Chapter 21

Ultrix

V1B 25th January 1989 Alan Trevennor/Pete Griffin

21.1 The first steps

21.1.1 Read me or suffer

Ultrix is very fussy about upper and lower case. Use all lower case letters, unless instructed to do otherwise.

The FILE SYSTEM is 'fragile', indiscriminate halting or improper shutdown may well cause unrecoverable corruption. (See Section 21.5.2).

FIELD is a powerful account, you can EASILY wreck the software from here, be careful.

Only one version of each file is kept. So THINK before you delete any file.

21.2 Logging into an Ultrix system

When you type something to begin a login sequence, Ultrix looks at what you have typed and from it makes some judgements about your terminal type. If, for example, your input is all in upper case it assumes that you are on a dumb terminal - for instance a teletype. So, in order that Ultrix can 'size' your terminal correctly you should set your terminal into seven bit mode, with even parity - and then type your login identifier in lower case. Unlike VMS, Ultrix is very case sensitive.

The first prompt you get is the "Login:" prompt. Ultrix is asking for your user name. Type it in. Then after a second or two it asks for your password. Don't type in your password until it is asked for, or it will echo on the screen.

Assuming these two items were typed correctly then you will be logged in and the system identifies itself by showing its ident information. Once you are logged into the system you get the system prompt - by default this is a hash symbol "#". This is how it might look (Except that the password won't echo!).

```
Ultrix-32 V3.0 (Rev 64) System #2: Sat Dec 17 14:13:30 WET 1988

                Digital Equipment Corporation
                Nashua, New Hampshire


   login: field
password: service

#
```

Did you spot the use of lower case?

The directory pathname for this account is /field (see Section 21.5.3). During the login sequence the commands contained in the file *.login* will be actioned (see Section 21.2).

In the field account there is a file which describes the exercisers available. The command:

more READMEFIRST will type these instructions a screen full at a time. Press the space bar to see the next screenful or Q to quit.

To logout type CTRL/D, this is sometimes disabled (especially over networks) so try *logout*. If 'logout' fails, because you have "processes active" then issue the command again It is often the case that Ultrix will do something if you insist enough times! See Section 21.7.1.

21.3 Shutting down the system

The general format of the command to shutdown Ultrix is:

/etc/shutdown time 'reason message'

Examples:

/etc/shutdown now 'for maintenance'	(Shutdown now and say its for maintenance)
/etc/shutdown +5 'fit memory upgrade'	(Shutdown in 5 minutes from now)
/etc/shutdown -r now 'Test new version'	(Shutdown now and reboot straight away)
/etc/shutdown -h now	(Shutdown rightaway and halt processor)

The stages in performing a manual shutdown are quite simple:

- Ensure that you are the system super-user (login as root user)
- */etc/umount -a* (unmount all file systems)
- */etc/sync* (Flush all disk buffers from memory)
- *halt* (Ultrix command to halt multiuser)

21.4 Booting Ultrix

To boot Ultrix on a VAX use the console command B, optionally specifying a device name after it. For example:

```
>>>B DUA0
```

The boot flags, contained in general purpose register 5 at boot time have different meanings in Ultrix to those used for VMS. Some examples:

```
>>>B/R5:X0000000 DUA0:

(on an 82x0 or 83x0 series machine) Boots device DUA0: - the
value X specifies which partition to boot the system from. 0
means partition a, 1 means partition b, 2 means partition c,
... 7 means partition h.

>>>B/1 DUA1

Boots drive DUA1 and tells the boot program to prompt
you for the pathname of the file containing the Ultrix
kernel you wish to run.

>>B/2 DUA2

Boots drive DUA2. The startup pauses in single user mode
after Ultrix has been loaded. This is roughly the same
as doing a minimum startup of VMS. Type CTRL/D when the
system is in single user mode to proceed with the rest
of the startup.
```

21.4.1 Booting Steps

— Print "LOADING xxxBOOT"

— Load BOOT program

BOOT FLAGS:

R10	R11
0=hp	0=Autoboot
3=hk	1=Prompt for filename
9=ra	2=Single User
b=rb	3=No sync
e=rl	10=Boot diagnostic supervisor

— Print "BOOT"

— Prompt for device and file if not specified

— Print "DEVICE and FILE", eg :hp(0,0)vmunix

— Load 11750 microcode if necessary.

