

# UNIX Workshop

<http://www.comp.nus.edu.sg/~melvin/UWS/>

Chua Wei Hang, Lim Jing Quan, Max Tan, Melvin Zhang

July 30, 2010

“It’s a UNIX system! I know this.”

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

## 1 Introduction to UNIX

The materials for this workshop are adapted from the following sources:

- UNIX Workshop 2005 notes by Mark Tan (SoC, NUS)
- CS1101 Lab 0 notes by Aaron Tan (SoC, NUS)
- UNIX/Linux Tutorial for Beginners by Michael Stonebank (University of Surrey)

As an SoC student, you have access to IT facilities exclusive to SoC. Most of these facilities are accessed via our **sunfire** server which runs the Solaris Operating System, a UNIX-like operating system.

UNIX is the name of the operating system developed by a group of AT&T researchers at Bell Labs in 1969. Solaris and many other modern operating system, such as Linux, Mac OS X, and BSD, are descendants of UNIX. All of these operating systems provide a UNIX-like environment to the user.

The UNIX environment provided by the Solaris OS on our servers are used for:

- writing programs for your programming labs/assignments
- learning about operating system concepts (CS2106, Operating Systems)
- hosting a database driven site (CS2102, Database Systems)
- accessing SoC printers and checking your print quota
- reading your SoC email account

An OS is the software that is in charge of scheduling resources and processes of a system as well as managing and maintaining interactivity between the software and hardware of the system. Figure 1 shows that the operating system acts as the bridge between your applications and the computer hardware.

### 1.1 Origins of UNIX

The first version of UNIX was developed by Ken Thompson in August of 1969, supposedly in just three week. In 1971, Dennis Ritchie invented the C programming language and UNIX was rewritten using C and became the first operating system to be written in a high level language. Most operating systems of the time, including the original UNIX, was written in assembly language, which was a special language that is

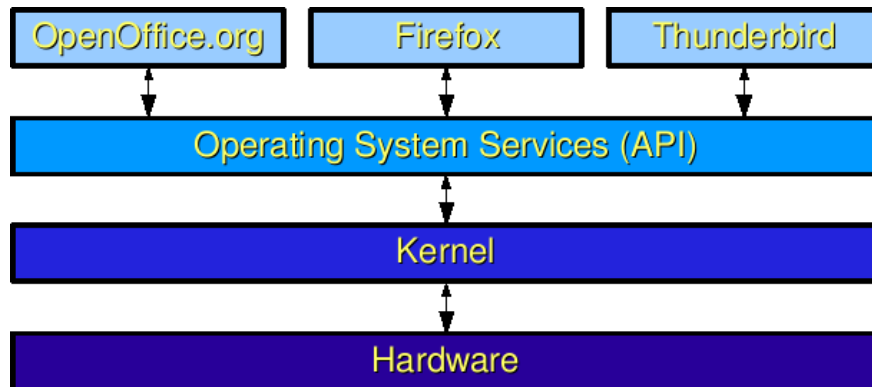


Figure 1: Relation between applications, OS and hardware



Figure 2: Ken Thompson (left) and Dennis Ritchie receiving the National Medal of Technology from President Clinton.

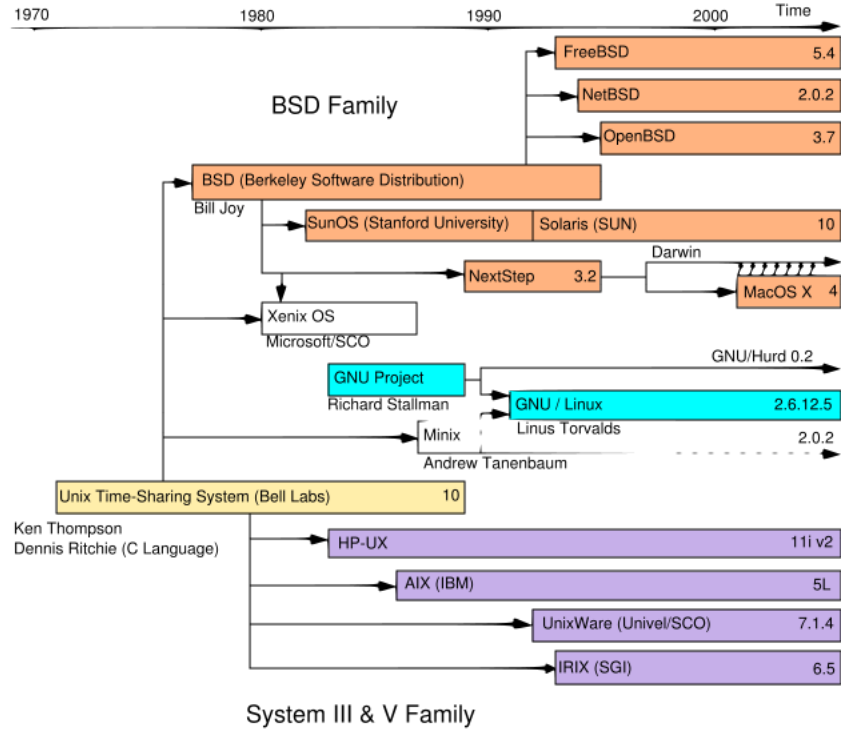


Figure 3: UNIX and its descendants

unique to a particular type of hardware. After UNIX was rewritten in C, it was possible to run UNIX on many different types of hardware.

