

# 정보처리기사 실기) 11장 응용 SW 기초 활용

by 코딩하는 핑가 2020. 10. 13.

의 상호작용할 수 있는 편리성을 제공

\* 인터페이스 : 서로 다른 시스템 또는 장치 사이에서 정보를 공유하는 시스템 또는 경계

|                |  |
|----------------|--|
| Windows        | GUI 제공, 선점형 멀티태스킹 방식 제공,<br>자동 감지 기능, OLE(Object Linking and Embeded) 사<br>용 |
| Unix           | 대화식 운영체제, 다중 작업 기능 제공, 다중 사용자 기능<br>제공,<br>이식성 제공, 계층적 트리구조 파일 시스템 제공        |
| Linux          | 유닉스 기반, 유닉스 특징, 오픈 소스,<br>다양한 리눅스 운영체제 종류                                    |
| MAC            | 유닉스 기반 GUI 운영체제 (Apple 사)  |
| 안드로이드(Android) | 리눅스 기반의 안드로이드 운영체제,<br>런 타임 라이브러리 제공, SDK                                    |

\* 안드로이드(Android) (2020 실기 2회 기출)

- 휴대전화를 비롯한 휴대용 장치를 위한 운영체제와 미들웨어, 사용자 인터페이스 그리고 표준 응용 프로그램 등  
을 포함하고 있는 소프트웨어 스택이자 리눅스 모바일 운영체제이다. 개발자들이 자바와 코틀린 언어로 응용 프로  
그램을 작성할 수 있게 하였으며, 컴파일된 바이트코드를 구동할 수 있는 런타임 라이브러리를 제공한다.

\* 운영체제 제어

- GUI (Graphic User Interface) : 마우스 클릭 등 그래픽 기반으로 제어
- CLI (Command Line Interface) : 사용자가 직접 명령어 입력 후 컴퓨터에 명령을 내리는 방식

#### \* 리눅스/유닉스 계열 기본 명령어 (2020 실기 2회 기출)

- 셸(Shell) : 컴퓨터 내부를 관리하는 커널과 사용자 간의 연결하는 명령어 창

#### \* 운영체제 핵심 기능

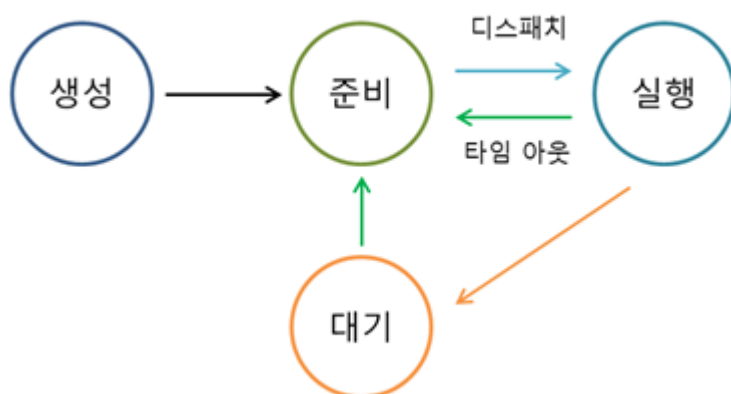
- 중앙처리 장치, 메모리, 스토리지, 주변기기 관리
- 메모리 관리, 프로세스 관리
- 메모리에 있는 프로그램을 CPU로 이동시켜 관리

#### \* MMU(단답형) (Memory Management Unit)

- CPU가 메모리에 접근하는 것을 관리하는 하드웨어 부품으로 가상 메모리를 실제 메모리 주소로 변환하는 장치

## 2) 프로세스 관리

- 생성 상태 : 사용자에 의해 프로세스가 생성된 상태
- 준비 상태 : CPU에 할당받을 수 있는 상태
- 실행 상태 : 프로세스가 CPU를 할당받아 동작 중인 상태
- 대기 상태 : 프로세스 실행 중 처리로 인해 CPU를 양도하고 대기 리스트에서 기다리는 상태
- 완료 상태 : 프로세스가 CPU 할당을 받고 주어진 시간에 완전히 수행한 상태



