

02. OS Overview

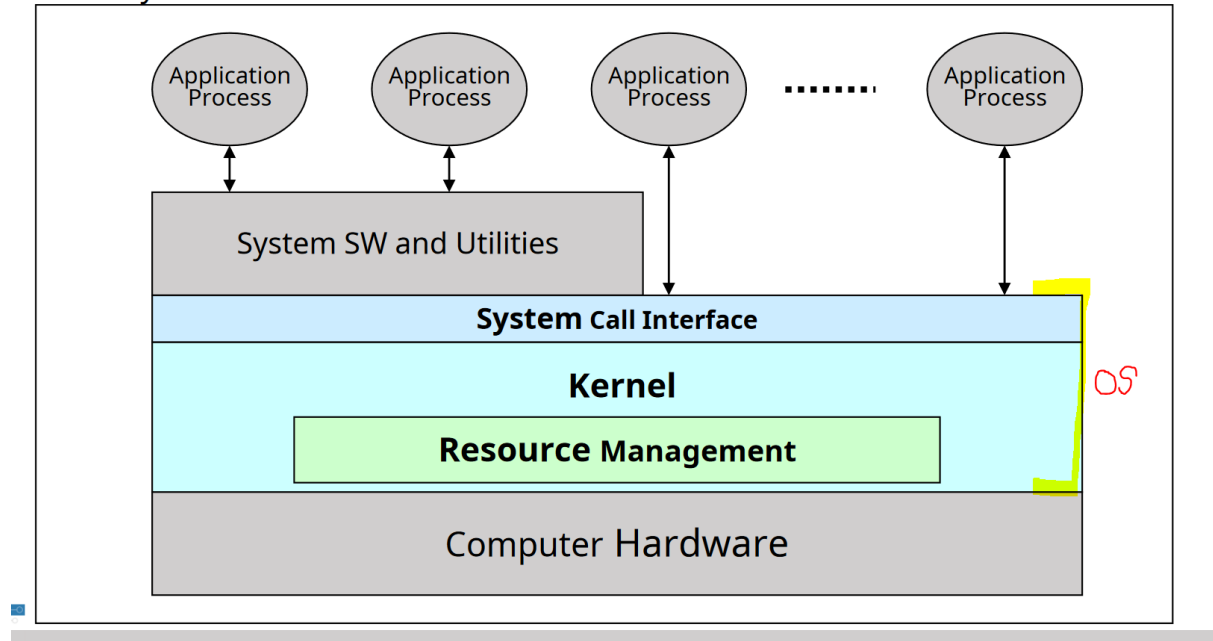
- 운영체제란?
 - 컴퓨터 HW를 효율적으로 관리해서 사용자/응용프로그램에게 서비스를 제공하는 것

운영체제 역할

1. user interface (편리성)
 - CUI: character user interface
 - GUI: graphical user interface - 지금 쓰는 거
 - EUCI: 특정 목적을 위해서 있는 운영체제, 해당 사용자를 위해서 특화됨 ex) MP3 화면 등
2. resource management (효율성)
 - HW 자원, SW 자원 관리
3. process & thread management
 - 프로세스: 프로그램 실행의 주체
 - Thread: 조금 가벼운 프로세스
4. system management (시스템 보호)
 - 사용자가 불법적으로 시스템을 사용하지 못하도록

컴퓨터 시스템 구성

Multi-layer architecture



HW 위에 OS, OS 위에 사용자가 사용하는 응용 프로그램 등

- 두 개 층이 OS가 됨: kernel & system call interface
 - kernel: 운영체제 핵심을 모아둔 것
 - system call interface: 사용자가 직접 커널에 접근하지 않고, 필요한 기능이 있으면 요청하는 통로
 - 보통 함수 형태 system call로 제공
 - 커널이 제공하는 기능 중 사용자가 사용할 수 있는 기능을 모아둔 것

운영체제 구분

- 동시 사용자 수
 - single user
 - 한명의 사용자가 모든 시스템 자원 독점
 - multi user: ex) linux, unix 와 같이 서버-클라이언트(1:N) 형태
 - 자원 소유권 관리 필요
 - 멀티태스킹 기능 필요, 복잡
- 동시 실행 프로세스 수
 - single tasking

