Bash (Origin: 1635–45)

- Verb
  - to strike with a crushing or smashing blow.
  - Chiefly British, Canadian. to hurl harsh verbal abuse at
- Noun
  - a crushing blow.
  - Informal. a thoroughly enjoyable, lively party.
- Idioms
  - have a bash (at), British. to attempt; make an attempt.
  - on the bash, British. working as a prostitute.

Dictionary.com, "bash," in *Dictionary.com Unabridged*. Source location: Random House, Inc. http://dictionary.reference.com/browse/bash. Available: http://dictionary.reference.com. Accessed: October 01, 2009.

David Notkin ● Autumn 2009 ● CSE303 Lecture 2

Odds and ends

- Final is on Tuesday December 15th – thanks for the good catch!
- Books
- Where and how to compute?
  - "Do I have to install Linux?" No
  - "Can I install Linux?" Yes
  - "Can I run on Windows?" Yes
  - "Can I run on a Mac?" Yes
  - "Can I run from home?" Yes
  - …

Rumors about 303: selection

- Big workload for a 3 credit class; harder than 143; Linux heavy; C is less friendly than Java
- Pointers in C and C++ may be confusing (or funnest thing since Slip and Slide); memory management is hard
- Learn [several|10] programming languages
- It is the 152nd odd number greater than zero. This does not worry me.
- Many of the group programming assignments are very challenging
- I have heard that the grading curve in CSE 303 is extraordinarily competitive and that it is typically dominated by a group of very experienced students who get perfect scores on nearly all assignments and tests making it very difficult for other students to attain good grades due to the curve.
- An [easy | fun |hard | not that bad |straightforward | awesome | time-consuming | useful | different] class
- It's a lot of a mess of junk that they put in one class.
- Other people know way more than you.
- You will get lost really quickly if you aren't fast in picking up Unix.
- We have to use comments.

Rumors about Notkin: selection

- Software researcher.
- I heard you are an interesting guy.
- You are a strict grader
- He is a hard grader
- He appears to have glasses.
- He's a funny dude

Rumors about beard: selection

- Beardly things. EPIC BEARD
- The beard [is old | grows mold | may come alive and swallow you whole | would make Moses cower | has existed since the 80s ("this worries me") | is over 25 years old ("incredibly awesome") | is fake | is kick ass ("confirmed true") | is nice | is long | was cursed by the Linux penguin | is older than myself].
- He keeps students he doesn't like trapped in his beard.
- No one knows what's hidden in his beard.
- Tiny gnomes hide in his beard, and that's where he gets his powers
- I haven't heard any rumors about David Notkin, but I have been told that the possesion of a beard of such calibre indicates a mastery of Unix.
- That a portal to an alien world is hiding in his beard, and the aliens (Nortkinians) may decide to overrun our planet at any time.

Tend to learn best?

- 16% From watching others do something
- 63% From trying it myself
- 6% From reading manuals
- 15% Other

Experience with Unix?

- 49% None
- 37% A little
- 11% Quite a bit
- 3% Pays my tuition

Experience with C/C++?

- 52% None
- 37% A little
- 9% Quite a bit
- 2% Pays my tuition

Where do you compute?

- 11% CSE labs
- 11% Own computer using CSE cycles
- 68% Own computer almost entirely
- 10% Other

Practice

- Almost all abilities demand a significant amount of relatively rote practice: piano and scales, baseball and playing catch, painting and figure drawing, etc.
- Over time, the rote activities become second nature and it is possible to chunk/abstract those activities to make it far easier to do more complex things
- The next couple of lectures and assignments will, to some degree, focus on some of these rote activities in using Unix commands
- For those of you who have already developed these rote abilities, a refresher is good once in a while

Demo: throughout today's lecture

Basics

- Basic shell commands
- Unix file system structure: "everything" is a file
- Commands for file manipulation, examination, searching
- Keyboard shortcuts and special characters

Shell commands

- A name followed by an optional set of arguments/parameters
  - Example: **cp** (copy) accepts a source and destination file path
- A program uses three streams of information
  - **stdin**, **stdio**, **stdout** (standard in, out, error)
  - Defaults: input from the keyboard, output and errors to the console
- Parameters vs. input
  - parameters: before [Enter] is pressed; sent to the program by the shell
  - input: after [Enter] is pressed; sent to the program during execution by the user

