Edinburgh Napier UNIVERSITY

CSN09101
Networked Services
Week 3 : Users, Permissions, Processes, and Pipes

Module Leader: Dr Gordon Russell
Lecturers: G. Russell

Edinburgh Napier UNIVERSITY

This lecture

- Users
- File permissions
- Processes
- Hard and soft links

Edinburgh Napier
UNIVERSITY

USERS

Edinburgh Napier UNIVERSITY

UID and GID

- In Unix, there are User Ids and Group Ids.
- User Ids uniquely identify a particular user.
- Group Ids allow users to be collected into groupings.
- Groups could be used to allow friends to share files, while stopping people not in that “group” of friends from reading the files.

Edinburgh Napier UNIVERSITY

Users

- User details are stored in 4 files.
  - /etc/passwd - General User details.
  - /etc/shadow - User passwords.
  - /etc/group - The user's groups.
  - /etc/gshadow - Passwords for groups.

Edinburgh Napier UNIVERSITY

> cat /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
...
```

- Username, *x*, uid, gid, text name, home directory, login shell.

Edinburgh Napier UNIVERSITY

> head –3 /etc/shadow

```
root:$1$RcFla0lb$bwl5dvTECg3M1ZgMQ7e6l.:12663:0:99999:7:::
bin:*:12621:0:99999:7:::
daemon:*:12621:0:99999:7:::
```

- Passwords are md5 encrypted.
- Shadow passwords can expire and have rules.

Edinburgh Napier UNIVERSITY

> tail –3 /etc/group

```
gdm:x:42:
dovecot:x:97:
mysql:x:27:
```

- Group contains group names, x, and the number which defines that group uniquely.
- After the last : can be a list of users who are in that group.

```
Friends:x:500:gordon,andrew
```

Edinburgh Napier UNIVERSITY

> tail –3 /etc/gshadow

```
gdm:x::
dovecot:x::
mysql:x::
```

- Allows people to change groups on a password.
- Not often used, but when done the password is placed here where the ‘x’ is.

Edinburgh Napier
UNIVERSITY

PERMISSIONS

Edinburgh Napier UNIVERSITY

Permissions

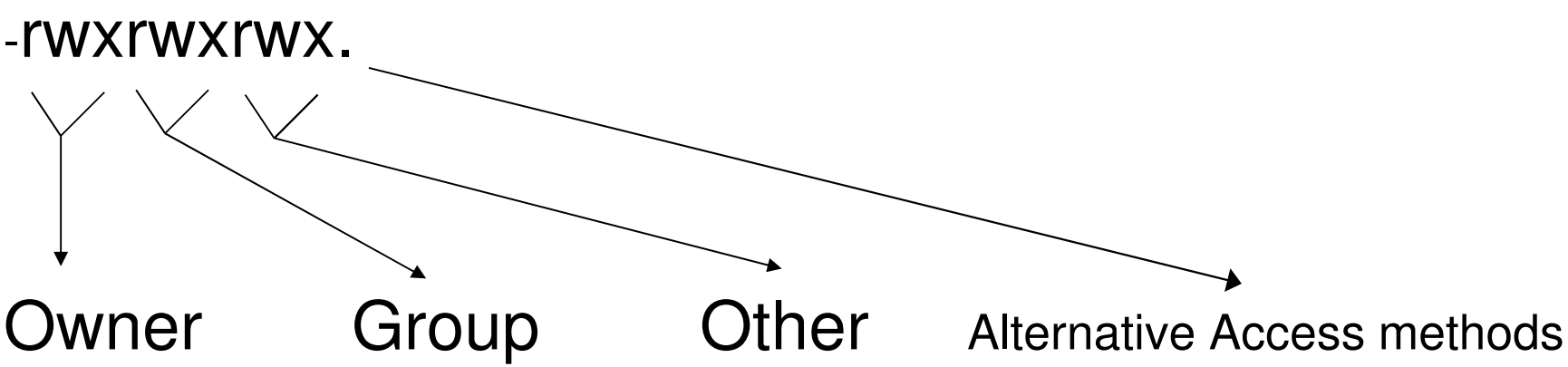
- A file or directory has various permissions and ownerships applied to it.
- Three file permissions:
  - r – read permission
  - w – write permission
  - x – execute permission
- Three permission levels:
  - u – User (the creator of the object)
  - g – Group (a group identifier)
  - o – Other (everyone not in the User or Group specified)

Edinburgh Napier UNIVERSITY

> ls –l /etc/passwd