— Print "TEXT, DATA and BSS information"

— Start vmunix from address zero.

— Print "Ultrix-32", and memory size. On a MicroVAX 2 this will be a few K short of real memory size.

— Allocates memory

— Print "Available memory"

— Print "Buffer information"

— Configures the system

— Print "CPU information"

— Size and Print "NEXUS space"

— The hardware is now running, a lot of software organising takes place depending on the type of boot.

— Print names of daemons (ie system background processes) that are running.

— Start network and print message.

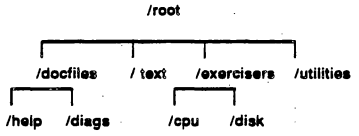
— Finally it prints the date and time, then logins are enabled.

21.5 The File system

Understanding the Ultrix file storage arrangements is the key to understanding Ultrix. The Ultrix file system is a "device transparent environment". This means that, unless they take the trouble to find out, users do not usually know, or need to know, on which device their files reside. For most purposes only the system manager needs to know the physical location of files.

Files are stored in tree structured file systems. In Figure 21–1 you can see an example of a simple Ultrix file system. At the very top of a file system is the / directory. This is always referred to as the root directory. A given Ultrix disk will usually contain more than one file system. File systems are connected together via the */etc/mount* command to form the file heirachy.

Figure 21–1: A simple Ultrix file system



21.5.1 Partitions

When Ultrix initialises a disk it is logically divided into a number of 'partitions'. This simply means that Ultrix zones off the disk into a number of areas which it calls partitions. Partitions are identified by a single letter, partition a, partition b, etc. It depends on the type of disk as to which of these partitions are actually created, and the size of each partition varies between disk types.

Furthermore the system allows you to specify how many partitions you want on the disk and how big they should be. Partition a always exists, and contains crucial information about the rest of the disk. Partition c always covers the whole disk, overlapping all the other partitions. Partitions are simply a convenient way of dividing the storage available on a given disk into easily identifiable chunks. To the system a partition looks just like a file! For example the file */dev/rra0g* refers to partition g on drive zero - which is an RA series drive. Whilst the file */dev/rrd3b* refers to partition b on drive 3 which is an rd series drive.

Inside each partition lives a file system, with a root directory at its top. So on a disk we have a number of separate file systems, each living in its own partition.

When the system is booted (by default from partition a of the system disk) Ultrix starts to build a map of its active file system. The top of the tree is the root directory of the file system in the partition which was booted.

Using the *mount* command the active file system is expanded. File systems within other partitions can be added onto the file tree using the *mount* command.

In this way the active file system can be continually extended in a device independent way. For almost all purposes the actual whereabouts of a file do not need to be known - only its pathname within the file system.

The Network File System (NFS) allows this concept to be taken a stage further, by allowing partitions on disks attached to other nodes on the network to be mounted in exactly the same way as partitions located locally.

partitions can be shared between nodes as well, presenting opportunities for sharing common databases, for example on-line manual pages. To see the partition table currently in use on a particular disk device use the command */etc/chpt -q /dev/rra0a* (but substitute the raw device name (see Table 21–1) of the desired disk for */dev/rra0a*). The separate "file systems" must be mounted either manually (using the */etc/mount* command) or automatically, at boot time. All partitions listed in the file */etc/fstab* the File System TABle are automatically mounted during the system startup.

To dismount a file system, use the */etc/umount* command.

Each File System contains top level directories something like this:-

/	- root directory
/bin	- general commands
/etc	- system manager commands
/dev	- device special files
/sys	- system dependant files
/usr	- userfiles

21.5.2 The Fast File System

The Berkeley fast file system - which is what the Ultrix file system is based upon - is a very clever thing. It optimises the file structure layout for the type of disk and processor taking into account disk rotation speeds and seek times, as well as the speed of interrupt response for the processor. (See the paper on the FFS in the Ultrix supplemnatry documents for more details).

To further assist in fast file i/o Ultrix assigns 10% of the available system memory as a disk cache (In Ultrix V3 onwards the amount is tailorable up to 90%). This cache is known as the Buffer Cache.

