

시스템 프로그래밍 실습

1차 과제 - Linux 기초





수업 일시 : 화 1,2
학과 : 컴퓨터 공학과
이름 : 최재은
학번 : 2013722095

1-1

■ Introduction

- 수업 및 과제 내용을 진행하기 위한 가상머신을 설치하고, 그 과정을 캡처하여 설명한다.

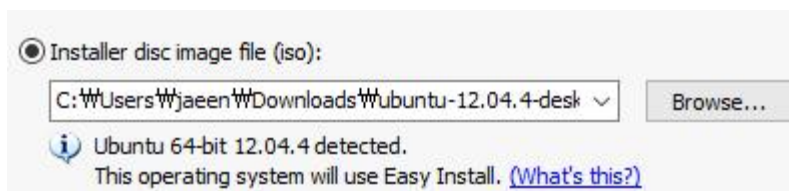
■ Result

	ubuntu-12.04.4-desktop-amd64	2017-03
	VMware-player-12.1.1-3770994	2017-03

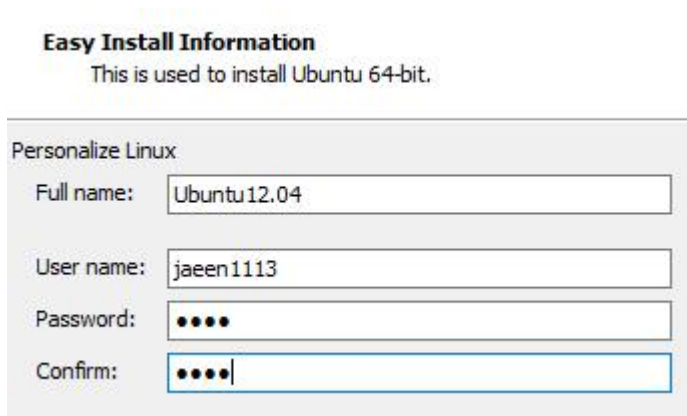
- ubuntu v12.04와 그것을 설치할 때 사용할 환경 tool을 준비한다.



- VMware를 설치하고 나서 위의 section을 클릭해서 새로 추가할 가상머신을 선택



- 두 번째 항목의 install image file을 선택하고 Browse...를 클릭해 이미지 파일이 저장되어 있는 폴더에서 ubuntu-12.04.4파일을 선택



- 아래의 내용을 적당히 입력해 주고 next 클릭

Virtual machine name:

Location:

- 설치할 가상머신(Ubuntu)를 어느 폴더에 저장해둘 것인지를 선택

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB):

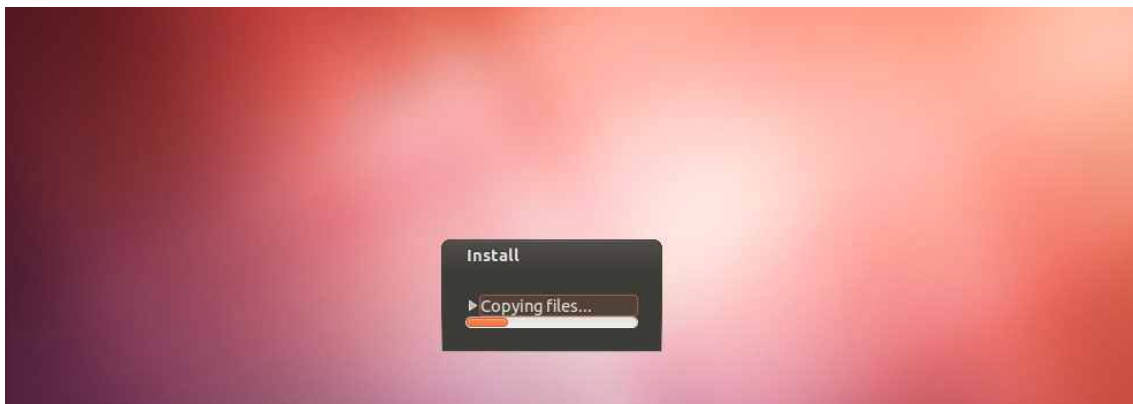
Recommended size for Ubuntu 64-bit: 20 GB

☐ Store virtual disk as a single file

☒ Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

- 해당 머신의 디스크 용량을 정해주는 부분, 넉넉하게 50정도 잡아주었다.
- 다음에 나오는 box에서 Finish를 눌러주면 아래와 같이 설치가 진행된다.



