Network File System

Concept: machines can share files.

File Server: must grant permission for access

Client: must import file system

Unix: permission and import is by subtree
a directory and its descendants

Unix attaches directories into a tree using `mount`

A `mount` of a subtree from a server will be used to attach the server's subtree into our file system tree

Linux: Networking components

`rpc.portmap`: NFS is based on remote procedure calls

`rpc.mountd`: The server runs this program, it handles mount requests from clients

`rpc.nfsd`: The server runs this program, it handles requests for files from client machines

Exports

file: `/etc/exports`

Purpose: say what is exported and to whom

Each line: A directory and a list of machines permitted to mount the subtree under that directory

```
/harddisk lab17.net.cecs.csulb.edu(rw,sync)
/usr/info lab19.net.cecs.csulb.edu(ro) lab20(rw)
/usr/bin *.net.cecs.csulb.edu(ro)
/oops    134.139.136.64/255.255.255.192(rw)
```

lab17 is allowed read/write access to `/harddisk`
lab19 is allowed read only access to `/usr/info` and lab20 is allowed read/write
`/usr/bin` is exported with a wild card
User sam on will have "sam" priviledges on `/harddisk`

`sync`: write immediate to keep files in sync
`root_squash`: root on the client machine doesn't get root priviledges
`no_subtree_check`: ensure file requests are in the exported subtree

`exportfs`: A command similar to mount; it causes the exports file to be reread. Pay attention command to the options!

Remote Programs

At boot the remote programs are started by the `rc.nfsd` script.

The script may be enabled by `chmod a+x rc.nfsd`
May be run by hand, but requires the parameter `start`.

The script will start the remote programs only if the `exports` file exists and contains some exports.

The `rc.nfsd` script can also be run by hand with either a start and stop.

You may also start `rpc.mountd` and `rpc.nfsd` by hand.
Start only one copy.

May also start from `inetd`.

Mounting Remote Files

Normal mount command systax is used
The fstab can be used

`/etc/fstab`
`lynx.net.cecs.csulb.edu:/u1 /lynx nfs bg,soft 0 0`

server machine-name:/directory
Where to attach it locally

`nfs`: the type of file system
other options include `ext2` or `msdos`

`bg`: if the mount times out, background it and keep trying
`soft`: report an error if an NFS operation times out
`intr`: allow a control C to abort an operation if it has timed out
`nolock`: file locking is local to this machine and does not attempt to lock against other NFS clients.

Caution: booting or a program will hang if you give options which require an NFS operation to succeed and the NFS server is down. (i.e. we recommend bg, soft, intr).

Review:
The machine owning the disk must *export*.
The machine wanting to use the files must *mount*.

Network File System

automatic mounts

Situation: you have alot of NFS mounts,
many are infrequently used.

Regular NFS mount: Error if server goes offline.

Solution: keep unused NFS mounts in the unmounted state.
automatically mount when needed.

A `cd`, `ls` or other program that needs something in an NFS area triggers a NFS mount.

If a NFS mount hasn't been used in a while,
(timeout period), unmount it.

Sun: `automount`
Linux: `automount`
BSD: `amd`

Linux automount

1) Attach an automount daemon to a directory.

2) Specify which subdirectories to automount

Invoking automount (sample):

```
/usr/sbin/automount /home/mount file /etc/autostuff
```

Automounts are done for subdirectories of `/home/mount`
The file `/etc/autostuff` tells us what to do.

Sample `autostuff`:

```
u1 -soft,intr d1.cecs.csulb.edu:/u1
u2 -soft,intr d1.cecs.csulb.edu:/u2
mail -soft,intr charlotte.cecs.csulb.edu:/var/mail
```

1) if the user wants `/home/mount/u1`,
automount `/u1` from `d1`.

2) if the user wants `/home/mount/u2`,

3) if the user wants `/home/mount/mail`,
automount `/var/mail` from `charlotte`.

automount (alternative config)

Automount has a master map that can be used to specify other files.

The name of this file is specified by the `MASTER_MAP_NAME` variable in `/etc/defaults/autofs`

Default master map entry (must be last)
`/net -hosts`
Meaning:
If `/net/d1/u2` is requested assume:
`d1` is the hostname
`u2` is the directory name on that host
mount that directory under `/net/d1/u2`

Typical master map override entry:
`/net/aardvark /etc/auto.aardvark`
Meaning:
If the directory `/net/aardvark` is requestd, see the file
`/etc/auto.aardvark for directions.`
`File format same as given on previous slide.`