



C언 - HW1

임베디드스쿨1기

Lv1과정

2021. 03. 08

차현호

1. 리눅스 명령어 복습

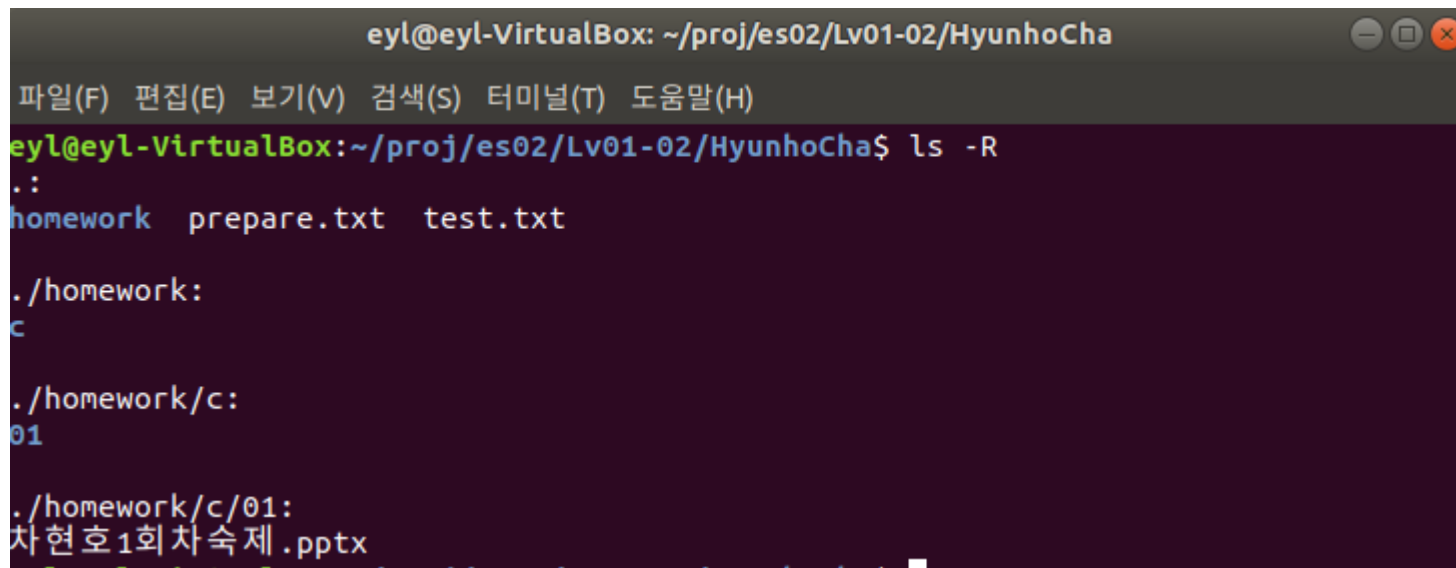
1. ls

현재 디렉토리안에 있는 파일 및 디렉토리를 보여준다

1) 옵션

Command 창에 `ls --help`를 치면 수 많은 옵션들이 나오는데 그 중 수업시간에 배운것들은 다음과 같다.

- `ls -R` : 하위 디렉토리에 있는 파일 및 디렉토리를 전부 보여준다.



```
eyl@eyl-VirtualBox: ~/proj/es02/Lv01-02/HyunhoCha
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha$ ls -R
.:
homework  prepare.txt  test.txt

./homework:
c

./homework/c:
01

./homework/c/01:
차현호1회차숙제.pptx
```

위 사진을 보면 현재디렉토리는 `~/proj/es02/Lv01-02/HyunhoCha` 이다 여기서 `ls -R` 명령어를 사용하면 하위 디렉토리인 `c`뿐 만아니라 `c`디렉토리의 하위 디렉토리인 `01`과 `01` 디렉토리 안의 파일까지 보여주는것을 확인할 수 있다.

1. 리눅스 명령어 복습

- ls -a : 숨긴 파일을 포함한 리스트를 보여준다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls -a
.  ..  .secret.txt  차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

사진에서 ls 명령어만 보이지 않던 .secret.txt 파일이 -a 옵션을 추가해서 보니 보였다. 이로서 파일명 앞에 .이붙으면 숨김파일이라는것을 알 수가 있다.