Memory	2 GB
Processors	2
Hard Disk (SCSI)	50 GB
CD/DVD (SATA)	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Specify the amount of memory for this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine:

64 GB -
32 GB -
16 GB -
8 GB -
4 GB -
2 GB -

- 설치를 마치면 위와 같이 Processor를 몇 개나 사용할 것인지, Ram은 어느 정도 끌어다가 쓸 것인지 설정할 수 있다. 사실 설치 전에 setting할 수도 있지만 깜빡했다.

#1-2

■ Introduction

- 강의자료 3에서 소개한 Linux명령어를 사용해보고 어떤 기능을 제공하는지 알아본다.

■ Result

1) man

```
jaeen1113@ubuntu:~/Downloads$ man pwd
```

위와 같은 man 명령어 뒤에 다른 명령어를 입력하면 아래 그림과 같이 해당 명령어의 사용 manual에 대해 보여준다.

```
NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by
    default).  Sort entries alphabetically if none of -cftuvSUX
    nor --sort is specified.

    Mandatory arguments to long options are mandatory for short
    options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..
```

2) cat

- 터미널 상의 텍스트 파일을 보는 명령어

```
jaeen1113@ubuntu:~/Documents$ cat 123.txt
file for cat command ^ ^
```

- 뒤에 파일이름 및 형식을 적어주면 해당 파일의 text를 출력해준다.

3) pwd

- 현재 위치의 경로를 보여주는 명령어

```
jaeen1113@ubuntu:~/Documents$ pwd
/home/jaeen1113/Documents
```

4) cd

- Current Directory의 약자로 디렉토리를 이동할 때 사용하는 명령어
- cd만 입력 시에 root Directory로 이동
- cd .. 입력 시 상위 Directory로 이동
- cd + Directory name 입력 시 해당 Directory로 이동(단, 해당 폴더가 현재 Directory에 존재할 경우)

```
jaeen1113@ubuntu:~/Documents$ cd ..  
jaeen1113@ubuntu:~$
```

(Documents directory의 상위 directory인 root로 이동하였다.)

```
jaeen1113@ubuntu:~$ cd Documents/happy/wheels/HoneyJam  
jaeen1113@ubuntu:~/Documents/happy/wheels/HoneyJam$
```

(cd 뒤에 찾아갈 Directory의 경로를 입력하면 그곳으로 이동한다.)

```
jaeen1113@ubuntu:~/Documents/happy/wheels/HoneyJam$ cd  
jaeen1113@ubuntu:~$
```

(cd만 입력 시 최상위 Directory인 root로 이동한다.)

5) ls

- 현재 Directory의 내용물을 보여주는 명령어

```
jaeen1113@ubuntu:~/Documents$ ls  
123.txt 123.txt~ happy
```

6) chmod

- 파일이나 디렉토리의 permission을 관리하는 명령어
- r: 읽기 권한(4) / w: 쓰기 권한(2) / x: 실행권한(1)이다

```
-rw-rw-r-- 1 jaeen1113 jaeen1113 25 Mar 12 00:17 123.txt
```

- 위를 보면 123.txt 파일이 664의 권한 값을 갖는데 이는 [디렉토리1][소유자권한 3][그룹권한3][전체권한3]으로 나뉘어서 읽을 수 있다.

- 아래처럼 명령어를 입력하면 그 권한이 바뀌는 것을 확인할 수 있다.

```
jaeen1113@ubuntu:~/Documents$ chmod 750 123.txt  
jaeen1113@ubuntu:~/Documents$ ls -al  
total 16  
drwxr-xr-x 3 jaeen1113 jaeen1113 4096 Mar 12 00:23 .  
drwxr-xr-x 19 jaeen1113 jaeen1113 4096 Mar 12 00:05 ..  
-rwxr-x--- 1 jaeen1113 jaeen1113 25 Mar 12 00:17 123.txt
```


7) mkdir

- 새로운 Directory를 만드는 명령어

```
jaeen1113@ubuntu:~/Downloads$ ls
jaeen1113@ubuntu:~/Downloads$ mkdir 123
jaeen1113@ubuntu:~/Downloads$ ls
123
```

- 123이라는 이름의 Directory가 만들어졌다.

