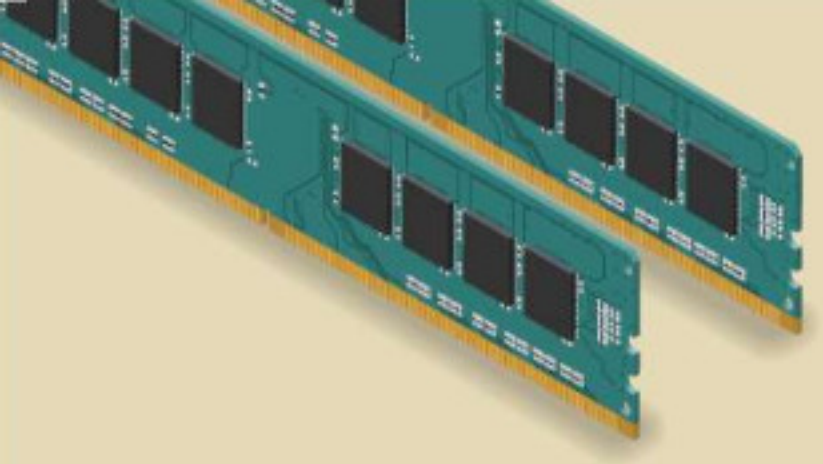


운영체제는 컴퓨터 시스템의 하드웨어를 관리하고 응용 소프트웨어를 실행하기 위해 하드웨어와의 연결을 제공하고 공통 시스템 서비스를 제공하는 시스템 소프트웨어이다. 입출력과 메모리 할당과 같이 하드웨어를 제어해야 하는 경우 운영체제는 응용 소프트웨어와 컴퓨터 하드웨어 사이에서 중재 역할을 하며 응용 소프트웨어의 코드는 일반적으로 하드웨어에서 직접 실행된다.

따라서 이번 단원의 학습을 통해서 운영체제의 구성과 종류를 알고 프로세스 관리, 주기억 장치 관리, 보조 기억 장치 관리, 가상 기억 장치 관리, 입출력과 파일 관리를 할 수 있는 역량을 기르며, 운영체제와 하드웨어의 특성을 이해하여 운영체제 운용 계획을 수립할 수 있도록 한다.

PART

III

운영체제의 자원 관리

> CHAPTER <

CHAPTER

01 운영체제의 구성

학습목표

- 운영체제의 개념과 종류, 운영체제를 구성하는 프로그램을 이해할 수 있다.
- 운영 방식에 따른 유형별 특징과 운영체제의 처리 방식의 차이를 이해할 수 있다.
- 컴퓨팅 시스템의 성능 저해 요인과 관리 방법을 이해할 수 있다.

여는 이야기

우리가 사용하는 운영체제

우리는 생활 속에서 많은 운영체제들을 알게 모르게 많이 활용하고 있다.

우리의 생활이나 일상 업무에서 활용되는 운영체제에는 어떤 것들이 있는지 찾아보고 어떤 원리에 의해서 운영체제가 작동되는지 생각해 보자.

단/원/학/습/안/내

이 단원에서는 우리가 일상 생활에서 다양하게 활용되고 있는 운영체제의 원리와 구성, 운영체제의 종류를 살펴봄으로써 실생활과 업무에서 운영체제와 관련된 직무를 수행하는 데 필요한 실무 능력을 기르도록 한다.

1 운영체제의 개념과 종류

운영체제는 사용자가 컴퓨터 시스템의 응용 프로그램을 쉽고 효율적으로 실행할 수 있는 환경을 제공한다. 그리고 컴퓨터 시스템의 하드웨어 자원과 소프트웨어 자원을 효율적으로 할당, 관리, 보호하는 목적을 가지고 있다.

운영체제는 제어 프로그램이므로 사용자 프로그램의 오류, 잘못된 자원 사용 등을 감시하는 것과 입출력 장치 등의 자원에 대한 운영과 제어를 관리한다.

(1) 운영체제의 개념 및 구성

운영체제는 사용자와 컴퓨터 하드웨어 사이에서 중재 역할을 하는 커널(Kernel)과 사용자가 컴퓨터 시스템을 운용할 수 있게 해 주는 셸(Shell)이라고 불리는 사용자 인터페이스(User Interface), 그리고 필수 유틸리티(Utility) 프로그램으로 구분할 수 있다.

그림 Ⅱ-1 운영체제의 역할

커널은 컴퓨터 하드웨어에 대한 중재 역할을 해야 하기 때문에 프로세스 관리, 동기화 및 통신, 메모리 관리, 입출력 관리 등의 핵심기능을 수행하며, 사용자 인터페이스는 사용자가 컴퓨터 시스템에 요청하는 동작들을 수행하는 역할을 한다. 예전에는 키보드와 모니터 화면을 사용하여 텍스트 메시지로 사용자들과 대화하였지만 현재는 아이콘, 단추 등을 사용하여 마우스를 통해 사용자들과 대화하는 것이 일반적이다.

필수 유틸리티는 서비스 프로그램, 언어 번역 프로그램, 문제 처리 프로그램 등과 같이 운영체제에서 기본적으로 제공되는 프로그램들을 의미한다.

1 커널

커널은 펌웨어(Firmware)와 장치 드라이버의 도움을 받아 모든 컴퓨터 하드웨어 장치에 대한 가장 기초 수준의 제어권을 얻는다.

그리고 주기억 장치(RAM)를 통해 프로그램을 위한 기억 장치 접근을 관리하며 어느 프로그램이 어느 하드웨어 자원에 접근할지를 결정하며 CPU의 동작 상태를 늘 최적으로 설정 및 초기화하고 디스크, 테이프, 플래시 메모리와 같은 비휘발성 기억 장치를 관리한다.

그림 Ⅱ-2 커널의 역할

2 사용자 인터페이스

모든 컴퓨터 시스템은 사용자가 컴퓨터와 소통할 수 있게 다양한 종류의 입력을 받을 수 있는 사용자 인터페이스가 필요하다. 초기의 많은 컴퓨터 시스템들이 키보드, 마우스와 같은 장치들을 이용하여 문자 형태로 입력을 받는 명령 줄 인터페이스를 사용했었지만 다양한 형태의 입력 장치가 발달함에 따라 마우스, 터치패드 등의 장치들을 이용하여 창, 단추, 아이콘 등을 통해 입력을 받는 그래픽 사용자 인터페이스가 주로 사용되고 있다.

개인용 컴퓨터의 운영체제는 그래픽 사용자 인터페이스가 주로 사용되고 있지만 서버, 슈퍼 컴퓨터 등의 운영체제에서는 아직도 명령 줄 인터페이스가 많이 사용되고 있다.

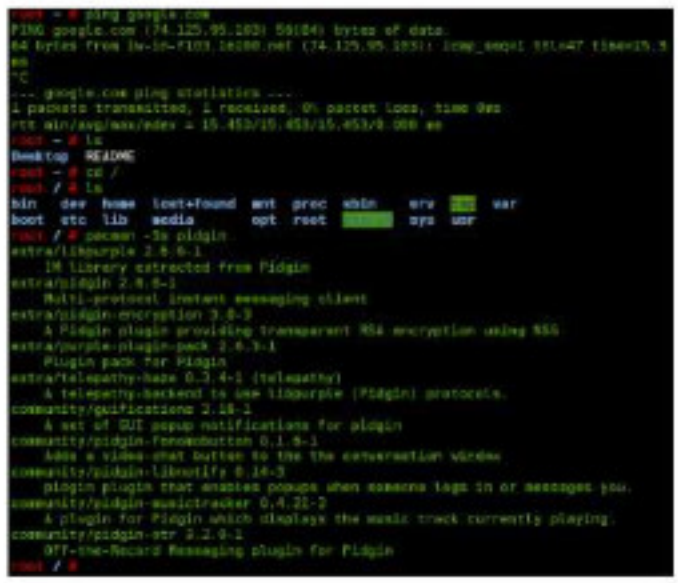


그림 Ⅲ-3 명령 줄 사용자 인터페이스

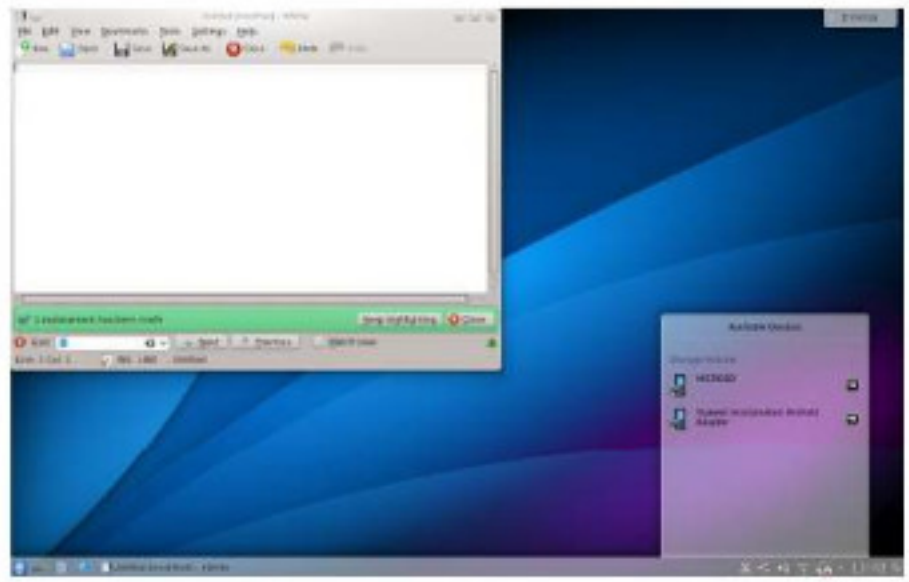


그림 Ⅲ-4 그래픽 사용자 인터페이스

(2) 운영체제의 종류

운영체제를 구분하는 방법은 매우 많지만 사용 범위와 운영되는 환경 등에 따라 데스크탑용 운영체제, 서버용 운영체제, 모바일 운영체제로 구분할 수 있다.

데스크톱용 운영체제로는 Windows, Mac OS, Linux가 대표적이고, 서버용 운영체제로는 Unix 서버, Linux 서버, Windows 서버가 대표적이다. 최근 사용이 많이 늘어나고 있는 모바일 운영체제로는 Android, iOS, Windows 모바일, Linux 모바일 등이 있다.

1 Windows

Windows는 마이크로소프트에서 개발한 대표적인 데스크톱용 운영체제로 현재 가장 많은 비율로 개인용 컴퓨터에서 사용하고 있으며 서버용 운영체제인 Windows 서버, 모바일 운영체제인 Windows 모바일도 사용이 늘어나고 있는 추세이다.

2 Mac OS

Mac OS는 애플이 매킨토시 컴퓨터용으로 개발한 대표적인 그래픽 사용자 인터페이스 운영체제이다. 일반적으로 매킨토시 컴퓨터와 호환되는 경우에만 설치하여 사용할 수 있으며 모바일 운영체제인 iOS의 기초가 되는 운영체제이다.

3 Linux

Linux는 자유소프트웨어와 오픈 소스 개발의 가장 유명한 표본이 되는 운영체제이며 서버용 운영체제에 더 적합하게 지원을 하기 때문에 폭넓게 사용되고 있다. 개인용 컴퓨터, 서버용 컴퓨터, 리눅스 모바일, 임베디드 시스템까지 광범위하게 이용되고 있다.

4 Unix

Unix는 대표적인 다중 사용자 운영체제로 Windows 계열을 제외한 대부분의 운영체제에 영향을 미친 운영체제이다. 서버용 운영체제로 많이 사용되고 있으며 개인용 또는 임베디드용으로도 사용되고 있다.

5 Android

Android는 대표적인 모바일 운영체제로 구글에서 모바일을 대상으로 소스까지 무료로 공개한 오픈소스 운영체제이다. 스마트워치와 같은 웨어러블 장치들에서도 많이 활용되고 있다.

6 iOS

iOS는 애플의 모바일 기기들을 대상으로 만든 운영체제이다. Mac OS를 기반으로 하는 운영체제이기 때문에 모바일 기기들과 매킨토시 컴퓨터간의 호환성과 최적화 등에서 뛰어난 특징이 있다.

개방형과 폐쇄형 운영체제

운영체제 소스 코드의 공개 여부에 따라 개방형과 폐쇄형으로 구분할 수 있다.
개방형 운영체제는 제조업체 사이의 협력이 쉽고 버전의 업그레이드가 수시로 이루어진다는 강점이 있지만 콘텐츠의 품질 저하나 보안 문제에 약점이 있다.
폐쇄형 운영체제는 오류 수정과 보안에 강점이 있지만 점유율의 확대가 느리다는 약점이 있다. 대표적인 개방형과 폐쇄형 운영체제는 다음과 같다.
개방형 운영체제: Linux, Android
폐쇄형 운영체제: Windows, iOS

직무 따라잡기

운영체제들의 점유율을 찾아보고 데스크톱용, 서버용, 모바일용을 비교해 보자.

순위	종류	비교 내용
1		
2		
3		

2 운영체제의 운영 방식

운영체제는 운영되는 방식과 시스템의 종류 등에 따라 다양한 형태로 분류할 수 있다. 단일 사용자와 다중 사용자와 같이 사용자의 형태로 구분할 수도 있고 명령의 수행 방법에 따라 분류할 수도 있으며 개인용 컴퓨터와 서버용 컴퓨터 등의 운영 시스템에 따라 분류할 수도 있다.

(1) 명령 처리 방식에 따른 분류

명령을 처리하는 방식에 따라 다양한 형태로 분류할 수 있다.

1 싱글태스킹/멀티태스킹 운영체제

싱글태스킹 운영체제는 한 번에 오직 하나의 프로그램만 실행할 수 있으나 멀티태스킹 운영체제는 하나 이상의 프로그램이 동시에 실행할 수 있게 한다. 이는 운영체제의 작업 스케줄링에 의해 여러 프로세스 사이에서 이용 가능한 프로세서 시간을 쪼개는 시분할을 통해 이루어진다.

멀티태스킹의 경우 선점형과 비선점형이 있다. 선점형 멀티태스킹 운영체제는 CPU 시간을 쪼개어 프로그램들 각각에 할당해 준다. 비선점형 멀티태스킹 운영체제는 프로세스가 CPU 시간을 할당 받아 실행이 시작되면, 그 프로세스가 종료될 때까지 CPU를 사용하여 계속 실행한다.

2 분산 운영체제

분산 운영체제는 구별된 컴퓨터 그룹을 관리하고 이들이 마치 하나의 컴퓨터인 것처럼 보이게 만들어 준다. 서로 연결되어 통신하는 네트워크화된 컴퓨터들이 개발되면서 분산 컴퓨팅이 활성화되었다. 분산되는 연산들은 하나 이상의 컴퓨터에서 수행된다. 하나의 그룹에 속하는 컴퓨터들이 협업을 할 때 분산 시스템을 형성하게 된다.

3 실시간 운영체제

실시간 운영체제는 특정한 짧은 시간 내에 이벤트나 데이터의 처리를 보증하는 운영체제이다. 실시간 운영체제는 싱글태스킹일 수도 있고, 멀티태스킹일 수도 있으며 멀티태스킹의 경우 특수한 스케줄링 알고리즘을 사용한다.

(2) 운영 시스템에 따른 분류

운영되는 시스템에 따라 다양한 형태로 분류할 수 있다.

1 개인용 컴퓨터 운영체제

일반적으로 단일 사용자 운영체제를 사용하고 있으며 개인용 컴퓨터가 등장하면서 메인프레임 운영체제를 수정하여 MS-DOS 등의 명령 줄 사용자 인터페이스를 사용하였으나 Mac OS 등의 그래픽 사용자 인터페이스가 등장하고 Microsoft의 Windows가 등장하면서 대부분의 개인용 컴퓨터 운영체제는 그래픽 사용자 인터페이스를 사용하고 있다. 주로 사용되는 운영체제로는 Windows, Linux, Mac OS 등이 있다.

2 메인프레임 운영체제

일반적으로 다중 사용자 운영체제를 사용하고 있으며 초기에는 명령 줄 사용자 인터페이스를 주로 사용하였으나 최근에는 그래픽 사용자 인터페이스를 많이 사용하고 있다. 주로 사용되는 운영체제로는 Unix, Solaris 등이 있다.

3 임베디드 운영체제

임베디드 운영체제는 임베디드 컴퓨터 시스템에서 사용할 수 있게 설계되어 있다. PDA처럼 조그마한 기계에 동작하도록 설계되어 있으며, 제한된 수의 자원으로 동작한다. 크기가 매우 작고 극히 효율적으로 설계되어 있다. 주로 사용되는 운영체제로는 Embeded Linux나 Windows CE 등이 있다.

직무 따라잡기

내가 가장 많이 사용하는 운영체제의 장점과 단점을 분석하여 보자.

사용 운영체제	장점	단점

3 컴퓨팅 시스템의 성능 저해 요인

컴퓨터를 사용하다 보면 여러 가지 원인에 의해서 성능이 저해된다. 예를 들어 프로그램을 설치하고 제거를 하는 것이 반복되면서 불필요한 정보가 쌓일 수도 있고 악성 코드나 바이러스에 감염될 수도 있다. 또한 동시에 실행되고 있는 프로그램의 수가 너무 많거나 저장 공간이 부족할 수도 있다. 성능이 저해되지 않도록 하기 위해서는 어떻게 해야 할까?

(1) 성능 저해로 발생할 수 있는 문제

성능이 저해되면 다음과 같은 문제가 발생할 수 있다.

