

UNIX Workshop 2010

<http://uws.assembla.me>

Angad Singh Aldrian Obaja Dang Cao Khoa
Chong Perry Saiaswin Saikannan Jialong Liu
Vu An Hoa Gary Kwong Melvin Zhang

August 4, 2010



<http://www.youtube.com/watch?v=dFU1AQZB9Ng>

Jurassic Park (1993)

“It’s a UNIX system! I know this.”

– Alexis “Lex” Murphy, Jurassic Park (1993)

Modern UNIX



Figure: Modern UNIX-like operating systems (Linux, BSD, Solaris)

What do these sites have in common?

The Google logo, featuring the word "Google" in its signature multi-colored font (blue, red, yellow, blue, green, red) with a trademark symbol.The Facebook logo, consisting of the word "facebook" in white lowercase letters on a solid blue rectangular background.The YouTube logo, featuring the word "You" in black and "Tube" in white inside a red rounded rectangle with a slight 3D effect.The Vimeo logo, featuring the word "vimeo" in a black, lowercase, sans-serif font.The FeedBurner logo, featuring a stylized flame icon in red and yellow to the left of the word "FeedBurner" in blue, with a trademark symbol.The Meemo logo, featuring the word "meebo" in a blue, lowercase, sans-serif font, with the final "o" being orange and having two small dots below it.

What do these sites have in common?

The Google logo, featuring the word "Google" in its signature multi-colored font.The Facebook logo, consisting of the word "facebook" in white lowercase letters on a blue rectangular background.The YouTube logo, with the word "You" in black and "Tube" in white inside a red rounded rectangle.The Vimeo logo, featuring the word "vimeo" in a black, lowercase, sans-serif font.The FeedBurner logo, which includes a stylized flame icon in blue and orange, followed by the word "FeedBurner" in blue.The Meelo logo, with the word "meelo" in a blue, lowercase, sans-serif font, where the final "o" is stylized with two dots below it.

67% of all web servers are running on UNIX¹

¹<http://w3techs.com>, August 2010

C was invented to write UNIX



PRENTICE HALL SOFTWARE SERIES

You will be programming in UNIX



- ▶ CS1010 labs - developing C programs in UNIX.
- ▶ CS1020 labs - developing Java programs in UNIX

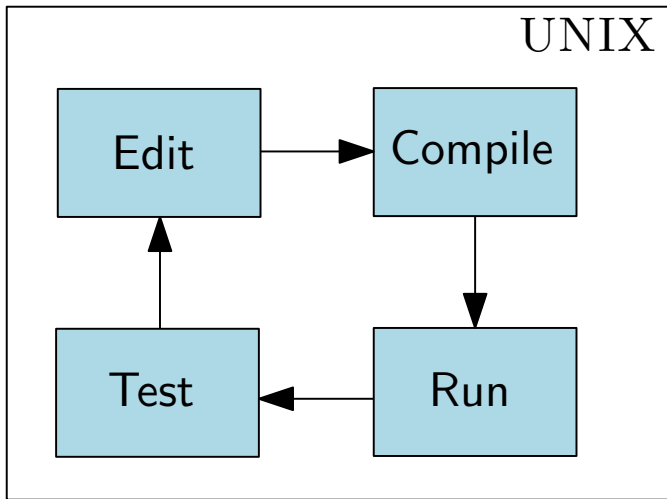


Figure: Workflow for writing programs

Activity: Login to NUSNET

1. Press Ctrl-Alt-Delete.



2. Type in your NUSNET user name, password and select the NUSSTU domain.



3. Click on the Ok button.

Activity: Creating your UNIX account

<https://mysoc.nus.edu.sg/~newacct>

Login using your NUSNET user name and password.

sunfire server in the old Machine Room



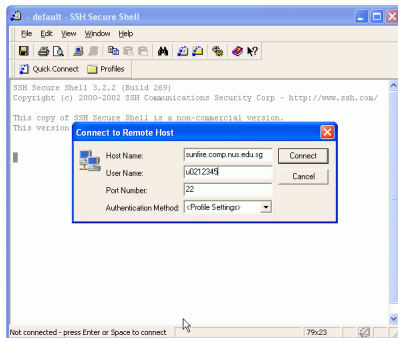
Figure: sunfire server located in the Machine Room with our Central Facilities staff. Clockwise from top-left: Tan Chee Sin, Tan Kwang Pon, Budiman Tsjin and Lai Zit Seng (Systems programmer, ITU).