1. 리눅스 명령어 복습

2. clear

Command 창을 깨끗하게 비워 주므로 커맨드창 정리시에 사용할 수 있다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 14 (delta 2), reused 14 (delta 2), pack-reused 0
오브젝트 묶음 푸는 중: 100% (14/14), 완료.
https://github.com/gusgh0758/Lv01-02 URL에서
eb73d6a..e65e8c6 main -> origin/main
업데이트 중 eb73d6a..e65e8c6
Fast-forward
...5\232\214\354\260\250\354\210\231\354\240\234.pptx" | Bin 0 -> 163891 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "HyunhoCha/homework/c/01/\354\260\250\355\230\204\355\230\27
01\355\232\214\354\260\250\354\210\231\354\240\234.pptx"
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha$ ls
homework prepare.txt test.txt
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha$ cd homework/
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework$ ls
c
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework$ cd c/
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c$ ls
01
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c$ cd 01/
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

clear

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

1. 리눅스 명령어 복습

3. cp

파일 또는 디렉토리를 복사하는 명령어이며 사용법은 cp 복사대상 사본의이름 순이다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test.txt 차현호1회차숙제 .pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ cp test.txt t
est1.txt
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test.txt test1.txt 차현호1회차숙제 .pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

위 사진을 보면 cp 명령어를 사용하여 test.txt파일을 복사하여 test1.txt 파일을 만들어낸것을 확일 할 수 있다.

1) 옵션

- cp -r : 아무런 옵션없이 디렉토리를 복사하려고 하면 복사가 실행 되지 않는다. 디렉토리를 복사하려면 -r 옵션을 사용해야만 한다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test 차현호1회차숙제 .pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ cp -r test/ t
est1
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test test1 차현호1회차숙제 .pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

1. 리눅스 명령어 복습

4. mkdir

Make directory의 약자로서 말그대로 디렉토리를 생성하는 명령어이다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ mkdir test
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test 차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

1) 옵션

- mkdir -p : 한개의 디렉토리 뿐만 아니라 하위 디렉토리까지 한번에 생성하려면 -p 옵션을 사용한다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test 차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ mkdir -p test
/01/02/03/04/05
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test 차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls -R
.:
test 차현호1회차숙제.pptx

./test:
01

./test/01:
02

./test/01/02:
03

./test/01/02/03:
04

./test/01/02/03/04:
05
```

1. 리눅스 명령어 복습

5. cd

Change directory의 약자로 디렉토리를 이동하는 명령어이다.

```
chh@chh-15ZD970-GX7SK: ~/proj
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~$ ls
proj test_proj 다운로드 바탕화면 사진 템플릿
snap 공개 문서 비디오 음악
chh@chh-15ZD970-GX7SK:~$ cd proj/
chh@chh-15ZD970-GX7SK:~/proj$
```

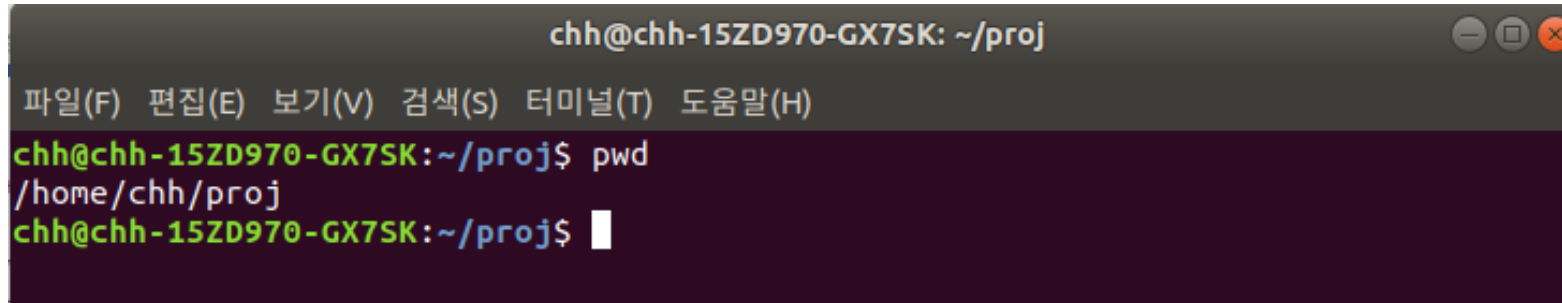
우분투에서 . 의 경우에 현재 위치를 의미하며 ..은 상위 디렉토리를 의미한다. ../../의 경우 2번 상위 디렉토리로 올라간다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ cd ..
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c$ cd ../../
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha$ cd .
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha$
```

