\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

130주차 - 배치 프로그램

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 배치(Batch) 프로그램의 개요

1) 사용자의 상호 작용 없이 여러 작업들을 미리 정해진 일련의 순서에 따라 처리하는 것

2) 자동으로 수행되는 주기에 따른 분류

- 정기 배치 : 정해진 기간에 정기적으로 수행

- 이벤트성 배치 : 특정 조건을 충족할 때 수행

- On-Demand 배치 : 사용자 요칭 시 수행

3) 배치 프로그램 필수 요소

- 대용량 데이터 : 대량의 데이터를 처리가 가능해야 됨

- 자동화 : 심각한 오류 제외하고 사용자 개입 없이 수행

- 견고성 : 잘못된 데이터나 중복 등 상황으로 중단되는 일 없이 수행

- 안정성/신뢰성 : 오류 발생 시 위치, 시간 등 추적 가능해야 됨

- 성능 : 다른 프로그램 방해 하지말고 지정 시간 내에 처리 가능해야 됨

2. 배치 스케줄러(=잡 스케줄러)

1) 일괄 처리(Batch Processing) 작업이 설정된 주기에 맞춰 자동으로 수행시켜주는 도구

2) 종류

- 스프링 배치

- 스프링 프레임워크, 오픈 소스

- 데이터베이스나 파일 데이터 교환하는데 필요한 컴포넌트 제공

- 로그 관리, 추적, 트랜잭션 관리, 작업 처리 통계, 작업 재시작 등의 기능

- 구성 요소

- Job : 수행할 작업 정의

- Job Launcher : 실행을 위한 인터페이스

- Step : Job 처리를 위한 제어 정보

- Job Repository : 작업 실행을 위한 모든 정보 저장

- Quartz

- 스프링 프레임워크, 오픈 소스

- 수행할 작업과 수행 시간을 관리하는 요소들을 분리하여 일괄 처리 작업에 유연성 제공

- 구성 요소

- Scheduler : 실행 환경 관리

- Job : 수행할 작업 정의

- JobDetail : Job의 상세정보

- Trigger : Job의 실행 스케줄 정의

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

144주차 - 스크립트 언어

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 스크립트 언어의 개요

1) HTML 문서 안에 직접 프로그래밍 언어 삽입, 기계어로 컴파일 되지 않고 별도의 번역기가 소스 분석, 동작

2) 주로 데이터베이스 처리 작업

3) 클라이언트 언어

- 클라이언트의 웹 브라우저에서 해석되어 실행

- 자바스크립트, VB 스크립트

서버 언어

- 서버에서 해석되어 실행된 후 결과만 클라이언트로 보냄

- ASP, JSP, PHP, 파이썬

4) 컴파일 없이 실행되므로 결과 바로 확인, 코드 읽고 해석해야 하므로 실행속도 느림

5) 런타임 오류 많이 생김

2. 스크립트 언어 종류

1) 자바스크립트

- 웹 페이지 동작제어

- 클래스 기반 객체 상속 지원, 객체지향 프로그래밍 언어 성격도 가짐

- Prototype Link와 Prototype Object를 통해 프로토타입 개념 활용

- Prototype Object : 객체가 생성될 때 그것의 원형

- Prototype Link : 객체와 원형을 연결하는 링크

2) VB스크립트(Visual Basic)

- 마이크로소프트 사에서 자바스크립트에 대응하기 위해 제작

- Active X를 사용하여 마이크로소프트 사의 앱을 컨트롤 할 수 있다.

- Active X : 마이크로소프트 사에서 윈도우 환경의 응용 프로그램을 웹과 연결함, 동적인 콘텐츠 제작 편리

3) ASP(Active Server Page)

- 서버 측에서 동적으로 수행되는 페이지를 만들기 위한 언어

- 마이크로소프트에서 만듦

- 윈도우에서만 사용 가능

4) JSP(Java Server Page)

- 자바로 만들어진 서버용 스크립트, 다양한 운영체제에서 사용 가능

5) PHP(Professional Hypertext Preprocessor)

- 서버용 스크립트 언어, Linux, Unix, Windows에서 사용 가능

- C, JAVA와 문법 유사, 웹 페이지 제작

6) 파이썬

- 귀도 반 로섬이 발표한 대화형 인터프리터 언어

- 객체지향 기능 지원, 플랫폼에 독립적, 문법 쉬움

7) 쉘 스크립트

- 유닉스/리눅스 계열의 쉘에서 사용되는 명령어들의 조합

- 컴파일 단계가 없어 실행 속도 빠름

- 저장 시 .sh가 붙음

- 쉘 종류 : Bash Shell, Bourn Shell, C Shell, Korn Shell

- 쉘 스크립트에서 사용하는 제어문

- 선택형 : if, case

- 반복형 : for, while, until

8) Basic

- 절차지향 기능 지원하는 대화형 인터프리터 언어, 쉬움

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

146주차 - 라이브러리

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 라이브러리의 개념

1) 자주 사용하는 함수나 데이터를 미리 만들어 놓은 집합체

2) 필요할 때 언제든지 호출하여 사용

3) 라이브러리는 모듈과 패키지 모두를 의미한다.

