

Operating Systems

# Index	0
📅 CreatedAt	
👤 Person	A Ally Hyeseong Kim
☰ Status	
☰ Tags	Computer Science Operating Systems
📅 UpdatedAt	

References

면접 시리즈5 - 운영체제

운영체제는 하드웨어를 관리하고, 응용 프로그램과 하드웨어 사이에서 인터페이스 역할을 하는 시스템 소프트웨어 cf) 여기서 인터페이스는 서로 다른 두 개의 시스템, 장치 사이에서 정보나

B <https://backtony.github.io/interview/2021-12-13-interview-20/>



<https://www.aladin.co.kr/shop/wproduct.aspx?ItemId=86832147>

References

1. 운영체제
2. 커널
4. 프로세스와 스레드
5. 컨텍스트 스위칭
6. IPC (Inter Process Communication)
7. 프로세스 종류
8. Deadlock
9. 전통적인 동기화 문제
10. Critical Section(임계영역)
11. 경쟁 상태(Race Condition)
12. 수준 스레드
13. 사용자 모드와 커널 모드
14. CPU Scheduling
15. 동시성과 병렬성

- 16. 인터럽트
 - 17. 시스템 콜
 - 18. 메모리구조 순서
 - 19. 메모리 관리
 - 20. 가상 메모리
 - 21. 캐시의 지역성
 - 22. 캐시 라인
 - 23. Memory Corruption
 - 24. Direct Memory Access (DMA)
-

1. 운영체제



운영체제는 응용 프로그램을 실행할 수 있는 기반 환경을 제공하고, 하드웨어를 효율적으로 사용할 수 있도록 다양한 기능을 제공하는 소프트웨어이다.



컴퓨터 시스템의 구성요소

- **사용자** : 작업을 수행하려고 컴퓨터를 사용하려는 사람이나 장치, 다른 컴퓨터
- **소프트웨어** : 컴퓨터가 기능을 수행하는 데 필요한 모든 프로그램
 - 시스템 소프트웨어: 컴퓨터 자원을 관리하고 응용 프로그램의 실행을 지원하여 컴퓨터를 제어하는 프로그램(운영체제, 장치 드라이버 등)
 - 유틸리티: 응용 프로그램보다 작지만 컴퓨터의 여러 처리 과정을 보조하여 시스템을 유지하고 성능을 개선하는 프로그램(운영체제를 돕는 역할)(화면 보호기, 바이러스 검사 소프트웨어 등)
 - 응용 프로그램: 특정 작업을 수행하려는 목적(웹 브라우저, 워드 프로세서, 워드 프로세서, 게임, 이미지 편집 프로그램, 은행 시스템 등)
- **하드웨어** : 기본 연산 자원을 제공하는 프로세서(CPU), 메모리, 주변 장치 등



운영체제의 역할

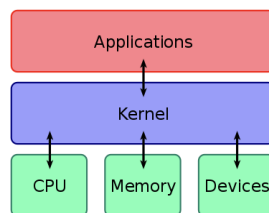
- **조정자** : 시스템 운영 요소를 적절하게 사용할 수 있도록 제어한다.
- **자원 할당자, 관리자** : 각 응용 프로그램에 필요한 자원을 할당하거나 효율적으로 운영하기 위해 자원을 할당하는 방법을 결정한다.
- **응용 프로그램, 입출력장치 제어자** : 응용 프로그램과 입출력장치를 제어한다.



운영체제의 기능

- 자원 관리
 - 메모리 관리
 - 프로세스 관리
 - 주변장치(입출력장치) 관리
 - 파일(데이터) 관리
- 시스템 관리
 - 시스템 보호(사용자 권한 부여)
 - 네트워킹(통신)
 - 명령 해석기

2. 커널



커널은 메모리에 상주하는 **운영체제**의 핵심 부분이다.

- 소프트웨어가 컴퓨터 시스템에서 수행되기 위해 메모리에 올라가 있어야 한다. 하지만 **운영체제** 전체가 항상 메모리에 올라가 있으면 공간 낭비가 심하므로 항상 필요한 부분만 올린다.