8) rmdir

- Directory를 삭제하는 명령어

```
jaeen1113@ubuntu:~/Downloads$ rmdir 123
jaeen1113@ubuntu:~/Downloads$ ls
jaeen1113@ubuntu:~/Downloads$
```

- mkdir로 만들었던 123 directory가 삭제되었다.

9) rm

- 파일이나 Directory를 삭제하는 명령어

```
jaeen1113@ubuntu:~/Documents$ ls
123.txt 123.txt~ happy
jaeen1113@ubuntu:~/Documents$ rm 123.txt
jaeen1113@ubuntu:~/Documents$ ls
123.txt~ happy
```

- 123.txt 파일을 지워보았다.

10) cp

- 파일이나 디렉토리를 복사할 때 사용하는 명령어
- cp [옵션][복사할 대상][복사할 위치]
- 아래와 같은 옵션으로 명령어를 수행할 수 있다.

-i : 복사대상 파일이 있을 경우, 사용자에게 복사에 대한 실행여부를 묻는다.
-f : 복사대상 파일이 있을 경우, 사용자에게 확인없이 강제로 복사한다.
-r : 디렉토리를 복사할 경우 하위 디렉토리도 모두 복사한다.
-v : 복사진행 상태를 출력한다.
-d : 복사대상 파일이 심볼릭파일이면, 심볼릭 정보를 그대로 유지한 상태로 복사한다.
-p : 원본파일의 소유주, 그룹, 권한, 시간정보를 보존하여 복사한다.
-a : 원본파일의 속성, 링크정보를 그대로 유지하면서 복사한다.

```
jaeen1113@ubuntu:~/Documents$ ls
123.txt~ happy
jaeen1113@ubuntu:~/Documents$ cd happy
jaeen1113@ubuntu:~/Documents/happy$ ls
sss.txt wheels
jaeen1113@ubuntu:~/Documents/happy$ cp -f sss.txt ~/Documents
jaeen1113@ubuntu:~/Documents/happy$ cd ..
jaeen1113@ubuntu:~/Documents$ ls
123.txt~ happy sss.txt
```

- happy Directory에 있던 sss.txt 파일이 Documents directory로 복사되었다.

11) mv

- 파일이나 디렉토리를 이동(잘라내기)할 때 사용하는 명령어

```
jaeen1113@ubuntu:~/Documents$ ls
123.txt~ happy sss.txt
jaeen1113@ubuntu:~/Documents$ cd happy
jaeen1113@ubuntu:~/Documents/happy$ ls
sss.txt wheels
jaeen1113@ubuntu:~/Documents/happy$ mv wheels ~/Documents
jaeen1113@ubuntu:~/Documents/happy$ ls
sss.txt
jaeen1113@ubuntu:~/Documents/happy$ cd ..
jaeen1113@ubuntu:~/Documents$ ls
123.txt~ happy sss.txt wheels
jaeen1113@ubuntu:~/Documents$
```

- happy에 있던 wheels Directory가 Documents Directory로 이동되었다.

12) ln

- 파일 시스템에서 링크파일을 만드는 명령어
- Symbolic / Hard link 두 가지 형태가 존재한다.
- Symbolic link는 원본파일을 가리키도록 링크만 시켜둔 것으로 '바로가기' 역할
- Hard link는 원본파일과 다른 이름으로 존재하는 파일. 동일한 내용의 다른 파일임.

```
jaeen1113@ubuntu:~/Documents$ ls
happy sss.txt wheels
jaeen1113@ubuntu:~/Documents$ ln sss.txt hard_link
jaeen1113@ubuntu:~/Documents$ ls
happy hard_link sss.txt wheels
```

- 위와 같이 sss.txt의 hard link를 만들고 내용을 확인하면 아래와 같이 동일함을 확인할 수 있음.

```
happy hard_link sss.txt wheels
jaeen1113@ubuntu:~/Documents$ cat sss.txt
a
b
c
d

jaeen1113@ubuntu:~/Documents$ cat hard_link
a
b
c
d
```

