

컴퓨터 구조 중간고사 정리

<컴퓨터 구조론> P.16 ~ 106 / 강의록 1 ~ 6 주차.

이시현

1. 컴퓨터의 발달 (교재 1장 내용 정리)

- A. 컴퓨터 : 논리 연산을 통해서 인간의 일(task)을 대신 수행할 수 있는 범용 계산기이다.
- B. System : 다양한 구성요소가 상호작용 가능하도록 하는 구조.
 - ① 컴퓨터 시스템 : 컴퓨터에게 일을 시키기 위한 모든 장치들의 집합(구조)과, 그 장치들이 상호작용 하기 위한 절차와 약속들. 크게 물리적 요소와, 논리적 요소로 나눌 수 있다.
- C. 소프트웨어
 - ① 정의 : 명령어의 집합. 혹은 작업지시서. Program과 유사한 용어임.
 - ② 응용 프로그램 : 특정한 일을 하도록 만들어진 프로그램(소프트웨어).
 - ③ 시스템 소프트웨어 : 하드웨어 구동과 관련된 모든 프로그램의 집합.
 - ④ 펌웨어(firmware) : ROM에 설치된 하드웨어 제어를 위한 소프트웨어. 하드웨어와 유사하게 동작하지만 소프트웨어이다.
- D. Data / Information
 - ① Data : 정의하여 만들어진 모든 것.
 - ② Information : Data를 습득하는 주체에게 의미 있는 것들을 지칭하는 용어.
- E. 신호에 대하여
 - ① 아날로그 신호 : 연속적 데이터.
 - ② 디지털 신호 : 불연속적 데이터.
 - ③ 아날로그 신호를 디지털로 변환할 때 데이터 일부가 손실됨. 그 반대도 마찬가지.
 - ④ 논리회로를 통해 신호를 연산하기 위하여 현대의 디지털 컴퓨터들은 입력 받은 모든 아날로그 신호를 디지털로 바꾸어 연산하고, 연산 결과를 다시 아날로그로 바꾸어 출력함.
 - ⑤ 왜 굳이 디지털로 변환하여 연산하는가? 이는 현대 사회가 디지털 기반으로 인프라가 구성되었기 때문이기도 하며, 현대의 디지털 컴퓨터가 훨씬 빠르기 때문이기도 하다.

F. 웹(Web) / 인터넷(Internet)

- ① 웹 : 논리적인 개념이다. 도메인 객체와 도메인들이 이루는 논리적 구조를 지칭한다.
- ② 인터넷 : 물리적인 개념이다. 컴퓨터와 컴퓨터를 연결하는 물리적 구조를 지칭한다. 이 경우 실물 서버 컴퓨터와, 라우터, 수많은 케이블 등이 인터넷을 구성하는 요소들인 것이다.

G. 서버(Server) / 클라이언트(Client)

- ① 서버 : 데이터를 전송하는 컴퓨터.
- ② 클라이언트 : 데이터를 수신하는 컴퓨터.
- ③ 서버와 클라이언트는 고정된 개념이 아니다. 어느 순간이든 데이터를 전송하고 있다면 서버인 것이고, 그 반대라면 클라이언트인 것이다. (PC는 매 순간 서버이며 클라이언트로서 동작하고 있다.)

H. 클라우드 서비스

- ① 정의 : 사용자가 자신의 컴퓨터가 아닌, 다른 장소에 있는 컴퓨터의 리소스를 빌려 쓸 수 있도록 제공되는 서비스. 다른 말로 하면, 서버사이드 프로그램을 서비스로서 지원하는 것. (사용자는 실제로 다른 장소의 컴퓨터를 사용하는 것이다.)
- ② 예시 : 온라인 문서 편집 프로그램 - 프로그램을 수행시키는데 필요한 리소스는 어딘가의 서버 컴퓨터가 지불하고 있다.

I. 클라우드 컴퓨팅

- ① 정의 : 서버(클라우드)의 리소스를 빌려 쓰는 것.
 - i. 이때 리소스는 하드웨어와 소프트웨어를 모두 아우르는 표현이다.
 - ii. 클라우드 서비스를 사용하는 행위는, 클라우드 컴퓨팅을 하고 있는 것이다.

