

#01. Memory(메모리 공간)

전미정

1. 정적 메모리 할당(Static Memory Allocation)

: 할당해야 할 메모리의 한계 크기를 명확히 알고 있는 경우 사용하며, 프로그램 시작시 메모리 공간을 고정/할당한다. 프로세스가 시작할때 정해진 메모리량은 시작할 때부터 프로세스가 종료될 때까지 유지된다. 정적 메모리 할당은 최대 한계 크기보다 약간 더 크게 잡기 때문에 약간의 메모리 낭비가 발생할 수 있다. 하지만 사용하기 쉬운 방법이며 버그가 발생할 확률이 적다는 장점이 있다. 동적 메모리 할당으로 발생할 수 있는 메모리 누수, 혼동되는 포인터의 사용 등의 문제를 피할 수 있으며, 안정적인 프로그램 작성시 주로 사용한다.

2. 동적 메모리 할당(Dynamic Memory Allocation)

: 프로그램을 실행하는 동안 필요한 메모리 공간을 할당하는 것을 의미하며, 메모리 사용이 끝나면 다른 프로그램이나 운영체제가 사용 할 수 있도록 해제해야한다. 기존의 원시 자료형이 아닌 크기가 큰 데이터를 담을 때 사용하며, 힙(heap)영역을 지칭한다. 힙에 필요한 만큼의 메모리가 할당되면 그 영역은 다른 스레드나 프로그램이 사용할 수 없으며, 포인터값으로 접근이 가능하다. 상황에 따라 필요한 만큼의 메모리가 할당되므로 경제적이고, 언제든지 메모리 크기 조절이 가능하지만 프로그래머가 메모리 해제 과정을 명시적으로 해제 해줘야하며 그렇지 않을 경우, 메모리 누수와 같은 버그의 원인이 되기도한다.

3. 프로세스 메모리 영역

1) code 영역

: 코드 자체를 구성하는 메모리 영역으로, 프로그램 명령어 및 기계어로 제어되는 영역

2) data 영역

: 전역변수, 정적변수, 배열, 구조체 등이 저장된다. 프로그램 실행시 생성되어 종료될때까지 유지된다.

3) heap 영역

: 동적 메모리 할당 공간으로 메모리 주소 값에 의해 참조된다. 런타임시 공간이 동적으로 변하며 프로그래머가 조절할 수 있는 구역이다. 프로세스의 사용이 끝나면 반드시 메모리 할당을 해제해야한다.

(heap이라는 단어는 자료구조의 heap과는 다른 것으로 생각되며, 단어 자체의 뜻 : (아무렇게나 쌓아 놓은) 더미로 추정됨)

