

Embedded Class

Linux 기초 와 C언어

-목차

1. Linux

- Linux 란
- 기초 명령어

2. C 언어

- 기초 프로그래밍 “Welcome to Embedded Class”
- int , float , double , long double
- function 이용 프로그래밍

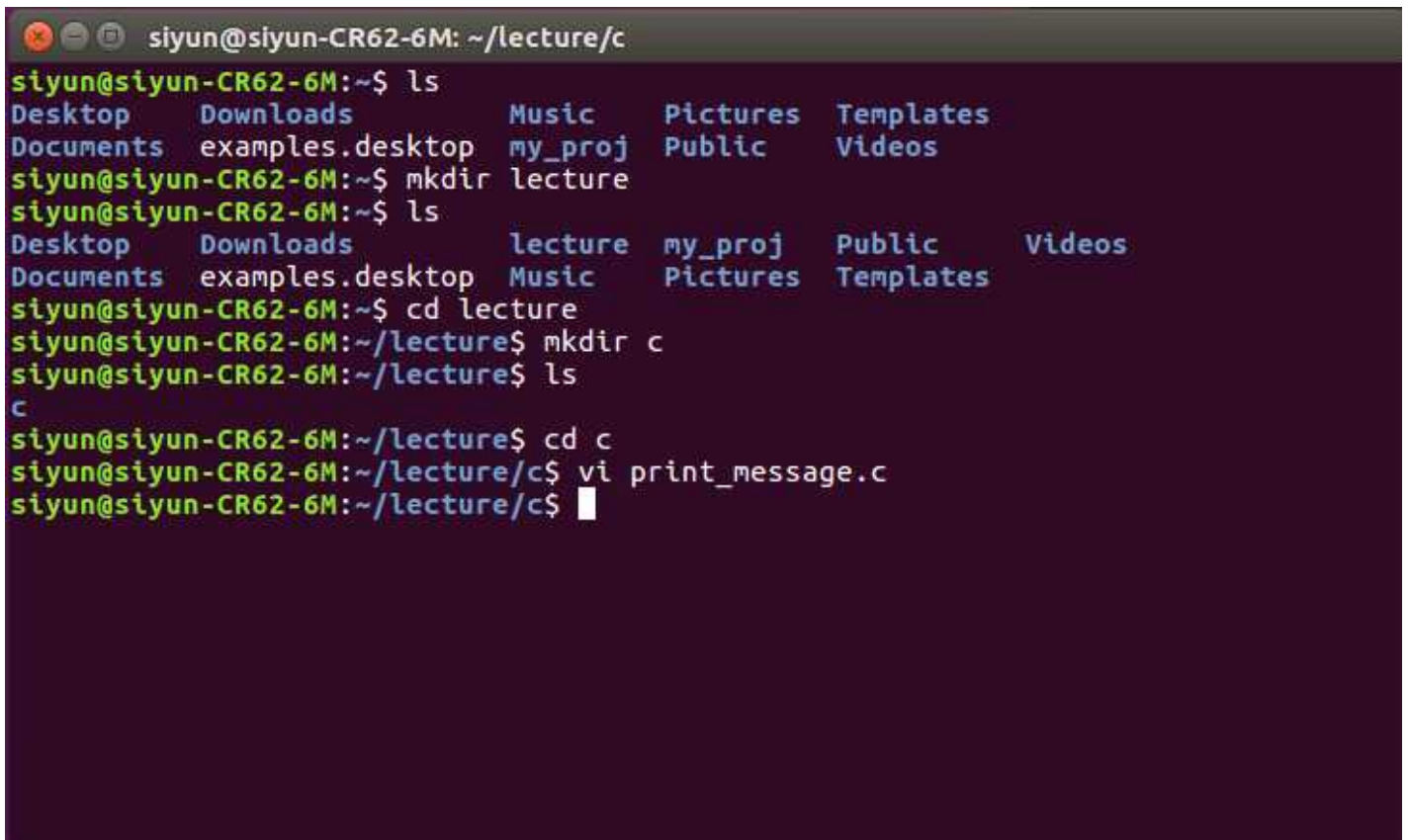
김 시 윤

1. Linux 란?

1991년 11월에 리누스 토르발즈(Linus Torvalds)가 버전 0.02을 공개한 유닉스 기반 개인컴퓨터용 공개 운영체제이다. 리눅스는 각종 주변기기에 따라 혹은 사용하는 시스템의 특성에 맞게 소스를 변경할 수 있다는 장점이 있다.