For the development and implementation of UNIX and their contributions to operating systems theory, Dennis and Ken were jointly awarded the A. M. Turing award in 1983 and the National Medal of Technology (Figure 2) in 1999. The A. M. Turing award is the equivalent of the Nobel prize in Computer Science.

Today, many of the operating system, such as the Mac OS X from Apple, Solaris from Sun and Linux, are descendants of the original UNIX system from Bell Labs (see Figure 3). The operating system you will be using on our UNIX server in SoC is the Solaris OS.

## 1.2 UNIX Concepts

UNIX consists of the following components:

**Kernel** The kernel of UNIX is the core of the operating system. It allocates time and memory to programs and handles the file system and communications in response to system calls.

**Shell** The shell acts as an interface between the user and the kernel. When a user logs in, the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter. It interprets the commands the user types in and arranges for them to be carried out. The commands are themselves programs: when they terminate, the shell gives the user another prompt.

**Files and processes** Everything in UNIX is either a file or a process. A process is an executing program identified by a unique process identifier. A file is a collection of data. They are created by users using text editors, running compilers etc.

**Utilities** An operating system is useless without useful applications that help the you to accomplish everyday tasks such as editing text, browsing the web etc. A UNIX-like environment provides a standard set of useful utilities.

## 2 UNIX in SoC

Every NUS student is provided with a NUSNET account which gives you access to university wide online services such as the Integrated Virtual Learning Environment (IVLE) and an email account with the nus.edu.sg domain.

In order to login to the PCs in the computer labs, you will need to use your NUSNET account.

### Activity: Login to NUSNET

1. Press **Ctrl-Alt-Delete**.
2. Type in your NUSNET username, password and select NUSSTU domain.
3. Click on the **Ok** button.

In addition, every SoC student is provided with a UNIX account which gives you access to SoC's **sunfire** server which run on Solaris. You are allowed to choose your own username, for your UNIX account.

In the following activity, you will create your UNIX account.

### Activity: Creating your UNIX account

1. Login to <https://mysoc.nus.edu.sg/~newacct> using your NUSNET username and password.
2. Read through the user-agreement and make sure you understand the obligations.
3. Decide your UNIX username. Your username should be between 5-8 characters and must be formed from your name. You may also use your NUSNET username.
4. Type in your new password (twice).
5. Submit your application.

Your new UNIX account comes with the following privileges:

- Email : `unix_username@comp.nus.edu.sg`
- Website : [http://www.comp.nus.edu.sg/~unix\\_username](http://www.comp.nus.edu.sg/~unix_username)
- Solaris zone: `unix_username-z.comp.nus.edu.sg`
- Disk quota: 2Gb
- Print quota: 50 pages/month

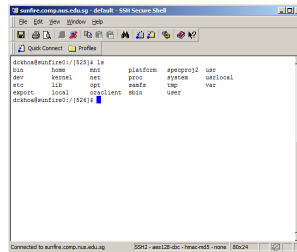
### 2.1 Checking your SoC email account

You can access your UNIX email account via mySoC Webmail, <http://mysoc.nus.edu.sg/~webmail>

Your mailbox part of your disk usage, which is 2Gb. You can forward your NUSNET email to your UNIX email using <https://exchange.nus.edu.sg/autoforward>.



Figure 4: **sunfire** server located in the Machine Room with our Networks staff. Clockwise from top-left: Tan Chee Sin, Tan Kwang Pon, Budiman Tsjin (has since left SOC) and Lai Zit Seng.



## 2.2 SoC's sunfire server

In the next activity, we will connect to the **sunfire** server (see Figure 4) and access the UNIX system via the Secure Shell protocol (ssh).

### 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 username
3. Click on **Connect**. Enter your UNIX password in the password dialog.

Once you have logged in, a copy of the shell is started and you are automatically placed in your home directory.

The shell is based on a command line interface (CLI), this is in contrast to the more commonly seen graphical user interface (GUI) which you are familiar with. Both types of interfaces are useful, depending on the kind of activity.

