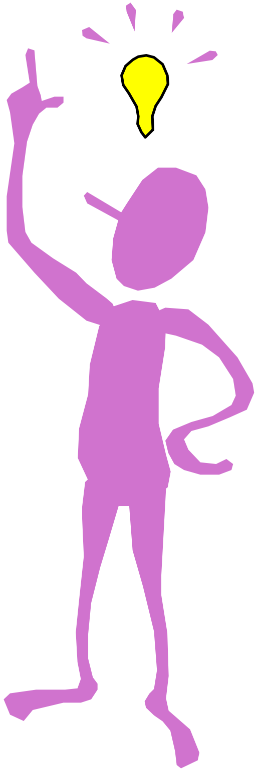
Introduction to Linux (Part I)



BUPT/QMUL

2017/03/01

Contents

1. Background on Linux
2. Starting / Finishing
3. Typing Linux Commands
4. Commands to Use Right Away
5. Linux help

continued

Contents

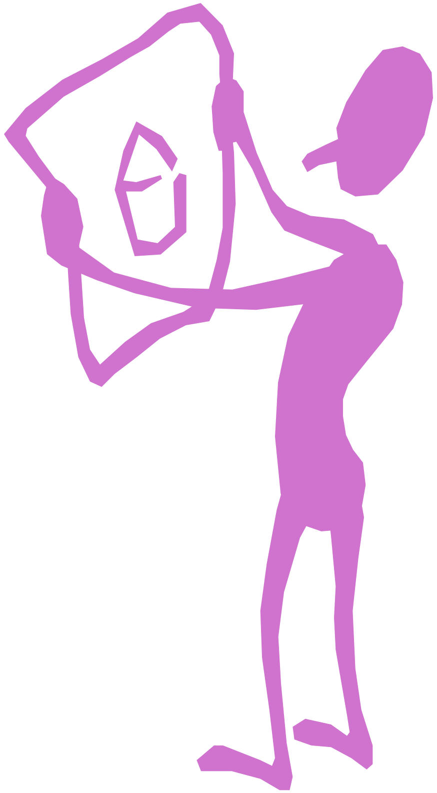
6. The File System
7. Working with Directories
8. Working with Files
9. Communicating with People

1. Background on Linux

1.1. What is Linux?

1.2. History

1.3. Why use Linux?



1.1. What is Linux?

- Linux [ˈ*lɪnəks*] **:** a Unix-like computer operating system but free and open-source.
- The Linux Operating System (OS) is a large program (mostly coded in C) that turns the computer into a useable machine.
- It provides a number of facilities:
  – management of hardware resources
  – directory and file system
  – loading / execution / suspension of programs

1.2. Brief History

- 1969 First UNIX at Bell Labs
- 1975 Bell Labs makes UNIX freeware
- 1970’s Berkeley UNIX (BSD)
- 1980’s TCP/IP
  MIT X-Windows
- 1990’s The Web,
  Linux Kernel (1991, by Linus Torvalds)
- 1993 Debian Linux
- 2003 Fedora Linux
- 2004 Ubuntu Linux, CentOS Linux

1.3. Why Use Linux?

- multi-tasking / multi-user
- lots of free software
- networking capability
- graphical (optional)
- easy to program
- stable
- portable (PCs, mainframes, super-computers)

continued

- free! (Ubuntu, CentOS, FreeBSD)
- open source
- popular
- not tied to one company
- active community
- lots of documents

2. Starting / Finishing

2.1. Your Account

2.2. Login to your Account

2.3. Password Tips

2.4. Logout from your Account

2.1. Your Account

- Each user has their own space, called their *account*.

- Type your login ID and password to enter your account.

- Only if the login ID and password match will you be let in.

2.2. Login to your Account

`login: ad`	You type your ID and RETURN.
`Password:`	You type your password and RETURN. It does not appear.
`$`	The Linux prompt (or similar). You can now enter commands.

`Access denied`
`Password:`
	Login ID and password not match

2.3. Password Tips

- NEVER tell anyone your password.
- Don’t write it down.
- A good password is:
  - 8 (or more) characters long
  - uses a mix of uppercase and lowercase letters, numbers, and symbols (e.g. #, %).
- You can change your password with the `passwd` command (see later).

