

Computer System

저녁이 있는 프로젝트
오상훈
6 Hours, 1 Month

Final Report

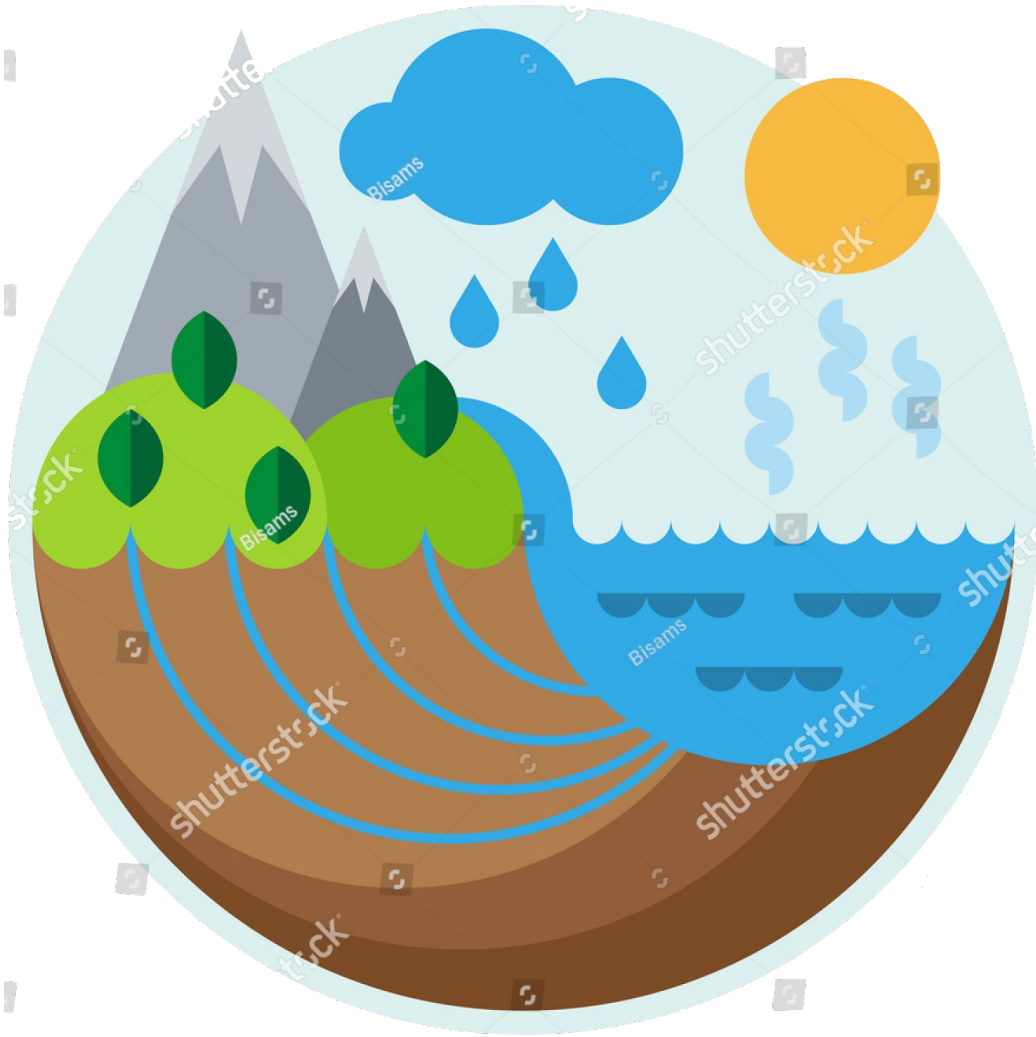
- ❖ 해보기
 - 컴퓨터 구성 마인드맵
 - 생각대로 컴퓨터 분해와 연결된 모든 것 표시 (이미지 or 글자)



컴퓨터 내부(본체)

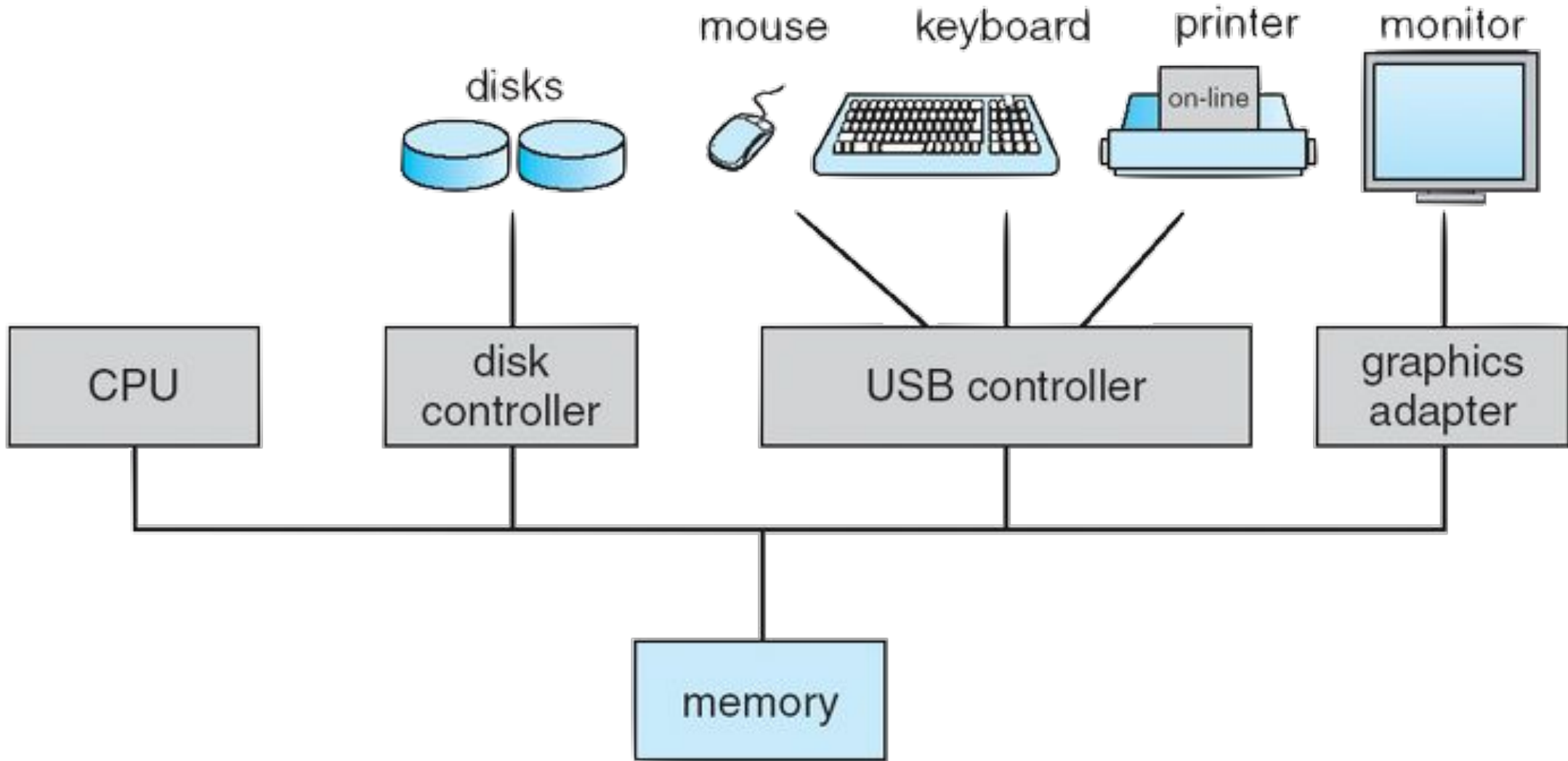
What's System !

