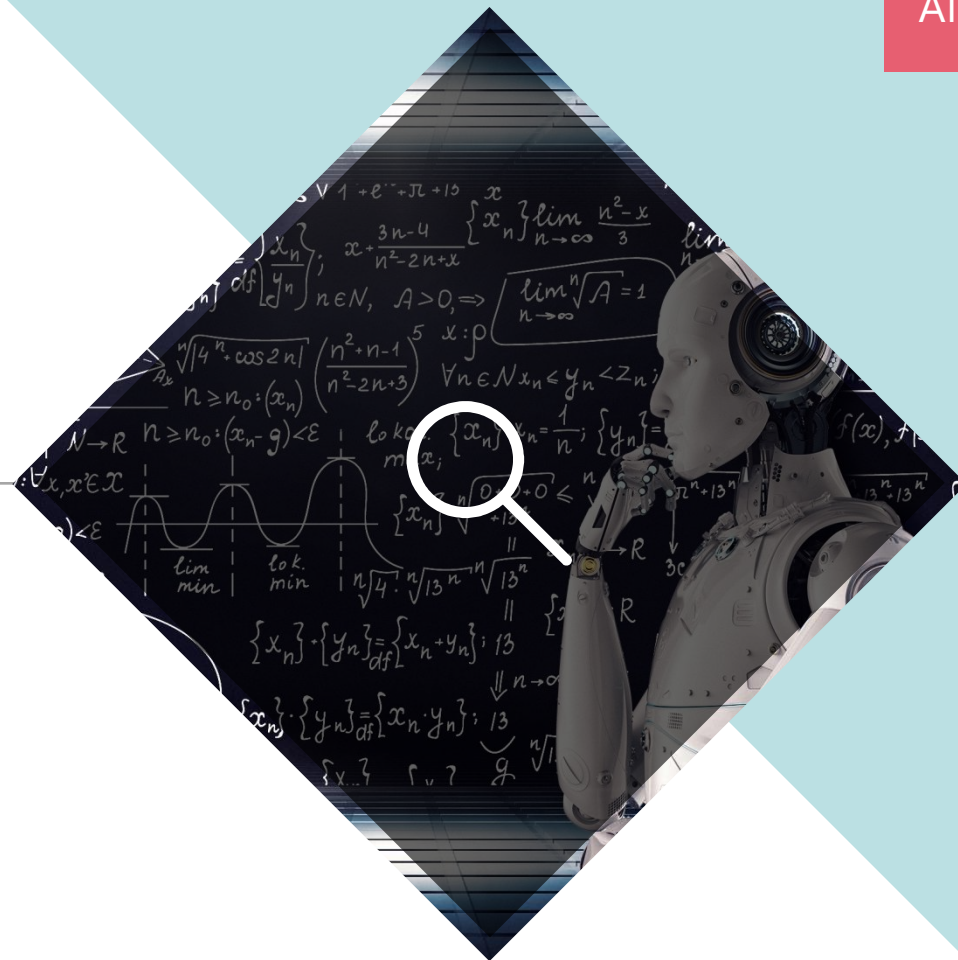
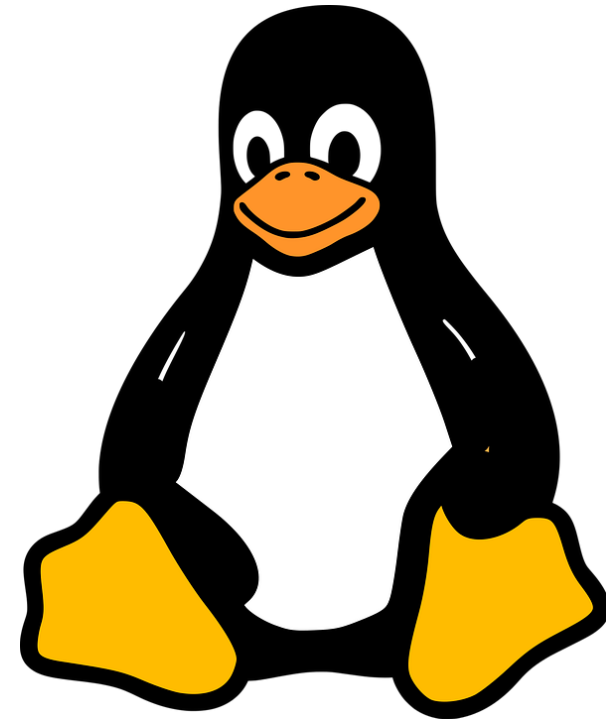


Kernel - 001



Contents

- I. 컴퓨터 구조
- II. 운영체제(OS)
- III. 커널이란?
- IV. Ubuntu Kernel 빌드/사용하기
- V. GRUB / DMESG



I. 컴퓨터 구조

컴퓨터 구조를 알아야 하는 이유



프로그래밍 언어 뿐 아니라
컴퓨터의 근간을 알아야 한다



- 문제 해결능력
- 성능 / 용량 / 비용

I. 컴퓨터 구조

컴퓨터 구조를 알아야 하는 이유 (cont.)



문법에 맞는 소스 코드를
컴퓨터에 입력만 하는 개발자

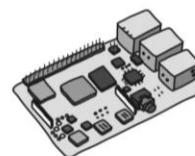


컴퓨터를 관조하며 문제를
해결할 수 있는 개발자

I. 컴퓨터 구조

컴퓨터 구조를 알아야 하는 이유 (cont.)

컴퓨터 구조는 결국
성능, 용량, 비용에 대한 이야기



작은 컴퓨터



스마트폰



노트북



서버 컴퓨터

세상에 존재하는
다양한 종류의 컴퓨터

<input type="radio"/>	r6a.large	2	x86_64	16	Up to 12.5 Gigabit	0.1134 USD per Hour	0.2054 USD per Hour
<input type="radio"/>	r6a.xlarge	4	x86_64	32	Up to 12.5 Gigabit	0.2268 USD per Hour	0.4108 USD per Hour
<input type="radio"/>	r6a.2xlarge	8	x86_64	64	Up to 12.5 Gigabit	0.4536 USD per Hour	0.8216 USD per Hour
<input type="radio"/>	r6a.4xlarge	16	x86_64	128	Up to 12.5 Gigabit	0.9072 USD per Hour	1.6432 USD per Hour
<input type="radio"/>	r6a.8xlarge	32	x86_64	256	12.5 Gigabit	1.8144 USD per Hour	3.2864 USD per Hour
<input type="radio"/>	r6a.12xlarge	48	x86_64	384	18.75 Gigabit	2.7216 USD per Hour	4.9296 USD per Hour

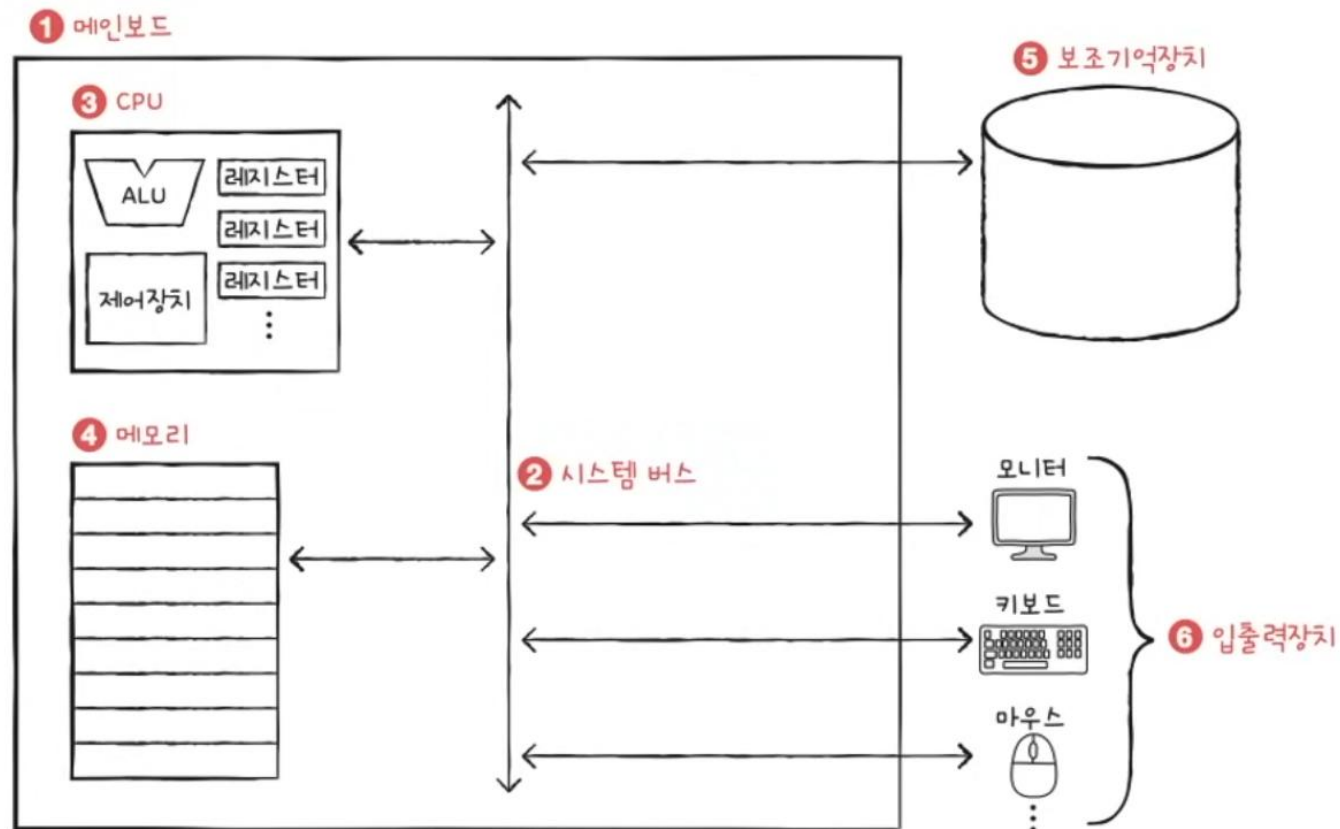
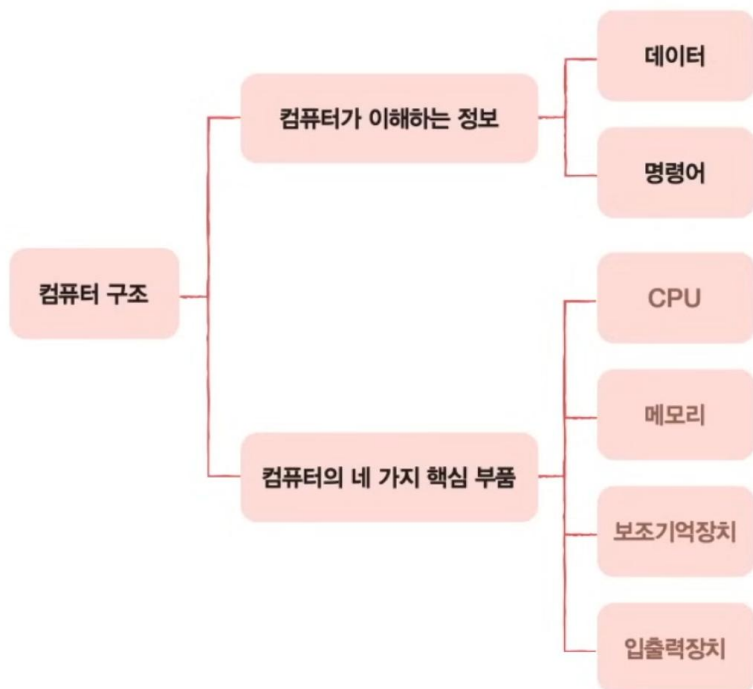
I. 컴퓨터 구조

