

url : <https://www.youtube.com/watch?v=ZFyRHylXlv4>

● 표준함수에 대한 이해

표준함수는 컴파일러가 다르더라도 최소한 이 라이브러리 안의 함수는 해당 CPU 구조에 맞춰서, 모든 기기에서 똑같은 기능을 하게끔 구현해준다.

결국, x86 시스템구조에서 printf나 scanf를 쓰던, MCU에서, scanf, printf를 쓰던 결과는 똑같이 나온다고 생각할 수 있다. 그러나, 세부적인 구조는 다르기 때문에 해당 컴파일러의 매뉴얼을 읽어보는 작업이 필요하긴하다.

1. scanf에 대한 이해

정수형 변수를 선언하고, 해당 변수에 문자 하나를 입력받는 함수를 선언하였다.

```
#include <stdio.h>

void main()
{
    // 표준함수는 컴파일러가 다르더라도 최소한 이 라이브러리 안의 함수는 구현해준다.
    // gcc 뿐만 아니라, gcc-arm 에서도 호환이 되는 문법이다.

    int A;

    printf("정수를 입력 : __\b\b ");
    scanf("%c",&A);

    printf("입력된 값 : %c\n",A); // 문자로 출력
```

이를 컴파일 하면, 약간의 경고가 뜨긴한다. (정수형을 왜 문자로 출력하려 하냐는 경고)

```
input.c:11:17: warning: format '%c' expects argument of type 'char *', but argument 2 has type
'int *' [-Wformat=]
   11 |         scanf("%c",&A);
      |         ^~
      |         |
      |         | int *
      |         char *
      |         %lc
```

그러나 무시해주고 다음과 같이 실행하였다.

```
root@LAPTOP-BQ4AK11N:/project/24-Cstudy06# ./input.out
정수를 입력 : 124
입력된 값 : 1
```

scanf는 입력버퍼의 값을 주어진 자료형에 맞춰서 해당 변수에 집어넣는 역할을 한다.

124를 입력했기에, 문자 1 2 4가 버퍼에 들어갔고, 해당 버퍼에서 1바이트만 pop해서, 정수형 변수에 넣었기 때문에, 1만 A 변수에 들어가게 된다.

그리고, 이를 A에 넣었기 때문에, 문자 1만 출력된걸 확인할 수 있다.

● 문자열

```
#include <stdio.h>

void main()
{
    char str[20] = "hello";

    char str1 = 'H';
    char str2 = 'E';
    char str3 = 'L';
    char str4 = 'L';
    char str5 = 'O';
    char str6 = '\0';

    printf("%s\n",str);

    printf("%c%c%c%c%c%c\n",str1,str2,str3,str4,str5,str6);
}
```

문자열과 문자 여러개를 출력해보면서 둘의 차이가 어떤건지 알아보는 실습을 진행하였다.

강사님이 char 형 여러개로 문자열을 흉내낸 것은 문자열과 다르게 연속적이지 않다는 말을 하였다.

그러나, 지역함수의 스택영역에 들어가는 지역변수는 스택구조로 되어있어서, 해당 코드에서 str 배열의 모든 요소가 스택에 저장되고, 지역변수들이 차례대로 들어가는 것으로 알고 있다.

str6 (1 바이트)	0x3a
str5 (1 바이트)	0x29
str4 (1 바이트)	0x28
str3 (1 바이트)	0x27
str2 (1 바이트)	0x26
str1 (1 바이트)	0x25
str 20바이트 (20바이트)	0x24
패딩 (4바이트)	0x10
이전 rbp의 값 (4바이트)	0x0c
caller 주소 (4바이트)	0x08

함수호출규칙에 의거하여

여기서, 이전 caller의 주소는 호출자가 callee를 호출했을때의 위치를 나타낸다. (push)

이전 rbp 레지스터의 값은 caller의 스택 베이스 포인터의 주소를 나타낸다. (push)

패딩은 이 레지스터, 주소값들과 지역변수들을 구분하기 위한 것이다.

그리고, 가장먼저 호출한 20개의 문자배열이다 (push)

그 뒤부터 차례대로 선언된 문자 변수들이다. (push)

그럼, str1부터 str6까지 포인터를 이용해 주소연산을하면 쉽게 문자들이 차례대로 나타
나야 한다.

이를 코드로 구현하면 밑으로 나타낼 수 있을 것 같다.

결과도 예상처럼 나타내게 되었다. 아마 강사님이 실수를 하신 것 같다.

```
#include <stdio.h>

void main()
{
    char str[20] = "hello";

    char str1 = 'H';
    char str2 = 'E';
    char str3 = 'L';
    char str4 = 'L';
    char str5 = 'O';
    char str6 = '\0';

    printf("%s\n",str);

    printf("%c%c%c%c%c%c\n",str1,*(&str1+1),*(&str1+2),*(&str1+3),*(&str1+4),*(&str1+5));
}
```

```
root@LAPTOP-BQ4AK11N:/project/24-Cstudy06# ./string.out
hello
HELLO
```