- ❖ 하나의 공통적인 목적을 수행하기 위해 조직화된 요소 **집합체**
- ❖ 태양계(Solar System), 생태계(EcoSystem), 인간몸(Human Body System) 등



What's Computer System ?

- ❖ 어떠한 작업을 잘 수행하기 위해 주의 깊게 선정된 요소 **집합체**.
- ❖ 헐리우드 키드 : 발전, 창조! 경험 통한 반영
- ❖ 여러분들 생각이 컴퓨터 전부인가 ?



Try - 업무 시스템 구조

- ❖ 해보기
 - 자신 업무 구조 (단, 최종 판결자 표시)

내부

외부

❖ Hex Editor extension for Visual Studio Code



2진수 소통

❖ 좋아하는게 달라 소통 필요 : 인간 16진수, 컴퓨터 2진수(bitwise operation)

➤ 영화로 보기 : <https://www.youtube.com/watch?v=Vy7RaQUmOzE>

➤ 변환 순서

- 진수 맞게 자리 수 묶기
- 중복 없게 가중치 부여
- 묶음 합산과 표기.

ex) 2진수 → 16진수

$$\begin{aligned} 101011 &\rightarrow (0010)(1010) \\ &(1^2 0^1)(1^8 0^4 1^2 1^1) \\ &(2+0)(8+0+2+1) \rightarrow (2)(11) \rightarrow 2B \end{aligned}$$

❖ 알아보기

➤ 2진수 → 10진수

- 진수 맞게 자리 묶기
- 중복 없게 가중치 부여
- 묶음 합산과 표기.

$$\begin{aligned} 111010 &\rightarrow (0011)(1010) \\ &(1^2 1^1)(1^8 0^4 1^2 0^1) \\ &(2+1)(8+0+2+0) \rightarrow (3)(10) \rightarrow 3? \end{aligned}$$

➤ 16진수 → 2진수

- 중복 없게 가중치 부여

$$4B \rightarrow (1^4 0^2 0^1)(1^8 0^4 1^2 1^1) \rightarrow 1001011$$

➤ Editer로 확인

❖ 해보기

➤ 2진수 → 16진수

- 1001, 110111, 1010011

➤ 16진수 → 2진수

- F, CE, D3

컴퓨터 문자 소통

- ❖ 문자 부호화(Encoding)와 문자 복원화(Decoding)
 - 기호 집합을 컴퓨터에서 저장, 통신 목적 사용할 경우
- ❖ 방식 : Remember Text Table on Computer
 - ASCII(American Standard Code for Information Interchange)
 - 1 Byte = 7 bit + 1 bit(에러코드)
 - ANSI(American National Standards Institute) Code
 - ASCII 문제 완화, 에러코드 삭제.
 - Unicode Code(<https://www.unicode.org/>, <https://unicode-table.com/>)
 - 비유럽권 불편 완화, UTF-8 권장.

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	

로마자, 로마자권 기호

기타 유럽 문자

아프리카 문자

중동·서남아시아 문자

남부와 중앙 아시아 문자

동남아시아 문자

동아시아 문자

CJK 문자

인도네시아, 오세아니아 문자

북미 및 남미 문자

Notational systems

기호

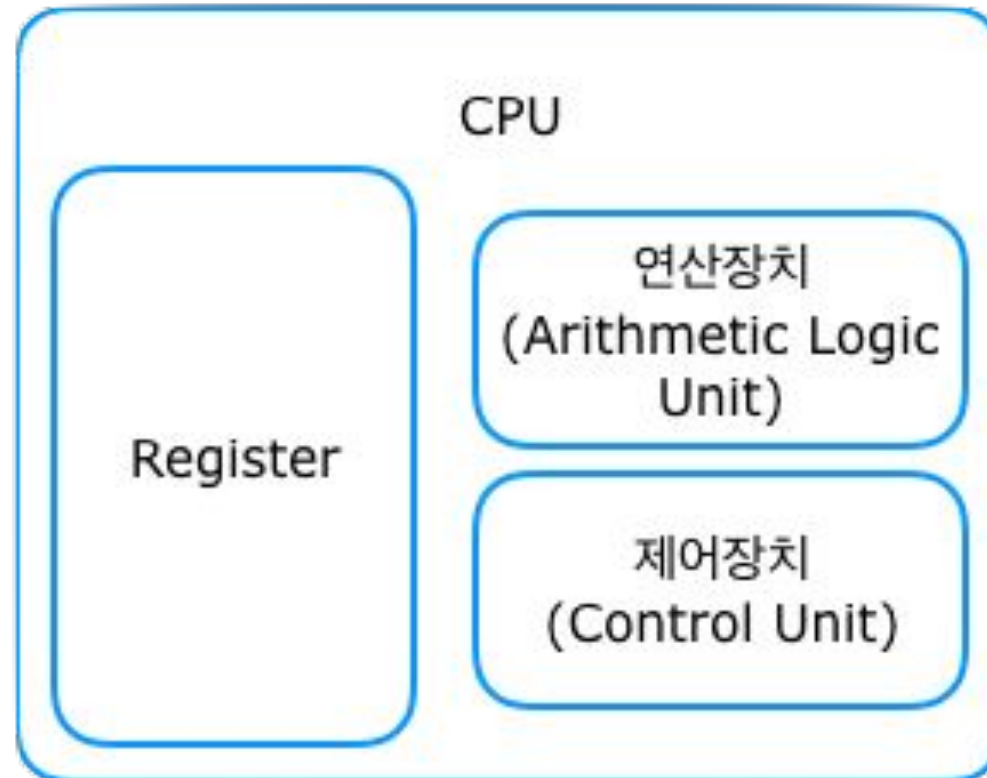
사용자 정의 영역

UTF-16 상·하위 대체 영역

쓰이지 않음

CPU

- ❖ 몸은 하나 할 일은 많다
 - **Interrupt** : 각각 디바이스 컨트롤러가 인터럽트 발생해 **CPU**에게 알리는 신호
 - 부족 자원 극대화 방식
 - 일괄 처리 시스템 : 급여명세서나 봉급계산, 수도세, 성적처리 등 이용
 - 다중 프로그래밍 : **CPU** 이용률을 증진, 유휴 상태 방지.
- ❖ **Timer** : 특정 프로그램이 무한루프에 빠져 **CPU**를 독점하는 경우를 막기위해 존재
 - 시분할 시스템 : **CPU** 스케줄링과 다중 프로그래밍 이용 각 사용자에게 분할
- ❖ **Mode bit** : 실행되는 프로그램이 프로그램인지 운영체제인지 구분(지연 시 **Kill** 대상)



Process - ps

❖ 알아보기

➤ **tail** [OPTION]... [FILE]... : 파일 마지막 부분 출력

```
~$ echo '1234567890' >> ./echo_tail.txt
```

```
~$ tail ./echo_tail.txt
```

```
1234567890
```

```
~$ tail -f ./echo_tail.txt
```

```
1234567890
```

```
~$ echo '0987654321' >> ./echo_tail.txt
```

other Terminal

❖ **ps** : 현재 프로세스 상태

➤ **STAT** : R(실행), S/D(수면), T(종료), Z(좀비), <(선순위), N(후순위)

```
~$ top
```

```
...
```

```
~$ ps -ef
```

```
...
```

```
~$ ps -ef | grep 'tail'
```

```
501  510  487  0  6:04PM ttys001    0:00.01 tail -f echo_tail.txt
```

```
501  656  519  0  6:36PM ttys004    0:00.00 grep tail
```

```
~$ sleep 100 &
```

```
~$ ps -ef | grep sleep
```

```
...
```

영혼은 존재하는가 ?

