78 %%%%%%%%%%%
Message from user AUDIT$SERVER on PSWVAX
Security alarm (SECURITY) and security audit (SECURITY) on PSWVAX, system id: 1025
Auditable event: Audit server starting up
Event time: 19-APR-2003 16:37:46.33
PID: 00000203
Username: SYSTEM

%STARTUP-I-AUDITCONTINUE, audit server initialization complete

The OpenVMS VAX system is now executing the site-specific startup commands.

%%%%%%%%%%% OPCOM 19-APR-2003 16:37:49.20 %%%%%%%%%%%
Message from user AUDIT$SERVER on PSWVAX
Security alarm (SECURITY) and security audit (SECURITY) on PSWVAX, system id: 1025
Auditable event: Identifier added
Event time: 19-APR-2003 16:37:49.12
PID: 00000203
Process name: STARTUP
Username: SYSTEM
Process owner: [SYSTEM]
Image name: PSWVAX$DUA0:[SYS0.SYSCOMMON.][SYSEXE]AUTHORIZE.EXE
Identifier name: SYS$NODE_PSWVAX
Identifier value: %X80010000
Attributes: none

%UAF-I-RDBADDMSG, identifier SYS$NODE_PSWVAX value %X80010000 added to rights database
%SET-I-INTSET, login interactive limit = 64, current interactive value = 0
SYSTEM job terminated at 19-APR-2003 16:37:53.28

Accounting information:
Buffered I/O count: 1525 Peak working set size: 1799
Direct I/O count: 568 Peak page file size: 4963
Page faults: 4754 Mounted volumes: 0
Charged CPU time: 0 00:01:21.45 Elapsed time: 0 00:01:38.25
```

Configuring OpenVMS

Congratulations! You're ready to log in for the first time.

```
<cr>
Welcome to OpenVMS (TM) VAX Operating System, Version V7.3

Username: system
Password: <password you selected above>

Welcome to OpenVMS (TM) VAX Operating System, Version V7.3

$
```

Bask in the glow of the long-lost DCL "**$** " prompt for a moment. Some Unix shells may also use the dollar-sign prompt, but it's just really not the same.

On with configuration: let's initialize two additional disks. These disks, like the virtual DUA0, are configured as RA92 disks in the vax.ini file that the simulator uses. It's possible, of course, to use other disk types or to use only the system disk if you'd rather.

```
$ initialize dua1: DATA1
$ initialize dua2: DATA2
$ mount/system dua1 data1
%MOUNT-I-MOUNTED, DATA1 mounted on _PSWVAX$DUA1:
```

```
$ mount/system dua2 data2
%MOUNT-I-MOUNTED, DATA2 mounted on _PSWVAX$DUA2:
```

Since you'll want to do this every time the system boots, you should put these commands in the **sys$manager:systartup_vms.com** site-specific startup file. Those familiar with VMS editors know that there are many shortcuts (search capabilities, the ability to scroll by pages, etc.) that can save time in the editing process. Explanation of those capabilities is far beyond the scope of this document, however; you should be able to stumble your way through editing files using this basic process:

```
$ set term/vt100
$ edit sys$manager:systartup_vms.com
```

Scroll down within the file until you find a place to insert these MOUNT commands, then edit the file appropriately. Use Control-Z to save and exit.

Now, let's add a user to the system. Here's how I added myself. Several notes on this:

- I'm putting user directories on the **DUA1** disk. If you're not, make appropriate changes.
- Every VMS user has to have an octal group and user code (known together as the user identification code, or UIC). I've used **200** for the group and **201** for the user code. UICs (the pairs) should be unique.
- User accounts are initially disabled, so it's necessary to indicate that the account should be unlocked with **/flag=nodisuser**.
- This user has full system privileges, though they're not on by default. Omit the **/priv=all** section if you just want a normal user.

```
$ set def sys$system
$ r authorize
UAF> add psw/password=temp/owner="Phil Wherry"/dev=dua1/dir=[psw]/uic=[200,201]/flag=nodisuser/priv=all
%UAF-I-PWDLESSMIN, new password is shorter than minimum password length
%UAF-I-ADDMSG, user record successfully added
```

This will generate a couple of security audit alarms (you're receiving these because you're currently logged in at the system's console). Then exit the user authorization facility application and create the user's directory. Once you're done with that, log out of the **SYSTEM** account.

