

Command-Line Unix

Charles Duan

cduan@fas.harvard.edu

FAS Computer Services

March 4, 2003

COMMAND-LINE UNIX

- Files and the File System
- Commands and Aliases
- Modifying your Workspace
- Printing

THE FIRST RULE

The most important thing to know about Unix:

THE FIRST RULE

The most important thing to know about Unix:



Ask this guy

FILES AND THE FILE SYSTEM

- Directories
- Finding, viewing, arranging, editing
- File permissions

DIRECTORIES

DIRECTORIES

- Directories are hierarchical folders

DIRECTORIES

- Directories are hierarchical folders
- Commands: `cd`, `mkdir`, `rmdir`

DIRECTORIES

- Directories are hierarchical folders
- Commands: `cd`, `mkdir`, `rmdir`
- Home directories: `~` and `/home/x/y/xyzabc`

DIRECTORIES

- Directories are hierarchical folders
- Commands: `cd`, `mkdir`, `rmdir`
- Home directories: `~` and `/home/x/y/xyzabc`
- “Recursive” traversal: the `-r` option

FINDING, VIEWING, ARRANGING, EDITING

FINDING, VIEWING, ARRANGING, EDITING

- Finding: `ls`, `find`, `grep`, `du`

FINDING, VIEWING, ARRANGING, EDITING

- Finding: `ls`, `find`, `grep`, `du`
- Viewing: `cat`, `less`

FINDING, VIEWING, ARRANGING, EDITING

- Finding: `ls`, `find`, `grep`, `du`
- Viewing: `cat`, `less`
- Arranging: `mv`, `cp`, `rm`

FINDING, VIEWING, ARRANGING, EDITING

- Finding: `ls`, `find`, `grep`, `du`
- Viewing: `cat`, `less`
- Arranging: `mv`, `cp`, `rm`
- Editing: `pico`, `vi`, `vim`

FINDING, VIEWING, ARRANGING, EDITING

- Finding: `ls`, `find`, `grep`, `du`
- Viewing: `cat`, `less`
- Arranging: `mv`, `cp`, `rm`
- Editing: `pico`, `vi`, `vim`
- Knowing a Unix text editor is **absolutely necessary**

FILE PERMISSIONS

FILE PERMISSIONS

- File permissions let other users access your files

FILE PERMISSIONS

- File permissions let other users access your files
- Where are they useful?

FILE PERMISSIONS

- File permissions let other users access your files
- Where are they useful?
 - Web pages

FILE PERMISSIONS

- File permissions let other users access your files
- Where are they useful?
 - Web pages
 - Group projects

FILE PERMISSIONS

- File permissions let other users access your files
- Where are they useful?
 - Web pages
 - Group projects
 - Finger: `.plan` file

FILE PERMISSIONS

- File permissions let other users access your files
- Where are they useful?
 - Web pages
 - Group projects
 - Finger: `.plan` file
- Can be tricky to use—ask someone!

COMMANDS AND ALIASES

- What are commands?
- Finding programs/the PATH variable
- Command aliases
- Where to get help

WHAT ARE COMMANDS?

WHAT ARE COMMANDS?

- Two ways to think about commands:

WHAT ARE COMMANDS?

- Two ways to think about commands:
 1. Special files that contain programs

WHAT ARE COMMANDS?

- Two ways to think about commands:
 1. Special files that contain programs
 2. Something typed at the command prompt

WHAT ARE COMMANDS?

- Two ways to think about commands:
 1. Special files that contain programs
 2. Something typed at the command prompt
- *Not* always the same! (But they usually are)

WHAT ARE COMMANDS?

- Two ways to think about commands:
 1. Special files that contain programs
 2. Something typed at the command prompt
- *Not* always the same! (But they usually are)
- Some general Unix command categories

WHAT ARE COMMANDS?

- Two ways to think about commands:
 1. Special files that contain programs
 2. Something typed at the command prompt
- *Not* always the same! (But they usually are)
- Some general Unix command categories
 - File system

WHAT ARE COMMANDS?

- Two ways to think about commands:
 1. Special files that contain programs
 2. Something typed at the command prompt
- *Not* always the same! (But they usually are)
- Some general Unix command categories
 - File system
 - Processing text files (sorting, searching, etc.)

WHAT ARE COMMANDS?