컴퓨터 구조 (4가지 핵심 부품)



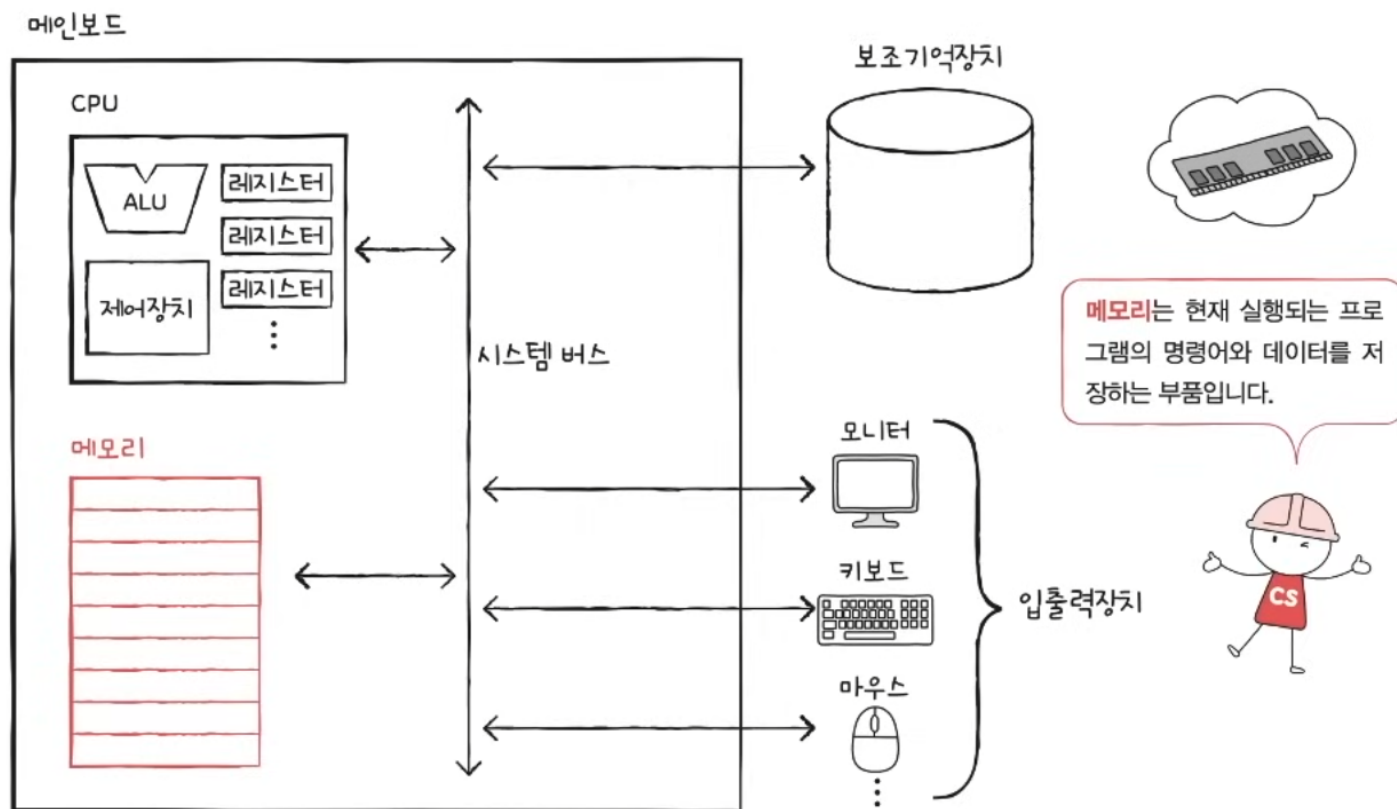
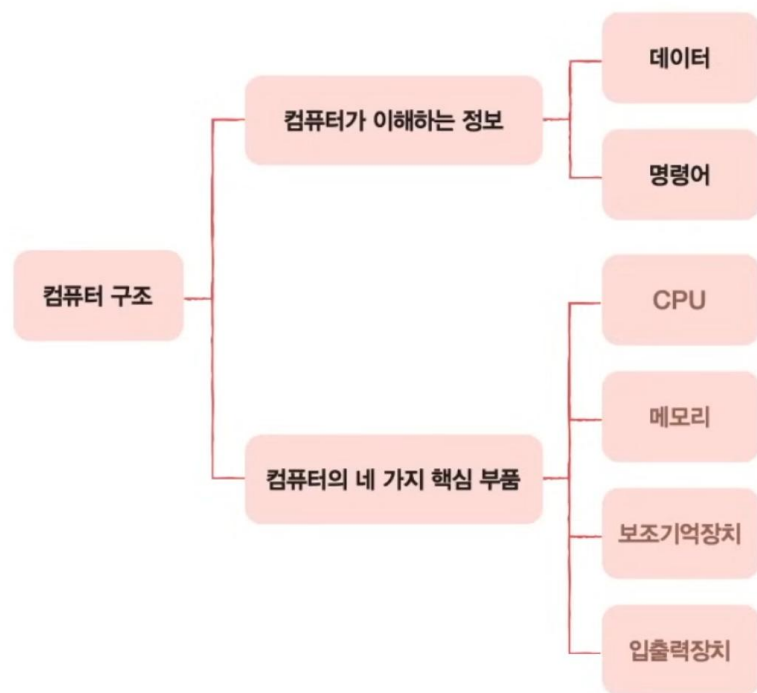
I. 컴퓨터 구조

컴퓨터 구조 (4가지 핵심 부품)



I. 컴퓨터 구조

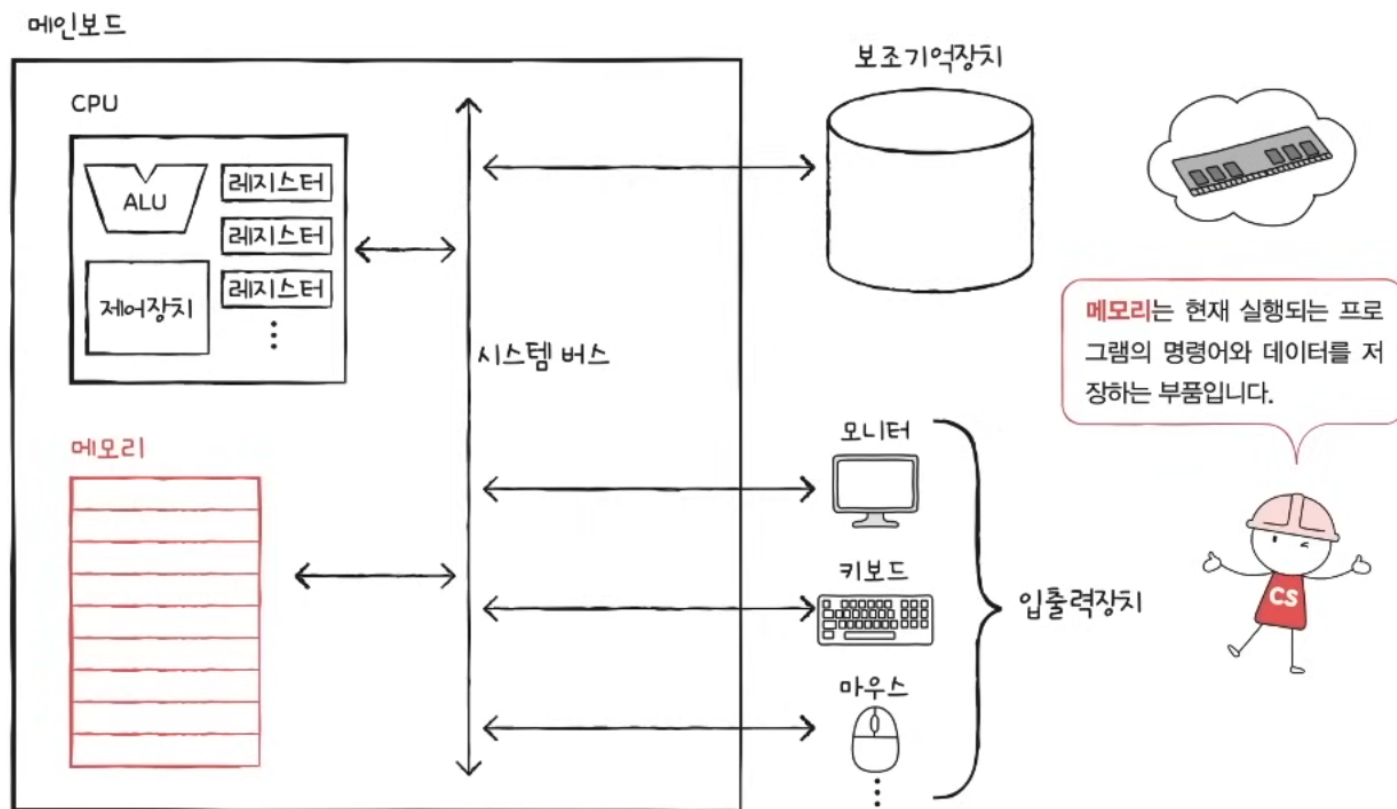
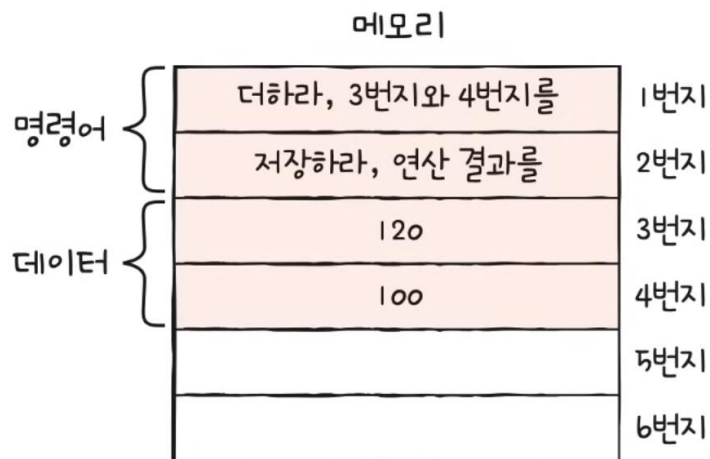
컴퓨터 구조 (4가지 핵심 부품 - 메모리)



