Introduction to Linux commands and Vim

Shuyue Hu

syhu@cse.cuhk.edu.hk

CSCI2100 Data Structures Tutorial 3

Basic Linux command

Command	Description
pwd	Display the pathname for the current directory.
ls [options]	List directory contents.
cd directorypath	Change to directory. cd .. to back to the parent directory
cp [options] source destination	Copy files and directories.
rm [options] directory	Remove (delete) file(s) and/or directories. rm -r directory to remove non-empty directory
mkdir [options] directory	Create a new directory.
rmdir [options] directory	Delete empty directories.
mv [options] source destination	Rename or move file(s) or directories.
man [command]	Display the help information for the specified command.
cat [filename]	Display file's contents to the standard output device (usually your monitor).
less [options] [filename]	View the contents of a file one page at a time.

Basic Linux command: cd

- Syntax: cd [*directory*]
- Change the directory/folder of the terminal's shell.
- Change to root directory:
- `$ cd /`
- Change to parent directory:
- `$ cd ..`
- Change to subdirectory *Documents*:
- `$ cd Documents`

Basic Linux command: cd

- Change to directory with absolute path */home/user/Desktop*:
- `$ cd /home/user/Desktop`
- Change to directory name with white space - *My Books*:
- `$ cd My\ Books`
- `$ cd "My Books“`
- `$ cd 'My Books'`

Basic Linux command: ls

- Syntax: ls [*options*] [*file*|*dir*]
- List information about the FILEs (the current directory by default).
- List directory *Doc/Books* with *relative* path:
- `$ ls Doc/Books`
- List directory */home/Doc/Books* with *absolute* path:
- `$ ls /home/Doc/Books`

Basic Linux command: ls

- Sort by date/time:
- `$ ls -t`
- Sort by file size:
- `$ ls -S`
- Recursive directory tree list:
- `$ ls -R`
- List only text files using wildcard:
- `$ ls *.txt`

Basic Linux command: cp

- Syntax: cp [*options*] *source dest*
- Copy files and directories.
- Copy single file *main.c* to destination directory *bak*:
- `$ cp main.c bak`
- Copy 2 files *main.c* and *def.h* to destination directory *bak*:
- `$ cp main.c def.h bak`

Basic Linux command: cp

- Copy all C files in current directory to subdirectory *bak*:
- `$ cp *.c bak`
- Copy directory *src* to absolute path directory */home/usr/* :
- `$ cp src /home/usr/`

Basic Linux command: mv

- Syntax: mv [*options*] *source dest*
- Move files and directories.
- Move *main.c def.h* files to */home/usr/* directory:
- `$ mv main.c def.h /home/usr/`
- Move all C files in current directory to subdirectory *bak* :
- `$ mv *.c bak`
- Rename file *main.c* to *main.bak*:
- `$ mv main.c main.bak`

Basic Linux command: rm

- Syntax: rm [*options*] [*file*|*directory*]
- Remove files and directories.
- Remove *main.c* file in the current directory:
- `$ rm main.c`
- Remove all C files in the current directory:
- `$ rm *.c`
- Remove the directory *mydirectory* and any files and directories it contains. :
- `$ rm –rf mydirectory`

Basic Linux command: cat

- Syntax: cat [*options*] *file1* [*file2*...]
- Display the content of text files and to combine several files to one file.
- View text file data:
- `$ cat list1.txt`
- `milk`
  `bread`

Basic Linux command: cat

- `$ cat list2.txt`
- `house`

- Combine 2 text files:
- `$ cat list1.txt list2.txt`
- `milk`
  `apples`

  `house`

Basic Linux command: cat

- Combine 2 text files to another file:
- `$ cat list1.txt list2.txt > todo.txt`

Basic Linux command: man

- Syntax: man [*options*] *keyword*
- Man is the interface used to view the system's reference manuals. The *keyword* is the exact name of the command or other item for which information is desired.
- `$ man ls`
- `$ man man`

