



코딩을 배우기 전에 읽는 컴퓨터 구조론

파이썬 프로그래밍

Contents

❖ 폰 노이만 구조

- 중앙처리장치
- 기억장치
- 입력과 출력

❖ 운영체제와 애플리케이션

❖ 소프트웨어는 무엇으로 만드는가

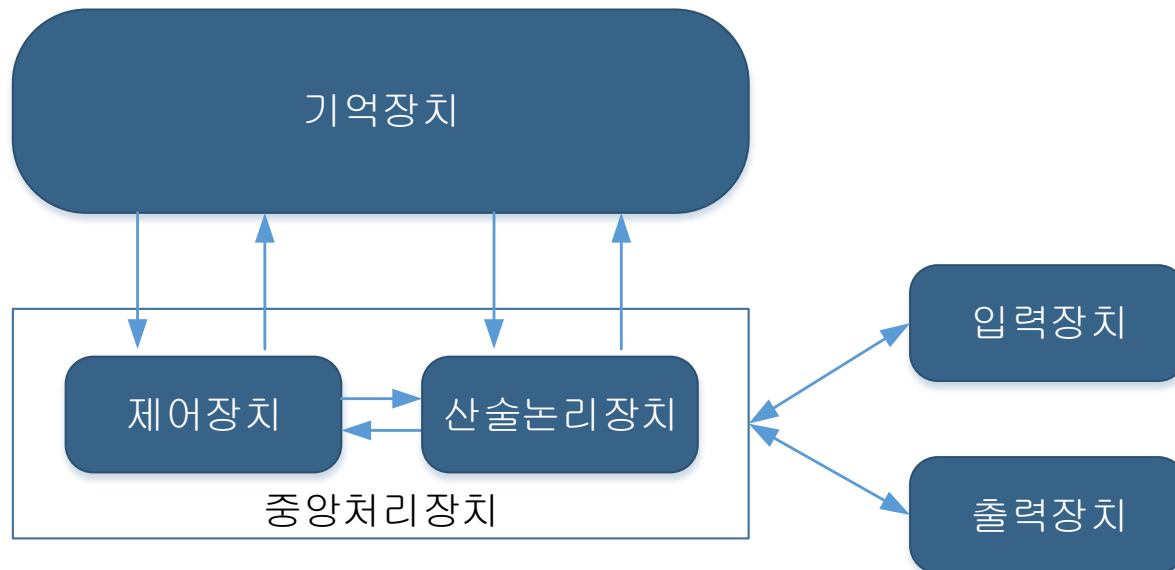
❖ 파이썬 프로그래밍 언어



폰 노이만 구조

❖ 오늘날의 컴퓨터들은 그 모습과 크기, 성능은 다를지 모르지만 기본적인 구조는 거의 똑같음.

- 1945년에 존 폰 노이만이 발표한 논문 <EDVAC에 관한 보고서>에서 기술한 컴퓨터의 구조를 그대로 따르고 있기 때문



폰 노이만 구조

- ❖ **EDVAC은 프로그램을 수정하는 일이 ENIAC에 비해 자유로웠음.**
 - EDVAC은 명령어를 기억장치에 내장하고 있었기 때문임.
 - 이렇게 기억장치에 명령어를 내장하는 컴퓨터를 **내장식 컴퓨터(Stored-program computer)**라고 함.
 - 내장식 컴퓨터는 폰 노이만의 논문에서 본격적으로 소개되었기 때문에 **폰 노이만 구조(von Neumann Architecture)**라는 이름으로 알려져 있음
- ❖ **PC의 폰노이만 구조**
 - 입력 장치 : 키보드와 마우스
 - 출력 장치 : 모니터와 프린터
 - 중앙처리장치 : 인텔 i7
 - 기억장치 : DDR3 DRAM



폰 노이만 구조 - 중앙처리장치

❖ 중앙처리장치

- 컴퓨터가 수행하는 계산은 모두 이 CPU를 통해 이루어짐.
(최근에는 그래픽 카드의 처리장치를 활용하기도 함.)
- CPU를 이루는 요소
 - 산술 논리 장치(ALU : Arithmetic Logic Unit)
 - 제어 장치(CU : Control Unit)

❖ 산술논리장치

- 덧셈, 뺄셈, 곱셈, 나눗셈 등의 산술 연산과 참과 거짓(또는 0과 1)을 다루는 논리 연산을 수행하는 회로를 가짐.