출처; 별랑님 블로그

#### \* 디스패치

- 준비 상태에 있는 프로세스를 스케줄링하여 CPU에 할당

**\* 타이머 런 아웃(타임아웃)**

- CPU에 할당된 프로세스가 지정된 시간이 초과되면 스케줄러에 의해 PCB 저장

**\* PCB(Process Control Block)**

- 운영체제가 프로세스의 정보를 표현한 것

**\* 문맥 교환(단답형)**

- CPU가 현재 실행하고 있는 프로세스의 상태를 PCB에 저장하고 다음 프로세스의 PCB로부터 문맥을 복원하는 작업

**(1) 프로세스 스케줄링 : CPU를 사용하려는 프로세스 사이의 우선순위를 관리하는 작업**

- 프로세스 스케줄링 이용 시 CPU 이용률 증가, 오버헤드 응답시간, 반환시간, 대기시간 감소

\* 서비스 시간 : 프로세스가 결과를 산출하기 전까지 소요되는 시간

\* 응답 시간 : Response Time으로 대기시간 + 수행시간

\* 대기 시간 : 프로세스가 할당되기 전까지 대기 큐에서 대기하는 시간

\* 응답률 :  $(\text{서비스 시간} + \text{대기시간}) / \text{서비스 시간} \rightarrow \text{HRN(Highest Response Ratio Next)}$  (2020 실기 1회 기출)

**(2) 선점형 스케줄링**

- 하나의 프로세스가 CPU에 할당되어 있을 때 **우선순위가 높은 다른 프로세스가 현재 프로세스를 중단시키고 CPU를 점유**하는 스케줄링 방식

- 라운드 로빈(RR), SRT(Shortest Remining Time First), 다단계 큐, 피드백 큐

**\* 라운드 로빈**

- CPU 할당 시간 설정 후 프로세스가 할당된 시간 내 처리하지 못할 경우 큐 리스트의 가장 맨 뒤로 이동하고 다음 프로세스를 CPU에 할당하는 기법

