Open Source Programming

Lecture-02

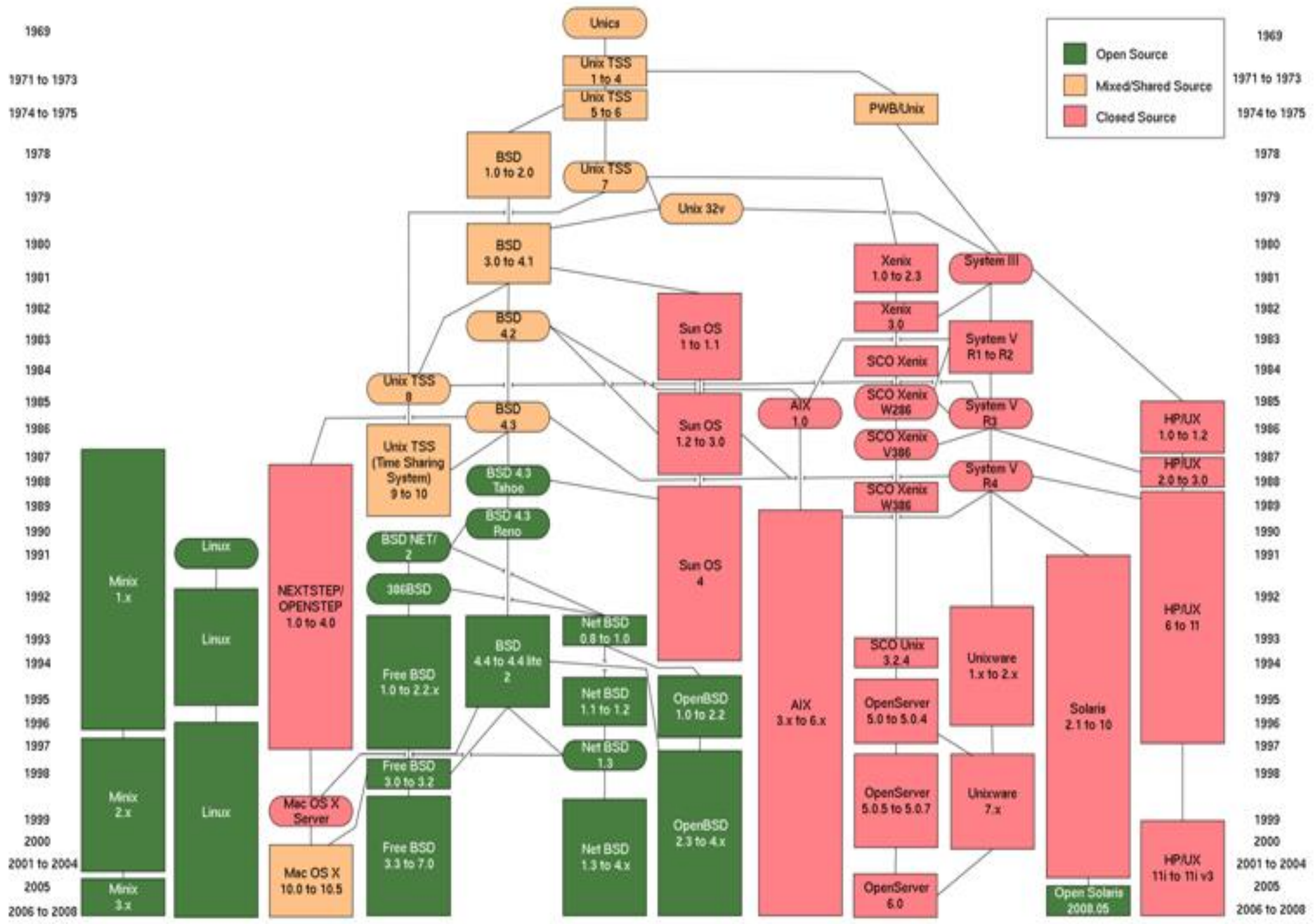
Linux Environment

UNIX

Unix 운영체제

- **1970년대 벨 연구소의 켄 톰슨, 데니스 리치가 개발**
- **다양한 버전**
  - 오늘날의 유닉스 시스템은 AT&T를 비롯한 여러 회사들과
  - 버클리대학(BSD 버전) 등 비영리 단체들이 개발한 다양한 버전
- **특징**
  - 다양한 종류의 시스템 사이에서 서로 **이식**할 수 있음
  - 다중 사용자 및 다중 작업을 지원
- **유닉스 시스템의 개념**
  - 일반 텍스트 파일
  - 명령줄 인터프리터
  - 계층적인 파일 시스템
  - 장치 및 특정한 형식의 프로세스 간 통신을 파일로 취급 등

Unix History



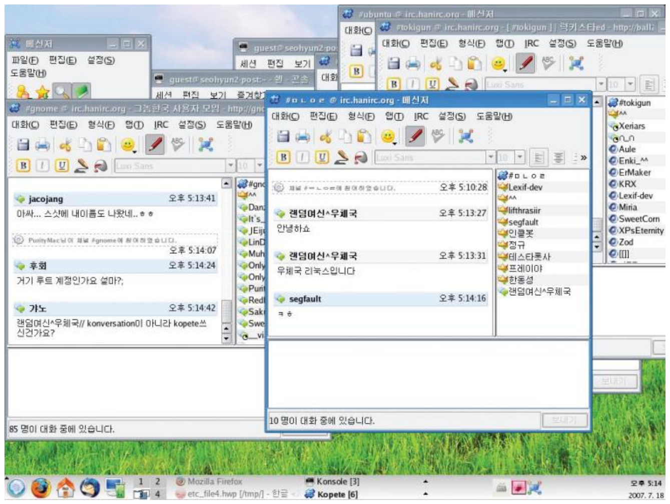
Unix 종류와 특성

- **Unix**
  - 1970년에 AT&T의 벨 연구소에서 개발한 운영체제로 처음에는 중형 컴퓨터에 사용하도록 고안
  - 여러 가지 유틸리티가 공개되면서 일반 사용자들에 확산
  - 특징
    - 다중 사용자, 다중 작업 처리 가능
    - 프로그램 개발이 쉬운 운영체제
    - 대부분 통신 서비스 프로그램은 Unix를 기반으로 하고 있음

```
Processes:  53 total, 2 running, 2 stuck, 49 sleeping... 300 threads
Load Avg:  0.12,  0.15,  0.17    CPU usage:  0.48% user,  0.96% sys, 98.55% idle      11:05:59
SharedLibs: num =    1, resident =   45M code, 4992K data, 4964K linkedit.
MemRegions: num =  7795, resident =  477M +  107M private,  262M shared.
PhysMem:  194M wired,  542M active,  270M inactive, 1006M used,   12M free.
VM: 12G + 140M   89770(1) pageins, 201400(0) pageouts

  PID COMMAND      %CPU   TIME    #TH #PRTS #MREGS RPRVT  RSHRD  RSIZE  VSIZE
  849 screencapt   0.0%  0:00.02   1    38     77  304K+ 2720K- 1916K   363M
  836 bash         0.0%  0:00.00   1    14     19  256K   624K   808K    74M
  835 login        0.0%  0:00.00   1    16     55  240K   260K   716K    75M
  817 top          3.6%  0:21.54   1    21     29 2832K   188K  2520K    74M
  814 iChatAgent   0.0%  0:00.21   2    73     73 1956K  2368K  5140K   344M
  778 bash         0.0%  0:00.01   1    14     19  240K   624K   808K    74M
  777 login        0.0%  0:00.01   1    16     55  240K   260K   716K    75M
  775 mdworker     0.0%  0:00.13   4    53     38  644K  2220K  1940K    87M
  762 Terminal     0.2%  0:13.04   7   109    105 4240K    11M    17M   419M
  749 Safari       0.0%  0:12.62  22   282    760   66M    31M    91M   585M
  744 iTunes       0.1%  0:26.49  15   311    505   45M    25M    75M   519M
  733 Front Row    0.1%  0:30.98  14   288    724  1438    18M   154M   627M
  732 mdworker     0.0%  0:00.45   5    76    182 1908K  3120K  4196K   188M
  306 Preview      0.0%  0:30.94  14   143    259 8596K  6944K    22M   448M
  240 launchd      0.0%  0:00.01   3    24     25   28K   272K   184K    74M
  211 ocspd        0.0%  0:00.01   1    19     21   40K   184K   240K    74M
  206 AppleFileS   0.0%  0:04.06   2    61     58  316K   192K  1200K    83M
  199 check_afp    0.0%  0:00.23   2    25     20   68K   176K   272K    74M
  154 Finder       0.0%  2:29.28  21   752    919  1618    49M   105M  1334M
  150 iCalAlarmS   0.0%  0:00.14   1    58     83  104K  2276K   996K   352M
  149 iTunesHelp   0.0%  0:00.09   1    49     42  124K  2112K   668K   346M
  142 ATSServer    0.0%  0:01.30   2    99     91 1244K  4104K  3368K   116M
  140 SystemUISe   0.0%  0:02.21   6   231    215 1272K  9952K  3364K   409M
```

Unix 종류와 특성

- **Linux**
  - 1991년 핀란드의 대학생이었던 리누스 토발즈(Linus Benedict Torvalds, 1969~)에 의해 만들어진 운영체제
  - 개인 컴퓨터용 UNIX에 주로 많이 사용
  - 특징
    - 프로그램 소스코드가 공개돼 있어 프로그래머가 원하는 대로 특정기능 추가 가능
    - 어느 플랫폼에도 포팅(porting)이 가능
    - 무료
  - 종류
    - 우분투(ubuntu), 레드햇(radhat), 데비안(debian) 등



#gnome @ irc.hanirc.org
대화(C) 편집(E) 형식(F) 탭(T) IRC 설정(S) 도움말(H)
#ㅁㄴㅇㄹ @ irc.hanirc.org - 메신저
85 명이 대화 중에 있습니다.
10 명이 대화 중에 있습니다.
Konsole [3]
Kopete [6]
Mozilla Firefox
2007. 7. 18

jacojang 오후 5:13:41
아싸... 스샷에 내이름도 나왔네.. ㅎㅎ

PurityMac님이 채널 #gnome에 참여하였습니다. 오후 5:14:07

후회 오후 5:14:24
거기 루트 계정인가요 설마?;

가노 오후 5:14:42
랜덤여신^우체국// konversation이 아니라 kopete쓰신건가요?

채널 #ㅁㄴㅇㄹ에 참여하였습니다. 오후 5:10:28

랜덤여신^우체국 오후 5:13:27
안녕하쇼

랜덤여신^우체국 오후 5:13:31
우체국 리눅스입니다

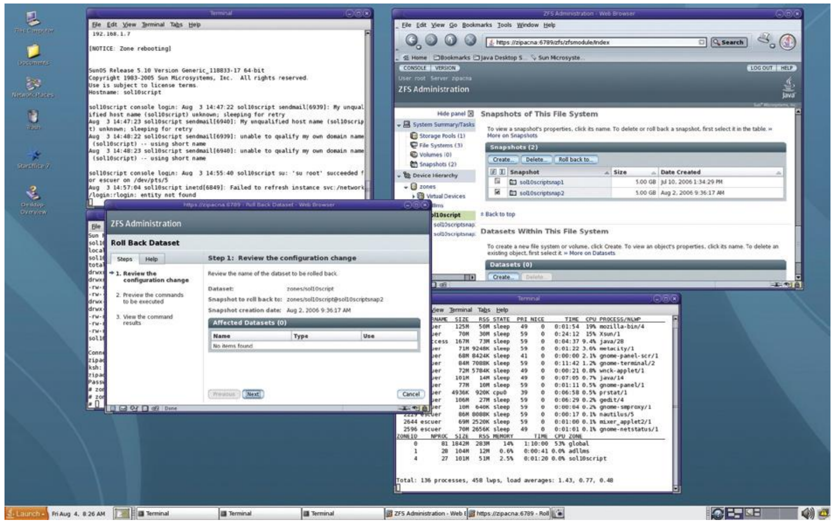
segfault 오후 5:14:16
ㅋㅎ

Unix 종류와 특성

- **Mac OS**
  - Apple 컴퓨터의 매킨토시 계열 개인용 컴퓨터나 워크스테이션용 운영체제로 개인용 컴퓨터에 GUI를 처음으로 도입
  - 1984년에 처음으로 세상에 선을 보인 이후로 거듭 발전
  - 특징
    - 문서편집이나 그래픽분야에서 많은 사랑을 받고 있음
    - Apple의 스마트폰인 iPhone, iPad와 디지털 미디어 재생기인 iPod Touch에도 내장되어 사용

Unix 종류와 특성

- **Solaris**
  - Unix 계열 운영체제 중 하나
  - Linux보다 훨씬 먼저 출시된 상용 운영체제
  - 특징
    - 처음에는 SUN에서 제작한 스팍(Sparc) CPU를 사용한 기종에서만 사용되는 전용 운영체제로 전문가들이 주로 사용
    - 인텔 아키텍처의 대량 보급으로 인하여 인텔용 Solaris도 출시 됨



```
192.168.1.7

[NOTICE: Zone rebooting]


SunOS Release 5.10 Version Generic_118833-17 64-bit
Copyright 1983-2005 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Hostname: sol10script

sol10script console login: Aug  3 14:47:22 sol10script sendmail[6939]: My unqual
ified host name (sol10script) unknown; sleeping for retry
Aug  3 14:47:23 sol10script sendmail[6940]: My unqualified host name (sol10scrip
t) unknown; sleeping for retry
Aug  3 14:48:22 sol10script sendmail[6939]: unable to qalify my own domain name
 (sol10script) -- using short name
Aug  3 14:48:23 sol10script sendmail[6940]: unable to qalify my own domain name
 (sol10script) -- using short name

sol10script console login: Aug  3 14:55:40 sol10script su: 'su root' succeeded f
or escuer on /dev/pts/5
Aug  3 14:57:04 sol10script inetd[6849]: Failed to refresh instance svc:/network
/login:rlogin: entity not found
```

ZFS Administration

User: root Server: zipacna

Snapshots of This File System

To view a snapshot's properties, click its name. To delete or roll back a snapshot, first select it in the table. » More on Snapshots

Snapshots (2)

	Snapshot	Size	Date Created
	sol10scriptsnap1	5.00 GB	Jul 10, 2006 1:34:29 PM
✓	sol10scriptsnap2	5.00 GB	Aug 2, 2006 9:36:17 AM

Datasets Within This File System

To create a new file system or volume, click Create. To view an object's properties, click its name. To delete an existing object, first select it. » More on Datasets

Datasets (0)

ZFS Administration

Roll Back Dataset

Step 1: Review the configuration change

1. Review the configuration change
2. Preview the commands to be executed
3. View the command results

Review the name of the dataset to be rolled back.

Dataset: zones/sol10script

Snapshot to roll back to: zones/sol10script@sol10scriptsnap2

Snapshot creation date: Aug 2, 2006 9:36:17 AM

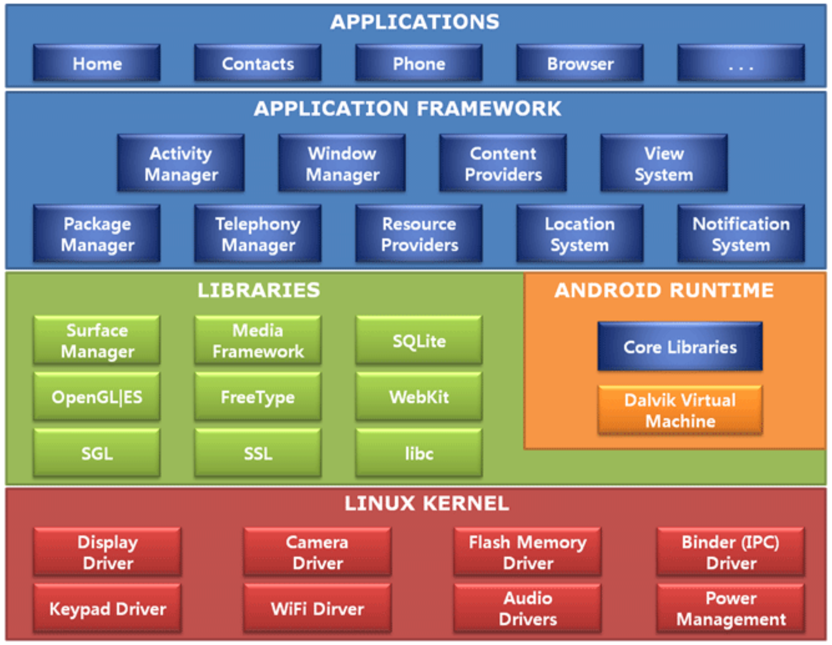
Affected Datasets (0)

Name	Type	Use
No items found		

```
RNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
uer    125M   50M sleep   49    0   0:01:54  19% mozilla-bin/4
uer     70M   30M sleep   59    0   0:24:12  15% Xsun/1
ccess  167M   73M sleep   59    0   0:04:37 9.4% java/28
uer     71M 9248K sleep   59    0   0:01:22 3.6% metacity/1
uer     68M 8424K sleep   41    0   0:00:00 2.1% gnome-panel-scr/1
uer     84M 7088K sleep   59    0   0:11:42 1.2% gnome-terminal/2
uer     72M 5784K sleep   49    0   0:00:21 0.8% wnck-applet/1
uer    101M   14M sleep   49    0   0:07:05 0.7% java/14
uer     77M   10M sleep   59    0   0:01:11 0.5% gnome-panel/1
uer   4936K  920K cpu0    39    0   0:06:58 0.5% prstat/1
uer    106M   27M sleep   59    0   0:06:29 0.2% gedit/4
uer     10M  640K sleep   59    0   0:00:04 0.2% gnome-smproxy/1
 escuer    86M 8088K sleep   59    0   0:00:17 0.1% nautilus/5
 2644 escuer    69M 2520K sleep   59    0   0:01:00 0.1% mixer_applet2/1
 2596 escuer    70M 2656K sleep   49    0   0:01:01 0.1% gnome-netstatus/1
ZONEID    NPROC  SIZE   RSS MEMORY      TIME  CPU ZONE
     0       81 1842M  283M    14%   1:10:00  53% global
     1       20  104M   12M   0.6%   0:00:41 0.0% adllms
     4       27  101M   51M   2.5%   0:01:20 0.0% sol10script


Total: 136 processes, 458 lwps, load averages: 1.43, 0.77, 0.48
```

Unix 종류와 특성

- **모바일 운영체제**
  - 현재 세계시장에 공급되는 범용으로 사용되는 모바일 운영체제
    - Windows Mobile, Symbian, Falm, BlackBerry, Linux 등
  - 모바일 운영체제들의 특징
    - MS사의 Windows Mobile과 Nokia의 Symbian 등은 사용하는데 편리함
    - Apple, MS 등은 좀 더 많은 개발자와 사용자들에게 API를 제공함
    - Linux는 아예 모든 것을 공개
  - Linux 기반 모바일 운영체제
    - 리모(LiMo, Linux Mobile)
    - 삼성 바다 (POSIX 리눅스 계열)
    - **안드로이드(Android)**



GNU

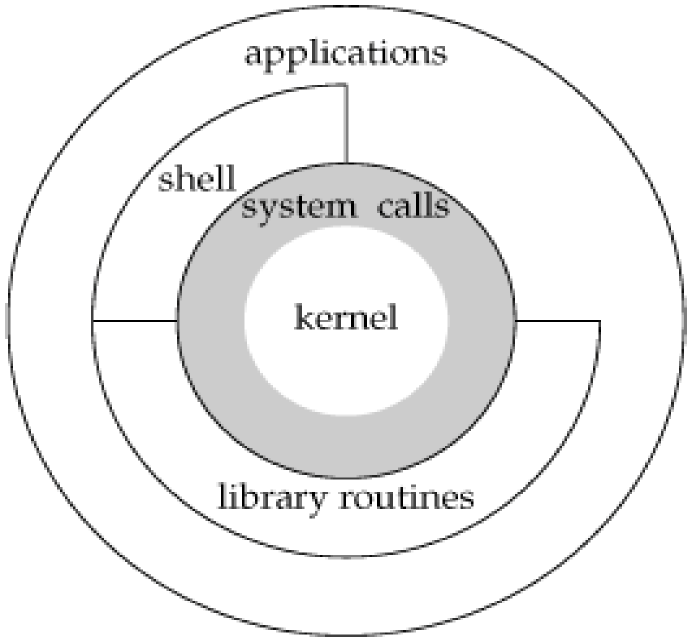
- **GNU project : 대표적인 open(free) source project**
  - Richard Matthew Stallman 에 의해 창립
  - 1970년 이후, 컴퓨터 사용자가 많아지면서 상업용 목적 소프트웨어들의 라이센스 비용에 반대
  - 자유 소프트웨어 : 누구나 소프트웨어의 복사나 사용, 연구, 수정 배포 등에 제한이 없음

- **GNU : 'GNU is Not Unix'**
  - unix의 기능을 포함하고 있지만, unix code를 포함하지 않음
  - 커널 부분을 제외한 모든 주요 구성 요소를 개발
  - 이후, 리누스 토발즈가 리눅스 커널을 개발하여 발표

- **License 정책**
  - GPL (GNU General Public License)
  - LGPL (GNU Lesser GHeneral Public License)

