

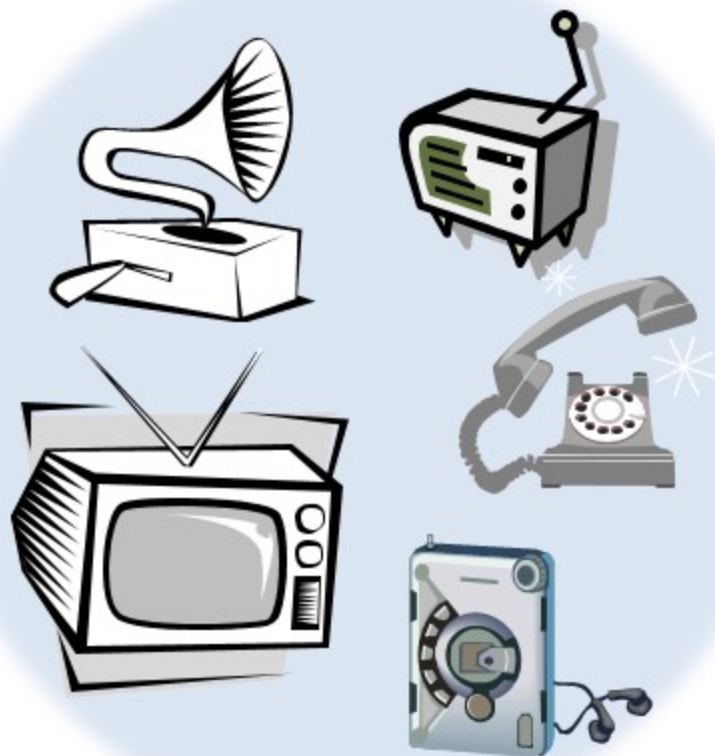
C++ 언어 강의자료

문정욱



기초이론 1: 컴퓨터 구조와 프로그래밍 언어

정보의 표현과 전달



아날로그 방식

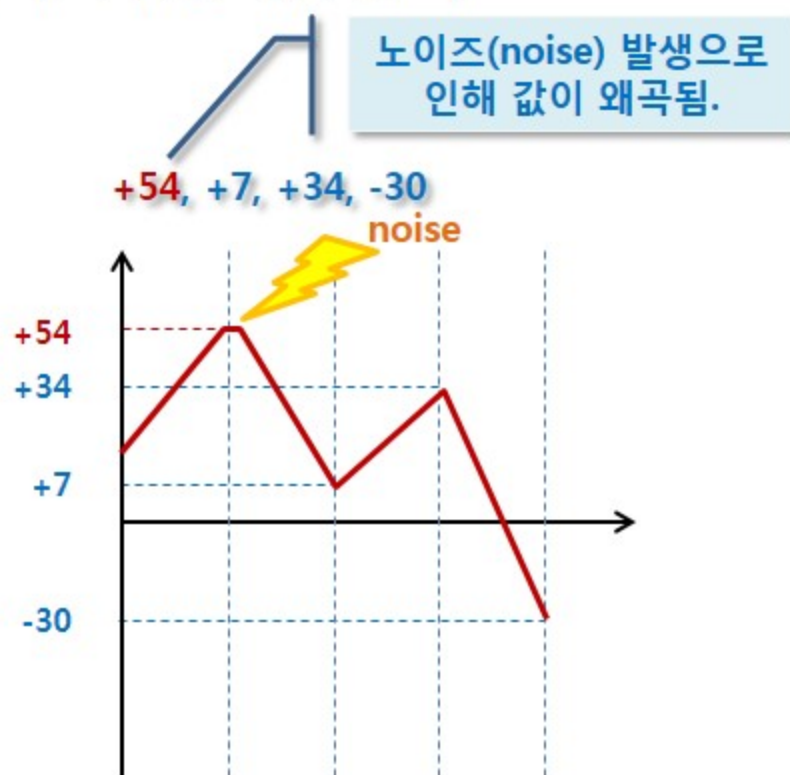
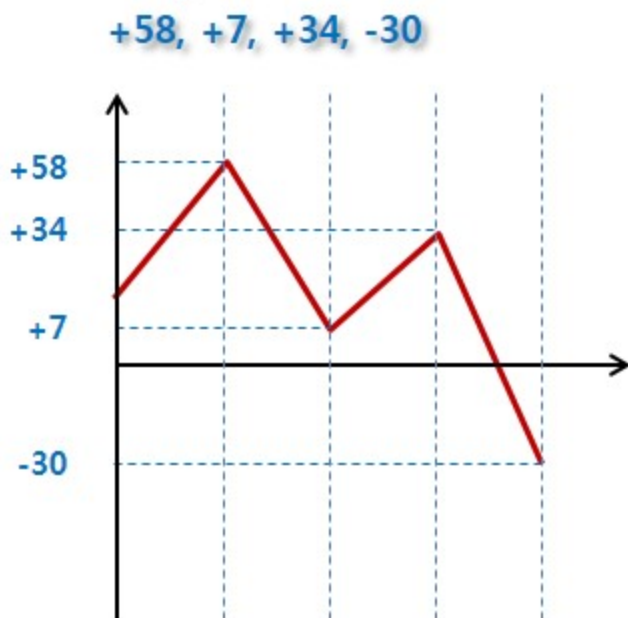