**\* SRT**

- 가장 짧은 시간이 소요되는 프로세스를 먼저 처리하고 남은 시간이 짧은 프로세스가 리스트 큐에 생기면 CPU에 할당하는 기법

**\* 다단계 큐**

- 작업들을 분할, 여러 개의 큐를 이용하여 상위 단계 작업에 의한 하위 단계 작업 선정 및 독자적인 스케줄링

**\* 다단계 피드백 큐**

- FIFO + RR의 혼합, 마지막 단계는 RR 방식으로 처리

[정보처리기사 필기/2과목 소프트웨어 개발] - 정보처리기사 필기 ) 선점 스케줄링

### (3) 비선점형 스케줄링

- 한 프로세스가 CPU를 할당받으면 작업 후 **CPU 반환 시까지 다른 프로세스가 CPU에 할당하지 못하는** 스케줄링 기법

- 우선순위, 기한부, FCFS, HRN, SJF

#### \* 우선순위

- 프로세스 별로 우선순위가 주어지고, 우선순위에 따라 CPU를 할당함

#### \* 기한부

- 작업들을 명시된 시간이나 기한 내 완료되도록 계획

#### \* FCFS

- 프로세스가 대기 큐 리스트에 도착한 순서에 따라 CPU를 할당

#### \* SJF

- 가장 작은 서비스 시간을 가진 프로세스를 우선순위로 하여 평균 대기 시간을 최소화시키는 기법

#### \* HRN(2020 실기 1회 기출)

- 대기 중인 프로세스 중 현재 응답률이 가장 높은 프로세스를 우선순위

- (서비스 시간 + 대기 시간) / 서비스 시간

[정보처리기사 필기/2과목 소프트웨어 개발] - 정보처리기사 필기 ) 비선점 스케줄링

### 3) 가상화

- 물리적인 리소스들을 사용자에게 하나로 보이게 하거나 하나의 물리적인 리소스를 여러 개로 보이게 하는 기술

- 플랫폼 가상화 : 플랫폼 위에 독립된 환경을 만드는 기술

- 리소스 가상화 : 독립된 하드웨어에서 소프트웨어가 실행되는 것처럼 활용하는 기법

- 컴퓨터 가상화(Hyper V) : 물리적인 리소스를 논리적인 리소스로 사용

- 분산처리 기술 : 여러 대 컴퓨터를 이용하여 대용량 처리

### 4) 컴퓨터 클라우딩

- 인터넷을 통해 클라우드 서버를 접속하여 리소스를 제공하고 자원들을 네트워크를 통해서 어디서든지 접속할 수 있는 기술

- 인프라형(IaaS) : 서버, 스토리지 같은 시스템 자원 클라우드 제공

- 플랫폼형(PaaS) : 인프라 생성 및 관리하는 복잡함을 없애고 애플리케이션 개발과 실행 시 사용

- 소프트웨어형(SaaS) : 데이터는 중앙에 호스팅되고 사용자는 웹 브라우저 등의 클라이언트를 통해 접속하여 소프트웨어를 서비스 형태로 이용

## 5) 데이터 베이스

- 시스템과 프로그램이 사용할 목적으로 통합하여 관리되는 데이터의 집합

### \* 데이터베이스 종류

- **파일 시스템** : 파일에 이름을 부여하고 어디에 위치시켜야 하는지 등을 정의한 뒤 DB 전 단계의 데이터를 관리하는 방식

- **계층형 데이터베이스 관리시스템** : 데이터를 **상하 종속적인** 관계로 계층화하여 관리하는 DB

- **망형 데이터베이스 관리시스템** : 데이터 구조를 **네트워크의 망 형태로** 논리적으로 표현한 DB 모델

- **관계형 데이터베이스 관리시스템** : 데이터를 저장하는 테이블의 일부를 다른 테이블과 **상하관계로 표시하며 상관 관계를 정리한 시스템** ( ORACLE, SQL Server, My SQL, Maria DB )

### \* 데이터베이스 도구

- DB 관리자가 데이터베이스를 편리하고 쉽게 다룰 수 있도록 도와주는 도구

- DB 생성 및 삭제, SQL 명령어 수행, 상태 모니터링, 사용자 계정 관리, DB 내보내기

### \* DBMS

- 데이터의 관리의 복잡성을 해결하는 동시에 데이터 생성 및 관리(데이터 추가, 삭제, 검색, 변경 및 보안, 백업, 권한 등의) 기능을 지원하는 소프트웨어

### \* 유형

- 키-값 DBMS : GET/PUT/DELETE 기능 제공

- 칼럼 기반 데이터 저장 DBMS : Key 안에 Column과 Value의 조합

- 문서 저장 DBMS : 값의 데이터 타입을 문서 타입으로 사용

- 그래프 DBMS : 그래프 데이터 표현

### \* 특징

- 데이터 무결성 : 데이터의 정확성, 일관성, 유효성이 유지되는 것

- 데이터 일관성 : 저장 후 데이터는 일정

- 데이터 회복성 : 장애 발생 시 회복 가능

- 데이터 보안성 : 불법적인 노출, 변경, 손실로부터 보호

- 데이터 효율성 : 응답시간, 저장 공간 활용 등 최적화

### \* ERD

- 업무 분석 결과로 도출된 엔티티(개체)와 엔티티(개체) 간의 관계를 도식화한 다이어그램

- 엔티티 : 현실 세계의 사물이나 사건으로 정의 ( 다이어그램에서 네모로 표현 )
- 속성 : 엔티티가 가지고 있는 요소 또는 성질 ( 다이어그램에서 원으로 표현 )
- 관계 : 두 엔티티 간의 관계를 정의 ( 다이어그램에서 마름모로 표현 )

## 6) 트랜잭션 ( 2020 실기 1회 기출 )

- DB 시스템에서 하나의 논리적인 기능을 수행하기 위한 작업의 기본 단위
- 원자성 (Atomicity) : 전체가 실패 또는 성공
- 일관성 (Consistency) : 트랜잭션 성공 이후 항상 일관된 DB 상태 유지
- 격리성 (Isolation) : 트랜잭션 실행 중 다른 트랜잭션 접근 불가
- 영속성 (Durability) : 성공된 트랜잭션 결과는 DB에 영속적으로 저장

### \* 트랜잭션의 특성

#### - Atomicity(원자성)

: 트랜잭션의 연산은 데이터베이스에 모두 반영되도록 **완료(Commit)**되든지 아니면 **전혀 반영되지 않도록 복구(Roll back)**되어야 한다.

