The Utilities

Learning Objectives

- Issue commands
- Use basic utilities to list files
- Copy, move, and remove files
- List special characters
- Search, sort, print, and compare text files
- Pipe commands
- Communicate with other users

Issue Commands

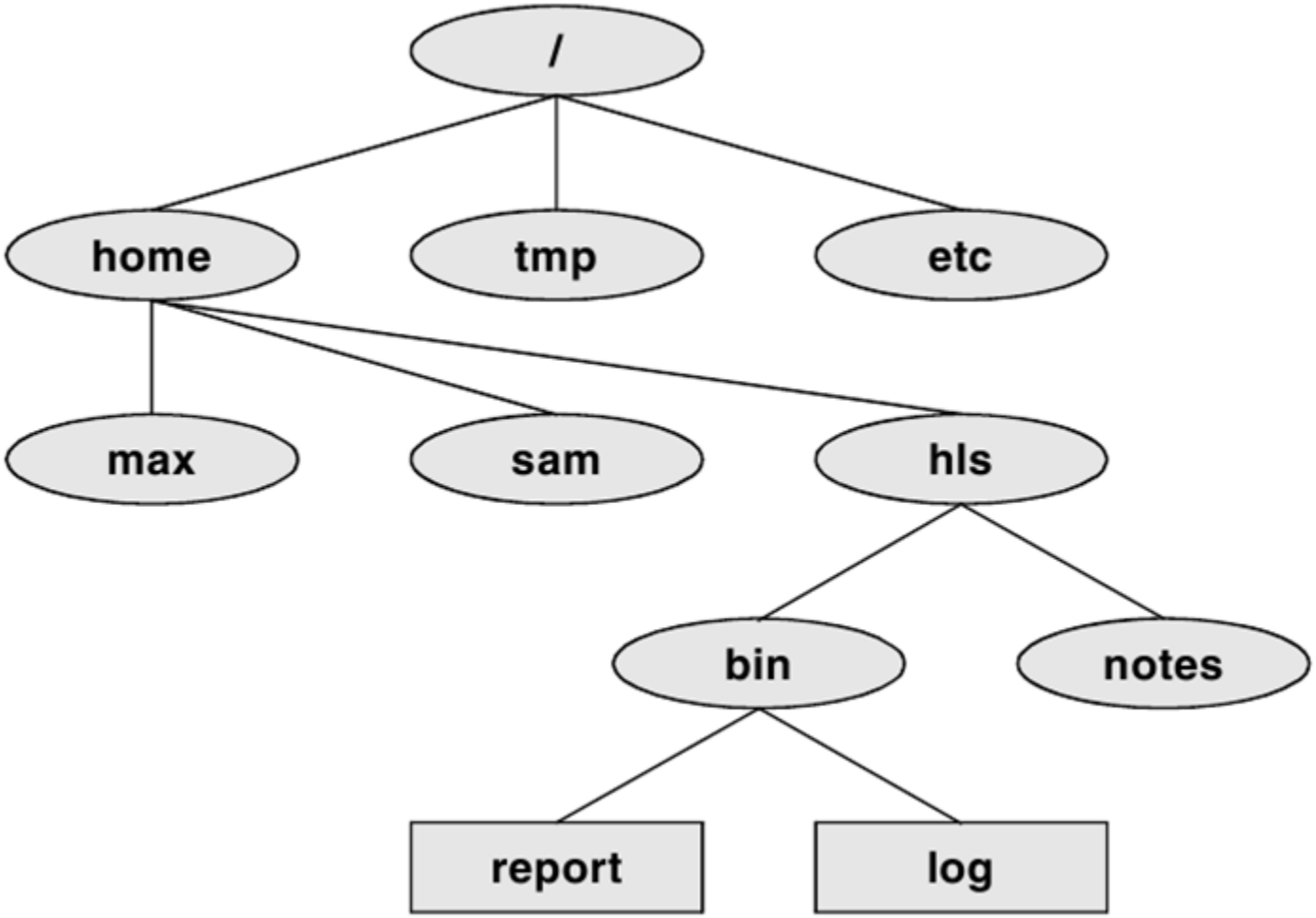
- Command <ENTER>
- Command ; Command; Command

```
who; date; ls
```

