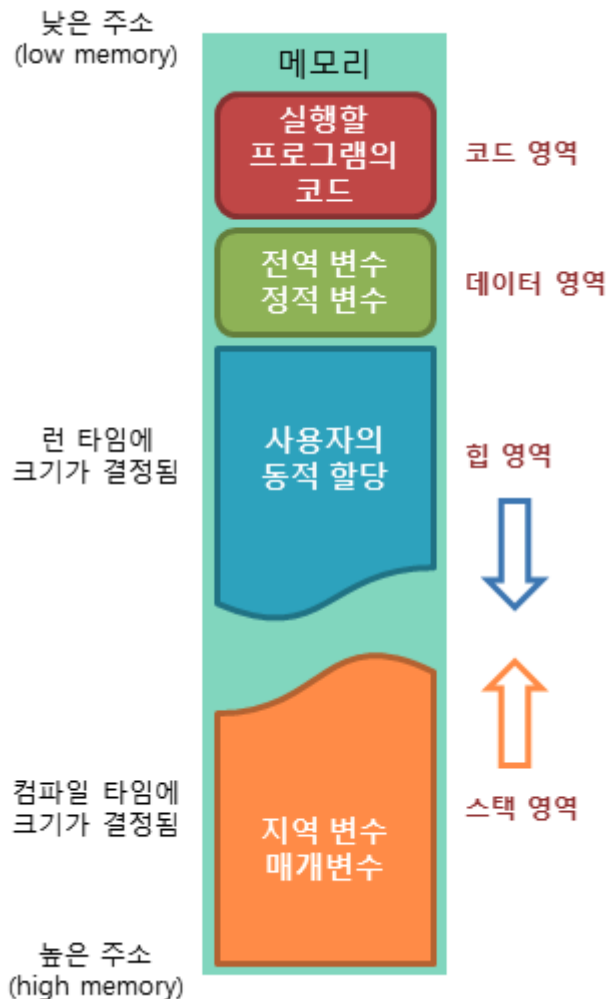


StackOverflow



코드(Code) 영역

실행할 프로그램의 코드가 저장되는 영역이다.

프로그램이 시작하고 끝날 때까지 메모리에 계속 남아 있다.

컴파일된 기계어가 들어가게 된다.

CPU는 코드 영역에 저장된 명령어를 하나씩 가져가서 처리한다.

데이터(Data) 영역

프로그램의 **전역 변수와 정적 변수, 문자열 상수가 저장되는** 영역이다.

프로그램이 시작하고 끝날 때까지 **메모리에 계속 남아** 있다.

스택(Stack) 영역

함수의 호출과 관계되는 **지역 변수와 매개변수가 저장되는** 영역이다.

함수의 호출과 함께 **할당되며 함수의 호출이 완료되면 소멸**한다.

이렇게 스택 영역에 저장되는 함수의 호출 정보를 스택 프레임(stack frame)이라고 한다.

프로그램이 자동으로 사용하는 임시 메모리 영역이다. **컴파일 시에 크기가 결정**된다.

힙(Heap) 영역

사용자가 직접 관리할 수 있고 해야만하는 영역이다.

힙 영역은 **사용자에 의해 메모리 공간이 동적으로 할당되고 해제**된다.

malloc()/new 연산자를 통해 할당하고, free()/delete 연산자를 통해서 해제가 가능하다. 런타임 시에 크기가 결정된다.

Stack vs Heap

Stack과 Heap은 OS/컴퓨터 구조에서 빈번하게 비교되는 주제이다.

- 메모리 구조상 Stack이 커지면 Heap이 작아지고, Stack이 작아지면 Heap이 커지는 구조이다.

그렇다면 할당 속도는 누가 더 빠를까?

Stack은 이미 할당 되어 있는 공간을 사용하므로 포인터의 위치만 바꿔주는 단순한 CPU Instruction이다. 하지만 Heap에서의 할당은 동적인 메모리 영역으로 고려해야하는 요소가 많아 Stack보다 많은 CPU Instruction을 필요로 한다. **결론은 Stack이 할당 속도가 훨씬 빠르다.**

