



# 4과목-프로그래밍 언어 활용

(Part 3. 응용 SW 기초 기술 활용-1)

# 프로그래밍 언어 활용 총 파트

---

프로그래밍 언어 활용 4과목은 총 3Part로 이루어져 있다.

1장 서버 프로그램 구현(0.69%)

2장 프로그래밍 언어 활용(44.83%)

3장 응용 SW 기초 기술 활용(54.48%)

# 프로그래밍 언어 활용

응용 SW 기초 기술 활용 Part는 17개의 섹션으로 구성되어 있다.

001 운영체제의 개념

002 Windows

003 UNIX / LINUX/ MacOS

004 기억장치 관리의 개요

005 주기억장치 할당 기법 등급

006 가상기억장치 구현 기법/페이지 교체 알고리즘

007 가상기억장치 기타 관리 사항

008 프로세스의 개요

009 스케줄링

010 주요 스케줄링 알고리즘

011 환경 변수

012 운영체제 기본 명령어

013 인터넷

014 OSI 참조 모델

015 네트워크 관련 장비

016 프로토콜의 개념

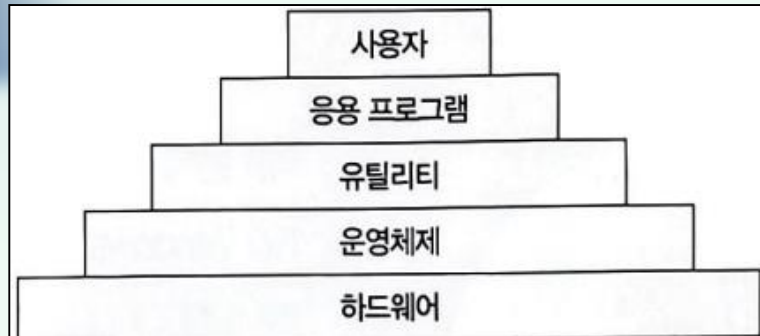
017 TCP/IP

# 4.응용 SW 기초기술 활용-SEC\_01(운영체제의 개념)

## 1) 운영체제의 개념

; 운영체제는 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임이다.

- 컴퓨터 사용자와 컴퓨터 하드웨어 간의 인터페이스로서 동작하는 시스템 소프트웨어의 일종으로, 다른 응용 프로그램이 유용한 작업을 할 수 있도록 환경을 제공해준다.



**자원(Resource)** : 자원은 시스템에서 사용할 수 있는 CPU, 주기억장치, 보조기억장치, 프린터, 파일 및 정보 등을 의미한다.

## 4.응용 SW 기초기술 활용-SEC\_01(운영체제의 개념)

### 2) 운영체제의 목적

; 운영체제의 목적에는 처리 능력 향상, 사용 가능도 향상, 신뢰도 향상, 반환 시간 단축 등이 있다.

- 처리 능력, 반환 시간, 사용 가능도, 신뢰도는 운영체제의 성능을 평가하는 기준이 된다.

처리 능력(Throughput)	일정 시간 내에 시스템이 처리하는 일의 양
반환 시간(Turn Around Time)	시스템에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간
사용 가능도(Availability)	시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도
신뢰도(Reliability)	시스템이 주어진 문제를 정확하게 해결하는 정도

## 4.응용 SW 기초기술 활용-SEC\_01(운영체제의 개념)

### 3) 운영체제의 구성

#### ● 제어 프로그램

- 제어 프로그램(Control Program)은 컴퓨터 전체의 작동 상태 감시, 작업의 순서 지정, 작업에 사용되는 데이터 관리 등의 역할을 수행하는 것으로 다음과 같이 구분할 수 있다.

감시 프로그램 (Supervisor Program)	제어 프로그램 중 가장 핵심적인 역할을 하는 것으로, 자원의 할당 및 시스템 전체의 작동 상태를 감시하는 프로그램
작업 관리 프로그램 (Job Management Program)	작업이 정상적으로 처리될 수 있도록 작업의 순서와 방법을 관리하는 프로그램
데이터 관리 프로그램 (Data Management Program)	작업에 사용되는 데이터와 파일의 표준적인 처리 및 전송을 관리하는 프로그램

#### ● 처리 프로그램

- 처리 프로그램(Processing Program)은 제어 프로그램의 지시를 받아 사용자가 요구한 문제를 해결하기 위한 프로그램으로, 다음과 같이 구분할 수 있다.

언어 번역 프로그램	사용자가 고급언어로 작성한 원시 프로그램을 기계어 형태의 목적 프로그램으로 변환시키는 것으로, 컴파일러, 어셈블러, 인터프리터 등이 있음
서비스 프로그램	• 사용자가 컴퓨터를 더욱 효율적으로 사용할 수 있도록 제작된 프로그램 • 분류/병합(Sort/Merge) , 유틸리티 프로그램 등이 여기에 해당됨

**분류/병합(Sort/Merge)** : 데이터를 일정한 기준으로 정렬하거나 정렬된 두 개 이상의 파일을 하나로 합치는 기능을 하는 서비스 프로그램

**유틸리티** : 컴퓨터 시스템에 있는 기존 프로그램을 지원하거나 기능을 향상 또는 확장하기 위해 사용하는 프로그램으로 디스크 관리, 화면 보호, 압축, 바이러스 검사/치료, 파일 백업 및 복구 프로그램 등이 있음

# 4.응용 SW 기초기술 활용-SEC\_01(운영체제의 개념)

## 4) 운영체제의 기능

- 프로세서(처리기, Processor), 기억장치(주기억장치, 보조기억장치), 입·출력 장치, 파일 및 정보 등의 자원을 관리한다.
- 자원을 효율적으로 관리하기 위해 자원의 스케줄링 기능을 제공한다.
- 사용자와 시스템 간의 편리한 인터페이스를 제공한다.
- 시스템의 각종 하드웨어와 네트워크를 관리 제어한다.
- 데이터를 관리하고, 데이터 및 자원의 공유 기능을 제공한다.
- 시스템의 오류를 검사하고 복구한다.
- 자원 보호 기능을 제공한다.
- 입·출력에 대한 보조 기능을 제공한다.
- 가상 계산기 기능을 제공한다.

**스케줄링(Scheduling)** : 스케줄링은 어떤 자원을 누가 언제, 어떤 방식으로 사용할지를 결정해 주는 것이다.

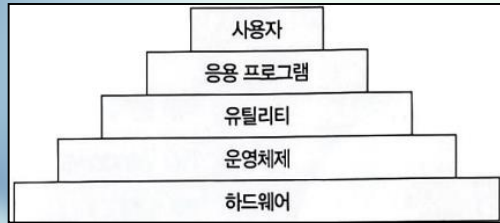
**가상 계산기(Virtual Computer)** : 가상 계산기는 한 대의 컴퓨터를 여러 대의 컴퓨터처럼 보이게 하는 가상 컴퓨터 운영체제에 의해 만들어지며, 사용자의 관점에서는 가상 컴퓨터가 실제 컴퓨터처럼 보일 수도 있고 아주 다르게 보일 수도 있다



# 응용 SW 기초기술 활용-SEC\_01(운영체제의 개념) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(운영체제의 개념)

1. 컴퓨터 시스템을 계층적으로 묘사할 때 운영체제의 위치는 다음 그림의 어느 부분에 해당되는가?



- ① 사용자와 응용 프로그램 사이
- ② 응용 프로그램과 유틸리티 사이
- ③ 유틸리티와 하드웨어 사이
- ④ 하드웨어 아래

운영체제는 '사용자와 하드웨어 간의 인터페이스 제공'이라는 것을 잊지 말자.

## 운영체제의 개념

운영체제는 컴퓨터 시스템의 여러 자원들을 스케줄링에 의거하여 효율적으로 관리하며, 컴퓨터를 사용자가 편리하고 효과적으로 사용할 수 있도록 GUI 및 명령어를 이용할 수 있는 환경을 제공 하는 여러 프로그램의 모임이다.

- 컴퓨터 사용자와 컴퓨터 하드웨어 간의 인터페이스로서 동작하는

3. 운영체제를 기능에 따라 분류할 경우 제어 프로그램이 아닌 것은?

- ① 데이터 관리 프로그램
- ② 서비스 프로그램
- ③ 작업 제어 프로그램
- ④ 감시 프로그램

## 운영체제의 구성

### 1. 제어 프로그램(Control Program)

- 제어 프로그램은 컴퓨터 전체의 작동 상태 감시, 작업의 순서 지정, 작업에 사용되는 데이터 관리 등의 역할을 수행하는 프로그램이다.

① 감시 프로그램(Supervisor Program) : 제어 프로그램 중에서 가장 핵심적인 역할을 하는 것으로, 자원의 할당 및 시스템 전체의 작동 상태를 감시하는 프로그램

② 작업 관리 프로그램(Job Management Program) : 작업이 정상적으로 처리될 수 있도록 작업의 순서와 방법을 관리하는 프로그램

③ 데이터 관리 프로그램(Data Management Program) : 작업에 사용되는 데이터와 파일의 표준적인 처리 및 전송을 관리하는 프로그램

### 2. 처리 프로그램(Processing Program)

- 처리 프로그램은 제어 프로그램에의 지시를 받아서 사용자가 요구한



# 응용 SW 기초기술 활용-SEC\_01(운영체제의 개념) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(운영체제의 개념)

### 5. 운영체제의 목적으로 거리가 먼 것은?

- ① 처리 능력의 향상
- ② 반환 시간의 최대화
- ③ 사용 가능도 증대
- ④ 신뢰도 향상

반환 시간은 작업을 사용자가 요청한 작업을 완료될 때까지의 시간을 의미하는데 이것이 최대화 되는 것을 목적으로 하지는 않는다.

### 운영체제의 목적

운영체제의 목적에는 처리 능력의 향상, 사용 가능도 향상, 신뢰도 향상, 반환 시간의 단축 등이 있다.

- 처리 능력, 반환 시간, 사용 가능도, 신뢰도는 운영체제의 성능을 평가하는 기준이 된다.

