ITP20004 – Open-Source Software Labs

Lab#2: Basic Linux Commands 2 + vim

Charmgil Hong

charmgil@handong.edu

Spring, 2023

Handong Global University

Announcements

- Team assignment for Weeks 3-5

학번	이름	학번	이름	학번	이름	학번	이름
18	마석재	18	송민준	20	나예원	18	박현우
20	이준형	21	김연희	20	비보시놉아잣	20	윤예람
20	김가현	20	유승준	20	김유겸	20	이상현
18	김두환	22	소종현	20	김승환	20	송산
20	정성호	22	반대준	18	정현준	20	방석민
22	곽철호	19	이지명	19	유건민	17	김홍찬
22	이채연	22	이온유	21	조유진	18	임건호
21	이선환	21	사우지아유인	20	정지원	22	황찬영
18	최정겸	21	송영은				
22	윤유원	21	서준예				

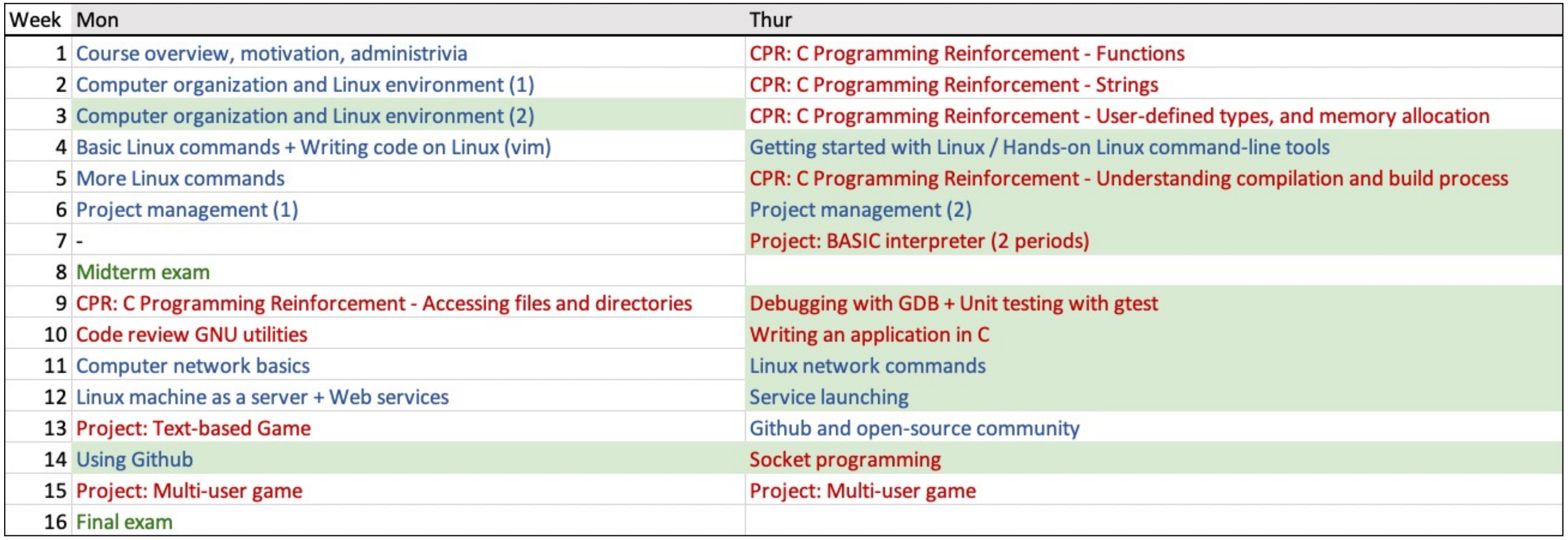
Announcements

- For each lab
  - Before a lab, every student submits a pre-lab report (worksheet-type assignment) – **individual work**
  - After a lab, each team sees and reports to the TA with the results – **team work**

