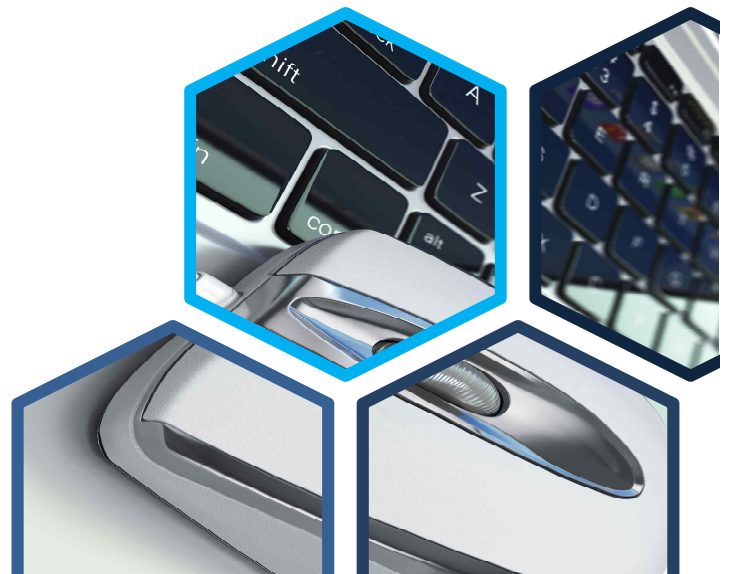


컴퓨터시스템

운영체제





학습목표

- 운영체제가 무엇인지 설명할 수 있다.
- 운영체제의 기능을 설명할 수 있다.
- 운영체제의 서비스에 대해 설명할 수 있다.
- 운영체제의 구조에 대해 설명할 수 있다.



학습내용

- 운영체제의 개요
- 운영체제의 기능
- 운영체제의 서비스
- 운영체제의 구조



운영체제의 개요

1 운영체제의 정의



컴퓨터 자원 관리면에서 운영체제의 정의

조정자

- 운영 요소를 적절하게 사용할 수 있도록 제어함
- 작업을 할 수 있는 환경만 제공하며 직접 일을 하지 않음

자원 할당자 또는 관리자

- 각 응용프로그램에 필요한 자원을 할당하고, 자원 할당 방법을 결정함

응용프로그램과 입출력 장치

- 다양한 입출력 장치와 응용프로그램을 제어함



사용자와 하드웨어 사이의 중간 매개체

응용프로그램의
실행 제어

자원 할당 및 관리

입출력 제어 및
데이터 관리



운영체제의 개요

1 운영체제의 정의



운영체제의 역할

- 하드웨어 및 사용자, 응용프로그램, 시스템 프로그램 사이의 **인터페이스 제공**
- 프로세서, 메모리, 입출력 장치, 통신 장치 등 **컴퓨터 자원**을 효과적으로 활용하기 위한 **조정 및 관리**
- 메일 전송, 파일 시스템 검사, 서버 작업 등 높은 수준의 서비스를 처리하는 **응용프로그램 제어**
- 다양한 사용자에게서 컴퓨터시스템 보호를 위한 **입출력 제어 및 데이터 관리**



운영체제의 개요

2 운영체제 발전의 목적



편리성

- 사용자에게 편리한 환경 제공
 - ① 프로그램 개발 환경뿐만 아니라 응용프로그램에 대한 사용자 인터페이스, 즉 사용자와 컴퓨터시스템이 정보 및 명령을 상호 교환할 수 있는 인터페이스 제공(PC의 GUI 환경)



효율성

- 시스템 성능 향상
 - ① 사용자가 많은 대형 컴퓨터시스템에서 특히 중요
 - ① 각 프로그램을 유기적으로 결합하여 시스템 전체 성능 향상
 - ① 시스템 성능의 평가 기준

처리능력

응답시간
(턴어라운드
타임)

신뢰도

사용 가능도
(가동률)



제어 서비스 향상

- 시스템 확장, 효율적 운영을 위해 새로운 기능의 효과적인 개발을 허용하는 방법으로 발전
 - ① 입출력 장치의 동작 관리 및 제어, 시스템 오류 예방 등으로 컴퓨터 자원을 여러 사용자에게 효율적으로 할당하고 관리할 수 있도록 함



