

DIRECTORIES

CREATING A DIRECTORY

`mkdir directory` Make a new directory called *directory*.

CHANGING DIRECTORIES

`cd directory` Change to the directory, *directory*.
`cd ~` Change to your home directory.
`cd ~/directory` Change to the directory, *directory*, which is in your home directory.
`cd ~userid` Change to another user's home directory.
`cd ..` Move one directory level up.

REMOVING A DIRECTORY

`rmdir directory` Delete a directory, which must be empty.

LISTING THE CURRENT DIRECTORY

`pwd` "Print the Working Directory," which means display the full path name of the directory you are currently in.

FILES

LISTING FILES

`ls` List the files in the current directory.
`ls -F` Directories are displayed with a trailing `/`. Executable files are displayed with a trailing `*`. Links are displayed with a trailing `@`.
`ls -l` Long listing of files, shows ownership, permissions, and links.
`ls -t` List file chronologically- newest to oldest

COPYING FILES

`cp file1 file2` Copy *file1* into *file2*.
`cp file1 directory1` Copy *file1* into the directory *directory1*.

MOVING FILES

`mv file1 file2` Rename *file1* to *file2*.
`mv file1 directory1` Moves *file1* into the directory *directory1*.

REMOVING FILES

`rm file1` Delete the file *file1*.
`rm *` Remove all files in the current directory.
`rm -i *` Prompts you before deleting each file.
`rm -f *` Removes all files without prompting.
`rm -r *` Removes all files and subdirectories in the current directory.

VIEWING FILES

`cat file1` Display *file1* to the screen.
`head file1` Display first few lines of *file1* to the screen.
`tail file1` Display the last few lines of *file1* to the screen.
`more file1` Interactively displays file on screen. Type `q` to quit more. Type `f` to move forward one screen, `b` to move backward one screen. Type `h` for help using more.

EDITING FILES

`pico file1` A simple text editor.
`axe file1` A graphical text editor. Needs X-windows.
`emacs file1` A more complicated and powerful text editor. Good for programming.
`vi file1` A really complicated text editor.

PRINTING FILES

`setenv PRINTER printer`
Set the default printer location to *printer*.
System default is `docuprint`.

Available printers are:

`docuprint` Xerox Docuprint laser printer with blue highlight color in Mudd. (default)
`vetpl` Line printer in Mudd. (ascii)

`ssclps` Postscript printer in SSCL (Sewall 101).
`sewall` Line printer in SSCL.
`lpr file1` Print *file1* to default location
`lpr -Pprinter file1` Print *file1* to location specified by *printer*.
`lpq -Pprinter` Check status of files sent to the printer location specified by *printer*.
`lprm -Pprinter` Cancel the printing of files sent to printer location specified by *printer*.
`/usr/sbin/lsvirprt` Lists all printers.

SAS

`sas` Run interactive SAS, requires X-windows.
`sas -dms -fsdevice ascii`
Run an interactive version of SAS on a telnet or terminal connection.
`sas sasfile` Run SAS in batch mode. SAS program is in the file *sasfile.sas*.
`sas sasfile &` Run SAS in batch mode and background process. This will immediately return you to the vet prompt and allow you to do other things while the SAS program runs. You can `log off` and the program will run until finished. If you are still logged on when the SAS program finishes you will see a message indicating that the program is done.

Other useful options when running SAS in batch mode are:

`-log filename` Change the name of the SAS log file.
`-print filename` Change the name of the SAS listing file.
`-linesize #` Change the number of columns in the SAS output files.
`-pagesize #` Change the number of rows in the SAS output files.

`sas -linesize 79 -pagesize 60 -log saslog -listing saslist sasfile`

Run SAS in batch mode. Save listing output in file *saslist*. Save log output in file *saslog*. Create output files *saslist* and *saslog 79* columns wide and 60 lines per page.

SPSS

`spss -m spssfile` Run SPSS in batch mode. SPSS program is in the file *spssfile*. Output is sent to screen.

`spss -m spssfile &`
Run SPSS in batch mode and background the process.

`spss -t spsslist -m spssfile`
Run SPSS in batch mode and save output in file *spsslist*. Output is also sent to screen.

`spss -t spsslist -m spssfile > /dev/null`
Run SPSS in batch mode, save output in the file *spsslist* and do not send output to the screen.

PROGRAMMING

`xlf [Options] file1.f file2.f`
FORTRAN compiler. Compile code in files *file1.f* and *file2.f*. Any number of files containing FORTRAN code can be compiled.

`xlc [Options] file1.c file2.c`
C compiler. Compile code in files *file1.c* and *file2.c*. Any number of files containing C code can be compiled.

Options:
`-o filename` Name executable file resulting from compilation *filename*. Default is *a.out*.
`-l library` Include a library in program compilation.

TAPES

For detailed information about working with tapes, see the document *Vet 2, Using Tapes on Vet*.

TAPE MANAGEMENT

`aixtape` Begins interactive tape management system. Requires X-windows.

`aixtape -v volser mount`
Mount a tape. The first tape is 3490a.

`aixtape -v volser -t 3490b mount`
Mount a second tape. This tape is 3490b.

`aixtape -v volser dismount`
Unmount a tape.

TAPE NAMES

`~/tape/3490a` Rewind device name for tape 3490a. The tape will be rewound after a tape command specifying this device name.

`~/tape/3490a.norewind`
No-rewind device name for tape 3490a. The tape will not be rewound after a tape command using this device name.

`~/tape/3490b` Rewind device name for tape 3490b. Typically the second tape mounted.

`~/tape/3490b.norewind`
No-rewind device name for tape 3490b. Typically the second tape mounted.

TAPE TO TAPE COPY

`tcopy ~/tape/3490a ~/tape/3490b`
Copy tape 3490a to tape 3490b.

THE TAR COMMAND

`tar -c -f ~/tape/3490a file1 file2`
Create a tar archive on tape 3490a including files *file1* and *file2*.

`tar -x -f ~/tape/3490a file1 file2`
Extract files, *file1* and *file2*, from the tar archive on tape 3490a.

`tar -c -v -f ~/tape/3490a *`
Create a tar archive on tape 3490a including all files and subdirectories in the current directory.

`tar -x -v -f ~/tape/3490a`
Extract all files in the tar archive on tape 3490a into the current directory.

THE DD COMMAND

`dd if=file1 of=~/tape/3490a bs=126b`
Write the file *file1* to tape 3490a using a block size of 126b.

`dd if=~/tape/3490a of=file1 bs=126b`
Read a file off tape 3490a using a block size of 126b and save it in a file called *file1*.

`dd if=file1 of=~/tape/3490a.norewind bs=126b`
`dd if=file2 of=~/tape/3490a.norewind bs=126b`
Write files *file1* and *file2* to tape 3490a and do not rewind the tape.

THE MT COMMAND

`mt -f ~/tape/3490a rewind`
Rewind the tape 3490a.

`mt -f ~/tape/3490a.norewind fsf 1`
Move the tape 3490a forward 1 file and do not rewind the tape.

`mt -f ~/tape/3490a.norewind bsf 2`
Move the tape 3490a backward 2 files and do not rewind the tape.