컴퓨터에서 발생 가능한 문제

① 컴퓨터가 켜지지 않거나 매우 느리게 켜짐
② 특별한 원인 없이 컴퓨터가 재부팅됨
③ 특정 응용 프로그램이 실행되지 않음
④ 인터넷 또는 네트워크를 사용할 수 없음
⑤ 특정 사이트로 이동하지 않아도 강제로 이동됨
⑥ 사용자 개인 정보가 외부로 유출됨

네트워크 환경에서 발생 가능한 문제

① 내부적 또는 외부적으로 연결된 네트워크를 사용할 수 없음
② 지정된 업무 파일 서버에 접근할 수 없음
③ 네트워크에 접근할 때 속도가 매우 느림
④ 공유 프린터를 사용할 수 없음
⑤ 메일을 보내고 받을 수 없음

(2) 관리 방법

성능이 저해된 경우에는 다음과 같은 방법으로 문제를 해결할 수 있다.

1 악성 코드와 바이러스 점검

악성 코드와 바이러스로부터 컴퓨터의 감염을 막기 위해 그림 Ⅲ-5, 그림 Ⅲ-6과 같이 악성 코드 검색 프로그램 및 백신 프로그램을 설치하고 실시간으로 관리할 수 있도록 설정해야 한다. 악성 코드나 바이러스의 경우는 성능 저하뿐만 아니라 시스템에 치명적인 오류를 발생시킬 수 있으니 유의해야 한다.

그림 Ⅲ-5 백신 프로그램

그림 Ⅲ-6 디펜더 프로그램

2 디스크 정리와 디스크 조각모음

컴퓨터를 사용하다 보면 많은 프로그램을 설치하고 삭제하는 작업이 반복적으로 이루어져 불필요한 정보가 쌓이고 파일의 내용이 순차적으로 기록되지 못하고 여러 곳에 나누어 저장하게 되는 경우가 발생한다. 이때는 그림 Ⅲ-7, 그림 Ⅲ-8과 같이 디스크 정리와 디스크 조각모음을 통해 성능을 개선할 수 있다.

그림 Ⅲ-7 디스크 정리

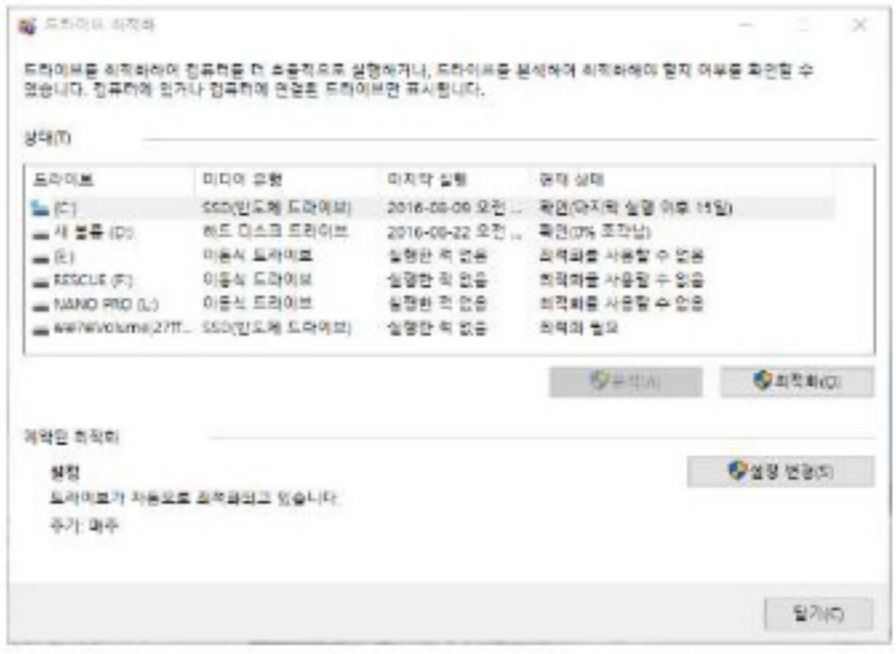


그림 Ⅲ-8 디스크 조각모음

3 불필요한 프로그램과 서비스 삭제하기

사용하지 않는 프로그램이나 서비스 운영으로 인해 성능이 저해되거나 오류가 발생할 수 있기 때문에 그림 Ⅲ-9, 그림 Ⅲ-10과 같은 기능을 사용하여 주기적으로 불필요한 프로그램과 서비스를 삭제하거나 중지시켜 성능을 개선할 수 있다.

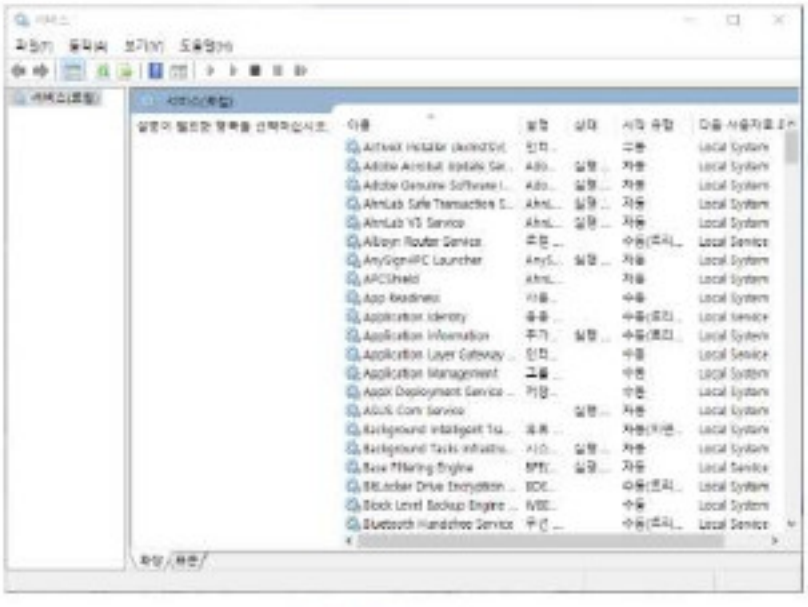


그림 Ⅲ-9 서비스

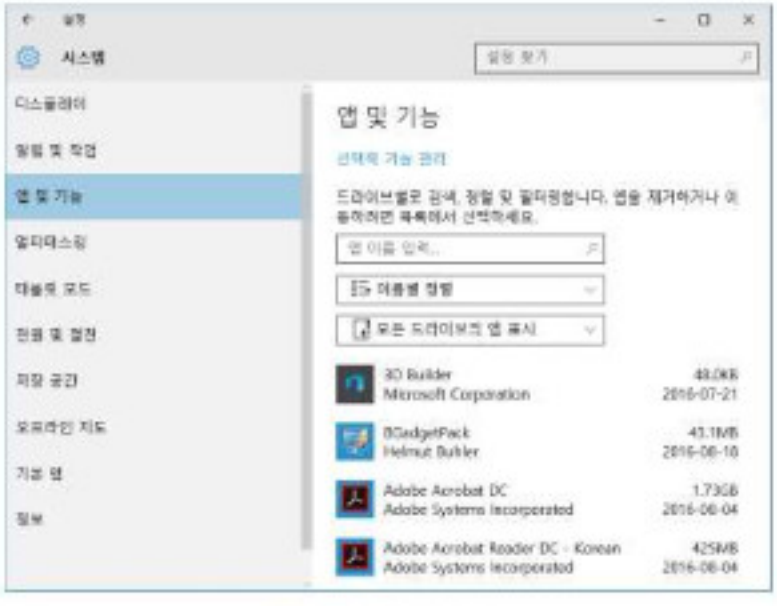


그림 Ⅲ-10 앱 및 기능

수행 활동

우리 생활 주변의 운영체제를 찾아 발표하기

활동 주제

우리 생활 주변의 운영체제를 찾아보고 특징 및 작동 원리를 정리하여 발표한다.

준비와 과정

- 모둠을 설정한다.
  - 4명 ~ 5명의 모둠을 설정한다.
  - 모둠별로 모둠의 이름을 정하고, 모둠에서의 역할을 정한다.

    모둠 이름:

    모둠장: 기록자: 검색자: 발표자:
- 생활 주변의 운영체제에 대한 자료를 찾고 운영체제의 특징을 찾아본다.
- 찾은 운영체제들의 작동 원리를 찾아본다.
- 정리한 운영체제 자료를 발표자가 발표한다.
  - 다른 모둠이 발표할 때, 자신이 속한 모둠의 운영체제와 비교해 본다.

활동 내용

1. 주변의 운영체제를 찾아보자.

2. 운영체제의 작동 원리를 찾아보자.

3. 운영체제의 작동 원리와 특징을 정리해 보자.

CHAPTER

02 프로세스 관리

학습목표

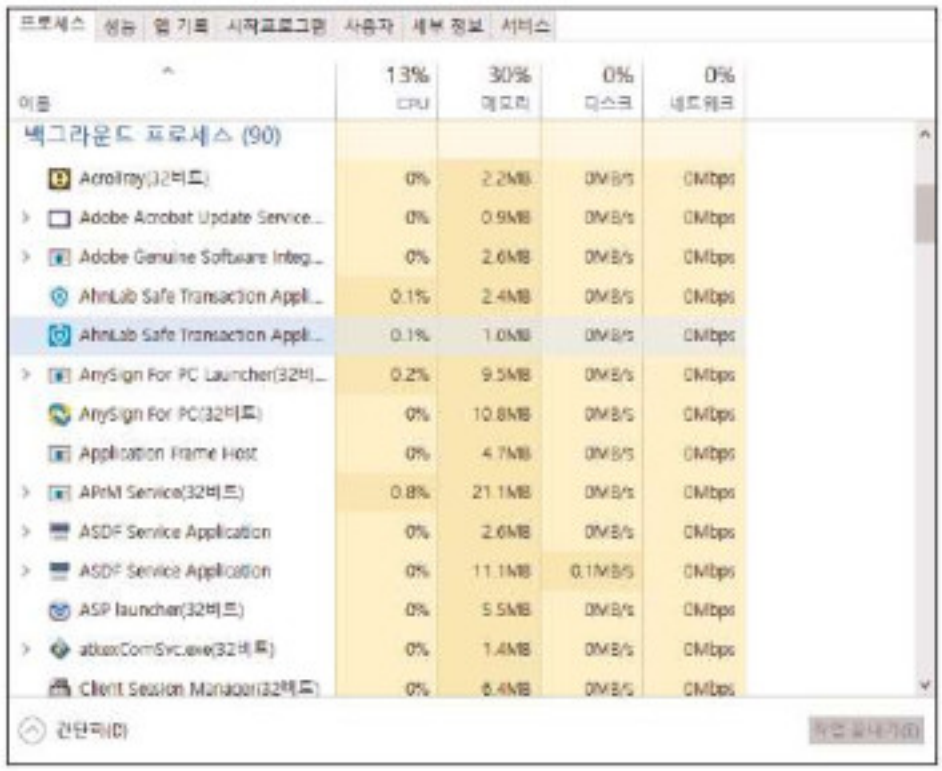
- 자원의 의미와 주요 자원을 이해할 수 있다.
- 프로세스의 개념과 프로세스의 상태와 상태 변화를 이해할 수 있다.
- 선점형, 비선점형의 프로세스 스케줄링 방법을 이해할 수 있다.
- 교착 상태의 개념과 발생하는 조건, 해결 기법을 이해할 수 있다.

여는 이야기

프로세스

우리들이 컴퓨터를 사용할 때 운영체제는 주기억 장치에 들어온 프로세스를 통해 프로그램을 실행하게 된다.

프로그램을 실행하기 위해 어떻게 프로세스들이 동작하는지 살펴보고 어떤 원리에 의해서 프로세스들이 관리되는지 생각해 보자.



단/원/학/습/안/내

이 단원에서는 자원과 프로세스의 개념을 이해하고 프로세스의 상태와 상태 변화를 살펴봄으로써 CPU에서 프로세스를 스케줄링하는 방법과 자원 관리 원리를 익히고 교착 상태를 해결하는 기법을 이해하여 직무를 수행하는 데 필요한 실무 능력을 기르도록 한다.

1 자원과 프로세스

대부분의 프로그램들은 보조 기억 장치에 저장되어 있다가 해당 프로그램 실행 명령을 내리면 주기억 장치로 옮겨져 중앙 처리 장치에 의해 사용 가능한 자원을 활용하여 실행된다. 그림 Ⅲ-11은 프로그램 실행 과정을 간략히 나타낸 것이다.

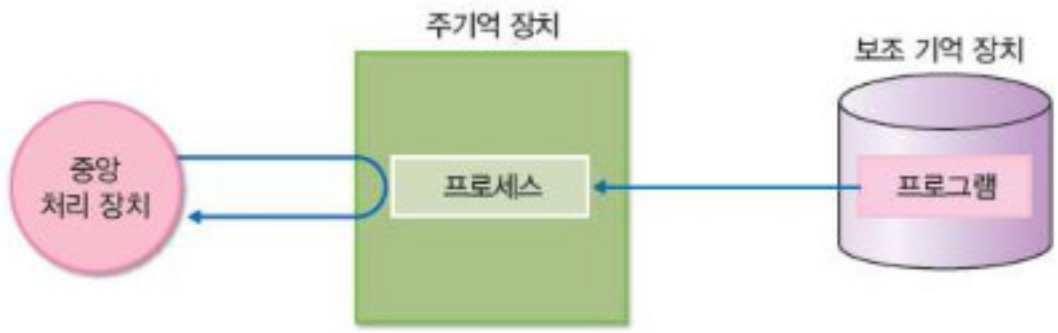


그림 Ⅲ-11 프로그램 실행 과정

(1) 자원

자원은 컴퓨터에서 사용될 수 있는 요소들을 말한다. 사용될 수 있는 요소들은 컴퓨팅 시스템을 이루는 하드웨어, 소프트웨어 또는 데이터 구성 요소를 말한다. 예를 들어 하드웨어 자원은 프린터, 디스크, 메모리 같은 장치나 네트워크 상에서 활용 가능한 서버, 프린터 등의 네트워크 자원을 말하며 소프트웨어 자원은 프로그램, 유틸리티, 프로그램 내의 구성 요소 등을 말한다. 데이터 자원은 파일, 데이터베이스 등이 포함된다.

(2) 프로세스(process)

프로세스는 실행되기 위해 주기억 장치에 옮겨진 프로그램을 말한다. 운영체제는 프로그램을 실행하기 위해 보조 기억 장치에 저장되어 있는 프로그램 전체 또는 일부를 주기억 장치의 적절한 위치로 올리는데 이것이 프로세스이다. 프로세스를 실행중인 프로그램이라고 정의할 수 있는데, 프로그램 코드뿐만 아니라 실행에 필요한 다양한 정보도 포함된다.

1 프로세스 제어 블록(PCB : Process Control Block)

프로세스 제어 블록은 프로세스 각각에 대한 정보를 관리하는 테이블을 말한다. 운영체제는 프로세스 제어 블록을 통해 프로세스를 관리한다. 프로세스 제어 블록은 프로세스가 생성될 때 같이 생성되고, 이 프로세스가 종료될 때 같이 사라진다.

프로세스 제어 블록에 저장되는 정보는 크게 다음과 같다.

① 프로세스의 상태 : 실행 상태인지 또는 실행을 위해 준비 상태인지 등에 대한 정보

② 프로그램 카운터 값 : 중앙 처리 장치의 프로그램 카운터라는 레지스터에 저장된 값으로, 다음에 실행될 명령어의 주기억 장치 주소를 의미

③ 스케줄링 정보 : 다음에 실행될 프로세스를 결정하는 데 필요한 정보로, 프로세스 스케줄링 정책, 우선 순위 등을 의미

④ 주기억 장치 정보 : 해당 프로세스가 주기억 장치의 어느 영역에 위치해 있는지에 대한 정보 저장

이외에도 프로세스 ID, 레지스터 등의 정보들을 포함하고 있다.

2 프로세스의 상태

프로세스는 생성되어 종료할 때까지 그림 Ⅲ-12와 같이 준비, 대기, 실행 상태를 거친다.

준비 상태에 있는 프로세스는 다음과 같은 세 가지 경우에 실행 상태가 될 수 있다.

