



1장. 운영체제 개요

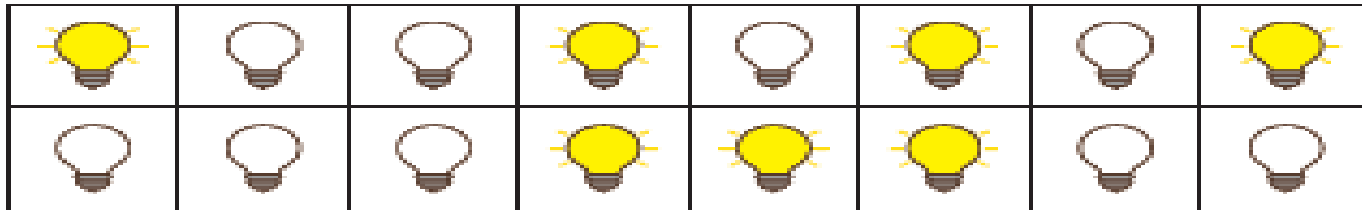
- 운영체제의 정의
- 운영체제의 필요성
- 운영체제의 진입점



학습목표

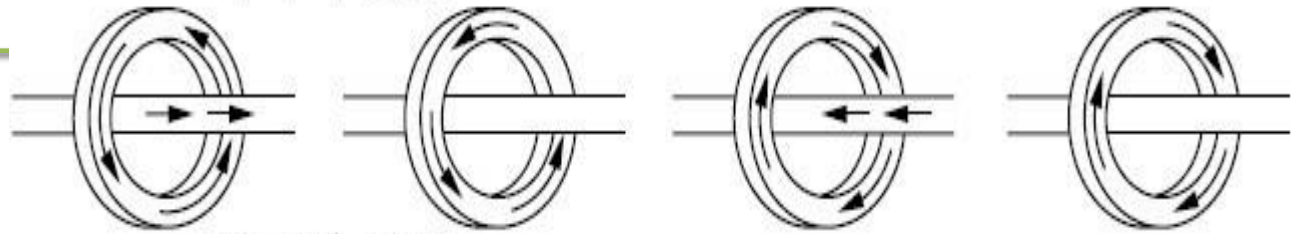
- ▶ 운영체제가 무엇인지 이해한다.
- ▶ 컴퓨터 동작 원리를 통해 운영체제의 필요성에 대하여 이해한다.
- ▶ 운영체제의 진입점을 통해 운영체제의 작동 원리를 이해한다.

사전 지식 : 컴퓨터의 발달

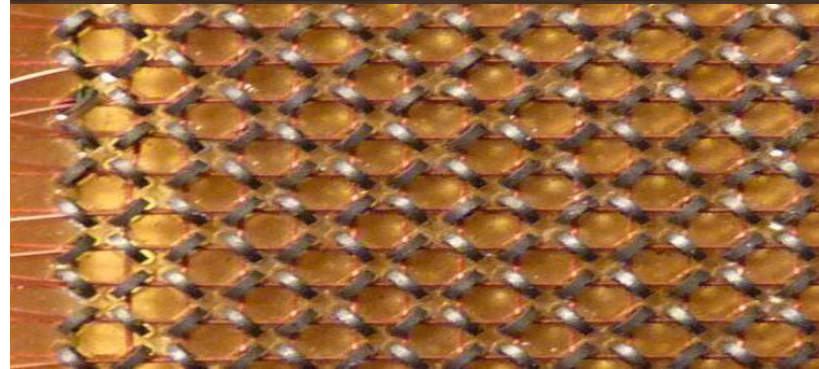
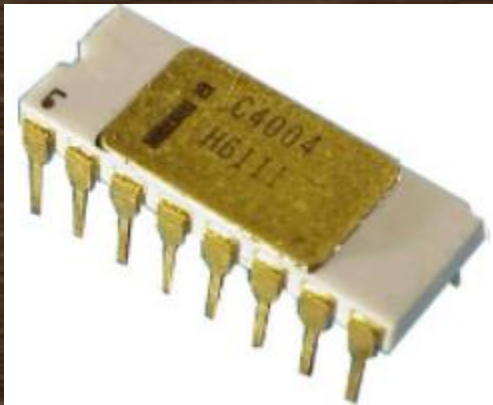


