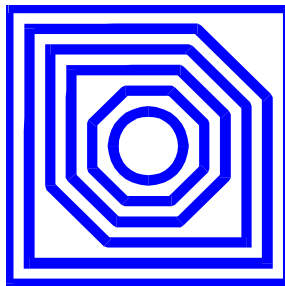


CHARON-VAX

MicroVAX Software Emulator

Hobbyist User Manual



For Red Hat Linux

November 2000

© 2000 Software Resources International S.A.

Email: info@softresint.com



Ownership Notice

Software Resources International SA owns all rights to a methodology for the execution of VAX applications and system software by emulating VAX hardware components in software, henceforth referred to as CHARON-VAX.

The commercial use licenses for the CHARON-VAX emulator allow the execution of the CHARON-VAX run-time system on a single host system or cluster. Software Resources International makes no representations that the use of CHARON-VAX as described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this manual imply the granting of licenses to make, use or sell equipment or software in accordance with the description.

The hobbyist releases of CHARON-VAX are provided by Software Resources International to help private individuals to enjoy the features and flexibility of VAX software on home computers; these releases are not for educational or commercial use. The functionality descriptions in this manual are indications only; Software Resources International does not provide guarantees concerning correctness of this manual or concerning performance, reliability or continuous availability of the CHARON-VAX hobbyist editions. You use the software and documentation entirely at your own risk.

This emulator is only usable in combination with an operating system for MicroVAX computers, which is not provided by Software Resources International. For hobbyists, such software is for instance available through the former DECUS user society. Please observe the licensing requirements for any system or application software you use with the CHARON-VAX emulator.

Copyright and trademarks

The copyright of this manual and the hobbyist edition of the CHARON-VAX emulator belong to Software Resources International. This manual and the hobbyist software can be freely copied for PRIVATE, NON-COMMERCIAL use, provided this copyright notice and the trademark acknowledgement are affixed to each copy, and the manual content and software are not altered.

The CHARON name / logo is a trademark of Software Resources International. PDP-11, VAX, Qbus, UNIBUS, VMS and OpenVMS are trademarks of Compaq Computer Corporation. Windows is a registered trademark in the United States and other countries, licensed exclusively through Microsoft Corporation, USA. Intel, and Pentium II (III) are registered trademarks in the United States and other countries, licensed exclusively through Intel Corporation, USA. All other trademarks and registered trademarks are the property of their respective holders.

Second edition, November 2000

Contents

1. Overview	1-1
Functionality	1-1
2. Installing and using CHARON-VAX	2-1
2.1 CHARON-VAX software installation	2-1
2.2 Installing your VAX software	2-1
2.3 Starting the emulator	2-2
2.4 The CHARON-VAX command interpreter	2-2
3. Emulation and connection of components	3-3
3.1 Emulating VAX disk drives	3-3
3.2 Serial line connections	3-3
4. CHARON-VAX components	4-1
4.1 The MicroVAX kernel components	4-1
4.1.1 The VAXBUS	4-1
4.1.2 The EPRBUS	4-1
4.1.3 The peripheral bridge	4-1
4.1.4 The Central Processing Unit (CPU)	4-1
4.1.5 The Memory Management Unit (MMU)	4-2
4.1.6 Time of year clock (TOY)	4-2
4.1.7 The MicroVAX real-time clock	4-2
4.2 The console port	4-2
4.3 The disk controller	4-2
4.4 Asynchronous serial lines	4-3
4.5 Terminal emulation	4-4
APPENDIX A: The Command interpreter	1
APPENDIX B: VMS operation notes	1

About This Manual

This manual assists you in using the hobbyist release of CHARON-VAX for Linux, the software emulator for MicroVAX hardware. CHARON-VAX can execute unmodified MicroVAX software by creating a model of a MicroVAX system as a software application on a host computer. This software model accurately executes binary software coded in the MicroVAX instruction set, such as the VAX/VMS operating system, diagnostic software etc. This hobbyist edition is intended for private, non-commercial use, and is a modified subset of the commercial CHARON-VAX products.

This manual explains how to install and use this version of CHARON-VAX and describes the overall CHARON-VAX design. The commercial versions of CHARON-VAX are based on a similar design as the hobbyist version, but have additional components and functionality. Users of the PDP-11 emulator CHARON-11 of Software Resources International will notice that the use of CHARON-VAX is quite similar.

