UNIX Commands Used in SVU

* Check the SVU home page for information and support
* Read the email send to you carefully
* Check your disk quota and take care of your files in common working directory
* Send your enquiries to ccesvuhelp@nus.edu.sg
* Share an account is not allowed.

1) Login and manage your account
   a. Remote login: ssh, e-svu portal or through MobileVisLab
   b. Always change your password on cougar1.nus.edu.sg: passwd
   c. Check your disk quota: quota -v
   d. ~ = /home/enduser/<UID>
   e. Common working directories:
                /svutmp (local directory for each hosts)
                /lsftmp (global common directories)
   f. Grace period for files in the common working directory
      /home/enduser/<UID>: permanent (daily backup)
      /lsftmp: 30 days (daily backup)
   g. file system in the hosts: df -k
   h. Disk usage for certain directory: du -s -k

2) On line reference manual:
   man -k <KEYWORDS>
   apropos <KEYWORDS>
   man <COMMAND>
   whatis: describe what a command is

3) ls: list contents of directory
     -l List in long format
     -a List all entries
     -t Sort by time stamp instead of by name.
     -F Put a slash (/) asterisk (*) an equal sign (=) (@)
     -i list the inode number

     * you can have sevral options at the same time!

4) Diplay the content of file

     more, page: browse or page through a text file
     less: the opposite of more.

     * In more, pg and less, you can search a pattern or forward or backward
     to a certain pattern using / or ?.

     cat: concaternate and print file.
     jot: mouse-base text editor (only on SGI)
     vi(-R), view: screen oriented display editor.
     pico:simple text editor in the style of Pine composer (Sun, SGI, Compaq)
     tail -n (from the end) +n (from the beginning) (only one file)
     head -n (with more than 3 files here)
     zmore: print compressed files, uncompressed, one screen at a time.
          similar to more, print compressed files(.Z or .zip) uncompressed,
one
          screen at a time.
     zcat: display compressed data

5) Directory management
mkdir: make a new directory
rmdir: remove directory
rm -r: recursively remove directory and the files inside
cd <Enter>: to user's home directory
cd .. : to one directory-level higher in the directory tree
cd / : to root directory
cd DIR1 to directory DIR1
cd /svutmp/ccezxh/demo/insight : to the "insight" directory through an absolute path
pwd: show path of the current directory

6) Copy, move, remove files or directories
mv -i file1 file2
rm file (wildcat)
-i turn interactive promoting (applicable to cp and mv)
-f force to remove files
cp file1 file2
-i interactive
-f force
-r copy directories recursively
-R recursively
-P keep the date of the file

7) Change the attribute of files and directories
chmod ugo+rwx filename
chmod ugo-rwx filename
chown user:group filename
chgrp group filename

Recursivlely change permission:
chmod -R u+rwx dir

8) Metacharacters & Special Characters:
? one character except black and tab
* zero or more character
[] a range of characters
$ used to print the value of a variables
~ abbreviation for a user's home directory

9) Redirect your output
< > redirection operator
| pipe operator
>> to append
>& redirect the error message into the standard output
>! to overwrite the output file

10) Process monitor and job control
Ctrl Z (suspend the current jobs)
fg job_id: run a job in forground
bg job_id: run a job in background
top: most CPU-intensive processes currently running
ps -ef |grep <STRING>: to check the status of a process
kill -9 <PID>: kill a process
time <COMMAND>: time a command execution

11) Set remote display:
    a) on local host, type: xhost + remote_host
    b) after login to the remote host, type: setenv DISPLAY local_host_ip:0.0

12) link files and directory
  hard link: ln file1 (path)file2 (have the same inode, in same file system)
  soft link: ln -s file1 (path)file2
  list link information: ls -il
  first character is l for softlink
  link count for hard link is 2, soft link is 1

13) Find file in a directory
  find path-name-list options expression
    -name True if file matches the current filename
    -user uname True if the file belongs to the user uname.
    -print Always true; causes the current pathname to be printed.
    -depth Always true; causes descent of the directory hierarchy
        to be done so that all entries in a directory are acted
        on before the directory itself.
    -local True if the file physically resides on the local system;
        causes the search not to descend into remotely mounted
        filesystems.
    -mount Always true; restricts the search to the filesystem
        containing the current element of the path-name-list.

