소프트웨어 프로젝트 1

컴퓨터 소프트웨어와 운영체제

2022학년도 1학기

국민대학교 소프트웨어학부

지난 수업에는… 정보의표현과컴퓨터하드웨어

2진수의 음수 표현

<2진수의 표현 방법 3가지(8bit)>

2진수	8비트 크기이며, MSB가 부호비트임		
	부호와 절대치	1의 보수	2의 보수
00000000	+0	+0	+0
00000001	+1	+1	+1
00000010	+2	+2	+2
00000011	+3	+3	+3
…	…	…	…
01111100	+124	+124	+124
01111101	+125	+125	+125
01111110	+126	+126	+126
01111111	+127	+127	+127
10000000	-0	-127	-128
10000001	-1	-126	-127
10000010	-2	-125	-126
10000011	-3	-124	-125
…	…	…	…
11111100	-124	-3	-4
11111101	-125	-2	-3
11111110	-126	-1	-2
11111111	-127	-0	-1

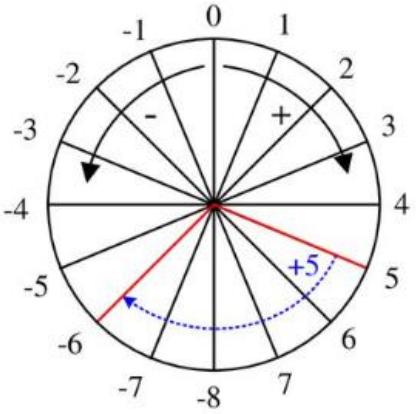
- **양수를 보수로 바꾸면 음수**
- **음수를 보수로 바꾸면 양수**

양수 ⇄ (보수 / 보수) 음수

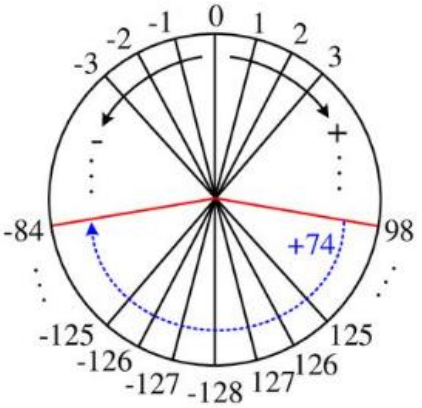
- 2진수와 그 수의 1의 보수와의 합은 모든 bit가 1이 된다.
- 2진수와 그 수의 2 보수와의 합은 모든 bit가 0이 된다.

(자릿수를 벗어나는 비트는 제외)

5+5 =?



98+74 =?



그럼, 예제의 135-62 =73 성립은?

소프트웨어란?

(Wikipedia) Computer software, or simply software, is a collection of data or computer instructions that tell the computer how to work.

즉, 이것이 없으면 (하드웨어만 있을 때에는)
컴퓨터는 무엇을 해야 할지 모른다!

(느슨한 분류)

프로그램 (소프트웨어) 에 의하여 동작이 결정되는 것 → 컴퓨터

(이에 비하여) 정해진 동작을 하도록 만들어진 것 → 전자기기

꼭 “사용자가 프로그래밍할 수 있는” 것을 의미하지는 않아요!

소프트웨어의 분류

	시스템 소프트웨어	응용 소프트웨어
기능	컴퓨터 시스템을 운용하거나 소프트웨어 개발을 지원하는 데 이용되는 소프트웨어의 모음	컴퓨터 시스템이 응용될 분야에 따른 기능을 수행하도록 만들어진 소프트웨어
종류	운영체제 (operating system) 컴파일러 (compiler) 링커/로더 (linker/loader) 시스템 운용 도구들	워드 프로세서 (word processor) 스프레드시트 (spreadsheet) 웹 브라우저 (web browser) 게임 (game) 등등…

운영체제 (Operating System)

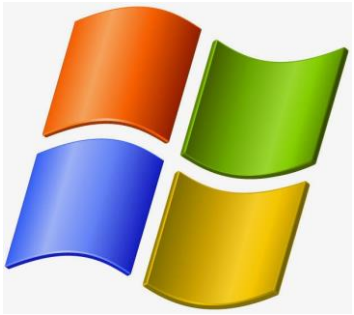
컴퓨터 시스템이 살아가기 위하여 반드시 필요한 소프트웨어 (의 모음)

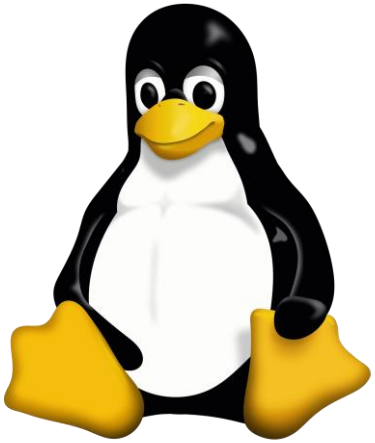
- 운영체제의 역할
  - 컴퓨터 시스템의 자원을 관리
    - (자원의 종류) 프로세서 시간, 메모리 공간, 입출력 장치, 파일 등
  - 사용자에게 시스템 자원을 활용할 수 있는 기능을 제공
    - 단일 사용자 시스템의 경우, 다소 간단
    - 다중 사용자 시스템의 경우, 다소 복잡

운영체제의 종류

이 외에도 아주 많아요!

