II. More commands, Editor and Environment

Wei, Fu-Jin

July 8th, 2019

2019 GS summer camp

Your own history !

- Your own history file : (for bash)

```
$ ~/.bash_history
```

- Display your commands history

```
$ less ~/.bash_history
$ history | less
```

- Re-use command
  - repeat last command

```
$ !!
$ !-1
```

  - repeat #10 command in the history

```
$ !10
```

Where is my command or file?!

- which - Display the full path of the default command

```
$ which blastn
```

- whereis - Display the path of command and it's manual

```
$ whereis ls
```

- locate - locate command path, but need pre-constructed database and authorization

```
$ locate sort
```

Find something - find

- find – find any path match your condition and more~
  - Match to the pattern

```
$ find /usr/local -name “*blast*”
```

  - Find the target and do something on it or for it

```
$ find /usr/local -name “*blast*” –exec ls -l ‘{}’ ‘;’
```

  - Find an executable file

```
$ touch doSomething ; chmod +x doSomething
$ find ./ -type f –executable
```

  - Find an accessible directory

```
$ mkdir goDir
$ find ./ -type d -executable
```

  - Find the files those were modified 5 days ago

```
$ touch –d “2019-07-02 14:05” 5days
$ find ./ -mtime +5
```

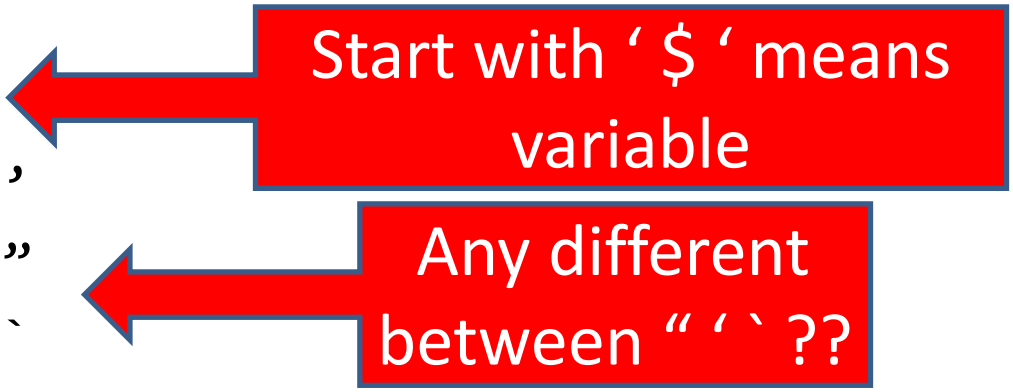
  - Find the files those were modified 5 to 10 days ago

```
$ touch –d “2019-06-19 13:05” 10days
$ find ./ -mtime +5
$ find ./ -mtime +5 –mtime -10
```

Simply print out

- echo : display a string to stand out

```
$ echo HOME
$ echo $HOME
$ echo ‘ls $HOME’
$ echo “ls $HOME”
$ echo `ls $HOME`
```



- cat (tac)

```
$ cat /etc/passwd
$ tac /etc/passwd
```

I/O stream

- `>`: output to or create a file.

```
$ ls -l /usr > ls_usr.txt
$ ls -l ls_usr.txt
```

Please check the file size !!

```
$ ls -l /usr >> ls_usr.txt
$ ls -l ls_usr.txt
```

Please check the file size, again.
Is there any different?)

- `<` : input from right side

```
$  cat < ls_usr.txt
$  tac < ls_usr.txt
```

| vs `>`

- | “pipe”

```
$ ls /usr/sbin | less
```

- head or tail

```
$ head -n 5 ls_bin.txt
$ tail -n 5 ls_bin.txt
```

- more or less

```
$ more /etc/passwd
$ less /etc/passwd
```

Partial, merge or different way to read files

- cut – display choiced bytes, columns

```
$ cat /etc/passwd | cut -d “:” -f 1,3,5,6
    > name_des_home.txt
$ cat /etc/passwd | cut -d “:” -f 1,7
  > name_shell.txt
```

- paste – join files in tabular separated columns

```
$ paste name_des_home.txt name_shell.txt
  > name_des_home_shell.txt
```

- file - determine file type

```
$ file ./*
```

- od – translate to different coding system

```
$ head -n 1 name_des_home_shell.txt
$ head -n 1 name_des_home_shell.txt | od –a
```

shell command (2.4)

- grep – display pattern matched line

```
$ grep 1002 /etc/passwd
$ grep -c 1002 /etc/passwd
$ grep -n 1002 /etc/passwd
```

- wc – calculate characters, words, lines number

```
$ wc /etc/passwd
$ wc -l /etc/passwd
```

- sort - sort lines of text files

```
$ sort -t”:” -k 3,3n /etc/passwd
$ sort -t”:” -k 3,3nr /etc/passwd | head
$ sort -t”:” -k 1,1r /etc/group
```

Standard I/O

- stdin : Standard input. Use to get input (keyboard)
  i.e. data going into a program.

```
$ ls -l /usr/sbin | sort –k 5nr | head –n 10
```

- stdout : Standard output. Use to write information (screen)

```
$ head name_des_home.txt > TenLine.txt
```

- `tee – branch pipe to a file`

```
$ head name_des_home_shell.txt | tee TenList.txt
```

The Unix / Linux standard I/O streams:

Handle	Name	Description
0	stdin	Standard input
1	stdout	Standard output
2	stderr	Standard error

Standard Error !!

- stderr : Standard error. Use to write error message (screen)

```
$ head name_des_home.txt nosuchfile.txt
$ head name_des_home.txt nosuchfile.txt > TenLine.txt
$ head name_des_home.txt nosuchfile.txt > TenLine.txt \
  2>lst_bin.log
$ head name_des_home.txt nosuchfile.txt > lst_bin.log \
  2>&1
$ head name_des_home.txt nosuchfile.txt &> lst_bin.log
```

Environment Variables

- Hierarchical inherit variables
  - /etc/profile, /etc/bash_profile, ~/.bash_profile, ~/.bashrc
- env
  - Display present environment variables
- export
  - Setting shell environment variables

```
$ export testV="What a nice day"
$ echo $testV
$ echo $PATH
```

Users and Groups

- What is your group belong to ?
  - Use command : groups
  - Check the configuration files

/etc/passwd

1. login name
2. optional encrypted password
3. numerical user ID
4. numerical group ID
5. user name or comment field
6. user home directory
7. optional user command interpreter

/etc/group

1. group_name
2. password
3. GID
4. user_list

Text Editor - vim

- vim
  - command mode
    - [y(n)] or [yy] : copy to clipboard
    - [p] : paste from clipboard
    - [x] : delete a character
    - [dw | dd] : delete a word or a line
  - change to edit mode
    - [a]:append text
    - [i]:insert text
    - [o]:append a line (row)
    - [O]:insert a line (row) before
  - command script mode
    - [:]: 命令開始
    - [:w]: 存檔
    - [:q]: 退出vim
    - [:wq]: 存檔並退出vim
    - [:q!]: 不存檔, 強制退出 vim

Edit a texture file

- Please edit a file named “.ncbirc” at your home

```
$ cd ; vi .ncbirc
```

```
[BLAST]
BLASTDB=/sharedata/blastdb
```