**OS 중간고사 정리**

Chapter 1. Introduction

**OS란? 하드웨어 관리를 하는 프로그램**

1. 프로그래머가 쉽게 개발할 수 있도록
2. 하드웨어를 효율적으로 사용할 수 있도록

**OS 정의**

Resource allocator: manages and allocates resources → 자원을 효율적으로, 공정하게 할당

Control program: controls the execution of user programs and operations of I/O devices

**Kernel**: the one program running at all times (all else being application programs)

“OS = kernel + system program”

\*system program: 운영체제를 사용하는 데 필수적인 프로그램 ex)탐색기

Middleware: 운영체제와 어플리케이션 사이. 공통된 것을 묶음

Firmware: 하드웨어와 소프트웨어와 중간.

① 하드웨어로 동작하던 것을 소프트웨어로 만든 것.

② Rom에 저장되어 바꿀 수 없는 소프트웨어

시험 출제

**컴퓨터 시스템 기본 구조**

CPU

Memory

Cache

I/O

I/O

Controller

Hard Disk

\* Bus: 하나만 이동! 그 이상 충돌

**CPU**: 명령어를 수행하는 장치 \* 산술연산, 논리연산, 데이터연산, 분기 필기 참고!

- 레지스터

IR (Instruction Register): PC가 가리키는 곳의 명령어를 읽어와서 저장.

PC (Program Counter): 다음 번에 수행해야 될 명령어를 가리킴.

\* PSW, General-purpose-register 필기 참고!

- 명령어 구조

RISC: CISC를 간단하게 만듦. 선↓, CPU클락↑, pipelining 가능 ⇒ 성능↑

CISC: 모든 명령어. 처음은 CISC로 시작.

Pipelining: 명령어 수행단계를 fetch, decode, execute, write back으로 나누고, 단계별로 수행하는 방법. 처리 속도를 4배 증가시킬 수 있다.

**메모리**

ROM (Read Only Memory): 컴퓨터가 꺼져도 유지되는 메모리

RAM (Random Access Memory): 전원이 꺼지면 날라가는 메모리

SRAM: 값이 유지, 속도 빠름 – cache memory

DRAM: 주기적으로 충전을 해주지 않으면 값이 변함. – 메인 메모리

Cache Memory: 사용된 명령어/데이터 저장. 빠르게 읽어올 수 있기 때문에 자주 쓰일수록 cache memory를 사용하는 것이 효율적이다.

**I/O 장치**

Controller: I/O장치 관리

Interrupt: Controller의 명령어 수행이 끝나거나 이벤트가 발생하면 CPU에게 알리는 것. CPU에 인터럽트가 걸리면 수행하던 것을 저장하고 인터럽트 서비스 루틴을 실행한다. 실행이 완료되면 저장해놓았던 것을 이어서 수행한다.

⇒ Exception: CPU가 자기자신한테 인터럽트를 걸어서 OS에게 알리는 것.

\*CPU가 놀고 있으면 효율↓, A프로세스 수행 중 할 일이 없어지면 CPU는 B프로세스를 수행한다!

System Call: 운영체제가 만들어놓은 함수를 호출하는 것.

DMA (Direct Memory Access): I/O장치가 메인 메모리에 직접적으로 저장한다. 단, CPU가 버스를 쓰지 않을 때 전송한다. (충돌주의)

- OS가 I/O장치 작업이 끝났는지 아는 방법

Poling: CPU가 하던 작업 중에 주기적으로 물어봄. (동기식)

Hardware interrupt: 작업이 끝나면 말해줌. (비동기식)

- I/O장치 방식

Synchronous/Blocking: 입력을 기다림. – 일반적

Asynchronous/Non-blocking: 입력을 기다리지 않고, 다른 동작을 진행함.

**Hardware Protection**

I/O Protection: I/O장치를 보호. OS만 I/O장치에 접근할 수 있다. 어플리케이션은 I/O장치에 접근하기 위해서는 OS에 요청해야 한다. (OS가 제어)

Dual-Mode Operation

1. User mode: OS에 서비스를 요청
2. Kernel mode: OS가 수행되는 모드. 모든 명령어 사용 가능.

\* In/Out과 같은 명령어는 커널 모드에서만 동작한다. 유저 모드의 경우 ‘exception’ – 강제종료

Memory Protection: 참조할 수 없는 메모리 참조 시 ‘exception’ 발생! 메모리를 막 바꾸는 것을 방지한다.

CPU protection: 하나의 프로세스가 CPU를 독차지 하는 것을 막는 것. Timer가 10ms마다 주기적으로 CPU에게 인터럽트를 걸어준다.

Timer: interrupts computer after specified period to ensure operating system maintains control