

운영체제 개요 / 기능, 구조와 발전 과정

운영체제는 컴퓨터 시스템의 핵심 소프트웨어로, 하드웨어와 사용자 간의 중간 매개체 역할을 합니다. 아래에서 운영체제의 **기능, 구조, 발전 과정**, 그리고 **프로세스 관리**와 관련된 주요 개념을 쉽게 설명하겠습니다.

운영체제 개요

운영체제의 주요 기능

운영체제는 다음과 같은 기능을 수행합니다

- **자원 관리**: CPU, 메모리, 저장 장치, 입출력 장치 등 시스템 자원을 효율적으로 관리합니다.
- **사용자 인터페이스 제공**: GUI(Graphical User Interface) 또는 CLI(Command Line Interface)를 통해 사용자가 시스템을 쉽게 사용할 수 있도록 돕습니다.
- **프로세스 관리**: 여러 프로그램이 동시에 실행될 수 있도록 프로세스를 생성, 스케줄링, 종료합니다.
- **파일 시스템 관리**: 데이터를 저장하고 접근할 수 있는 구조를 제공합니다.
- **보안 및 보호**: 사용자와 시스템을 외부 위협으로부터 보호합니다

운영체제의 구조

운영체제는 크게 ****커널(Kernel)****과 ****유틸리티(Utility)****로 나뉩니다:

커널: 운영체제의 핵심 부분으로, 항상 메모리에 상주하며 하드웨어와 소프트웨어를 연결합니다.

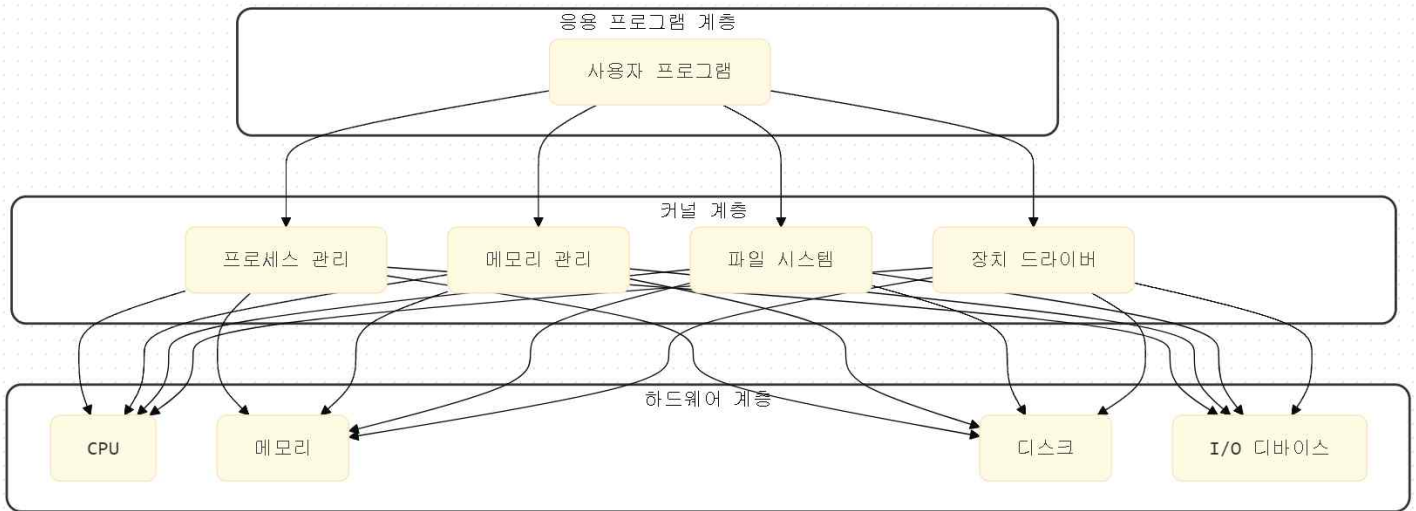
예를 들어, 메모리 관리, 프로세스 스케줄링, 입출력 제어를 담당합니다.

유틸리티: 사용자가 필요할 때만 메모리에 적재되는 프로그램으로, 파일 관리, 통신, 상태 정보 제공 등의 역할을 합니다.