I. 컴퓨터 구조

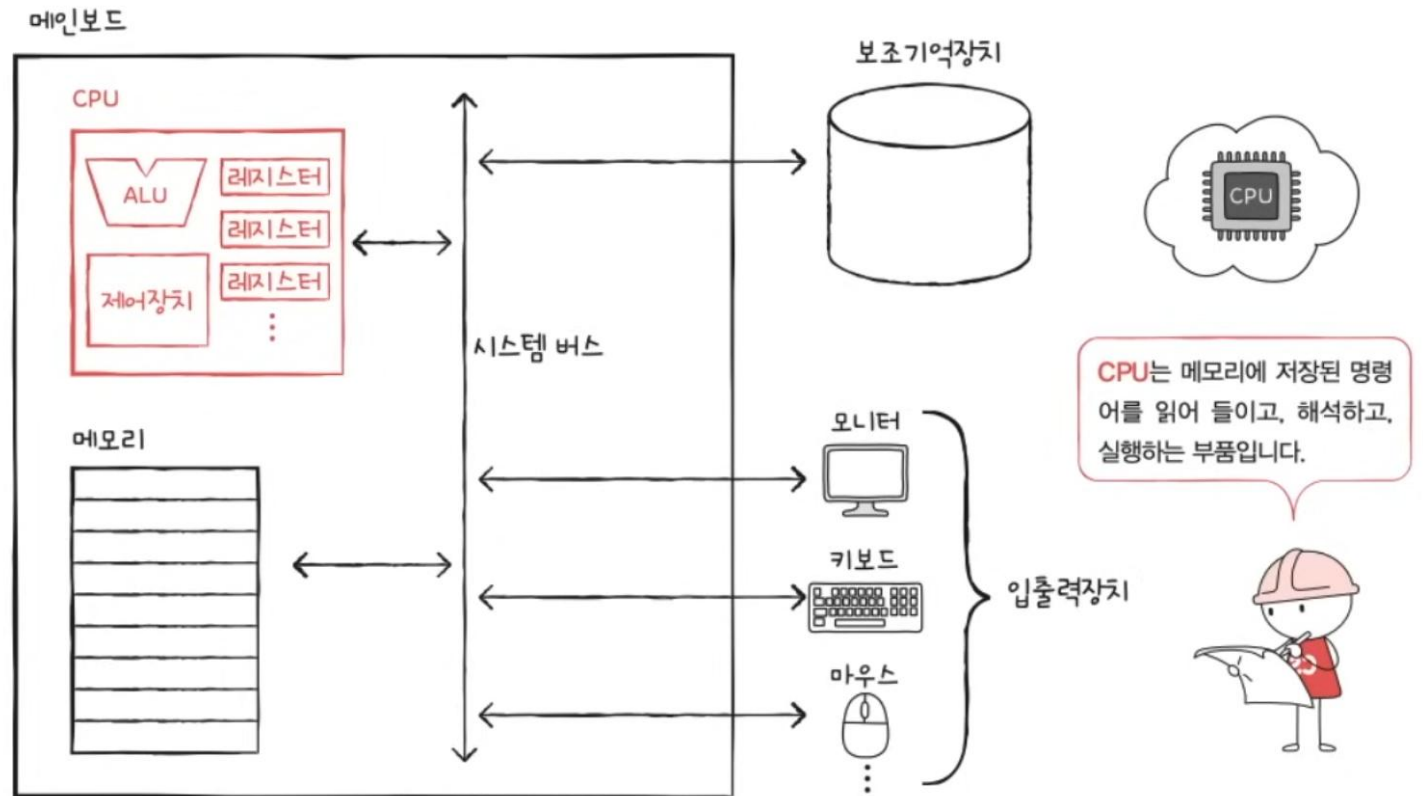
컴퓨터 구조 (4가지 핵심 부품 - 메모리)

- 메모리는 실행되는 프로그램의 명령어와 데이터를 저장한다
- 프로그램이 실행되려면 메모리에 저장되어 있어야 한다
- 메모리에 저장된 값의 위치는 주소로 알 수 있다



I. 컴퓨터 구조

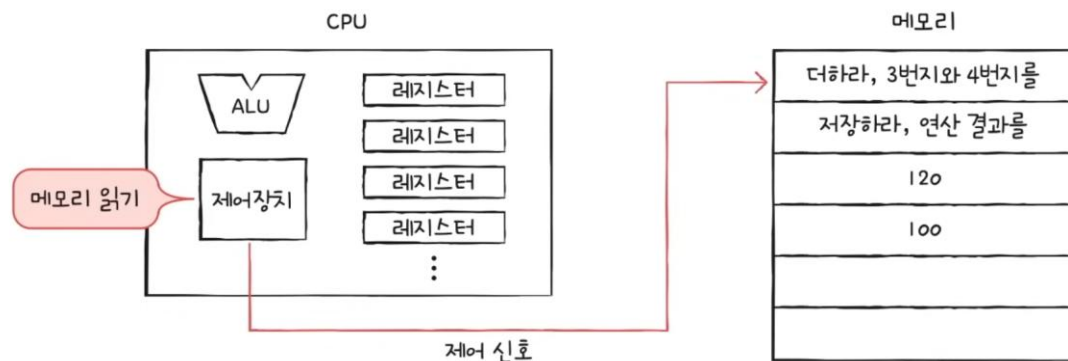
컴퓨터 구조 (4가지 핵심 부품 - CPU)



I. 컴퓨터 구조

컴퓨터 구조 (4가지 핵심 부품 - CPU)

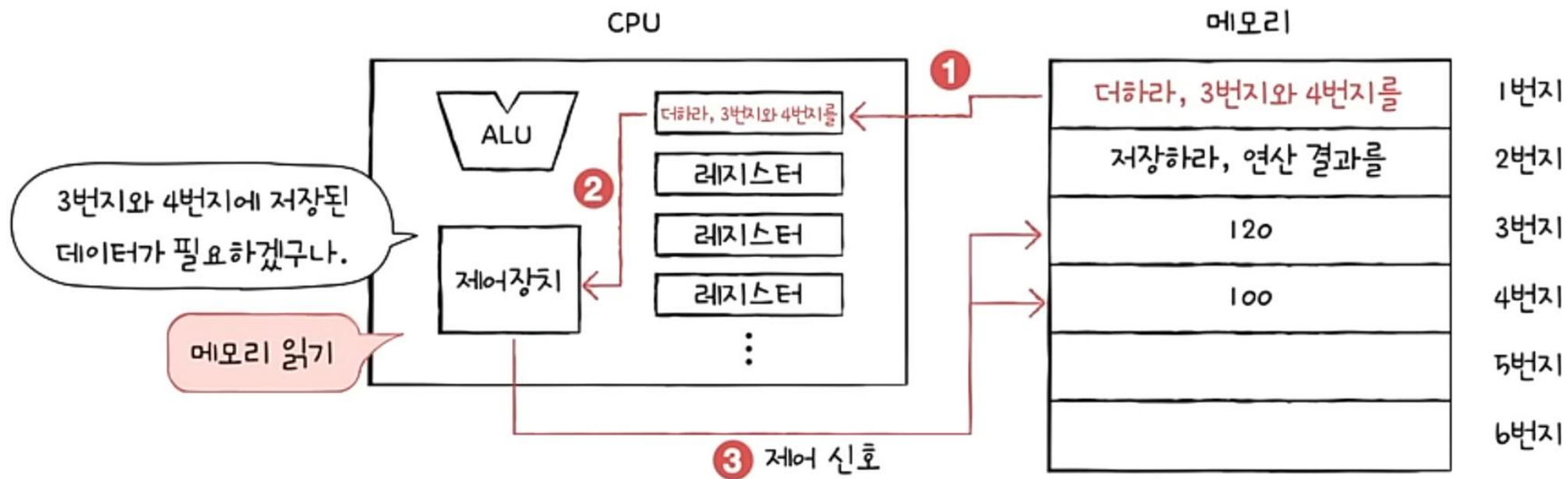
- CPU는 메모리에 저장된 값을 읽어 들이고, 해석하고, 실행하는 장치다
- CPU 내부에는 ALU, 레지스터, 제어장치가 있다
 - ALU는 계산하는 장치
 - 레지스터는 임시 저장 장치
 - 제어장치는 제어 신호를 발생시키고 명령어를 해석하는 장치



I. 컴퓨터 구조

컴퓨터 구조 (4가지 핵심 부품 - CPU)

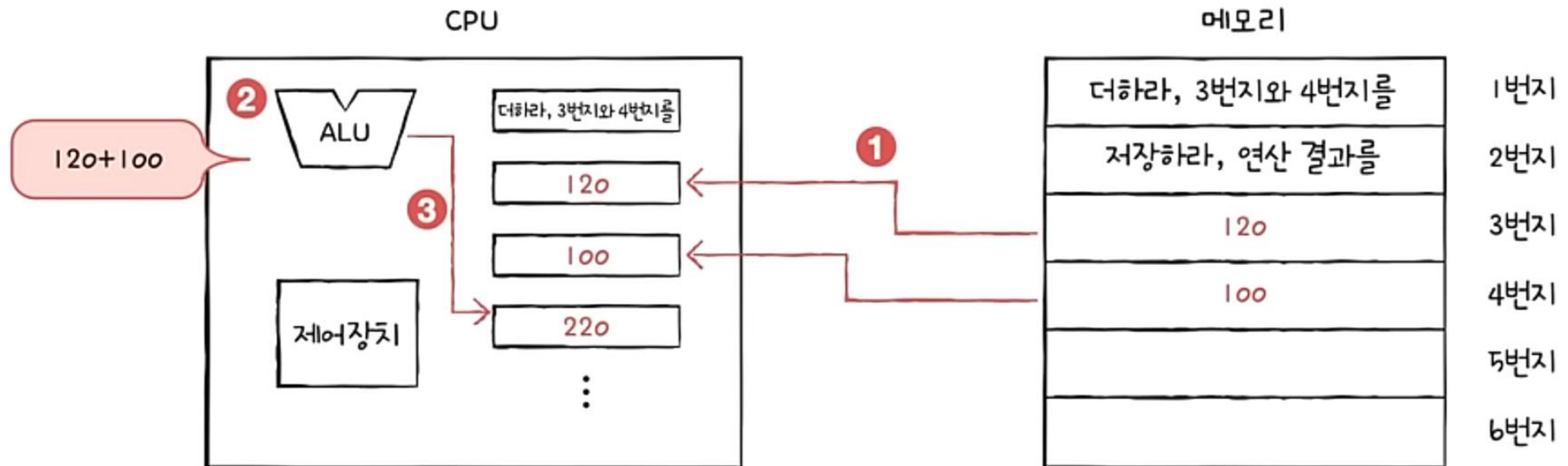
“ $120 + 100 = ?$ ” 이 CPU 에서 처리되는 과정



I. 컴퓨터 구조

컴퓨터 구조 (4가지 핵심 부품 - CPU)