Ultrix tries to keep as much as possible of the most recently accessed directory and file information in the Buffer Cache. Information in the buffer cache is written back to the disk when the system is idle, or if the system is busy every 30 seconds. This means that there are times when a version of a block in the buffer cache differs from the disk resident version. If the system is halted without issuance of a *sync* command - which flushes the buffer cache - then the file system can be corrupted.

For this reason it is not wise to halt the system indiscriminately, for example by powering it off whilst it is running. In spite of all this the system needs to be very busy for powering it off to fatally corrupt a disk, and *fsck* can usually recover things during the next boot, but fatal corruption can happen - so better be safe than sorry.

For your information Figure 21–2 shows a simplified overview of the internals of the Ultrix operating system.

21.5.3 PATHNAMES

When you want to specify a file name in Ultrix you provide its PATHNAME. This simply means that you tell the system the route it must take through the file system to arrive at the desired file.

For example, the file name */vmunix* means that the file lives directly below the root directory. The filename */usr/users/myfile* means a file called myfile, which lives in a directory called users, which is a subdirectory of the directory usr, which is itself a subdirectory located under the root directory.

21.5.4 Your current working directory

As with VMS when you log into an Ultrix system you have a default directory. Your default directory is the one from which files will be fetched or created if you don't provide a pathname. The *pwd* (Print Working Directory) command shows the pathname of your current directory. Unless you provide a complete pathname in a file specification, Ultrix assumes that it should take your current working directory as the start point for locating the file.

21.5.4.1 File system related commands

df	Lists the free space, a full partition may hang the software. A console message is normally given if a partition becomes full.
/etc/mount	Describes the current file system.
cat /etc/fstab	Contains a list of partitions to be mounted at system startup time

21.5.4.2 Corrupted file systems

A consistency check can be run on an unmounted file system by invoking the /etc/fsck command. The user can be prompted for an action to cope with an error. This is normally run automatically when rebooting after a system crash.

21.6 Device names and addresses

There is no Auto-Configure mechanism as in VMS, devices should be at their prime addresses.

The distributed version is pre-configured to recognise SOME devices at ASSUMED addresses, this is built upon by the System Manager altering the Configuration File (/sys/conf/system_name).

At boot time a console printout shows a list of the devices found, if this is different to the devices specified in the Configuration file several things could happen:-

- If the devices found are more than expected, then they will not be used.
- If a device has gone missing you may get a sensible message (ie a LP or RP missing) but a missing Nexus could cause the boot sequence to hang (see Section 21.8.)

21.6.1 Device names

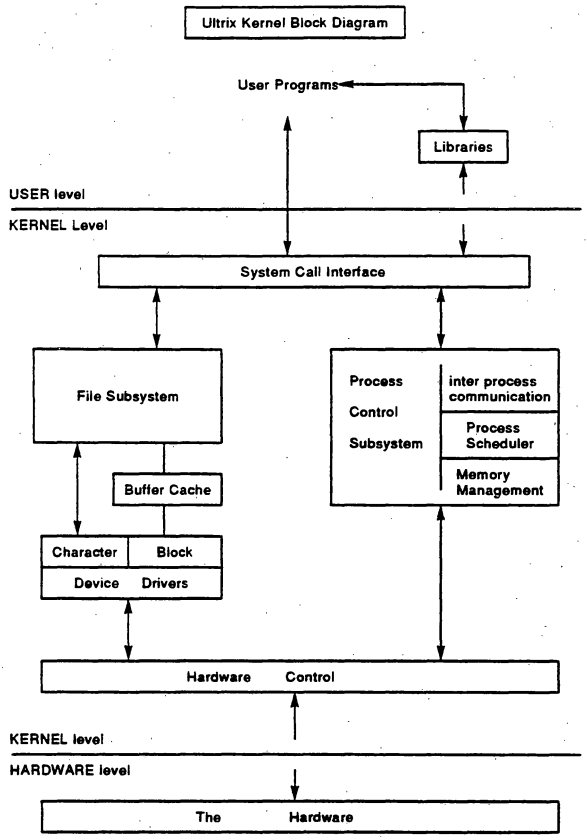
Table 21–1: Device names

TAPES and DISKS	COMMS
ra = RA60, RA80, RA81	de = DEUNA
hk = RK07	dz = DZ11, DZ32
rl = RL02	dmf= DMF32
rb = RL02 or R80 (IDC)	dmc= DMC11
hp = RM03, RM05, RM80, RP06, RP07	css= IMP11A
ht = TE16, TU77	pcl= PCL11
ts = TS11, TU80	
mt = TU78	