- Two ways to think about commands:
 1. Special files that contain programs
 2. Something typed at the command prompt
- *Not* always the same! (But they usually are)
- Some general Unix command categories
 - File system
 - Processing text files (sorting, searching, etc.)
 - Network/Internet (e-mail, IM, diagnostics)

FINDING PROGRAMS

FINDING PROGRAMS

- Typing the *full path* for a command

FINDING PROGRAMS

- Typing the *full path* for a command
 - `/usr/local/bin/ls`

FINDING PROGRAMS

- Typing the *full path* for a command
 - `/usr/local/bin/ls`
 - `./dining`

FINDING PROGRAMS

- Typing the *full path* for a command
 - `/usr/local/bin/ls`
 - `./dining`
- But if you just type the command name without the path, what do you run?

FINDING PROGRAMS

- Typing the *full path* for a command
 - `/usr/local/bin/ls`
 - `./dining`
- But if you just type the command name without the path, what do you run?
- The `which` command:

FINDING PROGRAMS

- Typing the *full path* for a command
 - `/usr/local/bin/ls`
 - `./dining`
- But if you just type the command name without the path, what do you run?
- The `which` command:

```
% which ls
```

FINDING PROGRAMS

- Typing the *full path* for a command
 - `/usr/local/bin/ls`
 - `./dining`
- But if you just type the command name without the path, what do you run?
- The `which` command:

```
% which ls
/usr/local/bin/ls
```

FINDING PROGRAMS

- Typing the *full path* for a command
 - `/usr/local/bin/ls`
 - `./dining`
- But if you just type the command name without the path, what do you run?
- The `which` command:

```
% which ls
/usr/local/bin/ls
```
- How do you *specify* which command to use?

THE PATH VARIABLE

THE PATH VARIABLE

PATH is a colon-separated list of directories

THE PATH VARIABLE

PATH is a **colon-separated list of directories**

Say there are `/bin/ls`, `/usr/bin/ls`, `/home/l/i/lib51/bin/ls` and

PATH is `/bin:/usr/bin`

THE PATH VARIABLE

PATH is a **colon-separated list of directories**

Say there are `/bin/ls`, `/usr/bin/ls`, `/home/l/i/lib51/bin/ls` and

PATH is `/bin:/usr/bin`

```
% echo $PATH
```

```
/bin:/usr/bin
```

THE PATH VARIABLE

PATH is a **colon-separated list of directories**

Say there are `/bin/ls`, `/usr/bin/ls`, `/home/l/i/lib51/bin/ls` and

PATH is `/bin:/usr/bin`

```
% echo $PATH
```

```
/bin:/usr/bin
```

```
% which ls
```

```
/bin/ls
```

THE PATH VARIABLE

PATH is a **colon-separated list of directories**

Say there are `/bin/ls`, `/usr/bin/ls`, `/home/l/i/lib51/bin/ls` and

PATH is `/bin:/usr/bin`

```
% echo $PATH
```

```
/bin:/usr/bin
```

```
% which ls
```

```
/bin/ls
```

```
% setenv PATH /home/l/i/lib51/bin:{$PATH}
```

```
% which ls
```

```
/home/l/i/lib51/bin/ls
```

THE PATH VARIABLE (CONTINUED)

Two things to note:

THE PATH VARIABLE (CONTINUED)

Two things to note:

1. Searches are done in order, and the first one found is selected

THE PATH VARIABLE (CONTINUED)

Two things to note:

1. Searches are done in order, and the first one found is selected
2. Make sure to **always** include the old PATH in the new PATH

THE PATH VARIABLE (CONTINUED)

Two things to note:

1. Searches are done in order, and the first one found is selected
2. Make sure to **always** include the old PATH in the new PATH

Alternate specification:

```
% set path = (/home/l/i/lib51/bin $path)
```

COMMAND ALIASES

COMMAND ALIASES

Aliases are “command nicknames”

COMMAND ALIASES

Aliases are “command nicknames”

Created by adding to the *.aliases* file

COMMAND ALIASES

Aliases are “command nicknames”

Created by adding to the *.aliases* file

Example:

```
alias sb sudo budget
```

Now, typing **sb** will run **sudo budget**

COMMAND ALIASES

Aliases are “command nicknames”

Created by adding to the *.aliases* file

Example:

```
alias sb sudo budget
```

Now, typing **sb** will run **sudo budget**

To use, either run **source .aliases** or logout and login again

WHERE TO GET HELP

WHERE TO GET HELP

- The `man` command

WHERE TO GET HELP

- The `man` command
- The `apropos` command

WHERE TO GET HELP

- The `man` command
- The `apropos` command
- The Internet

WHERE TO GET HELP

- The `man` command
- The `apropos` command
- The Internet
- Refer to Slide 3

MODIFYING YOUR WORKSPACE

- Login/logout settings
- Shell settings

LOGIN/LOGOUT SETTINGS

LOGIN/LOGOUT SETTINGS

- *.cshrc* and *.login* files run on login

LOGIN/LOGOUT SETTINGS

- *.cshrc* and *.login* files run on login
 - Only the *.login* should print out anything

