① Unlike the Windows File system,
where it uses <u>DRIVE LETTERS</u>, Linux
stores everything within a single directory
structure called a <u>virtual directory</u>.

WINDOWS

e.g. c:\ Users \ Rich \ Documents \ test.doc
indicates that <u>test.doc</u> is located in <u>Documents</u>,
which itself is located in directory <u>Rich</u>,
<u>Rich</u> is contained in directory <u>Users</u>,
which is located on the hard disk partition
assigned letter C. { C ← usually the first Hard
Drive on the PC }

LINUX

e.g. /home /rich /Documents /test.doc indicates
<u>only</u> that the file <u>test.doc</u> is in directory
<u>Documents</u>, under the directory rich, which
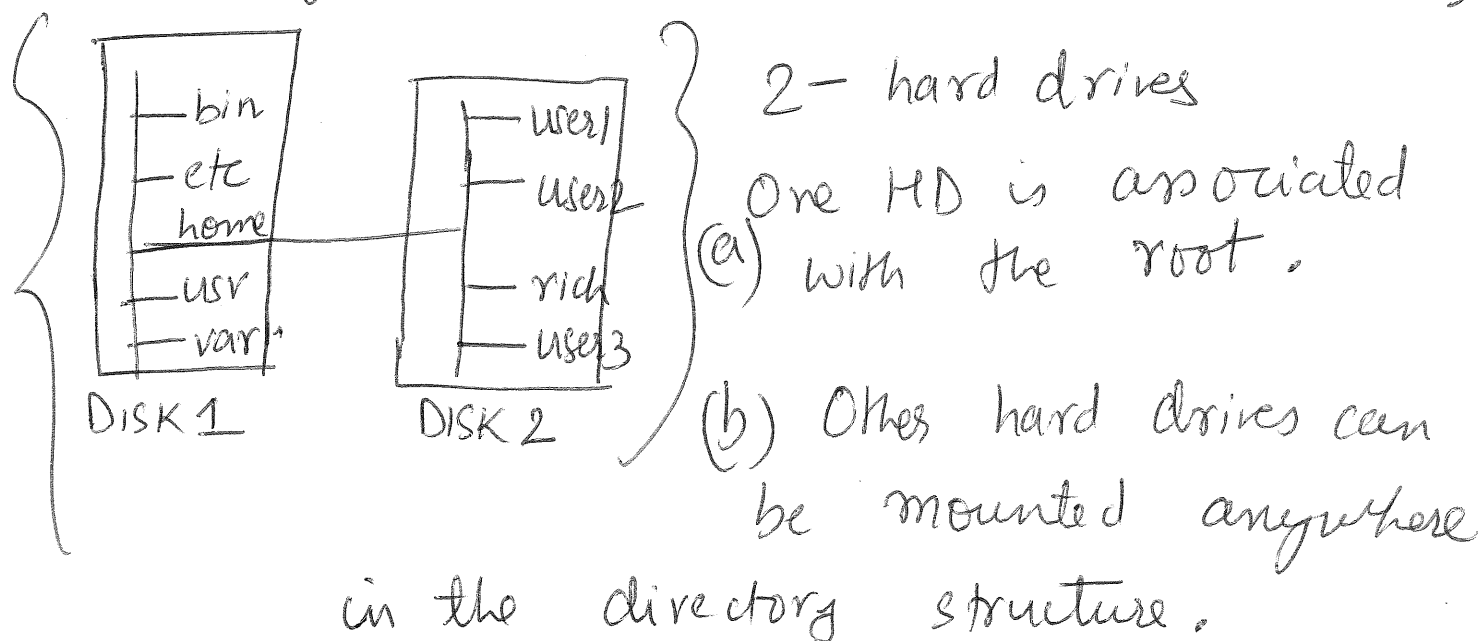is contained in the directory home

IT DOES NOT PROVIDE INFORMATION AS
TO WHICH PHYSICAL DISK ON THE PC THE FILE IS STORED

HOW DOES LINUX INCORPORATE EACH STORAGE DEVICE?