Online Basic Linux Commands tutorial

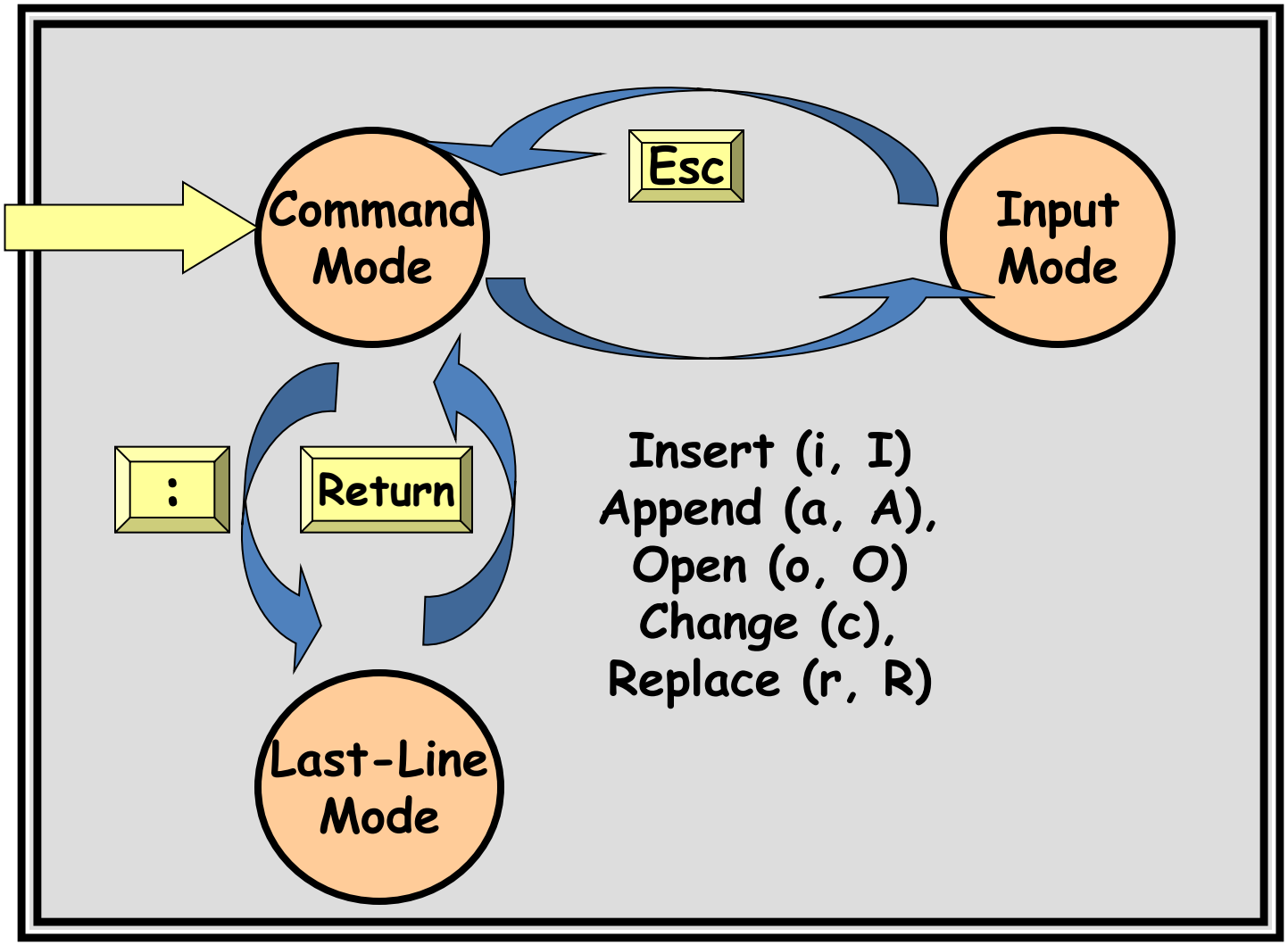
- https://youtu.be/IVquJh3DXUA
- http://www.dummies.com/how-to/content/common-linux-commands.html

The vi Editor

- short for: visual editor
- available on all UNIX systems
  - original vi part of BSD Unix
    - written by Bill Joy in 1976
  - many derived, improved versions available
  - open source vim (vi improved)

    is part of GNU/Linux
- vi has multiple modes of operation:
  - input mode, command mode, last-line mode

vi Editing modes



VIM TUTORIAL

- Never start something you can't exit
  - To end vi tutorial in the middle of the session, execute the command :q!
  - :q! = quit without saving
  - :wq = write out (save) and quit
- F1 = help
  - or :help
  - :help <command>
  - :q to exit help window

COMMANDS

- Delete characters
  - x deletes character under the cursor
- Insert characters
  - i converts to insert mode
  - then type characters
  - <esc> to exit insert mode
- Motion in command mode:
  - h,j,k,l: left,up,down,right
  - 0,$: move to begin/end of current line

COMMANDS

- Insert lines
  - o = open line below cursor
  - O = open line above cursor
  - <esc> to exit insert mode
- Append characters
  - A converts to insert mode at end of a line
  - then type characters
  - <esc> to exit insert mode

COMMANDS

- Deletion
  - d$ deletes to end of line
  - dw deletes to beginning of next word
  - de deletes to end of current word
  - d + motion
- Using motions for movement
  - Use any of the motions above
  - Use count for repetition
  - 2w = move cursor two words forward
  - 0 = start of line

COMMANDS

- Using repetition as part of deletion
  - 2dw deletes next two words
- Deleting a line
  - dd = delete line
  - 2dd = delete two lines
- Undo
  - u = undo one command
  - U = restore a line
  - ctrl-R = redo a command

COMMANDS

- p = put back the deleted text (in new place)
  - one of the delete command above + put = cut-and-paste
- More general cut-and-paste
  - v = start visual mode (start block)
  - move cursor to end of block
  - y = yank (copy to buffer)
  - then p = put in new place

COMMANDS

- Location
  - ctrl-g = show position in file
  - G = go to bottom of file
  - gg = go to top of file
  - <number>g = go to line <number>