운영체제의 개요

3 운영체제의 발전과정



1960년대: 다중 프로그래밍·시분할·다중 처리·실시간 시스템

- **장치 독립성**을 이용한 편리한 하드웨어 관리와 다중 프로그래밍
 - ① 여러 프로그램을 메모리에 나눠 적재한 후 프로세서를 작업 간에 번갈아 할당함
 - ② 프로세서 사용을 극대화하여 여러 **사용자 프로그램을 동시에 실행**



장치독립성

- 프로그램을 다른 입출력 장치와 함께 실행할 수 있도록 하는 것
- 시분할, 다중 처리, 실시간을 이용한 시스템의 처리 능력 향상

| 시분할 시스템 | 다중 처리 시스템 | 실시간 처리 시스템 |
|--|--|------------|
| 다중 프로그래밍 시스템에 프로세서 스케줄링이라는 개념을 더한 것 | 하나의 시스템에서 프로세서를 여러 개 사용하여 처리 능력을 높인 것 | 즉시 응답 |

- 미항공사의 SABRE 예약 시스템을 개발
 - ① 멀리 떨어진 사용자가 단말기를 이용, 중앙 컴퓨터 시스템과 통신하는 **트랜잭션 처리 시스템의 효시**

| 트랜잭션 | 트랜잭션 처리 시스템 |
|---------------------------------|---|
| 정보의 교환이나 관련 작업에 대한 일련의 처리 단위 | 사용자와 컴퓨터시스템이 서로 대화를 하되, 사용자의 간단한 요구에 빠르게 응답하는 것 |



운영체제의 개요

3 운영체제의 발전과정



1970년대 중반~1990년대: 분산 처리 시스템, 병렬 계산과 분산 처리

- 컴퓨터 네트워크와 온라인 처리 방법 널리 사용
 - ① 네트워크 이용하여 멀리 떨어진 컴퓨터를 사용할 수 있게 됨
- 마이크로프로세서 등장
 - ① 개인용 컴퓨터 보유
- 사용자가 지역적으로 원격의 여러 시스템과 통신 가능
 - ① 정보 보호가 보다 중요하게 됨
- 1970년대: 명령어 중심의 시스템 사용
- 1980년대: 사용자에게 편리한 메뉴 지향적인 시스템
- 1990년대: GUI(Graphical User Interface) 시스템
- 데이터베이스 시스템의 중요성이 커짐



운영체제의 개요

3 운영체제의 발전과정



2000년대 이후: 모바일 및 임베디드, 가상화 및 클라우드 컴퓨팅

- ☁ 스마트폰이나 태블릿 같은 **모바일 기기**가 대중화됨
- ☁ **모바일 운영체제(Mobile Operating System):**
모바일 장치나 정보 기기 제어 운영체제

스마트폰용

노키아(심비안), 구글(안드로이드 Android), 애플(iOS) 등이 대표적임

- ☁ **사물 인터넷 IoT(Internet of Things) 기술 등장:**
각종 사물에 컴퓨터칩과 통신 기능이 내장되어 인터넷으로 연결됨



운영체제의 개요

4 운영체제의 유형

일괄 처리
시스템

다중
프로그래
밍 시스템

다중 처리
시스템

시분할
시스템

실시간
처리
시스템

분산 처리
시스템



다중 프로그래밍 시스템

- 프로세스가 다른 작업 수행 시 입출력 작업 불가능하여 프로세서와 메인 메모리의 활용도 떨어지는 일괄 처리 시스템의 큰 문제를 다중 프로그래밍 도입하여 해결



다중 프로그래밍 시스템 | 특징

- 프로세서가 항상 수행할 작업을 가지도록 하여 프로세서 사용률을 극대화함
- 많은 사용자의 프로그램이 거의 동시에 프로세서를 할당 받는 듯한 느낌을 갖게 됨
- 운영체제가 아주 복잡함
- 여러 작업을 준비 상태로 두려면 이를 메모리에 보관, 일정 형태의 메모리 관리가 필요함
- 여러 작업이 수행할 준비를 갖추고 있으면, 이 중 하나를 선택하는 결정 방법이 필요함

| | | | | | | | |
|--------|---|------|------|------|------|------|---|
| 프로그램 A | ① | 유힬시간 | ③ | 유힬시간 | ⑤ | 유힬시간 | ⑦ |
| 프로그램 B | | ② | 유힬시간 | ④ | 유힬시간 | ⑥ | |

