Text Editors

Chonnam National University
School of Electronics and Computer Engineering

Kyungbaek Kim

Bunch of Editors

- Line-based text editors
  - sed
    - Stream editor. According to a command with arguments, the entire file contents are modified.
- Console-based text editors
  - vim
    - **V**i **IM**proved
    - Improved version of “vi” editor
  - Emacs
    - Also has GUI-version
  - Nano
- GUI-based text editors
  - gedit
    - A GUI-text editor for GNOME
  - kedit
    - A GUI-text editor for KDE

Why do we need to learn text based editors?

- Sometimes, it is the only available editor
  - Maintenance and recovery operations almost never take place during X Window sessions.
  - When any bad thing happens, you can not log into the X Window system
- You can edit text files through remote shell session
  - Some servers will not host graphical desktops.
- Mouse movements slow down the touch-typist

VI and VIM

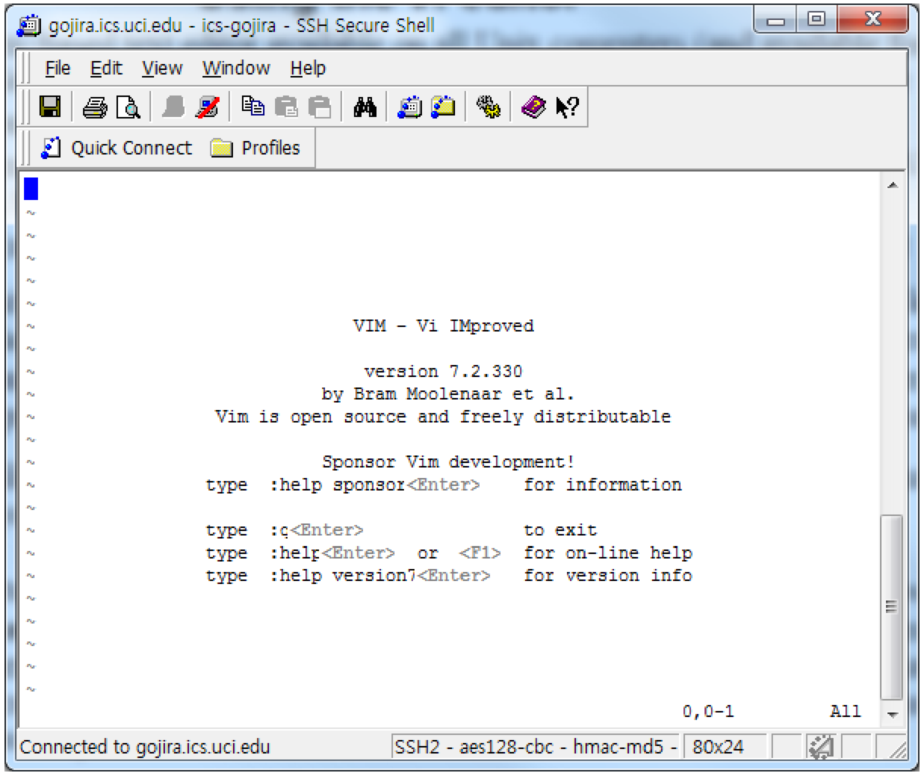
- A console-based text editor
  - Available on all UNIX/LINUX computers
  - Vim is usually aliased as vi in a shell
- How to start an editing session?

```
$ vi file.txt
```

  - If there is the file, it is opened based on the permission of the file
  - If there is not the file, virtually opened first
    - Later the file can be created when vi saves the file

