프로그래밍 기초

컴퓨터의 구조와 프로그래밍

프로그램이란?

- 프로그램은 명령어의 집합
- 개발자가 제시한 순서에 따라 컴퓨터(cpu)가 실행해야 할 명령어 목록

- 명령어
  - 컴퓨터에게 시킬 모든 작업에 해당하는 키워드
  - 입/출력, 계산(연산)

- 프로그래밍 언어란 컴퓨터(기계)가 알 수 있는 키워드를 사용하여 컴퓨터에게 작업을 지시하기 위해 사용하는 언어
  - 컴퓨터가 알 수 있는 언어를 사용해야 함 -> 기계어(비트)
  - 기계어는 2진법을 사용(0과 1만 사용)

컴퓨터의 역사

- 최초의 디지털 컴퓨터 콜로서스(영국)
  - 1943~1945년
  - 2차 세계대전 주에 독일의 암호문 해독을 위해 개발
  - 1970년 대까지 1급 비밀로 공개되지 않음
  - 스위치와 플러그를 사용하여 프로그래밍 및 작동

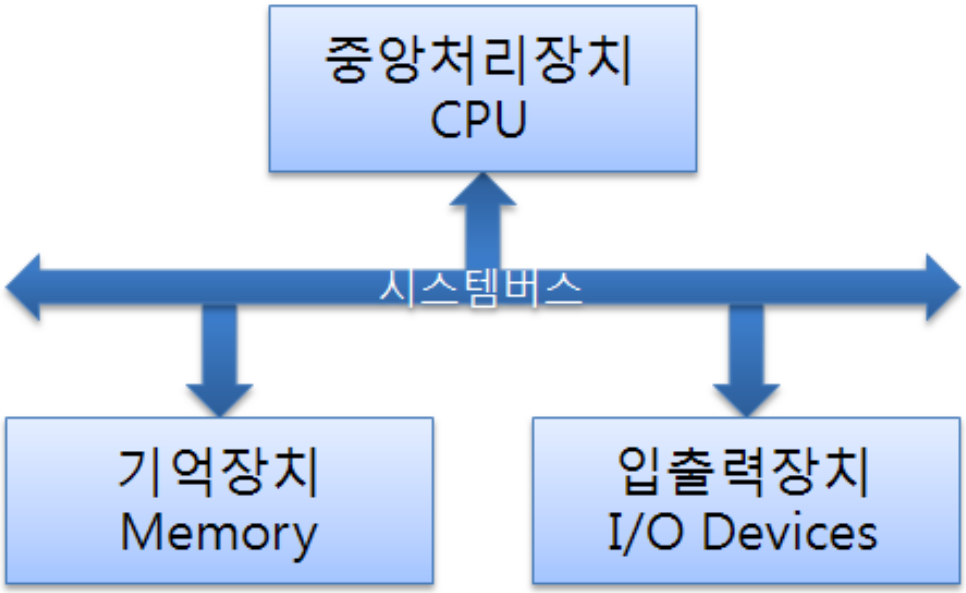
- 세간에 알려진 최초의 디지털 컴퓨터 에니악(미국)
  - 1946년
  - 탄도 계산을 위해 개발
  - 배선판에 일일이 배선하는 방식(프로그램 변경은 배선판 교체)

- 두 컴퓨터의 공통점
  - 진공관(전구)을 사용 -> 2진법을 활용

컴퓨터의 물리적 구조

- 폰 노이만 구조
  - 존 폰 노이만 - 헝가리 출신으로 미국에서 활동한 수학자
  - 프로그램을 기억장치에 내장하는 방식의 컴퓨터를 제안

- 기본 3 요소



컴퓨터의 물리적 구조

- CPU
  - 중앙처리 장치로써 연산을 담당한다.
- Memory
  - RAM이라는 저장장치로 구성되는 메인 메모리는 컴파일이 완료된 프로그램 코드가 올라가서 실행되는 영역이다.
  - 프로그램 실행을 위해 존재하는 메모리
- System Bus
  - 컴퓨터를 구성하는 구성요소 사이에서 데이터를 주고 받기 위해 사용되는 경로
  - Address Bus, Data Bus, Control Bus 세가지로 구분된다.
  - 하드디스크, CPU, 메인 메모리등 모두가 버스에 연결되어있어 전송, 입출력을 가능하게 한다.
- Input/Output Devices
  - 기본 입력(키보드), 출력 장치(모니터)

메모리

- 폰 노이만 구조를 사용한 모든 컴퓨터(즉, 현존하는 모든 컴퓨터)에서 실행되는 모든 프로그램은 메모리(기억장치)에 내장되어 실행된다!
  - (프로그램을 구성하는) 모든 명령어는 메모리에 저장됨.
  - 입력 장치를 통해 입력된 데이터는 모두 메모리로 들어감.
  - 메모리에 저장된 모든 명령어와 데이터를 CPU로 전달됨.
  - CPU가 처리한 모든 결과는 다시 메모리에 저장됨.
  - 메모리에 저장된 결과는 출력 장치를 통해 출력됨.

- 프로그래밍 언어에서 '변수'는 프로그램에서 사용하는 데이터를 저장하는 메모리 공간을 말함.
  - 데이터를 처리하는 모든 명령은 반드시 변수가 필요함.
  - 입력이란 사용자가 컴퓨터에게 데이터를 주는 행위
  - 출력이란 컴퓨터가 사용자에게 데이터를 주는 행위

자료(Data)와 정보(Information)

- Data
  - 현실 세계에서 벌어지는 일들을 단순한 방법으로 관찰하고 측정해서 얻은 값
  - 덧셈에서 더할 두 수
  - 한 사람의 이름, 키, 몸무게, 나이 등은 각각의 데이터

- Information
  - 데이터를 처리해서 얻을 수 있는 결과
  - 덧셈에서 더해진 두 수의 합
  - 이름, 키, 몸무게, 나이를 모두 묶으면 한 사람의 정보
  - 정보는 어떤 규칙에 따라 모으거나 추출되거나 생성된 데이터, 데이터 집합
  - 정보를 데이터로 하여 새로운 정보를 생성할 수도 있음.