- There will be two re-shuffles of the teams on Weeks 6 and 12
  - A peer evaluation will be conducted at the end of each team-cycle

Announcements

- Reminder: Weekly Schedule



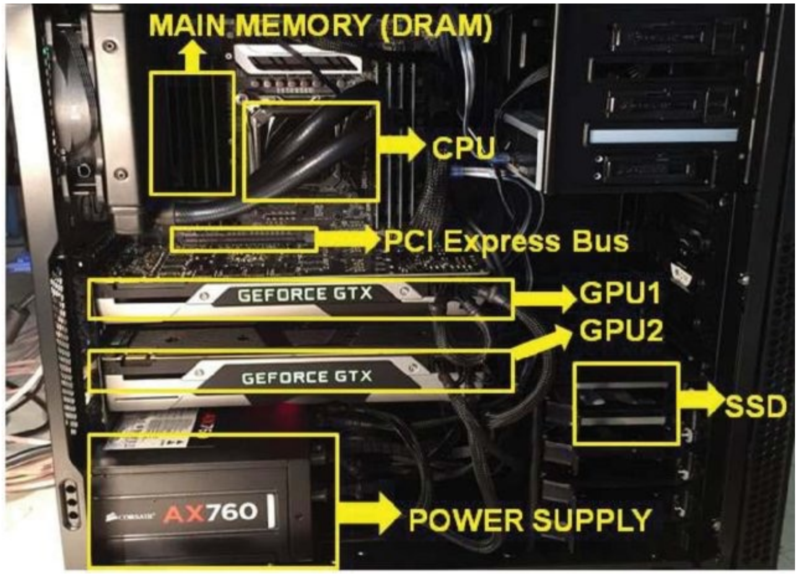
Week	Mon	Thur
1	Course overview, motivation, administrivia	CPR: C Programming Reinforcement - Functions
2	Computer organization and Linux environment (1)	CPR: C Programming Reinforcement - Strings
3	Computer organization and Linux environment (2)	CPR: C Programming Reinforcement - User-defined types, and memory allocation
4	Basic Linux commands + Writing code on Linux (vim)	Getting started with Linux / Hands-on Linux command-line tools
5	More Linux commands	CPR: C Programming Reinforcement - Understanding compilation and build process
6	Project management (1)	Project management (2)
7	-	Project: BASIC interpreter (2 periods)
8	Midterm exam	
9	CPR: C Programming Reinforcement - Accessing files and directories	Debugging with GDB + Unit testing with gtest
10	Code review GNU utilities	Writing an application in C
11	Computer network basics	Linux network commands
12	Linux machine as a server + Web services	Service launching
13	Project: Text-based Game	Github and open-source community
14	Using Github	Socket programming
15	Project: Multi-user game	Project: Multi-user game
16	Final exam	

Announcements

- Submission of your work
  - Make sure to submit your work before each deadline
  - Late submissions will be accepted within 24 hours after the deadline with a penalty of -20% of the assignment grade
    - Submissions made after 24 hours from the deadline will be rejected
  - For additional extensions, reasonable excuse should be requested before the deadline