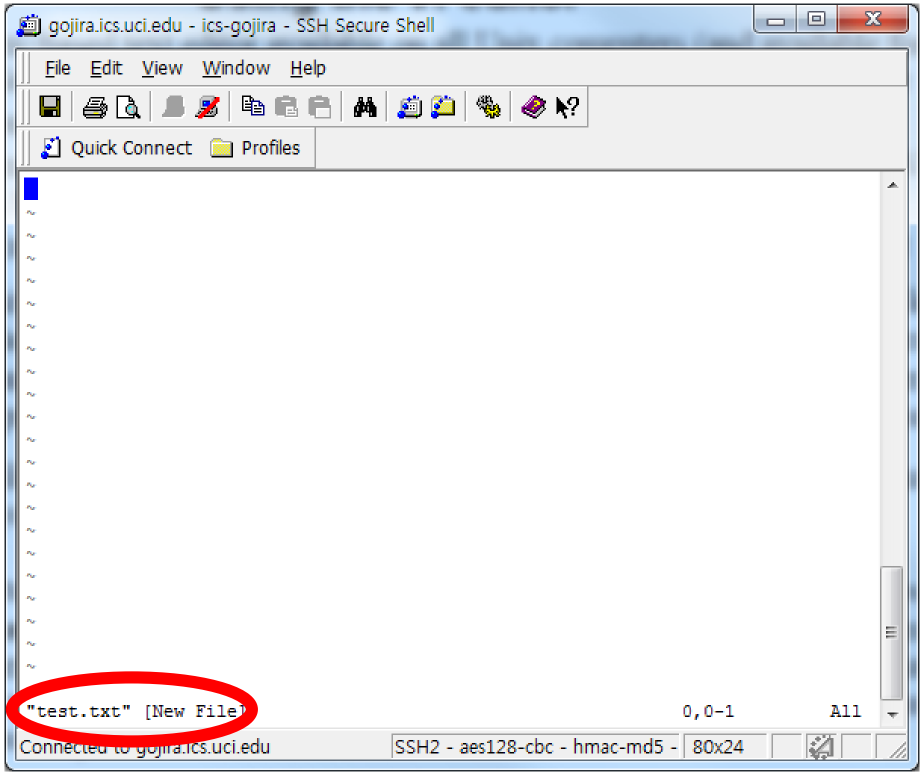
Example

- –R option : for read only



```
VIM - Vi IMproved

version 7.2.330
by Bram Moolenaar et al.
Vim is open source and freely distributable

Sponsor Vim development!
type  :help sponsor<Enter>    for information

type  :q<Enter>               to exit
type  :help<Enter>  or  <F1>  for on-line help
type  :help version7<Enter>   for version info

                                        0,0-1         All
```

gojira.ics.uci.edu - ics-gojira - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles
Connected to gojira.ics.uci.edu
SSH2 - aes128-cbc - hmac-md5 -
80x24



```
"test.txt" [New File]                   0,0-1         All
```

gojira.ics.uci.edu - ics-gojira - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles
Connected to gojira.ics.uci.edu
SSH2 - aes128-cbc - hmac-md5 -
80x24

Without filename :
Empty view

Filename :
view with the filename

Other ways to open

- vi + filename
  - Place the cursor on last line of file
- vi +n filename
  - Place the cursor on line "n" of file.
- vi +/pat filename
  - Place cursor on line with first occurrence of "pat"tern

Getting out from VI

- You need to save the file or exit from VI
  - First, switch to command mode
  - Then there are many options (in line mode)
    - :w filename
      - To save the edited file to a new file “filename” and quit
    - :w!
      - Write the file to disk even if read/only
    - :q
      - To quit VI, only if the file is already saved
    - :wq
      - To save the edited file and quit
    - :q!
      - To quit VI without any saving changes
    - ZZ
      - Same command to “:wq”