The System Architecture of UNIX

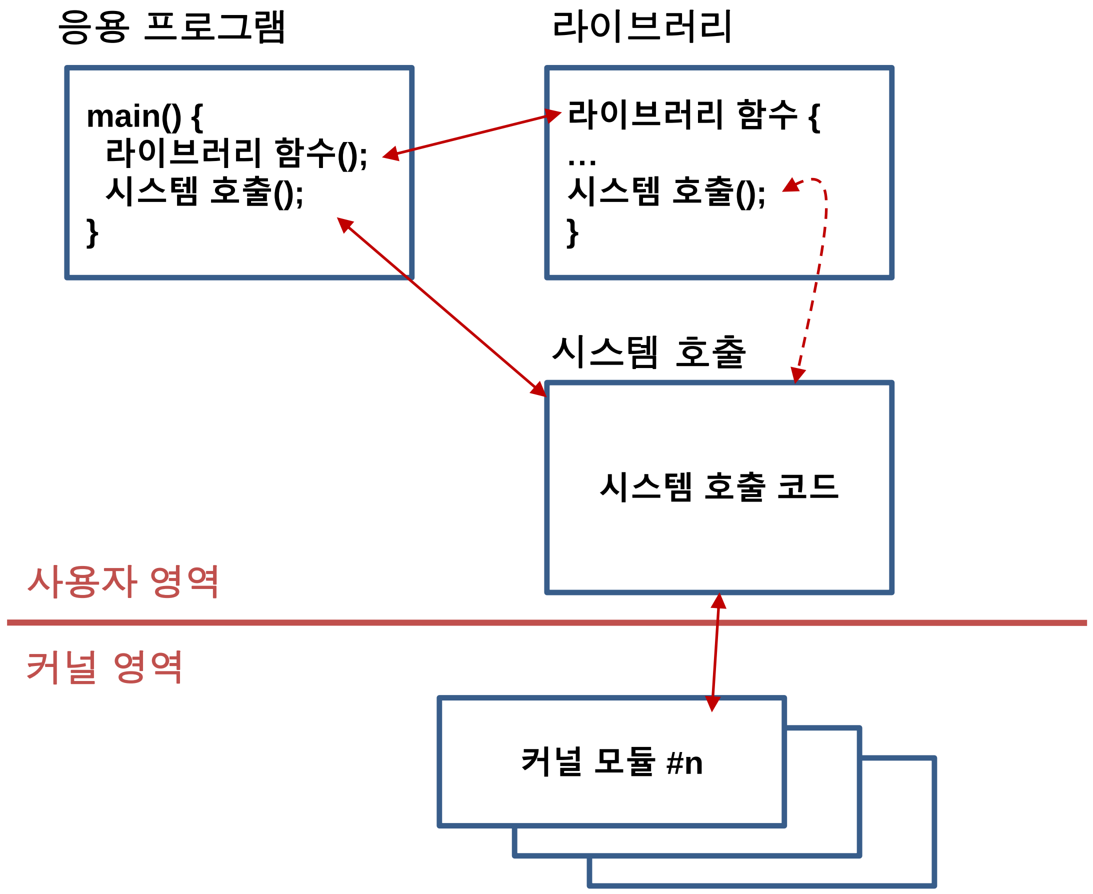
- **Hardware**
  - CPU, Memory, Disk, Peripherals
- **Kernel**
  - Process management
  - File management
  - Memory management
  - Device management



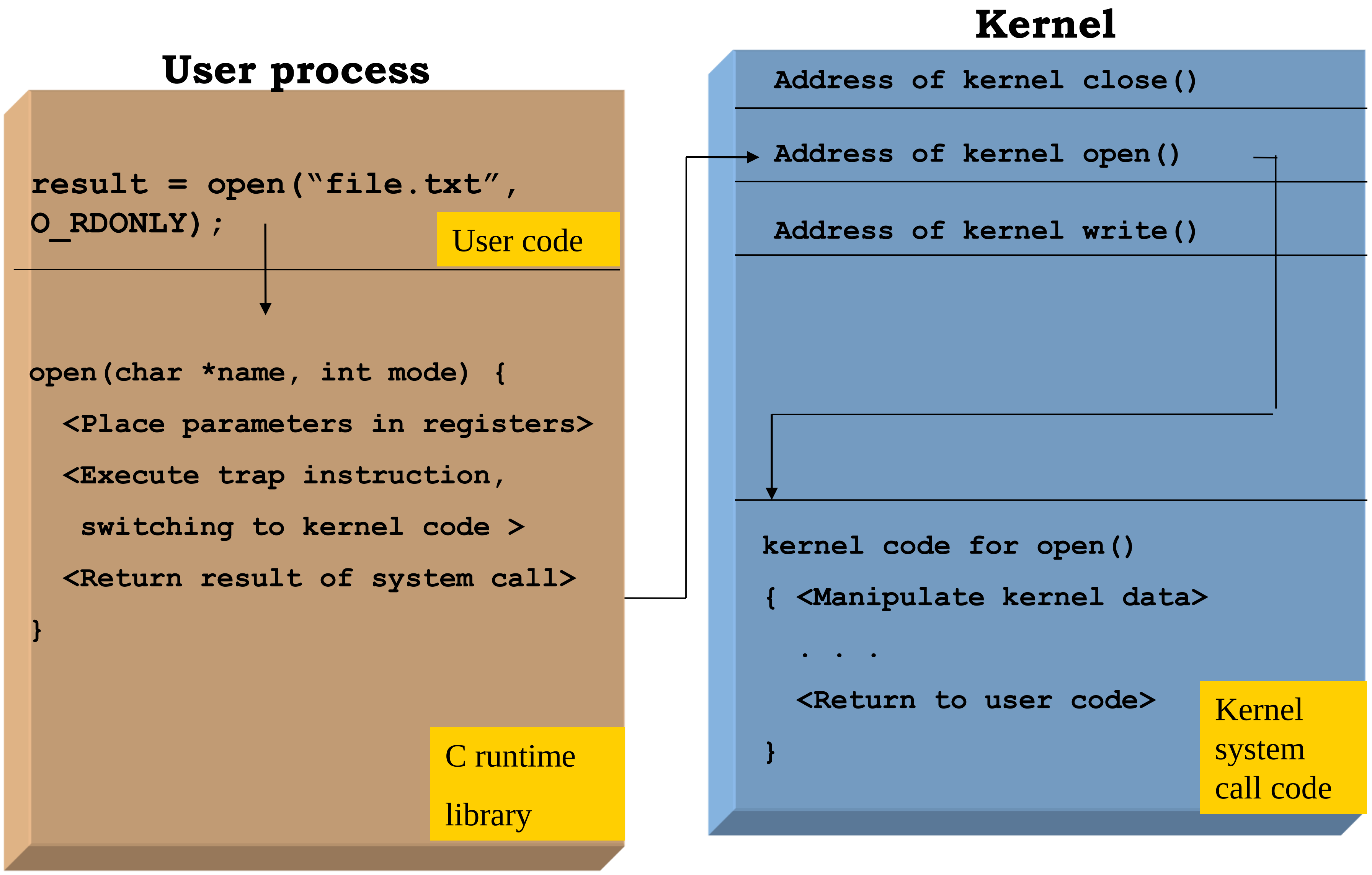
- **System call**
  - the programmer's functional interface to the UNIX kernel
- **Commands, Utilities, Application programs**
  - kernel services using library routines or
  - system calls

System Call

- Application programs talk to the OS via system calls
- Programmer's functional interface to the UNIX kernel



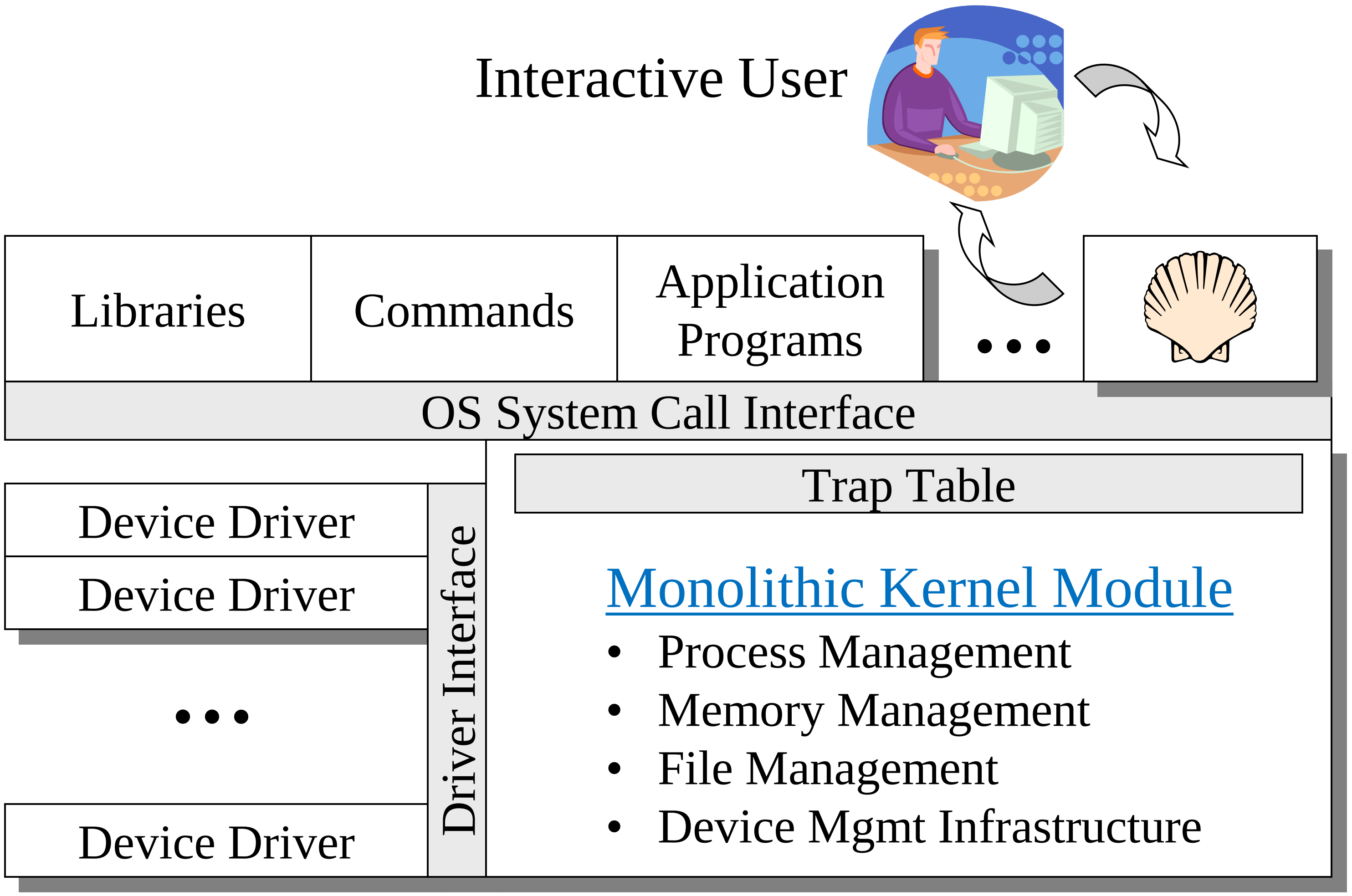
User Mode / Kernel Mode



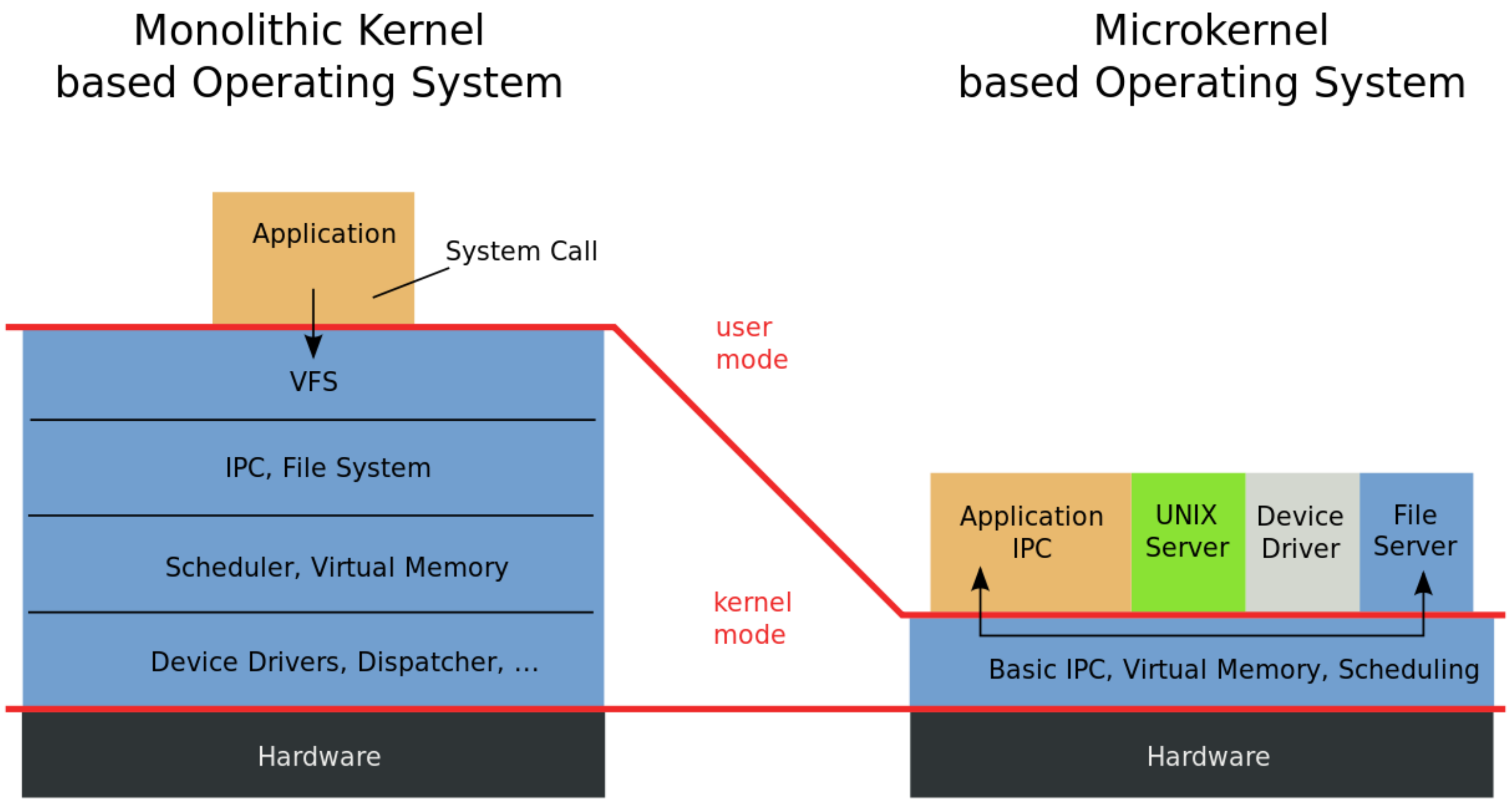
```
result = open("file.txt",
O_RDONLY);

open(char *name, int mode) {
  <Place parameters in registers>
  <Execute trap instruction,
   switching to kernel code >
  <Return result of system call>
}
```

```
Address of kernel close()
Address of kernel open()
Address of kernel write()

kernel code for open()
{ <Manipulate kernel data>
    . . .
  <Return to user code>
}
```

The UNIX Architecture



Monolithic Kernel vs. Microkernel



e.g. Windows, UNIX, etc.

e.g. MINIX, Mach

Commands

Working with Directories

- Unix organizes files into a tree-structured directory system.

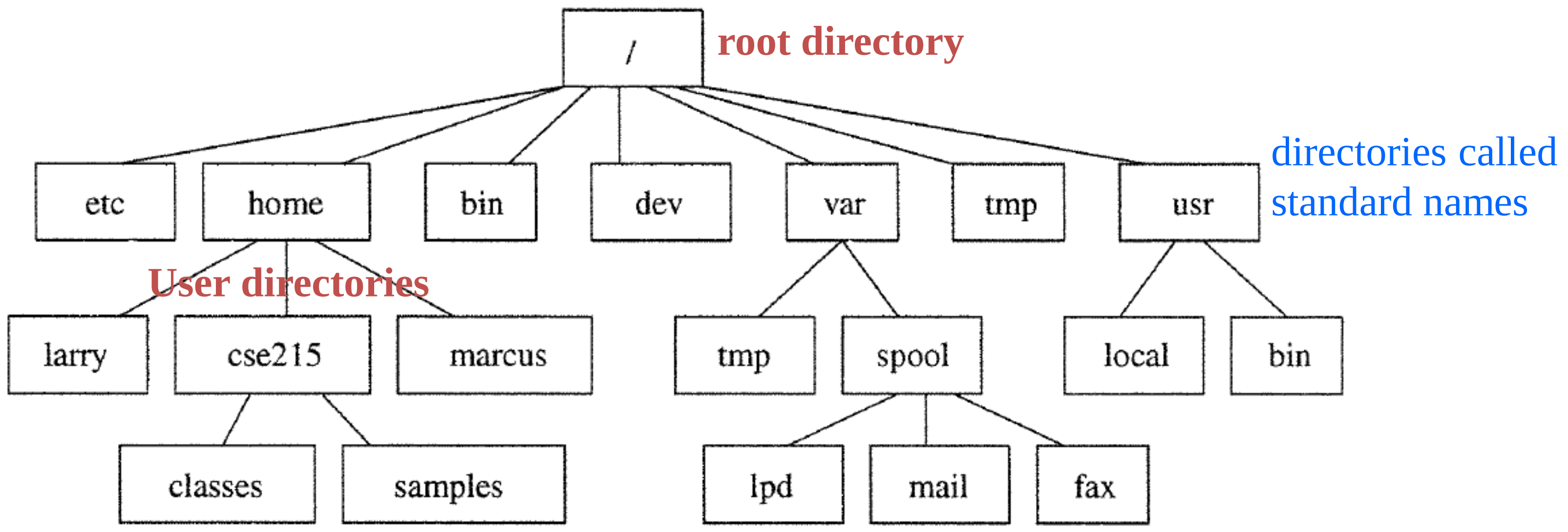
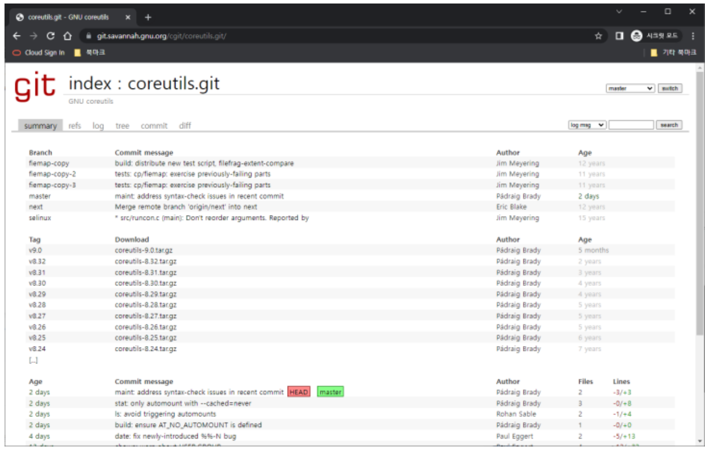


FIGURE 1.8

Part of the directory tree.

- GNU core utilities
  - https://www.gnu.org/software/coreutils/
  - https://git.savannah.gnu.org/cgit/coreutils.git/



git index : coreutils.git

GNU coreutils

Branch	Commit message	Author	Age
fiemap-copy	build: distribute new test script, filefrag-extent-compare	Jim Meyering	12 years
fiemap-copy-2	tests: cp/fiemap: exercise previously-failing parts	Jim Meyering	11 years
fiemap-copy-3	tests: cp/fiemap: exercise previously-failing parts	Jim Meyering	11 years
master	maint: address syntax-check issues in recent commit	Pádraig Brady	2 days
next	Merge remote branch 'origin/next' into next	Eric Blake	12 years
selinux	* src/runcon.c (main): Don't reorder arguments. Reported by	Jim Meyering	15 years

Tag	Download	Author	Age
v9.0	coreutils-9.0.tar.gz	Pádraig Brady	5 months
v8.32	coreutils-8.32.tar.gz	Pádraig Brady	2 years
v8.31	coreutils-8.31.tar.gz	Pádraig Brady	3 years
v8.30	coreutils-8.30.tar.gz	Pádraig Brady	4 years
v8.29	coreutils-8.29.tar.gz	Pádraig Brady	4 years
v8.28	coreutils-8.28.tar.gz	Pádraig Brady	5 years
v8.27	coreutils-8.27.tar.gz	Pádraig Brady	5 years
v8.26	coreutils-8.26.tar.gz	Pádraig Brady	5 years
v8.25	coreutils-8.25.tar.gz	Pádraig Brady	6 years
v8.24	coreutils-8.24.tar.gz	Pádraig Brady	7 years
[...]			