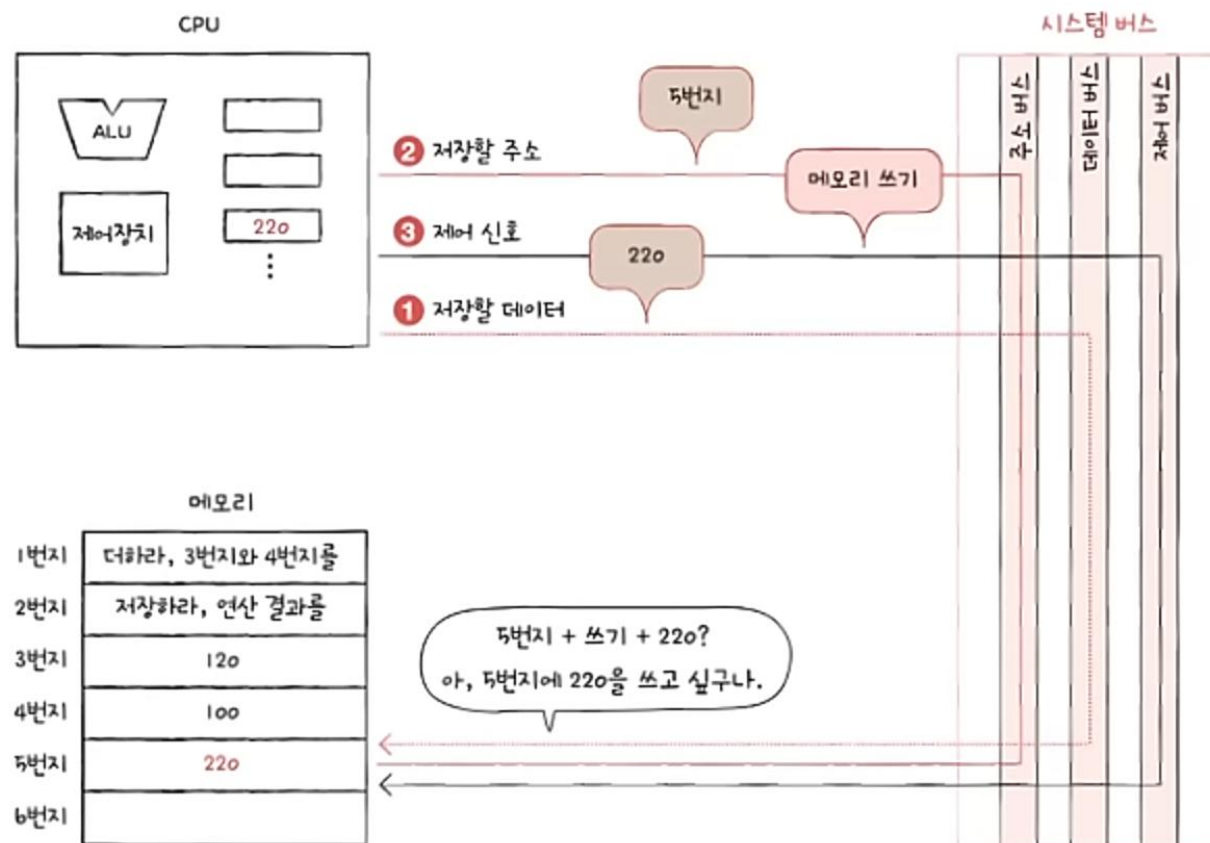
“ $120 + 100 = ?$ ” 이 CPU 에서 처리되는 과정



I. 컴퓨터 구조

컴퓨터 구조 (4가지 핵심 부품 - CPU)

“120 + 100 = ?” 이 CPU 에서 처리되는 과정



I. 컴퓨터 구조

컴퓨터 구조 (4가지 핵심 부품 - CPU)

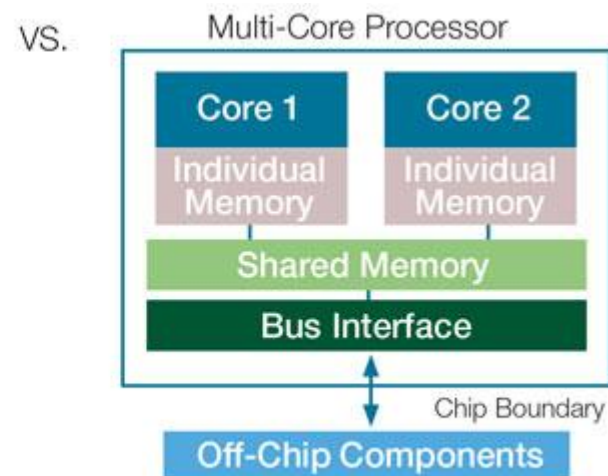
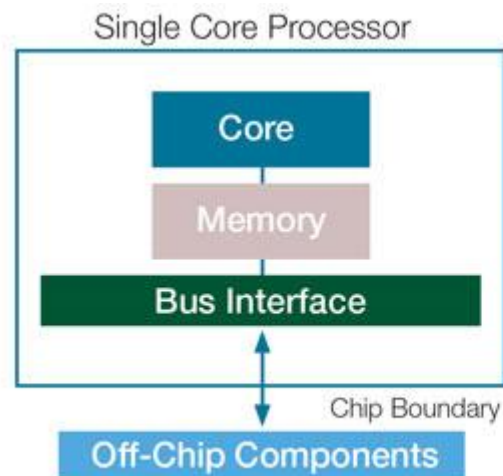
Multi-Core? - 하나의 CPU 칩 안에 2개 이상의 코어(연산 장치)를 내장한 구조

싱글 코어 CPU가 하나의 코어로 작업을 처리하는 반면, 멀티 코어 CPU는 여러 코어가 병렬로 작업을 처리하여 동시에 여러 작업을 수행하거나 더 복잡한 작업을 효율적으로 처리함

듀얼 코어 (Dual-core): 2개의 코어를 가진 CPU.

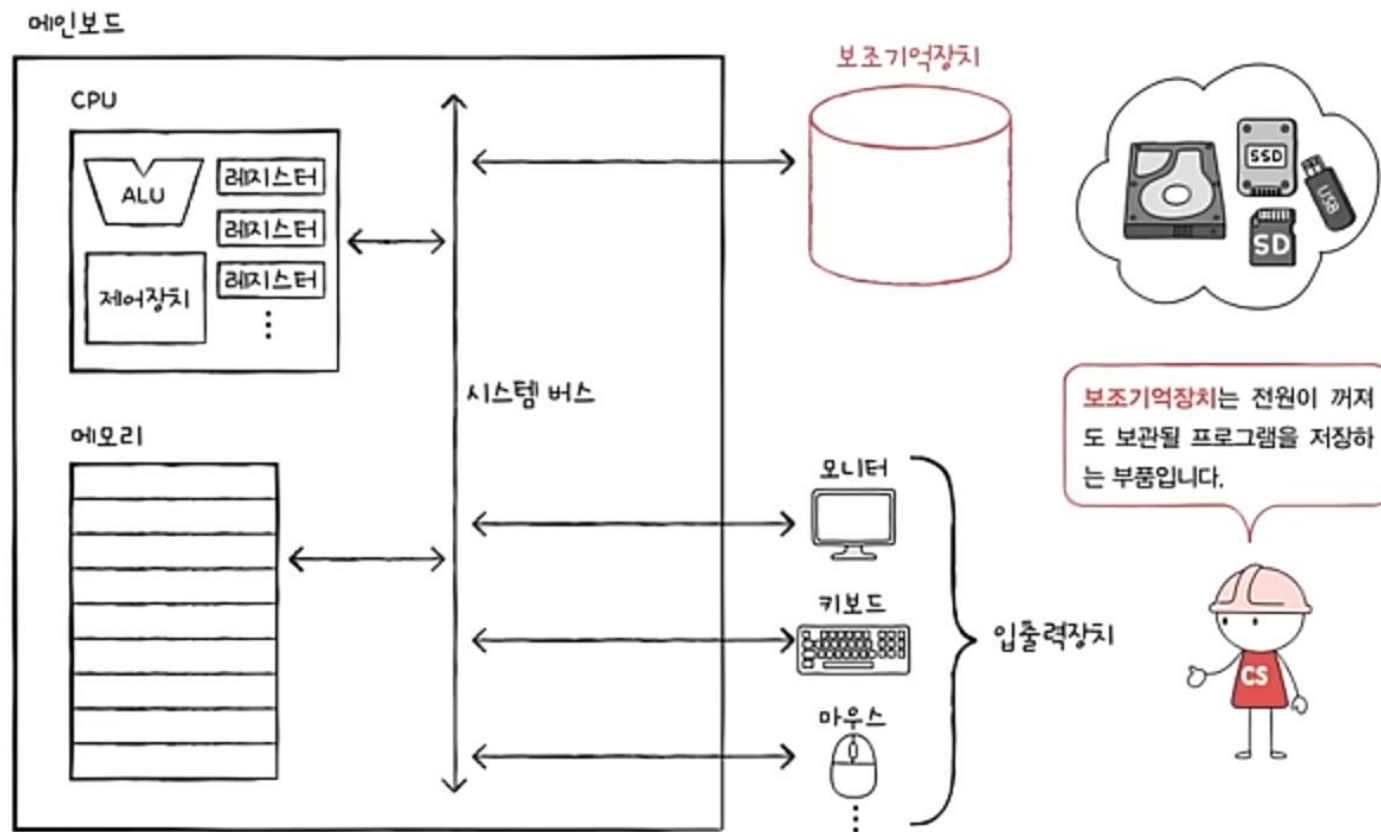
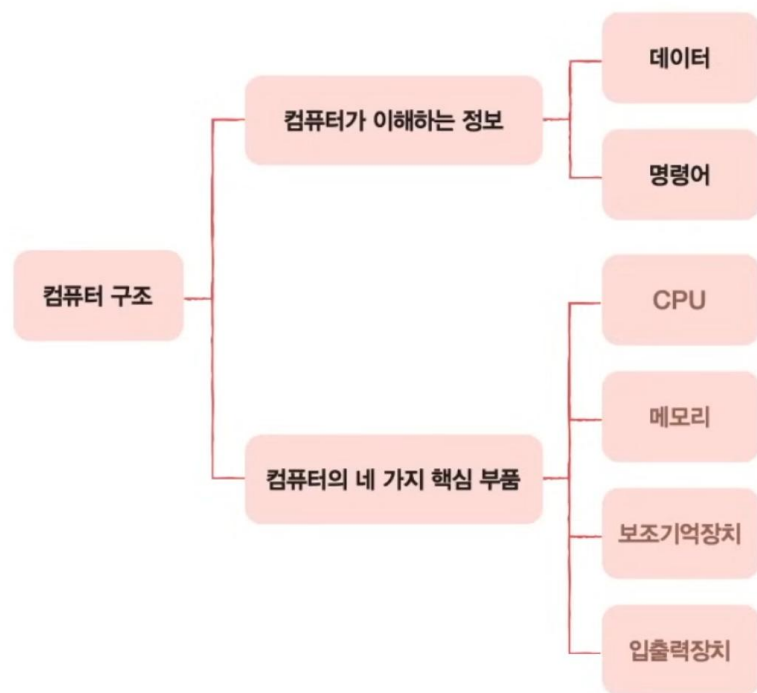
쿼드 코어 (Quad-core): 4개의 코어를 가진 CPU.

옥타 코어 (Octa-core): 8개의 코어를 가진 CPU.