21.6.2 Raw devices and character devices

UNIX systems in general, including Ultrix, distinguish between buffered and unbuffered device access. A buffered access is the norm for most puposes but character by character access (known as 'raw' access) is required for low level access - eg file system modifications. For example if you have a disk called ra0 it will be called */dev/ra0* for normal block mode access, and */dev/rra0* for raw access. In general buffered access to block structured devices is via the buffer cache - see the overview of Ultrix in Figure 21–2.

Figure 21–2: Ultrix Internal overview



21.7 Using Ultrix

This section deals with using Ultrix. It concentrates on commands that enable you to find out about the status of the system and its peripherals. Error logging is not covered here - see Section 21.9

You will often hear users refer to "The Shell", this is the Command Line Interpreter. There are many shells available, but the two most common are the Bourne Shell - named after its creator Steve Bourne - and the C shell - so named because it uses the same syntax as the C langauge.

In Ultrix the default shell is the C shell - and this is what most customers use. Another shell called the TSHELL is available in the Ultrix kit. This shell features VMS style command line recall - to any depth. You might like to encourage customers to use it.

The following is a table of command equivalents between the C shell and VMS DCL.

Table 21–2: Common Ultrix equivalents of VMS commands

VMS	Ultrix
APPEND file1 file2	cat file1 >> file2
COPY file1 file2	cp file1 file2
CREATE/DIR [name]	mkdir name
DELETE file	rm file
DEL/CONFIRM file	rm -i file
DIFF filea fileb	diff filea fileb
DIR/FULL	ls -a
DIR file	ls file
DIR [directory]	ls /directory
DIR [*...]file	find / -name file -print
DIR/PROT/DAT/SIZ	ls -al
EDIT file	vi file
HELP command	man command
MAIL	mail
PRINT file	print file
RECALL n	!n ie recall cmmnd by number
RECALL xxxx	!xxxx ie recall cmmnd by name
RENAME filea fileb	mv filea fileb
REPLY/TERM=tt	write tt
REPLY/ALL	wall
SEARCH	grep
SET DEF dev:[dir]	cd pathname
SET DEF sys$login	cd
SET DEF [-]	cd ..
SET HOST	tip
SET TERM	stty
SHOW DEF	pwd
SHOW MEMORY	pstat -s OR vmstat
SHOW NETWORK	netstat
SHOW PROCESS	printenv
SHOW QUEUE lp	lpq OR lpc OR lpd
SHOW QUOTA	quota
SHOW SYSTEM	ps OR pstat
SHOW TERMINAL	stty everything OR who am i
SHOW USERS	who
TYPE file	cat file

Table 21–3: System status and information

/etc/pstat -p	print the active process table
/etc/pstat -i	" inode "
/etc/pstat -x	" text "
/etc/pstat -T	" free slots in system tables
/etc/pstat -s	" swap space usage information
iostat	display I/O rates and system modes
who	print current users and ports
pagesize	show the system memory page size (usually 1024 bytes - 4096 on DECstation 3100)
ps [options]	display information about processes
uptime	shows uptime since booted
vmstat	reports all stats?
/etc/license	shows how many users the system is licensed for

21.7.1 Background jobs

To do a job in background mode, just put a '&' at the end of the command e.g. 'ls -l >listfile &' creates a directory listing in file "listfile".

Use the *ps* to see the background job running. Notice that this shows the Process Identification (pid) of each job. If you want to stop a job you need the pid to kill it. The kill command is of the format *kill -9 pid*. The number after the minus sign is a signal which gets sent to the process, -9 means stop now. You can also signal a process to do a number of other things. Type *man 3 signal* to find out about these.

21.7.2 Prompts

Primary prompt, this is generated by the shell, and can be changed by doing *set prompt="Ultrix>"*.

21.7.3 Redirecting terminal input/output

Standard input, output and error messages use the keyboard and terminal.

The *ls -la* command will produce a full directory listing on the users terminal, to create a file and put the listing in the file instead do *ls -ls > filename*. Use >> if you want to append the output to an existing file.