초기의 컴퓨터 개념 1

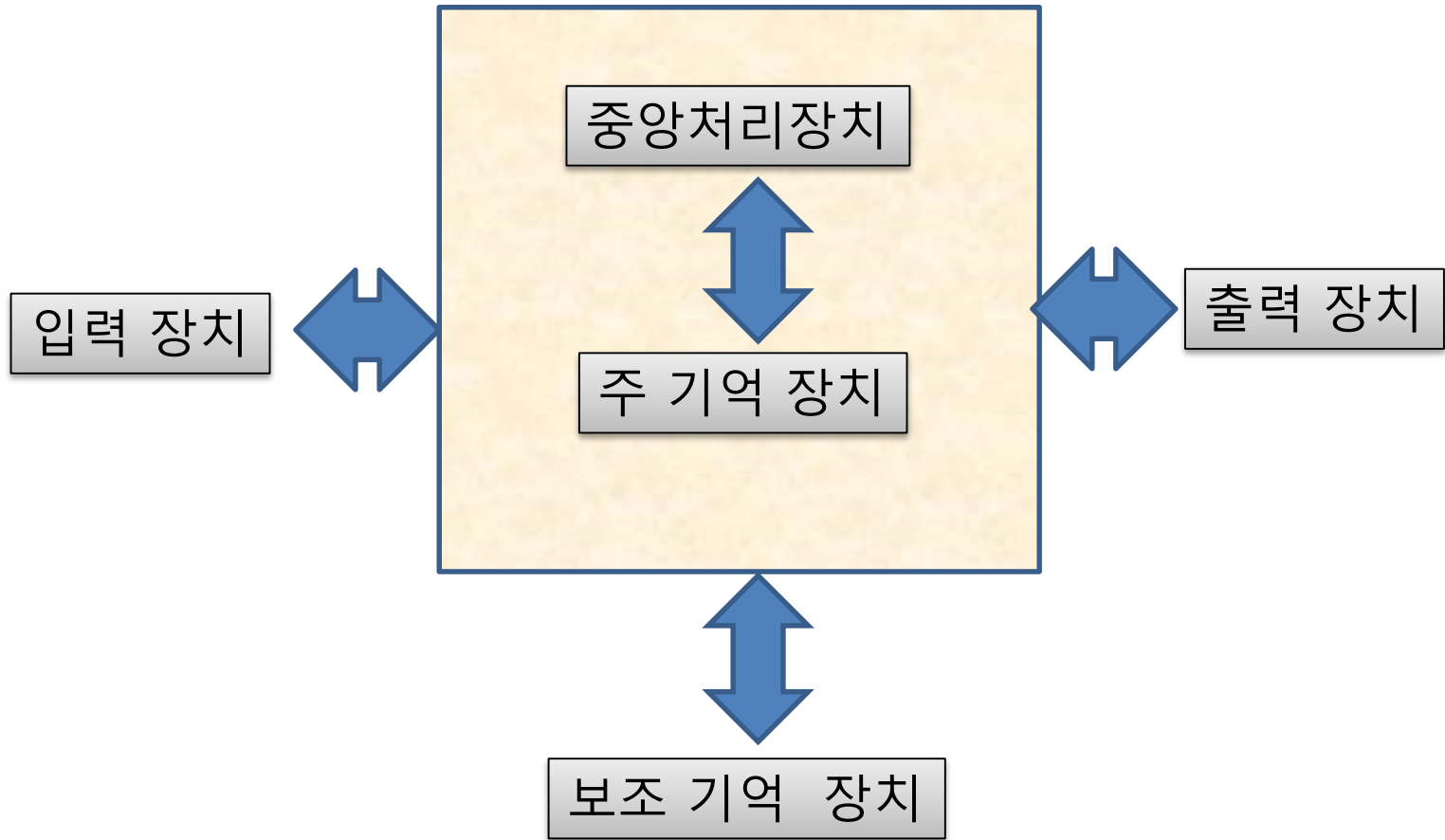
자화의 방향



전류의 방향



초기의 컴퓨터 개념 2

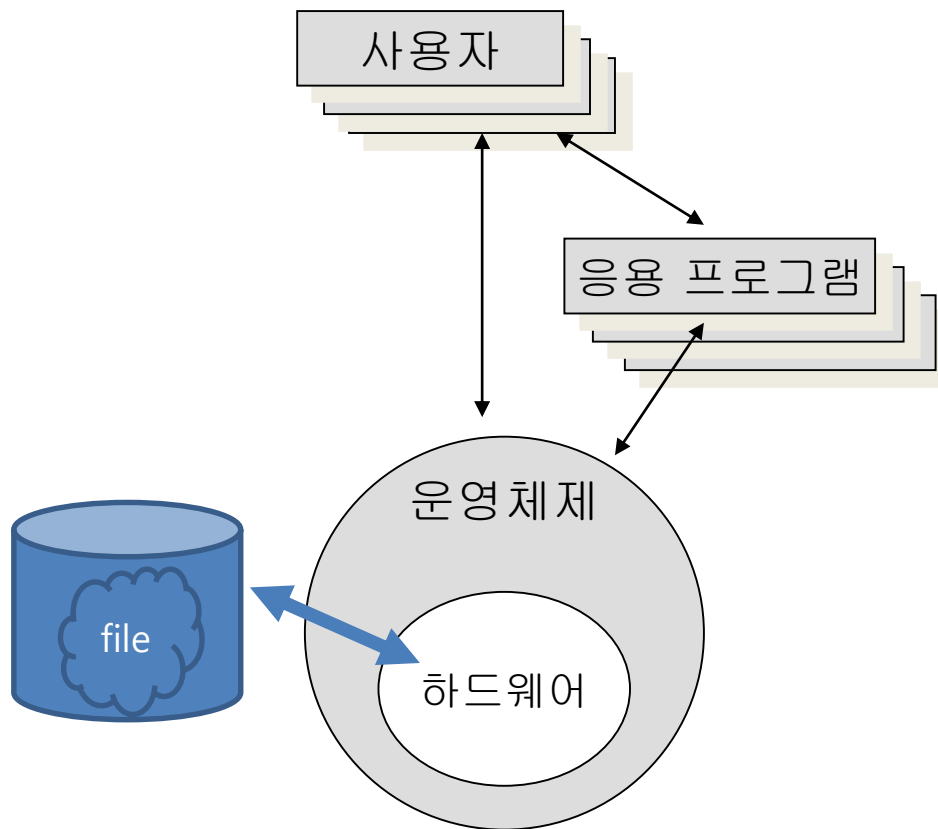


현재 컴퓨터 기본 구조

컴퓨터 구조와 운영체제

그림은 '단일 처리기 시스템에서 다중 사용자의 시분할 처리를 포함한 다중 처리'의 개념도임.

*. 향후 대부분 '다중 처리'로 표현.



User

Application

System S/W
(O/S : Shell & Kernel)

Hardware

운영체제(Operating System) 란?

❖ 정의 : 프로그램의 실행을 효율적으로 제어하고, 입출력 장치와 사용자 사이의 인터페이스를 하는 프로그램

- 사용자 관점 : 컴퓨터 시스템을 보다 편리하게 사용할 수 있는 환경을 제공하는 프로그램 => 편의성 제공
- 시스템 관리자 관점 : 컴퓨터 시스템 자원을 효율적으로 이용할 수 있게 제어하는 프로그램 => 자원의 효율적 관리
- 시스템 관점 : 목적 달성을 위해 여러 구성요소들이 유기적으로 결합된 집합체
 - 목적 : 컴퓨터가 보유하고 있는 자원들을 효율적으로 관리하며, 응용프로그램의 실행을 제어하여 성능을 높이고, 입출력 장치와 사용자 사이의 인터페이스를 수행하여 사용자에게 편의성을 제공을 위한 시스템 S/W

❖ 자원의 종류

- H/W 자원 : CPU, Memory, Disk, Printer, Keyboard...
- S/W 자원 : Service Routine, File ...