If you have questions...

If you have questions or suggestions about the hobbyist version of CHARON-VAX, feel free to contact us via Email at vaxinfo@softresint.com.

If you find any errors in the software we would like to hear from you; there is a software problem report form (SPR Form) on our website www.vaxemulator.com. We are also interested in your views on how to extend the functionality or about potentially interesting applications of a VAX emulator. Our work on commercial projects has priority, but we will try to respond as soon as we can make the time available.

1. Overview

The CHARON-VAX emulator is a software package that emulates the *hardware of a computer system*, a MicroVAX II in the case of this hobbyist version. It provides an exact model of the emulated system hardware, which runs as a normal application on a host system. This in contrast to an operating system emulator that imitates an existing software environment, for instance the VAX/VMS operating system.

The important advantage of hardware emulation is that existing binary systems and application code of the old system will execute as long as the emulator model is built correctly. You don't have to re-implement the complexity of the old operating systems on the new platform, and the existing application behavior and user interface are exactly preserved.

That is not to say that building a hardware emulator is simple. While technical information of older systems is often in the public domain, the interaction between emulated components in real time is very complicated. This is especially true if the components of the emulator are designed to operate asynchronously, as is the case with CHARON-VAX. This asynchronous operation permits efficient utilization of the host system, and allows obtaining a reasonable performance of the emulated environment. The executable should be as small as possible to benefit from cache memory. The CHARON-VAX executable is less than 500 KB.

The various commercial versions of CHARON-VAX are primarily designed to prolong the use of existing VAX applications without forcing the user to have the code converted or rewritten. Nevertheless, like any small VAX, CHARON-VAX also runs compilers and other software development tools and can be used as a small VMS development system on for instance a laptop computer.

Functionality

The hobbyist edition version of CHARON-VAX provides a model of a MicroVAX II system with a CPU, an RQDX3 controller, 16 MB memory, the console interface, a DHV11 with 2 serial lines enabled and all the internal bus and clock components required for a working system model. Unlike the commercial versions, the CHARON-VAX system components in the hobbyist edition are permanently loaded and do not have to be specified at startup. Specific device settings, like the location of a virtual system disk, are specified in a configuration file.

This package is designed to run on Red Hat Linux release 6.0 and higher. Since a PC platform is much faster than MicroVAX systems, you gain system performance with the emulator despite the overhead associated with emulation. A 500 MHz Intel Pentium system provides with this emulator about 3 times the performance of a MicroVAX II, when running VMS 6.2, with disk I/O up to 10 times faster.

Several utilities are included to make it easier to work with CHARON-VAX. For instance, MKDISK provides the tools to transfer user and system files from a MicroVAX.

By default an X-Window System terminal window is used as a display device for the console port, but you can reconnect the port to an external terminal or to a different terminal emulator.

Note: Although nothing prevents you could run multiple copies of the CHARON-VAX emulator simultaneously, your total emulated VAX system performance is not increasing, unless you use a multi-CPU system. (You will also need to start CHARON-VAX with the `-f` parameter to point to different configuration files).

2. Installing and using CHARON-VAX

2.1 CHARON-VAX software installation

CHARON-VAX requires Red Hat Linux release 6.0 or higher with the installed X-Window System, the RPM Package Manager and the gless file viewer.

NOTE: This release CHARON-VAX for Linux was not tested with earlier Red Hat releases or other Linux implementations.

The application should be installed and de-installed using the Linux RPM Package Manager. The installation procedure of the hobbyist edition of CHARON-VAX by the RPM package follows standard Linux installation rules. During setup, the installation procedure installs all software and data in the directory `/usr/charon-vax`. The files existing in the installation directory will be overwritten without asking for confirmation. There is no need to reboot the system after installation but a logout/login should be executed.

You have to start CHARON-VAX from the X Window System terminal window, not from the Linux console.

2.2 Installing your VAX software

Before you can use CHARON-VAX, you must provide a VAX system disk image or bootable VMS CD-ROM from which the emulator can boot. Although in principle any operating system written for VAX hardware can be executed by the emulator, we assume that with this hobbyist manual you will use the VAX/VMS hobbyist CD with the VMS 7.2 installation kit, as is provided through the DECUS program.

