



C basic language

임베디드스쿨 2기

Lv1과정

2021. 03. 12

김효창

리눅스 명령어 복습 (내용)

항 목	내 용	입 력
사진 1	list (ls) : 현재 디렉토리에 있는 내용을 출력 recursive (-R) : 현재 위치부터 하위 디렉토리까지 출력 path (/) : 디렉토리 구분 기호	ls proj : proj 폴더에 있는 내용을 출력
사진 2	all (-a) : 숨겨진 파일과 디렉토리 포함 모두 출력 온점 (.) : 현재 위치 (..) : 상위 디렉토리 copy (cp) : 파일 생성 후 동일하게 만들고자 하는 파일을 지정 후 내용 복사	ls -a cd proj/es02/Lv01-02/HyochangKim : HyochangKim 디렉토리로 이동 cp practice practice2 : cp 로는 폴더를 복사 할 수 없으니 cp + -r 을 사용! cp -r : 디렉토리 전체를 복사 (recursive)
사진 3, 4	adduser : 계정 추가 deluser : 계정 삭제	mkdir -p : 상위, 하위 디렉토리 만들 수 있다. cd ../.. : 상위 2칸 이동 pwd : 내가 어디에 있는지 알려준다. cd ~ : 사용자 home 디렉토리로 이동 sudo adduer (deluser) + 계정명
사진 5, 6	remove (rm) : 파일 삭제 recursive force (-rf) : 디렉토리를 강제로 삭제 visual (v) : 현재 커서부터 선택 visual (V) : 줄 (라인) 단위 선택 yank (y) : vim 복사 put (p) : vim 붙여 넣기 set number (:set nu) : vim 라인 번호 출력 set nonumber (:set nonu) : vim 라인 번호 해제 Insert (i) : vim 끼워 넣기 write : 디스크에 쓰기 quit : vim 종료	vim 텍스트 편집기 순서대로 y + 커서 시작 부분에서 어디까지 숫자 + y + 원하는 줄에 p (예 : 숫자 8 입력 시 0~7 까지) D 숫자 d : 삭제 : \$ 입력 시 마지막 줄로 이동 차카뎃 esc + u : vim 실행 취소 ctrl + r : vim 재실행 ctrl + F (forward) : 다음 페이지 ctrl + B (Back) : 이전 페이지
사진 7, 8	substitute (%s) : vim 치환 [옵션] global (g) : 전체 (전역) confirm (c) : 검색된 모든 문자열에 대해 변경 여부 확인	shift + i : 해당 라인 커서 맨 앞으로 이동하면서 끼워 넣기 모드 진입 shift + a : 해당 라인 커서 맨 뒤로 이동 %s / [원본단어] / [변경할 단어] / [옵션]
사진 9		-o : 출력파일의 이름 지정하는 옵션, 지정하지 않으면 모든 이름은 a.out 으로 생성 gcc (소스파일이름) -o (실행파일이름) 프로그램 실행 시 ./b.out (주의 디렉토리 경로 확인 필수)