❖ 자원의 효율적 관리

- 여러 사용자가 또 여러 프로그램이 자원을 공유하여 프로그램을 실행하므로 프로그램 수행에 문제가 생기지 않는 범위에서 자원의 효율적인 관리 기능(프로그램)이 필요(성능도 함께 고려)

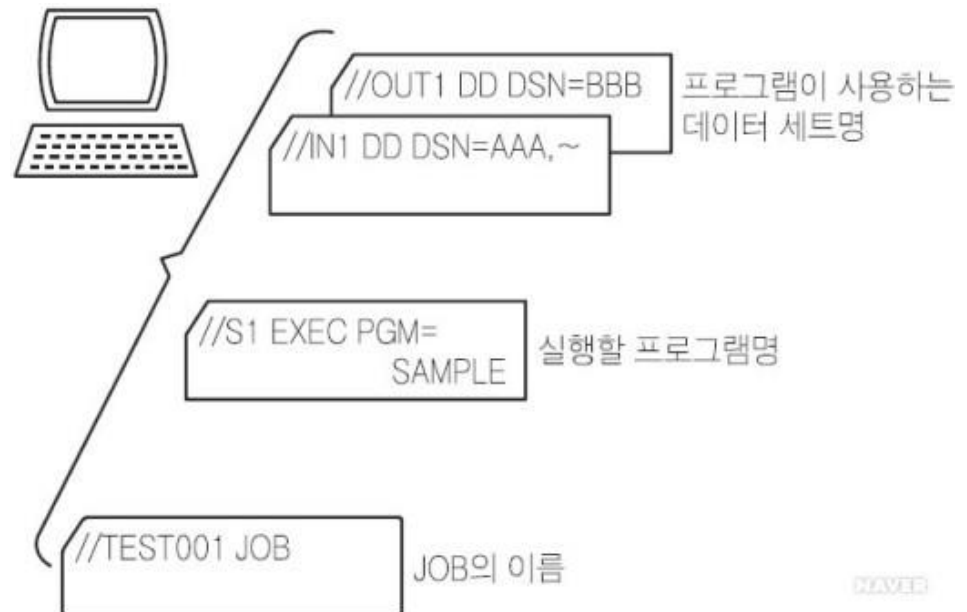
자원의 편의성 제공

❖ 프로그램 작성 및 실행의 편리

- 프로그램 작성 : 기계어 프로그램 => 고급 언어 프로그램
- 프로그램 실행 : Job Control Language => Double Click...

```
001001 11101 11101 1111111111111000
001000 00001 00000 0000000000001010
001000 00001 00001 0000000000000010
101011 11101 00001 0000000000000000
001000 00010 00001 0000000000000100
101011 11101 00010 0000000000000100
001001 11101 11101 0000000000001000
```

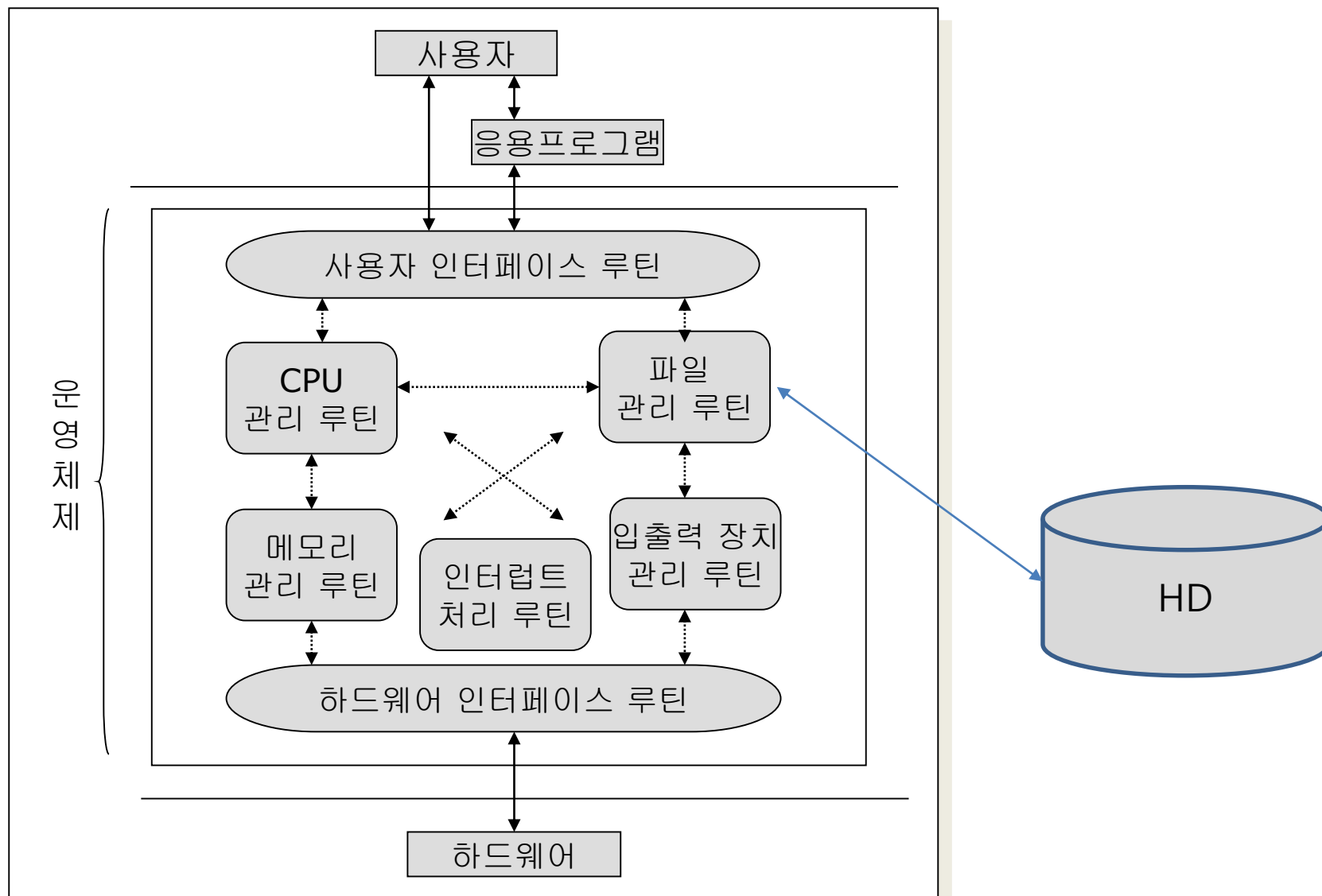
$x = 10 + 2$
 $y = x + 4$



❖ 다중 처리 환경에서 '자원 이용률의 향상'과 '사용자의 편리성 도모'는 이율 배반적 상황.

- 편리성을 위해 한 사용자/프로그램에게 많은 자원을 할당하면 자원의 이용률이 낮아지고, 적게 할당하면 이용률은 증가하나, 편의성은 하락하는 상황이 발생.