- ① **처리 능력(Throughput)** : 일정 시간 내에 시스템이 처리하는 일의 양
- ② **반환 시간(Turn Around Time)** : 시스템에 작업을 의뢰한 시간으로부터 처리가 완료될 때까지의 걸린 시간
- ③ **사용 가능도(Availability)** : 시스템을 사용할 필요가 있을 때 즉시

### 7. 운영체제의 성능평가 요인 중 다음 설명에 해당하는 것은?

이것은 컴퓨터 시스템 내의 한정된 각종 자원을 여러 사용자가 요구할 때, 어느 정도 신속하고 충분히 지원해 줄 수 있는지의 정도이다. 이는 사용 가능한 하드웨어 자원의 수나 다중 프로그래밍 정도 등의 요소가 좌우하는 것으로 같은 종류의 시스템 자원수가 많을 경우에는 이것이 높아질 수 있다.

- ① Throughput(처리 능력)
- ② **Availability(사용 가능도)**
- ③ Turn Around Time(반환 시간)
- ④ Reliability(신뢰도)

위의 지문을 요약하자면 '자원이 필요할 때 즉시 사용할 수 있는 정도'를 의미하고 있기에 이것은 사용 가능도(Availability)에 해당된다.

### 8. 운영체제의 일반적인 역할이 아닌 것은?

- ① 사용자들 간의 하드웨어의 공동 사용
  - ② 자원의 효과적인 운영을 위한 스케줄링
  - ③ 입·출력에 대한 보조 역할
  - ④ 실행 가능한 목적(Object) 프로그램 생성
- 목적(실행) 프로그램을 생성하는 것은 컴파일러의 역할이다.

# 응용 SW 기초기술 활용-SEC\_01(운영체제의 개념) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(운영체제의 개념)

9. 운영체제를 계층 구조로 나눌 때 ㉠ ~ ㉤에 들어갈 내용이 차례로  
옳게 나열된 것은?

프로세서(CPU)관리 - 메모리(RAM) 관리 - ( ㉠ ) - ( ㉡ )

- ( ㉢ )

[보기]

㉠ 프로세스 관리    ㉡ 파일 관리    ㉢ 주변장치 관리

① ㉠ - ㉡ - ㉢

② ㉠ - ㉢ - ㉡

③ ㉡ - ㉡ - ㉠

④ ㉡ - ㉠ - ㉡

**운영체제 계층에는 5개의 계층이 존재한다.**

1계층(프로세서(CPU)관리) : 주로 프로세스의 동기화와

CPU스케줄링을 담당한다. 여기서는 스케줄링은 처리해야 할 일들의  
순서를 정하는 일 또는 중앙처리장치(CPU)를 사용할 수 있는 순서를  
정하는 일을 의미한다.

2계층(메모리(RAM)관리) : 메모리 할당과 회수하는 역할을 담당한다.

3계층(프로세스 관리) : 프로세스를 생성하거나 제거하는 역할을

# 4.응용 SW 기초기술 활용 - SEC\_02(Windows)

## 1) Windows의 개요

; Windows는 1990년대 마이크로소프트(Microsoft) 사가 개발한 운영체제이다.

- Windows의 버전에는 95, 98, me, XP, Vista, 7, 8, 10, 11 등이 있다.
- Windows의 주요 특징에는 GUI, 선점형 멀티 태스킹, OLE, PnP 등이 있다.

### OS의 역사

1970년대 : 업무를 수행할 때 타자기를 사용. 문서를 복사하려면 등사판(틀에 잉크 등을 발라서 찍어내는 형식, 프린터기의 조상) 이용함.

이런 불편한 업무 방식에 빌게이츠와 폴엘런은 그때 당시 있던 마이크로 컴퓨터가 미래를 위한 열쇠로 보고 마이크로소프트(Microsoft)를 설립, 개발을 시작. 목표는 모든 가정에서 컴퓨터를 사용하게 하는 것.

1974년 개발된 인텔 8080용 8비트 운영체제

메인 프레임 말고 마이크로 컴퓨터의 최초의 운영체제이며 당시 8비트 컴퓨터 시장을 완전히 점령한 운영체제이다. 이 후 16비트 컴퓨터 시장에서(특히 IBM PC 호환 기종) MS-DOS에 패배한 비운의 운영체제.

1980년대 : MS-DOS(Microsoft의 운영체제 데뷔작이자 IBM PC에 들어간 운영체제(OS)

1981년 출시된 마이크로 컴퓨터이자 IBM의 첫 개인용 컴퓨터. 최초의 16비트 PC. 역사적인 컴퓨터 모델. 이 당시 OS는 컴퓨터를 살 때 선택하는 것으로 옵션임. IBM PC가 나올 때 그 당시 8비트 컴퓨터의 표준이었던 CP/M과 협상했으나 결렬된 후, 마이크로소프트의 눈치 작전에 발 빠른 MS-DOS가 IBM PC와 함께하게 되며 대성공을 이뤄냄.(이런 이유로 CP/M이 사장됨)

1990년 이후 ~ 2020년 이전 개인용 PC는 매킨토시를 제외하고 모두 IBM PC 호환기종이라고 봐도 무방함.

즉, 통일된 이후 운영체제 빼고 컴퓨터 내부 구조와 구성품들이 거의 통일됨. IBM PC 아키텍처가 표준화 됨.

**선점형 멀티태스킹(Preemptive Multi-Tasking)** : 운영체제가 제어권을 응용 프로그램에게 부여하는 것으로 응용 프로그램이 제어권을 독점하는 것을 방지 하여 안정적인 작업 환경을 지원하는 체제를 의미하는 용어이다.

**OLE(Object Linking and Embedding)** : 다른 여러 응용 프로그램에서 작성된 문자나 그림 등의 개체(Object)를 현재 작성 중인 문서에 자유롭게 연결(Linking)하거나 삽입(Embedding)하여 편집할 수 있게 하는 기능.

**PnP(Plug and Play, 자동 감지 기능)** : 컴퓨터 시스템에 프린터나 사운드 카드 등의 하드웨어를 설치했을 때, 해당 하드웨어를 사용하는 데 필요한 시스템 환경을 운영체제가 자동으로 구성해 주는 기능.

## 4.응용 SW 기초기술 활용 - SEC\_02(Windows)

### 2) Windows의 역사

- Window 1.0(1985년) : 운영체제라기 보다는 MS-DOS의 응용프로그램으로, Apple의 Mac OS와의 계약문제 때문에 휴지통 기능과 창을 겹치는게 안되서 타일처럼 배치한게 특징.
- Window 2.0(1987년) : 창 겹치기가 가능해진 Windows 2.0은 최초로 Microsoft Word와 Microsoft Excel의 사용이 가능해졌다는 것에 의의가 있음.
- Window 3.1(1992년, 그래픽의 시대) : 본격적인 가상메모리의 사용으로 멀티태스킹 능력 강화, 그래픽 카드 성능 향상으로 화려한 화면을 제공. 쓰기 편하고 쉬운 GUI와 자체 API를 통해 윈도우 전용 프로그램을 제작할 수도 있었음. 업무용으로 제격임. Windows 3.1의 등장 이후 윈도우 탑재 컴퓨터가 됐고, IBM PC 호환 기종은 점유율 90%대를 찍게 됨.
- Windows 95(1995년) : 본격적인 디지털 시대의 시작을 알린 운영체제. 그 전까지는 MS-DOS의 그래픽 셸 프로그램이었지만 Window 95 부터는 운영체제가 됨. 오늘날도 통용되는 windows의 UI를 정립(시작메뉴, 최소화/ 최대화, 작업표시줄, 바탕화면 등) 이 운영체제를 기점으로 일반 사용자도 컴퓨터라는 물건이 보편화되기 시작함. 또한 32비트 운영 체제 시대의 서막이며, Internet Explorer를 탑재함.

## 4.응용 SW 기초기술 활용 - SEC\_02(Windows)

### 2) Windows의 역사

- Windows 98(1998년) : 소비자용으로 설계된 윈도우의 첫 번째 버전. 이 당시 직장과 가정에 보편적으로 PC가 보급되어 업무와 여가 모두 적합할 수 있는 PC로 개발됨. 인터넷을 통해 업그레이드를 배포한 첫 버전. 마지막 MS-DOS 기반 버전이며 플로피 디스크로 발매한 마지막 Windows임. 이 때 내문서 아이콘이 처음으로 바탕화면에 제공됨.
- Windows XP(2001년) : 초원이 배경화면인 2001년도 개발된 OS. 기존의 칙칙한 고전 테마에서 벗어난 컬러풀한 테마. 호환성이 가장 막강한 버전.
- Windows Vista(2006년) : 5개월 후에 Windows 7이 나오는 바람에 거의 등한시되게 됨. 그래도 망한 애들 (Vista, Me, 8, 8.1)중에 그나마 나았다는 평가.
- Windows 7(2009년) : 실패한 Vista를 기반으로 제작. 데스크톱보다 노트북 판매량이 늘어난 시대. 공공 핫스팟과 개인 네트워크 접속이 일반화 되기 시작. SSD와 터치스크린을 지원하기 시작.
- Windows 8(2013년) : 네모네모한 UI로 바뀌고 시작버튼이 없어짐. (결국 10에서 돌아왔다) 최초의 태블릿 PC 정식 지원.
- Windows 10(2015년) : 업그레이드를 고객에게 무료로 제공함. 다양한 제품군이 하나의 통합된 플랫폼을 사용하고, 하나의 스토어를 사용 해서 PC에서 사용하던 앱을 핸드폰이나 대화면에서도 사용이 가능해짐.

## 4.응용 SW 기초기술 활용 - SEC\_02(Windows)

### 3) 그래픽 사용자 인터페이스(GUI : Graphic User Interface)

; 그래픽 사용자 인터페이스는 키보드로 명령어를 직접 입력하지 않고, 마우스로 아이콘이나 메뉴를 선택하여 모든 작업을 수행하는 방식을 말한다.

- 초보자도 쉽게 사용할 수 있는 그래픽 사용자 인터페이스(GUI)를 채용하였다.

