

## 운영체제

- 운영체제를 왜 배워야 할까

운영체제를 알아야 알맞게 프로그램을 만들 수 있다.

응용프로그램은 운영체제와 사용자 사이에 있기 때문이다.

예전에는 원하는 목적을 위해 매번 프로그램을 만들어야 했다. 다양한 목적으로 컴퓨터를 사용해야 할 필요성이 대두되었기 때문에 공통적으로 하드웨어를 제어하고 기본적인 기능을 제공하는 소프트웨어가 필요하게 되었다.

OS는 크게 세 분류 UNIX, Linux, Windows 이 분류는 커널의 차이인데 커널은 운영체제가 어떻게 동작하는지에 대한 뼈대이다.

- 시스템 하드웨어관리

사용자 프로그램의 오류나 잘못된 자원 사용을 감시, 입출력 장치 등의 자원에 대한 연산과 제어를 관리

- 가상시스템 서비스제공

사용자에게 컴퓨터 프로그램을 쉽고 효율적으로 실행할 수 있는 환경 제공

- 자원관리

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

컴퓨터 성능이 한정되어 있으므로 효율적 사용을 위해 관리

- 프로세스 : 프로그램이 메모리에 올라가있는 상태. 살아있는 프로그램

프로세스 상태 : 생성, 준비, 실행, 대기, 종료

대기 : 어떤 사건이 일어나기를 기다린다. CPU에 가기 전에 대기함.

준비 : 중앙처리장치에 의해 프로세스가 실행되기를 기다린다.

실행 : 중앙처리장치에 의해 프로세스가 실행된다.

생성 : 프로세스가 생성된다.

종료 : 프로세스가 종료된다.

- 프로세스 관리하는 몇 가지 방법
- FCFS(First-Come First-Served): 먼저 들어온 업무를 먼저 처리하는 방식. 단점은 앞에 실행시간 긴 놈이 다 잡아먹을 수 있다.
- SJF(Shortest Job First Scheduling): 실행시간이 짧은 프로세스에게 먼저 CPU할당. 실행시간이 긴 프로세스는 짧은 프로세스에 밀려 무한 연기상태 발생가능, 실행시간 예측이 어려움.
- Round Robin Scheduling: FCFS 기법 변형, 각 프로세스는 시간 할당량 동안만 실행, 완료되지 않으면 다음 프로세스에게 넘어가고 기존 프로세스는 준비상태 큐의 가장 뒤로 배치, 할당된 시간이 클수록 FCFS와 비슷하다, 할당시간이 작을수록 문맥 교환과 오버헤드가 자주 발생하지만 응답시간이 짧은 편이라 실시간 시스템에 유리.
- Priority Based Scheduling: 프로세스마다 우선순위 부여 우선순위가 동일한 경우 FCFS 기법으로 할당, 가장 낮은 순위 프로세스 무한 연기 발생가능
- Multi Queue Scheduling: 프로세스를 특정 그룹으로 분류할 수 있을 경우 각 그룹에 따라 각기 다른 준비단계 큐 사용, 준비상태 큐 마다 다른 스케줄링 기법 사용가능, 다른 준비상태 큐로 이동 불가

■ 주기억장치 관리: 프로그램을 만들 때 관리하는 코드를 작성하는 방법도 있음

단순 관리

가상메모리 – 보조기억장치를 주기억장치처럼 활용, 보조기억장치의 속도는 주기억장치보다 월등히 느리지만 중요도가 좀 낮은 경우 가상메모리에서 사용할 수 있게 함.

■ 파일관리.

파일을 입출력 할 때 보조기억장치에 있는 빈 공간을 찾거나, 이미 쓰여진 데이터를 운영체제가 찾아서 가져온다.

파일시스템이 시스템마다 다르기 때문에 한 운영체제에서 사용하던 스토리지를 다른 운영체제로 가져가면 사용할 수가 없다.

## 커널(kernel)

커널은 운영 체제의 핵심 부분으로서, 운영 체제의 다른 부분 및 응용 프로그램 수행에 필요한

여러 가지 서비스를 제공한다.

