



C언어 (CSE2035) (Chap9. Pointer Applications) (2-2)

Sungwon Jung, Ph.D.

Dept. of Computer Science and Engineering

Sogang University

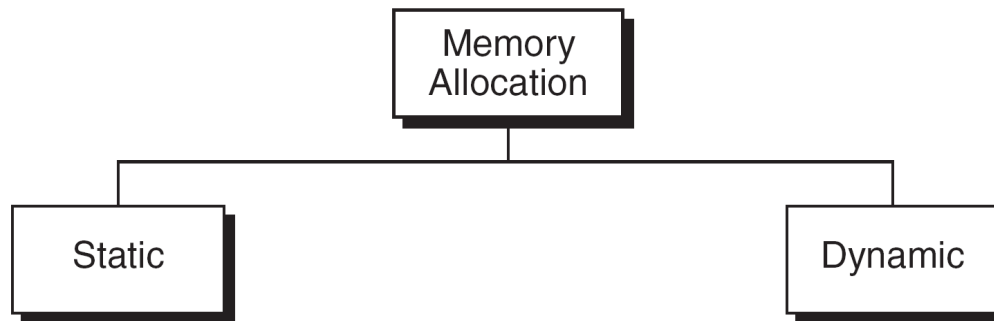
Seoul, Korea

Tel: +82-2-705-8930

Email : jungsung@sogang.ac.kr

Memory allocation functions

- C언어를 사용하면서 변수 영역의 메모리를 할당 받는 것으로 다음의 두 가지 방법이 가능하다.



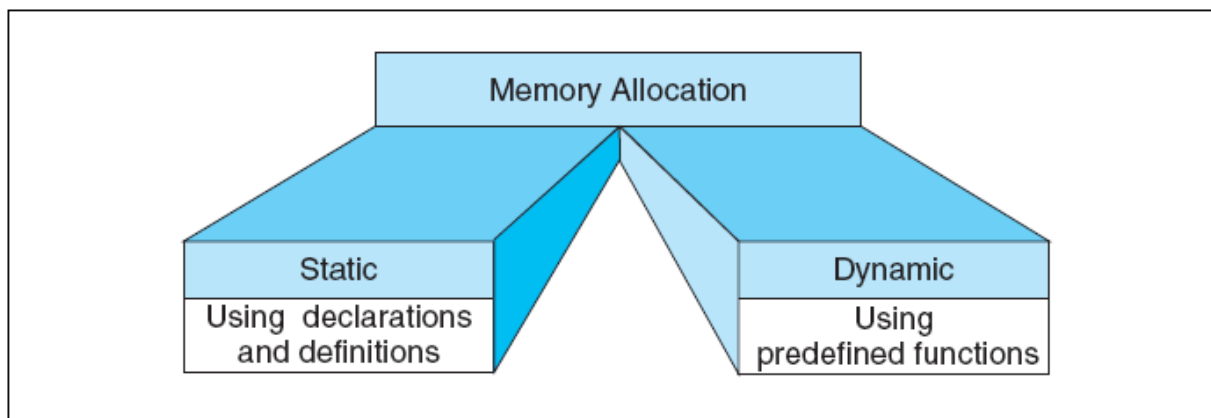
- **Static memory allocation**

- 정적 메모리 할당은 소스프로그램 안에서 메모리의 선언과 정의가 완벽하게 명시되어 있어야 한다.
- 실행 시간 동안에 할당된 byte 수 등이 변할 수 없다.
 - 입력되는 데이터 길이에 대해 정확하게 알고 있어야만 사용 가능하다.