1. FIRST Hard DRIVE INSTALLED IN A LINUX PC is called root drive.

   root drive contains core of the virtual directory. Everything else builds from there.

2. On the ROOT DRIVE, Linux creates mount points (these are special directories) where you assign additional storage devices)

DISK 1

DISK 2

2- hard drives

(a) One HD is associated with the root.

(b) Other hard drives can be mounted anywhere in the directory structure.

2nd HD is mounted on /home, where user directories are located.

Linux System [ls-F] Directories

Directories created during installation of Linux

Need to know about them in case your scripts need to refer to them

/ Root directory ← files stored here

core of the directory system

```
        /
        |
 ┌──┬──┬──┬──┬──┬──┬──┐
/bin /boot /dev /etc /home /lib /mnt /ro...
```
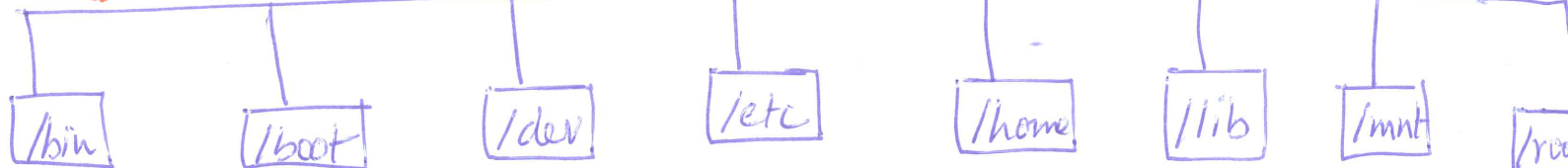
/bin — Binary utilities such as GNU user programs Firefox

/boot — Boot directory, GRUB [Boot loader] Linux Kernel

/dev — Device Files such as Sound card, etc.

/etc — Configuration files, Useful for programs to know how to behave

/home — User directory

/lib — Library files used by the System [Important for booting System]

/mnt — temporary connected devices such as USB, Flash drives, CDROM

/ro... — sup... Use... di...

/sbin , **/tmp** ← temporary files

System binary files used by (executables) the root only (e.g. clock, update, shutdown)

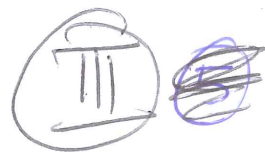**/var** — Typically contain log files

Your home directory path is stored in Variable called $HOME
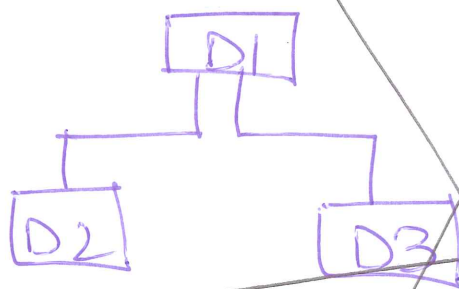
☐ / → Forward slash (Linux)
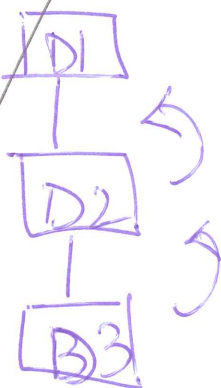
☐ \ → Back slash (Windows)

