

170111 – 운영체제,자료구조,알고리즘

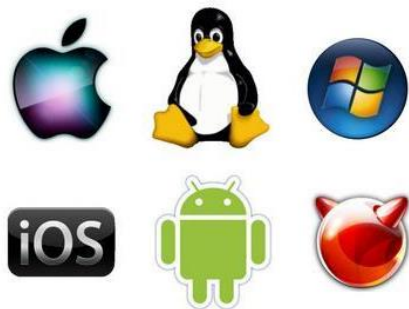
운영체제(Operating System)란?

배워야 하는 이유 : 우리가 사용할 소프트웨어가 어디에서 어떻게 사용되는지 알아야하기 때문

운영체제는 응용프로그램과 하드웨어 사이의 중재자이다.

운영체제의 탄생 배경 : 옛날 컴퓨터 사용목적이 점점 다양해지고 있는데 프로그램을 바꾸려면 매번 새로운 구동 시스템을 만들어야 했다. 하지만 그것은 너무 비효율적이기 때문에 공통적으로 하드웨어를 제어하고, 공통적으로 프로그램을 받아들이는 시스템이 필요했기 때문에 Operating System이 생긴 것입니다.

- 운영체제의 다양한 종류 -



커널이란 – 운영체제의 근간이 되는 뼈대

Ex) 리눅스 커널에서 안드로이드, Fedora CentOS라는 운영체제를 쓰는 것

운영체제가 하는 일

- 시스템 하드웨어관리

사용자 프로그램의 오류나 잘못된 자원 사용을 감시

입출력 장치 등의 자원에 대한 연산과 제어를 관리

- (가상)시스템 서비스 제공

사용자에게 컴퓨터의 프로그램을 쉽고 효율적으로 실행할 수 있는 환경 제공

- 자원관리

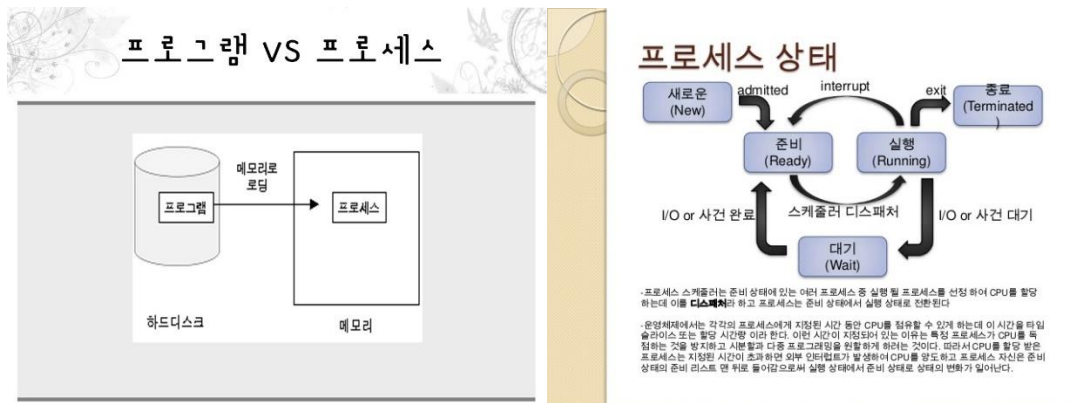
컴퓨터 시스템 하드웨어 및 소프트웨어 자원을 여러 사용자 간에 효율적 할당, 관리, 보호

* 자원관리

프로세스 – 메모리에 명령이 올라가 있는 것

프로그램(보조기억장치) 주기억장치(프로세스) 중앙처리 장치(CPU)

프로그램이 하드디스크에서 주기억장치(RAM)으로 가져오면 프로세스가 되는 것



Cpu는 한번에 한가지 일 밖에 처리하지 못한다.

Cpu가 일을 처리하기를 기다렸다가 순서에 맞게 보내주는 것이 운영체제가 하는 일

우리가 흔히 보는 2.7Ghz는 1초에 2.7기가의 계산을 한다는 것(2의 30승)

그런데 프로세스가 어떻게 이뤄질지 스케줄을 짜주는 방법들이 여러가지가 있다.

FCFS(first – come first-served)

준비 상태 큐에 도착한 순서에 따라 차례로 CPU를 할당

단점 : 더 빨리 처리할 수 있는 일을 처리 안하고 순서대로 처리한다.