2.4. Logout from your Account

```
logout
```

or

```
^D
```

Press CONTROL and D together

or

```
exit
```

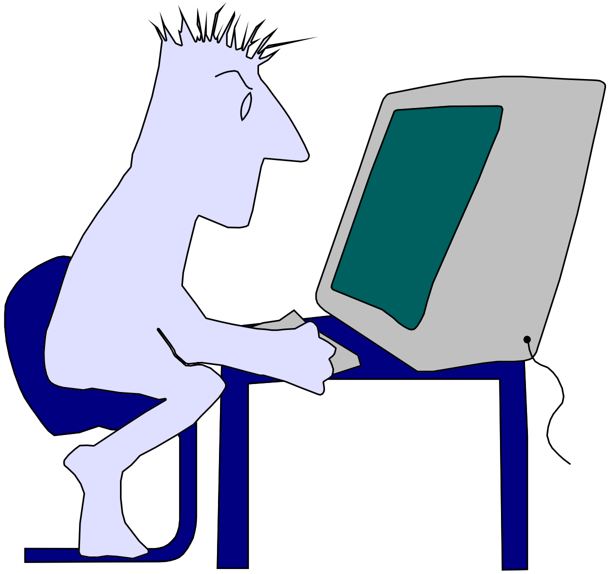
3. Typing Linux Commands

3.1. The Shell

3.2. Typing Commands

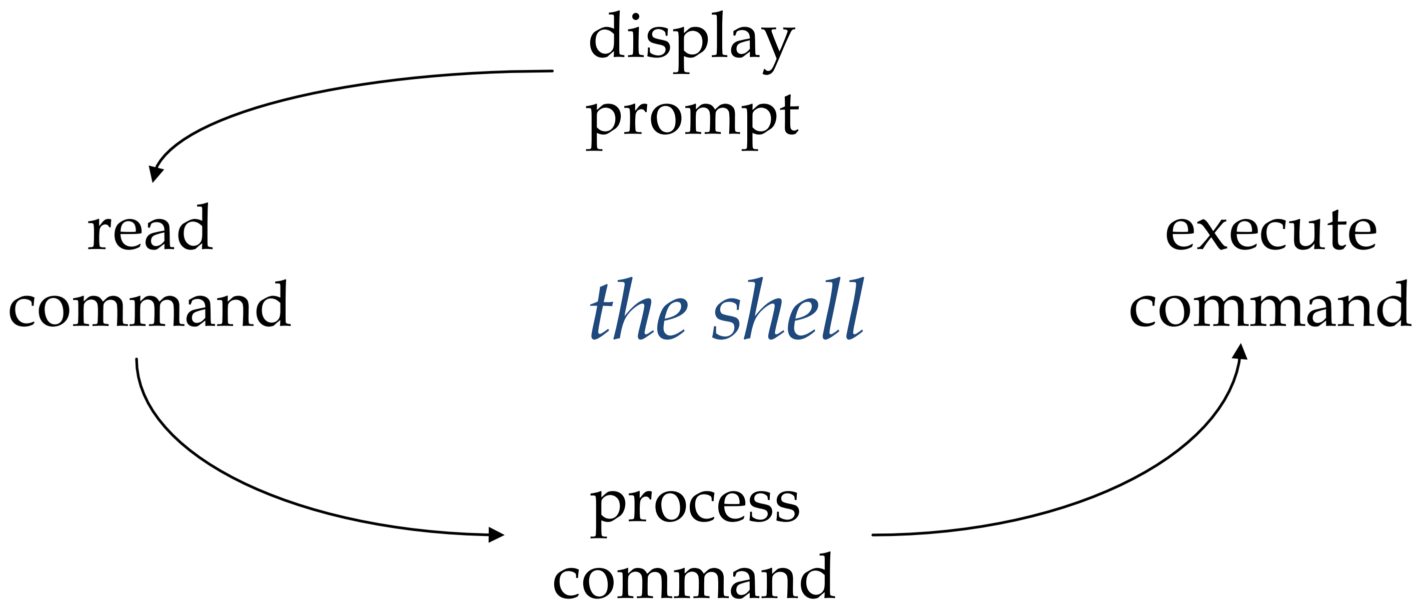
3.3. Control Characters

3.4. Changing your Password



3.1. The Shell

- The Linux user interface is called the *shell*.
- The shell does 4 jobs repeatedly:



display prompt

read command

the shell

execute command

process command

3.2. Typing Commands

- Try these:

```
date
cal 3 1997
who
ls
ifconfig
man cal
```

Press enter for next line;

Press spacebar for next page;
^C or q to stop

```
clear
```

3.3. Control Characters

- Erasing characters

`Backspace`	delete last character
`^H`	delete last character (press CONTROL and H together).
`^W`	delete last word
`^U`	delete the line

- Very useful control characters

`^C`	terminate command
`^S`	suspend output
`^Q`	resume output

3.4. Changing your Password

- The command is:

```
passwd
```

- It will ask you for the new password twice.

4. Command to Use Right Away

4.1. Date Commands

4.2. You and the System

4.3. Calculators

4.1. Date Commands

- **`date`** Gives time and date

- **`cal`** Calendar

```
cal
cal 1997
cal 3
cal 7 1962
cal 9 1752
```

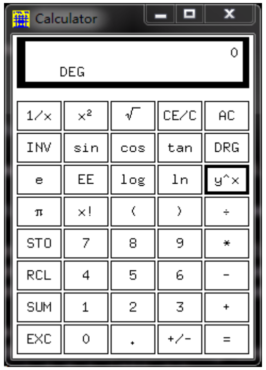
Not a mistake. Why?

4.2. You and the System

`uptime`	Machine’s ‘up’ time
`hostname`	Name of the machine
`whoami`	Your account name

4.3. Calculators

- `xcalc` Requires x11-apps (sudo apt-get install x11-apps)



- **`expr`** `e` Simple arithmetic

  ```
  expr 3 + 5 + 7
  ```

- **`bc`** Programmable Calculator

Using bc

- `bc`

```
3 + 5 + 7
   15          Output
^D
```

- `bc -l`   Use Maths library

```
scale=3        Set display to 3 dp
150/60
l(30)          natural log function
^D
```

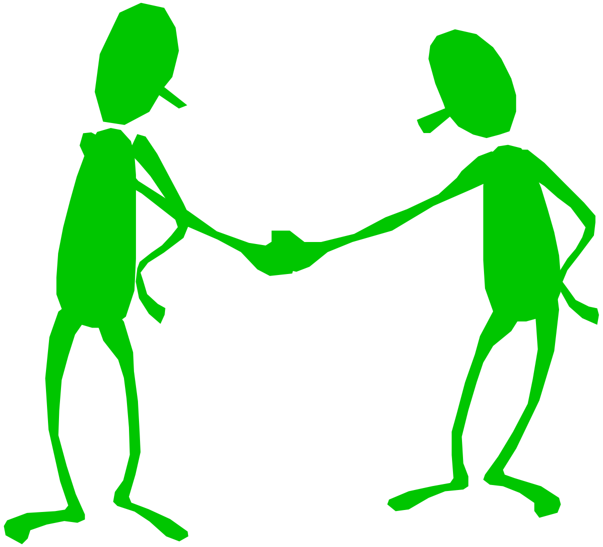
- `bc`

```
obase=2               Output base
ibase=16              Input base
FFC1
^D
```

5. Linux Help

5.1. On-line Help

5.2. Linux Reference



5.1. On-line Help

- **`man`** Manual pages

```
man cal
man man
```

- `apropos` *`topic`* Lists commands related to `topic`

```
apropos game
apropos help
```

man -k topic	Same as apropos
whatis cmd	One-line description
whatis find	
where cmd	Location of command
which cmd	Location

- **`locate`** `cmd` List files with `cmd` in their name (or path)

```
locate game
```

- Output of these commands can be very long. See one screenful at a time with: `| more`

```
locate game | more
apropos print | more
```

- Press enter/spacebar to go on; `^C/q` to stop.

