

터미널키기 : Ctrl + Alt + T

현재 디렉토리가 어디있는지 : pwd

현재 디렉토리에 무엇이 있는지 : ls

C 프로그램 컴파일하기 : gcc test.c

디렉토리 만들기 : mkdir (디렉토리 이름)

프로그램 디버깅하기 : gdb

디렉토리 이동하기 : cd

*cd 명령어에는 '절대경로' 방식과 '상대경로' 방식이 존재

절대경로 방식이란 '/' 최상위 root 에서 가고싶은 위치까지를 지정하는 방식

'절대경로' 방식

cd /home/id/lecture/result

'상대경로' 방식

cd ../result

* 항공기나 우주선등에 고성능 아키텍처를 사용하지 못하는 이유

결론: 우주 방사선 때문임

* 우주 방사선이 위험한 이유

결론: 맥스웰 방정식

cd = change directory

../ result

rm -rf / * 조심 - C 드라이브 삭제

입력하고 여러분이 만들고싶은 코드

vi print_message.c

명령모드 (아무메시지가 없을때)

편집모드 a, i

A, I

편집모드 들어가면 기록이 됨

Shift + U : 되돌리기

Shift + R : 앞으로가기

d3d : 3 줄 지우기

: % s / 바꿀거, 바뀐후

/g : 치환

yy : 줄복사

p : 붙여넣기

/ 찾고자 하는것

n (아래부터 검색), N(위로부터 검색)

:\$: 맨끝이동

:숫자 : 특정라인 이동

ctrl + F : 페이지 다운

ctrl + B : 페이지 업

4x : 4 개지우는거

wq : 나가기

편집모드 안들어가고 그냥 빨리보고싶다 -> cat

변수의 정의

메모리에 정보를 저장할 수 있는 공간

메모리의 주소가 "포인터"

-메모리-

stack : 지역변수가 위치하는 영역

Heap : 동적할당된 것들이 위치하는 영역

Data : 전역변수 및 static 으로 선언된 것들이 위치하는 영역, 초기화되지 않은
모든것은 0 으로 저장됨

Text : Machine Code 가 위치하는 영역

```
includ<stdio.h>
int main(void)
/*int, short, char float, double, long double*/
{
int num1=3;
int num2=7;
float num3=7.7;
double num4=3.3;
printf("num1=%d, num2=%d,num3=%f, num4=%f Wn");
res1=num1*num3;
res2=num1*num2;
return 0;
}
```

gcc 실행하면 실행파일 a.out 가 생긴다.

./a.out 실행

a.out 이름 수정

gcc -o print_msg print_message.c <-수정할 소스언어

./print_msg 실행파일

gcc -g -> 디버깅

gcc -g -o print.msg print_message.c

gdb test

breakpoint 2, main() at func1, c;

{

(gdb)disas