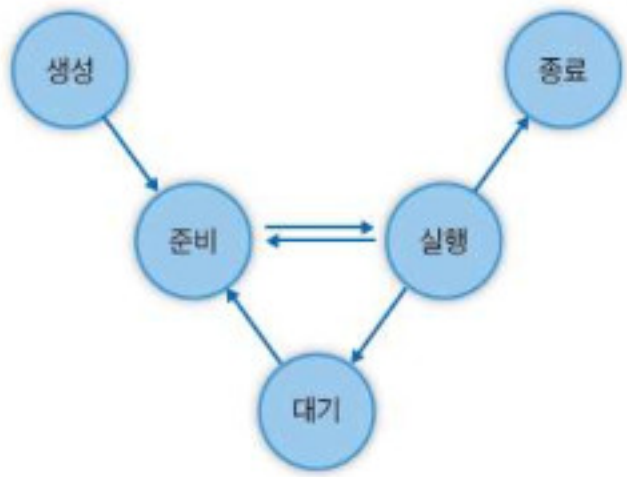
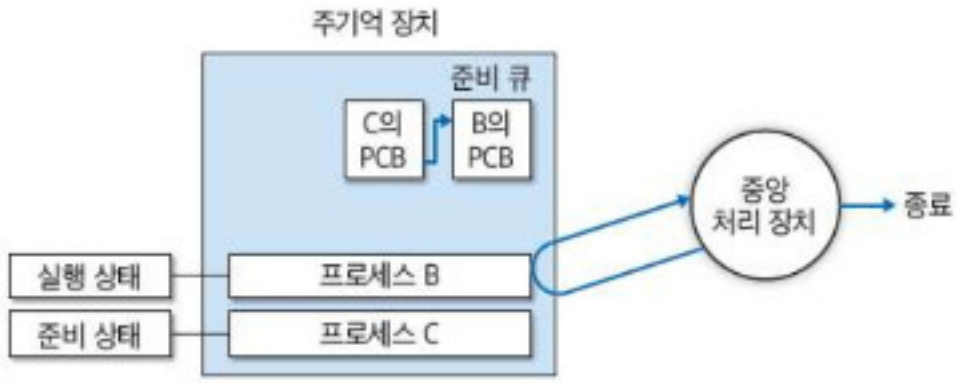


그림 Ⅲ-12 프로세스 상태

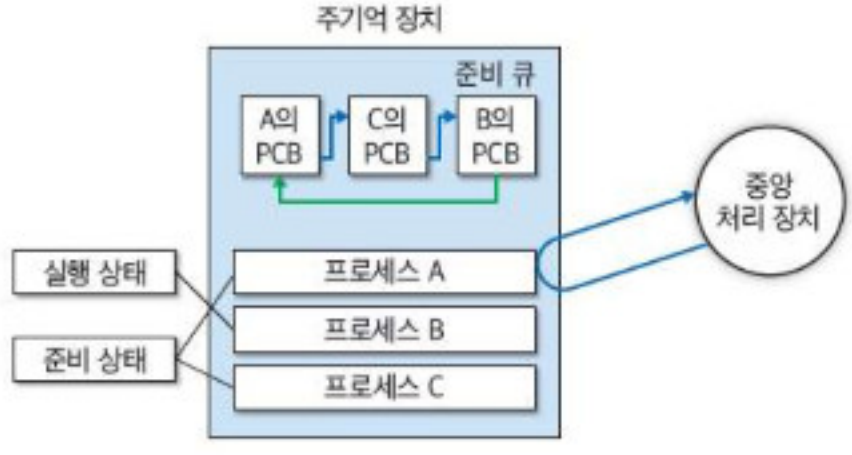
① 준비 큐에서 이전 프로세스의 실행이 종료된 경우

중앙 처리 장치에서 실행되고 있던 이전 프로세스의 실행이 종료된 경우 운영체제가 준비 상태에 있는 프로세스들 중 프로세스 스케줄링 정책에 따라 가장 먼저 실행될 프로세스가 실행 상태가 된다.



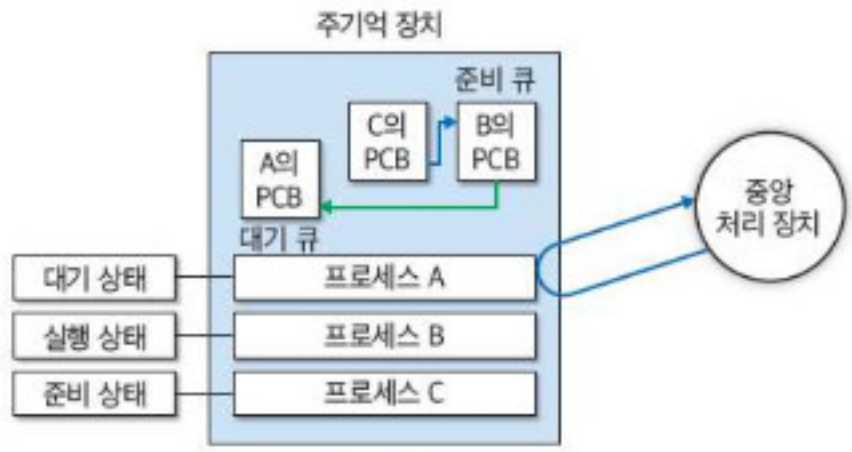
② 이전 프로세스가 중앙 처리 장치를 오래 차지하고 있는 경우

프로세스가 중앙 처리 장치를 일정 시간 이상 차지하고 있는 경우 운영체제가 프로세스를 준비 상태가 되게 하고, 가장 먼저 실행될 프로세스가 실행 상태가 된다. 이때 중앙 처리 장치의 일부 레지스터에 저장되어 있는 이전 프로세스의 정보와 새 프로세스의 정보 일부를 중앙 처리 장치의 레지스터에 저장하는 것을 문맥 전환(context switch)이라 한다. 대표적으로 프로그램 카운터 값을 저장하고 있어야 프로세스가 다시 실행될 때 원하는 명령어부터 실행된다.



③ 실행 프로세스가 디스크 입출력 명령을 실행해야 하는 경우

디스크 입출력 명령이 발생하면 입출력 동작이 완료될 동안 중앙 처리 장치가 멈추게 된다. 이런 경우 운영체제가 해당 프로세스를 대기 큐로 보내고, 가장 먼저 실행될 프로세스가 실행 상태가 된다. 이때의 이전 프로세스 상태를 대기 상태라 하고 이 경우에도 문맥 전환이 일어난다.



더 알아보기 프로세스의 생성 실행 및 준비

■ 프로세스의 생성

운영체제가 다음에 실행할 새로운 프로세스 A를 주기억 장치에 가져오면 프로세스 A의 프로세스 제어 블록(PCB)이 준비 큐에 등록된다. 이러한 프로세스 상태를 프로세스의 생성이라고 하고 그림 Ⅲ-13과 같이 나타낼 수 있다.

그림 Ⅲ-13 프로세스의 생성

■ 프로세스의 실행

운영체제는 준비 큐의 가장 앞에 위치한 프로세스 제어 블록을 중앙 처리 장치가 실행할 수 있도록 한다. 그림 Ⅲ-14와 같이 프로세스 A의 프로세스 제어 블록이 가장 앞에 위치해 있다면 제어 블록의 정보가 중앙 처리 장치의 레지스터에 저장되고 프로세스 A가 수행된다. 이러한 프로세스 상태를 프로세스의 실행이라고 한다.

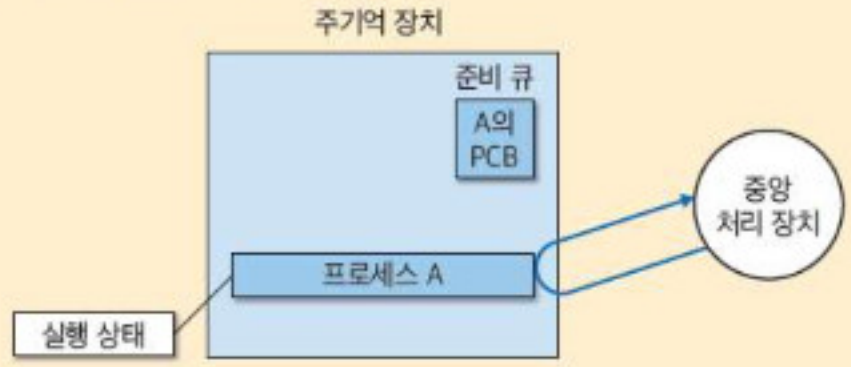


그림 Ⅲ-14 프로세스의 실행

■ 프로세스의 준비

프로세스 A가 실행되고 있는 상태에서 프로세스 B와 C가 생성되면 프로세스 B와 C의 프로세스 제어 블록이 준비 큐에 연결된다. 이때의 프로세스 B와 C의 프로세스 상태를 프로세스의 준비라고 하고 그림 Ⅲ-15와 같이 나타낼 수 있다.

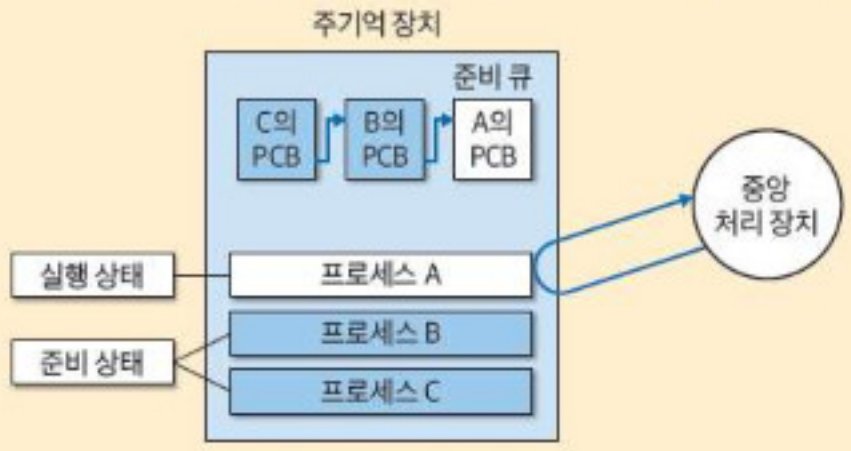


그림 Ⅲ-15 프로세스의 준비

2 프로세스 스케줄링

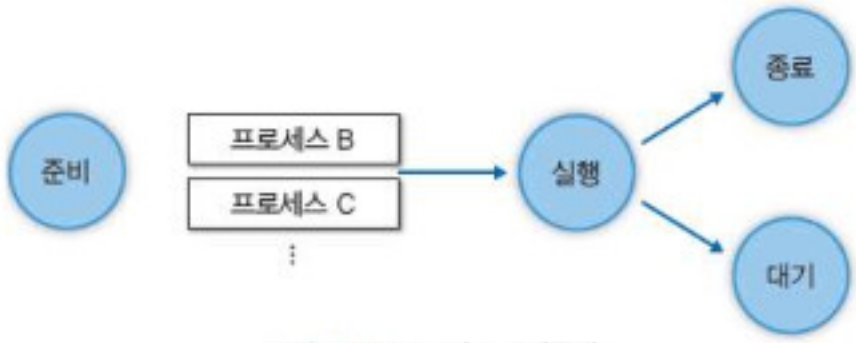


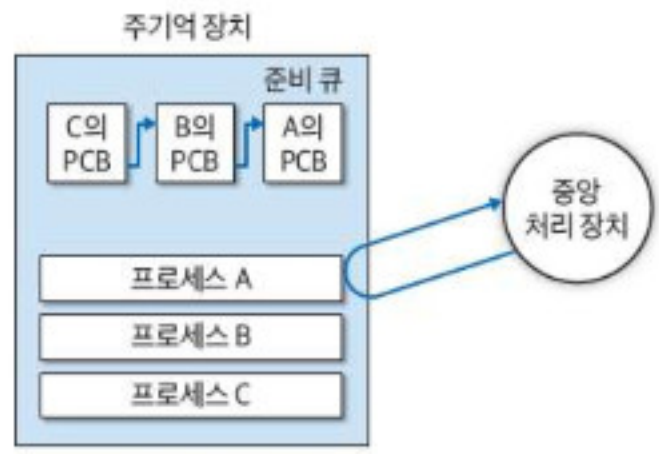
그림 Ⅲ-16 프로세스 스케줄링

최근 대부분의 시스템은 여러 개의 프로세스를 주기억 장치에 적재하여 중앙 처리 장치 이용률을 최대화하는 다중프로그래밍(multiprogramming) 방식으로 동작하기 때문에 준비 상태의 프로세스 중에서 가장 먼저 실행 상태가 될 프로세스를 결정하는 프로세스 스케줄링(process scheduling)이 필요하게 된다. 그림 Ⅲ-16은 프로세스 스케줄링을 간략히 나타낸 것이다.

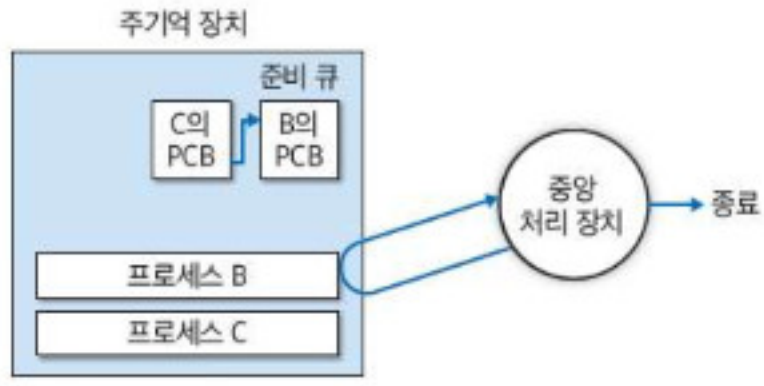
대표적인 프로세스 스케줄링에는 FCFS 스케줄링, 라운드로빈 스케줄링, 우선순위 스케줄링 등이 있다.

(1) FCFS 스케줄링(First-Come First-Served)

FCFS 스케줄링은 먼저 도착한 프로세스를 먼저 실행하는 방법으로 가장 간단한 정책이다. 프로세스가 A, B, C 순으로 생성되면 운영체제는 가장 먼저 생성된 프로세스 A에게 중앙 처리 장치를 배정하여 실행되도록 한다. 그리고 프로세스 B와 C는 프로세스 A가 종료될 때까지 준비 큐에서 아래 그림과 같이 기다린다.



프로세스 A의 실행이 종료되면 다음 프로세스인 B가 아래 그림과 같이 실행된다.

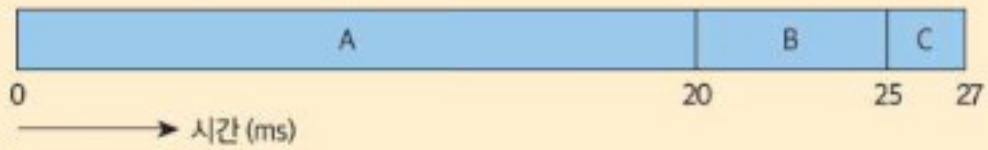


더 알아보기 FCFS 스케줄링의 대기 시간

이때 표와 같이 중앙 처리 장치 시간을 필요로 하는 세 개의 프로세스들이 있을 때의 대기 시간을 계산해 보면 다음과 같다.

프로세스	중앙 처리 장치 시간
프로세스 A	20ms
프로세스 B	5ms
프로세스 C	2ms

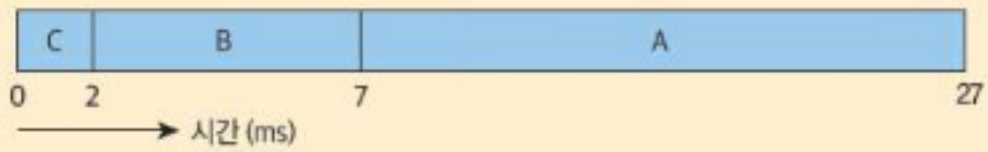
① 세 개의 프로세스들이 A, B, C 순으로 연속적으로 생성되면 다음과 같은 순서로 실행된다.



② 이 경우 프로세스 A의 대기 시간은 0ms, 프로세스 B는 20ms, 그리고 프로세스 C는 25ms이기 때문에 평균 대기 시간은 15ms가 된다.

$$\frac{0+20+25}{3} = 15(ms)$$

③ 그런데 만약 프로세스가 C, B, A 순서로 생성된다고 하면 프로세스들의 평균 대기 시간은 다음과 같다.



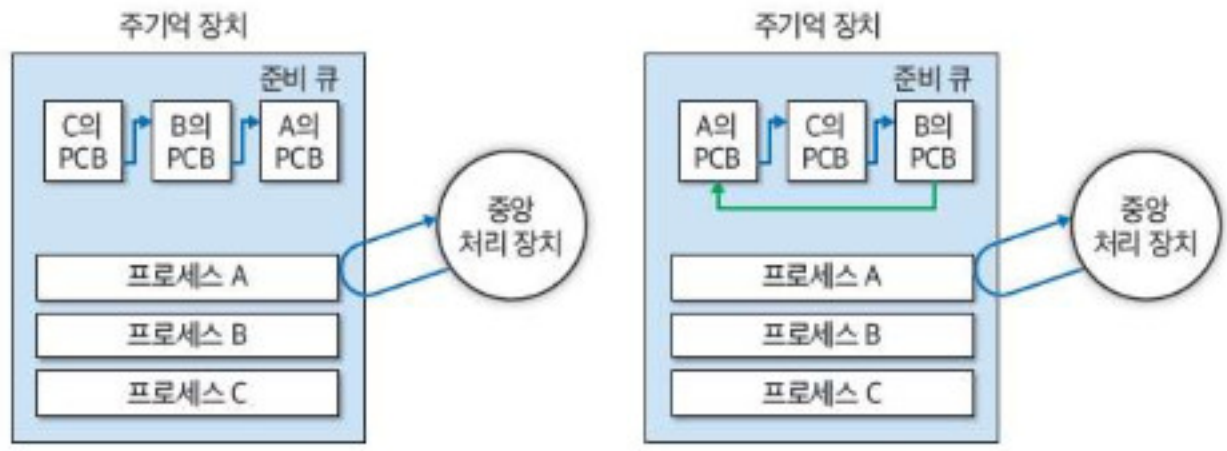
$$\frac{0+2+7}{3} = 3(ms)$$

이와 같이 FCFS 스케줄링은 프로세스의 실행 순서에 따라 평균 대기 시간의 차이가 크다.

(2) 라운드 로빈 스케줄링

라운드 로빈 스케줄링은 프로세스가 종료될 때까지 중앙 처리 장치를 차지하는 것이 아니라 여러 프로세스들이 중앙 처리 장치를 돌아가며 할당받아 실행되는 방식으로 리눅스를 포함한 대부분의 시스템에서 사용하는 방식이다.

프로세스들은 시간 할당량(time quantum) 동안 중앙 처리 장치를 할당받아 실행되며 시간 할당량 내에 종료하지 못하면 준비 상태로 전환되고 다음 프로세스가 중앙 처리 장치를 할당받아 오른쪽 그림과 같이 실행된다.

