

01. Computer System Overview

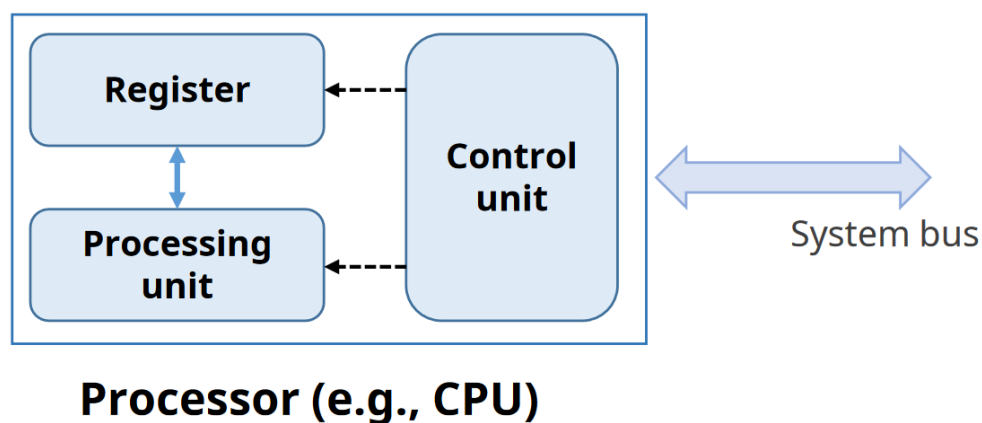
- 운영체제란?
 - 컴퓨팅 자원을 효율적으로 관리해서 사용자/응용 프로그램에 서비스를 제공하는 소프트웨어

컴퓨터 하드웨어

1. 프로세서: 연산
 - CPU, GPU, 응용 전용 처리 장치 등
2. 메모리: 저장
 - 주 기억장치, 보조 기억 장치 등 (ex. DRAM, 디스크)
3. 주변 장치
 - 키보드/마우스 (입력장치), 모니터/프린터 (출력장치)

프로세서

| 중앙 처리 장치: 연산 수행 및 컴퓨터의 모든 장치의 동작 제어



레지스터

- 프로세서 내부의 메모리
 - 특징: 프로세서가 사용할 데이터 저장, 컴퓨터 메모리 중 제일 빠름
 - 종류

- 용도에 따른 분류: 전용/범용 레지스터
- 사용자가 정보 변경 가능 여부에 따른 분류: 사용자 가시/불가시 레지스터
- 저장하는 정보의 종류에 따른 분류: 데이터/주소/상태 레지스터

[레지스터 종류]

- 사용자 가시 레지스터
 - 데이터 레지스터: 함수 연산에 필요한 데이터 저장
 - 주소 레지스터
- 사용자 불가시 레지스터
 - 프로그램 카운터(Program Counter): 다음에 실행할 명령어의 주소를 보관하는 레지스터
 - 명령어 레지스터(Instruction Register): 현재 실행하는 명령어의 주소를 보관하는 레지스터
 - 누산기(ACC): 데이터를 일시적으로 저장하는 레지스터

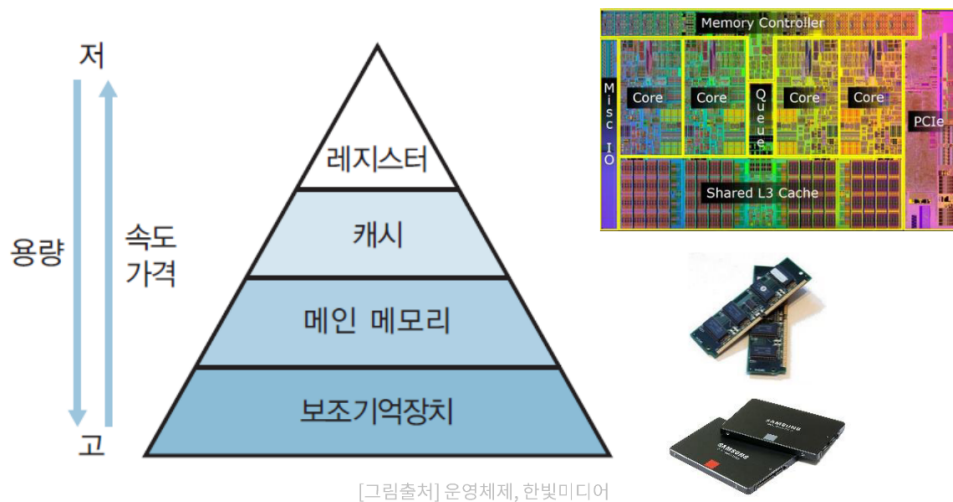
운영체제와 프로세서의 관계

- 운영체제는 프로세서에게 처리할 작업을 할당 및 관리한다.
 - 프로세스 생성 및 관리 (Process Management)
- 운영체제는 프로그램의 프로세서 사용을 제어한다.
 - 프로그램의 프로세서 사용 시간 관리, 복수 프로그램 간 사용 시간 조율 등 (Thread Management, Process Scheduling)