다중 프로그래밍 시스템의 처리방법 예



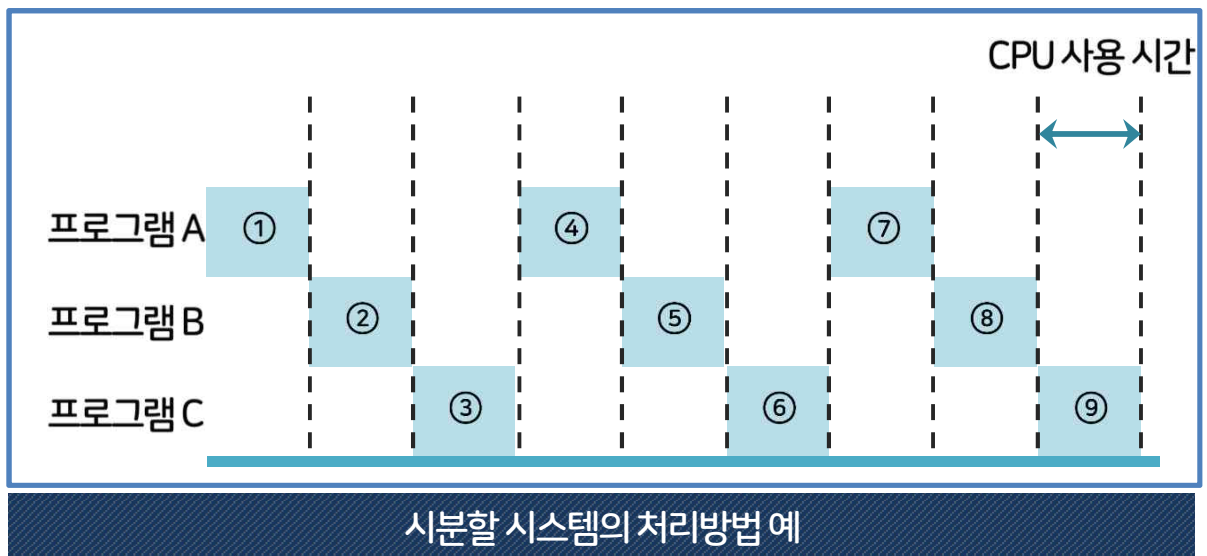
운영체제의 개요

4 운영체제의 유형



시분할 시스템(TSS, Time Sharing System)

- 다중 프로그래밍을 논리적으로 확장한 개념으로 프로세서가 다중 작업을 교대로 수행함
- 다수의 사용자가 동시에 컴퓨터의 자원을 공유할 수 있는 기술
- CTSS(Compatible Time Sharing System): MIT에서 개발, 1961년 IBM 709에 탑재하여 사용함





운영체제의 개요

4 운영체제의 유형



시분할 시스템(TSS, Time Sharing System) | 장/단점



장점

- ① 빠른 응답 제공
- ① 소프트웨어의 중복 회피 가능
- ① 프로세서 유휴시간 감소



단점

- ① 신뢰성 문제
- ① 다수의 사용자가 접근하는 파일의 권한 부여 필요성
- ① 사용자 프로그램과의 데이터의 무결성(한 작업이 다른 작업의 데이터를 변경 가능)



