

CS 344 - Unix Operating System Fundamentals

Francisco Hernández
(hernandf@cis.uab.edu?subject=CS-344)

Office: CH138 & Undergrad Lab
Office Hours: Tue & Thu 4-6 PM

Other times only by appointment
<http://www.cis.uab.edu/cs344>

Based on slides created by
Dr. Bangalore for the
Spring 2005 offering of the
course

Objectives

1. Introduce UNIX OS principles.
2. Provide hands-on experience using UNIX through assignments.

Grading Policy

Homework	50%
Mid Term Exam	20%
Final Exam	30%

Department of Computer and Information Sciences, UNB

Tentative Schedule

Date	Topics	Comments
June 3	Introduction and Overview	
June 10	Chapter 2 and 3 - Touring the Features and Editor	HW-1
June 17	Chapter 4 and 5 - Using Basic UNIX Utilities and Scripts	HW-2 HW-1 Due
June 24	Chapter 7 and 9 - Accessing and Changing Previous Commands and Setting File and Directory Permissions	
July 1	Mid Term Exam	HW-3 HW-2 due
July 8	Chapter 8, 11, and 12 - Specifying Instructions to the Shell, Controlling User Processes and Managing, Printing, and Archiving Large Files	
July 15	Chapter 13 and 14 - Shell Programming	HW-4 HW-3 Due
July 22	Chapter 15 and 16 - grep & sed	
July 29	Miscellaneous Topics and Review	HW4 Due
Aug 5	Final Exam	

6/2/2005 4

Department of Computer and Information Sciences, UNB

Operating System

- Acts as an interface between the user and the computer hardware
- A program that controls the execution of other programs
- Responsible for managing different computer resources (CPU, memory, disk, display, keyboard, etc.)

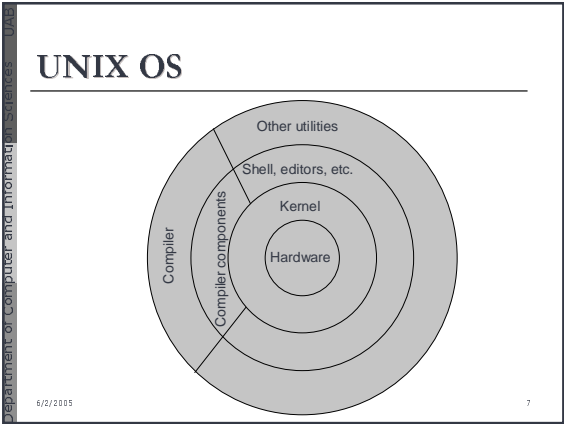
6/2/2005 5

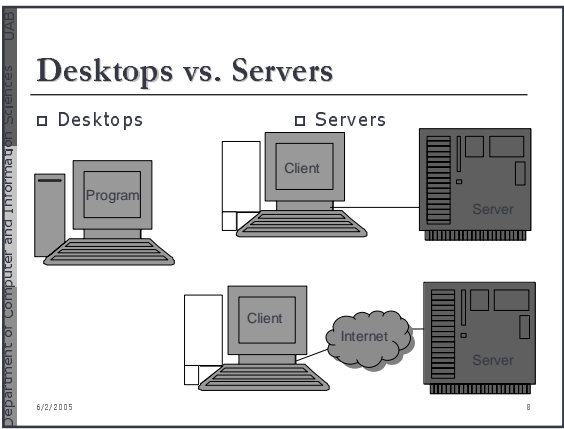
Department of Computer and Information Sciences, UNB

History

- Bell Laboratories in Murray-Hill New Jersey
- Developed by Ken Thompson in 1969
- Multi-user/Multitasking OS
- Many "flavors" (System VII, BSD, Solaris, Ultrix, Linux, ...)

