Introduction to Computing

Michaelmas Term

Prof D W Murray

Lecture 1

1. the machines and network
2. how to login
3. the windowing environment
4. how your filestore should be organized.
5. moving around the filestore
6. redirection
7. using electronic mail
8. how to logout
1 The machines and the network

All you need to know about hardware is to distinguish between the keyboard, mouse and screen, but ...

![Diagram of the main components of the workstation](image)

Figure 1: The main components of the workstation.

There are two main methods of conveying your intentions to the computer:

1. Via the keyboard
   - type command line
   - interpreted by the OS as program to execute, etc

2. Via the Mouse
   - mouse position + button clicks ordrag
   - interpreted according to software context
What is the Operating System?

The OS is a program which

- runs continually
- monitors requests for system resources
- allocates resources, but in limited amounts

The Suns use a version of the Unix operating system

- “kernel” deals with the details of hardware,
- “layers” of software protect the user from the detail.

Programs

Many programs for housekeeping come with the OS itself — some you use (like cp, mkdir, etc.) — others grind away unseen in the background.

Other programs are tailored to engineers

- programming languages (like C, Matlab etc),
- scientific word processing packages (eg Latex),
- powerful text-editors

Finally, there are programs that you the user write.
Filestore

- You have a private store for your own programs and data files

- Your files are not held on the local disks, but on much larger disks attached to a *fileserver*

- All the machines are linked via an Ethernet to each other and to a fileserver.

- The server "exports" the filesystem to all machines in the Solarium.

- So ...

  1. A single copy of a program can be used by many users
  2. You can use any machine in the Solarium
The network

The local network attaches, via a gateway machine, to the University-wide backbone.

As do other local networks in Colleges and other Departments.

At OUCS, the University backbone connects to the national network and thence to the international network.

The network allows

- sending of electronic mail to others (mail)
- direct logging in to remote machines (rlogin, telnet)
- retrieval of files from other machines (ftp)
- connection to WWW servers (netscape)
Two Warnings about Hardware

- **Never switch off the computer’s system box ...**
  - it will corrupt the file system
  - will destroy the work of any “remotely logged in” users.

- **Never move or jolt the system box ...**
  - it can cause a disk crash
2 Logging in

- Type in username and password..

  \[
  \text{login: u98xxx} \leftarrow \\
  \text{password: secret} \leftarrow \\
  \]

- After messages appears you will obtain prompt like:

  \text{b42.ecs:/ugrads/u98xxx} %

- The prompt tells you
  - your machine — \text{b42.ecs}
  - your “current working directory” — \text{/ugrads/u98xxx}

- Just after logging in, the os sets you in your \textbf{home directory}.

- Follow the lab sheets, and change the given password to something only you know. Type

  \text{b42.ecs:/ugrads/u98xxx} % \text{ passwd} \leftarrow

  and follow the instructions.

- \textbf{NEVER release your password}
- either either outside or inside the Dept
The Unix prompt

- The prompt is a prompt for you to type a command to the operating system. Unix is ready and waiting.

```
b42.ecs:/u/ecs/u98xxx% command ←
```

- Commands are the names of executable programs, which we saw a moment ago either
  - come with the OS, or
  - are packages already installed, or
  - are programs you have written

- E.g., try the Unix program `cal`

```
b42.ecs:/u/ecs/u98xxx% cal ←
```

and a calendar for this month will appear.

- However, using a workstation from the console alone is wasteful.

```
- You need to exploit a GUI
- a Graphical User Interface
```
3 Openwindows

- Type

\[ \texttt{b42.ecs:/users/u98xxx\% openwin} \]

A windowing system built on X-windows, called Openwindows, appears.

- The initial appearance of your workspace has
  - a command tool "cmdtool" CONSOLE in the top left
  - a cmdtool window in the bottom left
  - a textedit window in the top right.

![Diagram of Openwindows workspace]

Figure 2: Your desktop after starting Openwindows

- You can customize the layout — but don’t bother at first.
What are they for?

- The CONSOLE cmdtool (top left) is running a Unix shell. We could type commands into it, but it is where system messages appear, so it is best to leave it alone.

- The other cmdtool at the bottom left is also running a Unix shell. Make sure the mouse pointer is lying within the window. As before, the prompt indicates Unix is ready and waiting.

- Now try typing

  \[ \texttt{b42.ecs:/users/u98xxx}\% \texttt{cal} \rightarrow \]

  and again a calendar for this month will appear.

- So,

  - You can have several cmdtools open
  - Each is an independent window into the machine

- The textedit window is for writing and editing files. It does not accept Unix commands!!!