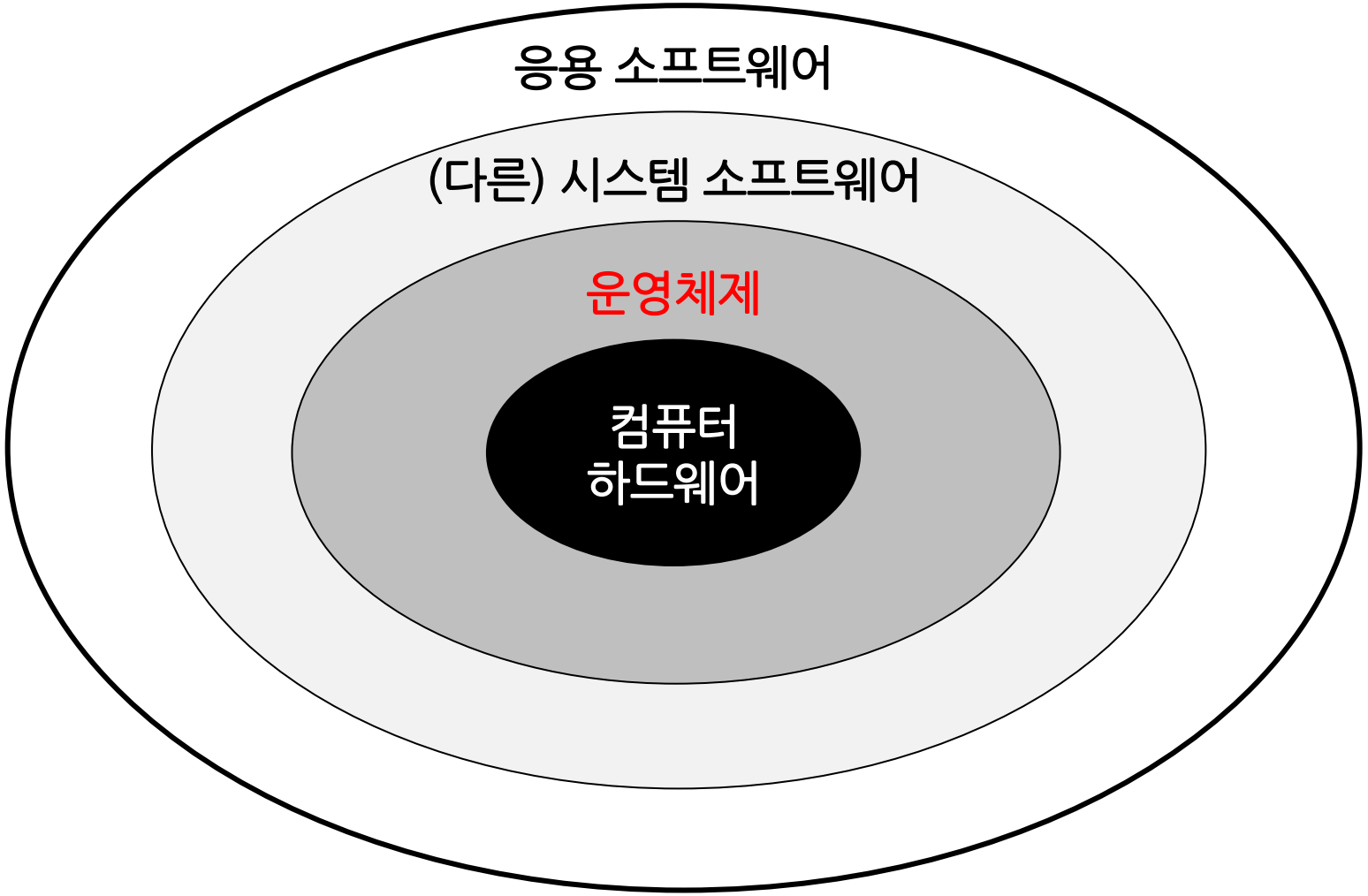
(그런데, 이들의 공통점은?)





UNIX®

컴퓨터 시스템에서의 운영체제



운영체제의 분류

- 운용 대상 시스템이 어떤 종류의 것인가?
  - 범용 (general purpose) / 내장형 (embedded)
- 여러 사용자들에 대해 서비스를 제공할 것인가?
  - 단일 사용자 (single-user) / 다중 사용자 (multi-user)
- 여러 개의 중앙처리장치로 이루어진 시스템을 지원할 것인가?
  - 단일 프로세서 (uniprocessor) / 다중 프로세서 (multiprocessor)
- 어떤 방식으로 복수의 작업을 행할 것인가?
  - 시분할 방식 (batch processing) / 다중 프로그래밍 (multiprogramming)

(연구과제) 앞에서 예로 든 시스템들을 이 기준에 따라 분류해 본다면?

운영체제의 역할

- 컴퓨터 시스템의 자원을 관리
  - 프로세스 관리
  - 메모리 관리
  - 파일 시스템 관리
  - 입출력 장치 관리
- 사용자에게 시스템 활용 도구를 제공
  - 텍스트 에디터
  - 시스템 관리 도구
  - 소프트웨어 개발 도구 (환경)
  - 웹 브라우저 (?)

프로세스 관리

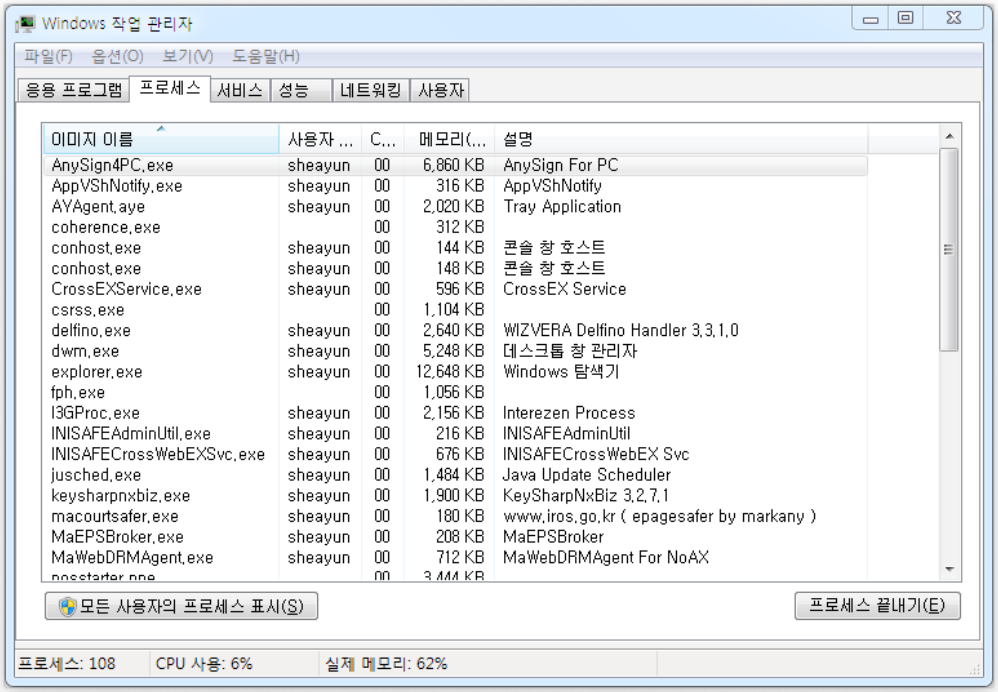
프로세스란? (대강 정의하면)

컴퓨터 시스템에서 CPU 및 메모리를 이용하여 실행 중인 작업

- 컴퓨터 시스템에서 실행되는 (거의) 모든 소프트웨어는

  하나 이상의 프로세스를 이룸
- 프로세스는 생명 주기 (life cycle) 를 가지며, 이것은 운영체제가 관리
- 프로그램 ≠ 프로세스
  - 프로그램은 실행이 준비된 소프트웨어의 덩어리
  - 프로세스는 지금 실행하고 있는 소프트웨어의 생명체