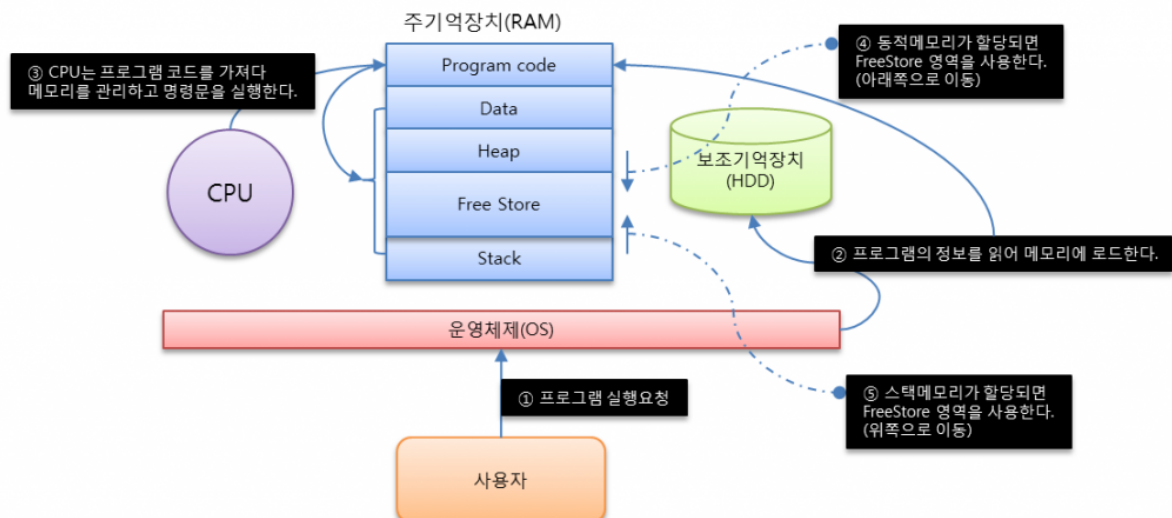
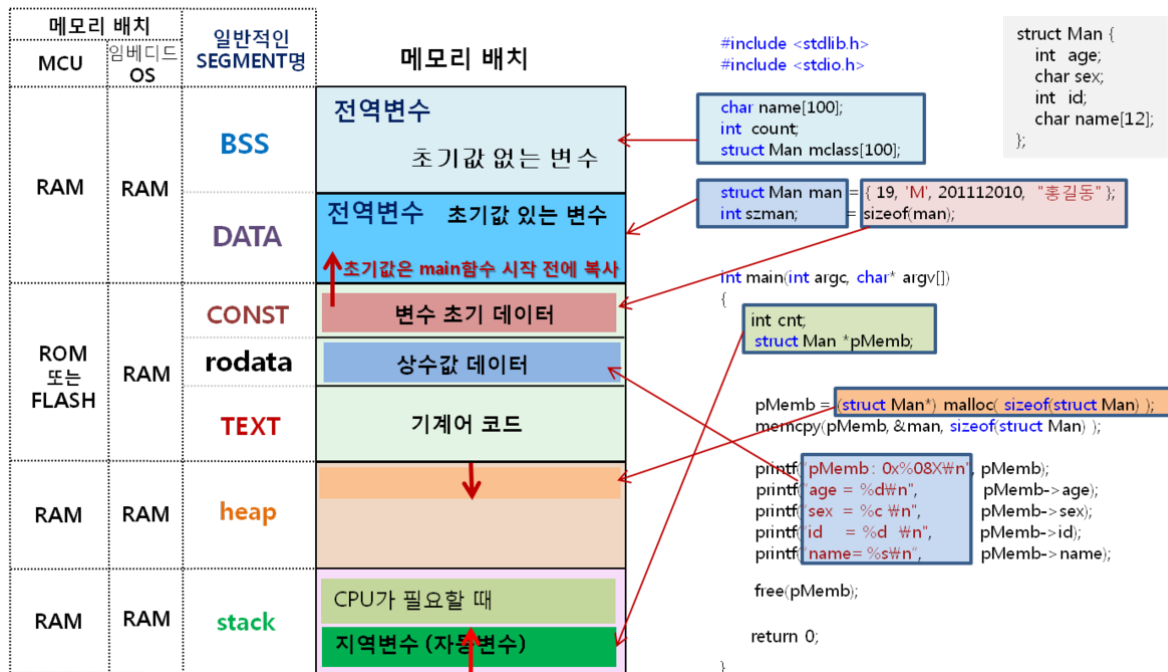
4) stack 영역

: 프로그램이 자동으로 사용하는 임시 메모리 영역으로 지역변수, 매개변수, 리턴 값 등을 저장하는 곳이다. 함수 호출 시 생성되고 함수가 끝나면 시스템에 반환된다. 컴파일 타임에 결정되며, 런타임에 스택 사이즈를 바꿀 수 없다. 함수 시작시 공간이 할당되고 함수가 종료되면 메모리를 반환한다.

Q. Objective-C에서의 메모리 할당은?

: 클래스에 의해 생성된 객체내 인스턴스, 객체 안의 변수 등도 힙에 저장된다.