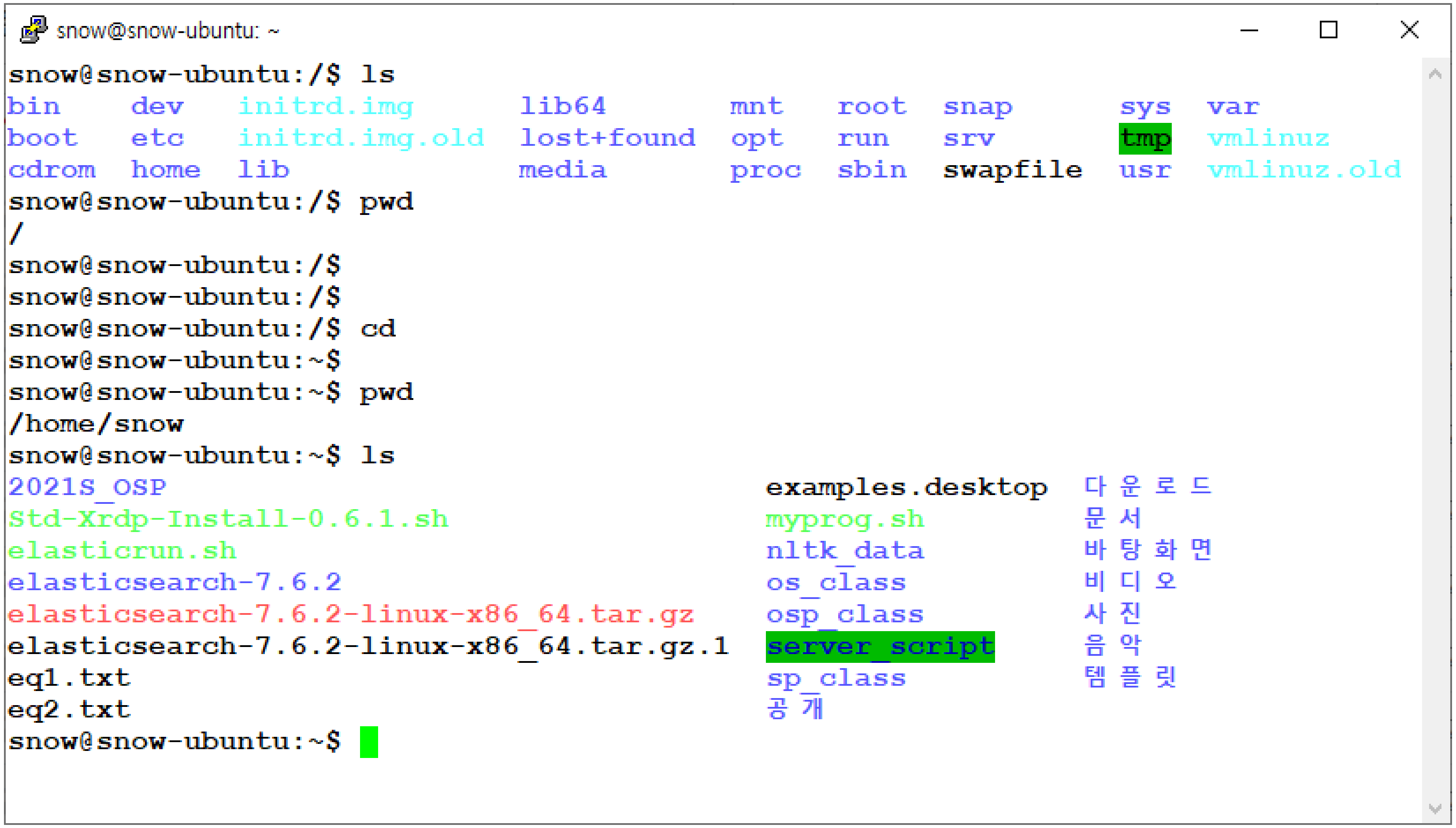
Age	Commit message	Author	Files	Lines
2 days	maint: address syntax-check issues in recent commit HEAD master	Pádraig Brady	2	-3/+3
2 days	stat: only automount with --cached=never	Pádraig Brady	3	-0/+8
2 days	ls: avoid triggering automounts	Rohan Sable	2	-1/+4
2 days	build: ensure AT_NO_AUTOMOUNT is defined	Pádraig Brady	1	-0/+0
4 days	date: fix newly-introduced %%-N bug	Paul Eggert	2	-5/+13

Working with Files

- **Users' personal files**
  - *stored in and below their home directories.*
- **The system files**
  - *stored in system directories.*
- **Filenames**
  - up to about 250 characters.
  - may contain any character except for the "/"

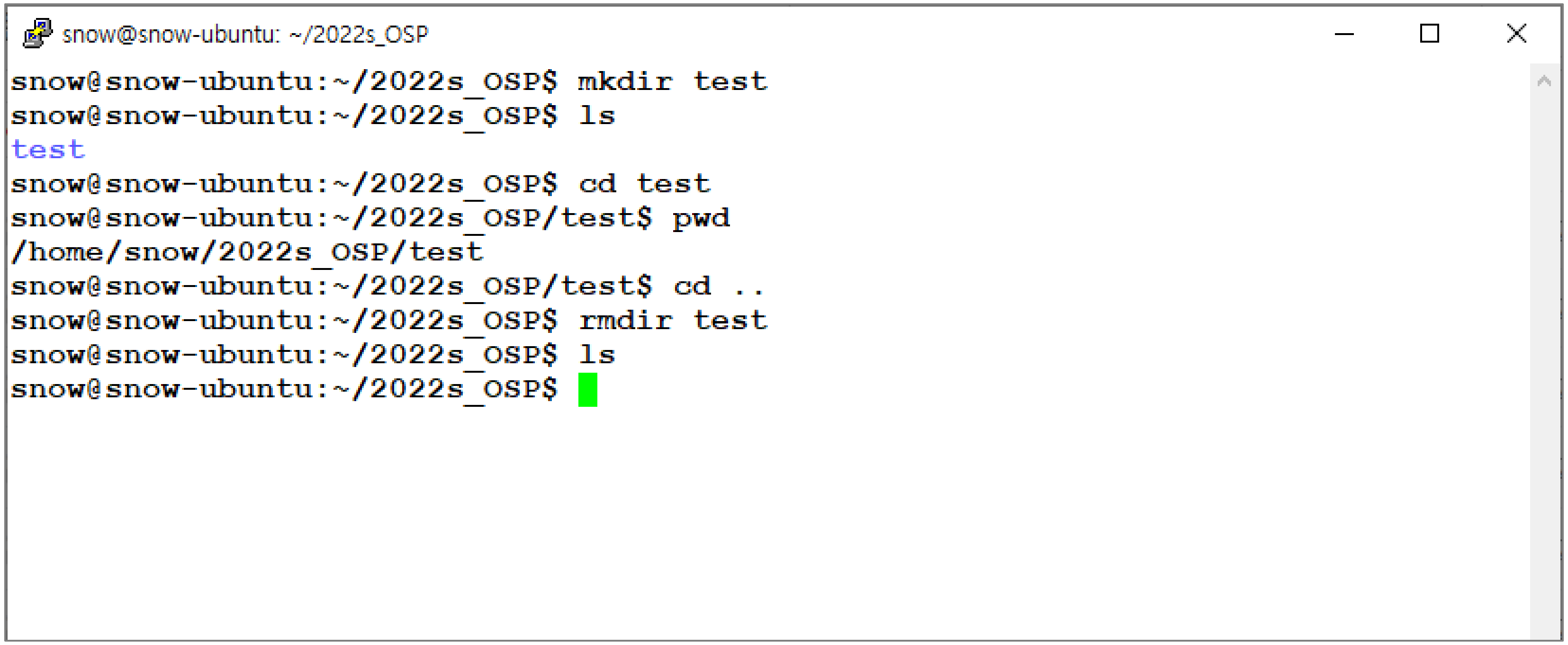
Commands for Working with Directories

- `ls` - list directory contents
- `pwd` - print path to current directory



```
snow@snow-ubuntu:/$ ls
bin    dev   initrd.img      lib64       mnt   root  snap      sys  var
boot   etc   initrd.img.old  lost+found  opt   run   srv       tmp  vmlinuz
cdrom  home  lib             media       proc  sbin  swapfile  usr  vmlinuz.old
snow@snow-ubuntu:/$ pwd
/
snow@snow-ubuntu:/$
snow@snow-ubuntu:/$
snow@snow-ubuntu:/$ cd
snow@snow-ubuntu:~$
snow@snow-ubuntu:~$ pwd
/home/snow
snow@snow-ubuntu:~$ ls
2021S_OSP                                  examples.desktop  다운로드
Std-Xrdp-Install-0.6.1.sh                  myprog.sh         문서
elasticrun.sh                              nltk_data         바탕화면
elasticsearch-7.6.2                        os_class          비디오
elasticsearch-7.6.2-linux-x86_64.tar.gz    osp_class         사진
elasticsearch-7.6.2-linux-x86_64.tar.gz.1  server_script     음악
eq1.txt                                    sp_class          템플릿
eq2.txt                                    공개
snow@snow-ubuntu:~$
```

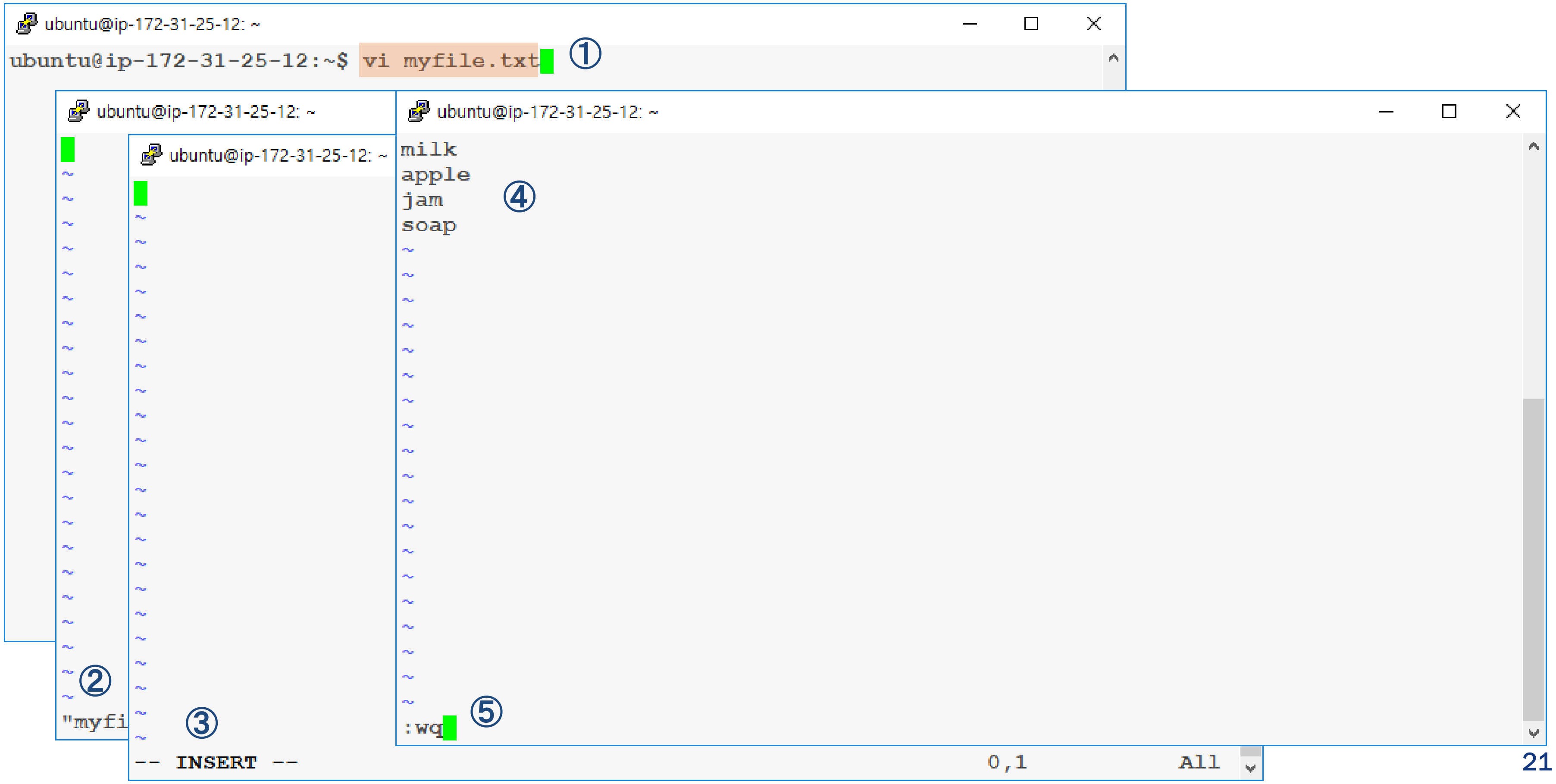
- `mkdir`, `rmdir` – **make and remove directories**
- `cd` – **change to a different directory**
  - $ cd /test
  - $ cd ..   Move up one level
  - $ cd   Move to user's home directory



```
snow@snow-ubuntu:~/2022s_OSP$ mkdir test
snow@snow-ubuntu:~/2022s_OSP$ ls
test
snow@snow-ubuntu:~/2022s_OSP$ cd test
snow@snow-ubuntu:~/2022s_OSP/test$ pwd
/home/snow/2022s_OSP/test
snow@snow-ubuntu:~/2022s_OSP/test$ cd ..
snow@snow-ubuntu:~/2022s_OSP$ rmdir test
snow@snow-ubuntu:~/2022s_OSP$ ls
snow@snow-ubuntu:~/2022s_OSP$ 
```

Commands for Working with Files

- **vi filename – create & edit file**
  - i – input mode
  - esc – command mode
  - :wq – save file



```
ubuntu@ip-172-31-25-12: ~
ubuntu@ip-172-31-25-12:~$ vi myfile.txt
```
①

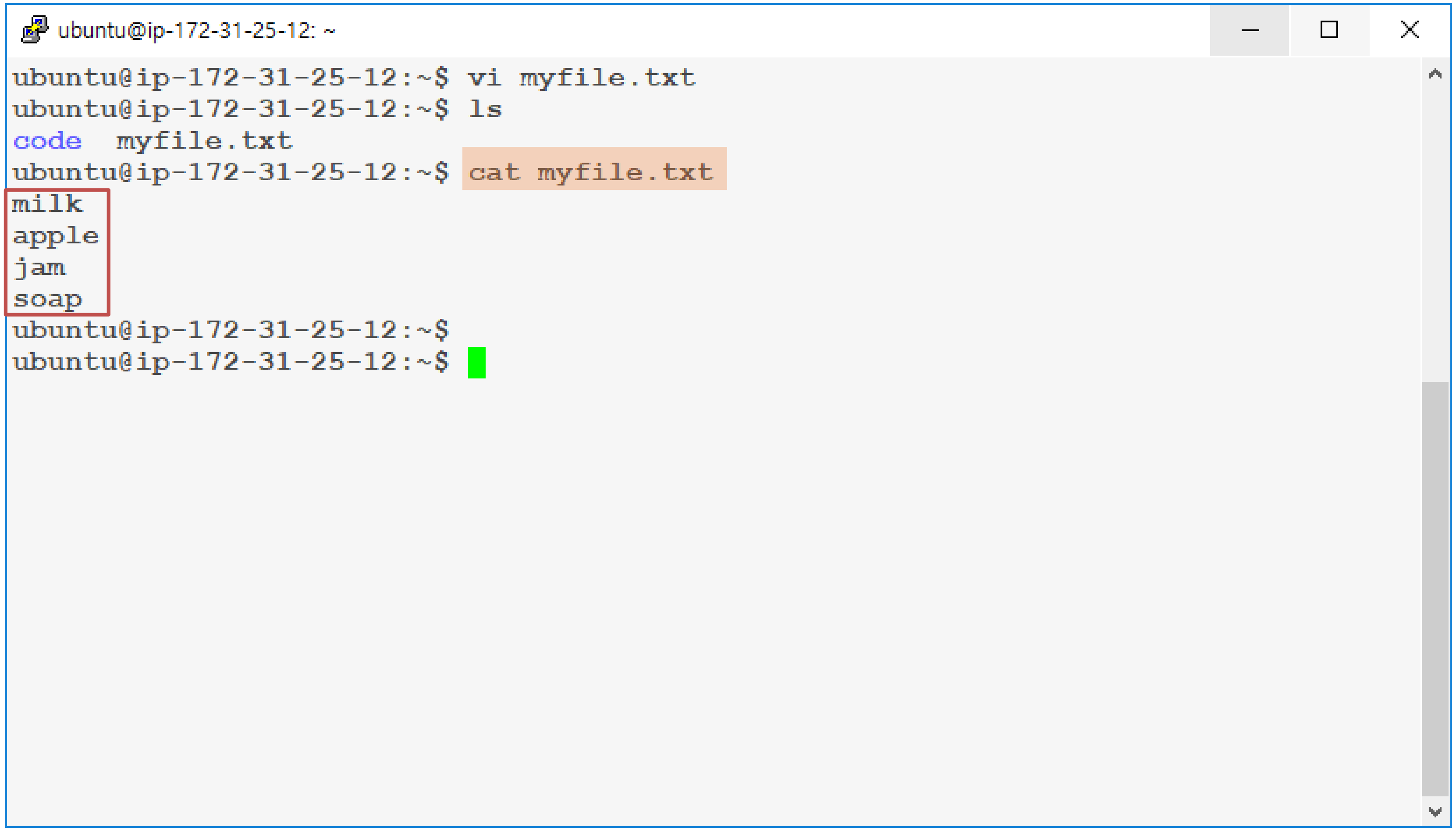
```
~
"myfi
```
②

```
-- INSERT --                                                    0,1          All
```
③

```
milk
apple
jam
soap
~
```
④

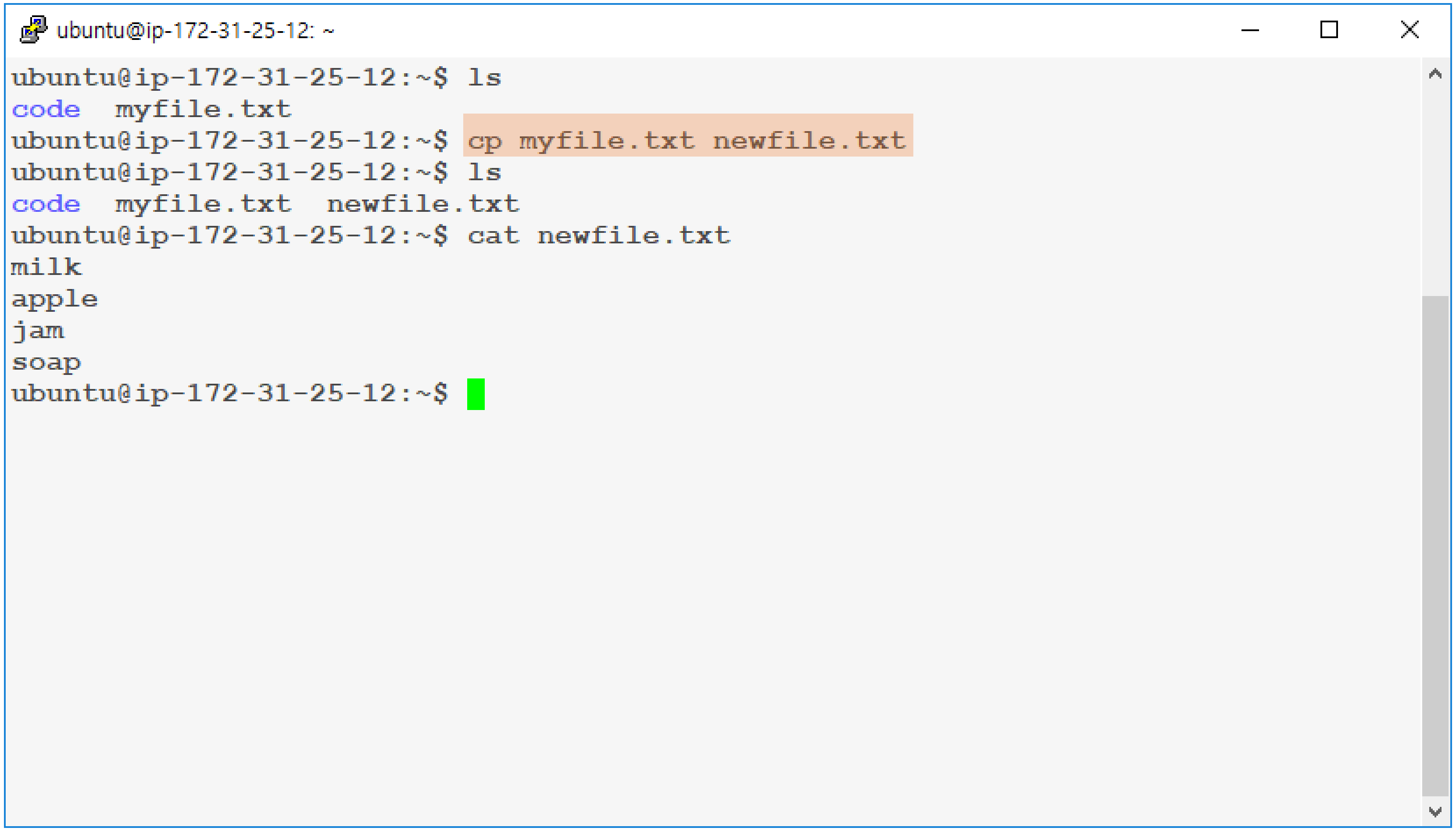
```
:wq
```
⑤

◆ `cat` – examine file contents



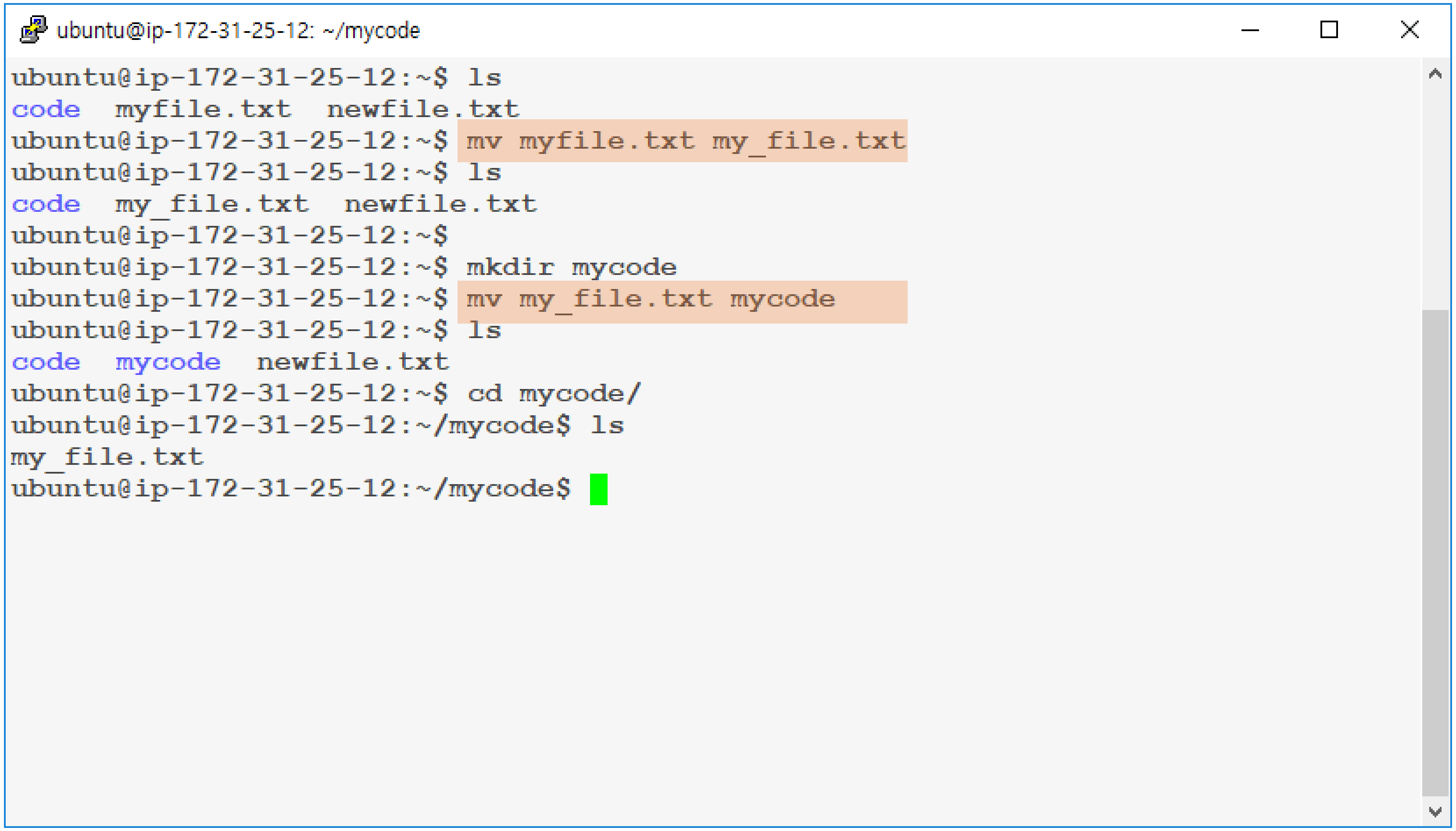
```
ubuntu@ip-172-31-25-12:~$ vi myfile.txt
ubuntu@ip-172-31-25-12:~$ ls
code  myfile.txt
ubuntu@ip-172-31-25-12:~$ cat myfile.txt
milk
apple
jam
soap
ubuntu@ip-172-31-25-12:~$
ubuntu@ip-172-31-25-12:~$
```