Memory allocation functions

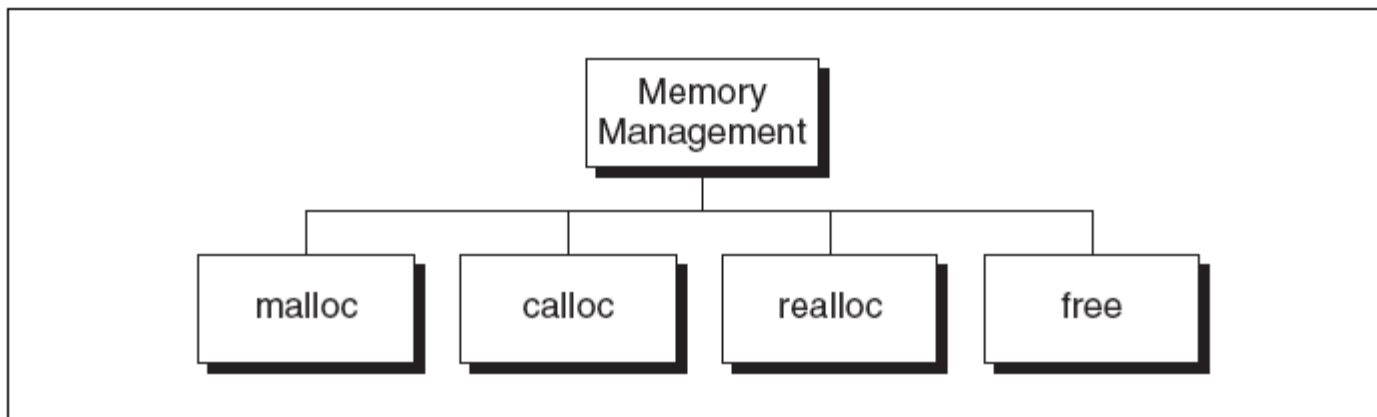
■ 동적 메모리 할당

- 메모리의 낭비를 최소화하기 위해 프로그램의 실행 중에 입력되는 데이터에 맞게 기억공간이 할당되어야 한다. ➔ 메모리의 동적 할당
- 메모리의 동적 할당과 해제 기능은 라이브러리로 구현되어 있고, 이를 이용하기 위해서는 반드시 포인터를 이용한다.
 - 동적 할당이란 프로그램 실행 중에 ,즉 `run time`때 메모리 영역을 할당한다는 뜻이다.



Memory allocation functions

■ Memory management functions



- 메모리의 동적 할당에 사용되는 메모리 할당 함수들
 - malloc(), calloc(), realloc()은 메모리를 할당하기 위한 함수
 - free()는 더 이상 사용하지 않는 메모리를 리턴하는 함수
- 함수를 사용하기 위해서는 “stdlib.h” 헤더파일을 포함해야 한다.
 - #include<stdlib.h>



malloc()

- **malloc()** 함수는 메모리를 할당하기 위한 함수이다.
 - 함수의 프로토타입은 다음과 같다.

```
void *malloc (size_t size);
```

- **malloc()** 함수는 할당한 메모리의 시작주소를 **void** 포인터로써 리턴한다. 할당에 실패한 경우 **NULL**을 되돌려 준다.
- **malloc()** 은 크기가 **size**인 객체(기억 장소)를 할당한다.
- **size_t** 타입은 **stdio.h**에서 정의된 타입으로 **unsigned integer**의 값을 저장한다.

malloc()

- 다음은 **malloc()** 함수의 일반적인 사용 예이다.

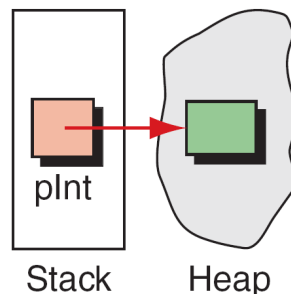
```
pInt = (int *)malloc (sizeof (int));
```

int 타입 포인터

- pInt에는 하나의 int형 데이터를 저장할 수 있는 크기의 메모리가 할당
- n개의 int 형 원소를 저장할 수 있는 메모리를 할당 받기 위해서는
 - **malloc(sizeof(int) * n)**
- 만일 malloc() 함수가 메모리 영역을 할당하는데 실패한다면 NULL 포인터를 리턴할 것이다.

```
if (!(pInt = malloc(sizeof(int))))
    // No memory available
    exit (100) ;
// Memory available
...
```

