

CPU 스케줄링 알고리즘(3)

경성대학교 양희재 교수님의 강의 자료를 정리한 것입니다.

CPU Scheduling Algorithms

CPU Scheduling Algorithms

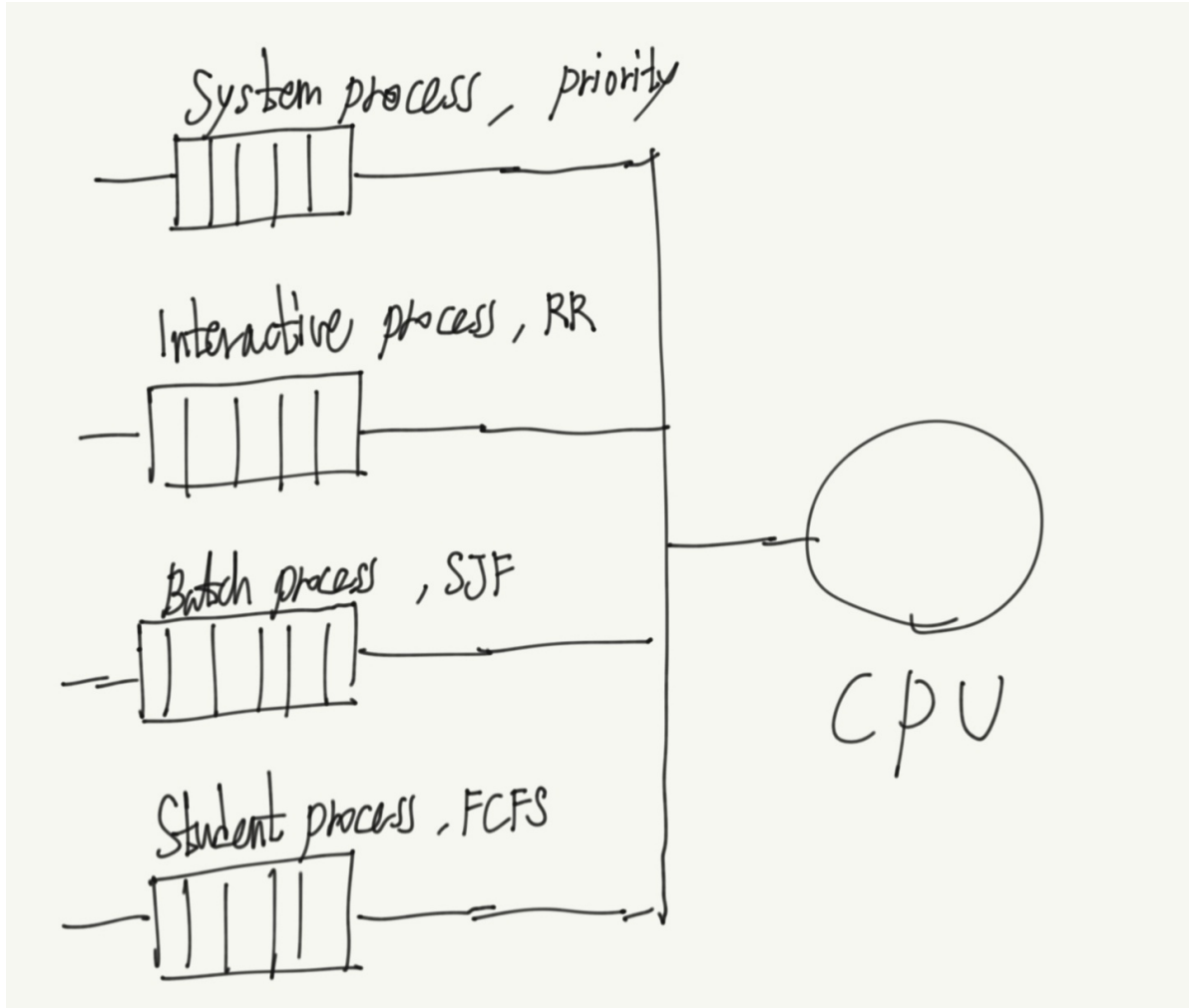
- First-Come, First-Served (FCFS)
- Shortest-Job-First (SJF)
 - Shortest-Remaining-Time-First
- Priority
- Round-Robin (RR)
- Multilevel Queue
- Multilevel Feedback Queue

