Linux System Guide for CS230 Students

Original Version by Taekyung Heo
Updated by Soojin Hwang

What is Linux?

- **What is Linux?**
  - Linux is a modern, free operating system.
  - First developed by Linus Torvalds in 1991.
  - **Features**
    - Multi-tasking, multi-user
    - Various distributions (Ubuntu, CentOS, …)
    - Fully customizable
- **Why do we need Linux?**
  - Knowing how to use Linux systems is important for computer engineers.
  - You will do your projects on Linux systems.

Connection to a Linux Machine

- How can I use a Linux machine?
  - You have to establish a connection to it.
- How can I connect to a machine?
  - Use ssh clients like PuTTY or VSCode.

PuTTY

PuTTY: A Free Telnet/SSH Client

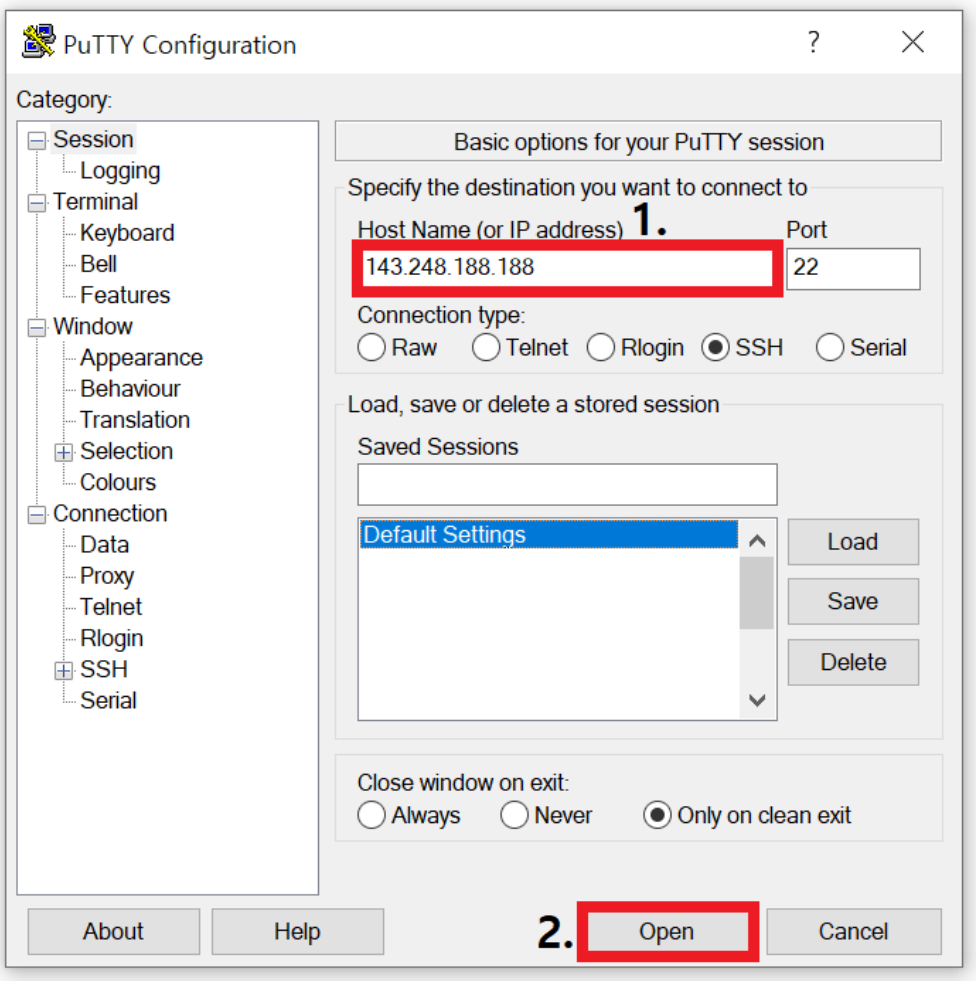
- Very light-weight ssh client
- Does not need installation
- Not user-friendly interface

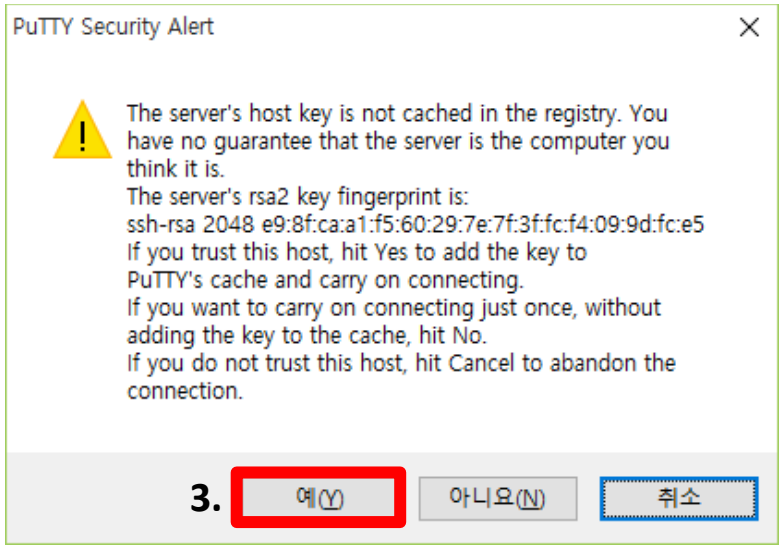
VSCode

VSCode (Visual Studio Code)

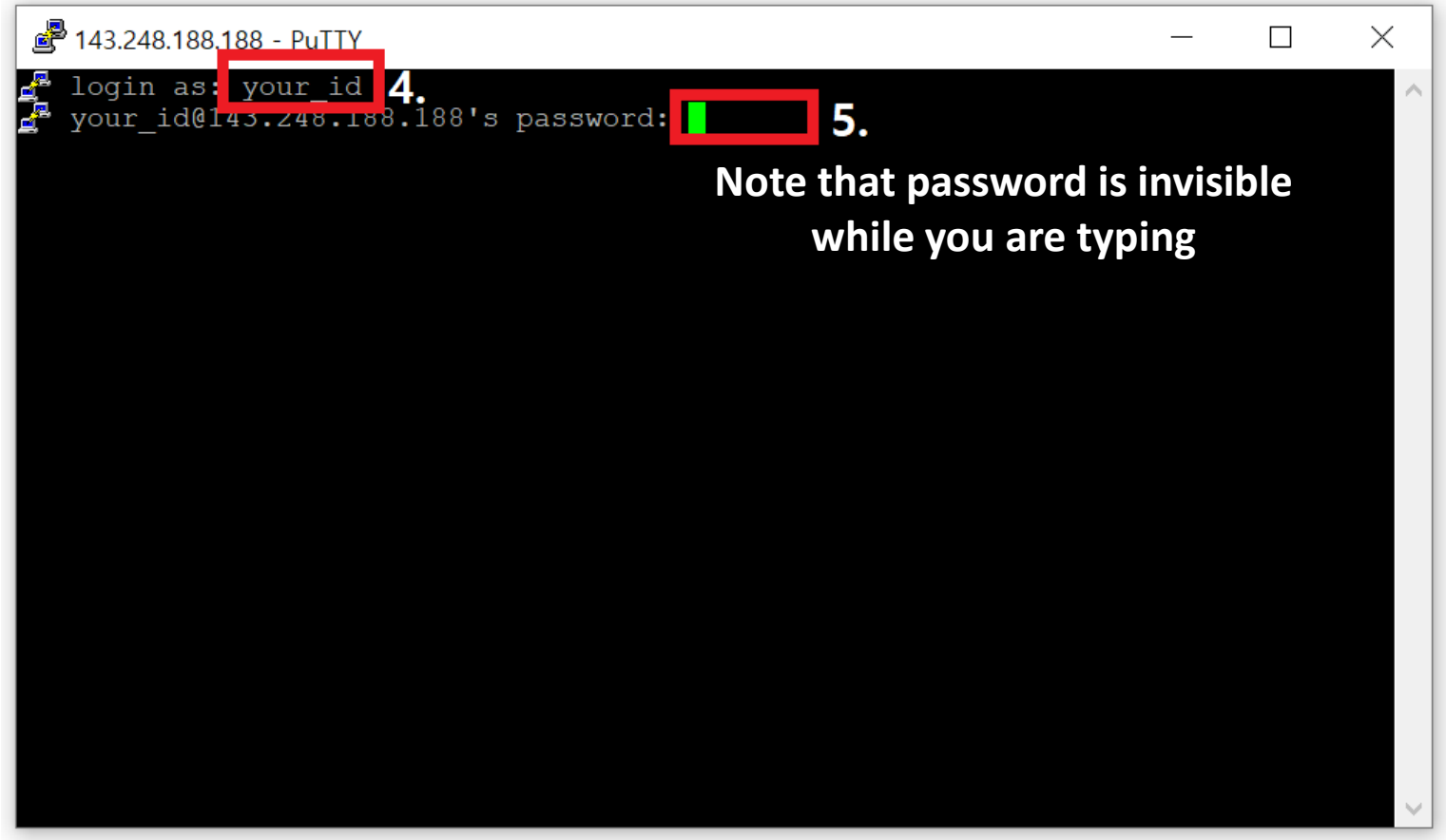
- Needs installation & setup
- User-friendly interface

