컴퓨터 구조를 알아야하는 이유

컴퓨터의 근간을 알아야한다

컴퓨터의 근간을 알게되면 문제해결능력과 성능 용량 비용에 대해 알 수 있게 된다

미지의 대상에서 분석의 대상으로 볼 수 있게 됨

컴퓨터가 이해하는 정보 - > 데이터와 명령어

컴퓨터의 네가지 핵심 부품 cpu 메모리(RAM)주기억장치 보조기억장치 입출력 장치

메모리는 현재 실행되는 프로그램 명령이나 데이터를 저장하는 부품

프로그램이 실행되려면 메모리에 저장되어 있어야한다

메모리에 저장된 값의 위치는 주소로 알 수 있다

Cpu는 메모리의 저장된 명령어를 읽어드리고 해석하고 실행하는 부품

CPU 내부구성 ALU,제어장치,레지스터

ALU : 계산기

레지스터 : CPU 내부의 작은 저장장치

제어장치 : 제어 신호를 내보내고 명령어를 해석하는 장치

운영체제란 ? 자원을 어떻게 효율적으로 관리할까 ? !!

자원을 관리하는 특별한 프로그램

실행 중인 프로그램(프로세스)을 관리하는 특별한 프로그램

동시에 실행되는 프로세스들이 문제가 없게끔 관리해주는 것

페이징과 스와핑의 개념 ?

동시에 실행 => 우리가 눈치채지 못할만큼 빠르게 번갈아가면서 실행되는 중

수많은 프로세스들이 자원에 마구 직접 접근해도 ㄷ괜찮을까 ?

A : 시스템 호출을 통해서 자원을 보호할 수 있음

동시성 – 한 순간에 여러가지 일이 아니라 짧은 전환으로 여러가지 일을 동시에 처리되는것처럼 보이는 것

메모리의 4가지 영역

Code 프로그램 코드

Data Static 변수 혹은 global 변수

Heap 동적 메모리 영역

Stack 지역변수,매개변수 등등 일시적인 데이터

쓰레드는 코드 데이터 힙영역을 공통된 자원으로 사용하고

각 쓰레드는 스택만을 따로 가지고 있음

공유된 자원이 있기 때문에 컨택스트 스위칭이 일어날 때 캐시 적중률이 올라감

멀티프로세스 멀티 스레드 란 처리방식의 일종

멀티프로세스는 각 프로세스는 독립적

Ipc를 이용한 통신이 필요함

자원 소모적 개별 메모리 차지

컨텍스트 스위칭 비용이 큼

동기화작업이 필요하지 않음

멀티 쓰레드

쓰레드끼리 긴밀하게 연결됨

공유 된 자원으로 통신 비용 절감

공유된 자원으로 메모리가 효율적

컨텍스트 스위칭 비용이 적음

공유자원 관리를 해야함

멀티프로세스를 사용하는 이유 :

멀티쓰레드는 긴밀하게 연결되어 있기 때문에 한 쓰레드에 문제가 생기면 전체 프로세스의 영향이 감

멀티코어는 하드웨어 측면에 가까움

동시성은 짧은 순간에 cpu의 시간을 분할해서 동시에 하는것처럼 보이게 하는 것

멀티코어는 병렬처리이며 물리적으로 여러 코어를 사용해서 다수의 실행단위를 한순간에 처리할 수 있게 해주는 것

리눅스 커널에서는 프로세스와 스레드를 동일하게 봄

스레드는 한 프로세스 내에서 나뉘어진 하나이상의 실행단위이다

컨텍스트 스위칭은 다음 상황에서 일어남

I/O interrupt

Cpu 사용시간 만료

자신 프로세스 fork

인터럽트 처리를 기다릴때

어떤 하나의 프로세스가 실행되고 있는 상태에서 인터럽트 요청에 의해 다음 프로세스가 실행되어야할 때 기존의 프로세스의 상태를 저장하고 cpu가 다음 프로세스를 수행하도록 새로운 프로세스 상태로 교체하는 작업

Context란 cpu가 해당 프로세스를 싱행하기 위한 해당 프로세스 정보들

이 context는 pcb에 저장됨

Pcb의 저장 정보

프로세스상태(생성,준비,수행,대기,중지)

PC

레지스터

프로세스번호

페이징기법 : 고정크기로 나누는 것 내부단편화가 있음 외부단편화 x

세그멘테이션기법 : 가변크기로 나누는 것 내부단편화가 없음 , 외부단편화가 있음