5.2. Linux Reference

- **“Linux Documentation Project Guides”, include:**
  - “Introduction to Linux ”
  - “GNU/Linux Command-Line Tools Summary”
  - “The Linux System Administrators' Guide”
  - “Advanced Bash-Scripting Guide”
  - “Bash Guide for Beginners”
  - “The Linux Programmer's Guide”

  (website: http://tldp.org/guides.html)
- **“Linux Network Administrators Guide”**
  - website: http://www.faqs.org/docs/linux_network
- **“Beej's Guide to Network Programming”**
  - website: http://beej.us/guide/bgnet

6. The Linux File System

6.1. An upside-down Tree

6.2. Some System Directories

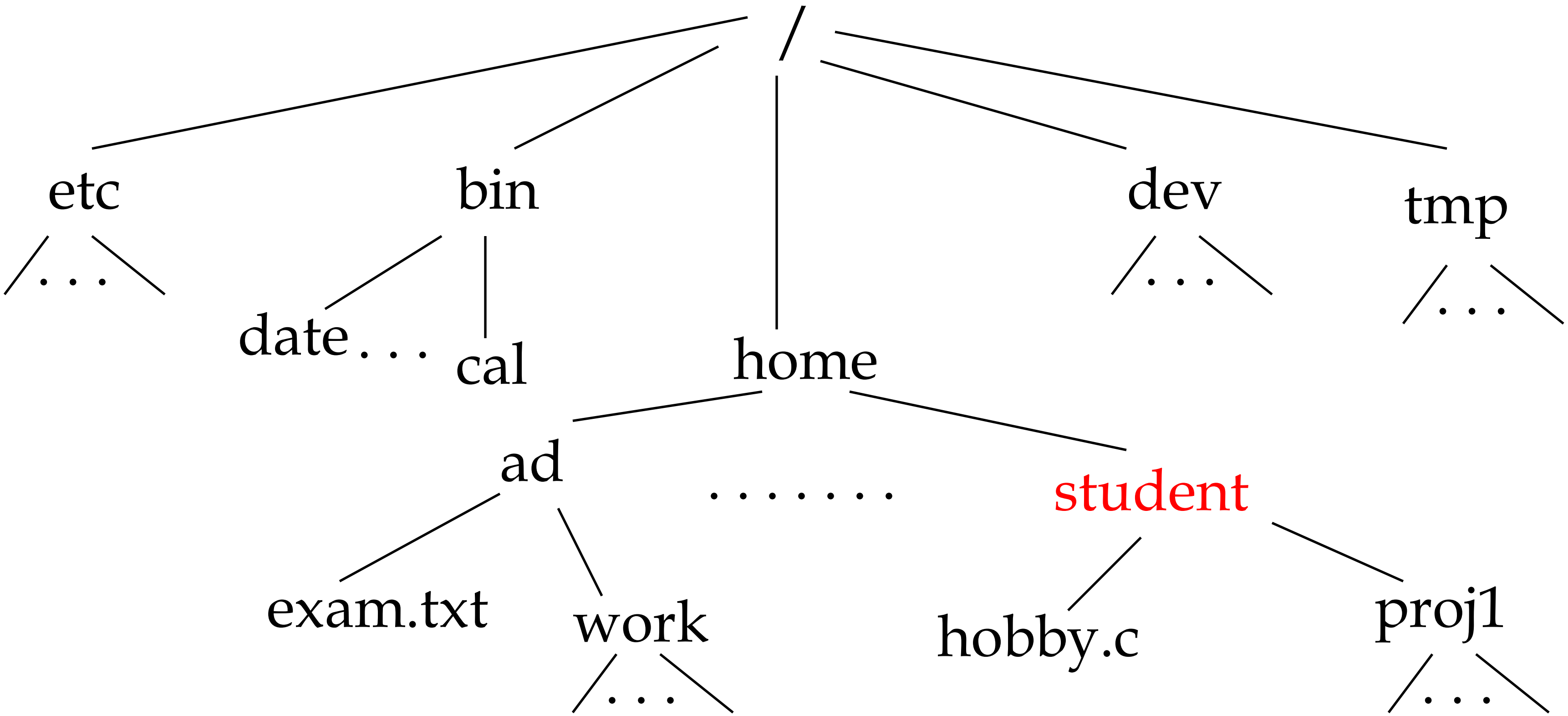
6.3. Where do you login?