### 4) 선점형 멀티 태스킹(Preemptive Multi-Tasking)

; 선점형 멀티태스킹은 동시에 여러 개의 프로그램을 실행하는 멀티태스킹을 하면서 운영체제가 각 작업의 CPU 이용 시간을 제어하여 응용 프로그램 실행 중 문제가 발생하면 해당 프로그램을 강제 종료시키고 모든 시스템 자원을 반환하는 방식을 말한다.

- 하나의 응용 프로그램이 CPU를 독점하는 것을 방지할 수 있어 시스템 다운 현상 없이 더욱 안정적인 작업을 할 수 있다.

**멀티태스킹(Multi-Tasking, 다중 작업)** : 멀티태스킹은 여러 개의 프로그램을 동시에 열어 두고 다양한 작업을 동시에 진행하는 것을 말한다. 예를 들면, MP3 음악을 들으면서 워드 프로세서 작업을 하다 인터넷에서 파일을 다운로드 하는 것

## 4.응용 SW 기초기술 활용 - SEC\_02(Windows)

### 5) PnP(Plug and Play, 자동 감지 기능)

; PnP는 컴퓨터 시스템에 프린터나 사운드 카드 등의 하드웨어를 설치했을 때, 해당 하드웨어를 사용하는 데 필요한 시스템 환경을 운영체제가 자동으로 구성해 주는 기능이다.

- 운영체제가 하드웨어의 규격을 자동으로 인식하여 동작하게 해주므로 PC 주변장치를 연결할 때 사용자가 직접 환경을 설정하지 않아도 된다.
- PnP 기능을 활용하기 위해서는 하드웨어와 소프트웨어 모두 PnP를 지원하여야 한다.

### 6) OLE(Object Linking and Embedding)

; OLE는 다른 여러 응용 프로그램에서 작성된 문자나 그림 등의 개체(Object)를 현재 작성 중인 문서에 자유롭게 연결(Linking)하거나 삽입(Embedding)하여 편집할 수 있게 하는 기능이다.

- OLE로 연결된 이미지를 원본 프로그램에서 수정하거나 편집하면 그 내용이 그대로 해당 문서에 반영된다.



## 4.응용 SW 기초기술 활용 - SEC\_02(Windows)

### 7) 255자의 긴 파일명

; Windows에서는 파일 이름을 지정할 때 VFAT(Virtual File Allocation Table)를 이용하여 최대 255자까지 지정할 수 있다.

- 파일 이름으로는 ₩ / : \* ? " < > | 를 제외한 모든 문자 및 공백을 사용할 수 있으며, 한글의 경우 127자까지 지정할 수 있다.

### 8) Single-User 시스템

; 컴퓨터 한 대를 한 사람만이 독점해서 사용한다.

**VFAT** : Windows 95와 함께 등장했다. 기존 MS-DOS의 파일 이름은 8자 이하, 확장자가 3자 이하(예: AUTOEXEC.BAT)로 길이가 제한되어 있었는데 최대 255자까지 지원하도록 한 FAT(파일 할당 테이블)이다.

**Single-User / Multi-User** : Windows 10과 같은 개인용은 하나의 컴퓨터를 한 사람이 사용하는 Single-User 시스템이고, UNIX, LINUX, Windows NT와 같은 서버용은 하나의 컴퓨터를 동시에 여러 사람이 사용하는 Multi-user 시스템이다.

# 응용 SW 기초기술 활용 - SEC\_02(Windows) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(Windows)

1. 윈도우에서 사용자가 사용하기 원하는 하드웨어를 시스템에 부착하면 자동으로 인식하여 동작하게 해주는 기능은?

- ① Folding                      ② Plug and Play
- ③ Coalescing                ④ Naming

### PnP(Plug and Play, 자동 감지 기능)

PnP는 컴퓨터 시스템에 프린터나 사운드 카드 등의 하드웨어를 설치했을 때, 해당 하드웨어를 사용하는데 필요한 시스템 환경을 운영체제가 자동으로 구성해 주는 기능이다.

- 운영체제가 하드웨어의 규격을 자동으로 인식하여 동작하게 해주므로 PC 주변장치를 연결할 때 사용자가 직접 환경을 설정하지 않아도 된다.

- PnP 기능을 활용하기 위해서는 하드웨어와 소프트웨어 모두 PnP 기능을 지원하여야 한다.

**Folding(폴딩)**이란 특정 코드 영역을 숨기는 기능을 의미한다.

**통합(Coalescing)**이란 기억공간 내에 인접한 둘 이상의 공백을 그들 사이의 경계를 없애고 하나의 공백으로 합치는 것을 의미한다.

**Naming(네이밍)**이란 명명 규칙을 말한다.

3. Windows에 대한 설명 중 옳지 않은 것은?

- ① PnP 기능을 지원한다.
- ② 멀티태스킹을 지원한다.
- ③ 네트워크 기능이 강화되었다.
- ④ 멀티 유저 시스템이다.

**Single-User / Multi-User** : Windows 10과 같은 개인용은 하나의 컴퓨터를 한 사람이 사용하는 Single-User 시스템이고, UNIX, LINUX, WINDOW NT계열과 같은 서버용은 하나의 컴퓨터를 동시에 여러 사람들이 사용하는 Multi-User 시스템이다.

**멀티태스킹(Multi-Tasking, 다중 작업)** : 멀티태스킹은 여러 개의 프로그램을 동시에 실행하고 다양한 작업을 동시에 진행하는 것을 의미한다.

4. 컴퓨터의 윈도우 창에 여러 윈도우를 열어 놓고 작업하는 것을 기억장치 처리 방법으로 무엇이라 하는가?

- ① 보조 프로그램
- ② 싱글 프로세싱
- ③ 멀티 프로그래밍
- ④ 리얼 타임 프로그래밍

# 응용 SW 기초기술 활용 - SEC\_02(Windows) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(Windows)

5. Windows에서는 파일 이름을 지정할 때 최대 255자까지 지정할 수 있게 하려면 무슨 기능을 이용하는가?

- ① FAT12                      ② FAT16
- ③ VFAT                        ④ NTFS

### 255자의 긴 파일명

Windows에서는 파일 이름을 지정할 때 VFAT(Virtual File Allocation Table)를 이용하여 최대 255자 까지 지정할 수 있다.

- 단, 파일 이름으로는 ₩ / : \* ? " < > | 를 제외한 모든 문자 및 공백을 사용할 수 있으며, 한글의 경우 127자까지 지정할 수 있다.

FAT12 : 1980년, MS-DOS의 전신인 86-DOS에서 사용하기 위해 CP/M

의 FAT를 8비트에서 12비트로 확장하고 여러 가지 정보를 추가하여 만들어진 것을 의미한다.

FAT16 : 1984년 IBM PC AT의 20MB 하드 디스크 드라이브를 지원하기 위해서 마이크로소프트사가 MS-DOS 3.0과 함께 등장했다. 16비트 클러스터를 사용한다.

NTFS : 마이크로소프트사의 Window의 파일 시스템으로 MS-

# 4.응용 SW 기초기술 활용 - SEC\_03(UNIX / LINUX / MacOS)

## 1) UNIX의 개요 및 특징

; UNIX는 1960년대 AT&T, 벨(Bell) 연구소, MIT, General Electric이 공동 개발한 운영체제이다.

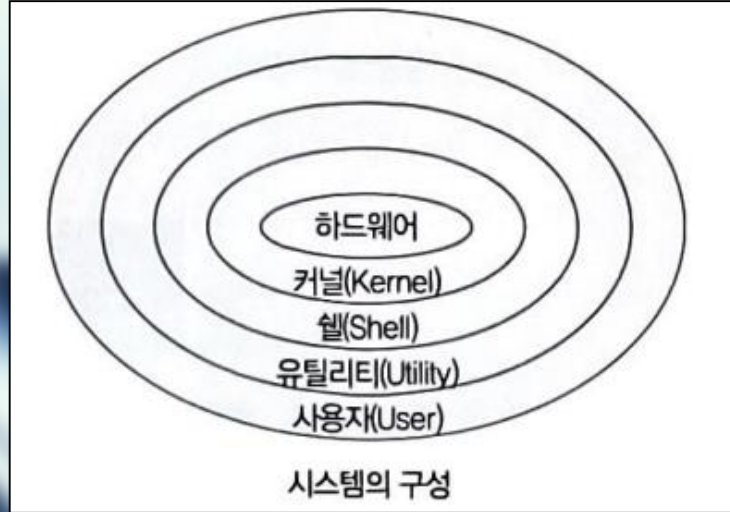
- 시분할 시스템(Time Sharing System)을 위해 설계된 대화식 운영체제로, 소스가 공개된 개방형 시스템(Open System)이다.
- 대부분 C 언어로 작성되어 있어 이식성이 높으며 장치, 프로세스 간의 호환성이 높다.
- 크기가 작고 이해하기가 쉽다.
- 다중 사용자(Multi-User), 다중 작업(Multi-Tasking)을 지원한다.
- 많은 네트워킹 기능을 제공하므로 통신망(Network) 관리용 운영체제로 적합하다.
- 트리 구조의 파일 시스템을 갖는다.
- 전문적인 프로그램 개발에 용이하다.
- 다양한 유틸리티 프로그램들이 존재한다.

**시분할 시스템(time-sharing)** : 컴퓨터를 대화식으로 사용하려는 시도에서 탄생하였다. 시분할 운영 체제는 CPU 스케줄링과 다중 프로그래밍을 이용해서 각 사용자들에게 컴퓨터 자원을 시간적으로 분할하여 사용할 수 있게 해 준다.

**다중 사용자(Multi-User)**는 여러 사용자가 동시에 시스템을 사용하는 것이고, **다중 작업(Multi-Tasking)**은 여러 개의 작업이나 프로그램을 동시에 수행하는 것을 의미한다.하나 이상의 작업을 백그라운드에서 수행하므로 여러 작업을 동시에 처리할 수 있다. 여러 개의 작업이 동시에 실행될 때 전면에서 실행되는 우선순위가 높은 작업을 **포그라운드 작업**이라 하고, 같은 상황에서 우선순위가 낮아 화면에 보이지 않고 실행되는 프로그램을 **백그라운드 작업**이라 합니다.