Activity: Connecting to sunfire

1. From the desktop, launch the SSH Secure Shell Client application.
2. Click on Quick Connect

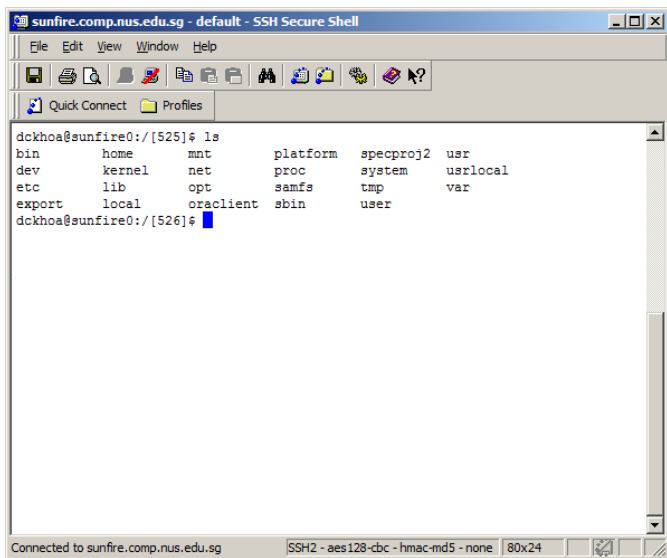
Host Name: sunfire.comp.nus.edu.sg

User Name: your UNIX user name



3. Click on Connect.
4. Click on "Yes" at the Host identification dialog.
5. Enter your UNIX password in the password dialog.

Command line interface



The image shows a terminal window titled "sunfire.comp.nus.edu.sg - default - SSH Secure Shell". The window has a menu bar with "File", "Edit", "View", "Window", and "Help". Below the menu bar is a toolbar with various icons for file operations. The main area of the window displays a command prompt "dckhoa@sunfire0:/[525]\$ ls" followed by the output of the "ls" command, which lists the contents of the current directory in a multi-column format:

bin	home	mnt	platform	specproj2	usr
dev	kernel	net	proc	system	usrlocal
etc	lib	opt	samfs	tmp	var
export	local	oraclient	sbin	user	

The prompt then changes to "dckhoa@sunfire0:/[526]\$". At the bottom of the window, a status bar shows "Connected to sunfire.comp.nus.edu.sg", "SSH2 - aes128-cbc - hmac-md5 - none", and "80x24".

Figure: Command line interface on sunfire

UNIX Directory Tree

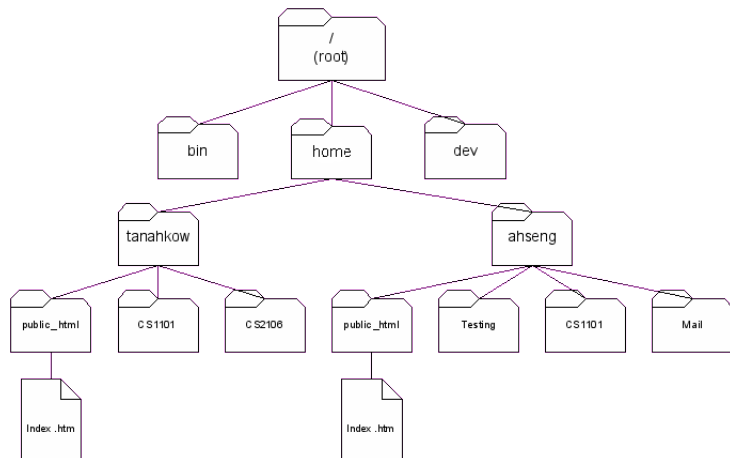


Figure: A subset of the UNIX directory tree showing home directories

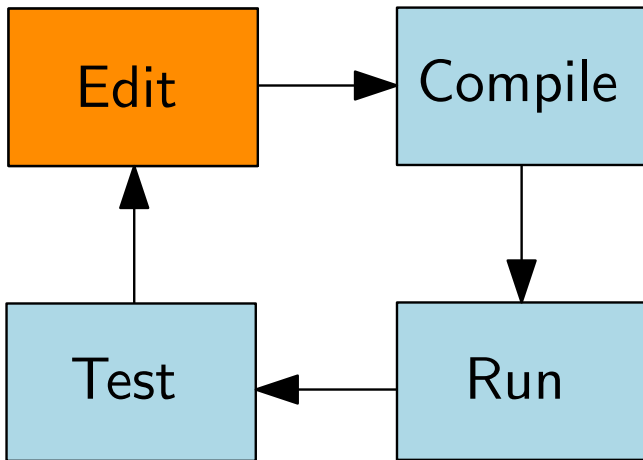
Parts of a command

```
$ program argument1 argument2 ...
```


