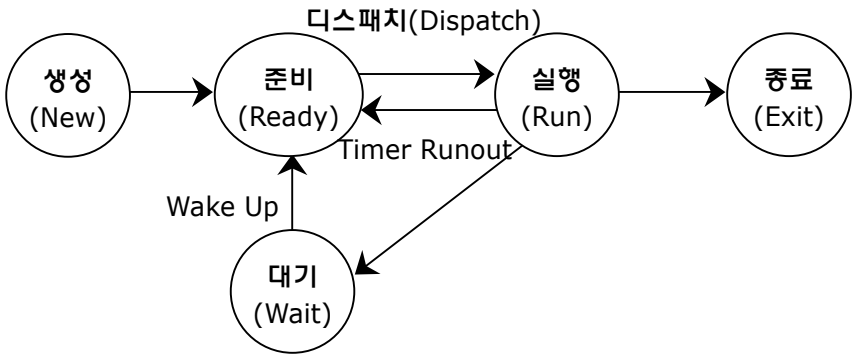
MEMO

1. 프로세스(Process)의 개념

- 실행 중인 프로그램(작업)
- Procedure가 활동 중인 것
- 운영체제(OS)가 관리하는 실행의 단위

2. 프로세스의 상태(Process State)



생성(New) **상태**

- 프로세스가 생성되었지만 아직 운영체제에 의해 실행가능하게 되지 못 한 상태

준비(Ready) **상태**

- 프로세스가 실행을 위해 CPU(Processor)를 할당받기를 기다리는 상태

실행(Run) **상태**

- 프로세스가 CPU(Processor)를 할당받아 실제로 실행되는 상태

디스패치(Dispatch) ?
Ready 상태의 프로세스가 CPU를 할당받아 Run 상태로 전이되는 과정

Timer Runout ?
Run 상태의 프로세스에 할당된 CPU 사용시간이 완료되어 Ready 상태로 되돌아가는 과정

종료(Exit) **상태**

- 프로세스의 실행이 완전히 끝나고 CPU 할당이 해제된 상태

MEMO

- **대기(Wait) 상태**
  - 프로세스가 CPU를 할당받아 실행되다가 어떤 사건이 발생(예. 입 · 출력 작업이 완료)할 때까지 멈추어 있는 상태

  Wake Up ?
  대기(Wait) 상태에 있던 프로세스가 기다리던 어떤 사건의 발생(예. 입 · 출력 작업의 완료)으로 인해 나머지부분의 실행을 위해 준비(Ready) 상태로 전이되는 과정

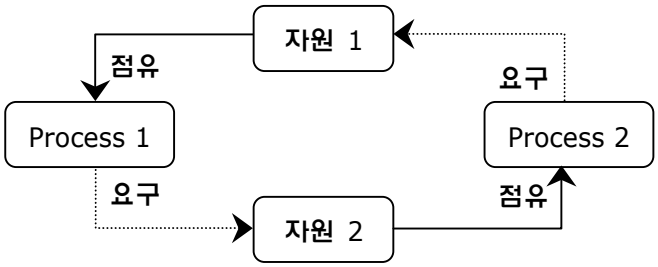
3. 프로세스 제어 블록(PCB)

- PCB(Process Control Block)란?
  - 프로세스가 실행될 때마다 프로세스의 정보를 기록해 두는 특별한 자료구조
  - 운영체제가 다른 프로세스에 제어를 넘겨줄 때 현재 실행중인 프로세스의 정보를 해당 PCB에 저장한 후 제어를 넘겨 줌.

- PCB의 주요 내용
  - 프로세스의 현재 상태
  - 프로세스의 고유 식별자
  - 프로세스의 우선순위
  - 프로세스가 적재된 주기억장치 주소
  - 프로세스에 할당된 자원에 대한 정보

4. 교착상태(Deadlock)

- **교착상태의 의미**
  - 다중 프로그래밍 환경 하에서 어떤 프로세스가 결코 발생되지 않을 사건을 무한정 기다리는 상태



  - 둘 이상의 프로세스가 서로 다른 프로세스가 차지하고 있는 자원을 요구하고 요청한 자원이 영원히 할당되지 않아 결국 해당 프로세스는 무한정으로 기다리는 교착상태에 빠지게 됨.

MEMO

교착상태 발생을 위한 필요충분조건 4가지

- 상호배제(Mutual Exclusion)
  - 각 프로세스들이 필요 자원에 대해 배타적 통제권을 요구하는 상태
- 점유 및 대기(Hold & wait)
  - 각 프로세스들이 현재의 자원을 점유하면서 다른 자원을 요구하는 상태
- 비선점(Non-Preemption)
  - 각 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 해제할 수 없는 상태
- 환형 대기(circular Wait)
  - 서로 다른 프로세스 간의 자원요구가 연속적으로 반복되는 상태

교착상태 해결 방안

- 예방(Prevention)
  - 교착상태의 발생조건을 제거하여 사전에 미리 예방하는 방법
- 회피(Avoidance)
  - 교착상태의 발생조건을 제거하지 않고, 다만 적절하게 피해 나가는 방법
- 발견(Detection)
  - 교착상태의 발생을 허용하고, 발생시 원인을 찾아 해결하는 방법
- 회복(Recovery)
  - 교착상태에 빠진 프로세스를 재시작하거나 원래 상태로 되돌림으로써 교착상태를 해결하는 방법

5. 스케줄링(Scheduling) 기법

스케줄링의 의미

- Process에 Processor(CPU)를 할당하는 것
- CPU를 포함한 모든 컴퓨터 자원은 사용되기 전에 스케줄링되어야 함.