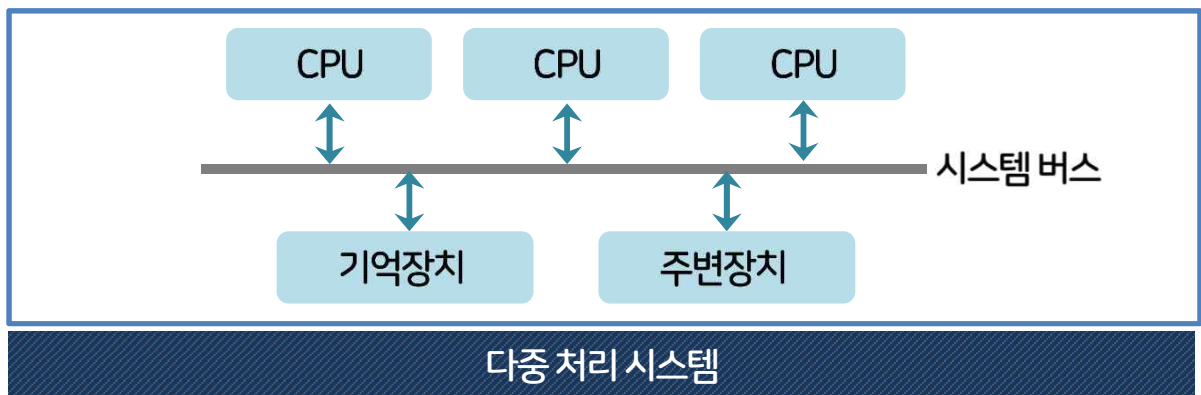
운영체제의 개요

4 운영체제의 유형



다중 처리(Multiprocessing) 시스템

- 단일 컴퓨터시스템 내에서 둘 이상의 프로세서 사용, **신뢰성, 가용성을 증대시킬 수 있음**
- 여러 프로세서와 시스템 버스, 클록, 메모리와 주변 장치 등을 공유함



실시간 처리 시스템(Real time processing system)

- 데이터 처리 시스템으로 정의**
- 온라인 시스템은 실시간으로 할 필요 없지만, 실시간 처리 시스템은 **항상 온라인 상태이어야 함**
- 보다 높은 적시 응답을 요구하는 장소에서 사용함**
- 데이터 흐름 또는 프로세서 연산에 **엄격한 시간 요구가 있을 때 사용함**
- 고정 시간 제약을 잘 정의하지 않으면 시스템이 실패할 가능성이 큼**



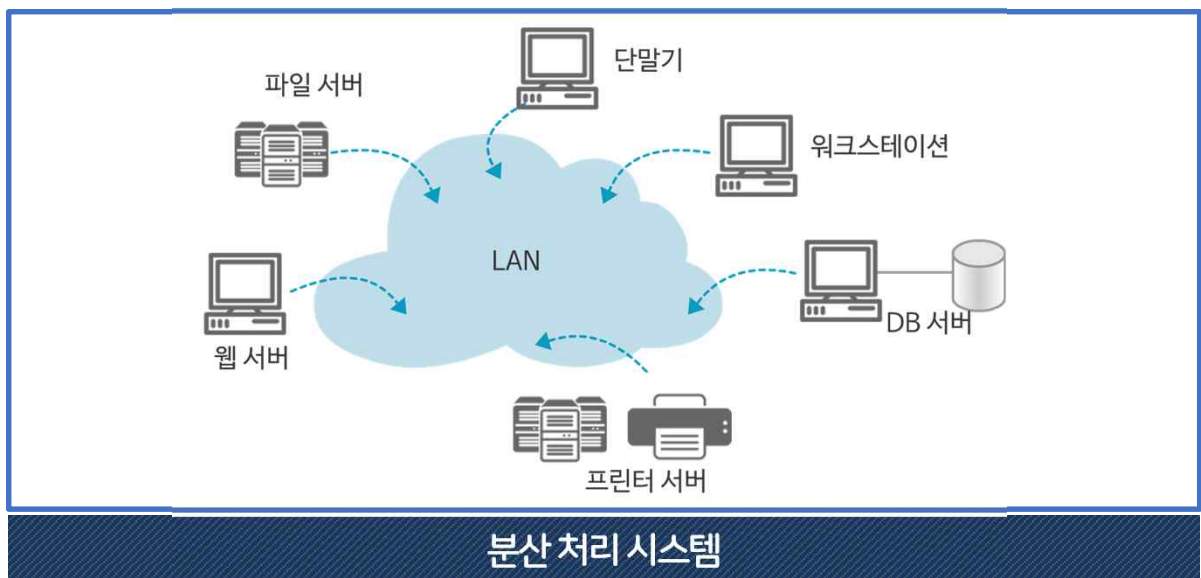
운영체제의 개요

4 운영체제의 유형



분산 처리 시스템(Distributed processing system)

- 시스템마다 독립적인 운영체제와 메모리로 운영, 필요 시에 통신하는 시스템
- 다수의 독립된 프로세서에서 실행하며 최근 컴퓨터 시스템의 대표적인 형태임
- 데이터를 여러 위치에서 처리·저장하며, 이를 여러 사용자가 공유함
- 하나의 프로그램을 여러 프로세서에서 동시에 실행





