Linux command Line Basics

ls → abbreviation for list, list all items in current directory

curl → download a file from the web

Terminal is an alternative interface to your computer

Not less capable than graphical interface

Terminals don't change much, they show
- information messages
- output from previous commands
- a prompt for new command

prompts contain (usually) the login user's name, the host name and maybe other information too

The terminal vs the shell

↓

draws text in a window, and lets you type things in a keyboard

You can use the terminal without the shell with a lot of terminal programs, you can tell it what program to run (shell is default)

Default unix shell is Bash, but there are others

When you write commands in a file, you're doing shell scripting. Not using the terminal

→

...But the terminal itself doesn't know what to do with that input, it needs another program to do that: The shell. The terminal sends what you typed to it and interprets what you wrote as a command, figures out the program to run, runs it, and sends the output back to the terminal

command line arguments are input you provide to a particular command or program when you run it

=

some commands print output always with the same format (like uptime)

others are more dynamic and the information they print depends on various factors (host ...)

=

In linux, the file system doesn't require files to have extensions on the end of their names

=

↑↓, shell keeps a history of commands you've run

history prints the current history (also commands run in other shell sessions)

Ctrl R, lets you search your shell history

=

unzip things.zip, unzips the file in the current directory

cat bivalves.txt gastropods.txt, prints out the contents of the files

↓
abbreviation for "concatenate" → concatenates files and outputs the results

=

pressing tab while typing a word, the shell will try to autocomplete it for you

If there are multiple choices, the shell will autocomplete as much as it can

wc, word count, tells how many words, lines and bytes are in a file

diff, compares files and shows you how they differ

=

all common shell programs come with a manual
man command

press q to exit the man page

=

ls doesn't show any files starting with ".", as those files are usually used on Linux for caching, configuration and other things

To display them use ls -a or ls --all

ls -l shows list of files with more info

=

Some programs take over the terminal for as long as they're running, and then you get the shell back when the program exits

ping 8.8.8.8

only stops with CTRL C (interrupt signal)
After stopping, it prints out a summary and you get your shell prompt back

eg sort, prints lines in sorted order, but keeps taking lines until the EOF (CTRL D) character

Some commands wait for the EOF, those commands are called line based programs

After EOF it will give you the shell back

—

man uses less, which takes the entire terminal, you can use less to display other files as well

SPACE D
scroll down a page at the time

↑ up

< > skip to the last / first line of the file

:3 ← goto line 3

:/otter ← (case sensitive) search after

n next occurrence

N previous

:/Regex

only type numbers and letters in search!

You can edit text files with editors like vim or emacs or gedit

pwd is an easy one built in utility

the two most important kind of objects in the linux filesystem are directories and files

Every object in the filesystem has a name (filename)

- can contain any character except slash. /
- when you write a filename with spaces or punctuation such as !$#[]%&; put the filename in quotes or precede the special character with \

Great Filename! → 'Great Filename!' → Great\ Filename\!

quoting

escaping

tell the shell to treat them as ordinary characters

directories are nested outermost inside each other

(ROOT) : unlike on Windows, there aren't separate roots for each disk partition (like C:), there's just one Filesystem root at the top of the Filesystem.

only / is the root directory. (Unix uses / instead Windows uses \ to separate directories)

No matter where the file is, you can unambiguously refer to the file by giving the full path which start at the root.

Directory names are separated by slashes

working directory, directory that is looking at

ls → lists files in the shell's current working directory

to know the path: pwd ← print working directory of that directory

cd : changes working directory

also called absolute path!

change directory

/var/log ↑ can also give the full path

cd log ↑ of a directory inside the current one

cd .. go to the parent of the working directory

cd . current directory

~ abbreviation for your own home directory

also called relative path!

cd with no argument takes you to your home directory

tab completion also works with directory names

=

mv item directory

can also be used to rename files

file ↑ destination

mv mustelidae.txt mustela.txt

=

to move several files into a directory — copying is similar

mv item1 item2 item3 directory — up item directory - (treat it a weird one)

commands have short names cause ↙

- quicker to type
- more efficient at the time when modem speed was measured in the hundreds of bits per second

mkdir path/of/new/directory

↑ takes either relative or absolute path

rmdir dirname, to remove a directory

if it has files in it, you can't rmdir, you must use rm -r dirname to remove everything inside first

mv mustel.txt junk/m.txt

move to junk and rename to m.txt

=

Anytime you want to operate on files with similar names, you can use a glob pattern to do it

Globbing is a kind of pattern matching for file names

the shell turns a glob pattern into a list of filenames that exist to match the pattern (remember, filenames are case sensitive!)

* , matches any string of characters

.html used at the beginning ← ... or at the end of a pattern → app

... or both ← *pp*

b*ng ← ... can also appear in the middle

app.{css,html}

↑ match files ending in either .css or .html

be?.png ← matches any one character

be??.png ← matches two characters instead

be[ea]r.png ← matches any of the characters in the brackets