1. 리눅스 명령어 복습

6. pwd

Command line 창에서 현재 디렉토리를 나타낸다.



```
chh@chh-15ZD970-GX7SK: ~/proj
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj$ pwd
/home/chh/proj
chh@chh-15ZD970-GX7SK:~/proj$
```


1. 리눅스 명령어 복습

7. sudo

사용자에게 root 권한을 부여하며 sudo 명령어를 사용하려면 사용자 password를 알아야한다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha$ apt-get update
패키지 목록을 읽는 중입니다... 완료
E: 잠금 파일 /var/lib/apt/lists/lock 파일을 열 수 없습니다 - open (13: 허가 거부)
E: /var/lib/apt/lists/ 디렉토리를 잠글 수 없습니다
W: /var/cache/apt/pkgcache.bin 파일을 삭제하는데 문제가 있습니다 - RemoveCaches (13: 허가 거부)
W: /var/cache/apt/srcpkgcache.bin 파일을 삭제하는데 문제가 있습니다 - RemoveCaches (13: 허가 거부)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha$ sudo apt-get update
[sudo] chh의 암호:
받기:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
기존:2 http://kr.archive.ubuntu.com/ubuntu bionic InRelease
받기:3 http://kr.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
받기:4 http://security.ubuntu.com/ubuntu bionic-security/main i386 Packages [926
```

처음에 일반 사용자 권한으로 apt-get update 명령을 사용하였을때는 위의 사진처럼 허가 거부라는 에러 메시지가 나온것을 확인 할 수 있었으나 sudo 명령어를 사용한후에는 잘 실행되는것을 확인 할 수 있다.

1. 리눅스 명령어 복습

8. adduser

사용자 계정을 추가하는 명령어이며 사용법은 `sudo adduser 계정명` 이다.

1. 리눅스 명령어 복습

9. rm

파일 또는 디렉토리를 삭제하는 명령어이다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test.txt 차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ rm test.txt
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

1) 옵션

- rm -rf : 디렉토리를 삭제할때는 -rf 옵션을 사용해야한다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ mkdir test
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test 차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ rm -rf test/
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

1. 리눅스 명령어 복습

9. rm

파일 또는 디렉토리를 삭제하는 명령어이다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test.txt 차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ rm test.txt
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

1) 옵션

- rm -rf : 디렉토리를 삭제할때는 -rf 옵션을 사용해야한다.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ mkdir test
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
test 차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ rm -rf test/
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
차현호1회차숙제.pptx
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

2. 데이터 타입

1. gcc

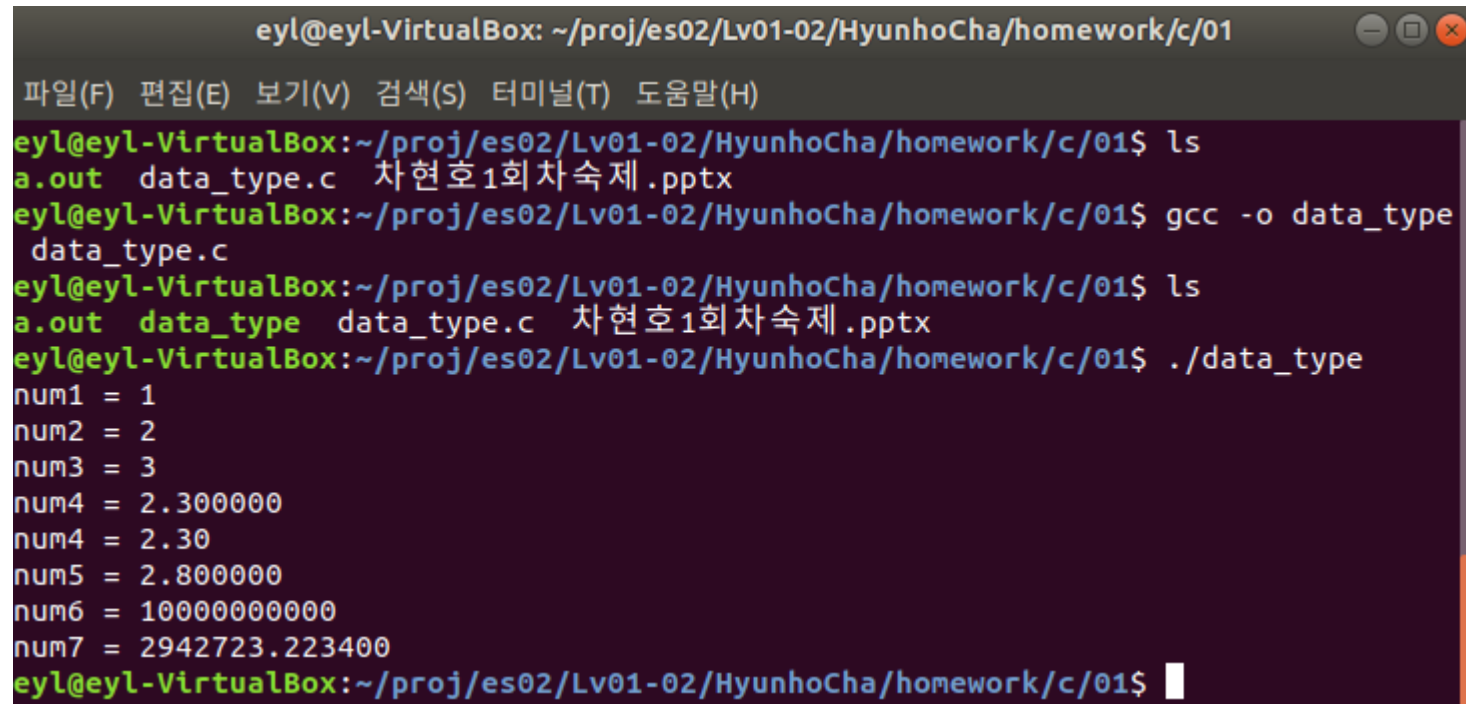
GNU C Compiler 의 약자이며 여러 다른 언어들의 컴파일도 지원하면서 GNU Compiler Collection으로 부르기도 한다.

