

1. 컴퓨터

1-1. 하드웨어

1-2. OS (Operating System)

1-3. 자바 버추얼 머신 (jvm)

1. 컴퓨터

컴퓨터는 명령어를 실행한다. 컴퓨터는 계산기라고 정의하지만 계산을 담당하는 주체는 cpu이다. cpu가 우리가 만든 프로그램의 명령어를 실행한다. 계산은 수학의 산수 영역을 말한다. 수와 연산자(오퍼레이터) 연산자는 수를 논리적인 규칙에 의해 수를 다른 숫자로 변형(매핑)하는 것이다. 컴퓨터는 여러 개의 계산 규칙을 가지고 있고, 그 흐름을 바꿀 수 있는 명령어를 가지고 있다. 선택, 반복의 개념. 단순 계산이 아니라 계산으로 의사결정을 할 수 있는 명령어가 포함되어 있다.

1-1. 하드웨어

cpu는 계산도 하지만 제어도 한다. 제어 유닛은 프로그램의 흐름을 바꿀 수 있다. 프로그램은 cpu에 명령을 준다. cpu는 다른 것과는 직접 소통하지 못하고 메모리를 통해서만 소통할 수 있다. 메모리 위에 프로그램이 있어야 cpu가 실행할 수 있다. 메모리의 옆에는 직접적, 지속적 저장장치(비휘발성 저장장치)인 하드디스크, 파일 등이 있다. 파일시스템이 저장장치를 관리하고 있다. (소프트웨어, OS의 일부)

파일은 저장된 독립적 단위이다. 파일은 크게 두 종류가 있다. 하나는 프로그램 파일, 하나는 데이터 파일이다. 프로그램은 cpu 구조의 집합이고 실행되는 것이다. 프로그램은 실행되는 단위이고, cpu가 프로그램을 실행한다.

데이터는 프로그램을 실행한 후 만들어지는 결과이다. 예를 들어 파워포인트 프로그램으로 만든 발표 자료는 데이터 파일이다. 프로그램 파일과는 다르다. 데이터 파일에 데이터를 만든 프로그램을 연결하고, 파워포인트가 실행되어 내 데이터 파일을 읽어온다. 데이터를 만든 프로그램을 알 수 없으면 열 수 없다.

우리가 다루는 것은 프로그램 파일이다. 프로그램은 기계어로 번역되어 실행 가능한 형태로 하드디스크에 저장되어있다. 하드디스크에서 메모리에 프로그램이 올라왔다는 것은 cpu가 실행하고 있음을 의미한다. 이러한 과정은 프로세스라고 한다. 프로세스는 실행중인 프로그램이다. 프로세스의 명령어를 cpu가 실행하면서 프로그램을 할 수 있다.

ctrl + alt + delete 키를 누르면 현재의 프로세스를 띄워준다. 여기서 cpu의 점유율을 보여준다. cpu가 짧은 시간 동안 오고 가며 계속 프로그램을 실행하고 있다. 매우 빨라서 동시에 하고 있는 것처럼 보인다. cpu가 메모리의 프로그램을 실행한 후에 꺼지면 하드디스크로 다시 돌아간다. 메모리는 휘발성이다. 실행할 때만 올라오고 지속적이지 않다. 오늘날 저장장치의 경계가 무너지고 있다. 전원을 꺼도 메모리가 그대로 남아있다. 전형적인 대용량 컴퓨터와

는 다르다. (스마트폰 등 소형기기와는 관계가 없다.)

“컴퓨터 하드웨어 아키텍처는 CPU, Memory, IO Device 와 이것들을 연결해주는 ‘시스템 버스’로 구성된다.”[네이버 지식백과] 시스템 버스에는 데이터 버스, 주소 버스, 제어 버스가 있다. “주소 버스에는 중앙처리장치가 주기억장치로부터 읽어야 할 데이터가 저장된 주기억 장치의 주소나 데이터를 쓸 주기억장치의 주소가 전달된다.”[네이버 지식백과] 데이터 버스로는 주기억장치로부터 중앙처리장치에 데이터가 전달된다. 제어 버스로는 ‘어느 위치에 무엇을 하라’는 정보가 전달된다.