❖ 제어장치

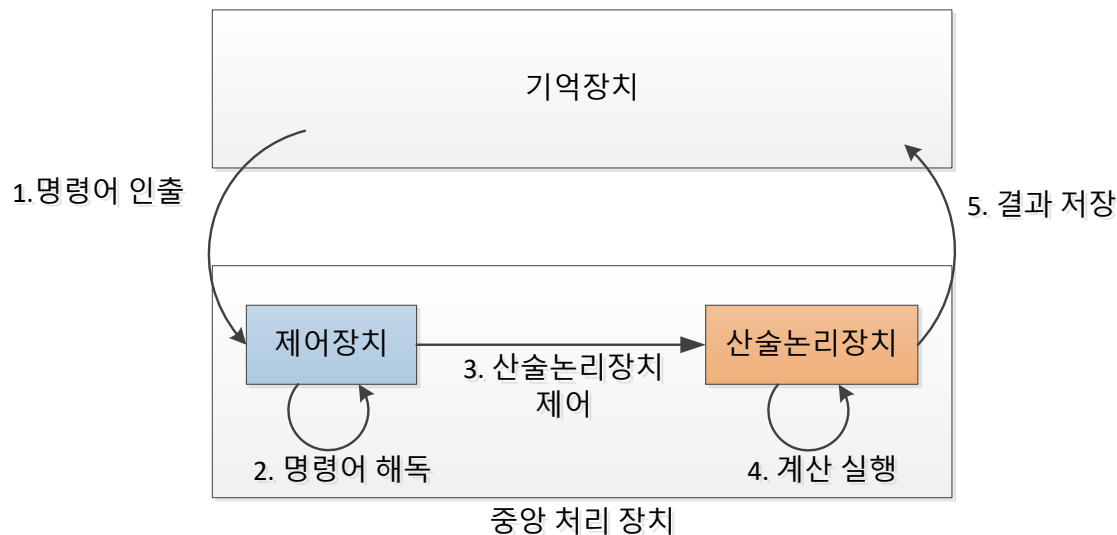
- 산술논리장치를 통제하여 계산을 수행함.



폰 노이만 구조 - 중앙처리장치

❖ 제어장치와 산술논리장치의 동작 과정

1. 먼저 제어장치가 기억장치로부터 명령어를 가져옵니다(Fetch).
2. 제어장치는 가져온 명령어를 해독(Decode)합니다.
3. 제어장치는 해독한 명령어에 따라 산술논리장치에 데이터를 옮기고 어떤 연산을 수행할지를 지시합니다.
4. 산술논리장치는 제어장치가 지시한 대로 계산을 수행(Execute)합니다.
5. 그리고 수행한 결과를 기억장치에 다시 저장(Store)합니다.



폰 노이만 구조 - 중앙처리장치

❖ 명령 주기(Instruction Cycle)

- 기억장치로부터 명령어를 불러오고 해독하며, 실행하는 주기
- CPU는 클럭(Clock)이라 부르는 시계를 갖고 있어서 이 클럭이 움직일 때마다 신호를 보내 명령 주기를 반복시킴.
- CPU가 명령 주기를 빠르게 반복할수록 컴퓨터는 주어진 시간 내에 할 수 있는 일이 많아짐.
- Hz(헤르츠)라는 단위를 이용하여 표현.
 - CPU의 클럭 주파수가 1GHz라고 하면, 그 CPU는 1초에 10억회의 명령 주기를 수행하는 성능을 지녔다는 뜻



폰 노이만 구조 - 기억장치

- ❖ **기억장치(Memory) : 데이터와 함께 명령어를 저장하는 역할 수행.**
- ❖ **메모리에 데이터를 기록하기 위해서는 CPU가 “쓰기 요청” 을 보내고 반대로 메모리로부터 데이터를 가져오기 위해서는 “읽기 요청” 을 보냄.**
 - 메모리의 성능은 이 요청에 응답하기까지의 시간과 이러한 읽기/쓰기 요청을 연속적으로 처리하는 주기에 의해 결정
- ❖ **캐시(Cache) 메모리**
 - CPU와 가장 가까이에 있는 기억장치. 성능이 좋지만 가격이 비쌘.
- ❖ **주기억 장치**
 - 캐시 메모리 아래에 위치한 기억장치. 대개 램(RAM)으로 구성
- ❖ **보조기억장치**
 - CPU로부터 가장 멀리 떨어져 있는 기억장치. 성능이 낮지만 가격도 낮음. 하드디스크가 대세였지만, 메모리기술이 발전하면서 플래시 메모리와 SSD(Solid State Drive)가 부상.



