**Xilinx zynq fpga.ti DSP,MCU기반의 프로그래밍 및 회로설계 전문가 과정**

**강사:INNOVA LEE(이상훈)**

[**Gccompil3r@gmail.com**](mailto:Gccompil3r@gmail.com)

**학생-윤지완**

[**Yoonjw7894@naver.com**](mailto:Yoonjw7894@naver.com)

**수업내용 정리**

**1.Typedef는 무엇인가 ?**

**자료형에 새로운 이름을 부여하고자 할 때 사용**

**주로 구조체나 함수 포인터에 사용한다. typedef는 c언어에서 예약어로도 불리고 있다.컴퓨터가 동작을 할 때 미리 이런 유형의 함수가 있다고 먼져 말해준다고 생각하면 된다.**

**2.malloc(),free()은 무엇을 하는가 ?**

**Memory 구조상 heap에 data를 할당함**

**data가 계속해서 들어올 경우**

**얼만큼의 data가 들어오는지 알 수 없음**

**들어올 때마다 동적으로 할당할 필요성이 있다.**

**Free()함수는 간단히 해체라는 단어가 제일 맞습니다. 이 함수는 malloc()의 반대의 기능을 한다고 보면됩니다.**

malloc 함수는 대충 보시면 이런 겁니다.

흠..malloc 함수에 마우스를 갖다댔더니 반환타입과 이름이 나옵니다.

반환타입은 void 군요.

반환되는 게 없다는 말입니다.

malloc(4 \* 3) 를 했을 경우,

이렇게 4byte공간이 3칸 확보된다고 보시면 됩니다.

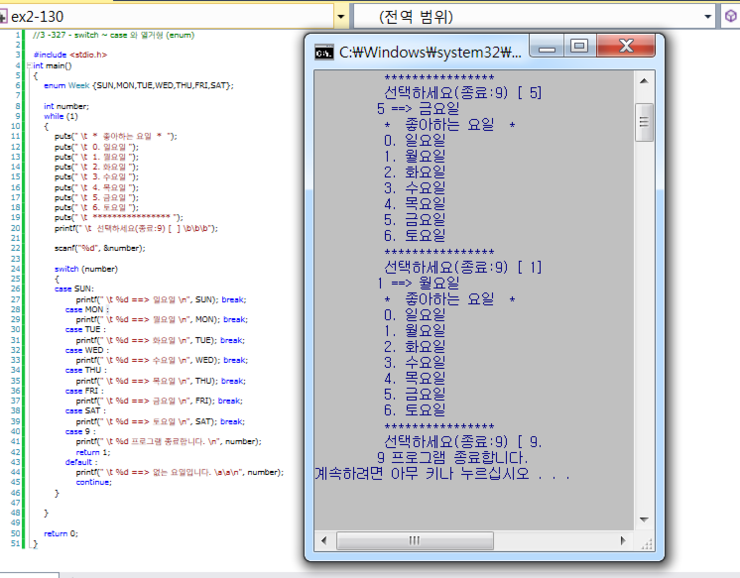
그리고 그것을 포인터변수로 캐스팅하면 다음과 같이 되죠.

이런 식으로 보시면 됩니다.

여기서 free()함수를 쓰면 이렇게 되죠.

**4.enum**

**이 함수는 쉽게 말해 열거형 함수라고 생각하면 됩니다. 열거형이란 정수를 열거해놓는다고 생각하면 된다.**



**이것을 쓰는 이유는 편하게 define하기가 쉽다는 것이다.**

**5.함수 포인터**

**void(\*signal(int signum, void(\*handler)(int)))(int);**

**함수 프로토타입이란?**

**리턴,함수명,인자에 대한 기술서**

**그렇가면 위 함수에 대한 프로토타입은?**

**이전에 배웟던 int(\*p)[2];->int 9\*0[2] p**

**리턴:void(\*)(int)**

**함수명:signal**

**인자:int signum과 void (\*handler)(int)**

**void(\*)(void):**

**void를 리턴하고 void 를 인자로 취하는 함수의 주소값을 저장할 수 잇는 변수 p**

**void ccc(void(\*p)(void))**

**리턴:void**

**이름:ccc**

**인자:void(\*p)(void)**

**이런식으로 사용이 된다.**

**6.함수 포인터를 사용하는 이유**

**1비동기 처리**

**2hw 개발 관점에서 인터럽트**

**3시스템 콜(유일한 sw인터럽트)**

**여기서 인터럽트들(sw,hw)은 사실상 모두 비동기 동작에 해당한다**

**결국 1번(비동기 처리)가 핵심이라는 의미다**

**그렇다면 비동기 처리라는 것은 무엇일까?**

**기본적으로 동기 처리라는 것은**

**송신하는 쪽과 수신하는쪽이 쌍방 합의하에만 당성된다**

**(휴대폰 통화등)**

**반면 비동기 처리는 이메일,카톡ㄷ으의 메신저에 해당한다.**

**그래서 그냥 일단 전져 놓으면 상대방이 바쁠땐 못보겠지만 그다지 바쁘지 않은 상황이라면 메시지를 보고 답변을 줄 것이다**

**이오 가타이 언제 어떤 이벤트가 발생할지 알 수 없는 것들을 다루는 녀석이 바로 함수 포인터다.**

**7.1858문제**

**#include<stdio.h>  
  
float test1(int a,int b){  
return a+b;  
}  
void test2(void(\*p)(void))  
{  
p();  
printf("no test call\n");  
}  
float test3(int x,int y){  
return (x+y)/2.0;  
}  
  
float (\*(\*test(void(\*p)(void))(float(\*)(int,int))(int,int)  
{  
float tal;  
tal=p(3,4);  
printf("%d",tal);  
return test1;   
}  
  
int main(void)  
{  
int total;  
total=test(test3)(5,1);  
printf("%d",total);  
return 0;  
}**