- 모듈 : 하나의 기능이 한 개의 파일로 구현된 형태

- 패키지 : 하나의 패키지 폴더 안에 여러 개의 모듈을 모아 놓은 형태

4) 표준과 외부가 있다.

- 표준 라이브러리 : 프로그래밍 언어에 기본적으로 포함되어 있는 라이브러리

- 외부 라이브러리 : 개발자들이 필요한 기능 만들어 인터넷에 공유해 놓은 것

2. C언어 대표 표준 라이브러리

1) stdio.h : 데이터 입출력 주요함수 : printf,scanf,fprintf,fscanf,fclose,fopen

2) math.h : 수학 함수 주요함수 : sqrt,pow,abs

3) string.h : 문자열 처리 주요함수 : strlen,strcpy,strcmp

4) stdlib.h : 자료형 변환,난수,메모리할당 주요함수 : atoi,atof,srand,rand,malloc,free

5) time.h : 시간 처리 주요함수 : time,clock

3. JAVA언어 대표 표준 라이브러리

1) java.lang :

- 기본적으로 필요한 인터페이스,자료형,예외처리

- import 문 없이 사용 가능

- 주요 클래스 : String,System,Process,Runtime,Math,Error

2) java.util

- 날짜 처리, 난수 발생, 복잡한 문자열 처리

- 주요 클래스 : Date,Calender,Random,StringTokenizer

3) java.io

- 파일 입출력

- 주요 클래스 : InputStream,OutputStream,Reader,Writer

4) java.net

- 네트워크

- 주요 클래스 : Socket,URL,InetAddress

5) java.awt

- 사용자 인터페이스

- 주요 클래스 : Frame,Panel,Dialog,Button,Checkbox

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

149주차 - 운영체제의 개념

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 운영체제의 정의

1) 시스템 자원을 효율적으로 관리, 사용자가 컴퓨터를 편하게 사용하도록 환경 제공

2) 사용자 -> 응용 프로그램 -> 유틸리티 -> 운영체제 -> 하드웨어

2. 운영체제 목적

1) 처리 능력 : 일정 시간 내에 시스템이 처리한느 일의 양

2) 반환 시간 : 시스템에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간

3) 사용 가능도 : 시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도

4) 신뢰도 : 시스템이 주어진 문제를 정확하게 해결하는 정도

3. 운영체제 구성

1) 제어 프로그램(깜짝데이트)

- 감시, 순서 지정, 데이터 관리 수행

- 감시 프로그램 : 가장 핵심적인 역할, 자원의 할당 및 전체 작동 감시

- 작업 관리 프로그램 : 작업의 순서와 방법 관리

- 데이터 관리 프로그램 : 사용되는 데이터 파일의 표준적인 처리 및 전송 관리

2) 처리 프로그램

- 제어 프로그램의 지시를 받아 사용자가 요구한 문제를 해결

- 언어 번역 프로그램 : 고급언어를 기계어 형태의 목적 프로그램으로 변환, 컴파일러,어셈블러,인터프리터

- 서비스 프로그램 : 효율적으로 컴퓨터 사용하게 해줌, 분류/병합, 유틸리티, 프로그램

- 유틸리티 : 기존 프로그램을 지원하거나 기능 향상 시키는 프로그램, 디스크 관리, 파일 백업

4. 운영체제 기능

1) 프로세스,기억장치,입출력장치,파일 및 정보 등의 자원 관리

2) 자원 스케줄링 기능 제공

3) 오류 검사 복구

4) 가상 계산기 기능 지원(한대의 컴퓨터를 여러대컴퓨터처럼 보이게 함)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

151주차 - UNIX/LINUX/MacOS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. UNIX의 개요 및 특징

1) 벨 연구소, MIT에서 개발

2) 시분할 시스템을 위해 설계된 대화식 운영체제, 오픈소스인 개방형 시스템

3) 대부분 C언어라 이식성 높고 프로세스 간 호환성 높음

4) 크기가 작고 이해하기 쉽다.

5) 다중사용자,다중작업 지원

6) 많은 네트워킹 기능 제공, 통신망 관리용으로 적합

7) 트리 구조의 파일 시스템 가짐

8) 전문적인 프로그램 개발에 좋음

9) 다양한 유틸리티 프로그램 존재

2. 유닉스 시스템 구성

1) 하드웨어

2) 커널

- 가장 핵심적인 부분

- 부팅될 때 주기억장치에 적재된 후 상주하면서 실행

- 하드웨어 보호, 프로그램과 하드웨어 간 인터페이스 역할

- 프로레스 관리, 기억장치 관리, 파일 관리, 입출력 관리, 프로세스 간 통신, 데이터 전송 및 변환

3) 쉘

- 사용자의 명령어를 인식해 프로그램을 호출하고 명령 수행하는 명령어 해석기

- 시스템과 사용자 간 인터페이스 역할

- DOS의 COMMAND.COM과 같은 기능 수행

- 주기억장치에 상주하지 않고, 명령어가 포함된 파일 형태로 존재

- 보조기억장치에서 교체 처리 가능

- 파이프라인 기능 지원, 입출력 재지정을 통해 입출력 방향 변경 가능

- 파이프라인 : 둘이상 명령을 묶어 처리한 결과를 다른 명령의 입력으로 전환

4) 유틸리티

- 일반 사용자가 작성한 응용 프로그램 처리에 사용됨

- DOS에서 외부 명령어에 해당된다.