읽어두기

커널의 기본 개념

- 하드웨어와 응용 프로그램 사이에서 중재자 역할을 수행
- 시스템 자원을 효율적으로 관리하고 할당
- 모든 하드웨어 장치와의 통신을 제어



위 다이어그램은 운영체제의 세 가지 주요 계층을 보여줍니다:

1. 응용 프로그램 계층

- 사용자가 직접 실행하는 모든 프로그램이 이 계층에 속함
- 웹 브라우저, 문서 편집기 등 일반적인 애플리케이션이 여기에 포함

2. 커널 계층

- 시스템의 핵심 기능을 제공하는 모듈들로 구성됨
- 각 모듈의 역할:
 - 프로세스 관리: 프로그램의 실행과 스케줄링
 - 메모리 관리: 메모리 할당과 보호
 - 파일 시스템: 데이터 저장과 접근 제어
 - 장치 드라이버: 하드웨어와의 통신 인터페이스

3. 하드웨어 계층

- 실질적인 컴퓨터 부품들(CPU, 메모리, 저장장치 등)
- 커널이 직접 제어하는 물리적 리소스

커널은 이러한 세 계층 사이에서 중개자 역할을 하며, 모든 요청은 반드시 커널을 통해서만 하드웨어에 접근할 수 있습니다.

운영체제의 발전 과정

운영체제는 다음과 같은 단계로 발전해 왔습니다:

1. 일괄 처리 시스템(Batch Processing): 1950년대, 작업을 한 번에 처리하는 방식.

다중 프로그래밍(Multiprogramming): 여러 프로그램을 메모리에 올려 동시에 실행하는 방식.

2. 시분할 시스템(Time-Sharing): CPU 시간을 나누어 여러 사용자가 동시에 작업하는 것처럼 보이게 하는 방식.

3. 실시간 시스템(Real-Time): 정해진 시간 내에 작업을 완료해야 하는 시스템, 예를 들어 로켓 제어 시스템.

4. 분산 시스템(Distributed Systems): 여러 컴퓨터가 협력하여 하나의 시스템처럼 작동하는 방식.

프로세스 관리

프로세스 상태

프로세스는 실행 중에 여러 상태를 거칩니다:

1. New(생성): 프로세스가 생성된 상태.

2. Ready(준비): CPU 할당을 기다리는 상태.

3. Running(실행): CPU를 할당받아 실행 중인 상태.

4. Blocked(대기): I/O 작업 등으로 인해 대기 중인 상태.

5. Exit(종료): 실행이 완료된 상태.

참고:

new 단계 : 프로세스가 생성되는 단계-> 프로그램을 실행시켰을 때의 첫 번째 상태가 됩니다.

준비(ready)단계 : 프로세스가 프로세서에 할당되기를 기다리고 있는 상태

실행(running)상태 : 프로세스가 실행 중인 상태

대기(waiting)상태 : 프로세스가 어떤 사건(event)을 기다리고 있는 상태

terminated상태 : 프로세스의 실행이 종료된 상태

디스패치(dispatch) : 준비(ready)상태에서 실행(running)상태로 전이되는 과정을 말하며, 이는 작업 스케줄러가 해당 프로세스를 선택하여 실행되어지는 것으로 이때 실행된 프로세스가 CPU를 점유하게 됩니다.

인터럽트(Interrupt) : 인터럽트 신호를 받게되면, 실행(running)중이던 프로세스는 준비(ready)상태로 전이되고, 우선순위(Priority)가 높은 프로세스를 실행(running)상태로 전이시키게 됩니다. (프로세스는 각각 우선순위를 부여받고, 우선순위에 따라 프로세스가 준비상태로 전이되거나, 실행상태로 전이됩니다.)

입출력 혹은 이벤트대기(I/O or event wait) : CPU를 점유하고 있는 프로세스가 입출력 처리를 해야만 하는 상황이라면, 실행되고 있는 프로세스는 실행(running)상태에서 대기/보류(waiting)상태로 바뀌게 됩니다. 그리고 대기상태로 바뀐 프로세스는 입출력 처리가 모두 끝날 때까지 대기상태로 머물게 됩니다. 그리고 실행상태이던 프로세스가 대기상태로 전이됨과 함께, 준비상태이던 또 다른 프로세스가 실행상태로 전이됩니다. 또한 대기상태인 프로세스는 우선순위가 부여되지 않으며 스케줄러에 의해 선택될 수 없습니다.

입출력 혹은 이벤트완료(I/O or event completion) : 입출력 처리가 끝난 프로세스는 대기상태(waiting)에서 준비상태(ready)로 전이되어 스케줄러에게 선택될 수 있게 됩니다. 추가로 프로세스를 종료(Terminate) 시킬 때에도 Blocked 상태를 거칠 수 있습니다.