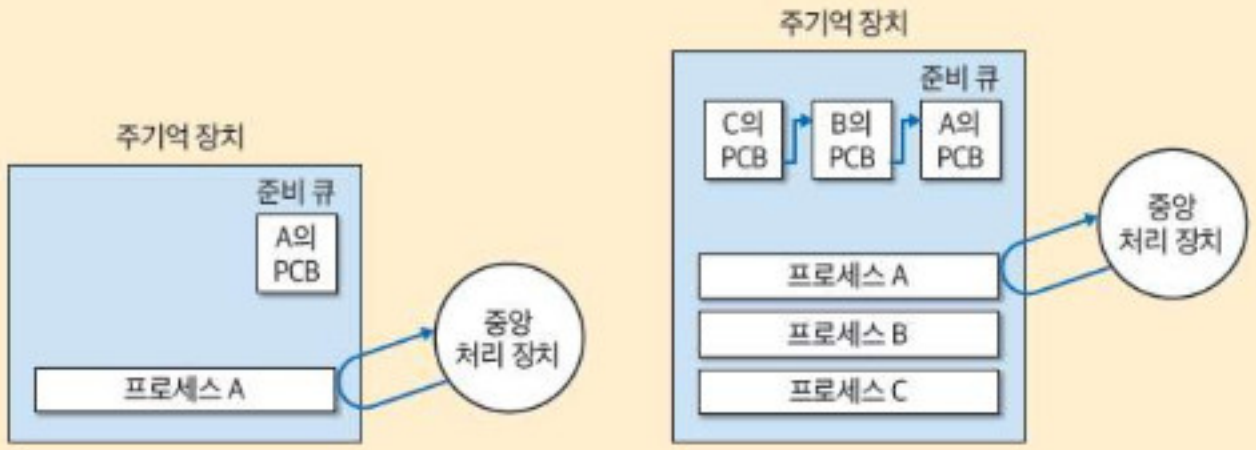


더 알아보기 라운드로빈 스케줄링의 대기 시간

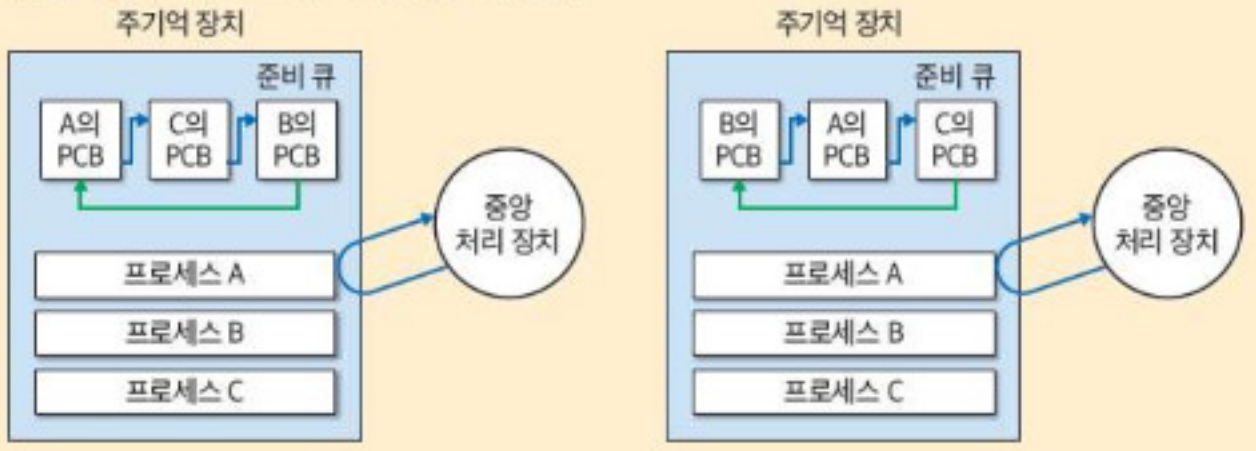
아래 표와 같이 중앙 처리 장치 시간을 필요로 하는 세 개의 프로세스들이 있고 시간 할당량이 4ms일 때의 대기 시간을 계산해보면 다음과 같다.

프로세스	중앙 처리 장치 시간
프로세스 A	20ms
프로세스 B	5ms
프로세스 C	2ms

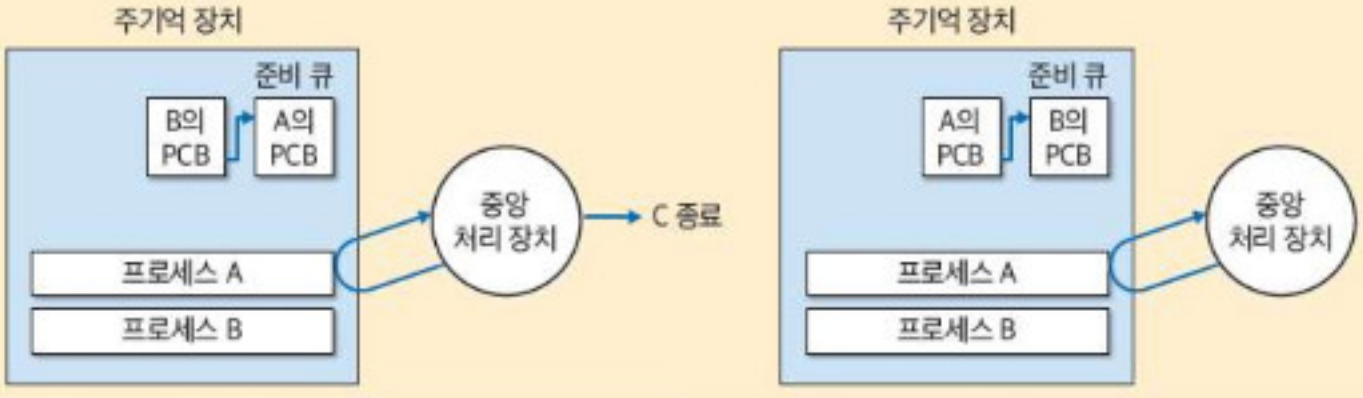
① 프로세스 A가 생성되어 중앙 처리 장치를 할당받아 실행되고 바로 프로세스 B와 C가 생성되면 프로세스 B와 C의 프로세스 제어 블록이 준비 큐에 연결된다.



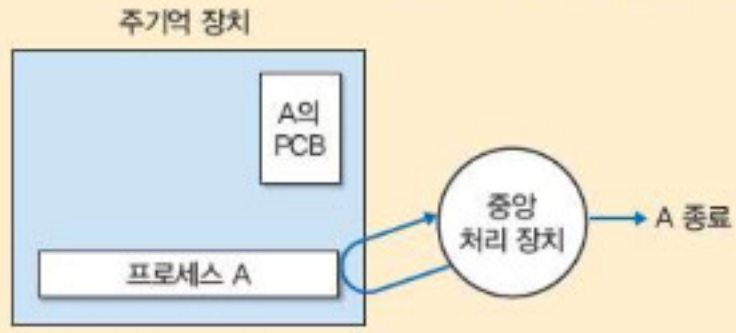
② 프로세스 A가 실행되고 시간 할당량(4ms)이 되면 준비 상태로 전환되고 프로세스 B가 중앙 처리 장치를 할당받아 실행된 후 같은 과정을 거쳐 프로세스 C가 실행되게 된다.



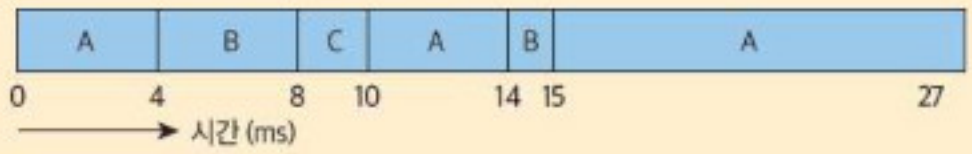
③ 프로세스 C는 2ms에 종료되기 때문에 다시 처음 프로세스인 A가 중앙 처리 장치를 할당받고 4ms 뒤에 프로세스 B가 중앙 처리 장치를 할당받는다.



④ 프로세스 B는 1ms에 종료되기 때문에 다시 프로세스 A가 중앙 처리 장치를 할당받고 종료될 때까지 실행된다.



⑤ 이런 과정을 시간을 축으로 하는 그림으로 나타내면 다음과 같다.



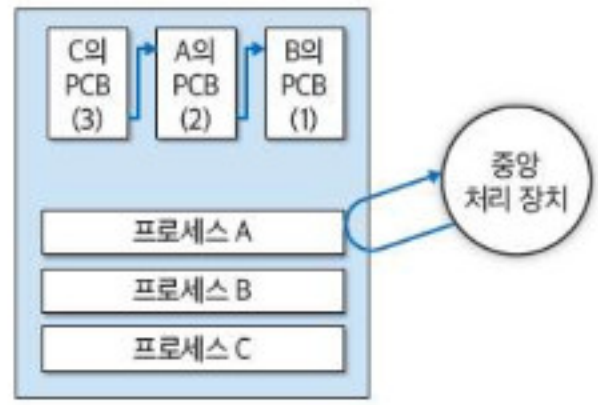
이때 프로세스 A의 대기 시간은 0+6(=10−4)+1(=15−14)=7ms, 프로세스 B는 4+6(=14−8)=10ms, 프로세스 C는 8ms다. 그러므로 평균 대기 시간은 다음과 같다.

$$\frac{7+10+8}{3} = 8.3(\text{ms})$$

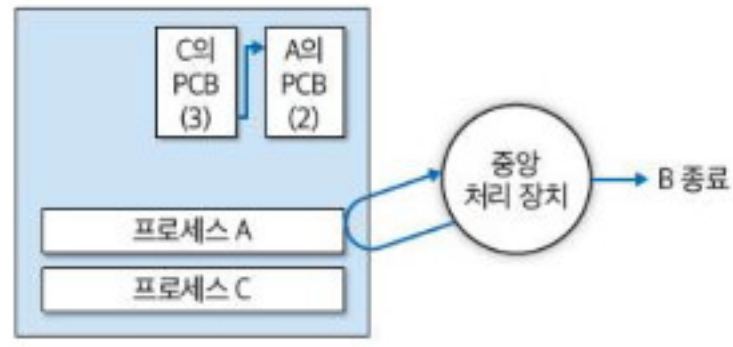
(3) 우선순위 스케줄링

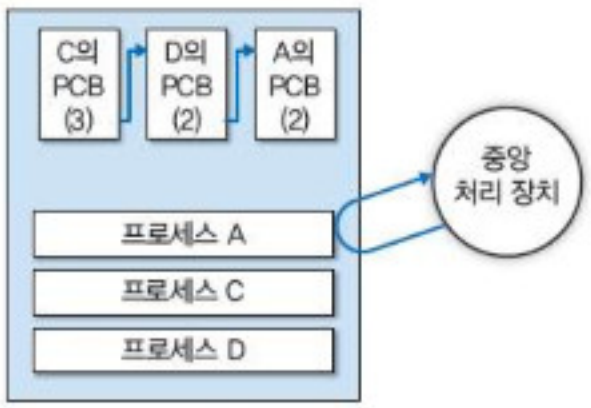
우선순위 스케줄링은 말 그대로 가장 높은 우선순위의 프로세스에게 먼저 중앙 처리 장치를 할당하는 방법이다. 우선순위는 프로세스 제어 블록에 저장된다.

① 만약 우선순위가 2, 1, 3인 프로세스 A, B, C가 있다면 운영체제는 우선순위가 가장 높은 프로세스 B에게 중앙 처리 장치를 배정한다.



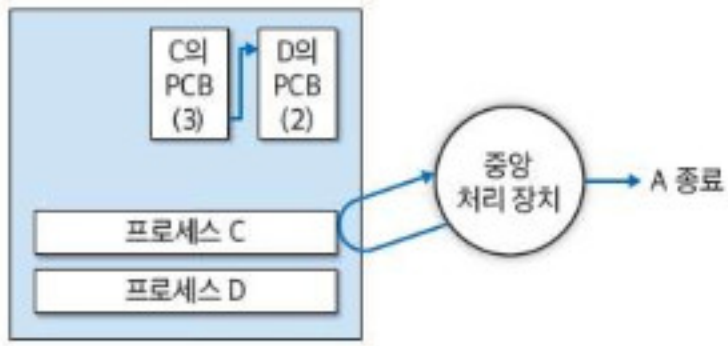
② 프로세스 B의 실행이 종료되면 다음으로 우선순위가 높은 프로세스 A를 실행한다.



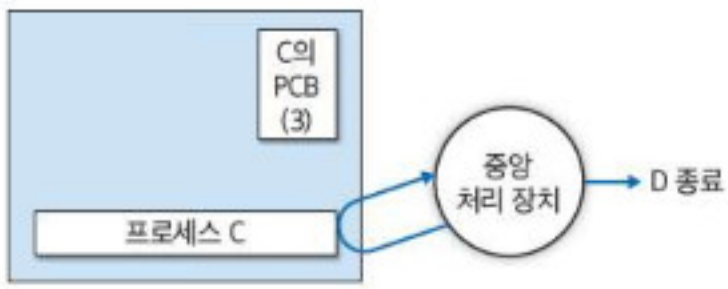


③ 이때 우선순위 2의 프로세스 D가 생성되면 프로세스 D의 프로세스 제어 블록이 준비 큐에 연결되는데, 우선순위가 2이므로 프로세스 C의 프로세스 제어 블록 앞에 위치한다.

④ 프로세스 A의 실행이 종료되면 프로세스 D가 실행된다.



⑤ 프로세스 D의 실행이 종료되면 프로세스 C가 실행된다.

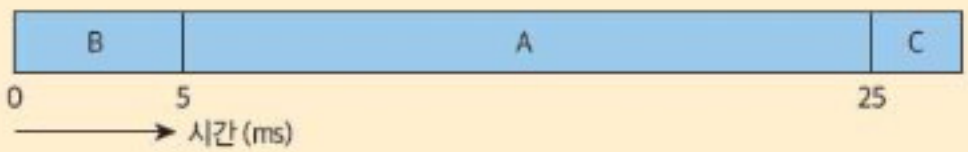


더 알아보기 우선순위 스케줄링의 대기 시간

아래 표와 같이 중앙 처리 장치 시간을 필요로 하는 세 개의 프로세스들이 있을 때의 대기 시간을 계산해 보면 다음과 같다.

프로세스	중앙 처리 장치 시간	우선 순위
프로세스 A	20ms	2
프로세스 B	5ms	1
프로세스 C	2ms	3

이런 A, B, C 세 개의 프로세스들은 다음과 같이 우선순위가 높은 순서로 실행된다.



이 경우 프로세스 B의 대기 시간은 0ms, 프로세스 A는 5ms, 프로세스 C는 25ms이므로 평균 대기 시간은 다음과 같이 된다.

$$\frac{0+5+25}{3} = 10(ms)$$

3 교착 상태

교착 상태(deadlock)란 두 개 이상의 작업이 다음 진행되어야 할 작업과 맞물려서 서로 상대방의 작업이 끝나기만을 기다리고 있기 때문에 결과적으로 아무것도 완료되지 못하는 상태를 가리킨다.

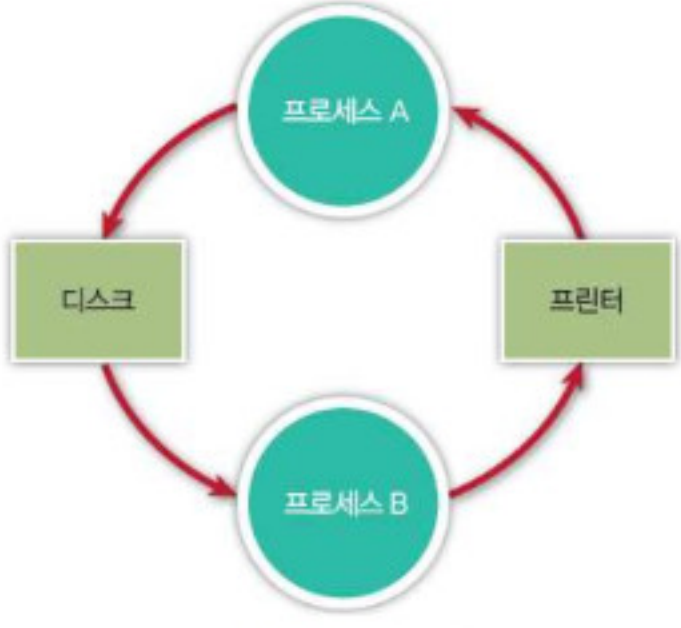


그림 Ⅱ-17 교착 상태의 예

예를 들어 프로세스 A가 프린터 자원을 점유하고 있으며 프로세스 B가 디스크 자원을 점유하고 있을 때 프로세스 A가 디스크 자원을 요청하고 프로세스 B가 프린터 자원을 요청하는 경우 각 자원이 사용중이므로 무한히 대기하게 되므로 교착 상태가 발생하게 된다.

(1) 교착 상태 발생 조건

교착 상태가 일어나려면 다음 네 가지 조건을 충족시켜야 한다.

① 상호 배제(Mutual exclusion) : 프로세스들이 필요로 하는 자원에 대해 배타적인 통제권을 요구한다.

② 점유 대기(Hold and wait) : 프로세스가 할당된 자원을 가진 상태에서 다른 자원을 기다린다.

③ 비선점(No preemption) : 프로세스가 어떤 자원의 사용을 끝낼 때까지 그 자원을 뺏을 수 없다.

④ 순환 대기(Circular wait) : 각 프로세스는 순환적으로 다음 프로세스가 요구하는 자원을 가지고 있다.

네 가지 조건 중 하나라도 만족하지 않으면 교착 상태는 발생하지 않는다.

(2) 교착 상태의 관리

현재 대부분의 운영체제들은 교착 상태를 막는 것은 불가능하기 때문에 교착 상태를 예방하거나 회피, 무시 또는 발견하는 방법으로 관리한다.

1 교착 상태의 예방

① 상호배제 조건의 제거

교착 상태는 두 개 이상의 프로세스가 공유 가능한 자원을 사용할 때 발생하는 것이므로 공유 불가능한, 즉 상호 배제 조건을 제거하면 교착 상태를 해결할 수 있다.