CHANGE DIRECTORY

cd /   ← change to root directory

cd ..   ← changes to parent directory

cd ..└─┘   ← changes to sibling directory

```
      ┌──┐
      │D1│
      └┬─┘
   ┌───┴───┐
┌──┴┐    ┌─┴─┐
│D2 │    │D3 │
└───┘    └───┘
```

cd ../../   ← two levels up

┌─────────────────────────┐
│ . ← Current   │
│   directory   │
│ .. ← Parent   │
│   directory   │
│ ~ ← Home directory   │
└─────────────────────────┘

```
┌──┐
│D1│
└┬─┘
┌┴─┐  ↰
│D2│
└┬─┘  ↰
┌┴─┐
│D3│
└──┘
```
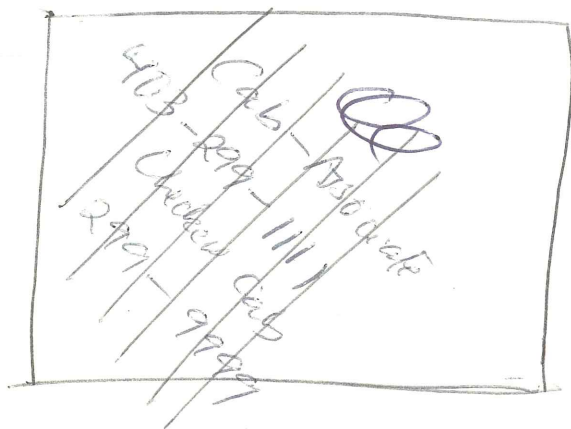
PWD and LS

Directory commands.

pwd   ← print working directory
       tells you where you are

ls /   ← lists directory

⑧ ⑥

$\{$ Absolute path name $\}$
$\{$ to my home directory $\}$

/afs/ec.auckland.ac.nz/users/a/m/amah811/unixhome

To refer to file1 in my home directory,
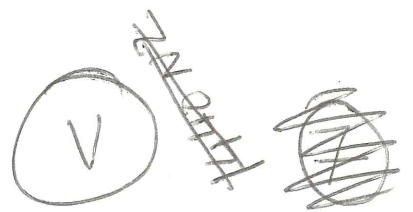the absolute path name:

$\Longleftrightarrow$ /afs/ec.auckland.ac.nz/users/a/m/amah811/unixhome/
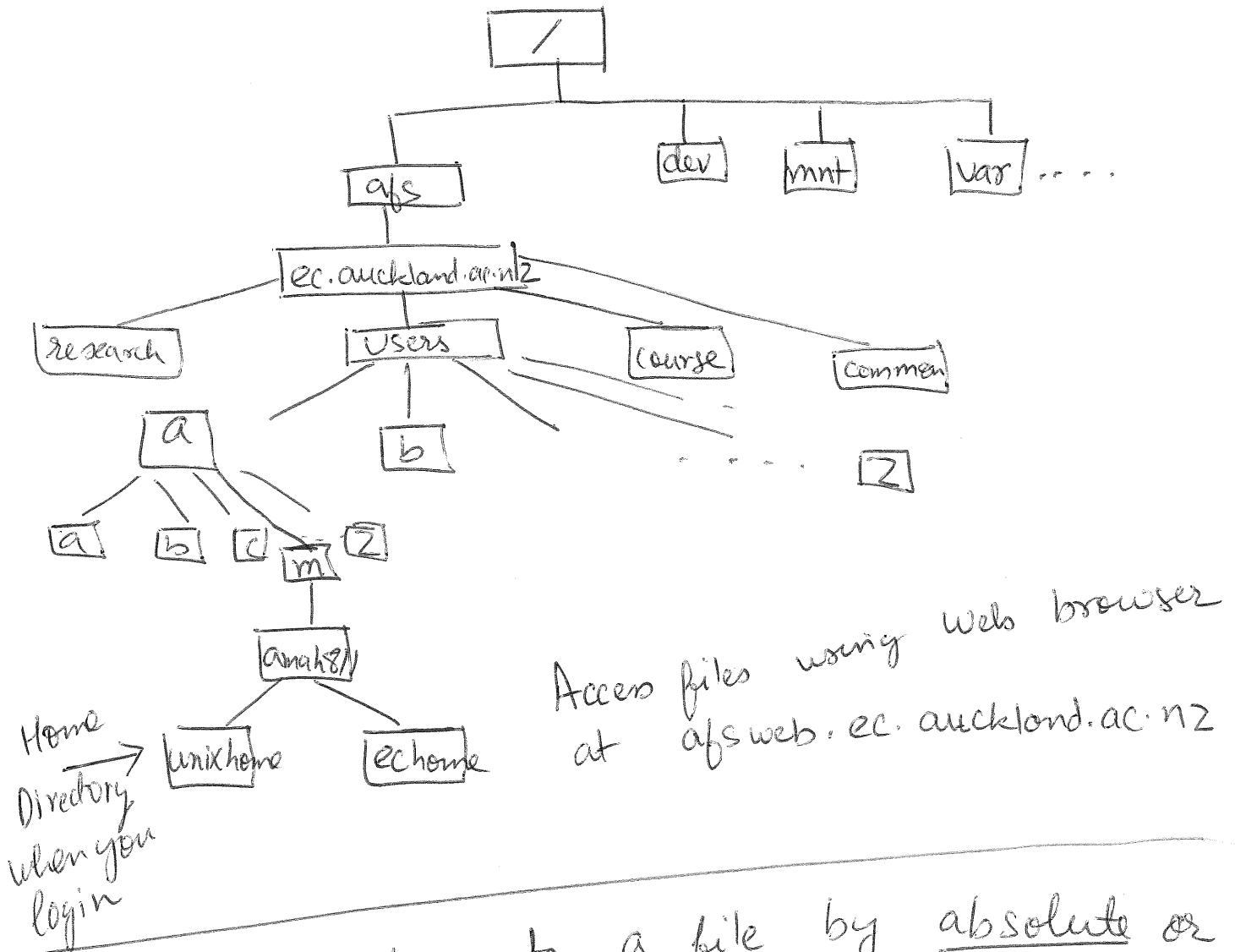file1.txt