13) touch

- 파일의 날짜와 시간정보를 변경하는 명령어

```
jaeen1113@ubuntu:~/Documents$ touch -t 120012152014 sss.txt
jaeen1113@ubuntu:~/Documents$ ls -al
total 24
drwxr-xr-x  4 jaeen1113 jaeen1113 4096 Mar 12 18:03 .
drwxr-xr-x 20 jaeen1113 jaeen1113 4096 Mar 12 18:03 ..
drwxrwxr-x  2 jaeen1113 jaeen1113 4096 Mar 12 17:58 happy
-rw-rw-r--  2 jaeen1113 jaeen1113    7 Dec 15 1200 hard_link
-rw-rw-r--  2 jaeen1113 jaeen1113    7 Dec 15 1200 sss.txt
drwxrwxr-x  3 jaeen1113 jaeen1113 4096 Mar 12 17:24 wheels
```

- sss.txt 파일의 날짜정보를 12월 15일로 바꾸었다.
- hard_link 역시 내용이 동기화되므로 날짜정보가 같이 바뀌었다.
- touch 명령어 뒤에 별다른 옵션을 주지 않고 파일명을 입력하면 linux의 현재시간으로 파일의 날짜 정보를 동기화시킨다.

14) exit

- 현재 shell을 닫는 명령어,

```
3542 pts/0    00:00:00 csh
3551 pts/0    00:00:00 ps
[1] + Terminated          yes
% exit
% exit
[1]+  Aborted              (core dumped) yes  (wd: ~/SP/assign1)
(wd now: ~)
jaeen1113@ubuntu:~$
```

- csh shell 환경을 종료하였다.

15) kill

- 프로세스에 특정 signal을 보내는 명령어로 보통 실행중인 프로세스에 종료신호를 보낸다. 일반적인 방법으로 중지하지 못한 프로세스를 종료시킬 때 주로 사용
- kill [option] [-signal number or -signal name] PID
- 아래와 같이 csh shell 환경에서 yes 라는 process를 실행시키고 kill하였다.

```
% ps
  PID TTY          TIME CMD
 3048 pts/0    00:00:00 bash
 3106 pts/0    00:00:00 yes
 3542 pts/0    00:00:00 csh
 3543 pts/0    00:00:00 yes
 3545 pts/0    00:00:00 ps
% kill 3543 3106
% ps
  PID TTY          TIME CMD
 3048 pts/0    00:00:00 bash
 3542 pts/0    00:00:00 csh
 3551 pts/0    00:00:00 ps
[1] + Terminated          yes
%
```


16) ps

- 현재 수행되고 있는 프로세스를 확인하는 명령어

```
jaeen1113@ubuntu:~$ ps
  PID TTY          TIME CMD
  4565 pts/0        00:00:00 bash
  4625 pts/0        00:00:00 ps
```

- 위와 같이 프로세스 ID 등의 정보를 보여준다.

17) pstree

- 프로세스의 상관관계를 트리형태로 출력해준다.

```
jaeen1113@ubuntu:~$ pstree
init--ManagementAgent--2*[{ManagementAgent}]
    |
    |--NetworkManager--dhclient
    |                   |
    |                   |--dnsmasq
    |                   --2*[{NetworkManager}]
    |
    |--VGAuthService
    |--accounts-daemon--{accounts-daemon}
    |--acpid
    |--anacron
    |--atd
    |--bamfdaemon--2*[{bamfdaemon}]
    |--bluetoothd
```

- 계층 관계를 한 눈에 파악할 수 있다.

18) passwd

- 계정의 비밀번호를 변경하는 명령어

```
jaeen1113@ubuntu:~$ passwd
Changing password for jaeen1113.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
Bad: new and old password are too similar
Enter new UNIX password:
Retype new UNIX password:
You must choose a longer password
Enter new UNIX password:
Retype new UNIX password:
Bad: new and old password are too similar
passwd: Authentication token manipulation error
passwd: password unchanged
```

- 기존 비밀번호를 입력하고 새 비밀번호를 입력하는데, 위에서 보드시피 기존의 것과 너무 유사하다, 너무 짧다든지 생각보다 깐깐했다.