Activity: Working with files and directories

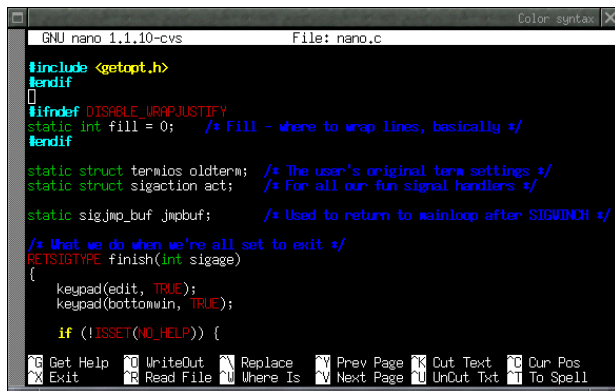
1. After login, you are placed in your home directory, e.g.
`/home/m/melvin`
2. You can check your working directory using the `pwd` command
`pwd`
3. The `ls` command shows you the files in your working directory
`ls`
4. Now create a new directory called `UNIXWorkshopFiles`
`mkdir UNIXWorkshopFiles`
5. Switch to the new folder using the `cd` command
`cd UNIXWorkshopFiles`
6. Use the `pwd` again command to check your working directory
`pwd`

UNIX



Text files are ubiquitous on UNIX

Program source code are stored as text files. A good text editor can dramatically improve your productivity.



```
GNU nano 1.1.10-cvs      File: nano.c

#include <getopt.h>
#endif
[]
#ifdef DISABLE_WRAPJUSTIFY
static int fill = 0; /* Fill - where to wrap lines, basically */
#endif

static struct termios oldterm; /* The user's original term settings */
static struct sigaction act; /* For all our fun signal handlers */

static sigjmp_buf jmpbuf; /* Used to return to mainloop after SIGWINCH */

/* What we do when we're all set to exit */
RETSIGTYPE finish(int sigage)
{
    keypad(edit, TRUE);
    keypad(bottomwin, TRUE);

    if (!ISSET(NO_HELP)) {
        Get Help      WriteOut  Replace  Prev Page  Cut Text  Cur Pos
        Exit          Read File  Where Is Next Page  UnCut Txt  To Spell
    }
}
```

Figure: Screenshot of nano

Activity: Text editing with nano

1. Download the sample GCD.c program from the UWS website using wget

```
wget http://uws.assembla.me/GCD.c
```

2. Edit the file using name nano

```
nano GCD.c
```

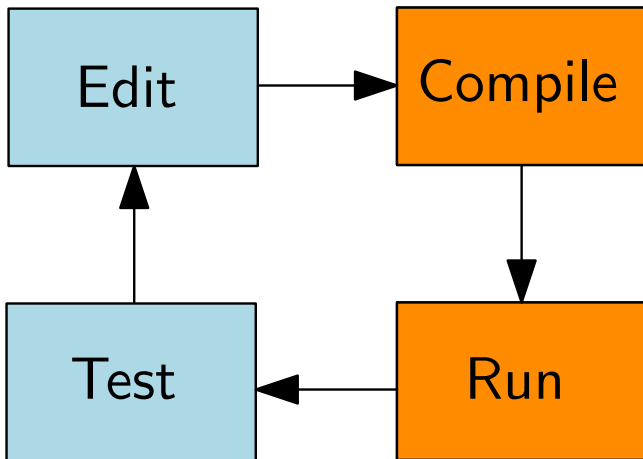
3. Type in your name and matric number as indicated in the file.
4. Save the file and exit nano by pressing

```
Ctrl-x
```