If my current directory is
"/afs/ec.auckland.ac.nz/users/a/m/amah811"

Relative path to file 1.txt in my home
directory is   unixhome/file1.txt

and relative path to my home directory
is unixhome.

University Directory ~~File~~ Structure

```
                          ┌─────┐
                          │  /  │
                          └─────┘
             ┌───────────────┼──────────┬────────┬────────┐
          ┌─────┐                     ┌─────┐ ┌─────┐ ┌─────┐
          │ afs │                     │ dev │ │ mnt │ │ var │ . . . .
          └─────┘                     └─────┘ └─────┘ └─────┘
             │
    ┌──────────────────┐
    │ ec.auckland.ac.nz │
    └──────────────────┘
    ┌──────────┬──────────┬──────────────┬────────────┐
 ┌────────┐  ┌───────┐              ┌────────┐   ┌─────────┐
 │research│  │ Users │              │ course │   │ commen  │
 └────────┘  └───────┘              └────────┘   └─────────┘
         ┌──────┼─────┬─────────┐
      ┌─────┐ ┌─────┐           ┌─────┐
      │  a  │ │  b  │  . . . .  │  z  │
      └─────┘ └─────┘           └─────┘
    ┌───┬───┬────┬────┐
  ┌───┐┌───┐┌───┐┌───┐
  │ a ││ b ││ c ││ z │
  └───┘└───┘└───┘└───┘
           ┌─────┐
           │  m  │
           └─────┘
              │
        ┌──────────┐
        │ amah8/   │
        └──────────┘
         ┌────────┴────────┐
    ┌──────────┐      ┌─────────┐
    │ unixhome │      │ echome  │
    └──────────┘      └─────────┘
```

Home Directory when you login →

Access files using web browser
at afsweb.ec.auckland.ac.nz

─ We can refer to a file by **<u>absolute</u>** or <u>relative **path**</u> names.

① Absolute path name: Specifies full path from the root to the desired directory or file.