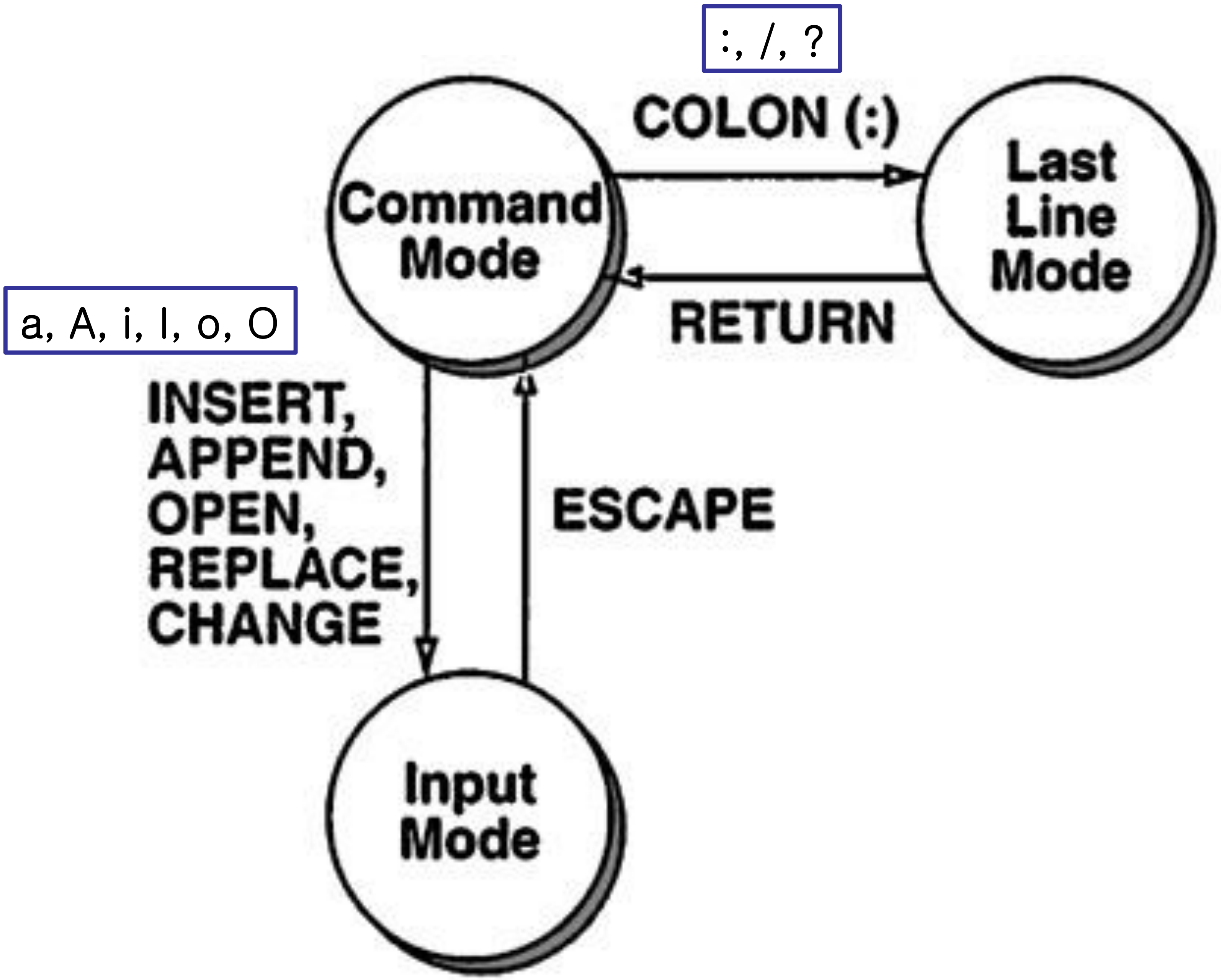
Recovery of file

- Sometimes, vi is interrupted and could not save the file properly
- Swap file
  - vi stored the things you changed in a swap file
  - A hidden file starting with "." in the same directory where the edited file locates
- With "–r" option
  - Case sensitive
    - C.f.) –R option : read only

vi –r → list all the files which can be recovered
vi –r filename → open vi and recover the previous contents

Three modes in VI

- VI has three modes
  - Insert (editing) mode
    - Editing texts
  - Command (or viewing) mode
    - Performing special functions
    - Initial mode
  - Line mode
    - Special mode to execute more complicated functions
- How to switch.
  - To switch to insert mode:
    - Press "i" or "a" or "o" or "I" or "A" or "O"
  - To switch to command mode:
    - Press ESC key
  - To switch to line mode :
    - In command Mode, press ":" or "/" or "?"


:, /, ?
COLON (:)
Command Mode
Last Line Mode
RETURN
a, A, i, I, o, O
INSERT, APPEND, OPEN, REPLACE, CHANGE
ESCAPE
Input Mode

Cursor and Moving around

- There is a blinked cursor
- In command mode, you can move the cursor
  - By using arrow keys
  - Or other keystrokes

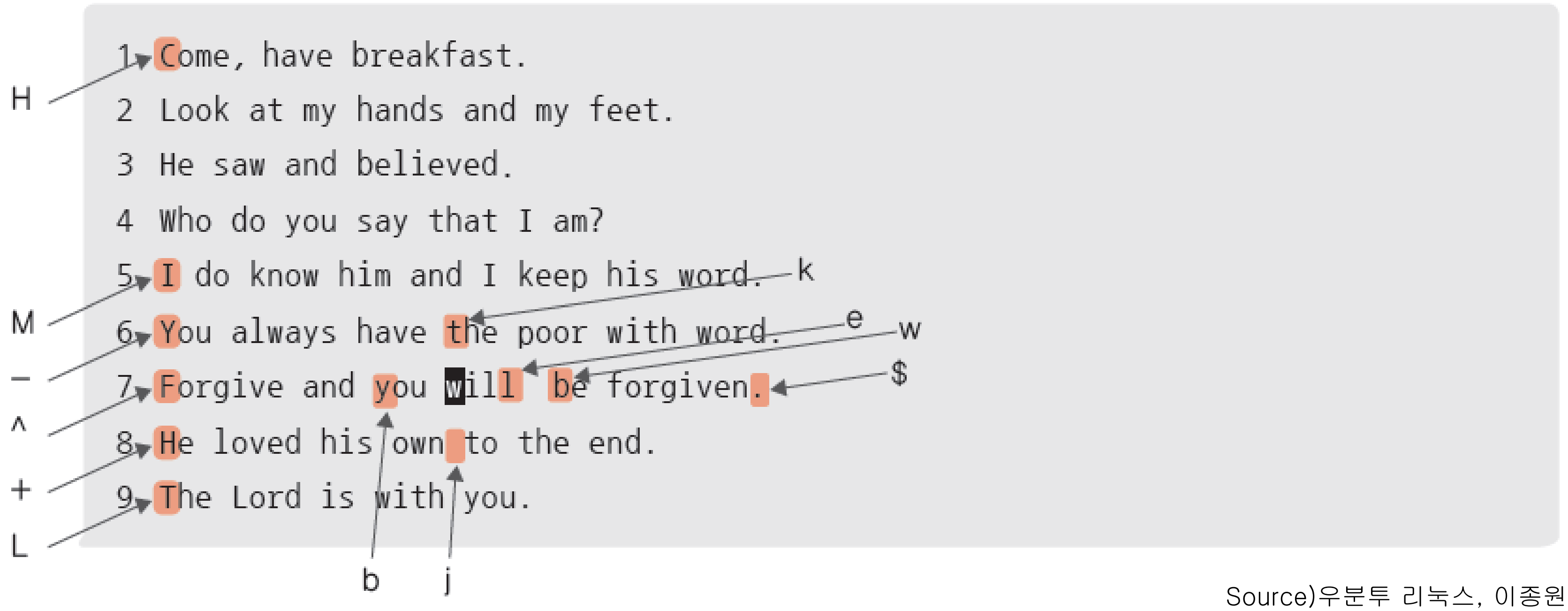
G → to the last line
1G → to the first line
17G → to line #17

h → left one character
l → right one character
k → up one line
j → down one line
b → back one word
w → forward one word

H → first line of screen
M → middle line of screen
L → last line of screen
e → end of next word
– → previous line
+ → next line