15) Repeat command and history
     in the .cshrc file
     set history=100
     set savehist=100
     The history will be stored in .history
     history [n]
     !comm: to repeat the last "commandx"
     !! : to repeat the very last command
      !-4 (offset) : to repeat the last but 4 command
     !m repeat command with number m as show in the history list

16) Identify
     who: check who is login now.
     w: who is on and what they are doing.
     id: print the user name and ID, and group name and ID
     whoami: display the effective current username
     hostname: print name of current host system
     which: locate a program file including aliases and path (csh only)
      finger: user information lookup

17) ftp file transfer program
    ftp
    mode: bi(nary) ascii
    general command: cd mkdir ls pwd rm rmdir (type help for more)
    transfer file: get mget put mput
    (before mass tranfer,type prompt to turn off interactive mode off)

    Transfer files between pc and svu servers
    SSH Secure File Transfer (web-shopping SSH Client)

18) Diff compare two files and display a list of the difference.
    diff file1 file2

xdiff file1 file2 : compare two files in a x-window

19) Compress and uncompress files
compress, uncompress
gizp, gunzip, zcat, gzcat
tar -cvf (create)
tar -tvf (table)list the tarred files in the tar file
tar -xvf (extract) files
* don't specify the absolute directory or use . to represent the current directory.

20) Capture and view the pictures in Unix:
ghostview
xpsview
xv
snapshot: Alt+f7-->Ctrl+leftbutton-->drag-->rightbutton-->save

21) Unix utilities
date: show date and time
cal: calendar
calc: calculator(SGI)
dc, xcalc (Compaq, HP)

22) Search a pattern
grep 'this pattern' filename (directory)
-l print only the name of the file with matching lines once, rated by new line.
-n procedes each line by its line number in the file.
-s supresses error message. work silently., display nothing
-c output only the count of the lines with the pattern.
-v output the lines that don't contain the pattern.
-i ignore uppercase/lowcase distinction during comparison.

23)tar: tape archive with DAT tape
mt -f /dev/tape
tar tvf /dev/tape
tar cvf /dev/tape
tar xvf /dev/tape

24) Editor
pico (all Linux systems)
emacs(All servers)
vi (All servers)

25) Jobs submitted through LSF:
a. Some commands for lsf
bsub: to submit jobs to lsf
bjobs: to check your jobs
bqueues: to check the queues (-m hostname, -l queuename)
bhist: to chech the jobs execution history
bkill(bdel): to kill a particulat job
bhosts: to check which host is among the lsf (-l hostname)
bpeek: to check the output
bmod: to modify the jobs
bhpart: to show current cumulative CPU usage for each user
lsload: displays load information for hosts
lsmon: displays load information for LSF hosts

b. Some examples for submitting jobs

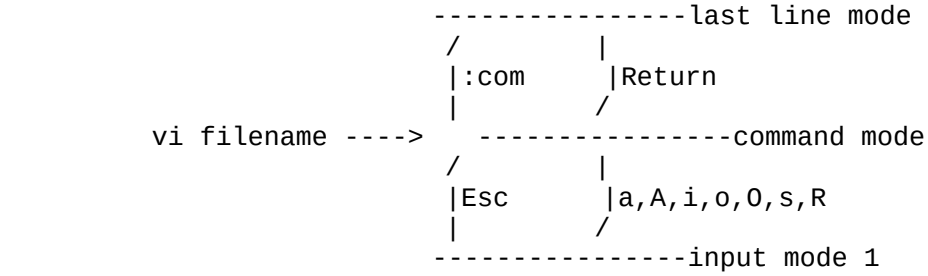
```
* bsub -o test1.eo -q linux -m cheetah3 ./exe_file
* bsub -o test1.eo -q linux < exe_file
* bsub -q linux < exe_file > out_file
* bsub -q sun -m callisto -r ./exe_file
* bsub < script_file
```

scriptfile example:

```
#!/bin/tcsh
#BSUB -q cpq
#BSUB -o error.eo
#BSUB -u ccezxh@nus.edu.sg
#BSUb -m em64t-compute
#BSUB -c cputime  #i.e. 3:30: three and half hours
exe_file < inputfile > outfile
```

vi editor

vi file
vi +n file (start editing from nth line)
vi +/Pattern file (start editing from Pattern)
in .exrc file: set showmode aoutoindent, etc.,



