

5957 객체지향 프로그래밍

60222117 이서현

목차

1. Process Memory Layout

- A. 명령하는 코드와 데이터 관련 코드
- B. 폰노이만 컴퓨터구조(Von Neumann architecture)

2. Memory Allocation

- A. Object type
- B. Primitive data type

주제 : Process

1. Process Memory Layout

A. 명령하는 코드와 데이터 관련 코드

컴파일러는 코드를 CPU가 실행을 하기 위한 코드와 준비하기 위한 코드로 나누어 해석한다. CPU가 준비하기 위한 코드는 저장공간을 만들고 메모리를 할당하는 역할의 코드를 의미한다. 실행을 하기 위한 코드는 흐름을 바꾸는 코드, 메모리를 읽고 쓰는 코드, 계산하는 코드로 나눌 수 있다. 예를 들어, 함수를 통해 순차적으로 진행하던 코드를 특정 위치의 코드에서 시작하도록 하는 것이 코드의 흐름을 바꾸는 코드라고 할 수 있다. 메모리를 읽고 쓰는 코드는 특정 값을 앞서 저장공간을 만든 곳에 넣고, 이를 이름과 연결하는 것을 예로 들 수 있다. 계산하는 코드는 특정 규칙에 따라 데이터를 변경하는 코드로, $a+b$ 를 예로 들 수 있다.

명령하는 코드는 code segment에 저장하고, 이를 실행하기 위해 data, heap, stack segment에 저장되어 있는 데이터를 이용한다. Heap segment는 동적할당으로 저장공간을 할당한 경우 즉, 객체가

저장된다. Data segment는 전역변수 및 static변수 등이 저장되며, stack segment는 지역변수, 매개변수, 리턴 값, 함수 호출 등과 관련된 정보가 저장된다.

B. 폰노이만 컴퓨터 구조 (“von neumann architecture”[참고문헌 1번-인용]))

폰노이만 컴퓨터 구조는 CPU가 code segment와 data,heap,stack segment를 이용해 계산, 흐름 컨트롤, 데이터 읽고 쓰기, I/O 디바이스 컨트롤 등의 역할을 하는 것을 의미한다. 계산은 특정 규칙에 맞게 데이터를 바꾸는 것을 의미하고, 코드 흐름 컨트롤은 어떤 순서로 프로그램을 실행하는지, 데이터 읽고 쓰기는 데이터가 저장되어 있는 메모리에서 데이터를 읽고 쓰는 것을 의미한다. 마지막으로 I/O 디바이스 컨트롤은 메모리의 확장구조로, 메모리 바깥에 있는 I/O 디바이스를 통해 입력/출력 받고 버퍼에 저장하는 것을 의미한다.

2. Memory Allocation

A. Object Type

객체의 이름을 만들어 놓으면, 포인터를 통해 주소를 할당하는데, 예를 들면, new Student()는 heap 메모리에 16 byte의 공간을 놓고, 그 공간에 Student 형태로 할당한다는 것을 의미한다.

B. Primitive Type

Primitive Type의 경우는, 객체의 구조를 가지고 있지 않고, 자바가 가지고 있는 것이다. 따라서 위의 경우와 달리 class가 없어도 만들 수 있다. 예를들면 int, float, boolean이 있다.

참고문헌

1)“폰노이만 컴퓨터 구조”, 나무위키, 인용 - 1번째 줄,

<https://namu.wiki/w/%ED%8F%B0%EB%85%B8%EC%9D%B4%EB%A7%8C%20%EA%B5%AC%EC%A1%B0>