Connection Example: PuTTY



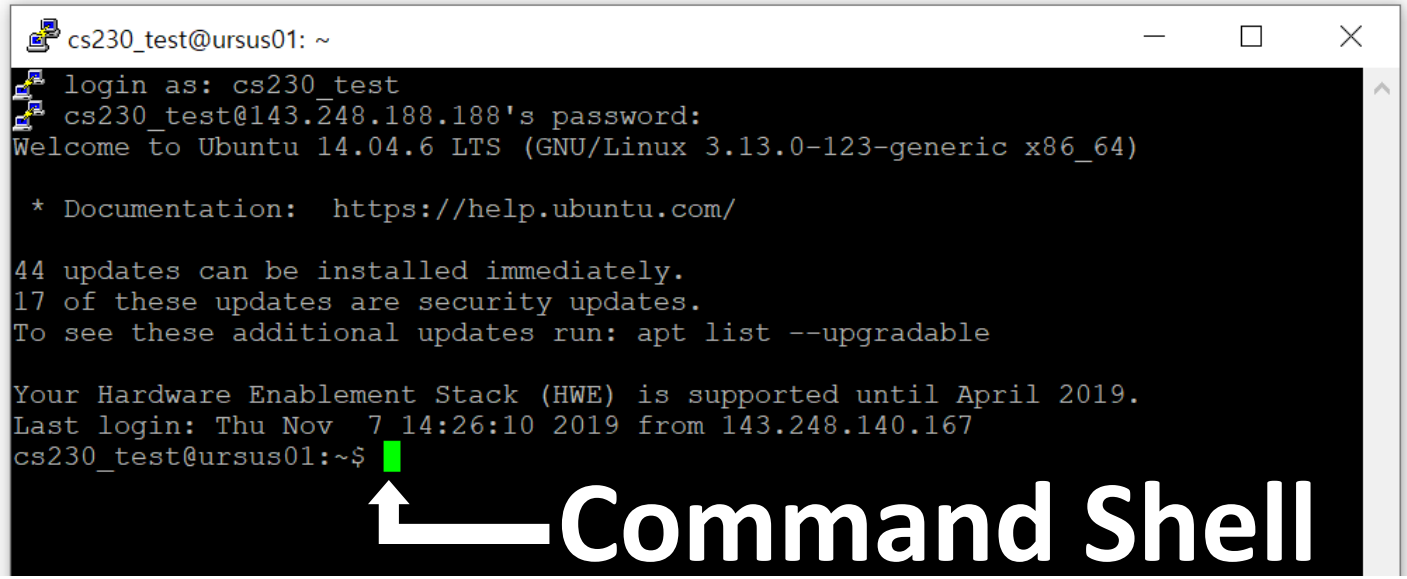


Connection Example: PuTTY



Linux Shell

Connected!



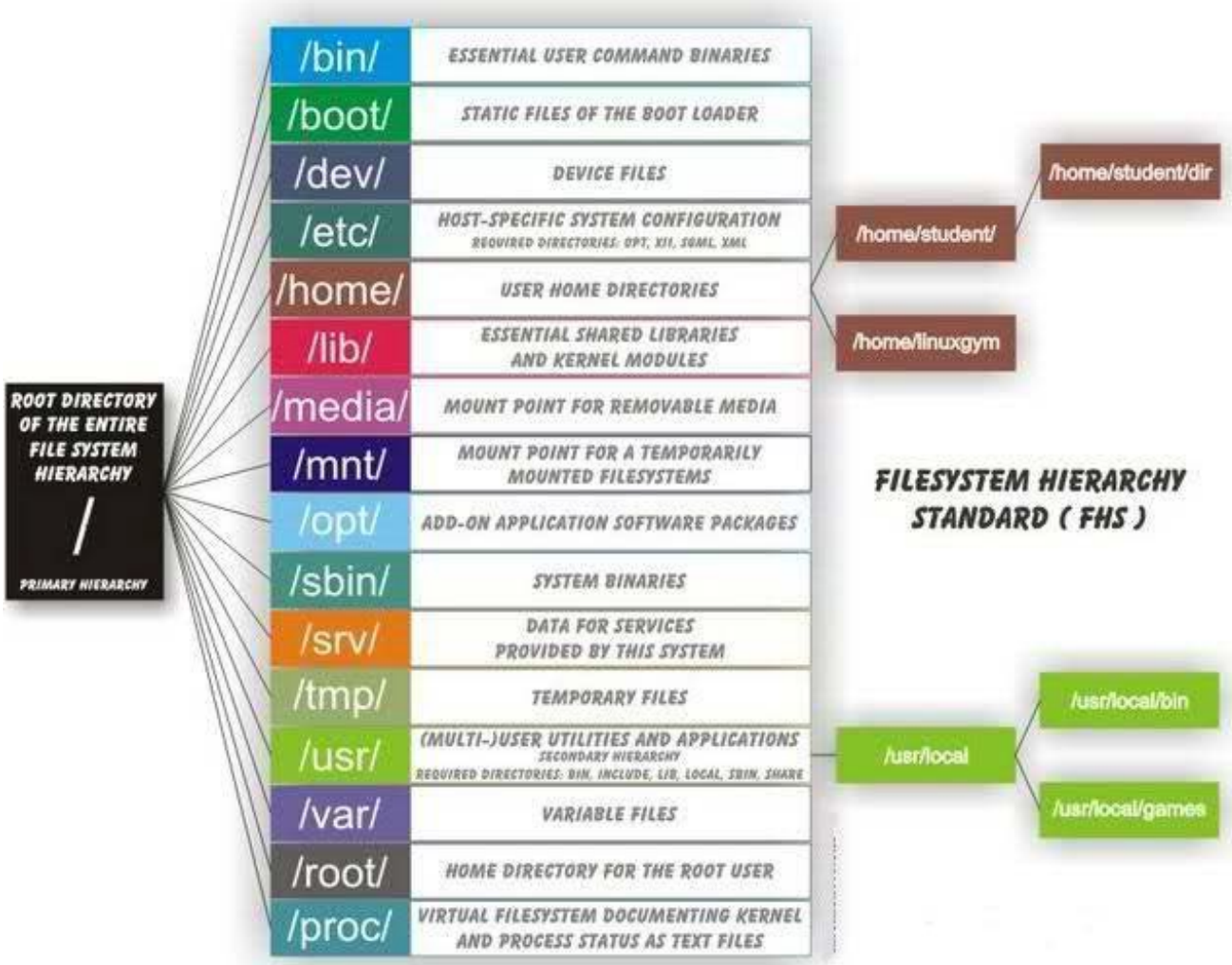
Shell?

- **a command line interface between a user and Linux machine**

Linux Directory

- / : root directory
- ~ : user's home directory (usually same as /home/[username])
- . : current(working) directory
- .. : upper(parent) directory

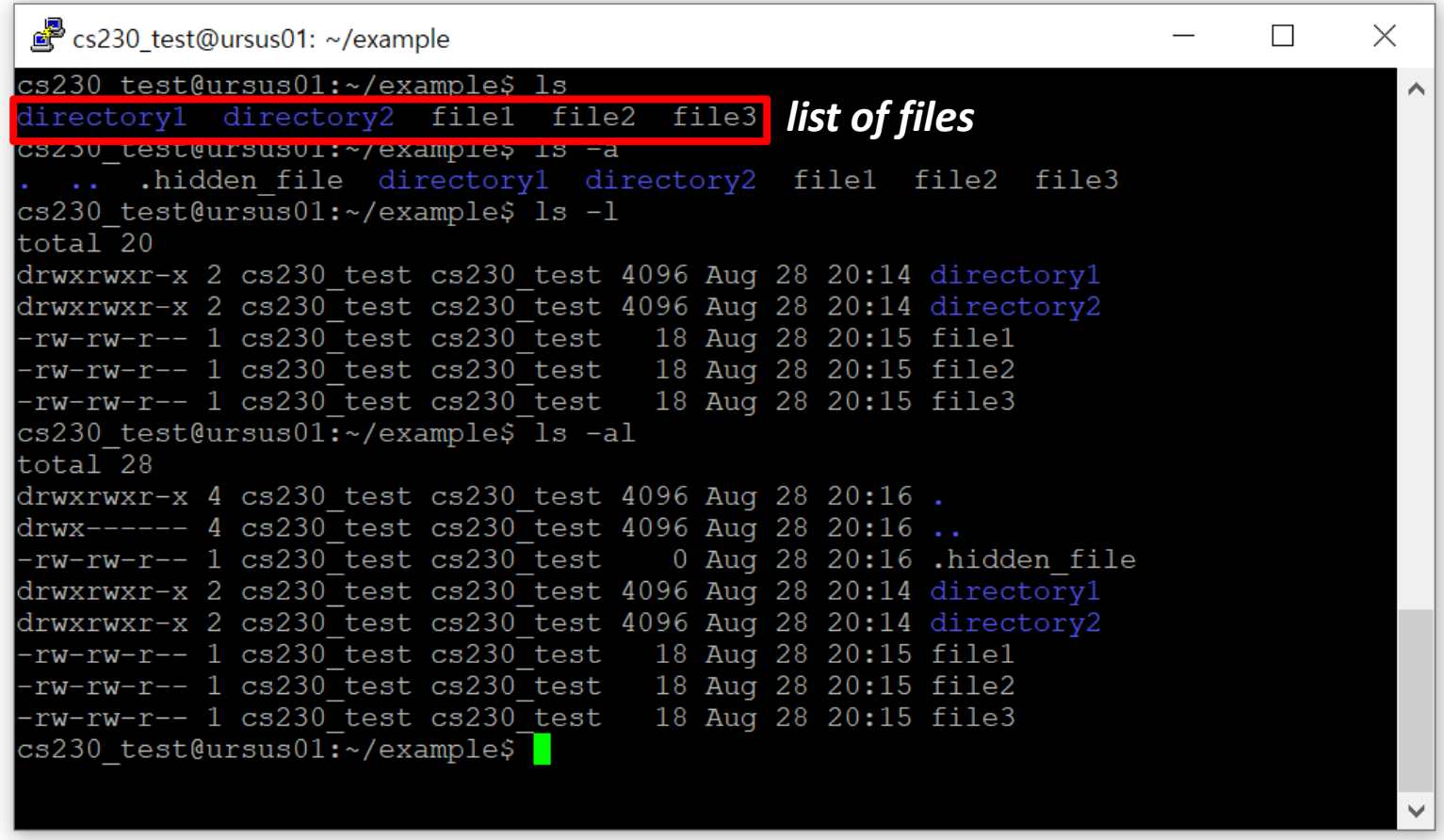
Linux Directory Structure



http://www.tecmint.com/linux-directory-structure-and-important-files-paths-explained/

