

2018. 3. 14 수 - 15 회차

과정 : TI, DSP, Xilinx Zynq FPGA, MCU 기반의 프로그래밍 전문가 과정

Prof. 이상훈

[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

Stu. 정상용

[fstopdg@gmail.com](mailto:fstopdg@gmail.com)

자료구조\_마지막

avl\_tree ins, del 재귀함수없이 실현

+

복습

# I. Array

Why → 변수가 많은 경우.

다수의 변수를 사용할 경우, for 문을 이용하여 간단히 작업 가능.  
 → index : 0부터 시작.

형제 : data type - name - [data type] = { ↓ };

Ex> int sensor-data[3] = {50, 45, 47}; 값들은 작음 (변수를 처리.)

→ 메모리상에서 순차적으로 배열이 되어있다.

	50	45	73	32	130	74	124	127
index	0	1	2	3	4	5	6	7

→ 0부터 시작

## I.i) Character type Array.

Why → 메모리에서 변경이 가능.

String인 문자열 "I'm SangYong" → 메모리에서 변경 불가능.

Char name[13] = "I'm SangYong" → 메모리에서 변경 가능  
 ↳ name[5] = 'Y';

+ 추가. [ ] → 실제 배열의 주소 + 1

↳ 값 넣지 넣었을 경우 문제가 생길 수 있다  
 다른 라인과 식으로 들어오는 경우.

## I.ii) High Dimension Array

다중 배열 (2차원 배열)  
 ∴ stack 개념

[ 2차원 배열: 행과 열에 용이.  
 3차원 배열: 진과 열과 높이에 용이.

Ex> 행렬 짜는 법 ⇒ B.C ★ 순서로 것 (for, in 사용법)

## I.iii) 배열의 초기화

↳ stack에 처음 값을 지정해주는 행위.

Arr[2][2][3] = { { { 10, 20 }, { 30, 40 } }, { { 50, 60 }, { 70, 80 } } }, { { { 90, 100 }, { 110, 120 } }, { { 130, 140 }, { 150, 160 } } }

{ { { 170, 180 }, { 190, 200 } }, { { 210, 220 }, { 230, 240 } } }

Arr[3][2] →  $\begin{matrix} \uparrow \\ [0] \\ [1] \\ [2] \\ [3] \\ [4] \\ [5] \end{matrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$   
 ↳ 4byte  
 ⇒ Arr[2][1] = 6  
 Arr[0][0] = 1



배열 = 주↑

int arr[4] = {10, 20, 30, 40}

배열의 주소 = 배열 맨 앞 자리의 주소.

%p: 주소를 출력

&: 주소를 가져옴

Ex) 위의 코드를 확인  $\Rightarrow$  10.C.