```
%UAF-I-RDBADDMSGU, identifier PSW value [000200,000201] added to rights database
UAF> exit
%UAF-I-DONEMSG, system authorization file modified
%UAF-I-RDBDONEMSG, rights database modified
$ create/dir dua1:[psw]
$ set directory/owner=psw dua1:[psw]
$ lo
SYSTEM logged out at 19-APR-2003 16:46:14.91
```

Now log back in with your newly-created identity. From here, we'll install TCP/IP.

```
Welcome to OpenVMS (TM) VAX Operating System, Version V7.3

Username: psw
Password: <password>

Welcome to OpenVMS (TM) VAX Operating System, Version V7.3
Last interactive login on Saturday, 19-APR-2003 16:44
```

Enable your privileges (this is the rough equivalent of the **su** command in Unix).

```
$ set proc/priv=all
```

TCP/IP requires more system resources than are available by default, so we need to edit a file called

modparams.dat in order to reserve the appropriate resources.

```
$ set def sys$system
$ edit modparams.dat
```

Add these lines to the file:

```
ADD_GBLPAGES=10000
ADD_GBLSECTIONS=100
ADD_NPAGEDYN=800000
ADD_NPAGEVIR=800000
MIN_SPTREQ=6000
```

Once this is done, we use the AUTOGEN facility to update the system:

```
$ set def sys$update
$ @autogen getdata shutdown nofeedback
%AUTOGEN-I-BEGIN, GETDATA phase is beginning.
%AUTOGEN-I-NEWFILE, A new version of SYS$SYSTEM:PARAMS.DAT has been created.
You may wish to purge this file.
%AUTOGEN-I-END, GETDATA phase has successfully completed.
%AUTOGEN-I-BEGIN, GENPARAMS phase is beginning.
%AUTOGEN-I-NEWFILE, A new version of SYS$MANAGER:VMSIMAGES.DAT has been created.
You may wish to purge this file.
%AUTOGEN-I-NEWFILE, A new version of SYS$SYSTEM:SETPARAMS.DAT has been created.
You may wish to purge this file.
%AUTOGEN-I-END, GENPARAMS phase has successfully completed.
%AUTOGEN-I-BEGIN, GENFILES phase is beginning.
%SYSGEN-I-EXTENDED, SYS$SYSROOT:[SYSEXE]SWAPFILE.SYS;1 extended

%AUTOGEN-I-REPORT, AUTOGEN has produced some informational messages which
have been stored in the file SYS$SYSTEM:AGEN$PARAMS.REPORT. You may
wish to review the information in that file.

%AUTOGEN-I-END, GENFILES phase has successfully completed.
%AUTOGEN-I-BEGIN, SETPARAMS phase is beginning.
%AUTOGEN-I-END, SETPARAMS phase has successfully completed.
%AUTOGEN-I-BEGIN, SHUTDOWN phase is beginning.

The system is shutting down to allow the system to boot with the
generated site-specific parameters and installed images.

You must manually reboot the system after it halts.
.
. (messages deleted)
.
SYSTEM SHUTDOWN COMPLETE - use console to halt system
Infinite loop, PC: 833E50D3 (BRB 833E50D3)
sim>
```

Once again, use the simulator to restart the virtual machine.

```
sim> boot cpu
```

Once the machine restarts, log in again. There's one more parameter to be updated; another reboot will be required.

```
$ set proc/priv=all
$ r sys$system:sysgen
SYSGEN> SET INTSTKPAGES 20
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
$ @sys$system:shutdown
```

(For the more-experienced: yes, it is possible to do both AUTOGEN and SYSGEN with only one reboot cycle. I thought it better not to combine the processes for the sake of clarity and generality.)

Note: If you intend to run DECNET Phase IV at some point, you'll also want to set the **SCSSYSTEMID** parameter by multiplying the DECNET area number by 1024, then adding the DECNET node number. So,

to prepare to be node number 3 in DECNET area 2, multiply the area number (2) by 1024, add the node number (3), to get 2051. The command **SET SYSSYSTEMID 2051** within the SYSGEN session above would be required to support DECNET. If you're not familiar with DECNET, or have no immediate plans to run it, you may safely ignore this step.

Installing TCP/IP

Boot the machine once again, then log in again; we'll install TCP/IP from here.