범용 운영체제의 구성 요소



운영체제의 필요성

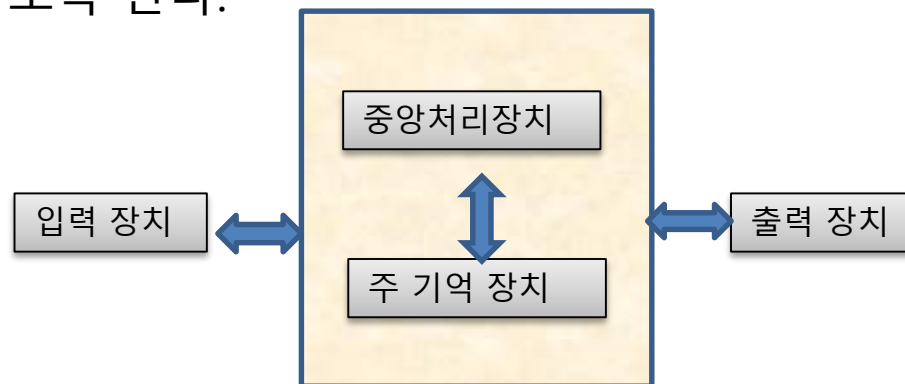
❖ 초기 컴퓨터는 O/S가 없었다. // HW 만으로 구성

- 프로그램 작성부터 모든 작업을 사용자가 해야만 했다.
- 사용이 불편하며, 컴퓨터 자원의 이용률이 저조하였다.

➔ 사용의 불편함을 덜어주고, 자원의 이용률을 향상시키기 위한 프로그램인 운영체제 개발이 요구되었다.

컴퓨터 시스템의 동작 원리

입력 장치를 통해 CPU 명령어들과 필요 자료를 주기억장치에 적재 시켜 놓고 CPU(중앙처리장치)로 하여금 그 명령어들을 하나씩 처리하되 중간 결과도 주기억장치에 기록하며 작업하여 필요한 결과를 출력장치로 출력하도록 한다.



- *. 폰 노이만의 Stored-program-computer 개념
 - 프로그램과 데이터를 내부에 저장
 - EDVAC 개발을 위하여 1945년에 발표.

*. 초기 컴퓨터 : Colossus / ENIAC / EDVAC / EDSAC

컴퓨터 시스템의 동작 원리

*. CPU :

CU + ALU + 다수의 레지스터

*. 레지스터 : PC, IR, MBR, MAR, SR...

CPU : Control Processing Unit

CU : Control Unit

ALU : Arithmetic & Logic Unit

PC : Program Counter

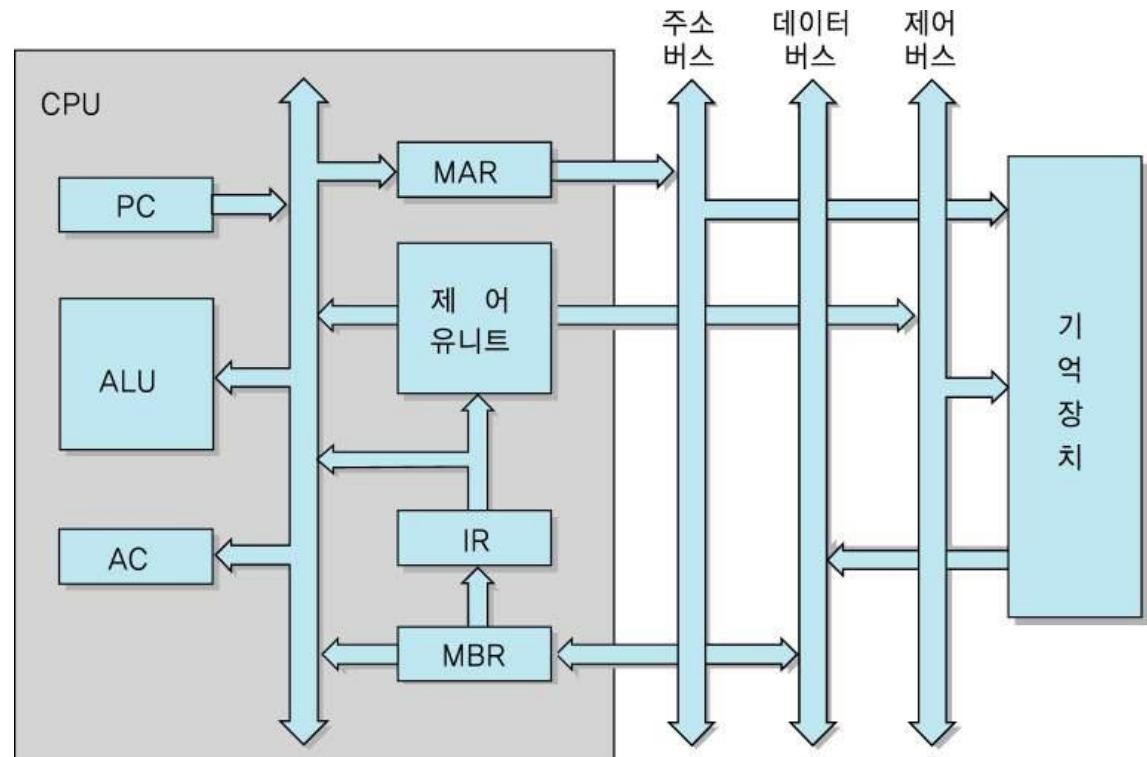
IR : Instruction Register

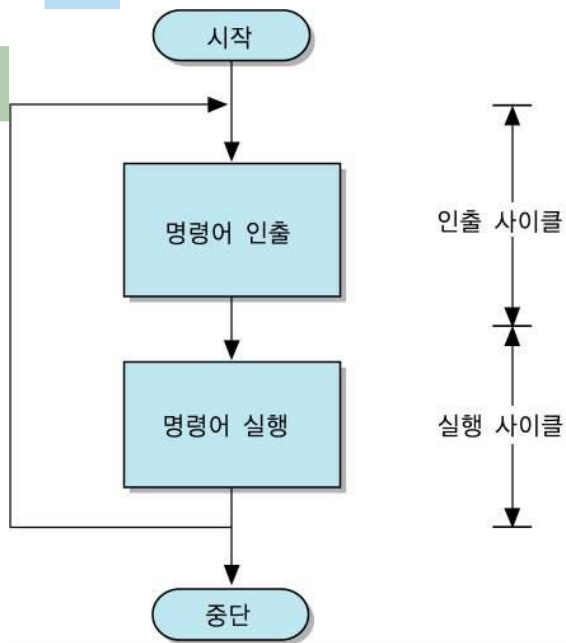
MBR : Memory Buffer Register

MAR : Memory Address Register

PSR : Program Status Register

H/W만으로 목적을 달성할 수 있을까?



