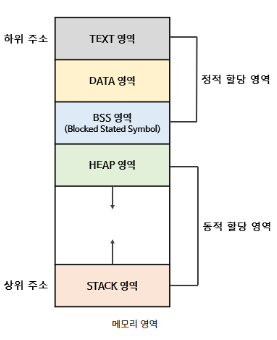
프로세스란? 운영체제가 자원을 할당하는 단위.

이 프로세스가 메모리를 할당 받으면, 그 메모리를 관리하기 위해 특정 구조로 관리하는데 이를 프로세스 주소 공간이라고 한다.

프로세스 주소공간은 스택영역, 힙영역, BSS(block stated symbol)영역, 데이터영역, 텍스트(코드)영역으로 나뉜다. 스택 영역과 힙 영역은 동적으로 할당된다.



1. Stack 영역

함수의 호출과 관계되는 지역 변수 + 매개 변수가 저장되는 영역이다.

함수의 호출과 함께 할당, 함수가 완료되면 소멸된다.

메모리의 높은주소 -> 낮은 주소의 방향으로 할당된다.(힙과는 반대방향)

재귀 함수가 깊게 호출되거나, 함수가 지역 변수를 너무 많이 가지는 등 stack 영역이 초과되면 stack overflow에러가 발생한다.

컴파일 시점에서 크기가 결정된다.

2. Heap 영역

런타임에 크기가 결정된다. 사용자에 의해 공간이 동적으로 할당 및 해제된다.

주로 클래스, 클로저 등의 참조형 데이터들이 할당된다.

메모리의 낮은주소 -> 높은 주소의 방향으로 할당된다.(스택과는 반대방향)

3. BSS 영역

초기 값이 없는 전역 변수나 Static 변수가 저장되는 영역이다.

초기화되지 않은 변수는 프로그램이 실행될 때 영역만 잡아주면 된다. -> 많아져도 실행 코드 사이즈가 늘어나지 않는다.

4. 데이터 영역

초기 값이 있는 전역 변수나 Static 변수 등 프로그램이 사용할 수 있는 데이터를 저장하는 영역.

프로그램의 시작과 함께 할당되며, 프로그램이 종료되면 소멸한다.

5. 텍스트(코드) 영역

프로그램이 실행될 수 있도록 CPU가 해석 가능한 기계어 코드가 저장되어 있는 공간.

컴파일 시점에서 결정되고, 중간에 코드가 변경되면 안되므로 ReadOnly 상태로 저장되어 있다.

구역을 나눈 이유

최대한 데이터를 공유하여 메모리 사용량을 줄이기 위해서이다.

코드 영역은 프로그램 자체에서 모두 같은 내용이므로 따로 관리하여 공유한다.

Stack과 데이터 영역이 나뉜 이유

스택 구조의 특성과(재귀함수 등) 전역 변수의 활용성을 위한 것.

프로그램의 함수와 지역 변수는 LIFO특성을 가진 스택에서 실행

이 함수들 안에서 공통으로 사용하는 변수, 함수들을 전역, static으로 선언하여 Data영역에 올려서 공유함으로써

똑같은 공간을 여러 개 만들지 않고 메모리를 아낄 수 있다.

스택 영역과 힙 영역

Stack 영역에서 원시 자료형이 아닌 String이나 클래스 등은 실제 객체를 갖고 있는 것이 아니라 참조를 갖고 있다.

실제 객체는 Heap영역에서 관리 되고, 그러므로 Stack영역의 크기는 실제로 클 필요가 없다.

그래서 앞에 말했듯이 스택은 컴파일 시점에서 크기가 결정된다.