

Fetchmail and NT (ems3)

- University mail server is ems3
 - a user joe has an account jbloggs on corporate
- also maintains a film mailing list
- requires three accounts
 - film@glam.ac.uk: post messages
 - film-request@glam.ac.uk: new users subscribe
 - film-admin@glam.ac.uk: users change passwords

Mailman

- handles mailing lists
 - web based and email based
- users subscribe, unsubscribe and post/receive email's
 - by email
 - or by html/cgi pages and scripts
- view all articles on web, sort by author, subject, date

Mailman

- installs out of the box
- each mailing list requires
 - *subject*@glam.ac.uk mailbox
 - *subject-admin*@glam.ac.uk mailbox
 - *subject-request*@glam.ac.uk mailbox
- you need to decide the *subject!* and set them up accordingly
 - can be done via cgi/html

Fetchmail script

- ```

set daemon 60 # fetchmail wakes up
 # every 60 seconds

#
joe's email mailbox
#
poll ems3.glam.ac.uk protocol imap
 username jbloggs@corporate with
 password hidden, is joe here;

#
films mailing list
#
poll ems3.glam.ac.uk protocol imap
 username film@corporate with
 password hidden, is film here;

```

## Fetchmail script

```
#
and the admin names for the film
mailing list
#
poll ems3.glam.ac.uk protocol imap
 username film-request@corporate
 with password hidden, is film-request here;
poll ems3.glam.ac.uk protocol imap
 username film-admin@corporate
 with password hidden, is film-admin here;
```

## Fetchmail script description

- NT domain is corporate
- usernames on ems3 do not need to be the same on a GNU/Linux box
- *one* fetchmail process running on GNU/Linux can handle *many* different user accounts
  - jbloggs, film, film-admin, film-request
- fetchmail allows optional *syntax sugar*
  - gui is available

## Network File System (Advantages)

- NFS available on many operating systems
  - macos, UNIX, GNU/Linux, Free BSD, Solaris, PCs (MSDOS), VMS, OS/2
  - Novell and NT if you pay extra
- NFS built on top of Sun's RPC (remote procedure call) mechanism
  - open standard, originally produced by Sun Micro-systems
  - uses RPC authentication

## Network File System (Disadvantages)

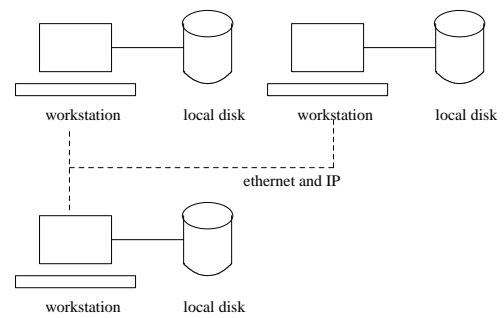
- uses RPC authentication
  - easily spoofed
- information sent over network is not encrypted
  - data could be replaced

## NFS Advantages

- stateless server and client
- server could be rebooted and user on client might be unaware of the reboot
- client/server distinction occurs at the application/user level *not* the system level
  - highly flexible, so we need to be disciplined in our administration/configuration

## A typical GNU/Linux network

■



- configuration is *very* flexible

## GNU/Linux configuration

- flexibility could be a disadvantage as system administrators have to fight their way through many different options for various components just to perform simple operations
- it is possible that all machines are configured identically (except for their hostname, IP address etc)
  - and ghost a hard disk
- could configure the network operating systems so that all local disks can be seen by all machines

## A typical GNU/Linux network

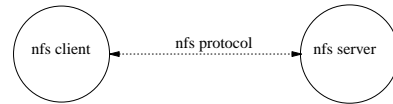
- alternatively machines could be very different (ie could build distinct client/server model similar to Novell)
- one machine has the common file system
- one machine has the system executables

## Disadvantages of GNU/Linux configuration

- complexity
- flexibility requires system administrators be disciplined otherwise overall system may become unmanageable
  - when more machines are added
- harder to configure than other operating systems
  - getting better (GNU/Linux is embracing Gtk/Tcl/Python)
  - but complexity still remains
  - few graphical tools to *identify* installation errors
    - however the problems are being addressed

## NFS configuration example

- can network file systems between different architectures
  - little endian, big endian - it makes no difference
- NFS design is built from three main parts: a protocol, a server and a client



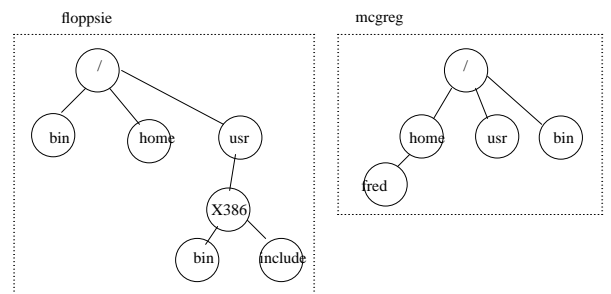
## NFS configuration example

- components fit into the ISO-7 layer model

|         |               |          |
|---------|---------------|----------|
| Level 7 | Application   | NFS      |
| Level 6 | Presentation  | XDR      |
| Level 5 | Session       | RPC      |
| Level 4 | Transport     | TCP UDP  |
| Level 3 | Network       | IP       |
| Level 2 | Data Link     | Logical  |
| Level 1 | Physical Link | Ethernet |

## NFS example

- suppose we have two machines



## NFS configuration example

- our task is to allow machine floppsie see directory /home/fred on mcgreg
- tell floppsie where this directory resides
- need to tell mcgreg that floppsie is trusted

## NFS configuration of floppsie

- edit /etc/fstab on floppsie and add the following line

```
remote fs mount point type options
mcgreg:/home /home/mcgreg nfs
```

## NFS configuration of mcgreg

- need to tell machine mcgreg that /home is exported and that floppsie is allowed to see it
- edit /etc/exports on mcgreg and add the following line

```
/home 193.63.128.0/255.255.255.0(rw) \
193.63.152.84(rw)
```

- why do we use 193.63.152.84 rather than floppsie?

## Laboratory work

- read about the program df using man
  - ie man df
- try running df and write down which file systems exist
- which file systems are networked (NFS)?
- continue working on your assignment