19) uname

- 현재 작동중인 가상머신과 운영체제에 대한 정보를 출력한다.

```
jaeen1113@ubuntu:~$ uname  
Linux
```

20) wc

- 문서가 몇 줄, 단어로 이루어져 있는지를 세어주는 명령어

```
apple is red  
banana is yellow  
comgong is TT
```

- 위와 같은 내용의 text file을 아래와 같이 나타내 준다.

```
jaeen1113@ubuntu:~/Documents$ wc sss.txt  
3  9 44 sss.txt
```

- 3줄 / 9 단어 / 44 글자 / 파일 이름 의 형식

21) more

- 한번에 보기 어려운 긴 내용의 문서들을 읽기 쉽게 표현해 주는 명령어

```
jaeen1113@ubuntu:~/Documents$ more -5 sss.txt  
apple is red  
banana is yellow  
comgong is TT  
i  
am  
--More--(60%)
```

- 위와 같이 -5로 옵션을 주면 한번에 5줄씩 보여주는데 그 아래의 글들은 more처리가 된다.

22) echo

- 텍스트를 출력해주는 기능을 가진 명령어
- 아래와 같이 환경변수를 출력해줄 수도 있다.

```
jaeen1113@ubuntu:~$ echo $PATH  
/usr/lib/lightdm/lightdm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/u  
sr/bin:/sbin:/bin:/usr/games
```

23) alias

- 자주 사용하는 명령어를 단축키로 설정할 때 사용한다.

```
jaeen1113@ubuntu:~$ vim /home/jaeen1113/.bashrc
```

- 위의 bash file로 들어가서 아래와 같이 입력해주고

```
[root@localhost ~]# /home/jaeen1113/.bashrc
alias
alias gag='cd Documents'
[root@localhost ~] source ~/.bashrc
```

- bash file을 다시 불러들이고나서

```
jaeen1113@ubuntu:~$ source ~/.bashrc
```

- 등록했던 명령어 gag를 치면 아래와 같이 Documents Directory로 이동한다.

```
jaeen1113@ubuntu:~$ gag
jaeen1113@ubuntu:~/Documents$
```

- 또는 아래의 사진처럼 명령어를 사용하여 등록하거나 해제 할 수 있다.

```
jaeen1113@ubuntu:~$ alias gogo='cd SP/assign1'
jaeen1113@ubuntu:~$ gogo
jaeen1113@ubuntu:~/SP/assign1$ unalias gogo
jaeen1113@ubuntu:~/SP/assign1$ gogo
No command 'gogo' found, did you mean:
  Command 'gogoc' from package 'gogoc' (universe)
  Command 'logo' from package 'ucblog' (universe)
  Command 'goo' from package 'goo' (universe)
gogo: command not found
jaeen1113@ubuntu:~/SP/assign1$
```

24) find

- 디스크에 저장되어 있는 각종 파일/디렉토리를 검색하는 명령어

```
jaeen1113@ubuntu:~/Documents$ find -name '*.txt' -ls
658557      4 -rw-rw-r--  2 jaeen1113 jaeen1113      81 Mar 12 18:32
./sss.txt
```

- 위의 그림은 현재 directory(Documents)에서 .txt.확장자를 가진 파일들을 -ls형태로 출력하는 명령어이다. -name 앞에 /를 붙여서 전체 디스크에서 탐색을 진행할 수도 있다.

25) grep

- 지정한 파일 내에서 특정 문자열을 검색할 때 사용

```
jaeen1113@ubuntu:~/Documents$ grep a sss.txt
apple is red
banana is yellow
am
a
are
a
yeah
```

- sss.txt파일에서 a가 들어가는 문자열을 보여준다.

- 명령어나 파일의 실행에 소요되는 시간을 보여주는 명령어
- 아래의 그림은 ps 명령어의 실제 소요시간을 보여준다.

- 커널에서 실행된 시간이 실제 소요 시간임을 확인할 수 있다.

- window의 notepad와 비슷하게 파일을 편집하는 기능을 제공한다.

```
VIM - Vi IMproved

        version 7.3.429
        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
```

```
--
--      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
--
-- -- INSERT --
```

0.1

- **I**를 누르면 입력모드로 전환되며 커서 위치 앞에서 삽입된다.

* 삭제

- 아래의 그림을 보면 현재 커서가 o에 있고 x를 누르면 현재 커서에 있는 문자가 삭제된다.

```
apple
banana
pineapple
orange
mango
```

```
apple
banana
pineapple
orange
man
```

* 데이터 저장하기

- fruit라는 이름의 파일로 저장하면

```
~
~
:w fruit.txt
```

```
jaeen1113@ubuntu:~$ ls
Desktop  Downloads  fruit.txt  Pictures
```

- 위와 같이 저장되는 것을 확인할 수 있다.