```
$ set proc/priv=all
$ mount/over=id dua3:
%MOUNT-I-WRITELOCK, volume is write locked
%MOUNT-I-MOUNTED, VAXVMS073 mounted on _PSWVAX$DUA3:
```

The **/over=id** qualifier on the mount command tells the operating system to ignore the volume label. This is really important if you don't know the label for the CD-ROM already. It's inconsistent with the **/SYSTEM** qualifier, however, so the CD-ROM will be visible only to you and will be dismounted automatically when you log out.

Change directories to the TCP/IP installation kit.

```
$ set def dua3:[tcpip_vax051.kit]
```

Before installing, register and activate the license. The UCX license from your layered product email will give you what you need. Note the use of the hyphen command-line continuation characters here.

```
$ LICENSE REGISTER UCX -
_$ /ACTIVITY=CONSTANT=100 -
_$ /AUTHORIZATION=DECUS-USA-NNNNNN-NNNNNN -
_$ /DATE=<date> -
_$ /ISSUER=DECUS -
_$ /PRODUCER=DEC -
_$ /TERMINATION=<date> -
_$ /UNITS=0 -
_$ /CHECKSUM=N-CCCC-CCCC-CCCC-CCCC
$ license load UCX
%LICENSE-I-LOADED, DEC UCX was successfully loaded with 0 units
```

Now, launch the installer.

```
$ product install *

The following product has been selected:
DEC VAXVMS TCPIP V5.1-15 Layered Product

Do you want to continue? [YES] <cr>

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product and for
any products that may be installed to satisfy software dependency requirements.

DEC VAXVMS TCPIP V5.1-15: Compaq TCP/IP Services for OpenVMS.

(c) Compaq Computer Corporation 2000. All Rights Reserved.

Compaq Computer Corporation

Compaq TCP/IP Services for OpenVMS offers several license options.

Do you want the defaults for all options? [YES] <cr>

Do you want to review the options? [NO] <cr>

Execution phase starting ...
```

```
The following product will be installed to destination:
DEC VAXVMS TCPIP V5.1-15 DISK$OVMSVAXSYS:[VMS$COMMON.]

Portion done: 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%
%PCSI-I-PRCOUTPUT, output from subprocess follows ...
% - HELP has been updated. You may purge SYS$COMMON:[SYSHLP]HELPLIB.HLB
%
%PCSI-I-PRCOUTPUT, output from subprocess follows ...
% TCPIP-W-PCSI_INSTALL
% - Execute SYS$MANAGER:TCPIP$CONFIG.COM to proceed with configuration of
% Compaq TCP/IP Services.
%
Portion done: 100%

The following product has been installed:
DEC VAXVMS TCPIP V5.1-15 Layered Product

DEC VAXVMS TCPIP V5.1-15: Compaq TCP/IP Services for OpenVMS.

Check the release notes for current status of the product.
```

Now, configure TCP/IP...

```
$ @sys$manager:tcpip$config
Compaq TCP/IP Services for OpenVMS Configuration Menu

Configuration options:

1 - Core environment
2 - Client components
3 - Server components
4 - Optional components

5 - Shutdown Compaq TCP/IP Services for OpenVMS
6 - Startup Compaq TCP/IP Services for OpenVMS
7 - Run tests

A - Configure options 1 - 4
[E] - Exit configuration procedure

Enter configuration option: 1
```

The most basic options are under option 1: core environment. These are the familiar TCP/IP configuration values; alter as appropriate for your network.

```
Compaq TCP/IP Services for OpenVMS Core Environment Configuration Menu

Configuration options:

1 - Domain
2 - Interfaces
3 - Routing
4 - BIND Resolver
5 - Time Zone

A - Configure options 1 - 5
[E] - Exit menu

Enter configuration option: 1
DOMAIN Configuration

Enter Internet domain: wherry.com

Enter configuration option: 2
QE0 is the Ethernet device XQA0:

QE0 has not been configured

Compaq TCP/IP Services for OpenVMS Interface QE0 Configuration Menu

Configuration options:

1 - Configure interface manually
2 - Let DHCP configure interface

[E] - Exit menu (Do not configure interface QE0)
```

```
Enter configuration option: 1
Enter fully qualified host name: vax.wherry.com
Enter Internet address for vax: 207.196.42.9 (note that this address should be different than the emulator machine's address, and should not be in use on your network)
Enter Internet network mask for vax [255.255.255.0]: <cr>
Enter broadcast mask for vax [207.196.42.255]: <cr>

The following parameters will be used to define the
Internet interface QE0:

Host name: vax
Internet address: 207.196.42.9
Network mask: 255.255.255.0
Broadcast mask: 207.196.42.255

* Is the above correct [YES]: <cr>

Enter configuration option: 3

DYNAMIC ROUTING Configuration

Dynamic routing has not been configured.

You may configure dynamic ROUTED or GATED routing.
You cannot enable both at the same time. If you want
to change from one to the other, you must disable the
current routing first, then enable the desired routing.

If you enable dynamic ROUTED routing, this host will use the
Routing Information Protocol (RIP) - Version 1 to listen
for all dynamic routing information coming from other
hosts to update its internal routing tables.
It will also supply its own Internet addresses to
routing requests made from remote hosts.

If you enable dynamic GATED routing, you will be able to
configure this host to use any combination of the following
routing protocols to exchange dynamic routing information
with other hosts on the network:
Routing Information Protocol (RIP) - Version 1 & 2
Router Discovery Protocol (RDISC)
Open Shortest Path First (OSPF)
Exterior Gateway Protocol (EGP)
Border Gateway Protocol (BGP-4)
Static routes

* Do you want to configure dynamic ROUTED or GATED routing [NO]: <cr>

A default route has not been configured.
* Do you want to configure a default route [YES]: <cr>
Enter your Default Gateway host name or address: 207.196.42.1

207.196.42.1 is not in the local host database.
If you want to enter the default gateway in the local host
database, enter its host name. Otherwise, enter <CR>.

Enter the Default Gateway host name []: <cr>

Enter configuration option: 4
BIND RESOLVER Configuration

A BIND resolver has not been configured.

Compaq TCP/IP Services for OpenVMS supports the Berkeley Internet Name
Domain (BIND) resolver. BIND is a network service that enables clients
to name resources or objects and share information with other objects
on the network.

Before configuring your system as a BIND resolver, you should
first be sure that there is at least one system on the network
configured as either a BIND primary or secondary server for
this domain.

You can specify a BIND server by its address or name;
however, if specified by name, an entry for it must
exist in the TCPIP$HOST database.

You will be asked one question for each server.
Press Return at the prompt to terminate the list.
```

```
Enter your BIND server name: 207.196.42.2

207.196.42.2 is not in the local host database.
If you want to enter the server in the local host
database, enter the server name. Otherwise, enter <CR>.