**** 단일 프로세스 시스템에서는 현재 실행상태에 있을 수 있는 프로세스가 오직 하나! but, 많은 프로세스가 waiting or ready상태에 있을 수 있습니다.!**

출처: <https://jwprogramming.tistory.com/15> [개발자를 꿈꾸는 프로그래머:티스토리]

프로세스는 실행되면서 매 순간 상태가 변한다. 이 상태를 정의하는 이름들은 OS의 종류에 따라 다르지만, 대부분의 OS에서 이름은 달라도 비슷한 개념을 갖는다. 보통은 Five-state라 하여 프로세스의 상태를 설명하는 것 같다. (two,seven-state도 있음)

새로운(new) : 프로세스가 생성중

프로세스 생성도 아니고 '중'이 붙는 이유가 뭘까? 확실히 잘 모르겠다..
프로세스가 생성되었지만 아직 OS에 의해 승인(admit)받지 못한 상태라 한다.

준비완료(ready) : 프로세스가 처리기에 할당 되기를 기다림

new 상태에서 admit된 상태. CPU를 제외한 다른 자원이 준비완료.
보조기억장치에 있는 프로그램을 실행시켜 메모리에 로드된 상태.
이 상태에는 여러개의 프로세스가 존재 할 수 있음. -> CPU를 할당받기를 기다림

실행(running) : 명령어들이 실행중

프로세스가 CPU를 할당받아 실제로 프로세스가 수행되고 있는 상태.
단일 처리기에서 단 하나의 프로세스가 이 상태를 갖음. (2개가 running 될 수 없겠다)

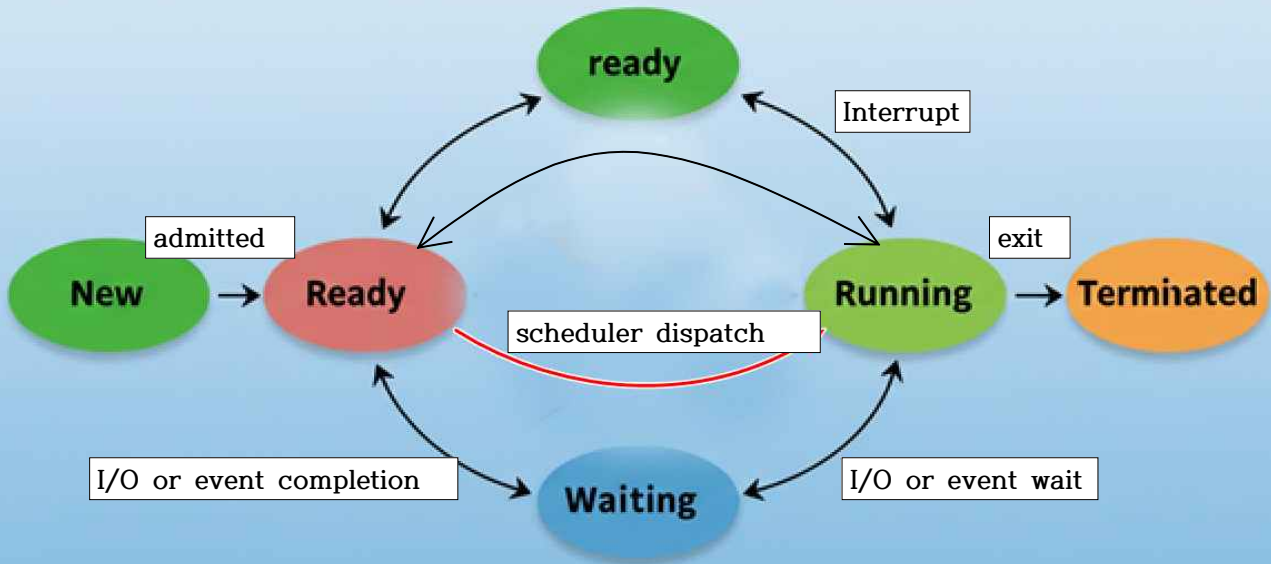
대기(waiting) : 프로세스가 어떤 사건(event)이 일어나기를 기다림 (Blocked 이라고도 함)

프로세스가 실행(running)되다가, 할당받은 CPU를 반납하고,
입/출력 작업이 완료를 기다리는 것 같은 특별한 사건을 기다림

종료(terminated) : 프로세스의 실행이 종료

프로세스의 실행이 완료되고 할당된 CPU를 반납.

