Shells

The UNIX Shells

- How shell works
  - Fetch command → Analyze → Execute

- Unix shells

Shell	Originator	System Name	Prompt
Bourne Shell	S. R. Bourne	/bin/sh	$
Csh	Bill Joy	/bin/csh	%
Tcsh	Ken Greer	/bin/tcsh	>
Korn Shell	David Korn	(shells/ksh93)	$
Z Shell	Paul Falstad	(shells/zsh)	%

Shell Startup Files

- sh
  - /etc/profile — login shell, system wide
  - ~/.profile — login shell
  - ENV
- csh
  - /etc/csh.cshrc — always, system wide
  - /etc/csh.login — login shell, system wide
  - ~/.cshrc — always
  - ~/.login — login shell
  - ~/.logout — logout shell
  - /etc/csh.logout — logout shell, system wide
- tcsh
  - ~/.tcshrc — login shell
- bash
  - /etc/profile → ~/.bash_profile or ~/.bash_login or ~/.profile
  - ~/.bashrc
  - BASH_ENV

Shell Environment Variables

- Controlling shell behaviors
  - There are many environment variables that control the shell behavior
- To dump them: env command
- To get value: $variable_name or ${variable_name}
- Useful Environment Variables

sh	csh	description
HOME		User's home directory
MAIL		User's mailbox
PATH		Search path
PS1	prompt	Primary prompt string (waiting for input commands)
PS2	prompt2	Secondary prompt string (after lines end with \)
	prompt3	Third prompt string (automatic spelling correction)
	history	Number of history commands

Variables and Strings Quotes

Char.	Purpose
var=value set var=value	Assign value to variable
$var ${var}	Get shell variable
`cmd`	Substitution stdout
'string'	Quote character without substitution
"string"	Quote character with substitution

- % varname=`/bin/date`
- % echo $varname
- % echo 'Now is $varname'
- % echo "Now is $varname"

- % set varname2=`/bin/date`
- % echo $varname2
- % echo 'Now is $varname2'
- % echo "Now is $varname2"

Mon Oct 24 19:42:49 CST 2011
Now is $varname
Now is Mon Oct 24 19:42:49 CST 2011

Global Variables

- Assignment

	Bourne Shell	C Shell
Local variable	my=test	set my=test
Global variable	export my	setenv my test

- Example:
  - $ export PAGER=/usr/bin/less
  - % setenv PAGER /usr/bin/less
  - $ current_month=`date +%m`
  - % set current_month =`date +%m`

- Use “env” command to display global variables

Shell Special Characters (1)

- Reduce typing as much as possible

Characters	Description
*	Match any string of characters
?	Match any single alphanumeric character
[...]	Match any single character within []
[!...]	Match any single character not in []
~	Home directory

- Example
  - If following files:

    test1 test2 test3 test4

    test-5 testmess

    are in current directory.

Command	Result
% ls test*	test1 test2 test3 test4 test-5 testmess
% ls test?	test1 test2 test3 test4
% ls test[123]	test1 test2 test3
% ls test[!345]*	test1 test2 test-5 testmess
% ls ~	List files under your home

Shell Special Characters (2)

Char.	Purpose	Example
#	Start a shell comment	# this is a comment
;	Command separator	% ls test*; ls test?
&&	executes the first command, and then executes the second if first command success (exit code=0)	% cd foo/bar && make install
||	executes the first command, and then executes the second if first command fail (exit code≠0)	% cp x y || touch y
\	(1) Escape character (2) Command continuation indicator	% touch test*; ls test* % ls \ > test*
&	Background execution	% make buildworld &

Built-in Shell Commands (1)

sh	csh	description
set/unset	set/unset	Set/Unset shell's parameters
	set/unset	Set/Unset a local variable
export	setenv/unsetenv	Set/Unset a global variable
set	@, set	Display or set shell variables
	login, logout	Logout
exit	exit	exit shell
cd	cd	change directory
	dirs	print directory stack
	popd, pushd	Pop/push directory stack
echo	echo	write arguments on stdout
alias/unalias	alias/unalias	command aliases
fg, bg	fg, bg	Bring a process to foreground/background

Built-in Shell Commands (2)

sh	csh	description
jobs	jobs	List active jobs
%[job no.]	%[job no.]	Bring a process to foreground
	kill	Send a signal to a job (%job | pid)
	stop	Suspend a background process (%job | pid)
exec	exec	execute arguments
	nice	Change nice value
	nohup	Ignore hangups
	notify	Notify user when jobs status changes
	history	Display history list
	rehash	Evaluate the internal hash table of the contents of directories
.	source	Read and execute a file

Built-in Shell Commands (3)

- References:
  - http://www.unet.univie.ac.at/aix/aixuser/usrosdev/list_bourne_builtin_cmds.htm
  - http://www.europa.idv.tw/UNIX-Shell/csh/V2-01-09.html
  - http://www.unix.org.ua/orelly/unix/unixnut/ch04_06.htm
  - http://publib.boulder.ibm.com/infocenter/pseries/index.jsp?topic=/com.ibm.aix.doc/aixuser/usrosdev/list_c_builtin_cmds.htm

  - sh(1)
  - tcsh(1)

Input/Output Redirection

- 3 default file descriptors
  - 0(stdin)、1(stdout)、2(stderr)

Method	Description
cmd < file	Open the file as stdin of cmd
cmd > file	Write stdout of cmd in the following file (noclubber)
cmd >> file	Append stdout of cmd to the following file
2>&1	Merge stdout with stderr
cmd1 | cmd2	Pipe stdout of cmd1 into stdin of cmd2