Bourne Shell Prompt

- PS1(default): $
- PS2: >
- PS3: not used

Linux File Structure



hostname: Displays the System Name

```
[victoryu@voyager:~/cis18a ] $hostname
voyager.deanza.edu
[victoryu@voyager:~/cis18a ] $hostname -a
voyager
[victoryu@voyager:~/cis18a ] $hostname -d
deanza.edu
[victoryu@voyager:~/cis18a ] $hostname -f
voyager.deanza.edu
[victoryu@voyager:~/cis18a ] $hostname -i
153.18.17.12
[victoryu@voyager:~/cis18a ] $hostname -s
voyager
[victoryu@voyager:~/cis18a ] $hostname -V
net-tools 1.60
hostname 1.100 (2001-04-14)
[victoryu@voyager:~/cis18a ] $hostname -y
(none)
[victoryu@voyager:~/cis18a ] $
```

File Name

- A file name can be any sequence of ASCII characters
- Some characters, although allowed, should not be used, e.g., >
- Make file name as meaningful as possible
- Never start a file name with a period (.) unless it’ s a hidden file

Matching Any Single Character

File	Matches				Does Not Match	
c?	c1	c2	c3	ca	ac	cat
c?t	cat	cet	cit	c1t	cad	dac
c??t	caat	cabt	cact	c12t	cat	daat
?a?	bat	car	far	mar	bed	cur

Matching A Single Character From a Set

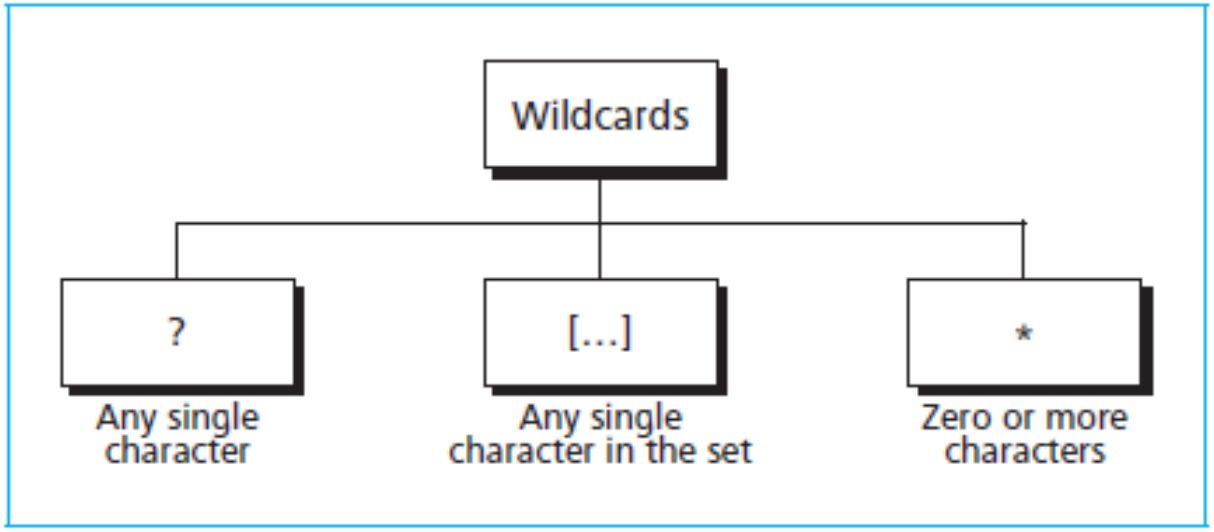
File	Matches				Does Not Match	
f[aoei]d	fad	fed	fod	fid	fud	fab
f[a-d]t	fat	fbt	fct	fdt	fab	fet
f[A-z][0-9]	fA3	fa3	fr2	f^2	FA3	fa33

Matching Zero or More Characters

Wildcard	Matches	Example	Does Not Match
*	every file		
f*	every file whose name begins with *f*	f5c2	afile, cat
*f	every file whose name ends in *f*	staff	f1, faF
.	every file whose name has a period	file.dat	bed, cur

Wildcards

- Tokens that represent one or more characters in a search
- There are three types of wildcards



Explore the System

- **`ls`**: lists directories and files
- **`cd`**: changes directory
- telnet: login in to a remote system
- **`ssh`**: connects to another system *securely*
- **`pwd`**: which directory am I in?