$ → to end of the line
^ → to beginning of the line
{ → up one paragraph
} → down one paragraph
^b or ^u → back one page
^f or ^y → forward one page

Example of cursor movement

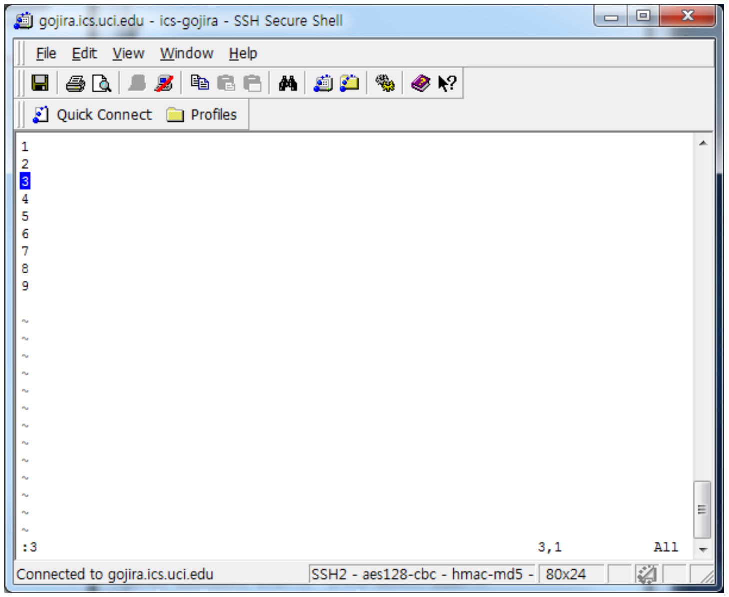


```
1 Come, have breakfast.
2 Look at my hands and my feet.
3 He saw and believed.
4 Who do you say that I am?
5 I do know him and I keep his word.
6 You always have the poor with word.
7 Forgive and you will be forgiven.
8 He loved his own to the end.
9 The Lord is with you.
```

Source)우분투 리눅스, 이종원

Moving with Line mode

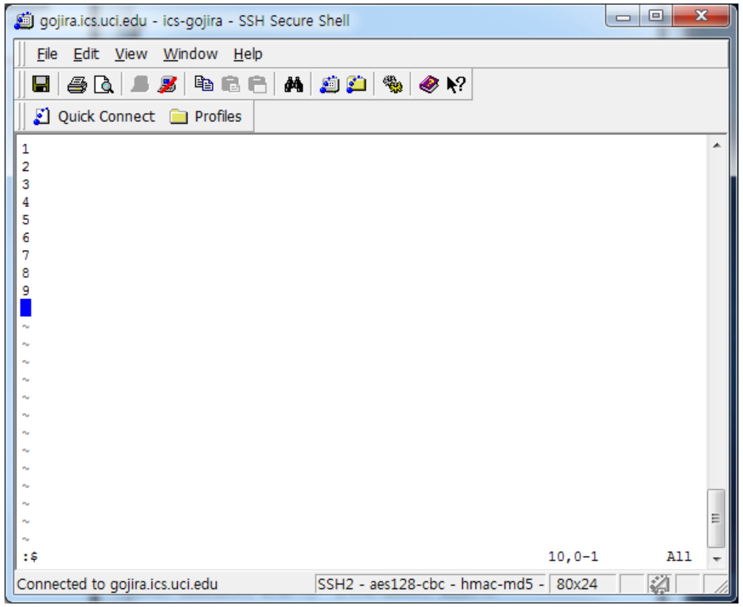
- Move commands
  - :n → move to n_{th} line of a file
  - :$ → move to last line of a file
  - :$= → print the total number of lines of a file
  - :.= → print the number of the current line



```
1
2
3
4
5
6
7
8
9
~
~
~
~
~
~
~
~
~
~
~
~
~
:3                                    3,1          All
```

Connected to gojira.ics.uci.edu | SSH2 - aes128-cbc - hmac-md5 - | 80x24

gojira.ics.uci.edu - ics-gojira - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles



```
1
2
3
4
5
6
7
8
9

~
~
~
~
~
~
~
~
~
~
~
~
~
:$                                    10,0-1       All
```

Connected to gojira.ics.uci.edu | SSH2 - aes128-cbc - hmac-md5 - | 80x24

gojira.ics.uci.edu - ics-gojira - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

