NHN 정리

1. 자료구조 & 알고리즘
   1. 리스트

배열

연결리스트

단일연결

원형연결

이중연결

* 1. 스택 & 큐

스택 :

큐 :

* 1. 트리

??

이진트리

완전이진트리

B+tree

* 1. 그래프
     1. DFS & BFS

1. 프로그래밍 언어
   1. JAVA
   2. C
   3. Python
2. 운영체제
   1. 페이징 알고리즘

보조기억장치를 이용하여 가상 메모리를 같은 크기의 블록으로 나눈것

페이징 기법을 사용하면 연속적이지 않은 공간을 활용할 수 있으므로 메모리 외부 단편화 문제를 해결 할 수 있다.

내부 단편화는 여전히 생길 수 있다.

OPT(최적 교체)

앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법

각 페이지의 호출 순서와 참조 상황을 미리 예측해야 하므로 실현 가능성이 희박

FIFO(First in First Out)

각페이지가 주 기억장치에 적재될 때 마다 시간을 기억시켜 가장 ㅓㄴ저 들어와서 가장 오래 있던 페이지를 교체하는 기법

이해하기 쉽고 프로그래밍 및 설계가 간단하며 벨레이디의 모순현상이 발생

벨레이디 모순현상 : FIFO알고리즘에서 기존 페이지 프레임의 개수를 늘리면 Page Fault 발생이 감소해야하나 오히려 증가하는 현상

LRU (Least Recently Used) 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법

각페이지마다 계수기나 스택을 두어 현시점에서 가장 오랫동안 사용하지 않은 페이지를 교체함

LFU(Least Frequently Used)

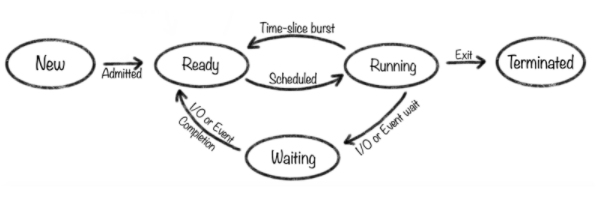
사용빈도가 가장 적은 페이지를 교체하는 기법

단점 : 프로그램 실행 초기에 많이 사용된 페이지가 그 후로 사용되지 않을 경우에도 프레임을 계속 차지하는 단점이 존재

NUR(Not Used Recently)

최근에 사용하지 않은 페이지를 교체하는 기법

SCR(Second Chance ReplaceMent) 가장 오랫동안 주 기억장치에 있던 페이지 중 자주 사용되는 페이지의 교체를 방지하기 위한 기법



스케줄링 : 프로세스가 생성되어 실행될 때 필요한 자원을 해당 프로세스에 할당하는 작업을 말합니다.

성능 측정지표 : 평균실행시간 평균대기시간 평균 반환시간

* 1. 선점 스케줄링

현 프로세스로부터 CPU를 회수하여 다른 프로세스에게 할당하는 행위

라운드로빈 스케줄링 : 준비큐를 원형큐로 간주하고 순환식으로 간주하고 각 프로세스에게 작은 단위의 시간량만큼 CPU를 할당

이건 동작과정 봐보자

다단계 큐 스케줄링

목적에 맞도록 우선순위들을 설정하고 그 우선순위마다 준비큐를 따로 설정

다단계 피드백 큐 : 다단계 큐에 동적인 프로세스 우선순위 변화를 적용한다.

이것도 동작과정 보자

* 1. 비선점 스케줄링

이미 할당된 자원을 다른 프로세스가 강탈 할 수 없음

응답시간의 예측이 편하며 일괄처리 방식에 적합

단점 : 덜 중요한 자원을 할등 받으면 중요한 작업이 와도 먼저 처리 될 수 없음

FCFS SJF HRN 우선순위 기한부

FCFS (선입선출)

먼저 들어온 것이 먼저 처리되는 FIFO 구조의 알고리즘

단점으로는 waiting time 이 너무 김

SJF 대기하는 작업중 실행시간이 가장 짧은 작업을 우선하는 하는 알고리즘

평균 대기 시간에 있어서는 최적이 된다.

도착시점을 고려하면 아직 도착하지 않은 프로세스는 스케줄링의 대상이 되지 않기 때문에 더 긴 프로세스가 먼저 할당 될 수 있다.

BURST TIME 이 같을경우 FCFS와 같음

HRN : 어떤 작업이 서비스 받을 시간과 그 작업이 서비스를 기다린 시간으로 결정

우선순위 대기식 = (대기시간 + 서비스 시간) / 서비스 시간

우선순위 : 준비상태 큐에서 대기하는 프로세스에게 부여된 우선순위가 가장 높은 프로세스에게 먼저 CPU를 할당하는 방식

단점 : 기아상태를 유발될 수 있음 - > 우선순위가 높은 작업이 계속적으로 들어올 경우 우선순위가 낮은 작업이 준비 상태에서 보장없이 머물 수 있다. 이런 기아상태는 Aging으로 해결될 수 있다. 시스템에 머무는 시간이 증가하면서 우선순위를 높여주는 것이다. 아무리 우선순위가 낮었던 프로세스라도 시간이 지나면서 우선순위가 높아져 CPU를 할당받을 수 있게 된다.

