NT Disks and Filesystems

- administrating filesystems include
  - making files and directories available to local and domain users
  - monitoring and managing finite disk resources
  - protecting against file corruption by establishing a well planned backup schedule
  - ensuring data confidentiality by limiting file and system access
  - connecting new storage when required

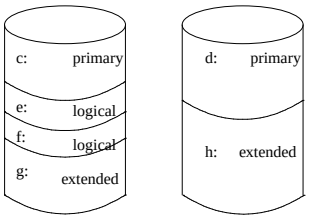
Making physical disks available

- install and configure the hardware
- partition the disk, into one or more logical independent pieces
- creating the filesystems on each partition, called *formatting* in the windows world
- adding files and directories to the new filesystem
- specifying access permissions for these files and directories
- enabling network access (*sharing*) for some or all of the filesystem

Managing disk partitions

- traditionally PC disk partitions are of two types:
  - *primary* partitions
  - *extended* partitions
- maximum of 4 primary partitions
- extended partitions can be subdivided into units
  - known as *logical drives*
- drive letters are assigned to Windows NT partitions

Drive letters

- drive letters assigned first to primary partitions
  - then to extended and logical drives
- 

PartitionMagic

- is a powerful and flexible disk administration tool
- it allows:
  - resizing existing partitions
  - moving a partition to a new location on the same disk
  - copying a partition to a different disk
  - changing the partition type
  - hiding a partition
    - marks it as inaccessible

NTFS filesystem features

- incorporates normal features of modern filesystems and has:
- journaling
  - before a change is made to the filesystem it is first recorded in a designated log file
    - implemented as a circular buffer and reused as necessary
  - on reboot the filesystem structure is synchronised with the log file
  - even after power loss the filesystem is returned to a consistent state
    - might incur some data loss
- enhanced file security
  - NTFS supports *discretionary access control* (DAC)
    - allows permitted and forbidden access to be specified in detail

NTFS filesystem features

- lazy write scheme
  - disk writes performed to a cache in memory rather than to a hard disk
  - cache periodically written to the disk in background
    - technique has been around in Computer Science for at least 30 years!
- `ntsync`
- is a small free utility which flushes the cache
- mirrored master file table and boot sector
  - stores a second copy of the above in the middle of the disk

Managing filesystems

- filesystems can be created via the disk administrators `Tools->Format` menu
- alternatively you can use the command line program `format`, which has the following syntax:
- `format [/fs:ntfs] [/V:label] [/C] [/A:n] x:`
  - where `x:` is the drive
  - `/C` for compression
  - `/A` for allocation units
- `format /C /fs:ntfs /V:user e:`
  - formats partition `e:`, named `user`, it will utilise compression

NTFS Features: Compression

- compression can be specified for
  - individual files
  - directories
  - complete filesystems
- if compression is not enabled for the filesystem
  - can enable it for a directory
- `compact [/C | /U] [options] [files]`
  - `/C` compressed
  - `/U` uncompress
  - options include:
    - `/S` recursive
    - `/I` ignore errors
    - `/Q` terse output

Compact example

- `compact /C /S Books`
- which marks all subdirectories and files below `Books` as compressed

Mounting and checking filesystems

- *mount*ing is the process by which a filesystem is made available to the system
- on Windows NT all local filesystems are mounted automatically at boot time
  - might want to make a filesystem unavailable
  - could use a public domain utility: `unmount`
- `unmount g:`
- `umount` will fail if there are any references to this filesystem
- package also contains a `mnt` program for the reverse

Checking filesystems

- ```
  chkdsk /f g:
  The type of the file system is NTFS.
  Chkdsk cannot run because the volume is in
  use by another process. Would you like to
  schedule this volume to be checked the
  next time the system restarts? (Y/N) Y
  ```
- `/f` checks and repairs any problems
  ```

Network Filesystem Access via Shares

- making a filesystem available to other computers is known as *sharing* in NT
  - a filesystem which is available to others is known as a *share*
- any directory or folder can be shared
  - files and subdirectories within them etc become available to network users
  - you can disable this, by protecting lower level directories with access lists
- protect via:
  - user lists
  - restricting number of users

Sharing command (on the server)

-
```
net share ntreskit=e:\ntreskit \
    /users:25 /remark:"Server Res Kit v4"
```
- the directory `ntreskit` on drive `e:`
  - number of simultaneous users is 25
  - the comment string: `Server Res Kit v4`

Sharing command (on the server)

- `net share`
- can be used to modify the properties of an existing share
-
```
net share ntreskit /users:unlimited
```
  - removes the upper limit of 25 users on `ntreskit`
    - note that we refer to `ntreskit` not `e:\ntreskit`
-
```
net share ntreskit /delete
```
  - stops `ntreskit` from being shared at all

Sharing command (on the client)

- to import a shared filesystem on a client can either use the GUI or command line
- in a GUI or application you can reference the share via:
-
```
\\lucy\ntreskit
```
- from the command line
-
```
net use g: \\lucy\ntreskit
```
  - from now on `g:` references the `ntnetkit`

Persistent drive settings

- may want shares to become active when a user logs in
  - may want to access shares via a user account, privilege
- allows us to manipulate the shares and then make the system remember the shares so that when we log in the environment is preserved
- ```
  net use k: \\lucy\cygwin /user:hacker *
             /persistent:yes
  ```
- means that the share `cygwin` is imported from server `lucy` and connected to the client as a user `hacker`
  - `hacker` will be asked for their password

Persistent drive settings

- are disabled by default
- to make all existing drive mapping persistent, type:
- ```
  net use /persistent:yes
  ```
- to `umount` a share on a client:
- ```
  net use k: /delete
  ```


net share

- a server command
- it is a good idea to type
- ```
  net share
  ```
- which displays all the shares the server is exporting

net use

- a client command
- ```
  net use
  ```
- will list all the shared resources imported from a server onto the current client
  ```