Inserting Text

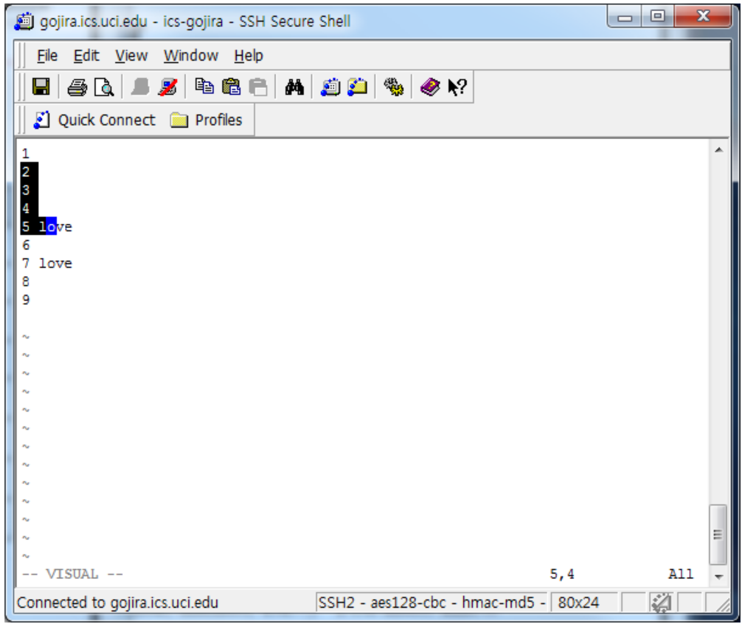
- Options of switching insert mode
  - i → just before the current cursor position
  - a → just after the current cursor position
  - o → into a new line below current cursor
  - I → at the beginning of the current line
  - A → at the end of the current line
  - O → into a new line above current cursor

Cutting, Copying, Pasting

- In command mode
  - x → delete(cut) character
  - 24x → delete(cut) 24 characters
  - dd → delete(cut) current line
  - 4dd → delete(cut) four line
  - D → delete(cut) to the end of line from the cursor
  - dw → delete(cut) to (remainder of) the current word
  - yy → copy (without cutting) current line
  - 5yy → copy (without cutting) 5 line from the current line
  - p → paste after current cursor position/line
  - P → paste before current cursor position/line

Visually Copying

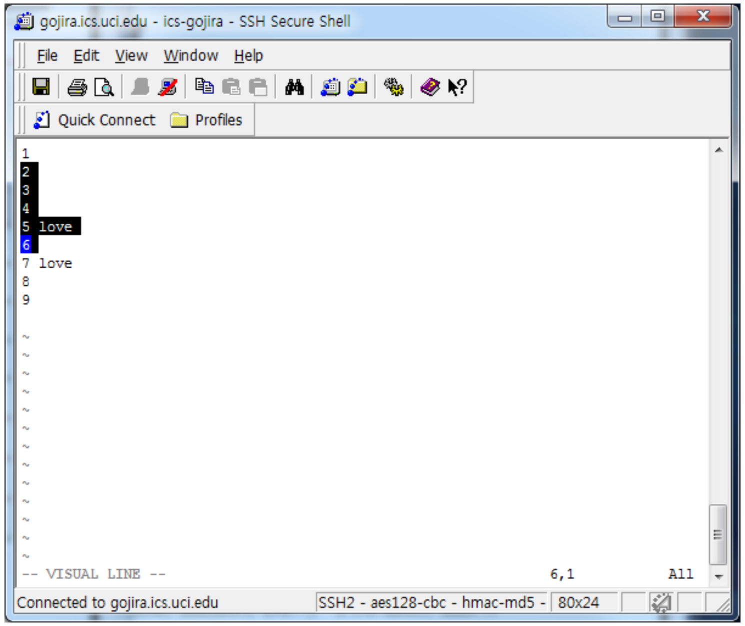
- v → select text for copying (unit of character)
- V → select text for copying (unit of line)



gojira.ics.uci.edu - ics-gojira - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
1
2
3
4
5 love
6
7 love
8
9
~
-- VISUAL --                                      5,4           All
```

Connected to gojira.ics.uci.edu | SSH2 - aes128-cbc - hmac-md5 - | 80x24



gojira.ics.uci.edu - ics-gojira - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
1
2
3
4
5 love
6
7 love
8
9
~
-- VISUAL LINE --                                 6,1           All
```

Connected to gojira.ics.uci.edu | SSH2 - aes128-cbc - hmac-md5 - | 80x24

After pressing v or V, selecting text by moving cursor with h,j,k and l.
Pressing y to finish selecting and copying the selected text

Replacing Text

- This combines two steps : deleting then inserting text
  - r → replace 1 character (under the cursor) with a given character
  - 8r → replace each of 8 characters with a given command
  - R → overwrite, replace text with typed input (ended with ESC key)
  - C → replace from cursor to the end of line, with typed input (ended with ESC key)
  - S → replace the entire line with typed input (ended with ESC key)
  - 4S → replace 4 lines with typed input (ended with ESC key)
  - cw → replace (remainder of) word with typed input (ended with ESC key)

Replacing text in line mode

- :[begin,end]s/pattern1/pattern2/flag
  - Change pattern1 to pattern2
  - [begin,end]
    - Define the range of lines for applying the changes
    - "%" represent the entire file
    - "$" represent the last line
    - "." represent the current line
    - Without [begin,end], the command is applied in <u>the current line</u>
  - Pattern1, Pattern2 → follows regular expression
  - Flag
    - g → every cases
      - C.f. without g option : replacing is applied only to the first matched pattern.
    - c → interactive

