

Shells

Shells are programs which provide a user interface - command line interpreter

1. Bourne shell (sh)
 - first UNIX shell
 - most widespread
 - generally associated with AT&T
2. “C” shell (csh)
 - Berkeley invention (Bill Joy)
 - widely used
 - write programs using “C”-like syntax
3. Korn shell (ksh)
 - default on vtaix
 - updated version of Bourne shell

Linux Shells

- bash - Bourne again shell
 - sh with advanced features
- tcsh - C shell
 - csh with advanced features
- ksh - Korn shell
- Others - ash, zsh

Variables/Environment

Variables

- environment variables exist in all shells
- set variables like this
 - ksh or sh
 - variableName = variableValue
 - export variableName
 - csh
 - setenv variableName
 - variableValue
 - set localVariableName = variableValue

Some important environment variables

- TERM
 - informs programs about your terminal
 - must have an entry in
 - /etc/termcap - file describing terminal capabilities
- DISPLAY
 - tells X programs where to display output

Setting Environment Variables

- ksh & sh
 - TERM=vt100
 - DISPLAY=myPC.lynchburg.edu:0.0
 - export DISPLAY
- csh
 - setenv TERM vt100
 - setenv DISPLAY myPC.lynchburg.edu:0.0

The Path Variable

- Lists directories that will be searched when a command is issued
- Csh
 - set path=(/usr/bin /usr/local/bin .)
- ksh & sh
 - path=/usr/bin:/usr/local/bin:

which, whence, whereis

- which or whence
 - gives full path for command
 - > which ls
/bin/ls
 - > whereis emacs
emacs: /usr/local/bin/emacs

Startup Files

- Bourne and Korn shells
 - /etc/profile - system-wide setup
 - .profile in home directory
- “C” shell
 - .cshrc
 - .login
 - .logout

Aliases

- **ksh**
 - alias name = [value]
 - alias ls = “ls -F”
 - alias dir = ls
 - alias cdt = “cd /home/cs201/proj/tests”
- **csh**
 - alias name [value]
 - alias ls ls -F
 - alias dir ls
 - alias cdt cd /home/cs201/proj/tests

Find command

`find <pathlist> <criteria>`

- `find . -name MyLostFile -print`
- `find / -name “*X11*R[56]” -print`
- `find / -name core -exec rm {} \;`
- `find . -size +100000c -exec \`
`ls -l {} \; > bigfiles`

History command

set history = 24

- !! re-execute last command
- !n re-execute command #n
- !-n re-execute current command
minus n
- !s re-execute last command
starting with s
- !!:1 !!:3 select words 1 and 3 from last
command
- !\$ last word in last command
- <command_ref>:s/old/new/
^old^new substitute *new* string for *old* in
previous command

History Examples

history

128 ls *.*[ch]

129 cd ~

130 pwd

find . -name "abe.*" -exec ls {} \;

^em^me

!128

!128:s/c/Cf/

Korn Shell

- `fc -l`
 - like history. Lists previous commands
- `fc -e - [old=new] line`
 - switch “-e -” means no editor
- `alias r=fc -e -`
- `r` commands
 - `r` re-execute last command
 - `r n` re-execute command #n
 - `r -n` re-execute current command number - n
 - `r cmd` re-execute command starting with cmd
 - `r old=new cmd` re-execute command with substitution

csH Job control

`^Z` suspends

`bg` moves job to background

`fg` moves job to foreground

`jobs` lists current jobs

`%n` resume nth job suspended

`nohup` prevents termination at hang-up

`nice` runs at altered priority

`&` at end of command runs

command in background