◆ cp – make a copy of a file



```
ubuntu@ip-172-31-25-12:~$ ls
code  myfile.txt
ubuntu@ip-172-31-25-12:~$ cp myfile.txt newfile.txt
ubuntu@ip-172-31-25-12:~$ ls
code  myfile.txt  newfile.txt
ubuntu@ip-172-31-25-12:~$ cat newfile.txt
milk
apple
jam
soap
ubuntu@ip-172-31-25-12:~$
```

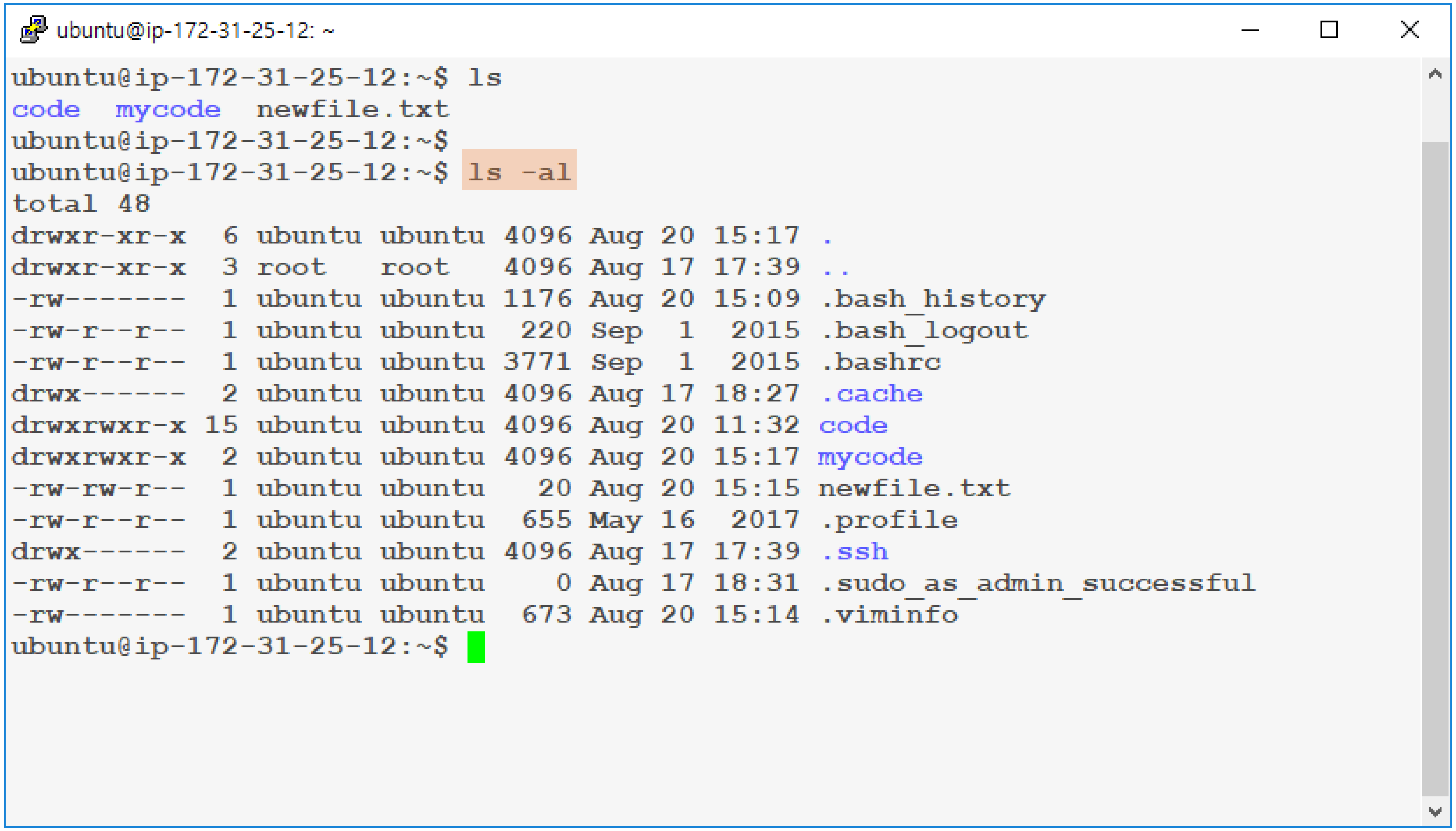
mv – rename or move a file


ubuntu@ip-172-31-25-12: ~/mycode

```
ubuntu@ip-172-31-25-12:~$ ls
code   myfile.txt   newfile.txt
ubuntu@ip-172-31-25-12:~$ mv myfile.txt my_file.txt
ubuntu@ip-172-31-25-12:~$ ls
code   my_file.txt   newfile.txt
ubuntu@ip-172-31-25-12:~$
ubuntu@ip-172-31-25-12:~$ mkdir mycode
ubuntu@ip-172-31-25-12:~$ mv my_file.txt mycode
ubuntu@ip-172-31-25-12:~$ ls
code   mycode   newfile.txt
ubuntu@ip-172-31-25-12:~$ cd mycode/
ubuntu@ip-172-31-25-12:~/mycode$ ls
my_file.txt
ubuntu@ip-172-31-25-12:~/mycode$
```

File Permission Attributes

- `ls -al` **shows attributes of a file:**



ubuntu@ip-172-31-25-12: ~

```
ubuntu@ip-172-31-25-12:~$ ls
code  mycode  newfile.txt
ubuntu@ip-172-31-25-12:~$
ubuntu@ip-172-31-25-12:~$ ls -al
total 48
drwxr-xr-x  6 ubuntu ubuntu 4096 Aug 20 15:17 .
drwxr-xr-x  3 root   root   4096 Aug 17 17:39 ..
-rw-------  1 ubuntu ubuntu 1176 Aug 20 15:09 .bash_history
-rw-r--r--  1 ubuntu ubuntu  220 Sep  1  2015 .bash_logout
-rw-r--r--  1 ubuntu ubuntu 3771 Sep  1  2015 .bashrc
drwx------  2 ubuntu ubuntu 4096 Aug 17 18:27 .cache
drwxrwxr-x 15 ubuntu ubuntu 4096 Aug 20 11:32 code
drwxrwxr-x  2 ubuntu ubuntu 4096 Aug 20 15:17 mycode
-rw-rw-r--  1 ubuntu ubuntu   20 Aug 20 15:15 newfile.txt
-rw-r--r--  1 ubuntu ubuntu  655 May 16  2017 .profile
drwx------  2 ubuntu ubuntu 4096 Aug 17 17:39 .ssh
-rw-r--r--  1 ubuntu ubuntu    0 Aug 17 18:31 .sudo_as_admin_successful
-rw-------  1 ubuntu ubuntu  673 Aug 20 15:14 .viminfo
ubuntu@ip-172-31-25-12:~$
```

```
drwxrwxr-x  2 ubuntu ubuntu 4096 Aug 20 15:17 mycode
-rw-rw-r--  1 ubuntu ubuntu   20 Aug 20 15:15 newfile.txt
```

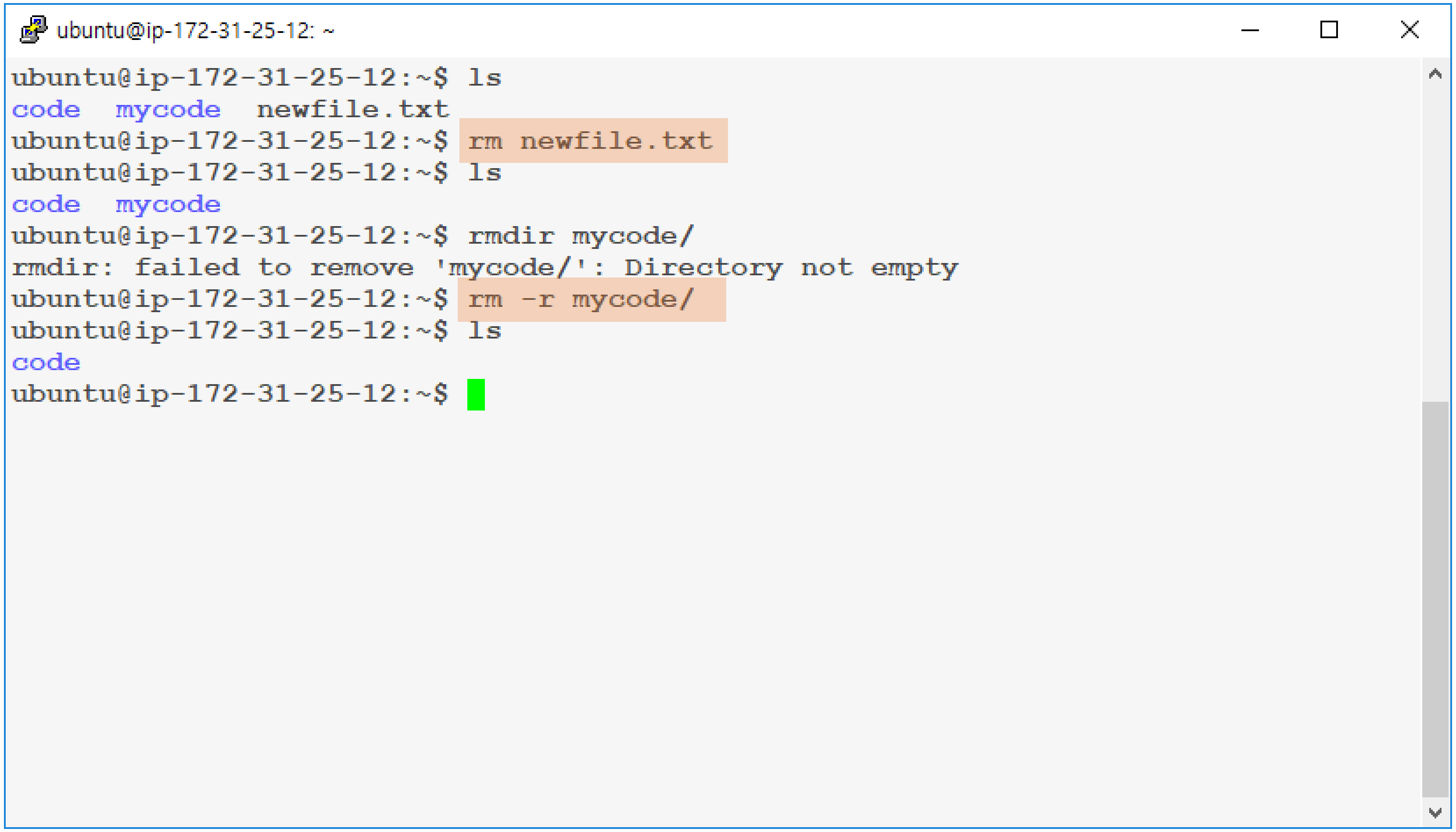
```
file          user       group   size   modification
permission    name       name           date and time
attributes    (owner/
              creater)
```

- **three groups of file permission attributes**

```
- r w x    r w x    r w x        r: read, w: write, x:execute
  user     group    other
```

- **rm filename – delete a file**
- **rm -r dirname – delete files & directory**

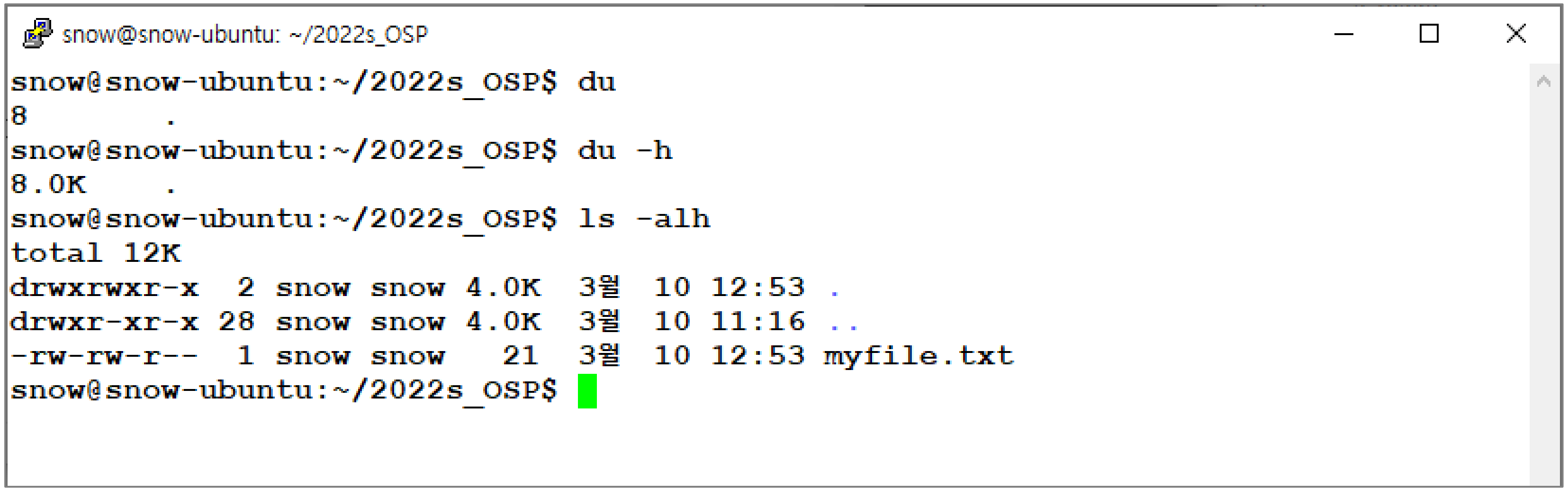
※ You have to delete or move the contents of a directory before you can delete it.



```
ubuntu@ip-172-31-25-12:~$ ls
code  mycode  newfile.txt
ubuntu@ip-172-31-25-12:~$ rm newfile.txt
ubuntu@ip-172-31-25-12:~$ ls
code  mycode
ubuntu@ip-172-31-25-12:~$ rmdir mycode/
rmdir: failed to remove 'mycode/': Directory not empty
ubuntu@ip-172-31-25-12:~$ rm -r mycode/
ubuntu@ip-172-31-25-12:~$ ls
code
ubuntu@ip-172-31-25-12:~$
```

Tree Commands

- **du (du -h)**
  - Stands for **disk usage.**
  - Reports the number of disk blocks used by a directory, the files it contains, and all the directories and files below it.



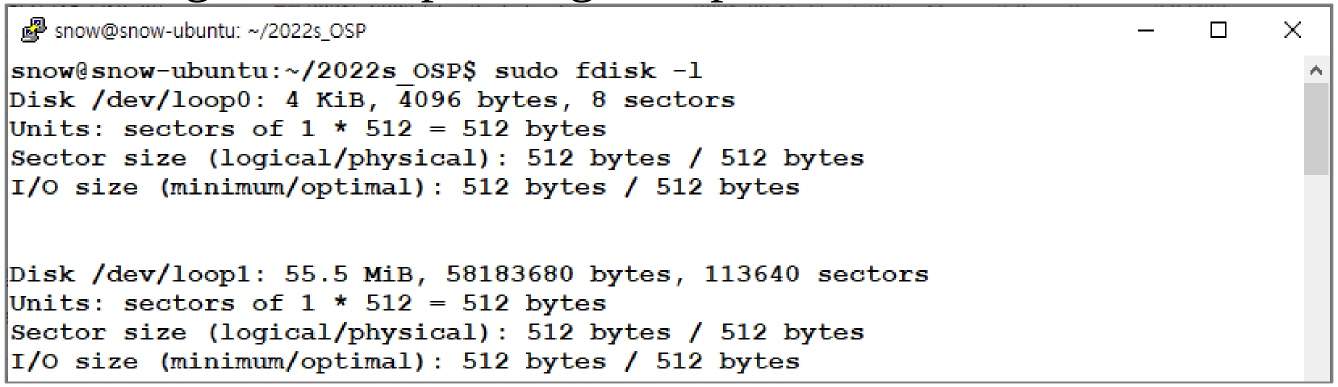
snow@snow-ubuntu: ~/2022s_OSP

```
snow@snow-ubuntu:~/2022s_OSP$ du
8       .
snow@snow-ubuntu:~/2022s_OSP$ du -h
8.0K    .
snow@snow-ubuntu:~/2022s_OSP$ ls -alh
total 12K
drwxrwxr-x  2 snow snow 4.0K  3월  10 12:53 .
drwxr-xr-x 28 snow snow 4.0K  3월  10 11:16 ..
-rw-rw-r--  1 snow snow   21  3월  10 12:53 myfile.txt
snow@snow-ubuntu:~/2022s_OSP$
```

- **find**
  - **Searches** a directory and all its subdirectories for files and directories that match a description specified on the command line.

