

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 윤연성

whatmatters@naver.com

```
/*
```

```
----- typedef -----
```

```
#include <stdio.h>
```

```
//typedef
```

```
typedef int INT; // int 를 INT 로 바꿈
```

```
typedef int* PINT; // int*를 PINT 로 바꿈
```

```
int main(void)
```

```
{
```

```
INT num = 3;
```

```
PINT ptr = &num;
```

```
printf("num = %d\n", *ptr);
```

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
typedef int INT[5];
```

```
int main(void)
```

```
{
```

```
int i;
```

```
INT arr = {1, 2, 3, 4, 5}; //INT arr
```

```
for(i = 0; i < 5; i++)
```

```
printf("arr[%d] = %d\n", i, arr[i]);
```

```
return 0;
```

```
}
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
char *str_ptr = (char *)malloc(sizeof(char)* 20);
```

```
printf("input string:");
```

```
scanf("%s", str_ptr);
```

```
if(str_ptr != NULL)
```

```
printf("string = %s\n", str_ptr);
```

```
free(str_ptr);
```

```
return 0;
```

```
}
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int *num_ptr = (int *)malloc(sizeof(int) * 2);
```

```
printf("Input Integer:");
```

```
scanf("%d %d", &num_ptr[0], &num_ptr[1]);
```

```
num_ptr = (int *)realloc(num_ptr, sizeof(int) * 3);
```

```
printf("Input one more Integer: ");
```

```
scanf("%d", &num_ptr[2]);
```

```
if (num_ptr != NULL)
```

```
printf("Integer = %d, %d, %d\n", num_ptr[0], num_ptr[1], num_ptr[2]);
```

```
free(num_ptr);
```

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
struct pos
```

```
{
```

```
double x_pos;
```

```
double y_pos;
```

```
};
```

```
int main(void)
```

```
{
```

```
double num;
```

```
struct pos position; // struct pos 까지 새로운 데이터타입
```

```
num = 1.2; //구조체 = 커스텀 데이터타입
```

```
position.x_pos = 3.3; //구조체 안의 x_pos 를 쓰겠다라는건 .을 써야됨
```

```
position.y_pos = 7.7;
```

```
printf("sizeof(position) = %lu\n", sizeof(position));
```

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
void add_arr(int *arr)
```

```
{
```

```
int i;
```

```
for(i=0; i<3; i++)
```

```
{
arr[i] += 7; //arr[i] + 7 = arr[i]
}
}
```

```
void print_arr(int *arr)
{
int i;
for(i =0; i <3; i++)
{
printf("arr[%d] = %d\n", i, arr[i]);
}
}
```

```
int main(void)
```

```
{
int arr[3] = {1, 2, 3};
add_arr(arr); //주소값전달
print_arr(arr); //주소값전달
return 0;
}
```

```
*ptr = 27;
```

```
printf("ptr value = %d\n", *ptr); // Segmentation fault (접근하지말아야할곳에 접근함)
return 0; //쓰레기값 0xcccccccc
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
{
char str1[33] = "pointer is important!";
char *str2 = "pointer is important!";
```

```
printf("str1 = %s\n", str1);
printf("str2 = %s\n", str2);
```

```
return 0;
```

```
}
```

```
// *별하나당 주소  
// int *p = &num;  
// int **p = &p;  
// p(vum)  
// p(*p)  
// p (**pp)
```

```
#include <stdio.h>
```

```
int main(void)  
{  
    int num1 = 3, num2 = 7;  
    int *temp = NULL;  
    int *num1_p = &num1; //&주소를 반환  
    int *num2_p = &num2; // * 내용을 반환  
    int **num_p_p = &num1_p;  
  
    printf("*num1_p = %d\n", *num1_p);  
    printf("*num2_p = %d\n", *num2_p);  
    temp = *num_p_p; // *주의!  
    *num_p_p = num2_p;  
    num2_p = temp;  
  
    printf("*num1_p = %d\n", *num1_p);  
    printf("*num2_p = %d\n", *num2_p);  
  
    return 0;  
}
```

```
#include <stdio.h>
```

```
int main(void)  
{  
    int i,j,n1,n2,n3;  
    int a[2][2] = {{10,20},{30,40}};  
    int* arr_ptr[3] = {&n1, &n2, &n3};  
    int (*p)[2] = a; //int (*p)[2] = int (*)[2] p int *p[2]  
    // int 형 두개짜리 포인터 8 바이트  
    //  
  
    for(i = 0; i < 3; i++)  
        *arr_ptr[i] = i;  
  
    for(i = 0; i < 3; i++)
```

```
printf("n%d = %d\n", i, *(arr_ptr[i])); // int* arr_ptr[3]이랑 *arr_ptr 다름
for(i = 0; i < 2; i++)
printf("p[%d] = %d\n", i, *p[i]);

return 0;

}
```