Unix Shell Environments

September 27, 2004
Class Meeting 6, Part I

Shell Characteristics
- Command-line interface between the user and the operating system
- Automatically starts on login, waits for user to type in commands
- Both a command interpreter and a programming language
- Shell script is a text file containing logic for shell interpretation

Shell Interactivity
- Command line parsing
- Environment
- Textual completion
- Aliases
- Command line editing
- Command history
- Configuration

Shell Programming
- Variables
- Control structures
  - Loops and conditionals
- Function definition and invocation
- Shell scripts
- Next lecture

Various Unix Shells
- `sh` (Bourne shell, original Unix shell)
- `ksh` (Korn shell)
- `csh` (C shell, developed at Berkeley)
- `tcsh`
- `bash` (Bourne again SHell)
  - Default user shell in Linux

Bourne Again SHell
- `bash` is the standard shell for this class
- Superset of the Bourne shell (`sh`)
- Borrows features from `sh`, `csh`, `tcsh`, and `ksh`
- Part of the GNU project
Environment Variables

- A set of variables the shell uses for certain operations
- Variables have a name and a value
- Current list can be displayed with the `env` command
- Send value of `varname` to standard output with `echo $varname`

Environment Variable Examples

```
[cs2204@acorn bin]$ echo $PS1
$`

[cs2204@acorn bin]$ echo $PATH
/usr/local/bin:/bin:/usr/X11R6/bin:/usr/games:/home/courses/cs2204/bin
```

Setting Environment Variables

- Set variable with `varname=value`
- `PS1=$USER@$HOSTNAME:`
  - Change default shell prompt
- `PATH=$PATH:$HOME/bin`
  - Append `:~:.` to `PATH`
- `DATE=` or `DATE=$(date)`
  - Capture output from `date` command

Textual Completion

- `<tab>` attempts to complete the current command or filename
- `pus<tab>` expands to `pushd space`
- `pu<tab>` gives the alternatives `pu` `pup` `pushd`
- `ln /etc, entering ll modu<tab>` gives `modules modules.conf modules.devfs`
  - `[cs2204@blackberry etc]$ ll modules`

Aliases

- `alias shortcut=command`

Examples

- `alias pu=pushd`
- `alias po=popd`
- `alias l="ls -F -C"`
- `alias ll="ls -l -l -F"`
- `alias d=dirs`
- `alias hide="chmod og-rwx"`
- `alias unhide="chmod og+r"`

Command History

- Use `history` command to list previously entered commands
- Use `fc -l <m> <n>` to list previously typed commands from `m` through `n`
- Use up and down cursor keys to scroll through history list
Editing on the Command Line

- **bash** provides a number of line editing commands
- Default **emacs**-mode commands
  - Esc-b Move back one word
  - Esc-f Move forward one word
  - Ctrl-a Move to beginning of line
  - Ctrl-e Move to end of line
  - Ctrl-k Kill text from cursor to end of line

Login Scripts

- Startup scripts executed at login
  - /etc/profile
  - ~/.bash_profile
  - ~/.bash_login (if no .bash_profile)
  - ~/.profile (if neither are present)
  - ~/.bashrc

- Script executed upon logout
  - ~/.bash_logout

Example .bash_profile (partial)

```bash
# .bash_profile
# include .bashrc if it exists
if [ -f ~/.bashrc ]; then
  . ~/.bashrc
fi

# Set variables for a warm fuzzy environment
export CVSROOT=~/.cvsroot
export EDITOR=/bin/vi
export PAGER=/usr/bin/less
```

Example .bashrc (partial)

```bash
# .bashrc
# abbreviations for some common commands
alias bye=logout
alias h=history
alias l='ls -F -C'
alias ll='ls -L -l -F'
alias po=popd
alias pu=pushd
```

Login and Other Shells

/etc/profile
~/.bash_profile
~/.bashrc
~/.bashrc

~/.bashrc
~/.bashrc
~/.bashrc

Interactive shell
~/.bashrc

Interactive shell
~/.bashrc

Interactive shell
~/.bashrc