If a command generates an error message, you can also send the error message to a file e.g. 'ls > filename 2> errorfilename'

All utilities take their input from your keyboard. The idea of using the keyboard is unusual, say, for a sort operations' input but thats the way it is! To sort the contents of a file you must redefine the input source e.g. 'sort < filename'.

21.7.4 Line Printers

/etc/lpq	will show the print queue
/etc/lpq -l	will give more information
/etc/lpq -P name	can specify a printer queue name
/etc/lpc status	asks Line Printer Control program for status
/etc/lpc stop [printer name]	stop spooling after this print
/etc/lpc start[printer name]	restart it

21.7.5 Terminal Lines

cat /etc/ttys	show default set up of each line
cat /etc/ttytype	show default terminal type on each line

21.8 Maintenance issues

In this section miscellaneous issues about maintaining Ultrix systems will be covered

21.8.1 Crashing a system

If you have a system which is hung, and you want to make it crash, and perform a dump/restart the following console mode commands will usually do this for you.

```
HALT
EXAM PSL                    ! To see if Kernel Mode (D24:25=00)
                            ! Set Kernel Mode if necessary
DEP PC 80000000
CONT
```

21.8.2 PANIC

When a 'panic' message is printed at the console, copies of the memory and kernel are dumped to the swap area. When it reboots, the /etc/rc.local script will execute the 'savecore' program which copies the dumps to the specified directory (usually /usr/adm/crash) as vmcore.? and vmunix.? (The ? is an incremented value). A message is also logged in /usr/adm/shutdown. The distributed copy of Ultrix has the 'savecore' line commented out of /etc/rc.local, this will need to be put back in to preserve these copies.

Analysing system crashes is not as easy as for VMS, usually it requires access to the Ultrix sources. Therefore the usual practice - if the panic message does not make clear what caused the crash - is to send a dump to the TSC for them to look at.

21.9 Ultrix Error logging

21.9.1 Error Logging prior to Ultrix-32 Version 2.0

Prior to Ultrix version 2 the method of error logging was to place messages (with very little content) into the */adm/messages* file. From Version 2.0 onward full error logging is available. (See Section 21.9.2).

The following is a quick guide to examining this file in various ways (for Ultrix V1.2) and may also be used on Berkeley UNIX (aka BSD UNIX) systems.

cat /usr/adm/messages	type the file on your terminal
tail -n /usr/adm/messages	type the last n lines on your terminal
lpr /usr/adm/messages	list the file on the printer
more -d +/1988/usr/adm/messages	type the file starting at the first occurance of '1988'
view /usr/adm/messages	run the editor on the file as read only
tail -f /usr/adm/messages	will type the errors as they are processed through the errorlog file buffer on their way to the disk file

21.9.2 Error Logging in Ultrix V2.0 - and after

For a quick guide to using the Ultrix error log report formatter refer to section Section 21.9.3.3.

21.9.2.1 Components of the error logging system

The components of Ultrix which are capable of detecting hardware errors are principally the device drivers, but the fault detection features built into VAX hardware can also cause an error to be logged by means of an interrupt. An example of this is a hardware detected memory error.

21.9.2.2 Error logging system overview

Figure 21–3 shows the general overview of error logging under Ultrix. The device drivers and the Ultrix kernel place information about errors into the Kernel Error Log buffers. The error logging daemon - elcsd - receives error information and places it into a file which is named in its configuration file - *elscd.conf*. The *eli* utility (see Section 21.9.4) controls the error logging daemon. Finally the *uerf* utility produces formatted reports of errors.

