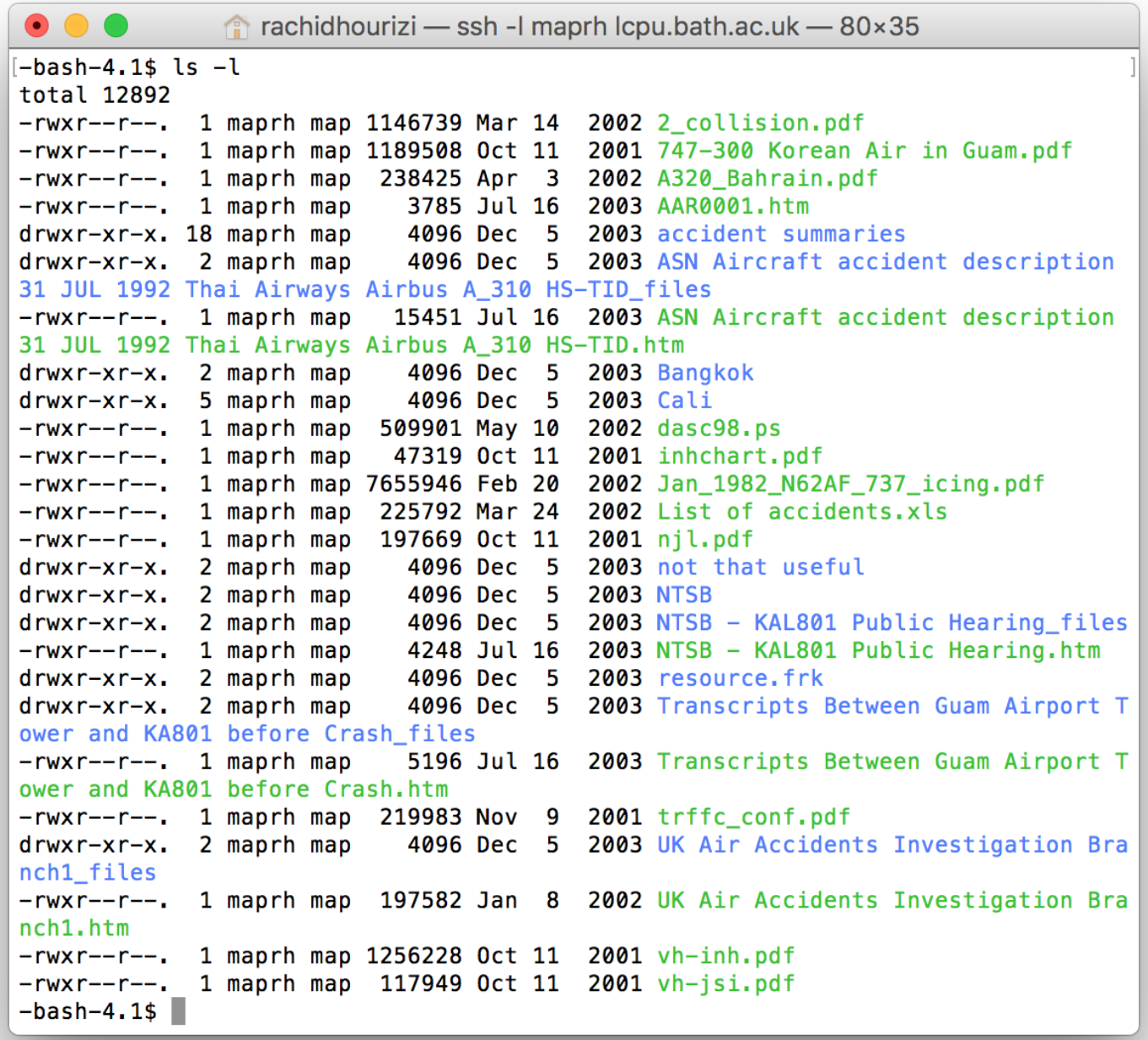
CM 10227/50258: UNIX 1

Dr. Rachid Hourizi and Dr. Michael Wright

October 13, 2016

Table of Contents

Introduction to UNIX


rachidhourizi — ssh -l maprh lcpu.bath.ac.uk — 80×35
-bash-4.1$ ls -l
total 12892
-rwxr--r--. 1 maprh map 1146739 Mar 14 2002 2_collision.pdf
-rwxr--r--. 1 maprh map 1189508 Oct 11 2001 747-300 Korean Air in Guam.pdf
-rwxr--r--. 1 maprh map 238425 Apr 3 2002 A320_Bahrain.pdf
-rwxr--r--. 1 maprh map 3785 Jul 16 2003 AAR0001.htm
drwxr-xr-x. 18 maprh map 4096 Dec 5 2003 accident summaries
drwxr-xr-x. 2 maprh map 4096 Dec 5 2003 ASN Aircraft accident description
31 JUL 1992 Thai Airways Airbus A_310 HS-TID_files
-rwxr--r--. 1 maprh map 15451 Jul 16 2003 ASN Aircraft accident description
31 JUL 1992 Thai Airways Airbus A_310 HS-TID.htm
drwxr-xr-x. 2 maprh map 4096 Dec 5 2003 Bangkok
drwxr-xr-x. 5 maprh map 4096 Dec 5 2003 Cali
-rwxr--r--. 1 maprh map 509901 May 10 2002 dasc98.ps
-rwxr--r--. 1 maprh map 47319 Oct 11 2001 inhchart.pdf
-rwxr--r--. 1 maprh map 7655946 Feb 20 2002 Jan_1982_N62AF_737_icing.pdf
-rwxr--r--. 1 maprh map 225792 Mar 24 2002 List of accidents.xls
-rwxr--r--. 1 maprh map 197669 Oct 11 2001 njl.pdf
drwxr-xr-x. 2 maprh map 4096 Dec 5 2003 not that useful
drwxr-xr-x. 2 maprh map 4096 Dec 5 2003 NTSB
drwxr-xr-x. 2 maprh map 4096 Dec 5 2003 NTSB - KAL801 Public Hearing_files
-rwxr--r--. 1 maprh map 4248 Jul 16 2003 NTSB - KAL801 Public Hearing.htm
drwxr-xr-x. 2 maprh map 4096 Dec 5 2003 resource.frk