- ❖ 우리 잘 죽기 위해 태어났다.
- ❖ 죽음은 몸에서 무엇인가 없어지는 것. = 동작되던 프로그램에서 무엇인가 변하는 거



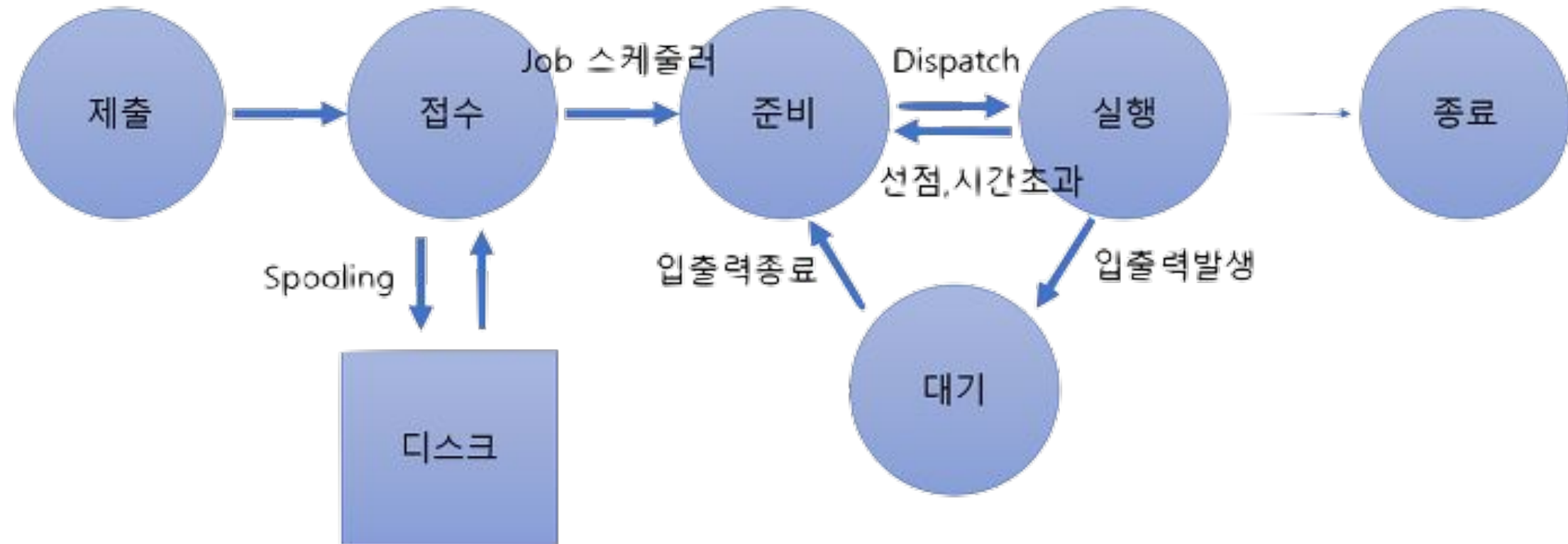
Instance

- ❖ 불러주었을 때 인스턴스가 되었다.
- ❖ 인스턴스(=프로시저)는 메모리가 소프트웨어가 올라간 상태.

2 네가 2의 이름을
불러주었을 때
2는 나에게로 와서
일하 되었다.
조남 김현수

What's Process !

- ❖ CPU 의해 실행중인 프로그램(사용자, 시스템)을 의미
 - 협동 프로세스 / 스레드(Thread)
 - 상호 배제와 동기화 / 세마포어



Process - kill

❖ **kill, killall** : 프로세스 시그널 종료, **kill -9 PID**

~\$ kill -9 510

~\$ ps -ef | grep 'tail'

501 669 519 0 6:39PM ttys004 0:00.01 grep tail

❖ **top** : 프로세스 상태 표시

➤ 과부하 상태 : **load average** >= CPU 코어수, 큰 CPU 사용률과 시간

➤ **Press C** : command 상세 표시

➤ 정렬 : **Press Shift+M, Shift+T, Shift + P**

~\$ pstree | less

~\$ top

~\$ vmstat

❖ **jobs** : 실행 작업 나열

❖ **bg(&)** : 프로세스 백그라운드 전환

➤ **fg** : 프로세스 포그라운드 전환

❖ **systemctl** : 프로세스 관리

~\$ sudo systemctl status ssh

~\$ sudo systemctl restart ssh

❖ 해 보기 : **GetProcessId.c**

Memory Simulation - variable

메모리판

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

변수

int	float	CH	CH	CH	P

값

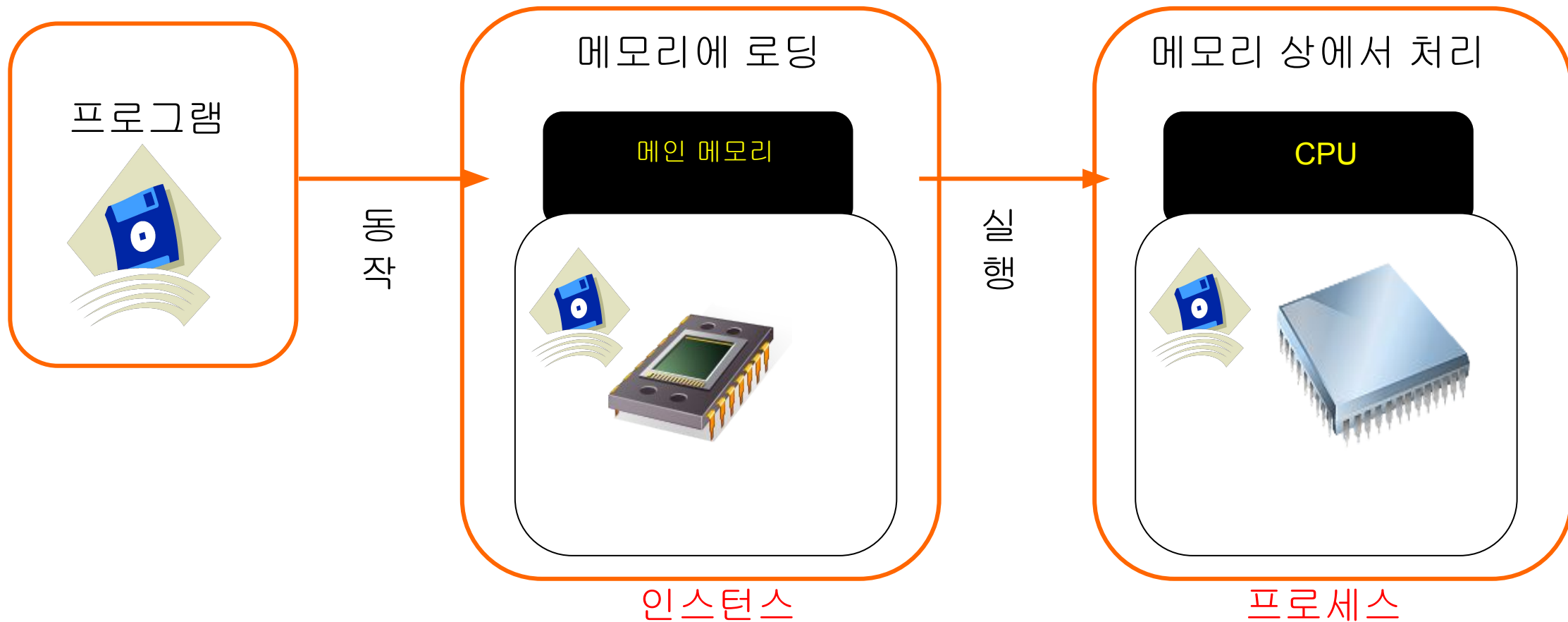
Trying - Variable Memory Simulation

❖ 해 보기

- `char i = '\\';`
`char i01 = 'A';`
- `int num = 0;`
`int num01 = 223;`
- `float val01 = 3.14f;`
`float val02 = 2.618f;`
- `int days[] = { 31, 23, 42 } ;`
`int nested[3][2] = { {1,2}, {2,4}, {3,6}};`
`char name[4];`
`char tel[] = "010-2345-6789";`