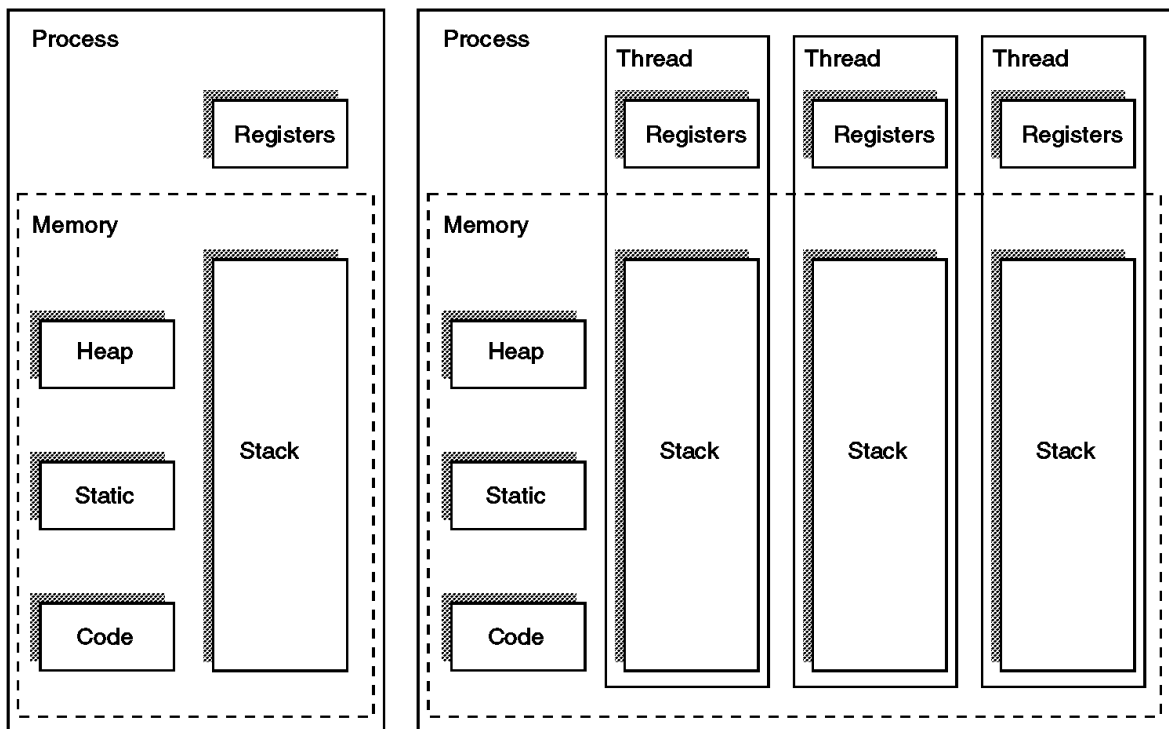
Q. 메모리 영역은 언어마다 달라질 수 있지만 보편적으로 위와 같은 내용과 같다.



**** 메모리 공간에서의 스레드(thread) :** 프로세스 안에서 존재는 실행 흐름인

스레드가 여러개일 경우, 프로세스 내의 heap, static, code영역을 공유한다.

스레드간 통신이 가능하며, 메모리내 자유로운 접근이 가능하다.



+ 추가: 전역 변수, 스택 로컬 변수(Global variable/Static local variable)

```
for (int i = 0; i < 5; ++i) {
    int n = 0;
    printf("%d ", ++n); // prints 1 1 1 1 1 - the previous value is lost
}
```

```
for (int i = 0; i < 5; ++i) {
    static int n = 0;
    printf("%d ", ++n); // prints 1 2 3 4 5 - the value persists
}
```

```
12
13 void f1(){
14     static int i = 0;
15
16     i++;
17 }
18
19 void f2(){
20     i = 3;
21 }
22
23
24
```

1 Use of undeclared identifier 'i'

전역변수와 스택 변수 모두 메모리의 데이터 영역에 저장되며, 한번 생성되면 함수가 끝나더라도 프로그램이 종료될 때까지 없어지지 않는다.

차이점은 다른 함수에서의 호출/접근 가능 여부인데, 전역변수는 모든 함수에서 호출/접근 가능하지만, 스택 로컬 변수는 해당 함수안에서만 호출/접근이 가능하다.