Easy or Hard

- This course is either very easy or very hard. If you don’t know and use the tools available you will spend a lot of time spinning your wheels
- The following will make this course easier
  - Understand how to connect to Unix and work with the Common Desktop Environment (CDE)
  - Read and understand man pages and documentation
  - Know how to use relevant Unix utilities
  - Use a full featured editor for coding
  - Use an IDE for compilation and debugging
- The actual scope of what you need to know is greater than can be covered here, but this is a start

Overview

- A basic connections
- The Common Desktop Environment
- Unix help via man pages
- Selected Unix utilities
- Editors – nedit
- IDEs – workshop
Basic Connection

Making a connection (3 ways)

- Login to a Unix workstation running CDE (best)
  - You can use all the graphical tools available to work
- Telnet from a PC (worse)
  - Use command line Unix
  - Use Unix line editors to work on files or use PC tools to create or manipulate files and FTP the files back and forth
- Turn a PC into a X-terminal and telnet (ok)
  - This allows all the benefits of a workstation login, but it can be slow

A PC telnet session

- Run telnet
  - On the start menu of an MS windows machine, there is a menu item called “Run”
  - Select this item and type “telnet” in the text box
  - When the DOS window opens with the telnet prompt, type “open” and the name of the machine to connect to:
    open paradox.sis.pitt.edu
  - When the machine responds, login to the remote host
  - When done, “exit” from the remote machine, and close the DOS window
A Basic telnet window

More on the telnet window

A Couple Commands in Telnet
Common Desktop Environment

Using a Workstation

• A workstation provides the most natural access to Unix resources
  • Login at the login screen
  • Start at least one xterm session
  • Run an graphical applications you wish
  • Tailor your environment

Login
The Basics of the CDE

- Look at how to navigate the various desktops
- Learn how to add your own menu items
- Learn enough about X resources to tailor your environment

Main Screen

The Desktop Control Panel
The Menus

<table>
<thead>
<tr>
<th>Workspace Menu</th>
<th>Applications</th>
<th>Local Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>Application Manager</td>
<td>Adobe Stuff...</td>
</tr>
<tr>
<td>Local Applications</td>
<td>System Control</td>
<td>Emacs</td>
</tr>
<tr>
<td>Cards</td>
<td>Audio and Video</td>
<td>FrameMaker...</td>
</tr>
<tr>
<td>Help</td>
<td>Calculator</td>
<td>FixIt</td>
</tr>
<tr>
<td>Fonts</td>
<td>Calendar</td>
<td>Flypick...</td>
</tr>
<tr>
<td>Help</td>
<td>Icon Editor</td>
<td>G VueWriter...</td>
</tr>
<tr>
<td>Doors</td>
<td>Image Viewer</td>
<td>Mathematics...</td>
</tr>
<tr>
<td>Unix</td>
<td>Java</td>
<td>Mplate...</td>
</tr>
<tr>
<td>Mail</td>
<td>DW Clock</td>
<td>Notit...</td>
</tr>
<tr>
<td>Tools</td>
<td>Snapshot</td>
<td>Network...</td>
</tr>
<tr>
<td>Windows</td>
<td>Text Editor</td>
<td>Star Office...</td>
</tr>
<tr>
<td></td>
<td>Text Note</td>
<td>S/2 chanting...</td>
</tr>
<tr>
<td></td>
<td>Vtext Note</td>
<td>Xemacs...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Xo...</td>
</tr>
</tbody>
</table>

Online Documentation
http://docs.sun.com/app/docs/prod/prod

Drill down
Using cygwin-x

- Download the setup program by going to x.cygwin.com
- You might get the users guide as well
- Follow the setup instructions about where you wish to set it up
- Make a shortcut to the batch file to start x – e.g. cygwin-x\usr\X11R6\bin\startxwin.bat
- Make a ssh connection to the machine you wish to start a connection with.
  - Set xhost plus locally
  - Set display variable as needed
    - Use ipconfig /all in a CMD window to determine your ip
**Man Pages**

---

**man**

the Basic Unix Help Facility

- It provides formatted help pages on every utility, system call, and API function.
- For functions, the man page defines the arguments, purpose, and returns.
- The man on man that tells you how to search, move around, and otherwise use man pages.
- The man pages are in sections, so be careful, you may need to use the section switch to get the right section.
- There is a -k switch, which lists all relevant man pages.

---

**man on signal**

---

```
#include <signal.h>

int signal(int sig, void (*func)(int));
```
**xman**

a Graphical Interface to **man**

- Allows for easier browsing
- Index and pages open simultaneously
- Recognition versus recall
- Allows model of “sections”
- Allows easier manipulation
- Scrolling back and forth
- Does not allow within page search
More Things to Do

- Read a couple include files
  - stdio.h
  - socket.h
- Read a complete standard – maybe something simple like http or smtp
- Read the man pages for:
  - the shell you use
  - grep
  - cc

Unix Utilities

Selected Unix Utilities & Commands

- There are literally 100’s of Unix utilities and commands that are useful to know. Only a few are covered here:
  - File transfer – ftp
  - File searching -- grep
  - Process control -- ps, kill, &, bg, fg, and tee.
  - File viewing – more, tail
  - File location – which and knowledge of file structure i.e. usr, etc, spool, var, tmp
  - Shortcuts and accelerators – history, tab, and ln
ftp