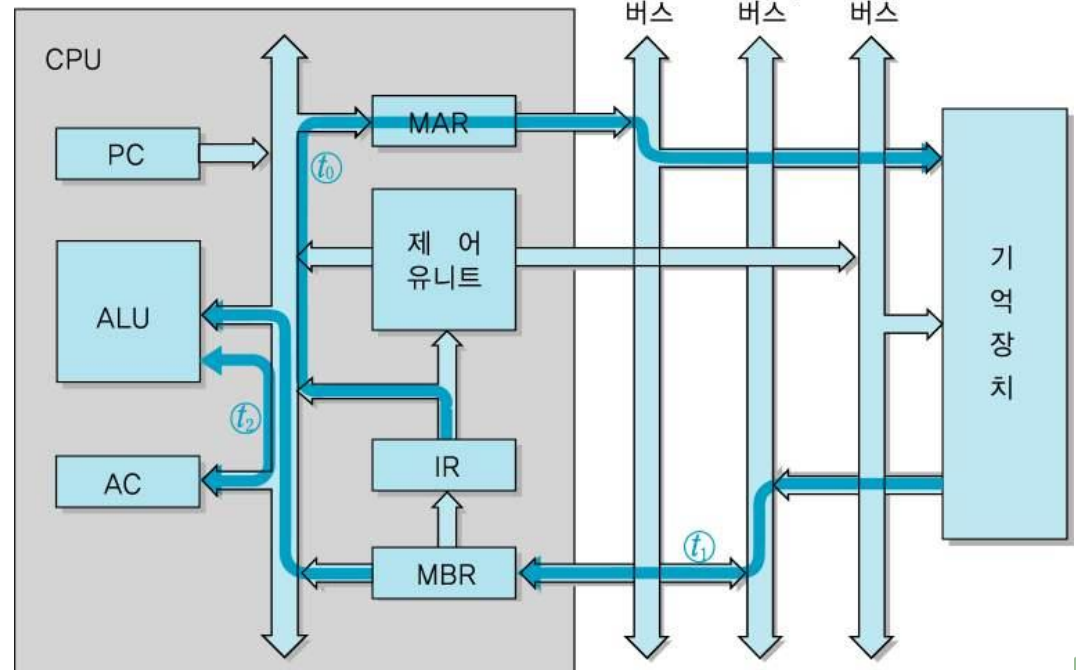
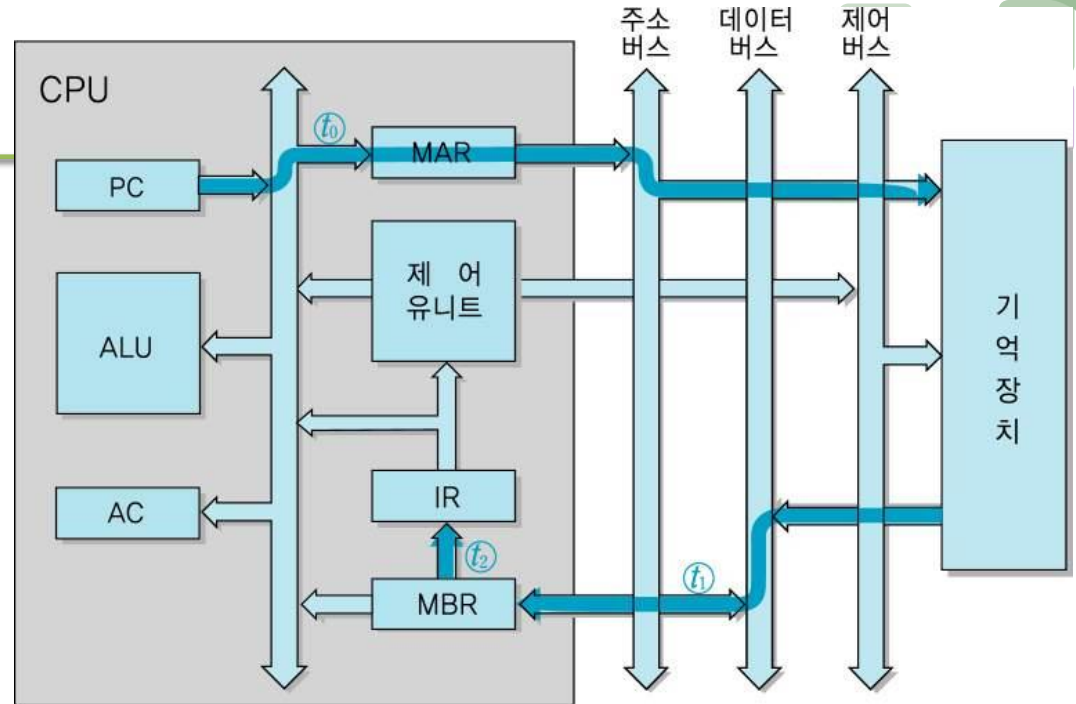


기본 명령어 사이클

<인텔 80x86>
교재 16p-18p

인출 사이클

실행 사이클



운영체제의 발전 과정

❖ 운영체제의 발전 과정

- No O/S > Batch processing(resident monitor) > Multiprogramming > Time-sharing

❖ No O/S

- 사람이 모든 프로그램과 자료 셀팅 등 뒷 일을 수행. 컴퓨터는 단지 계산만 수행

❖ 일괄처리(상주 모니터) 프로그램

- 한 번에 여러 개를 처리 할 수 있었지만 입출력 시 CPU idle

❖ 다중프로그래밍(Multiprogramming) – 여러 개의 작업 수행

- CPU는 비싼 자원
- 빠른 CPU 느린 I/O // 키보드 입력 속도 vs 게이트의 반응 속도
- 메모리에 여러 개의 프로그램을 올려 놓고 수행 중인 프로그램이 입출력을 수행하면 즉 CPU가 idle할 때 다른 프로그램이 CPU를 사용하여 작업
- CPU Scheduling, Memory Management, O/S safety 기능 등이 요구됨