- 에디터,컴파일러,인터프리터,디버거

5) 사용자

★ UNIX에서 프로세스 간 통신(IPC : Inter Process Communication)(프로그램 사이 데이터 주고받는것)

1) 각 프로세스는 시스템 호출을 통해 커널 기능 사용

2) 종류

- 시그널 : 간단한 메시지를 이용한 통신, 초기 UNIX시스템에서 사용됨

- 파이프 : 한 프로세스의 출력이 다른 프로세스의 입력으로 사용되는 단방향 통신 방식

- 소켓 : 프로세스 사이 대화를 가능하게 하는 쌍방향 통신 방식

3. LINUX의 개요 및 특징

1) 리눅스 토발즈가 UNIX 기반으로 개발

2) UNIX와 완벽하게 호환

4. MacOS의 개요 및 특징

1) 애플 사가 UNIX 기반을 개발

★ 파일 디스크립터(파일 서술자)

1) 파일에 대한 정보를 가진 제어 블록

2) 파일 제어 블록(FCB)로 불린다.

3) 파일마다 독립적으로 존재, 시스템에 따라 다른 구조를 가질 수 있다.

4) 보조기억장치에 있다가 파일 오픈될 때 주기억장치로 옮겨짐

5) 파일 시스템이 관리하므로 사용자가 직접 참조 불가

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

152주차 - 기억장치 관리의 개요

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 기억장치 계층 구조의 특징

1) 레지스터 <-> 캐시 기억장치 <-> 주기억장치 <-> 보조기억장치

<- 접근 속도, 비트당 가격 증가

<- 접근 시간, 기억 용량 감소

2) 상위 쪽(레지스터 쪽)으로 갈수록 접근 속도와 접근 시간이 빨라짐, 기억용량이 적고 고가

3) 주기억장치는 자신의 주소를 갖는 워드 또는 바이트로 구성되어 있으며, 주소를 이용해 접근 가능

4) 레지스터,캐시기억장치,주기억장치는 CPU가 직접 액세스 가능, 보조기억장치는 직접 액세스 불가

5) 보조기억장치에 있는 데이터는 주기억장치에 적재된 후 CPU에 의해 액세스 가능

2. 반입(Fetch) 전략

1) 보조기억장치에 보관중인 ㅍ로그램이나 데이터를 언제 주기억장치로 적재할 것인지

- 요구 반입(Demand Fetch) : 실행중인 프로그램이 특정 데이터 등의 참조를 요구할 때 적재하는 방법

- 예상 반입(Anticipatory Fetch) : 실행중인 프로그램이 특정 데이터 등의 참조를 예상해 미리 적재하는 방법

3. 배치(Placement) 전략

1) 새로 반입되는 프로그램이나 데이터를 주기억장치 어디에 위치시킬 것인지

- 최초 적합(First Fit) : 빈 영역 중 첫 번째 분할 영역에 배치

- 최적 적합(Best Fit) : 빈 영역 중 단편화를 가장 작게 남기는 영역에 배치

- 최악 적합(Worst Fit) : 빈 영역 중 단편화를 가장 크게 남기는 영역에 배치

- 단편화 : 분할된 영역이 데이터보다 크거나 작아서 생기는 빈 기억 공간

- 내부 단편화 : 영역이 커서 남음

- 외부 단편화 : 영역이 작아서 할당될 수 없다, 분할된 전체 영역

4. 교체(Replacement) 전략

1) 주기억장치의 모든 영역이 사용중인 상태에서 새로 배치하려 할 때, 어느 영역을 교체하여 사용할 것인지

2) 교체 전략 종류 : FIFO,OPT,LRU,LFU,NUR,SCR

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

154주차 - 가상기억장치 구현 기법/페이지 교체 알고리즘

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 가상기억장치의 개요

1) 보조기억장치의 일부를 주기억장치처럼 사용하는 것, 용량이 작은 주기억장치를 마치 큰 용량을 가진 것처럼 사용

2) 프로그램을 여러 작은 블록단위로 나눠 가상기억장치에 보관 후 실행 시 요구되는 블록만 주기억장치에 불연속적으로 할당

3) 주기억장치 용량보다 큰 프로그램 실행하기 위해 사용

4) 가상기억장치의 주소를 주기억장치의 주소로 바꾸는 주소변환작업(사상,Mapping) 필요

5) 페이징 기법, 세그먼테이션 기법

2. 페이징 기법

1) 가상기억장치에 보관된 프로그램과 주기억장치의 영역을 동일한 크기로 나눈 후 나눠진 프로그램(페이지)을 동일하게

나눠진 주기억장치의 영역(페이지 프레임)에 적재시켜 실행

- 페이지 : 프로그램을 일정한 크기로 나눈 단위, 일반적으로 1~4KB

- 페이지 프레임 : 페이지 크기로 일정하게 나눠진 주기억장치의 단위

2) 외부 단편화는 발생하지 않음, 내부 단편화는 발생 가능

3) 주소 변환을 위해 페이지 위치정보를 가진 페이지 맵 테이블 필요

4) 페이지 맵 테이블 사용으로 비용이 증가되고, 처리 속도가 감소된다.

