

이화여자대학교

YONSEI UNIVERSITY

프로세스의 개념

Process is a program in execution

프로세스의 문맥(context)

CPU 수행 상태를 나타내는 것들까지 포함한다.

이드웨어 문맥

Program Counter

각종 register

프로세스의 주소 공간

code, data, stack

프로세스 관련 커널 자료 구조

PCB (Process Control Block)

Kernel stack

변역 코드 문제

어떤 내용을 어디까지 읽었나

stack

data

code

PC

ALU

R1

R2

Rn

프로세스 A의 주소공간

stack

data

code

AI의 커널 스택

AI의 PCB

커널 주소공간

0:00:25

-59:34

1x

10...

· 프로세스란, 실행 중인 프로그램이다.

↳ 이에 따라, 실행 중 어떤 상태냐에 따라 정의함. 그래서 어떤 상황인지 문맥 파악을 위해,

- 하드웨어 문맥
- 대포리 문맥
- 커널 문맥

파악함.

· 커널코드: 사용자 프로그램이 공용함.

그래서, 사용자 프로그램이 stack과 data가 달리, 커널의 stack과 data면 누려 커널 스택인지, PCB인지 표시함.

프로세스 문맥 설명이슈.

· 파일 시스템, 메모리 시스템 등이 의해 다른 프로그램 사용자에게 포대를 프로그램이 CPU 할당.

그래서 백업하지 않으면(문맥을 파악 못하면) 처음부터 재실행해야 할 수 있다!

이 수업에선 파종코어는 논하지 않는다.

이화여자대학교

YONSEI UNIVERSITY

프로세스의 상태 (Process State)

프로세스는 상태(state)가 변경되며 수행된다

Running

CPU를 잡고 instruction을 수행중인 상태

Ready

CPU를 기다리는 상태(메모리 등 다른 조건을 모두 만족하고)

Blocked (wait, sleep)

CPU를 주어도 당장 instruction을 수행할 수 없는 상태

Process 자신이 요청한 event(예: I/O)가 즉시 만족되지 않아 이를 기다리는 상태

(예) 디스크에서 file을 읽어와야 하는 경우

New: 프로세스가 생성 중인 상태

Terminated: 수행(execution)이 끝난 상태

경우에 따라 이 두가지 정리할 것이 많아짐을 위해

상태도 고려.

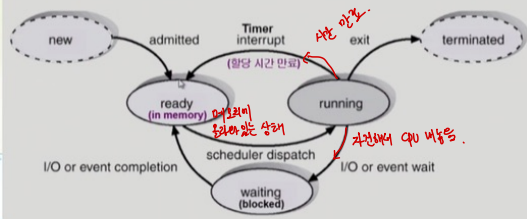
0:10:54

-49:06

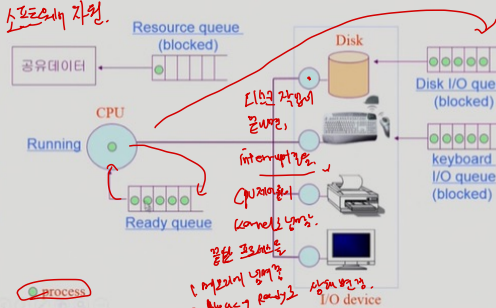
1x

1080p

### 프로세스 상태도

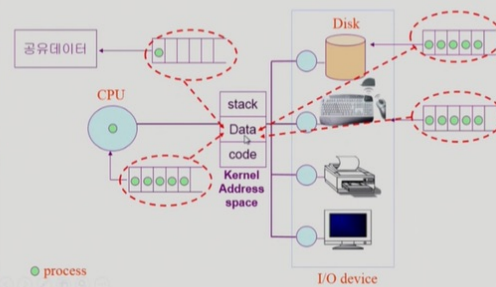


### 프로세스의 상태



오래 기다리는 작업이 생기면 해당 resource queue에서 기다리게 됨.

### 프로세스의 상태



결국 queue는 운영체제 data에서 운영되는 것임.

## Process Control Block (PCB)

▶ PCB

- ✓ 운영체제가 각 프로세스를 관리하기 위해 프로세스당 유지하는 정보
- ✓ 다음의 구성 요소를 가진다 (구조체로 유지)
  - (1) OS가 관리상 사용하는 정보
    - Process state, Process ID
    - scheduling information, priority
  - (2) CPU 수행 관련 하드웨어 값
    - Program counter, registers
  - (3) 메모리 관련
    - Code, data, stack의 위치 정보
  - (4) 파일 관련
    - Open file descriptors...

4)

①. Process state: ready, run, blocked  $\frac{\pi}{0}$ .

② process ID : 프로그램마다 숫자씩씩 씌어.

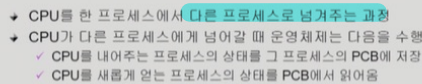
② schedu... : 우선순위 기법.

(2) 문맥을 고려.

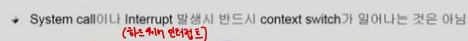
(3) 메르스 관련 문맥.

14) 파일 관련 문맥(resource 관련)

### 문맥 교환 (Context Switch)



### 문맥 교환 (Context Switch)



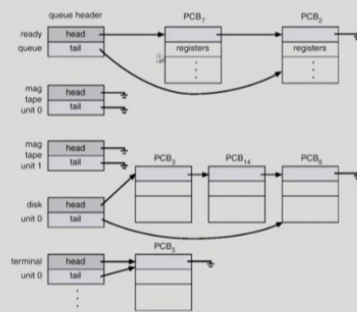
CPU는 다룬 프로그램!

A-키로 갱신시 cache memory flush 모두  
사건 처리해야 함.