J. 비트(Bit) : Binary Digit, 컴퓨터가 이해하는 최소의 데이터 단위. 2진법의 한 자릿수.

K. 인코딩(Encoding) / 디코딩(Decoding)

- ① 인코딩 : 암호화. 어떤 문장을 특정 코드표를 통해 변환시키는 것.
- ② 디코딩 : 복호화. 변환된 문장을 본래 문장으로 변환시키는 것.
- ③ 사람과 컴퓨터는 인코딩과 디코딩이 반대이다. 이해하는 언어가 깔끔하게 이분되어 있기 때문.
- ④ 사람 입장에서 C언어로 프로그램을 작성했다고 가정하자. 이때 소스코드가 컴파일 되는 것은 아래와 같다.
 - i. 사람 입장 : 인코딩. (해독 가능)C언어 -> (해독 불가)바이너리 코드
 - ii. 컴퓨터 입장 : 디코딩. 해독 불가 코드 -> 해독 가능 코드.

L. 코드 표

- ① 인코딩/디코딩 하기 위한, 문자들이 1 대 1 대응된 표.
- ② 컴퓨터에 사용되는 코드 표들은 2진수와 문자가 대응된 경우가 많다. (아스키 코드나 UTF-8 등)

M. 컴퓨터 워드

- ① 정의 : CPU가 한번에 처리 가능한 데이터 크기.
- ② 지금 시대에는 일반적으로 32bit 혹은 64bit임. CPU마다 컴퓨터 워드의 크기가 다르다.
- ③ 64bit CPU가 처리 속도 면에서 우월할 수도 있으나, 작은 일을 수행할 때 낭비가 있을 수 있다. 물론 64bit라고 항상 우월한 것은 아니다.

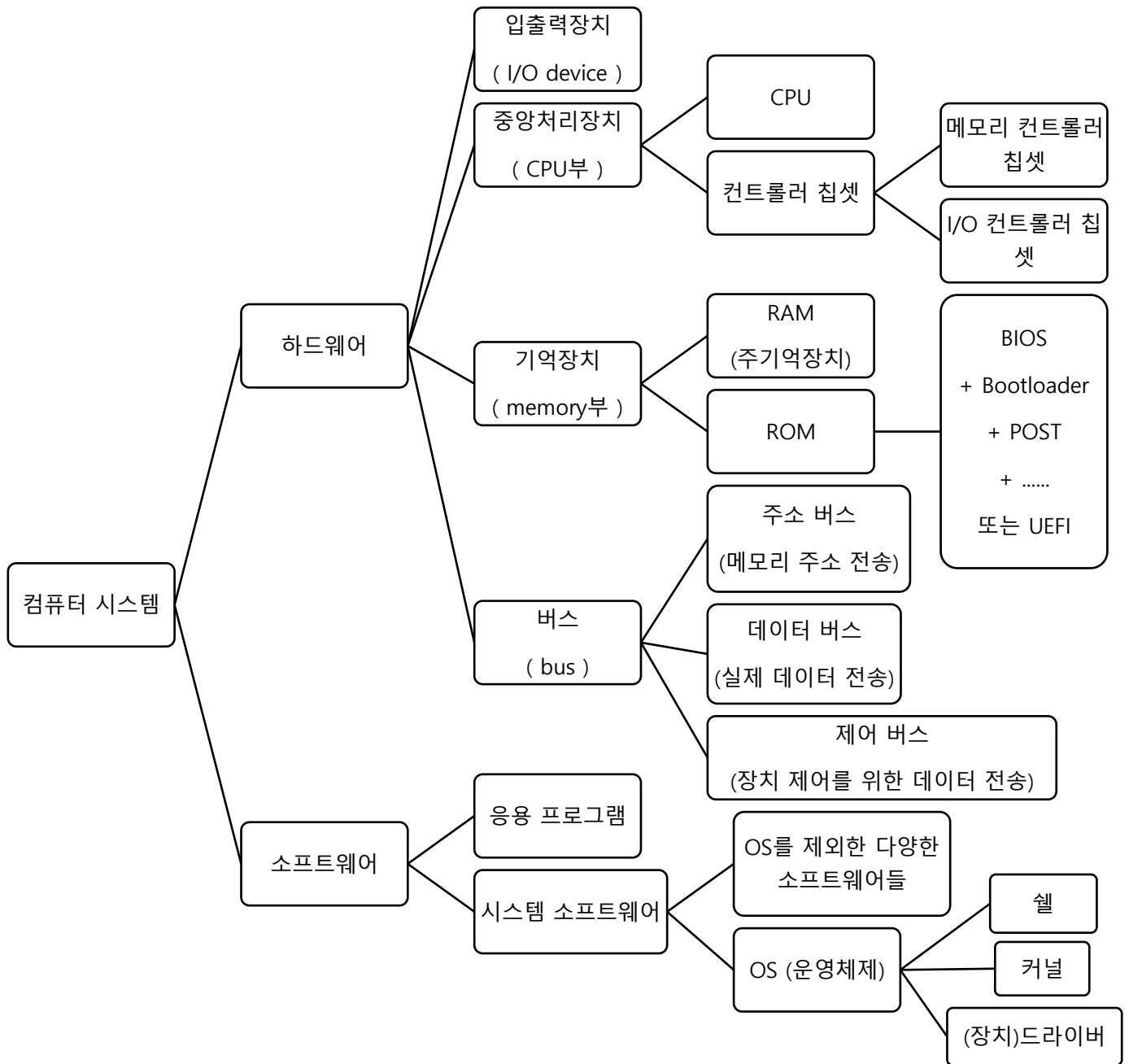
N. 처리(Processing) : 데이터를 가공하는 것. 제거, 수정, 작성 등등 모두 처리다.