Directory commands

command	description
`ls`	list files in a directory
`pwd`	output the current working directory
`cd`	change the working directory
`mkdir`	create a new directory
`rmdir`	delete a directory (must be empty)

Command line arguments

- Most options are given after the command name using a dash followed by a letter: **-c**, **-h**, **-S**, …
- Some options are longer words preceded by two dashes:
  --count, **--help**
- Parameters can be combined: **ls -l -a -r** can be **ls -lar**
- Many programs accept a **-help** parameter; others provide help if run with no arguments
- Many commands accept a file name parameter: if it is omitted, the program will read from standard input

Relative naming

directory	description
`.`	the directory you are in ("working directory")
`..`	the parent of the working directory (`../..` is grandparent, etc.)
`~`	your home directory (on many systems, this is `/home/`***username***)
`~`***username***	***username***'s home directory
`~/Desktop`	your desktop

Shell/system commands

command	description
`man or info`	get help on a command
`apropos (man -k)`	search for commands by keyword
`clear`	clears out the output from the console
`exit`	exits and logs out of the shell

command	description
`date, time`	output the system date/time
`cal`	output a text calendar
`uname`	print information about the current system

- "man pages" are a very important way to learn new commands

File commands

command	description
`cp`	copy a file
`mv`	move or rename a file
`rm`	delete a file
`touch`	update a file's last-modified time stamp (or create a new empty file)

- CAUTION: the above commands do not prompt for confirmation, so it's easy to overwrite/delete a file
- This setting can be overridden (how?)

Links

command	description
`ln`	create a link to a file
`unlink`	remove a link to a file

- *hard link*: two names for the same file
  - **`ln foo bar`**
  - This links **`bar`** as a duplicate name for **`foo`**
  - If one is modified, the other is too; if one is deleted, both will go away
- *soft (symbolic) link*: A reference to another file
  - **`ln -s snork fork`**
  - This creates a reference **`fork`** to the file **`snork`**
    - **`fork`** can be used as though it were **`snork`**
    - If **`fork`** is deleted, **`snork`** will be unaffected

File examination

command	description
`cat`	output a file's contents on the console
`more or less`	output a file's contents, one page at a time
`head, tail`	output the first or last few lines of a file
`wc`	count words, characters, and lines in a file
`du`	report disk space used by a file(s)
`diff`	compare two files and report differences

- Suppose you are writing a paper, and the teacher says it can be anything as long as it is at least 200 words long and mentions 303...

Searching and sorting

command	description
`grep`	search a file for a given string
`sort`	convert an input into a sorted output by lines
`uniq`	strip duplicate lines
`find`	search for files within a given directory
`locate`	search for files on the entire system
`which`	shows the complete path of a command

- **`grep`** is a very powerful search tool; more later...
- *Exercise* : Given a text file **`students.txt`**, display the students arranged by the reverse alphabetical order of their last names.
  - Can we display them sorted by first name?

Keyboard shortcuts

^***KEY*** means hold `Ctrl` and press ***KEY***

key	description
`Up arrow`	repeat previous commands
`Home/End or ^A/^E`	move to start/end of current line
"	quotes surround multi-word arguments and arguments containing special characters
*	"wildcard" , matches any files; can be used as a prefix, suffix, or partial name
`Tab`	auto-completes a partially typed file/command name
`^C or ^\`	terminates the currently running process
`^D`	end of input; used when a program is reading input from your keyboard and you are finished typing
`^Z`	suspends (pauses) the currently running process
`^S`	don't use this; hides all output until ^G is pressed

File system

directory	description
`/`	root directory that contains all others (drives do not have letters in Unix)
`/bin`	programs
`/dev`	hardware devices
`/etc`	system configuration files ▪ `/etc/passwd` stores user info ▪ `/etc/shadow` stores passwords
`/home`	users' home directories
`/media,/mnt,...`	drives and removable disks that have been "mounted" for use on this computer
`/proc`	currently running processes (programs)
`/tmp, /var`	temporary files
`/usr`	user-installed programs

Questions?