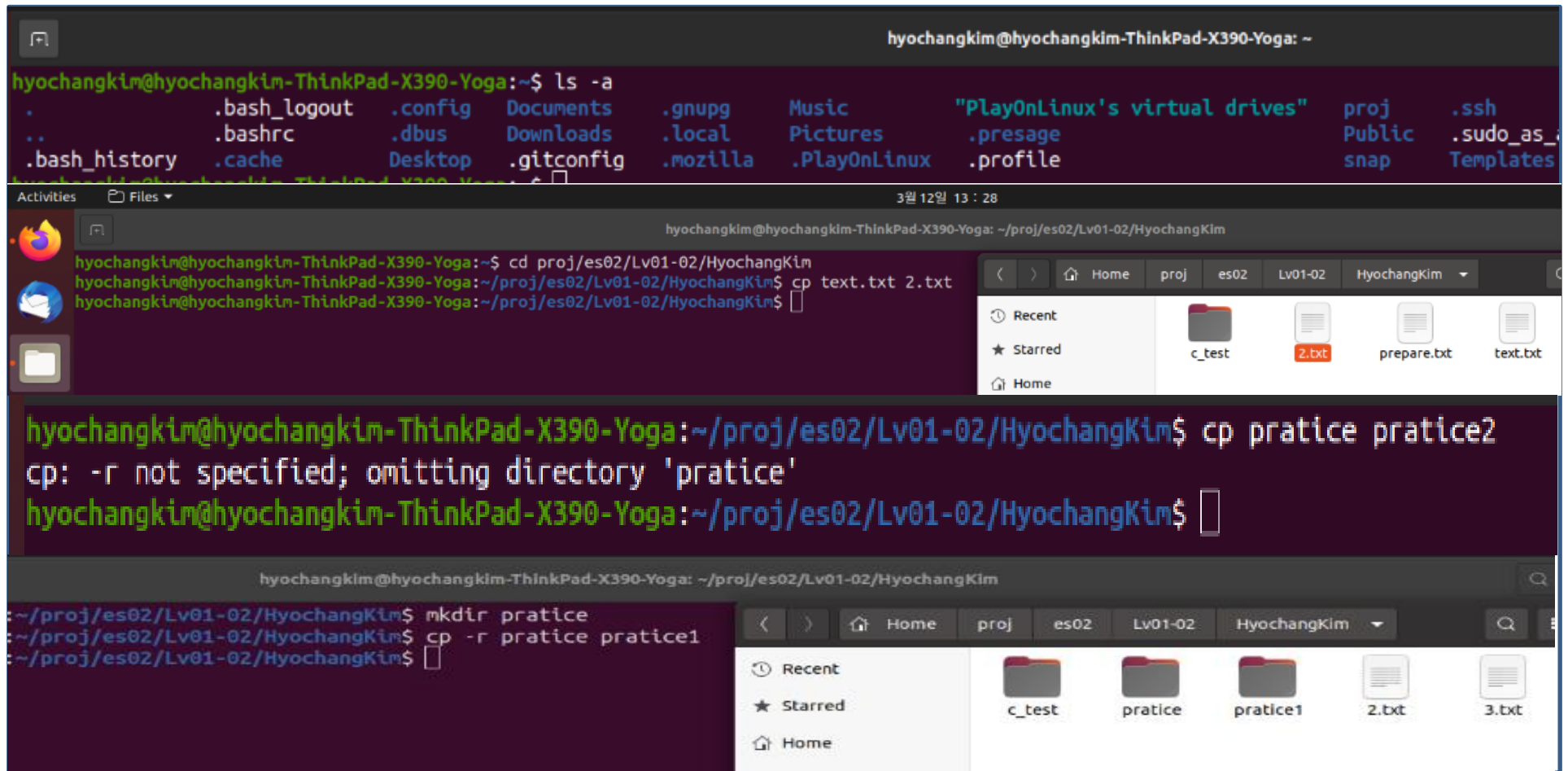
리눅스 명령어 복습 (내용)

항 목	내 용	입 력

리눅스 명령어 복습 과정 (사진 1)

```
hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~  
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~$ ls  
Desktop Documents Downloads Music Pictures "PlayOnLinux's virtual drives" proj Public snap Templates Videos  
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~$ ls proj  
es02  
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~$  
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~$ ls -R  
.:  
Desktop Documents Downloads Music Pictures "PlayOnLinux's virtual drives" proj Public snap Templates Videos  
./Desktop:  
./Documents:  
Git2.odp Git.pptx 'Untitled 1.odt'  
./Downloads:  
'file send.pptx' 'file sends.pdf' 'FoxitReader.enu.setup.2.4.4.0911(r057d814).x64.run' FoxitReader.enu.setup.2.4.4.0911.x64.run.tar  
./Music:  
./Pictures:  
'Screenshot from 2021-03-06 10-37-15.png' 'Screenshot from 2021-03-12 13-17-35.png'  
./proj:  
es02  
./proj/es02:  
Lv01-02  
./proj/es02/Lv01-02:  
ChungjaeLee HyochangKim HyunhoCha init.md LICENSE README.md TaeinPark
```