디지털 방식

정보의 표현과 전달

■ 아날로그(analog) 방식

- 전달하고자 하는 수치 값을 매체의 크기로 전달하는 방식

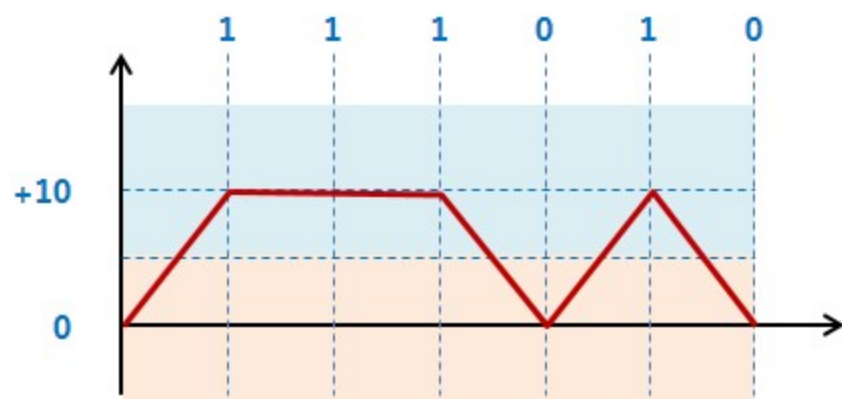


정보의 표현과 전달

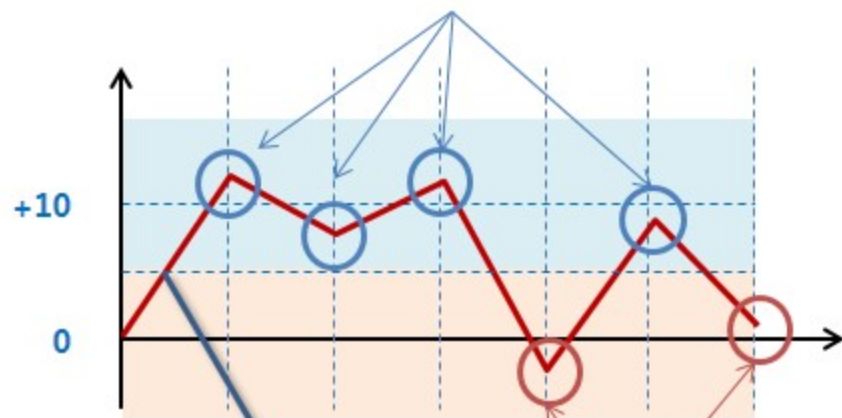
■ 디지털(digital) 방식

- 전달하고자 하는 수치 값을 이진수로 변환하여 비트 단위로 정보를 전달하는 방식

$$58 = 111010_{(2)}$$



신호의 크기를 10으로 간주
→ 1로 해석함



노이즈(noise) 발생으로
인해 값이 왜곡

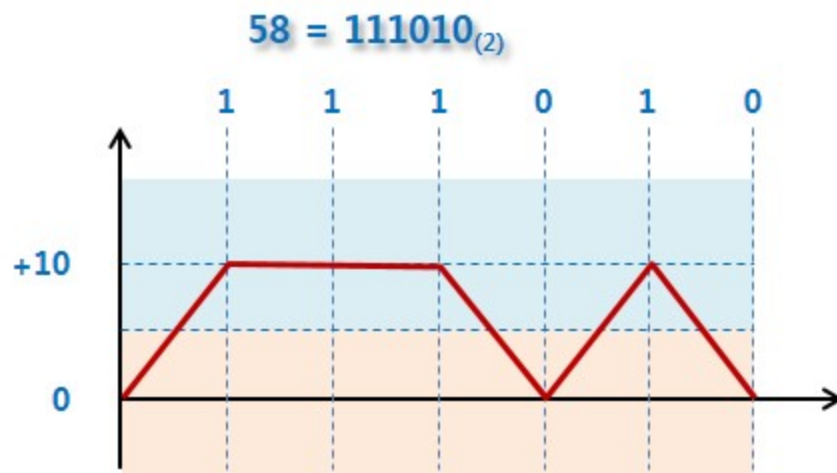
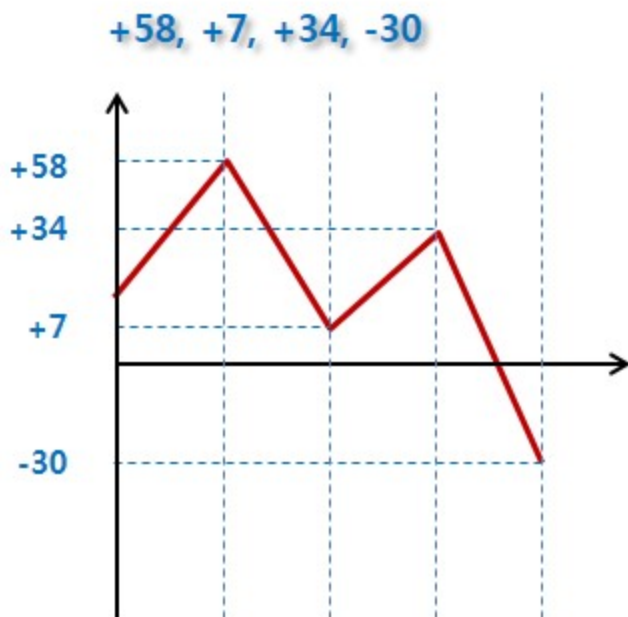
신호의 크기를 0으로 간주
→ 0으로 해석함

정보의 표현과 전달

- 디지털(digital) 방식 vs 아날로그(analogue) 방식

특징	디지털	아날로그
속도	느리다	빠르다
정확도	높다	낮다

컴퓨터가
디지털 기기인 이유



구조(architecture)의 의미

구조(architecture)란 무엇인가?

공기나 물은 구조가 있는가?



공기나 물은 구조가 없다.

자동차는 구조가 있는가?



자동차는 구조가 있다.

자동차의 모든 부품을 흩어 놓았을 때
이를 자동차라고 할 수 있는가?



부품을 흩어 놓으면
자동차라고 할 수 없다.

자동차의 모든 부품을 흩어 놓았을 때
자동차 구조가 보이는가?



부품을 흩어 놓으면
구조가 있다고 볼 수 없다.

구조(architecture)의 의미

구조(architecture)란 무엇인가?

구조를 이루기 위해서는 **부품(요소)**가 필요하다.

부품(요소)들이 제대로 **결합(관계)**되어 있어야 한다.



구조란?

구성 요소(element)를 나열하고
그 유기적인 관계(relationship)를 표현한 것

구조(architecture)의 의미

구조(architecture)란 무엇인가?

- 사람의 몸은 구조가 있는가? 만일 있다면 그 구성요소를 나열하라.

사람의 몸 = 머리 + 몸통 + 팔 + 다리

사람의 몸 = 뼈 + 근육 + 내장기관 + 혈관 + 피 + 피부

사람의 몸 = 머리 + 가슴 + 배 + 피부 + 팔

 잘못된 요소의 나열

이유: 피부는 다른 부분들과 겹친다. 다리가 빠져있다.

구조(architecture)의 의미

구조(architecture)란 무엇인가?

- 구조
 - 구성 요소(element)를 나열과 그 유기적인 관계(relationship)의 표현
- 요소를 나열할 때 주의할 점
 - 빠진 구성 요소가 있어서는 안 된다.
 - 구성요소들 간에 겹치는 부분이 있어서는 안 된다.
- 구조의 핵심 내용
 - 구성 요소들 간의 **관계(relationship)**가 구조(architecture)의 핵심적 내용이다.

컴퓨터 프로그램

- 컴퓨터 프로그램(program)의 개념
 - 컴퓨터가 수행해야 할 **명령의 목록(instruction sequence)**.
 - 명령 수행에 필요한 **자료(data)**가 포함되기도 한다.
- 컴퓨터 명령의 표현
 - 정확도가 중요, 디지털 방식 사용
 - **컴퓨터는 명령을 0과 1로 표현**한다.
 - 컴퓨터 명령을 표현하기 위해 규칙이 필요하다.

명령 표현 규칙 (ISA)

덧셈 = 010
곱셈 = 110
뺄셈 = 001
나눗셈 = 101

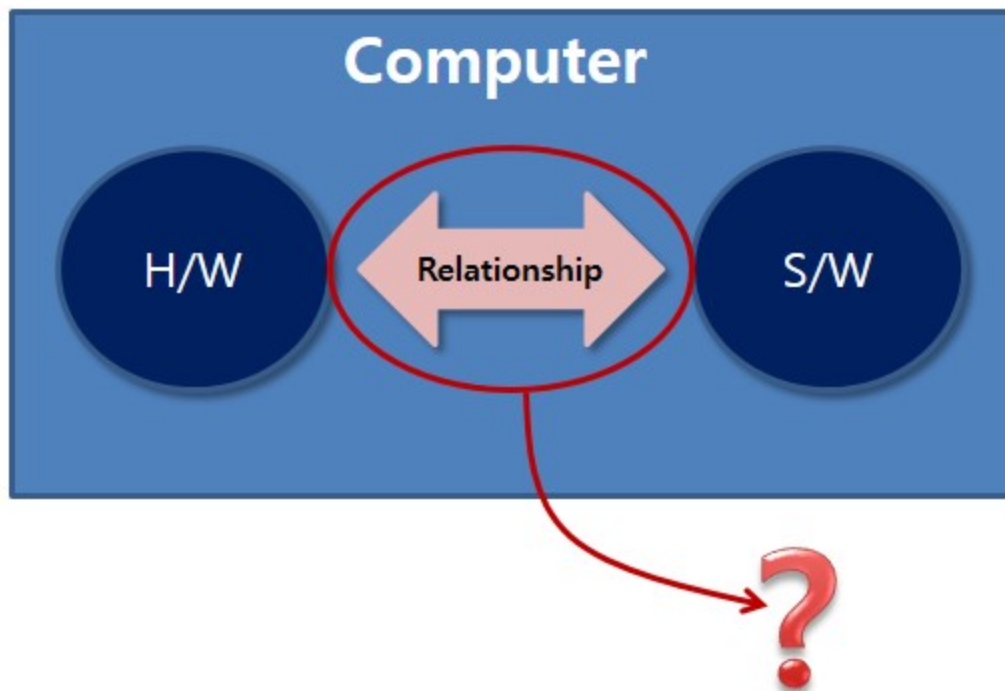
...