COMMANDS

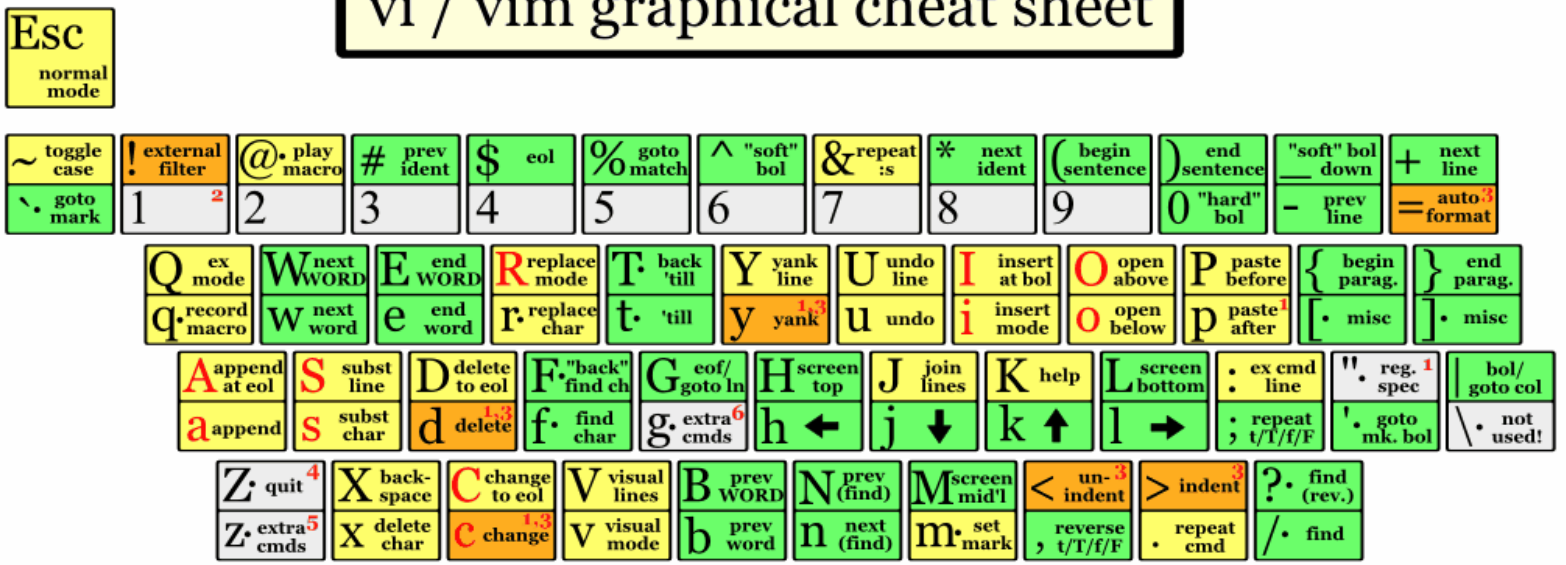
- Search
  - /<phrase> = search
  - /<phrase>\c = ignore case
  - ?<phrase> = search backwards
  - n = repeat search
  - N = repeat search in the other direction
- Search for matching parentheses
  - Put cursor on (, [or {
  - % = go to matching one
  - % = go to first one again

COMMANDS

- Substitute (replace)
  - :s/thee/the = changes first one
  - :s/thee/the/g = changes all (global change)
  - :s/thee/the/gc = change all with query
  - :#,#/thee/the/g = only change within that line range

version 1.1
April 1st, 06

vi / vim graphical cheat sheet



motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line,
mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDs: `quux(foo, bar, baz);`

Main command line commands ('ex'):

:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:%s/x/y/g (replace 'x' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important commands:

CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

(1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay$ to copy rest of line to reg 'a')

(2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)

(3) duplicate operator to act on current line (dd = delete line, >> = indent line)

(4) ZZ to save & quit, ZQ to quit w/o saving

(5) zt: scroll cursor to top, zb: bottom, zz: center

(6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to **www.viemu.com** - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Customizing vim

- Modify the ~/.vimrc file
- Some common syntax
  - set nu
  - set syntax=on
  - set history=1000

Vim plugins

- Many online resources
- Great color scheme, highlight keywords, etc.
- https://github.com/amix/vimrc

Vim Demonstration

Online vim tutorial

- http://www.openvim.com/
- http://tips.webdesign10.com/another-vim-tutorial
- https://www.youtube.com/watch?v=71YTkxUNwmg

gcc

- gcc is the C compiler developed by GNU project
- Widely adopted as the default compiler of UNIX-like systems

gcc

- hello.c

```
#include <stdio.h>
int main()
{
    printf("Hello, world!\n");
    return 0;
}
```

- Compiling hello.c into an executable file called “hello" is:
  - gcc hello.c -o hello
- Execute the executable file “hello”:
  - ./hello

gcc

- gcc also provides options that help you to optimize or debug your code
- Compile your code with debugging information:
  - gcc -g -o garbage garbage.c
- For other optimization/debug options, you may need to check the manual:
  - man gcc

References

- www.faculty.cs.niu.edu/~freedman/330/editors.ppt
- http://www.dummies.com/how-to/content/common-linux-commands.html
- http://cseweb.ucsd.edu/classes/wi11/cse141/tutorial_gcc_gdb.html