Multilevel Queue

Multilevel Queue Scheduling

- Process groups
 - System processes
 - Interactive processes
 - Interactive editing processes
 - Batch processes
 - Student processes
- Single ready queue → Several separate queues
 - 각각의 Queue 에 절대적 우선순위 존재
 - 또는 CPU time 을 각 Queue 에 차등배분
 - 각 Queue 는 독립된 scheduling 정책

- System processes : OS안에서 OS 나름의 작업을 하는 것 (가상메모리 읽기, 파일 매핑 등)
- Interactive processes : 사용자와 대화하는 것 / 마우스, 키보드 등을 이용 (게임 등)
- Interactive editing processes : Interactive processes 중 대표적인 것으로 편집하는 것 (워드프로세서 등)
- Batch processes : 대화형X, 꾸러미를 컴퓨터가 일괄처리, interaction을 안하므로 좀 느리게 처리되어도 됨
- Student processes : 학생 정보 처리

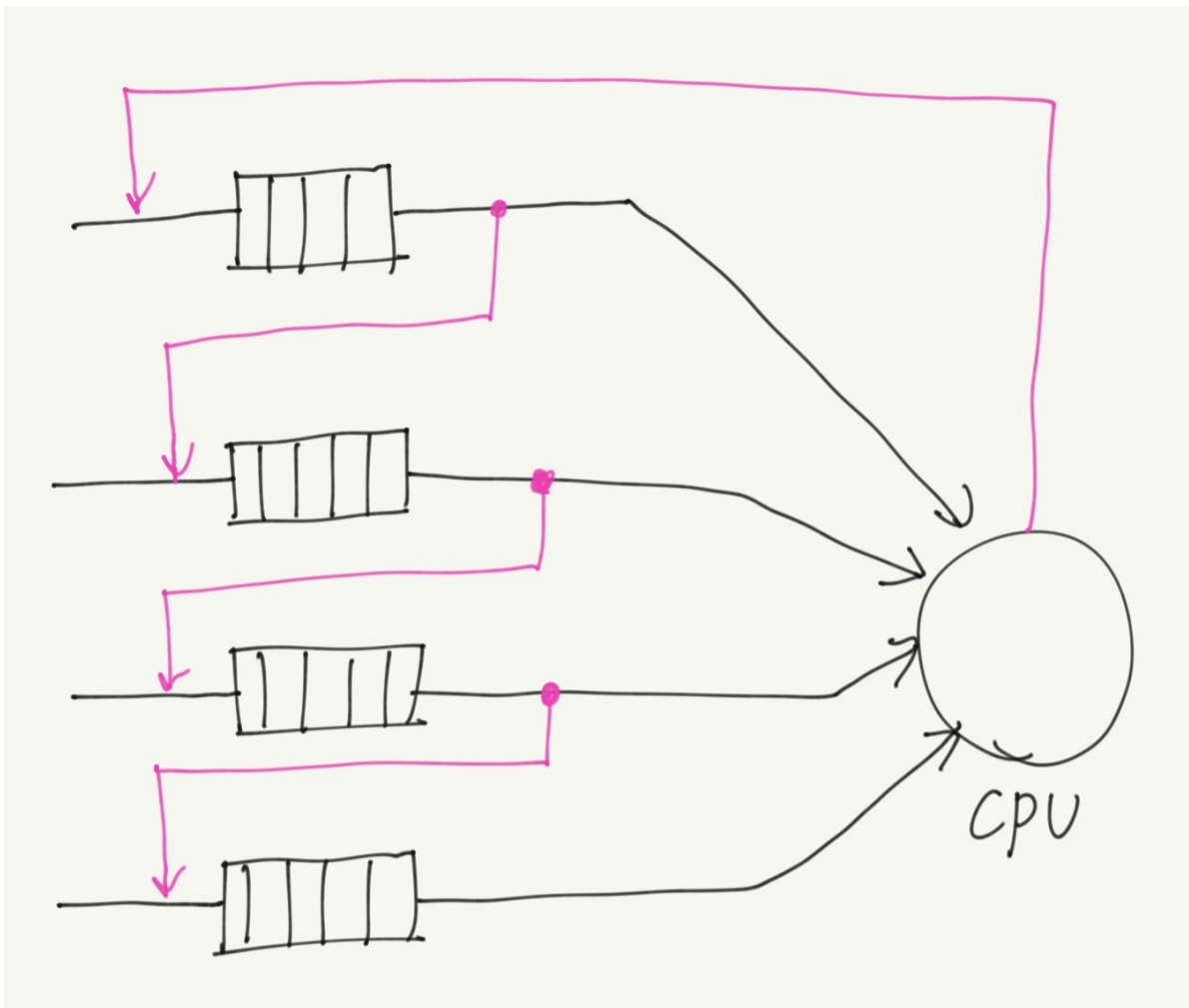


