

Networks and Operating Systems Practical 1

Craig Duffy 2/10/98, modified Nigel Gunton 10/09/02

Module: Computer Networks and Operating Systems
Courses: CRTS, Computing, Computer Science, Software Engineering
Prerequisites: Knowledge of UNIX

Background

This worksheet should introduce a number of features about the UNIX file system and how it is implemented. The first part of the worksheet concentrates on UNIX commands, the second part on C programs to manipulate the file system.

If the machine is not switched on or running Windows NT4.0 then you should either switch the machine on or press ctrl alt del and reboot the machine. After the reboot sequence you should get a lilo prompt, at this select linux and press return, after a lot of boot information you should get a login prompt. You should use the netlabs id and password. You can try running the X-windows interface with the command startx, although it is not strictly necessary to do any of the work in the work sheet.

Exploring and programming the UNIX file system

A large amount of information can be discovered about the structure and implementation of the UNIX file system from fairly mundane commands. The ls command is a good example. Normally it is just used to report on files and file sizes in the current or specified directory, however with certain flag settings it is possible to find out quite a lot more - there are 24 separate flags for ls!

Using the ls command on a system directory /usr or /usr/local you'll find that the print out can be quite complicated, as this print out from the CEMS faculty's server milly shows

```
cduffy@milly [35] ls -l
ls -l
total 122
drwxr-xr-x  3 root    other      512 Mar 10  1997 Browsers
drwxr-xr-x  2 root    other      512 Mar 10  1997 Fonts
drwxr-xr-x  6 root    other      512 Mar 10  1997 Reader
drwxr-xr-x  2 root    other     15872 Aug 29  1996 adainclude
drwxr-xr-x  2 proto13 11         6656 Aug 20 16:17 bin
drwxr-sr-x  2 root    staff      512 Dec 18  1997 doc
drwxr-xr-x  4 proto13 11         512 May 22 11:51 etc
drwxr-sr-x  2 root    staff      512 Sep 22 05:42 find
drwxr-xr-x  3 root    other      512 Jun  2 16:41 flexlm
drwxr-xr-x  5 root    other      512 Oct 15  1996 gnu
drwxr-sr-x  3 root    staff      512 Jan 14  1998 hylafax
drwxr-xr-x  7 root    other      512 Feb 24  1997 i386-unknown-linux
drwxr-xr-x  8 proto13 11        1024 Jul  2 16:37 include
```

```
drwxr-xr-x  2 root    other    4096 Jul  2 16:37 info
drwxr-xr-x 32 proto13 11      2560 Aug 12 11:48 lib
drwxr-xr-x  3 root    other    512  Jul 31 1997 libexec
drwxr-xr-x  2 root    other   8192  Aug  7 1996 lost+found
drwxr-xr-x  7 root    other    512  Jan 22 1997 m68k-sun-sunos4
drwxr-xr-x  9 proto13 11      512  Jan  9 1997 man
lrwxrwxrwx  1 root    other     7  Jul  1 1997 maple -> mapleV4
drwxr-xr-x 15 root    other    512  Apr 17 1997 mapleV4
drwxr-sr-x  8 root    staff   512  Aug  8 1997 netscape
drwx--l---  3 root    staff   512  May 11 11:19 samba
drwxr-xr-x  2 root    other    512  Aug 14 1996/sbin
drwxr-xr-x  6 root    other    512  Nov 10 1997/share
lrwxrwxrwx  1 root    other   33  Jul  1 1997 sparc-sun-solaris2.4 ->/usr/
drwxr-xr-x  5 root    other    512  Aug 13 1996 sparc-sun-solaris2.5.1
drwxr-xr-x  3 root    other    512  Dec 20 1996 tclX
drwxr-xr-x  3 root    other    512  Dec 20 1996 tkX
drwxr-sr-x  2 root    staff   512  Jul 31 1997 var
cduffy@milly [36]
```

Use the manual page for ls to figure out what each of the columns in the ls output means.

Create a sub-directory called file_test, in it create a file with some text in it. Using the commands ls with the -il options and the commands mv, cp, and ln check out that you understand what these commands do and what effect they have on the underlying file system. In particular, explain the differences between the mv, cp and ln commands. Why does the inode number (inumber) change after a cp command and not with mv and ln? After an ln command alter the data in the file and look at what has happened to the links. What do you think will happen to the links if you delete the original file? You need to use the ls -l command to see what has changed. You could also try using the stat command, which gives a more detailed description of the file's inode.

Using the df command identify that different file systems there are on your system. What discs are there and how have they been partitioned? Use the ln command to link two files on different file systems. NB you may need to link to an existing file on another file system because of permissions problems. Why do you think that ln needs a special option for linking across file systems?

The file systems displayed by the df command were created by the system administrator, probably when the system was installed. The command to make them is called mkfs (make file system). After a disc has been formatted it then needs a file system built upon it, in this process the system administrator can set various parameters, such as the file system size, no of files permitted and so on. All UNIX systems require, at least, a root (/) file system, the other file systems being mounted on various directories using the mount command. This process is done during boot up and specified in the file /etc/fstab (file system table). Linux permits the mounting of lots of different file systems, for example MSDOS ones.

As your system has a floppy drive it is possible for you to mount your own file system. Using the manual entry for mount, plus the knowledge that the floppy drive is /dev/fd0, mount your floppy into the machine's file system. In order to do this you will need a MSDOS formatted floppy disc - don't do it on a disc that holds useful/important information as it will be lost with some of the activities we will do later on. Once you have done this you could then create a file system on this disc - it is currently mounting the MSDOS FAT file system. I want you to create a minix file system on the disk using the mkfs command, in order to do this you will need to unmount the disc using the umount command. You will need to reference both the mkfs and minix mkfs- man pages, in order to know which parameters to pass for the minix file system. Try varying the number of inodes on the system, you could even verify this by repeatedly creating files on the new system until it runs out of inumbers. After creating the file system on the drive, mount it into /mnt/flop, you can verify

what you have done by using the `df` command, you can also run a file system consistency check, `fsck`, with the `-t` option set to `minix`, to check for corruption, although the file system will have to be unmounted, using the `umount` command, first. The output from `fsck` is likely to be a bit boring on a non-corrupt, empty file system.

Points for discussion.

- Why can't UNIX/Linux have a useful utility like `undelete`?
- Why should file systems require a utility like `fsck`?
- The `mkfs` command allows the system administrator the opportunity to vary the number of inodes in a file system, why is this useful?