SJF(Shortest Job First)

실행 시간이 가장 짧은 프로세스에 먼저 CPU 할당

평균 대기시간이 가장 적은 알고리즘

실행시간이 긴 프로세스에 밀려 무한 연기상태 발생가능

실행시간 계산 불가

Round Robin Scheduling

시분할 시스템을 위해 고안된 방식

FCFS 기법 변형

각 프로세스는 시간 할당량 동안만 실행

완료되지 않으면 다음 프로세스에게 CPU를 넘겨주고 준비상태 큐의 가장 뒤로 배치

할당된 시간이 클수록 FCFS와 비슷

할당시간이 작을수록 문맥교환과 오버헤드

(일처리 시간보다 스위칭 시간이 많이 발생하는 것)가 자주 발생

Priority Based Scheduling

프로세스마다 우선순위 부여

우선 순위가 동일한 경우 FCFS 기법으로 할당

가장 낮은 순위를 부여받은 프로세스의 무한 연기 발생가능

Multi Queue Scheduling

프로세스를 특정 그룹으로 분류할 수 있을 경우 그룹에 따라 각기 다른 준비단계 큐 사용

준비상태 큐 마다 다른 스케줄링 기법 사용가능

다른 준비상태 큐로 이동 불가

하위단계 준비 큐에 있는 프로세스를 실행하는 도중이라도 상위 단계 준비상태 큐에

프로세스가 들어오면 상위단계 프로세스에 CPU를 할당

주기억장치 관리

단순 관리

가상 메모리

보조기억장치를 주기억장치처럼 활용

(중요하지 않고, 느려도 되는 것들은 용량이 많은 보조기억장치에 할당하자)

우리가 작성하는 모든 코드는 운영체제에게 보내는 메시지,

커널

실질적으로 운영체제는 커널을 겉으로 둘러싸고 있는 것이라고 보면 편하다.

운영체제의 핵심

운영체제의 정체성

보안, 자원관리, 추상화

컴퓨터의 동작

사용자 -> 입력장치 -> os -> app -> 주기억장치 -> 연산장치 -> 제어장치 -> 출력장치 -> 사용자

상상해보자 - 노래를 틀고 싶다면?

사용자 - 입력장치 - 주기억장치 - 연산장치 - 주기억장치 - os - 보조기억장치 - app - os - 주기억장치 - 연산장치 - 주기억장치 - 제어장치 - 출력장치 -> 사용자

사용자 - 입력장치 - os - app - os - hdd - 메모리 - app - os - cpu - 출력장치

즉 os가 모든 것을 통제하고 중재하고 있다는 것

But 중요한 것은 과정을 계속 생각해보는 것!

사용자 : 연산명령을 입력장치를 통해 전달

입력장치 : 응용프로그램에게 사용자 입력 전달

응용프로그램 : 주기억장치로 명령복사

주기억장치 : CPU로 명령복사

CPU : 명령 수행 후 주기억장치로 결과 복사

OS: 응용프로그램에게 연산이 종료되었음을 알림

알고리즘

문제해결을 위한 절차/방법

어떠한 문제를 해결하기 위한 여러 동작들의 모음

Ex) 여행가방을 어떻게 싸지? 책가방에 책을 어떻게 넣지?

Ex) 테트리스 어떤 방식으로 넣어 놓을 것이냐

자료구조

자료를 효율적으로 이용할 수 있는 방법론

데이터를 구조적으로 표현하는 방식

Ex)블록 모양을 다 네모로 만들면 테트리스를 완벽하게 할 수 있음

Ex) 여행가방을 쌀 때 옷을 깔끔하게 잘 넣는 방법

자료구조의 예)

원시구조

- 정수, 실수, 문자 ...

선형구조

- 배열, 연결 리스트, 스택, 큐, 덱

비선형구조

- 트리, 그래프

물리적구조

정수, 실수, 문자

배열, 연결리스트

추상적구조

스택, 큐, 덱, 트리, 그래프

배열이란?

배열은 쉽게 말해 값을 저장해놓는 주차장을 만드는 것이다. 주차장의 주소만 알면 바로 그 값을 찾아낼 수 있다.