② 점유와 대기 조건의 제거

작업을 수행하기 전에 각 프로세스는 필요로 하는 모든 자원을 요청하여 획득하거나, 프로세스가 자원을 전혀 갖고 있지 않을 때만 자원을 요청할 수 있도록 하는 방법이다. 자원 과다 사용으로 인한 효율성, 프로세스가 요구하는 자원을 파악하는 데에 대한 비용, 자원에 대한 내용을 저장 및 복원하기 위한 비용, 기아 상태, 무한 대기 등의 문제점이 있다.

② 비선점 조건의 제거

비선점 프로세스에 대해 선점 가능한 프로토콜을 만들어 준다.

② 환형 대기 조건의 제거

자원의 종류에 일련의 순서를 부여하여 각 프로세스는 순서가 증가하는 순으로만 자원을 요청할 수 있도록 하는 방법이다.

2 교착 상태의 회피

자원이 어떻게 요청될지에 대한 추가 정보를 제공하도록 요구하는 것으로 교착 상태의 회피가 가능하며, 시스템에 순환 대기가 발생하지 않도록 자원 할당 상태를 검사한다.

교착 상태를 회피하기 위한 알고리즘으로 크게 두 가지가 있다.

① 자원 할당 그래프 알고리즘 (Resource Allocation Graph Algorithm)

② 은행원 알고리즘 (Banker's algorithm)

3 교착 상태의 무시

예방 혹은 회피 기법을 프로그래밍을 통해서 할 수 있으나 성능에 큰 영향을 미칠 수 있게 된다. 그렇기 때문에 교착 상태의 발생 확률이 비교적 낮은 경우 별다른 조치를 취하지 않는다.

4 교착 상태의 발견

감시/발견을 하는 검출 알고리즘으로 교착 상태 발생을 체크하는 방식으로 할 수 있으나 이 역시 성능에 큰 영향을 미칠 수 있다.

수행 활동

생활 주변의 프로세스 관리 기법 적용 발표하기

활동 주제

우리 생활 주변에서 프로세스 관리 기법을 적용하여 효율적으로 관리할 수 있는 것을 정리하여 발표한다.

준비와 과정

- 모둠을 설정한다.
  - 4명 ~ 5명의 모둠을 설정한다.
  - 모둠별로 모둠의 이름을 정하고, 모둠에서의 역할을 정한다.

    모둠 이름: 프로세스코리아

    모둠장: 기록자: 검색자: 발표자:
- 주변에서 프로세스 관리 기법을 적용할 수 있는 것을 찾아본다.
- 프로세스 관리 기법을 적용했을 때 효율적인지 정리해 본다.
- 정리한 프로세스 관리 기법 적용 자료를 발표자가 발표한다.
  - 다른 모둠이 발표할 때, 자신이 속한 모둠의 적용 자료와 비교해 본다.

활동 내용

1. 주변에서 프로세스 관리 기법을 적용할 수 있는 것을 찾아본다.

2. 프로세스 관리 기법을 적용했을 때 효율적인지 정리해 본다.

3. 프로세스 관리 기법을 개선할 방법이 있는지 정리해 본다.

CHAPTER

03 기억 장치 관리

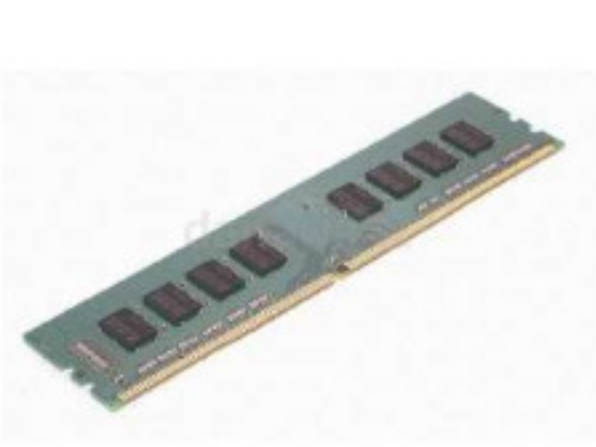
학습목표

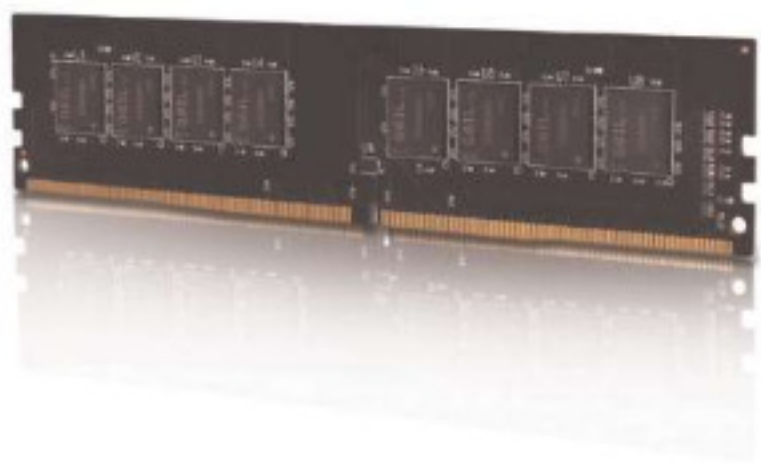
- 주기억 장치의 자원 관리 기법의 반입 기법과 배치 기법을 이해할 수 있다.
- 주기억 장치의 자원 관리 기법인 교체 기법을 이해할 수 있다.
- 단편화의 개념과 단편화를 해결하는 방법을 이해할 수 있다.

여는 이야기

RAM

모든 프로그램들은 주기억 장치에 들어온 프로세스를 통해 실행하게 된다. 따라서 프로그램을 실행하기 위해서는 보조 기억 장치에서 프로그램을 가져와서 주기억 장치에 배치하는 방법과 기존의 프로세스를 교체하는 방법이 필요하게 된다. 운영체제가 프로세스를 배치하는 방법과 교체하는 방법에 대해 생각해 보자.





단/원/학/습/안/내

이 단원에서는 주기억 장치의 반입 기법과 배치 기법을 이해하고 프로세스의 교체 전략을 살펴봄으로써 주기억 장치의 자원 관리 원리를 익히고 단편화를 해결하는 방법을 이해하여 직무를 수행하는 데 필요한 실무 능력을 기르도록 한다.

1 주기억 장치 반입 기법과 배치 기법

보조 기억 장치의 프로그램을 실행하기 위해서는 운영체제가 필요한 내용을 주기억 장치로 가져오는 방법과 가져온 내용을 주기억 장치의 적당한 위치에 배치하는 방법이 필요하게 된다.

(1) 반입 기법

반입 기법은 보조 기억 장치에 저장되어 있는 프로그램이나 데이터를 언제 주기억 장치로 가져올 것인지를 결정하는 기법을 말한다.

1 요구 반입(demand fetch)

요구 반입은 실행중인 프로그램이 보조 기억 장치에 있는 특정 프로그램이나 데이터 등의 참조를 요구할 때 주기억 장치로 가져오는 방법을 말한다.

2 예상 반입(anticipatory fetch)

예상 반입은 실행중인 프로그램에 의해 참조될 보조 기억 장치에 있는 특정 프로그램이나 데이터를 예상하여 미리 주기억 장치로 가져오는 방법을 말한다.

(2) 배치 기법

배치 기법은 보조 기억 장치로부터 가져온 프로그램이나 데이터를 주기억 장치의 어느 위치에 저장할 것인가를 결정하는 기법을 말하며, 최초 적합(first fit), 최적 적합(best fit), 최악 적합(worst fit) 등의 방법이 있다.

1 최초 적합

최초 적합 전략은 저장할 프로그램이나 데이터의 크기보다 큰 최초의 영역에 배치하는 방법을 말한다.

2 최적 적합

최적 적합 전략은 저장할 프로그램이나 데이터의 크기와 가장 유사한 영역에 배치하는 방법을 말한다.

3 최악 적합

최악 적합 전략은 저장할 프로그램이나 데이터의 크기가 들어갈 수 있는 영역 중 가장 큰 영역에 배치하는 방법을 말한다.

2 주기억 장치 교체 기법

프로그램이나 데이터가 필요에 의해 주기억 장치에 배치되어야 하는데 만약 배치할 수 있는 공간이 없다면 운영체제는 이미 사용되고 있는 영역 중에서 어느 영역을 교체하여 사용할 것인지를 결정하는 기법이 필요하게 된다. 대표적인 교체 기법으로는 FIFO(First In First Out), LRU(Least Recently Used), LFU(Least Frequently Used)가 있다.

(1) FIFO

FIFO 교체 기법은 이미 사용되고 있는 영역 중에서 주기억 장치에 배치된 시간이 가장 오래된 영역을 교체하는 기법이다.

다음 그림의 왼쪽과 같이 주기억 장치에 영역이 있다고 할 때 배치될 내용이 A, B, C, D 순서로 들어왔다고 하면 배치될 영역이 있는 C까지는 오른쪽 그림과 같이 배치될 수 있다.

운영체제 영역

들어온 순서대로 배치 시 →

운영체제 영역
A
B
C

더 알아보기 배치 기법의 적용

예를 들어 다음 그림과 같이 주기억 장치에 영역이 분할되어 있다고 할 때 저장할 크기가 110인 내용을 배치하는 각 전략의 배치 영역을 살펴보자.

운영체제 영역
150 (A)
사용
120 (B)
사용
80 (C)
사용
300 (D)

- 최초 적합

  크기가 110인 내용이 들어갈 수 있는 첫 번째 영역인 A에 배치하게 된다.

- 최적 적합

  크기가 110과 가장 유사한 영역인 B에 배치하게 된다.

- 최악 적합

  110인 내용이 들어갈 수 있는 영역 중 가장 큰 영역인 D에 배치하게 된다.

그렇지만 D의 경우는 배치될 공간이 없기 때문에 가장 먼저 배치된 A를 보조 기억 장치로 이동시키고 그 영역에 D를 오른쪽과 같이 배치하게 된다.

운영체제 영역
D
B
A

이와같은 FIFO 전략을 적용하기 위해서는 주기억 장치에 배치될 때 그 시간을 저장해야 한다.

(2) LRU

LRU 교체 전략은 이미 사용되고 있는 영역 중에서 주기억 장치에 배치된 다음, 가장 오랫동안 사용되지 않은 영역을 교체하는 전략이다.

다음 그림과 같이 주기억 장치에 A, B, C 순서로 배치되었고 내용이 A, B, C, C, C, C, B, A, A 순서로 사용되었다고 하면 아래 그림과 같이 표현할 수 있다.

운영체제 영역	1	2	3	4	5	6	7	8	9
A	V							V	V
B		V					V		
C			V	V	V	V			

현재 상황에서 D가 배치되어야 한다고 하면 가장 오랫동안 사용되지 않은 영역인 C를 보조 기억 장치로 이동시키고 그 영역에 D를 오른쪽 그림과 같이 배치하게 된다.

운영체제 영역
A
B
D

LRU 전략은 최근에 사용된 내용이 앞으로도 사용될 가능성이 많다는 것에 착안한 방법이며 적용하기 위해서는 배치된 영역이 주기억 장치에 접근할 때마다 그 시간을 저장해야 한다.

(3) LFU

LFU 교체 전략은 이미 사용되고 있는 영역 중에서 주기억 장치에 배치된 다음 가장 사용된 횟수가 적은 영역을 교체하는 전략이다.

다음 그림과 같이 주기억 장치에 A, B, C 순서로 배치되었고, 내용이 A, B, C, C, C, C, B, A, A 순서로 사용되었다고 하면 아래 그림과 같이 표현할 수 있다.

운영체제 영역	1	2	3	4	5	6	7	8	9
A	V							V	V
B		V					V		
C			V	V	V	V			

현재 상황에서 D가 배치되어야 한다고 하면 사용된 횟수가 가장 적은 영역인 B를 보조 기억 장치로 이동시키고 그 영역에 D를 오른쪽 그림과 같이 배치하게 된다.

운영체제 영역
A
D
C

LFU 전략은 많이 참조된 내용이 앞으로도 참조될 가능성이 많다는 것에 착안한 방법이며 적용하기 위해서는 배치된 영역이 주기억 장치에 접근할 때마다 그 횟수를 저장해야 한다.

3 단편화(fragmentation)

단편화는 기억 장치의 빈 공간 또는 데이터가 여러 조각으로 나뉘는 현상을 말한다. 단편화가 생기게 되면 기억 장치의 사용 가능한 저장 공간이 줄어들게 되며 내용을 가져오고 저장하는 속도가 늦어지는 문제점이 발생하게 된다.

(1) 단편화의 종류

단편화는 기억 장치의 저장 공간이 프로그램과 데이터를 배치하고 교체하는 과정에서 남게 되는 작은 조각 공간을 말한다. 크게 내부 단편화와 외부 단편화로 구분할 수 있다.

1 내부 단편화

내부 단편화는 필요한 크기에 비해서 실제 배정된 공간의 크기가 커서 해당 공간 내에서 사용하지 않는 부분이 있을 때 발생한다.

예를 들어 주기억 장치에 180만큼의 영역을 저장하려고 하는데 배정하는 기준이 100단위라고 가정한다면 200만큼의 영역을 배정받게 되기 때문에 20만큼의 단편화가 발생하게 된다.

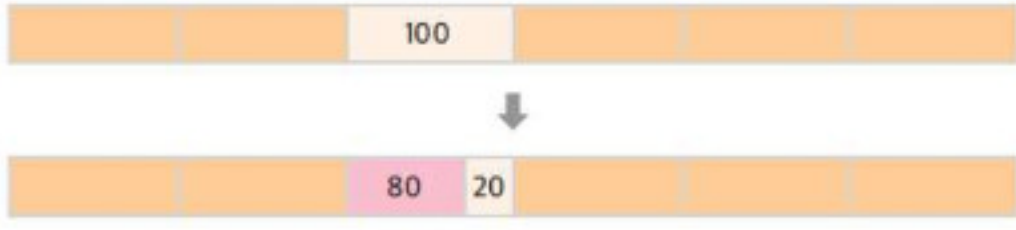


그림 Ⅲ-18 내부 단편화

2 외부 단편화

외부 단편화는 필요한 크기에 비해서 실제 배정받을 수 있는 저장 가능 공간이 작아서 사용되지 않는 부분이 있을 때 발생한다. 예를 들어 주기억 장치에 180만큼의 영역을 저장하려고 하는데 배정할 수 있는 공간이 100만큼의 나뉜 공간이 있다면 단편화가 발생하게 된다.

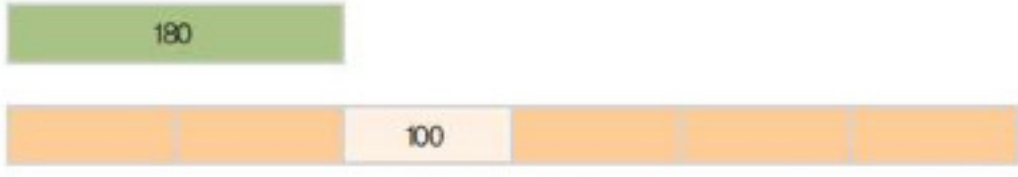


그림 Ⅲ-19 외부 단편화

(2) 단편화의 해결 방법

단편화를 해결하려면 단편화가 된 공간을 모아서 사용할 수 있는 공간으로 만들어야 한다. 단편화를 해결하는 방법은 크게 통합 기법과 압축 기법이 있다.

1 통합 기법

통합 기법은 인접해 있는 단편화된 공간을 하나의 공간으로 만드는 것을 말한다.

그림 Ⅲ-20 통합 기법

예를 들어 위의 그림과 같이 4개의 배정된 공간 중에서 가운데 2개의 공간이 사용할 수 있는 공간이 되었다면 인접한 2개의 공간을 합쳐 하나의 공간으로 만드는 방법이다.

2 압축 기법

압축 기법은 단편화된 빈 공간을 결합하여 사용할 수 있는 큰 공간으로 만드는 것을 말한다.

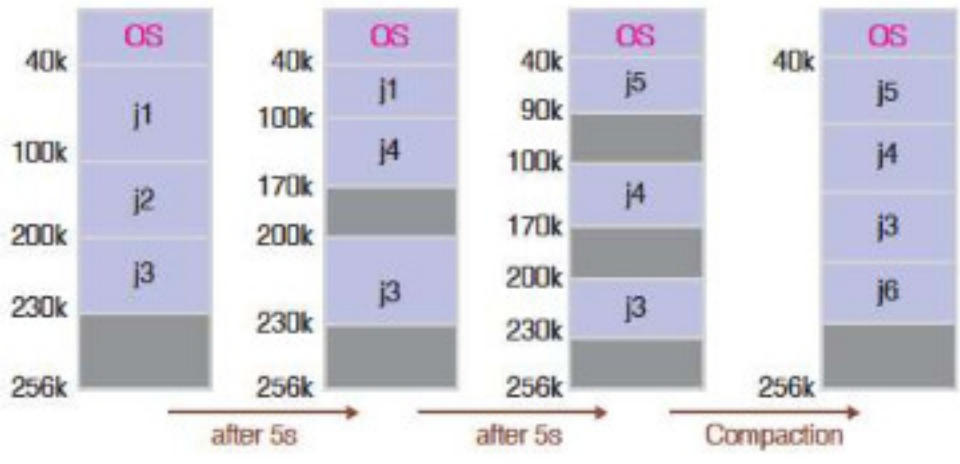


그림 Ⅲ-21 압축 기법

예를 들어 위의 그림과 같이 단편화가 진행된 경우 배정된 공간들을 가능한 인접할 수 있도록 이동시 남아 있는 공간을 가능한 크게 만드는 방법이다. 즉, 5초 후에 j2가 완료되어 j4가 배정되고 다시 5초 후에 j1이 완료되어 j5가 배정되면 j4, j3을 빈 공간으로 이동하여 남은 공간을 가능한 크게 만들고 다음 작업인 j6을 배정한다.