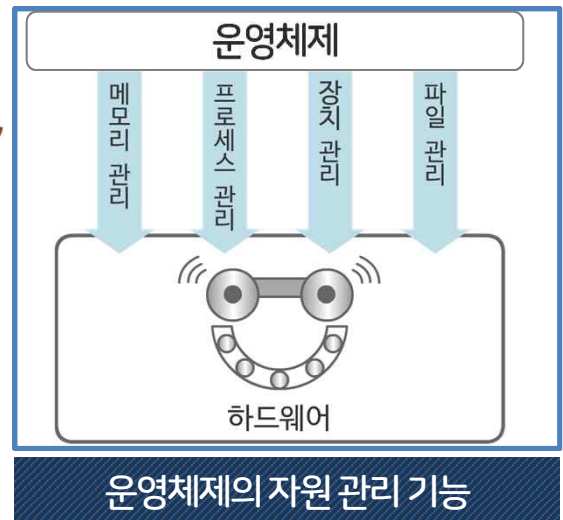
운영체제의 기능

1 자원관리



자원

- 컴퓨터시스템의 메모리, 프로세스, 장치, 파일 등 구성 요소





운영체제의 기능

1 자원관리



메인 메모리 관리

- 메인 메모리: 프로세서가 직접 주소로 지정할 수 있는 유일한 메모리



메모리 관리의 기능

- 메모리의 어느 부분을 사용하고, 누가 사용하는지 점검함
- 메모리에 저장할 프로세스를 결정함
- 메모리를 할당하고 회수하는 방법을 결정함



보조 기억 장치 관리

- 메인 메모리는 공간이 제한되어 데이터와 프로그램을 계속 저장할 수 없어 보조 기억 장치를 이용하게 됨



보조 기억 장치 관리의 기능

- 빈 여유 공간을 관리함
- 새로운 파일 작성 시 저장 장소를 할당함
- 메모리 접근 요청 스케줄링
- 파일 생성 및 삭제



운영체제의 기능

1 자원관리



프로세스 관리

프로세스

- ① 하나의 프로세스는 프로세서, 메모리, 파일, 입출력 장치와 같은 자원으로 구성
- ② 자원은 프로세스를 생성할 때 제공하거나 프로세스 실행 중에도 할당 가능함

시스템

- ① 프로세스의 집합
- ② 시스템 코드를 수행하는 운영체제 프로세스와 사용자 코드를 수행하는 사용자 프로세스로 구분됨
- ③ 모든 프로세스는 프로세서를 분할하여 사용함으로써 병행 수행이 가능함



프로세스 관리를 위한 운영체제의 기능

- 프로세스와 스레드 스케줄링
- 사용자 프로세스와 시스템 프로세스 생성 및 제거
- 프로세스 중지, 재수행
- 프로세스 동기화 방법 제공
- 프로세스 통신 방법 제공
- 교착 상태(Deadlock) 방지를 위한 방법 제공



운영체제의 기능

1 자원관리



주변 장치(입출력 장치) 관리

- 운영체제

- ① 특수 프로그램인 장치 드라이버를 사용하여 입출력 장치와 상호작용을 함

- 장치 드라이버

- ① 특정 하드웨어 장치와 통신할 수 있는 인터페이스를 제공하는, 특정 하드웨어에 종속된 프로그램



파일(데이터) 관리

- 입출력 파일의 위치, 저장, 검색 관리

- 컴퓨터시스템

- ① 물리적으로 다양한 형태(자기 테이프나 자기 디스크) 등으로 파일 저장 가능

- 운영체제

- ① 데이터의 효율적 사용을 위해 단일화된 정보 저장 형태를 제공함
- ① 파일의 용이한 사용을 위해 보통 디렉터리로 구성함
- ① 다수의 사용자가 여기에 접근하려고 할 때는 누가 어떤 방법으로 접근할 지를 제어함



파일 관리를 위한 운영체제의 기능

- 파일 생성과 삭제

- 디렉터리 생성과 삭제

- 보조 기억 장치에 있는 파일의 매핑

- 안전한(비휘발성) 저장 매체에 파일을 저장함



운영체제의 기능

2 시스템 관리



시스템 보호(사용자 권한 부여)