6.4. Pathnames

6.5. Commands and Pathnames

6.1. An upside-down Tree

- A simplified Linux directory/file system:

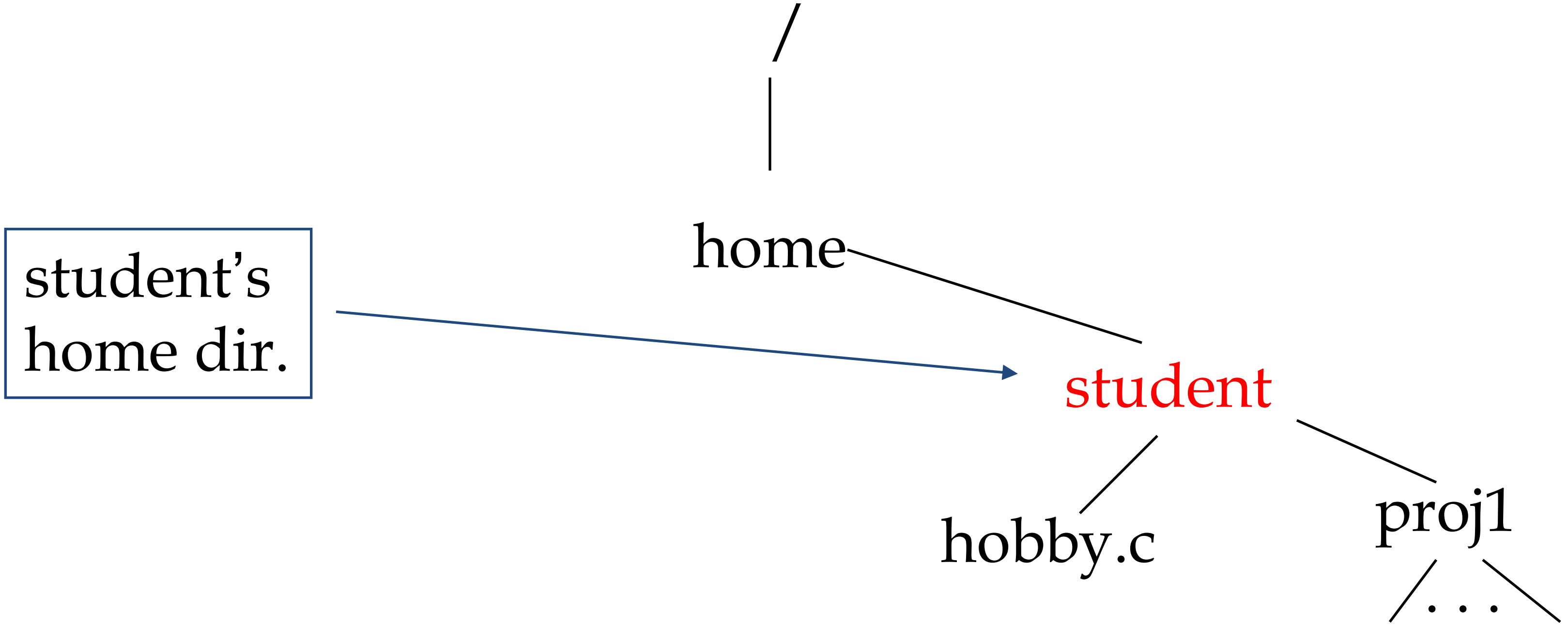


6.2. Some System Directories

- `/` *root* directory
- `/bin` commands
- `/etc` system data files
  (e.g. `/etc/passwd`)
- `/dev` files representing I/O devices

6.3. Where do you login?

- Your *home directory*, which is named after your login ID.
- **`pwd`** showing current directory



root Account

- root: super user, who has the full rights to all resources
- prompt: #
- home directory: /root
- changing to root account: $sudo -i
- add a new user account:

```
#adduser account-name
```

- open a new terminal window: ALT+F2
- switching among terminals: ALT+F1, ALT+F2, …
- try command who and whoami

6.4. Pathnames

- A *pathname* is a sequence of directory names (separated by /’s) which identifies the location of a directory.