수행 활동

교체 전략 개선 방안 발표하기

활동 주제

주기억 장치 교체 전략을 개선할 수 있는 방법을 모둠별로 생각해 보고 이를 정리하여 발표한다.

준비와 과정

- 모둠을 설정한다.
  - 4명 ~ 5명의 모둠을 설정한다.
  - 모둠별로 모둠의 이름을 정하고, 모둠에서의 역할을 정한다.

    모둠 이름:

    모둠장: 기록자: 검색자: 발표자:
- 주기억 장치 교체 전략을 개선할 수 있는 방법을 생각해 본다.
- 기존의 방법과 비교하여 장점과 단점을 분석하여 정리해 본다.
- 정리한 주기억 장치 교체 전략을 발표자가 발표한다.
  - 다른 모둠이 발표할 때, 자신이 속한 모둠의 적용 자료와 비교해 본다.

활동 내용

1. 주기억 장치 교체 전략을 개선할 수 있는 방법을 생각하여 정리해 본다.

2. 기존의 방법과 비교하여 어떤 면에서 효율적인지 정리해 본다.

3. 우리 모둠의 방법에서 발생할 수 있는 문제가 있는지 정리해 본다.

CHAPTER

04 보조 기억 장치 관리

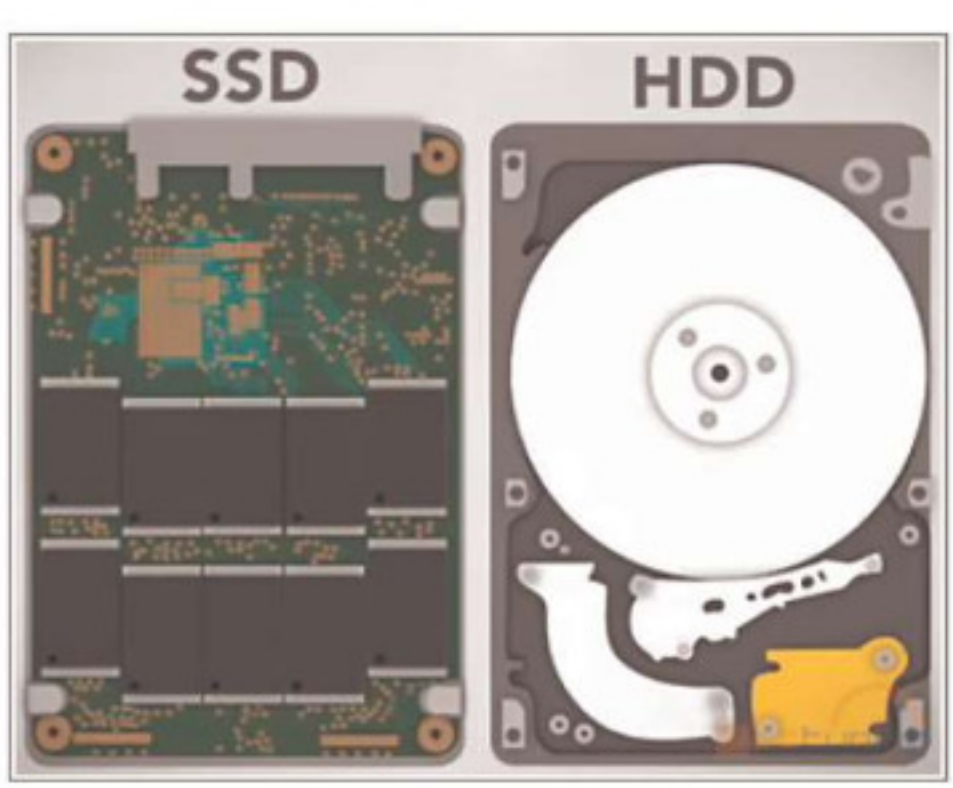
학습목표

- 보조 기억 장치 관리의 개념을 이해할 수 있다.
- 디스크 스케줄링의 개념과 종류를 이해할 수 있다.
- 디스크 스케줄링의 기법을 이해할 수 있다.

여는 이야기

HDD, SSD

주기억 장치의 용량이 크지 않고 전원이 없으면 내용이 삭제되기 때문에 보조 기억 장치가 반드시 필요하게 된다. 그 중에서 가장 많이 사용되고 있는 디스크의 접근 방법을 이해하고 자료에 접근하기 위한 스케줄링 방법에 대해 생각해 보자.



대표적인 보조 기억 장치는 속도가 빠른 SSD와 용량이 큰 HDD가 있다.

단/원/학/습/안/내

이 단원에서는 보조 기억 장치의 자료 접근 방법을 이해하고 가장 많이 사용되고 있는 디스크의 자료에 접근하는 방법을 살펴봄으로써 효율적으로 스케줄링하는 방법을 이해하여 직무를 수행하는 데 필요한 실무 능력을 기르도록 한다.

1 보조 기억 장치 관리의 개요

주기억 장치는 공간이 제한적이고 전원이 끊어지면 저장된 내용이 지워지는 휘발성 기억 장치이기 때문에 프로그램과 데이터를 계속 저장할 수 없다. 그래서 운영체제는 보조 기억 장치를 이용해 프로그램과 데이터를 영구적으로 저장한다.

(1) 보조 기억 장치 관리의 개념

보조 기억 장치는 주기억 장치에 비해 속도가 느리기 때문에 프로그램과 데이터를 효율적으로 저장하고 읽을 수 있는 방법을 찾는 것으로 탐색 시간을 최소화할 수 있다. 그러므로 보조 기억 장치 관리를 위한 스케줄링 방법은 매우 중요하다.

(2) 보조 기억 장치 자료 접근 방법

보조 기억 장치는 자료 접근 방법에 따라 순차 접근 방식(Sequential Access)과 직접 접근 방식(Direct Access)으로 구분한다.

1 순차 접근 방식

순서대로 진행하며 접근하는 방식을 말하며 대표적으로 테이프 장치가 있다.

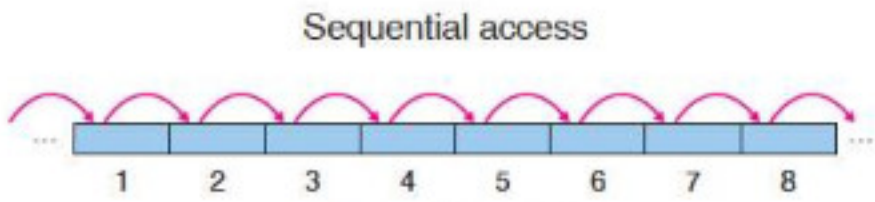


그림 Ⅲ-22 순차 접근 방식

1부터 순서대로 진행하며 접근하게 된다.

2 직접 접근 방식

저장 위치에 관계없이 직접 접근이 가능한 방식을 말하며 대표적으로 디스크 장치가 있다.

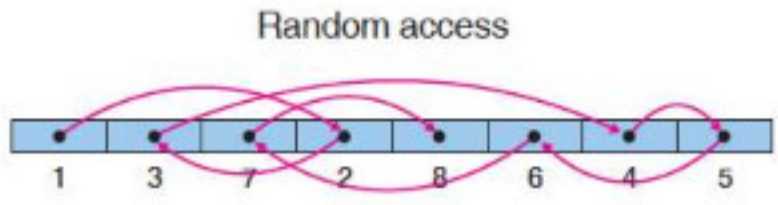


그림 Ⅲ-23 직접 접근 방식

1부터 임의로 직접 접근하며 순서대로 진행한다.

2 디스크 스케줄링

보조 기억 장치에는 자기 디스크, 광디스크, 자기 테이프 등이 있지만 가장 많이 사용되고 있는 것은 자기 디스크이다. 그리고 순차 접근 방식의 보조 기억 장치는 순서대로 접근하기 때문에 스케줄링의 개념이 필요하지 않다.

(1) 디스크 스케줄링의 개념

디스크 스케줄링은 필요한 프로그램과 데이터가 디스크 여러 곳에 저장되어 있을 경우 자료에 접근하기 위해 디스크의 헤드가 움직이는 경로를 결정하는 기법을 말한다. 디스크 스케줄링은 일반적으로 탐색 시간을 최적화하기 위해 수행되며 일정 시간에 요구를 최대한 많이 처리하고 요청에 대한 결과가 나오는 시간을 짧게 하며 각 요청의 응답 시간들의 차이가 많이 나지 않도록 하려는 목적을 가지고 있다.

(2) 디스크 스케줄링 기법

디스크 스케줄링 기법에는 FCFS(First Come First Served), SSTF(Shortest Seek Time First), SCAN, C-SCAN(Circular SCAN) 등이 있다.

1 FCFS 스케줄링

FCFS 스케줄링은 가장 간단한 스케줄링 기법으로 요청 대기 큐에 먼저 들어온 요청이 먼저 서비스를 받는 기법이다. 이 기법은 탐색 거리를 최소화하지 못한다는 단점이 있다.

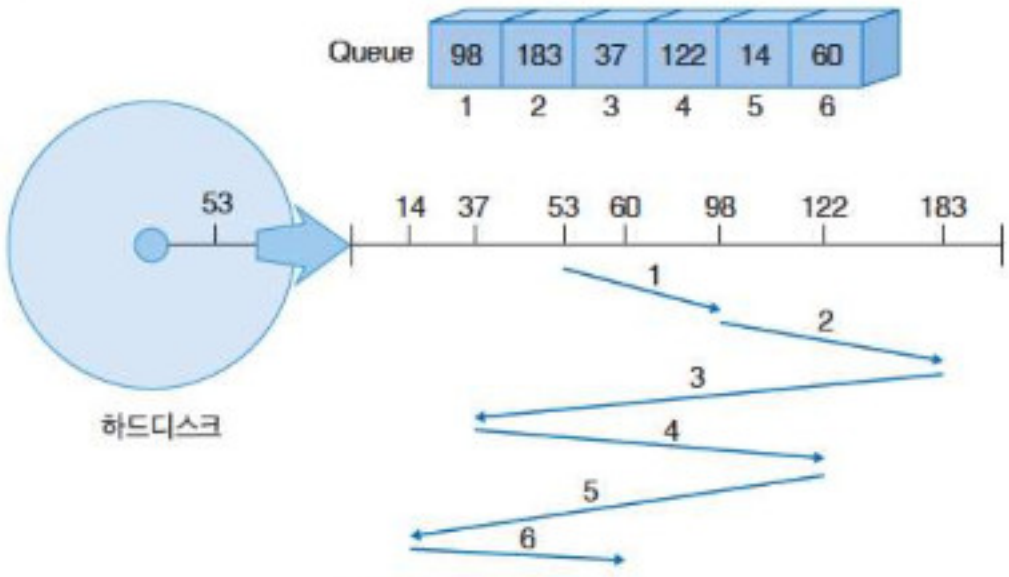


그림 Ⅲ-24 FCFS 스케줄링

53에서 출발하여 가장 가까운 60으로 이동하고 다음으로 가까운 37, 14로 이동한다. 그리고 안쪽으로 98, 122, 183 순서로 이동한다.

2 SSTF 스케줄링

SSTF 스케줄링은 FCFS 스케줄링의 단점을 보완하기 위한 기법으로 탐색 거리가 가장 짧은 요청이 먼저 서비스를 받는 기법이다. 이 기법은 응답 시간의 편차가 커서 요청에 대한 서비스를 계속 기다리게 되는 기아 상태가 발생할 수 있다는 단점이 있다.

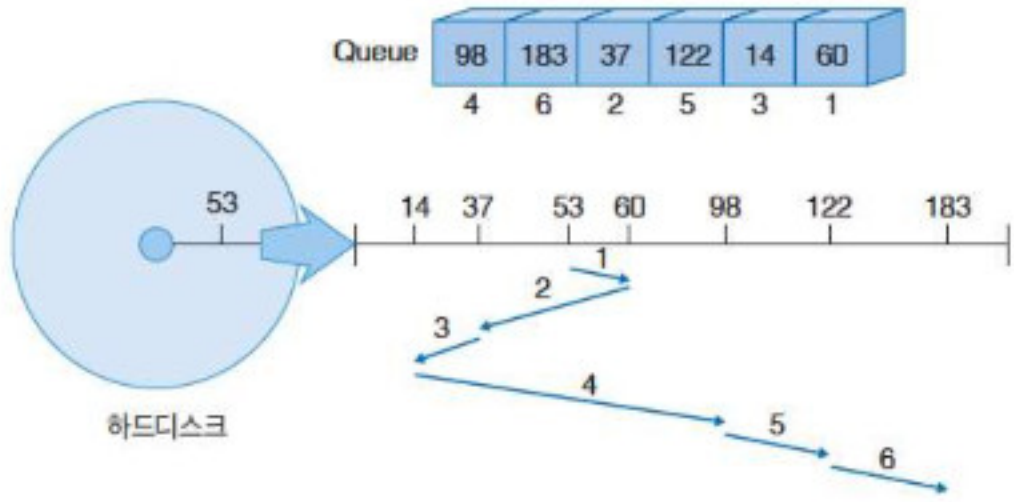


그림 Ⅲ-25 SSTF 스케줄링

53에서 출발하여 진행 방향으로 가장 가까운 60으로 이동하고 다음으로 가까운 37, 14로 이동한다. 그리고 안쪽으로 98, 122, 183 순서로 이동한다.

3 SCAN 스케줄링

SCAN 스케줄링은 SSTF 스케줄링의 단점인 응답 시간의 편차를 줄이기 위한 기법으로 진행 방향의 짧은 거리에 있는 요청을 먼저 처리하는 기법이다. 헤드가 요청된 트랙에 대해 디스크의 한 끝에서 다른 끝으로, 다른 한쪽 끝에 도달했을 때는 역방향으로 이동하면서 처리한다. 이 기법은 진행 방향으로 요청이 오면 바로 서비스 받을 수 있지만 진행 반대 방향이라면 헤드가 끝까지 이동하고 나서 되돌아 올 때까지 기다려야 하는 단점이 있다.

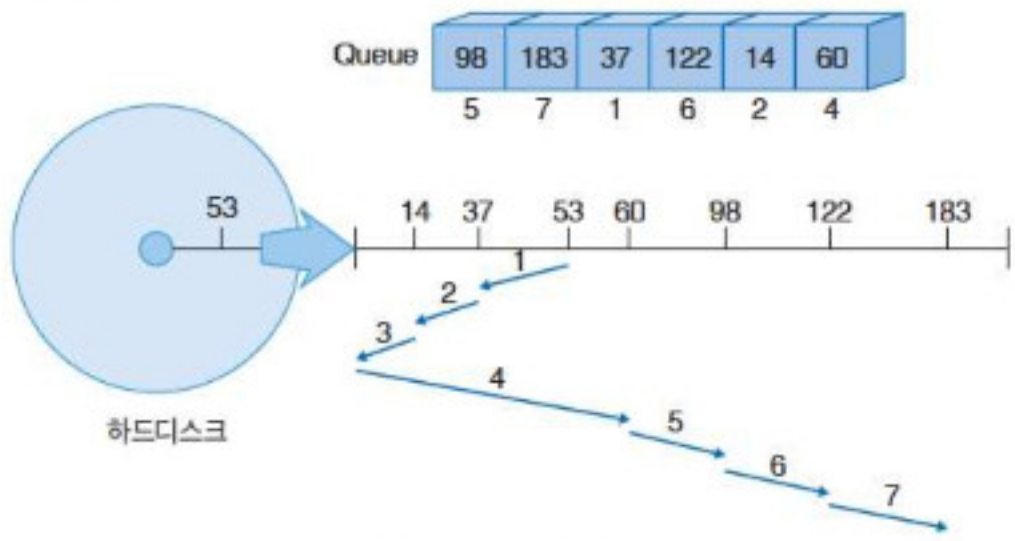


그림 Ⅲ-26 SCAN 스케줄링

53에서 출발하여 탐색 거리가 짧은 37로 이동하고 다음 37에서 탐색 거리가 짧은 14로 이동한다. 다시 반대쪽인 60, 98, 122, 183으로 이동한다.

4 C-SCAN 스케줄링

C-SCAN 스케줄링은 SCAN 스케줄링을 수정한 기법으로 헤드가 항상 바깥쪽에서 안쪽으로 이동하면서 가장 짧은 탐색 거리를 갖는 요청을 먼저 처리하는 기법이다.

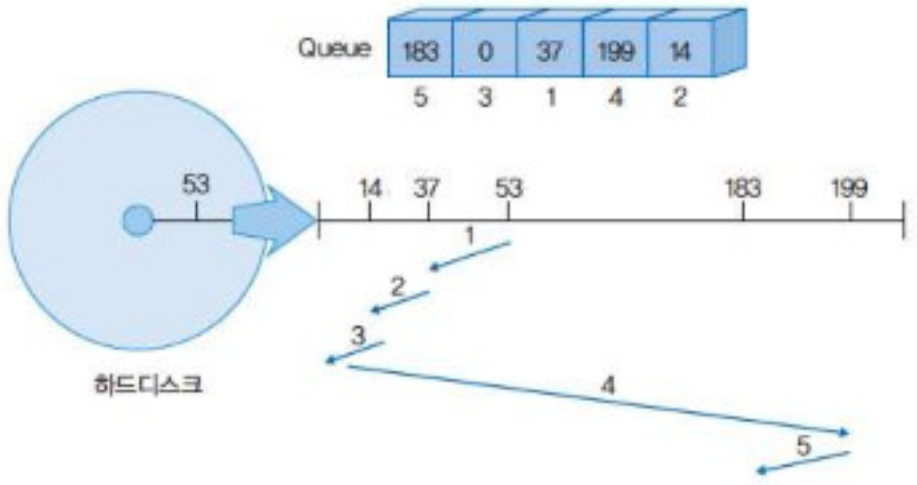


그림 Ⅲ-27 C-SCAN 스케줄링

53에서 출발하여 안쪽인 37, 14으로 이동하고 가장 안쪽 트랙인 0으로 이동했다가 다시 바깥쪽 트랙인 199로 이동 후 183으로 이동한다.

