



EDDI

Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv1 과정

제 4기

2022. 08. 26

유진선

LINUX(리눅스) 터미널 기본 명령어

ls (list)

디렉토리 목록 확인

- 1) ls
- 2) ls 디렉토리 명
- 3) ls -R (현재 위치부터 하위의 **모든 정보**를 보여줌)
- 4) ls -a

리눅스에서 숨김 파일은 맨 앞에 '.'이 온다.

숨김파일을 포함해서 리스트를 보여 준다.

```
jlinseon@jlinseon-Inspiron-16-5620: ~/EmbeddedMasterLv1/47 |
jlinseon@jlinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47|$ ls
ChoSeungSu HanJeongHun JinDongMin KimYoungJoong KimYoungJu LeeSangHoon YuJinSeon prepare.txt
jlinseon@jlinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47|$ ls YuJinSeon/
TEST1st.txt prepare.txt
jlinseon@jlinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47|$ ls -R
.:
ChoSeungSu HanJeongHun JinDongMin KimYoungJoong KimYoungJu LeeSangHoon YuJinSeon prepare.txt
./ChoSeungSu:
prepare.txt
./HanJeongHun:
prepare.txt
./JinDongMin:
prepare.txt
./KimYoungJoong:
prepare.txt
./KimYoungJu:
prepare.txt
./LeeSangHoon:
c prepare.txt
./LeeSangHoon/c:
1
./LeeSangHoon/c/1:
variable.c
./YuJinSeon:
TEST1st.txt prepare.txt
jlinseon@jlinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47|$ ls -a
. . . ChoSeungSu HanJeongHun JinDongMin KimYoungJoong KimYoungJu LeeSangHoon YuJinSeon prepare.txt
jlinseon@jlinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47|$
```

LINUX(리눅스) 터미널 기본 명령어

pwd (print working directory)

현재 작업중인 디렉토리 정보 출력

```
jinseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/4기$ pwd  
/home/jinseon/EmbeddedMasterLv1/4기  
jinseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/4기$ █
```

LINUX(리눅스) 터미널 기본 명령어

cd (change directory)

경로 이동

절대 경로와 상대 경로로 이동 가능하다.

절대경로란 최상위 디렉토리 (/)부터 시작해서 목표 디렉토리까지 가는 경로를 전부 기술하는 방식이다.

절대경로로 경로를 기술할 때에는 항상 맨 앞에 최상위 디렉토리 (/)가 붙는다.

상대경로는 '현재 자신이 있는 위치를 기준으로' 이동을 하는 것이다.

그리고 현재 자신이 있는 위치는 .(마침표) 로 표기한다.

'.' : 현재 디렉토리

'..' : 상위 디렉토리

ex) 현재 내 위치가 '/emlv1/lsh/test'라면 '/emlv1/lsh/c'로 이동할 때 아래와 같은 명령을 작성할 수 있음

= cd ../c → 이동완료

```
jinson@jinson-Inspiron-16-5620:~$ ls
EmbeddedMasterLv1 snap 공개 다운로드 문서 바탕화면 비디오 사진 음악 템플릿
jinson@jinson-Inspiron-16-5620:~$ cd EmbeddedMasterLv1/
jinson@jinson-Inspiron-16-5620:~/EmbeddedMasterLv1$ cd 47|/YuJinSeon/
jinson@jinson-Inspiron-16-5620:~/EmbeddedMasterLv1/47|/YuJinSeon$ ls
TEST1st.txt prepare.txt
jinson@jinson-Inspiron-16-5620:~/EmbeddedMasterLv1/47|/YuJinSeon$ cd ./
jinson@jinson-Inspiron-16-5620:~/EmbeddedMasterLv1/47|/YuJinSeon$ cd ../
jinson@jinson-Inspiron-16-5620:~/EmbeddedMasterLv1/47|$ cd ../47|
jinson@jinson-Inspiron-16-5620:~/EmbeddedMasterLv1/47|$ cd ../37|
jinson@jinson-Inspiron-16-5620:~/EmbeddedMasterLv1/37|$
```

LINUX(리눅스) 터미널 기본 명령어

mkdir (make directory)