컴퓨터 구조

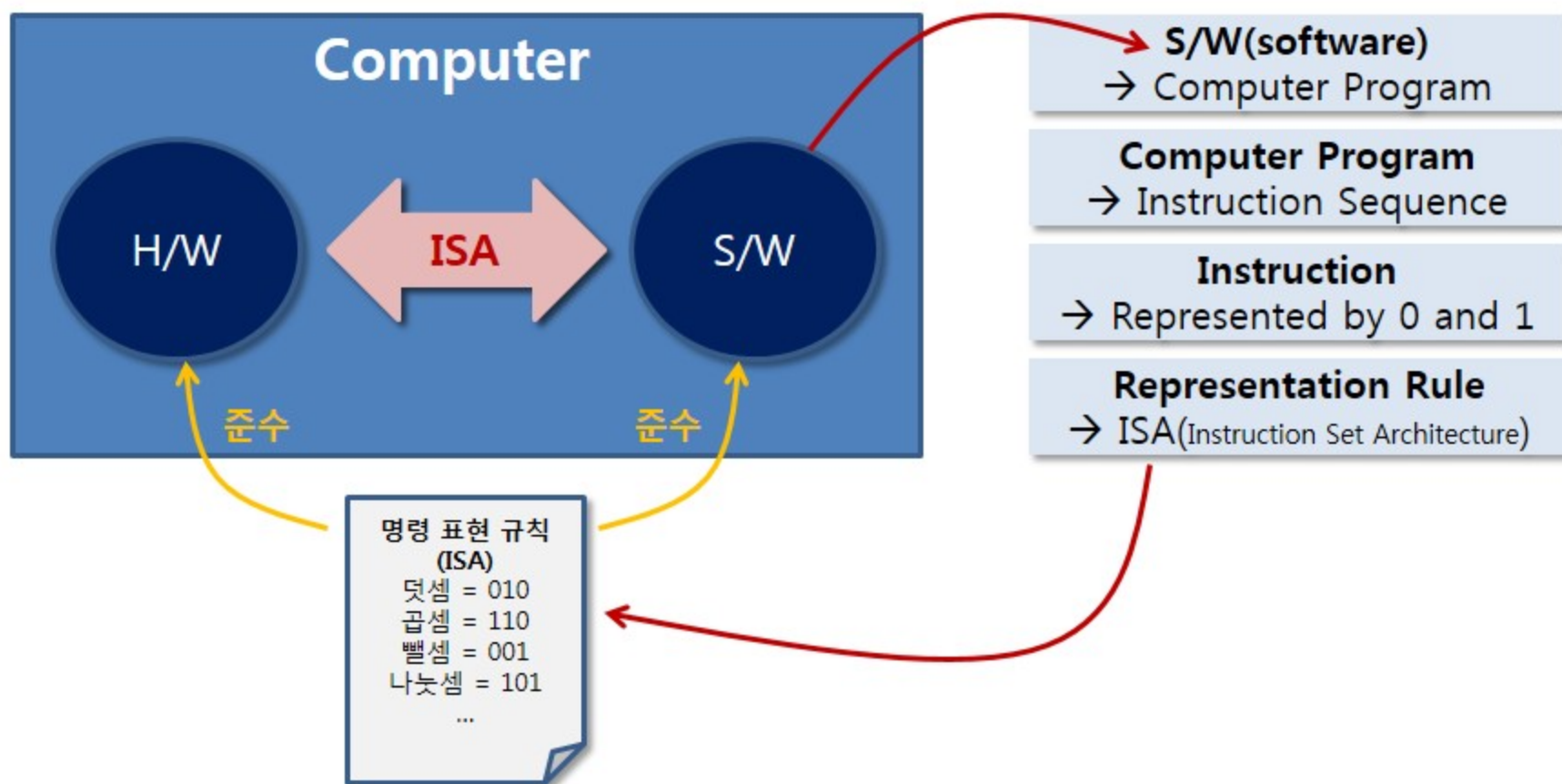
■ 컴퓨터 구조

- 컴퓨터의 **구성요소**와 그 유기적인 **관계**(결합 및 통신 규약)의 표현



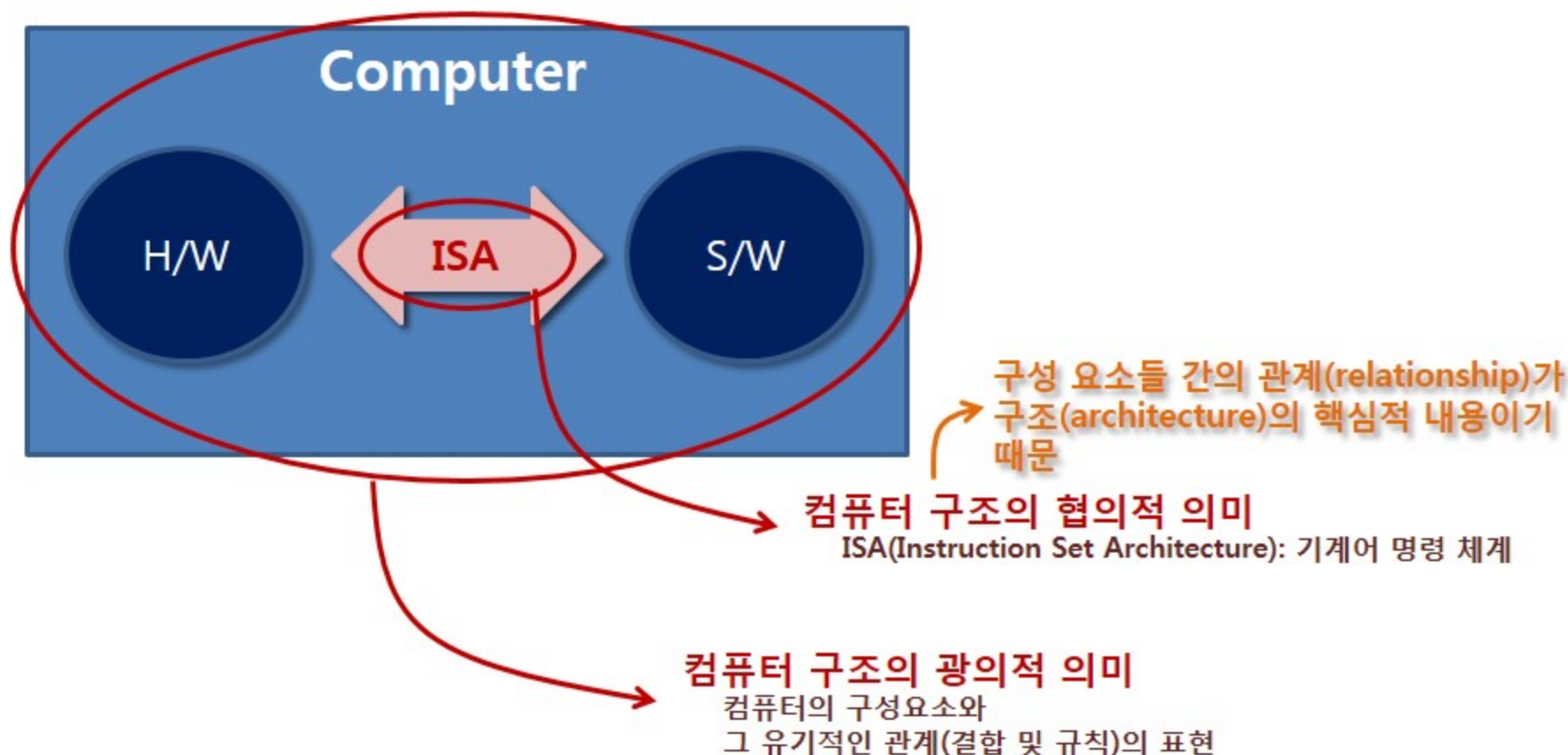
컴퓨터 구조

■ 컴퓨터 구조(Computer Architecture)



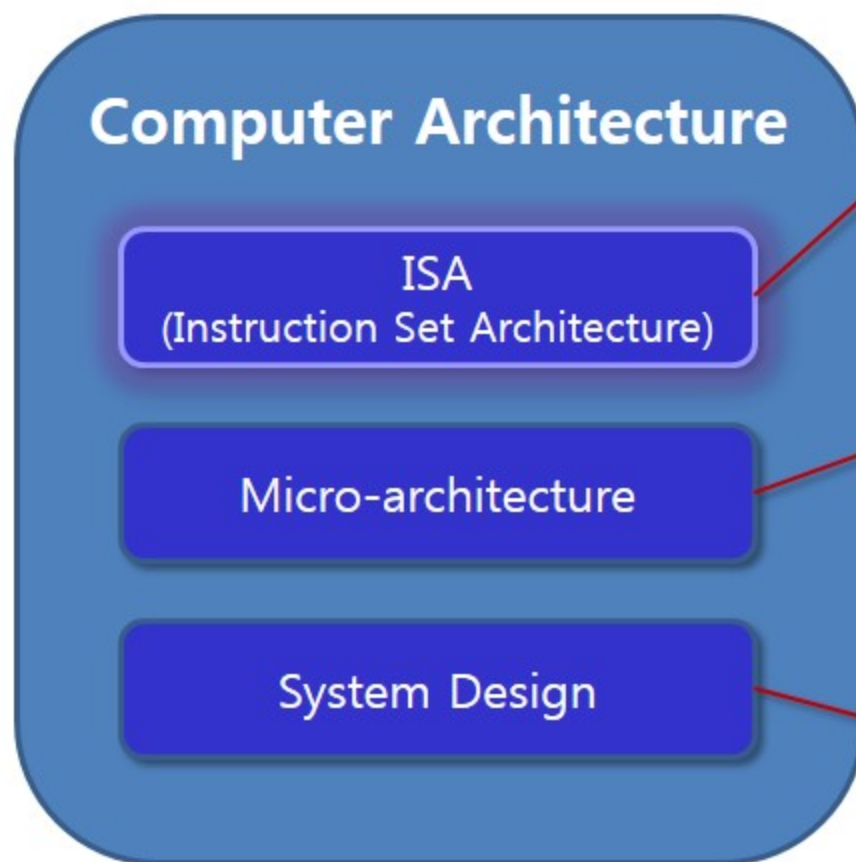
컴퓨터 구조

■ 컴퓨터 구조(Computer Architecture)