: 트랜잭션 내의 모든 명령은 반드시 완벽히 수행되어야 하며, 모두가 완벽히 수행되지 않고 어느 하나라도 오류가 발생하면 트랜잭션 전부가 취소되어야 한다.

: 타인에게 송금하는 도중 장애가 발생하였을 경우 부작용을 방지하는 특성

#### - Consistency(일관성)

: 트랜잭션이 그 실행을 **성공적으로 완료**하면 **언제나 일관성 있는 데이터베이스 상태로 변환**한다.

: 시스템이 가지고 있는 **고정 요소**는 트랜잭션 수행 전과 트랜잭션 수행 완료 후의 **상태가 같아야** 한다.

#### - Isolation(독립성, 격리성, 순차성)

: 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 **연산이 끼어 들 수 없다.(접근 불가)**

: 수행중인 트랜잭션은 완전히 **완료될 때까지 다른 트랜잭션에서 수행 결과를 참조할 수 없다.**

#### - Durability(영속성, 지속성)

: **성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 함**

### \* 트랜잭션 상태

- 활동 상태 : 초기 상태, 트랜잭션이 실행 중인 상태
- 부분 완료 상태 : 마지막 명령문이 실행되고 난 후 가지는 상태
- 완료 상태 : 트랜잭션이 성공적으로 처리된 후 완료된 상태
- 실패 상태 : 정상적인 실행이 더 이상 진행되지 않는 상태

- 철회 상태 : 트랜잭션이 취소되고 DB가 트랜잭션 시작 전 상태로 환원

### \* 용어

- TCL : 트랜잭션의 결과를 허용하거나 취소하는 목적으로 사용

- COMMIT : 모든 작업을 정상적으로 처리하겠다고 트랜잭션을 확정하는 명령어

- ROLLBACK(2020 실기 2회 기출) : 트랜잭션의 실패로 작업을 취소하고, 이전 상태로 되돌리는 데이터 제어어

- CHECKPOINT : 트랜잭션을 저장하는 시기를 설정하는 명령어

- Trigger : 트랜잭션을 취소하는 이외의 조치를 명세할 필요가 있는 경우 메시지를 보내 어떤 값을 자동으로 갱신하도록 프로시저를 가동시키는 방법

## 7) CRUD

### \* 데이터베이스의 기본 연산

- CREATE : 생성 (INSERT INTO)

- READ : 조회 (SELECT FROM)

- UPDATE : 갱신 (UPDATE SET)

- DELETE : 삭제 (DELETE FROM)

## 8) 11장 응용 소프트웨어 단답형 및 서술형 예상 내용

### \* NoSQL (NOT Only SQL)

- 전통적인 RDBMS(관계형 데이터베이스 관리 시스템)와 달리 데이터 저장에 필요한 테이블 스키마가 필요하지 않고 Join 연산도 하지 않는 수평적 확장이 가능한 DBMS

### \* 온톨로지(단답형)

- 설계 시 개념, 속성, 관계를 컴퓨터가 이해할 수 있도록 서술

### \* 시맨틱 웹(단답형)

- 온톨로지를 이용하여 서비스 검색, 조합 등 기능들을 자동화하는 웹

### \* 데이터 마이닝(2020 실기 1회 기출)

- 대규모로 저장된 데이터 안에서 체계적이고 자동적으로 규칙이나 패턴을 찾아내는 기술

### \* 빅데이터

- 주어진 비용과 시간을 초과한 페타바이트 크기의 정형 데이터

## 9) 네트워크

- 원하는 정보를 수신자에게 정확하게 전달하기 위한 인프라
- WAN : 광대역 네트워크로 전송거리가 넓고 라우팅 알고리즘이 필요한 네트워크
- LAN : 근거리 네트워크로 작은 지역을 커버하는 네트워크