CHARON-VAX uses 'virtual' VAX disks, which are represented in the Linux file system as files.

CHARON-VAX for Linux supports direct access a CD-ROM device. CDs in VMS format are readable by CHARON-VAX, which means that VMS can be booted and installed directly from a bootable VMS CD-ROM.

You can install an OpenVMS system by following this procedure:

1. Tell CHARON-VAX where it can find the installation kit, and on which disk you want to install VMS. You configure the system by placing instructions in the *charon.ini* file in the same directory. In this example, you want to associate one virtual disk with the physical CD-ROM drive, and assign a second virtual disk to become your future system disk. The CHARON-VAX installation procedure has already generated an (empty) image of an RD54 with the name *vaxsys.dsk*.

Modify the *charon.ini* file in the same directory as follows:

```
set OPA line="run terminal"
set DUA file[0]="vaxsys.dsk"
set DUA file[1]="/dev/hdd"
#set DUA file[2]=""
#set DUA file[3]=""
```

The # sign tells CHARON-VAX to ignore the text behind it. Make sure that the two lines defining *vaxsys.dsk* and */dev/hdd* are un-commented.

NOTE: CD-ROM device name might be different depending on the type of CD-ROM drive. You can use the following syntax as well:

```
set DUA file[1]="/dev/cdrom"
```

2. Start CHARON-VAX from the Linux X Window System terminal by command "*charons*". The console terminal emulator window (called with the statement `set OPA line="run terminal"`) will open. You will see the MicroVAX Diagnostic countdown: 7..6..5..4..3 and the console prompt `>>>`
3. Boot with command "`BOOT DUA1:`" The standalone boot program start. After you have a \$ prompt issue the following command:

```
BACKUP VMS072.B/SAVE DUA0:/LOG
```

This will restore a VMS installation kit on VAX drive DUA0: which is in effect the disk file *vaxsys.dsk*

4. Restore the CHARON-VAX command window (from which you have started the CHARON-VAX) and stop the emulator by typing "exit". Restart the emulator, boot this time from the disk image *vaxsys.dsk* with the command "BOOT DUA0:" and follow up the OpenVMS7.2 installation procedure.
5. When the installation procedure asks you to specify a drive holding the OpenVMS distribution media enter the name of the second drive, in this case DKA1:
6. Since this version of the emulator does not contain emulated graphics hardware or an Ethernet card, DECWindows, DECNet and DECNet Phase IV should not be installed.

2.3 Starting the emulator

Once CHARON-VAX is installed, you can start it from the Linux X Window System terminal by command "*charons*". At startup, CHARON-VAX will look for the file *charon.ini* in the current directory. If it does not find that file, it will startup in a default configuration, and try to boot from the disk image *vaxsys.dsk* in the same directory.

You can override the default configuration by pointing CHARON-VAX to a different parameter file with the `-f` command:

```
charons -f myvax.ini
```

In this example, CHARON-VAX reads the parameter file *myvax.ini* from the your current directory, where possibly the disk image(s) for system1 are stored as well. If you work with many different system configurations, it is convenient to keep the definition of each system in a separate directory.

Depending on the commands in your configuration file, one or more terminal windows will open at startup. Any error messages will be displayed in the CHARON-VAX command window.

2.4 The CHARON-VAX command interpreter

While the CHARON-VAX emulator is running, its command interpreter continues to accept input and execute valid commands. During execution, you interact with the interpreter by means of commands that you type in the CHARON-VAX console window, until you issue the EXIT command to terminate the CHARON-VAX task. In general, you use the command interpreter to:

- Control the operation of the CHARON-VAX emulator itself. Examples are START, STOP and EXIT.
- Set peripheral device parameters in the emulated MicroVAX system with the SET command. Although you would normally set the parameters in the *charon.ini* file, you can dynamically change device characteristics when you stop the emulated CPU and restart it afterwards. Whether the VAX software can handle these dynamic changes depends on the software and what you do. If you dynamically change a disk image you will likely crash the system.
- Exchange information with the emulator during execution. Examples are the SHOW command.

The interpreter commands are documented in Appendix A.