- 시스템 내 하나의 작업만 존재, 프로그램 실행 마친 후 다음 프로그램 실행 가능
ex) MS-DOS
- mutli tasking
 - 동시에 여러 작업(프로세스) 수행 가능
 - 작업들 사이 동시 수행, 동기화 등 관리 필요
- 작업 수행 방식 (사용자가 느끼는 사용환경)
 - batch processing system: 일괄처리
 - time sharing system: 시분할 처리
 - distributed processing system
 - real time system

작업 수행 방식에 따른 운영체제 구분

- 순차 처리: 운영체제 개념 존재 X
 - 사용자가 직접 기계어로 프로그램 작성
 - 실행 작업별 준비 시간 소요 및 각 작업 순차 처리

Batch Systems: 일괄처리 시스템

- 중앙 관리 시스템: 모든 시스템을 중앙에서 관리
 - 시스템 지향적(System-oriented)
 - 같은 유형의 작업을 모아서 한번에 처리 ⇒ 작업별 준비 시간 감소
- 장점: 많은 사용자가 시스템 자원 공유 가능, 처리 효율(throughput) 향상
- 단점: 생산성 저하, 긴 응답 시간(turnaround time)

Time Sharing Systems: 시분할 처리 시스템

- 여러 사용자가 동시에 자원 사용: 프로그램별 CPU 사용 시간을 나눠가짐
 - 사용자 지향적
 - 동시에 사용되는 자원을 관리하기 위해서 OS에 파일 시스템 및 가상 메모리 개념 등장, 관리
- 장점: 응답시간(response time) 단축, 생산성 향상(프로세서 유휴 시간 감소)
- 단점: 통신 비용 증가, 개인 사용자 체감 속도 저하

Personal Computing

- 개인이 시스템 전체 독점 ex) PC
 - CPU 활용률이 고려의 대상 X, 사용자 편리성이 더 우선 순위
- 장점: 빠른 응답 시간
- 단점: 성능 낮음

Parallel Processing Systems: 병렬 처리 시스템

- 단일 시스템 내 둘 이상의 프로세서 사용
 - 메모리 등 자원 공유 (Tightly coupled system): 여러 CPU가 기억 장치/주변 장치 등 공유
- 사용 목적
 - 성능 향상: 기존 PC의 낮은 성능을 개선하기 위해
 - 신뢰성 향상: 하나 고장나도 다른 프로세서 있음
- 한계: 넣을 수 있는 CPU 개수에는 한계가 있음

Distributed Processing Systems: 분산 병렬 처리 시스템

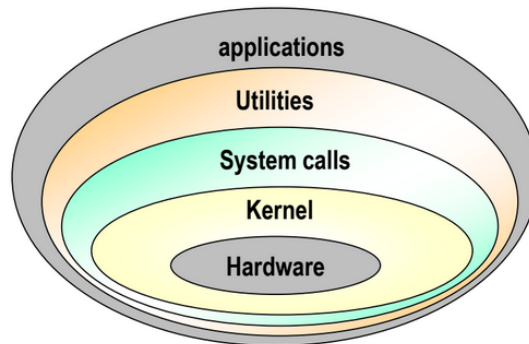
- 네트워크 기반으로 구축된 병렬 처리 시스템(Loosely coupled system)
 - NW를 통해 여러 컴퓨터를 붙여서 만든 병렬 처리 시스템
- 사용자는 분산 운영체제를 통해 하나의 프로그램/자원처럼 사용 가능(은폐성, transparency)
 - 물리적으로 분산되어 있고 다른 운영체제를 쓸 수도 있지만 사용자 입장에서는 하나처럼
 - ex) cluster system, client-server system, p2p 등
- 장점: 자원 공유를 통한 높은 성능, 고신뢰성, 높은 확장성
- 단점: 구축 및 유지보수 어려움