- 컴퓨터 시스템에 의해 정의된 자원에 대해 프로그램, 프로세스, 사용자 접근을 제어하는 방법
- 운영체제
 - ① 파일 사용 권한을 부여하거나, 데이터 암호화 등 서비스를 제공함
- 컴퓨터시스템
 - ① 여러 프로세스를 동시에 실행할 수 있으므로 상호 보호해야 함
- 네트워크로 파일 공유 사이트에 접속 시
 - ① 다른 사용자의 프로그램으로부터 보호되어야 함



네트워킹(통신)

- 프로세서
 - ① 다양한 방법으로 구성된 네트워크 이용, 완전 접속과 부분 접속 방법으로 다른 프로세서들과 연결
- 연결된 프로세서끼리 통신을 할 경우
 - ① 경로 설정, 접속 정책, 충돌, 보안 등의 문제를 고려해야 함(운영체제가 관리)



운영체제의 기능

2 시스템 관리



명령 해석기(Command interpreter)

- 운영체제를 위한 중요한 시스템 프로그램 중의 하나임
- 대화형으로 입력한 명령어를 이해하고 실행하는 사용자와 운영체제의 인터페이스
- 사용자가 입력한 명령은 제어문으로 운영체제에 전달하는데, 이 전달을 명령 해석기가 담당함



운영체제의 서비스

1 부팅 서비스



부팅(Booting) 또는 부트스트래핑(Bootstrapping)

- 운영체제를 메인 메모리에 적재하는 과정
- 부트 로더는 부트스트랩 로더(Bootstrap loader)줄인 말로 ROM에 저장된 소규모 프로그램
- 하드디스크와 같은 보조 기억 장치에 저장된 운영체제를 메인 메모리에 적재시킴



운영체제의 서비스

2 사용자 서비스



사용자 인터페이스 제공



사용자 인터페이스

- 사용자와 컴퓨터 간의 상호작용 발생 공간
 - ① 명령 라인 인터페이스(CLI: Command Line Interface)
 - ① 메뉴 인터페이스
 - ① 그래픽 사용자 인터페이스(GUI: Graphical User Interface)



프로그램 실행

- 프로그램을 메모리에 적재하여 실행함
- 정상적, 비정상적으로 프로그램 실행을 종료함



입출력 동작 수행

- 수행 중인 프로그램은 입출력을 요구할 수 있고, 이를 위해 파일이나 입출력장치가 필요함
- 운영체제
 - ① 입출력 동작을 직접 수행할 수 없는 사용자 프로그램의 입출력 동작 방법을 제공함



운영체제의 서비스

2 사용자 서비스



파일 시스템 조작

사용자

- ① 디스크에서 파일 열고, 저장, 삭제하는 등 다양하게 파일 조작
- ① 디스크에 파일을 저장하면 특정 블록에 할당하여 저장함
- ① 파일을 삭제하면 파일 이름 제거되면서 할당한 블록이 자유롭게 됨

운영체제

- ① 파일 시스템 조작 서비스를 제공함
- ① 사용자가 파일 관련 작업을 쉽게 할 수 있게 함



통신(네트워크)

프로세스가 다른 프로세스와 정보를 교환하는 방법

- ① 동일한 컴퓨터에서 수행하는 프로세스 간의 정보 교환
- ① 네트워크로 연결된 컴퓨터시스템에서 수행하는 프로세스 간의 정보 교환

운영체제

- ① 다중 작업 환경에서 공유 메모리를 이용하거나 메시지 전달을 통해 다양한 유형의 프로세스와 통신이 가능하도록 지원함



운영체제의 서비스

2 사용자 서비스



오류 탐지

- ☁ 운영체제는 가능한 모든 오류를 탐지하고 시스템을 모니터링하여 조정함으로써 하드웨어 문제를 예방함
 - ① 입출력 장치에 관련된 오류
 - ① 메모리 오버플로우
 - ① 하드디스크의 불량 섹터 문제
 - ① 부적당한 메모리 접근과 데이터 손상 등



운영체제의 서비스

3 시스템 서비스



의미

- 사용자가 아닌 **시스템 자체의 효율적인 동작을 보장함**
- 여러 사용자가 사용하는 시스템에서 **컴퓨터 자원을 공유**하여 시스템 자체의 효율성을 높임