리눅스 명령어 복습 과정 (사진 2)



The screenshot displays a Linux environment with a terminal window and a file manager. The terminal window shows the following commands and output:

```
hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~$ ls -a
.          .bash_logout  .config  Documents  .gnupg    Music      "PlayOnLinux's virtual drives"  proj  .ssh
..         .bashrc       .dbus    Downloads  .local    Pictures   .presage  Public  .sudo_as_
.bash_history  .cache       Desktop  .gitconfig  .mozilla  .PlayOnLinux  .profile  snap    Templates

hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~/proj/es02/Lv01-02/HyochangKim$ cd proj/es02/Lv01-02/HyochangKim
hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~/proj/es02/Lv01-02/HyochangKim$ cp text.txt 2.txt
hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~/proj/es02/Lv01-02/HyochangKim$

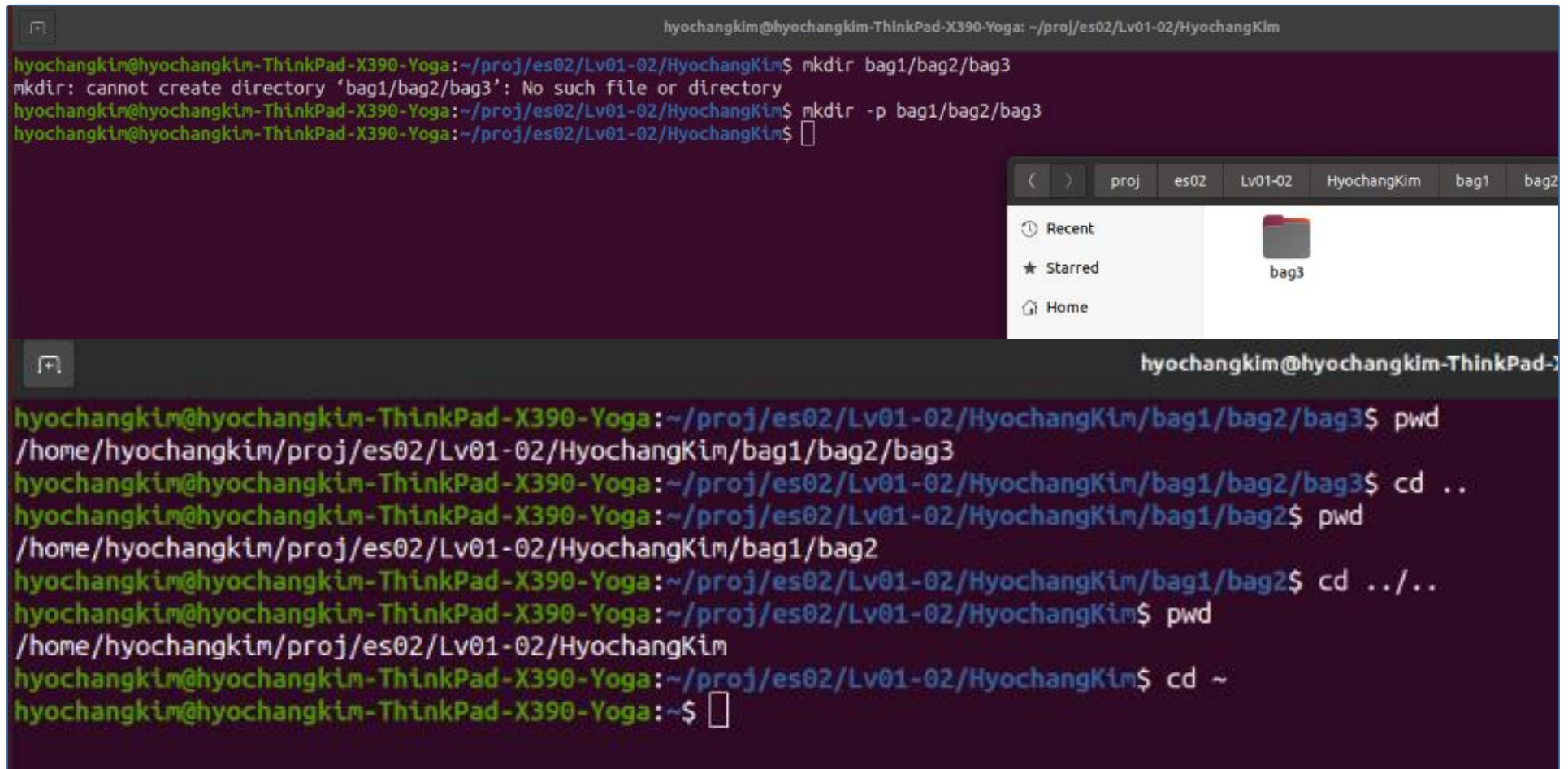
hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~/proj/es02/Lv01-02/HyochangKim$ cp pratice pratice2
cp: -r not specified; omitting directory 'pratice'
hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~/proj/es02/Lv01-02/HyochangKim$

hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~/proj/es02/Lv01-02/HyochangKim$ mkdir pratice
hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~/proj/es02/Lv01-02/HyochangKim$ cp -r pratice pratice1
hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~/proj/es02/Lv01-02/HyochangKim$
```