프로세스 관리

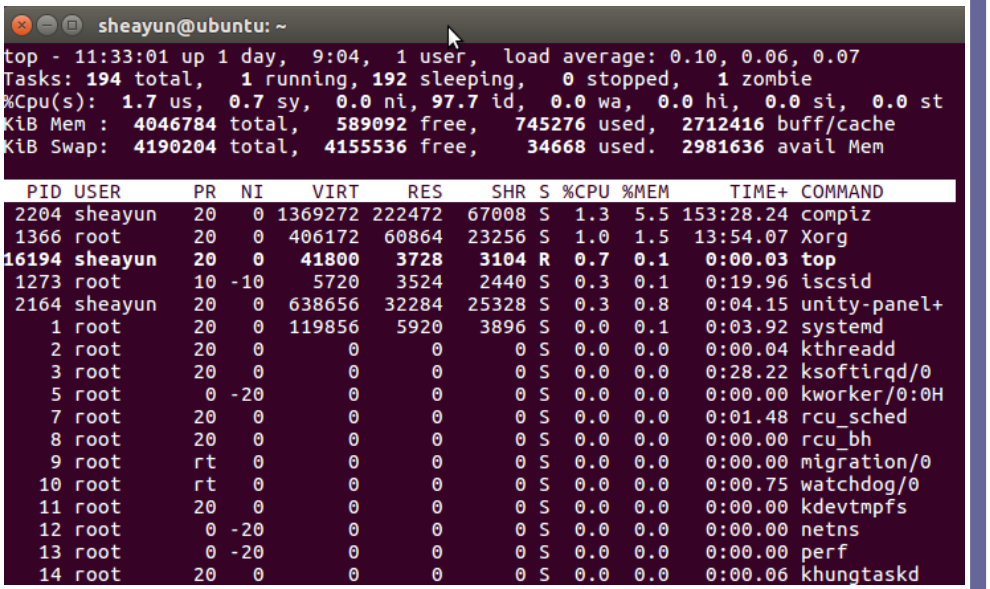


운영체제는

각 프로세스의 상태를 유지관리하고

각 프로세스가 시스템 자원을 사용할 수 있도록

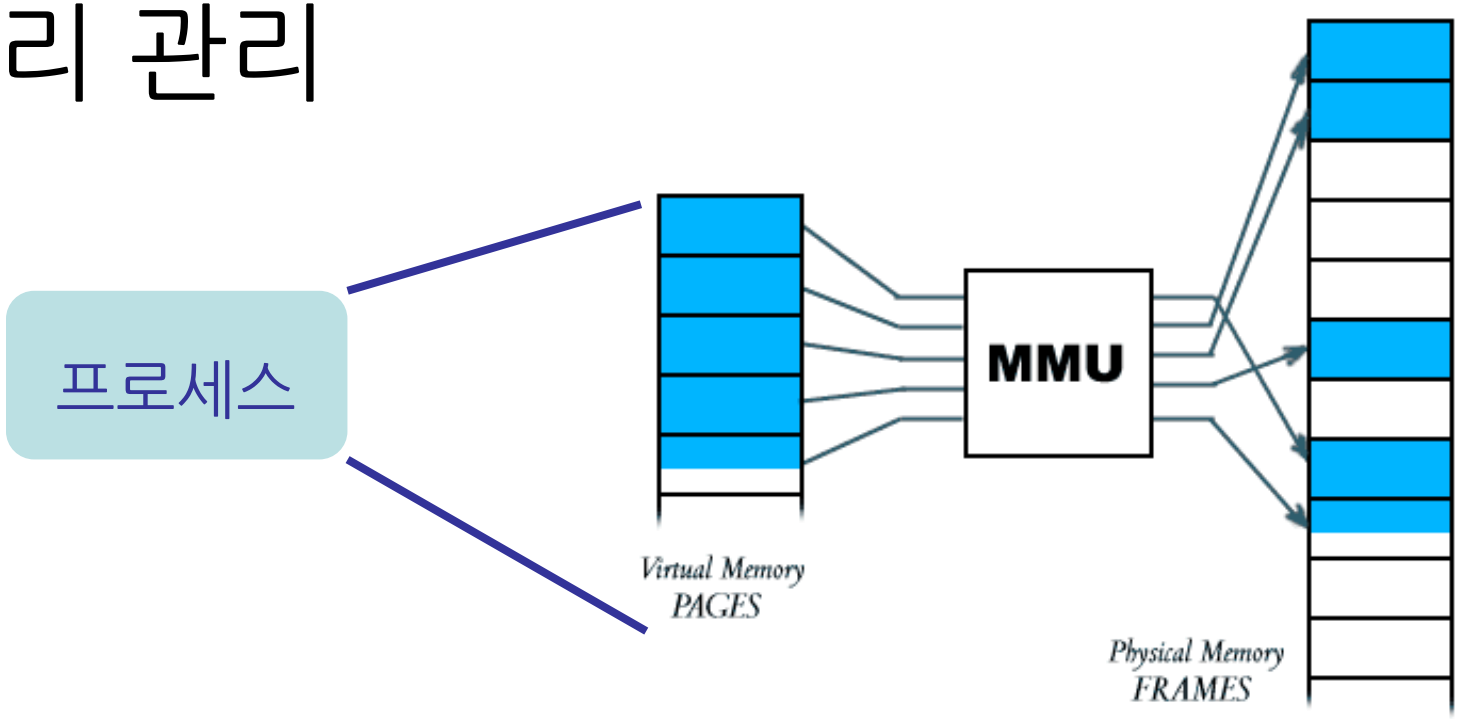
관리하고 있답니다!



(연구과제) 아무도 시스템 자원을 관리하지 않으면?

- “보호” 의 측면도 생각해야

메모리 관리



프로세스에게 메모리를 할당해 주고

프로세스가 반환하는 메모리를 “이용 가능한 것” 으로 관리하고

서로 다른 프로세스가 남의 메모리를 침범하지 못하도록 하는 일들은,

모두 운영체제의 몫이랍니다!