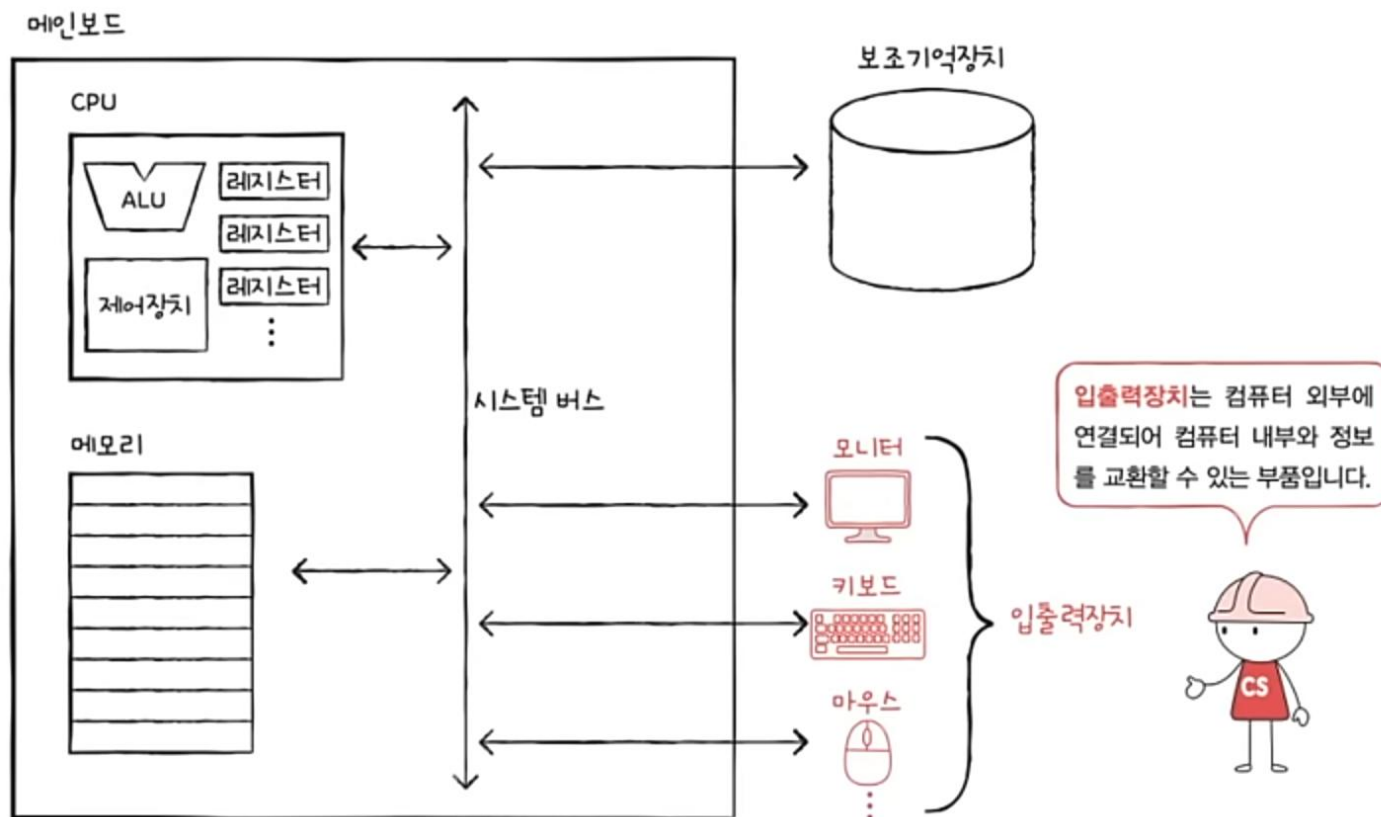
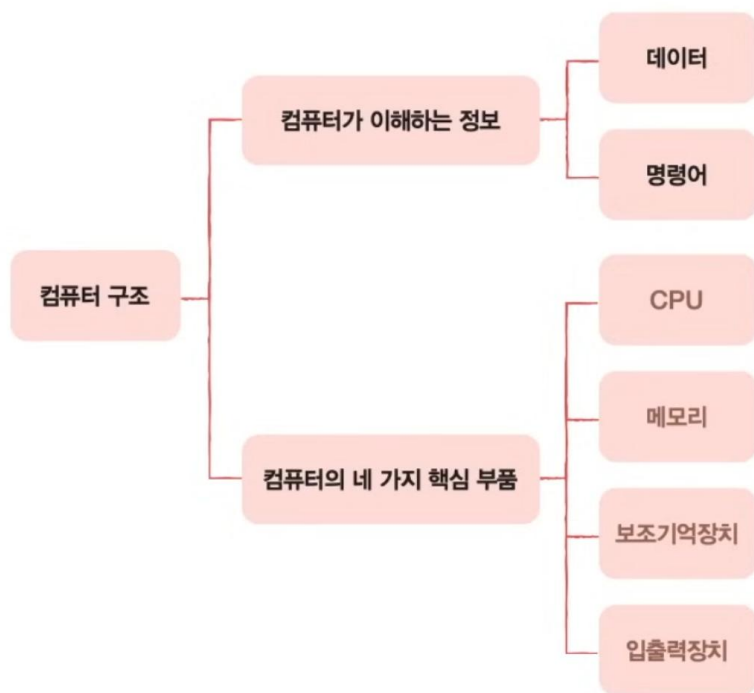
I. 컴퓨터 구조

컴퓨터 구조 (4가지 핵심 부품 - 보조기억장치)



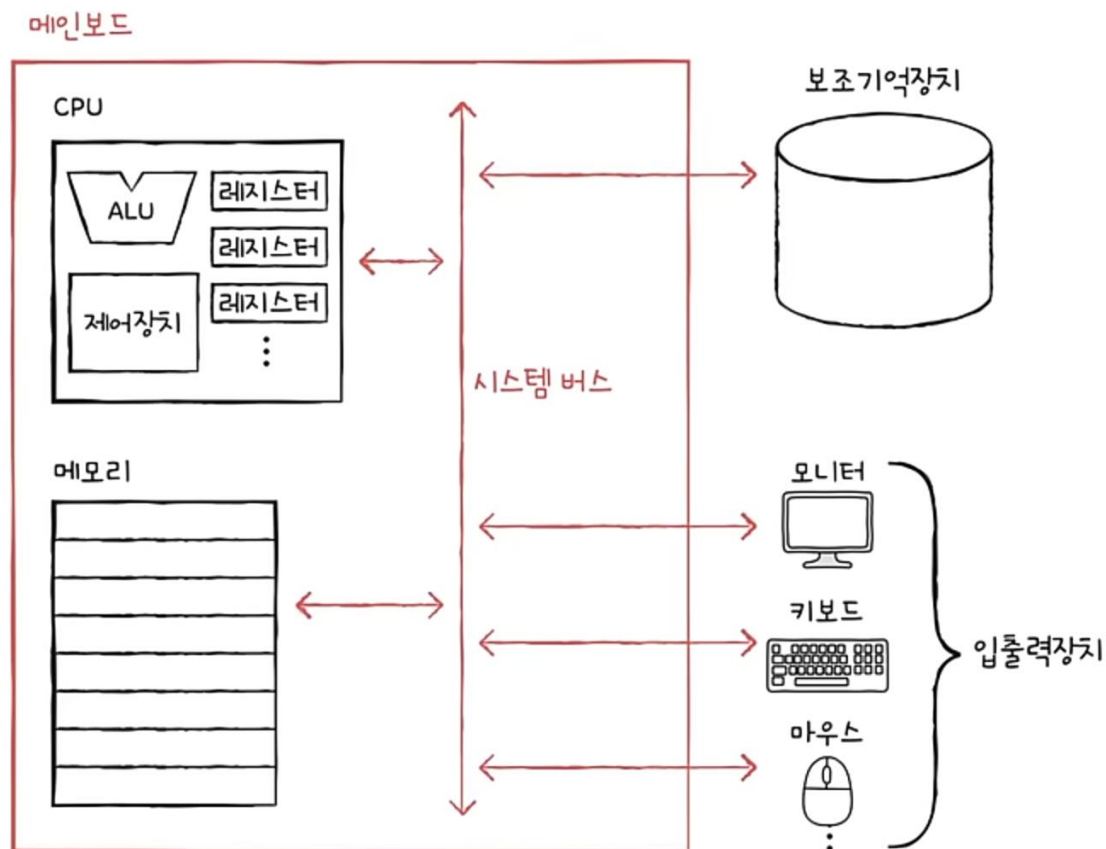
I. 컴퓨터 구조

컴퓨터 구조 (4가지 핵심 부품 - 입출력 장치)



I. 컴퓨터 구조

컴퓨터 구조 (메인 보드 & 시스템 버스)



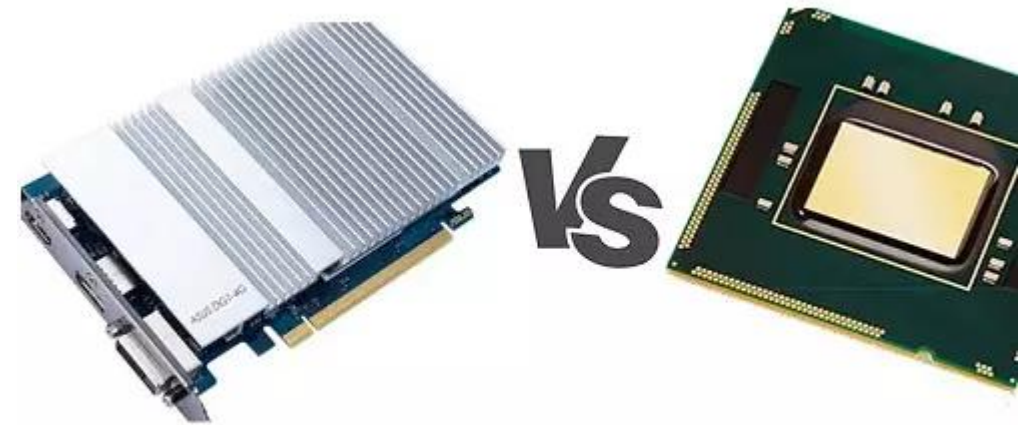
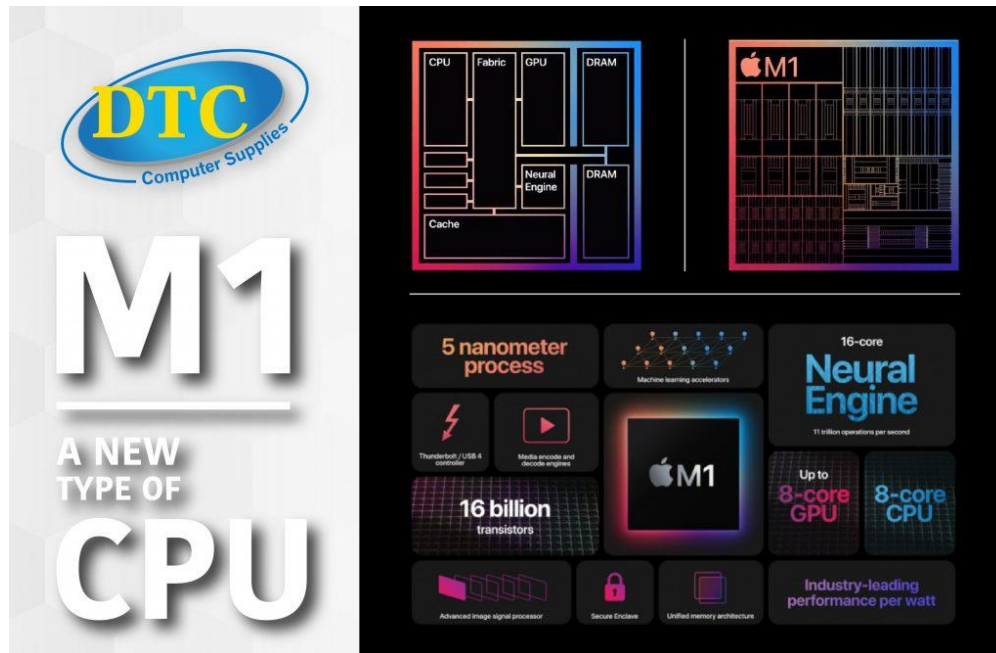
메인보드 & (시스템) 버스