```
eyl@eyl-VirtualBox: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
data_type.c 차현호1회차숙제.pptx
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ gcc data_type.c
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
a.out data_type.c 차현호1회차숙제.pptx
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ./a.out
num1 = 1
num2 = 2
num3 = 3
num4 = 2.300000
num4 = 2.30
num5 = 2.800000
num6 = 100000000000
num7 = 2942723.223400
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

2. 데이터 타입

1) 옵션

- gcc -o : -o 옵션을 주면 a.out 이 아닌 사용자가 원하는 파일명으로 컴파일 결과물을 만들 수 있다.



```
eyl@eyl-VirtualBox: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
a.out data_type.c 차현호1회차숙제.pptx
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ gcc -o data_type
data_type.c
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ls
a.out data_type data_type.c 차현호1회차숙제.pptx
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ./data_type
num1 = 1
num2 = 2
num3 = 3
num4 = 2.300000
num4 = 2.30
num5 = 2.800000
num6 = 10000000000
num7 = 2942723.223400
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

2. 데이터 타입

1) int

4바이트 크기의 부호 가지는 정수형 데이터 타입이며 숫자의 범위는 $-2^{31} \sim 2^{31} - 1$ 이다.

```
eyl@eyl-VirtualBox: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ./data_type
int min : -2147483648
int max : 2147483647
min - 1 : 2147483647
max + 1 : -2147483648
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

위 사진을 보면 int 형 데이터의 최솟값과 최댓값을 확인 할 수 있다. 여기서 최솟값 - 1 또는 최댓값 + 1을 하면 어떻게 될까?