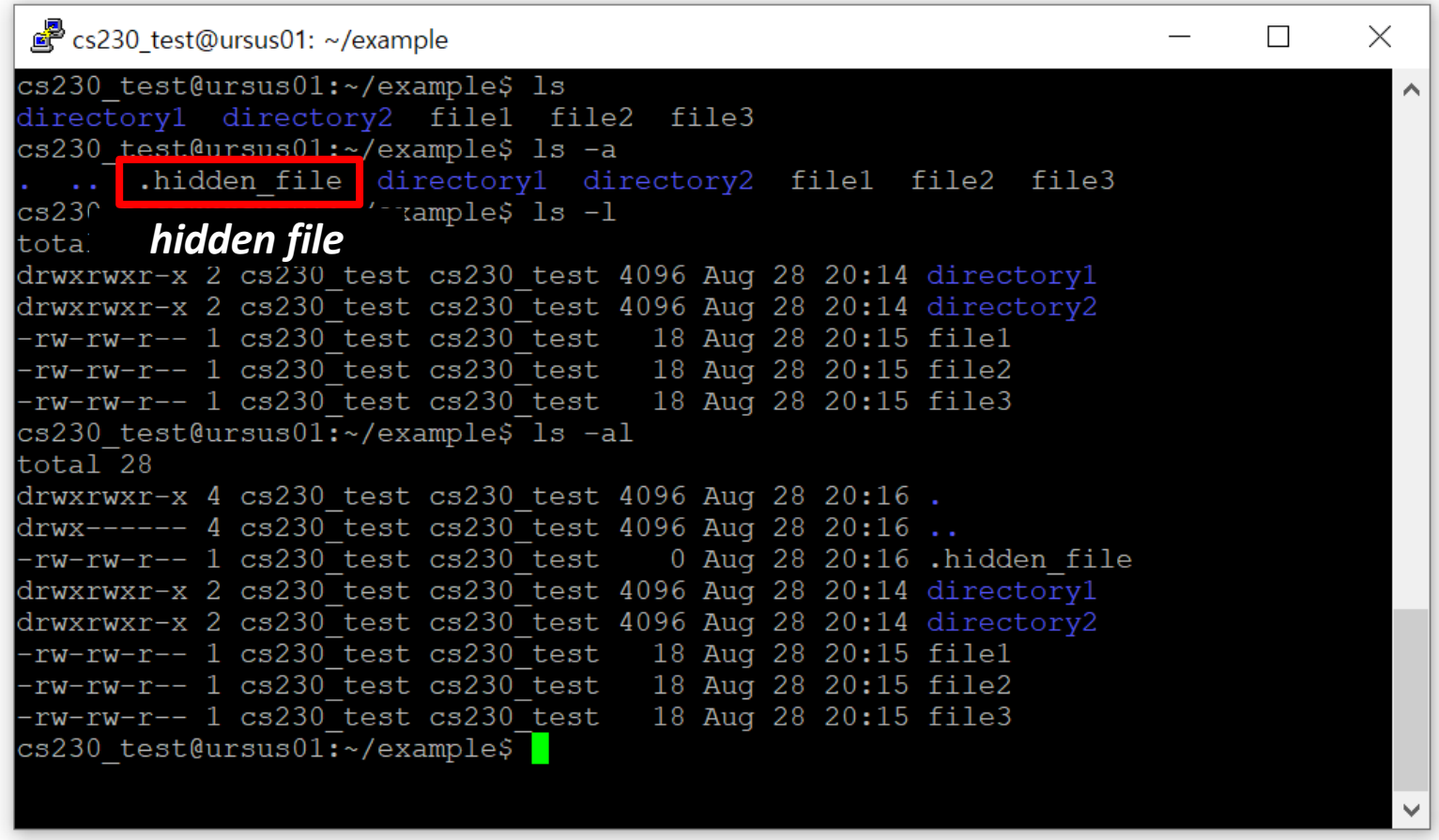
Linux Commands

- ls [directory] (empty for working directory)
  - list directory contents



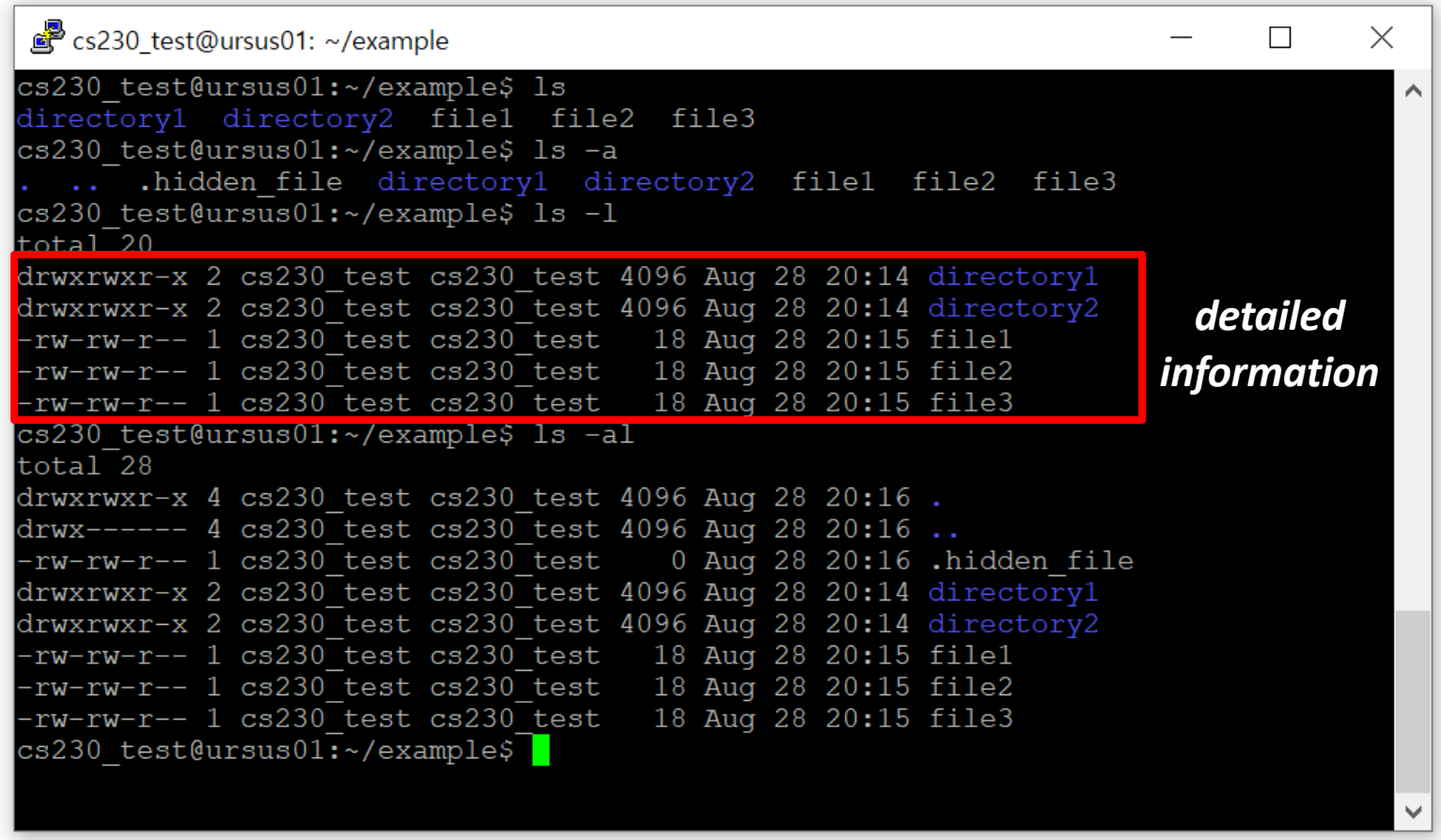
Linux Commands

- ls [directory] (empty for working directory)
  - -a : Print the list including hidden contents (. name)