drwxr-xr-x. 2 maprh map 4096 Dec 5 2003 Transcripts Between Guam Airport T
ower and KA801 before Crash_files
-rwxr--r--. 1 maprh map 5196 Jul 16 2003 Transcripts Between Guam Airport T
ower and KA801 before Crash.htm
-rwxr--r--. 1 maprh map 219983 Nov 9 2001 trffc_conf.pdf
drwxr-xr-x. 2 maprh map 4096 Dec 5 2003 UK Air Accidents Investigation Bra
nch1_files
-rwxr--r--. 1 maprh map 197582 Jan 8 2002 UK Air Accidents Investigation Bra
nch1.htm
-rwxr--r--. 1 maprh map 1256228 Oct 11 2001 vh-inh.pdf
-rwxr--r--. 1 maprh map 117949 Oct 11 2001 vh-jsi.pdf
-bash-4.1$

Why Command-line?

- Most modern tools have a graphical user interface (GUI)
  - Because they're easier to use
- But command-line user interfaces (CLUIs) still have their place
  - Easier (faster) to build new CLUI tools
    - ★ Building a GUI takes time
    - ★ Building a good GUI takes a lot of time
  - Easier to see and understand what the computer is doing on your behalf
    - ★ Which is part of what this course is about
  - Most important: it's easier to combine CLUI tools than GUI tools
    - ★ Small tools, combined in many ways, can be very powerful

The Shell

The most important command-line tool is the command shell (often just called the shell)

- Manages a user's interactions with the operating system by:
  - Reading commands from the keyboard
  - Figuring out what programs the user wants to run
  - Running those programs
  - Displaying their output on the screen
- Looks (and works) like an interactive terminal circa 1980