3. 세그먼테이션 기법

1) 세그먼테이션 개요

- 가상기억장치에 보관된 프로그램을 다양한 크기의 논리적 단위로 나눈 후 주기억장치에 적재시켜 실행

- 세그먼트 : 배열, 함수 등 논리적 크기로 나눈 단위, 각 세그먼트는 고유한 이름,크기를 가짐

- 기억장치의 사용자 관점을 보관하는 기억장치 관리 기법

- 기억공간을 절약하기 위해 사용

- 주소 변환을 위해 세그먼트 위치정보를 가진 세그먼트 맵 테이블 필요

- 세그먼트가 주기억장치에 할당될 때 다른 영역을 침범할 수 없고, 방지하기 위해 기억장치 보호키가 필요함

- 내부 단편화는 발생하지 않음, 외부 단편화는 발생 가능

2) 세그먼테이션 기법의 일반적인 주소 변환

- 주소 형식에 따른 주소와 세그먼트 맵 테이블의 구성

- 가상주소 형식 = 세그먼트 번호 | 변위값

- 실기억주소 형식 = 실기억주소(세그먼트 기준번지 + 변위값)

- 주소 변환 순서

- 세그먼트 맵 테이블 = 세그먼트 번호 | 세그먼트 크기 | 기준번지

1) 가상주소의 세그먼트 번호로 세그먼트 맵 테이블에서 해당 세그먼트의 기준번지와 세그먼트 크기 구함

2) 가상주소의 변위값과 세그먼트의 크기를 비교

3) 변위값이 작거나 같으면 기준번지와 변위값을 더하여 실기억주소를 만들어 주기억장치를 액세스

4) 변위값이 크면 다른 영역을 침범하게 되므로 실행 권한을 운영체제에게 넘기고 트랩 발생

4. 페이지 교체 알고리즘

1) 페이지 부재가 발생했을 때 가상기억장치의 필요한 페이지를 주기억장치에 적재해야 됨, 이 때 주기억장치 모든 프레임이

사용 중일 때 어떤 프레임을 교체할 지 결정

2) OPT(OPTimal replacement, 최적 교체)

- 앞으로 가장 오랫동안 사용하지 않을 페이지 교체

- 벨레이디가 제안, 페이지 부재 횟수가 가장 적게 발생

3) FIFO

- 페이지가 주기억장치에 적재된 시간을 기억해 가장 먼저 들어와 가장 오래있었던 페이지 교체

- 이해 쉽고 간단

4) LRU(Least Recently Used)

- 최근에 가장 오랫동안 사용하지 않는 페이지 교체

- 각 페이지마다 계수기나 스택을 두어 가장 오래전에 사용된 페이지 교체

5) LFU(Least Frequently Used)

- 사용 빈도가 가장 저근 페이지 교체

- 활발한 페이지는 사용횟수가 많아 교체되지 않고 사용

6) NUR(Nt Used Recently)

- 최근에 사용하지 않은 페이지 교체

- LRU에서 나타나는 시간적인 오버헤드를 줄일 수 있다.

- 각 페이지마다 두 개의 비트, 참조비트와 변형 비트 사용

- 참조비트 0 0 1 1

변형비트 0 1 0 1

교체순서 1 2 3 4

7) SCR(Second Chance Replacement, 2차 기회 교체)

- 가장 오랫동안 주기억장치에 있던 페이지 중 자주 사용되는 페이지의 교체를 방지

- FIFO 기법 단점 보완

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

155주차 - 가상기억장치 기타 관리 사항

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 페이지 크기

1) 페이지 크기가 작을 경우

- 페이지 단편화 감소, 한 페이지를 주기억장치로 이동하는 시간 감소

- 불필요한 내용이 적재될 확률이 적어 효율적인 워킹 셋 유지

- Locality에 더 일치할 수 있기 때문에 기억장치 효율 높아짐

- 페이지 맵 테이블 크기 커지고, 매핑 속도 늦어짐

- 접근 횟수가 많아져 입출력시간 늘어남

2) 페이지 크기가 큰 경우

- 페이지 맵 테이블 크기 작아짐, 매핑 속도 빨라짐

- 접근 횟수가 줄어들어 전체적인 입출력 효율성 증가

- 페이지 단편화 증가, 이동 시간 증가

- 불필요한 내용도 적재될 수 있다.

2. Locality(=국부성,지역성,구역성,국소성)

1) 프로세스가 실행되는 동안 주기억장치를 참조할 때 일부 페이지만 집중적으로 참조하는 성질이 있다.

2) 스래싱을 방지하기 위한 워킹 셋 이론의 기반이 되이ㅓㅆ다.