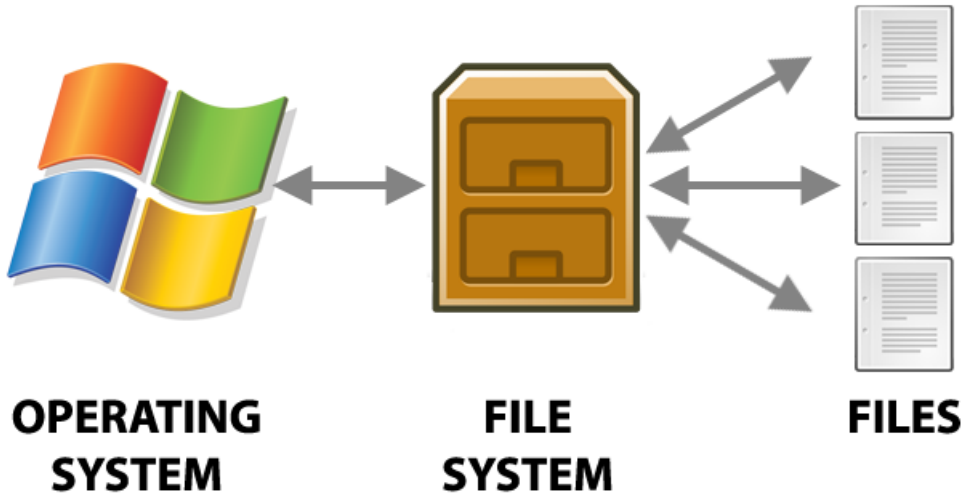
파일시스템 관리

디스크 드라이브도 입출력장치의 한 종류이지만,

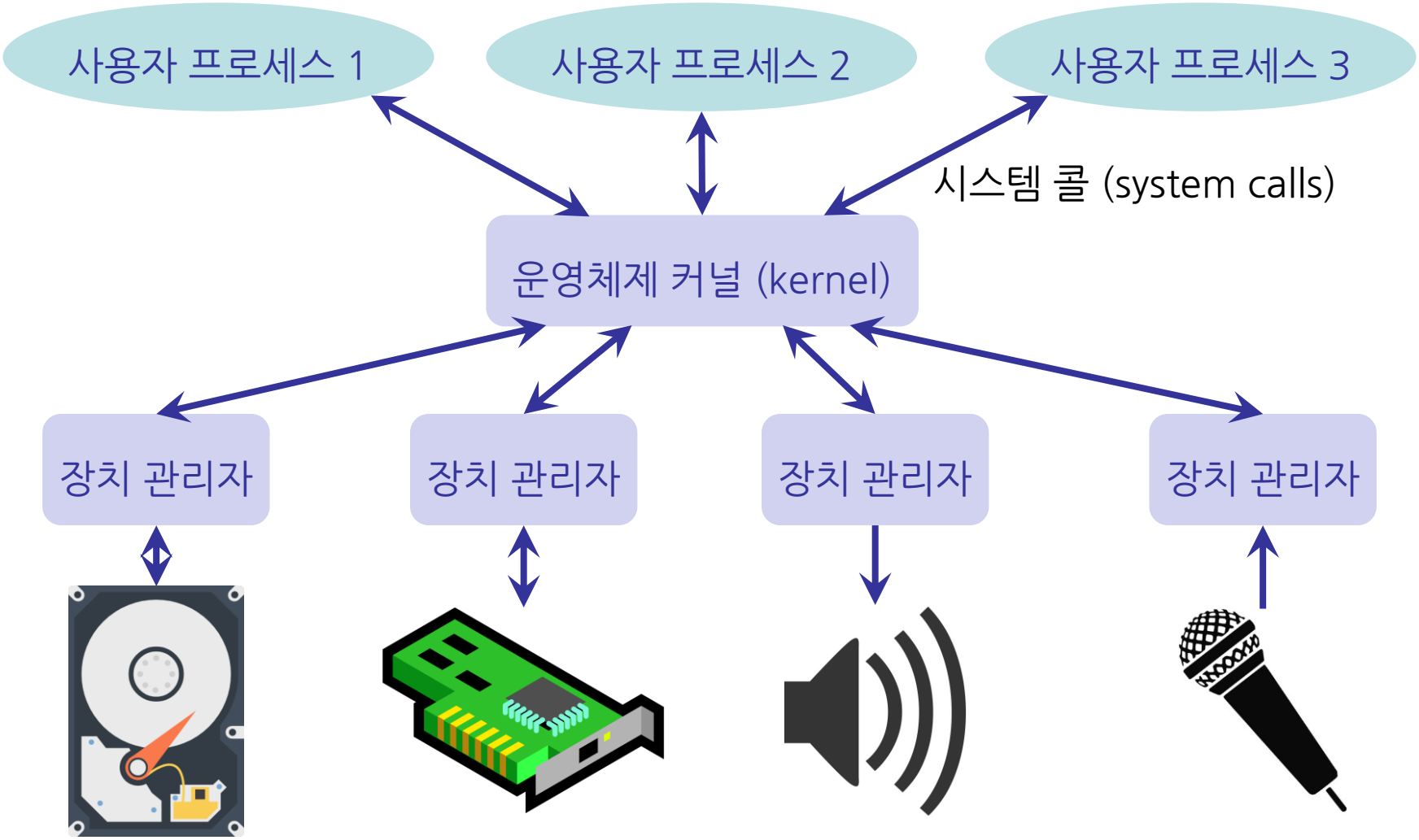
“파일시스템” 은 장치로서의 디스크 드라이브를 가리키는 것이 아니라

데이터 저장 매체를 이용하여 그 위에 “추상화” 된

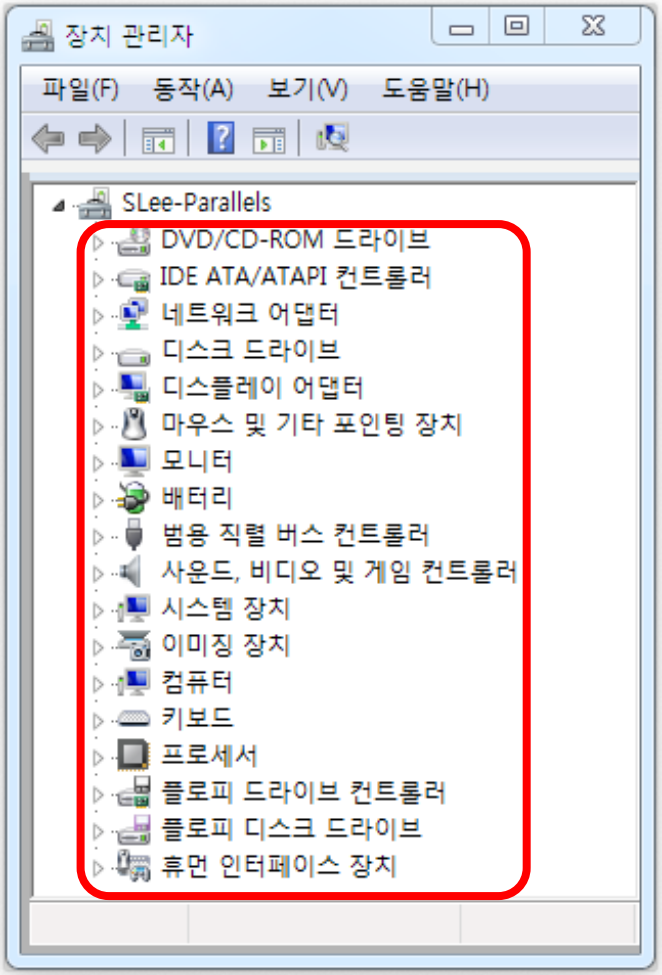
논리적인 구조를 뜻합니다!