Find Out Who Is Online

- **`who`**: shows who's logged in
- **`finger`**: looks up user information
- **`w`**: shows who's logged in and what they are doing

Display File Contents

- **cat**
- **echo**

```
$echo 'My Text' > myfile
$echo f?t
f-t flt fat fbt fgt fwt
```

Display Partial File Contents

- **`more`** or **`less`**: display one page at a time
- **`head`**: display the beginning of a file
- **`tail`**: display the end of a file

grep: Find Text

- Find a string in a specific file, and print all the lines that contain the text

```
$grep eggs basket
```

- Display the line that contains the text and the surrounding lines

```
$grep -5 eggs basket
```

- Find out how many times a string appears:

```
$grep –c eggs basket
```

cp: Copy a File

```
$ ls
memo
$ cp memo memo.copy
$ ls
memo memo.copy
```

rm and Its Danger

```
$ ls
practice
$ cat practice
This is a small file that I created
with a text editor.
$ rm practice
$ ls
$ cat practice
cat: practice: No such file or directory
$
```

Figure 3-1 Using ls, cat, and rm on the file named **practice**

mv: Rename a File

- mv: rename a file without making a copy
- Caution: mv can destroy a file:
  - Use the –i option as a safe guard

```
$ ls
memo
$ mv memo memo.0130
$ ls
memo.0130
```

`date`: Displays the Time and Date

- **`date`** displays the current date and time
  - You can specify the format

```
$date +"%A, %B %d, %Y“
$Tuesday, January 21, 2014
```

`cal`: Displays Calendar

- **`cal`** displays the calendar of a month or year
  - If no arguments are provided, it prints the current year
  - If both a month and a year are provided, it prints one month

```
$cal -1
January 2014
Su Mo Tu We Th Fr Sa
           1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

`sort` It Out

- **`sort`** sorts the line according to ASCII order
  - **`-d`** option: in dictionary order
  - **`-f`** option: ignore case
  - **`-r`** option: in reverse order
  - **`-t`** option: change the default delimiter
  - **`-k n[,m]`**: starts at the *n*th field, where the fields are numbered beginning with 1, and ends at the *m*th field

`uniq`: Eliminate Duplicates

- **`uniq`** finds only identical, adjacent (sorted) lines.

  `use -d` option to display only the duplicate lines

- You can sort and eliminate duplicates in one step with:

  `sort –u long.address.book`

`write`: Communicate with Others

- **write** sends a quick message to others who logged in to the same system.
  1. `write userid`
  2. `Wanna meet for lunch?`
  3. `Ctrl+D`
- The message will suddenly appear on the recipient's screen and can be intrusive
- Keep other people from sending you message

  `mesg n`

Special Characters

Characters with special meaning to the shell

- Whitespaces: RETURN, SPACE, TAB
- Quotes:
  - 'text' - protects the contents
  - "text" - expands the contents
  - `text` - executes as a command

Special Characters

- `*` substitutes for zero or more characters
- `?` substitutes for any single character
- `[charset]` substitutes for any single character in the set `charset`

Locate a Utility

- `which`
- `whereis`

Running Commands in Background

```
$xclock &
[1] 11446
$jobs
[1]+ Running                 xclock &
$fg 1
xclock
```

ctrl-c

```
$
```

Pipe and Redirect Output

- Pipe (|): connect the output of one program to the input of another program
- Redirect (>): redirect the output to a file
- Redirect and append (>>): redirect the output and append to a file

Questions (1)

- Which commands can you use to determine who is logged in on a specific terminal?
- What happens when you give the following commands if the file named done already exists?

```
$cp to_do done
$mv to_do done
```

- How can you find out which utilities are available on your system for editing files? Which utilities are available for editing on your system?

Questions (2)

- Which commands can you use to look at the first few lines and the end of a file named **`status_report`**?
- Try giving these two commands:

  `$echo cat`

  `$cat echo`

  Explain the differences between the output of each command