3) 프로세스가 집중적으로 사용하는 페이지 알아내는 방법, 가상기억장치 관리의 이론적 근거가 됨

4) 데닝 교수에 의해 구역성 개념 증명, 캐시 메모리 시스템의 이론적 근거

5) 종류

- 시간 구역성(Temporal Locality)

- 일정 시간 동안 집중적으로 액세스

- 한 번 참조한 페이지는 가까운 시간 내에 참조할 가능성이 높음

- 시간 구역성 이루어지는 기억 장소 : 반복,스택,counting,집계에 사용되는 변수

- 공간 구역성(Spatial Locality)

- 일정 위치의 페이지를 집중적으로 액세스

- 한 번 참조한 페이지는 그 근처 페이지를 참조할 가능성이 높음

- 공간 구역성 이루어지는 기억 장소 : 배열순회,순차적코드실행,근처에서 선언한 변수

3. 워킹 셋(Working Set)

1) 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합

2) 데닝 제안한 모델, 프로그램의 Locality 특성 이용

3) 자주 참조되는 워킹 셋을 주기억장치에 상주시켜 페이지 부재,교체 현상 감소, 기억장치 사용 안정됨

4) 시간에 따라 변화함

4. 페이지 부재 빈도 방식(PFF : Page Fault Frequentcy)

1) 페이지 부재율을 적정 수준으로 유지

2) 부재율이 상한선을 넘어가면 더 많은 페이지 프레임 할당, 하한선을 넘어가면 페이지 프레임 회수

5. 프리페이징(Prepaging)

1) 처음부터 과도한 페이지 부재를 방지하기 위해 필요할 것 같은 페이지를 한꺼번에 페이지 프레임에 적재

2) 사용되지 않은 페이지가 많을 수도 있다.

6. 스래싱(Thrashing)

1) 프로세스 처리 시간보다 페이지 교체에 소요되는 시간이 많아지는 현상

2) 전체 시스템 성능 저하

3) 다중 프로그래밍의 정도가 높아짐에 따라 CPU사용량이 늘어가다가 정도가 너무 커지면 스래싱이 나타나고

CPU 사용량이 급격하게 감소

4) 스래싱 현상 방지 방법

- 다중 프로그래밍의 정도를 적정 수준으로 유지

- 페이지 부재 빈도 조절하여 사용

- 워킹 셋 유지

- 부족한 자원 증설, 일부 프로세스 중단

- 임계치를 예상하여 운영

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

156주차 - 프로세스의 개요

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 프로세스의 정의

1) 프로세스는 일반적으로 프로세서(CPU,처리기)에 의해 처리되는 실행중인 프로그램을 의미, 작업(Job), 태스크(Task)

2) 프로세스 형태

- PCB 가진 프로그램

- 실기억장치에 저장된 프로그램

- 프로시저(프로세스를 잘게 쪼갠거)가 활동중인 것

- 비동기적 행위 일으키는 주체

- 운영체제가 관리하는 실행 단위

2. PCB(Process Control Block)

1) 운영체제가 프로세스에 대한 중요한 정보 저장해 놓는 곳

2) 저장된 정보 종류

- 프로세스 현재 상태

- 포인터

- 프로세스 고유 식별자

- 스케줄링 및 프로세스의 우선순위

- CPU 레지스터 정보 : 프로그램 카운터(PC), 누산기

- 주기억장치 관리 정보 : 기준 레지스터, 페이지 테이블

- 입출력 상태 정보

- 계정 정보

3. 프로세스 상태 전이

1) 제출 -> 접수 -> 준비 -> 실행 -> 종료

└ ┐ └ <┘

디스크 대기

2) 제출(Submit)

- 사용자가 작업을 시스템에 제출한 상태

3) 접수(Hold)

- 작업이 스플 공간인 디스크의 할당 위치에 저장된 상태

4) 준비(Ready)

- 프로세스가 프로세서를 할당받기 위해 기다리고 있는 상태

- 준비상태 큐에서 실행 준비

- 접수 상태에서 준비 상태로의 전이는 Job 스케줄러에 의해 수행

5) 실행(Run)

- 수행이 완료되기 전 프로세스에게 주어진 프로세서 할당 시간이 종료되면 프로세스는 준비 상태로 전이됨

- 실행중인 프로세스에 입출력이 필요하면 대기 상태로 전이

- 준비 상태에서 실행 상태로의 전이는 CPU(프로세서) 스케줄러에 의해 수행

6) 대기(Wait), 보류, 블록(Block)

- 입출력 처리가 완료될 때까지 대기하고 있는 상태

7) 종료(Terminated, Exit)

- 프로세스 할당이 해제된 상태

4. 프로세스 상태 전이 관련 용어

1) Dispatch

- 준비 상태에서 대기하는 프로세스 하나가 프로세서 할당받아 실행 상태로 전이되는 과정

2) Wake Up

- 입출력 작업이 완료되어 대기 상태에서 준비 상태로 전이되는 과정

3) Spooling

- 입출력장치 공유 및 상대적 느린 입출력장치 처리 속도 보완, 다중 프로그래밍 성능 향상을 위해 나중에 한꺼번에

입출력하기 위해 디스크에 저장한는 과정

4) 교통량 제어기(Traffic Controller)