sudo fdisk -l

- Creating and manipulating disk partition table

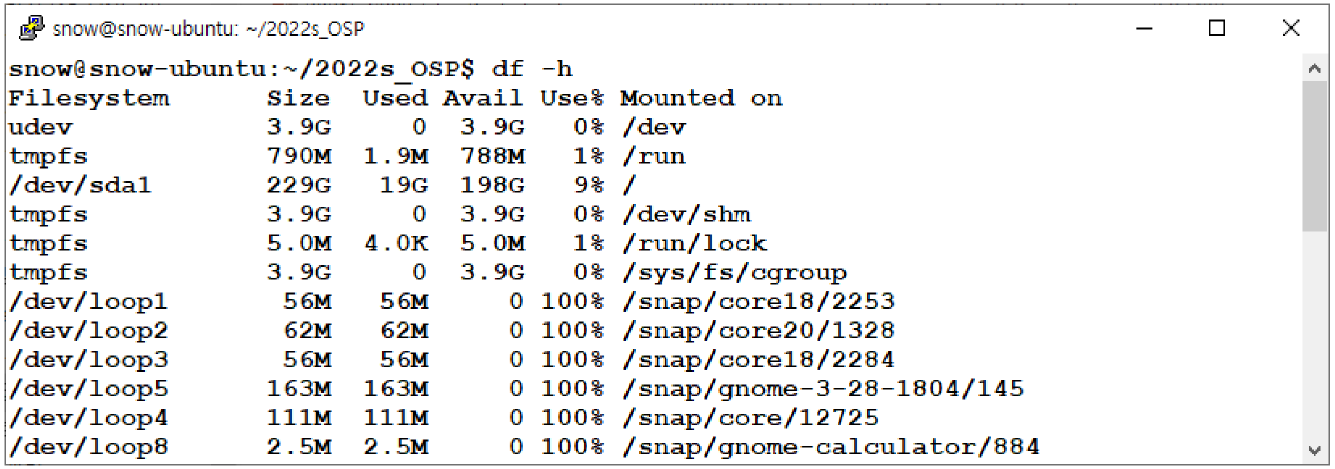


snow@snow-ubuntu: ~/2022s_OSP

```
snow@snow-ubuntu:~/2022s_OSP$ sudo fdisk -l
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop1: 55.5 MiB, 58183680 bytes, 113640 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

df (df -h)

- Disk free: check disk space

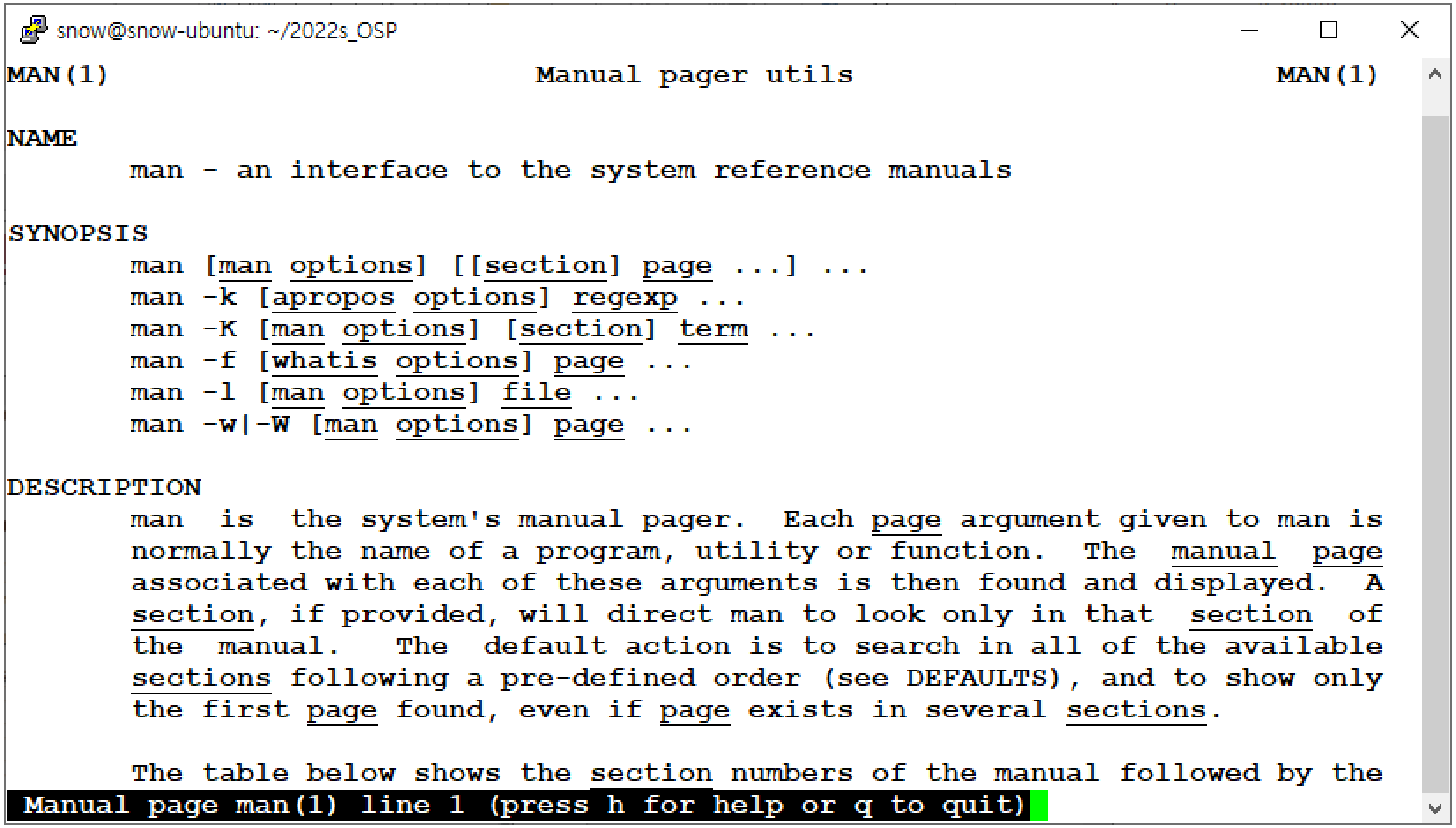


snow@snow-ubuntu: ~/2022s_OSP

```
snow@snow-ubuntu:~/2022s_OSP$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G     0  3.9G   0% /dev
tmpfs           790M  1.9M  788M   1% /run
/dev/sda1       229G   19G  198G   9% /
tmpfs           3.9G     0  3.9G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/loop1       56M   56M     0 100% /snap/core18/2253
/dev/loop2       62M   62M     0 100% /snap/core20/1328
/dev/loop3       56M   56M     0 100% /snap/core18/2284
/dev/loop5      163M  163M     0 100% /snap/gnome-3-28-1804/145
/dev/loop4      111M  111M     0 100% /snap/core/12725
/dev/loop8      2.5M  2.5M     0 100% /snap/gnome-calculator/884
```

Linux Manual

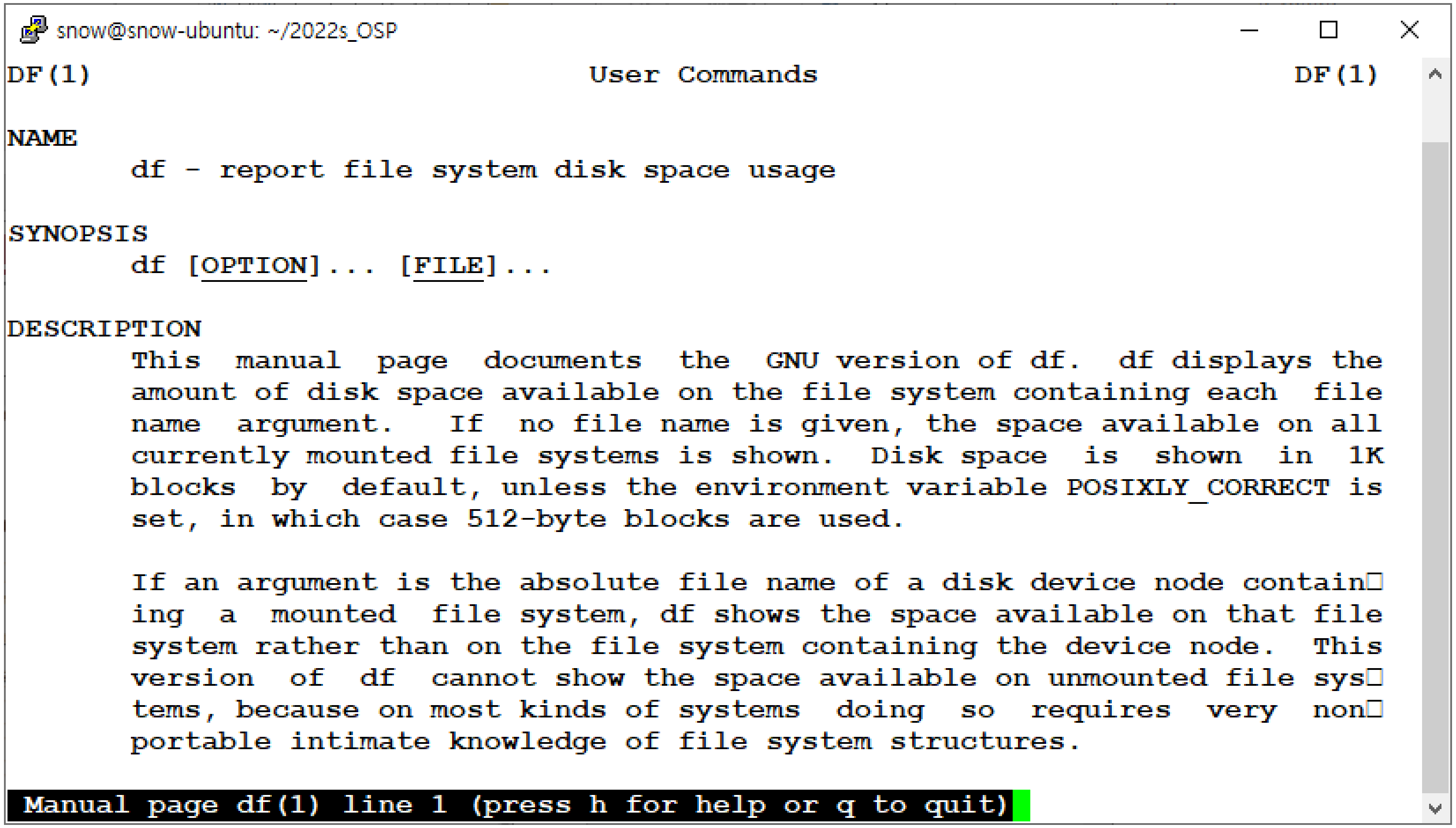
- `man man`



```
MAN(1)                      Manual pager utils                      MAN(1)

NAME
       man - an interface to the system reference manuals

SYNOPSIS
       man [man options] [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [man options] [section] term ...
       man -f [whatis options] page ...
       man -l [man options] file ...
       man -w|-W [man options] page ...

DESCRIPTION
       man  is  the system's manual pager.  Each page argument given to man is
       normally the name of a program, utility or function.  The  manual  page
       associated with each of these arguments is then found and displayed.  A
       section, if provided, will direct man to look only in that  section  of
       the  manual.   The  default action is to search in all of the available
       sections following a pre-defined order (see DEFAULTS), and to show only
       the first page found, even if page exists in several sections.

       The table below shows the section numbers of the manual followed by the
 Manual page man(1) line 1 (press h for help or q to quit)
```

```
q : quit
space : next page
b : before page
```

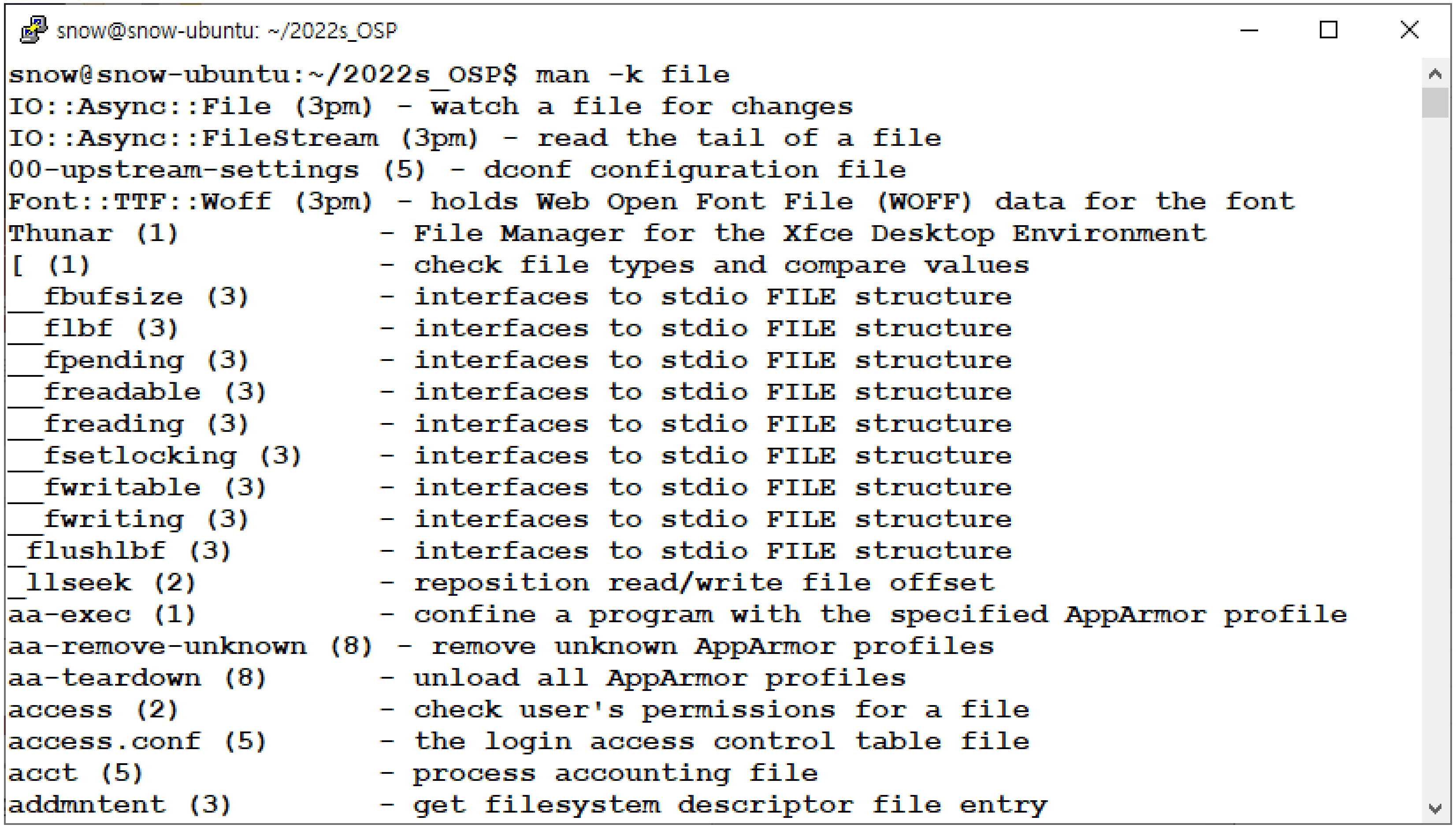
◆ man df



snow@snow-ubuntu: ~/2022s_OSP

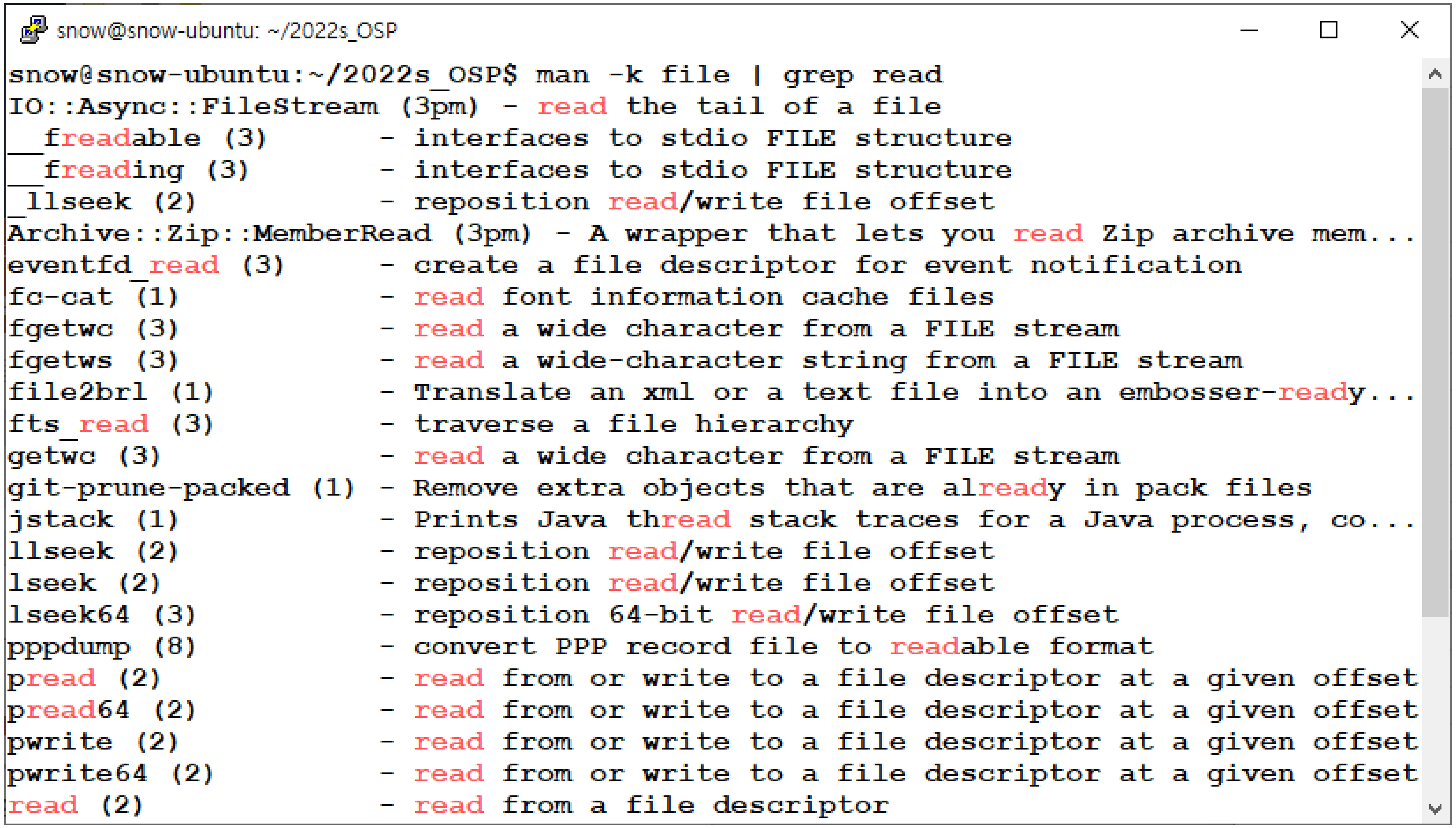
```
DF(1)                            User Commands                           DF(1)

NAME
       df - report file system disk space usage

SYNOPSIS
       df [OPTION]... [FILE]...

DESCRIPTION
       This  manual  page  documents  the  GNU version of df.  df displays the
       amount of disk space available on the file system containing each  file
       name  argument.   If  no file name is given, the space available on all
       currently mounted file systems is shown.  Disk space  is  shown  in  1K
       blocks  by  default, unless the environment variable POSIXLY_CORRECT is
       set, in which case 512-byte blocks are used.

       If an argument is the absolute file name of a disk device node contain‐
       ing  a  mounted  file system, df shows the space available on that file
       system rather than on the file system containing the device node.  This
       version  of  df  cannot show the space available on unmounted file sys‐
       tems, because on most kinds of systems  doing  so  requires  very  non‐
       portable intimate knowledge of file system structures.

 Manual page df(1) line 1 (press h for help or q to quit)
```

◆ `man -k file` : keywords


snow@snow-ubuntu: ~/2022s_OSP

```
snow@snow-ubuntu:~/2022s_OSP$ man -k file
IO::Async::File (3pm) - watch a file for changes
IO::Async::FileStream (3pm) - read the tail of a file
00-upstream-settings (5) - dconf configuration file
Font::TTF::Woff (3pm) - holds Web Open Font File (WOFF) data for the font
Thunar (1)           - File Manager for the Xfce Desktop Environment
[ (1)                - check file types and compare values
__fbufsize (3)       - interfaces to stdio FILE structure
__flbf (3)           - interfaces to stdio FILE structure
__fpending (3)       - interfaces to stdio FILE structure
__freadable (3)      - interfaces to stdio FILE structure
__freading (3)       - interfaces to stdio FILE structure
__fsetlocking (3)    - interfaces to stdio FILE structure
__fwritable (3)      - interfaces to stdio FILE structure
__fwriting (3)       - interfaces to stdio FILE structure
_flushlbf (3)        - interfaces to stdio FILE structure
_llseek (2)          - reposition read/write file offset
aa-exec (1)          - confine a program with the specified AppArmor profile
aa-remove-unknown (8) - remove unknown AppArmor profiles
aa-teardown (8)      - unload all AppArmor profiles
access (2)           - check user's permissions for a file
access.conf (5)      - the login access control table file
acct (5)             - process accounting file
addmntent (3)        - get filesystem descriptor file entry
```

- **grep : print lines matching a pattern**
- `man -k file | grep read`

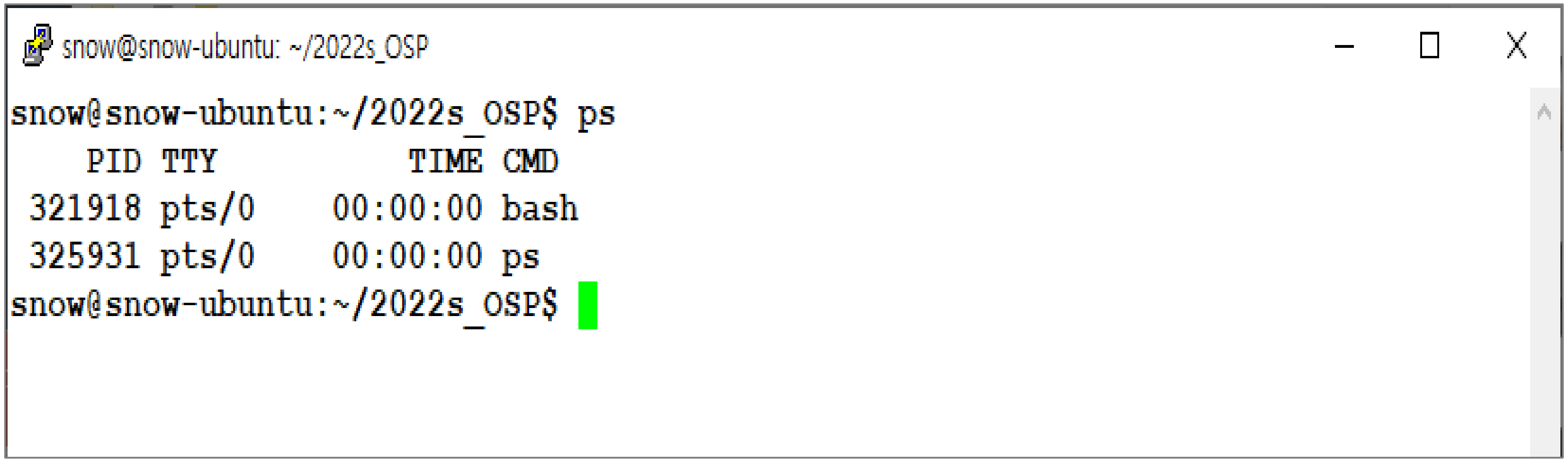


snow@snow-ubuntu: ~/2022s_OSP