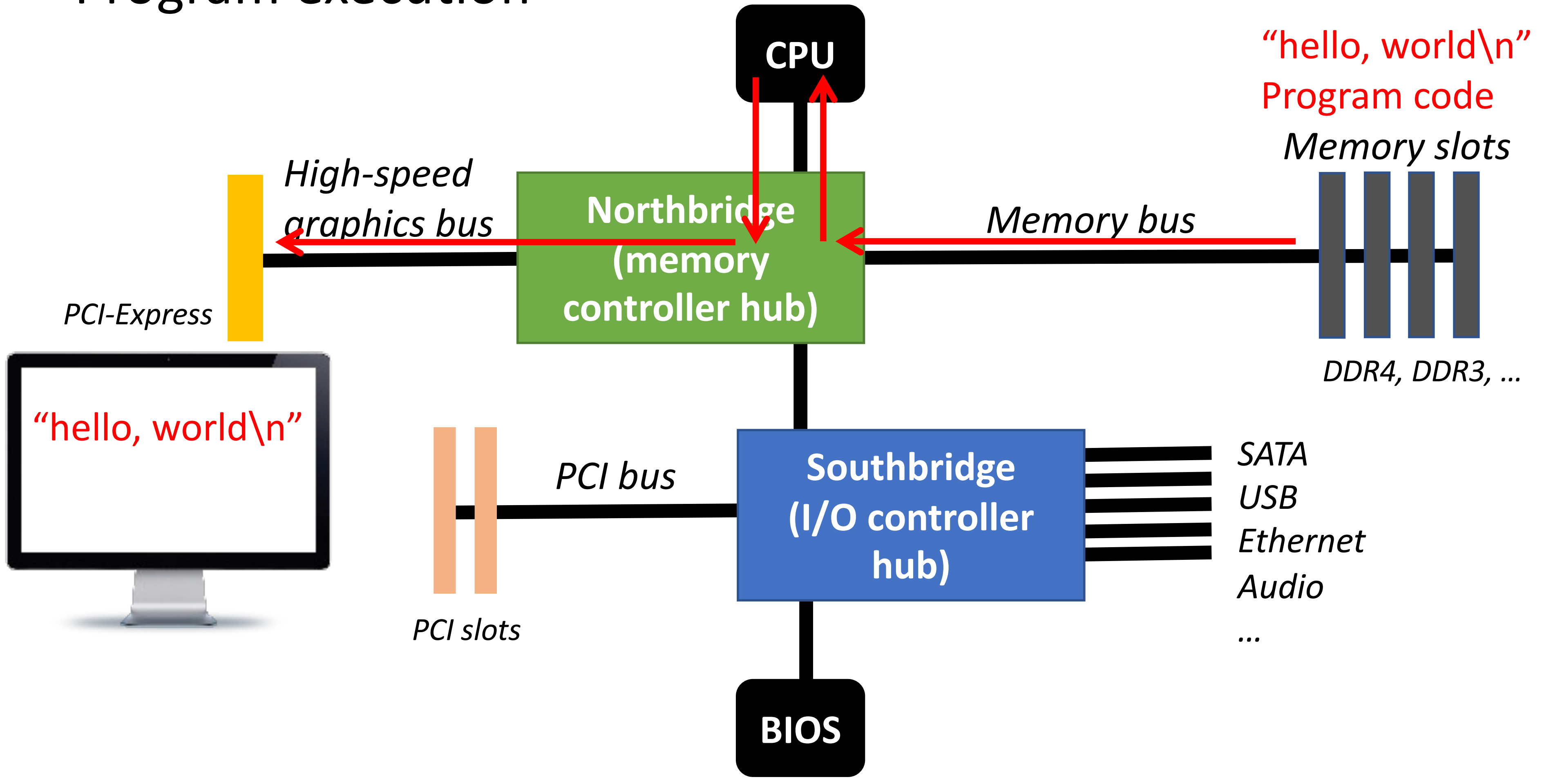
Last Lecture: Hardware Organization

- Main components: CPU (central processing unit), Memory, Input/Output Devices, GPU (graphics processing unit)
  - Communicate over buses
    - **Bus**: A communication system that transfers data between components