Process Lifecycle



new -> ready : OS에 의해 프로세스를 승인 - **admit**

ready -> running : 처리기가 프로세스를 수행하기 위해 CPU 할당(시간도 할당됨) - **dispatch**

running -> ready : 할당된 시간이 지나면 time out interrupt가 발생 - **interrupt**

running -> waiting : time out 전에 I/O 요청이 발생(sleep,block) - **event wait**

waiting -> ready : I/O 요청이 완료되면 다시 ready 상태로 전이 - **event completion**

running -> terminated : 프로세스 종료 - **exit**

ready -> terminated : **exit**

waiting -> terminated : **exit**

웹 참조 : <https://m.blog.naver.com/4717010/60207137085>

프로세스와 스레드

- **프로세스(Process):** 프로세스는 실행 중인 프로그램의 인스턴스를 의미하며, 운영체제로부터 독립적인 메모리 공간(Code, Data, Stack, Heap)을 할당받아 실행됩니다. 각 프로세스는 독립적이며 다른 프로세스의 메모리에 직접 접근할 수 없습니다. 프로세스 간 통신은 *IPC(Inter-Process Communication)*를 통해 이루어집니다.
- **스레드(Thread):** 스레드는 프로세스 내에서 실행되는 작은 실행 단위입니다. 스레드는 프로세스의 자원을 공유하며, 독립적인 스택을 가집니다. 스레드 간 통신은 프로세스 내에서 이루어지므로 비용이 적고 효율적입니다. 스레드는 프로세스의 메모리 공간(Code, Data, Heap)을 공유하지만, 각 스레드는 고유한 레지스터와 스택을 가집니다.

문제 1 : 운영체제의 주요 기능에 해당하지 않는 것은 무엇인가요?

- ㄱ) 자원 관리
- ㄴ) 사용자와 하드웨어 간의 인터페이스 제공
- ㄷ) 데이터베이스 관리
- ㄹ) 오류 검사 및 복구

정답: ㄷ) 데이터베이스 관리

정답 이유: 운영체제는 컴퓨터 시스템의 자원을 효율적으로 관리하고, 사용자와 하드웨어 간의 매개체 역할을 하며, 오류를 검사하고 복구하는 기능을 제공합니다. 하지만 데이터베이스 관리는 운영체제의 역할이 아니라 데이터베이스 관리 시스템(DBMS)의 역할입니다.

문제 2 : 운영체제의 구조에서 "커널(Kernel)"의 역할로 적절한 것은 무엇인가요?

- ㄱ) 사용자 인터페이스 제공
- ㄴ) 하드웨어 자원 관리
- ㄷ) 응용 프로그램 실행
- ㄹ) 파일 편집

정답: ㄴ) 하드웨어 자원 관리

정답 이유: 커널은 운영체제의 핵심 부분으로, 하드웨어 자원을 관리하고 응용 프로그램이 하드웨어에 직접 접근하지 못하도록 제어합니다. 사용자 인터페이스 제공은 셸(Shell)의 역할이며, 파일 편집은 응용 프로그램의 역할입니다.

문제 3 : 운영체제의 발전 과정에서 "시분할 시스템"의 주요 특징은 무엇인가요?

- ㄱ) 한 번에 하나의 작업만 처리
- ㄴ) 여러 사용자가 동시에 시스템을 공유
- ㄷ) 실시간 데이터 처리
- ㄹ) 단일 사용자 환경

정답: ㄴ) 여러 사용자가 동시에 시스템을 공유

정답 이유: 시분할 시스템은 여러 사용자가 동시에 시스템을 사용할 수 있도록 CPU 시간을 분할하여 각 사용자에게 할당하는 방식입니다. 이는 다중 사용자 환경을 지원하며, 단일 사용자 환경이나 실시간 처리와는 구분됩니다.

문제 4 : 프로세스 관리에서 "준비 상태(Ready State)"에 있는 프로세스는 어떤 상태인가요?

- ㄱ) 프로세서에서 실행 중
- ㄴ) 특정 이벤트를 기다리는 중
- ㄷ) 프로세서에서 실행될 준비가 된 상태
- ㄹ) 종료된 상태

정답: ㄷ) 프로세서에서 실행될 준비가 된 상태

정답 이유: 준비 상태는 프로세스가 실행되기 위해 프로세서 할당을 기다리는 상태를 의미합니다. 실행 중인 상태는 "Running", 특정 이벤트를 기다리는 상태는 "Blocked" 상태로 구분됩니다.

문제 5 : 운영체제의 주요 목적에 해당하지 않는 것은 무엇인가요?

