

AN-010 Using MKIMAGE to write CHARON-VAX disk images

Author Kirill Nikolaev & Vladimir Chugunov Date 19-Oct-2000

The CHARON-VAX emulator uses disk images in the form of files on the host system to represent the disks of an emulated VAX hardware platform. The MKIMAGE utility creates these disk images from a physical device like a distribution CD or a disk on a VAX system. MKIMAGE has two functions:

- Collecting the disk data on the VAX system (MKIMAGE works in *client mode*).
- Generating a file containing the binary disk image (MKIMAGE works in *server mode*).

A CHARON-VAX disk image file has approximately the same size as the original VAX disk, independent of the amount of data stored on it. This implies that on VAX systems with a single disk the image file cannot be stored locally; such images must be directly transferred to another system, for instance the host system running CHARON-VAX. On a VAX system with multiple disks the image file could be stored locally and later transferred via network services or tape. The network transfer time of a disk image depends on the effective transfer rate of the connection and can be several hours depending on the disk size.

To cover the various platforms of CHARON-VAX and to provide the facility to transfer the data over TCP/IP, the MKIMAGE utility is provided in four binary executable versions:

- MKIMAGE-VMS-VAX.EXE, which runs on VAX/VMS systems.
- MKIMAGE-VMS-AXP.EXE, which runs on OpenVMS Alpha systems.
- MKIMAGE-W32-I386.EXE, which runs on Intel compatible Windows systems.
- MKIMAGE-LNX-I386, which runs on Intel compatible Linux systems.

MKIMAGE can directly transfer data through a TCP/IP network connection to a remote system (*client* and *server* part are separate) or generate a disk image on the same system or on a system connected via DECnet (in this case the instance of MKIMAGE is both *client* and *server*). A disk image can for example be transferred from a VAX to an OpenVMS Alpha or PC Windows system via TCP/IP, or an image of a VAX/VMS CD-ROM can be generated locally on an Intel or Alpha system.

MKIMAGE is called from the command line and has two arguments: an input stream and an output stream. Invoked without arguments the MKIMAGE utility displays the usage information. The same MKIMAGE utility executable can work as *client*, *server* or both, depending on the parameters specified in command line. The format is:

```
MKIMAGE if=<input name> of=<output name>
```

If MKIMAGE runs on the VAX/VMS system from which you want to obtain a disk image, <input name> must be the logical name of the physical disk to copy. The <output name> specifies in this case the name of the file that will contain the disk image (it is recommended to use a .disk extension for it), either on the same system or on a remote DECnet node. If <output name> specifies a TCP/IP address (specified as tcpip:host:port) MKIMAGE on the VAX/VMS system switches to *client* mode. You must specify a TCP/IP port number greater than 1024 that is not used by any other applications.

When running MKIMAGE on a CHARON-VAX host platform, <input name> can be the device that will be the source for a disk image, for instance a VAX CD-ROM in drive D: of a Windows system. If <input name> specifies the TCP/IP address and the port number where a MKIMAGE *client* is sending data, MKIMAGE switches to *server* mode. You should specify the same port number for the client and server transmissions.

When running MKIMAGE on a CHARON-VAX host platform, <output name> is the name and location of the resulting disk image.

CHARON-VAX application note

In the following examples it is assumed that symbols are defined in OpenVMS that invoke MKIMAGE, for example in VAX/VMS:

```
$ MKIMAGE ::= "$ USER$DISK:[USERS.CHARON-VAX]MKIMAGE-VMS-VAX.EXE"
```

In OpenVMS/Alpha:

```
$ MKIMAGE ::= "$ USER$DISK:[USERS.CHARON-VAX]MKIMAGE-VMS-AXP.EXE"
```

In order to transfer data from a VAX system you must have one of the following TCP/IP support packages installed on that VAX/VMS system: UCX, TCPware (from PROCESS Software) or Digital TCPIP Services.