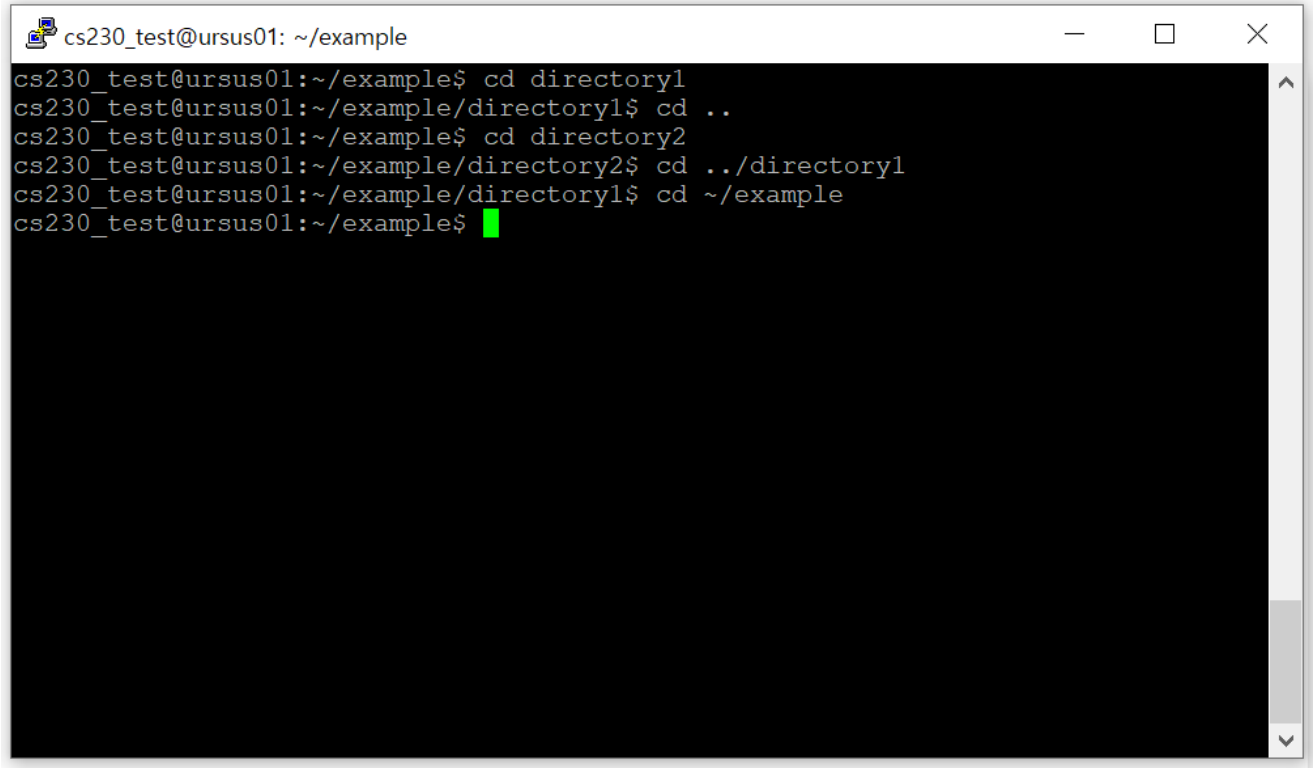
Linux Commands

- ls [directory] (empty for working directory)
  - -l : Print the list of files with detailed information
    - Permissions, number of links, owner name, owner group, size, last modification time, name