- 프로세스의 상태에 대한 조사와 통보 담당

5. 스레드(Thread) = 경량 프로세스

1) 프로세스 내 작업 단위, 여러 자원 할당받아 실행

2) 단일 스레드 : 하나의 프로세스에 하나의 스레드

다중 스레드 : 하나의 프로세스에 하나 이상의 스레드

3) 독립적인 스케줄링 최소 단위

4) 동일 프로세스 환경에서 서로 독립적인 다중 수행 가능

5) 분류

- 사용자 수준 스레드

- 사용자가 만든 라이브러리 이용해 스레드 운용

- 커널 모드 전환 없어 오버헤드 줄음

- 속도 빠름 구현 어려움

- 커널 수준 스레드

- 커널에 의해 스레드 운용

- 한 프로세스가 운영체제 호출할 때 전체 프로세스가 대기하지 않으므로 시스템 성능 높아짐

- 여러 스레드가 커널에 동시접근가능

- 독립적인 스케줄링 가능

- 구현 쉬움 속도 느림

6) 스레드 사용 장점

- 하나 프로세스를 여러개 스레드로 나눠 병행성 증진

- 처리율 향상

- 응답 시간 단축

- 기억장소 낭비 줄음

- 프로세스 간 통신 향상

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

158주차 - 주요 스케줄링 알고리즘

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 시간

1) 대기 시간 : 프로세스가 대기한 시간

2) 반환 시간 : 대기시간 + 실행시간

2. FCFS(First Come First Service) = FIFO

3. SJF(Short Job First, 단기 작업 우선)

- 가장 적은 평균 대기 시간

- 실행 시간 긴 프로세스가 무한 연기 상태 될 수 있음

4. HRN(Hightest Response-ratio Next)

- 실행시간이 짧거나 대기시간이 긴 프로세스에게 우선순위 줌

- 우선순위 계산식 = (대기 시간 + 서비스 시간)/서비스 시간

- 우선순위 높은 것 = 계산값이 큰 것

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

159주차 - 환경 변수

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 환경 변수의 개요

1) 시스템 소프트웨어의 동작에 영향을 미치는 동적인 값들의 모임

2) 변수명과 값으로 구성됨

3) 자식 프로세스에 상속됨

2. Windows의 주요 환경 변수

1) 앞뒤에 % 기호 붙음

2) set을 입력하면 모든 환경 변수와 값 출력

3. UNIX/LINUX의 주요 환경 변수

1) 앞에 $ 기호 붙음

2) set, env, printenv, setenv 중 하나 입력하면 모든 환경 변수와 값 출력

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

162주차 - OSI 참조 모델

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. OSI(Open Systme Interconnection) 참조 모델의 개요

1) 다른 시스템 간 원활한 통신을 위함 ISO 통신 규약

2) 7단계

- 응용 계층

- 표현 계층

- 세션 계층

- 전송 계층

- 네트워크 계층

- 데이터 링크 계층

- 물리 계층

2. OSI 참조 모델에서의 데이터 단위

1) 프로토콜 데이터 단위(PDU : Protocol Data Unit)

- 동일 계층 간에 교환되는 정보의 단위

- 물리 계층 : 비트

- 데이터 링크 계층 : 프레임

- 네트워크 계층 : 패킷

- 전송 계층 : 세그먼트

- 세션, 표현, 응용 계층 : 메시지

2) 서비스 데이터 단위(SDU : Service Data Unit)

- 서비스 접근점(SAP)을 통해 상하위 계층끼리 주고받는 데이터 단위

- 서비스 접근점 : 하위 계층을 통해 상위 계층에 서비스 제공, 하위 계층과 상위 계층의 통신 경계점

3. 물리 계층

1) 두 장치 간 실제 접속,절단 등 기계적,전기적,기능적,절차적 특성에 대한 규칙 정의

2) 물리적 전송 매체와 전송 신호 방식 정의, RS-232C, X.21 등 표준 존재

3) 관련 장비 : 리피터, 허브

4. 데이터 링크 계층

1) 인접한 개방 시스템들 간에 신뢰성 있고 효율적 정보 전송을 위한 연결 설정과 유지 및 종료 담당

2) 송수신 측 속도 차이를 해결하기 위한 흐름제어기능

3) 프레임 시작 끝을 구분하기 위한 동기화 기능

4) 오류 검출,회복을 위한 오류 제어 기능

5) 프레임 순서를 위한 순서 제어 기능

6) HDLC, LAPB, LLC, MAC, LAPD, PPP 등 표준 존재

7) 관련 장비 :랜카드, 브리지, 스위치

5. 네트워크 계층

1) 개방 시스템 간 네트워크 연결관리 및 데이터 교환, 중계

2) 발신지,목적지의 논리 주소가 추가된 패킷을 이용해 최종 목적지가지 전달하는 책임

3) 경로 설정, 트래픽 제어, 패킷 정보 전송 수행

4) X.25, IP 등 표준 존재

5) 관련 장비 : 라우터

6. 전송 계층

1) 논리적 안정과 균일한 데이터 전송 서비스 제공, 종단 시스템 간 투명한 데이터 전송

2) 하위 3계층과 상위 3계층 사이 인터페이스 담당

3) 주소 설정, 다중화, 오류 제어, 흐름 제어 수행

4) TCP, UDP 등 표준 존재

5) 관련 장비 : 게이트웨이

7. 세션 계층

1) 송수신 간 관련성 유지 및 대화 제어

2) 대화구성 및 동기 제어, 데이터 교환 관리 기능

3) 토큰 사용

4) 전송하는 정보 일부분에 체크점을 두어 수신 상태 체크, 체크점을 동기점이라함(Synchronization Point)