# 4.응용 SW 기초기술 활용 - SEC\_03(UNIX / LINUX / MacOS)

## 2) UNIX 시스템의 구성



### 커널(Kernel)

- UNIX의 가장 핵심적인 부분이다.
- 컴퓨터가 부팅될 때 주기억장치에 적재된 후 상주하면서 실행된다.
- 하드웨어를 보호하고, 프로그램과 하드웨어 간의 인터페이스 역할을 담당한다.
- 프로세스(CPU 스케줄링) 관리, 기억장치 관리, 파일 관리, 입·출력 관리, 프로세스간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행한다.

# 4.응용 SW 기초기술 활용 - SEC\_03(UNIX / LINUX / MacOS)

## 2) UNIX 시스템의 구성

### 셸(Shell)

- 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기이다.
- 시스템과 사용자 간의 인터페이스를 담당한다.
- DOS의 COMMAND.COM과 같은 기능을 수행한다.
- 주기억장치에 상주하지 않고, 명령어가 포함된 파일 형태로 존재하며 보조 기억장치에서 교체 처리가 가능하다.
- 파이프라인 기능을 지원하고 입·출력 재지정을 통해 출력과 입력의 방향을 변경할 수 있다.
- 공용 Shell(Bourne Shell, C Shell, Korn Shell)이나 사용자 자신이 만든 Shell을 사용할 수 있다.

**COMMAND.COM** : 도스와 윈도우 95, 98, ME 등을 기본으로 하는 운영 체제 셸의 파일 이름으로 명령 줄 해석기라 부른다. 시동 직후 첫 프로그램이 실행되면 AUTOEXEC.BAT 구성 파일을 실행하여 시스템을 설정할 책임을 가지고 뒤따르는 다른 과정을 밟게 된다.

**파이프라인** : 둘 이상의 명령을 함께 묶어 처리한 결과를 다른 명령의 입력으로 전환하는 기능이다.

**Bourne Shell** : 벨 연구소의 스티브 본(Stephen Bourne) 개발하였고, 많은 셸 스크립트의 기반이 되는 셸이다.

**C Shell** : C언어 구문과 유사하며, Bourne Shell을 확장하여 히스토리, 작업제어, 엘리어스 등 기능 추가하여 개발자들에게 유용한 기능들을 제공한다.

**Korn Shell** : Bourne Shell과 호환되며 C Shell의 많은 기능을 포함하고 있으며, Unix 계열에서 많이 사용된다.

# 4.응용 SW 기초기술 활용 - SEC\_03(UNIX / LINUX / MacOS)

## 2) UNIX 시스템의 구성

### Utility Program

- 일반 사용자가 작성한 응용 프로그램을 처리하는 데 사용한다.
- DOS에서의 외부 명령어에 해당된다.
- 유틸리티 프로그램에는 에디터, 컴파일러, 인터프리터, 디버거 등이 있다.

### UNIX에서의 프로세스 간 통신

; 각 프로세스는 시스템 호출을 통해 커널의 기능을 사용하며, 프로세스 간 통신은 시그널, 파이프 소켓 등을 사용한다.

- 시그널(Signal) : 간단한 메시지를 이용하여 통신하는 것으로 초기 UNIX 시스템에서 사용됨
- 파이프(Pipe) : 한 프로세스의 출력이 다른 프로세스의 입력으로 사용되는 단방향 통신 방식
- 소켓(Socket) : 프로세스 사이의 대화를 가능하게 하는 쌍방향 통신 방식

**프로세스 간의 통신(IPC; Inter Process Communication)** : 프로세스 간의 통신은 여러 프로세스 즉 프로그램 간에 서로 데이터를 주고받는 것을 의미한다.

**시스템 호출** : 프로세스가 커널에 접근하기 위한 인터페이스를 제공하는 명령어이다.



## 4.응용 SW 기초기술 활용 - SEC\_03(UNIX / LINUX / MacOS)

### 3) LINUX의 개요 및 특징

; LINUX는 1991년 리누스 토발즈(Linus Torvalds)가 UNIX를 기반으로 개발한 운영체제이다.

- 프로그램 소스 코드가 무료로 공개되어 있기 때문에 프로그래머가 원하는 기능을 추가할 수 있고, 다양한 플랫폼에 설치하여 사용이 가능하며, 재배포가 가능하다.
- UNIX와 완벽하게 호환된다.
- 대부분의 특징이 UNIX와 동일하다.

### 4) MacOS의 개요 및 특징

; MacOS는 1980년대 애플(Apple)사가 UNIX를 기반으로 개발한 운영체제이다.

- 아이맥(iMac)과 맥북(MacBook) 등 애플사에서 생산하는 제품에서만 사용이 가능하다.
- 드라이버 설치 및 install과 uninstall의 과정이 단순하다.

## 4.응용 SW 기초기술 활용 - SEC\_03(UNIX / LINUX / MacOS)

### 5) 파일 디스크립터(File Descriptor, 파일 서술자)

; 파일을 관리하기 위한 시스템(운영체제)이 필요로 하는 파일에 대한 정보를 가진 제어 블록을 의미하며, 파일 제어 블록(FCB; File Control Block)이라고도 한다.

- 파일 디스크립터는 파일마다 독립적으로 존재하며, 시스템에 따라 다른 구조를 가질 수 있다.
- 보통 파일 디스크립터는 보조기억장치 내에 저장되어 있다가 해당 파일이 Open될 때 주기억장치로 옮겨진다.
- 파일 디스크립터는 파일 시스템이 관리하므로 사용자가 직접 참조할 수 없다.

# 응용 SW 기초기술 활용 - SEC\_03(UNIX / LINUX / MacOS) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(UNIX / LINUX / MacOS)

### 1. 파일 디스크립터(File Descriptor)에 대한 설명으로 틀린 것은?

- ① 파일 관리를 위해 시스템이 필요로 하는 정보를 가지고 있다.
- ② 보조기억장치에 저장되어 있다가 파일이 개방(Open)되면 주기억장치로 이동된다.
- ③ 사용자가 파일 디스크립터를 직접 참조할 수 있다.
- ④ 파일 제어 블록(File Control Block)이라고도 한다.

### 파일 디스크립터(File Descriptor, 파일 서술자)

파일을 관리하기 위한 시스템(운영체제)이 필요로 하는 파일에 대한 정보를 가진 제어 블록을 의미하며, 파일 제어 블록 (FCB; File Control Block)이라고도 한다.

- 파일 디스크립터는 파일마다 독립적으로 존재하며, 시스템에 따라 다른 구조를 가질 수 있다.
- 보통 파일 디스크립터는 보조기억장치 내에 저장되어 있다가 해당 파일이 Open될 때 주기억장치로 옮겨진다.
- 파일 디스크립터는 파일 시스템이 관리하므로 사용자가 직접 참조할 수 없다.

### 2. 운영체제에서 커널의 기능이 아닌 것은?

### 3. UNIX의 셸(Shell)에 관한 설명으로 옳지 않은 것은?

- ① 명령어 해석기이다.
  - ② 시스템과 사용자 간의 인터페이스를 담당한다.
  - ③ 여러 종류의 셸이 있다.
  - ④ 프로세스, 기억장치, 입출력 관리를 수행한다.
- 프로세스, 기억장치, 입출력 관리를 수행하는 것은 커널이다.

### 셸(Shell)

- 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기이다.
- 시스템과 사용자 간의 인터페이스를 담당한다.
- DOS의 COMMAND.COM과 같은 기능을 수행한다.
- 주기억장치에 상주하지 않고, 명령어가 포함된 파일 형태로 존재 하며 보조 기억장치에서 교체 처리가 가능하다.
- 파이프라인 기능을 지원하고 입·출력 재지정을 통해 출력과 입력의 방향을 변경할 수 있다.
- 공용 Shell(Bourne Shell, C Shell, Korn Shell)이나 사용자 자신이 만든 Shell을 사용할 수 있다.

### 4. UNIX 운영체제에 관한 특징으로 틀린 것은?

## 응용 SW 기초기술 활용 - SEC\_03(UNIX / LINUX / MacOS) 기출 및 출제 예상 문제

### 기출 및 출제 예상 문제(UNIX / LINUX / MacOS)

5. UNIX 시스템에서 커널에 대한 설명으로 옳지 않은 것은?

- ① UNIX 시스템의 중심부에 해당한다.
- ② 사용자와 시스템 간의 인터페이스를 제공한다.
- ③ 프로세스 관리, 기억장치 관리 등을 담당한다.
- ④ 하드웨어를 캡슐화한다.

사용자와 시스템 간의 인터페이스를 제공하는 것은 셸이다.

프로그램과 하드웨어 간의 인터페이스 역할을 담당한다.

6. UNIX 시스템의 특징으로 옳지 않은 것은?

- ① 대화식 운영체제이다.
- ② 소스가 공개된 개방형 시스템이다.
- ③ 멀티 유저, 멀티 태스킹을 지원한다.
- ④ 효과적으로 구현할 수 있는 이중 리스트 구조를 사용한다.

UNIX는 트리(Tree) 구조를 사용한다.

7. UNIX 시스템의 셸(Shell)의 주요 기능에 대한 설명이 아닌 것은?

- ① 사용자 명령을 해석하고 커널로 전달하는 기능을 제공한다.
- ② 반복적인 명령 프로그램을 만드는 프로그래밍 기능을 제공한다.
- ③ 셸 프로그램 실행을 위해 프로세스와 메모리를 관리한다.
- ④ 초기화 파일을 이용해 사용자 환경을 설정하는 기능을 제공한다.

프로세스와 메모리를 관리하는 것은 커널(Kernel)의 기능이다.

8. 유틸리티 프로그램에는 해당하지 않는 것은?

- ① 에디터
- ② 컴파일러
- ③ 인터프리터
- ④ 레지스터

레지스터는 컴퓨터의 프로세서 내에서 자료를 보관하는 아주 빠른 기억 장소이다. 일반적으로 현재 계산을 수행중인 값을 저장하는 데 사용된다. 대부분의 현대 프로세서는 메인 메모리에서 레지스터로 데이터를 옮겨와 데이터를 처리한 후 그 내용을 다시 레지스터에서 메인 메모리로 저장하는 방법(로드-스토어 설계)을 사용하고 있다.