수행 활동

디스크 스케줄링 기법 개선 방안 발표하기

활동 주제

디스크 스케줄링 기법을 개선할 수 있는 방법을 모둠별로 생각해 보고 이를 정리하여 발표한다.

준비와 과정

- 모둠을 설정한다.
  - 4명 ~ 5명의 모둠을 설정한다.
  - 모둠별로 모둠의 이름을 정하고, 모둠에서의 역할을 정한다.

    모둠 이름:

    모둠장: 기록자: 검색자: 발표자:
- 디스크 스케줄링 기법을 개선할 수 있는 방법을 생각해 본다.
- 기존의 방법과 비교하여 장점과 단점을 분석하여 정리해 본다.
- 정리한 디스크 스케줄링 기법을 발표자가 발표한다.
  - 다른 모둠이 발표할 때, 자신이 속한 모둠의 적용 자료와 비교해본다.

활동 내용

1. 디스크 스케줄링 기법을 개선할 수 있는 방법을 생각하여 정리해 본다.

2. 기존의 방법과 비교하여 어떤 면에서 효율적인지 정리해 본다.

3. 우리 모둠의 방법에서 발생할 수 있는 문제가 있는지 정리해 본다.

CHAPTER

05 가상 기억 장치 관리

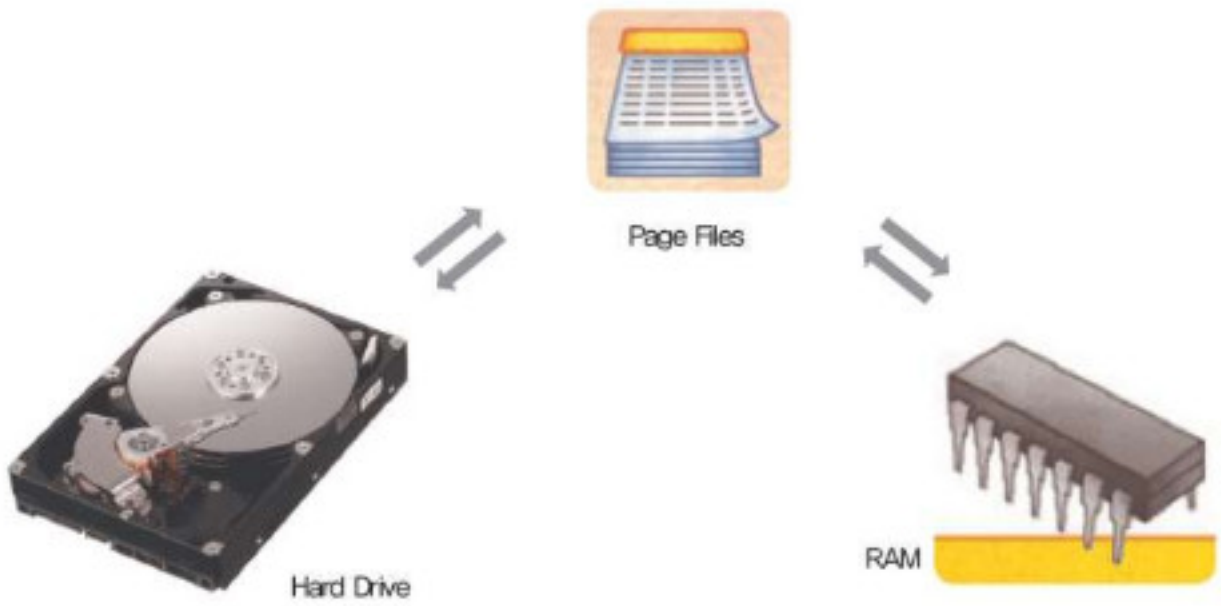
학습목표

- 가상 기억 장치의 개념을 이해할 수 있다.
- 가상 기억 공간을 이용할 경우의 장단점을 이해할 수 있다.
- 컴퓨팅 시스템에서 가상 메모리를 설정하고 관리할 수 있다.

여는 이야기

가상 기억 장치

주기억 장치의 용량이 제한되어 있기 때문에 프로그램에서 필요한 용량이 큰 경우 보조 기억 장치의 일부를 주기억 장치처럼 사용하는 방법이 가상 기억 장치이다. 자신이 사용하고 있는 운영체제에서 가상 기억 장치를 관리하는 방법에 대해 알아보자.



[Page Files을 이용한 가상 기억 장치]

단/원/학/습/안/내

이 단원에서는 가상 기억 장치의 장점과 단점을 이해하고 우리가 사용하고 있는 운영체제의 가상 기억 장치를 관리하는 방법을 살펴봄으로써 가상 기억 장치를 효율적으로 활용하는 방법을 이해하여 직무를 수행하는 데 필요한 실무 능력을 기르도록 한다.

1 가상 기억 장치의 개요

가상 기억 장치는 실행될 프로그램이나 데이터가 주기억 장치보다 크거나 여러 개인 경우 주기억 장치 공간이 부족하여 프로그램이 제대로 실행되지 못하는 상황이 발생할 것을 막기 위해 실행에 필요한 부분만 주기억 장치에 저장하고 나머지는 보조 기억 장치에 두고 동작하도록 하여 문제를 해결하는 방법을 말한다.

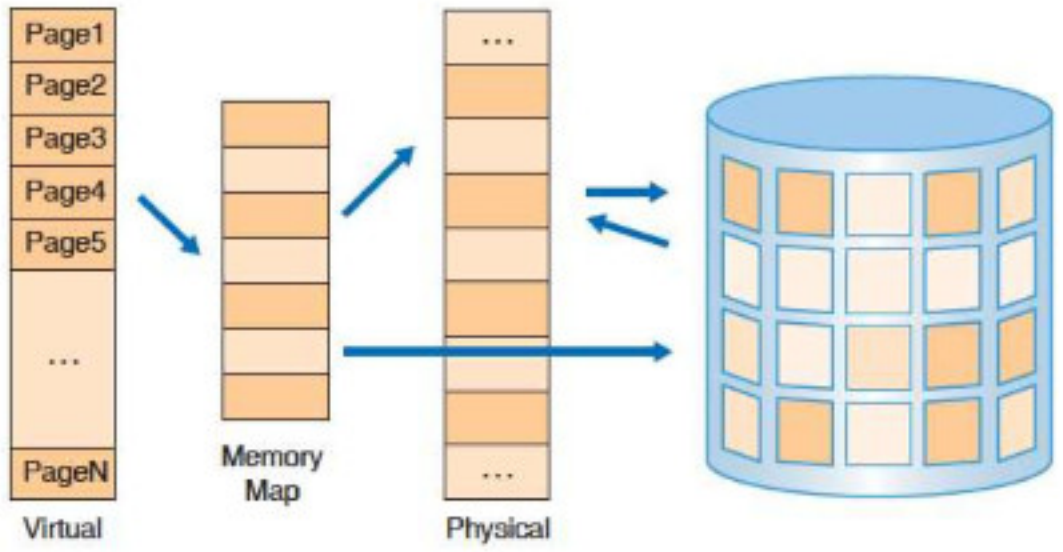


그림 Ⅲ-28 가상 기억 장치

주기억 장치의 내용을 메모리 지도를 통해 가상 기억 장치의 내용과 연결하여 활용할 수 있다.

가상 기억 장치는 다음과 같은 장단점이 있다.

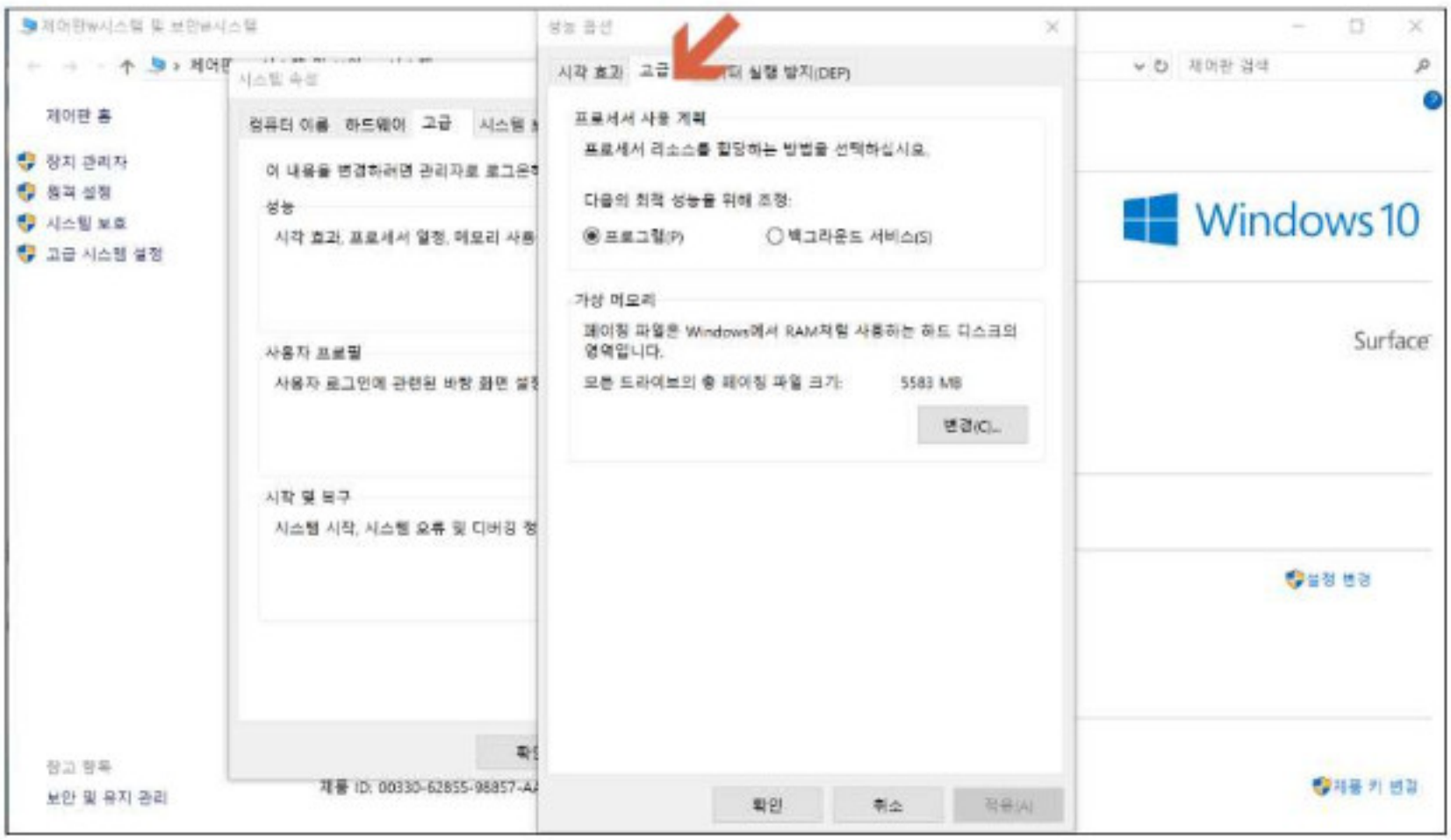
장점		단점
시스템 관리적 측면의 장점 • 어떠한 크기를 가진 프로세스도 주기억 장치 용량에 구애 받지 않고 실행 가능하다. • 우선 순위를 고려하여 물리적 기억 장치 할당하여 실행 속도를 조절할 수 있다.	**사용자 입장의 장점** • 프로그래머들이 가상 기억 장치 관리를 위한 세부적 사항들을 고려할 필요가 없다.	• 가상 기억 장치는 일반적으로 가상 기억 장치를 채택하지 않는 시스템에서의 실행 속도보다 느리며 소프트웨어를 통해 추가로 관리해야 한다.

2 가상 메모리 관리

Windows 운영체제에서는 가상 메모리를 관리할 수 있는 방법을 제공하고 있다.

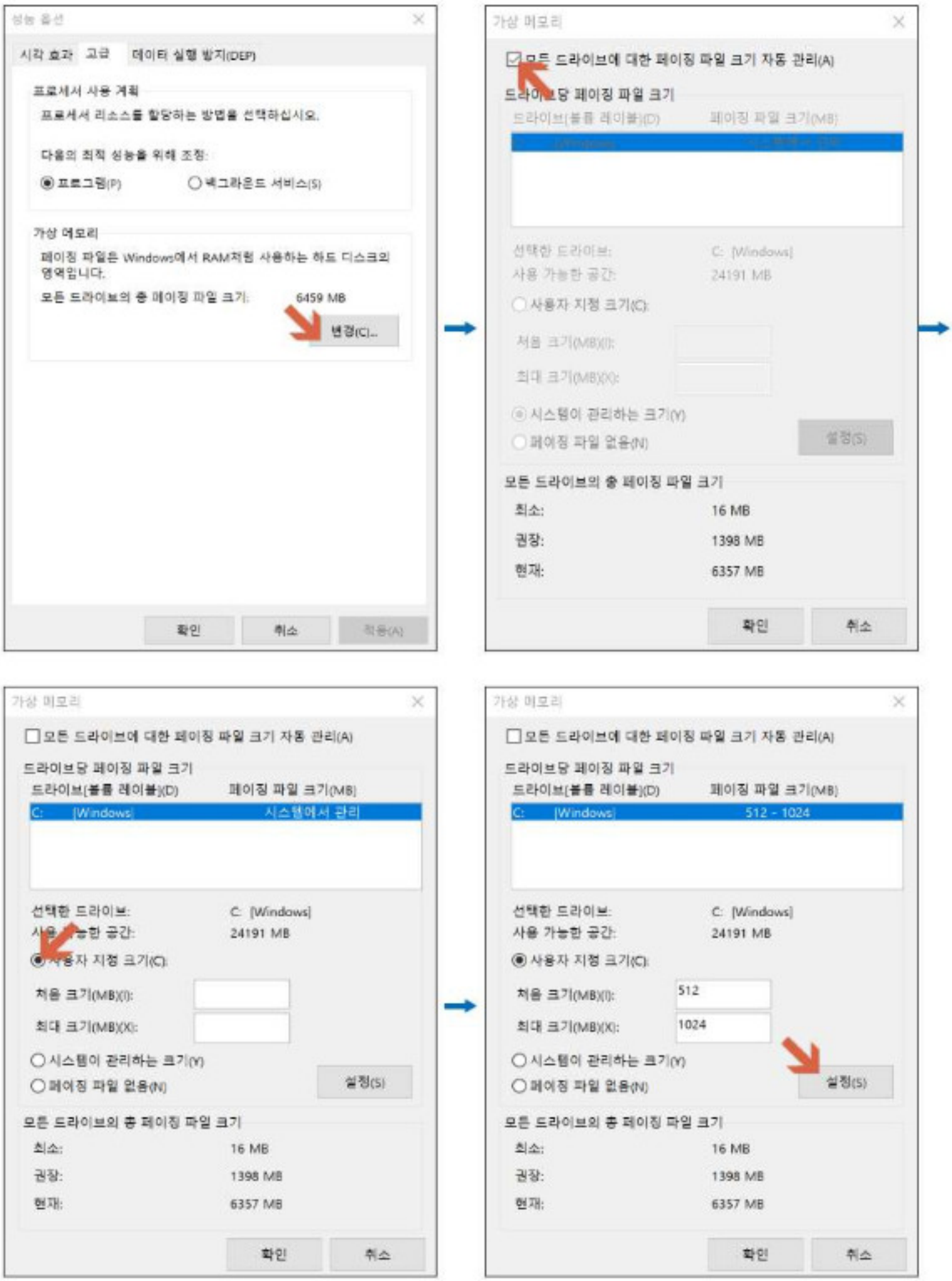
(1) 가상 메모리 실행 순서(Windows에서의 예)

탐색기를 실행 ➡ 컴퓨터 ➡ 속성 ➡ 고급 시스템 설정 ➡ 고급 탭 ➡ 성능(설정) 버튼 ➡ 고급 탭 ➡ 가상 메모리(변경) 버튼 수행



(2) 가상 메모리 관리 방법

시스템이 자동으로 관리하는 방법을 사용하지 않고 크기를 직접 설정한다.



수행 활동

실생활에서 가상 기억 장치와 유사한 내용 발표하기

활동 주제

실생활에서 가상 기억 장치와 유사하게 활용되는 것을 모둠별로 생각해 보고 이를 정리하여 발표한다.

준비와 과정

- 모둠을 설정한다.
  - 4명 ~ 5명의 모둠을 설정한다.
  - 모둠별로 모둠의 이름을 정하고, 모둠에서의 역할을 정한다.

    모둠 이름:

    모둠장: 기록자: 검색자: 발표자:
- 실생활에서 가상 기억 장치와 유사하게 활용되는 방법을 생각해 본다.
- 가상 기억 장치와 유사한 점과 차이점을 정리해 본다.
- 정리한 가상 기억 장치와 유사한 점과 차이점을 발표자가 발표한다.
  - 다른 모둠이 발표할 때, 자신이 속한 모둠의 적용 자료와 비교해 본다.

활동 내용

1. 실생활에서 가상 기억 장치와 유사하게 활용되는 것을 정리해 본다.

2. 실생활과 가상 기억 장치의 유사한 점을 정리해 본다.

