Chapter 1 컴퓨터 시스템 기본

01 컴퓨터 시스템 구성

* 컴퓨터 시스템은 최종 사용자, 응용 소프트웨어, 시스템 소프트웨어, 하드웨어 자원, 운영 체제의 다섯 가지 구성 요소로 나눌 수 있다.
* **최종 사용자**: 일반 사용자, 프로그래머, 운영자, 다른 컴퓨터를 일컫는 말
* **응용 소프트웨어**: 최종 사용자에게 일반적인 도구를 제공하는 프로그램 (MS Office 등)
* **시스템 소프트웨어**: 특정 응용 프로그램을 만들 수 있게 프로그래밍 환경을 제공하는 프로그램 (컴파일러)
* **운영 체제**: 시스템의 효율을 극대화하려고 H/W와 S/W 자원을 관리하는 프로그램
* **하드웨어 자원**은 CPU, 기억, 입출력 장치 등을 포함한다.

02 컴퓨터 하드웨어 기본 구성

* **중앙 처리 장치(CPU)**: ‘프로그램 실행’과 ‘데이터 처리’라는 중추적인 기능의 수행을 담당하는 요소. **마이크로프로세서**는 CPU를 단일의 IC칩에 집적한 여러 형태 중 하나이다. 개인용 컴퓨터에서 주로 이용한다. **마이크로 컨트롤러**는 단일 칩은 물론, 메모리(ROM, RAM)와 입출력 제어 인터페이스 회로까지 내장한 것이다.
* **기억 장치**: CPU가 실행할 프로그램과 데이터를 저장하는 장치
* **주기억 장치**: 1차 기억장치, 고속 액세스, 가격 높고 면적 넓음, 용량 한계, 휘발성
* **보조기억 장치**: 2차 기억장치, 저속 액세스, 가격 낮고 저장 밀도 높음, 영구 저장
* **순차적 액세스**: 처음부터 순서대로 액세스 (자기 테이프)
* **직접 액세스**: 직접 이동한 다음, 순차적 검색 통하여 위치에 도달하는 방식 (CD-ROM)
* **임의 액세스**: 주소에 의해 직접 장소를 찾아 액세스. 어떤 장소든 시간 동일 (반도체)
* **연관 액세스**: 저장된 내용의 특정 비트들을 비교하여 일치하는 내용을 액세스 (연관 기억 장치)
* **DRAM(Dynamic)**: 캐피시터에 전하를 충전하는 방식. 시간이 지나면 방전하므로 주기적인 재충전이 필요하다. 밀도 높고, 가격 낮다. 그렇기에 주기억 장치로 사용한다.
* **SRAM(Static)**: 플립플롭을 이용한다. 재충전 없이 데이터 유지 가능, 속도 빠름 (캐쉬)
* 계층적 기억 장치 시스템: 가장 중요한 요소는 **용량**, **속도**, **가격**이다.
* **내부 기억 장치**: CPU가 직접 액세스 **(CPU 레지스터, 캐쉬, 주기억 장치, 디스크 캐쉬)**
* **외부 기억 장치**: CPU가 직접 액세스 X **(디스크, 자기 테이프)**
* **버스**: 정보(데이터, 주소, 제어 명령)를 주고받을 수 있게 해 주는 통로
* **내부 버스**: 컴퓨터 프로세스 내부에서 레지스터, 연산 장치, 메모리와의 연결 담당
* **외부 버스**: 프로세서와 메모리, 입출력 장치 간 연결 담당. 보통 시스템 버스라 부름.
* **시스템 버스**: CPU와 시스템 내의 다른 요소들 사이에 정보를 교환하는 통로
* **주소 버스**: 주소 정보를 전송하기 위한 주소 선들의 집합. 단방향성 버스.

주소 버스의 비트 수가 16이면 2^16개의 주소를 지정 가능하다.

* **데이터 버스**: 데이터를 전송하기 위한 신호 선들의 집합. 양방향 버스.

데이터 버스의 폭이 32비트면 데이터 전송은 32 비트씩 가능하다.

