

# TNE30019/TNE80014 – Unix for Telecommunications

## Unix File Sharing Using NFS

Dr. Jason But

Swinburne University

Dr. Jason But

TNE30019/TNE80014 – File Sharing (NFS)

## File Sharing – NFS

- **Network File System (NFS)**
- Traditional UNIX file Sharing developed by Sun Microsystems
- Allow single copies of files that all systems can access
- Implemented using Open Network Computing Remote Procedure Calls (RPC)
- Initially based on UDP, but TCP support since version 3
- Standardised in IETF RFC 1813 (v3) and IETF RFC 3530 (v4)
- Need to configure two components
  - Client
  - Server

Dr. Jason But

TNE30019/TNE80014 – File Sharing (NFS)

## Outline

- File Sharing with Unix using NFS
- Configuring client
  - Mounting remote shares
  - Automating mounts
- Starting server
- Configuring shares
  - NFSv3
  - NFSv4

Dr. Jason But

TNE30019/TNE80014 – File Sharing (NFS)

## Enabling NFS Client (FreeBSD)

```
NFSv4 – /etc/rc.conf
nfsuserd_enable='YES'
nfscbd_enable='YES'
```

### Mounting remote NFS share

- Same as mounting local device
- Remote share becomes sub-directory of local file system
- Once mounted can almost ignore the fact it is NFS mounted

Example: Mount /home from server on 192.168.0.1 to local /mount/home

```
mount 192.168.0.1:/home /mount/home
```

- Can use DNS names instead of addresses
- How are user privileges mapped?

Dr. Jason But

TNE30019/TNE80014 – File Sharing (NFS)

## NFS – User Permissions

- Until NFS version 3 permissions based on Unix user/group IDs
- User/group IDs must be synchronised between client(s) and NFS server
- By default server maps requests from user root to user nobody
- Optionally can map all user IDs to nobody
- Optionally can map user root to user root on server
- NFS version 4 adds more security features
  - User authentication
  - ACLs

## Example Linux /etc/fstab

```
> cat /etc/fstab
/dev/sda1 swap                swap          defaults      0 0
/dev/sda2 /                        ext4          acl,user_xattr 1 1
/dev/sda3 /home              ext4          acl,user_xattr 1 2
/dev/sdb1 /data              ext4          defaults      1 2
192.168.0.1:/data /data2        nfs           soft,timeo=10,intr 0 0
proc /proc                  proc          defaults      0 0
sysfs /sys                    sysfs         noauto        0 0
usbfs /proc/bus/usb             usbfs         noauto        0 0
devpts /dev/pts              devpts        mode=0620,gid=5 0 0
```

## Automating Mounts via /etc/fstab

- File contains list of regularly used mount-points
- Makes mounting easier – *Don't need to remember all information*
- Automated mounting during system boot
- Simplifies mount command to `mount /mount/home`
- Extends to all mount types – physical disks, Samba, etc.

## Enabling NFS Server (FreeBSD)

### NFSv4 – /etc/rc.conf

```
nfs_server_enable="YES"
nfsv4_server_enable="YES"
nfsuserd_enable="YES"
```

### At startup following daemons run

- **nfsd** – Accepts requests from remote clients
- **mountd** – Performs tasks for nfsd (e.g. mount)
- **rpcbind** – Allows clients to discover NFS server
- **nfsuserd** – Loads user and group information into kernel

### Restarting after changes to configuration

```
/etc/rc.d/mountd reload
```

## NFS – Server Configuration

### Defining Shares – /etc/exports

Need to specify root under which all shares can be found

- V4: `local_dir`

Local directory on system, not exported but only sub-directories of this can be exported

### Remote Access

Individual sub-directories of `local_dir` specified

- Allows different permissions for different sub-dirs
- Clients specify path relative to `local_dir`
- See `man exports` for more help

## NFS – Example shared directories

### Sample /etc/exports

```
# Export entire file system - only master can connect and
# root user IDs are preserved
/      master(rw, no_root_squash)

# /usr can only be mounted by all in 192.168.(0/1).* but
# read-only User IDs are preserved except for root
/usr   192.168.0.*(ro) 192.168.1.*(ro)

# /home can be mounted by all in 192.168.0.*. User IDs are
# preserved except for root
/home  192.168.0.*(rw)

# /pub can be mounted by all but user IDs are not preserved
/pub   (rw,all_squash)
```

## NFS – Server Configuration

### Defining Shares – /etc/exports

Each line specifies which directories are exported using NFS

- `sub_dir clients`

### `sub_dir`

Sub-directory of V4 root

### `clients`

List of clients who are allowed to mount/access this directory  
Each client can be specified with range of options

- `rw` – allow read/write as opposed to read-only
- `root_squash`(default) – if mounted by a root user then change their user id to nobody (also `no_root_squash`)
- `sync` – confirm all writes locally before responding to client

## Warning

- Be careful with online documentation, /etc/exports format for NFSv4 is very different under Linux vs. BSD
- Error messages can be obtuse
- Only one V4: line allowed