컴퓨터 구조

컴퓨터 구조(Computer Architecture)의 협의적 해석들



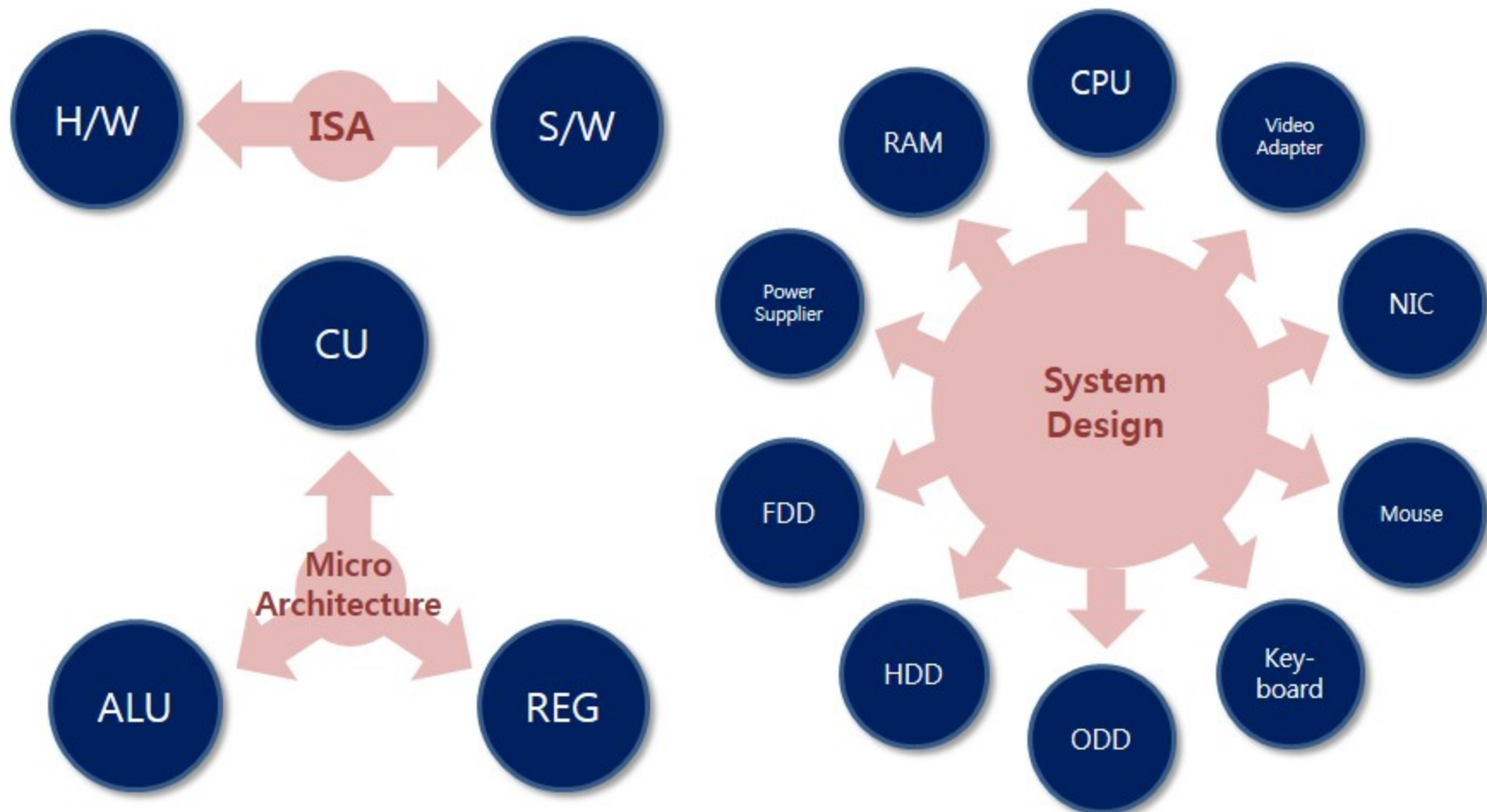
- 기계어 명령 체계
- 하드웨어와 소프트웨어간의 관계 및 약속 (protocol)
- 기계어 프로그래머들이 보는 컴퓨터 시스템에 대한 추상적인 이미지
- 기계어 집합, 워드 사이즈, 메모리 주소 모드, CPU의 레지스터, 주소, 데이터 포맷 등이 포함

- ISA의 각각의 명령들을 구현하기 위한 CPU의 구성요소들과 그것들의 결합에 대한 상세한 시스템 명세

- CPU와 나머지 하드웨어 구성요소들과의 연결에 대한 설계
- 버스, 스위치 같은 시스템 연결 구조(메인 보드)
- 메모리 컨트롤러와 계층구조, DMA(direct memory access)와 같은 CPU 독립 구조(CPU off-load mechanism), 다중처리(multiprocessing)을 위한 구조

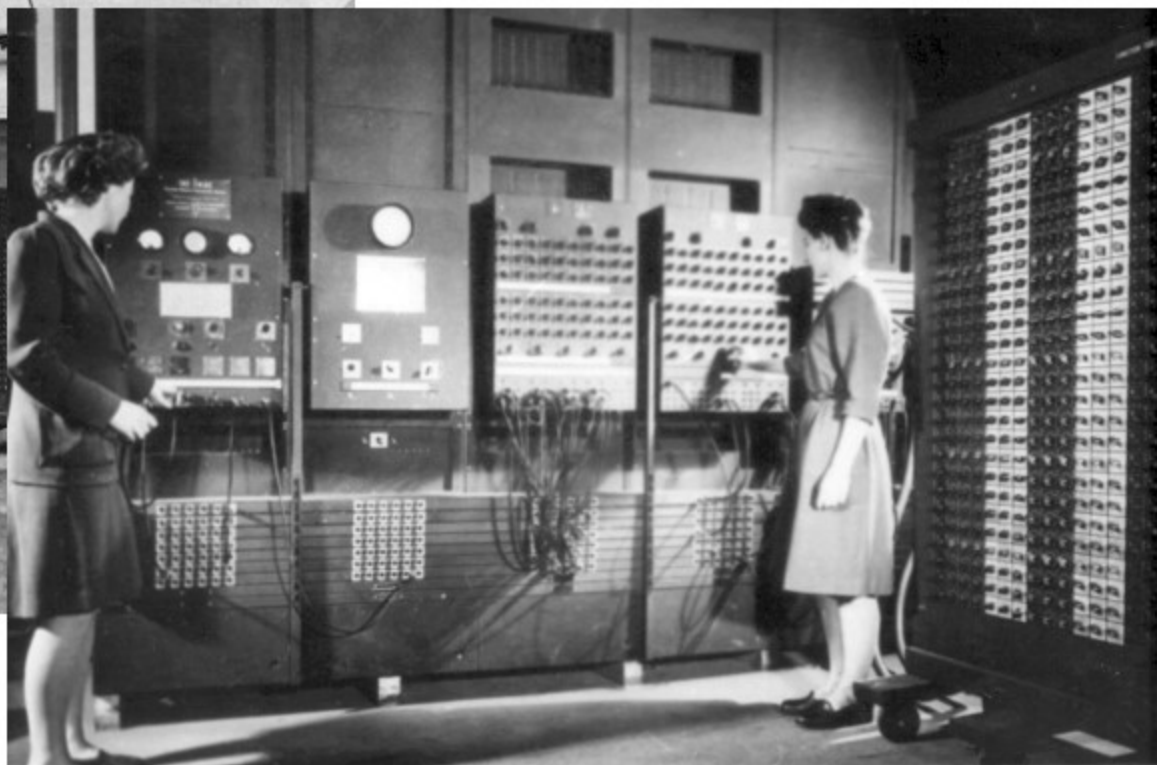
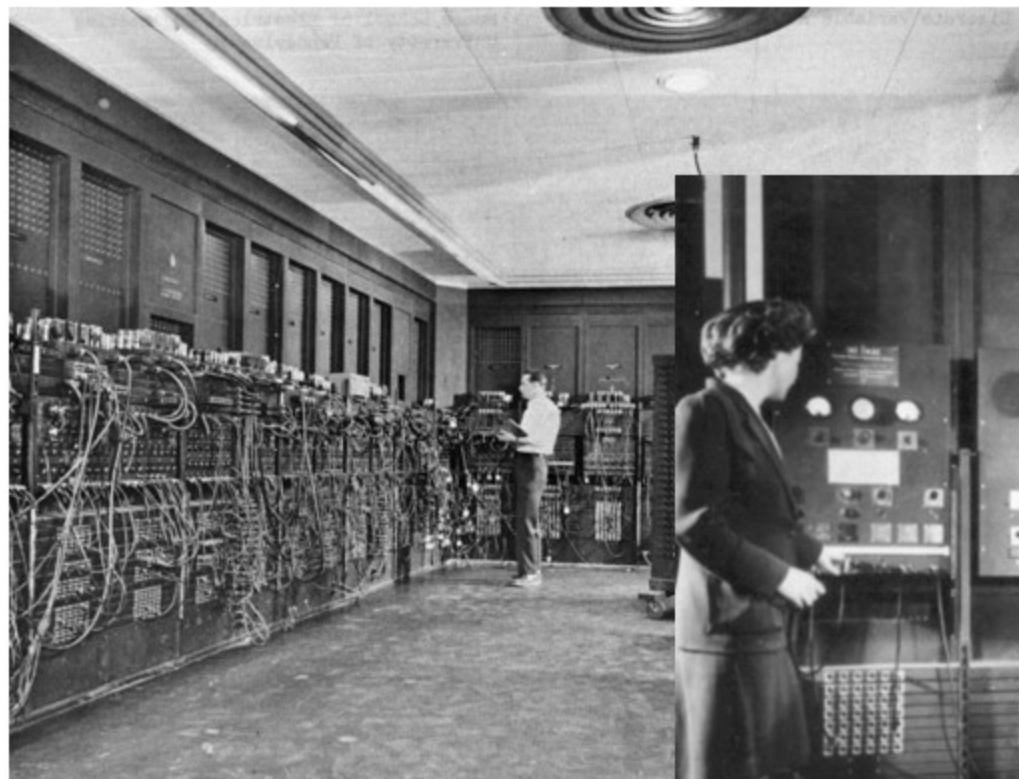
컴퓨터 구조

