Linux Command Line

Nasser Giacaman
SoftEng 206: Software Engineering Design

Graphical User Interfaces

- Easy for users, interaction becomes user-friendly
- Sometimes faster to just click here and there (rather than remembering commands)
- Some tasks are naturally visual, eg painting
- Not so good to use remotely
- Can be tedious for repetitive tasks
- More computation and memory intensive

Command Line

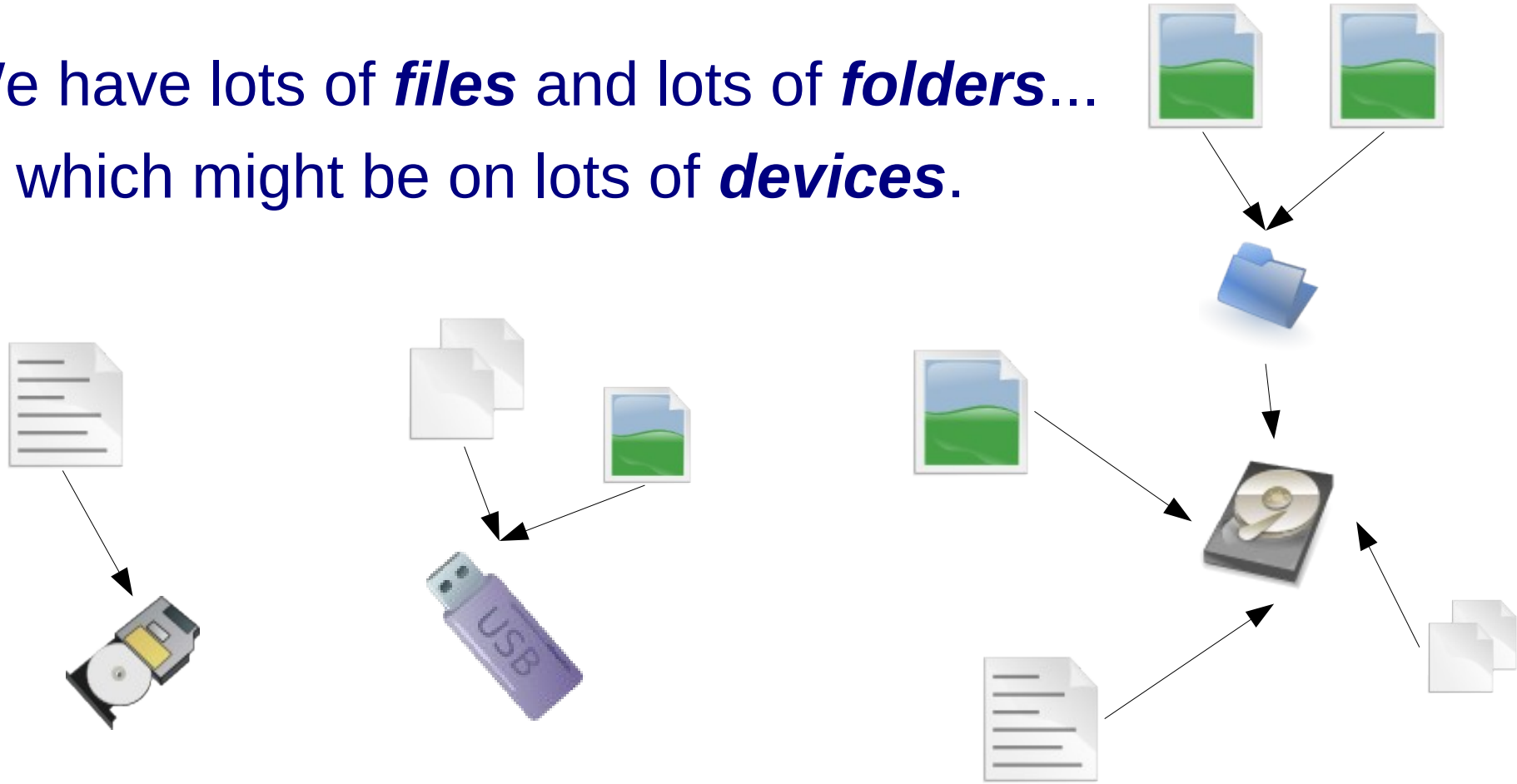
- More control, more options
- For many things is faster, no scrolling, clicking, just typing
- Awesome to connect to remote machines, in fact this is sometimes the only possibility to connect to some machines!
- Learning curve, requires practice, learning commands..
  - But there's always the “man” pages, and Google
- Awesome at repeating common tasks
- Can be used in combination with GUI (e.g. create a script file that launches some GUI application by setting up various options.. easier than navigating the program list)
- People (who only know GUIs) will think you're brainy (but actually...)

