60200307 우연정

목차

1. Naming Service

```
class Student {
        int id;
        int name;
        int address;
        void f1(int x) { … }
}

Int main() {
        Student student;
        student = new Student();
        sudent.id = 1;
        student.name = 5;
        student.f1(5);
}
```

Naming Service는 이름을 통해서 객체를 찾아 접근할 수 있도록 도와주는 서비스이다. 이 서비스는 분산 시스템에서 주로 사용되며, 객체의 위치나 주소를 쉽게 찾을 수 있게 한다.

프로그램에서 객체나 변수를 선언할 때, 그 객체나 변수는 메모리 상의 특정 주소를 참조한다. 예를 들어, Student student; 라는 코드는 student라는 이름의 객체를 메모리에 선언하는 것이다.

프로그램의 메모리는 크게 Data, Stack, Heap 세 가지 영역으로 나눠진다. Data Segment는 초기화된 전역 변수와 정적 변수가 저장되는 영역이다. Stack Segment는 함수와 지역 변수가 저장되는 영역이다. Heap Segment는 동적 메모리 할당을 통해 생성된 객체나 변수가 저장되는 영역이다. new Student()를 호출할 때, Student 객체는 힙 영역에 할당된다.

1-1. Source Code Level
소스 코드 레벨은 사용자가 작성하는 코드이다. 위의 Student 클래스와 main 함수가 이에 해당한다.

1-2. Executable Code Level
소스 코드가 컴파일러를 통해 변환된 실행 가능한 바이너리 코드의 수준이다.

1-3. Process Level
실행 중인 프로그램은 OS에 의해 프로세스로 관리된다. 각 프로세스는 독립적인 메모리 영역을 가진다.

코드가 컴파일 되는 과정은 소스 코드 레벨, 파일 시스템, Executable Code Level, 메모리, CPU, Process Level로 나눌 수 있다.

1. Source Code Level: Student 클래스와 main 함수가 소스 코드로 작성된다.
2. File System: 이 소스 코드는 파일 시스템에 저장된다.
3. Executable Code Level: 컴파일러는 소스 코드를 읽어들여 바이너리 코드로 변환한다.
4. Memory: 프로그램이 실행되면, 메모리의 Stack 영역에 student 객체가 할당되고, Heap 영역에 new Student()로 생성된 객체가 할당된다.
5. CPU: main 함수 내의 명령어들이 CPU에서 순서대로 실행된다.
6. Process Level: 프로그램 실행 도중, 해당 프로그램은 OS에 의해 프로세스로 관리된다.