6/2/2005 6





- Department of Computer and Information Sciences, UNB
- # Accessing UNIX systems
- ▣ Using client programs on Windows machines
 - PuTTY - <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
 - SSH Secure Shell Client - <http://www.icrew.org/mirror/ssh/SSHSecureShellClient-3.2.9.exe>
 - ▣ Other options:
 - Login to one of the SUN desktops in the lab
 - Install Linux on your computer (dual boot)
 - Use Knoppix Live CD (no need to install anything)
 - Use Cygwin/VM Ware (runs as a windows process)
- 6/2/2005 9

Enter password

6/2/2005 13

Now you are ready to enter commands at the UNIX Shell prompt

6/2/2005 14

Some sample commands

6/2/2005 15

Department of Computer and Information Sciences, UAB

Interacting with the Shell

- Type a command and press ENTER
- Shell starts a new process for executing the requested command (utility)
- The new process executes the command and the shell displays any output generated by the command
- When the process completes, the shell displays the prompt and is ready to take the next command
- Additional arguments after the command are used to pass specific information to the command
- The command "exit" or CTRL-D kills the shell

6/2/2005 16

Department of Computer and Information Sciences, UAB

Sample Commands

- Identifying current directory
 - pwd
- Listing files
 - ls
- Passing arguments
 - Long listing: ls -l
- Listing processes
 - ps
 - ps -u <userid>
 - ps -ef

6/2/2005 17

Department of Computer and Information Sciences, UAB

"man" Command

man provides on-line documentation of the UNIX system itself.

6/2/2005 18

Department of Computer and Information Sciences, UALB

Redirecting Output

- By default output from a utility is sent to the terminal/screen
 - `ls -l > myfiles`
- Output can be redirected to a file or another utility
- The ">" symbol tells the shell to open a new file and send the output to that file
 - `ls -l > myfiles`
- The "|" symbol tells the shell to connect the output of the first process to the input of the second process
 - `ls -l | wc -l`
 - Two processes are created for each of the commands and the output of the first process is sent to the input of the second process

6/2/2005 22

Department of Computer and Information Sciences, UALB

Redirecting Output

The diagram shows three scenarios of output redirection:

- Utility: ls, Args: -l** with **Command: ls -l**. The output (out) is directed to a **Screen**.
- Utility: ls, Args: -l** with **Command: ls -l > myfiles**. The output (out) is directed to a **File: myfiles**.
- Utility: ls, Args: -l** with **Utility: wc, Args: -l** and **Command: ls -l | wc -l**. The output (out) of the first utility is directed to the input (in) of the second utility, which then outputs (out) to a **Screen**.

6/2/2005 23

Department of Computer and Information Sciences, UALB

Other Directives

- Input redirection - "<"
 - e.g., `pr < myfiles`
- Append output to a file - ">>"
 - e.g., `cat newfile >> outfile`
- Execute multiple commands - ";"
 - e.g., `clear; date`

6/2/2005 24

Department of Computer and Information Sciences, UIUC

Miscellaneous

- How does the shell distinguish arguments from utilities?
 - ABC | DE | F > G
- Repeat commands (depends on the shell)
 - Repeat previous command - !!
 - Repeat a command starting with some string - !p (execute last command starting with p)
- History - using the "history" command
- Shortcuts - using the "alias" command
 - Type: alias h=history and press ENTER
 - When you type h, the shell will execute the corresponding utility

6/2/2005 25

Department of Computer and Information Sciences, UIUC

Directory Structure

- Similar to "Folders" in Windows
- Natural way to organize files into directories and sub directories
- "/" (forward slash) is used as a separator between directories and files
- "/" also denotes the root directory (highest directory in hierarchy)
- When we login we will be in the HOME directory (e.g., /home/hernadf)
- "pwd" command prints the current directory we are in
- "cd" command is used to change directories