메모리

| 기억 장치: 데이터를 저장하는 장치

메모리의 종류



1. 레지스터
2. 캐시: CPU
3. 메인 메모리: DRAM
4. 보조 기억 장치: HDD
 - 프로그램과 데이터를 메인 메모리에 옮겨야 실행 가능

메인 메모리 (주기억장치)

- 프로세서가 수행할 프로그램 및 데이터 저장
 - 프로세서가 직접 접근 가능한 마지막 계층
- CPU와 디스크 속도 차이 때문에 발생하는 디스크 입출력 병목 현상(I/O bottleneck)을 중간에서 해소하는 역할 수행 (CPU ↔ 메인 메모리 ↔ 디스크)

캐시

- CPU 내부의 메모리
- 메인 메모리 입출력 병목 현상 해소 (CPU ↔ 캐시 ↔ 메인 메모리)

[캐시의 동작]

- 일반적으로 HW적으로 관리됨
- Cache Hit: 프로세서에서 필요한 데이터 블록이 캐시에 존재하는 경우, 메인 메모리에서 가져올 필요 X, 시간 감소

- Cache Miss: 필요한 데이터 블록이 캐시에 존재하지 않는 경우, 메인 메모리에서 캐시로, 또 캐시에서 프로세서로 전달

[캐시가 효과적인 이유: 지역성]

- Spatial Locality: 공간적 지역성
 - 참조한 주소와 인접한 주소를 참조할 가능성이 높다는 특징 ex) 순차적 프로그램 수행
- Temporal Locality: 시간적 지역성
 - 한번 참조한 주소를 곧 다시 참조하는 특성 ex) for문
- 지역성은 캐시 적중률(cache hit ratio와 밀접)
 - ex) 캐시를 가져올 때, cache line(block) 단위로 가져옴. 프로그래밍할 때 캐시 미스가 나지 않도록 $a[c][r]$ 형식이 아니라 $a[r][c]$ 형식으로 접근하는 것이 좋음

보조 기억 장치

- 프로그램과 데이터 저장
- 프로세서 직접 접근 불가 (주변 장치)
 - 주기억장치를 거쳐서 접근
 - 데이터를 메모리에 올려두고 사용할 수 있음
- 프로그램/데이터가 주기억장치보다 큰 경우: 가상 메모리 활용. 하드디스크 일부를 메모리처럼 활용

메모리와 운영체제

- 메모리 할당 및 관리
 - 프로그램 요청에 따른 메모리 할당 및 회수, 할당된 메모리 관리 Deadlock
- 가상 메모리 관리
 - 가상 메모리 생성 및 관리, 논리주소 → 물리주소 변환

시스템 버스

- 하드웨어(프로세서, 메인메모리, 주변장치 사이) 간 데이터 및 신호를 주고받는 물리적인 통로
 - 데이터 버스: 데이터 전송
 - 주소 버스: 시스템 구성 요소 식별 위한 주소 전송

- 제어 버스: 시스템 구성 요소 제어 위해 사용, 제어 신호를 사용해 연산 및 메인 메모리의 R/W 작업 결정
- ex) 시스템 버스 동작 과정
 - PC에 저장된 명령어를 프로세서 내부 버스를 사용해 MAR에 전달
 - MAR에서 주소 버스에 전달 → 주소 버스 통해서 메모리로 전달
 - 제어 장치가 메모리 Read 위한 제어 신호 발생
 - Read한 데이터를 데이터 버스로 MBR로 이동, 명령어 저장 후 IR로 전달
 - IR에서 또 CPU로 다시 전달

주변 장치와 운영체제

- 장치 드라이버 관리: 주변 장치 사용을 위한 인터페이스 관리
- 인터럽트 처리: 주변 장치의 요청 처리
- 파일 및 디스크 관리: 파일 생성 및 삭제, 디스크 공간 관리 등