- 메인보드에 연결된 부품은 **버스**를 통해 정보를 주고 받음
- 버스는 컴퓨터의 부품끼리 정보를 주고받는 일종의 **통로**
- 다양한 종류의 버스가 있음
- 컴퓨터의 핵심 부품을 연결하는 버스는 **시스템 버스**

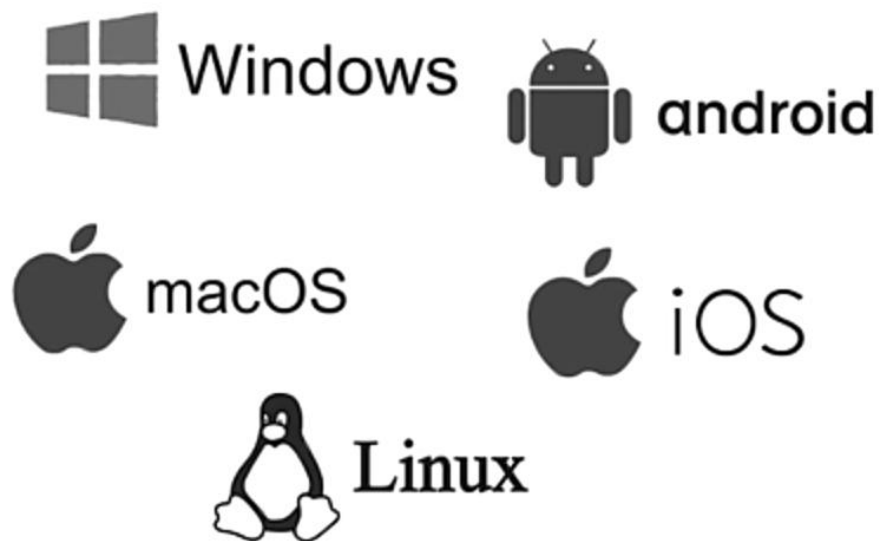
I. 컴퓨터 구조

컴퓨터 구조 (GPU)

Graphics Processing Unit - 다양한 장치에서 컴퓨터 그래픽 및 이미지 처리 속도를 높이도록 설계된 전자 회로, 내장 GPU 와 외장 GPU (discrete GPU) 로 크게 나뉨



II. 운영체제 (OS)



운영체제는 무엇이고,
개발자는 왜 운영체제를
알아야 할까?



II. 운영체제 (OS)

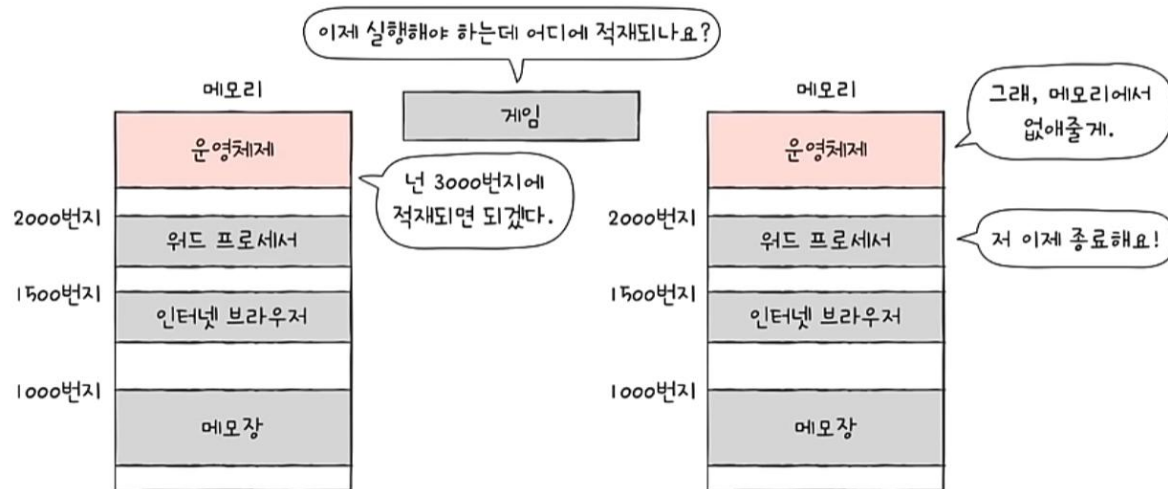
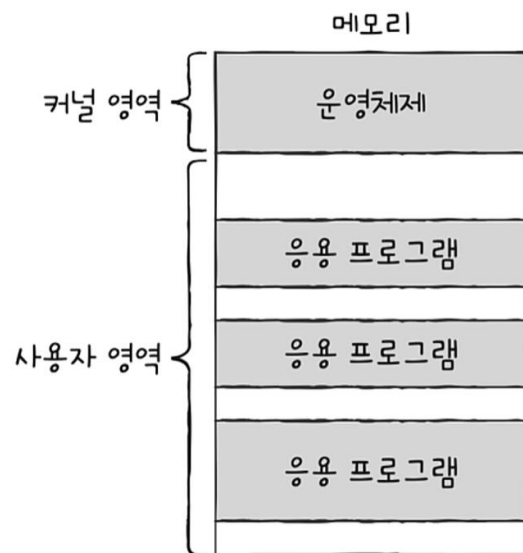
자원을 관리하는 특별한 프로그램

프로세스를 관리하는 특별한 프로그램

(시스템) 자원

- 프로그램이 실행되기 위해 마땅히 필요한 요소
- 컴퓨터의 네 가지 핵심 부품 포함

운영체제의 자원(메모리) 관리

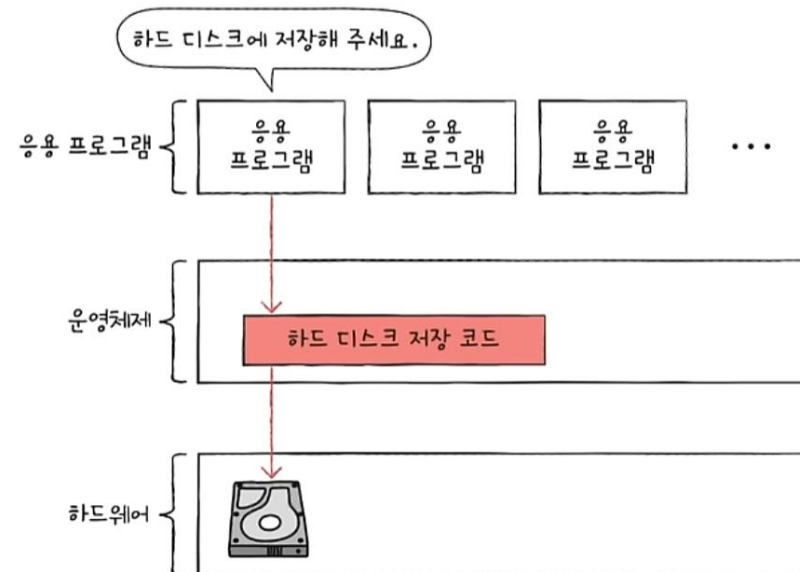
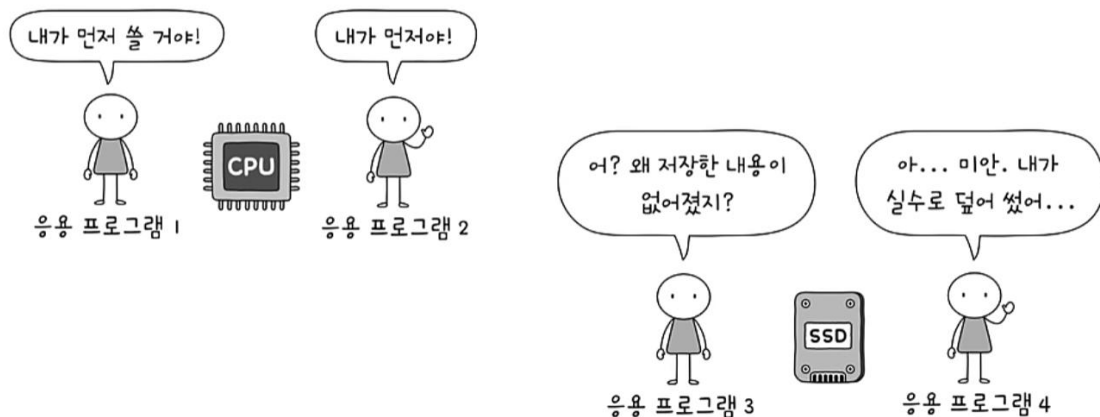