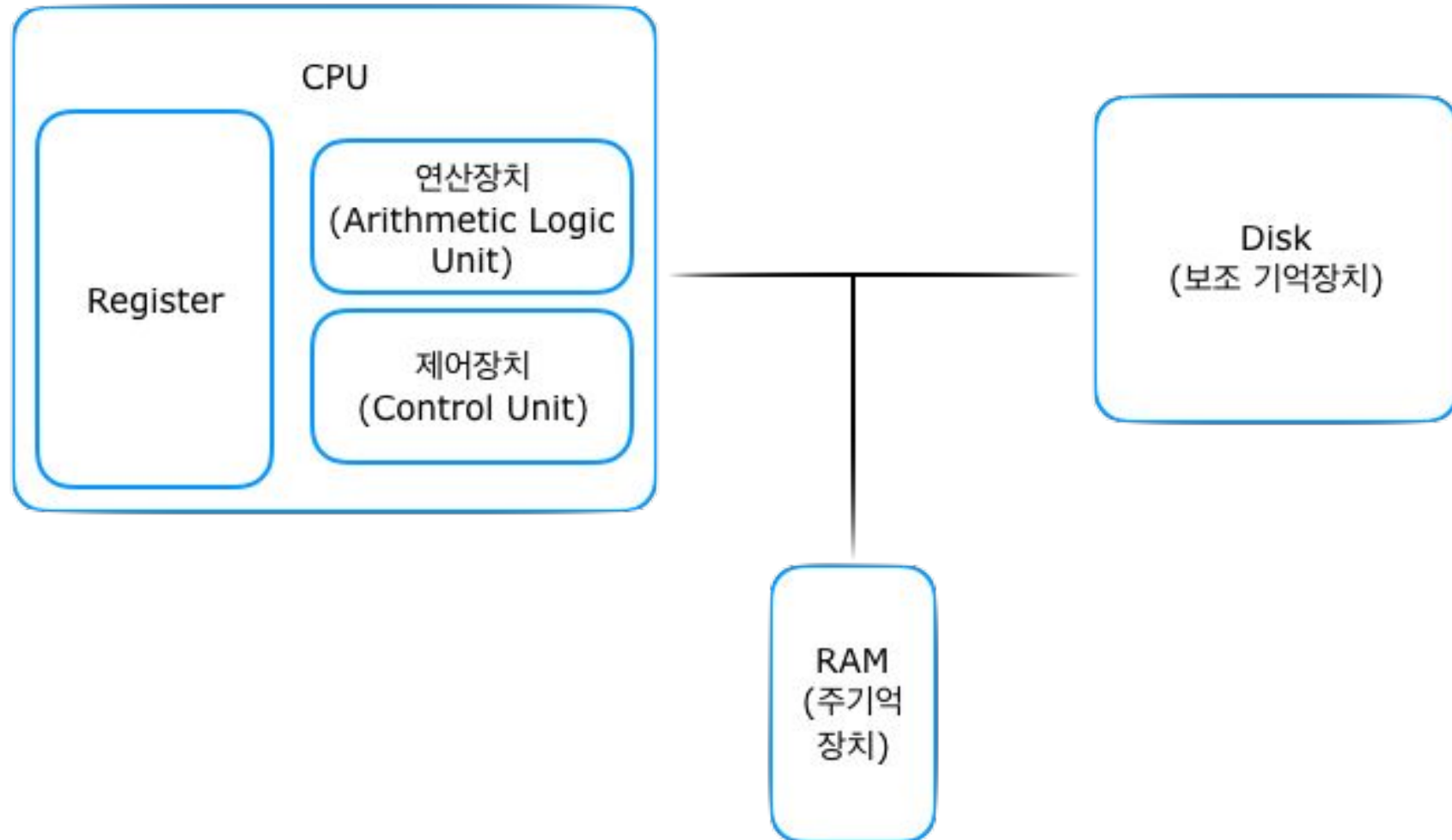
Memory

- ❖ CPU Register 한계 극복
- ❖ 스택영역 : 호출된 함수 종료 후 원래 함수 위치로 돌아오기 위해 복귀 주소 저장
- ❖ 데이터 영역 : 전역변수 등 프로그램이 사용하는 각종 데이터가 저장되는 공간
- ❖ 코드 영역 : 프로그래머가 작성한 코드가 기계어 명령형태로 저장되는 영역



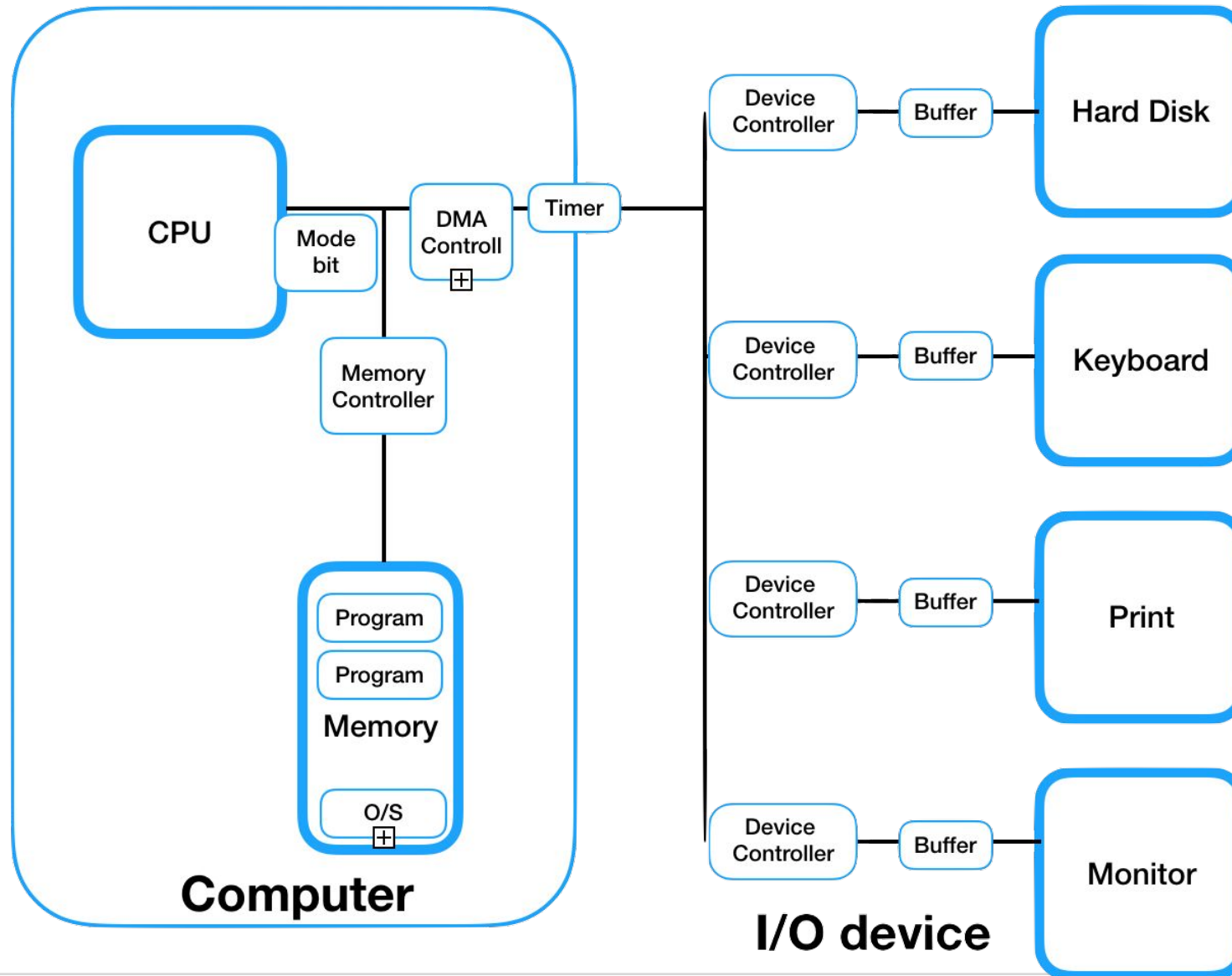
Memory Per Device

- ❖ CPU, Memory, I/O(Input/Output) Device → 장치간 소통은 ?
- ❖ 저장 장치 종류
 - Register : CPU 위한 것. 휘발성.
 - RAM, ROM : 프로그램(인스탄스) 위한 것. 일부 펌웨어(로봇 사용).
 - Disk : 예비 프로그램 위한 것. 비휘발성. → 파일 시스템 때 설명.



In / Output(I/O) Device

- ❖ 드라이브 설치에 인터럽트 위한 것 : Ordering Devices with Interrupt Vector
- ❖ DMA(Direct Memory Access) : 주변기기 인터페이스 장치에 제어권 주어 직접 주기억장치와 데이터 주고 받는 방식



File System

❖ 주말농장 사장님! 어떻게 해야 잘 나누죠 ?



Disk - Type

❖ 정보 영구 저장

~\$ df -h

❖ Disk Type

- IDE : 거의 사라짐.
- SATA : 많이 쓰임.
- SCSI : Hot Plug In 기능으로 Server System 많이 사용.



물리 디스크 추가

부팅 또는 장치 인식

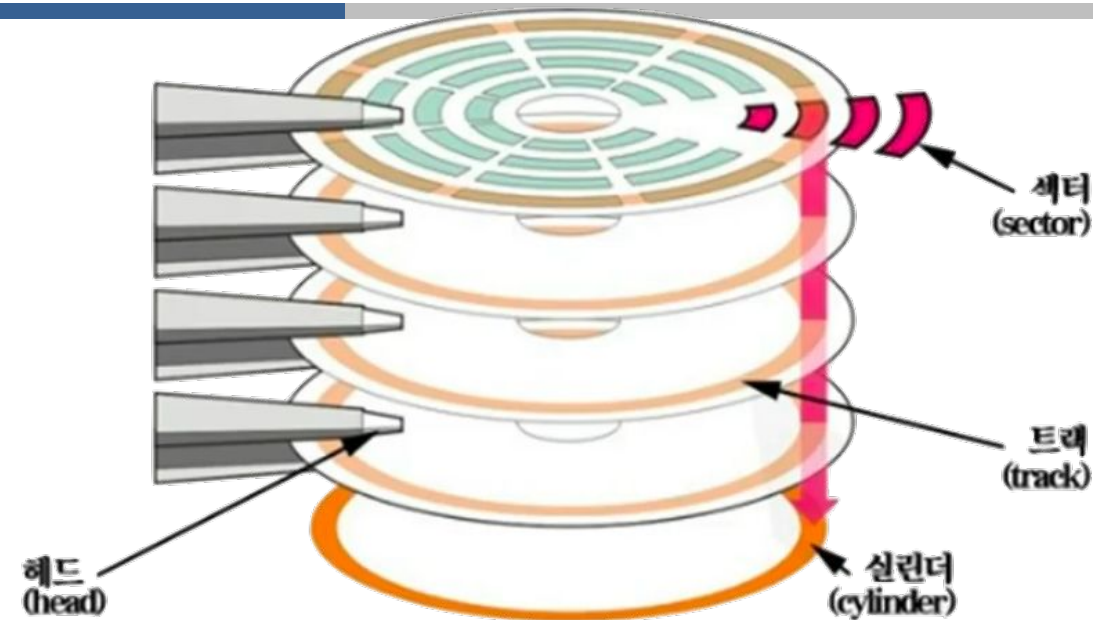
파티션 설정

파일 시스템 설치
(포맷)

사용

Disk - Format

- ❖ Data 정리 위해 구역 정리
~\$ fdisk -l



	EXT2	EXT3	EXT4
출시	1993	2001	2006
최대 파일 크기	16GB ~ 2TB	16GB ~ 2TB	16GB ~ 16TB
최대 파일 시스템 크기	2TB ~ 32TB	2TB ~ 32TB	1EB
특징	저널링X	저널링O	Extents Multi-block Allocation

Disk(Linux) - File System(1)

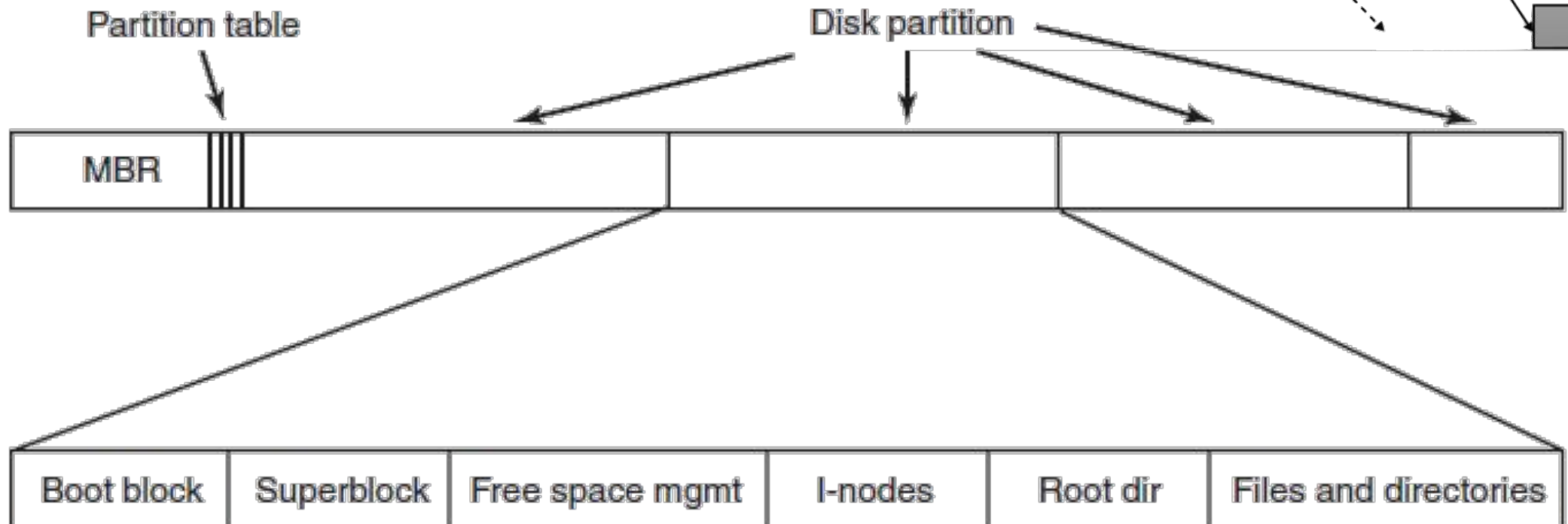
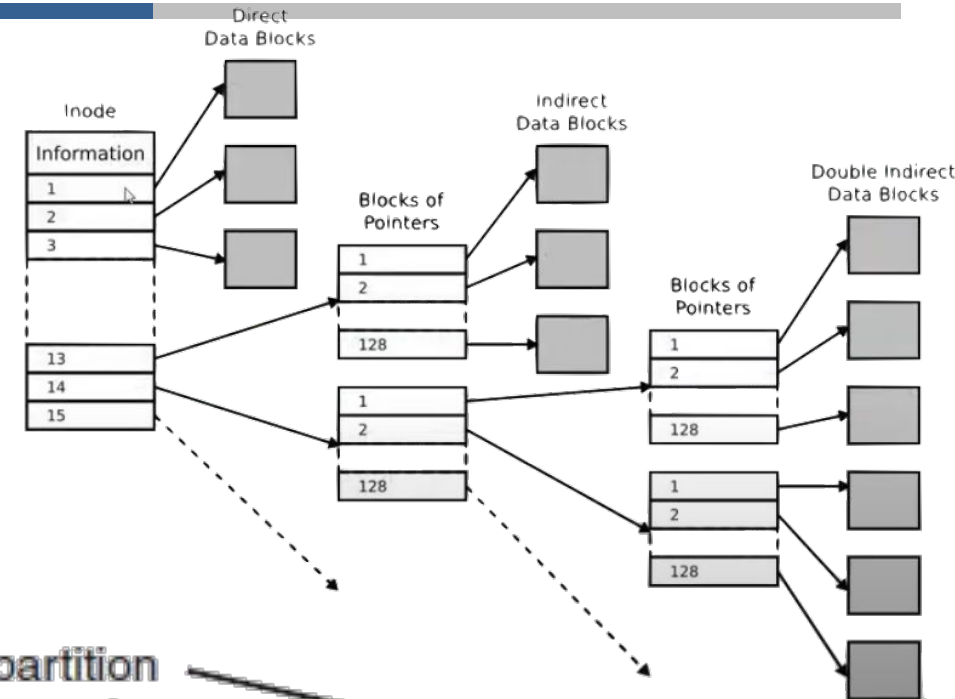
- ❖ MBR(Master Boot Loader)
- ❖ 파티션 (=? 마운트)
- ❖ 파일 포맷 (FAT32, EXT3, EXT4, ...)
- ❖ 색인(inode) 리스트, 디렉토리/데이터 블록

~\$ ls -il

794663 drwx----- ... 27 17:37 Applications

444383 drwx----- ... 22 16:15 Desktop

444367 drwx----- ... 21 17:01 Documents

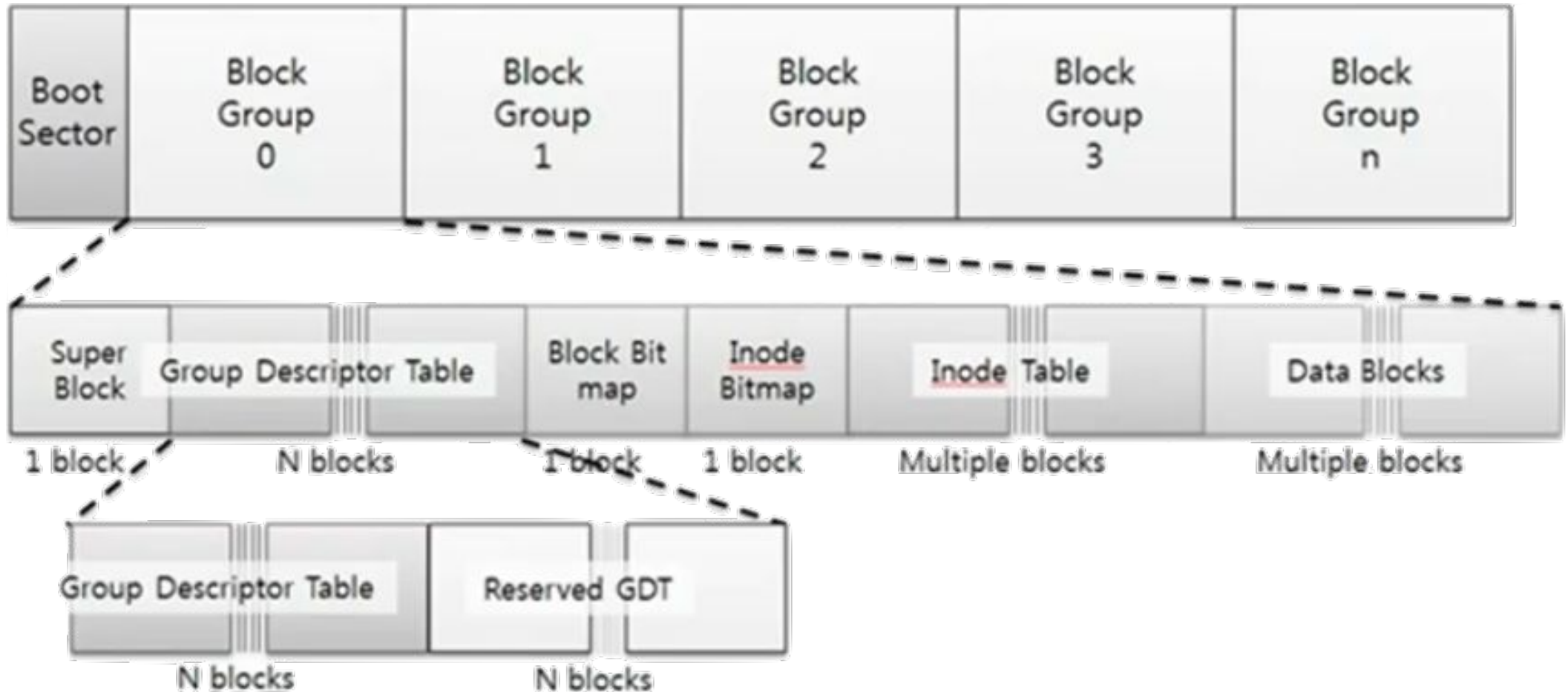


Disk(Linux) - File System(2)

❖ Manage Data : 퍼져 있는 정보 창고에서 보다 빠르게 정보 가져오기.

~\$ mkfs /dev/sdb1

Block Area Format



❖ 해 보기 : Stat.cpp

Try - Manage Disk

```
~$ df -h                                     # or ~$ lsblk
Filesystem      Size  Used Avail Use% Mounted on
...
~$ sudo fdisk -l
...
/dev/sdb1        2048 15630335 15628288 59.5G  c W95 FAT32 (LBA)
~$ sudo fdisk /dev/sdb
...
Command (m for help) : m
...
  d  delete a partition
  n  add a new partition
...
Command (m for help): w                      # w that you want to apply it
~$ lsblk
...
sdb      8:16   1 59.5G  0 disk
└─sdb1    8:17   1 59.5G  0 part
~$ sudo mkfs.ntfs -v /dev/sdb1
...
```

mount

❖ mount : 장치와 리눅스 커널 인식 경로 설정.

~\$ mount

...

gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse...

/dev/sdb1 on /media/rapa/847320a9-f217-4c2b...

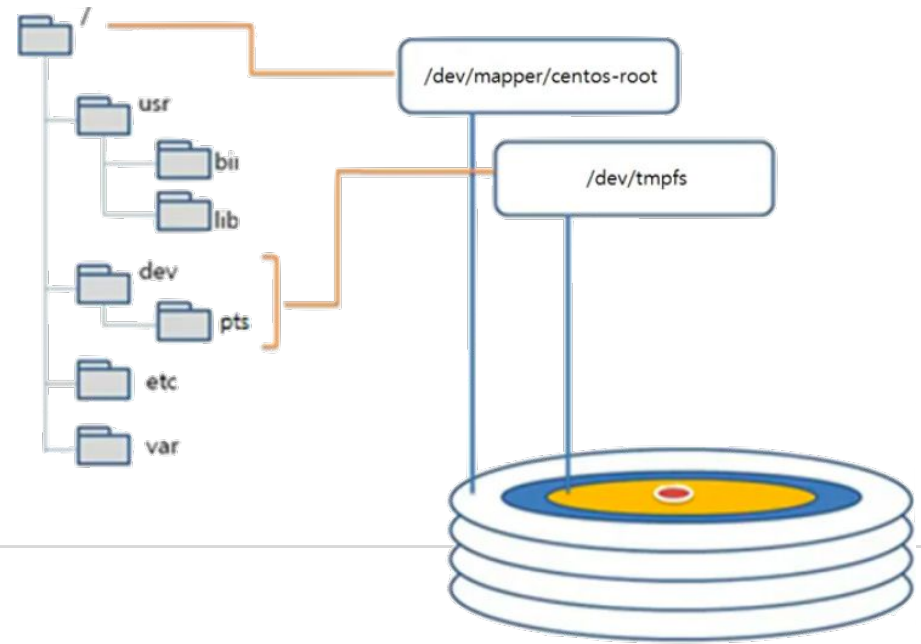
~\$ cat /etc/fstab

UUID=ca77... / ext4 errors=remount-ro 0 1

UUID=6893... /boot/efi vfat umask=0077 0 1 /swapfile ...

~\$ findmnt

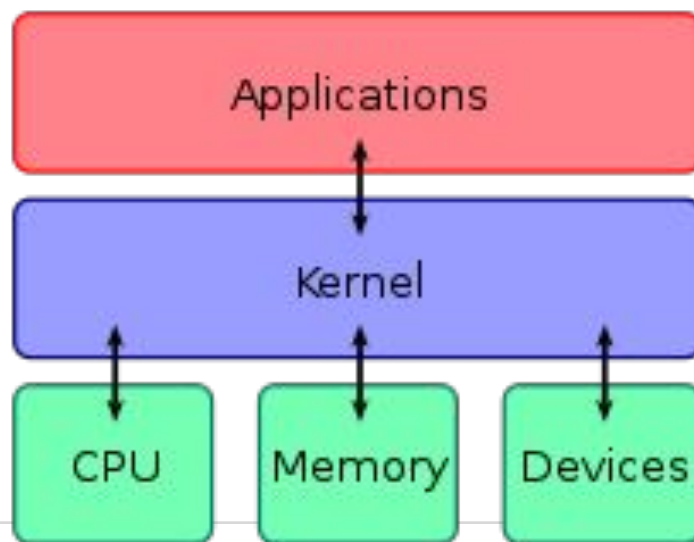
TARGET	SOURCE	FSTYPE	OPTIONS
/	/dev/sda2	ext4	rw,relatime,errors=remount-ro
└─/sys	sysfs	sysfs	rw,nosuid,nodev,noexec,relatime
...			



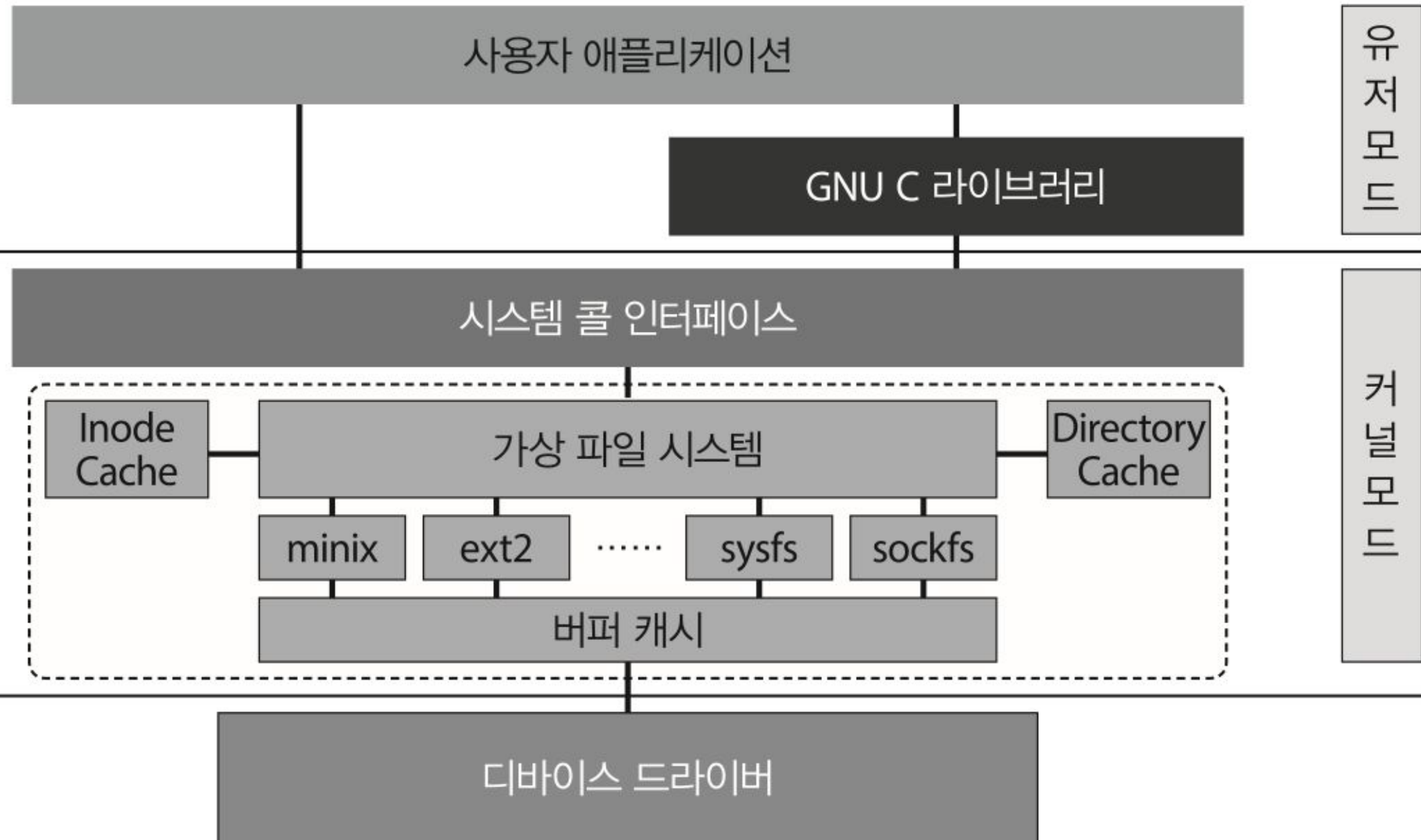
- ❖ **tar** [OPTIONS] ARCHIVE ARCHIVE : 여러 파일 묶음 해제
~\$ tar cvf file-digit.tar pinky* → 옵션 f 제거 시 tape backup
~\$ tar xvf file-digit.tar
- ❖ **gzip, gunzip** : 파일 압축 및 압축 해제, 단일 파일 압축 용도.
~\$ cp /usr/bin/pinky ./Downloads/
~\$ gzip pinky
~\$ ls -l pinky* → 압축 사이즈 확인
~\$ gunzip pinky.gz
- ❖ **bzip2, bunzip2** : 블록 단위 파일 압축 및 해제
~\$ bzip2 pinky
~\$ ls -l pinky* → 압축 사이즈 확인
~\$ bunzip2 pinky.bz2 → **bzip2 -kd** pinky.bz2
- ❖ **zip, unzip** : 파일 묶고 압축
~\$ zip file-digit.zip pinky*
~\$ unzip file-digit.zip → 옵션 -f 넣고 실행

Kernel

- ❖ refer : [https://en.wikipedia.org/wiki/Kernel_\(operating_system\)](https://en.wikipedia.org/wiki/Kernel_(operating_system))
- ❖ A computer program at the core of a computer's operating system with complete control over everything in the system
 - Linux OS : Created in **1991** by Linus Torvalds^[9] for his i386 based PC, It was soon adopted as the kernel for the GNU Operating System which was created as an open source and free software, and based on UNIX
 - official : <https://www.kernel.org/>
~\$ uname -r
 - Mac OS : first released by Apple Inc. in **2000**, It is mostly POSIX-compatible, but has never, by itself.
 - refer : [https://en.wikipedia.org/wiki/Darwin_\(operating_system\)](https://en.wikipedia.org/wiki/Darwin_(operating_system))
 - Windows OS : Basic code is similar in function to MS-DOS. As a 16-/32-bit hybrid



Linux System



Kernel - Wake up Computer(Linux)

- ❖ Booting Order : 제품 개발 업체간 약속 위한 규칙 준수. main() 필요성.
 - Power ON → run BIOS → run GRUB → Swapper/Init Process → Login prompt
- ```
~$ sudo cat /boot/grub2/grub.cfg | grep Core # booting kernel menu
~$ cat /var/log/boot.log # booting log
~$ ls -la /boot/*vmlinuz*
~$ uname -a # check kernel version
```



## Run Level(Linux)

❖ Defines the state of the machine after boot

~\$ who -r # check up current Run level

➤ init : before CentOS v5, ubuntu v16

~\$ ls /etc/rc5.d/

~\$ init 6 # System Reboot

➤ systemd : after Linux version, init 단점 보완, 추가 기능 구현 가능.

~\$ systemctl status ssh.service # ssh.service 대신 ssh 가능

~\$ systemctl list-unit-files --type service

~\$ reboot # System Reboot

| ID | Name                               | Description                                      |
|----|------------------------------------|--------------------------------------------------|
| 0  | Halt                               | Shuts down the system.                           |
| 1  | Single-user mode                   | Mode for administrative tasks. <sup>[2][b]</sup> |
| 2  | Multi-user mode                    | Does not configure network interfaces            |
| 3  | Multi-user mode with networking    | Starts the system normally. <sup>[1]</sup>       |
| 4  | Not used/user-definable            | For special purposes.                            |
| 5  | Start the system normally with GUI | Same as runlevel 3 + display manager.            |
| 6  | Reboot                             | Reboots the system.                              |

# File Descriptor - Linux

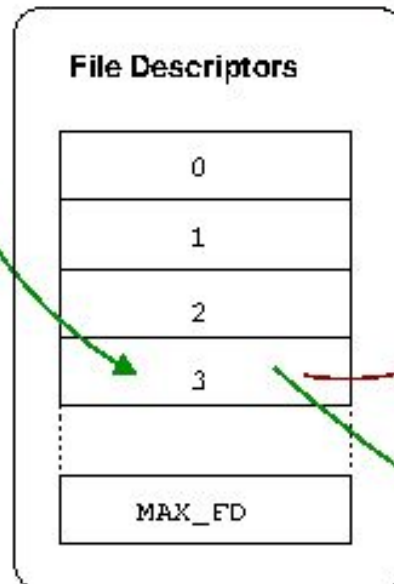
2

Opening the file  
associates a descriptor  
with the associated device

```
int fd = open("/dev/sr0");
int ret = read(fd, &input, count);
```

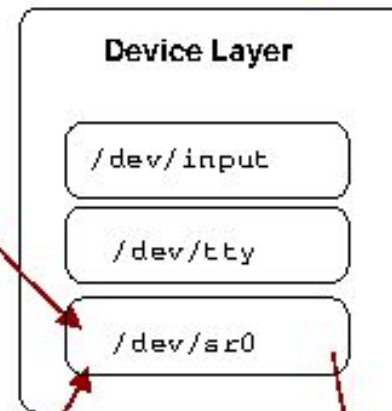
3

Further references  
to the descriptor  
are routed to the device

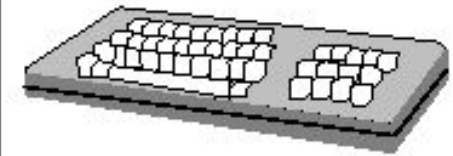


1

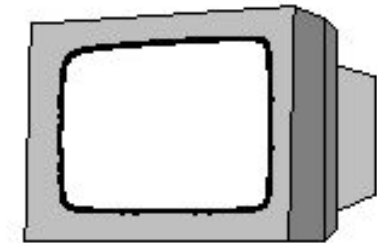
Devices register  
with the kernel  
which gives them a file



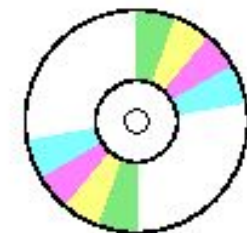
## Device Drivers



```
device_read()
device_write()
```



```
device_read()
device_write()
```



```
device_read()
device_write()
```

## Try - File Descriptor - xinput

```
~$ sudo apt install xinput
```

```
~$ xinput --list
```

```
[Virtual core pointer id=2 [master pointer (3)]
| ↳ zuomunt keyboard voice id=10 [slave pointer (2)]
| ↳ Logitech USB Optical Mouse id=13 [slave pointer (2)]
[Virtual core keyboard id=3 [master keyboard (2)]
...
 ↳ Sleep Button id=8 [slave keyboard (3)]
 ↳ zuomunt keyboard voice id=9 [slave keyboard (3)]
...
```

```
~$ xinput --list-props 10
```

```
Device 'zuomunt keyboard voice':
```

```
Device Enabled (155): 1
```

```
...
```

```
~$ xinput --test 10
```

```
redgsehdsfbs shdgsdb^[[C^[[A^[[D^[[C^[[24~^[[
```

```
~$ xinput --test 13
```

```
motion a[0]=2405
```

```
...
```

❖ 해 보기 : ReadMouse.c

# Syslog

- ❖ refer : <https://en.wikipedia.org/wiki/Syslog>
- ❖ Standard for message logging, the software that reports and analyzes them.  
[Facility].[Level] [Action] → ~\$ less /etc/rsyslog.conf

➤ Facility

| code | Keyword | Description                              |
|------|---------|------------------------------------------|
| 0    | kern    | Kernel messages                          |
| 1    | user    | User-level messages                      |
| 3    | daemon  | System daemons                           |
| 4    | auth    | Security/authentication messages         |
| 5    | syslog  | Messages generated internally by syslogd |

➤ Severity level

| Value | Severity      | Keyword |
|-------|---------------|---------|
| 0     | Emergency     | emerg   |
| 1     | Alert         | alert   |
| 2     | Critical      | crit    |
| 3     | Error         | err     |
| 4     | Warning       | warning |
| 5     | Notice        | notice  |
| 6     | Informational | info    |
| 7     | Debug         | debug   |

## Try - Syslog

❖ 해 보기 : BasicInputOutput.cpp

➤ 아래 표 로그 확인

~\$ less /var/log/messages

...

|          |                    |                             |
|----------|--------------------|-----------------------------|
| 시스템 로그   | /var/log/messages  | 시스템 전반적인 로그                 |
| 보안 로그    | /var/log/secure    | Inetd에 의한 로그                |
| 메일 로그    | /var/log/maillog   | 메일로그                        |
| 크론 로그    | /var/log/cron      | 작업 스케줄링 로그                  |
| 부팅 로그    | /var/log/boot.log  | 시스템 부팅시의 로그                 |
| Dmesg 로그 | /var/log/dmesg     | 부팅시 기록되는 로그                 |
| Utmp 로그  | /var/run/utmp      | 현재 시스템에 로그인한 각 사용자의 상태를 출력  |
| Wtmp 로그  | /var/log/wtmp      | 로그인, 로그아웃, 시스템의 재부팅에 대한 정보  |
| Last 로그  | /var/log/lastlog   | 계정 사용자들이 마지막으로 로그인한 정보      |
| 아나콘다     | /var/log/anaconda  | 리눅스 설치시 installer 과정에 대한 로그 |
| Su 로그    | /var/log/sulog     | Su 명령어를 통한 로그인시 정보 기록       |
| Pacct 로그 | /var/account/pacct | 로그인한 모든 사용자의 실행한 프로그램 정보 기록 |
| Btmp 로그  | /var/log/btmp      | 실패한 로그인 시도를 기록              |