ssh

- Sometimes command line is the only way to connect to a remote machine, especially servers, or special multi-user machines
- You can do this by using Secure SHell:
  - `> ssh ngia003@shell.ece.auckland.ac.nz`
- Try this out yourself! You should have access to this machine
- To log out when you finish:
  - `> exit`
- You can do all the things we talk about in the following lectures on that machine (or any other linux machine you have access to)

The Linux filesystem

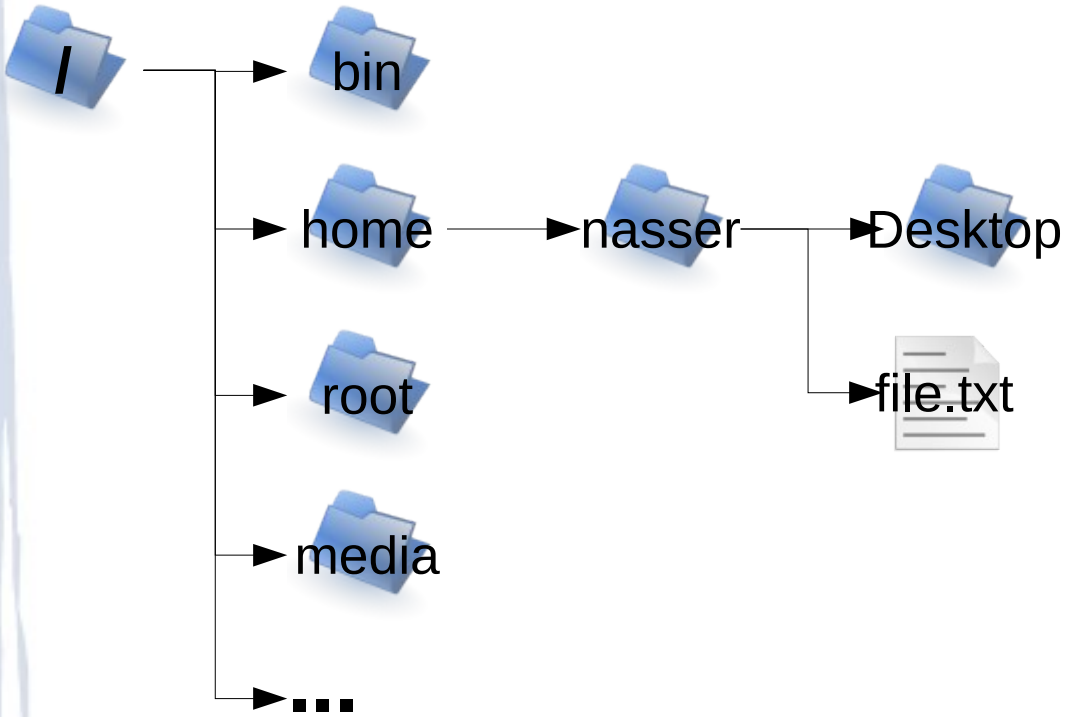
- We have lots of ***files*** and lots of ***folders***...
- ... which might be on lots of ***devices***.



- But Linux has the idea of a ***single hierarchy*** of files...