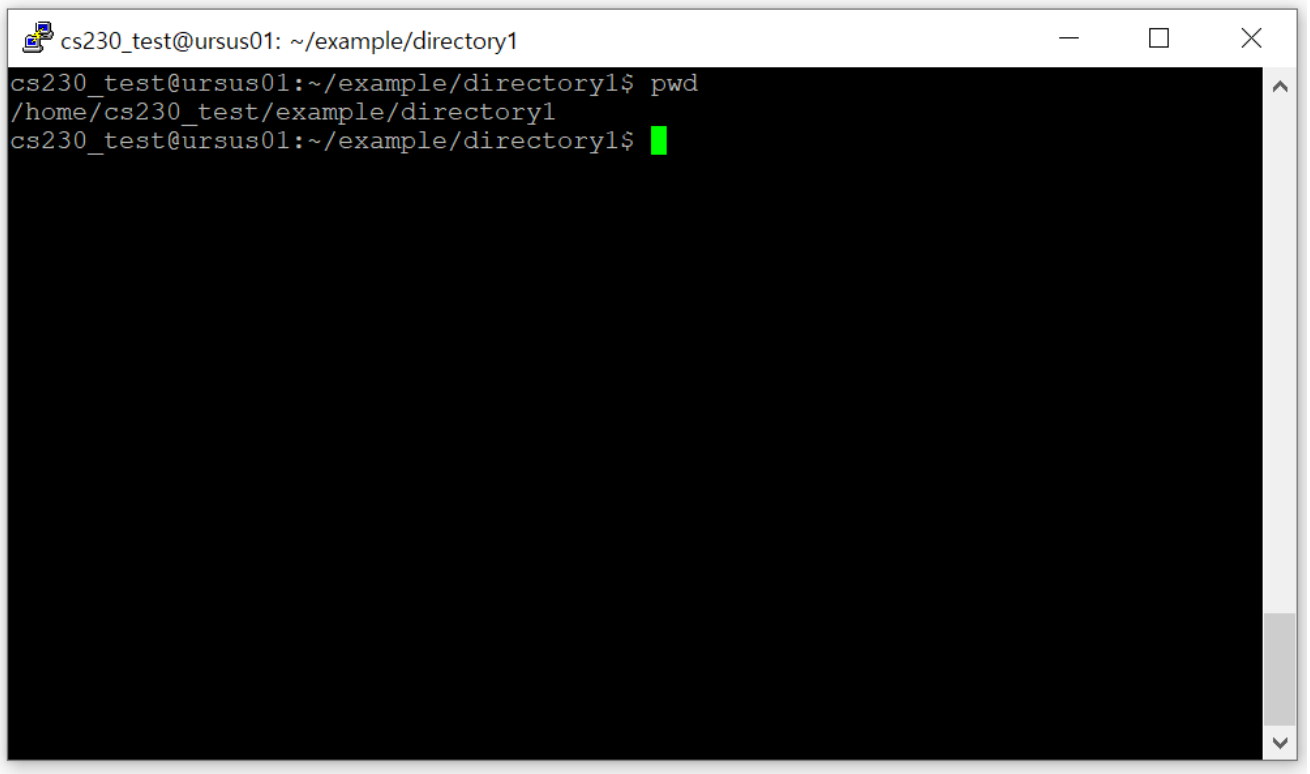
Linux Commands

- cd [directory]
  - change working directory



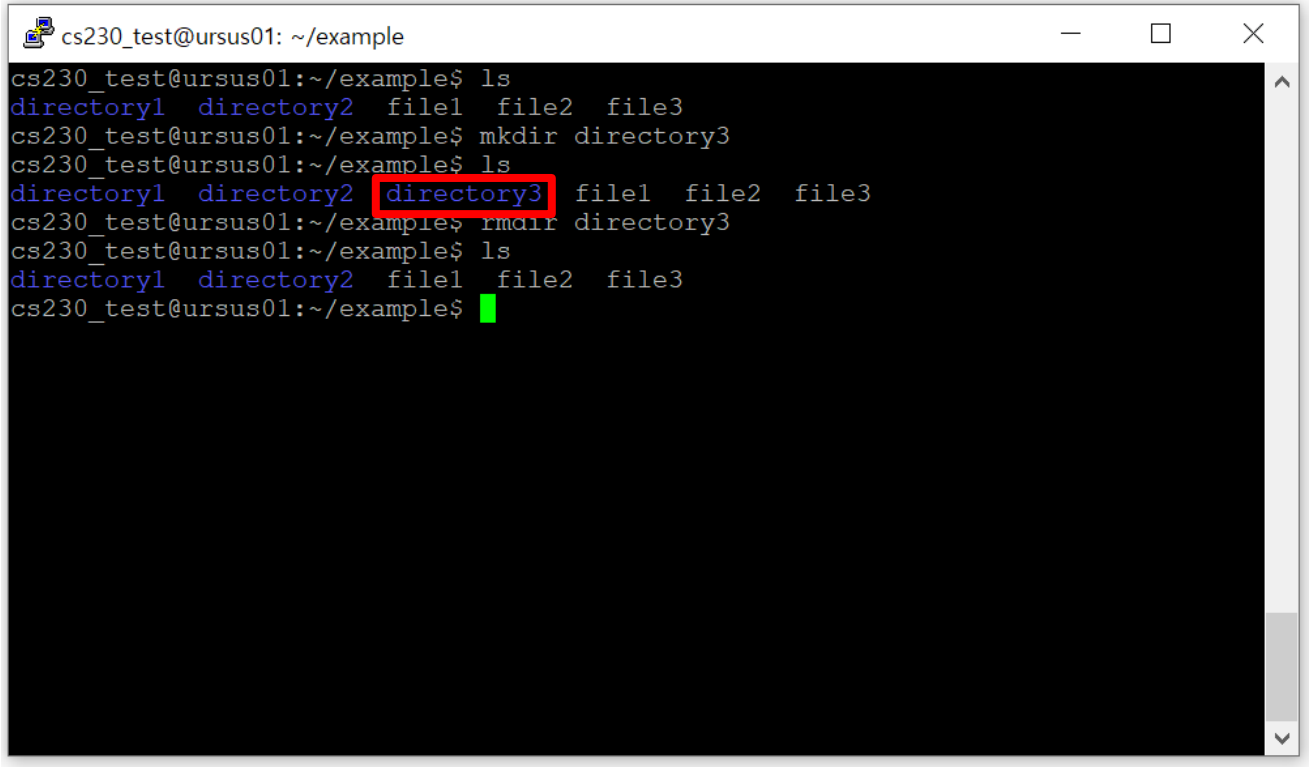
Linux Commands

- pwd
  - print name(absolute path) of working directory



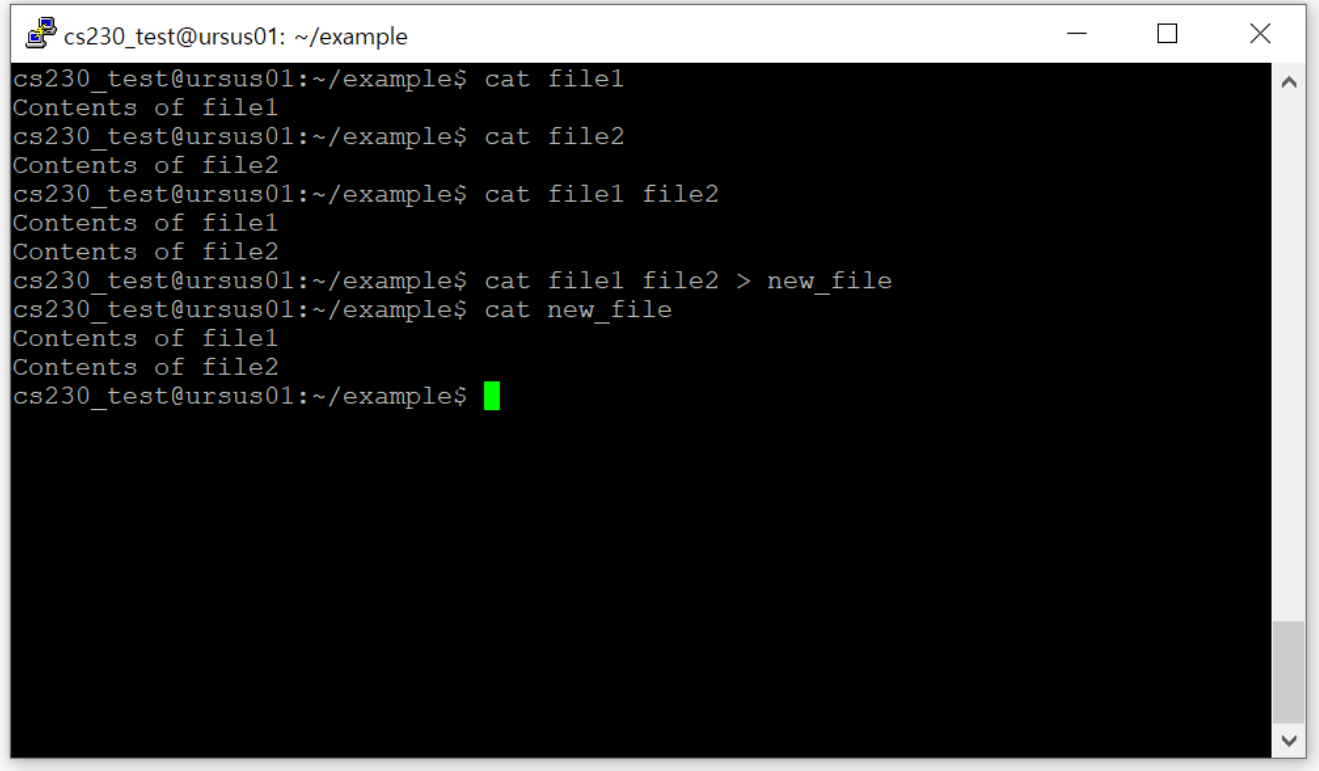
Linux Commands

- mkdir [name], rmdir [name]
  - mkdir – make directories
  - rmdir – remove empty directories



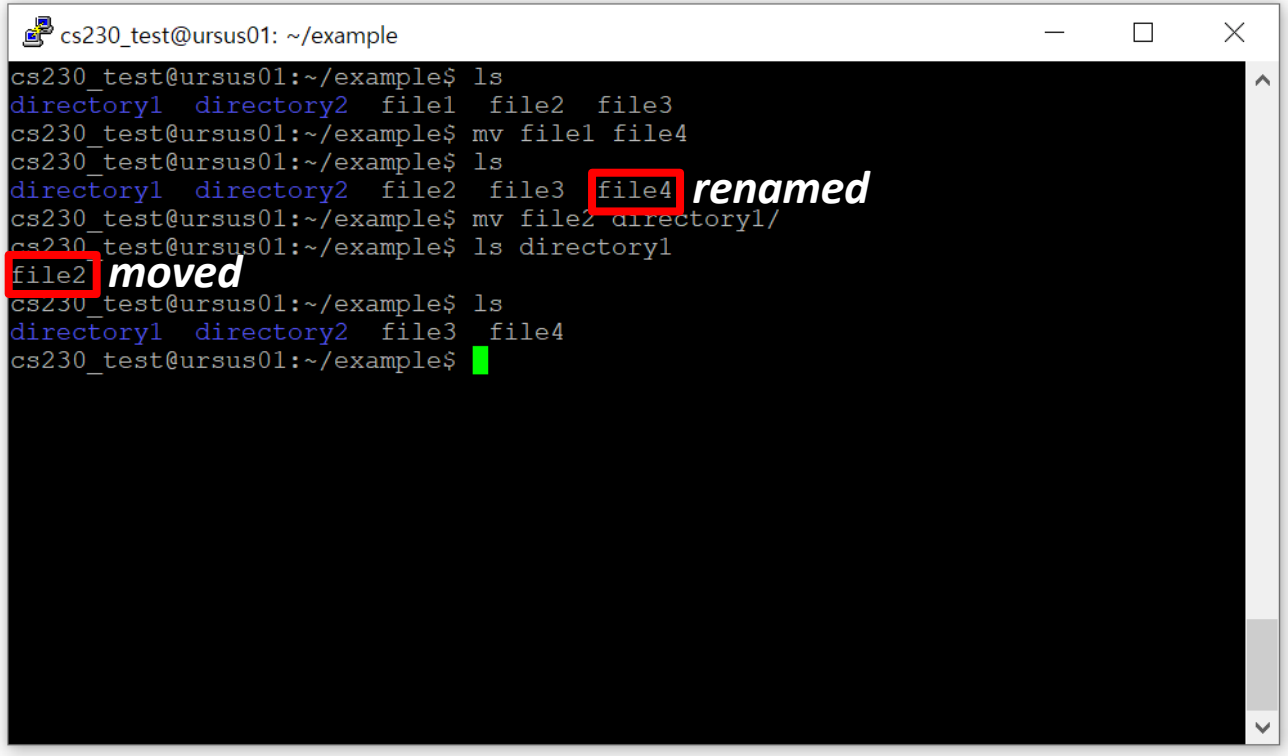
Linux Commands

- cat
  - concatenate files and print on the standard output
  - > : redirects standard output to a file (overwrite)
  - >> : redirects standard output to a file (append)