2. 기초명령어

여러 가지 명령어들을 ubuntu 터미널로 명령 실습을 해보았다.

A terminal window titled 'siyun@siyun-CR62-6M: ~/lecture/c' showing a series of Linux commands and their outputs. The commands are: 'ls' (listing files in the home directory), 'mkdir lecture' (creating a new directory), 'ls' (listing files again to show 'lecture'), 'cd lecture' (changing to the new directory), 'mkdir c' (creating a subdirectory), 'ls' (listing files in the current directory to show 'c'), 'cd c' (changing to the subdirectory), and 'vi print_message.c' (opening a file in the vi editor). The terminal output shows the directory structure: Desktop, Downloads, Music, Pictures, Templates, Documents, examples.desktop, my_proj, Public, Videos, and the newly created lecture and c directories.

```
siyun@siyun-CR62-6M: ~/lecture/c
siyun@siyun-CR62-6M:~$ ls
Desktop    Downloads  Music      Pictures   Templates
Documents  examples.desktop my_proj    Public     Videos
siyun@siyun-CR62-6M:~$ mkdir lecture
siyun@siyun-CR62-6M:~$ ls
Desktop    Downloads  lecture    my_proj    Public     Videos
Documents  examples.desktop Music      Pictures   Templates
siyun@siyun-CR62-6M:~$ cd lecture
siyun@siyun-CR62-6M:~/lecture$ mkdir c
siyun@siyun-CR62-6M:~/lecture$ ls
c
siyun@siyun-CR62-6M:~/lecture$ cd c
siyun@siyun-CR62-6M:~/lecture/c$ vi print_message.c
siyun@siyun-CR62-6M:~/lecture/c$
```

[그림1] Linux 기본 명령어

[그림1]에서는 기본명령어와 폴더생성을 해보았다.

ls(list) 는 현재 위치에 있는 파일 및 폴더 즉 direct를 나타내준다.

mkdir(make direct) direct를 생성해주는 명령어다.

[그림1]에서는 lecture 와 c 라는 폴더를 생성하였다.

vi print_message.c

print_message.c 라는 c파일을 생성해 열어준다.

```
siyun@siyun-CR62-6M: ~/lecture/c
hi hello nihao
say
tt
uaa
hi
bye
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

[그림2]Linux 기본 명령어

[그림1]에서 생성한 print_message.c 에 hi hello nihao 등 여러 단어를 입력하였다.

```
siyun@siyun-CR62-6M: ~/lecture/c
hi hello nihao
hi hello nihao
hi hello nihao
hi hello nihao
hi hello nihao
hi hello nihao
nihao
esjoiwwef
nihao
esjoiwwef
nihao
esjoiwwef
nihao
esjoiwwef
nihao
esjoiwwef
nihao
esjoiwwef
~
~
~
```

그 다음은 명령모드에서 yy 명령어로 복사한후 p로 붙여넣기한 모습이다.

```
siyun@siyun-CR62-6M: ~/lecture/c
121 hi hello nihao
122 hi hello nihao
123 hi hello nihao
124 hi hello nihao
125 hi hello nihao
126 hi hello nihao
127 nihao
128 esjoiwwef
129 nihao
130 esjoiwwef
131 nihao
132 esjoiwwef
133 nihao
134 esjoiwwef
135 nihao
136 esjoiwwef
137 nihao
138 esjoiwwef
139 nihao
140 esjoiwwef
~
~
~
:set nu                                     126,1      Bot
```

명령모드에서 :set nu 로 줄의 숫자를 나타내주었다.

siyun@siyun-CR62-6M: ~/lecture/c	siyun@siyun-CR62-6M: ~/lecture/c
121 hi hello nihao	121 hi hello hello
122 hi hello nihao	122 hi hello hello
123 hi hello nihao	123 hi hello hello
124 hi hello nihao	124 hi hello hello
125 hi hello nihao	125 hi hello hello
126 hi hello nihao	126 hi hello hello
127 nihao	127 hello
128 esjoiwwef	128 esjoiwwef
129 nihao	129 hello
130 esjoiwwef	130 esjoiwwef
131 nihao	131 hello
132 esjoiwwef	132 esjoiwwef
133 nihao	133 hello
134 esjoiwwef	134 esjoiwwef
135 nihao	135 hello
136 esjoiwwef	136 esjoiwwef
137 nihao	137 hello
138 esjoiwwef	138 esjoiwwef
139 nihao	139 hello
140 esjoiwwef	140 esjoiwwef
~	~
~	~
~	~
:%s/nihao/hello/g	88 substitutions on 88 lines

nihao를 hello로 바꾸어보았다.