MEMO

스케줄링의 목적

- 프로세스의 공정성 : 모든 프로세스는 공정하게 취급되어 무한정 기다리는 경우가 없어야 함.
- 처리량의 최대화 : 단위 시간 내에 최대한 많은 프로세스에 서비스를 제공할 수 있어야 함.
- 반응시간의 최소화 : 사용자에게 응답을 최소의 시간에 할 수 있어야 함.
- 수행시간의 예측가능성 : 시스템 부하에 관계없이 작업은 예측된 시간 내에, 예측된 비용으로 수행되어야 함.
- 시스템의 과부하 방지 : 시스템에 부하가 많이 걸릴 경우 스케줄링 기법으로 새로운 프로세스의 생성을 자제하거나 각 프로세스의 서비스를 줄여 부하를 방지할 수 있음.
- 균형있는 자원 활용 : 시스템 내의 유휴자원이 없도록 균형있게 자원을 할당해야 함.
- 프로세스 수행의 무한정 연기 방지 : 자원을 오래 기다리는 프로세스는 우선순위를 높여서 자원할당 기회를 높여줌.

스케줄링 기법의 종류

선점형(Preemption) 스케줄링

- 한 프로세스가 CPU(자원)를 점유하고 있을 때 다른 프로세스가 CPU(자원)를 빼앗을 수 있는 방식
- R-R, SRT 등

비선점형(Non-Preemption) 스케줄링

- 일단 한 프로세스에게 CPU(자원)가 할당되면 작업이 완료되기 전까지는 CPU(자원)를 다른 프로세스에 할당할 수 없는 방식
- FIFO, SJF, 우선순위, 기한부 등

- FIFO(First In First Out)
  - 가장 간단한 스케줄링 기법으로 CPU를 요구하는 순서에 따라 CPU를 할당하는 방식
  - 자원 요청 프로세스들의 줄(대기행렬)에 들어온 순서대로 CPU를 할당(=FCFS : First Come First Served)
- 우선순위(Priority) Scheduling
  - 각 프로세스에 우선순위를 부여하여 우선순위가 높은 프로세스에 CPU를 할당하는 방식

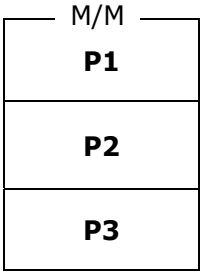
- SJF(Shortest Job First)
  - **예상 작업시간이 가장 짧은 프로세스에 먼저 CPU를 할당하는 방식**
- SRT(Shortest Remaining Time)
  - **남은 작업시간 추정치가 가장 적은 프로세스에 CPU를 할당하는 방식**
  - **선점형 SJF라 할 수 있음.**
- R-R(Round Robin)
  - **FIFO처럼 먼저 들어온 프로세스가 먼저 실행되나 각 프로세스는 정해진 시간동안만 CPU를 사용하는 방식**
  - **선점형 FIFO라 할 수 있음.**
  - **시분할 처리(TSS)에 가장 적합한 방식**
- **기한부**(Deadline) Scheduling
  - **주어진 제한된 시간 내에 프로세스가 반드시 완료되도록 스케줄링하는 방식**

MEMO

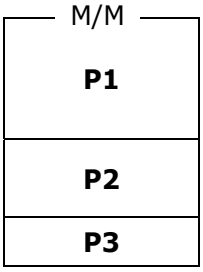
6. 주기억장치 관리

주기억장치 공간 분할 방식

- **고정 분할**(Fixed Partition) **방식**
  - **주기억장치 공간을 동일한 크기의 고정된 부분으로 미리 나누어 할당하는 방식**
  - **정적**(Static) **분할 방식이라고도 함.**



- **가변 분할**(Variable Partition) **방식**
  - **각 부분의 크기가 고정되어 있는 것이 아니라, 프로세스를 처리하는 과정에서 필요한 만큼의 공간을 할당하는 방식**
  - **동적**(Dynamic) **분할 방식이라고도 함.**



MEMO

주기억장치 공간 할당 기법

- **최적 적합**(Best Fit)
  - **적재가능한 공간 중 가장 작은 분할공간에 적재하는 방식**
- **최초 적합**(First Fit)
  - **적재가능한 공간 중 가장 먼저 발견된 분할공간에 적재하는 방식**
- **최악 적합**(Worst Fit)
  - **적재가능한 공간 중 가장 큰 분할공간에 적재하는 방식**

7. 가상기억장치 관리

가상기억장치(Virtual Memory)란 ?

> **주기억장치의 공간 확대를 위해 보조기억장치(예. 하드디스크)의 일부를 주기억장치처럼 사용하는 방식**

가상기억장치 구현 방식

- **페이징**(Paging) **기법**
  - **프로그램을 동일한 크기의 물리적 단위**(Page)**로 나누어, 이 페이지**(Page) **단위로 가상기억장치를 구현하는 기법**
- **세그먼테이션**(Segmentation) **기법**
  - **프로그램을 가변적인 크기의 논리적 단위**(Segment)**로 나누어, 세그먼트**(Segment) **단위로 가상기억장치를 구현하는 방식**

가상기억장치의 페이지 교체 기법

- **최적화 기법**
  - **이후에 가장 사용되지 않을** Page**를 교체하는 기법**
  - **이론적으로는 가장 최적의 방식이나 현실적으로 구현이 불가능함.**
- FIFO(First In First Out) **기법**
  - **주기억장치 적재순서에서 제일 처음 적재되었던** Page**를 교체하는 기법**
- LRU(Least Recently Used) **기법**
  - **가장 오랫동안 사용되지 않은** Page**를 교체하는 기법**

- LFU(Least Frequently Used) **기법**
  - **사용횟수가 가장 적은** Page**를 교체하는 기법**
- NUR(Not Used Recently) **기법**
  - **최근에 전혀 사용되지 않은** Page**를 교체하는 기법**

MEMO