The file manager window shows the directory structure and files:

- Initial state: The file manager shows the directory `~/proj/es02/Lv01-02/HyochangKim` with files `c_test`, `2.txt`, `prepare.txt`, and `text.txt`.
- After `cp pratice pratice2`: The file manager shows the directory `~/proj/es02/Lv01-02/HyochangKim` with files `c_test`, `pratice`, `pratice1`, `2.txt`, and `3.txt`.

리눅스 명령어 복습 과정 (사진 3)



The image displays two screenshots from a Linux environment. The top screenshot shows a terminal window with the following commands and output:

```
hyochangkim@hyochangkim-ThinkPad-X390-Yoga: ~/proj/es02/Lv01-02/HyochangKim
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim$ mkdir bag1/bag2/bag3
mkdir: cannot create directory 'bag1/bag2/bag3': No such file or directory
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim$ mkdir -p bag1/bag2/bag3
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim$
```

Below the terminal is a file manager window showing the directory structure. The breadcrumb path is: < > proj es02 Lv01-02 HyochangKim bag1 bag2. The main area shows a folder named 'bag3' under the 'bag2' directory. The left sidebar has 'Recent', 'Starred', and 'Home' options.

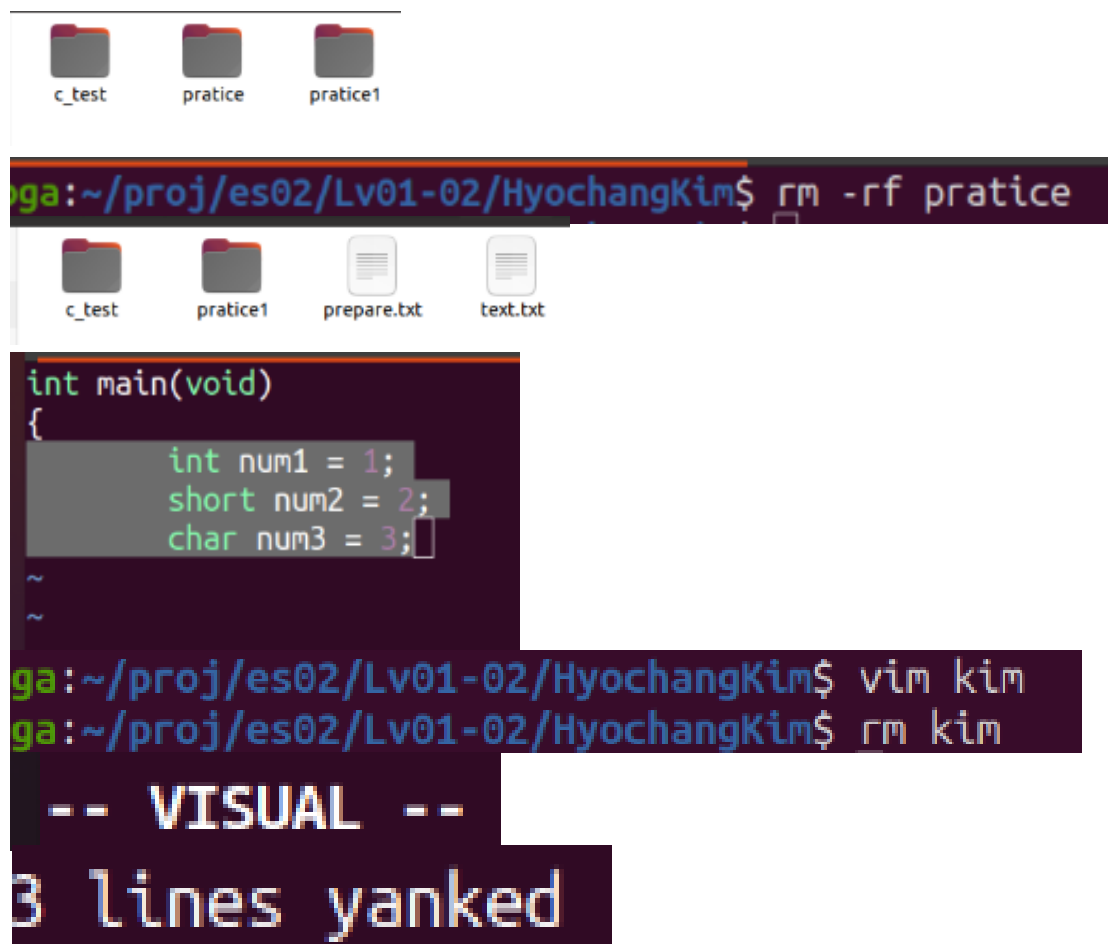
The bottom screenshot shows the terminal again, demonstrating directory navigation:

```
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim/bag1/bag2/bag3$ pwd
/home/hyochangkim/proj/es02/Lv01-02/HyochangKim/bag1/bag2/bag3
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim/bag1/bag2/bag3$ cd ..
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim/bag1/bag2$ pwd
/home/hyochangkim/proj/es02/Lv01-02/HyochangKim/bag1/bag2
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim/bag1/bag2$ cd ../../
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim$ pwd
/home/hyochangkim/proj/es02/Lv01-02/HyochangKim
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim$ cd ~
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~$
```

```
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~$ sudo adduser head
[sudo] password for hyochangkim:
Adding user 'head' ...
Adding new group 'head' (1001) ...
Adding new user 'head' (1001) with group 'head' ...
The home directory '/home/head' already exists.  Not copying from '/etc/skel'.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for head
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~$ sudo deluser head
Removing user 'head' ...
Warning: group 'head' has no more members.
Done.
```

```
head@hyochangkim-ThinkPad-X390-Yoga:~$ pwd
/home/head
head@hyochangkim-ThinkPad-X390-Yoga:~$ 
```

리눅스 명령어 복습 과정 (사진 5)