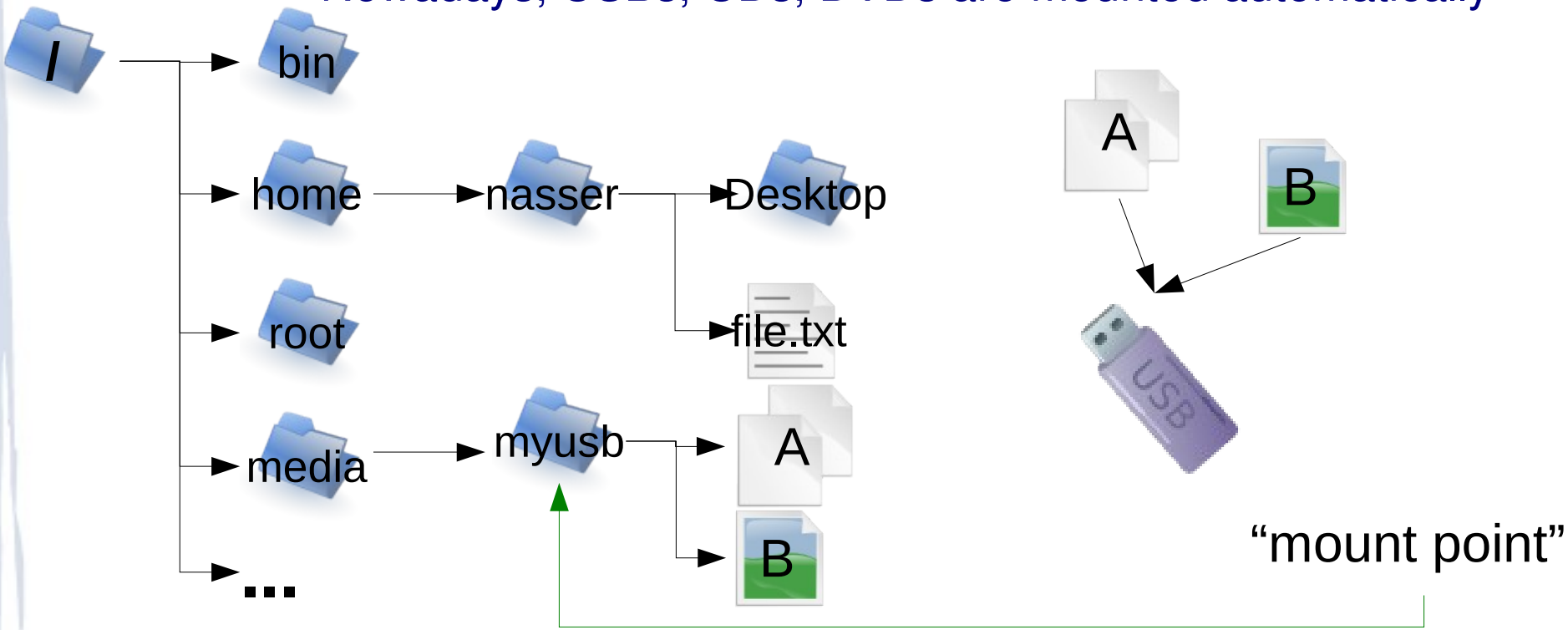
The Linux filesystem

- Single hierarchy
  - Appears as if all files are under one folder – the root “/”
  - Do not confuse the root folder “/” with the “root” folder



The Linux filesystem

- Mounting
  - If you want to access files from a new device, you need to specify where in the tree those files should appear
  - Nowadays, USBs, CDs, DVDs are mounted automatically



The file system

- The top of the filesystem is the root folder, denoted by /
- There is no concept of “drive letters” as in Windows, instead you mount a new device as a folder somewhere under “/”
- In most cases, you will only bother with what's inside your “home” folder
  - /home/nasser (on your own system)
  - /afs/ec.auckland.ac.nz/users/n/g/ngia003/unixhome (on uni labs)
  - Your home folder is also represented as a tilde ~
- When you attach removable media, typically they are mounted under /media

Shells and terminals

- Shell
  - This is the program that takes the commands you type and interprets them. There are different shells you can use, but we will use bash (Bourne Again SHell).. others: sh, ksh, csh, etc:
    - > cat /etc/shells
- Terminal
  - It is a window that opens up to allow you to interact with a shell.. for example konsole, gnome-terminal, xterm, etc
  - Notice something? gnome-terminal → Gnome, konsole → KDE
- From a terminal, try this:
  - `echo $SHELL`
  - `ps -p $$`
- Now change to another shell, try echo and ps again...