BigEndian 방식 :

낮은 주소에 데이터의 높은 바이트부터 저장하는 방식

메모리에 저장된 순서 그대로 읽을 수 있음

Little endian 방식 : 높은 주소의 데이터에서 낮은 주소로 저장하는 방식

1. 네트워크

서브넷 네트워크 : 하나의 큰 네트워크를 논리적으로 분리하여 만든 부분 네트워크를 의미

11111111 11111111 11111111 ~ (255.255.255.~) 전부 연속적으로 이루어진 것들

서브네팅을 이용하는 이유는 ip낭비를 방지하고

지역적인 한계성이 있을경우

네트워크 환경을 구성하려는 물리적인 하부 구조에 제한있을경우

케이블 길이 등의 제한

특정 지역에 한정된 트래픽으로 제한

보안상의 이유로 서로 공유할수 없을 경우

OSI 7 Layer

계층을 나눈 이유는 통신이 일어나는 과정이 단계별로 파악가능

1 . 물리계층 (Physical Layer)

통신 케이블로 데이터를 전송 이계층은 데이터를 전달만 할 뿐 전송하려는 데이터가 무엇인지 어떤 에러가 있는지 등에는 신경 쓰지 않는다.

케이블 , 리피터 , 허브를 통해 전송

2. 데이터 링크계층 (DataLink )

포인트 투 포인트 간 신뢰성 있는 전송을 보장하기 위한 계층 CRC 기반의 오류제어와 흐름제어가 필요 물리 계층에서 발생할 수 있는 오류를 찾아내고 수정한는데 필요한 기능적 절차적 수단을 제공

주소값은 물리적으로 할당 받는데 네트워크 카드가 만들어질 때부터 맥주소가 정해져있다는 뜻 주소 체계는 계층이 없는 단일 구조인데 데이터 링크 계층의 가장 잘 알려진 예는 이더넷이다. 이 외에도 HDLC나 ADCCP같은 포인트 투 포인트 프로토콜이나 패킷 스위칭 네트워큰나 LLC ,ALOHA 같은 근거리 네트워크용 프로토콜

프레임에 주소부여 , 에러검출 , 제전송 흐름제어

3. 네트워크 계층

데이터를 목적지까지 가장 안전하고 빠르게 전달한느 기능

대표장비는 라우터 요즘 스위치에 라우팅 기능을 장착한 layer 3 스위치도 있다.

여러 개의 노드를 거칠때마다 경로를 찾아주는 역할 다양한 길이의 네트워크들을 통해 전달하고 그 과정에서 전송 계층이 요구하는 서비스 품질을 제공하기 위한 기능적 절차적 수단을 제공한다.

네트워크 계층은 라우팅 흐름제어 세그멘테이션 오류제어 인터네트워킹등을 수행

데이터를 연결하여 다른 네트워크를 통해 전달함으로서 인터넷이 가능하게 만드는 계층

논리적인 주소구조(ip) 네트워크 관리자가 직접 주소를 할당하는 구조를 가지며 계층적이다.

IP계층 TCP/IP 상에서 IP게층은 네트워크의 주소를 정의하고 IP 패킷의 전달 및 라우팅을 담당하는 계층

OSI 7 Layer계층모델의 관점에서 보면 IP계층은 네트워크 계층에 해당

즉 패킷을 목적지까지 전달하는 역할 및 그에 수반되는 기타 역할을 함

데이터링크 계층의 하드웨어적인 특성에 관계없이 독릭적인 역할을 수행

패킷의 전달을 책임지는 IP

패킷 전달 에러의 보고 및 진단을 위한 ICMP

복잡한 네터워크에서 인터 네트워킹을 위한 경로를 찾게해주는 라우팅 프로토콜

IP 프로토콜 : TCP/IP 기반의 인터넷 망을 통하여 데이터그램 전달을 담당하는 프로토콜

주요기능 : IP계층에서 IP 패킷의 라우팅 대상이 됨 IP 주소 지정

신뢰성 및 흐름제어 기능이 전혀없다 . IP계층위의 TCP와 같은 상위 트랜스포트 계층에 의존 비연결성 데이터그램 방식으로 전달되는 프로토콜

패킷의 완전한 전달(소실 , 중 복 , 지연,순서바뀜등이 없게함을 보장하지 않음

IP패킷 헤더 내 바이트 전달 순서 : 최상위 바이트 먼저보냄

경우에 따라 단편화가 필요

TCP UDP ICMP IGMP등이 IP 데이터그램에 실려서 전송

1. 데이터베이스