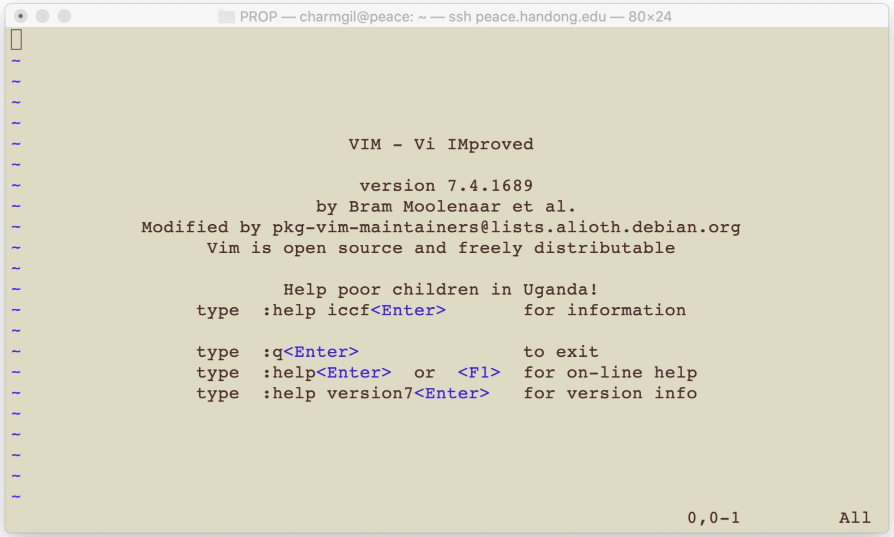
Last Lecture: Hardware Organization

- Program execution



Last Lecture: vim

- Vi Improved, a programmer's text editor
  - "vi" is a text editor from the early days of Unix
    - As the name suggests, "vim" adds a lot of functionalities to the original vi interface
  - Virtually every Linux machine has vim – *wherever you go, you can edit files in the same environment!*



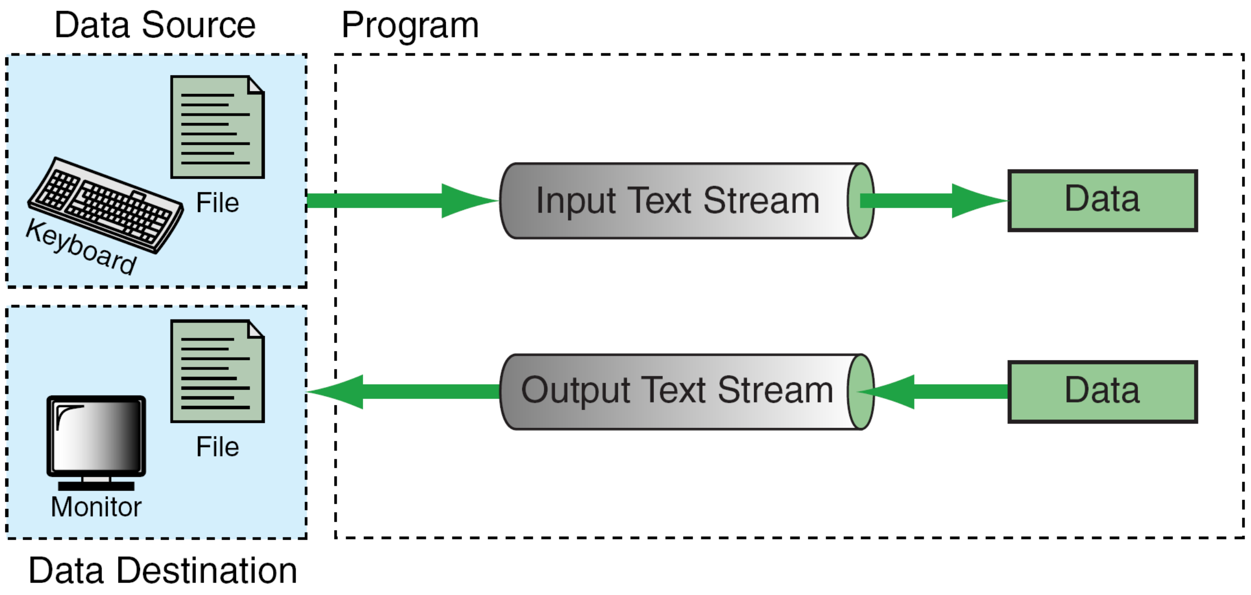
```
~
~
~
~
~                        VIM - Vi IMproved
~
~                        version 7.4.1689
~                    by Bram Moolenaar et al.
~          Modified by pkg-vim-maintainers@lists.alioth.debian.org
~                Vim is open source and freely distributable
~
~                       Help poor children in Uganda!
~               type  :help iccf<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
```