레지스터는 메모리 계층의 최상위에 위치하며, 가장 빠른 속도로 접근 가능한 메모리이다. 최신 프로세서에서 레지스터는 대개 레지스터 파일로

# 응용 SW 기초기술 활용 - SEC\_03(UNIX / LINUX / MacOS) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(UNIX / LINUX / MacOS)

9. 1991년 리누스 토발즈(Linus Torvalds)가 UNIX를 기반으로 개발한 운영체제는 무엇인가?

- ① UNIX                      ② LINUX
- ③ MacOS                    ④ Windows

LINUX는 1991년 리누스 토발즈(Linus Torvalds)가 UNIX를 기반으로 개발한 운영체제이다.

- 프로그램 소스 코드가 무료로 공개되어 있기 때문에 프로그래머가 원하는 기능을 추가할 수 있고, 다양한 플랫폼에 설치하여 사용이 가능하며, 재배포가 가능하다.
- UNIX와 완벽하게 호환된다.
- 대부분의 특징이 UNIX와 동일하다.

10. 애플(Apple)사가 UNIX를 기반으로 개발한 운영체제는 무엇 인가?

- ① LINUX                    ② UNIX
- ③ Windows                ④ MacOS

MacOS는 1980년대 애플(Apple)사가 UNIX를 기반으로 개발한 운영체제이다.

- 아이맥(iMac)과 맥북(MacBook) 등 애플사에서 생산하는

11. UNIX에서의 프로세스 간 통신에 사용되는 수단이 아닌 것은?

- ① 시그널(Signal)            ② 파이프(Pipe)
- ③ 스레드(Thread)            ④ 소켓(Socket)

스레드(Thread)는 하나의 프로세스에 속하는 작업 단위를 말한다.

각 프로세스는 시스템 호출을 통해 커널의 기능을 사용하며, 프로세스 간 통신은 시그널, 파이프, 소켓 등을 사용한다.

- 시그널(Signal) : 간단한 메시지를 이용하여 통신하는 것으로 초기 UNIX 시스템에서 사용됨
- 파이프(Pipe) : 한 프로세스의 출력이 다른 프로세스의 입력으로 사용되는 단방향 통신 방식
- 소켓(Socket) : 프로세스 사이의 대화를 가능하게 하는 쌍방향 통신 방식

12. UNIX는 대부분 어떤 언어로 작성되어 있는가?

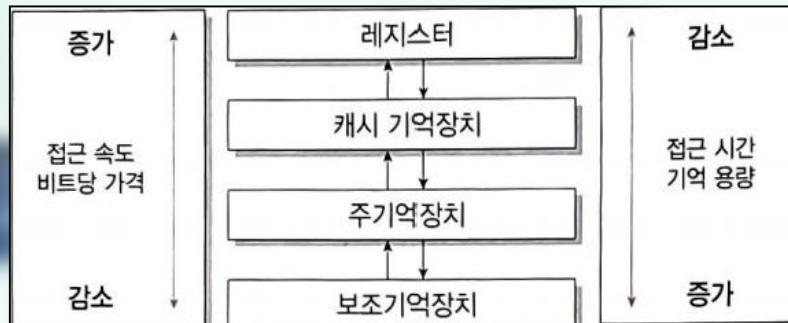
- ① JAVA                      ② Python
- ③ C++                      ④ C

대부분 C 언어로 작성되어 있어 이식성이 높으며 장치, 프로세스 간의 호환성이 높다.

## 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

### 1) 기억장치 계층 구조의 특징

; 기억장치는 레지스터, 캐시 기억장치, 주기억장치, 보조기억장치를 다음과 같이 계층 구조로 분류할 수 있다.



- 계층 구조에서 상위의 기억장치일수록 접근 속도와 접근 시간이 빠르지만, 기억용량이 적고, 고가이다.
- 주기억장치는 각기 자신의 주소를 갖는 바이트들로 구성되어 있으며, 주소를 이용하여 액세스할 수 있다.
- 레지스터, 캐시기억장치, 주기억장치의 프로그램과 데이터는 CPU가 직접 액세스 할 수 있으나 보조 기억장치에 있는 프로그램이나 데이터는 직접 액세스할 수 없다.
- 보조기억장치에 있는 데이터는 주기억장치에 적재된 후 CPU에 의해 액세스될 수 있다.

**레지스터** : 컴퓨터의 프로세서 내에서 자료를 보관하는 아주 빠른 기억 장소이다. 일반적으로 현재 계산을 수행중인 값을 저장 하는데 사용된다. CPU 내부에 존재하며 연산제어 및 디버깅 목적으로 사용된다.

**캐시 메모리** : CPU의 처리 속도와 주기억장치의 접근 속도 차이를 줄이기 위해 사용한다. 데이터 지역성을 활용하여 메인 메모리에 있는 데이터를 캐시 메모리에 불러와 두고, 프로세서가 필요한 데이터를 캐시 메모리에서 먼저 찾도록 하면 시스템 성능을 향상할 수 있다. 주기억장치와 CPU사이에 위치하며, 자주 사용하는 프로그램과 데이터를 기억한다.

# 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

## 2) 주기억장치에 저장되는 프로그램

; 주기억장치에 저장되는 프로그램은 응용프로그램과 시스템 프로그램이 있다.

- 응용 프로그램은 실행될 때만 주기억장치에 저장되었다가, 수행이 종료되면 다른 프로그램으로 대체되거나 삭제되고 전원이 꺼지면 삭제된다.
- 시스템 프로그램은 컴퓨터가 구동을 시작해서 종료될 때까지 주기억장치에 유지한다.

### ① 사용자 응용프로그램 영역

- 일반 프로그램이 기억되는 곳, 시스템 프로그램의 제어에 의해 동작한다.
- 여러 부분으로 분할해 독립된 프로그램을 기억한다.
- 다중 프로그래밍 방식으로 동작한다.
- 운영체제가 사용자 응용프로그램 각각의 독립된 영역을 보호해주는 기억 보호를 수행한다.

### ② 시스템 프로그램 영역

- 운영체제가 저장되는 곳을 상주 구역과 비상주 구역을 분류된다.
- 상주 구역 : 언제라도 바로 실행될 수 있는 운영체제의 기본적인 기능 저장공간을 의미하며, 자주 사용되는 프로그램들이 기억되는 곳이다.
- 비상주 구역 : 자주 사용되지 않는 프로그램을 필요할 때만 보조기억장치에서 인출해 저장하며, 처리가 끝나면 다른 프로그램이 다시 사용할 수 있는 구역



## 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

### 3) 사용자 응용프로그램 영역을 효율적으로 사용하기 위한 고려 사항

- 주기억장치에 한 번에 몇 개의 프로그램을 적재할 것인지 고려하여야 한다.
  - 한 개의 프로그램만 가능하게 하거나 여러 개의 프로그램을 함께 공존하는 것을 고려한다.
- 여러 개의 프로그램을 함께 적재할 때(각 프로그램에 할당되는 공간의 크기)
  - 동일하게 할지 다르게 할지 고려
- 일정한 크기의 공간 할당할 지 고려한다.
  - 해당 크기를 유지할지 상황에 따라 변경할지를 고려한다.
  - 연속한 작은 공간들을 할당할지 한 덩어리의 커다란 공간을 할당할지를 고려한다.

## 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

### 4) 주기억장치를 할당하는 방법

#### ① 단일 사용자 할당 기법

; 운영체제가 차지하는 부분을 제외한 나머지 기억 공간을 한 사용자 응용프로그램이 독점 사용하도록 하는 기법을 의미한다.

##### 장점

- 사용자에게 융통성을 최대한 제공
- 최대의 단순성과 최소의 비용을 만족
- 특별한 하드웨어가 없어도 되며, 운영체제 소프트웨어도 필요 없다.

##### 단점

- 사용자가 사용하는 이외의 부분은 낭비가 될 수 있다.
- 입력과 출력을 수행하는 동안 중앙처리장치를 계속 쓸 수 없어 유휴 상태가 되므로 활용도가 매우 낮다.
- 프로그램이 주기억장치의 용량보다 크면 수행시키기 어렵다.

# 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

## 4) 주기억장치를 할당하는 방법

### ② 고정 분할 할당 기법

; 고정 크기의 분할된 구역을 고정 할당하는 방법을 의미한다.

#### 장점

- 한 프로그램이 적재되고 남은 공간에 다른 프로그램을 적재하여 수행한다.
- 프로세서와 기억장치와 같은 자원의 활용도를 크게 향상된다.
- 동시에 여러 프로그램을 주기억장치에 적재하여 수행하는 다중 프로그래밍 기법이 가능하다.

#### 단점

- 할당되는 저장 공간이 작고 저장될 프로그램이 클 경우 프로그램이 작은 단위로 쪼개지는 단편화 문제가 발생한다.
- 프로그램과 할당된 분할 구역의 크기가 일치하지 않으면 남은 공간이 생긴다.

**단편화** : 주기억장치의 분할된 영역에 프로그램이나 데이터를 할당할 경우, 분할된 영역이 프로그램이나 데이터보다 작거나 커서 생기는 빈 기억 공간을 의미한다.

•**내부 단편화** : 분할된 영역이 할당될 프로그램의 크기보다 크기 때문에 프로그램이 할당된 후 사용되지 않고 남아 있는 빈 공간

•**외부 단편화** : 분할된 영역이 할당될 프로그램의 크기보다 작기 때문에 프로그램이 할당될 수 없어 사용되지 않고 빈 공간으로 남아있는 분할된 전체 영역

## 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

### 4) 주기억장치를 할당하는 방법

#### ③ 가변 분할 할당 기법

첫 번째 방법으로는 기억 장소의 집약을 만든다. 주기억장치를 검사하여 공백을 존재하는 빈 영역을 하나의 커다란 빈 영역을 만드는 방법이며, 운영체제는 사용 중인 블록을 한 곳에 모으고, 비어 있는 기억 장소를 하나의 커다란 공백으로 만든다.

#### 장점

- 사용 가능한 큰 영역을 만들어 낭비를 줄일 수 있다.