결과는 위에 나와있는것처럼 **최솟값 - 1 == 최댓값**, **최댓값 + 1 == 최솟값**이 되는것을 확인할 수 있다.

```
eyl@eyl-VirtualBox: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ./data_type
int min : 10000000 00000000 00000000 00000000
int max : 01111111 11111111 11111111 11111111
int min-1 : 01111111 11111111 11111111 11111111
int max+1 : 10000000 00000000 00000000 00000000
eyl@eyl-VirtualBox:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

이를 확인하기 위해 2진수로 출력해보았다. 2진수로 출력시 max + 1과 min - 1 의 값이 이진수 계산으로 잘된것을 확인 할 수 있었다.

2. 데이터 타입

2) Char

1바이트 크기의 부호 가지는 정수형 데이터타입이며 정수 범위는 $-2^7 \sim 2^7 - 1$ 이다.

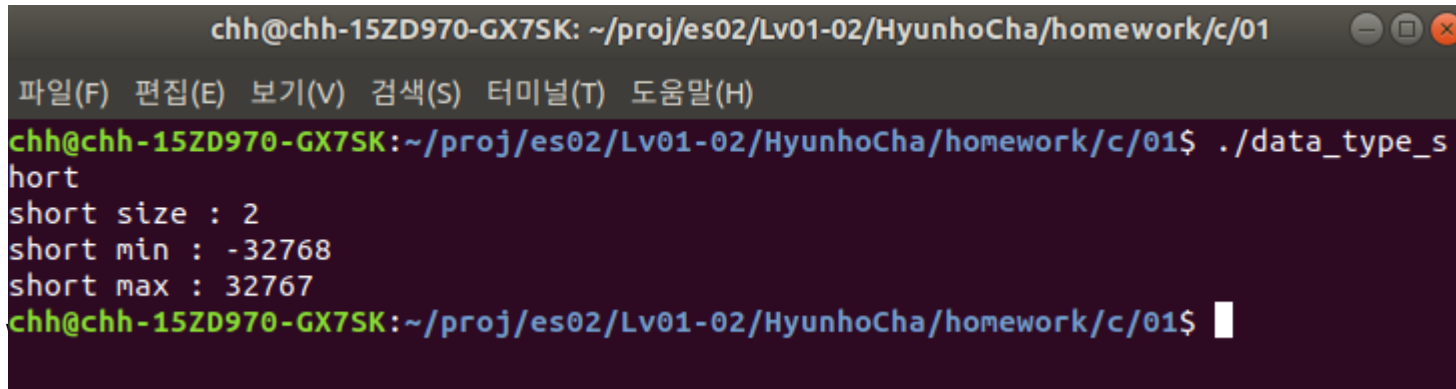
```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ./data_type_c
har
char min : -128
char max : 127
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

Char 형 데이터타입의 최대값과 최소값을 확인해보면 -2^7 , $2^7 - 1$ 인것을 확인해 볼 수 있다.

2. 데이터 타입

3) short

2바이트 크기의 부호 가지는 정수형 데이터타입이며 정수 범위는 $-2^{15} \sim 2^{15} - 1$ 이다.

A terminal window with a dark background. The title bar shows the user 'chh' on a machine 'chh-15ZD970-GX7SK' in the directory '~/proj/es02/Lv01-02/HyunhoCha/homework/c/01'. The menu bar includes '파일(F)', '편집(E)', '보기(V)', '검색(S)', '터미널(T)', and '도움말(H)'. The command prompt shows the user running './data_type_short'. The output displays the size and range of a short integer: 'short size : 2', 'short min : -32768', and 'short max : 32767'. The prompt returns to the shell.

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$ ./data_type_s
hort
short size : 2
short min : -32768
short max : 32767
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

Char 형 데이터타입의 사이즈와 최대값과 최솟값을 확인해보면 2, -2^{15} , $2^{15} - 1$ 인것을 확인해 볼 수 있다.

2. 데이터 타입

4) float

4바이트 데이터 크기의 실수형 데이터 타입이며 IEEE 754 표준을 따르고있다.

```
float size : 4
number1 : 0.100000
number2 : -0.100000
chh@chh-15ZD970-GX7SK:~/proj/es02/Lv01-02/HyunhoCha/homework/c/01$
```

float형 데이터타입의 사이즈는 4바이트 0.1을 넣었을때와 -0.1을 넣었을때의 값을 확인 할 수 있다.
Number1, number2의 메모리에 저장된 값이 궁금하여 gdb를 이용해 메모리 값을 출력해 보았다.