Processes

- `> ps -p $$`
  - What does that mean!?!?!?
  - `> $$`
- `> ps`
- `> ps ux`
- `> ps aux`
  - `> ps aux | grep -i firefox`
- `> pstree`
- `> top`
- `> htop`
- Want more? `> man ps`

Processes

- `> xeyes`
  - Ctrl + C (terminate)
  - Ctrl + Z (suspend)
    - `> bg`
- `> xeyes &`
  - `> fg`
- `> kill`
- `> killall xeyes`
- `> jobs`
  - `> jobs -l`
  - `> fg n` `> bg n`
- `> nohup xeyes &`

Navigation

- Print Working Directory:
  - `> pwd`
- Change Directory:
  - `> cd ~`
  - `> cd .`
  - `> cd ..`
  - `> cd assignments/se206/a1`
  - `> cd ./assignments/se206/a1`
  - `> cd /lib`
  - `> cd .kde` [the “.” makes a file/folder hidden]
- Push & pop:
  - `> pushd myfolder`
  - `> popd`

Path & PATH

- A path is like a set of instructions to get to a folder somewhere on the filesystem
  - **Relative**: are respective to the current directory
    - `> cd assignments/se206`
  - **Absolute**: start with /
    - `> cd /media/myusb`
    - `> cd ~/documents`
- PATH is a series of paths separated by colons
  - /home/nasser/bin:/usr/bin:/bin
  - `> echo $PATH`
  - `> export PATH=$PATH:/home/nasser/test`

Environment variables

- PATH is an environment variable, where to look for executables
- There are other environment variables
  - `> env`
- You can make your own environment variables
  - `> echo $MYVAR`
  - `> export MYVAR=/some/abs/path:some/relative/path:.`
  - `> export MYVAR=$MYVAR:new/path`
- The above is temporary for the current shell session.. For long term solutions, save it to one of the following files:
  - ~/.bashrc
  - ~/.profile
- > source .bashrc

Array variables

- A variable containing many values
- `FRUITS=(apple banana pear pineapple)`
- `> echo FRUITS`
- `> echo $FRUITS`
- `> echo $FRUITS[@]`
- `> echo ${FRUITS[@]}`
- `> echo ${FRUITS[index]}`
- `> echo ${#FRUITS[@]}`
- `> echo ${#FRUITS}`

Array variables

- Other ways to populate/edit an array
- `> FRUITS[0]=apple`

  `> FRUITS[1]=banana`
- `> FRUITS=(${FRUITS[@]} pear)`
- `> unset FRUITS[2]` // doesn't remove it, just unsets value
- Remove completely by combining indexes 0-1 and 3+

  `> FRUITS=(${FRUITS[*]:0:2} ${FRUITS[*]:3})`
- `> GREETINGS=(good morning good afternoon)`

  `> GREETINGS=('good morning' 'good afternoon')`
- `> HELLOS=(${GREETINGS[@]})`
- `> HELLOS=("${GREETINGS[@]}")`

Evaluating expressions

- `> whoami`
- `> echo "Hello `whoami`"`
- `> echo "Hello $(whoami)"`

- ```
  > x=2
  > y=3
  > echo $(($x+$y))
  ```
  ```

Random numbers and arithmetic

- `> echo $RANDOM`

- The ((…)) allows for arithmetic evaluation