```
-rw-r--r--.  1 root root 1639 Sep 14 14:38 /etc/passwd
```

- Owned by root, with group root.
- 1639 bytes in size.
- Created on Sep 14th at 14:38.
- 1 link.
- rw by user root
- r by group root
- r by other



- The first character indicates the type of the object.

Edinburgh Napier UNIVERSITY

File types

- - means normal file
- d means directory
- c means a character device (mouse, keyboard)
- b means a block device (ide disk, scsi disk)
- There are more types to discover!

Edinburgh Napier UNIVERSITY

> ls –ld /home

drwxr-xr-x. 2 root root 4096 Jul 27 13:38 /home

- /home is a directory
- Owned by root in group root.
- UID root can do anything, group root can rx
- All others can rx.
- Size is not really useful for directories.

```
> touch /tmp/test
> ls -l /tmp/test
-rw-r--r--.  1 root root 0 Sep 23 15:47 /tmp/test
> chmod og+wx /tmp/test
> ls -l /tmp/test
-rw-rwxrwx.  1 root root 0 Sep 23 15:47 /tmp/test
> chown ftp.mem /tmp/test
> ls -l /tmp/test
-rw-rwxrwx.  1 ftp mem 0 Sep 23 15:47 /tmp/test
> chgrp root /tmp/test
> ls -l /tmp/test
-rw-rwxrwx.  1 ftp root 0 Sep 23 15:47 /tmp/test
```

Edinburgh Napier UNIVERSITY

> ls –ld /home

drwxr-xr-x. 2 root root 4096 Jul 27 13:38 /home

- The “.” immediately after the permissions indicates that alternative access methods exist.
- If this is a “ “ (space) there are no additional methods.
- “.” (dot) indicates a SELinux security context
- “+” (plus) indicates a combination of access methods.

Edinburgh Napier UNIVERSITY

Alternative Access Methods

- Cover this in more detail in a later lecture.
- ACL access methods allow you to set fine-grained permissions:

```
> touch test
> setfacl –m user:root:rwx test
> ls -l test
-rw-rw-r--+ 1 gordon gordon 0 Aug 30 15:25 test
> getfacl test
    user::rw-
    user:root:rwx
    group::rw-
    mask::rw-
    other::r--
```

- SELinux access methods map complex process rules to file context information, e.g. The web server can only see files in the “httpd_user_context_t” context.

Edinburgh Napier UNIVERSITY

Numeric Notation

- An older way of looking at permissions.
- Still needed for some commands, and a fast way of changing multiple permissions.
- Based on octal, 4 digits long.
- Digit 0 is usually 0, 1 is OWNER, 2 GROUP, 3 OTHER.
- Values:

Octal	Binary	Perms	Octal	Binary	Perms
7	111	rwx	3	011	-wx
6	110	rw-	2	010	-w-
5	101	r-x	1	001	--x
4	100	r--	0	000	---

Edinburgh Napier UNIVERSITY

Example

- If User rwx, Group rx, Other rx,
  - Symbolic –rwxr-xr-x
  - Numeric 0755
- If User rwx, Group x, Other none
  - Symbolic –rwx--x---
  - Numeric 0710

Edinburgh Napier UNIVERSITY

> umask 022

- When a command creates a file or directory the default is:
  - rwxrwxrwx – for directories
  - rw-rw-rw- – for files
- The value of your umask is SUBTRACTED from the numeric protection code.
- So removing write for group and other you need to know that 2 stands for w, and thus for:
  - rw-r--r-- (644) - Write only for owner.
  - This is numerically, 666-022 => 644
  - So the umask is 022.

Edinburgh Napier UNIVERSITY

The umask mask

- 0022
  - Col 0 is always 0, Col 1 is OWNER
  - Col 2 is GROUP, Col 3 is OTHER
- Values:

Octal	Binary	Perms	Octal	Binary	Perms
0	000	rwx	4	100	-wx
1	001	rw-	5	101	-w-
2	010	r-x	6	110	--x
3	011	r--	7	111	---

Edinburgh Napier
UNIVERSITY

PROCESSES

Edinburgh Napier UNIVERSITY

Processes

- Processes are running programs.
- They have their own ID (pid)
- Some processes are part of the filesystem and can be found.
- Some processes are special, and cannot be found, and these are usually described [brackets].
- The INIT process is the boss process in linux.

Edinburgh Napier UNIVERSITY

> ps aux