```
snow@snow-ubuntu:~/2022s_OSP$ man -k file | grep read
IO::Async::FileStream (3pm) - read the tail of a file
__freadable (3)        - interfaces to stdio FILE structure
__freading (3)         - interfaces to stdio FILE structure
_llseek (2)            - reposition read/write file offset
Archive::Zip::MemberRead (3pm) - A wrapper that lets you read Zip archive mem...
eventfd_read (3)       - create a file descriptor for event notification
fc-cat (1)             - read font information cache files
fgetwc (3)             - read a wide character from a FILE stream
fgetws (3)             - read a wide-character string from a FILE stream
file2brl (1)           - Translate an xml or a text file into an embosser-ready...
fts_read (3)           - traverse a file hierarchy
getwc (3)              - read a wide character from a FILE stream
git-prune-packed (1)   - Remove extra objects that are already in pack files
jstack (1)             - Prints Java thread stack traces for a Java process, co...
llseek (2)             - reposition read/write file offset
lseek (2)              - reposition read/write file offset
lseek64 (3)            - reposition 64-bit read/write file offset
pppdump (8)            - convert PPP record file to readable format
pread (2)              - read from or write to a file descriptor at a given offset
pread64 (2)            - read from or write to a file descriptor at a given offset
pwrite (2)             - read from or write to a file descriptor at a given offset
pwrite64 (2)           - read from or write to a file descriptor at a given offset
read (2)               - read from a file descriptor
```

Process Commands

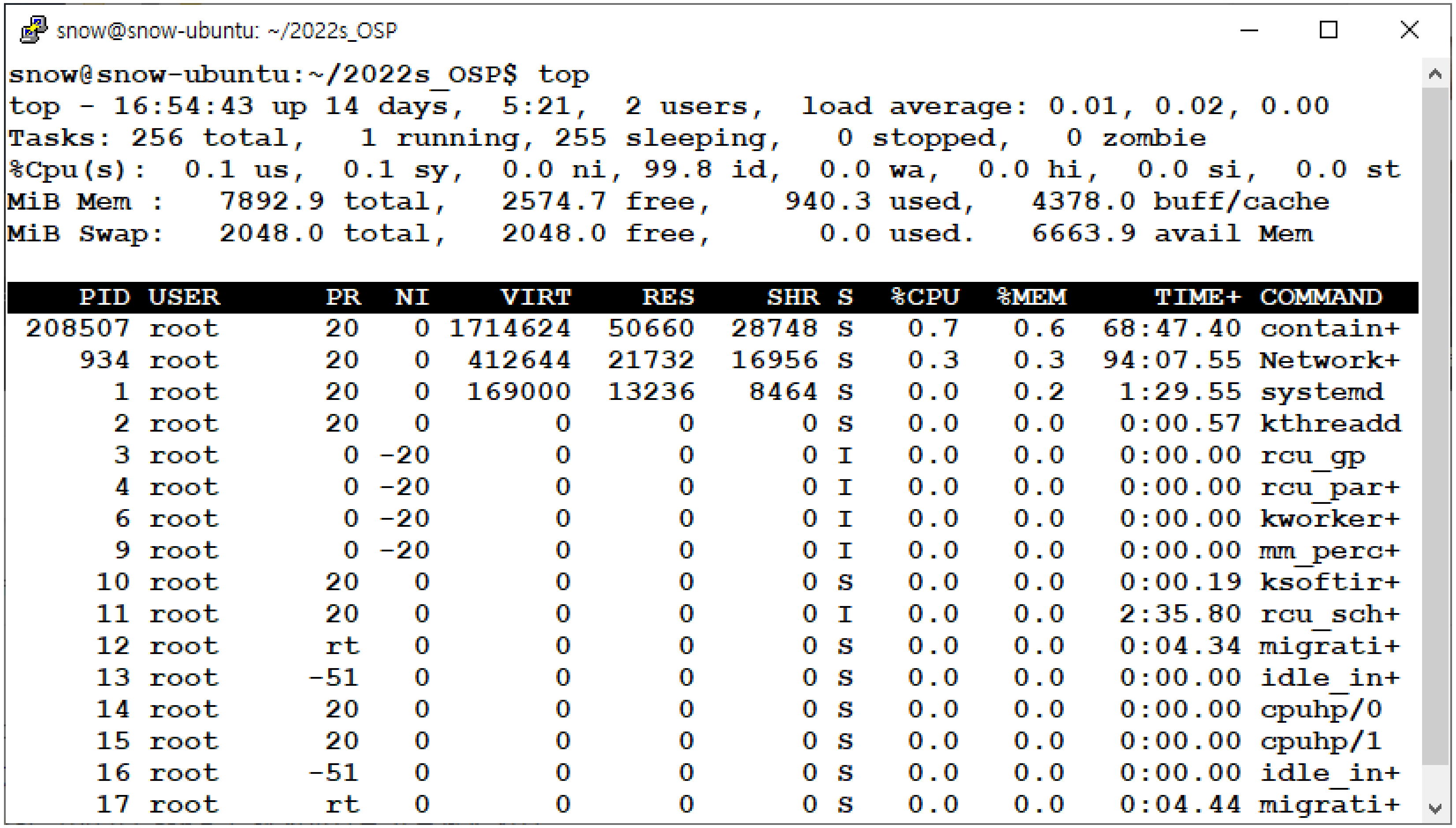
- `ps (ps -la, ps -a)`



```
snow@snow-ubuntu:~/2022s_OSP$ ps
    PID TTY          TIME CMD
 321918 pts/0    00:00:00 bash
 325931 pts/0    00:00:00 ps
snow@snow-ubuntu:~/2022s_OSP$ 
```

- `kill -9 pid#` **: kill a process #**

- **top : a real-time view of running processes in Linux and displays kernel-managed tasks**



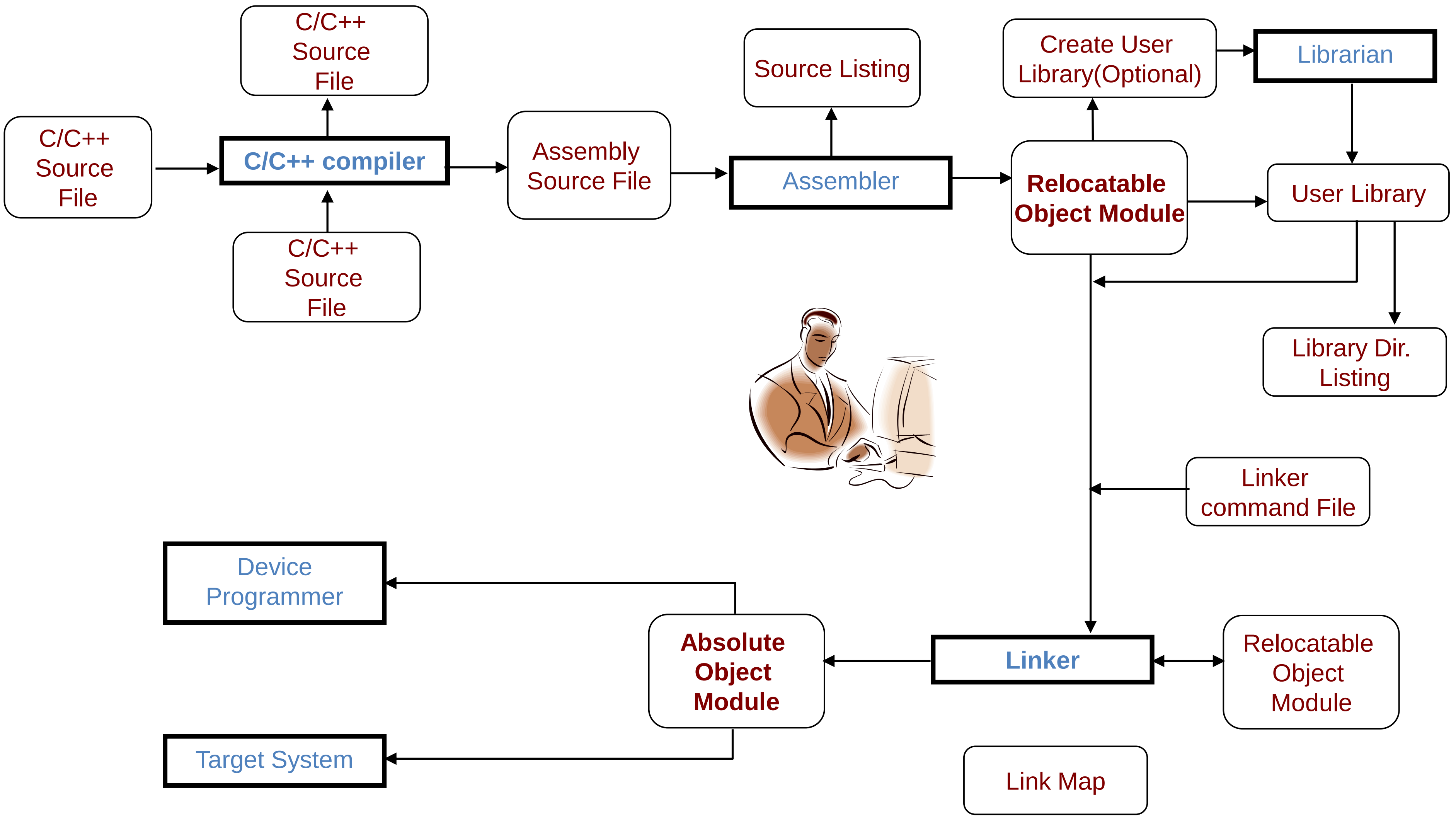
snow@snow-ubuntu: ~/2022s_OSP

```
snow@snow-ubuntu:~/2022s_OSP$ top
top - 16:54:43 up 14 days,  5:21,  2 users,  load average: 0.01, 0.02, 0.00
Tasks: 256 total,   1 running, 255 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.1 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7892.9 total,  2574.7 free,   940.3 used,   4378.0 buff/cache
MiB Swap:   2048.0 total,  2048.0 free,     0.0 used.   6663.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 208507 root      20   0 1714624  50660  28748 S   0.7   0.6  68:47.40 contain+
    934 root      20   0  412644  21732  16956 S   0.3   0.3  94:07.55 Network+
      1 root      20   0  169000  13236   8464 S   0.0   0.2   1:29.55 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.57 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.19 ksoftir+
     11 root      20   0       0      0      0 I   0.0   0.0   2:35.80 rcu_sch+
     12 root      rt   0       0      0      0 S   0.0   0.0   0:04.34 migrati+
     13 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_in+
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_in+
     17 root      rt   0       0      0      0 S   0.0   0.0   0:04.44 migrati+
```

```
Exit: Ctrl + C
```

C Programming Environment in GNU

System Software Development Process



Utilities for Systems Programming in UNIX

- **Editor : Vi / Vim**
- **C/C++ Compiler and Library Builder**
  - CC (C Compiler)
  - GCC (GNU Compiler Collection) : C, C++, Pascal, Ada, etc.
- **Make**
- **Debugger**
- **System Libraries**

Development System Roadmap

- **Programs :**

  /bin, /usr/bin

  /usr/local/bin

  .$ ./mypgm

- **Header Files : to provide definitions of constant and declarations for system and library function calls**

  /usr/include

  /home/mypgm/include

  $gcc -I/usr/openwin/include fred.c

  $grep EXIT_ *.h

- **Library Files :**

  /usr/lib

  /home/mypgm/lib

  xxx.a *for traditional, static libraries*

  xxx.so and xxx.sa *for shared libraries*

  $gcc -o fred fred.c /usr/lib/libm.a

  $gcc -o x11fred -L/usr/openwin/lib x11fred.c -lx11

Create & Use Static Libraries

fred.c

```
#include <stdio.h>
void fred(int arg)
{
    printf(“fred: you passed %d\n”,arg);
}
```

bill.c

```
#include <stdio.h>
void bill(int arg)
{
    printf(“bill: you passed %s\n”,arg);
}
```

```
$gcc -c bill.c fred.c
$ls *.o
```

lib.h

```
/* This is lib.h. */
void fred(char *);
void bill(int);
```

mypgm.c

```
#include “lib.h”
int main()
{
    bill(“Hello World”);
    exit(0);
}
```

```
$gcc -c mypgm.c
$gcc -o mypgm mypgm.o bill.o
$./mypgm
```

$ar crv libfoo.a bill.o fred.o

→ create static library

$ranlib libfoo.a

→ generate library index

$gcc -o mypgm mypgm.o libfoo.a

or

$gcc -o mypgm mrpgm.o -L. -lfoo

Make

- Command generator using a description file and some general templates
- creates a sequence of commands for execution by the UNIX shell
- commands commonly relate to the maintenance of the files comprising a software development project.
- “Maintenance” refers to a whole array of tasks, ranging from status reporting and the purging of temporary files, to building the final, executable version of a complex group of programs
- most naturally used to sort out dependency relations among files
- Even relatively small software projects, If you modify one or more source files, you must relink the program after recompiling some of the sources.
- This process is normally repeated many times during the course of a project.

➔ We need **make importantly !**

How to write a Simple Makefile

$make mypgm

- want to “make” a version - usually the latest version -- of “mypgm” program.
- if mypgm is an executable file, to perform all necessary compilation and liking required to create the file.
- Instead of entering a great many compile and linking commands by hand, you use ***make*** to automate the process
- We call “mypgm” program the **target** of the operation. And “mypgm” is built from one or more files, called **prerequisites** or **dependents**.
- In ***make***, specifying the dependencies in a description file, defaulted file name is “***Makefile***”

The Description File

- **Suppose you are writing a program that consists of :**
  - Three C language source files - main.c iodat.c dorun.c
  - Assembly language code in lo.s, called by one of the C source.
  - A set of library routines in /usr/fred/lib/crtn.a

- **If you built the program by hand,..:**

```
$gcc -c main.c
$gcc -c iodat.c
$gcc -c dorun.c
$as -o lo.o lo.s
$gcc -o program main.o iodat.o dorun.o lo.o -L/usr/fred/lib -lcrtn.a
```

- **Need every times whenever rebuild, @.@**

The Description File (Cont.)

```
$ vi Makefile
```

```
program : main.o iodat.o dorun.o lo.o /usr/fred/lib/crtn.a
        cc -o program main.o iodat.o dorun.o lo.o /usr/fred/lib/crtn.a
main.o : main.c
        cc -c main.c
iodat.o : iodat.c
        cc -c iodat.c
dorun.o : dorun.c
        cc -c dorun.c
lo.o : lo.s
        as -o lo.o lo.s
```

```
$ make program
```

Dependency Checking

- The numbering in Makefile show the execution sequence of commands
- Therefore, first of all, make checks ***main.o,iodat.o,dorun.o,lo.o***,and ***crtn.a*** to see whether any of them are newer than ***program*** by Line 1
- Because all files in UNIX have the touched timing information
- If ***program*** was built since the latest modifications of all its prerequisites, ***make*** may decide that there is no need to rebuild it, and exit without issuing any commands
- If ***main.c*** in Line 3 was modified after the last time main.o was made, make executes the compile command in Line 4 and thus create a new, up-to-date ***main.o***
- Assuming that ***main.c*** is the only file that changed since program was last built, the command executed by ***make*** are :

```
$cc -c main.c
$cc -o program main.o iodat.o dorun.o lo.o /usr/fred/lib/crtn.a
```

Minimizing Rebuilds

Suppose a program that can exist in several different versions:

```
plot_prompt : basic.o prompt.o
        cc -o plot_promt basic.o prompt.o
plot_win : basic.o window.o
        cc -o plot_win basic.o window.o
basic.o : basic.c
        cc -c basic.c
prompt.o : prompt.c
        cc -c prompt.c
window.o : window.c
        cc -c window.c
```

➔ **$make plot_prompt** or **$make plot_win**

Invoking *make*

$make ***target***

? `target` is up to date

$make ***nontarget***

→ ***if no description in Makefile***

make: Don't know how to make nontarget. Stop.

$make

→ the **first target** contained in the description file is made

Basic Rules in Syntax

- **Don’t use tab as a first char in a line**
- **(\) at the end of the line, to continue a long line**
- **(#) beginning of comment line**

Tips :

```
clean :
        /bin/rm -f core *.o
```

$make clean

Macros

- **Macro**

```
name = hello world          ;macro definition
$(name) or ${name}          ;references
→ hello world
```

- **Example :**

```
LIBES = -lX11
objs = drawable.o plot_points.o root_data.o
CC=gcc
23 = “This is the (23)rd run”
DEBUG_FLAG = #empty now, but assign -g for debugging
BINDIR = /usr/local/bin

plot : ${objs}
        ${CC} -o plot ${DEBUG_FLAG} ${objs} ${LIBS}
        mv plot ${BINDIR}
```

Macros (Cont.)

- **Nested macros :**

```
SOURCE = ${MY_SRC} ${SHARED_SRC}
MY_SRC = parse.c search.c
SHARED_DIR = /users/project/src
SHARED_SRC = ${SHARED_DIR}/depend.c
```

- **Predefined Macros**

```
${CC}, ${LD}
```

- **Conventional Macros**

```
${CFLAGS}, ${LDFLAGS}
```

- **Internal Macros**

```
plot_prompt : basic.o prompt.o
        cc -o $@ basic.o prompt.o
libops : interact.o sched.o gen.o
        ar r $@ $?
```

Debugging with GDB

Purpose of a source-code debugger

- **A source-code debugger is a piece of software which can be used to find semantic (i.e. run-time) program errors.**

- **It should not be used to find syntax (i.e. compile-time) program errors.**

The gdb Source-Code Debugger

- **The gdb source-code debugger is available in the UNIX operating system environment.**
- **gdb can be used to perform the following operations which are very helpful in the process of debugging a compiled program :**
  A. Setting break-points: Program execution can be temporarily suspended at specified points (called "break-points"). At the point of program suspension, specific values/outcomes can be displayed to determine their correctness. Upon program suspension, the programmer can interact with gdb and use its full set of commands to investigate the performance of the executing program before resuming program execution.
  B. Displaying program values and attributes: gdb(dbx) can be made to display the current contents of variables as the program executes.
  C. Step through a program line-by-line. Each line of the executable program can be executed one line at a time.

Compiling a Program that is to be Debugged with gdb

- **The (source-code) program that is to be debugged using gdb must first be successfully compiled (no compilation errors found during compilation).**

- **Once the program source-code compiles successfully, compile it one more time using the *"-g"* compiler option as in:**

```
$gcc -g source_code_file.cpp
```

- **The use of the *"-g"* compiler option will cause the compiler to build special files/tables of data that gdb will need for subsequent debugging.**

Invoking the gdb Debugger

- **After compiling the source code file in the manner described above, the gdb debugger can be invoked to assist the programmer in debugging his/her program.**
- **To invoke gdb, get to the UNIX command-line prompt and enter:**

  ```
  $gdb executable_file
  ```

  where "executable_file" is the name of the compiled executable form of the program (which will be "a.out" unless you have changed its name).
- **The gdb debugger will access the special files/tables created by using the "-g" compiler option.**
- **The gdb prompt "(gdb)" will be displayed and you will now be in the gdb environment and will be able to enter only gdb commands (no UNIX commands will be recognized).**

Controlling the gdb debugger

- **To abort your program you can type ctrl-c. From there you can see what line it was executing, look at the values of variables, etc.**
- **To restart a program from a stopped point. Type cont (as in continue) to keep running it from where you were.**
- **To exit the gdb(dbx) debugger and return to the UNIX command-line prompt, enter:**

  ```
  $(gdb) quit
  ```

  from the (gdb) command line.
- **How to get more info about gdb.**