- ㄱ) 처리 능력 향상

- ㄴ) 반환 시간 단축
- ㄷ) 하드웨어 제조
- ㄹ) 시스템 신뢰성 향상

정답: ㄷ) 하드웨어 제조

정답 이유: 운영체제의 주요 목적은 처리 능력(Throughput) 향상, 반환 시간(Turnaround Time) 단축, 시스템 신뢰성(Reliability) 향상 등입니다. 하지만 하드웨어 제조는 운영체제의 목적과 관련이 없습니다.

문제 1: 벡터 프로세서의 특징으로 옳지 않은 것은 무엇인가?

- ㄱ) 벡터 명령은 여러 독립된 작업을 동시에 처리할 수 있다.
- ㄴ) 벡터 프로세서는 데이터 의존성을 검사할 필요가 없다.
- ㄷ) 벡터 프로세서는 스칼라 프로세서보다 일반적으로 느리다.
- ㄹ) 벡터 프로세서는 병렬 처리를 통해 계산 속도를 향상시킨다.

정답: ㄷ) 벡터 프로세서는 스칼라 프로세서보다 일반적으로 느리다.

정답 이유: 벡터 프로세서는 병렬 처리를 통해 대량의 데이터를 동시에 처리할 수 있어 스칼라 프로세서보다 빠른 계산 속도를 제공합니다. 따라서 "느리다"는 설명은 틀렸습니다.

문제 2: 멀티코어 시스템의 장점으로 가장 적절한 것은?

- ㄱ) 단일 코어에서 클럭 속도를 높이는 방식으로 성능을 향상시킨다.
- ㄴ) 여러 코어가 병렬로 작업을 처리하여 처리량을 증가시킨다.

ㄷ) 코어 간 데이터 공유가 불가능하다.

ㄹ) 전력 소모가 단일 코어보다 항상 적다.

정답: ㄴ) 여러 코어가 병렬로 작업을 처리하여 처리량을 증가시킨다.

정답 이유: 멀티코어 시스템은 여러 코어가 병렬로 작업을 나누어 처리함으로써 처리량을 증가시키고, 동시에 성능을 향상시킵니다. 이는 단일 코어의 클럭 속도를 높이는 방식보다 효율적입니다.

문제 3: 병렬 처리 시스템의 특징으로 옳은 것은?

ㄱ) 단일 프로세서에서만 작업을 처리한다.

ㄴ) 여러 프로세서가 동시에 작업을 처리하여 성능을 향상시킨다.

ㄷ) 병렬 처리 시스템은 항상 단일 메모리만 사용한다.

ㄹ) 병렬 처리는 데이터 의존성이 높은 작업에 적합하다.

정답: ㄴ) 여러 프로세서가 동시에 작업을 처리하여 성능을 향상시킨다.

정답 이유: 병렬 처리 시스템은 다수의 프로세서를 활용하여 작업을 동시에 처리함으로써 성능을 크게 향상시킵니다. 이는 특히 대규모 데이터 처리나 계산 집약적인 작업에서 유리합니다.

문제 4: 버스 구조의 주요 구성 요소가 아닌 것은?

ㄱ) 데이터 버스

ㄴ) 주소 버스

ㄷ) 제어 버스

ㄹ) 캐시 메모리

정답: ㄹ) 캐시 메모리

정답 이유: 데이터 버스, 주소 버스, 제어 버스는 시스템 내에서 데이터를 전송하고 제어하는 데 필수적인 버스 구조의 구성 요소입니다. 반면, 캐시 메모리는 CPU와 메인 메모리 간의 속도 차이를 줄이기 위한 저장 장치로, 버스 구조의 구성 요소가 아닙니다.

문제 5: 주기억장치(RAM)의 특징으로 옳지 않은 것은?

- ㄱ) 데이터를 읽고 쓸 수 있다.
- ㄴ) 전원이 꺼지면 데이터가 유지된다.
- ㄷ) CPU가 직접 접근할 수 있다.
- ㄹ) DRAM과 SRAM으로 나뉜다.

정답: ㄴ) 전원이 꺼지면 데이터가 유지된다.

정답 이유: RAM은 휘발성 메모리로, 전원이 꺼지면 저장된 데이터가 사라집니다. 이는 ROM과의 주요 차이점입니다.

문제 6: 보조기억장치의 특징으로 옳은 것은?

- ㄱ) 전원이 꺼지면 데이터가 사라진다.
- ㄴ) 주기억장치보다 속도가 빠르다.
- ㄷ) 데이터를 영구적으로 저장할 수 있다.
- ㄹ) CPU가 직접 접근하여 데이터를 처리한다.

정답: ㄷ) 데이터를 영구적으로 저장할 수 있다.