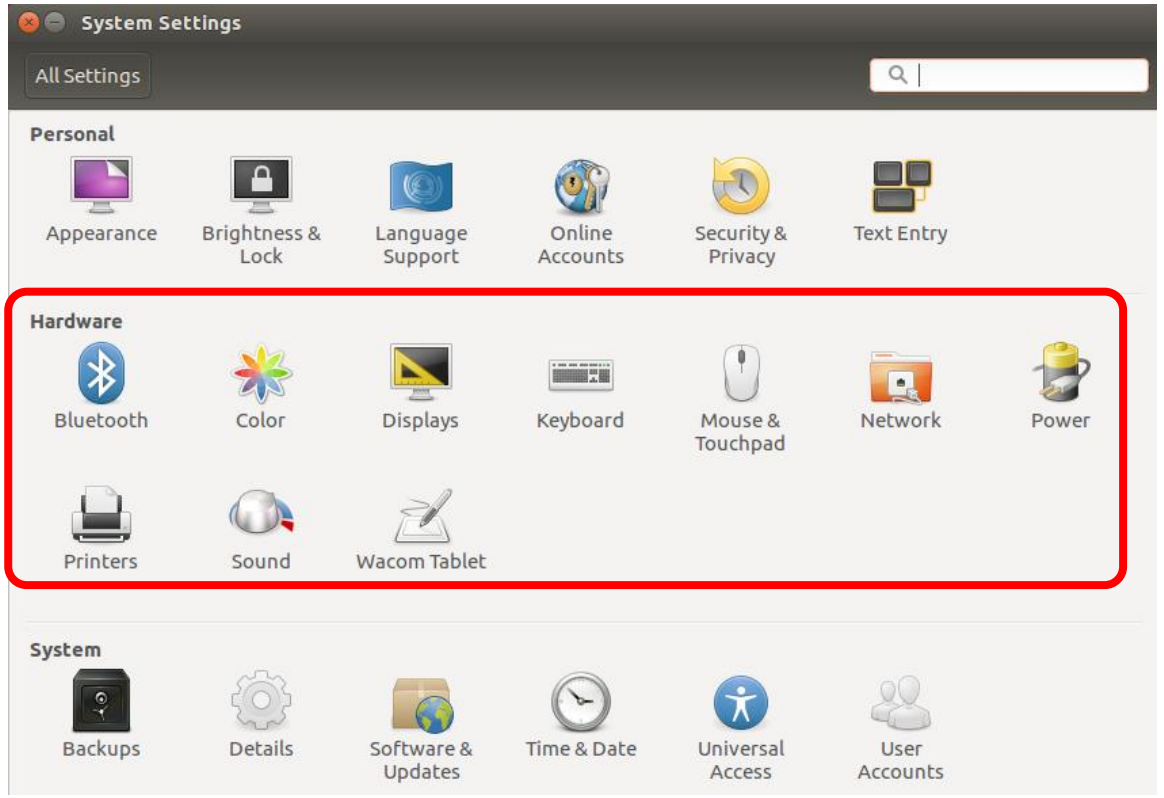


입출력 장치 관리


사용자 프로세스 1
사용자 프로세스 2
사용자 프로세스 3
시스템 콜 (system calls)
운영체제 커널 (kernel)
장치 관리자
장치 관리자
장치 관리자
장치 관리자

입출력 장치 관리





장치 관리자 (device driver) 는

하드웨어 제조사에 의하여 제공되는 경우도 많지만

운영체제의 일부로 보는 것이 타당합니다.

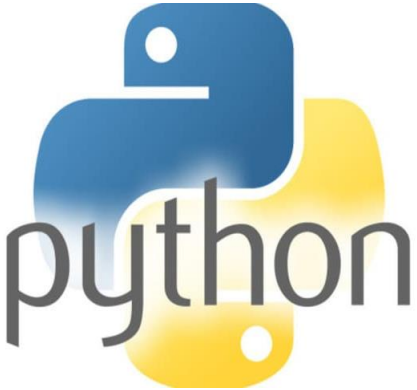
소프트웨어의 개발

소프트웨어의 개발 ? 프로그래밍

- 소프트웨어를 개발하는 데에는 프로그래밍 (코딩) 이외의 일들도 많이 필요
  - 기획, 설계, 디자인, 테스트, 문서화, …
- 하지만 프로그래밍 없이는 소프트웨어가 만들어지지 않음

- 쉽사리 빠져들 수 있는 오해들:
  - 프로그래밍만 잘 하면 됐지 다른 건 신경쓸 일 아님! (누군가 하겠지…)
  - 난 소프트웨어 개발 기획 일을 할 거니까 프로그래밍 좀 못 해도 될 거야!

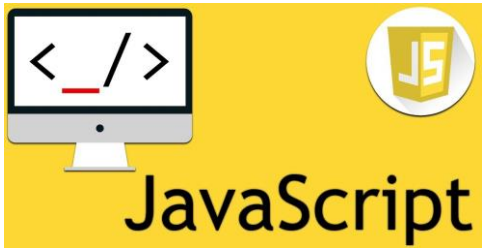
프로그래밍 언어



(FAQ)

이걸 다 알아야 하나요? → 아니오

많이 알면 쓸모 있나요? → 예



한두 가지 하다 보면 비슷한 면이 있어서

여러 가지에 익숙해지게 됩니다!

프로그래밍 언어의 분류

사람이 생각하는 방식에 가까운

언어 구조와 표현

고수준 프로그래밍 언어:

C++, Java, Python, …

C ?

어셈블리

(Assembly)

기계어

(Machine language)