#### 단점

- 기억 장소를 집약하는 동안 시스템은 지금까지 수행해오던 일들을 일단 중지해야 하며, 집약하는데 시간적으로 소모가 된다.
- 수행 중이던 프로그램과 데이터를 주기억장치 내의 다른 장소로 이동시키기 때문에 각 위치 수정이 불가피하다.

## 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

### 4) 주기억장치를 할당하는 방법

#### ③ 가변 분할 할당 기법

두 번째 방법으로는 배치 전략을 의미하는 공백 영역을 찾는 알고리즘이다.

- 최초 적합 방법 : 여러 공백을 차례로 검색하다가 새로운 프로그램을 저장할 수 있을 만큼의 크기를 가진 공간을 최초로 찾으면 그 공간에 할당한다.
- 최적 적합 방법 : 요구하는 크기보다 크면서 가장 크기가 비슷한 공간을 선택하여 할당하며 매우 작은 공백만 생긴다는 장점이 있다.
- 최악 적합 방법 : 큰 부분을 찾아 할당한다. 프로그램이 할당되고 남은 공간이 비교적 크다면, 나머지 부분을 다른 프로그램에 할당하여 사용한다.

## 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

### 5) 기억장치의 관리 전략의 개요

; 기억장치의 관리 전략은 보조기억장치의 프로그램이나 데이터를 주기억장치에 적재시키는 시기, 적재 위치 등을 지정하여 **한정된 주기억장치의 공간을 효율적으로 사용하기 위한 것으로 반입(Fetch) 전략, 배치(Placement) 전략, 교체(Replacement) 전략**이 있다.

### 6) 반입(Fetch) 전략

; 반입 전략은 보조기억장치에 보관중인 프로그램이나 데이터를 언제 주기억장치로 적재할 것인지를 결정하는 전략이다.

- **요구 반입(Demand Fetch)** : 실행중인 프로그램이 특정 프로그램이나 데이터 등의 참조를 요구할 때 적재하는 방법이다.
- **예상 반입(Anticipatory Fetch)** : 실행중인 프로그램에 의해 참조될 프로그램이나 데이터를 미리 예상하여 적재하는 방법이다.

## 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

### 7) 배치(Placement) 전략

; 배치 전략은 새로 반입되는 프로그램이나 데이터를 주기억장치의 어디에 위치시킬 것인지를 결정하는 전략이다.

- **최초 적합(First Fit)** : 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 첫 번째 분할 영역에 배치시키는 방법
- **최적 적합(Best Fit)** : 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 작게 남기는 분할 영역에 배치시키는 방법
- **최악 적합(Worst Fit)** : 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 많이 남기는 분할 영역에 배치시키는 방법

**단편화** : 주기억장치의 분할된 영역에 프로그램이나 데이터를 할당할 경우, 분할된 영역이 프로그램이나 데이터보다 작거나 커서 생기는 빈 기억 공간을 의미한다.

•**내부 단편화** : 분할된 영역이 할당될 프로그램의 크기보다 크기 때문에 프로그램이 할당된 후 사용되지 않고 남아 있는 빈 공간

•**외부 단편화** : 분할된 영역이 할당될 프로그램의 크기보다 작기 때문에 프로그램이 할당될 수 없어 사용되지 않고 빈 공간으로 남아있는 분할된 전체 영역



## 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

### 7) 배치(Placement) 전략

예제) 기억장치 상태가 다음 표와 같다. 기억장치 관리 전략으로 First Fit, Best Fit, Worst Fit 방법을 사용하려 할 때, 각 방법에 대하여 10K의 프로그램이 할당 받게 되는 영역의 번호는?

영역 번호	영역 크기	상태
1	5K	공백
2	14K	공백
3	10K	사용 중
4	12K	공백
5	16K	공백

- ① 먼저 10K가 적재될 수 있는지 각 영역의 크기를 확인한다.
- ② First Fit : 빈 영역 중에서 10K의 프로그램이 들어갈 수 있는 첫 번째 영역은 2번이다.
- ③ Best Fit : 빈 영역 중에서 10K 프로그램이 들어가고 단편화를 가장 작게 남기는 영역은 4번이다.
- ④ Worst Fit : 빈 영역 중에서 10K 프로그램이 들어가고 단편화를 가장 많이 남기는 영역은 5번이다.

## 4.응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요)

### 8) 교체(Replacement) 전략

; 교체 전략은 주기억장치의 모든 영역이 이미 사용중인 상태에서 새로운 프로그램이나 데이터를 주기억장치에 배치하려고 할 때, 이미 사용되고 있는 영역 중에서 어느 영역을 교체하여 사용할 것인지를 결정하는 전략이다.

- 교체 전략에는 FIFO(First Input First Output), OPT(Optimal replacement), LRU(Least Recently Used), LFU(Least Frequency Used), NUR(Not Used Recently), PFF(Page Fault Frequency), SCR(Second Chance Replacement) 등이 있다.

프로그램을 일정한 크기로 나눈 단위를 **페이지(Page)**라고 하고, 페이지 크기로 일정하게 나누어진 주기억장치의 단위를 **페이지 프레임(Page Frame)**이라고 한다.

- OPT(Optimal Replacement)** : 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체한다.
  - FIFO(First Input First Output)** : 주기억장치 내에 가장 오래된 페이지를 교체한다.
  - LRU(Least Recently Used)** : 사용(참조된) 지 가장 오래된 페이지를 교체한다.
  - LFU(Least Frequency Used)** : 사용(참조된) 횟수가 가장 적은 페이지를 교체한다.
  - NUR(Not Used Recently)** : 참조 비트와 변형 비트를 사용하여 교체한다.(최근 사용하지 않은 페이지를 제거)
  - PFF(Page Fault Frequency)** : 워킹 셋에 속하지 않은 페이지 중에 최근에 자주 사용하는 페이지를 교체한다.
  - SCR(Second Chance Replacement)** : 가장 오래 동안 주기억장치에 있던 페이지중 자주 사용되는 페이지의 교체를 방지하기 위한 것으로, FIFO 기법의 단점을 보완하는 기법이다.
- 워킹 셋은 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합이다

# 응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(기억장치 관리의 개요)

1. 기억장치 관리 전략은 주기억장치 자원을 가장 잘 사용하도록 설계되어야 하는데, 다음 중 주기억장치 관리 전략과 거리가 먼 것은?

- ① Fetch 전략
- ② Placement 전략
- ③ Paging 전략
- ④ Replacement 전략

**페이징(Paging)**기법은 **가상기억장치(Virtual Memory)**에 보관되어 있는 프로그램과 주기억장치의 영역을 동일한 크기로 나눈 후 나뉜 프로그램(페이지)을 동일하게 나뉜 주기억장치의 영역(페이지 프레임)에 적재시켜 실행하는 기법이다.

**가상기억장치(Virtual Memory)**라는 것은 보조기억장치(SSD, HDD 등) 일부를 마치 주기억장치(RAM)처럼 사용하는 것으로 용량이 작은 주기억장치를 큰 용량을 가진 것처럼 사용하는 기법이다.

아울러 프로그램을 여러 개의 작은 블록 단위로 나누어서 가상기억장치에 보관해놓고, 프로그램 실행 시 요구되는 블록만 주기억장치에 불연속으로 할당하여 처리한다.

**반입(Fetch) 전략**

3. 기억공간이 15K, 23K, 22K, 21K 순으로 빈 공간이 있을 때 기억장치 배치 전략으로 "First Fit" 을 사용하여 17K의 프로그램을 적재할 경우 내부 단편화의 크기는 얼마인가?

- ① 5K                      ② 6K
- ③ 7K                      ④ 8K

4. 빈 기억공간의 크기가 20KB, 16KB, 8KB, 40KB 일 때 기억장치 배치 전략으로 "Best Fit"을 사용하여 17KB의 프로그램을 적재할 경우 내부 단편화의 크기는 얼마인가?

- ① 3KB                      ② 23KB
- ③ 64KB                    ④ 67KB

# 응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(기억장치 관리의 개요)

5. 메모리 관리 기법 중 Worst Fit 방법을 사용할 경우 10K 크기 의 프로그램 실행을 위해서는 어느 부분에 할당되는가?

영역 번호	메모리 크기	사용 여부
NO.1	8K	FREE
NO.2	12K	FREE
NO.3	10K	IN USE
NO.4	20K	IN USE
NO.5	16K	FREE

- ① NO.2                      ② NO.3
- ③ NO.4                      ④ NO.5

**최악 적합(Worst Fit) :** 반입되는 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 많이 남기는 분할 영역에 배치시키는 방법이다.

**단편화 :** 주기억장치의 분할된 영역에 프로그램이나 데이터를 할당할 경우, 분할된 영역이 프로그램이나 데이터보다 작거나 커서 생기는 빈 기억공간을 의미한다.

● **내부 단편화 :** 분할된 영역이 할당될 프로그램의 크기보다 크기 때문에 프로그램이 할당된 후 사용되지 않고 남아 있는 빈 공간을

7. 150K의 작업요구 시 First Fit과 Best Fit 전략을 각각 적용할 경우, 할당 영역의 연결이 옳은 것은?

할당영역	운영체제
1	50k
	사용중
2	400k
	사용중
3	200k

- ① First Fit : 2, Best Fit : 3
- ② First Fit : 3, Best Fit : 2
- ③ First Fit : 1, Best Fit : 2
- ④ First Fit : 3, Best Fit : 1

150KB 작업을 최초 적합(First Fit)으로 할당할 경우는 150KB의 데이터 나 프로그램을 저장할 공간이 150KB 이상이 되어야 되며, 아울러 그 중 에서 제일 첫 번째 발견되는 할당영역에 배치하게 되며 여기서는 2번에 해당하는 400KB 공백에 저장되며, 250KB만큼 내부 단편화가 생긴다. 최적 적합(Best Fit)으로 할당할 때는 150KB이상이며 그 중에 가장 작은 내부 단편화가 이루어져야 하므로 200KB에 저장되며 아울러 50KB 의 내부 단편화가 이루어진다.