정답 이유: 보조기억장치는 데이터를 영구적으로 저장하며, 전원이 꺼져도 데이터가 유지됩니다.
하지만 주기억장치보다 속도가 느리고, CPU가 직접 접근하지 못합니다.

문제 7: 캐시 메모리의 역할로 가장 적절한 것은?

- ㄱ) 데이터를 영구적으로 저장한다.
- ㄴ) CPU와 주기억장치 간의 속도 차이를 줄인다.
- ㄷ) 보조기억장치의 데이터를 직접 처리한다.
- ㄹ) 데이터를 순차적으로만 접근할 수 있다.

정답: ㄴ) CPU와 주기억장치 간의 속도 차이를 줄인다.

정답 이유: 캐시 메모리는 CPU와 주기억장치 사이에 위치하여 자주 사용되는 데이터를 저장함으로써 데이터 접근 속도를 높이고, CPU와 메모리 간의 속도 차이를 줄이는 역할을 합니다.

문제 8: CPU 내부 구성 요소에 포함되지 않는 것은?

- ㄱ) 산술논리연산장치(ALU)
- ㄴ) 제어장치(Control Unit)
- ㄷ) 레지스터(Register)
- ㄹ) 하드디스크(HDD)

정답: ㄹ) 하드디스크(HDD)

정답 이유: CPU는 ALU, 제어장치, 레지스터와 같은 구성 요소로 이루어져 있으며, 하드디스크는

보조기억장치로 CPU 내부 구성 요소가 아닙니다.

문제 9: DRAM과 SRAM의 차이로 옳은 것은?

- ㄱ) DRAM은 비휘발성 메모리이고, SRAM은 휘발성 메모리이다.
- ㄴ) DRAM은 속도가 빠르고, SRAM은 속도가 느리다.
- ㄷ) DRAM은 재충전이 필요하고, SRAM은 재충전이 필요 없다.
- ㄹ) DRAM은 캐시 메모리로 사용되고, SRAM은 주기억장치로 사용된다.

정답: ㄷ) DRAM은 재충전이 필요하고, SRAM은 재충전이 필요 없다.

정답 이유: DRAM은 데이터를 유지하기 위해 주기적으로 재충전이 필요하며, SRAM은 전원이 공급되는 동안 데이터를 유지할 수 있어 재충전이 필요 없습니다. SRAM은 주로 캐시 메모리로 사용됩니다.

문제 10: 보조기억장치의 예로 적절하지 않은 것은?

- ㄱ) 하드디스크(HDD)
- ㄴ) 솔리드 스테이트 드라이브(SSD)
- ㄷ) 광학 디스크(CD/DVD)
- ㄹ) 레지스터

정답: ㄹ) 레지스터

정답 이유: 레지스터는 CPU 내부에 위치한 고속 메모리로, 보조기억장치가 아닙니다. 보조기억장치에는 HDD, SSD, CD/DVD 등이 포함됩니다.

문제 11: 메모리 계층 구조에서 속도가 가장 빠른 계층은?

- ㄱ) 보조기억장치
- ㄴ) 주기억장치
- ㄷ) 캐시 메모리
- ㄹ) 레지스터

정답: ㄹ) 레지스터

정답 이유: 메모리 계층 구조에서 레지스터는 CPU 내부에 위치하며, 가장 빠른 속도를 제공합니다. 캐시 메모리와 주기억장치는 레지스터보다 느리고, 보조기억장치는 가장 느립니다.

Quize

문제 12 : 운영체제의 주된 목적은 무엇인가요?

- ㄱ) 사용자 인터페이스 제공
- ㄴ) 하드웨어 자원 관리
- ㄷ) 데이터베이스 관리
- ㄹ) 네트워크 보안

정답: ㄴ) 하드웨어 자원 관리

정답 이유: 운영체제는 컴퓨터의 하드웨어 자원을 효율적으로 관리하여 사용자와 응용 프로그램이 하드웨어와 상호작용할 수 있도록 돕는 소프트웨어입니다. 이는 CPU, 메모리, 저장 장치 등 다양한 자원을 관리하고, 여러 프로그램이 동시에 실행될 수 있도록 자원을 배분하는 역할을 포함합니다.

문제 13: 운영체제가 제공하는 기능 중 어떤 것이 아닌가요?