기계어를 (거의) 1:1 로

사람이 알아볼만한 기호로 표현

기계 (CPU) 가 바로 이해할 수 있는

(사람은 이해하기 어려운) 언어

복잡한 설계와 소프트웨어 상호작용을

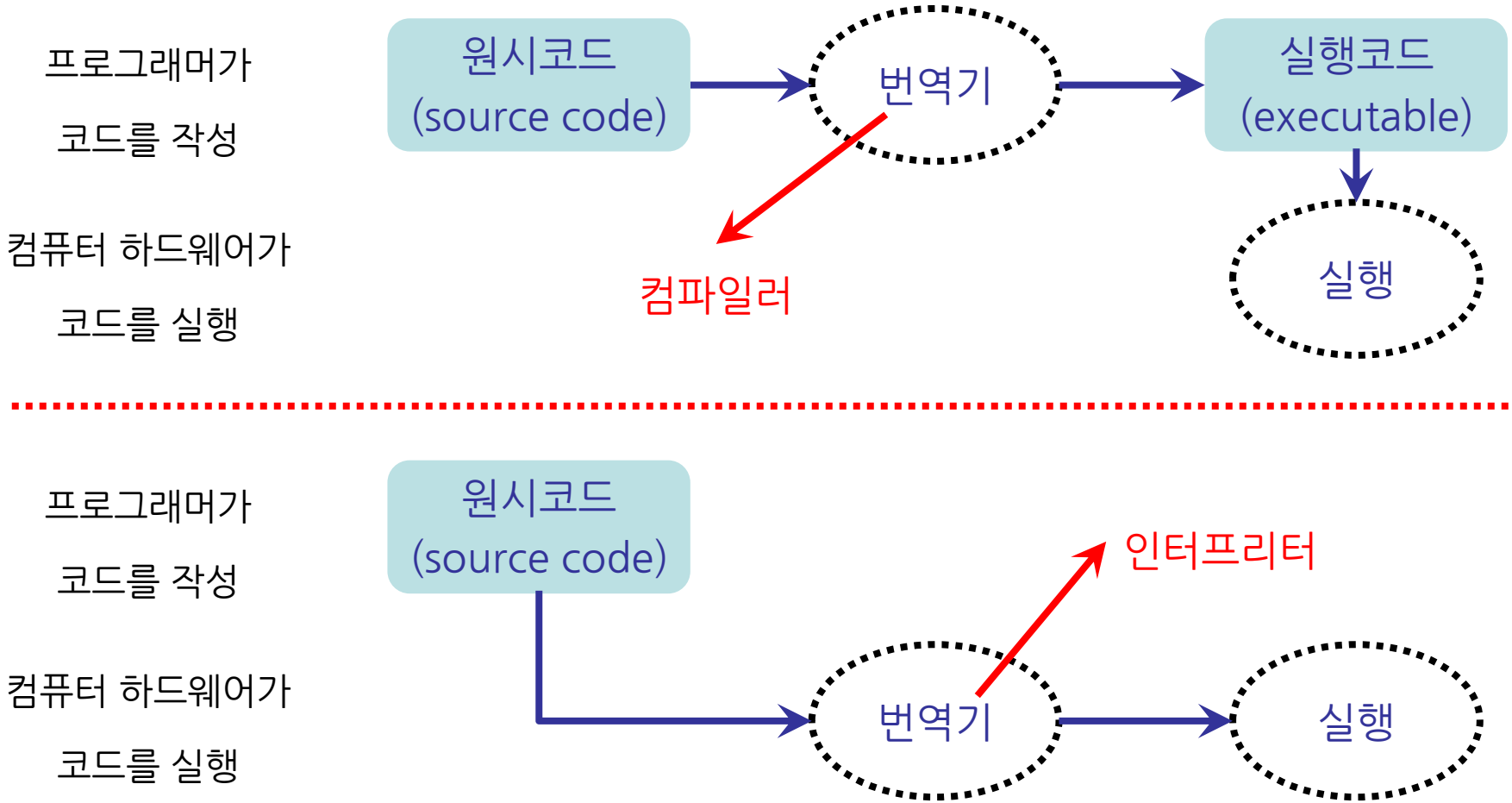
표현하기에 적합/용이하고

하드웨어 의존성이 최소

하드웨어에 특화된 프로그래밍 가능

→ 실행 효율이 높(을 수 있)음

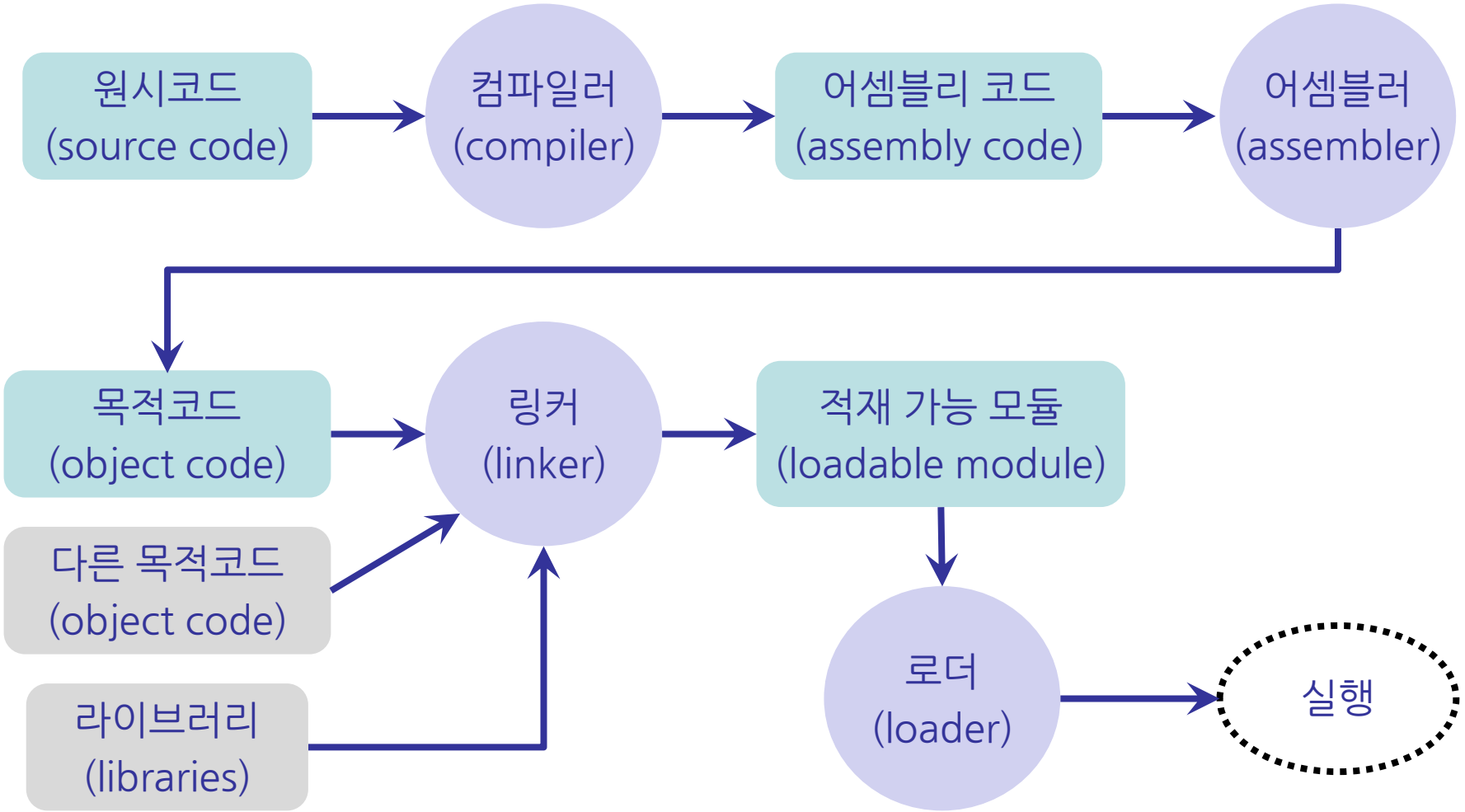
프로그램 번역/실행의 두 가지 모델



프로그램 번역/실행의 모델 비교

- 컴파일러 방식의 장점
  - 번역을 실행 이전에 해 두므로 코드를 공들여 최적화할 수 있다.
  - 따라서 코드 실행의 효율이 높다.
- 인터프리터 방식의 장점
  - 다른 컴퓨터로 옮겨 실행하는 것이 편리하다.
  - 대화형 개발이 가능하다.
- 이 두 방식의 장점을 택한 (어중간한) 방법들도 이용되고 있습니다!