O. ISO / ANSI : 규약을 정하는 표준화 기구들.

- ① ISO : International Organization for Standardization (국제 표준화 기구)
- ② ANSI : American National Standards Institute (미국 국가 표준 협회)

2. 컴퓨터의 기본 구조와 동작 (교재 2장 내용 정리)

본격적인 정리 이전에, 이 그림을 보라.



A. 칩(Chip) : 얇고 작은 무언가.

① 칩셋(Chip-set) : 얇고 작은 반도체 집합체.

B. RAM : Random Access Memory

① 이진탐색으로 빈 공간을 찾아서 쓰기를 수행한다. 즉, 빠르다.

C. ROM : Read Only Memory

- ① 물리적으로는 읽기 쓰기가 가능한 메모리이나, 논리적으로 쓰기가 불가능하도록 막아 둔 것이다. 내장된 시스템 소프트웨어의 손상을 막기 위한 조치이다.
- ② 반대는 RWM(Read Write Memory).

D. 인터페이스(Interface) : 접속부. 데이터를 주고 받는 통로.

- ① 모든 장치는 인터페이스를 갖는다. 물리적이든 논리적이든 말이다.
- ② 무엇이든 입력과 출력이 만나는 접속부는 모두 인터페이스이다.
- ③ 소켓이란 용어와 유사하다. 웹 소켓이나, PCI-e 슬롯 소켓, 배터리 소켓 등의 그 단어!

E. 버스(Bus) : 이진 신호가 실제로 이동하는 물리적 신호선들의 집합. 3개의 종류가 있다.

- ① 버스를 나누는 이유 : 모든 버스에서 동일한 이진 전기 신호가 이동하기 때문에, 중요한 데이터들을 따로 입출력 하도록 분리시킨 것이다.
- ② 주소 버스 : CPU가 외부로 내보내는 주소 신호를 전송하는데 사용.
- ③ 데이터 버스 : CPU가 메모리나 I/O장치와 데이터를 주고받는데 사용.
- ④ 제어 버스 : CPU내외부의 각종 장치를 동작시키는데 사용.
- ⑤ 일반적으로 '버스'라고 하면, 위의 세가지 버스를 묶어서 지칭하는 것.