- 커널의 역할

보안: 컴퓨터 하드웨어와 프로세스의 보안을 책임진다.

자원관리: 한정된 시스템 자원을 효율적으로 관리하여 프로그램의 실행을 원활하게 한다.

추상화: 일반적으로 커널은 운영 체제의 복잡한 내부를 감추고 깔끔하고 일관성 있는 인터페이스를 하드웨어에 제공하기 위해 몇 가지 하드웨어 추상화(같은 종류의 장비에 대한 공통 명령어의 집합)들로 구현된다. 이 하드웨어 추상화는 프로그래머가 여러 장비에서 작동하는 프로그램을 개발하는 것을 돕는다.

## 컴퓨터의 동작

음악프로그램을 동작시킨다면 어떤 순서로 컴퓨터가 동작할까?

나의 예상 : 사용자 - 입력장치 - 보조기억장치로부터 프로그램을 주기억장치로 로딩 - 주기억장치에서 중앙처리장치로 이동 후 프로세서 실행(음악 재생프로그램 실행) - 음악 파일 로딩을 위해 메모리 요청 - 주기억장치에서 보조기억장치에 있는 음악 파일 로딩 - 중앙처리장치에서 음악 재생프로그램을 통해 음악파일 압축해제 - 스피커 출력

실제 : 사용자 입력 - OS가 입력을 받아 app에게 알려줌 - 음악파일을 불러와야겠다!! - app이 OS에게 특정 경로의 음악파일을 요청함 - 가져와서 메모리에 로딩 - app에게 음악파일 로딩을 알림 - app이 OS에게 스피커에 음악파일 출력을 요청 - 음악 출력

실험: 중앙처리장치 , 주기억장치 , 보조기억장치, 운영체제, 입출력장치, 사용자, 어플리케이션 의 역할을 한명 씩 맡아 정보의 이동과정을 role play 해보았다. 왜 CPU의 성능이 좋아야 하는지 알 수있었다....

## 알고리즘

문제 해결을 위한 절차/방법

어떠한 문제를 해결하기 위한 여러 동작들의 모음

Ex)테트리스 : 테트리스를 어떻게 쌓아야 빈 공간이 가장 적게 쌓을 수 있을까?

Ex)여행가방을 꾸리려면? 리스트작성부터, 속옷부터 넣는다, 큰 물건부터 넣는다, 잘 안 꺼내는

물건부터 넣는다. 각각 다른 방법으로 문제를 해결하려 함

- 대표적 정렬 알고리즘

선택정렬 : 전체 스캔 후 가장 작은 것을 맨 앞에 놓는 작업을 반복함.

버블정렬 : 근접한 두 값씩 비교하여 순차적 정렬함.

삽입정렬 : 하나를 선택해서 그것보다 작은 것과 그것보다 큰 것 사이에 집어넣음

병합정렬 : 구간 나누어놓고 합친다. 구간사이에서 어떤 정렬을 쓰느냐에 따라 여러 차이가있다.

퀵정렬 : 평균 값을 기준 값으로 지정하고 기준보다 작은 값과 큰 값을 나누고 작은 값들의 모임과 큰 값들의 모임에서 같은 작업을 반복한다. - 평균적으로 가장 좋은 성능을 보인다.

- 시간복잡도

알고리즘이 실행되는데 소요되는 시간 분석 : 1번의 실행이 되는데  $x$ 의 시간이 걸린다고 생각할 수 있으므로 최소 실행이 될수록 시간이 적게 걸리므로 좋다. 시간복잡도 그래프를 그려보면 데이터 개수  $n$ 을  $x$ 축으로 최대 실행횟수를  $y$ 축으로 놓는다.

점근 표기법(대문자 O표기법) : 입력 값에 따라 출력 값이 얼마나 커지는가를 표현한다. - 강사님 설명은 최악의 경우를 생각한다고 함.