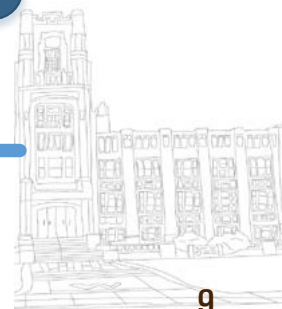
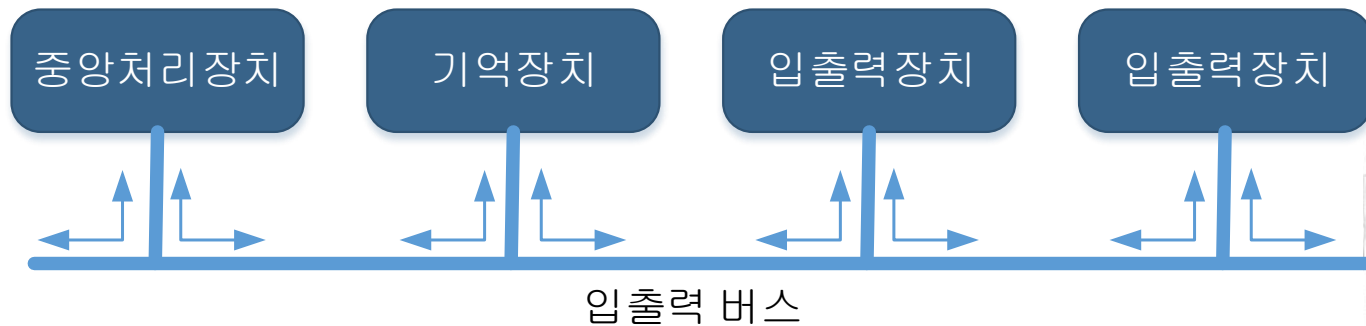
폰 노이만 구조 - 입력과 출력

❖ 입출력장치는 컴퓨터와 바깥세계와 소통하는 수단.

- 입력장치의 예)
 - 마우스, 키보드, 터치스크린, 마이크, 게임패드, 동작 인식 장치
- 출력장치의 예)
 - 모니터, 프린터, 3D 프린터, 스피커, 빔 프로젝터, E-Ink 전자 종이

❖ 입출력 버스(I/O BUS)

- 버스 : 컴퓨터의 정보 전송 회로
- 중앙처리장치와 기억장치, 그리고 입출력 장치는 아래와 같이 버스로 연결되어 있음.



폰 노이만 구조 - 입력과 출력

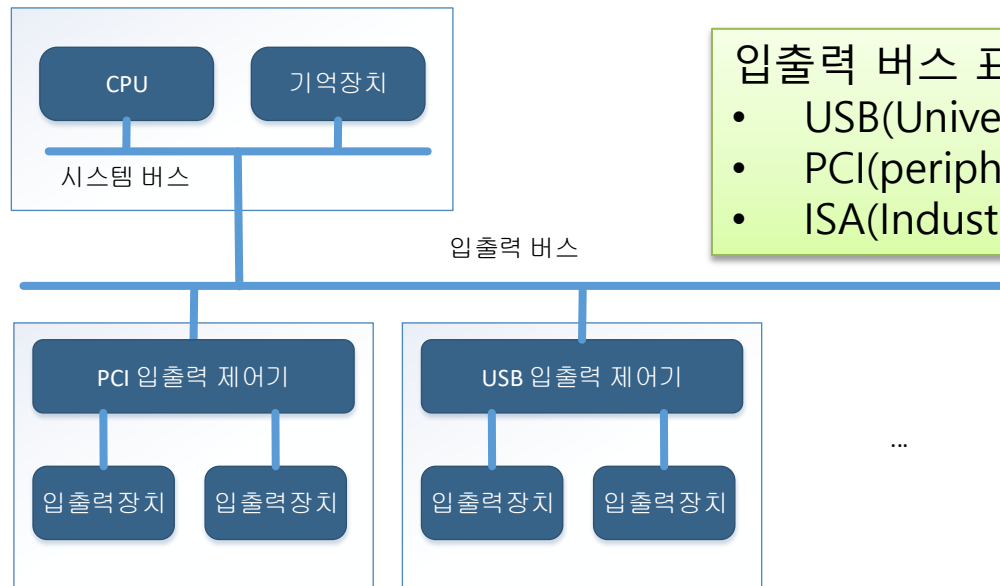
❖ 입출력 버스의 문제점

- 중앙처리장치, 기억장치, 입출력장치가 동일한 입출력버스를 사용함으로써 빠르게 동작하는 중앙처리장치가 가장 느린 입출력 장치 때문에 제 성능을 낼 수 없는 문제가 발생

→ CPU가 빠른 속도로 진화하면서 더욱 문제가 불거짐.