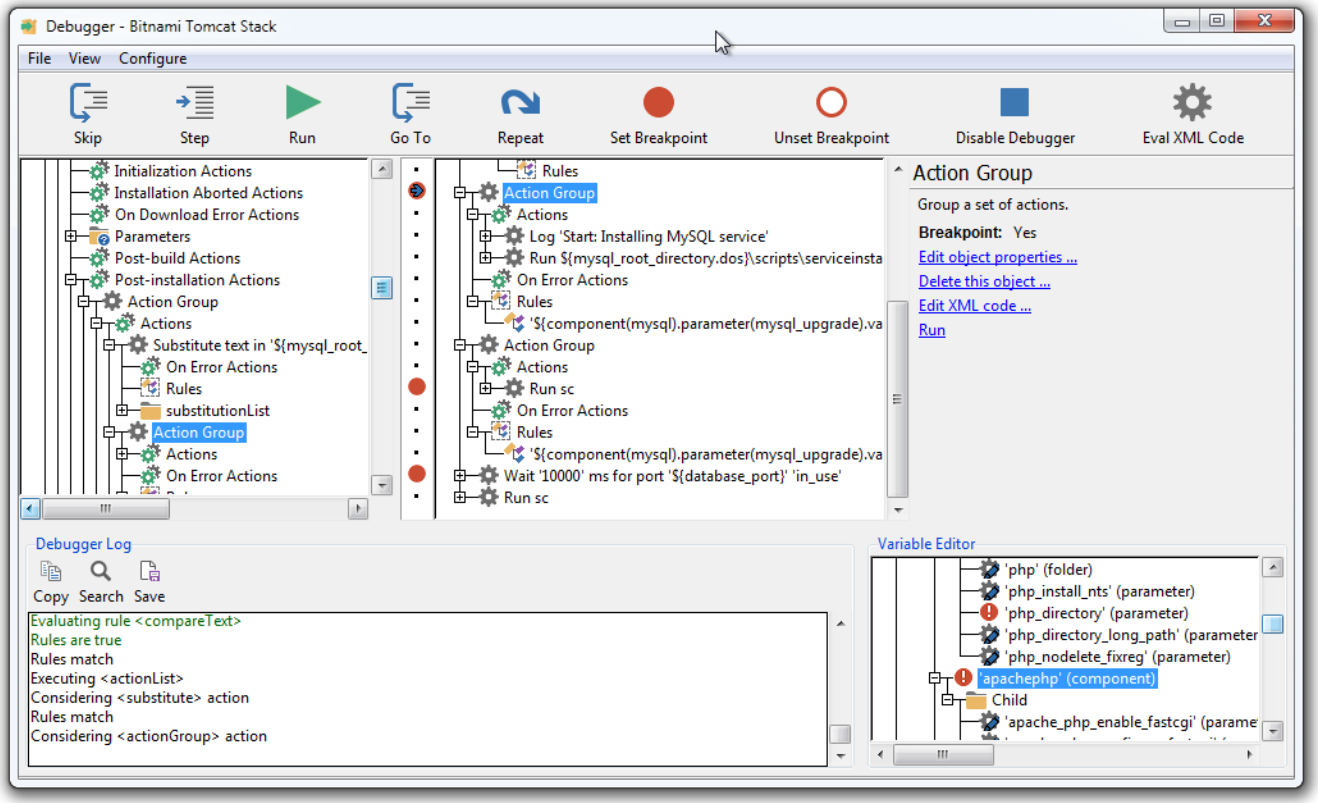
컴파일러와 링커/로더


원시코드
(source code)
컴파일러
(compiler)
어셈블리 코드
(assembly code)
어셈블러
(assembler)
목적코드
(object code)
다른 목적코드
(object code)
라이브러리
(libraries)
링커
(linker)
적재 가능 모듈
(loadable module)
로더
(loader)
실행

디버거

Debugger

- 다른 대상 프로그램을 테스트하고 동작을 관측, “버그” 를 찾아내어 수정하는 데 쓰이는 도구



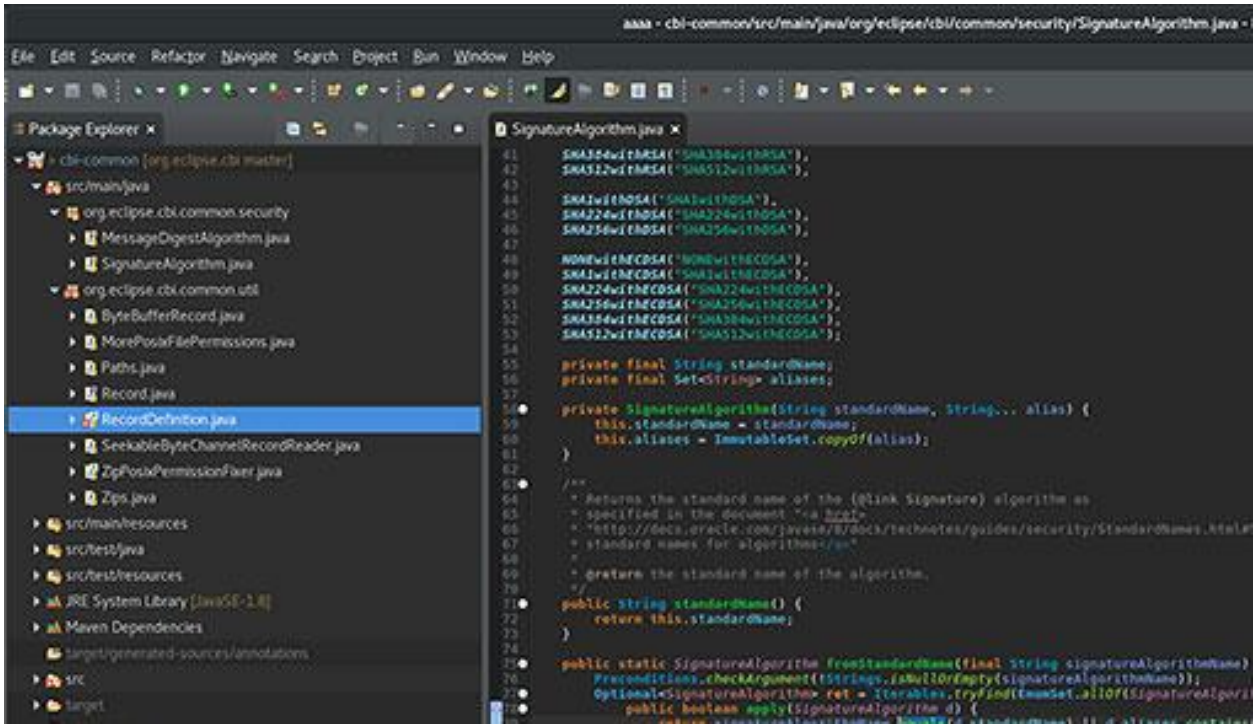
통합 개발 환경 (IDE)

Integrated Development Environment

- 코딩, 디버깅, 컴파일, 배포 등 프로그램 개발에 관련된 작업을 하나의 소프트웨어 안에서 처리

(예)