❖ 시분할 시스템 – 여러 사람이 동시 사용 효과

- 강제 절환, interactive system(대화형 시스템)
- 가상메모리, 프로세스간 통신, 동기화 방법 고안

- 프로세서(CPU), 메모리(ROM, RAM), HD, 키보드, 모니터, ...
- POST(Power-On-Self-Test)
- Bootstrap Loader : ROM
- BIOS : Basic Input Output System

❖ 전원을 켜면

- CS:EIP 초기값은 F000:FFF0(FFFF0)로 설정되어 있음.
 - CS:EIP – Program Counter(PC) 역할
- 컴퓨터를 켜자마자 메모리의 FFFF0 번지 명령어를 수행하라는 의미(FFFF0번지에는 ROM-BIOS 프로그램이 있음을 의미함) => POST 수행
- ROM-BIOS 프로그램은 POST 후 Boot Program을 설정된 보조기억 장치로부터 가져와(Bootstrap Loader) 7C00번지부터 적재하고 CS:EIP를 07C0:0000으로 설정.

❖ 7C00부터 명령어 실행

- 7C00부터의 프로그램은 윈도우와 같은 운영체제 프로그램.
- 운영체제 프로그램이 수행(초기화)된다.
- 운영체제 초기화 후 무한루프 돌면서 사용자의 명령어를 기다린다.

❖ 응용 프로그램 실행

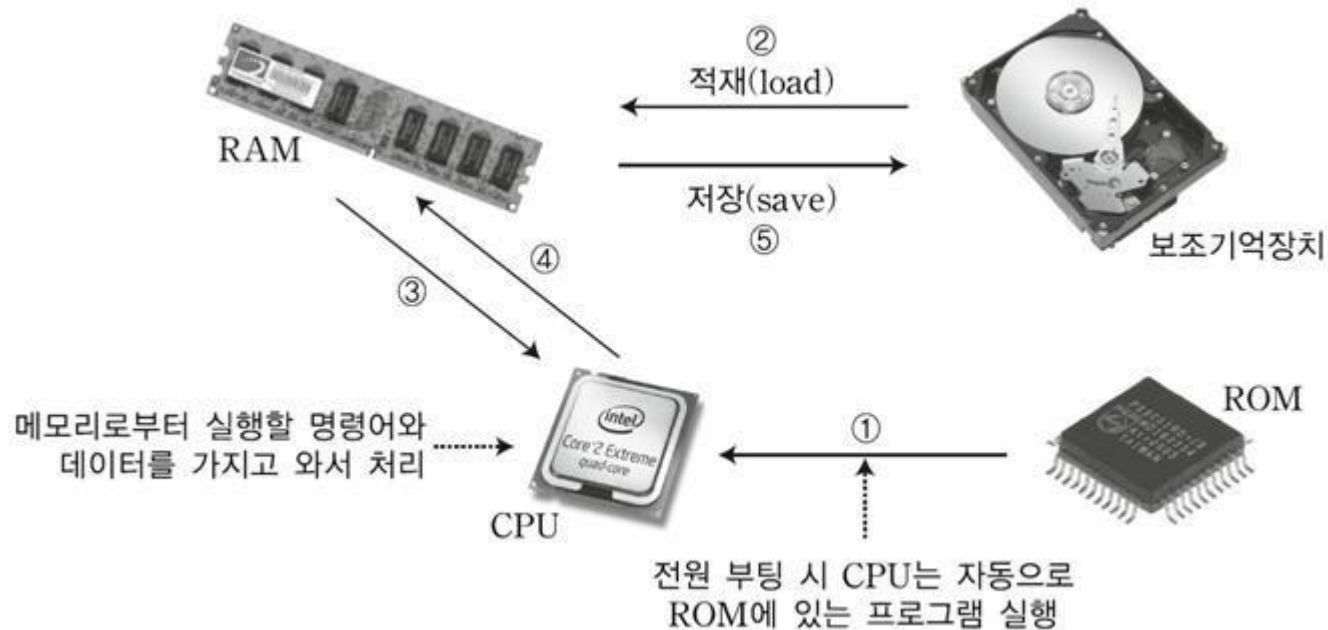
- 더블클릭 : 마우스의 더블클릭 인터럽트 처리(인터럽트 처리)
- 응용프로그램 탐색 : 보조기억장치에서 해당 응용프로그램 탐색(파일관리)
- 주기억장치 빈 공간 탐색 : 프로그램을 적재할 주기억장치 빈 공간 탐색(메모리관리)
- 프로그램 적재 : 보조 기억장치의 응용프로그램을 메모리에 적재(입출력장치관리)
- CS:EIP 값 조작 : 프로그램이 적재된 시작 주소를 CS:EIP에 저장하여 수행할 수 있게 한다.(CPU 관리)

PC 부팅과정

인텔 80x86 기준

*. 운영체제가 있는
컴퓨터의 부팅 모습을
볼 수 있다.

- 프로세서(CPU), 메모리(ROM, RAM),
HD, 키보드, 모니터, ...
- POST(Power-On-Self-Test)
- Bootstrap Loader : ROM
- BIOS : Basic Input Output System



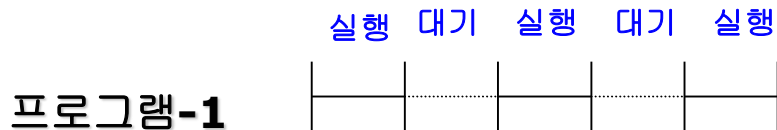
❖ 다중 프로그래밍 개념의 등장 배경

- Batch processing O/S에서 한 번에 여러 개의 Job을 처리할 수는 있었지만 수행하는 프로그램이 입출력 하는 동안 CPU는 idle(유휴). 즉 CPU의 이용률과 프로그래밍 처리율(throughput)이 낮았음.
- CPU의 idle time을 다른 Job이 이용하자는 아이디어 대두.

❖ 다중 프로그래밍의 기본 개념

- 하나의 프로그램이 수행하는 과정에서 입출력 요구가 발생하게 되면 입출력을 완료할 때까지 CPU는 작업을 하지 않게 됨.
- 이러한 CPU 유휴 시간을 사용하기 위해, 여러 개의 프로그램을 메모리에 적재 시켜 놓고,
- 실행중인 프로그램이 입출력을 수행 시 대기중인 프로그램이 CPU 사용을 함으로 CPU 이용률과 프로그램 처리율을 향상시키는 것.