```
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.1  0.8   1480   496 ?        S    12:57   0:00 init [5]
root          2  0.0  0.0      0     0 ?        SWN  12:57   0:00 [ksoftirqd/0]
root          3  0.0  0.0      0     0 ?        SW<  12:57   0:00 [events/0]
root          4  0.0  0.0      0     0 ?        SW<  12:57   0:00 [khelper]
root         16  0.0  0.0      0     0 ?        SW   12:57   0:00 [kjournald]
...
root        527  0.0  0.9   1464   576 ?        S    12:58   0:00 syslogd -m 0
rpc         553  0.0  0.9   1544   584 ?        S    12:58   0:00 portmap
rpcuser     573  0.1  1.3   1644   812 ?        S    12:58   0:00 rpc.statd
root        658  0.4  2.4   3656  1484 ?        S    12:58   0:00 /usr/sbin/sshd
gordon    15521  0.0  0.1   3992   760 pts/1    R    20:41   0:00 ps aux
```

Edinburgh Napier UNIVERSITY

State Codes

- Standard Codes
  - D uninterruptible sleep (usually IO)
  - R runnable (on run queue)
  - S sleeping
  - T traced or stopped
  - W paging
  - X dead
  - Z a defunct ("zombie") process

- Additional Codes
  - W has no resident pages
  - < high-priority process
  - N low-priority task
  - L has pages locked into memory (for real-time and custom IO)

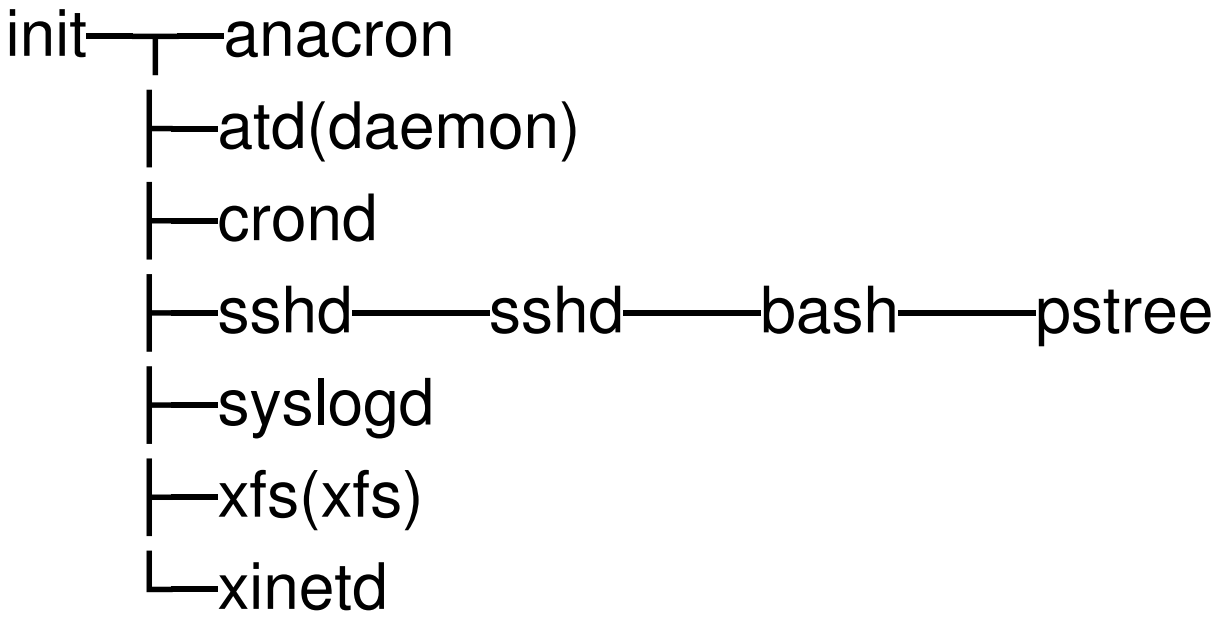
Edinburgh Napier UNIVERSITY

Process Relationships

- Processes form “trees” of parentage.
- All processes have the parent INIT.
- If a process starts another process, that new process has a parent of the old process.
- I now run pstree, in the bash shell, after logging in to the machine using ssh (controlled by sshd).

Edinburgh Napier UNIVERSITY

> pstree



Edinburgh Napier UNIVERSITY

/proc

- Processes are represented as files in /proc
- They appear as a directory with a name equal to the PID of the process.
- They have things in the directory which define the process in question.

Edinburgh Napier UNIVERSITY

> ls –l /proc/668