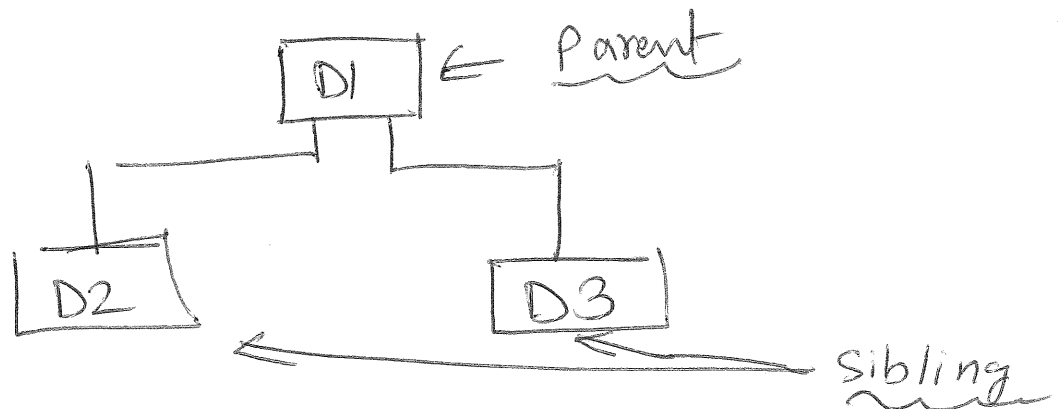② Relative path name: Path from current directory to a file or directory.

(IV)

Navigating directories

cd   /   ← change directory to root
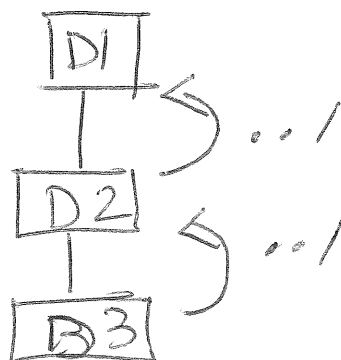   directory

cd   ..   ← changes to parent
                                directory

cd   ../directoryname   ← changes to
                                        sibling directory

{
. ← current directory
.. ← parent directory
~ ← HOME directory
}

```
        ┌────┐
        │ D1 │ ← Parent
        └────┘
       ┌────┴────┐
   ┌────┐      ┌────┐
   │ D2 │      │ D3 │
   └────┘      └────┘
       ←─────────── sibling
```

cd   ../../   ← two levels up

```
  ┌────┐
  │ D1 │  ⟩ ../
  └────┘
    │
  ┌────┐
  │ D2 │  ⟩ ../
  └────┘
    │
  ┌────┐
  │ D3 │
  └────┘
```

(IV)

File Protection System

$\Longrightarrow$ Each user belongs to at least one group.

$\Longrightarrow$ Groups of a user are determined by the system administrator.

$\downarrow$

Super user $\longrightarrow$ have access to any file and directory in the system.

ls -l in unix home

total 4 $\longleftarrow$ total # of blocks (Total disk allocation for all files in that directory)

FILE TYPE

Permissions

number of links

| - | rw- | r-- | r-- | 1 | amah | all | 81 | Mar·18 21:26 | grocery |

owner   Group   Others.

