

CS-333 - Unix Operating System Fundamentals

Lecture 2

Touring the Features of UNIX/Linux & Editing with the Visual Editor

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

Syntax for UNIX Commands

- *command* *[[-]option(s)]*
[option argument(s)]
[command argument(s)]
- **Examples:**
 - *ls*
 - *ls -la*
 - *ls -la m**
 - *lpr -Pspr -n 3 proposal.ps*

8/21/2006

3

Sample Commands

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

8/21/2006

4

Miscellaneous

- How does the shell distinguish arguments from utilities?
 - *A B C | D E | 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

8/21/2006

5

“man” Command

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

```
moat.cs.uab.edu - UAB - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
Reformatting page. Please Wait... done
Dear Commands man(1)
NAME
man -- find and display reference manual pages
SYNOPSIS
man [ - ] [ -adflit ] [ -M path ] [ -T macro-package ] [
  -w position ] name ...
man [ -M path ] -k keyword ...
man [ -M path ] -f file ...
DESCRIPTION
The man command displays information from the reference
manuals. It displays complete manual pages that are selected
by name, or one-line summaries selected either by keyword
(-k) or by the name of an associated file (-f). If no
manual page is located, man prints an error message.
Source Format
man(1)
Connected to moat.cs.uab.edu SSH2 - aes128-cbc - hmac-md5 - none 80x24
8/21/2006
```

8/21/2006

6

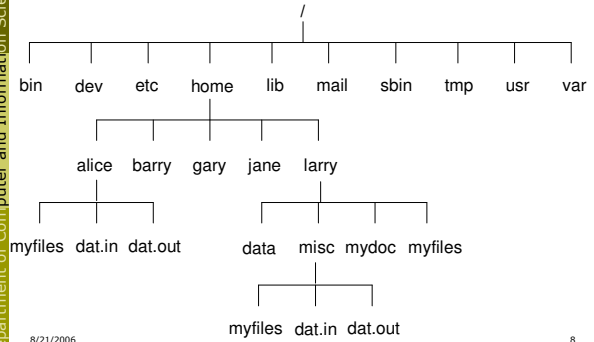
Directory Structure

- 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/afgane)
- "pwd" prints the current directory we are in
- "cd" is used to change directories

8/21/2006

7

Typical Directory Structure



8/21/2006

8

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/afgane/mydir; mkdir ../mydir`
- Remove directory
 - `rmdir <options> <directory-name>`
 - `rmdir mydir; rmdir /home/afgane/mydir; rmdir ../mydir`
- Details about a command can be obtained using the "man command" – e.g., `man mkdir`

8/21/2006

9

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`
 - List all the files including hidden files: `ls -a`
 - see man page for more options

8/21/2006

10

Simple Operations with Files I

- Creating files – `cat > myfile`
- Sorting contents of a file – `sort`
- View Files with "more"
 - search for a string → `/<string>`
 - move forward and backwards → `n` and `b`
- Other options to view files – `cat`, `head`, `tail`
- Count lines, words, characters – `wc`

8/21/2006

11

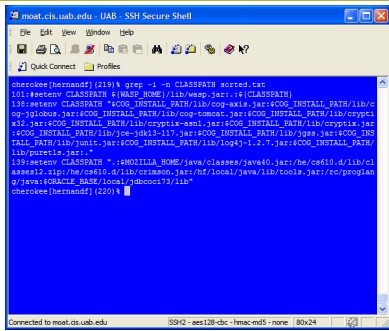
Simple Operations with Files II

- Copy files – `cp`
- Rename files – `mv`
- Remove files – `rm`
- Locate specific lines in a file – `grep`

8/21/2006

12

grep Example



8/21/2006

13

Device Independence

- Devices (printer, terminal, disk files, modems) all appear as files to UNIX programs
- Output Redirection
 - Instruct a command to send the output to any device or file
- Input Redirection
 - Instruct a command to receive its input from any device or file
- Input/Output are device independent

8/21/2006

14

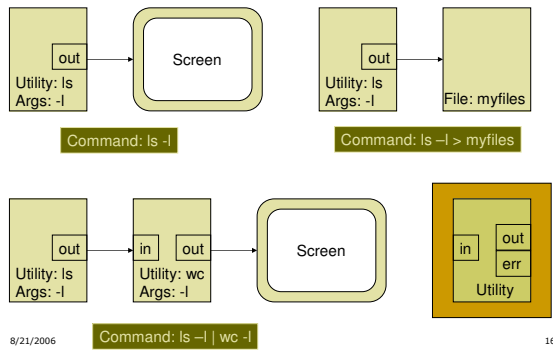
Redirecting Output I

- By default output from a utility is sent to the terminal/screen
- 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 > myfile`
- 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

8/21/2006

15

Redirecting Output II



8/21/2006

16

Other Directives

- Input redirection - "<"
 - e.g., `pr < myfile`
- Append output to a file - ">>"
 - e.g., `cat newfile >> outfile`
- Redirect error messages - "2>"
 - e.g., `cat nofile 2> errors`
- Not display error messages
 - e.g., `cat nofile 2> /dev/null`
- Execute multiple commands - ";", ";&"
 - e.g., `clear; date`

8/21/2006

17

Creating Files

- We have already seen creating files using
 - output redirection with the ">" directive
 - appending a file using the ">>" directive
- Using cat and ">" directive


```

$ cat > myfile
This is some text being entered into file - myfile
CTRL-D
$ cat myfile
This is some text being entered into file - myfile
$
            
```
- By default the keyboard is connected to the input stream of "cat" utility

8/21/2006

18

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.*)

8/21/2006

19

"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`

8/21/2006

20

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/afgane/old-dir/`

8/21/2006

21

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/afgane/old-dir/`

8/21/2006

22

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`

8/21/2006

23

Recap ... (1)

- Simple commands:
 - `pwd` → identify current directory
 - `man` → help on commands
 - `ls` → list files
 - `ps` → list processes
 - `cd` → change directory
 - `cat, more, head, tail` → display files
 - `cp` → copy files
 - `mv` → move (also rename) files
 - `rm` → remove files

8/21/2006

24

Recap ... (2)

- Other commands:
 - history
 - clear
 - date
 - alias
- Input/Output redirection
 - > to file (rewrites the file)
 - < from file
 - >> append
- Execute more than one command
 - ;