1-2. OS (Operating System)

OS는 프로그램을 실행하는데 필요한 제반을 제공해준다. OS는 유저 프로세스의 실행을 관리해준다. 간단한 하드웨어 시스템은 OS가 필요하지 않은 경우가 있다.

여러 명의 유저가 들어가는 순간 하드웨어는 제대로 쓸 수 없고, 이를 관리해주는 관리자가 필요한데, 그 역할을 하는 것이 OS이다. OS는 리소스 매니저라고도 한다. 리소스는 하드웨어 디바이스이다. 하드웨어 디바이스를 쓸 수 있게 해주는 유틸리티가 있다. 대표적으로 프로세스 매니저가 있다. 멀티 태스킹, 멀티 유저가 개입되는 순간 한정적인 자원(컴퓨터)을 관리하는 관리자가 있어야 한다. 운영체제는 여러 명의 사용자와 프로세스, 데이터를 동시에 실행시킬 수 있게 해주는 관리자이다.

1-3. 자바 버추얼 머신

유저 프로세스는 프로그램이 실행되는 프로세스의 환경을 말한다. 유저 프로세스가 OS에 실행 명령을 내린다. OS가 유저 프로세스 실행(하드웨어)에 관여한다. 어떤 명령어들은 CPU 명령어를 직접 전달하기도 한다. 유저 프로세스를 만들 때 명령어가 같을 수 없다. 같은 프로그램이지만 명령어가 다르기 때문에 따로 프로그램을 짜야 한다. 그래서 하나로 짜서 사용하는 방법을 고민했고, 플랫폼의 독립성을 유지하기 위해서 번역을 할 수 있는 기능을 달아 중립적 언어를 만들었다. 그것이 바로 ‘자바 버추얼 머신’이다. 나에게 일방적으로 서비스를 제공하면 플랫폼이고(한 방향), 버추얼 머신도 일종의 플랫폼이라고 할 수 있다.

버추얼 머신을 미리 깔아놓고 운영체제에 따라 번역하는 방식으로 작동된다. ‘플랫폼 독립성’을 유지하기 위해서 자바가 있고, 자바가 자기 자신의 기계어를 만들었다. 이는 바이트 코드이다. 자바 버추얼 머신의 기계어는 어디서든 똑같다. 즉 자바로 프로그램 하나를 짜면 어디에서든 돌아간다.

jdk를 깔면 그 안에 자바 버추얼 머신(jvm)이 있는 것을 확인할 수 있다.

유저 프로세스는 기계어로 만들어져 있다. windows pc, mac, 자바 버추얼 머신 용으로 만들어서 사용한다. 자바 버추얼 머신은 전기 신호로 구성되어 있다. 컴퓨터가 한글이나 알파벳을 알고 있는 이유는 세계 각국의 언어를 고유 숫자로 대응시켜놓았기 때문이다. 몇백만 개의 언어를 만들기 위해 자릿수를 늘렸다. 컴퓨터가 점점 감성적으로 변하는 것이 아니라 경우의

수를 더 많이 처리하게 될 수 있게 된 것이다.

코드로 프로그램을 짜면 0101로 바꾼다(전기신호). 전기의 단계가 모호하면 안되기 때문에 0과 1로 만들어 놓은 것이다. 유저 프로그램이 유저 프로세스를 만들어 낼 때 컴파일러나 편집기가 개입을 한다. (IDE) . 소스코드를 만들 때, 우리가 한글 등 언어를 입력하면 이미 그려놓은 폰트를 메모리에서 읽어서 모니터에 보여주는 것이다.

컴파일러는 윈도우에서는 실행 프로그램을 윈도우 기계어로 바꾼다. “FORTRAN, COBOL, ALGOL 등의 컴파일러 언어로 쓰여진 소스 프로그램을 번역해서 각각 목적의 기계가 해독할 수 있는 기계어로 고치기 위한 프로그램이다. 따라서 컴파일러는 각 기계에 고유한 것이 개발되어 메이커에서 사용자에게 공급되는 것이 보통이다.”[네이버 지식백과] 컴파일러 [compiler] (컴퓨터인터넷IT용어대사전, 2011. 1. 20., 전산용어사전편찬위원회)