Multilevel Feedback Queue

Multilevel *Feedback* Queue Scheduling

- 복수 개의 Queue
- 다른 Queue 로의 점진적 이동
 - 모든 프로세스는 하나의 입구로 진입
 - 너무 많은 CPU time 사용 시 다른 Queue 로
 - 기아 상태 우려 시 우선순위 높은 Queue 로

앞의 정책의 프로세스로 실행하다 되지 않으면 다음 정책의 프로세스로도 실행해 봄



실제 계산이 필요한 앞의 4가지 프로세스가 더 중요함!

프로세스 생성과 종료

Process Creation

Process Creation

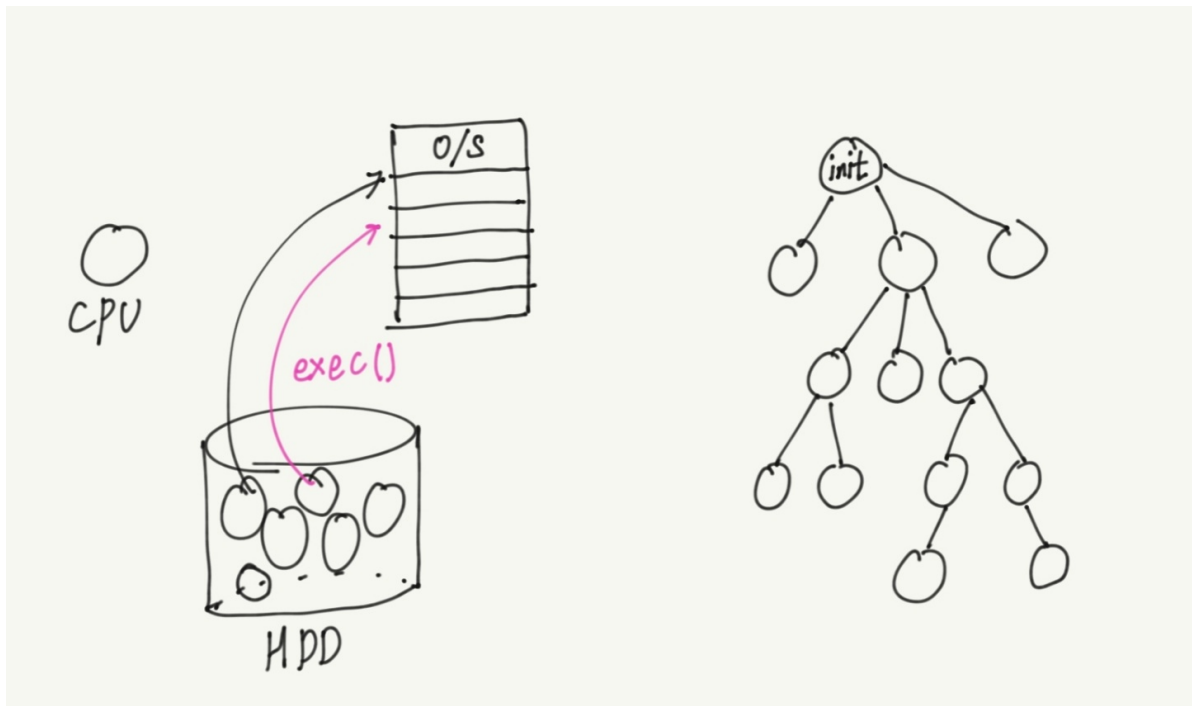
- 프로세스는 프로세스에 의해 만들어진다!
 - 부모 프로세스 (Parent process)
 - 자식 프로세스 (Child process)
 - cf. Sibling processes
 - 프로세스 트리 (process tree)
- Process Identifier (PID)
 - Typically an integer number
 - cf. PPID
- 프로세스 생성
 - `fork()` system call - 부모 프로세스 복사
 - `exec()` - 실행파일을 메모리로 가져오기