Note: The CHARON-VAX interpreter commands are different from the console input to the emulated MicroVAX II, which can be entered after the >>> prompt in the console terminal window. These console commands are also emulated, and the >>> prompt will appear as soon as the MicroVAX CPU is loaded and has completed the first part of its internal diagnostics.

3. Emulation and connection of components

At startup, CHARON-VAX assembles a model of a MicroVAX system by loading a MicroVAX CPU, memory and system buses, a disk controller, a console port, a DHV11 and some 'internal' devices like the Time Of Year clock. The CHARON-VAX emulator creates such peripherals in several ways:

1. By mapping to files on the host system. Emulated disk drives work this way. Mapping the (system) disks as files makes it very convenient to archive and restart such systems.
2. By one-to-one mapping to host system peripherals. For instance when a line of the emulated DHV11 is routed to the serial port on a PC.
3. By emulating the functionality completely in software, e.g. the CPU, buses or the console terminal emulator.

Most MicroVAX hardware peripherals have on-board switches or jumpers to customize their settings. In CHARON-VAX, the emulated peripherals have the same settings by means of software parameters. These parameters are specified in a small text file; the default parameter file is charon.ini in the CHARON-VAX directory.

Each configuration parameter has a unique name and gets a value assigned to it. When you change the configuration parameter with the SET command, the value you specify should have a type valid for the parameter. If a device has several units or drives, it is possible to refer to any of them by specifying the corresponding unit number enclosed in the square brackets at the end of the parameter name. For example, to enable write-protection on the first disk drive include the following line in the configuration file:

```
SET DUA WRITEPROTECT[0]=YES
```

When reading from the charon.ini file, CHARON-VAX needs less than one second to assemble a specific MicroVAX configuration and to start the boot sequence of the emulated CPU.

Parameters can also be set with a SET command in the CHARON-VAX command interpreter and are accepted as long as the emulated CPU is in the stopped state. You can change the CPU state with the START and STOP commands. (That does not guarantee that the VAX software you are using can handle dynamically changed parameters like disk images or terminal ports).

Chapter 4 discusses device-specific configuration parameters.

3.1 Emulating VAX disk drives

Instead of using a dedicated disk on your PC, a MicroVAX disk image is associated with a disk file. The collection of one or more disk images and the charon.ini parameter file together define a complete emulated MicroVAX system; such systems can be archived as a collection of files on your PC and restarted in seconds. This simplifies the work with multiple emulated MicroVAX systems.

3.2 Serial line connections

Emulated serial lines can be connected to outside world in several ways. The connection method is specified by **line = <parameter>**. The following parameter forms are currently supported for an emulated serial line (Console port or DHV11 lines):

- “<physical port>” for example “/dev/ttyS0” or “/dev/cua0”.
NOTE: Serial lines driver name might be different in the different HW configuration. In this case the device names “ttyS0” or “cua0” might be different.
- “run <program_name>” this type of line starts a program with path <program_name>. Arguments to the program can be supplied if needed, and are appended to the command line. (Connection is via a local domain socket, named <process_id>-<device_instance_name>-<unit_number>). Using the console terminal emulator is an example of this command. Please take care of the availability of the path to your program by means of the PATH environment variable.
- “link <port>” creates a TCP/IP socket at port number <port>.
- “link <name>” creates a local domain socket with name <name>.

- “**telnet <port>**” is similar to “**link <port>**” but connection is expected to be in the telnet protocol. This means that some characters will be specially interpreted and several will be filtered out.

Note that features like support for modem control, setting speed, etc. are only supported only for connections to physical devices (set line=<device>). All other types of connection are just bi-directional byte streams.

Examples of connection:

```
SET TXA LINE[0]="/dev/ttyS0"
```

```
SET TXA LINE[0]="run <program name>"
```

Program name could be for instance a utility to route serial printer output from the MicroVAX to the system printer.

```
SET TXA LINE[0]="run terminal"
```

The program *terminal* is a standard Linux X Window System terminal (xterm or dxtterm) supporting VT52/VT100 functionality and is NOT part of the CHARON-VAX distribution.

```
SET TXA LINE[0]="link -host <hostname> -port <port_number>"
```

OR

```
SET TXA LINE[0]="link <hostname>:<port_number>"
```

This example associates a serial line with a TCP/IP agent on the specified host, which accepts TCP/IP connection requests on the specified port. This method supports also the connection of a third party terminal emulator.