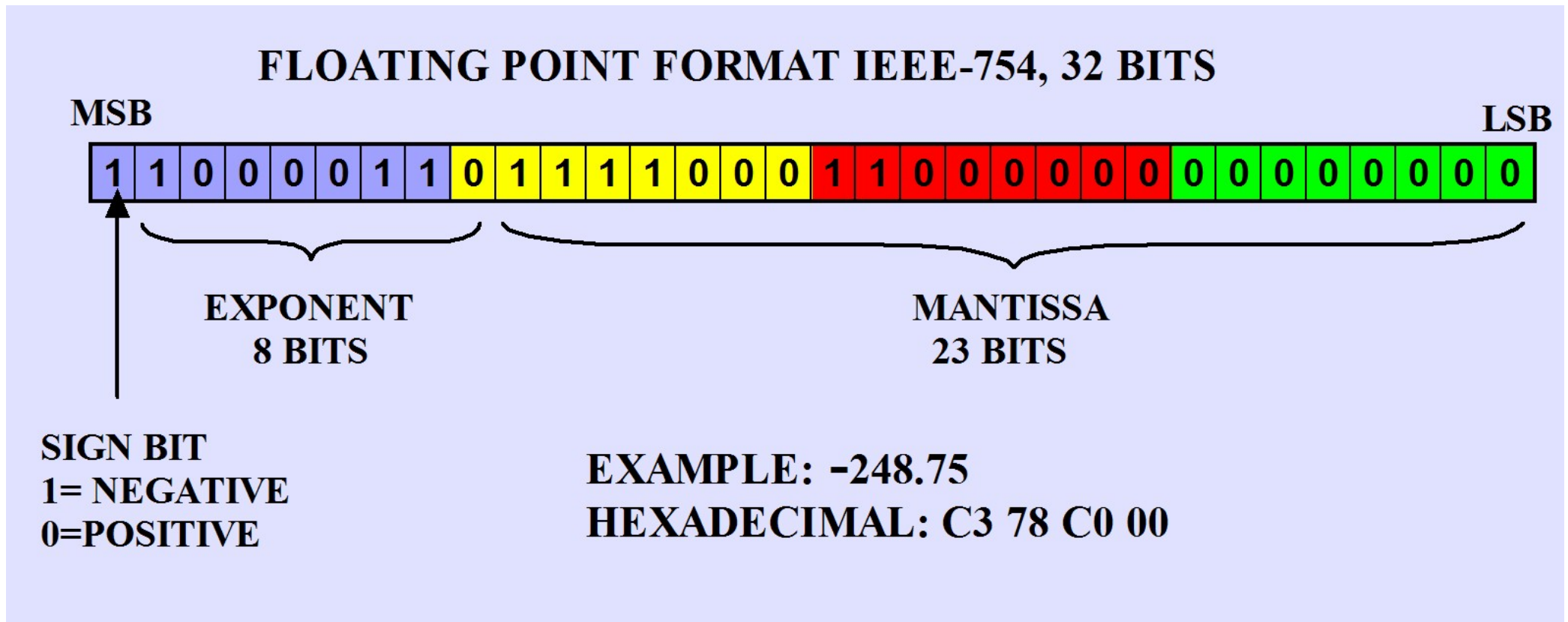
```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
(gdb) p &number1
$12 = (float *) 0x7fffffffdf48
(gdb) p &number2
$13 = (float *) 0x7fffffffdf4c
(gdb) p number1
$14 = 0.100000001
(gdb) p number2
$15 = -0.100000001
(gdb) x /wt 0x7fffffffdf48
0x7fffffffdf48: 00111101110011001100110011001101
(gdb) x /wt 0x7fffffffdf4c
0x7fffffffdf4c: 10111101110011001100110011001101
(gdb)
```

gdb로 메모리에 저장된 값을 확인해보면 맨 앞의 상위비트 1bit가 부호를 나타내는 값인것을 확인할 수 있다. 왜이렇게 나왔는지 IEEE 754 표준을 보며 확인해보자

2. 데이터 타입

4) float

IEEE 754



컴퓨터가 소수점을 표현하는 방식은 부동 소수점 방식이며 IEEE 754 표준을 따르고 있다. 4바이트기준으로 IEEE 754 표준으로 표현하면 최상위 1비트는 부호를 그다음 상위 8비트는 지수비트를 그다음 23비트는 가수를 나타낸다.