```
-r--------.  1 root root 0 Sep 21 16:32 auxv
-r--r--r--.  1 root root 0 Sep 21 16:31 cmdline
lrwxrwxrwx.  1 root root 0 Sep 21 16:32 cwd -> /
-r--------.  1 root root 0 Sep 21 16:32 environ
lrwxrwxrwx.  1 root root 0 Sep 21 16:32 exe -> /usr/sbin/sshd
dr-x------.  2 root root 0 Sep 21 16:32 fd
-r--r--r--.  1 root root 0 Sep 21 16:32 maps
-rw-------.  1 root root 0 Sep 21 16:32 mem
-r--r--r--.  1 root root 0 Sep 21 16:32 mounts
lrwxrwxrwx.  1 root root 0 Sep 21 16:32 root -> /
-r--r--r--.  1 root root 0 Sep 21 16:31 stat
-r--r--r--.  1 root root 0 Sep 21 16:32 statm
-r--r--r--.  1 root root 0 Sep 21 16:31 status
dr-xr-xr-x.  3 root root 0 Sep 21 16:32 task
-r--r--r--.  1 root root 0 Sep 21 16:32 wchan
```

Edinburgh Napier UNIVERSITY

> ls –l /proc/668/fd

```
lrwx------.  1 root root 64 Sep 21 16:32 0 -> /dev/null
lrwx------.  1 root root 64 Sep 21 16:32 1 -> /dev/null
lrwx------.  1 root root 64 Sep 21 16:32 2 -> /dev/null
lrwx------.  1 root root 64 Sep 21 16:32 3 -> socket:[4230]
```

- Files which that process has open.
- 0,1,2 are STDIN,STDOUT,STDERR.
- 668 : sshd – listening on a socket for people logging in with ssh.

Edinburgh Napier UNIVERSITY

> sleep 20 > /tmp/hia &
[1] 854
> ls –l /proc/854
> ls –l /proc/854/fd

```
lrwxrwxrwx.  1 root root 0 Sep 21 16:45 cwd -> /root
-r--------.  1 root root 0 Sep 21 16:45 environ
lrwxrwxrwx.  1 root root 0 Sep 21 16:45 exe -> /bin/sleep
dr-x------.  2 root root 0 Sep 21 16:45 fd

lrwx------.  1 root root 64 Sep 21 16:45 0 -> /dev/pts/0
l-wx------.  1 root root 64 Sep 21 16:45 1 -> /tmp/hia
lrwx------.  1 root root 64 Sep 21 16:45 2 -> /dev/pts/0
```

Edinburgh Napier UNIVERSITY

Daemons

- A Daemon is a process started when you boot which runs in the background.
- Not all things started when booting stay running (e.g. they set something up and then die).
- To help us, daemons usually have a name which ends with a “d”. (e.g. syslogd, sshd).

Edinburgh Napier UNIVERSITY

> top

