



운영체제 - 반효경 (이화여자대학교)

운영체제란 무엇인가?

운영체제(Operating System, OS)란?

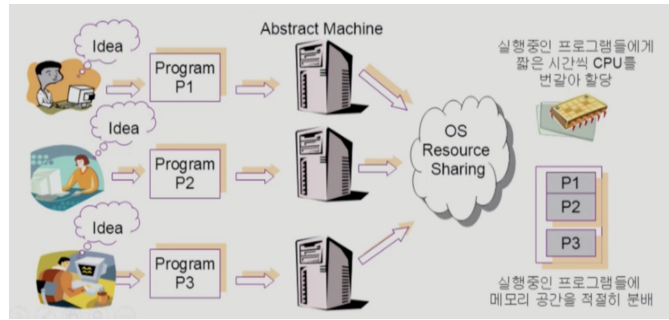
: 컴퓨터 하드웨어 바로 위에 설치되어 사용자 및 다른 모든 소프트웨어와 하드웨어를 연결하는 소프트웨어 계층



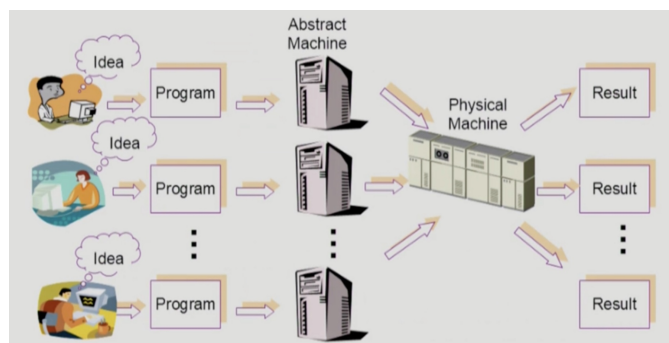
- 협의의 운영체제(커널) : 운영체제의 핵심부분으로 메모리에 상주하는 부분
 - 좁은 의미의 운영체제, 전공자 입장에서 운영체제는 보통 커널을 의미
- 광의의 운영체제 : 커널 뿐 아니라 각종 주변 시스템 유틸리티를 포함한 개념 - 메모리에 상주하지 않는 일종의 별도의 프로그램들
 - 넓은 의미의 운영체제

운영체제의 목적

- 컴퓨터 시스템의 **자원을 효율적으로 관리**



- 프로세서, 기억장치, 입출력 장치 등의 효율적 관리
 - : 사용자간의 형평성 있는 자원 분배 : 형평성
 - : 주어진 자원으로 최대한의 성능을 내도록 : 효율성
- 사용자 및 운영체제 자신의 보호
- 프로세스, 파일, 메시지 등을 관리 : 소프트웨어 자원도 관리를 해야할 책임을 가지고 있음
- 컴퓨터 시스템을 편리하게 사용할 수 있는 환경을 제공



- 운영체제는 동시 사용자/프로그램들이 각각 독자적 컴퓨터에서 수행되는 것 같은 환상을 제공
- 하드웨어를 직접 다루는 복잡한 부분을 운영체제가 대행

운영체제의 분류

1. 동시작업 가능 여부

- 단일 작업(single tasking)
 - 한 번에 하나의 작업만 처리

ex) MS-DOS 프롬프트 상에서는 한 명령의 수행을 끝내기 전에 다른 명령을 수행시킬 수 없음

- 다중 작업(multi tasking)
 - 동시에 두 개 이상의 작업 처리

ex) UNIX, MS Windows 등에서는 한 명령의 수행이 끝나기 전에 다른 명령이나 프로그램을 수행할 수 있음

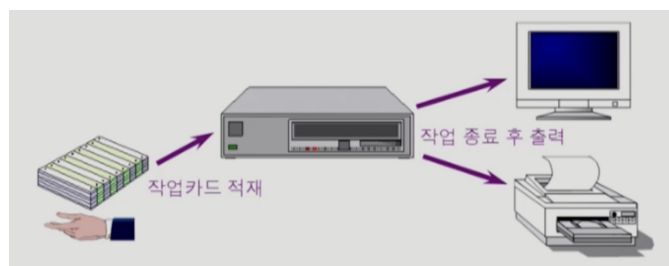
2. 사용자의 수

: 여러 사용자의 계정을 만들어서 동시접근을 할 수 있느냐 없느냐에 따른 기준으로 나눔

- 단일 사용자(single user)
 - ex) MS-DOS, MS Windows
- 다중 사용자(multi user)
 - ex) UNIX, NT server

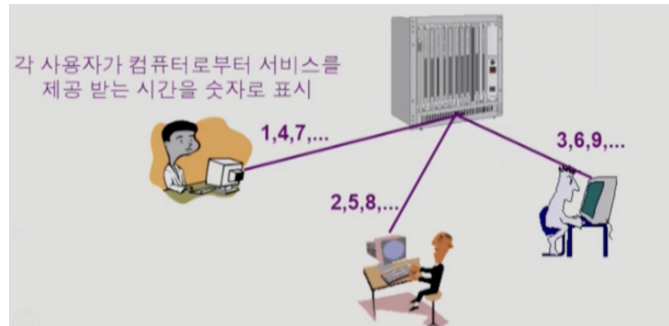
3. 처리방식

- 일괄처리(batch processing)