- There are two sorts of pathnames
  - absolute pathnames
  - relative pathname

Absolute Pathnames

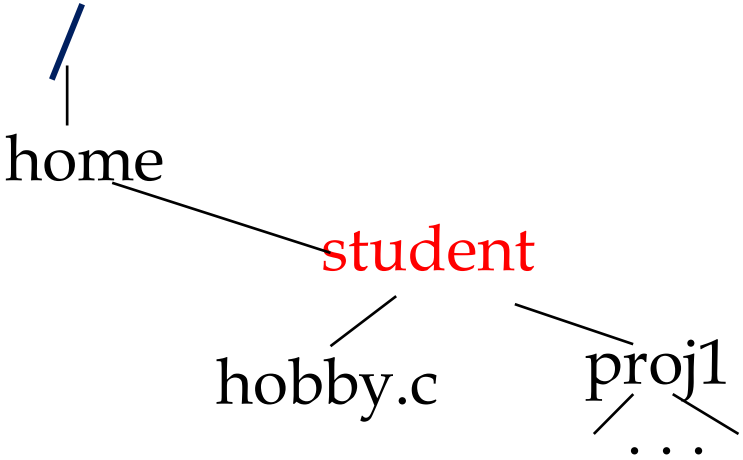
- The sequence of directory names between the top of the tree (the *root*) and the directory of interest.
- For example:

```
/bin
/etc/terminfo
/home/ad
/home/student/proj1
```

Relative Pathnames

- The sequence of directory names below the directory where you are now to the directory of interest.
- If you are interested in the directory `proj1`:

`proj1`	if you are in `student`
`student/proj1`	if you are in `home`



6.5. Commands and Pathnames

- Commands often use pathnames.

- For example:

```
cat /etc/passwd
```

List the password file

7. Working with Directories

7.1. Moving between Directories

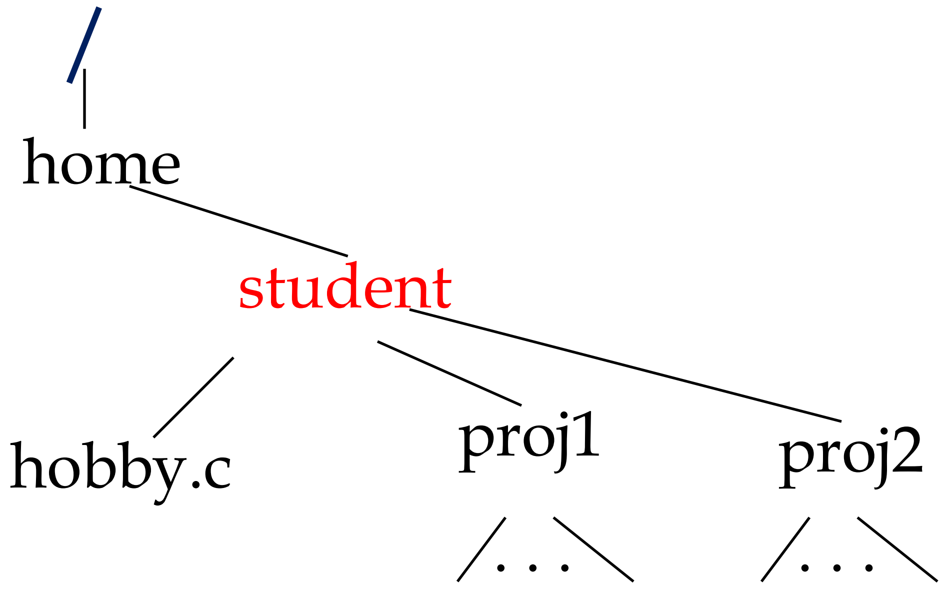
7.2. Special Directory Names

7.3. Investigate the System

7.4. Making / Deleting / Renaming Directories

7.1. Moving between Directories

- student's home directory:



- If you are in directory `student` how do you move to directory `proj1`?

```
cd proj1
```

- You are now in `proj1`. This is called the *current working directory*.