TRY TO WRITE THE MODIFICATION FROM TIME TO TIME

a. Command Mode

The basic editing operators are:

```
c  begin a change
d  begin a delete
y  begin a yank(or copy)
p  put the line somewhere
x  delete one character
G  jump to the end of file
r  replace the current character
u  undo the last command
.  redo the last command
```

Some representatives

```
w  word
```

() sentense
{} paragraph
[] section [[very beginning,]] very end
. current line
$ end of line or a paragraph or the whole text
0 beginning

Movement of the cursor:
h, j(+), k(-, enter), l(spacebar), left, down, up, right
b w: move backward and forward of a word
H L M: highest, lowest and middle of the screen
Ctrl-f Ctrl-b forward and backward one full screen
Ctrl-u Ctrl-d up and down half of a screen
z enter, z. z- put the current line the top middle and bottom line
/pattern/z. 35z. line contain pattern or line 35 put in the middle
nG put to line n, G end of the file , last line is the default.
cG to change to the end of the file.
dG delete the rest of the file from the current line.

/pattern search the pattern
/\<word\>/ search only the exact the word
/\<word/ search including pattern start with word
/word\>/ search including pattern end with word
d/pattern delete from the current position to the pattern

% command will locate a corresponding openning or closing
parenthesis, bracket or braces.

The command m followed by a letter marks that line with the
letter. A single quote placed before the letter reference
the line. You can use a mark in any command in place of the
line number.
ma mark the line with a
'ad delete from current line to the marked line

Single quota also used to reference the previous position.

dd delete one line 3dd
dw delete a word
d{ delete to the beginning of the paragraph d(
d} delete up to next paragraph d)
d^ delete back to beginning of the line
d/pat delete up to the first occurance of the line
dL delete up to the last line
dG delete to the end of the file cG
d'(mark) delete to the marked line) backword or forward
X delete the character before the cursor
x delete the character after the cursor
D delete all text from cursor to the end of the line: d$

yy (equals to Y)copy the line to a buffer
vi has 9 default buffers, each one has a number with the
latest has the lowerst number, a double quota with the
number represents the paticular buffer
vi can also have 26 named buffers, represent by 26 letters
yw copy a word
y) copy a sentence

y} copy a paragraph

p put the buffer after the current line (P before)
"f7yy-----> "fp : 7 lines put into buffer f
"m9yy-----> "mp : 9 lines put into buffer m

/exp search expresion forward
?exp search expresion backward

u undo the last command
U undo all the change to the current line.

b. Input command
a Append after cursor
A Append at the end of the line (equals $a)
i insert before cursor
I insert at the beginning of the line (equals 0I)
o open a line below the current line
O open a line above the current line
r replace one character
R begin overwrite text
s substitute a character
S substiture the entire line (equals cc)
c change text
C change the reminder of the line

c. Last line command
:m1, m2d<ret> delete from line m1 to line m2
:,m1d<ret> delete from current line to line m1
:.,$d<ret> delete from currunt lime to the end of the file
:q quit
:q! quit without saving
:w write the change
:w filename: write the buffer to a file
:w >>filename: write the buffer to append to a file
:w! force write
:wq write and quit
:r file1<ret> read file1 to the cursor position
:set nu: setting the numbering of lines
:set nonu: turn of the numbering of the lines
:set wm=70 (set the word wrap margin to the 70th column)
:[address]s/search_string/replace_string/[g]: search and substitute
:m1,m2s/t1/t2/g (. current $ last)
:s/^t1/t2/g
:g/R/s//r/g (g flag means do it globally, make the substitution if it occurs more than once on the same line)(R: to be replaced, r: replaced string)