F. 시스템 버스 / I/O버스

- ① 시스템 버스 : CPU부 내부의 버스를 지칭 + 메모리 컨트롤러와 메모리 사이의 버스를 지칭.
- ② I/O버스 : I/O컨트롤러와 I/O장치들 사이의 버스를 지칭.
- ③ 이때 이 두가지 구분은, 버스의 용도에 따라서 구분해 부르는 것이다. 실제로는 주소/데이터/제어의 세가지 버스를 묶어서 지칭하는 것이다.
- ④ 시스템 버스의 정보 송수신량이 I/O버스보다 많기 때문에 실제로 시스템 버스를 더 빠르게 제작한다. 이는 효율과 경제성의 문제로 인하여 I/O버스를 구분하여 더 낮은 성능으로 제작하는 것이다.

- G. 마이크로 컴퓨터 : 마이크로프로세서를 사용한 컴퓨터. (거의 모든 컴퓨터)
- ① 마이크로프로세서 : 아주 작은 반도체로 만든 집적회로, 중앙처리장치.
- H. API : 인터페이스를 통해서 라이브러리를 제공하는 시스템 소프트웨어.
- ① 라이브러리는 사용하지 않는 코드도 모두 바이너리 코드로 다운로드 받아서 사용해야 하는 반면, API는 필요한 기능만 요청해서 사용할 수 있다는 차이가 있다.
 - ② API를 활용하여 응용 프로그램 개발 시, 일관된 사용자 인터페이스를 유지하기 쉽다.
 - ③ Application Programming Interface - 응용 프로그래밍 인터페이스.
- I. 클립보드 : 운영체제에서 관리하는 임시 저장소.
- J. 언어 : 서로 이해 가능한 정보(데이터)를 주고받기 위한 도구.
- ① 언어는 곧 규칙이며 규약이고, Protocol이다.
- K. 프로그래밍 언어 : 컴퓨터에게 명령을 내리기 위해 만들어진 언어.
- ① 고급언어 : 인간 친화적.
 - ② 저급언어 : 기계 친화적.
 - ③ 중요한 것은, 체계적으로 정확하게 명령을 내릴 수 있는 언어라는 것!
- L. 컴퓨터 언어 : 컴퓨터와의 의사소통에 사용될 수 있는 모든 언어.
- ① 프로그래밍 언어를 포함하며, 기계어나 어셈블리어를 포괄하는 개념이다.
- M. 컴퓨터 시스템 시동 과정
- ① 스위치의 신호가 ROM의 시동 소프트웨어 실행.
 - ② 시동 1-POST(power-on self-test)프로그램으로 하드웨어 테스트 수행.
 - ③ 시동 2-주기억장치에 BIOS를 복사.
 - ④ 시동 3-부팅. OS를 메모리로 읽어 들임. (부트섹터에서 부트스트랩 로더를 찾아서 실행)

N. 시스템 소프트웨어

- ① 정의 : 사용자가 시키는 일(Task)을 수행하는 응용 프로그램을 작동할 수 있게 하는 소프트웨어.
- ② 여러 종류가 있다. 하지만 그 종류를 외울 필요는 없다. 중요한 것은, 시스템 소프트웨어가 ‘응용 프로그램’의 동작이 가능하게끔 하는 소프트웨어라는 것이다.
 - i. 사용자와 응용 프로그램 사이를 매개.
 - ii. 사용자와 하드웨어 사이를 매개.
 - iii. 응용 프로그램과 하드웨어 사이를 매개.

O. 운영체제(OS)

- ① 하드웨어와 다른 소프트웨어를 관리하고, 정보의 매개를 수행하는 소프트웨어.
- ② 다양한 소프트웨어를 내장하며, 3가지 대표적인 프로그램을 내장한다.
 - i. 장치 드라이버 : 특정한 하드웨어와 어떻게 상호작용 해야 하는지를 모아둔 명령어 집합(OS와 장치를 연결시키는 소프트웨어). OS에 내장된 시스템 소프트웨어이다. (논리적 장치다)
 - ii. 셸 : 커널을 감싸고 있는 외부 층. 사용자와 커널을 매개하는 인터페이스로서 만들어진 프로그램. 명령어 해석기(사용자의 명령어를 해석하기에)라고도 함.
 - iii. 커널 : 운영체제의 핵심 부분. 이하 항목에서 설명하겠음.