자원 할당

- 운영체제는 다수의 사용자나 다수의 작업에 **동시에 실행될 때 자원을 각각 할당**하도록 관리
- 프로세서 사이클, 메인 메모리, 파일 저장 장치 등은 특수한 할당 코드를 갖지만, **입출력 장치** 등은 더 일반적인 요청과 해제 코드 가질 수 있음



계정

- 운영체제는 각 사용자가 어떤 컴퓨터 자원을 얼마나 많이 사용하는지 **파악하기 위해 계정정보를 이용함**
- 사용 통계 정보는 사용자 서비스 개선을 위해 시스템 재구성하는 연구자에게 귀중한 도구가 됨



운영체제의 서비스

3 시스템 서비스



보호와 보안

- 운영체제는 다중 사용자 컴퓨터시스템에 저장된 **정보 소유자의 사용을 제한할 수 있음**
- 서로 관련이 없는 여러 작업을 동시에 수행할 때는 **한 작업이 다른 작업이나 운영체제를 방해하지 못하게 해야 함**

보호

시스템 호출 하려고 전달한 모든 매개변수의 타당성 검사하고, 시스템 자원에 모든 사용자 접근을 제어하도록 보장하는 것

보안

잘못된 접근 시도에서 외부 입출력 장치 방어, 외부에 사용자 인증을 요구하는 것



운영체제의 서비스

4 시스템 호출(System call)



정의와 역할

- 실행 중인 프로그램과 운영체제 간의 인터페이스로, API(Application Programming Interfaces)라고도 함
- 사용자 프로그램은 시스템 호출을 통해 운영체제의 기능을 제공 받음

핵심 커널 서비스와
통신

새로운 프로세스의
생성과 실행

하드웨어 관련
서비스 등



시스템 호출 방법

- 프로그램에서 명령이나 서브루틴의 호출 형태로 호출
- 시스템에서 명령 해석기를 사용하여 대화 형태로 호출



운영체제가 제공하는 일반적인 시스템 호출

| 호출 서비스 | 설명 |
|---------|---|
| 프로세스 제어 | <ul style="list-style-type: none"> • 종료와 취소 • 적재(Load)와 실행 • 프로세스 생성과 종료 • 프로세스 속성 획득과 지정 • 대기와 대기 이벤트, 신호 이벤트 • 메모리 할당과 해제 |
| 파일 조작 | <ul style="list-style-type: none"> • 파일 생성과 삭제 • 파일 열기(Open)와 닫기(Close) • 파일 읽기와 쓰기, 파일 재배치(Reposition) • 파일 속성 획득과 지정 |
| 장치 조작 | <ul style="list-style-type: none"> • 장치 요구와 해제 • 장치 읽기와 쓰기, 재배치 • 장치 속성 획득과 설정 • 논리적 부착이나 장치 제거 |
| 정보 관리 | <ul style="list-style-type: none"> • 시간과 날짜의 설정과 획득 • 프로세스, 파일 장치속성의 설정과 획득 • 데이터의 설정과 획득 |
| 통신 | <ul style="list-style-type: none"> • 통신 연결의 생성과 제거 • 메시지의 송수신 • 정보 상태 전달 • 원격 장치의 부착 및 제거 |



운영체제의 구조

1 단일구조 운영체제



등장 배경

점점 더 다양한 하드웨어와
소프트웨어를 지원하면서
복잡해진 구조

복잡한 시스템은 설계, 구현,
테스트, 유지 보수 등
모든 면에서 어려움

어려움 해결을 위해 운영체제를 설계하는 다양한 방법 등장



단일(Monolithic) 구조(또는 모놀리식 커널 구조) 운영체제

- 초기에 생겨난 가장 보편적 형태
- 운영체제의 모든 기능을 커널과 동일한 메모리 공간에 적재 후 시스템 호출만으로 사용
- 작고 간단하면서 시스템 기능이 제한된 구조



운영체제의 구조

1 단일구조 운영체제



특징

- 대부분의 기능을 **커널에 그룹화**해서 구현, 직접 통신하여 시스템 자원을 효율적 관리
- 커널 크기가 상대적으로 커지면서 **버그의 원인이나 기타 오류 구분 어려움**
- 새 기능을 추가하는 **수정과 유지 보수 매우 어려움**
- 동일한 메모리에서 실행하므로 **한 부분에서 발생한 문제로 시스템 전체에 심각한 영향 가능**
- 악성 코드로 피해 입기 쉬움