### Command line interface

Launching program by typing name follow by parameters. Programs as functions,  $f(x)$

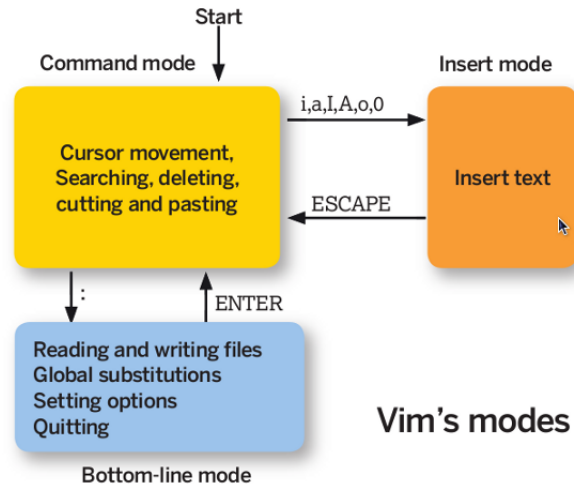


Figure 5: Different modes of Vim and how to switch between them

### Command line interface

- Silence is golden: there is usually no output when a program runs successfully
- Easy to automate repetitive tasks
- Default interface when accessing remote servers

## 2.3 Creating text files

Types of files you will encounter on UNIX include the following:

- a document (report, essay etc.)
- the text of a program written in some high-level programming language
- instructions comprehensible directly to the machine and incomprehensible to a casual user, for example, a collection of binary digits (an executable or binary file)
- a directory, containing information about its contents, which may be a mixture of other directories (subdirectories) and ordinary files.

Text files are used extensively on a UNIX system for storing system data, program configuration files, scripts and source code. They are preferable to binary files because they can be easily read/modified. The most important program is a text editor, a program to interactively create/edit text files. Examples of editors on UNIX are Vim, Emacs, nano, and pico.

In this workshop, we will introduce you to the Vim editor. Vim stands for Vi IMproved, it is created by Bram Moolenaar (now working at Google) as an improvement of an earlier editor called vi by Bill Joy (co-founder of Sun Microsystems). Vim is a modal editor. This means, depending on the current mode, different keys have different effects (see Figure 5). For example in Normal mode, pressing 'x' key deletes the character under the cursor, while in Insert mode, it adds the character 'x' to the text file. This allows Vim to use almost all the keys on your keyboard for issuing commands.

In the following activity, we will use Vim to create a simple text file.

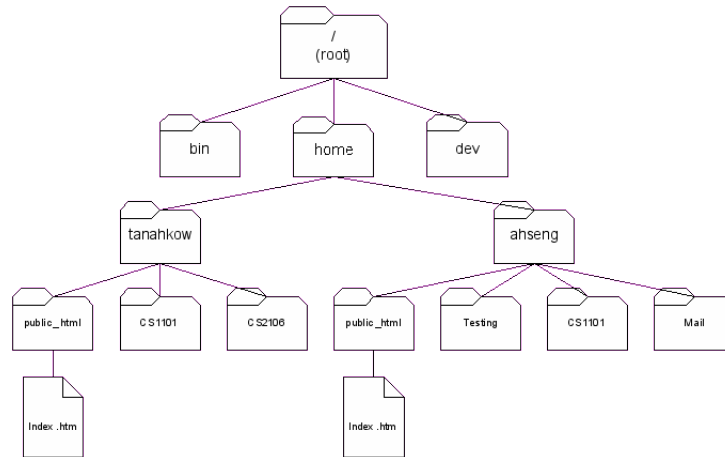


Figure 6: A subset of the UNIX directory tree showing home directories

### Activity: Text editing with Vim

1. From the Secure Shell Client window start Vim and create a new file using the command

```
vim hello.txt
```

2. Vim puts you in Normal mode by default. Switch to Insert mode using the ‘i’ key.

```
i
```

3. Type a short message to introduce yourself.
4. Now return to Normal mode by pressing the Escape key.

```
<Esc>
```

5. Save the file and exit Vim by pressing ‘ZZ’

```
ZZ
```

## 2.4 Organising your home directory

Files on UNIX are arranged in a hierarchical structure, like an inverted tree (see Figure 6). The top of the hierarchy is traditionally called root (written as a slash /)

Directories allow you to organising your files on UNIX. Certain directories have special meaning in UNIX. For example, the contents of the directory named `public.html` in your home directory is interpreted by the web server as your homepage at [http://www.comp.nus.edu.sg/~unix\\_username](http://www.comp.nus.edu.sg/~unix_username)

Sometimes you might want to share a file with your friends but it is too large to send through email. You can make use of the 2Gb in your UNIX account to host the file on your website.