  ```
  $(gdb) help
  ```

More debugging skills

```
(gdb) help
        List of classes of commands:
        running -- Running the program
        stack -- Examining the stack
        data -- Examining data
        breakpoints -- Making program stop at certain points
        files -- Specifying and examining files
        status -- Status inquiries
        support -- Support facilities
        user-defined -- User-defined commands
        aliases -- Aliases of other commands
        obscure -- Obscure features
        internals -- Maintenance commands
```

A typical debugging session might follow these steps:

① **Invoke *gdb* on the executable file compiled with the *-g* option.**

② **Enter the gdb command *break main***

③ **Enter the gdb command *run* to start your program running Your program will suspend execution as soon as it gets to function "main" because the command above said to break in function "main"**

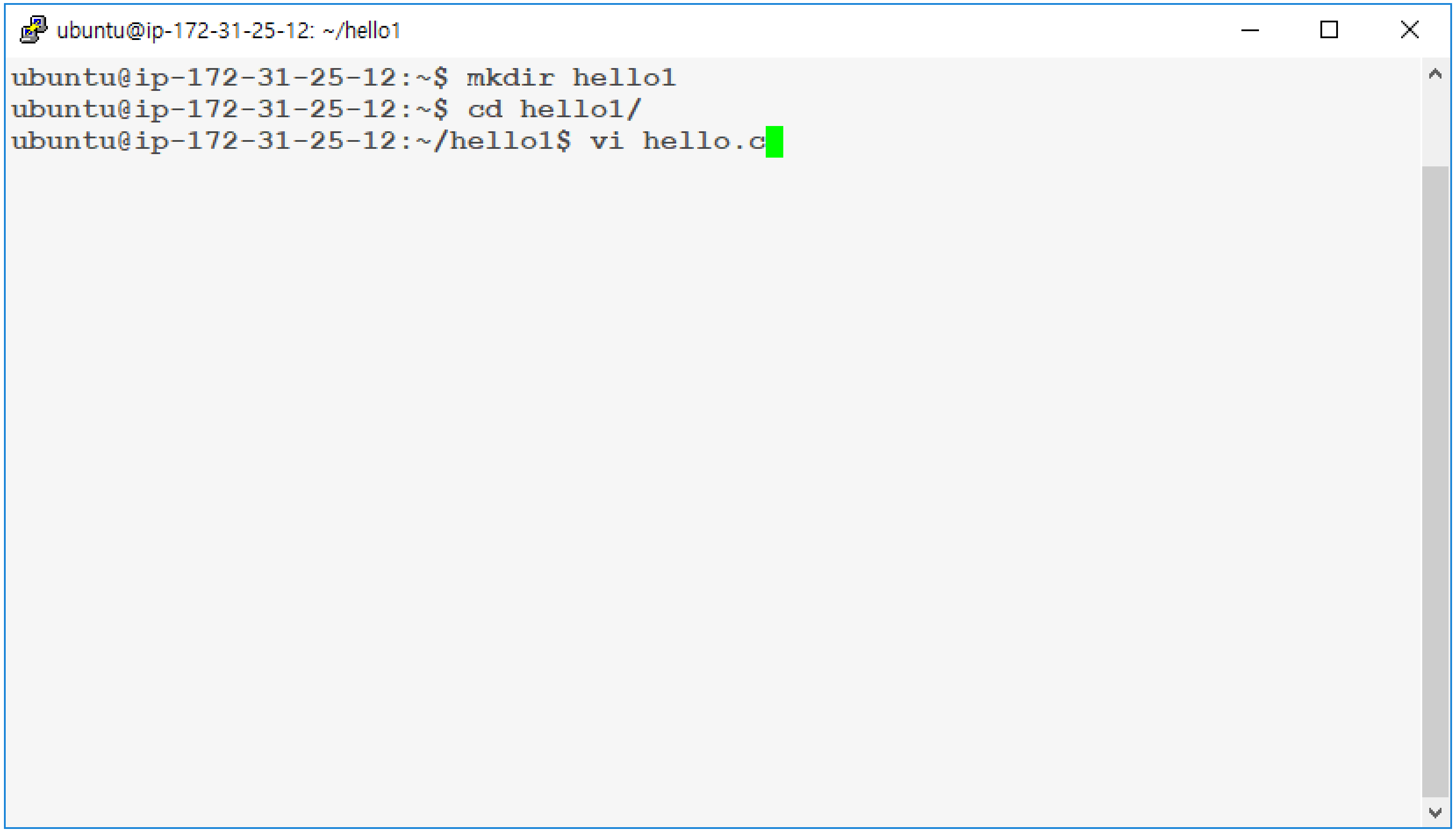
④ **Enter the gdb command *step* or *next* to execute the current line of the program. Remember "*step*" will jump into a function call, while "*next*" will jump over a function call.**

⑤ **Enter the gdb command *print var_name* to see the value stored in variable "var_name".**

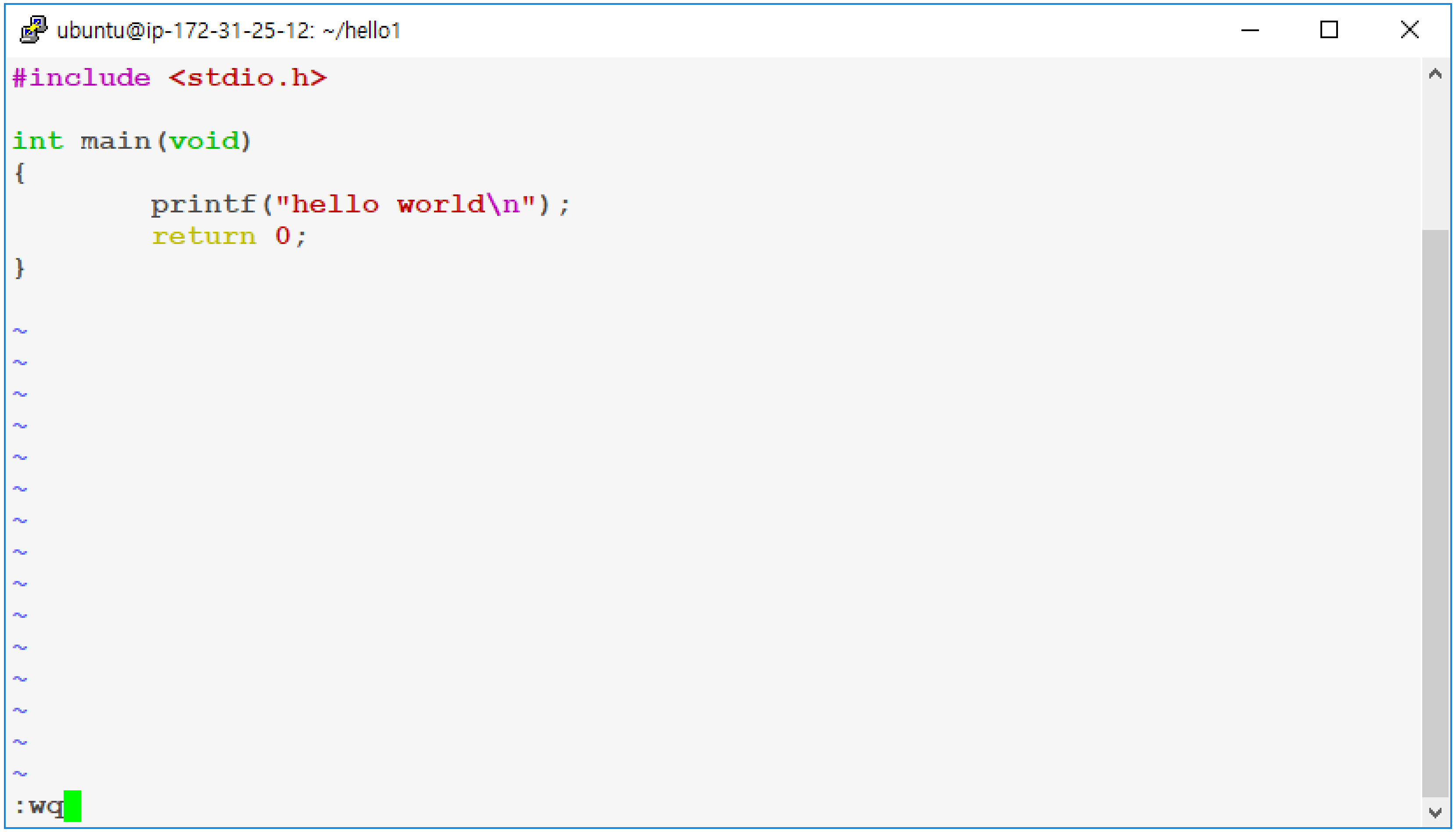
Simple Program

Write Sample code

- Create “hello.c” that prints “hello world”



```
ubuntu@ip-172-31-25-12:~$ mkdir hello1
ubuntu@ip-172-31-25-12:~$ cd hello1/
ubuntu@ip-172-31-25-12:~/hello1$ vi hello.c
```



```
#include <stdio.h>

int main(void)
{
        printf("hello world\n");
        return 0;
}

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

Compile C/C++ Program

- **cc filename**
  - C compiler
- **gcc filename**
  - GNU C Compiler from the GCC (GNU Compiler Collection)
- **g++ filename**
  - GNU C++ Compiler from the GCC

- **(In general) use gcc ➔ call cc**

- **Option**

  -c : create object file

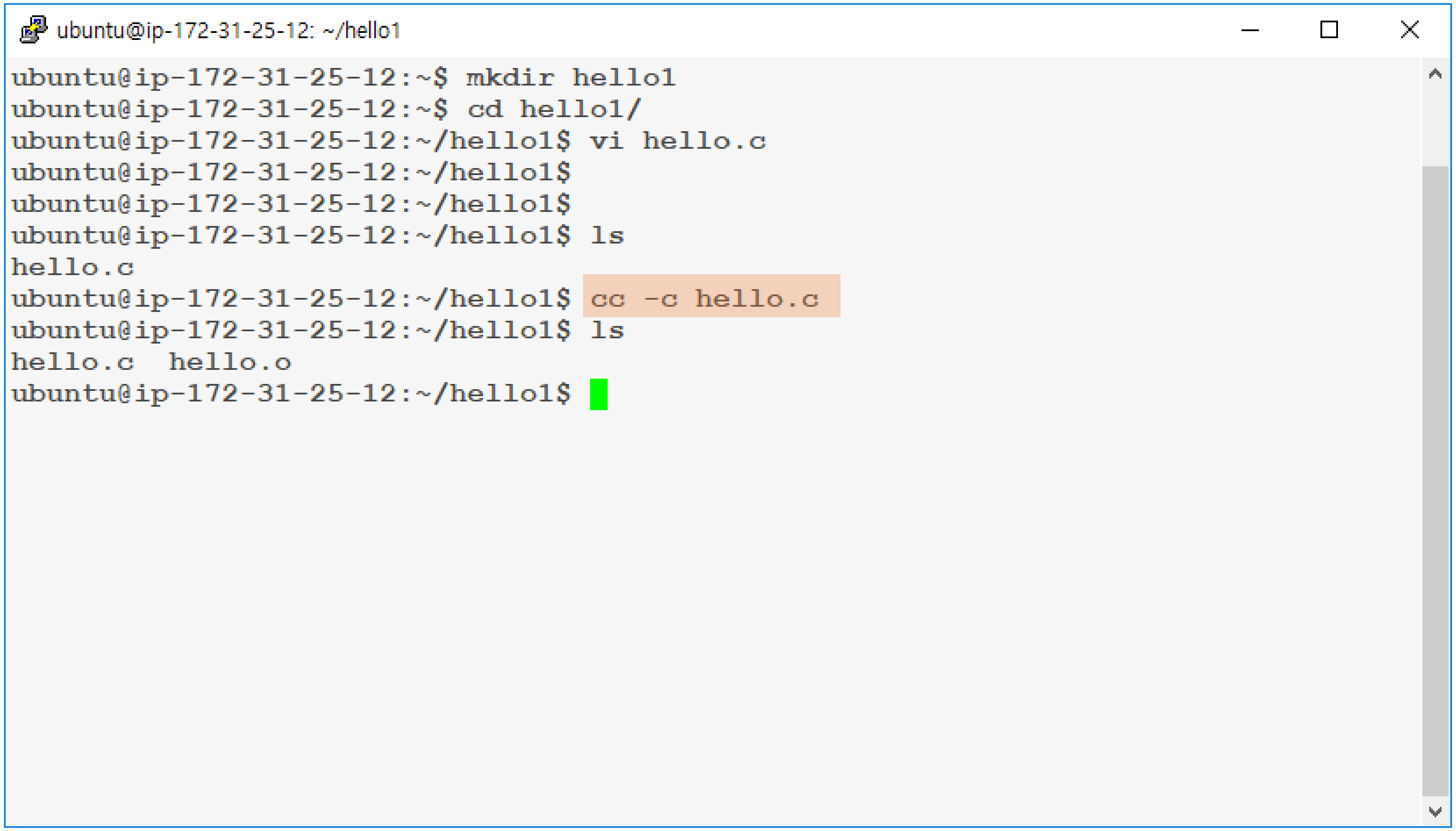
  -o : create output(executable) file

  (Without this option, the execute file name is a.out)

  -g : add debugging info

Create object file

- cc –c hello.c



ubuntu@ip-172-31-25-12: ~/hello1

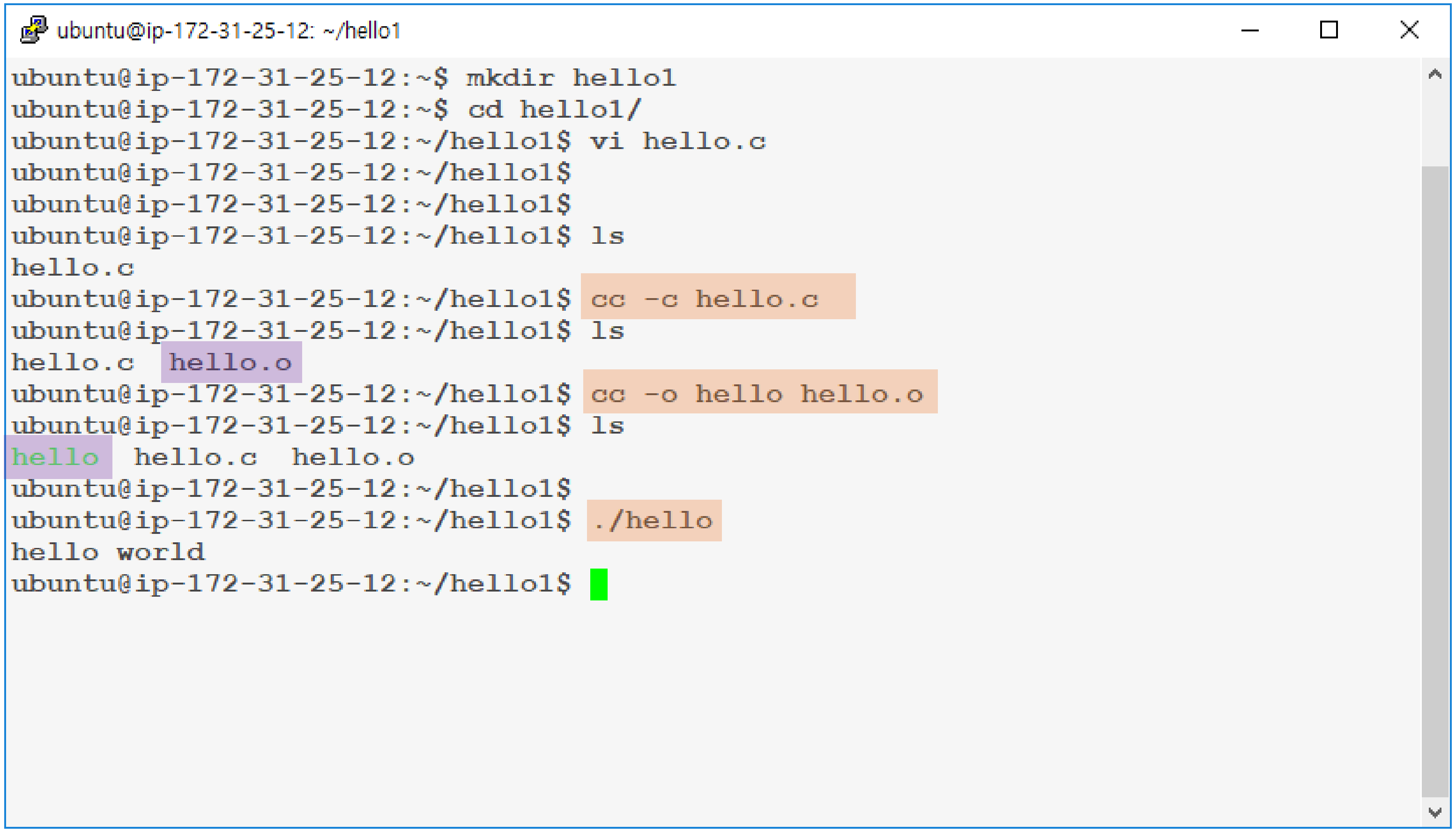
```
ubuntu@ip-172-31-25-12:~$ mkdir hello1
ubuntu@ip-172-31-25-12:~$ cd hello1/
ubuntu@ip-172-31-25-12:~/hello1$ vi hello.c
ubuntu@ip-172-31-25-12:~/hello1$
ubuntu@ip-172-31-25-12:~/hello1$
ubuntu@ip-172-31-25-12:~/hello1$ ls
hello.c
ubuntu@ip-172-31-25-12:~/hello1$ cc -c hello.c
ubuntu@ip-172-31-25-12:~/hello1$ ls
hello.c  hello.o
ubuntu@ip-172-31-25-12:~/hello1$
```

Create execute file

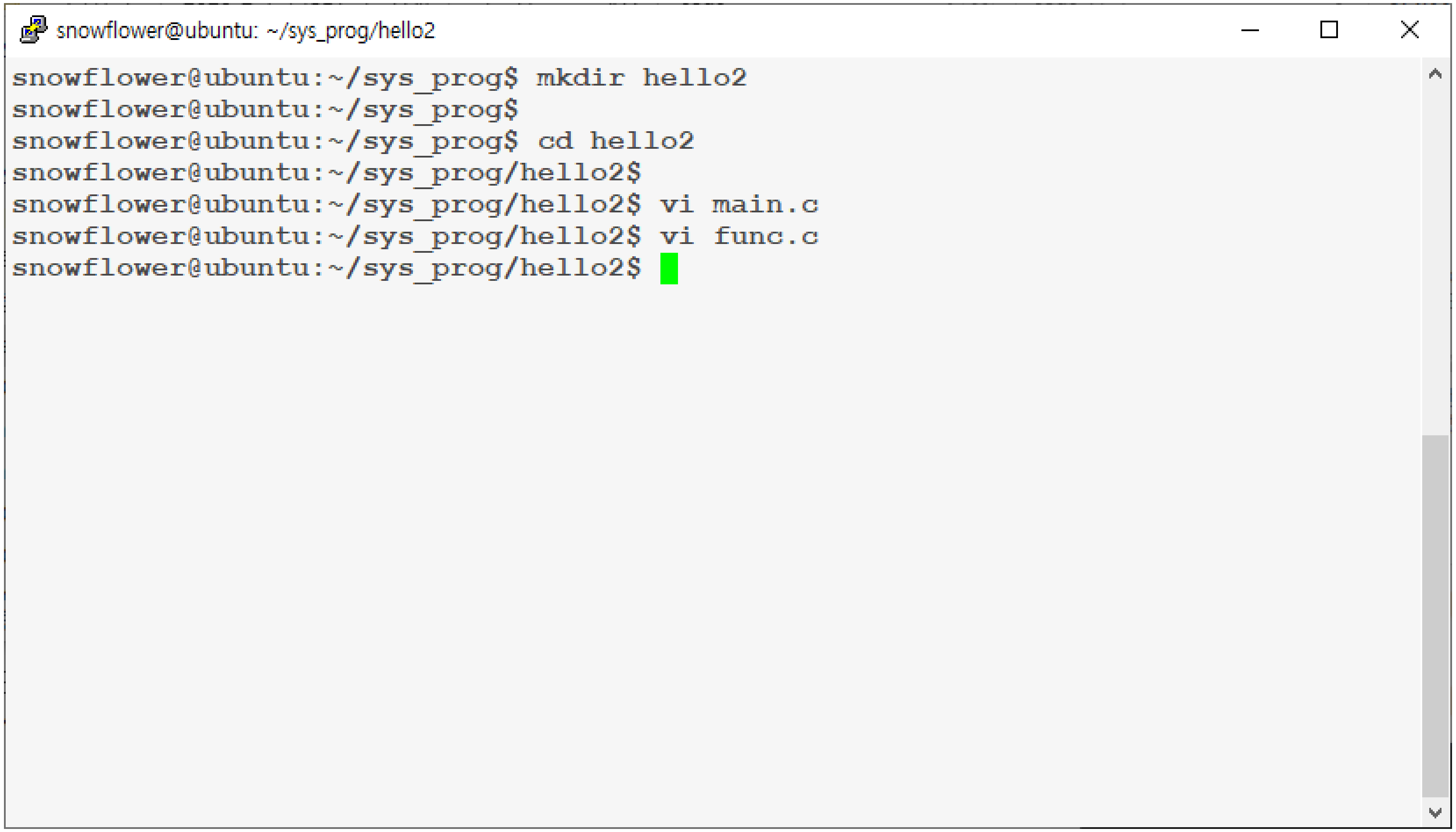
- cc –o hello hello.o

Run program

- ./hello



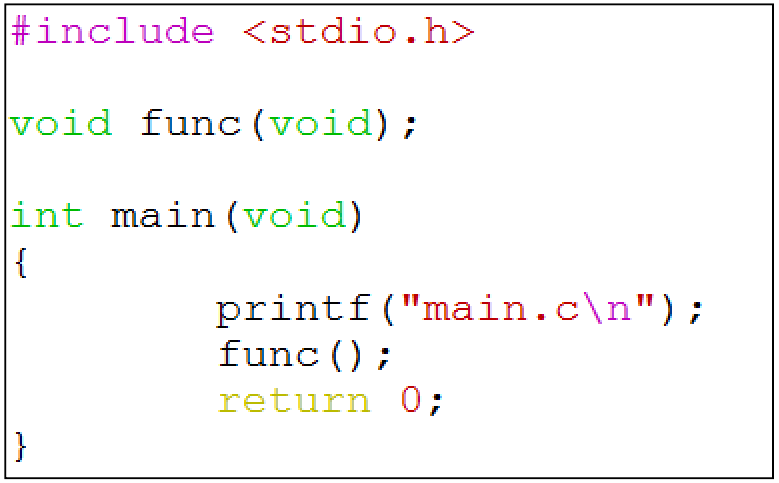
```
ubuntu@ip-172-31-25-12:~$ mkdir hello1
ubuntu@ip-172-31-25-12:~$ cd hello1/
ubuntu@ip-172-31-25-12:~/hello1$ vi hello.c
ubuntu@ip-172-31-25-12:~/hello1$
ubuntu@ip-172-31-25-12:~/hello1$
ubuntu@ip-172-31-25-12:~/hello1$ ls
hello.c
ubuntu@ip-172-31-25-12:~/hello1$ cc -c hello.c
ubuntu@ip-172-31-25-12:~/hello1$ ls
hello.c  hello.o
ubuntu@ip-172-31-25-12:~/hello1$ cc -o hello hello.o
ubuntu@ip-172-31-25-12:~/hello1$ ls
hello  hello.c  hello.o
ubuntu@ip-172-31-25-12:~/hello1$
ubuntu@ip-172-31-25-12:~/hello1$ ./hello
hello world
ubuntu@ip-172-31-25-12:~/hello1$
```



```
snowflower@ubuntu:~/sys_prog$ mkdir hello2
snowflower@ubuntu:~/sys_prog$
snowflower@ubuntu:~/sys_prog$ cd hello2
snowflower@ubuntu:~/sys_prog/hello2$
snowflower@ubuntu:~/sys_prog/hello2$ vi main.c
snowflower@ubuntu:~/sys_prog/hello2$ vi func.c
snowflower@ubuntu:~/sys_prog/hello2$
```

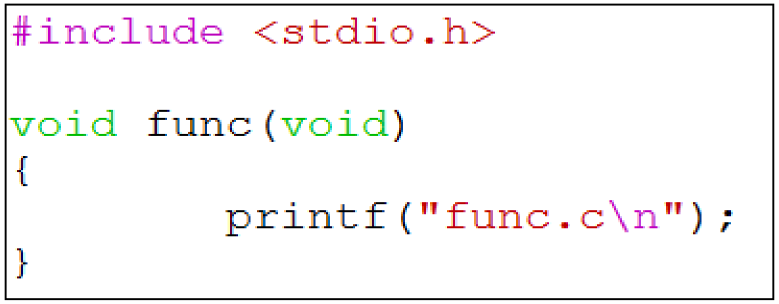
A program with 2 source files

- **main.c**



```c
#include <stdio.h>

void func(void);

int main(void)
{
        printf("main.c\n");
        func();
        return 0;
}
```