운영체제의 구조

2 계층구조 운영체제



등장배경

- 운영체제가 점점 커지고 복잡해지면서 순수 단일 구조만으로는 다루기가 어려워지는 문제점을 해결하기 위해 등장함



특징

- 계층 구조에서는 비슷한 기능을 수행하는 요소를 그룹화하여 계층적으로 구성
- 사용자 프로세스의 요청을 수행할 때 한 계층에서 다음 계층으로 데이터를 전달할 때마다 추가적인 시스템 호출 발생
- 호출 한 번으로 서비스를 받는 단일 구조보다는 성능이 우수함
- 단일 구조 운영체제보다 모듈화가 잘 되어 있음
- 시스템 검증과 오류 수정이 단일계층 구조에 비해 용이함
- 첫 번째 계층은 기본 하드웨어 사용하여 기능을 만들어 나머지 시스템에 의문을 가지지 않고 오류를 수정할 수 있음
- 첫 번째 계층의 오류를 수정하면 기능이 정확하다고 가정하여 두 번째 계층 만듦(이 과정 반복)
- 특정 계층에서 오류를 발견해도 하위 계층은 오류를 수정했기 때문에 오류가 없다고 판단됨
- 시스템을 계층으로 나누면 시스템 설계나 구현이 단순해짐
- THE 운영체제에서 처음 사용함(1968년 다익스트라[Dijkstra] 개발)
- 각 계층은 자신의 하위 계층만 사용할 수 있으므로 신중히 설계해야 함



운영체제의 구조

3 마이크로커널 구조 운영체제



등장배경

- 단일 커널의 문제점 해결 위해 1980년대 카네기멜론 대학교에서는 모듈화된 마이크로커널(Microkernel)을 사용하여 매크(Mach)를 만들






특징

- 커널 기능을 대폭 축소시킴
- 커널에는 최소 기능만 포함시켜 크기를 대폭 줄이고 기타 기능은 사용자 공간으로 옮겨 사용자 영역에서 수행되는 서버로 구현함
- 하드웨어 초기화, 메모리 관리(주소 공간 관리), 프로세스(스레드) 관리, 프로세스 간 통신, 프로세스 간 협력 동기화 기능 등 기본 기능만 커널에서 실행함
- 네트워크 시스템, 파일 시스템 상호작용과 장치 관리 등 대부분의 운영체제 구성 요소는 커널 외부(사용자 영역의 서버)로 옮겨 구현함
- 프로세스 간 통신 발생을 최소화시키는 것이 중요 과제임
- 응용프로그램과 서버 간에 자료를 교환을 위해 커널을 출입하는 문맥 교환 때문에 속도가 느려짐
- 커널 내부에서 발생 지연이 적고 예측 가능하여 실시간 시스템에 활용됨
- 대표적인 운영체제 중 마이크로커널 구조를 전적으로 선택한 것은 없지만, 모듈화된 구성 요소 개념은 많이 활용함





핵심요약

1 운영체제의 개요

-  사용자와 하드웨어 사이의 중간 매개체로 응용프로그램의 실행을 제어하고, 자원을 할당 및 관리하며, 입출력 제어 및 데이터 관리와 같은 서비스를 제공하는 소프트웨어
-  하드웨어 및 사용자, 응용프로그램, 시스템 프로그램 사이에서 인터페이스 제공
-  프로세서, 메모리, 입출력 장치, 통신 장치 등 컴퓨터 자원을 효과적으로 활용하도록 조정·관리함

2 운영체제의 기능

-  자원관리: 컴퓨터시스템의 메모리, 프로세스, 장치, 파일 등 구성 요소를 관리하는 것
-  시스템 관리: 시스템 보호를 위해 행하는 계정관리 등



📌 핵심요약

3 운영체제의 서비스

📚 부팅서비스, 사용자 서비스, 시스템 서비스, 시스템 호출 등이 있음

4 운영체제의 구조

📚 운영체제는 단일구조 운영체제, 계층구조 운영체제, 마이크로커널 구조 운영체제로 구성됨