4. CHARON-VAX components

This chapter describes which emulated MicroVAX components are included in the CHARON-VAX hobbyist edition. The components are pre-loaded at system start, and their parameters can be seen from the CHARON-VAX command window by typing: **SHOW DEVICES**

Most MicroVAX hardware peripherals have on-board switches or jumpers to customize their settings. In CHARON-VAX, the emulated peripherals have the same settings in software and are present to the usual hardware default values.

Some devices have settable parameters, as indicated below. In general, the values can be of one the following types:

- Boolean type. Valid values are YES, NO or 1 and 0. A value of YES or 1 means that the option is enabled, a value NO or 0 means that the option is disabled.
- Character type. Values of this type are character strings enclosed by quotation marks ("string parameter").
- Numeric type. Where numeric data is required, it can be entered in octal, decimal or hexadecimal format:
 - A number starting with 0 is considered octal; use the symbols 0 – 7.
 - A number starting with 1 - 9 is considered decimal.
 - A number starting with 0x is hexadecimal; use 0 - 9 and a – f.

For the SHOW command in the command windows, components (e.g. VAXBUS) should be specified in uppercase. The following components are present:

4.1 The MicroVAX kernel components

4.1.1 The VAXBUS

A VAX CPU can be viewed as connected to two buses. One is the local bus, connecting memory, some peripheral devices and the adapters for the other one, the IO bus. The local bus normally supports 32 or even 34 bit addressing. Each family of VAX hardware uses it's own type of local bus. The buses must be loaded first, so that the emulator can connect other components to them.

For emulation purposes these local buses are quite similar; and in the current MicroVAX implementation a single component is used to emulate these local buses. The emulator component for this bus is called VAXBUS.

4.1.2 The EPRBUS

Another bus defined by the VAX architecture is the bus where all peripheral devices that are accessed via the Internal Processor Registers (IPRs) are placed. This bus is longword addressable and has an 8-bit address. The emulator component for this bus is called EPRBUS.

4.1.3 The peripheral bridge

One of the components of a MicroVAX II system is the QBUS to local bus adapter. This bridge contains the scatter-gather registers to map the QBUS peripherals to the internal buses, the inter-processor communication register (IPCR) and the control register. The bridge emulator connects both to the VAXBUS and the EPRBUS. On the VAXBUS it defines the address windows for IPCR, scatter-gather registers, QBUS IO page address space and Q-BUS address space. On the EPRBUS it defines the address window for the control register. The emulator component for this bus is called QBUS.

4.1.4 The Central Processing Unit (CPU)

The hobbyist release of CHARON-VAX emulates the standard MicroVAX II CPU. This version of CHARON-VAX does not support multi CPU configurations.

Note that CHARON-VAX does not implement the PDP-11 emulation mode, since that mode is available in its companion product CHARON-11.

4.1.5 The Memory Management Unit (MMU)

The memory emulation component VAXRAM is by default set to the logical MEM with the maximum size for MicroVAX II systems of 16 MB. With a set command in the charon.ini file the memory size can be set between 1024 and 16384 KB in increments of 1024.

For example: `SET MEM size=8192.`

4.1.6 Time of year clock (TOY)

The MicroVAX contains a Time-Of-Year (TOY) clock to provide the time and date information. CHARON-VAX stores the time as the difference between the value provided by the user and the current host system time. When the TOY Clock is requested for time, it returns the system time plus adjustment. Initially the adjustment is 0, so any change made to CHARON-VAX time and date does not affect host operating system.

4.1.7 The MicroVAX real-time clock

The MicroVAX real-time clock contains an interval timer that posts an interrupt request every 10 ms. Special algorithms are used to perform the correction of timer ticks to synchronize time measured by the emulated clock with the real clock of the operating system. This also requires that the time and date within the operating system must not change subsequently, since it may cause wrong synchronization and change the real frequency of timer ticks.

4.2 The console port

The console serial line port is defined with the logical name OPA. It is typically connected to the CHARON-VAX internal terminal emulator, but can be redirected to a host system port or a telnet session. Via telnet, you can connect a 3th party terminal emulator running in the same host system. Most serial line parameters, like baud rate, can be set but have no effect on its operation.