❖ 개선된 입출력 버스

- CPU와 기억장치는 시스템 버스(System Bus)로 묶고, 그 다음 다양한 입출력장치들은 입출력 버스로 묶어 CPU의 입출력 모듈에 연결



입출력 버스 표준 예:

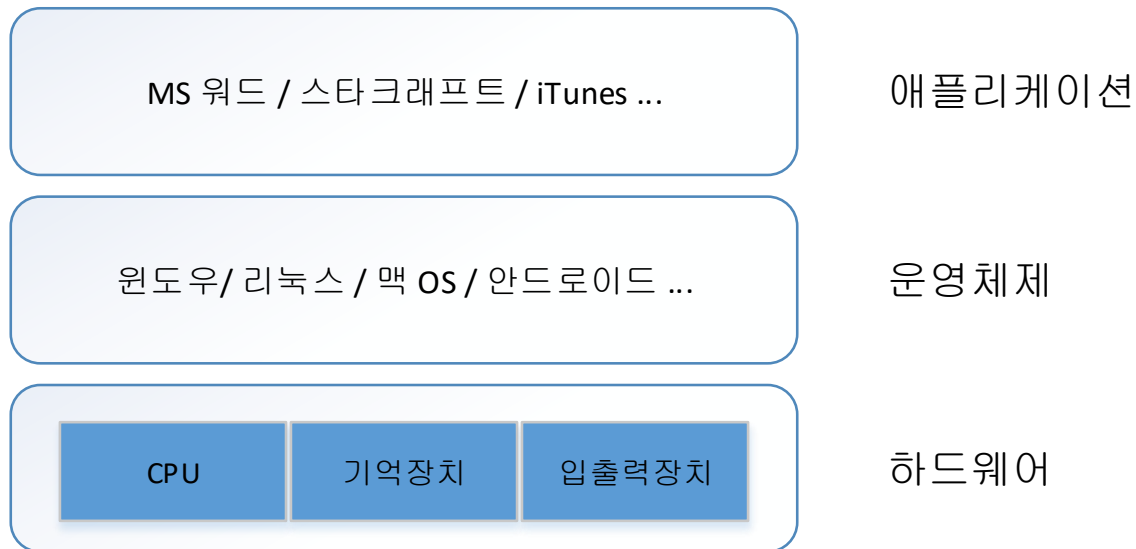
- USB(Universal Serial Bus)
- PCI(peripheral component interconnect)
- ISA(Industry Standard Architecture)



운영 체제와 애플리케이션

❖ 운영체제(Operating System)

- 애플리케이션이 다양한 하드웨어(CPU, 기억장치, 입출력장치 등등) 위에서 동작할 수 있도록 하는 시스템 소프트웨어
- 운영체제는 애플리케이션에게 API(Application Programming Interface)를 제공하여 운영체제가 제어하고 있는 하드웨어를 이용할 수 있게 함.



소프트웨어는 무엇으로 만드는가?

❖ 프로그래밍 언어(Programming Language)

- 프로그램을 작성하기 위해 만들어진 인공 언어 체계

❖ 컴파일 방식과 인터프리트 방식 프로그래밍 언어

	언어	실행 속도	이식성
컴파일 방식	C, C++, 파스칼, 에이다	CPU가 실행할 수 있는 기계 코드로 컴파일되므로 실행속도 빠름 .	원래 애플리케이션이 개발된 CPU/운영체제와 다른 CPU/운영체제용으로 옮기는 경우, 대체로 코드를 그대로 사용할 수 없음. 이식성 낮음 . ※ C#, 자바 등 가상머신 기반의 언어는 이식성 높음
인터프리트 방식	베이직, 파이썬, 루비, PHP, 펄	애플리케이션을 실행할 때마다 인터프리터가 소스 코드를 기계 코드로 번역하는 절차를 거치므로 실행속도 느림 .	인터프리터만 대상 CPU/운영체제를 지원한다면 애플리케이션 코드는 변경할 필요 없이 어떤 환경에서나 실행 가능. 이식성 높음 .



파이썬 프로그래밍 언어

- ❖ 파이썬은 1989년부터 귀도 반 로섬(Guido van Rossum)이 개발을 시작한 언어
- ❖ 만인을 위한 프로그래밍(CP4E : Computer Programming for Everybody) 언어로써 개발됨
- ❖ 읽고 쓰기 쉬운 언어
- ❖ 인터프리트 방식
- ❖ 생산성이 높고 다른 언어로 작성된 코드와 결합하는 능력이 탁월
- ❖ 다양하고 풍부한 라이브러리
- ❖ 무료





Thank You !

파이썬 프로그래밍