디렉토리 생성

1) mkdir 디렉토리명

2) mkdir -p 디렉토리명1/디렉토리명2/디렉토리명3/

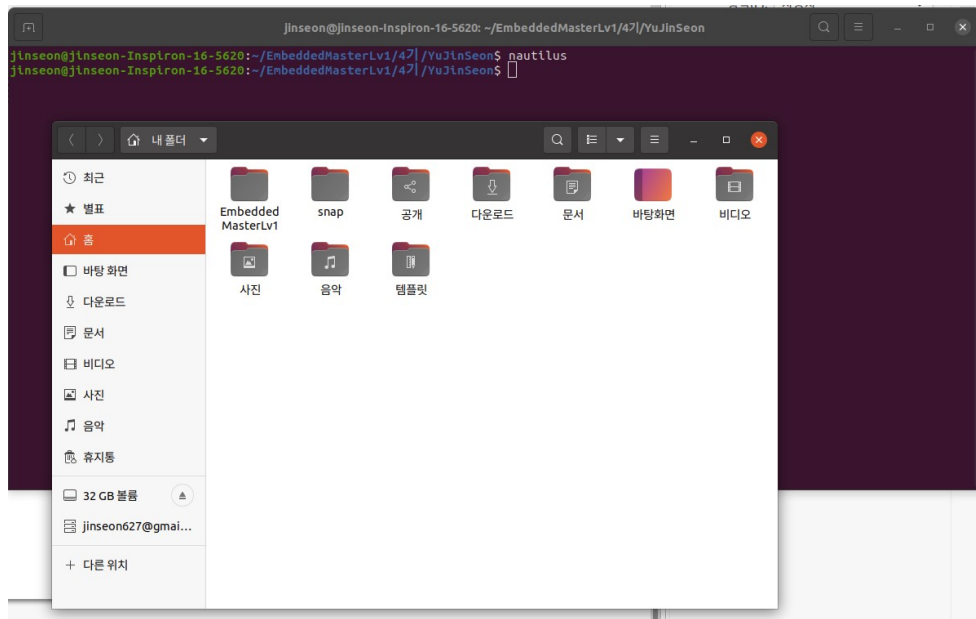
이와 같이 만들면 디렉토리1 내부에 디렉토리2가 만들어지고 디렉토리2 내부에 디렉토리3 생성.

```
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/37$ cd ../
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1$ ls
3기 4기 README.md documents homework '임베디드 속제 2주차.pdf'
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1$ cd 47
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47$ ls
ChoSeungSu HanJeongHun JinDongMin KimYoungJoong KimYoungJu LeeSangHoon YuJinSeon prepare.txt
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47$ cd YuJinSeon/
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon$ ls
TEST1st.txt prepare.txt
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon$ mkdir mktest1/
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon$ ls
TEST1st.txt mktest1 prepare.txt
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon$ cd mktest1
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1$ mkdir -p mktest2/ mktest3/mktest4
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1$ ls
mktest2 mktest3
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1$ ls
mktest2 mktest3
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1$ cd mktest2
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1/mktest2$ ls
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1/mktest2$ ls
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1/mktest2$ cd ../
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1$ ls
mktest2 mktest3
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1$ cd mktest3
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1/mktest3$ ls
mktest4
jInseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/47/YuJinSeon/mktest1/mktest3$
```


LINUX(리눅스) 터미널 기본 명령어

Nautilus 노틸러스

GUI 상태로 현재 디렉토리를 볼수 있음
우분투로 terminal 을 사용시 현재 사용자가 있는 위치로 파일 관리자를 연다.



LINUX(리눅스) 터미널 기본 명령어

vi 파일제목.c → 파일제목.확장자명 이런식으로 파일 생성 하면서 해당 파일 편집창으로 이동됨

Vi 편집기 : → 커맨드(ESC키) , 편집 (l or a || shift + l : 맨앞, shift + a : 맨뒤)

