

5주 1강

컴퓨터 시스템의 소프트웨어 (1)



3. 컴퓨터 시스템의 소프트웨어

◆ 운영체제의 개념과 역할

■ 운영체제

- 컴퓨터의 모든 움직임을 제어하는 시스템 소프트웨어의 한 종류
- 시스템이 가지고 있는 자원(프로세서, 기억장치, 입출력 장치, 파일 및 데이터, 분산 및 보호 기능)들을 효율적으로 관리
- 사용자와 컴퓨터 시스템 간의 편리한 인터페이스 기능을 제공
- 사용자 간의 하드웨어 공동 사용을 허용

3. 컴퓨터 시스템의 소프트웨어

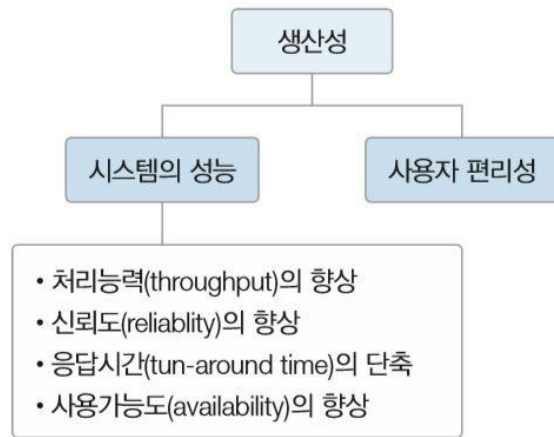
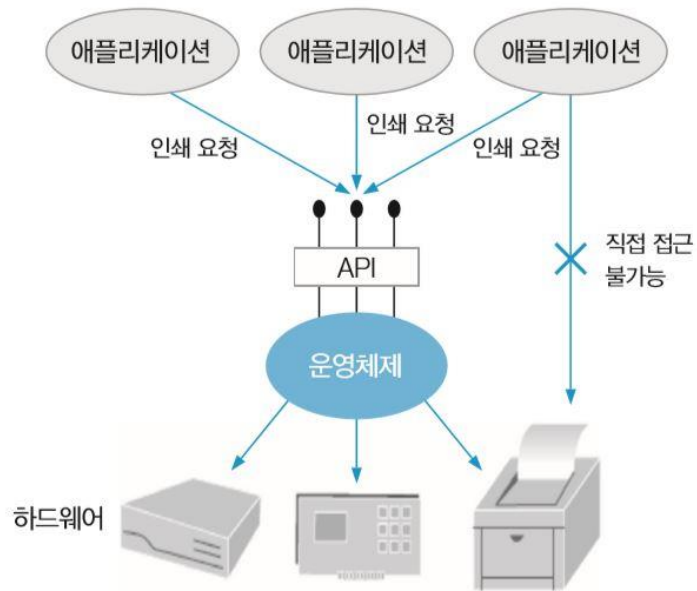


그림 3-23 운영체제의 역할

3. 컴퓨터 시스템의 소프트웨어

◆ 운영체제의 개념과 역할

■ 운영체제

- 기본적인 기능은 컴퓨터 시스템 자원 관리
- 주로 프로세서와 메모리, 주변장치, 파일 등을 관리
- 부팅 작업과 사용자 인터페이스 작업, 작업 관리

3. 컴퓨터 시스템의 소프트웨어

◆ 운영체제의 개념과 역할

■ 운영체제

- 다중 프로세싱
 - 여러 개의 프로세서를 사용하는 것
- 다중 태스킹
 - 같은 시간에 여러 개의 프로그램을 띄우는 것
- 다중 프로그래밍
 - 여러 개의 프로그램들이 단일 프로세서 상에서 동시에 실행되는 것
 - 진정한 의미로는 여러 개의 프로그램이 동시에 수행된다고 볼 수 없음
 - 운영체제의 통제 하에 한 프로그램이 일부 수행되고 나서, 또 다른 프로그램이 일부 수행되는 식으로 진행
- 교착상태
 - 다중 프로그래밍 환경에서 흔히 발생
 - 두 개 이상의 작업이 서로 상대방의 작업이 끝나기만을 기다리고 있어 결과적으로 아무것도 완료되지 못하는 상태

3. 컴퓨터 시스템의 소프트웨어

◆ 운영체제의 역사

▪ 최초의 운영체제

- 제너럴 모터스 연구소, 1950년대 초반에 처음 개발
- IBM-701 컴퓨터에서 사용목적

표 3-9 운영체제의 역사

구분(연대)	1950	1960	1960년대 후반 ~ 1970년대 중반	1970년대 후반 ~ 현재
세대	1세대	2세대	3세대	4세대
이용 시스템	일괄처리 시스템	다중 프로그래밍 다중처리 시스템	다중모드 시스템 시분할처리 시스템	마이크로 프로세서 개발 개인용 컴퓨터 개발

3. 컴퓨터 시스템의 소프트웨어

◆ 운영체제의 구성

- 제어 프로그램과 처리 프로그램으로 나뉨
- 비동기 인터페이스
 - 컴퓨터 본체와 주변장치 간에 직렬 데이터 전송을 하기 위해 사용
 - 컴퓨터 내에 있는 입출력 시스템의 전송속도와 입출력 장치의 속도가 현저하게 다를 때 사용
- 동기 인터페이스
 - 컴퓨터와 입출력 장치가 데이터를 주고받을 때 일정한 클록 신호의 속도에 맞추어 약정된 신호에 의해 동기를 맞추는 방식