- **func.c**



```c
#include <stdio.h>

void func(void)
{
        printf("func.c\n");
}
```

- **Create object & execute file**

```
$ cc –o test main.c func.c
$ ./test
```

($ cc main.c func.c –o test)

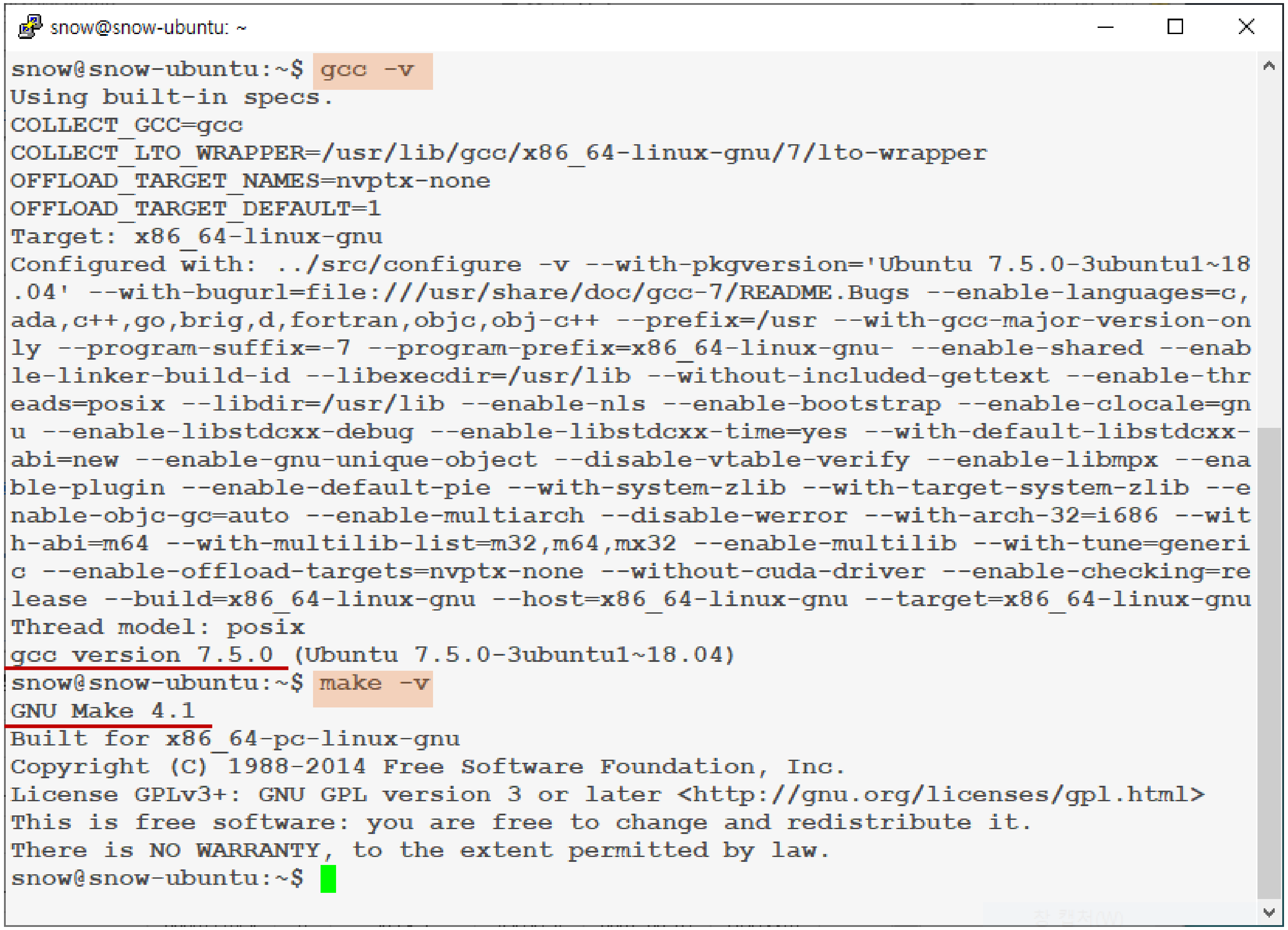
```
$ cc –c main.c
$ cc –c func.c
$ cc –o test main.o func.o
$ ./test
```

($ cc main.o func.o –o test)

```
$ cc –c main.c func.c
$ cc –o test main.o func.o
$ ./test
```

gcc & make

- **gcc, make version**

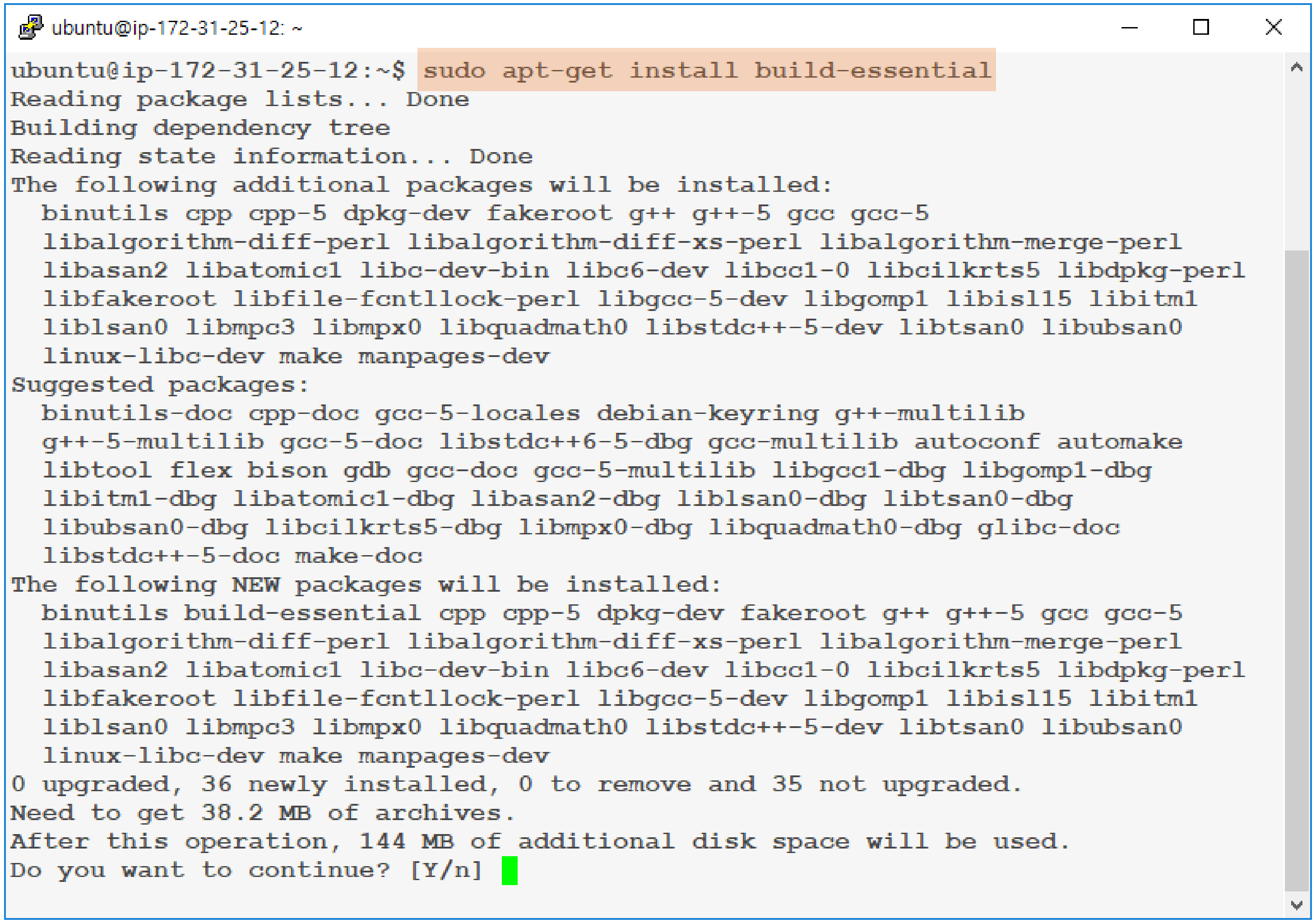


```
snow@snow-ubuntu:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 7.5.0-3ubuntu1~18
.04' --with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=c,
ada,c++,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-on
ly --program-suffix=-7 --program-prefix=x86_64-linux-gnu- --enable-shared --enab
le-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-thr
eads=posix --libdir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gn
u --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-
abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-libmpx --ena
ble-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib --e
nable-objc-gc=auto --enable-multiarch --disable-werror --with-arch-32=i686 --wit
h-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generi
c --enable-offload-targets=nvptx-none --without-cuda-driver --enable-checking=re
lease --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)
snow@snow-ubuntu:~$ make -v
GNU Make 4.1
Built for x86_64-pc-linux-gnu
Copyright (C) 1988-2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
snow@snow-ubuntu:~$
```

gcc & make install

- install make package

  $ sudo apt-get install build-essential

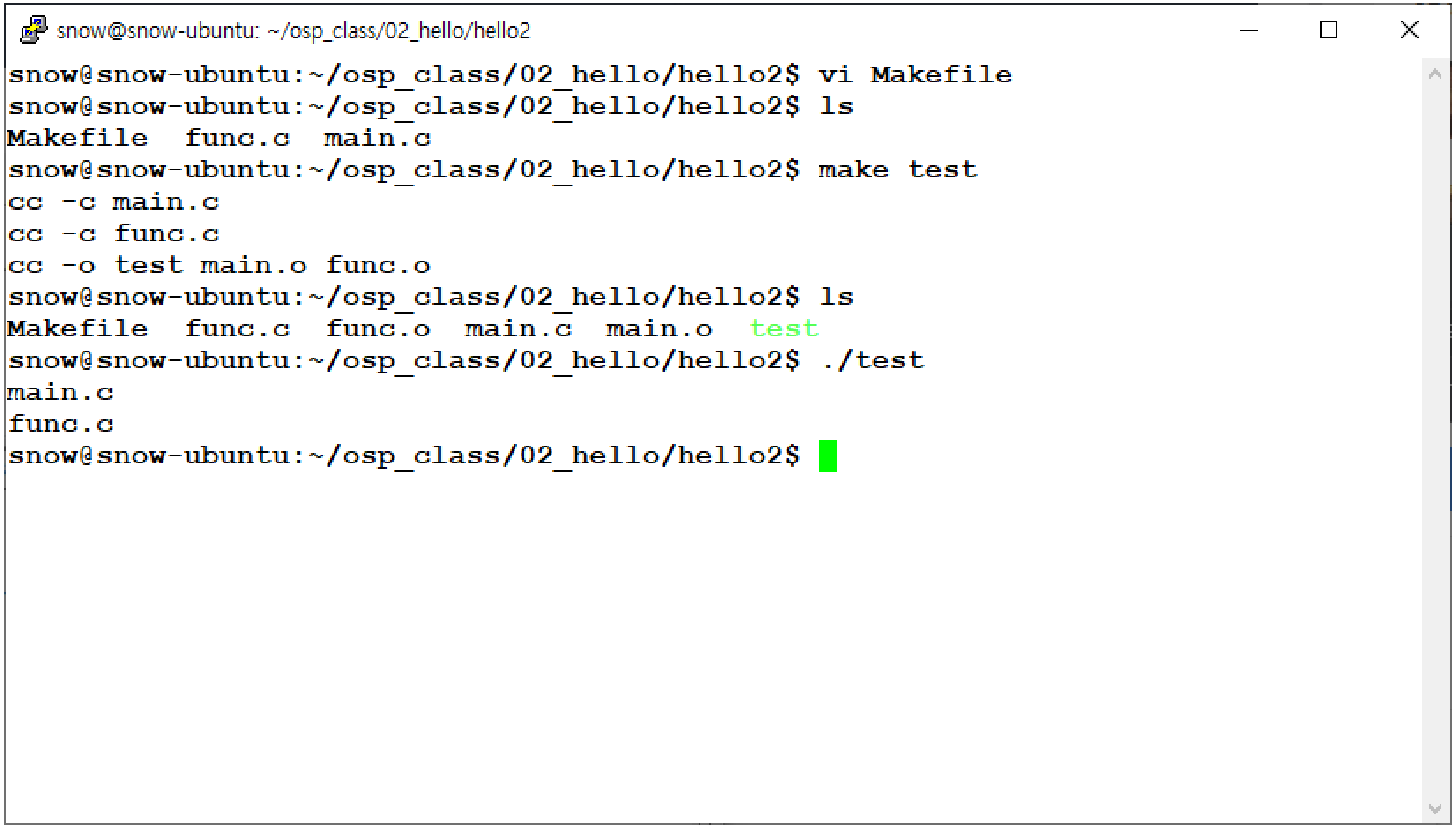


```
ubuntu@ip-172-31-25-12:~$ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils cpp cpp-5 dpkg-dev fakeroot g++ g++-5 gcc gcc-5
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libasan2 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrts5 libdpkg-perl
  libfakeroot libfile-fcntllock-perl libgcc-5-dev libgomp1 libisl15 libitm1
  liblsan0 libmpc3 libmpx0 libquadmath0 libstdc++-5-dev libtsan0 libubsan0
  linux-libc-dev make manpages-dev
Suggested packages:
  binutils-doc cpp-doc gcc-5-locales debian-keyring g++-multilib
  g++-5-multilib gcc-5-doc libstdc++6-5-dbg gcc-multilib autoconf automake
  libtool flex bison gdb gcc-doc gcc-5-multilib libgcc1-dbg libgomp1-dbg
  libitm1-dbg libatomic1-dbg libasan2-dbg liblsan0-dbg libtsan0-dbg
  libubsan0-dbg libcilkrts5-dbg libmpx0-dbg libquadmath0-dbg glibc-doc
  libstdc++-5-doc make-doc
The following NEW packages will be installed:
  binutils build-essential cpp cpp-5 dpkg-dev fakeroot g++ g++-5 gcc gcc-5
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libasan2 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrts5 libdpkg-perl
  libfakeroot libfile-fcntllock-perl libgcc-5-dev libgomp1 libisl15 libitm1
  liblsan0 libmpc3 libmpx0 libquadmath0 libstdc++-5-dev libtsan0 libubsan0
  linux-libc-dev make manpages-dev
0 upgraded, 36 newly installed, 0 to remove and 35 not upgraded.
Need to get 38.2 MB of archives.
After this operation, 144 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

- **$ vi Makefile**

```
test : main.o func.o
	cc –o test main.o func.o

main.o : main.c
	cc –c main.c

func.o : func.c
	cc –c func.c
```

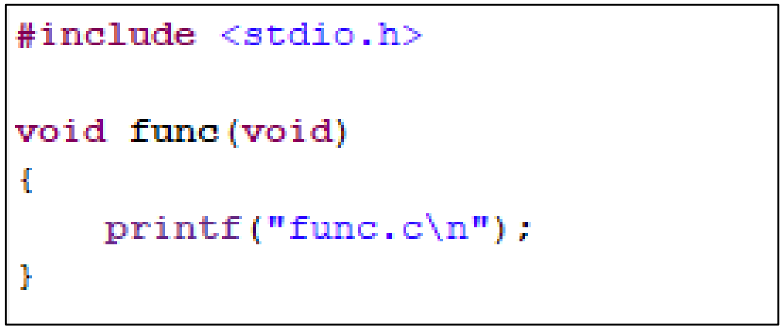
- **$ make test**
- **$ ./test**



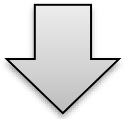
snow@snow-ubuntu: ~/osp_class/02_hello/hello2

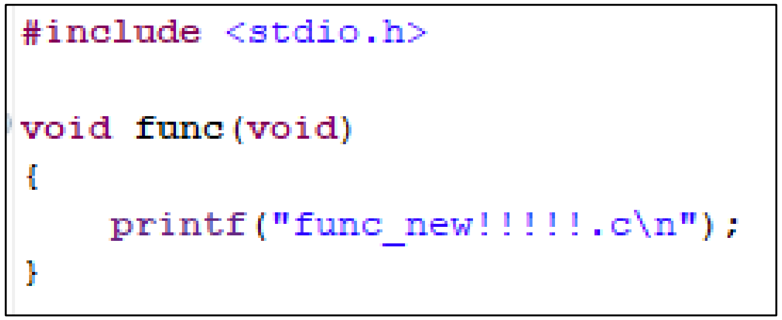
```
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ vi Makefile
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ ls
Makefile  func.c  main.c
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ make test
cc -c main.c
cc -c func.c
cc -o test main.o func.o
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ ls
Makefile  func.c  func.o  main.c  main.o  test
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ ./test
main.c
func.c
snow@snow-ubuntu:~/osp_class/02_hello/hello2$
```

- modify a file "func.c"



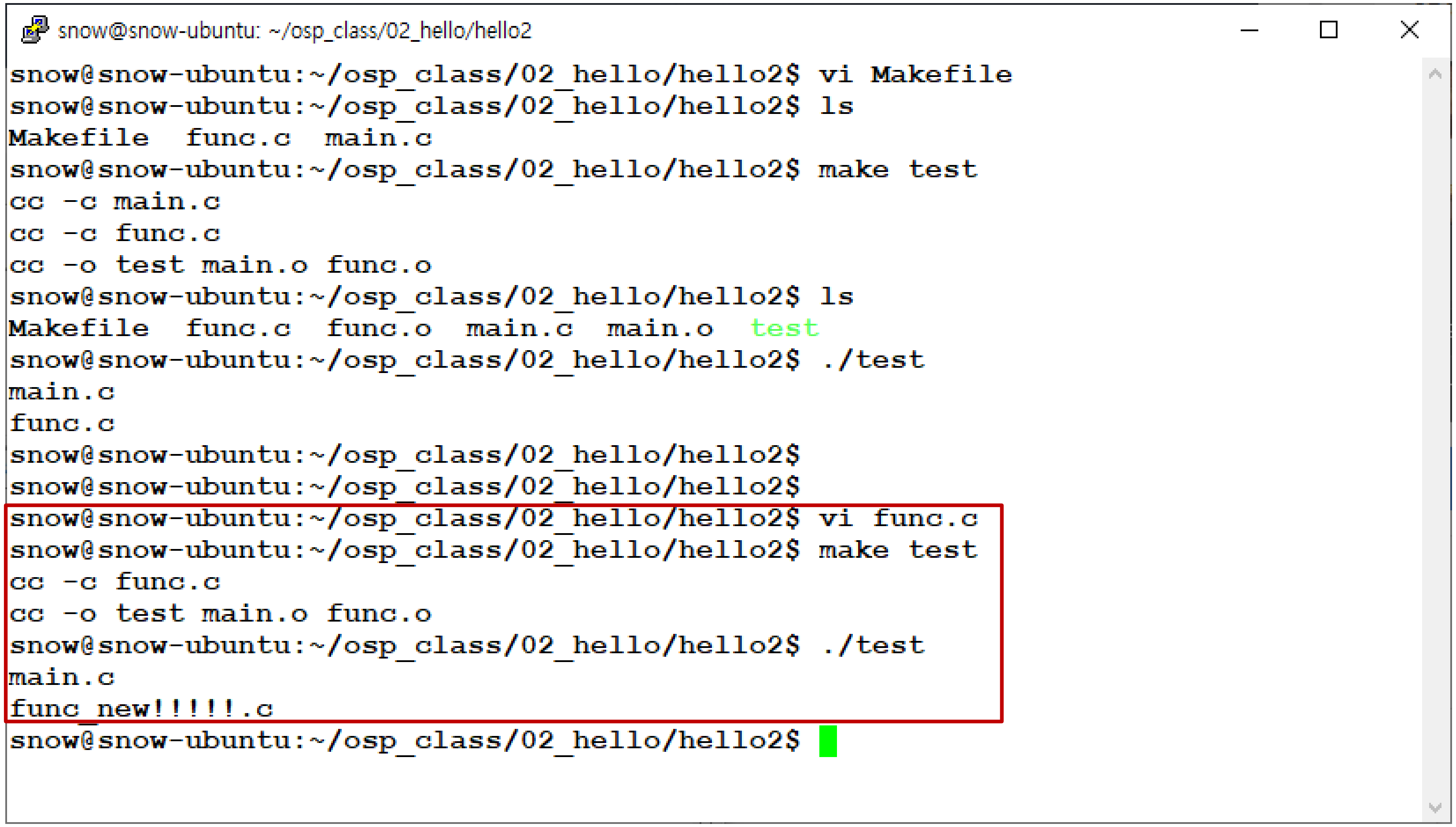
```
#include <stdio.h>

void func(void)
{
	printf("func.c\n");
}
```





```
#include <stdio.h>

void func(void)
{
	printf("func_new!!!!!.c\n");
}
```

- **$ make test**
- **$ ./test**



snow@snow-ubuntu: ~/osp_class/02_hello/hello2

```
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ vi Makefile
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ ls
Makefile  func.c  main.c
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ make test
cc -c main.c
cc -c func.c
cc -o test main.o func.o
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ ls
Makefile  func.c  func.o  main.c  main.o  test
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ ./test
main.c
func.c
snow@snow-ubuntu:~/osp_class/02_hello/hello2$
snow@snow-ubuntu:~/osp_class/02_hello/hello2$
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ vi func.c
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ make test
cc -c func.c
cc -o test main.o func.o
snow@snow-ubuntu:~/osp_class/02_hello/hello2$ ./test
main.c
func_new!!!!!.c
snow@snow-ubuntu:~/osp_class/02_hello/hello2$
```

Any Questions...
Just Ask!