- **`pwd`** Print name of current working directory

- Move back to directory `student` (the parent directory):

```
cd ..
```

- When in `proj1`, move to `proj2` with one command:

```
cd ../proj2
```

- `../proj2` is a *relative* pathname

7.2. Special Directory Names

• /	The root directory
• .	The current working directory
• ..	The parent directory (of your current directory)
• ~	Your home directory
• `~user`	Home directory of `user`

Examples

• `cd /`	Change to root directory
• `cd ~`	Change to home directory
• `cd`	(Special case; means `cd ~`)
• `cd ~ad`	Change to ad’s home dir.
• `cd ../..`	Go up two levels.

7.3. Investigate the System

- Use `cd`
- **`cat`** `file` List `file`

```
cd /etc
cat passwd
```

- **`ls`** Directory listing

`ls`	List current dir.
`ls /etc`	List `/etc`
`ls -F`	-F option shows types

```
          a.tcl*          ns-allinone-2.29/
```

7.4. Making / Deleting / Renaming Directories

- Usually, you can only create directories (or delete or rename them) in your home directory or directories below it.

`mkdir`	Make a directory
`rmdir`	Delete a directory
`mv`	Rename a directory

- Create a `lab` directory in your home directory:

```
cd ~
mkdir lab
```

- Create two directories inside the `lab` directory:

```
cd lab
mkdir week1
mkdir week2
```

- Delete the `week1` directory:

```
rmdir week1
```

- Change the name of `week2` to `all-weeks`

```
mv week2 all-weeks
```

8. Working with Files

8.1. Creating a Text File

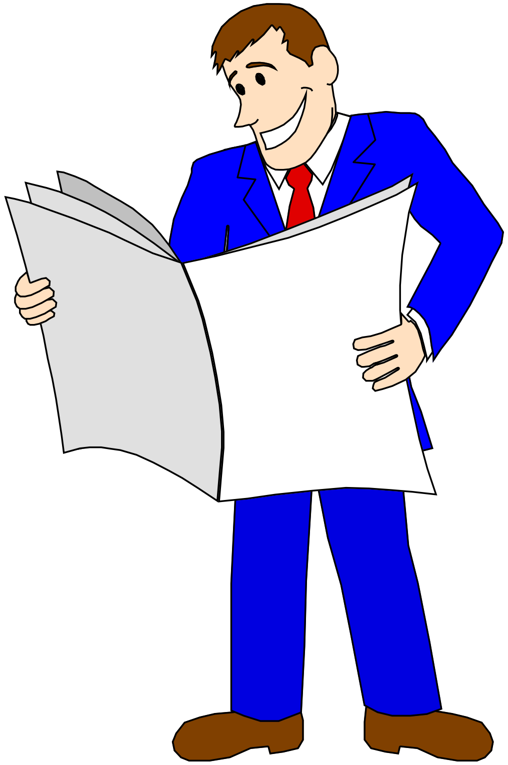
8.2. Listing Files

8.3. Filename Conventions

8.4. Other Basic Commands

8.5. Printing

8.6. I/O Redirection



8.1. Creating a Text File

- A quick way:

```
cal > file
```

- This will feed the text you type at the keyboard into `file` until you type `^D` (CONTROL and a D together).

- A more powerful way is to use **`vi`**, a full screen editor (see later).

8.2. Listing Files

- **`cat`** `file` List the file

```
cat hobby.c
cat /etc/passwd
```

- **`more`** `file` List the file a screen at a time. Type spacebar to go on; `^C` to stop

- **`less`** `file` Like `more` but more powerful
- **`head`** `file` List the *first* few lines
- **`tail`** `file` List the *last* few lines

8.3. Filename Conventions

- Many files have a name and an extension:

`file.c`	A C program
`file.cpp`	A C++ program
`file.txt`	A text file

- However, you can call a file *anything*. It doesn't have to have an extension.