Command	Parameter	Type	Value
SET OPA	line	String	Connected process or port
Examples	<pre>SET OPA line="/dev/ttyS0" SET OPA line="run terminal" SET OPA line="link -host localhost -port 500"</pre>		

4.3 The disk controller

The emulated system uses the model of an MSCP disk controller type RQDX3, with logical name DUA. The parameters that can be set cover both the emulated hardware device like controller address (all with standard defaults) and options to adjust the characteristics of the emulator.

Command	Parameter	Type	Value
SET DUA	FILE [<unit>]	String	The name of a file with a disk image
	REGISTER	Numeric	Device address
	VECTOR	Numeric	Vector value
	PRIORITY	Numeric	Bus priority
	WRITEPROTECT [<unit>]	Boolean	The write protection flag. The default value is false or write enabled.

	OFFLINE [<unit>]	Boolean	The off-line flag. The default value is false ("no") or online.
	SIZE [<unit>]	Numeric	The size of a disk image in 512-byte blocks
	CACHING [<unit>]	Numeric	Defines whether the drive is cached, where <number> specifies type of caching: 0 - Disable caching (write-through cache) 1 - Use the operating system cache, this is the default value. 2 - Map the file into memory
	SECTOR_SIZE [<unit>]	Numeric	Defines size of each sector for given drive
	DELAY_FACTOR [<unit>]	Numeric	Determines priority of disk I/O operations on the given unit. Instructs CHARON-VAX to postpone I/O operation in the case of a heavily loaded environment. The default is set to 1
Example	<pre>SET DUA file[0]="./cfg1/mscgai.dsk" SET DUA file[1]="user1.dsk" SET DUA writeprotect[1]=YES</pre>		

4.4 Asynchronous serial lines

CHARON-VAX emulates the DHV11 asynchronous 8-line multiplexer (the number of the lines is limited to 2 in the Hobbyist release), and gives it the logical name TXA. The following parameters can be changed with the SET command; it is recommended to leave them at their default value unless a change is needed:

Command	Parameter	Type	Value
SET TXA	LINE [unit]	String	Connected process or port.
	ADDRESS	Numeric	Device addresses in I/O page.
	VECTOR	Numeric	Interrupt Vector address.
	PRIORITY	Numeric	Bus Request level.
	SPEED [<unit>]	String	Number (baud rate).
	PARITY [<unit>]	Odd/ None/ Even	Specifies parity of transmission. Default = None.
	CHARACTER [<unit>]	Number	Specifies character length (5-8). Default = 8
	STOP [<unit>]		Specifies stop bit (1 or 2). Default = 1.

	TTSYNC [<unit>]	Yes/No	Enables XON/XOFF emulation. When enabled, turns on software synchronization and output-flow control on the underlying communication device. Also instructs the serial line emulator to synchronize the line with the VAX operating system. Default = No.
	HSYNC [<unit>]	Yes/No	Enables synchronization of receipt of data. Default = No.
	RX_BUF_SIZE [<unit>]	Number	Receive buffer length. The default is set to 1024.
	TX_BUF_SIZE [<unit>]	Number	Transmit buffer length. The default is set to 1024.
	TX_TRACE_FILE [<unit>]	String	Specifies the name for the trace file where the TX trace of the line will be stored. Also enables the TX trace. Default=null.
	RX_TRACE_FILE [<unit>]	String	Specifies the name for the trace file where the RX trace of the line will be stored. Also enables the RX trace. Default=null.
	IO_TRACE_FILE [<unit>]	String	Specifies the name for the trace file where the IO (access to device registers) trace of the line will be stored. Also enables the IO trace. Default=null.
Examples	SET TXA line[1]="/dev/ttyS1"		

Note that parameters are normally set by the VMS operating system and this continues under CHARON-VAX. The serial lines accept settings for the SPEED, PARITY, CHARACTER, and STOP parameters directly from VMS, and override any special settings for these parameters in the charon.ini file. These parameters are implemented for special purposes.

If a device has an owner (process), only the owner can change the device characteristics such as any control bit of a serial line. When no process owns the device anyone can change the characteristics. For example, a terminal must be logically detached in VMS (user should perform a logout) before the characteristics of the specified terminal line can be changed from the CHARON-VAX console terminal.