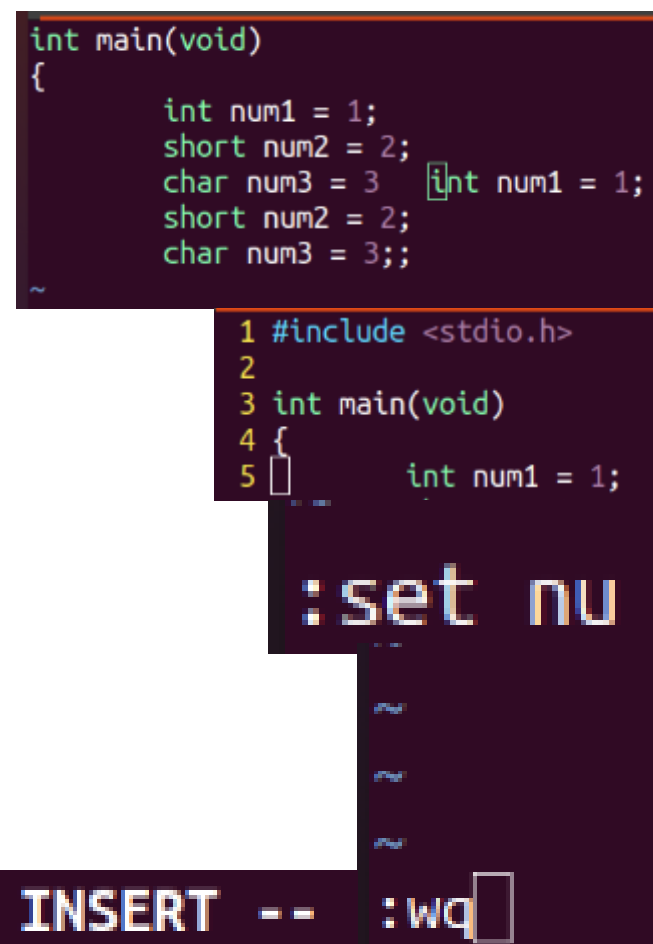
```
ga:~/proj/es02/Lv01-02/HyochangKim$ rm -rf pratice
```

```
int main(void)
{
    int num1 = 1;
    short num2 = 2;
    char num3 = 3;
}

~
~

ga:~/proj/es02/Lv01-02/HyochangKim$ vim kim
ga:~/proj/es02/Lv01-02/HyochangKim$ rm kim

-- VISUAL --
3 lines yanked
```



```
int main(void)
{
    int num1 = 1;
    short num2 = 2;
    char num3 = 3;
}

~
~

1 #include <stdio.h>
2
3 int main(void)
4 {
5     int num1 = 1;
}

:set nu

-- INSERT -- :wq
```



```
30
31 0
32 1
33 2
34 3
35 4
36 5
37 6
38 7
39
40
41 0
42 1
43 2
44 3
45 4
46 5
47 6
48 7
49

41 0
42 1
43 2
44 3
45 4
46 5
47 6
48 7
49

51 0
52 1
53 2
54 3
55 4
56 5
57 6
58 7

0
1
2
3
4
5
6
7
...
:set nonu

8 lines yanked
```

```
long long int num6 = 10000000000;  
long double num7 = 1942723.2234;
```

```
long long int num6 = 10000000000;  
long double num7 = 1942723.2234;
```

```
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim/c_test/01$ gcc homework.c -o b.out  
homework.c: In function 'main':  
homework.c:26:26: warning: format '%d' expects argument of type 'int', but argument 2 has type 'long unsigned int' [-Wformat=]  
26 | printf("sizeof(char) = %d\n", sizeof(char));  
   | ~^ ~~~~~  
   | int long unsigned int  
   | %ld  
homework.c:27:27: warning: format '%d' expects argument of type 'int', but argument 2 has type 'long unsigned int' [-Wformat=]  
27 | printf("sizeof(short) = %d\n", sizeof(short));  
   | ~^ ~~~~~  
   | int long unsigned int
```

```
printf("num1 = %lu\n", num1);  
printf("num2 = %lu\n", num2);  
printf("num3 = %lu\n", num3);  
  
printf("num4 = %f\n", num4);  
printf("num4 = %.2f\n", num4);  
printf("num5 = %lf\n", num5);  
  
printf("num6 = %lld\n", num6);  
printf("num7 = %Lf\n", num7);  
  
printf("sizeof(char) = %lu\n", sizeof(char));  
printf("sizeof(short) = %lu\n", sizeof(short));  
printf("sizeof(int) = %lu\n", sizeof(int));  
printf("sizeof(long long int) = %lu\n", sizeof(long long int));  
  
printf("sizeof(float) = %lu\n", sizeof(float));  
printf("sizeof(double) = %lu\n", sizeof(double));  
printf("sizeof(long double) = %lu\n", sizeof(long double));
```

```
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim/c_test/01$ gcc homework.c -o b.out
hyochangkim@hyochangkim-ThinkPad-X390-Yoga:~/proj/es02/Lv01-02/HyochangKim/c_test/01$ ./b.out
num1 = 1
num2 = 2
num3 = 3
num4 = 2.300000
num4 = 2.30
num5 = 2.800000
num6 = 1000000000
num7 = 1942723.223400
sizeof(char) = 1
sizeof(short) = 2
sizeof(int) = 4
sizeof(long long int) = 8
sizeof(float) = 4
sizeof(double) = 8
sizeof(long double) = 16
```