컴퓨터 구조(Computer Architecture)의 협의적 해석들



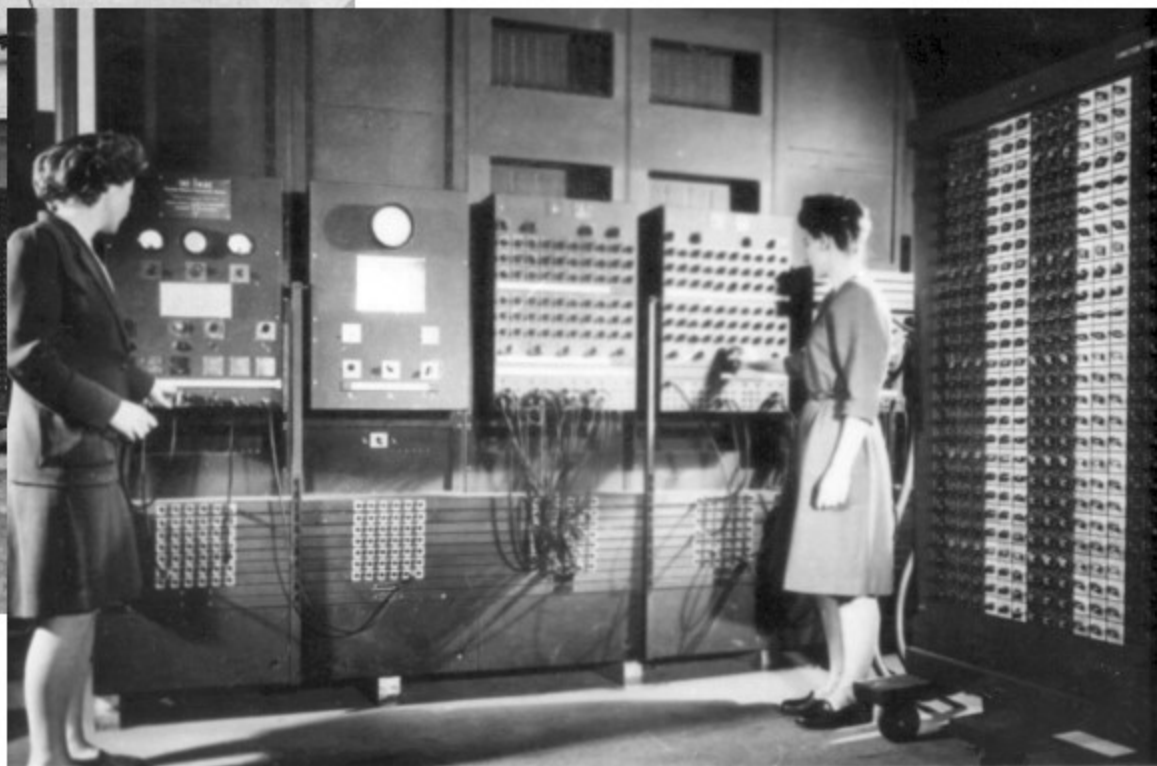
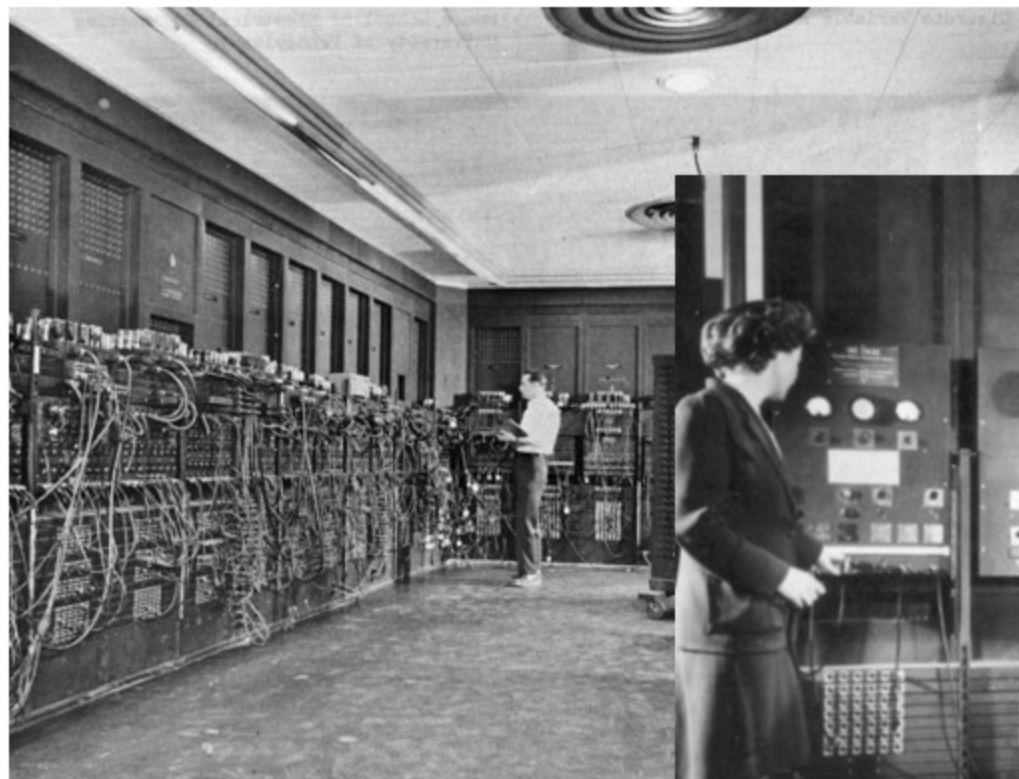
폰 노이만 컴퓨터 구조

ENIAC: Program-controlled Computer(Wired Programming)

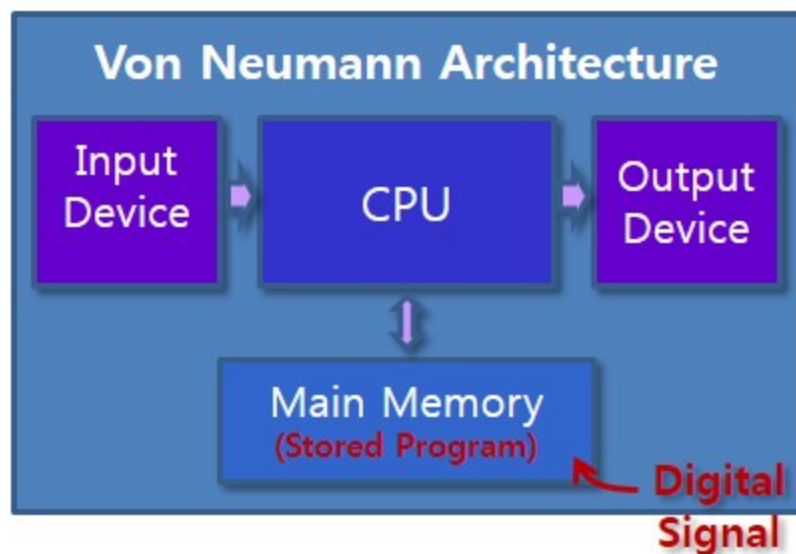
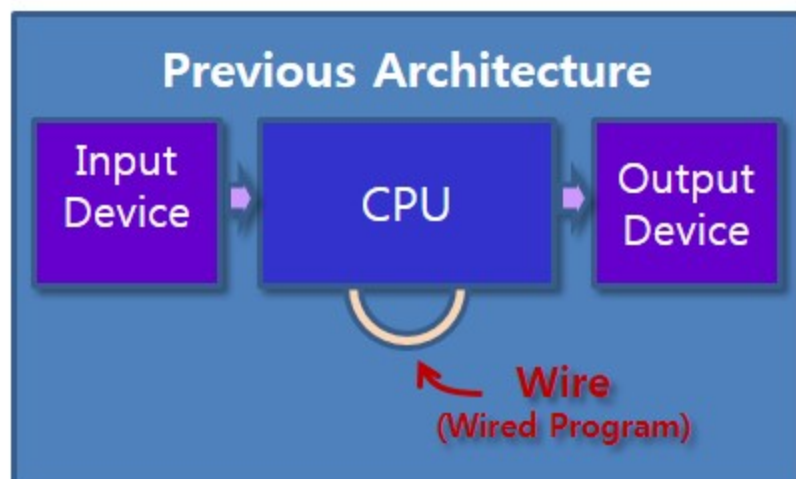


폰 노이만 컴퓨터 구조

ENIAC: Program-controlled Computer(Wired Programming)



폰 노이만 컴퓨터 구조



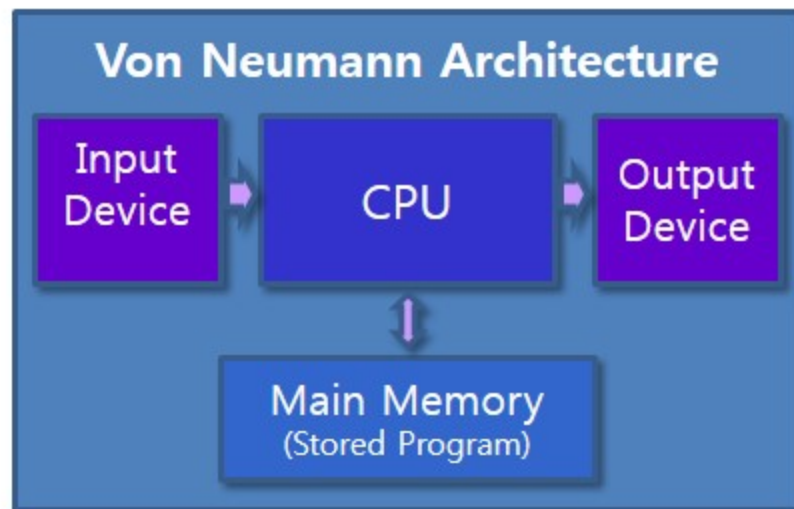
■ 폰 노이만 컴퓨터 구조 (Von Neumann Computer)

항목	내용
제안	1945년 컴퓨터 과학자였던 존 폰 노이만(John von Neumann)이 제안한 컴퓨터 구조
요소	입력 장치, 출력장치, 중앙처리장치(연산장치, 제어장치), 메인 메모리 로 구성
특징	프로그램 내장 방식(stored-program digital computer)
장점	효율적인 프로그래밍으로 인해 프로그램 개발 비용과 시간이 절약됨.