P. 커널(Kernel)

- ① 정의 : OS가 하드웨어와 소프트웨어를 관리하기 위해서 필요한 핵심 부분. OS의 일부분임을 잊지 마라.
- ② 주기억장치에 항상 상주함.
 - i. 실행 : 보조기억장치의 프로그램을 주기억장치로 옮기는 것.
 - ii. 설치 : (OS와 CPU에 맞는 명령어로 구성된 명령어만 골라서) 응용 프로그램을 다른 컴퓨터에서 옮겨오는 것.
 - iii. 소프트웨어의 주기억장치 상주 : 실행된 이후, 계속 메모리 점유.

Q. 플랫폼(Platform) : 기반. 편평한 곳.

- ① 정의 : 무언가 이동하기 위한 시작과 끝을 제공하는 기반.
- ② 예 : 유튜브 - 사용자가 영상을 서버에 올리고 내릴 수 있는 플랫폼
- ③ 예2 : 메인보드 - 전기신호의 물리적 플랫폼.
- ④ 예3 : OS - 응용 소프트웨어와 사용자가 정보를 주고받을 수 있게 하는 논리적인 플랫폼.

R. 허브(Hub) : 활동의 중심지. 무엇인가 모여있는 장소.

- ① 예 : 메인보드는 컴퓨터 시스템의 허브. 전기신호의 허브.

S. 웹 기반 앱의 장점(클라우드 컴퓨팅)

- ① 플랫폼 독립적이다. 어디서든 동작. Java가 JVM에서 동작하듯이, 웹 앱은 웹 브라우저에서 동작.

T. 시뮬레이션(Simulation) / 에뮬레이션(Emulation)

- ① 시뮬레이션 : 소프트웨어적 모의 실험.
- ② 에뮬레이션 : 하드웨어를 논리적으로 모방.
 - i. 에뮬레이터(Emulator) : 하드웨어를 가상으로 구현하는 프로그램.
 - ii. Java의 JVM도 에뮬레이터이고, VMware도 에뮬레이터이다.
 - iii. 이는 하드웨어의 리소스를 사용하여 또다른 하드웨어의 동작을 계산하는 것이다.

U. 스레드(Thread) : 단어 - 실. 이야기 줄거리.

- ① 정의 : 프로그램마다 가진 처리해야 할 일들에 대한 일정표. 동시에 프로그램이 실행될 때 내부적으로 순차적으로 명령을 수행하는 흐름의 단위. 하나의 통로. 동시에 복수의 명령을 순환하며 실행 가능하도록 하는, 동일 명령의 작업을 연결하는 기록물.

V. 스케줄러(scheduler)

- ① 여러 작업이 시스템 자원을 공유할 때, 작업 순서와 일정을 조정해주는 프로그램.
- ② 스케줄 : 메모리 주소의 집합체인 시간표.

W. 버퍼 : 메모리상의 임시 저장소. 데이터의 충돌을 방지해준다.

- ① 송수신 장치의 속도를 완충해줄 수 있다.
- ② 대부분의 소프트웨어 작업은 임시로 데이터를 저장할 버퍼를 필요로 한다.
- ③ 양방향으로 송수신이 가능한 버퍼의 경우, 양측에서 번갈아 전송을 요구할 때, 임시로 신호를 저장할 버퍼가 필요하다.

이상. 시험범위에 속하는 내용의 정리가 끝났다.

내 필기를 바탕으로 중요해 보이는 부분만 모아서 정리해 본 것이다.

교재 : <컴퓨터 구조론> 윤남일, 이강우 지음 / 생능출판.

범위 : p.16 ~ 106

작성일 : 2023년 10월 20일

(*)