Enter remote BIND server name []: <cr>

Enter next BIND server name: <cr>

Enter configuration option: e
```

It's worth a little digression at this point to talk about how the Ethernet adapter for the VAX is emulated in SIMH. Experts will notice some significant oversimplification in this section; if you're one of those folks, I hope you'll agree that the simplifications don't distort the underlying message much.

Under normal circumstances, an Ethernet card sends data to the operating system on which it's running when it receives packets destined for its particular MAC (media access control) address. This is a hardware identifier that's supposed to be unique to each Ethernet card, so you usually don't have to worry about it. It also sends data to the operating system when it receives packets destined for the special MAC address FF-FF-FF-FF-FF-FF (the "broadcast" address). It ignores everything else. This keeps load on the system reasonable, since it can ignore packets on the network that don't concern it.

In order to sort out what packets should go to the VAX emulation, and which packets should go to the emulator machine on which it runs, it's necessary to have the Ethernet card respond to a *second* MAC address in addition to its native address. Some cards can be given a list of addresses, but it's certainly not universal. So, the SIMH simulator puts the network adapter in the colorfully-named *promiscuous mode*, in which the Ethernet card sends every packet it sees on to the host operating system and its application (in this case Linux and SIMH). The operating system reacts to the packets destined for the hardware MAC address on the Ethernet card, as always. But the SIMH application also gets the packets, and it can choose to react to any of them. Since it's looking for an alternate MAC address, the server on which the emulator runs and the VAX emulator itself appear to be completely separate.

There are three consequences to this approach, however:

- The emulator must be run as root. For security reasons, Linux won't permit just any application to put the Ethernet adapter in promiscuous mode, as it allows snooping on network traffic to and from the Linux machine in addition to the traffic of any other machine on the same Ethernet segment. The security risk posed by this fact isn't huge, but it is real. To take advantage of this fact, an attacker would have to be able to cause the emulator to malfunction in a way that gives it undesired access to the underlying Linux machine. Since this would generally involve simultaneous successful attacks on both the emulated machine (the VAX) and the Linux server on which it runs, it's unlikely. But the risk isn't zero, either; the cautious (or paranoid) administrator will likely run the VAX emulation on its own dedicated machine.
- CPU and/or interrupt loading on the Linux server may become a problem if the Ethernet is very busy. This is because *every* packet received by the Ethernet card must be examined by the operating system, rather than only those packets bearing the adapter's MAC address or the broadcast address. This is often much less of a big deal that it would appear, however, since the emulator machine is frequently connected to an Ethernet switch. The Ethernet switch will pass through packets destined for either the broadcast address or any of the MAC addresses to which the connected Ethernet adapter responds. The net effect is that all traffic not needed by the server or the emulation running on it is filtered out, just as would be the case if the Ethernet were not in promiscuous mode. For users running their servers connected to a *hub* rather than a *switch*, however, this may pose a performance problem.
- It's likely you'll have problems getting the emulator and the environment in which it runs (say, Linux) to

communicate with each other on the same machine. The reason for this is simple: the Ethernet card sends a packet out onto the network under one MAC address looking for a MAC address that's actually also served by the same card. By design, Ethernet cards don't receive their own traffic, so the result is that the packet disappears, well, into the ether. If this is a problem, the simplest solution is to put a second Ethernet card into the machine and dedicate it to SIMH; communication between the emulator and its host environment will then take place via the network just as if the emulator were running on a physically separate machine.

Once the core environment is configured, configure client and server components. I recommend enabling FTP and TELNET services as both client and server applications. Configuring the clients for these applications will also enable the servers, as the listing below indicates.

```
Compaq TCP/IP Services for OpenVMS Configuration Menu

Configuration options:

1 - Core environment
2 - Client components
3 - Server components
4 - Optional components

5 - Shutdown Compaq TCP/IP Services for OpenVMS
6 - Startup Compaq TCP/IP Services for OpenVMS
7 - Run tests

A - Configure options 1 - 4
[E] - Exit configuration procedure

Enter configuration option: 2

Compaq TCP/IP Services for OpenVMS Client Components Configuration Menu

Configuration options:

1 - FTP Disabled Stopped
2 - NFS Client Disabled Stopped
3 - REXEC and RSH Disabled Stopped
4 - RLOGIN Disabled Stopped
5 - SMTP Disabled Stopped
6 - TELNET Disabled Stopped
7 - DHCP Disabled Stopped
8 - Telnetsym Disabled Stopped

A - Configure options 1 - 8
[E] - Exit menu

Enter configuration option: 1

FTP CLIENT Configuration

Service is not enabled.
Service is stopped.

FTP CLIENT configuration options:

1 - Enable service on this node

[E] - Exit FTP_CLIENT configuration

Enter configuration option: 1

The FTP SERVER is not enabled.

* Do you want to configure FTP SERVER [NO]: y

Enter configuration option: e

Compaq TCP/IP Services for OpenVMS Configuration Menu

Configuration options:

1 - Core environment
2 - Client components
```

```
3 - Server components
4 - Optional components

5 - Shutdown Compaq TCP/IP Services for OpenVMS
6 - Startup Compaq TCP/IP Services for OpenVMS
7 - Run tests

A - Configure options 1 - 4
[E] - Exit configuration procedure

Enter configuration option: 3

Compaq TCP/IP Services for OpenVMS Server Components Configuration Menu

Configuration options:

1 - BIND Disabled Stopped 11 - NTP Disabled Stopped
2 - BOOTP Disabled Stopped 12 - PC-NFS Disabled Stopped
3 - DHCP Disabled Stopped 13 - POP Disabled Stopped
4 - FINGER Disabled Stopped 14 - PORTMAPPER Disabled Stopped
5 - FTP Enabled Stopped 15 - RLOGIN Enabled Stopped
6 - LBROKER Disabled Stopped 16 - RMT Disabled Stopped
7 - LPR/LPD Disabled Stopped 17 - SNMP Disabled Stopped
8 - METRIC Disabled Stopped 18 - TELNET Enabled Stopped
9 - NFS Disabled Stopped 19 - TFTP Disabled Stopped
10 - LOCKD/STATD Disabled Stopped 20 - XDM Disabled Stopped

A - Configure options 1 - 20
[E] - Exit menu

Enter configuration option: e
```

Once finished, you can proceed to start up the TCP/IP service.