※ **프로그램 내장 방식(stored-program):**
 주기억장치(main memory)에 저장된 명령을 중앙처리장치로 가져와 하나씩 수행.
 이때 필요에 따라 주 기억장치에서 데이터도 가져옴.

폰 노이만 컴퓨터 구조

EDVAC: stored-program digital computer

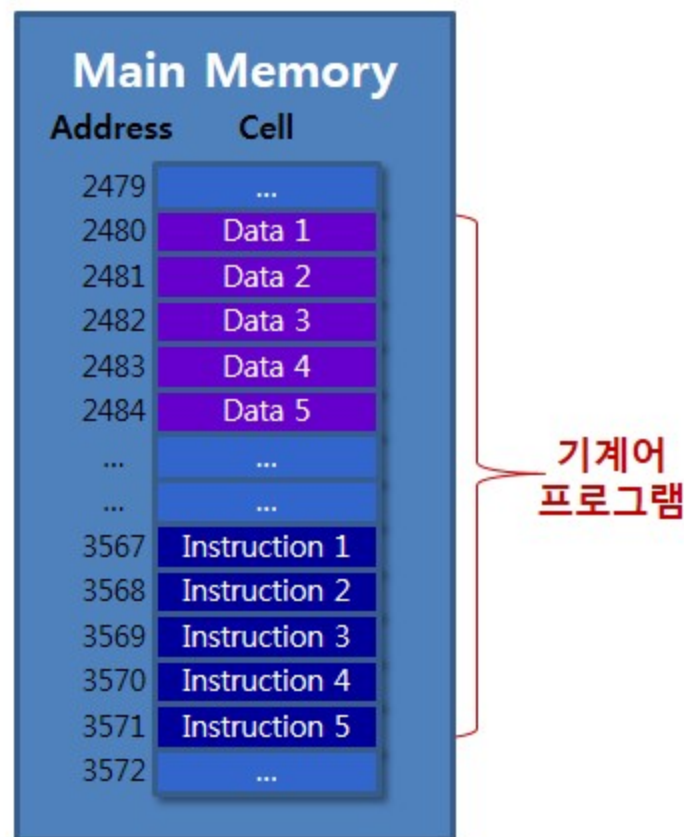


Hardware vs. Software

Computer Program
(메인 메모리에 내장됨, 보이지 않음)

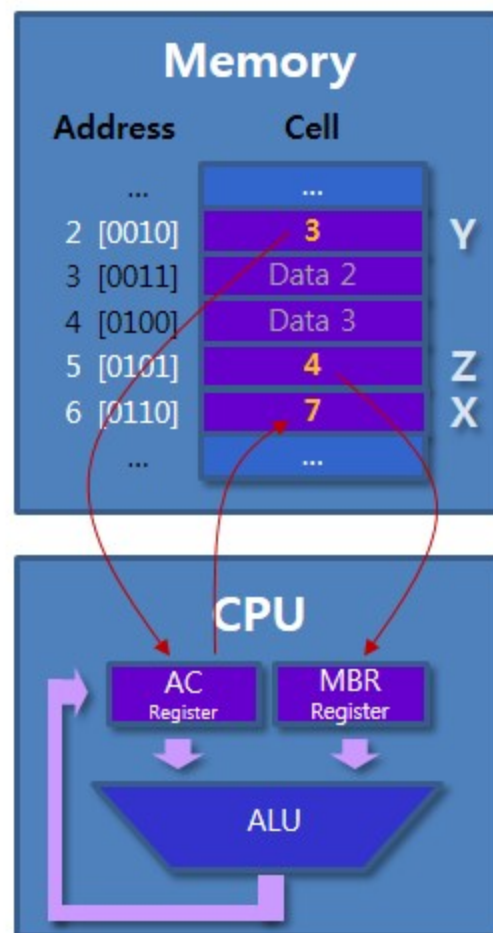
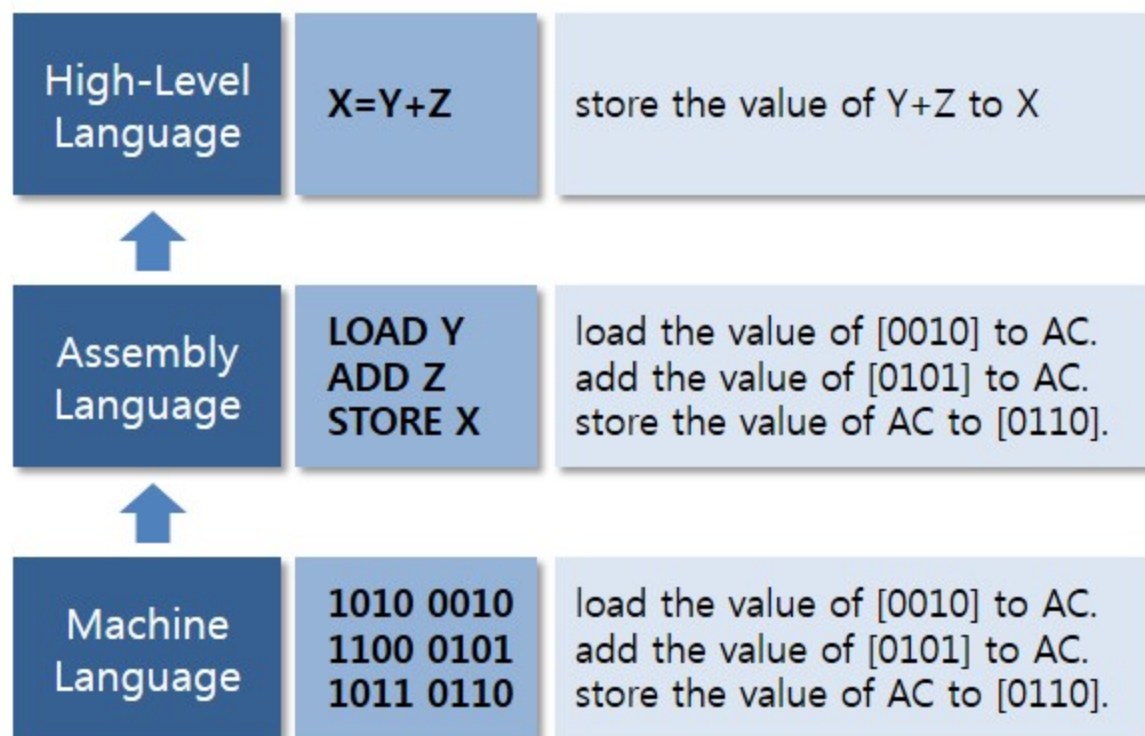
프로그램의 실행 원리

- 컴퓨터 프로그램이란?
 - 기계어 프로그램
 - 0과 1로 표현됨
 - ISA(기계어 명령 체계)를 따라 표현됨.
 - CPU가 이해할 수 있고 바로 수행할 수 있는 유일한 프로그램.
 - 사람이 이해하기 어려움.



프로그래밍 언어

발전 단계에 따른 프로그래밍 언어의 세 가지 분류



프로그래밍 언어

각 단계별 프로그래밍 언어의 특징

비교 항목	기계어 (machine language)	어셈블리어 (assembly language)	고급어 (high-level language)
언어 표현 방식	0과 1로 표현	간단한 단어로 표현 (기계어와 1대1 대응)	수식 및 간단한 단어로 표현
H/W의 이해 여부	가능	불가능	불가능
사람의 이해 용이성	어려움	약간 쉬움	쉬움
이종 ISA 호환 여부	호환 불가	호환 불가	호환 가능 (해당 ISA 전용 번역기를 통해 번역 가능)
번역기	필요 없음	어셈블러(assembly)	컴파일러(compiler)
번역기 제작	필요 없음	번역기 제작 쉬움	번역기 제작 어려움