### \* OSI 7계층

- 파일 참고

OSI 7계층.hwp  
0.06MB

- \* **프로토콜** : 이기종 시스템간 데이터 교환을 할 수 있도록 하는 표준화 통신 규약
- \* **프로토콜의 기본 요소(2020 실기 1회 기출)** : 구문, 의미, 타이핑
- \* **네트워크 프로토콜** : 컴퓨터나 통신장비 사이에서 메시지를 교환하는 통신규약이나 규칙

### \* TCP(서술형/단답형)

- 근거리 통신망이나 인터넷에서 신뢰성있는 전송을 수행하는 프로토콜
- 특징 : 신뢰성 보장, 연결 지향적, 흐름제어, 혼잡제어

### \* UDP(서술형/단답형)

- 비연결성이고 신뢰성이 없으며 순서화되지 않는 데이터그램 서비스를 제공하는 통신 프로토콜
- 특징 : 비 연결성, 비 신뢰성, 순서화되지 않은 데이터그램 서비스 제공, 단순 헤더

[네트워크관리사] - 네트워크관리사2급) TCP/UDP

### \* IPv4

- 데이터 교환을 위해 32bit 주소체계를 갖는 네트워크 계층의 프로토콜
- 유니캐스트, 멀티캐스트, 브로드캐스트

### \* IPv6



- 차세대 인터넷 프로토콜로 IPv4의 주소 자원 고갈, 보안성 등 문제점을 해결하기 위해 개발된 128bit 주소체계를 갖는 프로토콜
- 유니캐스트, 멀티캐스트, **애니캐스트**

#### \* 패킷 스위칭

- 작은 블록의 패킷으로 데이터를 전송하며 전송하는 동안만 네트워크를 사용

#### \* X.25(단답형)

- 두 단말장치가 패킷 통신망을 통해 원활한 패킷 전송을 하기 위한 프로토콜

#### \* 프레임 릴레이

- ISDN을 사용하기 위한 프로토콜

#### \* ATM(단답형)

- 비동기 전송모드인 광대역 전송에 사용되는 스위칭 기법

#### \* 서킷 스위칭

- 네트워크 리소스를 특정 사용계층이 독점하여 통신하는 방식

### 10) 라우팅 프로토콜(단답형)

#### \* 거리 벡터 알고리즘

- 인접 라우터와 정보를 공유하여 목적지까지의 방향과 거리를 결정
- **RIP** : 자율 시스템(AS) 내에서 사용하는 거리 벡터 알고리즘을 기반으로 개발된 통신규약으로 최대 홉수를 15개로 제한한 라우팅 프로토콜
- **IGRP** : RIP의 문제점을 개선하기 위해 개발된 통신규약으로 RIP보다 대규모로 복잡한 통신망에서의 운용을 상정한 규격으로 되어 있으며, 거리 벡터 라우팅 알고리즘을 채용하여 목적 통신망까지의 경로는 지연, 회성의 대역폭, 신뢰성 및 부하를 고려하여 결정하는 프로토콜

#### \* 링크 상태 알고리즘

- 링크 정보를 모든 라우터에게 전달하여 최적의 경로를 구성
- **OSPF**
- **BGP** : 규모가 큰 네트워크를 상호 연결하는 알고리즘

- \* 라우팅 프로토콜(Routing Protocol)
- 최종 목적지까지 패킷을 최소 비용 또는 최소 홉수를 알고리즘을 통해 적절한 경로를 설정해주는 프로토콜

|               | RIP<br>(Routing Information Protocol)                          | OSPF<br>(Open Shortest Path First)                             |
|---------------|--|--|
| 개념            | AS(Autonomous System) 내에서 사용하는 거리벡터 알고리즘에 기초하여 개발된 내부 라우팅 프로토콜 | RIP의 단점을 개선하기 위해 자신을 기준으로 하여 링크상태 알고리즘 기반으로 최단 경로를 찾는 라우팅 프로토콜 |
| 라우팅 테이블 전송 시기 | 일정주기(30초)  | 변화 발생시에만   |
| 최대 지원 네트워크 크기 | 15개  | 255개   |