다중 프로그래밍 기법의 효율



*. 실선은 CPU 사용시간 점선은 CPU 유휴시간



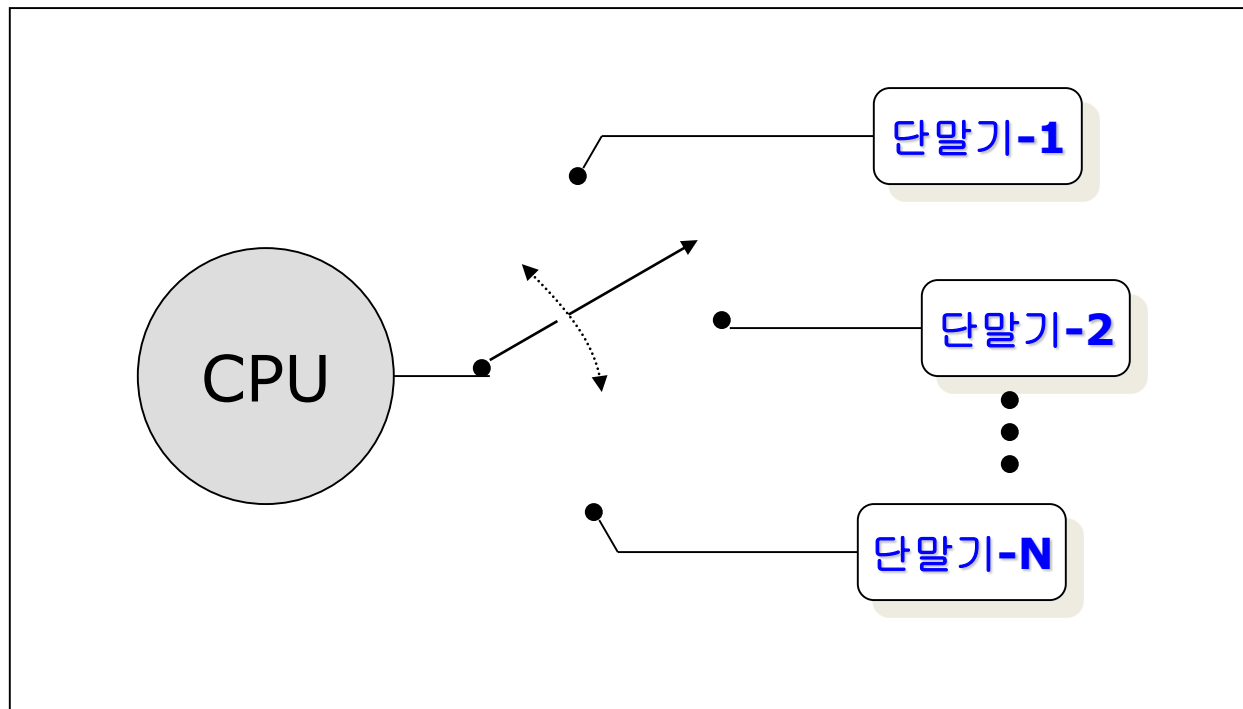
(a) 다중 프로그래밍 기법을 이용하지 않을 경우



(b) 다중 프로그래밍 기법을 이용할 경우

- 메모리를 어떻게 분할하고 할당할까? (분할전략 : 고정, 가변 / 할당전략 : 연속, 불연속)
- CPU 유휴 시간은 왜 발생할까?
- CPU 유휴 시간을 어떻게 이용할까?
- CPU 이용률과 시스템 처리율은 향상될까?
- 사용자들은 동시 처리 효과를 느낄 수 있을까?
- 규칙적으로 CPU 사용을 반납하지 않으면 어떻게 할까?

시분할 처리



* 단말기 :
모니터 + 키보드.

여러 사용자가 각자의
단말기에서 자기
프로그램을 운용.
자기가 컴퓨터를 쓰는 느낌.

강제 절환, interactive system(대화형 시스템)
가상메모리, 프로세스간 통신, 동기화

운영체제의 기능(다중-시분할 환경)

다중-시분할 환경이 범용
운영체제 기본 환경임

❖ 기능의 종류

- CPU 관리(프로세스 관리)
 - 프로그램의 실행 / 프로그램 종료 / 프로그램 간 CPU 공유 처리
- 메모리 관리(주기억장치 관리)
 - 메모리 사용 영역과 빈 영역 유지 및 관리 / 새로운 영역 요구와 사용 영역 반납 처리 / 사용하지 않는 공간 파악 및 사용 않는 공간 회수 기능
 - 가상 메모리(Virtual memory) 사용 시 논리 주소의 물리 주소로 변환
- 파일 관리 기능
 - 보조 기억 장치에 파일 생성 및 저장 / 요청 파일의 읽기, 생성된 파일 삭제 처리
- 입출력 장치 관리
 - 실행 중인 프로그램에게 필요 입출력 장치를 사용할 수 있도록 처리
- 보호 기능
 - 공유하는 자원에 대한 보호 기능
- 오류 처리와 복구 기능
 - 컴퓨터 시스템의 오류 탐지 및 복구 또는 사용자에게 경고 처리
- 네트워킹 기능
 - 컴퓨터 시스템 간 통신을 위한 기능

운영체제 진입점

- ❖ 다중 프로그래밍 환경에서 운영체제를 포함하여 다수의 프로그램이 메모리에 적재되어 있음.
- ❖ 프로그램에서 컴퓨터 자원을 사용하려면 운영체제의 도움을 받아야 함.
- ❖ 운영체제의 도움이란 운영체제 명령어가 수행되는 것.
- ❖ 사용자 프로그램의 명령어가 수행되다 운영체제 도움이 필요하면 PC 레지스터 값을 운영체제 범위 주소로 변경되어야 함.
- ❖ 진입점이란 PC 레지스터 값이 운영체제 범위 주소로 바뀌고 프로그램 실행이 운영체제가 되는 것을 말함.