II. 운영체제 (OS)

자원을 관리하는 특별한 프로그램

프로세스를 관리하는 특별한 프로그램

문지기 역할(시스템 호출)을 통한 자원 보호

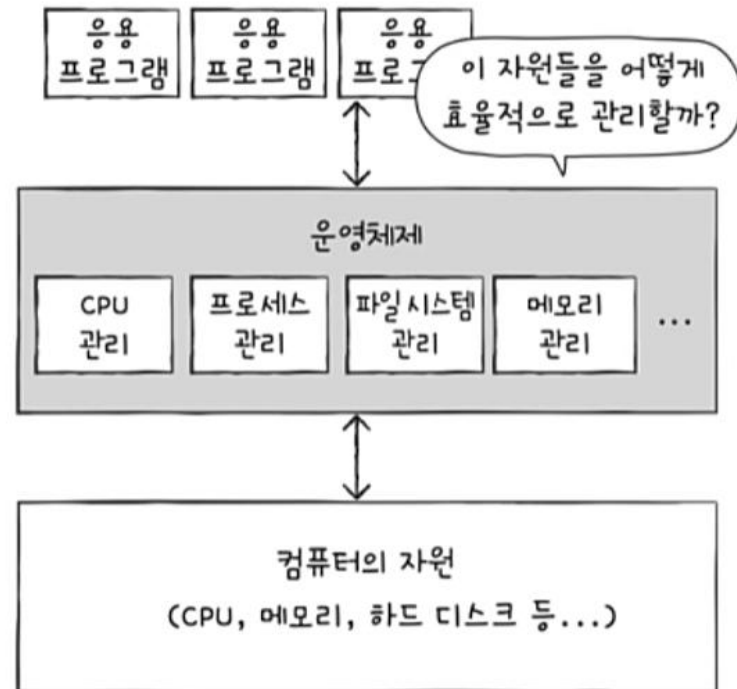
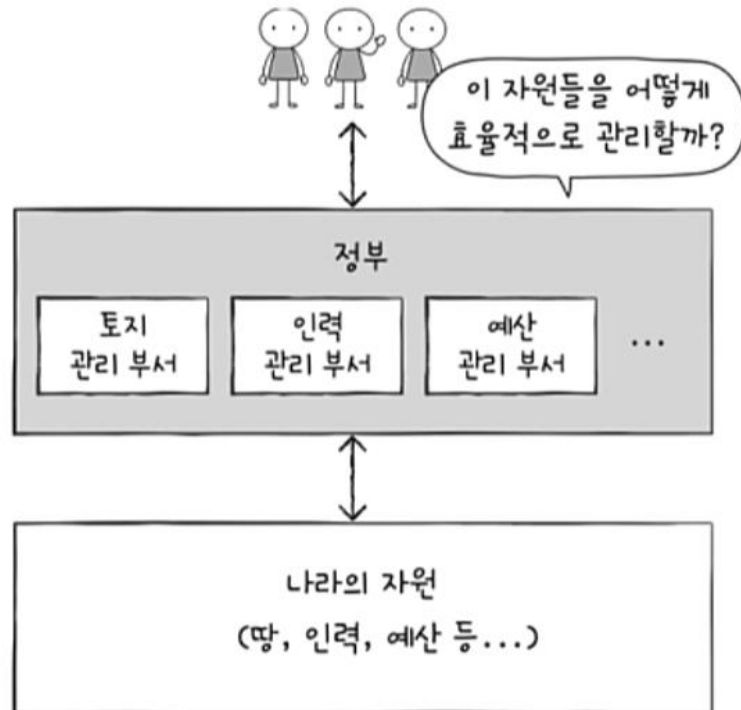


II. 운영체제 (OS)

자원을 관리하는 특별한 프로그램

프로세스를 관리하는 특별한 프로그램

마치 정부와도 같은..

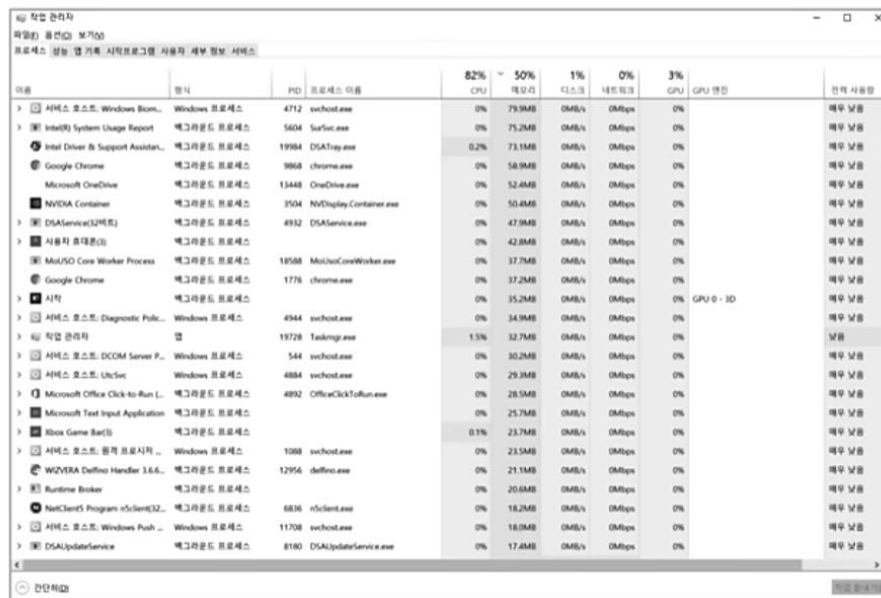


II. 운영체제 (OS)

자원을 관리하는 특별한 프로그램

프로세스를 관리하는 특별한 프로그램

Q. 이렇게 많은 프로그램들이 동시에 실행되는데,
누가 일목요연하게 실행을 관리해주지?



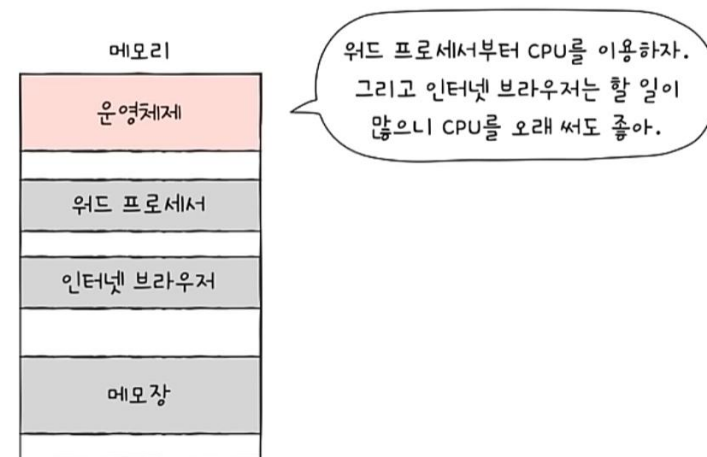
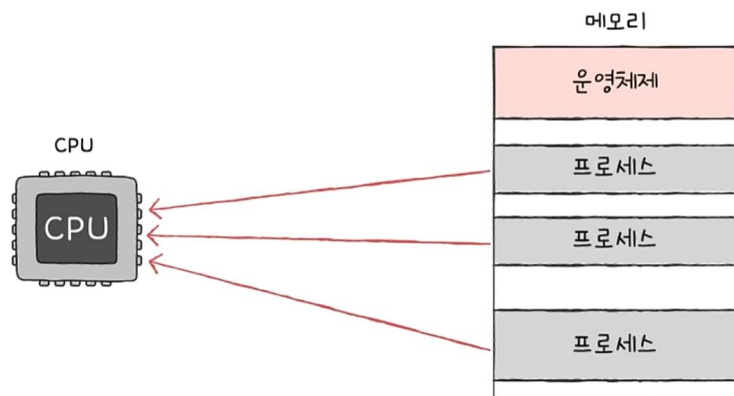
이름	원격	PID	프로세스 이름	82% CPU	50% 메모리	1% 디스크	0% 네트워크	3% GPU	GPU 엔진	현재 사용량
> 서비스 호스트: Windows Biom...	Windows 프로세스	4712	svchost.exe	0%	79.9MB	0MB/s	0Mbps	0%		매우 낮음
> Intel(R) System Usage Report	백그라운드 프로세스	5604	SurSvc.exe	0%	75.2MB	0MB/s	0Mbps	0%		매우 낮음
Intel Driver & Support Assistan...	백그라운드 프로세스	19984	DSATray.exe	0.2%	71.1MB	0MB/s	0Mbps	0%		매우 낮음
Google Chrome	백그라운드 프로세스	9868	chrome.exe	0%	58.9MB	0MB/s	0Mbps	0%		매우 낮음
Microsoft OneDrive	백그라운드 프로세스	13488	OneDrive.exe	0%	52.8MB	0MB/s	0Mbps	0%		매우 낮음
NVIDIA Container	백그라운드 프로세스	3704	NVDisplayContainer.exe	0%	50.8MB	0MB/s	0Mbps	0%		매우 낮음
DSAService(32비트)	백그라운드 프로세스	4932	DSAService.exe	0%	47.9MB	0MB/s	0Mbps	0%		매우 낮음
사용자 휴대전화	백그라운드 프로세스			0%	42.8MB	0MB/s	0Mbps	0%		매우 낮음
MALWARE Core Worker Process	백그라운드 프로세스	18588	MalwareCoreWorker.exe	0%	37.7MB	0MB/s	0Mbps	0%		매우 낮음
Google Chrome	백그라운드 프로세스	1776	chrome.exe	0%	37.2MB	0MB/s	0Mbps	0%		매우 낮음
시작	백그라운드 프로세스			0%	35.2MB	0MB/s	0Mbps	0%	GPU 0 - 3D	매우 낮음
서비스 호스트: Diagnostic Polic...	Windows 프로세스	4944	svchost.exe	0%	34.9MB	0MB/s	0Mbps	0%		매우 낮음
작업 관리자	임	19728	Taskmgr.exe	1.5%	32.7MB	0MB/s	0Mbps	0%		낮음
서비스 호스트: DCOM Server P...	Windows 프로세스	544	svchost.exe	0%	30.2MB	0MB/s	0Mbps	0%		매우 낮음
서비스 호스트: UtcSvc	Windows 프로세스	4884	svchost.exe	0%	29.3MB	0MB/s	0Mbps	0%		매우 낮음
Microsoft Office Click-to-Run L...	백그라운드 프로세스	4892	OfficeClickToRun.exe	0%	28.5MB	0MB/s	0Mbps	0%		매우 낮음
Microsoft Text Input Application	백그라운드 프로세스			0%	25.7MB	0MB/s	0Mbps	0%		매우 낮음
Xbox Game Bar()	백그라운드 프로세스			0.1%	23.7MB	0MB/s	0Mbps	0%		매우 낮음
서비스 호스트: 원격 프로시저 ...	Windows 프로세스	1088	svchost.exe	0%	23.5MB	0MB/s	0Mbps	0%		매우 낮음
WIZVERA Dellino Handler 3.6.6...	백그라운드 프로세스	12956	dellino.exe	0%	21.1MB	0MB/s	0Mbps	0%		매우 낮음
Runtime Broker	백그라운드 프로세스			0%	20.6MB	0MB/s	0Mbps	0%		매우 낮음
NetClient5 Program xSClient32...	백그라운드 프로세스	6836	xSClient.exe	0%	18.2MB	0MB/s	0Mbps	0%		매우 낮음
서비스 호스트: Windows Push ...	Windows 프로세스	11708	svchost.exe	0%	18.0MB	0MB/s	0Mbps	0%		매우 낮음
DSASUpdateService	백그라운드 프로세스	8180	DSASUpdateService.exe	0%	17.6MB	0MB/s	0Mbps	0%		매우 낮음