```
> result=$((2+3))
> echo $result
```

- `dice=$((RANDOM%6+1))`

```
> echo $dice
```

Directory contents

- `> ls`
- `> ls -l`
- `> ls -a`
- `> ls -al`
- `> ls -l ~`
- `> ls -a /bin`
- `> man ls`

- `find`

File contents

- `> less fileName`
  - Page up, page down
  - G, g
  - /chars, n
  - q
- `> cat fileNames`
- `> file fileName`
- `> wc fileName`
  - `> wc –w filename`
  - `> wc –l filename`
- `> grep pattern file`

Redirection and pipelines

- `> ls -al > contents.txt`
- `> java BenchmarksA > results.txt`
- `> java BenchmarksB >> results.txt`

- `> ls -al > contents.txt`
  - Followed by `> less contents.txt`
- Or in 1 step `> ls -al | less`

- `> find | grep -i softeng`
- `> cat .profile .bashrc | grep bin | wc -l`

Redirect input, $ and help

- `> sort < fileName.txt`
- `> sort < fileName.txt > sortedFile.txt`
  - `> cat fileName.txt | sort > sortedFile.txt`

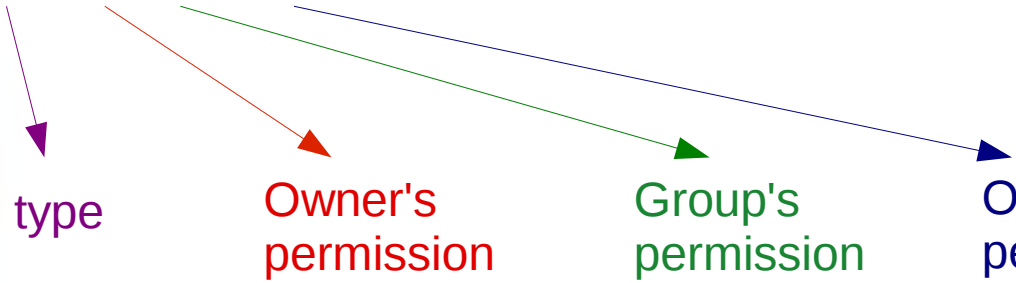
- `> echo “Hello, here we have: $(ls -l)”`
- `> echo Hello, here we have: $(ls -l)`

- `> type`
  - alias, built-in shell command, executable
- `> help`
- `> man`
- `> which`

File permissions

- `> ls -l`

```
-rw-r--r--  1 ngia003 scstud  37977 2012-03-11 15:41 handout.pdf
drwxr-xr-x 14 ngia003 scstud    392 2012-07-10 16:21 se206stuff
-rwxr-xr-x  1 ngia003 scstud    582 2010-11-20 17:10 run.sh
```



- read, write, execute...
  - e.g. permissions for run.sh are 755
    - 7=111(rwx), 5=101(r-x), 5=101(r-x)
- `> chmod 744 run.sh`
  - Only the owner can execute run.sh
- `> chown` `> chgrp`

Regular expressions

- `*`
- `Do*` → Documents, Downloads
- `?` → any character
- `[abc]` `[!abc]`
- `[:alnum:]` `[:digit:]` `[:alpha:]`
- `[:upper:]` `[:lower:]`

- `> echo [![:upper:]]*`
- `> echo *.java`
- `> echo ???`
- `> echo Do*`

Regular expressions and arrays and files

- `> echo Do*`

  `Downloads Documents`
- `FILES=\`echo Do*\``

  `FILES=$(echo Do*)`
- `FILES=(\`echo Do*\`)`

  `FILES=($(echo Do*))`
- Assume that SomeFile contains one word per line
- `LINES=\`cat SomeFile\``
- `LINES=(\`cat SomeFile\`)`
- `LINES=($(cat $(echo ResultFile*)))`

File manipulation

- `> cp [-r] SOURCE DEST`
- `> cp SOURCE... DIR`
- `> mv SOURCE DEST`
- `> mv SOURCE... DIR`
- `> rm -i FILE...`
- `> rm -r DIR...`
- `> mkdir DIR`

- `> mv src/*.class bin`
- `> rm ../*~`
- `> cp ??.txt myfolder`

Editors

- There are plenty of editors you can use from the command line
  - vi, emacs, gedit
  - I personally use vi, just because I'm used to it, but you can use any
- `> vi somefile`
  - Use arrows or page-up/down to move the cursor
  - Escape key goes into neutral mode
  - `i` will go into insert mode
  - `dd` will delete a line
  - `yy` copies a line (yank), `3y` will copy 3 lines
  - `p` will paste
  - Plenty more powerful commands
  - `:q` `:wq` `:q!`