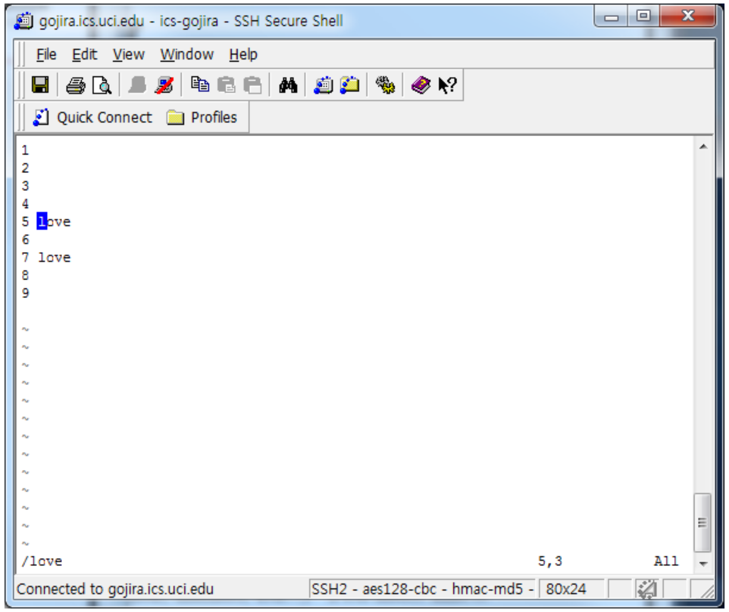
```
:1,10s/peterpan/hook/g
:%s/love/hate/gc
```

Undo and redo and .

- u →undo last changes in the last insert mode
- ^r → redo last changes which were undone.
- . → repeat the last command
  - Every command is applicable

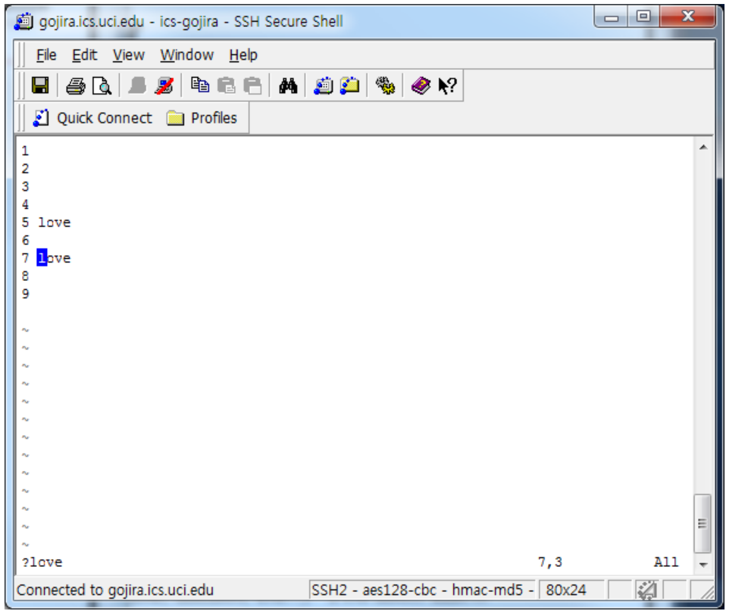
Searching for Text in Line mode

- Use “/” or “?” character
  - /love → jump forward to the next occurrence of the string “love”
  - ?love → jump backward to the previous occurrence of the string “love”
  - n → repeat the last search given by “/” or “?”


gojira.ics.uci.edu - ics-gojira - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
1
2
3
4
5 love
6
7 love
8
9
~
~
~
~
~
~
~
~
~
~
~
~
~
/love                                              5,3          All
```

Connected to gojira.ics.uci.edu | SSH2 - aes128-cbc - hmac-md5 - | 80x24


gojira.ics.uci.edu - ics-gojira - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
1
2
3
4
5 love
6
7 love
8
9
~
~
~
~
~
~
~
~
~
~
~
~
~
?love                                              7,3          All
```

Connected to gojira.ics.uci.edu | SSH2 - aes128-cbc - hmac-md5 - | 80x24

More option for saving in line mode

- :[begin,end]w filename
  - saving the text from begin line to end line into a given filename
- :1,.w filename
  - saving the text from the 1_{st} line to the current line into a given filename
- :1,.w! filename
  - saving the text from the 1_{st} line to the current line into a given filename (if the file exist, overwrite!!)
- :1,.w >> filename
  - appending the text from the 1_{st} line to the current line into a given filename
- :3,$w >> filename
  - saving the text from the 3_{rd} line to the last line into a given filename