II. 운영체제 (OS)

자원을 관리하는 특별한 프로그램

프로세스를 관리하는 특별한 프로그램

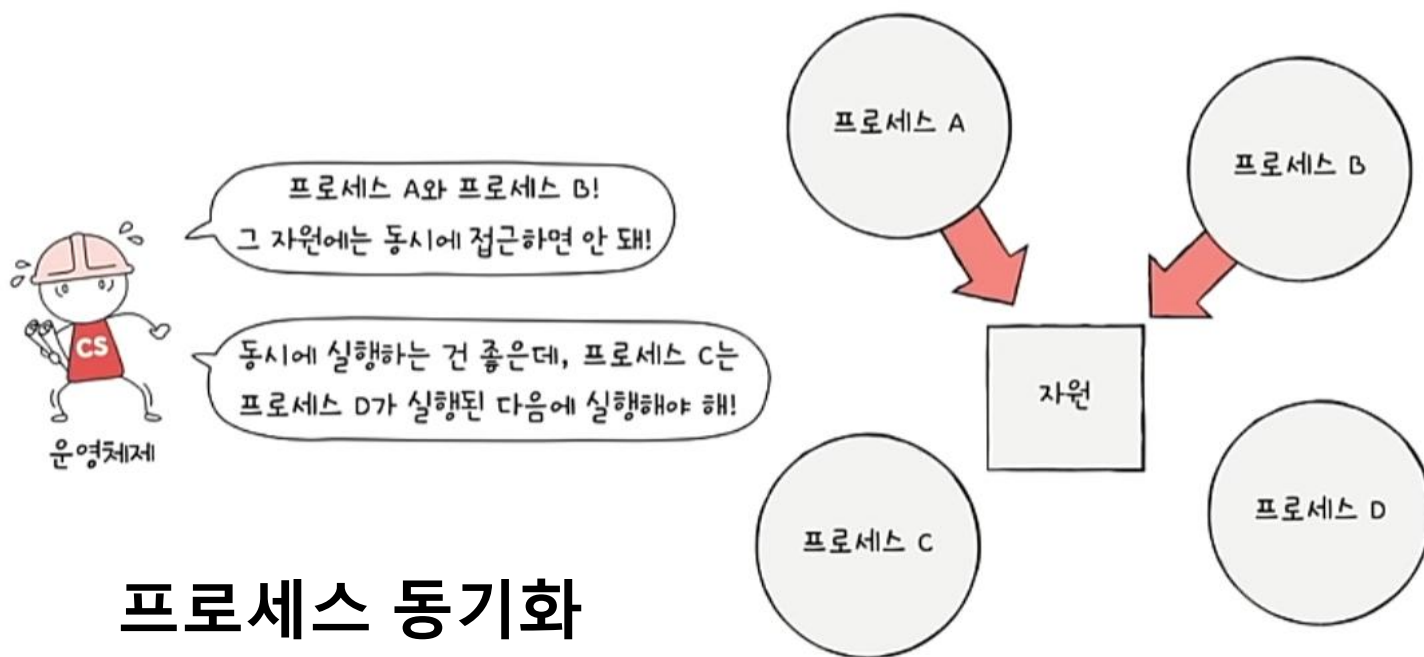
Q. 어떤 프로세스를 먼저, 얼마나 오래 실행할까?



II. 운영체제 (OS)

자원을 관리하는 특별한 프로그램

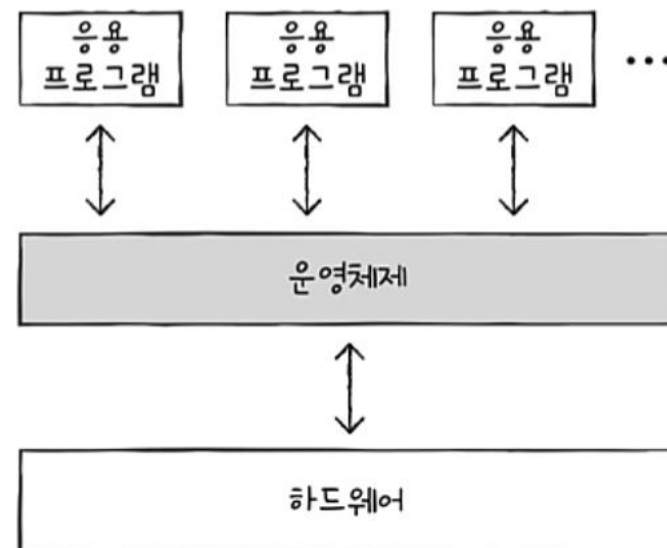
프로세스를 관리하는 특별한 프로그램



II. 운영체제 (OS)

정리

- 운영체제는 컴퓨터를 동작하는데 도움을 주는 SW
- HW 와 SW 사이의 매개체 역할
- 운영체제의 역할
 - 사용자를 위한 역할: 사용자에게 컴퓨터를 손쉽게 이용할 수 있는 인터페이스 제공
 - 하드웨어를 위한 역할:
 - 프로세스 관리 - 컴퓨터에서 프로세스들을 연속적으로 실행 가능하게
 - 메모리 관리 - 한정된 메모리 공간에 프로그램 메모리 할당
 - HW 관리 - 사용자의 직접 접근 저지, 파일 효율적으로 관리
- 운영체제의 구조
 - Kernel: 운영체제의 가장 핵심적인 부분으로 프로세스, 메모리, 저장장치 관리로 커널 자체를 좁은 의미에서의 OS 라고 부르기도 함
 - 인터페이스: 사용자가 커널에 직접 접근 할 수 없어 인터페이스(System Call) 를 통해 커널에 명령을 전달하고 결과를 전달받음



III. Kernel

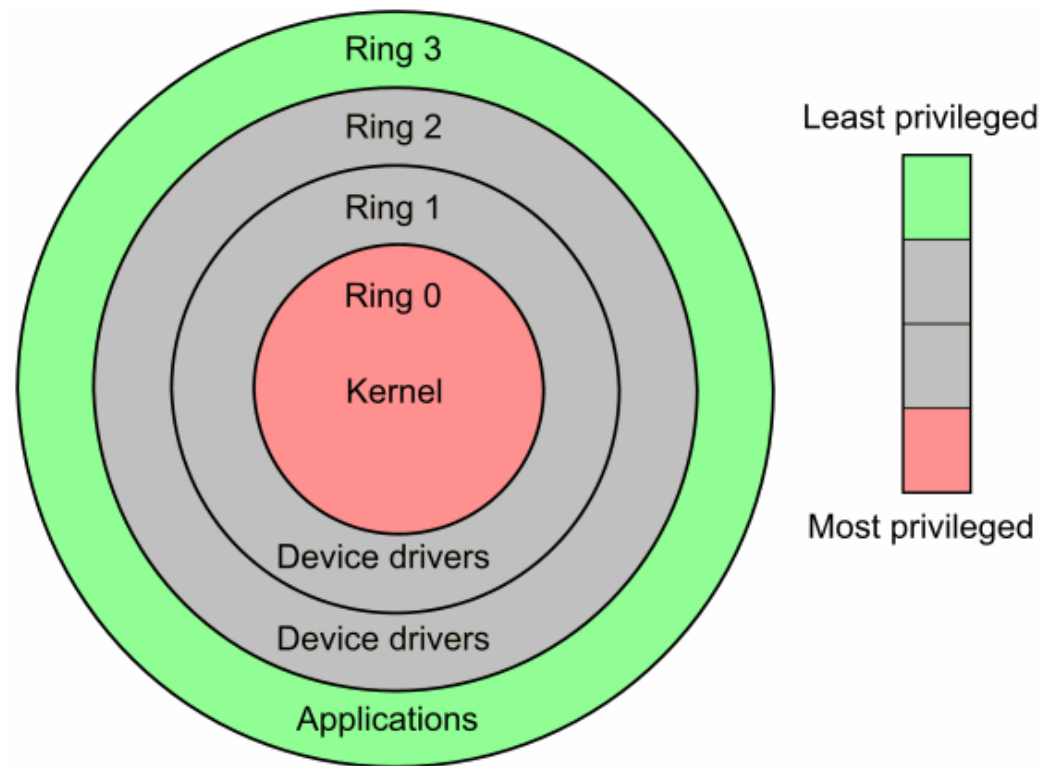
Kernel 이란?

- 커널(Kernel)은 운영체제의 핵심 부분으로, 하드웨어와 직접 상호작용하며 OS의 주요 기능들을 담당하는 중앙 관리 프로그램
- 운영체제 전체가 집이라면 커널은 그 토대(핵심)에 해당하며, 응용프로그램이 요청하는 저 수준 작업 들을 수행
- 커널은 일반적으로 커널 모드에서 동작하며, 사용자 프로그램은 유저 모드에서 동작하여 커널에 직접 접근하지 못하게 격리
- 커널의 역할:
 - **메모리 관리**: 시스템의 메모리 사용 현황을 추적하고, 프로세스들에게 메모리를 할당/회수하며, 페이지 교체 등 의 메커니즘을 통해 효율적으로 메모리를 관리
 - **프로세스 관리**: 프로세스 생성과 종료, CPU 스케줄링, 프로세스 간 통신(IPC) 등을 수행하여 여러 프로그램이 원활하게 CPU를 공유
 - **장치 드라이버 제어**: 각종 하드웨어 장치에 대한 디바이스 드라이버를 관리하고, 응용프로그램의 입출력 요청을 해당 드라이버를 통해 하드웨어에 전달. 커널은 하드웨어와 프로세스 간의 중재자로서 드라이버를 통해 장치를 제어
 - **시스템 호출 처리 및 보안**: 사용자 프로세스가 커널 기능을 요청할 수 있도록 시스템 콜 인터페이스를 제공하고, 이 때 권한 체크 및 보안을 적용. 커널 모드에서만 수행가능한 작업과 자원 접근을 통제하여 시스템 전체의 안정성을 유지

III. Kernel

Kernel mode vs User mode

- **유저모드:**
일반 프로그램이 동작하는 안전한 영역.
→ 시스템의 핵심 자원(메모리, 하드웨어 등)에 직접 접근 못함.
- **커널모드:**
운영체제(커널) 코드가 실행되는 특권 영역.
→ 하드웨어와 시스템 자원에 직접 접근/제어 가능.



III. Kernel

System Call 실습

- System Call 이란?
 - 유저 모드에서 커널 모드로 진입하는 공식 통로
 - 파일 입출력, 프로세스 관리, 메모리 할당 등 주요 OS 기능은 반드시 system call 을 통해 요청
 - strace 는 리눅스에서 프로그램이 어떤 시스템콜(system call)을 호출하는지 실시간으로 보여주는 명령어/도구로 이를 통해 System Call 이 어떻게 호출되는지 확인 가능함
- strace 를 활용해서 리눅스 명령어 따라가 보기

```
strace ls
```

```
strace ps aux
```

`strace -c ls` : 시스템콜별 호출 횟수/시간 요약

`strace -e trace=open,read,write ls` : 파일 열기/읽기/쓰기만 추적

III. Kernel

Ubuntu 에서 커널 정보 확인하기

- 커널 정보 출력하는 명령어 – uname

```
uname -r # 실행 중인 커널의 버전 번호를 출력  
uname -a # 커널 버전뿐 아니라 시스템의 전체 정보를 출력  
uname -v # 커널 컴파일 시각 및 빌드 정보
```

Tip: uname -r 로 출력된 버전과 일치하는 설정 파일이 /boot 디렉토리에 존재함 (예: /boot/config-5.15.0-75-generic), 이게 문지 확인해 보기

- 가상 파일 시스템 /proc 아래 명령어

```
/proc/version # 커널 버전과 빌드 정보  
/proc/meminfo # 메모리 용량 및 사용 현황  
/proc/cpuinfo # CPU 모델, 코어 수 등 프로세서 정보  
/proc/sys/kernel/hostname # Host 이름  
/proc/sys/kernel/osrelease # OS 릴리스 버전
```



THANK YOU