1. HDD 에 여러 프로그램과 OS 존재
2. 가장 처음 올라간 OS가 첫번째 프로세스를 만듦 (Unix의 경우 init이라는 프로세스 생성)
3. 프로세스가 자식 프로세스들을 생성

Sibling processes : 부모가 같은 프로세스

PID : 프로세스 고유의 ID, 양수의 정수값, 중복X

PPID : 부모의 PID



- 프로세스 생성
 1. `fork()` : `fork()` system call을 이용해 새로운 프로세스 생성

2. exec() : 실제로 일을 시키기 위해서 실행파일을 하드디스크에서 메모리로 가져오는 것

예제: Windows 7 프로세스 나열



Window10의 경우

작업 관리자						
파일(F) 옵션(O) 보기(V)						
프로세스	성능	앱 기록	시작프로그램	사용자	세부 정보	서비스
이름	PID	상태	사용자 이름	CPU	메모리(활성...	UAC 가상화
System Interrupts	-	실행 중	SYSTEM	00	0 K	
시스템 유휴 시간 프로세스	0	실행 중	SYSTEM	98	8 K	
System	4	실행 중	SYSTEM	00	24 K	
svchost.exe	8	실행 중	SYSTEM	00	2,016 K	허용 안 함
Registry	148	실행 중	SYSTEM	00	5,836 K	허용 안 함
smss.exe	480	실행 중	SYSTEM	00	256 K	허용 안 함
fontdrvhost.exe	564	실행 중	UMFD-0	00	1,796 K	사용 안 함
WinStore.App.exe	600	일시 중단됨	landform....	00	0 K	사용 안 함
svchost.exe	632	실행 중	NETWORK...	00	7,388 K	허용 안 함
csrss.exe	640	실행 중	SYSTEM	00	1,000 K	허용 안 함
wininit.exe	800	실행 중	SYSTEM	00	684 K	허용 안 함
csrss.exe	808	실행 중	SYSTEM	00	1,216 K	허용 안 함
services.exe	872	실행 중	SYSTEM	00	4,456 K	허용 안 함
lsass.exe	892	실행 중	SYSTEM	00	6,836 K	허용 안 함
svchost.exe	1020	실행 중	SYSTEM	00	9,572 K	허용 안 함
svchost.exe	1028	실행 중	SYSTEM	00	2,124 K	허용 안 함
winlogon.exe	1084	실행 중	SYSTEM	00	1,284 K	허용 안 함
amdow.exe	1156	실행 중	landform....	00	512 K	사용 안 함
Mattermost.exe	1208	실행 중	landform....	00	32,300 K	사용 안 함
dwm.exe	1212	실행 중	DWM-1	00	34,848 K	사용 안 함
GamePlay.exe	1220	실행 중	landform....	00	1,056 K	사용 안 함