```
top - 16:03:17 up  1:04,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  35 total,   2 running,  33 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0% us,  0.0% sy,  0.0% ni, 100.0% id,  0.0% wa,  0.0% hi,  0.0%
   si
Mem:     59764k total,    52308k used,     7456k free,     6192k buffers
Swap:   205816k total,        0k used,   205816k free,    32472k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
  807 root      16   0  1624  728 1412 R  0.0  1.2   0:00.02 in.telnetd
  934 root      16   0  1828  872 1628 R  0.0  1.5   0:00.00 top
```

Edinburgh Napier UNIVERSITY

SYSLOG

- The syslogd daemon is your friend.
- It helps other daemons record what is going on into a file.
- On the website, you can click on “syslog output” and see what syslogd has noticed.
- This output, known as the syslog, can also be seen from the prompt using “dmesg”.

Edinburgh Napier UNIVERSITY

> who

root pts/0 Sep 21 15:59 (hub1-gw)

- I am root, logged on from hub1-gw.
- My session is linked to a device which handles my screen and keyboard, called pts/0
- This refers to /dev/pts/0

Edinburgh Napier UNIVERSITY

> ls –l /dev/pts/0

crw--w----. 1 root tty 136, 0 Sep 21 16:49 /dev/pts/0

- This is a character device.
- The person connected via it always owns it.
- There are no sizes with block or char devices.
- 136 is the major device number
- 0 is the minor device number.

Edinburgh Napier UNIVERSITY

mknod

- Devices are usually created automatically.
- To create a new file to represent a device, use mknod.

> mknod /tmp/screen c 136 0

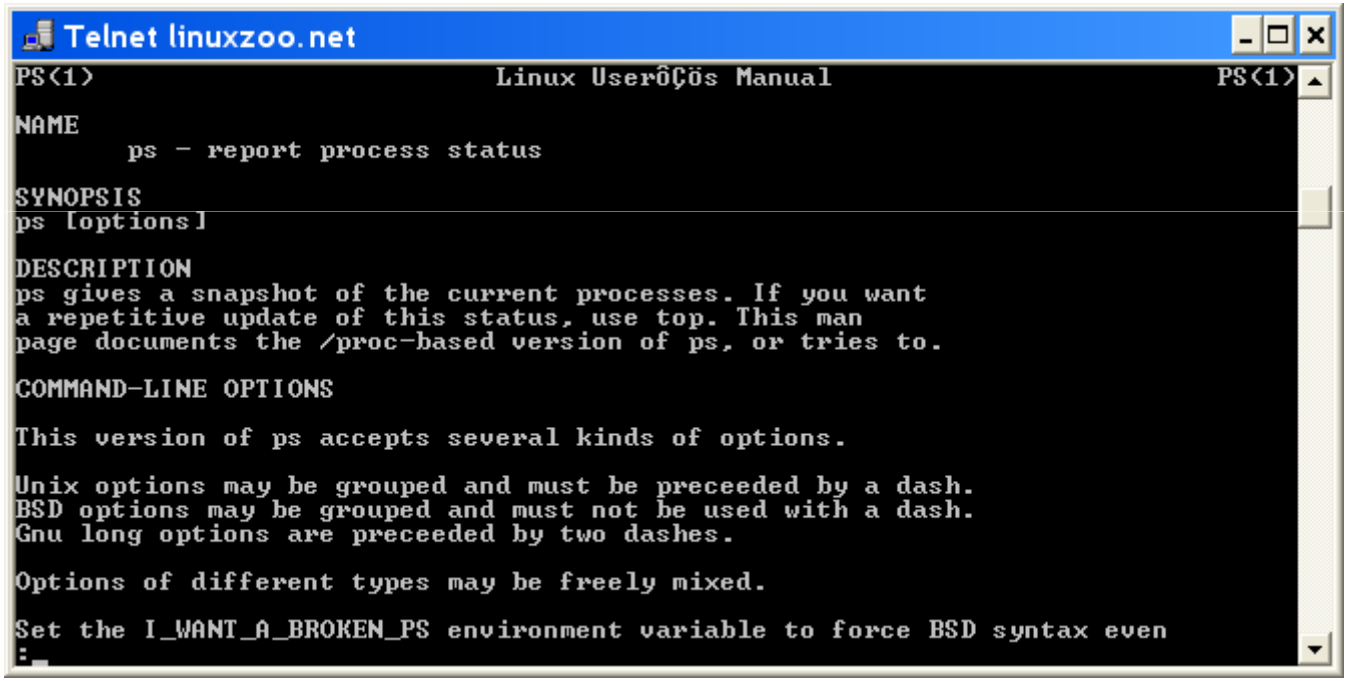
> echo “hello there” > /tmp/screen

hello there

Edinburgh Napier UNIVERSITY

- Find things out for yourself!

> man ps



Edinburgh Napier
UNIVERSITY

PIPES

Edinburgh Napier UNIVERSITY

Pipes

- In the last lecture you saw “>” and “<“ as redirections.
- For example, to copy file a to file b you could do:

$ cp a b

- But you could do the following (ugly) command

$ cat < a > b

- Remember the cat command prints what it gets, and here it gets from a and puts to b. Don’t do this, as its too ugly for a real admin to do.

Edinburgh Napier UNIVERSITY

- These redirections work fine, unless you want “>” to give its output to another program (rather than a file).
- Example, I am looking for all the users who have a username beginning with “a”. I will use a regular expression for this, “^a”