Example of generating a CHARON-VAX disk image on another local VAX disk

Assuming that the VAX disk of which an image is to be made is called DKA100: and the result is stored on DKA200: we can use the following sequence:

```
$ MKIMAGE IF=DKA100: OF=DKA200:[000000]MY_DISK.DSK
(C) Copyright 2000, Software Resources International
USER$DISK:[USERS.CHARON-VAX]MKIMAGE-VMS-VAX.EXE;1, utility to copy
data from a physical device to a disk image file
Total transferred: 681574400 bytes
$
```

Example of generating a CHARON-VAX disk image on a Windows TCP/IP node:

To create a TCP/IP connection with the MKIMAGE utility between VAX/VMS and PC/Windows, *first start the server part on the PC* (MKIMAGE-W32-I386.EXE) with the proper parameters, and then the *client* part on VAX. Assuming that the VAX disk is called DKA100: we can use the following sequence:

On the Windows system:

```
MYNOTE> mkimage-w32-i386 if=tcpip:mynote.mydomain:6000 of=my_disk.dsk
(C) Copyright 2000, Software Resources International
Utility to copy data from a physical device to a disk image file
Total transferred: 681574400 bytes
MYNOTE>
```

On the VMS system:

```
$ MKIMAGE IF=DKA100: OF=TCPIP:MYNOTE.MYDOMAIN:6000
(C) Copyright 2000, Software Resources International
USER$DISK:[USERS.CHARON-VAX]MKIMAGE-VMS-VAX.EXE;1, utility to copy
data from a physical device to a disk image file
Total transferred: 681574400 bytes
$
```

Example of generating a VAX disk image on an OpenVMS/Alpha system via TCP/IP:

On the OpenVMS/Alpha system start MKIMAGE in server mode, for example:

```
$ MKIMAGE if=tcpip:192.168.1.2:5000 of=mydisk.dsk
```

NOTE: Since in server mode MKIMAGE is using its local TCP/IP address as part of the input parameter, the host component of the TCP/IP address can be omitted, i.e.

```
$ MKIMAGE if=tcpip:5000 of=mydisk.dsk
```

On the VAX/VMS system start MKIMAGE in client mode, for example:

```
$ MKIMAGE if=DKA100: of=tcpip:192.168.1.2:5000
```

In the above example the *server*, an Alpha system running OpenVMS, creates a CHARON-VAX container file by listening for data on port 5000 of TCP/IP address 192.168.1.2 for a file from the MKIMAGE *client*, which writes the disk DKA100: to the specified port.

Example of copying to a remote DECnet node:

Assuming that the VAX disk is called DKA100: we can use the following sequence:

```
$ MKIMAGE IF=DKA100: OF=MYNODE::MY_DISK.DSK
(C) Copyright 2000, Software Resources International
USER$DISK: [USERS.CHARON-VAX]MKIMAGE-VMS-VAX.EXE;1, utility to copy
data from a physical device to a disk image file
Total transferred: 681574400 bytes
$
```

Example of generating a VMS CD-ROM image on a Windows NT node:

Assuming that the CD-ROM drive is called D: we can use the following sequence:

```
MYNOTE> mkimage-w32-i386 if=\\.D: of=my_disk.dsk
(C) Copyright 2000, Software Resources International
mkimage-w32-i386, utility to copy data from a physical device to a
disk image file
Total transferred: 681574400 bytes
MYNOTE>
```

Example of generating a VMS CD-ROM image on a Linux node:

Assuming that the CD-ROM drive is called /dev/cdrom we can use the following sequence:

```
bash$ mkimage-lnx-i386 if=/dev/cdrom of=my_disk.dsk
(C) Copyright 2000, Software Resources International
mkimage-lnx-i386, utility to copy data from a physical device to a
disk image file
Total transferred: 681574400 bytes
bash$
```

The MKIMAGE utility was tested on:

- VAX VMS 5.5 with UCX 4.1
- VAX VMS 5.5 with TCPWARE 5.4
- OpenVMS/VAX 6.2 with UCX 3.3
- OpenVMS/VAX 6.2 with UCX 4.1
- OpenVMS/VAX 6.2 with TCPWARE 5.4
- OpenVMS/VAX 7.2 with UCX 4.1
- OpenVMS/VAX 7.2 with TCPWARE 5.4
- OpenVMS/VAX 7.2 with DIGITAL TCPIP SERVICES 5.0
- OpenVMS/Alpha 7.1 with DIGITAL TCPIP SERVICES 5.0
- OpenVMS/Alpha 7.2 with DIGITAL TCPIP SERVICES 5.0
- Windows NT 4.0
- Windows 2000
- Windows 95
- Windows 98
- RedHat Linux 6.0 kernel release 2.2.5-15

Note: Reading of physical media (Floppy disk or CD-ROM) is not supported on Windows 95 and 98.

[30-18-010]