5) 소동기점 : 하나의 대화 단위 내 데이터 전달 제어, ack 받지 않음

대동기점 : 전송하는 데이터 처음과 끝에 사용, 반드시 ack 받음

8. 표현 계층

1) 정보를 다른 계층으로 보낼 때 적당한 형태로 변환

2) 서로 다른 데이터 표현 형태를 갖는 시스템 간 상호 접속할 때 필요

3) 코드 변환, 데이터 암호화, 데이터 압축, 구문 검색, 정보 형식 변환, 문맥 관리 기능

9. 응용 계층

1) 서비스 제공

2) 파일 전송, 가상 터미널 등

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

163주차 - 네트워크 관련 장비

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 네트워크 인터페이스 카드(NIC : Network Interface Card)

1) 컴퓨터와 컴퓨터, 컴퓨터와 네트워크를 연결하는 장치

2) 정보 전송 시 케이블 통해 전송될 수 있도록 정보 형태 변경

3) 이더넷 카드(LAN카드), 네트워크 어댑터라고 함

2. 허브(Hub)

1) 가까운 거리 컴퓨터 연결 장치, 각 회선을 통합적 관리, 신호증폭하는 리피터역할도 함

2) 더미 허브

- 네트워크에 흐르는 모든 데이터를 단순히 연결

- LAN이 보유한 대역폭을 컴퓨터 수만큼 나누어 제공

- 네트워크에 연결된 각 노드를 물리적인 성형 구조로 연결

3) 스위칭 허브

- 네트워크상에 흐르는 데이터 유무 및 흐름 제어하여 각 노드가 허브의 최대 대역폭을 사용할 수 있는 지능형 허브

- 최근 대부분 사용

3. 리피터

1) 신호가 왜곡되거나 약해질 경우 원래 신호로 재생하여 다시 전송

2) 물리 계층에서 동작

3) 근접 네트워크 간 신호 전송, 전송 거리 연장 또는 배선 자유도 높이기 위한 용도

4. 브리지

1) LAN과 LAN 연결하거나 LAN안에서 컴퓨터 그룹(세그먼트) 연결

2) 데이터 링크 계층 중 MAC(Mandatory Access Control) 계층에서 사용하므로 MAC브리지라 불림

3) 트래픽 병목 현상 줄일 수 있음

4) 보안성 높임

5) 브리지를 이용한 서브넷 구성 시 전송 가능한 회선 수는 브리지가 n개일 때 n(n-1)/2

5. 스위치

1) 브리지처럼 LAN과 LAN을 연결하여 훨씬 더 큰 LAN 만드는 장치

2) 하드웨어 기반 처리, 전송 속도 빠름

3) 포트마다 다른 전송 속도 지원 가능, 수십 수백개 포트 제공

4) 데이터링크계층에서 사용

6. 라우터

1) 브리지처럼 LAN과 LAN 연결 기능에 최적 경로 선택할 수 있는 기능 추가

2) 네트워크 계층

3) 라우팅 제어표에 저장하여 보관

4) 3계층까지 프로토콜 구조가 다른 네트워크 간의 연결을 위해 프로토콜 변환 기능을 추가한다.

5) 브라우터 : 브리지와 라우터의 기능을 모두 수행,

- 브리지 : 내부 네트워크를 분리하기 위한 용도

- 라우터 : 외부 네트워크에 연결하는 용도

7. 게이트웨이

1) 전 계층의 프로토콜 구조가 다른 네트워크의 연결 수행

2) 상위 3계층 연결하여 데이터 형식 변환, 주소 변환, 프로토콜 변환 수행

3) LAN에서 다른 네트워크에 데이터 보내거나 받을 때 출입구 역할

8. 전처리기(FEP : Front End Processor)

1) 통신 회선 및 단말장치 제어, 메시지 조립과 분해, 전송 메시지 검사 등 미리 수행, 컴퓨터 부담 줄여줌

2) 호스트 컴퓨터와 단말장치 사이에 고속 통신 회선으로 설치

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

165주차 - TCP/IP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. TCP/IP의 개요(Transmission Control Protocol/Internet Protocol)

1) 서로 다른 기종의 컴퓨터들이 데이터 주고받을 수 있도록 하는 표준 프로토콜

2) ARPA에서 개발, ARPANET에서 사용

3) UNIX의 기본 프로토콜로 사용되었고, 현재 인터넷 범용 프로토콜로 사용

4) TCP

- 전송 계층

- 신뢰성 있는 연결형 서비스(논리적 연결 후 데이터 전송)

- 패킷의 다중화,순서제어,오류제어,흐름제어

- 스트림 전송 기능

- TCP 헤더에 source/destination port number, sequence number, acknowlegement number, checksum 있음

5) IP

- 네트워크 계층

- 데이터그램 기반의 비연결형 서비스(논리적 연결하지 않고 데이터 전송)

- Best Effort 원칙

- 패킷의 분해/조립, 주소 지정, 경로 선택

- 헤더 길이 최소 20byte 최대 60byte

- IP 헤더에 Version,Header Length,Total Packet Length,Header Checksum,Source IP Address,Destination IP Address 있음