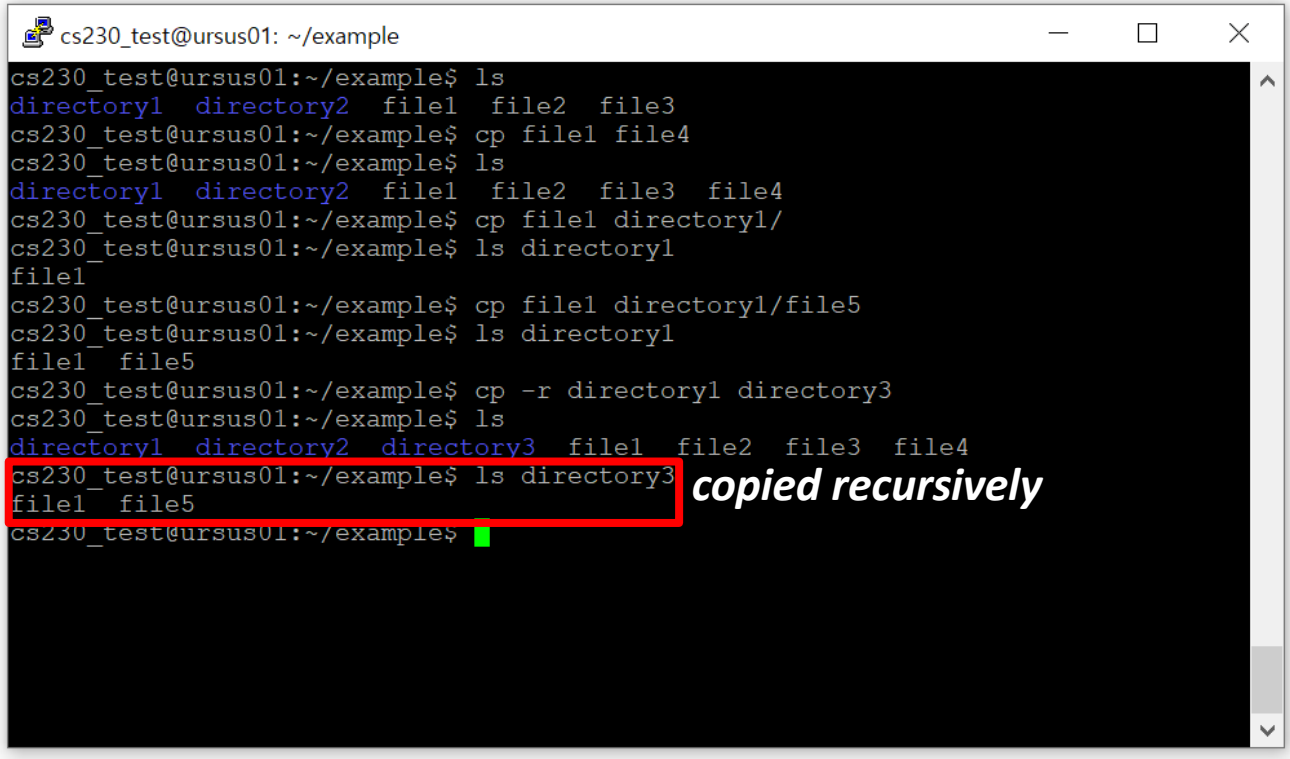
Linux Commands

- mv [source] [destination]
  - Move or rename files or directories



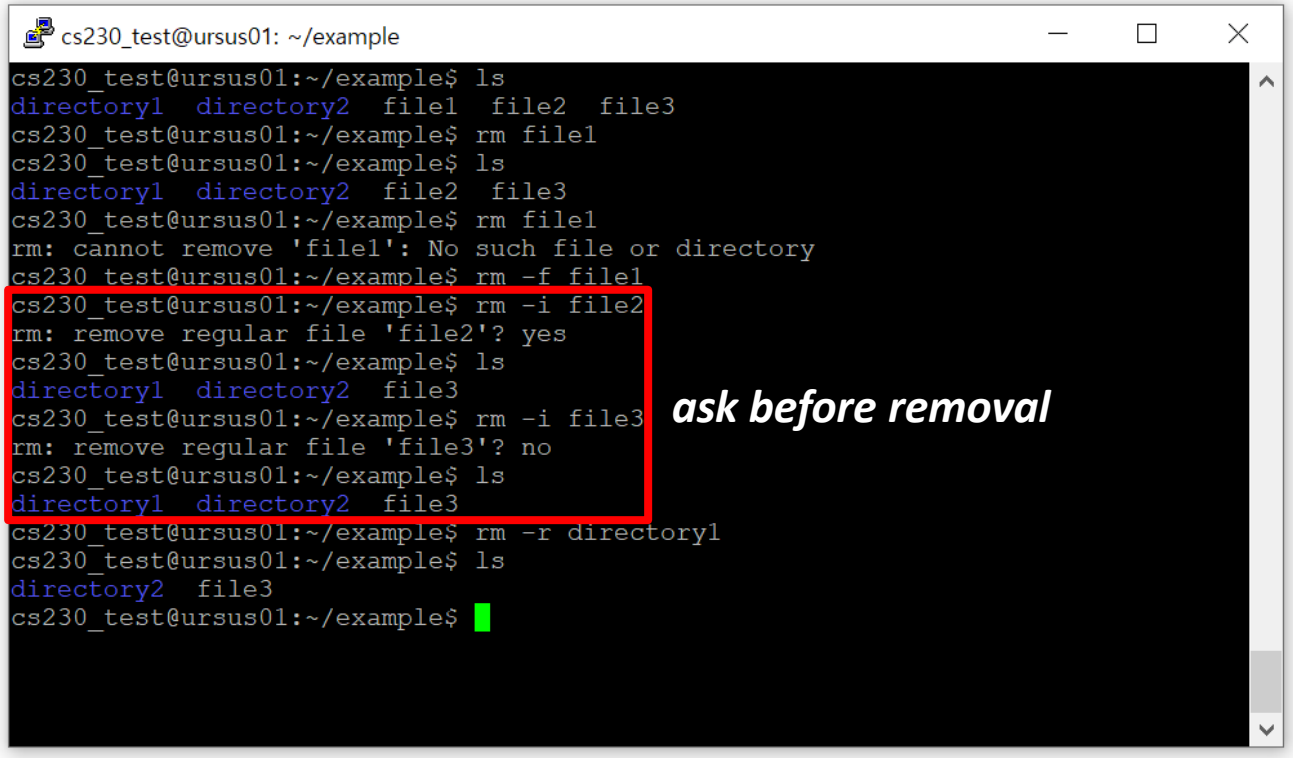
Linux Commands

- cp [source] [destination]
  - Copy one or more files to specified location
  - -r : copy directories recursively



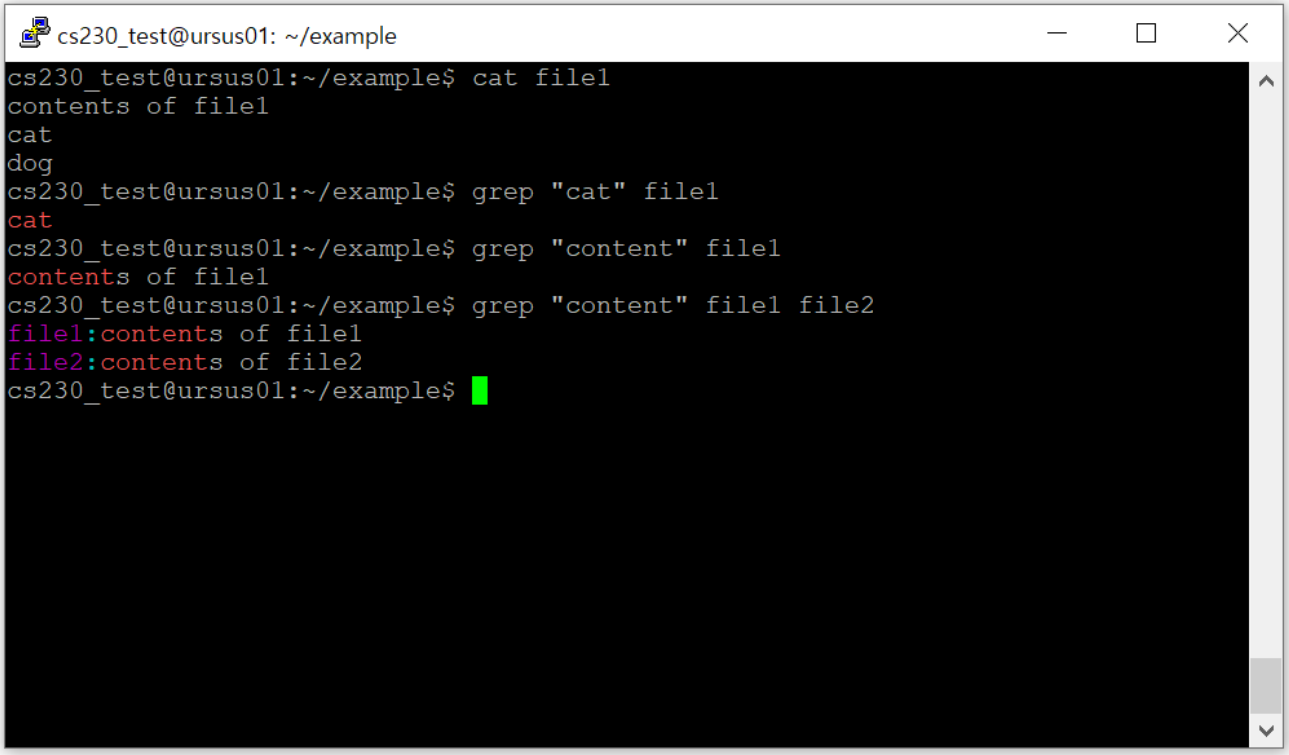
Linux Commands

- rm [file/directory]
  - remove files or directories
  - -f : ignore nonexistent files, never ask
  - -i : ask whether really remove file or not
  - -r : Remove directories and their contents recursively



Linux Commands

- grep [pattern] [files]
  - print lines matching a pattern



Linux Commands

- man [command name]
  - an interface to the on-line reference manuals

```
GREP(1)                          General Commands Manual                          GREP(1)



NAME
       grep, egrep, fgrep, rgrep - print lines matching a pattern

SYNOPSIS
       grep [OPTIONS] PATTERN [FILE...]
       grep [OPTIONS] [-e PATTERN | -f FILE] [FILE...]

DESCRIPTION
       grep  searches  the  named input FILEs (or standard input if no files are named, or if a single hyphen-minus
       (-) is given as file name) for lines containing a match to the given PATTERN.  By default, grep  prints  the
       matching lines.

       In  addition,  three  variant  programs egrep, fgrep and rgrep are available.  egrep is the same as grep -E.
       fgrep is the same as grep -F.  rgrep is the same as grep -r.  Direct invocation as either egrep or fgrep  is
       deprecated, but is provided to allow historical applications that rely on them to run unmodified.

OPTIONS
   Generic Program Information
       --help Print  a  usage message briefly summarizing these command-line options and the bug-reporting address,
              then exit.

       -V, --version
              Print the version number of grep to the standard  output  stream.   This  version  number  should  be
              included in all bug reports (see below).

   Matcher Selection
       -E, --extended-regexp
              Interpret PATTERN as an extended regular expression (ERE, see below).  (-E is specified by POSIX.)
```