# 응용 SW 기초기술 활용 - SEC\_04(기억장치 관리의 개요) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(기억장치 관리의 개요)

9. 운영체제가 차지하는 부분을 제외한 나머지 기억 공간을 한 사용자 응용프로그램이 독점 사용하도록 하는 기법은 무엇인가?

- ① 사용자 할당 기법                      ② 고정 분할 할당 기법
- ③ 가변 분할 할당 기법                  ④ 동적 할당 기법

### 단일 사용자 할당 기법

운영체제가 차지하는 부분을 제외한 나머지 기억 공간을 한 사용자 응용 프로그램이 독점 사용하도록 하는 기법을 의미한다.

#### 장점

- 사용자에게 융통성을 최대한 제공
- 최대의 단순성과 최소의 비용을 만족
- 특별한 하드웨어가 없어도 되며, 운영체제 소프트웨어도 필요 없다.

#### 단점

- 사용자가 사용하는 부분 이외의 부분은 낭비가 될 수 있다.
- 입력과 출력을 수행하는 동안 중앙처리장치를 계속 쓸 수 없어 유휴 상태가 되므로 활용도가 매우 낮다.
- 프로그램이 주기억장치의 용량보다 크면 수행시키기

11. 기억장치의 분류에 속하지 않는 것은?

- ① 레지스터                                      ② 캐시
- ③ 주기억장치                                    ④ 스위치

기억장치에는 레지스터, 캐시 기억장치, 주기억장치, 보조기억장치가 다음과 같은 계층 구조로 분류할 수 있다.

### 레지스터

### 캐시 기억장치

### 주기억장치

### 보조기억장치

- 계층 구조에서 상위의 기억장치일수록 접근 속도와 접근 시간이 빠르지만, 기억용량이 적고, 고가이다.
- 주기억장치는 각기 자신의 주소를 갖는 바이트들로 구성되어 있으며, 주소를 이용하여 액세스를 할 수 있다.
- 레지스터, 캐시 기억장치, 주기억장치의 프로그램과 데이터는 CPU가 직접 액세스를 할 수 있으나 보조 기억장치에 있는 프로그램이나 데이터는 직접 액세스할 수 없다.
- 보조기억장치에 있는 데이터는 주기억장치에 적재된 후 CPU에 의해 액세스가 되어 실행될 수 있다.

## 4.응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법)

### 1) 주기억장치 할당의 개념

; 주기억장치 할당 기법은 프로그램이나 데이터를 실행시키기 위해 주기억장치에 어떻게 할당할 것인지에 대한 내용이며, 연속 할당 기법과 분산 할당 기법으로 분류할 수 있다.

연속 할당 기법	프로그램을 주기억장치에 연속으로 할당하는 기법으로, 단일 분할 할당 기법과 다중 분할 할당 기법이 있다. • 단일 분할 할당 기법 : 오버레이, 스와핑 • 다중 분할 할당 기법 : 고정 분할 할당 기법, 동적 분할 할당 기법
분산 할당 기법	프로그램을 특정 단위의 조각으로 나누어 주기억장치 내에 분산하여 할당하는 기법으로, 페이징 기법과 세그멘테이션 기법으로 나눌 수 있다.

**분산 할당 기법** : 가상기억장치의 내용을 주기억장치에 할당하기 위한 기법으로, 가상기억장치 관리 기법이라고도 한다.

**페이징(Paging)기법** : 가상기억장치에 보관되어 있는 프로그램과 주기억장치의 영역을 동일한 크기로 나눈 후 나뉜 프로그램(페이지)을 동일하게 나뉜 주기억장치의 영역(페이지 프레임)에 적재시켜 실행하는 기법이다.

**세그멘테이션 기법** : 가상기억장치에 보관되어 있는 프로그램을 다양한 크기의 논리적인 단위로 나눈 후 주기억장치에 적재시켜 실행시키는 기법이다.

**가상기억장치** : 보조기억장치의 일부를 주기억장치처럼 사용하는 것으로 용량이 작은 주기억장치를 마치 큰 용량을 가진 것처럼 사용하는 기법이다. 프로그램을 여러 개의 작은 블록 단위로 나누어서 가상기억장치에 보관해놓고, 프로그램 실행 시 요구되는 블록만 주기억장치에 불연속적으로 할당하여 처리한다.

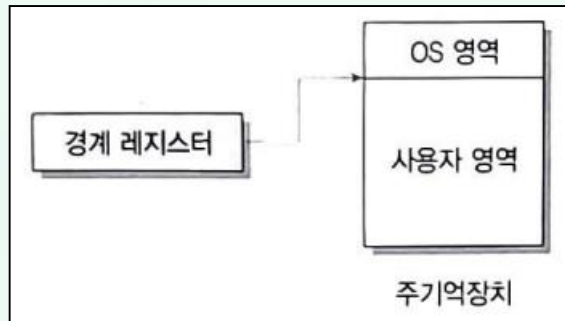


# 4.응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법)

## 2) 단일 분할 할당 기법

; 단일 분할 할당 기법은 주기억장치를 운영체제 영역과 사용자 영역으로 나누어 한 순간에는 오직 한 명의 사용자만이 주기억장치의 사용자 영역을 사용하는 기법이다.

- 가장 단순한 기법으로 초기의 운영체제에서 많이 사용하던 기법이다.
- 운영체제를 보호하고, 프로그램이 사용자 영역만을 사용하기 위해 운영체제 영역과 사용자 영역을 구분하는 경계 레지스터(Boundary Register)가 사용된다.
- 프로그램의 크기가 작을 경우 사용자 영역이 낭비될 수 있다.
- 초기에는 주기억장치보다 큰 사용자 프로그램은 실행할 수 없었으나 오버레이 기법을 사용하면서 이 문제가 해결되었다.



**경계 레지스터(Boundary Register)** : 사용자 영역에 있는 사용자 프로그램이 운영체제 영역에 접근하지 못하도록 보호하는 레지스터로 사용자 영역이 시작되는 주소를 기억하고 있다.

**오버레이 기법** : 주기억장치보다 큰 사용자 프로그램을 실행하기 위한 기법이다.



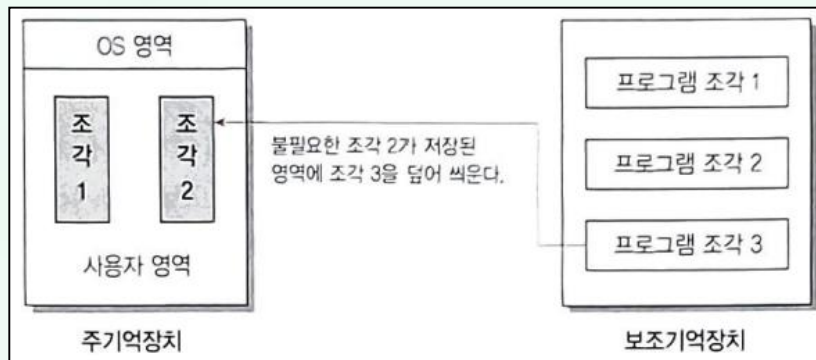
# 4.응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법)

## 2) 단일 분할 할당 기법

### 오버레이(Overlay) 기법

; 오버레이 기법은 주기억장치보다 큰 사용자 프로그램을 실행하기 위한 기법이다.

- 보조기억장치에 저장된 하나의 프로그램을 여러 개의 조각으로 분할한 후 필요한 조각을 차례로 주기억장치에 적재하여 프로그램을 실행한다.
- 프로그램이 실행되면서 주기억장치의 공간이 부족하면 주기억장치에 적재된 프로그램의 조각 중 불 필요한 조각이 위치한 장소에 새로운 프로그램의 조각을 중첩(Overlay)하여 적재한다.
- 프로그램을 여러 개의 조각으로 분할하는 작업은 프로그래머가 수행해야 하므로 프로그래머는 시스템 구조나 프로그램 구조를 알아야 한다.



오버레이 기법이 가능한 이유 : 하나의 프로그램을 여러 개의 조각으로 분할하여 처리할 수 있는 것은 프로그램의 모든 부분이 동시에 실행되는 것이 아니기 때문이다. 예를 들어 한글을 사용할 때 한글의 모든 기능을 동시에 사용하는 것이 아니라 파일을 여는 기능, 저장하는 기능, 색을 지정하는 기능 등이 분할되어 따로 실행될 수 있다.

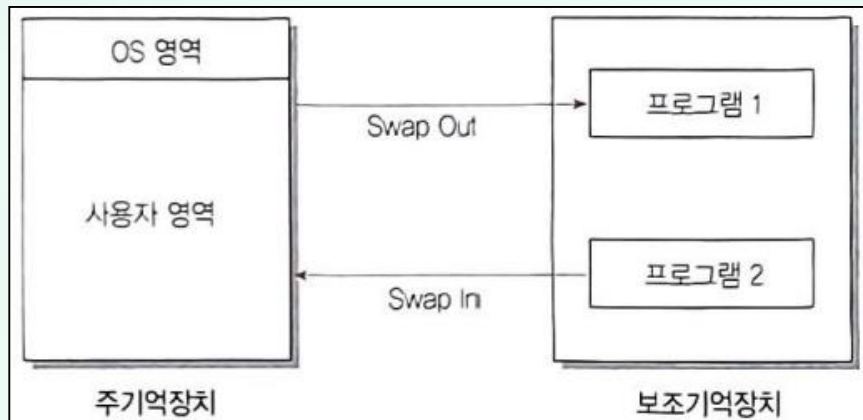
## 4.응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법)

### 2) 단일 분할 할당 기법

#### 스와핑(Swapping) 기법

; 스와핑 기법은 하나의 프로그램 전체를 주기억장치에 할당하여 사용하다 필요에 따라 다른 프로그램과 교체하는 기법이다.

- 주기억장치에 있는 프로그램이 보조기억장치로 이동되는 것을 Swap Out, 보조기억장치에 있는 프로그램이 주기억장치로 이동되는 것을 Swap In이라고 한다.
- 하나의 사용자 프로그램이 완료될 때까지 교체 과정을 여러 번 수행할 수 있다.
- 가상기억장치의 페이징 기법으로 발전되었다.