8.4. Other Basic Commands

• **`cp`** `file1 file2`	Copy `file1`, making `file2`
• **`mv`** `file1 file2`	Rename `file1` as `file2`
• **`rm`** `file` `rm -i file`	Delete `file` check first

- **`wc`** `file` Counts the lines, words, characters in `file`

- **`grep`** `string file` Search `file` for `string`

  e.g., `grep abc test`

- List lines containing ‘Andrew’ in `/etc/passwd`

```
grep Andrew /etc/passwd
```

- Lines containing ‘printf(‘ in `hobby.c`

```
grep ‘printf(‘ hobby.c
```

- Lines starting with ‘loca’ in `/usr/dict/words`

```
grep ^loca /usr/dict/words
```

8.5. Printing

- `lpr file` Print `file`

- `lpq` List the print queue. Each print job has a number.

- `lprm job-number` Remove that print job

- You may have to name the printer with the `-P` option:

```
lpr -Plj5 hobby.c
```

- `lpq` and `lprm` understand the `-P` option

8.6. I/O Redirection

- Most commands output to the screen

```
ls
```

- Output can be *redirected* to a file with‘>‘:

```
ls > dir.txt
cal 1997 > year1997
```

- Output can be *appended* to a file with ‘>>‘

```
cal 1997 > years
cal 1998 >> years
```

- Concatenate two files:

```
cat f1 f2 > fs
```

- Input redirection (less common) uses ‘<‘

```
wc < years
```

- Combine input and output redirection:

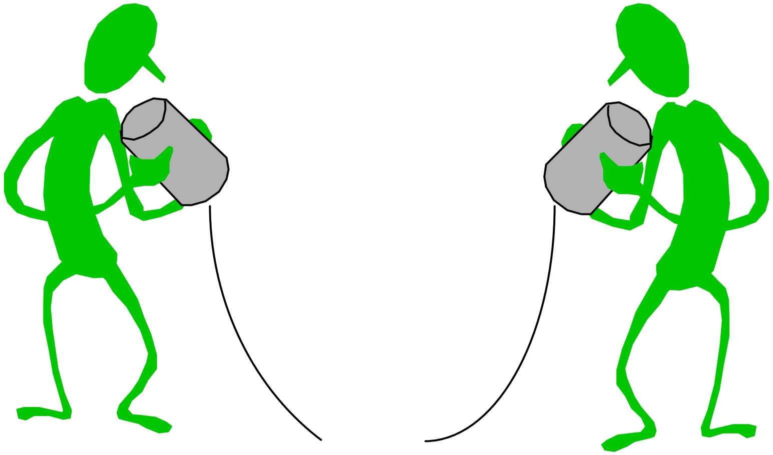
```
wc < years > year-counts
```

9. Communicating with People

9.1. Information on Others

9.2. Fingering People

9.3. Talking



9.1. Information on Others

- **`users`** Who else is logged on?
- **`who`** Information on current users
- **`ps`** What are people doing?

- **`w`** What are people doing?

  `w -sh` A shorter report

- Examine password info:

```
more /etc/passwd
grep s38 /etc/passwd
```

9.2. Fingering People

• **`finger`**	Info. on current users
`finger -l`	Longer information
• `finger user`	Information on `user` (need not be logged in)
`finger ad`	

- `finger @machine-name` User info. for that machine

```
finger @catsix
finger @ratree.psu.ac.th
```

- **`ping`** `machine-name` Is machine alive (on)?

```
ping catsix
```

(`^C` to stop)

Your Finger Information

- `chfn`   Change your finger entry
- `finger` also prints the contents of the `.plan` and `.project` files in your home directory. List '.' files with:

```
 ls -a
```

9.3. Talking

- `talk user` Talk to `user` (on any machine)

```
talk ad
talk bill-gates@ratree.psu.ac.th
```

Get out by typing `^C`