The Terminal

rachidhourizi — ssh -l maprh lcpu.bath.ac.uk — 80×35

```
-bash-4.1$ ls -l
total 12892
-rwxr--r--.  1 maprh map 1146739 Mar 14  2002 2_collision.pdf
-rwxr--r--.  1 maprh map 1189508 Oct 11  2001 747-300 Korean Air in Guam.pdf
-rwxr--r--.  1 maprh map  238425 Apr  3  2002 A320_Bahrain.pdf
-rwxr--r--.  1 maprh map    3785 Jul 16  2003 AAR0001.htm
drwxr-xr-x. 18 maprh map    4096 Dec  5  2003 accident summaries
drwxr-xr-x.  2 maprh map    4096 Dec  5  2003 ASN Aircraft accident description
31 JUL 1992 Thai Airways Airbus A_310 HS-TID_files
-rwxr--r--.  1 maprh map   15451 Jul 16  2003 ASN Aircraft accident description
31 JUL 1992 Thai Airways Airbus A_310 HS-TID.htm
drwxr-xr-x.  2 maprh map    4096 Dec  5  2003 Bangkok
drwxr-xr-x.  5 maprh map    4096 Dec  5  2003 Cali
-rwxr--r--.  1 maprh map  509901 May 10  2002 dasc98.ps
-rwxr--r--.  1 maprh map   47319 Oct 11  2001 inhchart.pdf
-rwxr--r--.  1 maprh map 7655946 Feb 20  2002 Jan_1982_N62AF_737_icing.pdf
-rwxr--r--.  1 maprh map  225792 Mar 24  2002 List of accidents.xls
-rwxr--r--.  1 maprh map  197669 Oct 11  2001 njl.pdf
drwxr-xr-x.  2 maprh map    4096 Dec  5  2003 not that useful
drwxr-xr-x.  2 maprh map    4096 Dec  5  2003 NTSB
drwxr-xr-x.  2 maprh map    4096 Dec  5  2003 NTSB - KAL801 Public Hearing_files
-rwxr--r--.  1 maprh map    4248 Jul 16  2003 NTSB - KAL801 Public Hearing.htm
drwxr-xr-x.  2 maprh map    4096 Dec  5  2003 resource.frk
drwxr-xr-x.  2 maprh map    4096 Dec  5  2003 Transcripts Between Guam Airport T
ower and KA801 before Crash_files
-rwxr--r--.  1 maprh map    5196 Jul 16  2003 Transcripts Between Guam Airport T
ower and KA801 before Crash.htm
-rwxr--r--.  1 maprh map  219983 Nov  9  2001 trffc_conf.pdf
drwxr-xr-x.  2 maprh map    4096 Dec  5  2003 UK Air Accidents Investigation Bra
nch1_files
-rwxr--r--.  1 maprh map  197582 Jan  8  2002 UK Air Accidents Investigation Bra
nch1.htm
-rwxr--r--.  1 maprh map 1256228 Oct 11  2001 vh-inh.pdf
-rwxr--r--.  1 maprh map  117949 Oct 11  2001 vh-jsi.pdf
-bash-4.1$
```

The Shell vs. the Operation System

- The shell is just one program among many
  - Many different ones have been written
  - **sh** was the first for Unix
    - ★ Most others extend its capabilities in various ways
    - ★ Which means that it's the lowest common denominator you can always rely on
  - We will use **bash** (the Bourne again shell)
    - ★ Available just about everywhere
    - ★ Even on Windows (thanks to Cygwin)

Aside: Cygwin

- Cygwin is (according to the project webpages at https://www.cygwin.com)
  - a large collection of GNU and Open Source tools which provide functionality similar to a Linux distribution on Windows.
  - a library (cygwin1.dll) which provides substantial POSIX API functionality.
- you can, if you would like, download, install and use cygwin on your Windows machines

- Note, however, that while Cygwin can look like the University UNIX machines (lcpu)
- It is not a University UNIX machine
- If you develop your coursework code using Cygwin, check that it works on lcpu before submitting it

The Shell vs. the Operation System

- As introduced above, the shell is just one program among many
- In contrast, the operating system is not just another program
  - Automatically loaded when the computer boots up
  - The only program that can talk directly to the computer's hardware
    - ★ I.e., read characters from the keyboard, or send drawing commands to the screen
  - Manages files and directories on the disk
  - Keeps track of who you are, and what you're allowed to do
  - You can run many instances of the shell on a computer at once, but it can only run one operating system at a time

The File System

- The file system is the set of files and directories the computer can access
  - Everything that stays put when you turn the computer off and restart it