- 작업 요청의 일정량 모아서 한꺼번에 처리
- 작업이 완전 종료될 때까지 기다려야 함
- ex) 초기 Punch Card 처리 시스템

- 시분할(time sharing) : 일반적인 범용 컴퓨터의 사용되는 운영체제에서 사용



- 여러 작업을 수행할 때 컴퓨터 처리 능력을 일정한 시간 단위로 분할하여 사용
- 일괄 처리 시스템에 비해 짧은 응답 시간을 가짐 ex) UNIX
- interactive 한 방식

- 실시간(Realtime OS) : 특수한 목적을 가진 시스템에서 사용, 정교한 시스템
 - 정해진 시간 안에 어떠한 일이 반드시 종료됨이 보장되어야 하는 실시간 시스템을 위한 OS
 - ex) 원자로/공장 제어, 미사일 제어, 반도체 장비, 로봇트 제어

- 실시간 시스템의 개념 확장
 - Hard realtime system(경성 실시간 시스템)
 - Soft realtime system(연성 실시간 시스템)

몇가지 용어 정리

- Multitasking
- Multiprogramming
- Time sharing
- Multiprocess
- 구분
 - 위의 용어들은 컴퓨터에서 여러 작업을 동시에 수행하는 것을 뜻함
 - Multiprogramming 은 여러 프로그램이 메모리에 올라가 있음을 강조

- Time Sharing 은 CPU 의 시간을 분할하여 나누어 쓴다는 의미를 강조

- Multiprocesseor : 하나의 컴퓨터에 CPU(processor) 가 여러 개 붙어있음을 의미

: 우리가 여러작업이 동시에 실행되는 걸 보통 멀티 태스킹이다 이렇게 많이 부른다. 멀티 태스킹은 좀 더 일반적인 용어이다. 사실은 태스크가 여러개 동시에 돌아가고 있지만 정확하게 엄밀히 말하면 CPU 에서는 매 순간은 하나의 작업만을 실행만 되고 있다. 그걸 짧은 시간으로 분할해서 할당하기 때문에 마치 동시에 실행되는 것처럼 보이지만 이런 식으로 진행되는 작업 컴퓨터 전반에서 이런식으로 진행되는 것을 멀티태스킹이라고 부른다.

하나의 프로그램이 끝나기 전에 다른 프로그램이 컴퓨터에서 실행이 가능하고 멀티태스킹이라고 부르고 또 다른 말로는 멀티프로그램이라고도 부른다. 근데 조금 전문적으로 구분을 하자면 멀티프로그래밍이라는 용어는 메모리에 여러 프로그램이 동시에 올라가는 그러한 방식은 우리가 멀티프로그래밍이라고 부른다. 물론 멀티태스킹이 되려면 즉 여러프로그램이 동시에 실행이 되려면 당연히 메모리에 여러 프로그램이 동시에 올라가야하지만 특별히 메모리측면을 강조해서 메모리에 동시에 여러 프로그램이 동시에 올라가는 그러한 방식을 멀티프로그래밍이란 용어로 설명을 한다.

그다음에 동시에 실행되는 걸 다르게 타임쉐어링이라는 말로도 사용함. 타임쉐어링은 시간을 공유해서 쓰는 방식인데 이건 아무래도 cPU를 강조한측면이 된다. 시간을 조금씩 쪼개서 이 프로그램이 실행되다가 저프로그램이 실행되다가 이런식으로 진행이 되기 때문에 타임쉐어링도 멀티태스킹이랑 유사한 용어지만 cpu쪽을 좀 더 강조한 용어라고 설명 가능

멀티프로세스도 사실 비슷한 용어라고 볼 수 있음 프로세스라는게 사실 실행중인 프로그램이고 여러프로그램이 동시에 실행된다.

네가지 용어는 유사한 용어로 우리가 사용이되고 우리 수업시간에 배우는 운영체제가 이러한 그 이러한 기능을 지원해주는 운영체제를 우리수업시간에 배우는 것

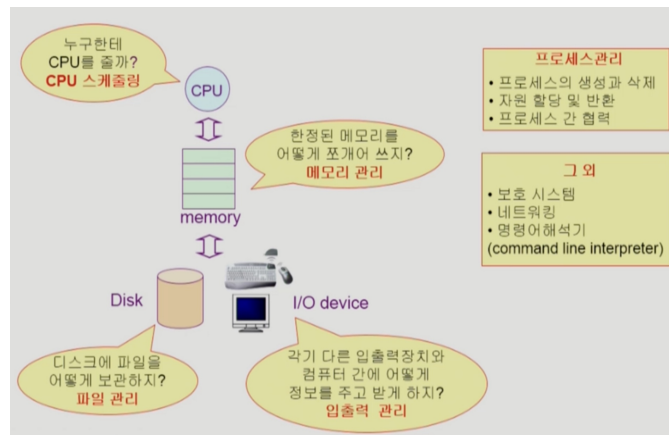
멀티 프로세서

프로세서라는 건 보통 cpu 를말함 처리기라고도 말함 다중처리기 , cPU가 여러개 있는 컴퓨터를 멀티 프로세서라고 부른다. cPU가 여러개 있는 컴퓨터에서는 여러 프로그램이 cpu가 동시에 들어가 있을 수 있다. cPU가 하나 있더라도 시간을 조금씩 나누어서 동시에 실행이 가능한 그런 시스템을 이야기하는데 비해서 멀티프로세서는 기본적으로 이런 멀티태스킹이 기본적으로 지원이 된다. cpu가 여러개 있어야 되는 하드웨어적으로 다른 시스템이라는 것