```
Compaq TCP/IP Services for OpenVMS Configuration Menu

Configuration options:

1 - Core environment
2 - Client components
3 - Server components
4 - Optional components

5 - Shutdown Compaq TCP/IP Services for OpenVMS
6 - Startup Compaq TCP/IP Services for OpenVMS
7 - Run tests

A - Configure options 1 - 4
[E] - Exit configuration procedure

Enter configuration option: 6

Begin Startup...

%TCPIP-I-INFO, TCP/IP Services startup beginning at 19-APR-2003 18:17:13.86
%TCPIP-I-INFO, creating UCX compatibility file SYS$COMMON:[SYSEXE]UCX$SERVICE.DAT
%TCPIP-I-NORMAL, timezone information verified
%RUN-S-PROC_ID, identification of created process is 00000211
%%%%%%%%%%% OPCOM 19-APR-2003 18:17:18.87 %%%%%%%%%%%
Message from user INTERnet on PSWVAX
INTERnet Loaded

%TCPIP-I-SETLOCAL, setting domain and/or local host
%TCPIP-I-STARTCOMM, starting communication
%%%%%%%%%%% OPCOM 19-APR-2003 18:17:22.48 %%%%%%%%%%%
Message from user INTERnet on PSWVAX
INTERnet Started

%TCPIP-I-SETPROTP, setting protocol parameters
%TCPIP-I-DEFINTE, defining interfaces
%%%%%%%%%%% OPCOM 19-APR-2003 18:17:24.25 %%%%%%%%%%%
Message from user INTERnet on PSWVAX
INTERnet ACP Created INTERnet interface: QE0

%TCPIP-I-STARTNAME, starting name service
%%%%%%%%%%% OPCOM 19-APR-2003 18:17:25.38 %%%%%%%%%%%
Message from user Proxy Server on PSWVAX
Loading proxy server image TCPIP$PROXY_SERVICES

%TCPIP-S-STARTDONE, TCP/IP Kernel startup completed
```

```
%TCPIP-E-PROXYERROR, error processing proxy request
-TCPIP-W-NORECORD, information not found
-RMS-E-RNF, record not found
%TCPIP-E-PROXYERROR, error processing proxy request
-TCPIP-W-NORECORD, information not found
-RMS-E-RNF, record not found
%TCPIP-I-LOADSERV, loading TCPIP server proxy information
%TCPIP-I-SERVLOADED, auxiliary server loaded with 0 proxy records
-TCPIP-I-SERVSKIP, skipped 0 communication proxy records
-TCPIP-I-SERVTOTAL, total of 0 proxy records read
%TCPIP-S-STARTDONE, TCPIP$PROXY startup completed
%%%%%%%%%%% OPCOM 19-APR-2003 18:17:42.58 %%%%%%%%%%%
Message from user INTERnet on PSWVAX
INTERnet ACP Activate FTP Server

%%%%%%%%%%% OPCOM 19-APR-2003 18:17:42.60 %%%%%%%%%%%
Message from user INTERnet on PSWVAX
INTERnet ACP NOLISTEN Process creation success: Service - FTP

%TCPIP-S-STARTDONE, TCPIP$FTP startup completed
%TCPIP-S-STARTDONE, TCPIP$FTP_CLIENT startup completed
%%%%%%%%%%% OPCOM 19-APR-2003 18:17:48.61 %%%%%%%%%%%
Message from user INTERnet on PSWVAX
INTERnet ACP Activate TELNET Server

%TCPIP-S-STARTDONE, TCPIP$TELNET startup completed
%TCPIP-S-STARTDONE, TCP/IP Services startup completed at 19-APR-2003 18:17:50.44