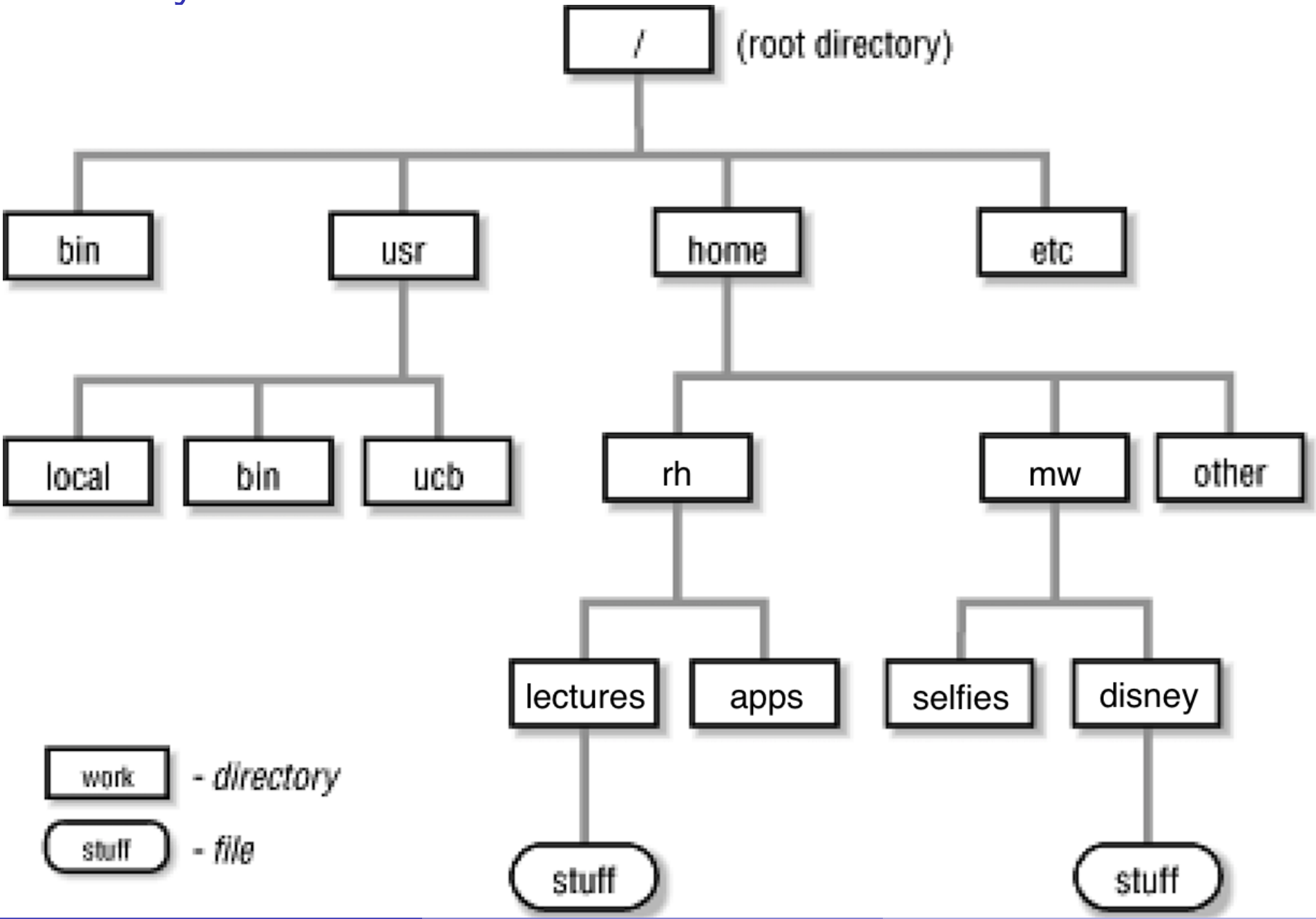
The File System

- Data is stored in files
  - By convention, files have two part names, like notes.txt or home.html
  - Most operating systems allow you to associate a filename extension with an application
    - E.g., .txt is associated with an editor,
    - and .html with a web browser
  - But this is all just convention: you can call files (almost) anything you want

The File System

- Files are stored in directories (often called folders)
  - Directories can contain other directories, too
  - Results in a directory tree

Directory Structure



Drives

- On Unix, the file system has a unique **root directory** called /
  - Every other directory is a child of it, or a child of a child, etc.
- On Windows, every **drive** has its own root directory
  - So C:\home\rh\notes.txt is different from J:\home\rh\notes.txt

Paths

A **path** is a description of how to find something in a file system

- An **absolute path** describes a location from the root directory down
  - Equivalent to a street address
  - Always starts with " /"
  - E.g., /home/rh is my home directory,
  - and /home/rh/courses/prog1/UNIX1 is this file
- A **relative path** describes how to find something from some other location
  - Equivalent to saying, Four blocks north of here, and seven blocks east
    - E.g., from /home/rh,
    - the relative path to this file is courses/prog1/UNIX1

Special Paths

- Two special names:
  - "." means the current working directory
  - ".." means the directory immediately above this one
    - ★ Also called the parent directory
  - In /home/rh/courses/prog1/,
  - .. is /home/rh/courses/

Special Paths

- Every program (including the shell) has a (current) working directory
  - Where am I?
  - Relative paths are deciphered relative to this location
  - Can change while a program is running

pwd and ls

- **pwd** prints (shows you) the current working directory
- **ls** lists the contents of that directory

```
-bash-4.1$ pwd
/u/w/maprh/accident reports
```

```
-bash-4.1$ ls
2_collision.pdf
747-300 Korean Air in Guam.pdf
A320_Bahrain.pdf
AAR0001.htm
accident summaries
ASN Aircraft accident description 31 JUL 1992 Thai Air
ASN Aircraft accident description 31 JUL 1992 Thai Air
Bangkok
Cali
dasc98.ps
inhchart.pdf
-bash-4.1$
```