Linux Commands

- Running a command in background :
  $ [command] &

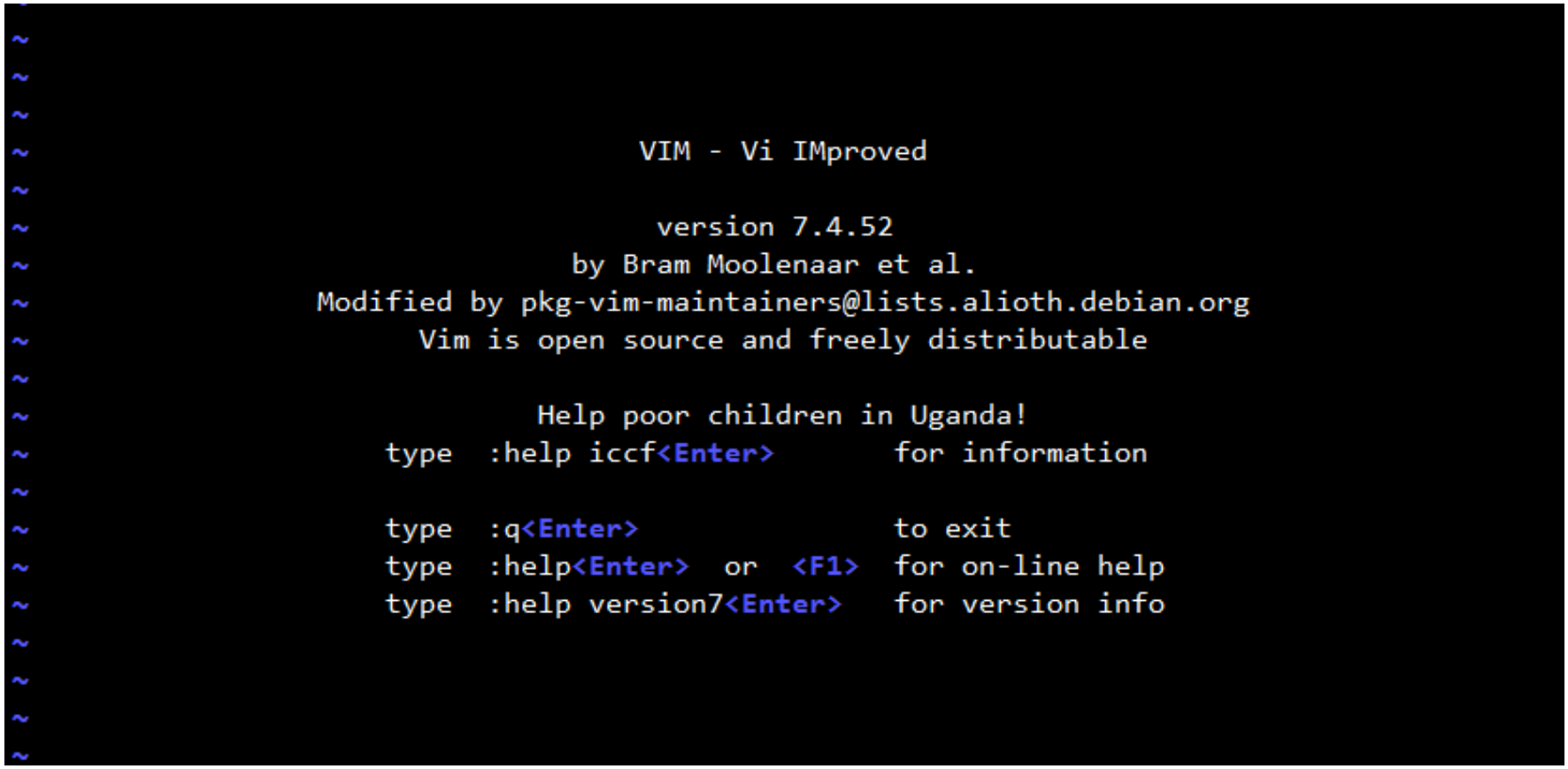
- Running many commands using a single line :
  $ [command1] ; [command2] ; [command3]

- Use the output of [command1] as an input of [command2] :
  $ [command1] | [command2]

- Save the printed output of command in a file :
  $ [command] > [file to save]

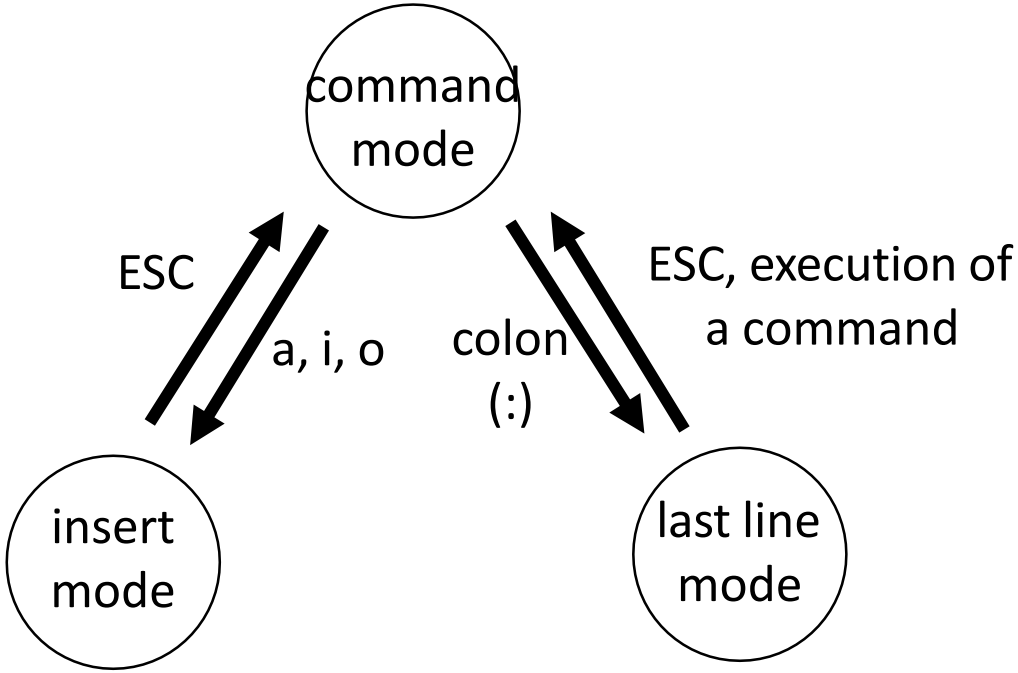
Vim

- vim
  - Vi IMproved, a programmer's text editor



Vim

- vim
  - Vi IMproved, a programmer's text editor
  - Three modes
    - Command mode
    - Insert mode
    - Last line mode

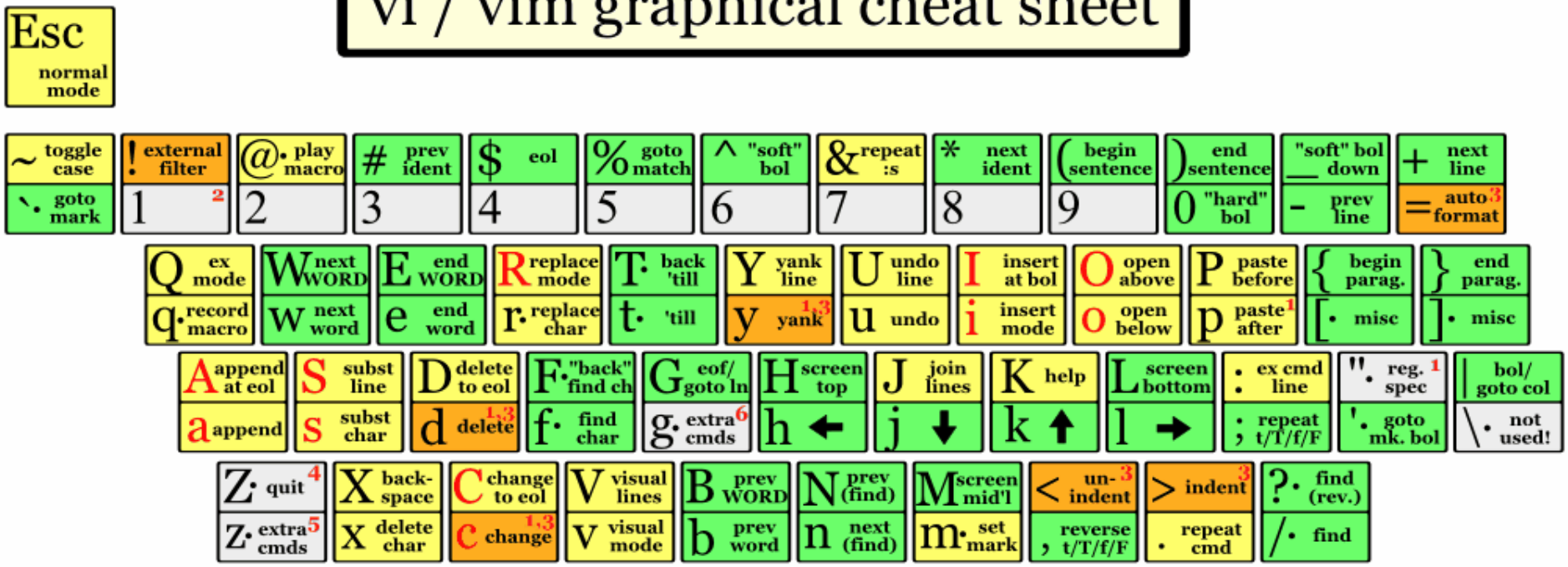


Vim

- vim
  - Vi IMproved, a programmer's text editor
  - Three modes
    - Command mode
    - Insert mode
    - Last line mode
  - Example
    - $ vi test.txt ← command mode
    - press "a" or "i" ← insert mode
    - input the contents
    - press "Esc" ← Return to command mode
    - :wq ← last line mode
      : save and quit