Startup request completed.
Press Return to continue ... <cr>
```

At this point, it should be possible to telnet to the VAX. At some point, you'll want to edit the command **`@sys$startup:tcpip$startup`** into the **`sys$manager:systartup_vms.com`** startup command file.

Other Tasks

I also recommend decompressing the system libraries for performance reasons. Use the command:

```
$ @SYS$UPDATE:LIBDECOMP.COM
```

to accomplish this.

Installing languages is fairly straightforward. Several popular languages (C, Pascal, FORTRAN, and Basic) are included on the Hobbyist CD. License keys should have been emailed to you (same email containing the UCX key for TCP/IP). I'd recommend installing the licenses first based on instructions in the email. Remember that licenses have to be installed, then loaded (if they're just installed, they won't be active until the next boot). The installation process for each language is quite similar; here's an example of the command used to install the C compiler, assuming the CD is mounted on the drive **`DUA3`**:

```
$ @SYS$UPDATE:VMSINSTAL CC064 DUA3:[CC064.KIT]
```

FORTRAN would be installed with:

```
$ @SYS$UPDATE:VMSINSTAL FORT066 DUA3:[FORT066.KIT]
```

Pascal installs with:

```
$ @SYS$UPDATE:VMSINSTAL PASCAL058 DUA3:[PASCAL058.KIT]
```

...and Basic installs with the command:

```
$ @SYS$UPDATE:VMSINSTAL BASIC039 DUA3:[BASIC039.KIT]
```

Additional Linux Integration Tasks

Once the OpenVMS operating system is installed and configured, there are a few other things that can be done with the SIMH installation to streamline operations. Here's a copy of the vax.ini file that I use operationally:

```
;
; Load CPU microcode
load -r /usr/local/vax/data/ka655.bin
;
; Attach non-volatile RAM to a file
attach nvr /usr/local/vax/data/nvram.bin
;
; This virtual machine has 64M memory
set cpu 64m
;
; Define disk drive types. RA92 is largest-supported VAX drive.
set rq0 ra92
set rq1 ra92
set rq2 ra92
set rq3 cdrom
;
; Attach defined drives to local files
attach rq0 /usr/local/vax/data/d0.dsk
attach rq1 /usr/local/vax/data/d1.dsk
attach rq2 /usr/local/vax/data/d2.dsk
;
; Attach the CD-ROM to its file (read-only)
attach -r rq3 /usr/local/vax/data/cd.iso
;
; Disable unused devices. It's also possible to disable individual devices,
; using a construction like "set rq2 disable" if desired.
;
set rl disable
set ts disable
;
; Attach Ethernet to a network interface
set xq mac=08-00-2B-AA-BB-CC
attach xq eth0
;
; Uncomment the line below to enable auto-boot
dep bdr 0
;
; Choose one of the following lines. SET CPU CONHALT returns control to the
; VAX console monitor on a halt event (where behavior will be further
; determined by whether auto-boot is set--see above. SET CPU SIMHALT will
; cause the simulator to get control instead.
set cpu conhalt
;set cpu simhalt
;
; Now start the emulator
boot cpu
;
; Exit the simulator
exit
```

The beginning of this file is identical to the vax.ini that we used during the configuration of OpenVMS. Some changes have been made near the end, however. Let's examine these in detail.

First, the line:

```
dep bdr 0
```

is used to deposit the value 0 in the BDR register. This register contains the boot jumper/switch flag used by the VAX console ROM code. If bit 7 is high (i.e. if we'd instead said "**`dep bdr 80`**"), the ROM will print the VAX console prompt (**>>>**) instead of autobooting. For those who remember the VAX 11/7xx series machines, setting BDR to 0 is the equivalent of setting the key switch to the DISABLE position; setting BDR to 80 [hex] is like turning the key switch to its ENABLE position. As an alternative, Dave Hittner points out that the VAX console command "**`set halt`** n" will allow this value to be set from within the VAX console

environment, and that this is the standard way of accomplishing this in a non-emulated environment. Some research on the Internet suggests that the "**set halt 2**" command will enable auto-boot and "**set halt 3**" will disable it.

The line:

```
set cpu conhalt
```

is related. It controls what happens on a CPU halt event. Control can go either to the VAX console ROM (which will then either display a console prompt or reboot depending on the value in BDR, described above), or to the simulator's command mode.

The combination of the BDR and the console halt mode settings above will cause the automatic reboot mode to work correctly.

As it turns out, the simulator still gets control in this configuration when the VAX CPU is commanded to shut down but not reboot. That said, let's examine the final two command lines in the file:

```
; Now start the emulator
boot cpu
;
; Exit the simulator
exit
```

The "**boot cpu**" command is fairly obvious; it causes the VAX to start up (and, in the case of this command file, autoboot).

The "**exit**" command that follows will be executed when the simulation halts for some reason; this is most likely the result of a non-reboot shutdown. The important thing here, though, is that it causes the SIMH VAX emulator process to exit.

This seems kind of pointless until one thinks about running the VAX emulator unattended. Consider this shell script:

```
#!/bin/bash
while /usr/local/bin/vax_enabled
   do /usr/local/vax/bin/vax </dev/tty8 >/dev/tty8 2>/dev/tty8
done
```

The file **/usr/local/bin/vax_enabled** is a symbolic link to either **/bin/true** or **/bin/false**. If it's a symlink to **/bin/true**, then the script will loop forever. One can cause the VAX process to stop gracefully, however, by first linking **/usr/local/bin/vax_enabled** to **/bin/false**, then shutting down the VAX without automatic restart. When control passes back to the simulator, it'll exit (because of the "**exit**" command in its initialization script) and will not be restarted.

There's one other subtlety to be found in this file. Note that input, output, and error output are redirected to **/dev/tty8**. On a typical Linux machine, this is an unused virtual console that can be accessed with the Alt-F8 keysequence (or Control-Alt-F8 if X is running). The significance of this is substantial: since the virtual machine now has a source of input and output, it can be run unattended (from, say, a machine startup shell script). Console I/O remains possible, however, from the physical console of the Linux machine.

After the original publication of this article, Stan Quayle wrote and suggested that the **screen** utility is a much more flexible choice than redirecting the input and output to/from **/dev/tty8**. The command:

```
screen -m -d -h 2000 /usr/local/vax/bin/vax
```

will start up the VAX emulator with a 2000-line scrollback buffer, but detached from any physical terminal. To connect to this virtualized console, the command

```
screen -r
```

may be used. Type Control-A, followed by D in order to disconnect. The shell script approach may still be helpful in cases where you'd like to re-invoke the VAX automatically after a crash/shutdown; if you choose to do this, don't redirect I/O to/from **`/dev/tty8`**; you can then invoke this script using **`screen`** (substitute the script's name in place of **`/usr/local/vax/bin/vax`** in the **`screen`** startup command above).

```
#!/bin/bash
while /usr/local/bin/vax_enabled
   do /usr/local/vax/bin/vax
done
```

Final Words

The VAX/VMS environment remains an interesting one; I hope that being able to work in an old, familiar environment once again brings back a few good memories. A couple of final observations:

- This project really serves as an impressive reminder of the progress that's been made in computing hardware in the last couple of decades. The computer on which I'm running the emulator would cost no more than $350 if purchased new, and it vastly outperforms the half-million-dollar machines of twenty years ago, even when handicapped by the fact that it's running old software atop an emulation platform. This is an impressive and humbling feat.
- It's similarly impressive to see the longevity and flexibility of well-designed, well-engineered software. I'm not sure that the field of computer science has been particularly well-served by recent practice in commercial software development, in which software has become a primary rather than an ancillary product. Since software has become a commodity to be bought and sold (rather than an enabler for hardware products), long-lasting design has become nearly antithetical to the ongoing revenue stream desired by software marketeers.
- Sharing and collaboration made this endeavor possible. The emulator and the server environment (Linux) on which I'm running it were both made possible by the community-mindedness of their creators. HP/Compaq/DEC deserve special credit for making the operating system available at no charge to the hobbyist community, as well; I think it took a real understanding of the historical importance of VMS (and no small amount of courage) to do this.

And, in closing, one more request for feedback: if you find errors, have suggestions, or desire clarification on this document, please don't hesitate to get in touch with me. Similarly, if you find these directions helpful, I'd love to hear how you're using the tools.

Phil Wherry
psw-at-wherry-dot-com