배열과 연결리스트의 차이

배열에 비해 연결리스트는 삭제와 삽입이 쉽다. 하지만 연결리스트가 가지는 단점은 값을 찾기 위해서 처음부터 끝까지 하나하나 찾아봐야 하는 구조이다. 즉 속도가 느리다. 또 용량이 더 커진다(주소를 찾기 위한). 또 일처리가 많다(삽입과 삭제를 위한).

그래서 나온 것이 이중연결리스트(양쪽의 주소를 알고 있는)

스택(Stack) – 어떠한 물건들이 쌓여 있는 모양

Ex) 인터넷 뒤로가기를 하면 순서대로 다시 나오는 것,

간단히 말해 마지막에 들어온 것이 먼저 나간다.

쌓는 것은 push, 나가는 것은 pop

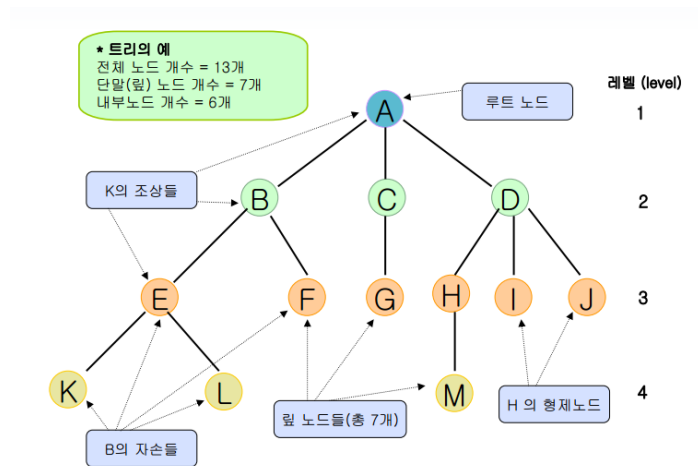
큐(Queue) 어디가서 줄 서 있으면 보통 큐라고 함. 맨 먼저 들어온 것이 맨 먼저 나가는 것

Ex) 작업 대기열에 많이 사용되고 있음

들어가는 것은 put, 꺼내 오는 것은 get

덱(Deque) 하나의 일렬데이터에서 앞에서도 꺼내고, 뒤에서도 꺼내는 것

트리(Tree)



장점 : 어떤 것을 찾아내는데 굉장히 용이한 구조

구조 - 부모 -> 자식 -> 자손처럼 가계도와 비슷한 구조

그래프(고등학교 수학에서 배우는 그래프)

대규모 관계를 찾기 위해서 필요함

'관계'를 찾는다는 것에 유의

**** 트리와 그래프의 차이에 유의하자**

알고리즘

일을 처리하는 순서/방법

대표적 알고리즘 - 정렬, 탐색, 재귀 등

정렬 알고리즘

선택 정렬 - 쪽 읽어보고 가장 작은거를 맨 앞으로, 쪽 읽어보고 다시 가장 작은거를 그 앞으로

버블 정렬 - 옆의 값과 비교를 해서 크면 뒤로 밀고, 아니면 옆으로 다시 가서 반복

삽입 정렬 - 계속 한 개씩 사이 사이에 넣는 과정을 만드는 것

병합 정렬 - 일정 구간을 나눠서 먼저 정렬하고, 일정 구간 나눠서 먼저 정렬하고,
계속해서 그것을 합치는 과정을 통해서 정렬

퀵 정렬 - 기준을 정해서 작으면 왼쪽 작으면 오른쪽으로 넘기는 방법을 계속해서 반복을
통해서 빠르게 정렬하는 것, **평균적으로 가장 빠른 정렬 방법**

* 퀵 정렬이 안 좋을 때 - 기준점을 잘못 잡았을 때, 자료의 크기가 비슷비슷할 때

알고리즘에 항상 정답은 없다. 따라서 여러가지 정렬 방법을 써야한다.

시간복잡도

알고리즘이 실행되는데 소요되는 시간분석

점근 표기법(대문자 O표기법) -> 최악의 경우를 표현한 것

정렬 알고리즘의 시간 복잡도

선택정렬 - $O(n^2)$

버블정렬 - $O(n^2)$

삽입정렬 - $O(n^2)$

병합정렬 - $O(n \log n)$

퀵정렬 - $O(n \log n)$

탐색 알고리즘의 시간 복잡도

선형탐색 - $O(n)$

이진 탐색 - $O(\log n)$