- ㄱ) 프로세스 관리
- ㄴ) 메모리 관리
- ㄷ) 웹 브라우징
- ㄹ) 파일 시스템 관리

정답: ㄷ) 웹 브라우징

정답 이유: 운영체제는 프로세스 관리, 메모리 관리, 파일 시스템 관리와 같은 기본적인 시스템 자원 관리 기능을 제공합니다. 반면, 웹 브라우징은 운영체제의 기능이 아니라 웹 브라우저와 같은 응용 프로그램의 기능입니다. 운영체제는 이러한 응용 프로그램이 원활하게 실행될 수 있도록 자원을 할당하지만, 직접적으로 웹 브라우징 기능을 제공하지는 않습니다.

문제 14 : 운영체제가 사용자에게 제공하는 주요 이점은 무엇인가요?

- ㄱ) 하드웨어의 직접 조작
- ㄴ) 프로그램 간의 충돌 방지
- ㄷ) 데이터 암호화
- ㄹ) 인터넷 연결

정답: ㄴ) 프로그램 간의 충돌 방지

정답 이유: 운영체제는 여러 프로그램이 동시에 실행될 때 자원을 효율적으로 관리하여 프로그램 간의 충돌을 방지합니다. 이는 다중 프로그래밍 환경에서 각 프로그램이 필요한 자원을 적절히 할당받고, 서로 간섭하지 않도록 조정하는 기능을 통해 이루어집니다. 따라서 운영체제는 사용자에게 안정적인 컴퓨터 환경을 제공합니다.

문제 1 : 프로세스가 생성(New) 상태에서 준비(Ready) 상태로 전환되는 이유는 무엇인가?

- ㄱ) 프로세스가 CPU를 할당받았기 때문이다.
- ㄴ) 프로세스가 필요한 자원을 모두 할당받았기 때문이다.
- ㄷ) 프로세스가 I/O 작업을 완료했기 때문이다.
- ㄹ) 프로세스가 종료(Terminated) 상태로 전환되었기 때문이다.

정답: ㄴ) 프로세스가 필요한 자원을 모두 할당받았기 때문이다.

정답 이유: 생성 상태는 프로세스가 운영체제에 의해 생성된 초기 상태로, 필요한 자원이 모두 할당되면 준비 상태로 전환된다. 준비 상태는 CPU에서 실행될 준비가 완료된 상태를 의미한다.

문제 2 : 프로세스가 실행(Running) 상태에서 대기(Blocked) 상태로 전환되는 이유는 무엇인가?

- ㄱ) 프로세스가 I/O 작업을 요청했기 때문이다.
- ㄴ) 프로세스가 CPU 시간을 초과했기 때문이다.
- ㄷ) 프로세스가 종료되었기 때문이다.
- ㄹ) 프로세스가 메모리 부족으로 인해 중단되었기 때문이다.

정답: ㄱ) 프로세스가 I/O 작업을 요청했기 때문이다.

정답 이유: 실행 상태의 프로세스는 I/O 작업이나 특정 이벤트를 기다려야 할 때 CPU를 반납하고 대기 상태로 전환된다. 이 상태에서 프로세스는 요청한 작업이 완료되기를 기다린다.

문제 3 : 프로세스가 준비(Ready) 상태에서 실행(Running) 상태로 전환되는 조건은 무엇인가?

- ㄱ) 프로세스가 I/O 작업을 완료했기 때문이다.
- ㄴ) 운영체제의 스케줄러가 CPU를 할당했기 때문이다.
- ㄷ) 프로세스가 종료되었기 때문이다.
- ㄹ) 프로세스가 메모리에서 제거되었기 때문이다.

정답:ㄴ) 운영체제의 스케줄러가 CPU를 할당했기 때문이다.

정답 이유:준비 상태의 프로세스는 CPU를 할당받아야 실행 상태로 전환될 수 있다. 스케줄러는 우선순위와 정책에 따라 CPU를 할당한다.

문제 4 : 프로세스가 대기(Blocked) 상태에서 준비(Ready) 상태로 전환되는 이유는 무엇인가?

- ㄱ) 프로세스가 CPU를 할당받았기 때문이다.
- ㄴ) 프로세스가 요청한 I/O 작업이 완료되었기 때문이다.
- ㄷ) 프로세스가 종료되었기 때문이다.
- ㄹ) 프로세스가 메모리 부족으로 인해 중단되었기 때문이다.

정답:ㄴ) 프로세스가 요청한 I/O 작업이 완료되었기 때문이다.

정답 이유:대기 상태의 프로세스는 I/O 작업이나 이벤트가 완료되면 준비 상태로 전환되어 CPU 할당을 기다린다.

문제 5 : 프로세스가 실행(Running) 상태에서 준비(Ready) 상태로 전환되는 이유는 무엇인가?