More on ls

What actually happens when I type ls is:

- The operating system reads characters from the keyboard
- Passes them to the shell (because it's the currently active window on my desktop)
- The shell breaks the line of text it receives into words
- Looks for a program with the same name as the first word (i.e., the command to run)
- Runs that program
- Reads the program's output and sends it back to the operating system for display

Flags

- **Flags are options that you can add to commands**
- E.g. can tell ls to produce more informative output by giving it some flags
- By convention, flags start with "-", as in "-a" or "-l"

Flags

- For example: show directories with trailing slash

```
-bash-4.1$ ls -F
2_collision.pdf*
747-300 Korean Air in Guam.pdf*
A320_Bahrain.pdf*
AAR0001.htm*
accident summaries/
ASN Aircraft accident description 31 JUL 1992 Thai Air
ASN Aircraft accident description 31 JUL 1992 Thai Air
Bangkok/
Cali/
```

- -a: gives you all files starting, including those, which are normally hidden
- -l: provides long listing format i.e. provides more information

Finding your way

- man pages: provide an overview of the functionality of a command.
  - !man ls!
- apropos: provides all commands related to a certain topic
  - !appropos(permissions)!
- !–help!: provides support for a specific command
  - !ls –help!

Manipulating Files and Directories

- Ultimately, we will want to create, use and delete our own files/directories
- in oder to do that, we will need additional Unix commands:
  - mkdir - create a new directory
  - cd - change directory
  - cp - copy a file

Manipulating Files and Directories

Working through an example:

- create a new directory called temp

```
mkdir temp
```

- Note: no output
- The -v (verbose) flag tells mkdir to print a confirmation message

Manipulating Files and Directories

- Now go into that directory

```
cd temp
```

- Changes the shell's notion of our current working directory

```
pwd
/home/rh/programming1/temp
```

Manipulating Files and Directories

- No files there yet:

```
ls -a

.  ..
```

- Use the editor of your choice (emacs,vim) to create a file called earth.txt with the following contents:

```
Name:  Earth
Period:  365.26 days
Inclination:  0.00
Eccentricity:  0.02
Object:  Planet
```

Manipulating Files and Directories

- The easiest way to create a similar file venus.txt is to copy the one we have

```
cp earth.txt venus.txt

ls -t
venus.txt    earth.txt
```

- Note: the syntax of the cp command (make a copy of earth.txt called venus.txt)
- Note also: the -t option tells ls to list newest first (i.e. list in time order)

Manipulating Files and Directories

- Check the contents of the file using cat (short for concatenate)
- prints the contents of a file to the screen
- You can also use more or less

```
cat venus.txt
```

Manipulating Files and Directories

- Edit the file as follows:

```
Name:  Venus
Period:  224.70 days
Inclination:  3.39
Eccentricity:  0.01
Object:  Planet
```

- Compare the sizes of the two files using wc (for word count)

```
wc earth.txt venus.txt

  4   9  69 earth.txt
  4   9  69 venus.txt
  8  18 138 total
```

Manipulating Files and Directories

- You can also compare the files using the **diff** command
- diff prints details of the differences between the files

```
diff earth.txt venus.txt

1,4c1,4
< Name: Earth
< Period: 365.26 days
< Inclination: 0.00
< Eccentricity: 0.02
---
> Name: Venus
> Period: 224.70 days
> Inclination: 3.39
> Eccentricity: 0.01
```

Manipulating Files and Directories

- Unix/Linux does not care about filename extensions.
- cp earth.txt earth.pdf! is valid
- though not a very sensible thing to do

- we can rename earth.pdf using mv earth.pdf earth2.txt

Manipulating Files and Directories

- Removing a file can be done using **rm**

```
rm earth2.txt
```

- A empty directory can be removed with rmdir or rm −r which recursively removed all files.

Next Steps

- A practical next step to work through the university pages on UNIX/Linux
  - http://www.bath.ac.uk/bucs/tools/unix/basicunixcommands/
- Since many of you have asked, it is also worth noting that the University maintains a page on how to access the campus machines from outside (e.g. from your own laptops)
  - http://www.bath.ac.uk/bucs/networking/ssh.html
- We will look at connecting to one computer from another in the next UNIX lecture
- .... but, for now, using the applications recommended by the University is not a bad starting point
- As always, help is available in the labs