$\uparrow$ owner

$\uparrow$ group

$\uparrow$ size in bytes

$\uparrow$ Date and Time file was Created.
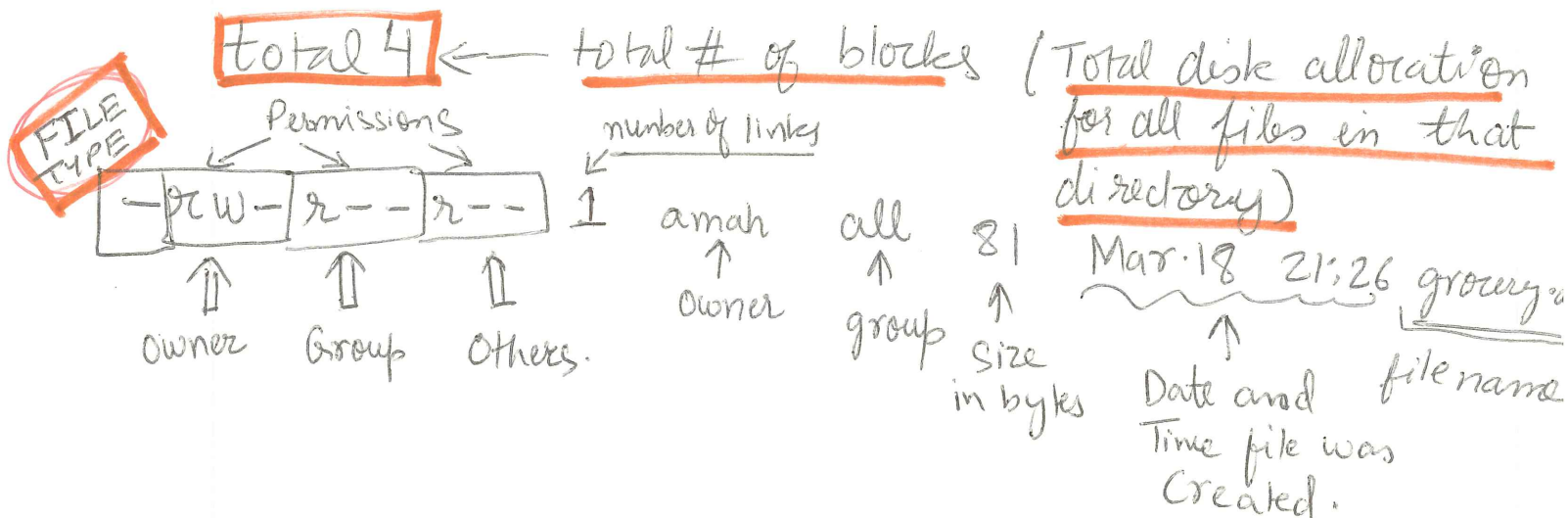
file name

Three permissions

r : Read

w : write and delete, rename

x : Execute

— : No permission

(V)

Difference between _Read_ and _Execute_

1) _Read_ _happens_ in case of _data_ files, which are opened by programs and parsed.

2) _Execute_ _happens_ when files are opened or run by the OS for the CPU. They contain instruction

---

Change file permissions

$$r \rightarrow 4$$
$$w \rightarrow 2$$
$$x \rightarrow 1$$

$$rwx \rightarrow 7$$
$$r-x \rightarrow 5$$
$$r-- \rightarrow 4$$

u ← user
g ← group
o ← others
a ← all three

(chmod 744 filename.)

Chmod a-rwx filename

⊕ ← add permission

⊖ ← remove permission

(VI)

(✳) File Types

| | | Unix looks at everything as a file |

- **—** A normal file
- **d** Directory
- **b** Block device sends data in blocks
  (e.g. HD, USB drive)
- **c** Character device sends single characters
  (eg. serial ports, sound cards)

S- socket (network connection)

Permissions

- r   Read
- w   Write
- x   Execute   ← For directories, means you can do a directory listing for sub directories.
- —   No access

Link counts

- 1 ← file
- 2 ← Directory } Starting.

ls -a   ← Shows hidden files

mkdir   ← Creates a directory

rmdir   ← deletes an empty directory

File Management Commands

COPY
- cp ← create copy of original
- cp [options] → source destination
- -i ← interactive mode

MOVE
- move command
- mv source destination
  - ↑ Renames in same directory

Source and destination can have same name only when moving file to another directory

REMOVE
- rm filenames ← deletes files permanently
- rm -i ← interactive mode
- rm -f ← delete forcibly
- rm -rf to recursively delete files in a directory without prompting