Lab #2

- Task 1 – Getting familiar with the Linux command-line
- Task 2 – Text processing with *vim*

Streams & Redirection

- Streams: Inputs to and outputs from programs
  - Data is read and wrote through stream
  - A stream can be associated with terminal, file, and other data sources or destinations
    - Usually comes from the keyboard; goes to the screen

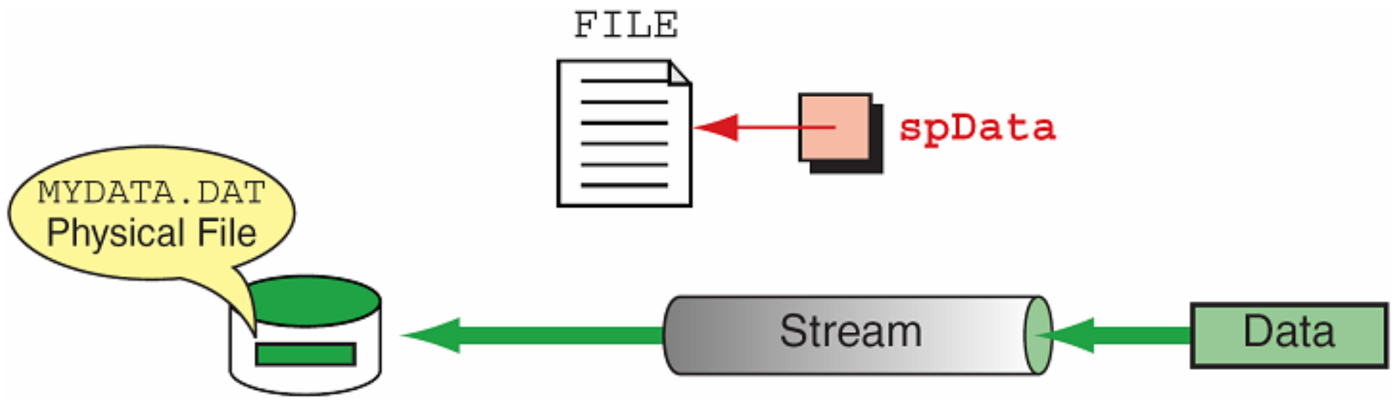


Streams & Redirection

- File open: prepares a file for processing
  - Syntax: `FILE* fopen(“filename”, “mode”);`
    - Filename: name of physical file
    - Mode: string to indicate how the file will be used
    - Return value: pointer to a stream (FILE*)
      - If it fails to open a file, return NULL.

E.g., `FILE* spData = fopen(“MYFILE.DAT”, “w”);`

```
FILE* spData = fopen(“A:\\MYFILE.DAT”, “w”);
```



Streams & Redirection

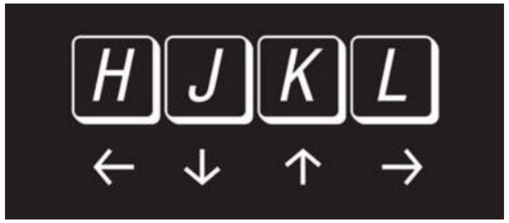
- Linux shells use three “standard” streams:
  - Standard input (`stdin`): usually the input from the keyboard
  - Standard output (`stdout`): displays the output from commands, usually to the terminal
  - Standard error (`stderr`): displays error output from commands
    - Usually sent to the same output as standard output
    - Can be redirected separately from `stdout`

vim

- *vim*
  - Cursor movements
  - Modes
  - Editing shortcuts
  - Search and replace

vim – Basic Operations

H, J, K, L	Maneuvering the cursor (the arrow keys work too)
:wq	Save and quit
:q!	Quit without save



I Am Devloper
@iamdevloper

Following

I've been using Vim for about 2 years now, mostly because I can't figure out how to exit it.