Overflow

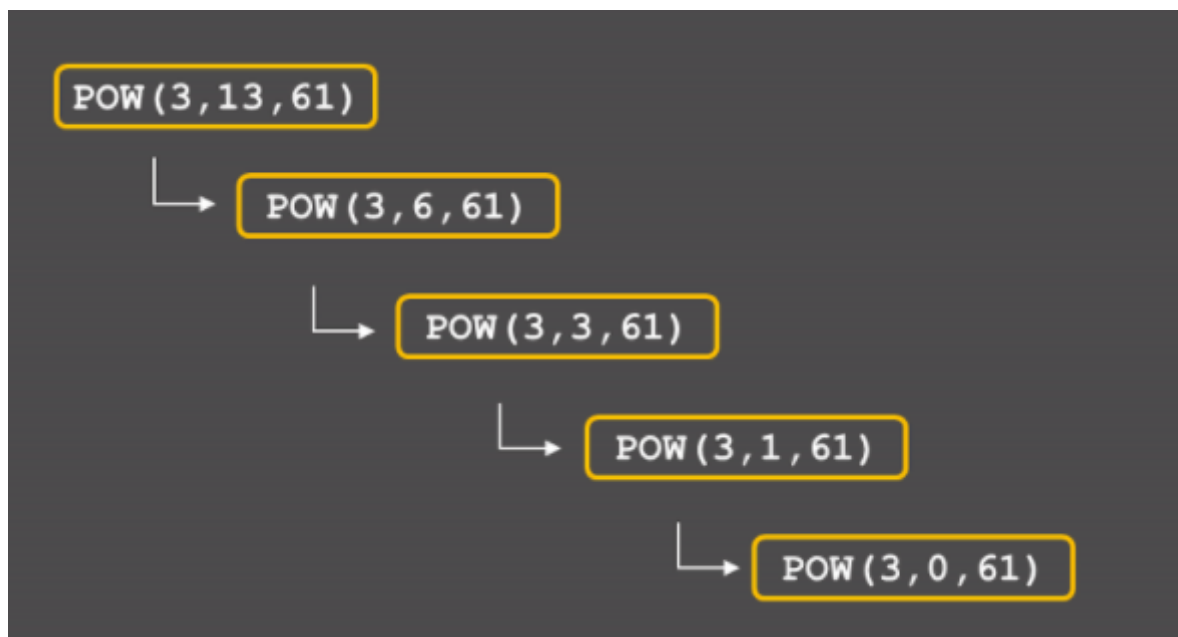
Overflow가 생기는 이유에 대해서 생각해본다면 Stack과 Heap에 채워지는 **메모리 주소의 방향이 달라서이다**. Stack은 주소값이 위에서부터 채워진다. 반면 Heap의 주소값은 아래에서부터 채워져 내려온다. 이때, 두 메모리 영역의 주소가 겹치게 되면 Overflow가 발생한다.

Heap Overflow

Heap이 아래에서부터 주소값을 채워져 올라가다 Stack영역을 침범한 경우이다.

Stack Overflow

Stack 영역이 Heap을 침범한 경우이다.



재귀함수가 stack overflow를 일으키는 이유

함수를 호출 시 함수의 입력 값(매개변수), 리턴 값, 리턴 됐을 때 돌아갈 위치 값 등을 스택에 저장한다. 재귀함수를 사용하면 함수가 끝나지 않은 채 연속적으로 함수를 호출함으로

stack에 메모리가 끊임 없이 쌓이게 됩니다.

즉, 재귀함수를 사용하면 호출시 마다 스택에 해당 함수관련 정보가 memory를 계속 쌓이게 되고, 함수가 return값을 반환해야 stack이 비워지는데, 함수가 끝나지 않은 도중에 계속 함수가 호출됨으로 함수의 깊이는 계속 깊어만 진다. 즉 계속 메모리가 쌓이게 되고 그래서 stack overflow가 일어난다.

⇒ 깊이(depth)를 늘리면 깊이가 줄어들 때마다 depth 만큼의 실행 컨텍스트가 저장될 메모리 공간 필요 << 반복문 기반 알고리즘을 통해 메모리 사용(=함수 호출의 비용) 절약 가능

알고리즘을 풀 때

- 메모리 제한(스택, 힙, 정적메모리 포함)을 초과하지 않는 범위 내에서는 재귀를 몇 번이고 들어가도 됨
- 다만 예외적으로, 파이썬과 같은 언어에서는 내부적으로 재귀 횟수에 제한이 있습니다. 이런 경우 제한을 풀 수 있는 방법을 사용하거나, 다른 언어를 사용해야 합니다.

파이썬에서 재귀

- Python3 기준 최대 재귀 깊이는 1000이라고 한다.
- 임의적으로 파이썬 기본 재귀 깊이 제한을 변경하면 해결되는 것이다. 이를 위해서는 파이썬 sys 라이브러리의 setrecursionlimit 메소드

```
import sys
sys.setrecursionlimit(2500)
```

Context Switching 관련

Context Switching이란?

면접에서 받았던 질문에 대해 정리하고 조금 더 자세하게 알기 위해 작성합니다. 해당 Task가 끝날때까지 다음 Task는 기다릴 수 밖에 없습니다. 또한 반응속도가 매우 느리고 사용하기 불편합니다. Computer

👤 <https://nesoy.github.io/articles/2018-11/Context-Switching>