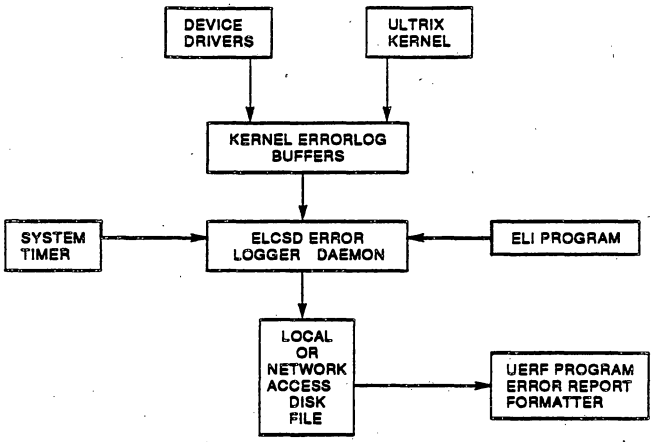
21.9.3 Obtaining error log reports

21.9.3.1 The uerf program

The program used to read the binary contents of the error log file and turn it into a report is called uerf (Ultrix Error Report Formatter). You do not need to be superuser to use uerf. By default uerf lives in:

```
/etc/uerf
```

Figure 21–3: Ultrix Error Logging overview



The command:

```
/etc/uerf -h
```

provides a summary of the options available for use with uerf. Also in the /etc directory are three supporting files. These are:

```
/etc/uerf.bin          # database of event information
/etc/uerf.hlp          # a text file giving brief help on uerf
/etc/uerf.err          # uerf's private error log.
```

If required, you can move uerf and the three associated files to another place on your file system and still have it work successfully, because it uses the following search list to locate its supporting files:

1. The directory where you ran uerf from
2. The /etc directory.
3. The pathname specified in your shell PATH variable. (See the description of the *setenv* command in section 8 of the Ultrix manual set).

21.9.3.2 Running uerf

uerf is invoked by typing:

```
/etc/uerf
```

and then you get the brief format report output to your terminal. If you are on a softcopy terminal then use the command:

```
/etc/uerf | more
```

to see the output a screen at a time. Often the default report obtained using these commands will be sufficient to give an idea of what has gone wrong. But what if your error log file is full of tape retry logs from yesterday, and you now want to see the details of an uncorrectable memory error which has occured today? This is where the numerous options available with uerf are used.

Below is a quick reference to the options you can use with uerf, and following that are some annotated examples.

21.9.3.3 uerf command options

```
/etc/uerf       -x              # negate all other supplied options
                                # EXAMPLE:    /etc/uerf -D -x
                                # means display all errors except disks

                -R              # Display newest errors first
                -n              # Display errors on the console immediately
                                # they occur.

                -h              # show available uerf options

                -H hostname     # Display errors from node hostname
                                # EXAMPLE:     /etc/uerf -H PERCY
                                # Means display all errors which have been
                                # logged which originated from node PERCY.

                -A              # Include hardware adapter errors
                   aie          # BVP controller (BI disk controller).
                   aio          # BVP controller (BI disk controller).
                   bla          # BI LESI adaptor
                   bua          # BI to UNIBUS adaptor
                   nmi          # Nautilus memory interconnect
                   uba          # VAX UNIBUS adaptor
                                # EXAMPLE:    /etc/uerf -A bua,bla
                                # means display all bua and bla errors.

                -c              # Select specific classes of errors
                   err          # Selects error reports from h/w or s/w
                  oper          # Selects only operational events

                -D              # Includes disk errors.
                   rd54         # Include disk errors on RD54 type drives
                   ra80         # Include disk errors on RA80 type drives
                   ra60         # Include disk errors on RA60 type drives
                   etc          # Any other DSA compatible drive type
                                # EXAMPLE:    /etc/uerf -D ra60
                                # means display all errors logged on ra60's.
                                # EXAMPLE:    /etc/uerf -D -o full
                                # means display all disk errors in full format

                -f filename     # Allows you to display errors from any
                                # current or archived error log file.

                -M              # requests that only errors to do with the
                   cpu          # central hardware elements of the machine
                   mem          # be displayed. The memory and the CPU..
                                # EXAMPLE:    /etc/uerf -M mem -o full
                                # means display all memory errors in full

                -o              # Selects output format. brief is the
                   terse        # default. Beware non-selective use of
                   brief        # full, it can generate a long listing!
                   full         #
```

```
-O              # Operating system detected errors.
   aef          # Arithmetic exceptions (can be h/w or s/w)
   ast          # async trap exception faults (h/w or s/w)
   bpt          # Break point instruction fault (often h/w)
   cmp          # Compatibility mode faults (h/w possibly s/w)
   pag          # Page fault exceptions (h/w)
   pif          # Privileged instruction faults (h/w or s/w)
   pro          # protection faults (h/w or s/w)
   ptf          # Page table faults (h/w or corrupt s/w)
   raf          # Reserved address Faults (h/w or corrupt s/w)
   rof          # Reserved Operand Faults (corrupt s/w usually)
   scf          # System Service Call Exception faults (either)
   seg          # Segmentation Faults (h/w usually)
   tra          # Trace exception faults (h/w or s/w)
   xfc          # Xfc instruction failed (h/w)
                # EXAMPLE:   /etc/uerf -O seg -o full
                # Means display full details of all
                # segmentation faults.

-r n            # Selects errors by record type - see manual.

-s nn-nn        # Each error is assigned a number when it
                # is logged. This option selects a specific
                # number or range of numbers to be reported

-S              # Give a summary of all errors.

-T              # Includes tape errors in report
   tk50         # Include errors which occurred on tk50
   tu81         # Include errors which occurred on tu81
    etc         # Any other Digital tape drive type

-t              # Selects by absolute or range of time.
```