Reply Retweet Favorite More

RETWEETS 4,846 FAVORITES 2,105

4:56 AM - 18 Feb 2014

vim – Modal Editor

- Four mode
  - Command mode: All keystrokes are interpreted as commands
  - Command line mode: Providing a command prompt
  - Insert mode: Regular typing as you expect from an editor
  - Visual mode: Highlighting

vim – Modal Editor

Four mode
- **Command mode**
- Command line mode
- Insert mode
- Visual mode

- Command mode (normal model)
  - Default when vim is opened
  - Keystrokes are commands
  - From other modes, use ‘esc’ to switch to normal mode

vim – Modal Editor

Four mode
- Command mode
- **Command line mode**
- Insert mode
- Visual mode

- Command line mode
  - From normal mode, press ‘:’ to trigger command line mode
  - Similar to the ”File” menu on common text editors
  - Used for opening, closing files; finding and replacing text; *etc.*

vim – Modal Editor

Four mode
- Command mode
- Command line mode
- **Insert mode**
- Visual mode

- Insert mode
  - More normal for modern editor users
  - Keystrokes insert text
  - Commands are possible with key combinations
  - From command mode, use ‘i’ to switch to insert mode

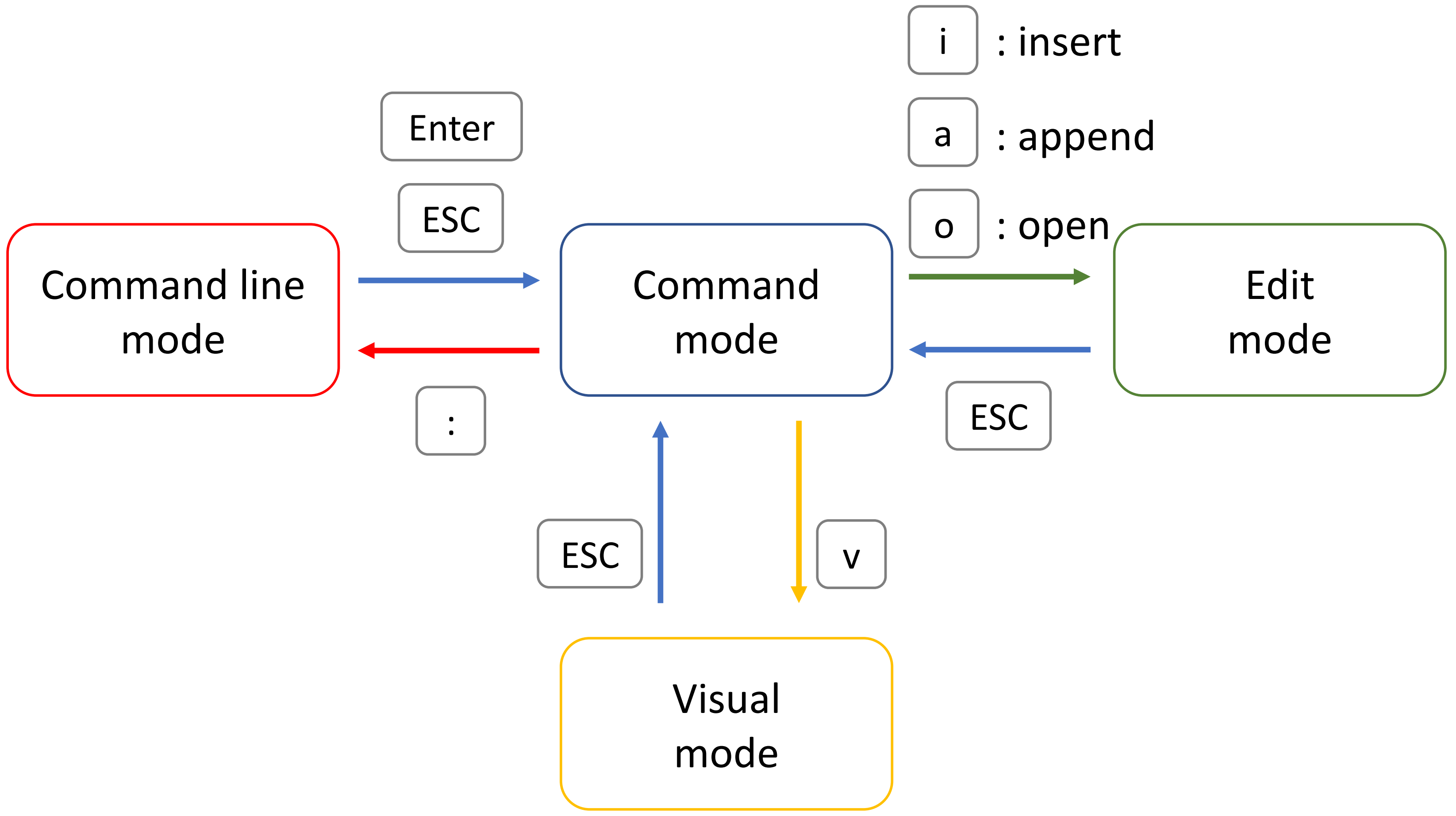
vim – Modal Editor

Four mode
Command mode
Command line mode
Insert mode
Visual mode

- Visual mode
  - For highlighting/text selection
  - Keystrokes are commands
  - From command mode, use ‘v’ to switch to visual mode

vim – Modal Editor

- Switching between modes



vim – Editing Shortcuts

- Switching to the intert mode
  - i Insert at the current position
  - a Insert at the next next position
  - o Insert at the next line
  - I Insert at the beginning of the current line