## 4.응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법)

### 3) 다중 분할 할당 기법

고정 분할 할당(Multiple contiguous Fixed parTition allocation, MFT) 기법 = 정적 할당(Static Allocation) 기법

; 고정 분할 할당은 프로그램을 할당하기 전에 운영체제가 주기억장치의 사용자 영역을 여러 개의 고정된 크기로 분할하고 준비상태 큐에서 준비중인 프로그램을 각 영역에 할당하여 수행하는 기법이다.

- 프로그램을 실행하려면 프로그램 전체가 주기억장치에 위치해야 한다.
- 프로그램이 분할된 영역보다 커서 영역 안에 들어갈 수 없는 경우가 발생할 수 있다.
- 일정한 크기의 분할 영역에 다양한 크기의 프로그램이 할당되므로 내부 단편화 및 외부 단편화가 발생하여 주기억장치의 낭비가 많다.
- 실행할 프로그램의 크기를 미리 알고 있어야 한다.
- 다중 프로그래밍을 위해 사용되었으나 현재는 사용되지 않는다.

**단편화** : 주기억장치의 분할된 영역에 프로그램이나 데이터를 할당할 경우, 분할된 영역이 프로그램이나 데이터보다 작거나 커서 생기는 빈 기억 공간을 의미한다.

•**내부 단편화** : 분할된 영역이 할당될 프로그램의 크기보다 크기 때문에 프로그램이 할당된 후 사용되지 않고 남아 있는 빈 공간

•**외부 단편화** : 분할된 영역이 할당될 프로그램의 크기보다 작기 때문에 프로그램이 할당될 수 없어 사용되지 않고 빈 공간으로 남아있는 분할된 전체 영역

## 4.응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법)

### 3) 다중 분할 할당 기법

고정 분할 할당(Multiple contiguous Fixed parTition allocation, MFT) 기법 = 정적 할당(Static Allocation) 기법

절대 번역과 적재, 재배포 번역과 적재

; 고정 분할 할당 기법은 다음과 같이 절대 번역과 적재, 재배포 번역과 적재로 구분할 수 있다.

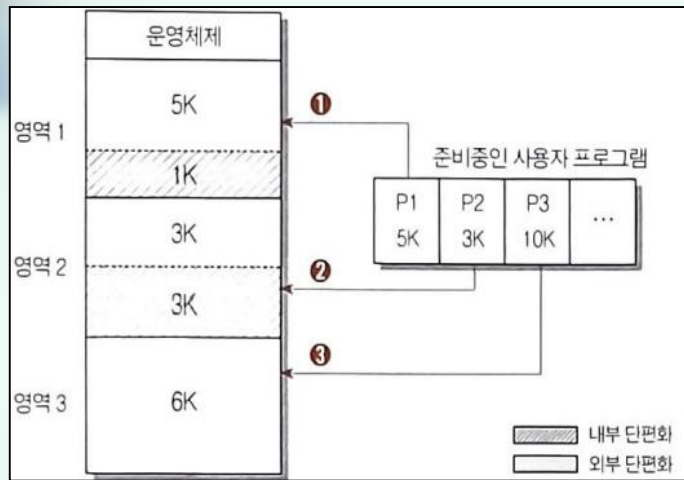
- **절대 번역과 적재** : 프로그램이 할당될 분할 영역을 어셈블러나 컴파일러가 지정하는 방식으로, 각 프로그램은 분할된 각 영역의 준비상태 큐에서 기다리며, 다른 분할 영역이 비어 있다 하더라도 지정된 분할 영역만을 사용해야 한다.
- **재배포 번역과 적재** : 프로그램이 할당될 영역이 미리 지정되지 않고, 하나의 준비상태 큐에서 기다린 순서대로 분할 영역에 할당되는 방식이다.

## 4.응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법)

### 3) 다중 분할 할당 기법

고정 분할 할당(Multiple contiguous Fixed parTition allocation, MFT) 기법 = 정적 할당(Static Allocation) 기법

예제) 18K 크기의 사용자 영역이 6K씩 3개의 영역으로 분할된 주기억장치에 준비 중인 5K, 3K, 10K 프로그램을 고정 분할 기법으로 할당하면 다음과 같다.



- ① 준비상태 큐에서 대기중인 P1(5K)를 영역 1에 할당시키면 1K의 내부 단편화가 발생한다.
- ② 준비상태 큐에서 대기중인 P2(3K)를 영역 2에 할당시키면 3K의 내부 단편화가 발생한다.
- ③ 준비상태 큐에서 대기중인 P3(10K)를 영역 3에 할당시키면 6K의 외부 단편화가 발생한다.

## 4.응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법)

### 3) 다중 분할 할당 기법

가변 분할 할당(Multiple contiguous Variable parTition allocation, MVT) 기법 = 동적 할당(Dynamic Allocation) 기법

; 고정 분할 할당 기법의 단편화를 줄이기 위한 것으로, 미리 주기억장치를 분할해 놓는 것이 아니라 프로그램을 주기억장치에 적재하면서 필요한 만큼의 크기로 영역을 분할하는 기법이다.

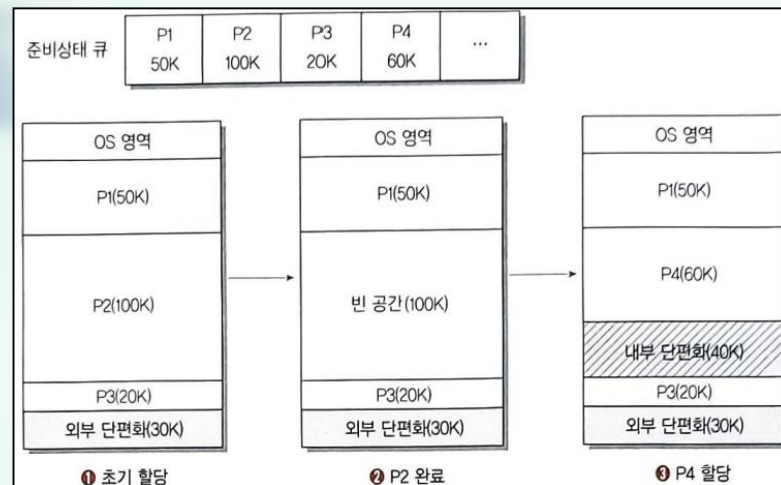
- 주기억장치를 효율적으로 사용할 수 있으며, 다중 프로그래밍의 정도를 높일 수 있다.
- 고정 분할 할당 기법에 비해 실행될 프로세스 크기에 대한 제약이 적다.
- 단편화를 상당 부분 해결할 수 있으나 영역과 영역 사이에 단편화가 발생할 수 있다.

## 4.응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법)

### 3) 다중 분할 할당 기법

가변 분할 할당(Multiple contiguous Variable parTition allocation, MVT) 기법 = 동적 할당(Dynamic Allocation) 기법

예제) 200K 크기의 사용자 영역으로 구성된 주기억장치에 준비중인 다음과 같은 프로그램들을 가변 분할 할당 기법으로 할당하면 다음과 같다.



- ① P1, P2, P3이 차례로 주기억장치의 사용자 영역에 할당되며 30K( $200 - 50 - 100 - 20$ )의 외부 단편화가 발생된다.
- ② P2의 수행이 완료되면 100K의 빈 공간이 생긴다.
- ③ 준비상태 큐에서 기다리던 P4는 P2가 사용했던 공간에 할당되며 40K( $100 - 60$ )의 내부 단편화가 발생된다.



# 응용 SW 기초기술 활용 - SEC\_05(주기억장치 할당 기법) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(주기억장치 할당 기법)

1. 주기억장치의 사용자 영역을 일정 수의 고정된 크기로 분할 하여 준비상태 큐에서 준비 중인 프로그램을 각 영역에 할당하여 수행하는 기법은?

- ① 가변 분할 기억장치 할당
- ② 고정 분할 기억장치 할당
- ③ 교체 기법
- ④ 오버레이 기법

**고정 분할 할당(Multiple contiguous Fixed parTition allocation, MFT) = 정적 할당(Static Allocation)**

고정 분할 할당 방식은 프로그램을 할당하기 전에 운영체제가 주기억장치의 사용자 영역을 여러 개의 고정된 크기로 분할하고 준비상태 큐에서 준비중인 프로그램이나 데이터를 각 영역에 할당하여 수행하는 기법이다.

- ▶ 프로그램을 실행하려면 프로그램 전체가 주기억장치에 위치 해야 한다.
- ▶ 프로그램이 분할된 영역보다 커서 영역 안에 들어갈 수 없는 경우가 발생할 수 있다.

3. 다음 기억공간 관리 중 고정 분할 할당과 동적 분할 할당으로 나누어 관리되는 기법은?

- ① 연속 로딩 기법
- ② 분산 로딩 기법
- ③ 페이징(Paging)
- ④ 세그먼트(Segment)

고정 분할 할당과 동적 분할 할당은 다중 분할 할당 기법에 해당되고 다중 분할 할당 기법은 연속 할당 기법에 해당한다. 로딩>Loading)을 한글로 번역하면 '할당'이다.

주기억장치 할당 기법은 프로그램이나 데이터를 실행시키기 위해서 주기억장치에 어떻게 할당할 것인지에 대한 내용이며, 연속 할당 기법과 분산 할당 기법으로 분류할 수 있다.

### 연속 할당 기법

프로그램을 주기억장치에 연속적으로 할당하는 기법을 말하며, 단일 분할 할당 기법과 다중 분할 할당 기법이 있다.

- ▶ 단일 분할 할당 기법 : 오버레이, 스와핑
- ▶ 다중 분할 할당 기법 : 고정 분할 할당 기법, 가변(동적) 분할 할당 기법

### 분산 할당 기법

- ▶ 프로그램을 특정단위의 조각으로 나누어 주기억장치 내에 분산하여 할당하는 기법으로, 페이징 기법과 세그먼테이션 기법으로 나눌 수 있다.

A close-up, low-angle shot of a white car's side mirror and door handle against a light blue sky. The car is on the left side of the frame, and the sky is on the right. The text "감사합니다." is overlaid on the right side of the image.

**감사합니다.**