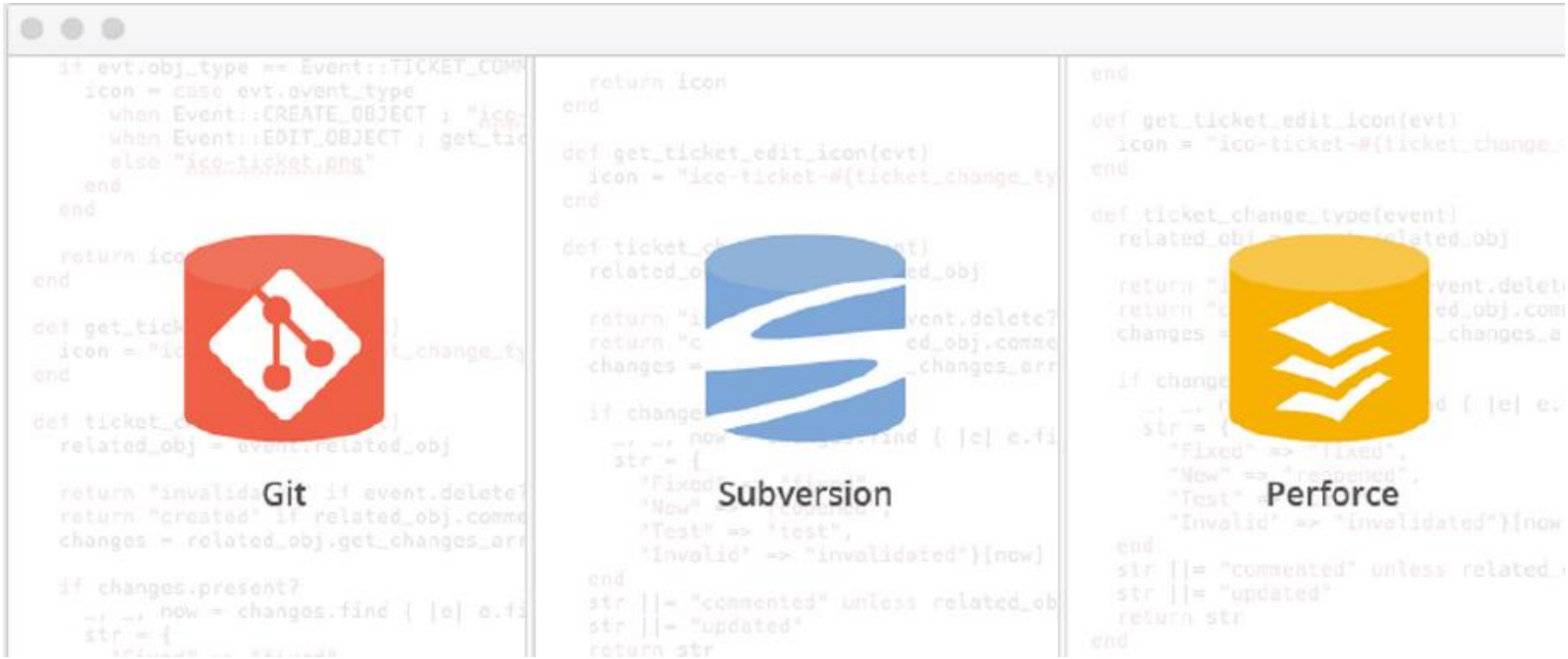
- Eclipse
- PyCharm
- Jupyter Notebook
- Visual Studio
- XCode



버전 제어 시스템 (VCS)

Version Control Systems

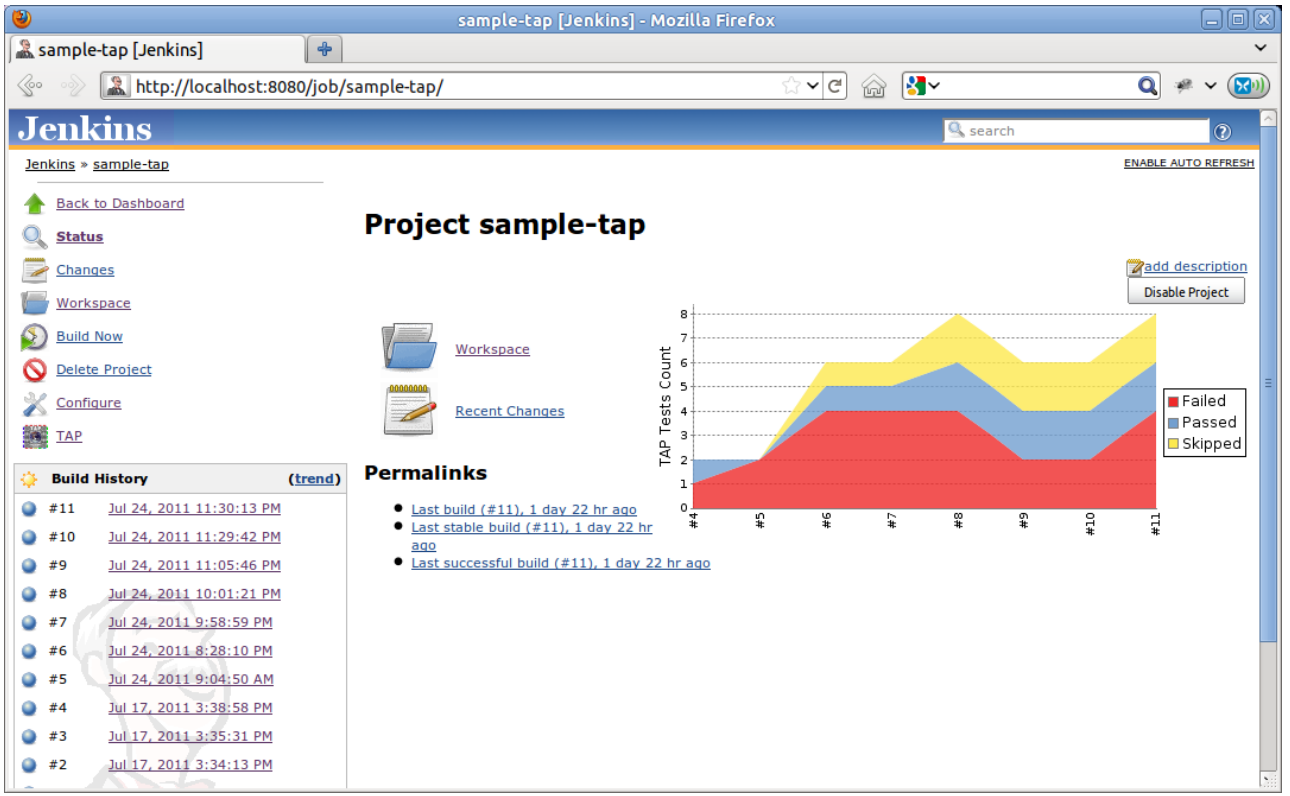
- 소프트웨어의 형상 (configuration) 을 추적, 관리하기 위해 이용하는 시스템



지속적 통합 도구

Continuous Integration

- 지속적으로 소프트웨어의 품질 관리 활동을 적용하는 프로세스의 실행



생각해 볼 과제

- 운영체제란 무엇이며, 어떤 역할을 하는가?
- 소프트웨어의 개발 과정에는 어떤 활동들이 포함될까?
- 아래 프로그램 번역/실행 모델은 어느 언어에서 주로 이용할까?
  - 컴파일러 (compiler)
  - 인터프리터 (interpreter)
- 좋은 소프트웨어 개발자가 갖추어야 할 역량은 무엇일까?

지금까지 배운 것들 정리해봅시다!