* 검색

- 입력한 패턴에 대해서 검색

```
apple
banana
pineapple
orange
mang
```

```
apple
banan
pineapple
orange
mang
```

```
apple
banana
pineapple
orange
mang
```

```
apple
banana
pineapple
orange
man
```

- 위의 그림들은 an에 대해 검색한 내용이다. 다음으로 넘어가기 위해 n을 누르면 된다.

* 치환

- 위의 파일중 'app'의 문자열에 대해 '00'으로 치환한 그림이다.

```
~
~
:%s/app/00
```

```
00le
banana
pine00le
orange
mang
```

27) make

- 컴파일 과정을 자동화하기 위해 사용한다.

```
#include <stdio.h>
void main ()
{
    printf("Hello! System Pro.!\n");
}
```

- 위와 같은 내용을 출력하는 .c 파일을 만들고

```
hi_run: hi.c
        gcc hi.c -o hi_run
clean:
        rm *.o hi_run
~
```

- 위와 같이 hi_run이라는 이름의 target에 hi.c를 실행하도록 Makefile을 설정.
- clean 명령어 실행 시 .o 파일을 삭제하도록 하였다.

```
jaeen1113@ubuntu:~/sys$ make
gcc hi.c -o hi_run
jaeen1113@ubuntu:~/sys$ ls
hi.c hi_run Makefile
```

- make 명령을 하면 위와 같이 target file인 hi_run이 만들어 진다.

```
jaeen1113@ubuntu:~/sys$ ./hi_run
Hello! System Pro.!
```

- hi_run 파일을 실행하면 위와 같이 문구가 출력된다.

28) gdb

- 프로그램을 실행하면 프로그램 내에서 어떤 일이 일어나는지 보여주는 툴로
- aa.c 파일을 컴파일하여 a.out을 만들고 10번째 line에 breaking point를 만든 뒤, 해당 파일을 실행한다.

```
jaeen1113@ubuntu:~/sys$ gcc -g aa.c
jaeen1113@ubuntu:~/sys$ gdb a.out
GNU gdb (Ubuntu/Linaro 7.4-2012.04-0ubuntu2.1) 7.4-2012.04
Copyright (C) 2012 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. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
For bug reporting instructions, please see:
<http://bugs.launchpad.net/gdb-linaro/>...
Reading symbols from /home/jaen1113/sys/a.out...done.
(gdb) b 10
Breakpoint 1 at 0x40059f: file aa.c, line 10.
(gdb) r hell! lab.
Starting program: /home/jaen1113/sys/a.out hell! lab.
warning: no loadable sections found in added symbol-file system-supplied DSO
```

```

Breakpoint 1, main () at aa.c:11
11             for(b= 0; b<10; b++)
(gdb) n
13             printf(" *");}
(gdb) step
11             for(b= 0; b<10; b++)
(gdb) print1
Undefined command: "print1". Try "help".
(gdb) print a
$1 = 0
(gdb) step
13             printf(" *");}
(gdb) print b
$2 = 1

```

- 10번째 line이 { 뿐이라 11 line으로 넘어갔다.
- n과 step을 눌러 다음 동작이 수행되게 해보았다.
- a와 b를 print하니 현 상태에서 각각의 값들이 출력되는 것을 볼 수 있다.

```

(gdb) jump 16
Continuing at 0x4005d8.
END!

```

- 16 line으로 jump하자 END가 출력되었다.

```

(gdb) q
jaeen1113@ubuntu:~/sys$

```

- q를 입력해서 gdb 탈출
- 사용한 소스는 아래와 같다.

```

#include <stdio.h>
#include <string.h>

void main ()
{
    int a;
    int b;
    printf(" SQUARE \n");
    for(a =0; a<10; a++)
    {
        for(b= 0; b<10; b++)
        {
            printf(" *");}
        printf("\n");
    }
    printf("END!\n");
}

```

■ Reference

- <http://lovelydays95.tistory.com/entry/cp%EB%AA%85%EB%A0%B9%EC%96%B4>(리눅스 명령어)
- <http://webdir.tistory.com/148>