커맨드 모드에서 ‘:wq’ → 저장하고 나가기

커맨드 모드에서 ‘:q!’ → 강제종료

커맨드 모드에서 y숫자y → 숫자만큼 라인 복사

커맨드 모드에서 d숫자d → 숫자만큼 라인 삭제

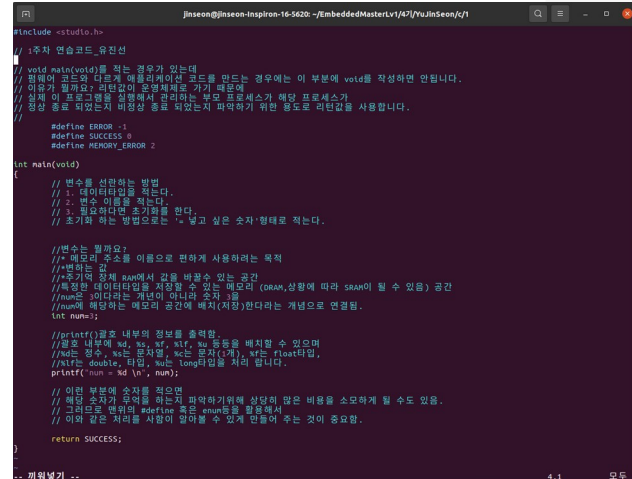
커맨드 모드에서 u → 뒤로 가기

커맨드 모드에서 ctrl + R → 앞으로 가기

커맨드 모드에서 p → 붙여넣기

커맨드 모드에서 ‘:q!’ → 강제 종료

```
jjinseon@jjinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/4기/YuJinSeon/c/1$ ls
variable.c
jjinseon@jjinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/4기/YuJinSeon/c/1$ vi variable.c
jjinseon@jjinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/4기/YuJinSeon/c/1$
```



```
jinseon@jinseon-Inspiron-16-5620:~/EmbeddedMasterLv1/4기/YuJinSeon/c/1
#include <stdio.h>
// 주석: 연습코드, 유원선
// void main(void)를 쓰는 경우가 있는데
// 컴파일러와 링크와 다른게 때문에 리눅스에서는 이 부분에 void를 작성하면 안됩니다.
// 이유가 될까요? 리눅스가 운영체제로 가기 때문에
// 실제 이 프로그램은 컴파일해서 실행하는 것은 프로세스가 해당 프로세스가
// 정상 종료 되었는지 비정상 종료 되었는지 파악하기 위한 용도로 리턴값을 사용합니다.
//
#define ERROR -1
#define SUCCESS 0
#define MEMORY_ERROR 2

int main(void)
{
    // 변수를 선언하는 방법
    // 1. 데이터타입을 적는다.
    // 2. 변수 이름을 적는다.
    // 3. 필요하면 초기화를 한다.
    // 초기화 하는 방법으로는 * 담고 싶은 숫자*형태로 적는다.

    //변수는 필요요!
    // 메모리 주소를 이용해서 변수를 사용하려는 목적
    //변수는 int
    //중간적 단계 안에서 값을 바꿀수 있는 공간
    //특정할 데이터타입을 저장할 수 있는 메모리 (main,상항에 따라 sizeof 할 수 있음) 공간
    //num은 3이라는 개념이 아니라 숫자 3을
    //num에 해당하는 메모리 공간에 배치(저장)한다는 개념으로 연결됨.
    int num;

    //printf()를로 내부의 정보를 출력함.
    //필요 부분에 %d, %s, %f, %lf, %u 등등을 배치할 수 있으며
    //%d는 정수, %s는 문자열, %c는 문자(1개), %f는 float타입,
    //%lf는 double, %i, %u는 long타입을 처리 합니다.
    printf("num = %d\n", num);

    // 이런 부분에 숫자를 적으면
    // 해당 숫자가 주어진다는지 파악하기위해 상항의 많은 비용을 소모하게 될 수도 있음.
    // 그러므로 확실히 %d, %f, %lf, %u 등등을 활용해서
    // 이와 같은 처리를 사람이 알아볼 수 있게 만들어 주는 것이 중요함.

    return SUCCESS;
}
```


LINUX(리눅스) 터미널 기본 명령어

컴파일 방법

gcc → 리눅스 전용 컴파일러 모음중 **C 컴파일러**에 해당함

1. gcc *.c → a.out

실행시 ./a.out (여기서 a.out은 실행 파일명)

2. gcc -o variable
variable.c

실행시 ./variable