5. Check the contents of the file using the cat command

```
cat GCD.c
```

UNIX



Activity: Compiling and running

1. C programs are compiled using the gcc compiler.

```
gcc GCD.c
```

2. To run a program, you must add ./ in front of its name. The default name used by gcc is a.out

3. Run the GCD program

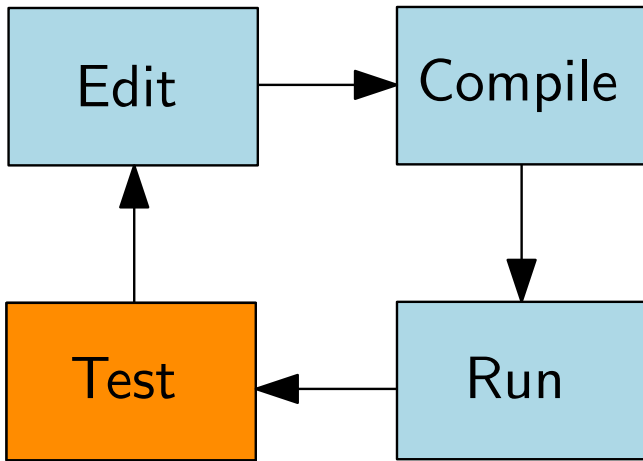
```
./a.out
```

4. Type in a pair of integers followed by the Enter key, for example

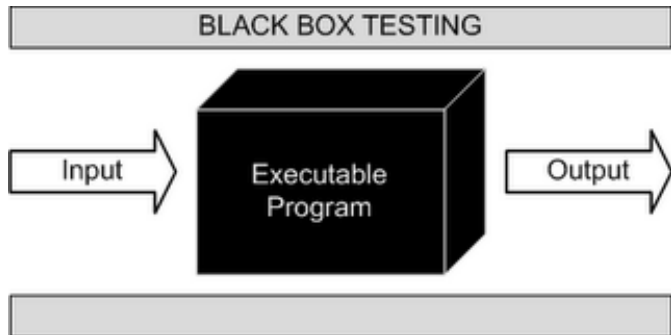
```
58 24
```

5. Repeat step 4 as many times as you like.
6. To force the program to end, press Ctrl-c

UNIX



Testing



Testing on UNIX

```
$ program < input > output
```

Activity: Creating the input

1. Instead of typing the input by hand as in the previous activity, we create an input file using nano.

```
nano input
```

2. Type in pairs of integers, one pair per line, for example

```
3 10
```

```
15 25
```

```
200 420
```

3. Save and exit nano.

Activity: Creating the correct output

1. Create an file for the correct output
`nano answer`
2. Type in the correct GCD for each pair of integers in the input
`GCD(3,10)=1`
`GCD(15,25)=5`
`GCD(200,420)=20`
3. Save and exit nano.
4. Run the GCD program on the test case.
`./a.out < input > output`

When things go wrong

9/9

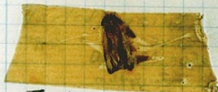
0800 Antam started
 1000 " stopped - antam ✓
 1300 (032) MP-MC ~~1.98264000~~ { 1.2700 9.037847025
 (033) PRO 2 2.130476415 ~~(-2)~~ 9.037846995 correct
 correct 2.130476415
 correct 2.130676415

Relays 6-2 in 033 failed special speed test
 in relay " 10,000 test.

Relay
 3145
 Relay 3376

1100 Started Cosine Tape (Sine check)
 1525 Started Mult+Adder Test.

1545



Relay #70 Panel F
 (moth) in relay.

1630 First actual case of bug being found.
 Antam started.
 1700 closed down.

Activity: Find the bugs!

1. Open output in nano and verify if it matches the correct answer.

```
nano output
```

2. If they differ, you've found the bug!, else try again with different input/answer files
3. Hint: manually checking whether two files are identical is boring, try using the diff command

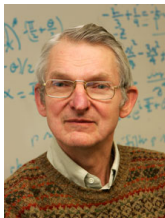
```
diff answer output
```

The UNIX Philosophy

Write programs that do one thing and do it well.

Write programs to work together.

Write programs to handle text streams, because that is a universal interface.



– Douglas McIlroy
(inventor of UNIX pipes)

Activity: SMS Word Count



Your friend from FASS is studying SMS language as part of a course project. She collected a number of SMS messages and would like to find out the frequency of each word.

Activity: SMS Word Count

For example, given the following text file:

```
U wan 2 haf lunch i'm in da
canteen now.
Haf u found him? I feel so
stupid da v cam was working.
Where r we meeting?
I went to ur hon lab but no
one is there.
```

The desired output is:

```
.
.
.
1 we
1 went
1 Where
1 working.
2 da
2 I
```


Activity: sort and uniq

Two UNIX utility programs are related to our task.

sort

Input:		Output:
dog		bat
bat	→	cat
log		dog
cat		log

uniq

Input:		Output:
dog		dog
dog	→	cat
cat		dog
cat		cat
dog		
cat		
cat		

Activity: SMS Word Count I

1. Download the file containing sms messages from <http://uws.assembla.me/SMSwords.txt> using `wget`
`wget http://uws.assembla.me/SMSwords.txt`
2. Sort the file.
`sort SMSwords.txt`
3. Sort and remove duplicates.
`sort SMSwords.txt | uniq`

Activity: SMS Word Count II

4. We need to use a particular option of `uniq` which counts the number of duplicates, read the manual page for `uniq`. Press `q` to leave the manual page.

```
man uniq
```

5. Sort and count words,

```
sort SMSwords.txt | uniq -???
```

6. Sort by the frequency, so that more frequent words appear later,

```
sort SMSwords.txt | uniq -??? | sort -n
```

Activity: Logging out of sunfire

To log out of sunfire, use the logout command,
logout

Useful programs/websites

- ▶ KiTTY, SSH client for Windows
<http://www.9bis.net/kitty/>
- ▶ Cygwin, UNIX-like environment for Windows
<http://www.cygwin.com/>
- ▶ Description of computing facilities in SoC
<http://docs.comp.nus.edu.sg/cf>
- ▶ mySoC, web service portal
<https://mysoc.nus.edu.sg>