자료(Data)와 정보(Information)

- 컴퓨터는 사람이 정보를 얻을 수 있도록 도와주는 시스템
- 프로그램은 컴퓨터가 사람이 정보를 얻을 수 있도록 돕기 위해 사용하는 도구.
  - 프로그램은 특정 정보를 얻기 위한 목적이 반영된 도구.
  - 즉, 프로그램은 사용하기 위한 목적이 존재.

- 프로그램은 기본적으로 자료를 취급하여 정보를 생성.
  - 사람은 프로그램에 자료를 입력하여 해당 결과인 정보를 획득.
  - 컴퓨터 시스템의 구조에 의해 프로그램이 동작하기 위해 입력 받은 자료를 저장하는 메모리 공간과 정보 출력을 위한 메모리 공간이 요구됨.
  - 이 공간을 변수라고 함.

Data의 처리

- 메모리(기억 장치)는 비어있는 공간
  - 계산을 할 때 공책에 숫자를 쓰듯이 컴퓨터의 메모리에 데이터를 쓴다.
  - 공책 = 메모리
- 공간의 크기는 유한함.
  - 최근에는 약 4~8GB를 많이 사용.
  - 확장은 가능하지만 무한히 확장할 수 없음(물리적 한계)
- 이러한 공간의 관리를 위해 동일한 크기로 분할하여 사용.
  - 이 크기를 데이터의 최소 단위라고 함.
  - 이 공간의 관리는 운영체제가 담당.(토지관리 공사?)
  - 모든 공간은 공유지이며, 독점하여 사용할 수 없음.
  - 모든 공간에 일련번호를 지정하여 관리.

Data의 처리

- 메모리(기억 공간)를 사용하는 방법
  - 데이터를 저장하는 데 필요한 만큼 운영체제에 요청하고 할당 받아서 사용한 후 다시 반납.
  - 즉, 프로그램이 실행되면 운영체제는 프로그램이 필요한 만큼 할당해 주고 프로그램 종료 시 다시 수거함.
  - 공간 계산을 명확하기 하기 위해 모든 공간을 동일하게 분할

- 공간의 크기 = 단위
  - 컴퓨터의 최소 단위 – 1 bit(비트)
  - 데이터의 최소 단위 – 1 byte(바이트)
  - 1 Byte는 8 bit로 구성.
    - 즉 메모리는 8개의 bit 열을 방 한 칸처럼 취급

프로그램의 실행

- (사람의 말로..)명령어 작성(코딩)
- 사람의 명령을 실행하기 전 사전 작업 처리(전처리)
  - 자바는 전처리 기능을 지원하지 않음.
- 전처리된 모든 명령을 기계어로 번역(컴파일)
- 프로그램 실행에 필요한 (함께 작성하지 않은) 다른 기계어 명령과 지금 컴파일된 기계어 명령을 연결(링크)
- 모든 명령어 실행

프로그램의 실행

- 프로그래밍 언어
  - 진공관(전구)에서 시작한 디지털 컴퓨터는 사용할 수 있는 신호(또는 데이터)가 0과 1 뿐임.
  - 이것은 바이너리(Binary)라고 함.
  - 1 bit란 0 또는 1을 저장하는 한 공간.
  - 1 bit로는 on/off 명령 밖에 줄 수 없음.
  - 여러 개의 bit를 묶어서 하나의 의미를 부여 -> 기계어
  - 8개의 bit를 묶어서 사용 -> 1 byte

- 사람이 bit열로 구성된 기계어를 사용하기는 어려움.
  - 사람이 사용하는 언어(영단어)를 사용하여 명령어를 작성.
  - 컴퓨터가 번역을 수행하여 프로그램으로 완성.
  - 이 과정을 컴파일(Compile)이라고 함.

프로그램의 실행

- 자바의 컴파일 과정

- 컴파일 – 알파벳을 바이트 코드로 변환(번역)
- 인터프리터 – 바이트 코드를 기계어로 변환(통역)
  - 바이트 코드(8bit)
  - 기계어(1bit) – 비트열

OOP(Object Oriented Programming)

- 객체 지향 프로그래밍
  - 프로그램의 모든 part(부분, 부품)을 객체(Object)로 취급하여 개발하는 방식

- 객체
  - 형상화된 모든 사물(모든 것은 객체다!)
  - 하나의 독립된 완성체이면서 동시에 다른 객체와 관계를 형성하여 더 큰 객체를 구성하는 부품
  - 예) 학생은 하나의 객체이면서 한 반(객체)을 구성하는 구성원

OOP(Object Oriented Programming)

- 객체
  - 객체는 상태(특성, 성질, 고유 값 등)와 행동(기능, 방법)으로 구성
  - 객체는 그대로 사용될 수도 있고, 또는 조금 변형되어 사용될 수도 있음.
    - 유리를 잘라서 다양한 크기로 사용할 수 있지만, 유리의 기본 성질인 빛만 통과한다는 것은 변하지 않음
  - 객체는 변하지 않는 부분과 변경할 수 있는 부분으로 구성.
    - 빛만 통과시킨다는 목적으로 유리를 사용하지만 모양과 크기는 변경할 수 있음.

- 객체의 프로그래밍
  - 상태는 저장할 데이터.
  - 행동은 명령어의 집합.