- So,

  - there are many types of specialist windows or screens (eg cmdtool, textedit, mailtool, calctool)
    each with its own purpose.
The mouse

- To make a different open window active, just point the mouse at it.
- In Openwindows, the three mouse buttons are use for Selection (LEFT), Adjusting (RIGHT) and Menu getting (RIGHT)

Moving, Resizing, Scrolling:

- Closing, Reopening: Top left button, double click on icon.
- Quitting: This option (and some of the above) is available by selecting from a menu when you hold down the Menu (RIGHT) button over the top bar of the window.
Starting up new windows

- Apart from memory, no limit to number of windows.
- To open a new windows:
  1. Holding down the Menu (RIGHT) button on blue desktop background
  2. "Workspace" Menu appears.
  3. Slide along an item through arrow
  4. Program selection menu appears
  5. Choose one by releasing button

In one motion, and holding down the button, you drag right across the menu item, and the submenu appears ...
4 Organizing your filestore

- Information stored on hard disk in logical chunks called files. Two important sorts of files are:
  - text (ascii) files — legible by you and the machine
  - binary files — legible only by machine
- Text files are created either
  - by using a text editor
  - as the output from a program
- Eg, the output of the date program can be store in datefile using the redirect arrow:

  b42.ecs:/users/u98xxx% date > datefile

- Now do a "long listing" of the files in the directory:

  b42.ecs:/users/u98xxx% ls -l
  total 1
  -rw-r--r--  1 u98xxx  29 Oct 12 15:20 datefile

- We could view it on the screen by typing

  b42.ecs:/users/u98xxx% more datefile
  Mon Oct 12 15:20:05 BST 1998

- We could carry on making files here in the home directory, but ...
  - it would end up a HUGE MESS.
- Much better to arrange files in related groups called directories.
Directories

- The Unix command to create directories is `mkdir` (make directory).

  `b42.ecs:/users/u98xxx% mkdir project1`  ➔
  `b42.ecs:/users/u98xxx% mkdir project2`  ➔
  `b42.ecs:/users/u98xxx% mkdir lsm2`  ➔

- Your filestore now looks like:

```
_u97xxx_
```

  ![Diagram of Directory Structure]

- and if you do "ls -l" you see

```
total 4
drwx--x--x  2 u98xxx   512 Oct 12 15:25 lsm2
-rw-------  1 u98xxx   29 Oct 12 15:20 datefile
drwx--x--x  4 u98xxx   512 Oct 12 15:26 project1
drwx--x--x  2 u98xxx   512 Oct 12 15:26 project2
```
5 Moving around the filestore

To move between directories use cd.

Change directory (cd) into project1 and make two new directories, progs and docs.

b42.ecs:/ugrads/u98xxx% cd project1
b42.ecs:/ugrads/u98xxx/project1% mkdir progs
b42.ecs:/ugrads/u98xxx/project1% mkdir docs

NB! the prompt reminds you of your new working directory.

Make some files in the the docs directory

b42.ecs:/ugrads/u98xxx/project1% cd docs
b42.ecs:/ugrads/u98xxx/project1/docs% date > datefile
b42.ecs:/ugrads/u98xxx/project1/docs% cal > october

Is the second datefile the same as the first?

NO! — but how are they distinguished?
Names may appear relative, but are really absolute

Commands like `mkdir project1` and `date > datefile` create directories or files in the current working directory.
It appears that filenames are described relative to the c.w.d.
This is not the whole story — file names are really absolute with respect to the filesystem’s root “/”.

The two datefiles we created have full names:

```
/ugrads/u98xxx/datefile.
/ugrads/u98xxx/project1/docs/datefile
```

To check this out let us sit in docs, and view the files using their full names:

```
+xxx/project1/docs% more /ugrads/u98xxx/datefile
Mon Oct 12 15:20:05 BST 1998
+xxx/project1/docs% more /ugrads/u98xxx/project1/docs/datefile
+xxx/project1/docs% more datefile
```

Clearly the second file was made after the first, and the third is equivalent to the second.

So, when you hand Unix a straightforward filename

- it sticks the path from the root to the c.w.d. in front
- makes an absolute filename with respect to the root.
Moving up as well as down

The above makes referencing files in the CWD and below (ie further from the root) very easy. Eg, sitting in the home directory, I view the datefile in docs using

```
b42.ecs:/ugrads/u98xxx% more project1/docs/datefile
Mon Oct 12 15:20:27 BST 1998
```

However, now we need a way to deal with directories above the current one.

Unix provides two shorthands which help in this:

1. a shorthand for your home directory, ~u98xxx or just ~
2. the ../ prefix refers to the parent of the CWD

Eg, suppose the current working directory is
```
/ugrads/u98xxx/project1/docs
```
Typing

```
b42.ecs:/ugrads/u98xxx/project1/docs% more datefile
```

gets us the second datefile we made.