선택, 버블, 삽입정렬  $O(n^2)$

병합정렬  $O(n \log n)$

퀵정렬  $O(n \log n)$

선형 탐색(배열)  $O(n)$

이진 탐색  $O(\log n)$

## 자료구조

자료를 효율적으로 이용할 수 있는 방법론

데이터를 구조적으로 표현하는 방식

- 이해를 위해

Ex)테트리스 - ㄱ 자 모양이었던 블록을 쌓았을 때 생기는 문제점을 ㄴ 자 모양 블록이라면 문제없이 차곡차곡 쌓을 수 있다.

Ex)여행가방을 꾸릴 때 - 물건을 넣는 순서보다 물건을 어떻게 넣어야 가방을 잘 활용할지 생각한다.

- 원시구조

정수, 실수, 문자

메모리에 숫자 하나, 문자 하나 달랑 있는 것.

- 선형구조

배열 : 데이터가 근접하여 저장되어 있으므로 자료를 찾기가 쉽다.

연결리스트 : 1번에게 2번주소가 있고 2번에게 3번주소가있는 점 조직 같은 구조. 데이터를 찾는데 오래 걸린다. 다음 데이터 주소가 저장되어야 하므로 저장공간 낭비가 있다.

스택 : First in last out. 이것은 물리적 적재와 비슷하다. 책이 쌓여있을 때 아래 있는 책을 먼저 뺄 수 없는 것처럼. 인터넷을 이용할 때 뒤로가기 버튼을 통해 우리는 스택을 이용한다.

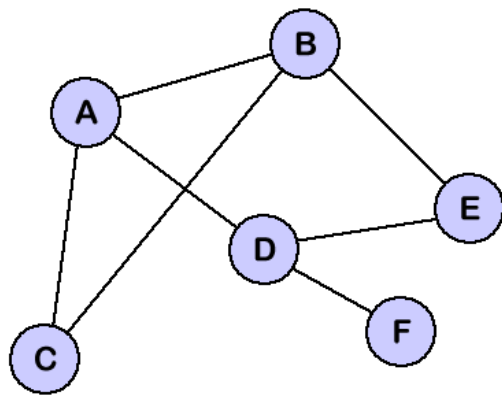
큐 : 맨 먼저 들어온 것이 맨 먼저 빠져나가는 형태. 작업 대기 열에 많이 사용된다.

덱 : 스택 + 큐. 앞에서도 꺼내고 뒤에서도 꺼냄. 중간은 못 꺼내는 형식. Double ended queue

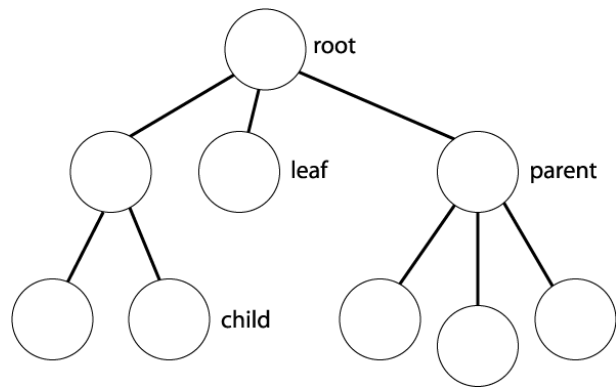
- 비선형구조

트리 : 정보를 찾을 때 유용한 구조. 출발점이 있으며 부모와 자식이 연결되어있는 모양이다. 위계가 있다.

그래프 : 관계를 표현할 때 많이 사용하는 구조. 복잡한 일 처리를 표현할 때



(트리)



(그래프)

나중에 알고리즘 책을 펴보면 무시무시한 일이 일어난다고 한다.

마크다운을 쓰는 이유

이것은 표준이다. 어떤 시스템이든 어떤 환경이든 쓰일 수 있는 것이 표준이다. 표준을 지키지 않는 것은 기본을 지키지 않는 것. 덧붙여 PDF도 표준.

## HOMEWORK

ActiveX가 무엇일까?

국내 웹에 ActiveX가 정착된 이유 및 배경

기술적 부채(Technical Debt)에 대해 알아보기

위 내용을 조사하며 느낀 점