6/2/2005 26

Department of Computer and Information Sciences, UIUC

Typical Directory Structure

```

graph TD
  root["/"] --- bin
  root --- dev
  root --- etc
  root --- home
  root --- lib
  root --- mail
  root --- sbin
  root --- tmp
  root --- usr
  root --- var
  home --- alice
  home --- barry
  home --- gary
  home --- jane
  home --- larry
  alice --- alice_myfiles[myfiles]
  alice --- alice_dat_in[dat.in]
  alice --- alice_dat_out[dat.out]
  larry --- larry_data[data]
  larry --- larry_misc[misc]
  larry --- larry_mydoc[mydoc]
  larry --- larry_myfiles[myfiles]
  larry_data --- larry_data_myfiles[myfiles]
  larry_data --- larry_data_dat_in[dat.in]
  larry_data --- larry_data_dat_out[dat.out]
  
```

6/2/2005 27

Department of Computer and Information Sciences, UAB

Working with Directories

- Change directory
 - `cd <directory-name>`
 - `cd`; `cd mydir`; `cd ./mydir`; `cd ..`; `cd ../misc`; `cd ~/mydir`;
- Create directory
 - `mkdir <options> <directory-name>`
 - `mkdir mydir`; `mkdir /home/puri/mydir`; `mkdir ../mydir`
- Remove directory
 - `rmdir <options> <directory-name>`
 - `rmdir mydir`; `rmdir /home/puri/mydir`; `rmdir ../mydir`
- Details about a command can be obtained using the "*man command*" – e.g., `man mkdir`

6/2/2005 28

Department of Computer and Information Sciences, UAB

Listing Directories and Files

- "*ls*" lists files and directories
- By default "*ls*" displays directories and files sorted in alphabetic order column-wise
- Other options can be used to display details in different formats
 - long listing: `ls -l`
 - long listing in reverse order: `ls -lr`
 - list sorted by time stamp: `ls -t`
 - see man page for more options

6/2/2005 29

Department of Computer and Information Sciences, UAB

View Files

- Using "*more*"
 - `more <filename>`
 - SPACEBAR to move forward
 - move forward and backwards – n and b
 - search for a string – `/<string>`
- Other options to view files – `cat`, `head`, `tail`
- Combining `cat` (`head`, `tail`) and `more`
 - `cat myfile | more`
 - `tail -200 myfile | more`
- Using editors (`vi`, `emacs`, etc.)

6/2/2005 30

Department of Computer and Information Sciences, UAB

“wc” command

- `wc <options> filename`
- Default behavior is to print number of lines, words, and characters followed by the filename
- Count lines – `wc -l <filename>`
- Count words – `wc -w <filename>`
- Count characters – `wc -c <filename>`
- Count lines & words – `wc -lw <filename>`
- Also “wc” can be used with input redirection
 - `cat <filename> | wc -l`
 - `ls | wc`

6/2/2005 31

Department of Computer and Information Sciences, UAB

Copy Files

- Copy files using “cp” command
- Usage: `cp <options> <source> <target>`
- Some common options:
 - Interactive (prompt when overwriting) – `i`
 - Preserve permissions and ownership – `p`
 - Recursive copy – `r`
- Examples:
 - `cp mydoc newdoc`
 - `cp mydoc ../`
 - `cp ../newdoc .`
 - `cp -r mydir /home/puri/old-dir/`

6/2/2005 32

Department of Computer and Information Sciences, UAB

Rename or Move Files

- Rename/move files using “mv” command
- Usage: `mv <options> <source> <target>`
- Some common options:
 - Prompt for confirmation – `i`
 - Move without confirmation (force) – `f`
- Examples:
 - `mv -i mydoc newdoc`
 - `mv -f mydoc ../`
 - `mv ../newdoc .`
 - `mv mydir /home/puri/old-dir/`

6/2/2005 33

Remove Files

- Remove files using “*rm*” command
- Usage: `rm <options> <filename>`
- Some common options:
 - Prompt for confirmation - `i`
 - Move without confirmation (force) - `f`
 - Recursive remove - `r`
- Examples:
 - `rm mydoc`
 - `rm -f mydoc ../newdoc`
 - `rm -i ../newdoc`
 - `rm -r mydir`