• ftp stands for file transfer protocol
  • It was one of the first network programs in Unix
  • The ftp works through a client/server program set
  • The host system runs a server called ftpd
  • The client runs a client program normally called ftp
  • Some systems allow restricted "anonymous" ftp
• PC based interfaces tend to hide the underlying ftp commands, Unix ftp is normally command line
  • ftp does automatic conversion of line endings as needed
  • To turn this feature off, use binary mode

Common ftp Commands

• get will get a single file
• put will put a single file
• binary specifies that files are to me moved byte for byte
  • text mode does pc to unix conversion of line endings
• prompt turns off prompting for mget and mput
  • mget will get multiple files
  • mput will put multiple files
• hash prints a hash mark for each block moved
• "! Command" allows a command to be executed locally from within ftp
grep

- The General Regular Expression Program or grep is a program to search streams
  - allows the user to search a standard I/O stream for terms pr patterns
  - Allows the numbering or lines
  - Allows non matches to be displayed
  - Allows counts to be developed
  - Helps with voluminous data streams

Process Control Commands

- ps can be used to list processes
  - Ps –ef lists all processes with full info
- To run a process in the background, type the command followed by an &
  - xman& runs xman in the background returning the prompt to the user
- When you forget the ampersand, type ‘^Z’ which will stop the process
  - Typing bg will then move the process to the background
- If you stop a process (‘^Z’) but do not move it to the background, the application will not accept input
File Commands

• When you execute a command, you will occasionally find what results is not what you expected. The command “which” allows you to identify the executable you are running – e.g. which nedit
• Knowing the basic Unix file structure is also very helpful – see next two slides
• At the command line you can page through a file or look at the beginning or ending using the commands more, head, and tail

Hierarchical File System

• Must have a root directory – /
• A user’s home directory is ~username
Important Directories

• Some standard directories include
  • /bin – user binaries
  • /sbin – binaries for system administration
  • /dev & /etc – device files system admin data files
  • /home – user file systems
  • /spool & /tmp – temp files for printing and other things
  • /usr – binary files (unix system resources)
    • /usr/lib – libraries for the linker
    • /usr/include – most of the include files
    • /usr/man – the man pages main location

Getting around and issuing commands

• Most students use Unix crudely because they don’t know better. Consider the following four things that can be done.
  • Most shells, including the default bash shell, have history. Cursor up to see commands you have already issued – allowing you to easily reissue the command.
  • History lines can be edited and the modified line run.
  • If you type the first few letters of a directory, filename, or executable and hit the tab key, Unix will complete the word or show the options.
  • Unix allows links (ln) that can make moving to a given directory as easy as typing just a couple letters.

A Couple GUI utilities

• Netscape is the default browser on Unix. Hotjava also is installed on solaris
• Netscape can be used:
  • For viewing web sites (netscape)
  • Editing webpages (netscape -e)
  • Reading mail and keeping address books (netscape -m)
• The mail tools used on Unix are multiple and all are imap and pop compliant
  • pine exists as a command line mail tool
  • mailtool is the default solaris mailtool
  • netscape provides a powerful mail capability
Editors

There are a variety of editors
- vi and pico
- emacs and xemacs
- nedit

The line editors are vi (universal), pico (very simple), emacs (very powerful).
The graphical editors are nedit (powerful but simple) and xemacs (very powerful and not too hard to learn).

Nedit as an editor

- You must be in an X Window System environment to run nedit
- It is relatively simple to use and has a little built in help
- It is important to keep in mind:
  - Preferences are at two levels, immediate and saved
  - Really useful for programming – syntax, braces, include files, indents, etc.
Some Important nedit Features

- If you select a line number from a compiler error listing, you can select “go to line number” under search.
- If you select an include file line, selecting “go to selected” under file to open the file.
- If you place the cursor after a parenthesis or brace, the corresponding brace will be highlighted.
- The window can be split to see the beginning and end of a very long procedure.
- Line numbers can be added to the display.
- Syntax is graphically highlighted.
Integrated Development Environments – IDEs

- Writing, compiling and linking complex applications is difficult.
- Integrated development environments make it easier.
- Sun has two IDEs, which appear to be moving toward integration and which are constantly changing names. They are:
  - sunstudio (for C, C++, Fortran, etc.)
  - netbeans (for Java)
- This presentation briefly introduces workshop

Sunstudio Functionality

- Sunstudio provides four main functions from our point of view
  - The ability to construct a project which allows a set of files to be identified and manages the make process
  - An integrated editor
  - A debugger that includes graphical breakpoints and the ability to debug multiple processes
  - A facility for developing GUIs in a WYSIWYG fashion
Using sunstudio

- To start sunstudio, type `sunstudio` at the command prompt
- Once open, mount a file directory in the filesystems tab of the explorer window
- Use the File/New option to create simple C programs and makefiles
- Use the make file and do a build all
- Use the debugging option to trace through

Main sunstudio screen

Help Viewer