### Activity: playing with diff and grep I

1. What is `diff`? – compare differences between files
2. Text editing usually leaves a lot of backup files ending with `~`. One day you want to figure out the differences between a file `text` and its backup `text~...` Open in two editors and then eye-ball?

```
diff firstFile secondFile
```

### 3. A quick how-to

#### Activity: playing with diff and grep II

Let's see how `text` and `text~` look like first

Output of `cat text`

```
same text
same text
Hello World!
still the same
still the same
```

Output of `cat text~`

```
same text
same text
Hello World~
still the same
still the same
```

#### Activity: playing with diff and grep III

Output of `diff text text~` :

```
3c3
< Hello World!
---
> Hello World~
```

#### Activity: playing with diff and grep IV

1. What is `grep`? \*- look for a pattern in file(s)

```
grep pattern file
```

2. Sometimes it is useful to find the occurrences of some word in a (list of) file. Say you suspect a typo in your source code, open a text editor and 'Find'?
3. But what if you made the same typo in a lot of files? 'grep' makes your life easier Let's find out how to 'grep'

#### Activity: playing with diff and grep V

Sample output of `grep h1 a.html`:

```
grep h1 a.html
```

```
<h1>Hello World!</h1>
```

Contents of `a.html`

```
<html>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```



A bit too easy, isn't it? Ready to get nasty?

1. Output of `grep h1 a.html*`

```
a.html:<h1>Hello World!</h1>
a.html~:<h1>Hello World~</h1>
```

2. Output of `grep -n h1 a.html*`

```
a.html:3:<h1>Hello World!</h1>
a.html~:3:<h1>Hello World~</h1>
```

3. Output of `grep -n -i 'heLlO wORlD' a.html*`

```
a.html:3:<h1>Hello World!</h1>
a.html~:3:<h1>Hello World~</h1>
```

4. Find out more in '`man grep`' !

5. `grep` on Linux is more fun! :p

## 2.5 Some useful applications on UNIX

A major part of the operating system that the user interacts with are the utilities that helps the user to perform everyday tasks such as editing files, printing and reading email.

You have already used a number of such utilities such as `mkdir` (make directory), `mv` (move), `chmod` (change modifiers/permissions) and `cd` (change directory).

In the next activity, you will learn a number of utilities related to printing. There are a number of printers that are exclusive to SoC students, they are located in COM1 level 1 and basement.

### Activity: Printing

- View the status of the print queue, use `lpq`,

```
lpq -P pstsc
```

- Remove a print job after it has been sent, use `lprm`,

```
lprm -P pstsc 89
```

- Check your print quota, use `pusage`,

```
pusage
```

UNIX utilities are simple focused programs operating using a common communication protocol. This is summed up by Douglas McIlroy (inventor of UNIX pipes) as the UNIX philosophy.

### 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

We will see the UNIX philosophy in action in our next activity, which makes use of three UNIX utility programs to analyse SMS messages.

### 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.

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

1. Download the file containing sms messages from <http://www.comp.nus.edu.sg/~melvin/UWS/SMSwords.txt> using wget

```
wget http://www.comp.nus.edu.sg/~melvin/UWS/SMSwords.txt
```

2. Sort the file.

```
sort SMSwords.txt
```

- Sort and remove duplicates.

```
sort SMSwords.txt | uniq
```

- 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
```

- Sort and count words,

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

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

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

If there is additional time during the workshop, you may wish to setup a homepage which is accessible at [http://www.comp.nus.edu.sg/~unix\\_username](http://www.comp.nus.edu.sg/~unix_username). The details are provided in the following activity.

### Activity: Setting up your homepage (Optional)

Instead of using `hello.txt`, create a file named `index.html` and put it in your `public.html` directory. Remember to change its permissions to readable by all.

```
<html>
<head>
<title>Sample index page</title>
</head>
<body>
Hello World
</body>
</html>
```

## 3 Summary

In this workshop, we have covered the following topics:

- UNIX from past to present
- Connecting to `sunfire` via `ssh`
- Text editing using `Vim`
- Using `sunfire` as a web host
- Manipulating text files using UNIX utilities

Finally, after we are done with what we need to do in the UNIX environment, we have to logout of the system.

### Activity: Logging out of `sunfire`

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

```
logout
```

## 4 Resources

### Computing Resources in SoC

- Description of facilities in SoC, <https://www.comp.nus.edu.sg/cf> and <https://mysoc.nus.edu.sg/~wiki>
- Web based services, mySoC, <https://mysoc.nus.edu.sg>
- SoC Webmail <https://mysoc.nus.edu.sg/~webmail>
- SSH Secure Shell Client 3.2.9, [http://www.comp.nus.edu.sg/~cs1101x/2\\_resources/SSHSecureShellClient-3.2.9.exe](http://www.comp.nus.edu.sg/~cs1101x/2_resources/SSHSecureShellClient-3.2.9.exe)