Introduction to UNIX

(Based on slides by Michael Siegenthaler)
Why Bother?

• Most programmers who learn UNIX end up finding it useful
• Provides powerful command-line interface
  – Many simple tasks are easier to accomplish
  – Possible to script repetitive operations
• Widely used in research and industry, and runs most of the servers on the Internet
UNIX Philosophy

• Multiuser / multitasking
• Toolbox approach
  – Combine multiple simple commands instead of using a single complex application
• Designed by programmers for programmers
Shelling into CSUG

• From Windows
  – PuTTY
  – Cygwin

• From MacOS, open a terminal and type
  – `ssh netid@csug01.csuglab.cornell.edu`
Transferring Files

- Use WinSCP
  [http://winscp.net/](http://winscp.net/)
Running Commands

• Commands follow the form:
  – `command <options> <arguments>`
  – Options modify the command
  – Arguments indicate what file to operate on

• Get help by typing `man command`

• Example:

  [msiegen@tiger ~]$ ls -l /usr
  total 301
  drwxr-xr-x  2 root root  69632 Oct 18 08:43 bin/
  drwxr-xr-x  2 root root  4096 Aug 12  2004 etc/
  drwxr-xr-x 117 root root 20480 Sep 12 20:40 include/
  ...
Plumbing

• I/O Redirection
  > Redirect standard output to file
  >> Append standard output to file
  < Get input from file

• Pipes (|) are used to take the output of one program and use it as input to another
  e.g. du -sk /home/* | sort -nr | head -10
      > disk_hogs.txt
Practical Tips

• Use **less** to view output that will not fit on your screen
  
  e.g. `ls -lR | less`

• Use **grep** to filter output, and **wc** to count lines
  
  e.g. `ps aux | grep "vim" | wc -l`

• Use **&&** to run multiple commands in sequence
  
  e.g. `./configure && make && make install`
File System

• Case sensitive!

• Moving around, working with directories
  
  cd
  Change working directory
  
  pwd
  Print working directory
  
  ls -la
  List all files in working directory
  
  mkdir
  Make directory
  
  rmdir
  Remove directory
  
  cp
  Copy file
  
  mv
  Move or rename file
  
  rm
  Delete a file

• Searching
  
  e.g. find -name Makefile
How to understand the File System

[da279@csug01 /]$ ls /
amd bin boot courses dev etc home initrd lib lib64 localdisk lost+found media misc mnt opt proc root sbin selinux srv sys tmp usr var
[da279@csug01 /]$ ls /courses
cs3110 cs3410 cs3410stf cs4411
[da279@csug01 /]$ ls /courses/cs3410
mipsel-linux README
[da279@csug01 /]$ ls /courses/cs3410/mipsel-linux/
bin include info lib libexec man mipsel-linux share

[da279@csug01 ~]$ ls /amd/daffodil/a/
ab397 al644 bc352 cek37 cs722 db493 dtt6 gey2 hs465 jjs87 js368 kk67 mjp63 ng292 pae26 rh335 rw347 sr533 tbw32 xl229...
[da279@csug01 ~]$ ls /amd/daffodil/a/da279/
for.c for.c~ for.s hello.c hello.s
Viewing File Contents

• **Use `cat` or `less`:**
  ```
  $ cat hw1.c  # use cat for short files
  #include "test-include.h"
  
  _start() {
  }
  
  $ less hw1.s  # use less for long files
  ```

• **You can also use vi or emacs!**
Comparing Files

• Use `diff`:

```bash
$ cat file1
Hello!
This is the contents of file1.
Goodbye.
$ cat file2
Hello!
This is the contents of file2.
Goodbye.
$ diff -u file1 file2
+++ file2  2007-10-11 04:25:45.000000000 -0400
@@ -1,3 +1,3 @@
Hello!
-This is the contents of file1.
+This is the contents of file2.
Goodbye.
```
How to use gcc

[da279@csug01 /tmp]$ ls
hello.c

[da279@csug01 /tmp]$ gcc -o hello hello.c

[da279@csug01 /tmp]$ ls
hello hello.c

[da279@csug01 /tmp]$ ./hello
Hello World!
How to use mipsel-linux-gcc

• To create the .o file:
  /courses/cs3410/mipsel-linux/bin/mipsel-linux-gcc -c foo.c
  - Object file is created and saved in foo.c

• To create the .s file:
  /courses/cs3410/mipsel-linux/bin/mipsel-linux-gcc -S foo.c
  - MIPS Assembly Code is created and saved in foo.s
  - You can actually run these instructions in your Processors!