운영체제의 예

- 유닉스(UNIX)
 - 코드의 대부분을 C언어로 작성
 - 높은 이식성
 - 최소한의 커널구조
 - 복잡한 시스템에 맞게 확장 용이
 - 소스 코드 공개
 - 프로그램 개발에 용이
 - 다양한 버전 : System V, FreeBSD, SunOS, Solaris / Linux
- DOS(Disk Operating System)
 - MS사에서 1981년 IBM-PC를 위해 개발
 - 단일 사용자용 운영체제, 메모리 관리 능력의 한계(주 기억장치 : 640KB)
- MS Windows
 - MS사의 다중 작업용 GUI 기반 운영체제
 - Plug and Play, 네트워크 환경 강화
 - DOS용 응용 프로그램과 호환성 제공
 - 불안정성(초창기)
 - 풍부한 지원 소프트웨어
- Handheld device를 위한 OS
 - PalmsOS, Pocket PC(WinCE), Tiny OS

운영체제의 구조



컴퓨터의 자원들

• CPU

: 어떤 프로그램한테 CPU를 할당할 것인가 - CPU 스케줄링

: 먼저 온 것을 먼저 처리해주는 것 - 일반적으로 생각하는 스케줄링 → 컴퓨터입장에서는 효율적이지 못함

• 메모리 관리

: 어떤 프로그램한테 얼마큼 메모리를 할당해야 하는가

: 메모리를 분배하는 이슈에 대해서는 프로그램들이 점점 많아지면 각각이 갖는 메모리공간이 너무 작아진다 → 프로그램이 원활하게 실행이 되지 않는다.

: CPU에서 어떤 프로그램이 실행이 되려면 최소한 그 프로그램을 실행에 필요한 부분은 메모리에 올라와 있어야 한다. 프로그램이 너무 많아서 메모리를 지나치게 조금씩 가지고 있다고 하면 cpu에 실행하려고 하면 그 부분이 메모리에 없다. 그러면 디스크에서 읽어와야하면 프로그램이 원활하게 실행이 되지 않는다.

: 메모리에다가 너무 많은 프로그램을 동시에 올려놓는 것 보다는 적어도 cPU 에서 원활하게 실행되게 하기 위해서 필요한 용량이 있다. 그 부분 만큼은 프로그램 하나가 그정도 메모리는 받을 수 있도록 하는 그런 것을 지원해주자 → 워킹셋 모델이라고 해서 너무 많은 프로그램이 동시에 실행되려고 하면 일부 프로그램만 어느정도 메모리 공간에 할당해주고 나머지는 디스크로 보내버려서 이미 올라가 있는 프로그램만이라도 cpu에서 원활하게 실행이 되게 하고 어느정도 실행이 됐으면 경우에 따라 좀 내려놓고 아까 보내졌던 친구들이 메모리에 올라오는 이런 식의 관리

무조건 n분의 일로 나누는것보다 그때그때 집중적으로 몰아주면 더 원활하게 실행이 될 수 있는 이러한 원리

메모리는 크기가 한정되어있기 때문에 꽉 차있으면 디스크로 보내야한다. 그래서 어떤것을 보낼것인지도 중요

CPU에서 다시 사용할 가능성이있는 것들은 보관하고 그렇지 않은 것을 보낸다.

미래는 어떻게 예측하는가 ?- 메모리에서 많이 사용이 됐으면 미래에도 사용될 가능성이 높겠구나 판단 / 최근에 사용된 적이 없거나 몇번 사용하지 않은 것들은 메모리에서 나가짐

- 파일 관리

: 조각조각을 나누어서 저장할지 연속적으로 저장할지

: 디스크에 특성에 맞게 파일을 관리하고 디스크도 스케줄링이 필요하다.

: 디스크는 헤드가 움직이면서 읽고 쓰기 때문에 디스크 헤드의 이동을 줄이면 성능이 좋아짐

: 디스크도 헤드가 이동하면서 중간에 요청이 들어오면 처리하면서 지나가는 것이 효율성이 좋음

cpu 스케줄링하고 디스크 스케줄링하고는 성격이 조금 다름

: cpu는 너무 빠르기 때문에 어떻게 관리할 것인가

: 디스크는 헤드가 움직이면서 처리하기 때문에 어떻게하면 움직임을 최소화하면서 요청을 빠르게 처리할 수 있는가

- I/O device 입출력 관리