IOT 환경의 임베디드 개발자 양성과정

첫번째 과제

1일차 내용 정리

1. 리눅스 설치

- https://www.ubuntu.com/download/desktop

  위 주소에서 Ubuntu Desktop 16.04.3 다운로드 받는다.

- https://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3/

  위 주소에서 Universal USB installer를 다운로드 받는다.

- Universal USB installer를 실행하여 Ubuntu 설치용 USB를 만든다.

  (http://cafe.naver.com/hestit/1986 : 내용참조)

- 부팅 중 BIOS에 진입하여 설치한다.

  (http://mangyeon.tistory.com/5 : 제조사 별 Bios진입 키)

2. 리눅스 기본 환경 구성

- sudo apt-get update : 소프트웨어 설치 가능 리스트
- sudo apt-get install vim : vi의 방향키 문제 해결
- sudo apt-get install git : github관리
- sudo apt-get install build-essential : 필수 개발툴 설치

3. 리눅스 명령어 정리

- Ctrl + Alt + T : 터미널 열기
- pwd : 현재 디렉토리 위치
- ls : 현재 디렉토리 내용
- mkdir : 디렉토리 생성
- cd : 디렉토리 이동
-

* cd 명령어에는 '절대경로' 방식과 '상대경로' 방식이 존재

절대 경로 방식이란 '/' 최상위 root 에서 가고 싶은 위치 까지를 지정하는 방식이다 ex)현재 위치는 '/home/id/lecture' 이다.

여기에 test 라는 디렉토리가 있고 result 라는 디렉토리가 있다.

'절대경로' 방식으로 test 에 가고 싶다고 가정

cd /home/id/lecture/test

'상대경로' 방식은 현재 위치를 기준으로 한다.

ex) 현재 위치는 '/home/id/lecture' 이다.

여기에 test 라는 디렉토리가 있고 result 라는 디렉토리가 있다.

'상대경로' 방식으로 test 에 가고 싶다고 가정

cd test

- '..' : 의 상위 디렉토리
- '.' : 현재 디렉토리

ex) 현재 위치는 '/home/id/lecture/test' 이다.

lecture 디렉토리 밑에 result 에 가고 싶다 가정

'절대경로' 방식 : cd /home/id/lecture/result

'상대경로' 방식 : cd ../result

- 명령모드(esc) / 편집모드(a,i)

u : 되돌리기

r : 앞으로 가기

yy : 한줄복사

p : 붙여넣기

shift + a : 맨 뒤로

shift + i : 맨 앞으로

a : 현재 커서 뒤

i :현재 커서 위치

y숫자y : 숫자만큼 복사

:%s /바꿀꺼 /변경 후 /g : 치환

:set nu

/찾고자 하는 것

n -> 아래로찾기

N -> 위로찾기

:set hlsearch (검색패턴 강조됨)

:$ 맨 끝으로 이동하기

:2000 2000번째 줄 보여줌

ctrl + f : 페이지 단위로 아래로

ctrl + b : 페이지 단위로 위로

x : 하나씩 지움

4x : 4개지움

cat 이름 : 내용확인

:wq : 파일 저장 후 나감

ctrl g : 현재 파일 이름 알 수 있음

vi [파일명] : 파일명을 편집

gcc [소스파일명] : 소스파일명을 컴파일해서 a.out이라는 실행파일을 생성함

(소스 파일이란 반드시 *.c 확장자를 가지고 있어야한다)

gcc -o [실행파일명] [소스파일명] : 소스파일명을 컴파일해서 실행파일명의 실행파일을 생성함(실행파일명은 소스파일명과 일치시키지 않아도 무방하지만 용도를 명확히 해줘야 이게 뭔지 파악을 하기에 좋다)

gcc -g -o [실행파일명] [소스파일명] : 위와 동일하게 컴파일을 하는데 디버깅 옵션을 추가적으로 주는 명령이다.

gcc -g [소스파일명] : 이와 같이 하여도 상관없지만 a.out이라는 실행파일이 만들어짐

cp [복사하고자하는 파일] [원하는 이름] : 복사하고자 하는 파일을 원하는 이름으로 복사함

pwd : 현재 디렉토리 위치를 확인하는 명령어

cd [디렉토리명] : change directory의 약자로 디렉토리명으로 이동을 하는 명령어

mv [이름을 바꾸고자 하는 파일] [원하는 이름] : 이름을 바꾸고자 하는 파일을 원하는 이름으로 변경!

[이동시키고자 하는 파일] [디렉토리 이름] : 이동시키고자 하는 파일을 디렉토리 이름으로 이동시킴

rm -rf [삭제하고자하는 이름] : 삭제하고자 하는 이름을 디렉토리, 파일에 관계없이 다 지움

gdb [실행파일 이름] : 실행파일을 디버깅 하고자 할 경우 사용한다.

4. C언어

- // : 한 줄 주석, /* 내용 */ : 여러 줄 주석
- 변수 : 메모리에 정보를 저장할 수 있는 공간 (정보가 주소가 된다면 포인터)
- basic of memory architecture (그냥 외우기)

  stack -> 지역 변수가 위치하는 영역

  heap -> 동적 할당된 녀석들이 위치하는 영역

  data -> 전역 변수 및 static으로 선언된 것들이 위치하는 영역 초기화 되지 않은 모든 것은 0으로 저장

  text -> Machine code가 위치하는 영역
- 자료형 (데이터타입)

  ->int, short, char, float, double, long double
- 함수 : 필요한 입력을 받아 원하는 어떤 기능을 수행한 후 결과를 반환하는 프로그램 단위

  이름, input, output, 기능으로 구성

  독립적인 기능을 가지는 구성요소(module)

  ex) 2,4,6,8,10으로 30만들기!

edit fork download copy

```c
#include <stdio.h>

int myfunc(int num1, int num2, int num3, int num4, int num5){
int add = num1 + num2 + num3 + num4 + num5;
return add;
}

int main(void){
int n1 = 2, n2 = 4, n3 = 6, n4 = 8, n5 = 10;
int result = myfunc(n1,n2,n3,n4,n5);
printf("%d",result);
return 0;
}
```

Success #stdin #stdout 0s 4360KB comments (0)

stdin copy

```
Standard input is empty
```

stdout copy

```
30
```

ex) 3,6으로 21.3만들기!

edit fork download copy

```c
#include <stdio.h>

float myfunc(int num1,int num2){
        float result = (num1 * num2) +3.3;
        return result;
}

int main(void) {
        // 3,6 -> 21.3

        int num1 = 3, num2 = 6;
        printf("%lf",myfunc(num1,num2));

        return 0;
}
```

Success #stdin #stdout 0s 4324KB comments (0)

stdin copy

```
Standard input is empty
```

stdout copy

```
21.299999
```