4.5 Terminal emulation

Terminal emulator is required for a convenience to avoid the need for a hardware MicroVAX VTxxx type console terminal. By default, CHARON-VAX for Linux uses an X-Window System terminal window as a display device for the console port. You can also reconnect the port to an external terminal or to a third party terminal emulators supporting VT52/VT100 functionality. In this case the path to the terminal you would like to use have to be added at the head of PATH environment variable.

For every association with a communication line a copy of the terminal emulator is started. Example:

```
SET TXA line[0]="run terminal"
```

When many serial lines are mapped to terminal windows, it can be difficult to determine which line corresponds with which window. To facilitate this, the title of each terminal window can be set in the configuration file with the 'unit' parameter:

```
SET OPA line="run terminal" unit="VAX Console"
```

Note: No special characters are allowed in the title string.

APPENDIX A: The Command interpreter

The CHARON-VAX command interpreter parameters are listed below. Where they refer to VAX system components like VAXBUS, these components should be specified in uppercase. Where numeric data is required, it can be entered in octal, decimal or hexadecimal format:

- A number starting with 0 is considered octal; use only the symbols 0 – 7.
- A number starting with 1 - 9 is considered decimal. Use a dot at the end of the number to indicate that the number is decimal.
- A number starting with 0x is hexadecimal; use the symbols 0 - 9 and a – f.

START <value>

Resets all devices and resumes (or starts) execution on the current program counter. The START command is valid only if the CPU is in the stopped state. <value> is the start address of the program counter.

Initially, the emulator is started automatically. After a stop command, use START 0x20040000 to restart the boot commands.

STOP

Instructs CHARON-VAX to stop (suspend) execution (emulation) of instructions. Brings the CPU into the stopped state, where the configuration can be changed.

EXIT

Stops execution (instruction emulation) and exits CHARON-VAX.

RUN <command>

Executes a shell program.

SET <device> <parameter>=<value>

Changes the configuration parameter for the device. The <parameter> argument must refer to a valid parameter name for the device specified by the <device> argument and <value> must be a valid value for this parameter. The SET command is valid only when the CPU is in the stopped state. Example:

```
SET DUA file[1]="rz25.dsk"
```

The SET command in this example specifies an (additional) drive for the DUA device and associates it with the file *rz25.dsk* as media.

HELP

Displays the commands that are available

SHOW [<device>]

The SHOW command is the main command to observe system status. In CHARON-VAX, devices are hierarchically grouped:

At the top level, SHOW DEVICES will display a list of implemented generic devices like SYSTEM, COMMUNICATION, DISK, etc.

SHOW DISK will display all implemented disk subsystems like an RQDX3.

SHOW RQDX3 displays the logical definition given to a loaded device of that type, like DUA. SHOW DUA lists the current configuration parameters of that device in the form of <parameter>=<value>. For devices with have multiple units, like disk subsystems or communication devices, each of the components is listed with the unit number in [].

Note that the <device> names are defined (and must be typed) in UPPER CASE.

About and Ver

Displays the name and version of the program.

APPENDIX B: VMS operation notes

The current implementation of CHARON-VAX does not detect the "idle" process in VMS. This implies that all CPU time in the host system assigned to the CHARON-VAX application used to execute MicroVAX instructions even when VMS is idle, resulting in a 100% CPU load.

However, as CHARON-VAX is a normal application, it will co-exist with other applications (or a copy of itself), with the scheduler handling the load distribution between applications as usual, effectively allowing other applications to operate normally.

Stop the execution of VMS by running the **SHUTDOWN** procedure as usual. You can quickly stop the emulator by typing **EXIT** in the command window, but you risk corruption of your emulated disks if VMS was writing in them at that point in time.

Reader's Comment Form

We appreciate your comments, suggestions, criticism and updates of this manual. You can Email us, or send this form by FAX or postal service to:

Software Resources International S.A.
P.O.Box 156
CH-1228 Plan-les-Ouates Switzerland
vaxinfo@softresint.com

CHARON-VAX hobbyist manual for Windows, November 2000
 Document number: 30-01-002

If you found any errors, please list them with their page number.

Your Name/Title:

Company/Department:

Internet address:

FAX number:

Mailing address:

Phone:
