

## 컴퓨터 역사

진공관 → 레지스터 → 집적회로(IC)  
의 CPU 발전 역사.

## 데이터 표현.

컴퓨터는 0과 1로 만들어짐.

↳ 정수, 실수, 문자를 encoding

## 컴퓨터

하드웨어 (입력, 출력, 기억, 연산, 제어)  
소프트웨어 (System SW vs 응용 SW)

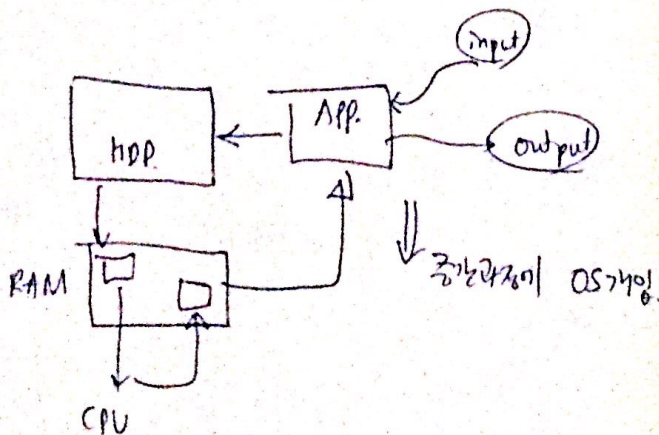
ex) OS → HW interaction

user와 interaction.

↳ int 4 byte ⇒ related to  
BUS (CPU bit)  
↓  
메모리 32 bit 연결 가능  
(CPU 처리 가능 한대치).

아) kernel HW와 software 연결하는 링크.

Data flow flow.



(연산)

→ AND, OR, NOT, XOR, NOR, NAND.  
만약. 다르면 1, 같으면 0. (XOR)  
둘다 참 → F, 하나만 참 → T (NAND)

비연산 논연산.

8 ≠ 88  
bool value.

1 ≠ 11.

RAM

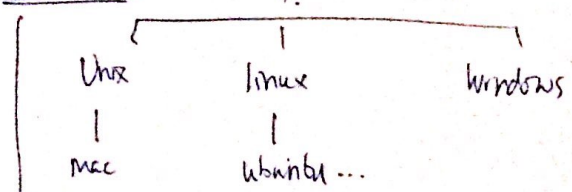
메모리장치.

↔

하드

메모리장치

운영체제



(연산)

시스템 HW 관리  
가장 ~~쉬운~~ system 리소스 제공  
관리.

자료구조

정수/실수/문자.

array: fast, random access, stable

연립리스트: 삭제·삽입 중간에 가능, slow search

stack: first in, last out, first out "pop" push

queue: first in, first out

tree: 1: 1만 가지 등... ⇒ 9진법으로 nlogn.

graph: 연결망 ⇒ facebook.



시간 복잡도  $\Rightarrow$  Big O notation.

process: 실행되고 있는 프로그램.



on RAM

생성  $\rightarrow$  준비  $\rightarrow$  대기  $\rightarrow$  실행  $\rightarrow$  종료

프로세스 스케줄링 by O/S.

① FCFS

② shortest job first

③ round robin scheduling

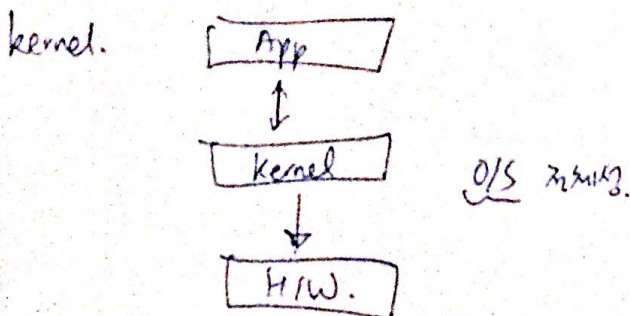
$\hookrightarrow$  한정 시간당 동안 작업  
 $\hookrightarrow$  multi-tasking!

④ priority based

⑤ multi queue.  $\rightarrow$  (1~4) 무한대.

memory dump  $\rightarrow$  memory 상태를 캡처!  
 $\downarrow$   
장기기보!

파일관리: 파일시스템이 다 다르지만 결국  
G/S가 위치를 기억  
 $\wedge$   
H/W에 있는 식별자.



DB  $\rightarrow$  (관계형)  $\Rightarrow$  SQL을 사용. (가장 일반적).

DBMS: OS of DB.

(경쟁)  $\Rightarrow$  경쟁자/상대.

" "  $\Rightarrow$  같음.

-  $\Rightarrow$  반대

\*  $\Rightarrow$  wildcard!

~  $\Rightarrow$  similar.

site: / link: / related: / mtext: / m

filetype: /

(네트워크)

LAN, WAN: 근거리 vs 광역망

server  $\leftrightarrow$  client

Internet: through TCP/IP.

WWW: 웹사이트 / 웹상 사이트.

protocol: http, https, ftp 등.

통신/정보교환의 규칙.

(GET): 자료 요청

POST: 자료 전송

PUT: URL에 저장

DELETE: 삭제

TELNET: 원격조정, text 기반 암호화 <



SSH: " 암호화 O.

SSL: 웹서버와 브라우저 간의 보안 protocol.  
secure socket layer.

IP: internet protocol address

통신사 식별자.

DNS: domain name system: IP  $\leftrightarrow$  domain.

MAC address: H/W와 network adapter에 특정한  
식별자.



암호화 : 대칭키  $\Rightarrow$  같은 키로 암호화 & 복호화.

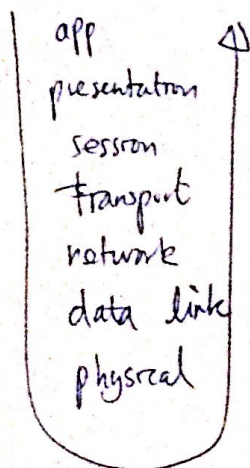
공개키  $\Rightarrow$  비밀키 공개 / public & private

두 개를 사용

[protocol] : // [HOST NAME] : [PORT] / [PATH]

protocol에 따라.  
 $\rightarrow$  다 변하는 것 같나.  
 ex) http: 80번

OSI 7계층



S/W 개발

Waterfall, prototyping, 사용...

agile  $\Rightarrow$  code-oriented

$\downarrow$   
 그때그때 코드 작성하며 수정은  
 다른 다른 작업자들...

UML: modelling 이 용어

결과 코드 를 시작하!

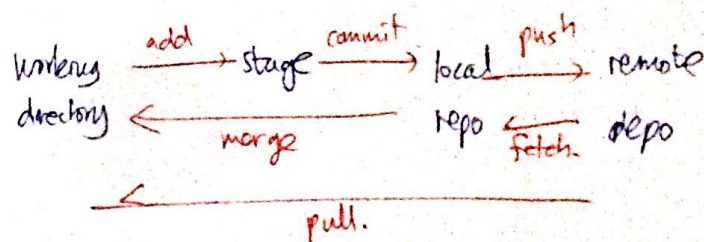
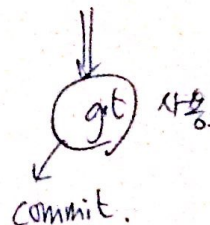
$\downarrow$   
 Model / class diagram.

$\downarrow$   
 간단한 function prototype까지  
 나옴.

TDD : 자동화된 test, 작업은 개발 cycle 때문.

PDD:  $\approx$  똑똑한 방법.

개발생각 :  $\Rightarrow$  버전관리 포함 개발.



HASH  $\Rightarrow$  특정 data를 일련적이지 않게  
 mapping!

$\hookleftarrow$  단계 구분하기  
 $\downarrow$   
 복호화할거!

인터프리터 lang (JS) compile lang

Python, Java, Javascript

C, C++

$\downarrow$   
 수정 용이 but slow.

fast 실행  
 slow 수정.

API  $\Rightarrow$  O/S의 자체성. 연결용 with app  
 $\hookrightarrow$  주로 함수.

library = 함수의 모음.

framework = S/W 제작시 주목되는 것들을  
모아 쉽게 가능하게 하는 틀.

---

객체 지향  $\rightarrow$  blueprint인 class 를 instantiate  
해서 객체로 생성.  
 $\downarrow$   
실행.

$\downarrow$  상속 가능  
override / overload  
등...

frei