```
$ grep "^a" /etc/passwd
adm:x:3:4:adm:/var/adm:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
andrew:x:501:500:Andrew Cumming:/home/andrew:/bin/bash
```

- That got the information, plus lots of other pieces of info.
- There is a command called “cut”, which chops things out of a line. It will split a field out of a line so long as it knows what character marks the end of 1 field and the start of another.
- In /etc/passwd, “:” splits each field, so cut is –d”:”
- The username is in field 1, so –f1

Edinburgh Napier UNIVERSITY

```
$ grep "^a" /etc/passwd > a
$ cut –d”:” –f1 < a
adm
apache
andrew
```

- We can do this in one line, instructing the prompt to give the output from grep as the input to cut. We use pipe “|” to do this.

```
$ grep "^a" /etc/passwd | cut –d”:” –f1
```

Edinburgh Napier
UNIVERSITY

HARD and SOFT LINKS

Edinburgh Napier UNIVERSITY

File links

- Sometimes we want to have the same file contents in two or more different places.
  - Edit one version and you edit all versions
  - You only use up the disk space for one version no matter how many copies you have.
  - This is done using a file link
- Links are common in the system directories, and are used for configuration as well as dynamic libraries.

Edinburgh Napier UNIVERSITY

Hard Link

- Lets assume that the current working directory is /home/john.
- We wish to create a link 'hardfile2' within the sub-directory projects from the file 'hardfile'.

% date > hardfile (create the file)

% ls -l

-rwx-xr-x. 1 john users 605 Nov 18 12:25 hardfile

% ln hardfile project/hardfile2

% ls -l projects/hardfile2

-rwx-xr-x. 2 john users 605 Nov 18 12:25 projects/hardfile2

Edinburgh Napier UNIVERSITY

% ls -l projects/hardfile2

-rwx-xr-x. 2 john users 605 Nov 18 12:25 projects/hardfile2

- The file 'hardfile' and its like 'hardfile2' are indistinguishable,
- if 'hardfile' is updated then 'hardfile2' is updated.
- Notice the link number has increased to 2. This would occur in both listings.

Edinburgh Napier UNIVERSITY

Soft Links

- Again, let us assume that the current working directory is /home/john and we wish to create a link 'softfile2' within the subdirectory projects to the file 'softfile'. Notice the '-s' switch:

% date > softfile (create the file)

% ls -l

-rwx-xr-x. 1 john users 605 Nov 18 12:25 softfile

% ln -s /home/john/softfile project/softfile2

% ls -l projects/softfile2

lrwx-xr-x. 1 john users 605 Nov 18 12:25 projects/softfile2 ->
/home/staff/john/softfile

Edinburgh Napier UNIVERSITY

Soft Links

lrwx-xr-x. 1 john users 605 Nov 18 12:25

softfile2 -> /home/staff/john/softfile

- Notice the appended pathname on the long listing, the link number has not changed, but the permissions show an 'l' at the beginning of the long listing rather than a '-'.
- Again any updates in 'softfile' will be reflected in 'softfile2'.

Edinburgh Napier UNIVERSITY

Discussion

Future of file permission:

- Is User/Group/Other sufficient?
- Simple control methods? ACL…
- Complex control methods? SELinux