3. 컴퓨터 시스템의 소프트웨어

◆ 운영체제의 종류와 점유율

- 대표적으로 마이크로소프트의 윈도우, 애플의 맥 OS, 리눅스 등
- 개인용 컴퓨터 PC가 널리 보급되며 운영체제 OS 시장도 성장
- 현재 전 세계에서 가장 많이 사용되는 OS : 안드로이드
 - PC: windows
 - 스마트폰: 안드로이드
- 스마트폰과 태블릿의 보급확대로 인해 안드로이드 점유율이 windows 점유율 추월

3. 컴퓨터 시스템의 소프트웨어

빌 게이츠(William Henry Gates III, 1955 ~)

- ✓ 마이크로소프트 사의 창업자
- ✓ 컴퓨터 운영체제인 윈도우를 만들어낸 대표적인 시스템 프로그래머
- ✓ 1975년 뉴멕시코 주 앨버커키에서 마이크로소프트사 설립
- ✓ 2007년 미국 ABC 뉴스가 선정한 세계에서 가장 영향력 있는 인물 1위
- ✓ 2007년 포브스지 선정 세계 갑부 순위 1위



3. 컴퓨터 시스템의 소프트웨어

◆ 소프트웨어의 개발 방법

- 넓은 의미 : 소프트웨어 개발 방법
- 좁은 의미 : 프로그램 작성 과정, 번역 과정, 실행 과정 등



그림 3-24 넓은 의미의 프로그래밍 과정(전통적인 개발 방법)

3. 컴퓨터 시스템의 소프트웨어

◆ 소프트웨어의 개발 방법

표 3-11 전통적인 개발 방법에 대한 각 단계의 설명

단계		설명
1단계	문제 분석	소프트웨어에 요구되는 기능, 성능, 인터페이스 등 사용자의 요구사항을 구체적으로 이해하는 단계이다.
2단계	알고리즘 설계	기본 설계와 상세 설계가 있다. 기본 설계는 개발될 소프트웨어에 대한 전체적인 하드웨어 및 소프트웨어 구조, 제어, 제어구조, 자료구조의 개략적인 설계를 작성하는 단계이다. 상세 설계는 각 단위 프로그램에 대한 사항을 상세히 기술하는 단계이다.
3단계	프로그램 코딩	설계 단계에서 만들어진 설계 사양서를 바탕으로 프로그램을 작성하는 단계로, 코딩과 디버깅, 단위 테스트를 수행하는 단계이다.
4단계	프로그램 시험 실행	단위 프로그램별로 구현된 것을 통합시키며 시험하는 단계이다. 전체 시스템이 정확하게 실행하는가를 확인하는 단계이기도 하다.
5단계	프로그램 문서화/유지보수	시스템의 사용 중에 발생하는 여러 변경 사항에 대해 적응하고, 변화에 대비하는 과정이다.

3. 컴퓨터 시스템의 소프트웨어

◆ 컴퓨터 프로그래밍 순서

표 3-13 컴퓨터 프로그래밍 순서

단계		내용
1단계	문제 분석	입력 자료의 종류, 처리 순서, 처리 결과의 형태를 결정한다.
2단계	입출력 설계	처리할 자료를 어떤 매체를 통해 입출력할 것인가를 결정한다.
3단계	순서도 작성	각 요소간의 관계를 표준 기호로 도표화한다.
4단계	프로그램 코딩	프로그래밍 언어로 명령문을 규칙에 따라 작성한다.
5단계	수정 작업	프로그램과 자료를 입력한다.
6단계	입력 매체 수록 작업	프로그램의 오류를 수정하며, 이 과정을 디버깅(debugging)이라 한다.
7단계	프로그램 실행	프로그램을 실행한다.
8단계	평가	프로그램을 평가한다.

3. 컴퓨터 시스템의 소프트웨어

◆ 순서도

- 문제 해결에 필요한 과정, 처리절차, 방법들을 도식화한 것

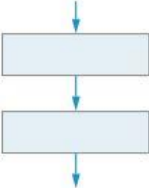
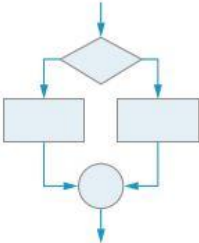
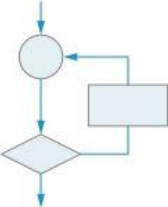
표 3-14 순서도의 기호

기호	의미	기호	의미	기호	의미
	시작/끝		작업 준비		데이터
	처리		비교/판단		흐름선
	온라인 수동입력		문서(출력)		디스플레이
	자기디스크		자기테이프 (순차파일)		온라인 저장매체
	오프라인 저장 매체		연결자		페이지 연결자
	커뮤니케이션 링크		수동 조작		keying peration

3. 컴퓨터 시스템의 소프트웨어

◆ 순서도

표 3-15 순서도의 유형

직선형		가장 단순한 형태로 판단에 의한 분기나 반복 과정 없이 위에서 부터 아래 방향으로 처리가 진행된다. 직사각형을 이용하여 처리 내용을 표현하고 실행 순서는 화살표를 사용한다.
분기형		분기형은 실행 과정에서 발생한 조건에 따라 실행 순서를 결정하는 형태이다. 분기의 조건은 두 요소 값의 증가, 대소, 논리관계에 따른다. 조건에 대한 판단을 표시하는 기호로는 마름모를 사용하며 판단 결과에 따라 참과 거짓으로 분기가 진행된다.
반복형		순서도의 일정한 부분을 조건이 만족할 때까지 반복해서 실행하는 과정을 표시하며 루프(loop)형 순서도라고도 부른다. 반복형은 직선형과 분기형을 이용해서 구성하는데 조건을 만족하는 동안 반복을 계속하는 형태와 조건을 만족할 때까지 반복을 계속하는 두 가지 형태가 있다.

수고하셨습니다.