```
siyun@siyun-CR62-6M: ~/lecture/c/class
Desktop Downloads lecture my_proj Public Videos
Documents examples.desktop Music Pictures Templates
siyun@siyun-CR62-6M:~$ cd lecture
siyun@siyun-CR62-6M:~/lecture$ mkdir c
siyun@siyun-CR62-6M:~/lecture$ ls
c
siyun@siyun-CR62-6M:~/lecture$ cd c
siyun@siyun-CR62-6M:~/lecture/c$ vi print_message.c
siyun@siyun-CR62-6M:~/lecture/c$ vi print_message.c
siyun@siyun-CR62-6M:~/lecture/c$ ls
print_message.c
siyun@siyun-CR62-6M:~/lecture/c$ vi homework.c
siyun@siyun-CR62-6M:~/lecture/c$ gcc homework.c
siyun@siyun-CR62-6M:~/lecture/c$ ls
a.out homework.c print_message.c
siyun@siyun-CR62-6M:~/lecture/c$ ./a.out
Wlecome to Embedded class
My name is Siyun Kim
siyun@siyun-CR62-6M:~/lecture/c$ mkdir class
siyun@siyun-CR62-6M:~/lecture/c$ cd class
siyun@siyun-CR62-6M:~/lecture/c/class$ ../a.out
Wlecome to Embedded class
My name is Siyun Kim
siyun@siyun-CR62-6M:~/lecture/c/class$
```

결과를 확인하기 위해 gcc homework.c 로 컴파일을 한 후
출력을 확인하였다.

./ =현재위치 인데 상위개념을 확실히 인지하기 위해
class 라는 폴더를 생성한후 컴파일한 파일의 출력을
상위 폴더의 a.out을 확인한다는 명령어로 같은결과가 나오는지 확인했다.
결과는 같았다. ../는 상위를 가리키는 말임을 확인하였다.

```
siyun@siyun-CR62-6M: ~/lecture/c
#include <stdio.h>

int myfunc(int num)
{
    return num << 1;
}

int main(void)
{
    int num=3,res;
    res = myfunc(num);
    printf("res = %d\n", res);

    return 0;
}
```

shift logic 에대한 함수를 좀더 작성하고싶어 작성하였다.
함수안에 있는 식은 return num << 1; 이다
여기서 num 은 3 일 때 2진수로 바꾸면 0011이 된다.
여기서 num을 왼쪽으로 한칸 shift 시키면 0110 이되고 10진수로 바꾸면 6이된다.
결론적으로 3이 들어가서 6이 나오는 함수가 완성이 된다.


```
siyun@siyun-CR62-6M: ~/lecture/c
siyun@siyun-CR62-6M:~$ ls
Desktop  Downloads  lecture  my_proj  Public  Videos
Documents examples.desktop Music  Pictures  Templates
siyun@siyun-CR62-6M:~$ cd lecture
siyun@siyun-CR62-6M:~/lecture$ ls
c
siyun@siyun-CR62-6M:~/lecture$ cd c
siyun@siyun-CR62-6M:~/lecture/c$ ls
a.out class home homework.c print_message.c
siyun@siyun-CR62-6M:~/lecture/c$ vi func1.c
siyun@siyun-CR62-6M:~/lecture/c$ gcc func1.c
siyun@siyun-CR62-6M:~/lecture/c$ ./a.out
res = 6
siyun@siyun-CR62-6M:~/lecture/c$
```

그러면 4가 들어가서 16이 나오는 함수는 어어떻게 작성해야할까?

다시 4를 2진수로 바꾸면 100이 된다. 여기서 16을 2진수로 바꿔보면 10000이 된다.

4의 최상위 비트 1이 왼쪽으로 2번 shift 해야 16의 2진수 형태와 동일해지는 것을 알 수 있다. 이를 토대로 4가 들어가서 16이 나오는 shift 함수를 짜보면

```
</> source code close shortcuts fullscreen ↗
1 #include <stdio.h>
2
3 int myfunc(int num)
4 {
5     return num << 2;
6 }
7
8
9 int main(void) {
10     int num=4,res;
11     res = myfunc(num);
12     printf("res = %d\n",res);
13     return 0;
14 }
15
```

input Output clear the output ☒ syntax highlight

Success #stdin #stdout 0s 4244KB

res = 16

이와 같이 짤 수 있으며 결과를 확인하면 16이 나오게 된다.