Homework

□ Data type (1 / 2)

type 자료형 : 이진 정보를 저장하는 메모리 공간

자료 저장을 위해 필요한 메모리 크기가 정해져 있다.

CPU가 한 번에 읽어서 처리할 수 있는 비트의 크기

unsigned : 음수 필요 없다, %u: 부호 없는(unsigned decimal)

short : 메모리 공간 절약 시 사용

long : 큰 값 사용, %lu : long unsigned

short int : 2 Bytes

long int : 4 Bytes , 8 Bytes

float : 유효 숫자 6 , %f

double : 유효 숫자 16 , %lf

(unsigned) long long : 8 Bytes = 64 bits = 18,446,744,073,709,551,616

최소값 : $- 2^{(63)} = - 9,223,372,036,854,775,808$

최대값 : $2^{(63)} = 9,223,372,036,854,775,807$

%lld : signed long long decimal 변수를 출력

%llu : unsigned long long 변수를 출력

long double : 출력은 %Lf 사용, 16 Bytes , double (8 Bytes) 보다 정밀도가 높다

Homework

□ Data type (2 / 2)

sizeof : 대상의 크기 확인, 단항 연산자이며 함수가 아니다.

단항 연산자 : !, ~, -, +, --, ++, *, &, (자료형), sizeof

함수 : 입력 값을 가지고 어떤 일을 수행한 다음에 그 결과물을 나오게 한다.

1's complement : 숫자가 음수일 경우 사용

2's complement : +0, -0 오류를 0으로 정정

□ 메모리 계층 구조

구성 : register, cache, main memory, hard disk

hard disk 용량보다 cache 메모리를 올리는 데 드는 비용이 높다.

CPU의 가격 절반 이상이 cache 값.

자주 사용되는 주소 번지의 데이터는 cache에 저장하여 main memory까지 접속하는 빈도수를 줄인다.

속도 향상 : hard disk → main memory → cache → register (반복적으로 사용하는 데이터 저장 순역순)

main memory 구성 :

code 영역 : 실행할 프로그램의 코드

data 영역 : 전역 변수, 정적 변수

heap 영역 : 사용자의 동적 할당

stack 영역 : 지역 변수, 매개 변수

Question

Q1. 텍스트 편집기로 만든 것을 컴파일과 실행파일을 만들려고 터미널에 "gcc homework.c -o b.out" 로 입력하였습니다.
수업 시간에 다른 방법도 말씀해 주셨는데... 기억이 나지 않습니다....

gcc homework.c (컴파일) + -o b.out 라는 명령을 함께 필수로 작성해야 될까요??

Q2. long double 은 16바이트가 고정인가요??

32bit 64bit 운영체제마다 다르게 나올 수 있나요??

Q3. main () 앞에는 기본형 int 가 아닌 다른 type 의 char, float, double 도 가능할까요??

main () 앞에 의미는 어떤 변수 type을 반환할 것인지 명시하여 주는 기능인가요??

Q4. return 0; 이라는 의미는 0 이면 정상 , 0 이 아니면 오류일까요??

다른 예로 return a; 는 a 값이면 정상, a 값이 아니면 비정상인가요???

Q5. int main () 의 괄호는 매개변수를 정의하려는 용도인가요?

빈 괄호이면 default 로 return 0; 이 될까요??

void main () 의 void는 변수가 없다?? 라고 이해하고 전체적 의미는 " 종료 상태가 없다 (return 반환이 없다)"

라고 해석해도 될까요??