2. TCP/IP 구조

1) 응용,표현,세션 계층

- TCP/IP : 응용 계층

- 기능

- 응용 프로그램 간 데이터 송수신 제공

- TELNET,FTP,SMTP,SNMP,DNS,HTTP

2) 전송 계층

- TCP/IP : 전송 계층

- 기능

- 호스트들 간의 신뢰성 있는 통신 제공

- TCP, UDP

3) 네트워크 계층

- TCP/IP : 인터넷 계층

- 기능

- 데이터 전송을 위한 주소 지정,경로 설정 제공

- IP,ICMP,IGMP,ARP,RARP

4) 데이터 링크,물리 계층

- TCP/IP : 네트워크 액세스 계층

- 기능

- 실제 데이터(프레임)를 송수신하는 역할

- Ethernet, IEEE802, HDLC, X.25, RS-232C, ARQ

3. 응용 계층의 주요 프로토콜

1) FTP(File Transfer Protocol)

- 컴과 컴, 컴과 인터넷 사이 파일을 주고받을 수 있도록 하는 원격 파일 전송 프로그램

2) SMTP(Simple Mail Tranfer Protocol)

- 전자 우편 전송 프로토콜

3) TELNET

- 멀리 떨어진 컴퓨터에 접속해 내 컴퓨터처럼 사용하는 서비스

- 프로그램 실행 등 시스템 관리 작업을 하는 가상 터미널기능 수행

4) SNMP(Simple Network Management Protocol)

- TCP/IP 네트워크 관리 프로토콜

- 라우터, 허브 등 네트워크 기기의 네트워크 정보를 네트워크 관리 시스템에 보내는 데 사용되는 표준 통신 규약

5) DNS(Domain Name System)

- 도메인 네임을 IP주소로 매핑하는 시스템

6) HTTP(HyperText Transfer Protocol)

- 월드 와이드 웹(WWW)에서 HTML문서 송수신하기 위한 표준 프로토콜

7) MQTT(Message Queuing Telemetry Transport)

- 발행-구독 기반 메시징 프로토콜

- IoT환경에서 자주 사용

4. 전송 계층의 주요 프로토콜

1) TCP(Transmission Control Protocol)

- 양방향 연결

- 가상 회선 연결

- 스트림 위주의 패킷 전달

- 신뢰성 있는 경로 확립 및 메시지 전송 감독

- 순서 제어, 오류 제어, 흐름 제어

- 패킷분실,손상,지연,순서오류 등 발생 시 투명성 보장되는 통신 제공

- 헤더는 20byte ≤ ≤ 60byte, 선택적으로 40byte 추가 가능 최대 100byte까지 확장 가능

- FTP,SMTP,TELNET,HTTP 등이 이용

2) UDP(User Datagram Protocol)

- 데이터 전송 전 연결 설정 않는 비연결형 서비스

- TCP에 비해 단순한 헤더 구조, 오버헤드 적고, 흐름제어나 순서제어가 없어 전송 속도 빠름

- 동시에 여러사용자에게 데이터 전달, 정기적 반복 전달 등의 경우에 사용

- 실시간 전송에 유리, 속도가 중요시되는 네트워크에서 사용

- UDP 헤더에는 Source Port Number,Destination Port Number,Length,Checksum

3) RTCP(Real-Time Control Protocol)

- 패킷의 전송 품질 제어하는 제어 프로토콜

- 세션에 참가한 각 참여자들에게 주기적으로 제어 정보 전송

- 하위 프로토콜은 데이터 패킷과 제어 패킷의 다중화를 제공

- 데이터 전송 모니터링, 최소한의 제어와 인증 기능 제공

- 항상 32비트 경계로 끝남

5. 인터넷 계층의 주요 프로토콜

1) IP(Internet Protocol)

- 전송 데이터에 주소 지정, 경로 설정

- 비연결형 데이터그램 방식 사용, 신뢰성 보장 안됨

2) ICMP(Internet Control Message Protocol)

- IP와 조합하여 통신중에 발생하는 오류 처리,경로 변경 등을 위한 제어 메시지 관리

- 헤더는 8바이트로 구성됨

3) IGMP(Internet Group Management Protocol)

- 멀티캐스트를 지원하는 호스트나 라우터 사이에서 멀티캐스트 그룹 유지를 위해 사용

4) ARP(Address Resolution Protocool)

- 호스트의 IP주소를 호스트와 연결된 네트워크 접속 장치의 물리적 주소(MAC Address)로 바꿈

5) RARP(Reverse Address Resolution Protocol)

- ARP와 반대로 물리적 주소를 IP주소로 변환

6. 네트워크 액세스 계층의 주요 프로토콜

1) Ethernet(IEEE 802.3)

- CSMA/CD 방식의 LAN

2) IEEE 802

- LAN을 위한 표준 프로토콜

3) HDLC

- 비트 위주의 데이터 링크 제어 프로토콜

4) X.25

- 패킷 교환망을 통한 DTE와 DCE 간의 인터페이스 제공 프로토콜

5) RS-232C

- 공중 전화 교환망(PSTN)을 통한 DTE와 DCE간의 인터페이스 제공 프로토콜