* **제어 버스**: 제어하기 위한 신호 선들의 집합. 단방향 버스 (기억 장치 읽기/쓰기 신호, I/O 읽기/쓰기 신호)
* CPU가 기억 장치의 특정 장소에 데이터를 저장하고나 참조하는 동작을 **액세스**라고 한다.
* CPU의 기능
* 명령어 인출: 기억 장치로부터 명령어를 인출.
* 명령어 해독: 인출된 명령어를 해독.
* 데이터 인출: (데이터가 필요한 경우) 기억 장치, 입출력 장치로부터 데이터를 인출.
* 데이터 처리: 데이터에 대한 산술적, 논리적 연산을 수행.
* 데이터 저장: 수행한 결과를 저장.
* CPU는 산술 논리 장치, 레지스터 세트, 제어 유니트(제어 장치), 내부 CPU 버스로 구성
* 산술 논리 장치: 각종 산술 연산들과 논리 연산들을 수행하는 회로
* 레지스터 세트: CPU 내부 레지스터들의 집합. 기억 장치들 중에서 액세스 속도 가장 빠름.
* 프로그램 카운터(PC): 다음에 인출할 명령어를 가지고 있는 레지스터. 분기(루프, 반복문) 등 명령어가 실행될 때 목적지 주소로 갱신된다.
* 누산기: 데이터를 일시적으로 저장하는 레지스터 (누적해서 계산)
* 명령어 레지스터: 가장 최근에 인출된 명령어 코드가 저장되어 있는 레지스터
* 기억 장치 주소 레지스터: 저장된 명령어 주소가 출력되기 전에 일시적으로 저장
* 기억 장치 버퍼 레지스터: 쓰여질 혹은 읽혀진 데이터를 일시적으로 저장
* 명령어 형식: 연산 코드 + 오퍼랜드
* 연산 코드: CPU가 수행할 연산을 지정, 오퍼랜드: 필요한 데이터가 저장된 주소를 지정
* 명령어 사이클: 인출, 실행 사이클 두 개로 구성
* 실행 사이클: 데이터 이동, 데이터 처리(연산 수행), 데이터 저장, 제어(실행 순서 결정)
* 인터럽트: 현재 실행 중인 프로그램의 수행을 연기하고 다른 프로그램의 수행을 요구하는 명령, 시스템의 처리 효율을 향상시키기 위해 사용, 다중 프로그래밍에 사용

Chapter 2 운영 체제 개관

01 운영 체제 개념

* 운영 체제는 H/W와 사용자 사이에 위치하여 사용자 대신에 컴퓨터 시스템의 각종 자원을 효율적으로 관리하고 운영할 수 있도록 도와주는 시스템 프로그램이다.
* 운영 체제의 목적
  + 편리성 제공
  + 처리 능력의 향상
  + 응답 시간 또는 반환 시간의 최소화
  + 사용 가능도의 향상
  + 신뢰도의 향상
* 운영 체제의 역할
  + 조정자: 적절하게 사용할 수 있도록 제어
  + 자원 할당자 또는 관리자
  + 입출력 장치와 사용자 프로그램 제어

02 운영 체제의 구성과 기능

* 슈퍼바이저 프로그램: 모든 동작 상태를 관리하고 감독하는 제어 프로그램
* 작업 관리 프로그램: 어떤 작업을 지시하는 명령을 해독, 준비, 마무리 담당 및 연속적인 작업이 이루어지게 통제
* 데이터 관리 프로그램: 입출력되는 데이터를 표준적으로 관리, 입출력 장치를 쉽게 사용
* 통신 관리 프로그램: CPU와 연결된 주변 장치간의 신호 교환이 원활히 이루어지게 통제
* 언어 번역 프로그램: 프로그래밍 언어 -> 기계어 (컴파일러)
* 서비스 프로그램: 사용자가 더 편리하게 컴퓨터를 사용할 수 있게 지원하는 프로그램
  + 연계 편집 프로그램: 서로 독립되어 컴파일된 여러 개의 목적 프로그램을 하나의 실행할 수 있는 로드 모듈로 만드는 기능을 수행
  + 매크로: 반복적으로 나타나는 명령들을 매번 반복적으로 열거하여 프로그램을 작성하지 않고, 명령들을 하나의 새로운 명령으로 정의할 수 있게 해 주는 기법

매크로 정의, 매크로 이름(명령), 매크로 호출, 매크로 확장

* + 로더: 컴퓨터가 프로그램을 수행할 수 있게 기억 장치에 적재하고 수행을 시작시키는 프로그램

할당(배정), 링킹(타 프로그램 연결), 재배치(주소 조정), 로딩(기억 장소에 적재)