3. 실생활과 가상 기억 장치의 차이점을 정리해 본다.

CHAPTER

06 입출력과 파일 관리

학습목표
- 입출력 자원 관리 방법을 이해할 수 있다.
- 파일의 자원 관리 방법을 이해할 수 있다.

여는 이야기

파일 관리

입출력 장치와 파일을 운영체제에서 사용하기 위해서는 중앙 처리 장치와 주기억 장치 등과 속도의 차이를 극복하기 위한 방법이 필요하다. 입출력 버퍼링과 파일을 운용하는 방법에 대해 알아보자.

[Windows에서의 예]

단/원/학/습/안/내

이 단원에서는 운영체제에서 입출력을 실행할 때의 속도 차리를 극복하기 위한 방법을 이해하고 파일을 운용 및 관리하는 방법을 살펴봄으로써 효율적으로 파일을 활용하는 방법을 이해하여 직무를 수행하는 데 필요한 실무 능력을 향상시킬 수 있도록 한다.

1 입출력 자원 관리 방법

입출력 자원 관리는 운영체제와 사용자 간의 인터페이스를 말한다. 입출력 장치는 제어 카드에 연결하고 제어 프로그램을 장치 관리자에 설치하여 작동한다.

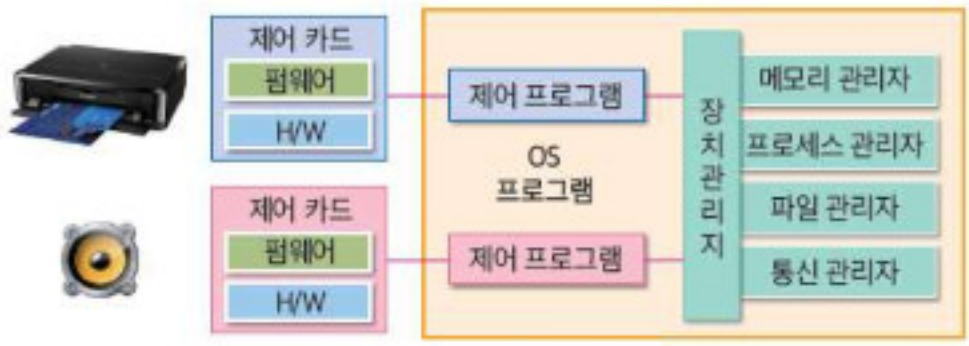


그림 Ⅱ-29 입출력 자원 관리

(1) 입출력 방법

초기의 컴퓨터에서는 중앙 처리 장치가 프로그램을 실행하는 도중 입출력 명령을 만나게 되면 직접 처리하는 방식을 사용하였으나 현재는 프로그램을 실행하는 도중 입출력 명령을 만나게 되면 전용 프로세서가 담당하여 처리하고 중앙 처리 장치는 그 동안 다른 프로그램을 수행하게 된다.

스풀링(spooling)

스풀링은 중앙 처리 장치와 입출력 장치의 속도 차이로 발생하는 대기 시간을 줄이기 위한 기법으로 스풀러(spooler) 프로그램이 인쇄하기 위한 파일들을 저장하고 있다가 순서대로 출력을 하는 방식이다.

(2) 입출력 버퍼링

입출력이 실행될 때마다 중앙 처리 장치가 명령을 실행하면 너무 많은 부하가 생기기 때문에 버퍼링을 이용하여 수행하는 것이 효율적이다.

1 단일 버퍼링

하나의 버퍼를 이용하여 입출력을 실행하게 되며 중앙 처리 장치는 버퍼를 채우고 비울 동안 기다려야 한다.

2 더블 버퍼링

하나의 버퍼를 채우는 동안 다른 버퍼의 데이터를 사용한다.

3 순환 버퍼링

순환 큐 데이터 구조를 이용한 버퍼링 방법으로 여러 개의 버퍼가 원형으로 데이터를 사용하므로 중앙 처리 장치는 거의 쉬지 않고 입출력을 수행한다.

2 파일의 자원 관리

(1) 파일 시스템의 기능

파일 시스템은 다음과 같은 기능을 제공해야 한다.

- 사용자가 파일을 생성하고 수정하며 제거할 수 있도록 한다.
- 여러 사용자가 공동으로 사용할 수 있도록 한다.
- 사용자의 목적에 적합한 구조로 파일을 구성할 수 있게 한다.
- 불의의 사고에 대비한 백업과 복구 능력이 있다.
- 파일의 무결성, 보안 유지 방안을 제공해야 한다.
- 주기억 장치와 보조 기억 장치 간의 파일 전송을 담당한다.

(2) 파일 자원 관리의 주요 기능

운영체제는 파일 자원을 관리하기 위해 다음과 같은 기능을 제공해야 한다..

1 디스크의 분할 및 포맷

- 디스크의 표면을 파티션으로 분할하거나 삭제할 수 있어야 한다.
- 파티션 영역을 사용할 수 있도록 주소를 할당하는 포맷 기능이 있어야 한다.

2 파일과 폴더(디렉터리)의 생성

- 파티션과 포맷이 완료되면 파일을 생성할 수 있어야 한다.
- 파티션 영역에 트리 형태의 파일 구조를 생성할 수 있어야 한다.

3 드라이브와 폴더의 내용 출력

- 드라이브와 폴더의 내용을 화면에 나타낼 수 있어야 한다.

4 파일과 폴더(디렉터리)의 이름 변경

- 파일과 폴더(디렉터리)의 이름을 지정하여 변경할 수 있어야 한다.

5 파일과 폴더(디렉터리)의 복사 / 삭제 / 이동

- 파일과 폴더(디렉터리)를 원하는 위치에 복사할 수 있어야 한다.
- 파일과 폴더(디렉터리)를 삭제할 수 있어야 한다.
- 파일과 폴더(디렉터리)를 이동할 수 있어야 한다.

수행 활동

실생활에서 파일 저장 방식과 유사한 내용 발표하기

활동 주제

실생활에서 파일 저장 방식과 유사하게 활용되는 것을 모둠별로 생각해 보고 이를 정리하여 발표한다.

준비와 과정

- 모둠을 설정한다.
  - 4명 ~ 5명의 모둠을 설정한다.
  - 모둠별로 모둠의 이름을 정하고, 모둠에서의 역할을 정한다.

    모둠 이름:

    모둠장: 기록자: 검색자: 발표자:
- 실생활에서 운영체제의 파일 자원 관리와 유사한 사례를 생각해 본다.
- 자신만의 파일 관리 방법을 정리해 본다.

활동 내용

1. 실생활에서 운영체제가 파일 자원을 관리하는 것과 유사한 사례를 찾아 발표해 보자.

2. 자신이 사용하는 컴퓨터의 파일 관리 방법을 발표해 보자.

Ⅲ 단원 학습 정리

01. 운영체제의 구성

(1) **운영체제의 구성:** 커널, 사용자 인터페이스, 필수 유틸리티

(2) **운영체제의 종류**

- 데스크톱 운영체제: Windows, Mac OS, Linux
- 서버용 운영체제: Unix, Windows 서버, Linux 서버
- 모바일용 운영체제: Android, iOS, Windows 모바일, Linux 모바일

(3) **운영체제 운영 방식**

- 명령 처리 방식에 따른 분류: 멀티태스킹, 분산, 실시간 등
- 운영 시스템에 따른 분류: 개인용 컴퓨터, 메인프레임, 임베디드 등

(4) **컴퓨팅 시스템 성능 저해 요인:** 프로그램 설치/제거 반복, 많은 프로그램의 실행 등

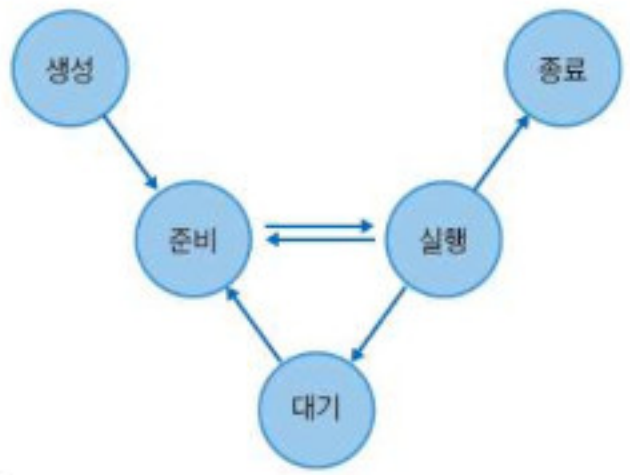
(5) **성능 저해 관리 방법:** 악성 코드와 바이러스 점검, 디스크 정리와 디스크 조각모음, 불필요한 프로그램과 서비스 삭제 등

02. 프로세스 관리

(1) **자원:** 컴퓨터에서 사용될 수 있는 항목. 예) 프린터, 디스크, 메모리, 네트워크, 소프트웨어 등

(2) **프로세스:** 실행되기 위해 주기억 장치에 들어간 프로그램

(3) **프로세스의 상태:**



(4) **프로세스 스케줄링**

- FCFS 스케줄링: 먼저 도착한 프로세스를 먼저 실행하는 방법
- 라운드 로빈 스케줄링: 프로세스가 종료될 때까지 중앙 처리 장치를 차지하는 것이 아니라 여러 프로세스들이 중앙 처리 장치를 돌아가며 할당받아 실행되는 방법
- 우선순위 스케줄링: 가장 높은 우선순위의 프로세스에게 먼저 중앙 처리 장치를 할당하는 방법

(5) **교착 상태:** 두 개 이상의 작업이 서로 상대방의 작업이 끝나기만을 기다리고 있기 때문에 결과적으로 아무것도 완료되지 못하는 상태

- 교착 상태 발생 조건: 상호 배제, 점유 대기, 비선점, 순환 대기
- 교착 상태 관리: 교착 상태의 예방, 교착 상태의 회피, 교착 상태의 무시, 교착 상태의 발견

03. 주기억 장치 관리

(1) **주기억 장치 반입 기법:** 요구 반입, 예상 반입

(2) **주기억 장치 배치 전략:** 최초 적합, 최적 적합, 최악 적합

(3) **주기억 장치 교체 전략**

- FIFO: 이미 사용되고 있는 영역 중에서 주기억 장치에 배치된 시간이 가장 오래된 영역을 교체하는 전략
- LRU: 이미 사용되고 있는 영역 중에서 주기억 장치에 배치된 다음 가장 오랫동안 사용되지 않은 영역을 교체하는 전략
- LFU: 이미 사용되고 있는 영역 중에서 주기억 장치에 배치된 다음 가장 사용된 횟수가 적은 영역을 교체하는 전략

(4) **단편화의 종류:** 내부 단편화, 외부 단편화

(5) **단편화 해결 기법:** 통합 기법, 압축 기법

04. 보조 기억 장치 관리

(1) **보조 기억 장치 자료 접근 방법:** 순차 접근 방식, 직접 접근 방식

(2) **디스크 스케줄링 기법**

- FCFS: 요청 대기 큐에 먼저 들어온 요청이 먼저 서비스를 받는 기법
- SSTF: 탐색 거리가 가장 짧은 요청이 먼저 서비스를 받는 기법
- SCAN: 진행 방향의 짧은 거리에 있는 요청을 먼저 처리하는 기법
- C-SCAN: 헤드가 항상 바깥쪽에서 안쪽으로 이동하면서 가장 짧은 탐색 거리를 갖는 요청을 먼저 처리하는 기법

05. 가상 기억 장치 관리

(1) **가상 기억 장치의 장점:** 물리적 용량에 구애받지 않고 실행 가능, 가상 기억 장치 세부 사항 고려 불필요

(2) **가상 기억 장치의 단점:** 실행 속도가 느림, 소프트웨어를 통한 추가 관리가 필요

06. 입출력과 파일 관리

(1) **입출력 버퍼링:** 단일 버퍼링, 더블 버퍼링, 순환 버퍼링

(2) **파일의 구성 방식:** 순차 접근 파일, 직접 접근 파일, 인덱스된 순차 접근 파일

대단원 평가문제

01 운영체제의 목적으로 가장 거리가 먼 것은?

① 사용자 인터페이스 제공
② 주변 장치 관리
③ 데이터 압축 및 복원
④ 신뢰성 향상
⑤ 잘못된 자원 사용 감시

02 운영체제의 기능에 해당하지 않는 것은?

① 사용자와 컴퓨터 사이의 인터페이스를 제공한다.
② 효율적인 자원 관리를 한다.
③ 시스템의 오류를 처리한다.
④ 두 개 이상의 목적 프로그램을 합쳐서 실행 가능한 프로그램으로 만든다.
⑤ 컴퓨터 하드웨어에 대한 중재 역할을 한다.

03 컴퓨터의 본체 및 각 주변 장치를 사용자가 가장 능률적이고 경제적으로 사용할 수 있도록 하는 프로그램은?

① 컴파일러 ② 매크로
③ 운영체제 ④ 로더
⑤ 인터프린터

04 다음 중 운영체제가 프로그램과 사용자들에게 제공하는 기능으로 틀린 것은?

① 프로그램의 실행
② 파일 시스템의 조작
③ 자원 할당
④ 사용자에게 무제한의 액세스 제공
⑤ 파일 스케줄링 관리

05 다음 설명에 해당하는 운영체제의 기능은?

> 프로그램, 프로세스 및 사용자가 시스템이나 자원에 접근하는 것을 제어하는 매커니즘

① 입출력 관리 ② 보호 시스템
③ 프로세스 관리 ④ 사용자 인터페이스
⑤ 커널

06 운영체제의 중요한 부분인 커널(kernel)에 대한 설명중 옳은 것은?

① 컴퓨터 자원을 관리한다.
② 항상 어셈블리 언어로 작성된다.
③ 사용자가 하드웨어와 통신할 수 있도록 한다.
④ 응용 프로그래머에 의해 작성된다.
⑤ 펌웨어와 장치 드라이버의 도움을 받는다

07 프로세스의 정의와 관련이 적은 것은?

① 실행 중인 프로그램
② PCB를 가진 프로그램
③ CPU가 할당되는 실체
④ 디스크에 저장된 프로그램
⑤ 주기억 장치에 들어간 프로그램

08 프로세스 제어 블록(Process Control Block)에 대한 설명 중 옳지 않은 것은?

① 프로세서의 현 상태를 알 수 있다.
② 프로세서의 우선순위에 대한 정보를 갖고 있다.
③ 프로세스에 할당된 자원에 대한 정보를 갖고 있다.
④ 부모 프로세스와 자식 프로세스는 프로세스 제어 블록을 공유한다.
⑤ 프로세스가 생성될 때 같이 생성되고 종료될 때 같이 사라진다.

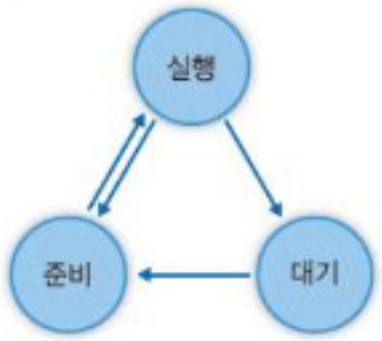
09 프로세스의 상태 전이에 있어서 실행 상태에 있던 프로세스가 할당 시간 종료로 인하여 바뀌어지는 형태는?

① 보류 상태 ② 종료 상태
③ 준비 상태 ④ 서스펜드 상태
⑤ 생성 상태

10 가장 바람직한 스케줄링 정책은?

① CPU 이용률을 줄이고 반환 시간을 늘린다.
② 응답 시간을 줄이고 CPU 이용률을 늘린다.
③ 대기 시간을 늘리고 반환 시간을 줄인다.
④ 반환 시간과 처리율을 늘린다.
⑤ 우선순위가 높고 낮은 것을 교대로 하여 자원을 할당한다.

11 다음의 그림은 프로세스의 상태도이다. 그림 중 실행의 상태에서 준비의 상태로 가게 되는 프로세스에 대한 설명으로 옳은 것은?



① 자신에게 할당된 시간 분할을 다 사용한 프로세스
② 입출력을 요청한 프로세스
③ 어떤 수를 0으로 나누는 연산을 시도하는 프로세스
④ 입출력이 끝났음을 통보받은 프로세스
⑤ 준비 큐에서 기다리는 프로세스

12 다음에서 설명하는 용어는 무엇인가?

자원이 할당되기를 오랜 시간 동안 기다린 프로세스에 대하여 기다린 시간에 비례하는 높은 우선 순위를 부여하고 가까운 시간 안에 자원이 할당되도록 하는 기법

13 다음에서 평균 대기 시간을 계산하시오.

라운드 로빈 알고리즘에서 A, B, C, D, E의 작업을 실행시킬 때 대기 시간은 다음과 같다. 평균 대기 시간은?

작업	A	B	C	D	E
대기시간	0	32	20	23	40

14 다음에서 설명하는 용어는 무엇인가?

- 각 프로세스는 한꺼번에 자신에게 필요한 모든 자원을 요구해야 하며 이 요구가 만족되지 않으면 작업을 진행하지 않는다.
- 만일 어떤 자원을 갖고 있는 프로세스가 더 이상 요구가 받아들여지지 않으면 원래 갖고 있던 자원을 반납하고 필요하면 다시 그 자원과 함께 다른 자원을 요구해야 한다.
- 모든 프로세스에 각 자원을 유형별로 할당 순서를 정한다.

15 운영체제가 해결해야 할 다음과 같은 여러 가지 문제점 발생의 직접적인 원인은?

메모리 관리, CPU와 자원의 할당 문제, 교착 상태 처리 문제, 병행성 제어 문제

16 운영체제의 목적을 나열하시오.

17 프로세스 스케줄링 방법들을 참고하여 개선된 프로세스 스케줄링 방법을 제시하시오.