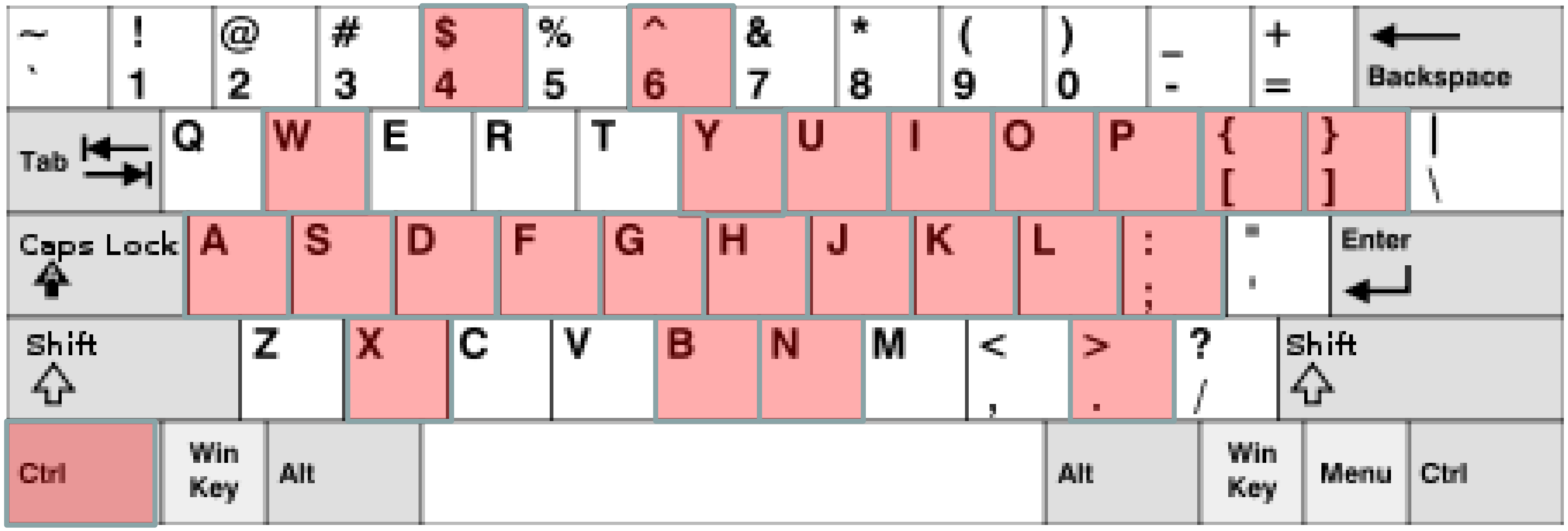
Other linemode commands

- :r filename
  - Read a given filename into the current position of cursor
- :e filename
  - Finish editing the current file (should be saved) and open a given file
- :! command
  - Execute shell command
  - E.g.) :! ls
- :r! command
  - Execute shell command and put the result to the file

Configurations for VI

- Setting environment parameters
  - :set number / :set nonumber
    - Show line number / hide line number (default)
  - :set list / :set nolist
    - Show special characters / hide special characters (default)
  - :set showmode / :set noshowmode
    - Show the current mode / hide the current mode (default)
  - :set tabstop=#
    - Set the number of spaces for tab key
- Method of configuration
  - Using "~/.exrc" file → on the permanent disk
  - Using "EXINIT" shell variable → on the current shell
    - e.g.) $export EXINIT='set nu tabstop=4'

Why is vi addictive…

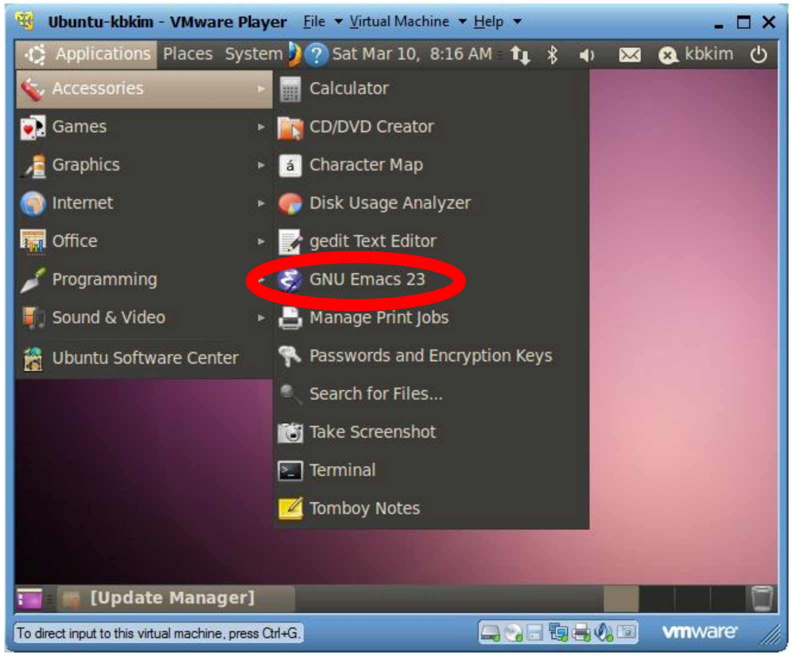


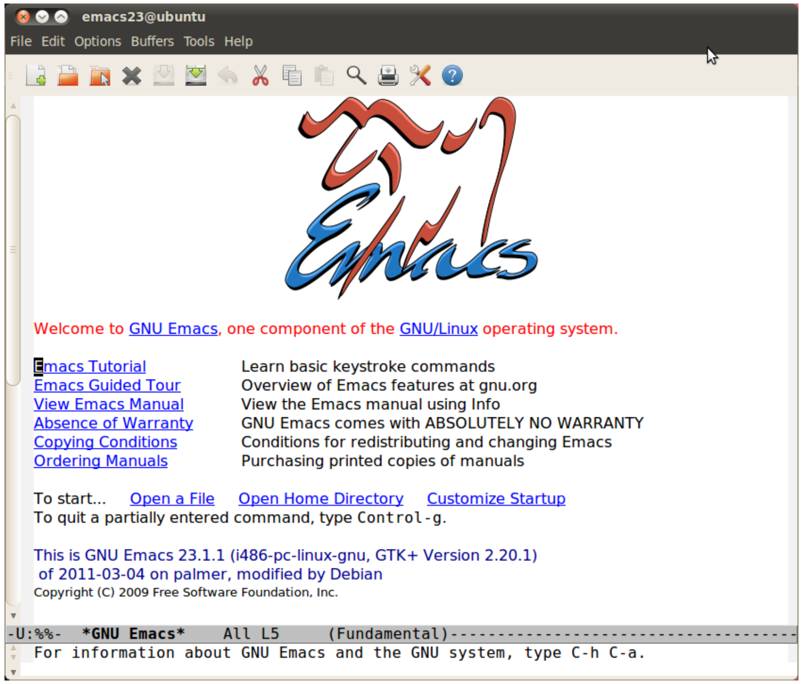
During editing or even commanding, the movement of you hand will be limited.