- ㄱ) 프로세스가 I/O 작업을 요청했기 때문이다.
- ㄴ) 프로세스가 종료되었기 때문이다.
- ㄷ) 프로세스가 CPU 시간을 초과했기 때문이다.
- ㄹ) 프로세스가 메모리 부족으로 인해 중단되었기 때문이다.

정답:ㄷ) 프로세스가 CPU 시간을 초과했기 때문이다.

정답 이유:실행 상태의 프로세스는 할당된 CPU 시간을 모두 사용하면 준비 상태로 전환되어 다시 CPU 할당을 기다린다. 이는 선점형 스케줄링에서 자주 발생한다.

문제 6 : 프로세스가 종료(Terminated) 상태로 전환되는 이유는 무엇인가?

- ㄱ) 프로세스가 작업을 완료했기 때문이다.
- ㄴ) 프로세스가 I/O 작업을 요청했기 때문이다.
- ㄷ) 프로세스가 CPU 시간을 초과했기 때문이다.
- ㄹ) 프로세스가 메모리 부족으로 인해 중단되었기 때문이다.

정답:ㄱ) 프로세스가 작업을 완료했기 때문이다.

정답 이유:종료 상태는 프로세스가 작업을 완료하거나 오류로 인해 실행을 중단했을 때 발생한다. 이 상태에서 프로세스는 시스템 자원을 해제하고 제거된다.

문제 7 : 프로세스가 준비(Ready) 상태에서 준비/보류(Ready/Suspend) 상태로 전환되는 이유는 무엇인가?

- ㄱ) 프로세스가 메모리 부족으로 인해 보조 저장장치로 이동했기 때문이다.
- ㄴ) 프로세스가 CPU를 할당받았기 때문이다.

ㄷ) 프로세스가 I/O 작업을 요청했기 때문이다.

ㄹ) 프로세스가 작업을 완료했기 때문이다.

정답: ㄱ) 프로세스가 메모리 부족으로 인해 보조 저장장치로 이동했기 때문이다.

정답 이유: 준비 상태의 프로세스는 메모리 부족 시 보조 저장장치로 이동하여 준비/보류 상태로 전환된다. 이는 메모리 관리의 일환이다.

문제 8 : 프로세스가 실행(Running) 상태에서 종료(Terminated) 상태로 전환되는 이유는 무엇인가?

ㄱ) 프로세스가 작업을 완료했기 때문이다.

ㄴ) 프로세스가 I/O 작업을 요청했기 때문이다.

ㄷ) 프로세스가 CPU 시간을 초과했기 때문이다.

ㄹ) 프로세스가 메모리 부족으로 인해 중단되었기 때문이다.

정답: ㄱ) 프로세스가 작업을 완료했기 때문이다.

정답 이유: 실행 상태의 프로세스는 작업을 성공적으로 완료하거나 오류가 발생하면 종료 상태로 전환된다. 이 상태에서 시스템 자원이 해제된다.

문제 9 : 프로세스가 대기(Blocked) 상태에서 대기/보류(Blocked/Suspend) 상태로 전환되는 이유는 무엇인가?

ㄱ) 프로세스가 메모리 부족으로 인해 보조 저장장치로 이동했기 때문이다.

ㄴ) 프로세스가 I/O 작업을 완료했기 때문이다.

ㄷ) 프로세스가 CPU를 할당받았기 때문이다.

ㄹ) 프로세스가 작업을 완료했기 때문이다.

정답: ㄱ) 프로세스가 메모리 부족으로 인해 보조 저장장치로 이동했기 때문이다.

정답 이유: 대기 상태의 프로세스는 메모리 부족 시 보조 저장장치로 이동하여 대기/보류 상태로 전환된다. 이는 메모리 관리의 일환이다.

문제 10 : 프로세스가 준비/보류(Ready/Suspend) 상태에서 준비(Ready) 상태로 전환되는 이유는 무엇인가?

ㄱ) 프로세스가 메모리에 다시 로드되었기 때문이다.

ㄴ) 프로세스가 CPU를 할당받았기 때문이다.

ㄷ) 프로세스가 작업을 완료했기 때문이다.

ㄹ) 프로세스가 I/O 작업을 요청했기 때문이다.

정답: ㄱ) 프로세스가 메모리에 다시 로드되었기 때문이다.

정답 이유: 준비/보류 상태의 프로세스는 메모리가 확보되면 다시 메모리에 로드되어 준비 상태로 전환된다. 이는 메모리 관리의 일환이다.