But to get the first we could do any of the following

```
+/u98xxx/project1/docs% more ~/u98xxx/datefile
Mon Oct 12 15:20:05 BST 1998
```

```
+/u98xxx/project1/docs% more ~/datefile
Mon Oct 12 15:20:05 BST 1998
```

```
+/u98xxx/project1/docs% more ../../datefile
Mon Oct 12 15:20:05 BST 1998
```
Changing directories upwards

These shorthands also provide a way of changing directory upwards.

b42.ecs:/ugrads/u98xxx/project1/docs% cd ..
b42.ecs:/ugrads/u98xxx/project1% cd ..
b42.ecs:/ugrads/u98xxx% cd ..
b42.ecs:/ugrads% cd ..
b42.ecs:% cd ~
b42.ecs:/ugrads/u98xxx%

where at the penultimate step we move up above our home directory.

The ability to visualize your file store

- as a "tree structure"
- in which you can move around by cd'ing
- in which you can move around files

is the most important step to becoming a competent user.
6 Using redirection

Many Unix commands (and programs you write) will accept input from the "standard input" and write output on the "standard output". Usually these will be directed from the keyboard and screen respectively. There are simple mechanisms to output and input to and from other files and programs.

We have already seen the use of redirection to out the standard output of a program into a file. Eg

```
% cal > month
```

This starts a new file month The double redirect appends to a file:

```
% cal 11 1998 >> month
```

would stick November at the bottom of October, rather than starting a new file.

We can "pipe" the output of one program into the input of another. Eg

```
% cal | grep 0 > lines0
```

would send the output of cal into the input of grep which outputs only those lines of input containing a particular string — here a zero 0. This is put into the file lines0, which will as a result contain

```
5 6 7 8 9 10 11
19 20 21 22 23 24 25
26 27 28 29 30 31
```

What would % cal | grep 0 | grep 4 > lines04 put into file lines04?
7 Electronic Mail

- Once you have a login name at the Department, you effectively have another email address of the form u98xyz@ecs.ox.ac.uk

- To avoid having to log-on to separate mailers:
  - Decide which is going to be the main one, and set the others to forward mail appropriately.
  - If you decide on u98xxx@ecs, then use the **Forward** facility in the OUCS Herald WWW interface, and supply
    u98xxx@ecs.ox.ac.uk
    as the forwarding address.
  - If you decide on you@college, then edit a file called `.forward` in your home directory (i.e. `~u98xxx/.forward`) to contain your college address eg jo.bloggs@st-trinians.ox.ac.uk

- Mailers complete the domain
  - To mail fellow ecs’ers mail u98abc
  - To mail eng’ers mail david.murray@eng. (Your tutors are eng’ers)
  - To mail college bods mail jo.bloggs@st-trinians
  - To mail other unis
    mail anne.other@mediastudies.poppleton.ac.uk

- Using mail under openwin
  
  Use **mailtool** from the menu on the screen background
8 Protecting your files

Earlier we saw an example of the output from the `ls -l` command.

```
total 4
drwx--x--x 2 u98xxx 512 Oct 19 17:32 lsm2
-rw------- 1 u98xxx 29 Oct 19 17:32 datefile
drwx--x--x 4 u98xxx 512 Oct 19 17:33 project1
drwx--x--x 2 u98xxx 512 Oct 19 17:32 project2
```

What does this mean?

```
0123456789
 rwxrwxrwx  <--- Read Write eXecute
 uuuggggooo  <--- User Group Others
```

The zeroth column indicates whether the file is a directory (d), a standard file (-) or something more esoteric. Columns 1-9 indicate the read, write, and execute permissions for the files and directories. The "user" is you, ie u98xxx, the "group" includes all u98’ers, and "others" embraces all other people.

Permissions can be changed using `chmod`, as explained more fully in the manual pages. For example, to give others permission to read file datefile, and remove execute permission from others and group on project1:

```
b42.ecs:/ugrads/u98xxx% chmod o+r datefile
b42.ecs:/ugrads/u98xxx% chmod og-x project1
```

The rules for permissions are a bit tricky. For example, for an “other” user to be able to read a file belonging to you, all directories in the path must be readable. Another rule is that for an “other” user to cd into a directory, it and all directories above it must be executable.
9 Logging out

- Before leaving a session you must log out.

- However, you cannot log out directly from Openwindows. First you must exit from windows.

1. From the desktop select the Workspace Menu using Menu (RIGHT) button, then pull down to “Exit”.
   Openwindows will shut down, leaving you back with the plain console.

2. Then type `logout` or, for those in a hurry, press `ctrl-d` (hold down the control key and d key together).

- It is very important not to forget to logout

  - failing to do so leaves your session open to others
  - your password and filestore become open to abuse.