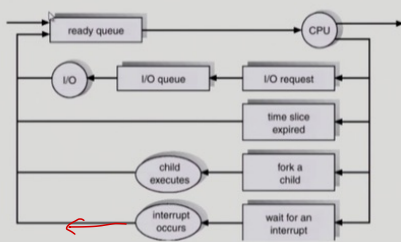
## 프로세스를 스케줄링하기 위한 큐

- **Job queue**
  - ✓ 현재 시스템 내에 있는 모든 프로세스의 집합
- **Ready queue**
  - ✓ 현재 메모리 내에 있으면서 CPU를 잡아서 실행되기를 기다리는 프로세스의 집합
- **Device queues**
  - ✓ I/O device의 처리를 기다리는 프로세스의 집합
- 프로세스들은 각 큐들을 오가며 수행된다

## Ready Queue와 다양한 Device Queue



## 프로세스 스케줄링 큐의 모습



중요! 중재는 Ready 상태가 넘어가는 것이야



이화여자대학교  
EWHA WOMANS UNIVERSITY

## 스케줄러 (Scheduler)

- **Long-term scheduler** (장기 스케줄러 or job scheduler)
  - ✓ 시작 프로세스 중 어떤 것들을 ready queue로 보낼지 결정
  - ✓ 프로세스에 **memory(및 각종 자원)**을 주는 문제
  - ✓ **degree of Multiprogramming**을 제어 (**메모리 용량에 맞는 프로세스 수 결정**)
  - ✓ time sharing system에는 보통 장기 스케줄러가 없음 (무조건 ready)
- **Short-term scheduler** (단기 스케줄러 or CPU scheduler) → **CPU 스케줄러**
  - ✓ 어떤 프로세스를 다음번에 running시킬지 결정
  - ✓ 프로세스에 CPU를 주는 문제
  - ✓ 충분히 빨리야 함 (**millisecond 단위**)
- **Medium-Term Scheduler** (중기 스케줄러 or Swapper) → **메모리 스케줄러**
  - ✓ 여유 공간 마련을 위해 프로세스를 통째로 메모리에서 디스크로 쫓아냄
  - ✓ 프로세스에게서 **memory**를 뺏는 문제
  - ✓ **degree of Multiprogramming**을 제어

0:41:50 -18:09 1x 1080p

Long-term scheduler.

new → ready 상태로 넣어줄 때 메모리 용량은 것을 리락.

이 메모리 줄지 안줄지 결정하는 것이 이 스케줄러.

모든 준비 ready 상태로 넣음. Medium-Term Scheduler도 있음.

이화여자대학교  
EWHA WOMANS UNIVERSITY

## 프로세스의 상태 (Process State)

- 프로세스의 상태
  - ✓ **Running**
    - CPU를 잡고 instruction을 수행중인 상태
  - ✓ **Ready**
    - CPU를 기다리는 상태(메모리 등 다른 조건을 모두 만족하고)
  - ✓ **Blocked (wait, sleep)**
    - I/O 등의 event를 (스스로) 기다리는 상태
    - (예) 디스크에서 file을 읽어와야 하는 경우
  - ✓ **Suspended (stopped)**
    - **외부적인 이유로** 프로세스의 수행이 정지된 상태
    - 프로세스는 통째로 디스크에 swap out 된다
    - (예) **사용자**가 프로그램을 일시 정지시킨 경우 (break key)
    - 시스템이 여러 이유로 프로세스를 잠시 중단시킴 (메모리에 너무 많은 프로세스가 올라와 있을 때)

★ Blocked: 자신이 요청한 event가 만족되면 Ready  
★ Suspended: 외부에서 resume해 주어야 Active  
사전에 계획된 경우

0:52:09 -7:50 1x 1080p

Medium-Term Scheduler 때문에 새로운 프로세스 정지.

Suspended.

이화여자대학교  
EWHA WOMANS UNIVERSITY

## 프로세스 상태도

```

graph TD
    RunningUser[Running (user mode)] -- "System call, Interrupt, trap" --> RunningMonitor[Running (monitor mode) (커널 모드)]
    RunningMonitor -- "Interrupt / interrupt 복귀" --> RunningUser
    RunningMonitor -- "Scheduler dispatch (preempted)" --> Ready[Ready]
    Ready -- "Wakeup (event occur, I/O completed)" --> Blocked[Blocked (Sleep, wait)]
    Blocked -- "Wakeup (event occur, I/O completed)" --> Ready
    RunningMonitor -- "Scheduler dispatch" --> Suspended[Suspended]
    Suspended -- "Resume" --> Ready
  
```

0:55:50 -4:09 1x 1080p

Running, Ready, blocked, Suspended 이라는 프로세스 상태는

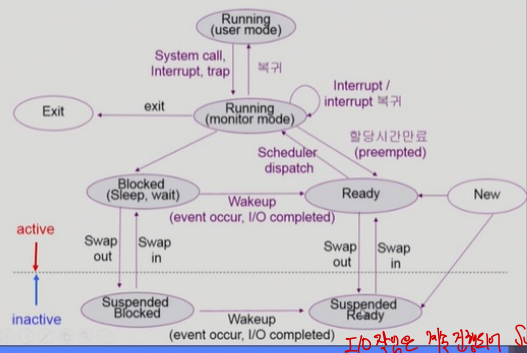
운영체제가 관리하기 편하게 구분시킨 것.

운영체제에 커널 코드가 실행될 때

따라서 커널 모드에서 Running하고 있다고 표현하지,  
운영체제가 Running하고 있다고 표현X.

Suspended 추가.

### 프로세스 상태도



IO 작업은 계속 진행되나 Suspended Ready로 넘어갈 수 있음

0:58:31

-1:28

1x

1080p