프로그래밍 언어

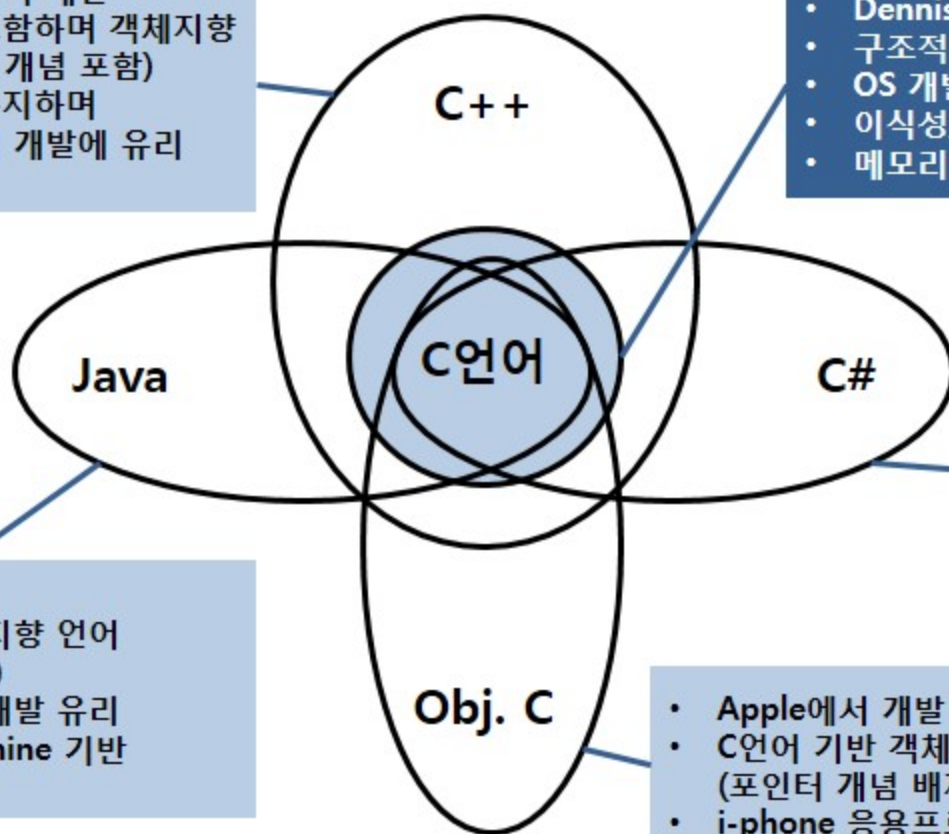
- 초기 고급 언어(High-level Language)의 예
 - FORTRAN(FORmula TRANslation)
 - 1957년 IBM의 존 배커스가 개발한 최초의 성공적인 고급어.
 - 단순화된 영어 단어와 수학기호로 표현.
 - 과학계산용으로 사용되며, 병렬처리 등에 탁월한 성능을 발휘함.
 - COBOL(Common Business Oriented Language)
 - 1960년 초에 개발됨.
 - 급여, 회계, 기타 사무용 프로그래밍 개발에 널리 사용된 고급어.
 - BASIC(Beginner's All-purpose Symbolic Instruction Code)
 - 1964년에 개발됨.
 - 학생들에게 프로그래밍 언어를 가르치기 위해 개발.
 - C Language
 - 1970년대 Dennis Ritchie가 개발한 고급어.
 - OS 개발 및 시스템 프로그래밍용 언어
 - 포인터 등의 개념을 지원하여 저급 언어에 가까운 정교한 프로그래밍이 가능하므로 중급어로 분류되기도 함.

프로그래밍 언어

C언어와 C언어 기반 객체지향 언어와의 관계

- Bjarne Stroustrup이 개발
- C언어를 완벽히 포함하며 객체지향 개념 추가 (포인터 개념 포함)
- C언어의 장점을 유지하며 대규모 소프트웨어 개발에 유리
- 배우기 어렵다

- Dennis Ritchie, Ken Thomson이 개발
- 구조적 프로그램 (객체지향 개념 없음)
- OS 개발 및 시스템 프로그래밍에 유리
- 이식성, 효율성, 간결성
- 메모리 관리가 중요함.



- Sun에서 개발
- C언어 기반 객체 지향 언어 (포인터 개념 배제)
- 웹 응용프로그램 개발 유리
- Java Virtual Machine 기반
- 플랫폼 독립성

- MS에서 개발
- C언어 기반 객체 지향 언어 (포인터 개념 배제)
- 컴포넌트 개발 유리
- .NET Framework 기반
- 개발 환경 독립성

- Apple에서 개발
- C언어 기반 객체 지향 언어 (포인터 개념 배제)
- i-phone 응용프로그램 개발용

프로그래밍 언어

C언어 기반 객체지향 언어들의 비교

	C언어	C++	Java	C#	Objective C
제작	Dennis Ritchie, Ken Thomson	Bjarne Stroustrup	James Gosling	Anders Hejlsberg	Brad Cox, Tom Love
제작사	AT&T, 1972	AT&T, 1983	SUN, 1994	MS, 1999	StepStone 1981, Apple 인수, 1996
목적	OS개발(기계제어)	범용 SW	가전제품/웹/안드로이드 응용SW	윈도우즈 응용SW	OS X, iOS 응용SW
지향점	ISA 독립 기계제어 SW의 개발	C언어의 객체지향 확장	플랫폼 독립성 (라이선스 수익)	개발환경 독립성 ¹⁾ (플랫폼 종속성)	-
특징	간결성, 효율성	C언어 계승	Java Virtual Machine	.Net Framework(CLI)	-
포인터	있음	있음	없음	없음	없음
메모리 관리	수동	수동	자동	자동	자동
객체 지향	없음	포함	포함	포함	포함

¹⁾ CLI: Common Language Infrastructure

프로그래밍 언어

■ C++ 언어 프로그램

- 고급 언어(high level language)에 속함.
- 알파벳, 숫자, 기호 등으로 표현됨.
- C++ 언어 문법 규칙을 따라 표현됨.
- 사람이 이해하기 쉬움.
- CPU가 이해할 수 없기 때문에 컴파일러를 통한 기계어 번역 작업이 필요함.
- **소스코드(source code)**라고도 함.

```
#include <iostream>

using namespace std;

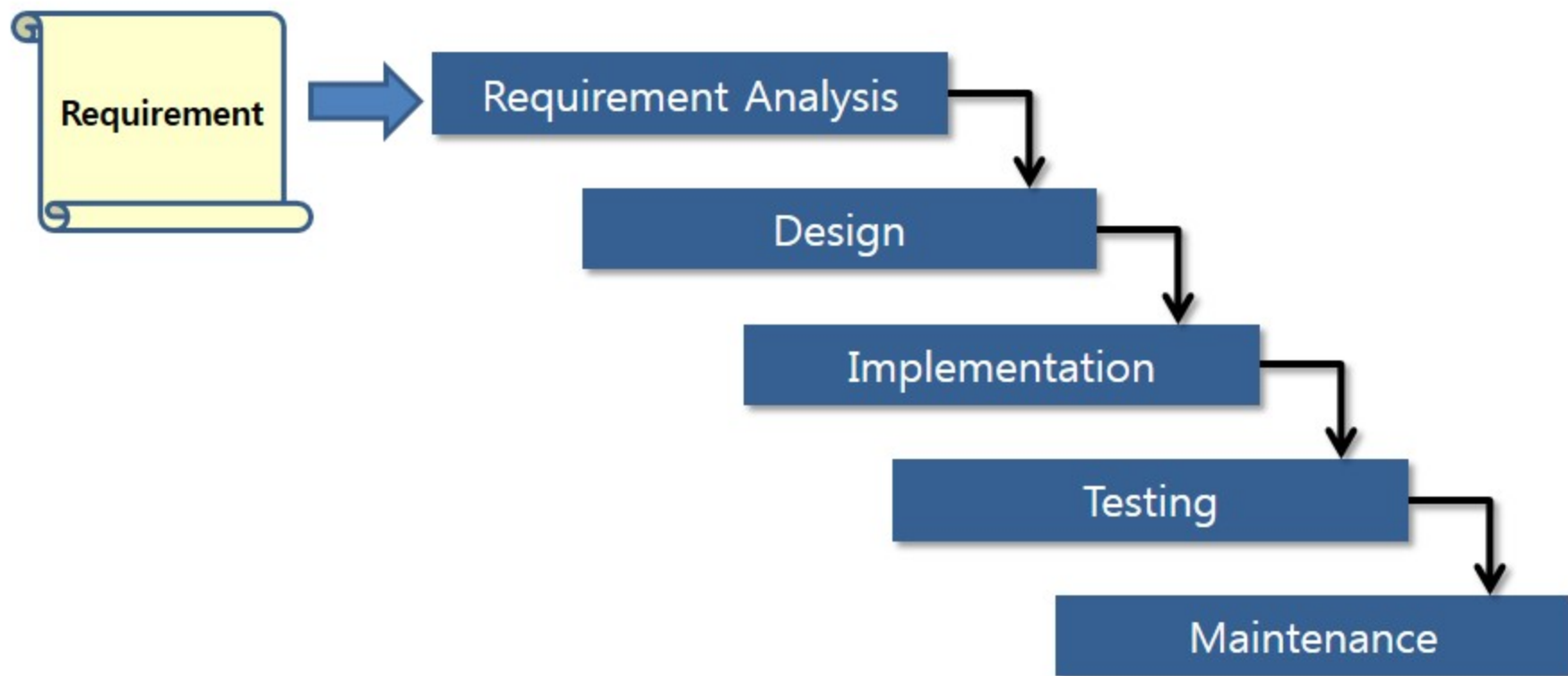
int gcd(int a, int b)
{
    for(int t=a; b!=0; t=a) {
        a = b;
        b = t%b;
    }
    return a;
}

int main(void)
{
    int a=12, b=16;
    cout << gcd(a,b) << endl;
    return 0;
}
```