Real-time Systems: 실시간 시스템

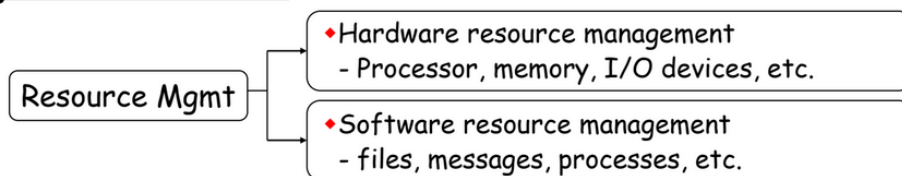
- 작업 처리에 제한 시간을 갖는 시스템: 제한 시간 내 서비스 제공 > 자원 활용 효율
- 종류
 - hard real-time task: time critical tasks

- soft real-time task: 동영상 재생 등
- non real-time task

운영체제 구조

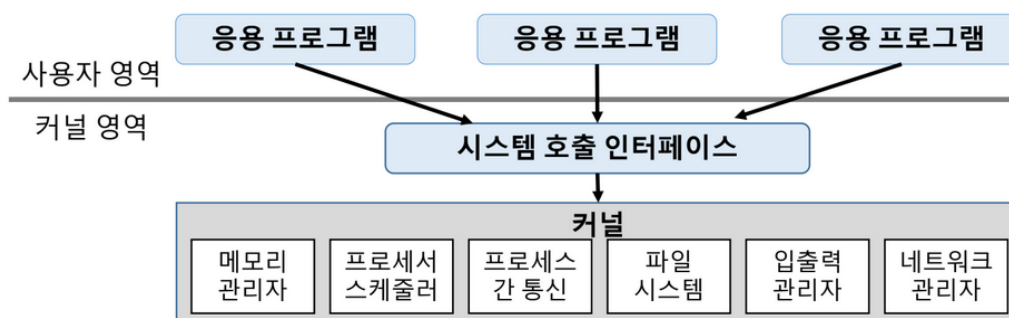


Major functions of Kernel



1. 커널: OS 핵심 부분 (항상 메모리에 상주)
 - 시스템 (프로세서, 메모리 등) 관리 등 제일 자주 쓰이는 기능 담당
2. 유틸리티: 그 외 (필요할 때만 메모리에 올림)

단일 구조



- 운영체제 기능을 하나의 큰 커널로 묶어둔 것
- 장점: 커널 내 모듈 간 직접 통신 가능 ⇒ 효율적 자원 관리 및 사용, 속도 빠름
- 단점: 커널의 거대화 ⇒ 유지보수 어려움, 문제 생기면 전체가 영향 받음

계층 구조

- 기능별 계층으로 분리
- 장점: 모듈화 ⇒ 계층 간 검증 및 수정 용이, 설계 및 구현 단순화
- 단점: 단일 구조 대비 성능 저하

마이크로 커널 구조

- 필수 기능(메모리/프로세스 관리, 프로세스 간 통신 등) 외 다른 기능은 사용자 영역에서 처리

운영체제 기능

- 프로세스 관리
- 프로세서 관리
- 메모리 관리
- 파일 관리
- 입출력 관리
- 보조 기억 장치 및 기타 주변 장치 관리

Process Management

- 프로세스
 - 커널에 등록된 실행 단위(실행 중인 프로그램)
- OS의 프로세스 관리 기능
 - 생성/삭제/상태 관리, 자원 할당
 - 프로세스 간 통신 및 동기화, deadlock 해결
 - 프로세스 정보 관리 위해 커널 내 PCB 사용

Processor Management

- 프로세서: HW (특히 CPU), 프로그램 실행하는 핵심 자원
- OS의 프로세서 관리
 - 프로세스에 대한 프로세서 할당 관리
 - 프로세스 스케줄링

Memory Management

- 주기억장치(DRAM 등): 작업 위한 프로그램/데이터 올려놓는 공간
- OS의 메모리 관리
 - multi-user, multi-tasking 시스템
 - 프로세스에 대한 메모리 할당 및 회수, 할당된 메모리 영역 접근 보호
 - 메모리 여유 공간 관리
 - 메모리 할당 방법(scheme) 종류
 - 전체 적재
 - 일부 적재: cf) virtual memory

File Management

- 파일: 논리적 데이터 저장 단위
- OS에서 파일 시스템 지원

I/O Management

- 주변 장치와의 입출력 관리
- 사용자 프로그램에서 커널에 요청, 커널을 통해서 주변 입출력 장치와 통신