운영체제의 구조

- 단일 구조 운영체제(초기 유닉스)
 - 운영체제의 모든 기능을 커널과 동일한 메모리 공간에 적재한 후 시스템 콜로만 사용할 수 있다.
 - 장점: 대부분의 기능을 커널에 그룹화해서 구현하므로 직접 통신하여 시스템 자원을 효율적으로 관리할 수 있다.
 - 단점: 커널 크기가 상대적으로 커서 버그 원인이나 기타 오류를 구분하기 어렵고, 새 기능을 추가하는 수정과 유지 보수가 매우 어렵다. 그리고 동일한 메모리에서 실행하여 한 부분의 문제가 전체 시스템에 심각한 영향을 줄 수 있다.
- 계층 구조 운영체제(윈도우)
 - 비슷한 기능을 수행하는 요소를 그룹으로 묶어 최하위 계층(0)인 하드웨어에서 최상위 계층(5)인 사용자 인터페이스까지의 계층으로 구성된다.
 - 장점: 모듈화가 잘 되어 있어 각 계층은 자신의 하위 계층 서비스와 기능만 사용하여 시스템 검증과 오류 수정을 쉽게 할 수 있다.
 - 단점: 사용자 프로세스의 요청을 수행할 때 여러 계층을 거쳐야 하므로 다음 계층으로 데이터를 전달할 때마다 추가적인 시스템 호출이 발생하여 단일 구조보다 성능이 떨어진다.
- 마이크로 커널 구조 운영체제(OS X)
 - 커널에 최소 기능만 포함시켜 크기를 줄이고 기타 기능은 사용자 공간으로 옮겨 사용자 영역에서 수행한다.
 - 장점: 운영체제 서비스를 사용자 영역의 독립적인 서버에서 수행하므로 서버에서 잘못 수행해도 다른 서버와 커널에 치명적인 영향을 주지 않는다. 그리고 운영체제의 많은 기능을 사용자 영역의 서버로 구현할 수 있어 서버 개발에 용이하고, 운영체제 기능을 쉽게 변경할 수 있다.
 - 단점: 모듈 간에 통신이 빈번하게 발생하여 성능이 떨어질 수 있다.

3.1. 스택 vs 힙

3.2. 힙 영역

4. 프로세스와 스레드

4.1. 프로세스 생성 과정

4.2. cron tab

4.3. 멀티 스레드 vs 멀티 프로세스

4.4. 스레드마다 스택을 독립적으로 할당하는 이유

4.5. 스레드에서 PC 레지스터를 독립적으로 할당하는 이유

5. 컨텍스트 스위칭

5.1. 진행 과정

6. IPC (Inter Process Communication)

7. 프로세스 종류

8. Deadlock

8.1. 교착 상태 해결 방법

9. 전통적인 동기화 문제

9.1. 식사하는 철학자 문제(Dining Philosophers)

9.2. 유한한 공유 버퍼 문제(Bounded-Buffer Problem)

9.3. 읽기 쓰기 문제(Readers-Writers Problem)

10. Critical Section(임계영역)

10.1. 해결방법

11. 경쟁 상태(Race Condition)

11.11. 해결방법

12. 수준 스레드

12.1. 커널 수준 스레드

12.2. 사용자 수준 스레드

13. 사용자 모드와 커널 모드

14. CPU Scheduling

14.1. 프로세스 스케줄러 종류

14.2. CPU 성능 척도

14.3. 스케줄링 방법

15. 동시성과 병렬성

16. 인터럽트

17. 시스템 콜

18. 메모리구조 순서

19. 메모리 관리

19.1. 메모리 관리 전략

20. 가상 메모리

20.1. 가상 주소 공간

20.2. 요구 페이징(Demand Paging)

20.3. 페이지 교체

20.4. Trashing

20.5. 메모리 고갈 상황과 CPU 사용률을 체크하는 이유

21. 캐시의 지역성

22. 캐시 라인

23. Memory Corruption

24. Direct Memory Access (DMA)