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

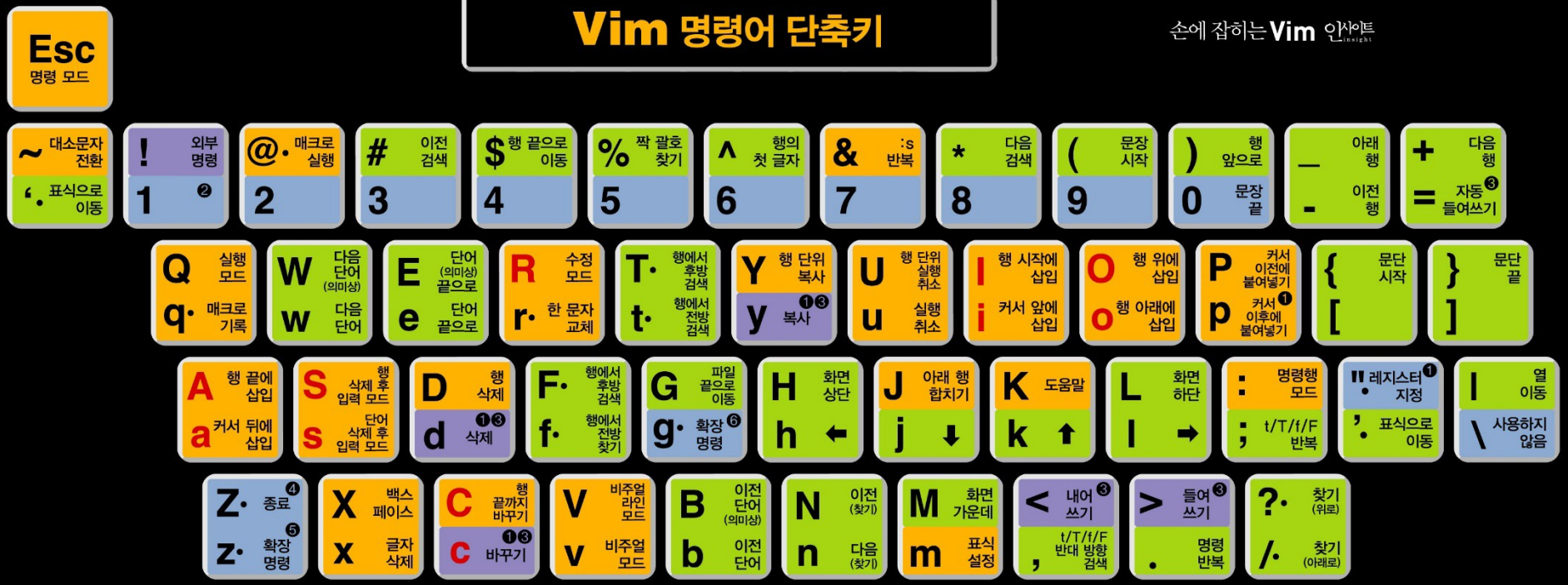
(5) zt: scroll cursor to top, zb: bottom, zz: center

(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

Vim 명령어 단축키

손에 잡히는 Vim 인사이트



동작 커서를 이동하거나, 연산자가 동작할 범위를 지정합니다.

명령 바로 동작하는 명령입니다. 빨간색은 입력 모드로 변경됩니다.

오퍼레이션 펜딩 모드 커서 위치부터 목적지까지를 대상으로 명령을 실행합니다.

확장 추가적인 키 입력이 필요합니다.

q· 입력 후 글자를 입력해야 합니다.

단어 공백 문자나 특수 문자로 구분된 단어
`test(123, 456, 789);`

단어 (의미상) 공백 문자로 구분된 단어
`test(123, 456, 789);`

• 명령행 모드의 주요 명령어

:w	저장
:q	종료
:q!	저장없이 종료
:e f	f 파일 열기
:%s/x/y/g	파일 전체에서 'x'를 'y'로 교체
:h name	name 명령에 대한 도움말
:new	새 파일

• 일반 모드의 주요 명령어

CTRL-R	재실행
CTRL-F/-B	페이지 위로/아래로
CTRL-E/-Y	스크롤 위로/아래로
CTRL-V	비주얼 모드

• 참고

❶ 복사/붙여넣기/지우기 명령을 사용하기 전에 "a를 입력하여 레지스터 a를 지정할 수 있습니다. (레지스터 이름은 a부터 z까지 사용 가능)
예를 들어 "ay$는 커서 위치부터 행 끝까지의 내용을 레지스터 a에 저장합니다.

❷ 명령어 입력 전 숫자를 지정하면, 해당 숫자만큼 명령어가 반복됩니다.

❸ 연속으로 입력하면, 현재 행에 반영됩니다.
예를 들어 dd는 현재 행이 지워집니다.

❹ ZZ는 저장 후 종료, ZQ는 저장 없이 종료입니다.

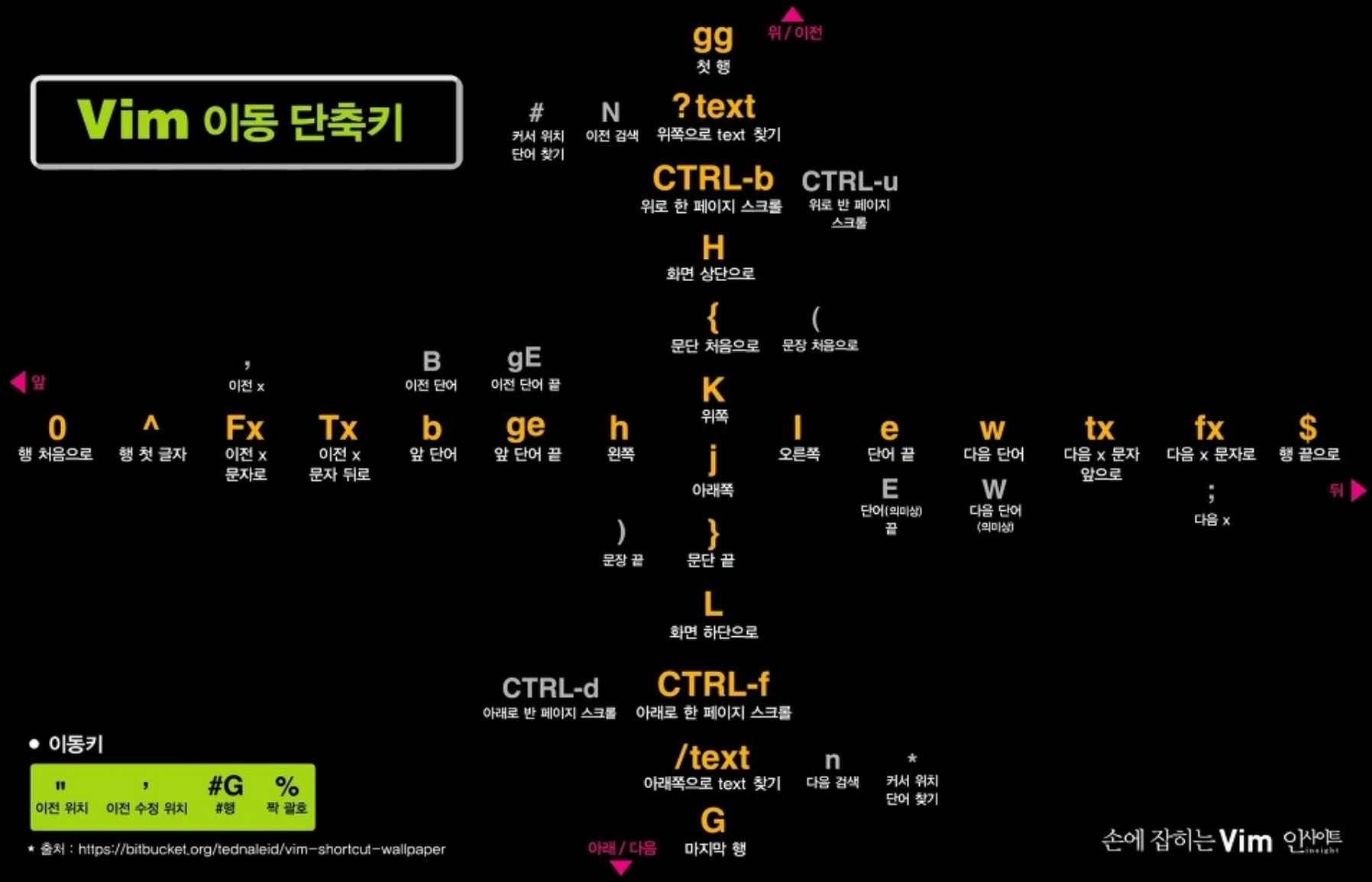
❺ zt는 커서가 위치한 부분을 화면 상단으로 스크롤합니다.
zb는 바닥으로, zz는 가운데로 스크롤합니다.

❻ gg는 커서를 파일 처음으로 이동합니다.
gf는 커서 위치의 파일명을 인식하여 파일을 엽니다.

★ 출처 : www.viemu.com ★ 번역 : Mr. Dust

http://www.insightbook.co.kr/post/872

Vim 이동 단축키



● 이동키

''	'	#G	%
이전 위치	이전 수정 위치	#행	짝 괄호

★ 출처 : https://bitbucket.org/tednaleid/vim-shortcut-wallpaper

손에 잡히는 Vim 인사이트

http://www.insightbook.co.kr/post/872

Vim

- Customization
  - see ~/.vimrc and vim plugins
  - you can see line numbers in Vim
  - auto indentation is available
  - filetype-aware auto completion is also available (snipmate plugin)

gcc

- gcc
  - gcc hello.c -E -o hello.i ⟵ preprocessed file
  - gcc hello.c -S -o hello.s ⟵ assembly file obj
  - gcc hello.c -c -o hello.o ⟵ ect file

make

- make
  - Make is a utility that automatically builds executable programs and libraries from source code by reading files called Makefiles (Wikipedia)

- Structure of Makefiles

```
target ... : prerequisites ...
        recipe
...
...
```

http://www.tecmint.com/a-brief-introduction-to-makefiles-in-open-source-software-development-with-gnu-make/

make

- A sample Makefile

```
$cat Makefile
final: main.o end.o inter.o start.o
        gcc -o final main.o end.o inter.o start.o
main.o: main.c global.h
        gcc -c main.c
end.o: end.c local.h global.h
        gcc c -c end.c
inter.o: inter.c global.h
        gcc c -c inter.c  s
tart.o: start.c global.h
        gcc -c start.c
clean:
        rm -f main.o end.o inter.o start.o
```

http://www.tecmint.com/a-brief-introduction-to-makefiles-in-open-source-software-development-with-gnu-make/

make

- A sample Makefile

```
$make
$ls | grep final
final
```

http://www.tecmint.com/a-brief-introduction-to-makefiles-in-open-source-software-development-with-gnu-make/

Self-study material

- Linux : https://www.youtube.com/watch?v=CpTfQ-q6MPU
- Vim : https://www.youtube.com/watch?v=ggSyF1SVFr4