- "Redirection" in sh(1), or "Input/Output" in tcsh(1)

File and Directory Related Commands

Command	Purpose
ls	List a directory's content
pwd	Print working directory
mkdir	Make(create) a new directory
rmdir	Remove existing empty directory
cat	Concatenate file
cp	Copy file
ln	Link files
mv	Move file
rm	Remove file
split	Split a file into n line chunks
stat	Display file status

Select and File Processing Related Commands (1)

Command	Purpose
head	Display first lines of a file
tail	Select trailing lines
grep	Select lines
diff	Compare and select difference in two files
wc	Count characters, words or lines of a file
uniq	Select uniq lines
cut	Select columns
tr	Transform character
sort	Sort and merge multiple files together
join	Join two files, matching row by row
sed	Edit streams of data
awk	Pattern scanning and processing language

Select and File Processing Related Commands (2)

- Example usage:
  - Look first few lines or last few lines
    - % head /var/log/message
    - % tail /var/log/message
  - Find the occurrence of certain pattern in file
    - % grep −l liuyh *
      - Print the filename that has "liuyh" as content
  - Print the line number when using grep
    - % grep −n liuyh /etc/passwd
  - Ignore case-sensitive
    - % grep -i liuyh /etc/passwd
      - List any line contains any combination of "liuyh"
    - % ps auxww | grep ^liuyh | wc −l
      - Count number of processes owned by liuyh

Select and File Processing Related Commands (3)

- List liuyh's id, uid, home, shell in /etc/passwd
  - % grep liuyh /etc/passwd | cut –f1,3,6,7 –d:
    - liuyh:1002:/home/liuyh:/bin/tcsh
- Cut out file permission and file name from ls output
  - % ls -l | grep –v ^total | cut -c1-12 -c45-

    drwxr-xr-x GNUstep/

    drwx------ Mail/

    drwx------ News/
- Use awk to generate the same behavior of cut
  - % awk −F: '{print $1 " " $6}' /etc/passwd
    - nobody /nonexistent
    - liuyh /home/liuyh
  - % ls −al | grep −v ^total | awk '{print $1 " " $9}‘

    drwxr-xr-x GNUstep/

    drwx------ Mail/

    drwx------ News/

Select and File Processing Related Commands (4)

- sort (useful arguments: -r, -u, -k, -n)
  - **-n** (numeric keys sorting),
  - % ls −al | sort –k 5,5 −r
    - List directory contents and sort by file size decreasingly
  - % sort −t: -k 1,1 /etc/passwd | grep −v ^#
    - List records in /etc/passwd increasingly by id
  - % sort -t. -n -k 1,1 -k 2,2 -k 3,3 -k 4,4 /etc/hosts
    - List records in /etc/hosts sorted by IPv4 address
- tr – Translate characters
  - % tr "A-Z" "a-z" < file1 > file2
  - % grep liuyh /etc/passwd | tr ":" "\n"
  - % tr −d "\t" < file1
    - Delete tab in file1
  - % tr −s " " " " < file1
    - Delete multiple space in file1

xargs Command

- xargs – construct argument list(s) and execute utility

  -n number

  -I replstr

  -J replstr

  -s size

  …

```
% ls
2.sh      3.csh     4.csh     4.sh      bsd1.ping   testin
% ls | xargs echo
2.sh 3.csh 4.csh 4.sh bsd1.ping testin
% ls | xargs -n1 echo
2.sh
3.csh
4.csh
4.sh
bsd1.ping
testin
```

```
% ls | xargs -I % -n1 echo % here %
2.sh here 2.sh
3.csh here 3.csh
4.csh here 4.csh
4.sh here 4.sh
bsd1.ping here bsd1.ping
testin here testin
```

```
% ls | xargs -J % -n1 echo % here %
2.sh here %
3.csh here %
4.csh here %
4.sh here %
bsd1.ping here %
testin here %
```

The Unix Way

- Lots of little tools, each good at one thing
  - Use them together to achieve your goal
- Example
  - Quest: To get all cs98 student id/account/cname/ename
  - Hints

    All user home dir are created by his/her student id.

    User command can get some useful info.

    % user liuyh

    username: liuyh studentID: 9755806 劉用翔 Yung-Hsiang Liu
  - Approach
    - % cd /u/cs/98
    - % ls # you will get all cs98 student id
    - % ls | xargs -n 1 # print student id each in one line
    - % ls | xargs -n 1 user # get data you want
    - % ls | xargs -n 1 user | awk '{print $4" "$2" "$5" "$6}'
      # format the data to get the result

Appendix

Command History in (t)csh

Command History in (t)csh

- !n - exec previous command line n
- !-n - exec current command line minus n
- !! - exec last command (the same as !-1)
- !str - exec previous command line beginning with str
- !?str? - exec previous command line containing str

```
% history
9   8:30        nroff –man ypwhich.1
10  8:31        cp ypwhich.1 ypwhich.1.old
11  8:31        vi ypwhich.1
12  8:32        diff ypwhich.1.old ypwhich.1
13  8:32        history
% !?old?
```

Command History in (t)csh

- !!:n - use the nth word of previous command
- !!:m-n - select words m ~ n of previous command
- !!:* - use all arguments of previous command
- !!:s/str1/str2/ - substitute str1 with str2 in previous command

```
% history
15  8:35        cd /etc
16  8:35        ls HOSTS FSTAB
17  8:35        history
% cat !-2:*:s/HOSTS/hosts/:s/FSTAB/fstab
```

- "History Substitution" in tcsh(1)