2. 데이터 타입

4) float

```
chh@chh-15ZD970-GX7SK: ~/proj/es02/Lv01-02/HyunhoCha/homework/c/01
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
(gdb) p &number1
$12 = (float *) 0x7fffffffdf48
(gdb) p &number2
$13 = (float *) 0x7fffffffdf4c
(gdb) p number1
$14 = 0.100000001
(gdb) p number2
$15 = -0.100000001
(gdb) x /wt 0x7fffffffdf48
0x7fffffffdf48: 00111101110011001100110011001101
(gdb) x /wt 0x7fffffffdf4c
0x7fffffffdf4c: 10111101110011001100110011001101
(gdb) █
```

IEEE 754 표준으로 number1 0.100000001 값이 저장되어있는 메모리 데이터를 보면 다음과 같이 분리할 수 있다.

부호비트 : 0

지수비트 : 01111011

가수비트 : 10011001100110011001101

부호비트를 우선보면 0이므로 양수인것을 확인 할 수 있다.

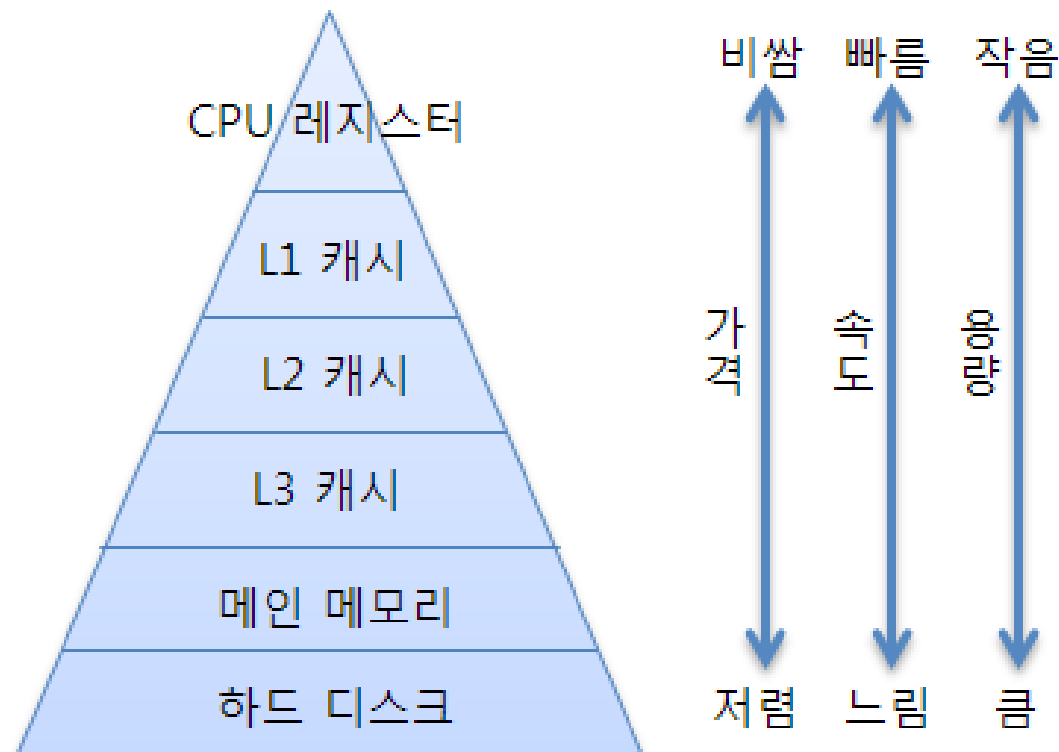
0.1을 이진수로 표현하면 0.00010011001100110011001101.... 이되며 이를 이진수 과학표기로 바꾸면 $0.10011001100110011001101 \times 2^{-3}$ 이된다.

여기서 가수부 값은 $127 + (-3) = 124 = 0b01111011$ 이된다.

그리고 과학표기법의 앞부분이 가수비트로 들어가는것을 확인할 수 있다.

3. 메모리 계층 구조

CPU가 메모리에서 데이터를 가져오는 속도는 대략 레지스터 > 캐시 > 메모리 > 하드디스크 순으로 빠르며 비용은 이와 반대로 레지스터순으로 가장 비싸다 이에 비용절감과 속도 저하를 막기위해 각 메모리를 계층별로 놓았다.



출처 : <https://constructor.tistory.com/18>