예제: Ubuntu Linux 프로세스 나열

```

hjyang@rn303:~$ ps -l
 F S   UID     PID   PPID  C PRI   NI ADDR SZ  WCHAN  TTY          TIME CMD
 0 S   1000    2197   2189  0  80    0  -   1799 wait   pts/0    00:00:00 bash
 0 R   1000    2368   2197  0  80    0  -   1177 -      pts/0    00:00:00 ps

hjyang@rn303:~$ ps -axl
 F   UID     PID   PPID  PRI   NI     VSZ   RSS WCHAN  STAT TTY          TIME COMMAND
 4     0         1         0  20    0    3536   1948 poll_s Ss   ?         0:00 /sbin/init
 1     0         2         0  20    0         0         0 kthrea S   ?         0:00 [kthreadd]
 1     0         3         2  20    0         0         0 run_k S   ?         0:00 [ksoftirqd/0]
 1     0         4         2  20    0         0         0 worker S   ?         0:00 [kworker/0:0]
 5     0         5         2  20    0         0         0 worker S   ?         0:00 [kworker/u:0]
 .....
 1  1000    1820         1  20    0   55944   3992 poll_s Sl   ?         0:00 /usr/bin/gnome-...
 4  1000    1831    1658  20    0   50924   9096 poll_s Ssl  ?         0:00 gnome-session --sessio...
 0  1000    2196    2189  20    0    2404    724 unix_s S   ?         0:00 gnome-pty-helper
 0  1000    2197    2189  20    0    7196   3572 wait   Ss   pts/0    0:00 bash
 0  1000    2370    2197  20    0    4708    708 -      R+   pts/0    0:00 ps -axl
hjyang@rn303:~$

```

ps : process status, 현재 내컴퓨터에서 돌아가는 process 보여줌

UID : user id

PRI : 우선순위

ProcessTermination

Process Termination

- 프로세스 종료
 - *exit()* system call
 - 해당 프로세스가 가졌던 모든 자원은 O/S 에게 반환
(메모리, 파일, 입출력장치 등)

쓰레드(Thread)

Thread?

- 쓰레드 (Thread)
 - 프로그램 내부의 흐름, 맥

```
class Test {  
    public static void main(String[] args) {  
        int n = 0;  
        int m = 6;  
        System.out.println(n+m);  
        while (n < m)  
            n++;  
        System.out.println("Bye");  
    }  
}
```

Multithreads

- 다중 쓰레드 (Multithreads)
 - 한 프로그램에 2개 이상의 맥
 - 맥이 빠른 시간 간격으로 스위칭 된다 ⇒ 여러 맥이 동시에 실행 되는 것처럼 보인다 (concurrent vs simultaneous)
- 예: Web browser
 - 화면 출력하는 쓰레드 + 데이터 읽어오는 쓰레드
- 예: Word processor
 - 화면 출력하는 쓰레드 + 키보드 입력 받는 쓰레드 + 철자/문법 오류 확인 쓰레드
- 예: 음악 연주기, 동영상 플레이어, Eclipse IDE, ...

concurrent : 빠른 스위칭으로 동시에 진행되는 것 처럼 보이는 것

simultaneous : 진정으로 한 순간에 두 개의 일을 하는 것 (CPU가 두 개 이상인 경우)

Thread vs Process

Thread vs Process

- 한 프로세스에는 기본 1개의 쓰레드
 - 단일 쓰레드 (single thread) 프로그램
- 한 프로세스에 여러 개의 쓰레드
 - 다중 쓰레드 (multi-thread) 프로그램
- 쓰레드 구조
 - 프로세스의 메모리 공간 공유 (code, data)
 - 프로세스의 자원 공유 (file, i/o, ...)
 - 비공유: 개별적인 PC, SP, registers, stack
- 프로세스의 스위칭 vs 쓰레드의 스위칭

한 프로세스 내에서 code나 data는 공유하나 stack은 쓰레드 별로 따로 사용!

- 옛날 : CPU가 프로세스 단위로 일을 수행함
- 요즘 : 프로세스 내에서도 스레드가 존재해서 일을 처리