운영체제 영역 진입 통제

❖ 운영체제 보호를 위해 운영체제 진입 통제

❖ 운영체제 보호 방법

- 프로그램 실행 상태를 '사용자 모드와 커널(kernel) 모드'로 구별
 - 커널 모드(Kernel mode) PC가 운영체제 범위 안의 주소를 가리키고 있음.
 - 운영체제 실행 상태. System Mode / Super visor mode.
 - 사용자 모드(User mode) PC가 사용자 프로그램 범위 주소를 가리키고 있음.
 - 사용자 프로그램 실행 상태.
- 프로그램 실행 상태 구별을 위해 '프로그램 상태 레지스터(PSW : Program Status Word) 이용

*. 운영체제가 수행하는 명령을 특권명령(Privileged Instruction)이라고 함

커널 모드 필요 예

❖ 사용자 모드 사용 제한 / 커널 모드에서만 실행

- 입출력 명령
- 다른 프로그램 메모리 접근
- 다른 사용자의 파일 접근 명령
 - 인쇄 내용 혼재 야기, 사용중인 프린터 리셋 야기, 남의 파일 읽기/쓰기

사용자 모드에서 커널 모드로 변환 종류

- ❖ 시스템 호출(system call)
 - ❖ 인터럽트(Interrupt)
 - ❖ 예외 현상(Exception)
- ❖ 위 내용을 처리해 주기 위한 서비스 루틴이 OS에 존재함.

❖ 운영체제 내부에 구현되어 있는 서비스(루틴)를 요청 방법

- 응용 프로그램과 운영체제의 인터페이스 수단
- API(Application Programming Interface)라고 부른다.
 - cf) API Programming
- 시스템 호출은 함수의 수행 과정과 유사하게 작동

인터럽트(Interrupt)

❖ 입출력 장치에서 CPU에게 처리해 주어야 할 일이 있음을 알리는 신호

- CPU와 입출력 장치 사이의 인터페이스 수단
 - CPU는 입출력 장치에게 명령을 보내고
 - 입출력 장치는 인터럽트로서 CPU가 처리해 주어야 할 일이 있음을 알림.

- *. 입출력 장치의 처리속도는 CPU 보다 훨씬 느리다.
- *. 입출력 장치는 고유의 장치 제어기(device driver : sw)에 의해 제어된다.
- *. CPU와 입출력 장치 사이의 인터페이스 : 폴링 혹은 인터럽트

- *. 폴링 : 대상을 주기적으로 감시하여 입출력 상황이 발생하면 해당처리를 실행하는 방법.
- *. 인터럽트 : 상대(입출력 장치)가 프로세서(중앙)에게 처리 요청을 하면 일 처리하는 방법.

예외 현상(Exception)

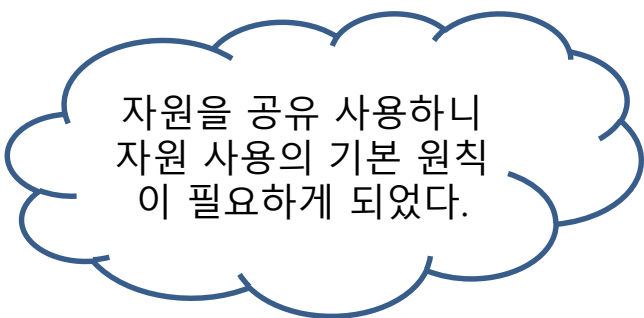
❖ 프로그램 실행 중에 발생하는 금지 상황.

- 금지 상황 예
 - divide by zero : 어떤 수를 0으로 나누기()
 - segment violation : 운영체제 영역 또는 다른 사용자 영역 침범
- 예외 현상 처리 루틴은 인터럽트 처리 루틴처럼 수행

자원 관리의 기본 원칙

❖ 기본 원칙

- 요청(request)
 - 사용자 프로그램이 자원 사용 허가를 운영체제에게 요청.
- 사용(usage)
 - 사용자 프로그램이 허가된 자원을 사용.
- 해제(release)
 - 사용자 프로그램이 자원 해제를 운영체제에게 통보한다.



자원을 공유 사용하니
자원 사용의 기본 원칙
이 필요하게 되었다.

입출력 장치 관리

- ❖ 인터럽트 처리
- ❖ 디바이스 드라이버(장치 구동기)
- ❖ 성능 향상을 위한 버퍼링 처리
- ❖ 사용자의 편의성을 위한 일관된 접근 방법 제공

- *. 입출력 장치의 처리속도는 CPU 보다 훨씬 느리다.
- *. 입출력 장치는 고유의 장치 제어기(device driver)에 의해 제어된다.
- *. CPU와 입출력 장치 사이의 인터페이스 : 폴링 혹은 인터럽트

- *. 폴링은 중앙에서 대상을 주기적으로 감시하여 입출력 상황이 발생하면 해당처리를 실행.
- *. 인터럽트는 상대(입출력 장치)가 프로세서(중앙)에게 처리 요청을 하면 일 처리를 실행.

❖ O/S는 다수의 사용자 프로그램에 의해 공유됨.

❖ 자원의 보호를 위해 이중 모드를 사용

■ .

■ 모드 상태는 CPU 내의 '프로그램 상태 레지스터'로 표시

❖ 운영체제 진입점

■ 운영체제로의 진입 통제 : 사용자 모드에서 커널 모드로 변환 요청.

• 시스템 호출 : 사용자 프로그램이 OS에게 서비스 요청(예 : 입출력). SW 인터럽트. API(Application Programming Interface)라고도 부름.

• 인터럽트 : HW Interrupt. 입출력 장치에서 입출력 할 것이 있을 때 보내는 신호.

❖ 하드웨어 보호 예

■ IO device Protection / Memory Protection / CPU Protection

❖ 한 사용자가 실수 또는 고의로 CPU를 독점하는 것을 방지

`while (i=1) { ... } //무한루프로 CPU 독점`

❖ 해결법

- Timer를 두고 일정 시간 후 Timer interrupt 발생.
- OS가 다른 프로그램 수행으로 강제 전환

❖ UNIX 운영체제의 특징

- 다중 처리(multi-tasking) 시스템
- 이식성(portability)
 - 기계어가 아닌 C언어로 작성 다양한 컴퓨터에 설치 가능
- 계층적 구조의 파일 시스템
- 입출력 장치를 보통 파일과 동일한 방법으로 취급
- 풍부한 유틸리티 및 응용 프로그램
- 대학 교육용으로 보급 ==> Berkely 대학 / 30년 동안의 가장 대표적인 운영체제로 군림
- UNIX와 매우 유사한 Linux 시대 도래

Linux 운영체제의 구조

❖ 모노리틱 구조

- 하나의 커다란 프로그램 덩어리

