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 3, 2010

Jurassic Park (1993)

> *"It's a UNIX system! I know this."*
>
> – Alexis "Lex" Murphy, Jurassic Park (1993)

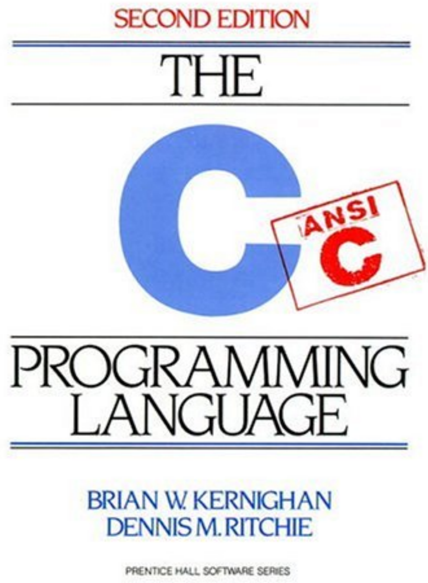
What do these sites have in common?

Google™
facebook
YouTube
vimeo
FeedBurner™
meebo

67% of all web servers are running on UNIX[1]

[1] http://w3techs.com, August 2010

C was invented to write UNIX

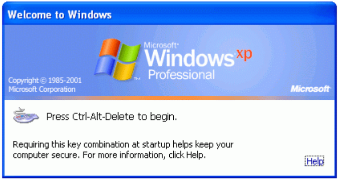


You will be programming in UNIX

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

Activity: Login to NUSNET

1. Press `Ctrl-Alt-Delete`.

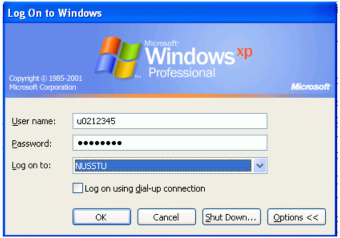


Welcome to Windows
Microsoft Windows xp Professional
Copyright © 1985-2001 Microsoft Corporation
Microsoft
Help

Press Ctrl-Alt-Delete to begin.

Requiring this key combination at startup helps keep your computer secure. For more information, click Help.

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



Log On to Windows
Microsoft Windows xp Professional
Copyright © 1985-2001 Microsoft Corporation
Microsoft
OK
Cancel
Shut Down...
Options <<

User name: u0212345

Password: ●●●●●●●●●

Log on to: NUSSTU

□ Log on using dial-up connection

3. Click on the `Ok` button.

Activity: Creating your UNIX account

`https://mysoc.nus.edu.sg/~newacct`

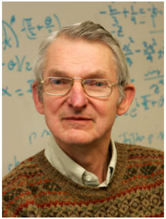
Login using your NUSNET user name and password.

Modern UNIX

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

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.



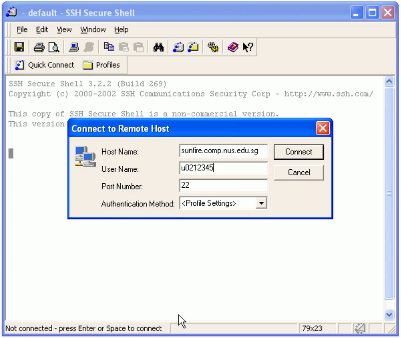
– Douglas McIlroy

sunfire server in the old Machine Room

Figure: 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.

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



```
SSH Secure Shell 3.2.2 (Build 269)
Copyright (c) 2000-2002 SSH Communications Security Corp - http://www.ssh.com/

This copy of SSH Secure Shell is a non-commercial version.
This version
```

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

User Name: u0212345

Port Number: 22

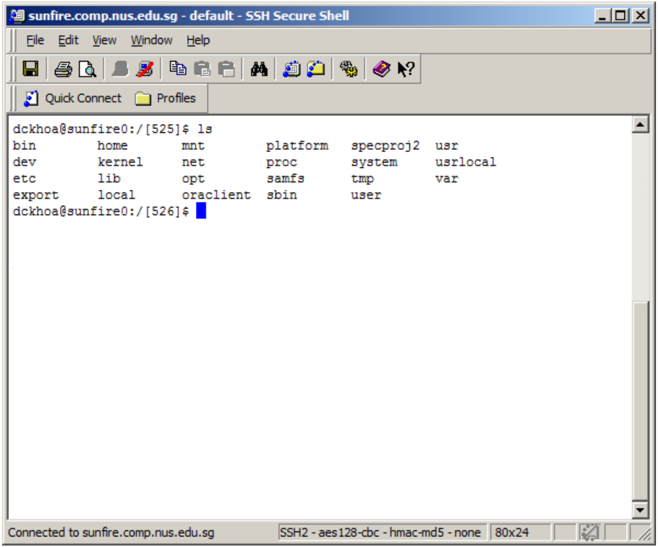
Authentication Method: <Profile Settings>

Not connected - press Enter or Space to connect

79x23

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

Command line interface



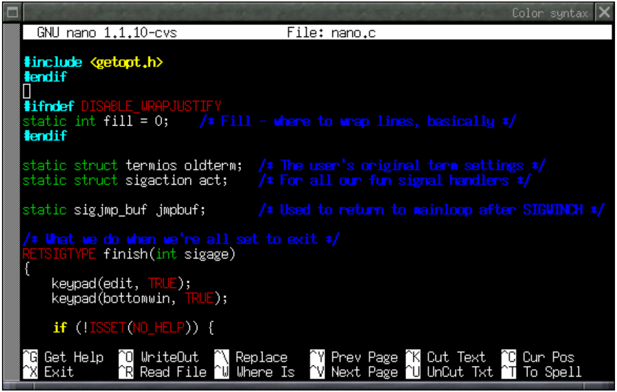
sunfire.comp.nus.edu.sg - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles

```
dckhoa@sunfire0:/[525]$ ls
bin         home        mnt         platform    specproj2  usr
dev         kernel      net         proc        system     usrlocal
etc         lib         opt         samfs       tmp        var
export      local       oraclient   sbin        user
dckhoa@sunfire0:/[526]$
```

Connected to sunfire.comp.nus.edu.sg | SSH2 - aes128-cbc - hmac-md5 - none | 80x24

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

#ifndef 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)) {
```

Figure: Screenshot of nano

Activity: Text editing with nano

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

   ```
   nano hello.txt
   ```

2. Type the following paragraph as carefully as possible.
3. Save the file and exit nano by pressing

   ```
   Ctrl-x
   ```

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 you 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?

Getting serious about grep

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

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	$\longrightarrow$	cat
log		dog
cat		log

`uniq`

Input:		Output:
dog		dog
dog	$\longrightarrow$	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
   ```

Summary

Activity: Logging out of `sunfire`

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

```
logout
```

Useful programs/websites

- KiTTY SSH client
  http://www.9bis.net/kitty/
- Computing facilities in SoC
  http://docs.comp.nus.edu.sg/cf
- mySoC (Web services)
  https://mysoc.nus.edu.sg