<malloc함수 호출 예>



Stack

Heap

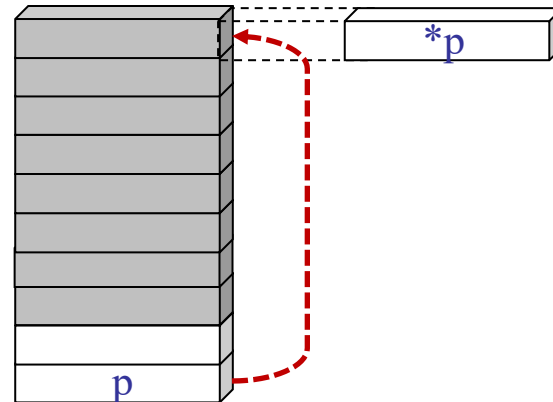
Dynamic allocation of array

■ Dynamic allocation of array

```
int *p;  
p = (int *)malloc(8 * sizeof(int));
```

- malloc은 8개의 int를 위한 기억 장소를 할당해 그 시작 주소값을 되돌려 준다.

malloc 함수로
할당된 영역



- 포인터 변수가 있으면 포인터가 가리키는 영역을 배열처럼 사용가능
- 포인터 변수 p가 가리키는 원소의 i번째 원소를 나타내는 표현식
 - $*(p+i)$, $p[i]$

malloc()

■ 예제프로그램 - malloc을 이용한 변수 할당

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

```
int main(void)
{
```

```
    int *p;
```

```
    p=(int*)malloc(1*sizeof(int));
```

```
    if(p==NULL) {
```

```
        printf("not allocated\n");
        return 1;
```

```
    }
```

```
    *p=1;
```

```
    printf("%d\n", *p);
    return 0;
```

```
}
```

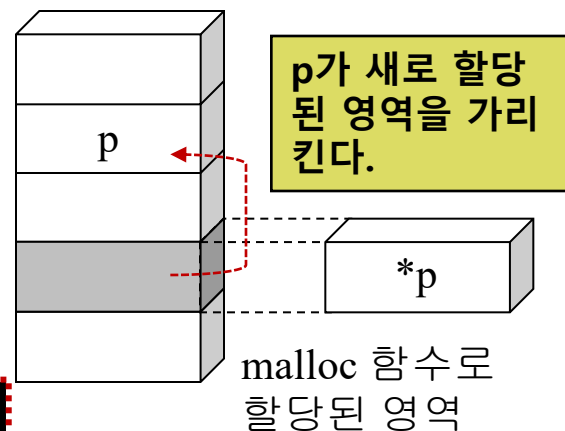
```
[root@mclab chap10]# ./chap10-5
```

```
1
```

```
[root@mclab chap10]#
```

int형 1개 크기의 메모리 할당하고 그 주소값을 p에 넘겨준다.

새로 할당된 메모리영역에 접근



malloc()

- 예제프로그램 - 메모리를 동적으로 할당 받아 배열로 사용하는 프로그램

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int *ip;
7     int i, sum = 0;
8
9     ip = (int *)malloc(5*sizeof(int));
10    if(ip==0) {
11        printf("short memory\n");
12        return 1;
13    }
14    printf("Input the age of 5 people : ");
15    for(i=0; i<5; i++) {
16        scanf("%d", ip+i);
17        sum+=ip[i];
18    }
19    printf("Average : %.11f\n", sum/5.0);
20    free(ip);
21    return 0;
22 }

```

20바이트
의 기억공
간 할당

```

[root@mclab chap10]# vi chap10-6.c
[root@mclab chap10]# gcc -o chap10-6 chap10-6.c
[root@mclab chap10]# ./chap10-6
Input the age of 5 people : 21 27 24 22 35
Average : 25.8
[root@mclab chap10]#

```

할당 받은 메모리
를 배열처럼 사용
하여 값을 입력 받
는 부분