  - A Insert at the end of the current line
  - O Insert at the previous line
  - R Replace from the current position

vim – Editing Shortcuts

- Deletions (cut)

x	Delete a character
*n*x	Delete *n* characters
dd	Delete a line
d*n*	Delete *n* lines
dw	Delete a word
d*n*w	Delete *n* words

- Undo/Redo

u	Undo
Ctrl + r	Redo

vim – Editing Shortcuts

- Copy and Paste
  - Copy (yank)
    - yy Copy a line
    - y*n* Copy *n* lines
    - yw Copy a word
    - y*n*w Copy *n* words

    - p Paste

    - v Select text (visual mode)

vim – Search and Replace

- Search
  - / Searches in the document for a string (can take regular expression)
  - n Goes to next occurrence of the search string
  - N Goes to previous occurrence of the search string

- Replace
  - :%s/foo/bar Replaces <u>an</u> occurrence of 'foo' with 'bar' (once)
  - :%s/foo/bar/g Replaces <u>all</u> occurrences of 'foo' with 'bar' (global)

- Count occurrences
  - :%s/foo//gn Counts all occurrences of 'foo'
    (make sure to have double-slashes)

vim – Search and Replace

- Search with regular expression
  - Regular expression: Sequence of characters that specifies a match pattern in text
    - *An introduction to regular expressions in Vim* [***https://youtu.be/4KwsijqA7tQ***](https://youtu.be/4KwsijqA7tQ)

vim – Links

- You can download vim on your Windows too
    - https://www.vim.org/
- vim plug-ins
    - https://vimawesome.com/
- vim tutorials/manuals
    - https://www.csie.ntu.edu.tw/~piaip/vim/vimbook-OPL.pdf
- References:
    - https://www.slideshare.net/BenMcCormick/vim-survival-guide-71763917
    - https://www.slideshare.net/brandonliu/introduction-to-vim

Lab#2: Basic Linux Commands 2 + *vim*

- **Happy hacking!**