Worst case

Emacs

- A class of text editor, usually characterized by their extensibility.
- Over 2000 built-in command
- Users can combine these commands into macros to automate work





Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial	Learn basic keystroke commands
Emacs Guided Tour	Overview of Emacs features at gnu.org
View Emacs Manual	View the Emacs manual using Info
Absence of Warranty	GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions	Conditions for redistributing and changing Emacs
Ordering Manuals	Purchasing printed copies of manuals

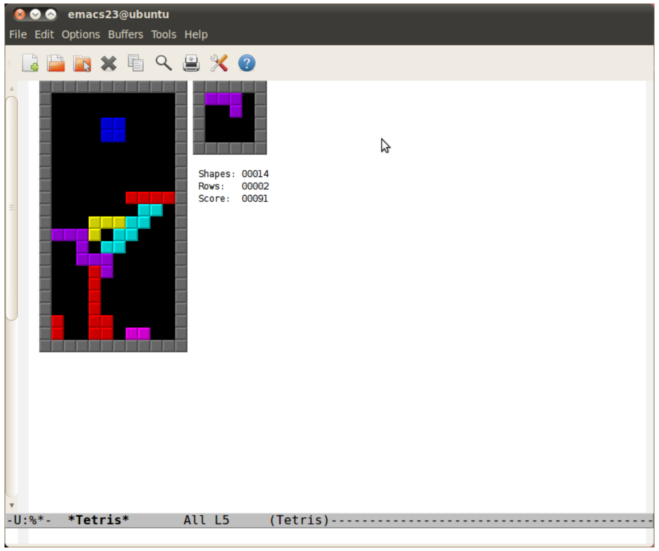
To start... Open a File Open Home Directory Customize Startup
To quit a partially entered command, type `Control-g`.

This is GNU Emacs 23.1.1 (i486-pc-linux-gnu, GTK+ Version 2.20.1)
of 2011-03-04 on palmer, modified by Debian
Copyright (C) 2009 Free Software Foundation, Inc.

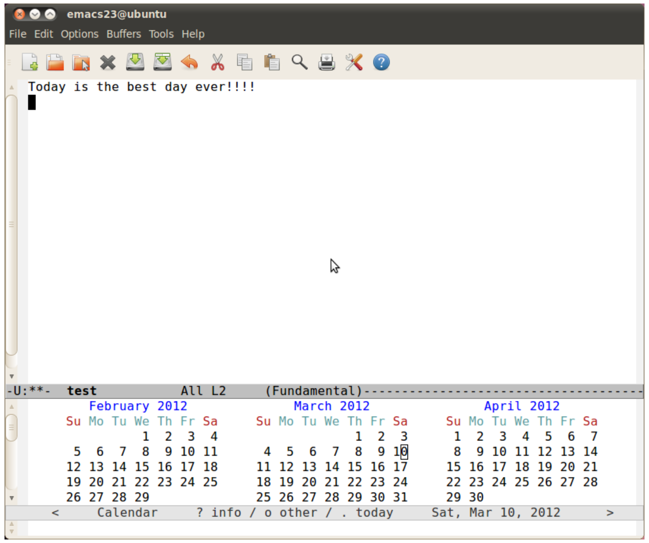
```
-U:%%-  *GNU Emacs*     All L5     (Fundamental)------------------------------------
For information about GNU Emacs and the GNU system, type C-h C-a.
```

Emacs as a flatform

- Emacs is a platform rather than a simple text editor
- You can plot a graph, use calendar or playing game in Emacs
- Lots of utilities can be programmed in Emacs
  - Emacs Lips programming language
  - Customizability and extensibility



```
Shapes:  00014
Rows:    00002
Score:   00091

-U:%*-  *Tetris*       All L5     (Tetris)---------------------------------------
```



```
Today is the best day ever!!!!

-U:**-  test           All L2     (Fundamental)----------------------------------
     February 2012           March 2012            April 2012
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4               1  2  3   1  2  3  4  5  6  7
 5  6  7  8  9 10 11   4  5  6  7  8  9 10   8  9 10 11 12 13 14
12 13 14 15 16 17 18  11 12 13 14 15 16 17  15 16 17 18 19 20 21
19 20 21 22 23 24 25  18 19 20 21 22 23 24  22 23 24 25 26 27 28
26 27 28 29           25 26 27 28 29 30 31  29 30
  <   Calendar    ? info / o other / . today   Sat, Mar 10, 2012    >
```