NL
District

F A C T F L A S H

Options Affected: DMV11
Submitted By: Barry Lowry
Date: 31-JAN-1989
Filing Instructions: File with your other comms flashes or in your minireference manual.

DMV11 Switches

If you have ever configured a DMV11 you may have found that the minireference manuals are not clear. This factflash should be used in conjunction with the minireference guide and will make things clearer. (I hope!)

The address and vector switch settings are clear.

The DDCMP address register is only used when the DMV11 is configured in a multipoint configuration. I have not seen any of these in the Welwyn District so you should not have to worry about them.

E107 on an M8064 module, or E101 on an M8053 module:

SWITCH 1: This should be off to allow switches 6,7 and 8 to define the mode.

SWITCH 2: This defines the unit number if the system is being booted via the DMV11 and there are two DMV11's on the system. This is unusual and unlikely to bother you.

SWITCH 3: This should normally be off. It makes the DMV11 assert DTR for use by the modem.

SWITCH 4: This should normally be on. If its set off then the DMV11 will try to request a boot from the far end.

SWITCH 5: This should normally be on. If its off then the DMV11 will let the system at the far end try to download it.

SWITCHES 6,7 and 8: The minireference guides are clear. Select half or full duplex depending the modems you have, or to match the far end.. Only select DMC11 compatible mode if there is a DMC11 installed at the far end. The other settings are for multipoint configurations.

SWITCH 9: This should be on for line speeds of less than 19.2k.

SWITCH 10: This should be on for V.35 and off for RS232 or RS423.

The RS232 interface panel switches should be S4,S8,S11,S14,S18 on, rest off. Do not put S7 on as per the minireference manual.

21.9.3.4 uerf command examples.

```
/etc/uerf -T tk50 -t s:01-jan-1989,00:00 e:05-feb-1989,23:59
                # Display all tk50 errors logged between midnight
                # 1st January 1989 and one minute before midnight
                # on 5th Febuary 1989. (If either time specifier
                # omitted with the -t option defaults to today).

/etc/uerf -D -t s: e:
                # Display all disk errors logged today in default
                # brief format.

/etc/uerf -M cpu
                # Display all errors detected on the VAX Central
                # Processing Unit.

/etc/uerf -O cmp,seg -t s:01-dec-1988 -o full
                # Display full details of all operating system detected
                # compatibility mode or segmentation errors which have
                # been logged between the 1st December 1988 and today.

/etc/uerf -s 233-299 -o terse
                # display terse details of errors whose sequence numbers
                # are between 233 and 299
```

21.9.4 The eli utility

The /etc/eli program provides a user interface to the error logger, allowing the system manager to change the error logging parameters during time sharing. Its most frequent use is to restart error logging after changing the contents of the configuration file. (see Section 21.9.2.2) To do this type:

```
/etc/eli -a
```

which stops elcsd and restarts it, thus forcing it to read the new configuration parameters. The most useful of the other options are shown below, use the -h option to obtain a full list.

```
/etc/eli -d     # Disables error logging by stopping elcsd
/etc/eli -e     # Enables error logging when the system is in multi user
                # mode.
/etc/eli -i     # Flushes the kernel error log buffer
/etc/eli -n     # Disables error logging to disk.
/etc/eli -s     # Enables single user mode error logging
```