* + 정렬 및 병합 프로그램: 어떤 값에 따라서 항목을 정렬하거나 또는 여러 개의 정렬된 순차 파일을 일정한 규칙에 따라 하나의 파일로 병합하는 기능
* 유틸리티 프로그램: 공통적으로 필요한 분야를 표준화시켜서 작성하여 공동으로 사용할 수 있게 하는 서비스 프로그램
  + 디버깅 유틸리티: 프로그램의 오류를 찾아주는 프로그램
* 자원 관리자의 관점 기능 (자원을 관리하는 수행 과정)
  + 1. 시스템 내 자원의 상태를 추적, 기억
  + 2. 어떤 CPU에게 어떤 자원을 얼마나 할당할 것인가를 결정하려는 정책 수립
  + 3. 요구된 자원 할당
  + 4. 사용이 끝난 자원 회수
  + 프로세스 관리 기능
  + 기억 장치 관리 기능
  + 장치 관리 기능
  + 정보 관리 기능
* 프로세스 관점의 기능: CPU가 처리할 수 있는 작업은 어느 한 시점에서 볼 때 오직 하나의 작업뿐이다.
* 계층 구조 관점의 기능
  + 커널 기능: 기계어 제공과 커널 기능으로 많은 능력을 제공함으로써 추가 능력 부여
  + 계층적 설계 이유: 오류 수정을 쉽게 하려고 사용. 상위 계층에서 하위 계층으로의 호출만 허용
    - 레벨 1 (프로세서 관리): 동기화, 스케줄링을 위한 프로세서 관리
    - 레벨 2 (기억 장치 관리)
    - 레벨 3 (프로세스 관리): 프로세스의 생성, 제거, 메시지 전달, 시작, 정지 등
    - 레벨 4 (주변 장치 관리)
    - 레벨 5 (파일과 데이터 관리)

03 운영 체제 서비스

3.1 부팅 서비스

* 부팅: 시스템을 시동하거나 재시동하는 작업, 웜 부팅(켜진 상태), 콜드 부팅(꺼진 상태)
* 부팅은 운영 체제를 디스크로부터 주기억 장치로 적재시키는 과정

3.2 사용자 서비스

* 사용자 인터페이스 제공
* 프로그램 수행 (프로그램들이 정상적, 비정상적 종료 방법 제공)
* 입출력 동작 (입출력 동작 방법 운영 체제에서 제공)
* 파일 시스템 조작 (파일 읽기/쓰기, 생성/삭제)
* 통신
* 오류 발견 (오류 종류에 대해 인터럽트 처리 방식 도입하여 적당한 조치 취함)

3.3 시스템 서비스

* 컴퓨터 자원의 관리: 운영 체제는 자원 추상화와 자원 공유 기법을 통해 효율적으로 사용할 수 있도록 한다.
* 자원 할당
* 보호: 다중 사용자 시스템 내에서 정보 보호, 시스템 보안
* 계정: 기록 보관, 시스템 사용 통계

04 운영 체제의 종류

4.1 일괄 처리 시스템

* 일괄 처리 시스템: 즉각적으로 처리하는 것이 아니라 일정 기간 또는 일정량 모아 두었다가 한꺼번에 처리하는 방식
  + 컴퓨터 시스템을 효율적으로 사용할 수 있다.
  + 작업 시간이 길어 반환 시간이 늦다.
  + 적절한 작업 관리 기능을 제공해야 한다.
  + 프로그램의 오류 수정 작업이 어렵다.
  + CPU가 유휴 상태로 되기 쉽다.

4.2 온라인 시스템

* 사용자가 운영 체제나 프로그램에 직접 명령을 주고 즉시 응답을 받을 수 있는 시스템
  + 사용자는 프로그램을 쉽게 실험할 수 있다.
  + 실행 결과를 곧바로 받아 볼 수 있어 응답 시간이 짧다.
  + 바로 실행 결과가 출력되므로 프로그래머에게 유효성을 제공해준다.
  + 이러한 기능 때문에 CPU는 상당한 유휴 시간이 생긴다.

4.3 시분할 시스템

* 동시에 많은 사용자들이 컴퓨터를 공유하도록 해준다.
* 응답 시간을 최소화하여, 각 사용자로 하여금 독점하도록 느끼게 만드는 것
  + 여러 사람이 공동으로 CPU를 사용할 수 있다.
  + 대부분 책임을 운영 체제에 전가시킴으로써 운영 체제를 복잡하게 한다.
  + CPU 할당 시간을 정의하여 스케줄링 해야 한다.
  + 실행 중인 프로세스 간의 통신 및 I/O 장치 사용의 제어와 관리 능력을 가져야 한다.
  + 여러 개의 프로그램을 주기억 장치에 적재시킨다. (스와핑)

4.4 다중 프로그래밍 시스템

* CPU의 효율을 극대화하려는 방법, 여러 개의 사용자 프로그램이 마치 동시에 실행되는 것처럼 처리하는 방식
  + CPU 1개, 주기억 장소에 여러 개의 프로그램 존재하여 다중 작업 구현
  + 일괄 처리의 반환 시간이 길어서 CPU의 유휴 시간이 길어진다.
  + 기억 장치 관리 기법이나 CPU 스케줄링 기법이 필요하다.
  + 스케줄링, 교착 상태, 병행 제어와 보호 문제 등이 신중히 고려되어야 한다.
  + 하나의 프로세스의 성능을 향상시키지는 못하지만 전반적인 시스템 성능을 향상