LOGIN/LOGOUT SETTINGS

- *.cshrc* and *.login* files run on login
 - Only the *.login* should print out anything
- *.logout* runs on logout

LOGIN/LOGOUT SETTINGS

- *.cshrc* and *.login* files run on login
 - Only the *.login* should print out anything
- *.logout* runs on logout
- All of these are *text files* containing commands, one per line

LOGIN/LOGOUT SETTINGS

- *.cshrc* and *.login* files run on login
 - Only the *.login* should print out anything
- *.logout* runs on logout
- All of these are *text files* containing commands, one per line
- To test, type **source .login** (or the appropriate file)

CHANGING YOUR PROMPT

Put this in your `.cshrc` file:

```
set prompt = ...
```

CHANGING YOUR PROMPT

Put this in your `.cshrc` file:

```
set prompt = ...
```

```
set prompt = Sample prompt
```

CHANGING YOUR PROMPT

Put this in your `.cshrc` file:

```
set prompt = ...
```

<code>set prompt =</code>	Sample prompt
<hr/>	
<code>"prompt%"</code>	<code>prompt%</code>

CHANGING YOUR PROMPT

Put this in your `.cshrc` file:

```
set prompt = ...
```

<code>set prompt =</code>	Sample prompt
<code>"prompt%"</code>	<code>prompt%</code>
<code>" [%M:\!] %~> "</code>	<code>[ice3:30] ~></code>

CHANGING YOUR PROMPT

Put this in your `.cshrc` file:

```
set prompt = ...
```

<code>set prompt =</code>	Sample prompt
<code>"prompt%"</code>	<code>prompt%</code>
<code>" [%M:\!] %~> "</code>	<code>[ice3:30] ~></code>
<code>"%M: %B%t%b %~: "</code>	<code>iso2: 11:42am ~:</code>

CHANGING YOUR PROMPT

Put this in your `.cshrc` file:

```
set prompt = ...
```

<code>set prompt =</code>	Sample prompt
<code>"prompt%"</code>	<code>prompt%</code>
<code>" [%M:\!] %~> "</code>	<code>[ice3:30] ~></code>
<code>"%M: %B%t%b %~: "</code>	<code>iso2: 11:42am ~:</code>

Many more options—read [man tcsh](#)

PRINTING

- Viewing printing status
- Printable files
- Printing files

VIEWING PRINTING STATUS

VIEWING PRINTING STATUS

- `echo $printers` to list printers

VIEWING PRINTING STATUS

- `echo $printers` to list printers
- `lpq -P[printer name]` to show active print jobs

VIEWING PRINTING STATUS

- `echo $printers` to list printers
- `lpq -P[printer name]` to show active print jobs
- You *can* put a space between the `-P` and the printer name

PRINTABLE FILES

PRINTABLE FILES

Plain text Contain only printable characters

PRINTABLE FILES

Plain text Contain only printable characters

- Check with `less`

PRINTABLE FILES

Plain text Contain only printable characters

- Check with `less`

Postscript A “page description language”

PRINTABLE FILES

Plain text Contain only printable characters

- Check with `less`

Postscript A “page description language”

- Files start with
“%!PS-Adobe-3.0” (or 2.0 or 1.0)

PRINTABLE FILES

Plain text Contain only printable characters

- Check with **less**

Postscript A “page description language”

- Files start with
“%!PS-Adobe-3.0” (or 2.0 or 1.0)
- Again, check with **less** or **head**

PRINTABLE FILES

Plain text Contain only printable characters

- Check with **less**

Postscript A “page description language”

- Files start with
“%!PS-Adobe-3.0” (or 2.0 or 1.0)
- Again, check with **less** or **head**

Don't print any other type of file!

PRINTING FILES

PRINTING FILES

Command	Plain text	Postscript	PDF

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript			

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript	Yes		

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript	Yes	No	

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript	Yes	No	No

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript	Yes	No	No
lpr			

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript	Yes	No	No
lpr	Yes		

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript	Yes	No	No
lpr	Yes	Yes	

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript	Yes	No	No
lpr	Yes	Yes	No

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript	Yes	No	No
lpr	Yes	Yes	No

Both commands are of syntax

`[command] -P[printer] [files]`

PRINTING FILES

Command	Plain text	Postscript	PDF
enscript	Yes	No	No
lpr	Yes	Yes	No

Both commands are of syntax

`[command] -P[printer] [files]`

Without the `-P` option you will print to `scb11`