C++ 언어
프로그램

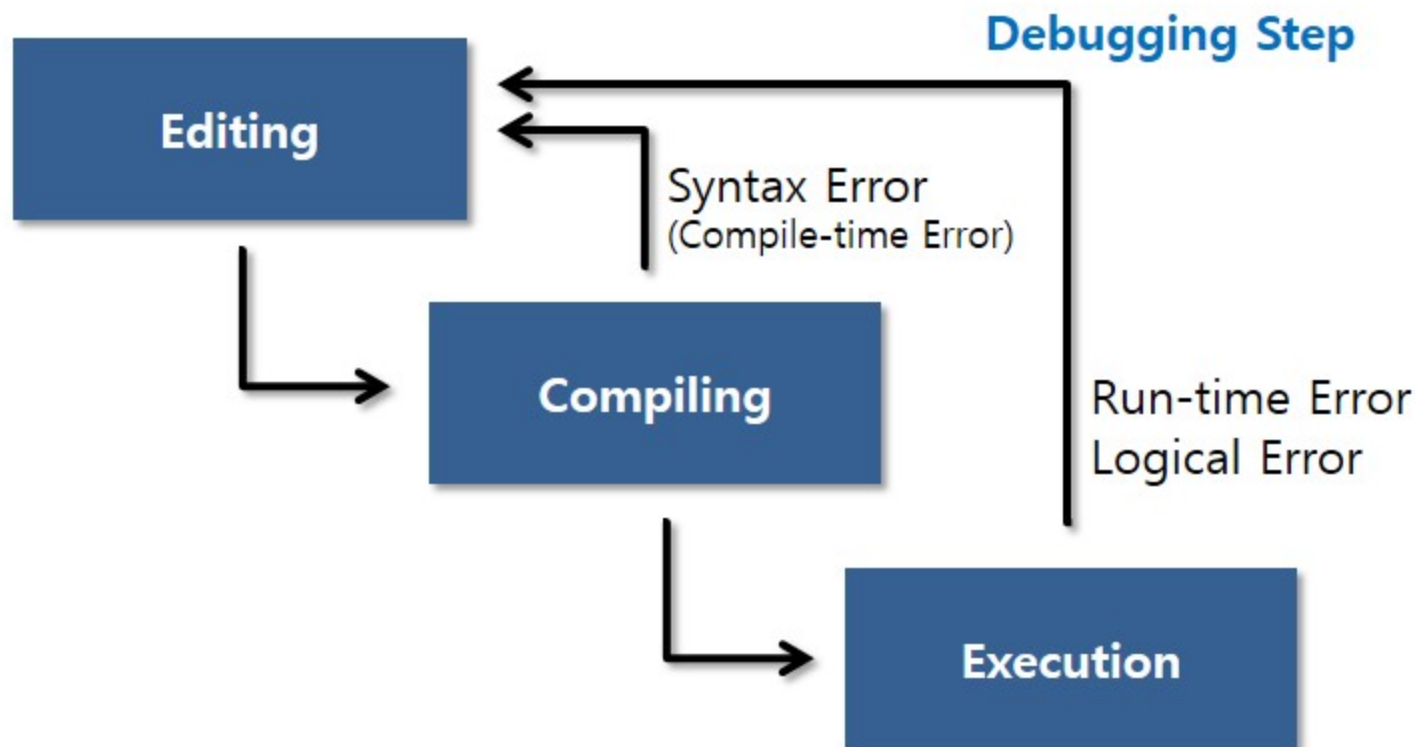
소프트웨어 개발

소프트웨어 개발 과정



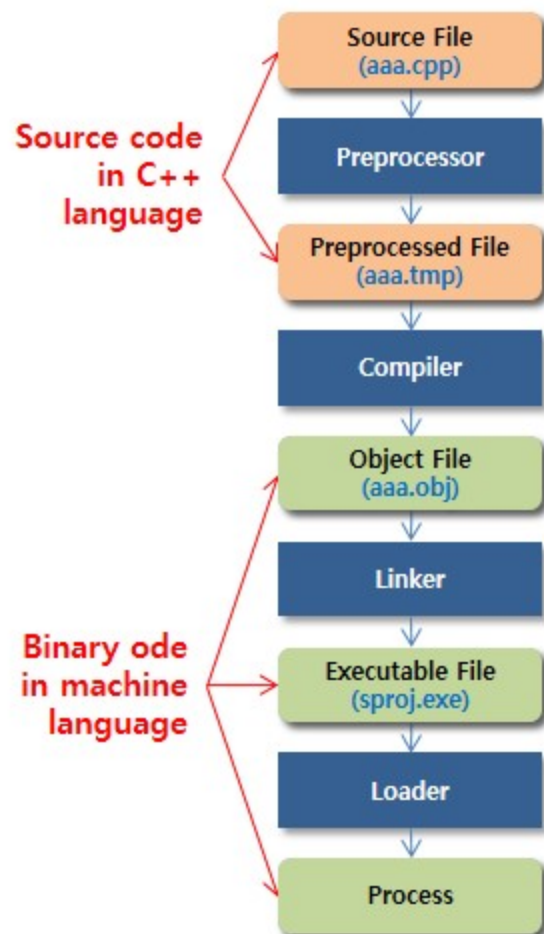
소프트웨어 개발

프로그램 작성과 실행

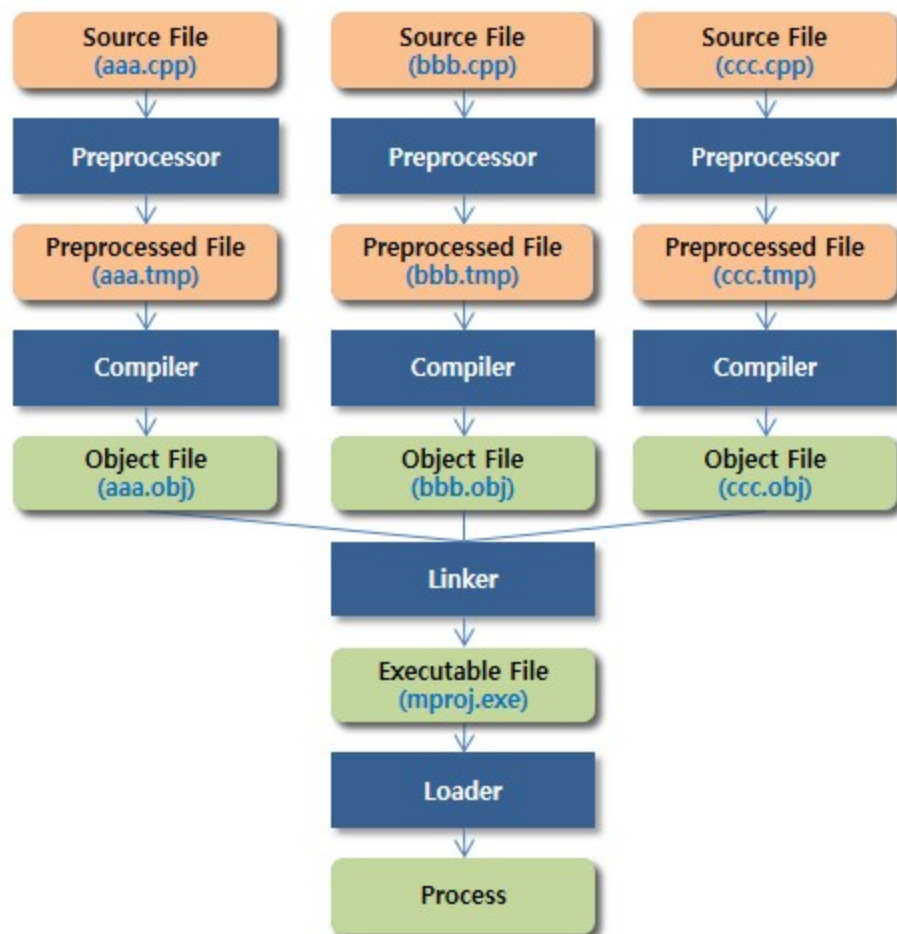


- 구문 오류(Syntax Error): 프로그램의 문법 오류
- 논리 오류(Logical Error): 프로그램의 기능 오류
- 실행 오류(Run-time Error): 프로그램이 비정상적으로 멈추거나 종료되는 치명적인 오류

소프트웨어 개발



**Single-source Project
(sproj.vcxproj)**



**Multi-source Project
(mproj.vcxproj)**

소프트웨어 개발

- 실행파일(Executable File)
 - **보조기억장치에 저장**되어 있는 기계어 프로그램.
 - 실행되고 있지 않음.
 - 보조기억장치(secondary memory)
 - hard disk, optical disk, flash memory
- 프로세스(Process)
 - **주기억장치에서 실행**되고 있는 기계어 프로그램
 - 실행파일이 메인 메모리로 옮겨지면 프로세스가 된다.
 - 주기억장치(main memory)
 - RAM(random access memory)
- 로더(Loader)
 - 실행파일을 프로세스로 만들어주는 프로그램.
 - 운영체제의 일부분이다.