4.5 다중 처리 시스템

* CPU가 다수 존재하여 다중 작업을 구현
  + 강 결합 다중 처리 시스템: 매우 밀접하게 동기화해야 하는 시스템
  + 약 결합 다중 처리 시스템: 높은 작업의 처리량을 제공하는 데 목적
  + CPU를 여러 개 사용하여 작업 속도와 신뢰성을 높일 수 있다.
  + 스케줄링 결정
  + CPU 간의 문제 고려

4.6 실시간 시스템

* 자료가 발생할 때마다 즉시 처리하여 응답하는 방식 (특수 목적)
  + 단말기를 통하여 직접 입출력되므로 사용자의 노력이 절감된다.
  + 처리 시간이 단축된다.
  + 처리 비용이 절감된다.
  + 무작위하므로 입출력 자료의 일시 저장과 대기가 필요하다.
  + 특정 상태의 재현을 할 수 없다.
  + 장애가 발생할 때 단순한 재실행을 할 수 없다.

4.7 분산 처리 시스템

* 하나의 대형 컴퓨터에서 수행하던 기능을 지역적으로 분산된 여러 개의 미니 컴퓨터에 분담시킨 후 통신망을 통하여 교신하고 처리하는 시스템
  + 자원 공유
  + 계산 속도 증가
  + 신뢰성 향상
  + 통신 기능

Chapter 3 프로세스 관리와 스레드

01 프로세스의 개념

* 프로세스란 CPU가 현재 수행하고 있는 사용자 프로그램과 시스템 프로그램으로, 시스템의 작업 단위를 가리킨다.
  1. 프로세스 정의
* 프로세스: 작업의 기본 단위로 프로그램에 입출력 상태를 결합한 형태를 갖는다.
  + 현재 실행 중이거나 곧 실행할 수 있는 PCB를 가진 프로그램
  + 실행 중인 프로시저를 말한다.
  + 실행 중인 프로그램을 말한다.
  + 비동기적인 행위
  + 프로세서에 할당되어 실행되는 개체로 디스패치를 할 수 있는 대상
  1. 프로세스 종류
* 순차 프로세스: 현재 실행 중인 하나의 프로세스
* 독립 프로세스: 다른 프로세스에 영향을 주고 받지 않으며, 데이터 공유X
* 협동 프로세스: 다른 프로세스에 영향 주고 받으며, 데이터 공유O
* 시스템 프로세스: 실행 순서를 제어하고, 감시하는 프로세스
* 사용자 프로세스: 사용자 프로그램을 수행하는 프로세스
  1. 프로세스 상태
* 보류 상태
* 준비 상태
* 실행 상태
* 대기 상태
* 교착 상태
* 완료 상태
* 준비 리스트: 우선 순위O
* 대기 리스트: 우선 순위X
* 디스패치: 준비 상태 -> 실행 상태
* 할당 시간 초과: 실행 상태 -> 준비 상태
* 대기: 실행 상태 -> 대기 상태
* 깨움: 대기 상태 -> 준비 상태

예상문제

1. 컴퓨터 시스템을 5가지로 구분하고 각각 설명을 하시오
2. 운영 체제란?
3. CPU란?
4. 액세스 방법에 따른 기억 장치의 분류
5. 제조 기술에 따른 기억 장치의 분류
6. 버스란?
7. 시스템 버스란?
8. 주소 버스란? 계산 방법?
9. 데이터 버스란? 계산 방법?
10. 제어 버스란?
11. CPU와 기억 장치 간 접속 순서 (읽을 때, 쓸 때)
12. CPU의 기능
13. 레지스터 세트 5개 쓰고, 각각 설명
14. 명령어의 구성?
15. 인터럽트란?
16. 운영 체제의 목적?
17. 운영 체제의 역할 1문장, 3개
18. 슈퍼바이저 프로그램?
19. 자원 관리자의 관점 기능
20. 부팅이란?
21. 사용자 서비스 6개
22. 시스템 서비스 4개
23. 일괄 처리 시스템이란? 장단점?
24. 온라인 시스템이란? 장단점?
25. 시분할 시스템이란? 장단점?
26. 다중 프로그래밍 시스템이란? 장단점?
27. 다중 처리 시스템이란? 장단점?
28. 실시간 시스템이란? 장단점?
29. 분산 처리 시스템이란? 장단점?
30. 프로세스의 정의? 5가지
31. 순차 프로세스란?
32. 독립 프로세스와 협동 프로세스의 차이점
33. 시스템 프로세스란?
34. 프로세스 상태 6가지
35. 디스패치란?