

## [운영체제 설계 과제물]

2015년 1학기

### 1. 주제: CPU 스케줄링 프로그램의 설계 및 구현

### 2. 수행 방법:

3인 1조 팀 단위로 프로젝트를 진행하며, 보고서에 각자가 수행한 역할에 대해서 상세하게 기술할 것

### 3. 수행 일정

5/12(화): 프로젝트 팀 구성 완료(팀원 중 1명을 팀장으로 위촉)

6/3(수): 프로젝트 최종 보고서 및 프로그램 제출

### 4. 입출력 데이터의 구성

#### (1) 프로그램의 입력 데이터의 구성

- 프로그램의 입력 데이터는 type, process\_id, priority, computing\_time의 4개 필드로 구성된다.

type: 0은 프로세스의 생성을 나타내며, process-id, priority, computing\_time을 입력 받게된다.

type: 1은 하나의 프로세스의 시간할당량이 다되어 스케줄이 되어야 함을 나타내며, process\_id, priority, computing\_time의 값은 0이 되며, 특별한 의미는 없다.(RR과 관련된 알고리즘에서만 사용)

type -1은 입력 완료를 의미한다. 입력 완료 이후에는 Queue에 있는 일반 프로세스들이 우선순위에 따라 차례대로 스케줄링 된다.

#### (2) 프로그램 입력 예제

- 입력

0	1	25	80
0	2	15	40
0	3	8	30
1	0	0	0
0	4	12	10
0	5	22	30
0	6	28	50
1	0	0	0
0	7	5	20
0	8	3	40
0	9	13	60
1	0	0	0
0	10	24	45
-1			

#### (3) 프로그램의 출력

- 프로그램의 출력 필드는 Process\_id, priority, computing\_time, turn\_around time의 4가지 필드를 가진다.
- Process\_id는 입력 받은 프로세스의 id를 출력한다.
- priority는 프로세스가 종료되기 직전의 변경된 priority를 출력한다.
- computing\_time은 입력시 지정된 computing\_time을 출력한다.
- turn\_around time은 프로세스가 종료된 시점에서 프로세스가 입력된 시점의 상대시간을 뺀 값을 출력한다.

#### (4) 프로그램 출력 필드

Process_id	priority	computing_time	turn_around time
2	12	10	40

### 5. 프로젝트의 설계

(1) 4의 입출력 데이터 구성을 가지는 조건에서, 다음의 CPU 스케줄링 알고리즘 중에 3가지를 골라서 설계를 하고, 프로그램으로 구현하시오.

- FCFS(First-Come-First-Served) 알고리즘
- Round-Robin 알고리즘(default time quantum = 20): 시간할당량(time quantum)은 쉽게 변경 가능하고, 프로그램 실행에 문제가 없도록 작성 할 것
- SPN(Shortest Process Next) 알고리즘
- SRTF(Shrotest Remaining Time First) 알고리즘
- HRRN(Highest Response Ration Next) 알고리즘
- Multi-level Feedback Queue 알고리즘
- 우선순위 기반 알고리즘
- 여러 개의 스케줄링 알고리즘을 혼합한 알고리즘

(2) 하나의 알고리즘을 한 명이 맡아서 설계 및 구현할 것(실력에 따라 쉬운 것과 어려운 것 배정: 실력 향상이 목적이므로 절대 포기하지 말 것)

(3) 선택한 3개의 알고리즘을 normalized average turn\_around time을 이용하여, 비교, 평가하시오.

### 6. 과제물 제출 방법

- e-class의 팀 프로젝트 게시판에 업로드할 것(프로그램 제출은 팀장의 학번으로 제출)
- 3개의 프로그램과 입력 데이터를 UNIX의 zip 프로그램으로 압축하여 제출할 것
- 학과 서버(computer.kpu.ac.kr)에서 gcc를 이용하여 컴파일한 후 검증할 예정임

### 7. 프로그램 설계 시 Tips

- turn\_around time을 어떻게 계산해야 되는지 고려
- 프로세스의 자료 구조에 새로운 필드가 추가되어야함(2개정도의 필드)

### 8. 기타 주의 사항

- 설계과제물 최종보고서 작성을 충실히 하기 바랍니다.(중요)
- 과제물 제출이 delay가 된 경우 감점합니다.

## 별첨. 프로젝트 설계 예제

(다음 설명을 만족하는 CPU 스케줄링 프로그램을 설계 및 구현하시오)

### (1) 프로그램의 입력 데이터의 구성

- 프로그램의 입력 데이터는 type, process\_id, priority, computing\_time의 4개 필드로 구성된다.  
type: 0은 프로세스의 생성을 나타내며, process-id, priority, computing\_time을 입력 받게된다.  
type: 1은 하나의 프로세스의 시간할당량이 다되어 스케줄이 되어야 함을 나타내며, process\_id, priority, computing\_time의 값은 0이 되며, 특별한 의미는 없다.  
type -1은 입력 완료를 의미한다. 입력 완료 이후에는 Queue에 있는 일반 프로세스들이 우선순위에 따라 차례대로 스케줄링 된다.

### (2) 생성되는 프로세스의 종류

- 생성되는 프로세스의 종류는 실시간 프로세스와 일반 프로세스로 구분되며, 실시간 프로세스의 priority 값은 음수가, 일반 프로세스의 priority 값은 양수가 된다.
- priority 값은 숫자가 작을수록 우선순위가 높다.
- 실시간 프로세스의 priority 값은 -1에서 -10까지 값이 할당될 수 있으며, 일반 프로세스의 priority 값은 1부터 31까지의 값이 할당될 수 있다.

### (3) 스케줄링 원칙

- 일반 프로세스는 과제물 1에서 설계한 3개의 priority queue로 구성되는 것을 가정한다. Queue1은 priority 1에서 10까지, Queue2는 priority 11에서 20까지, Queue3는 priority 21에서 30까지가 배정된다. priority가 같은 프로세스의 경우 FIFO(First\_In, First\_Out)의 규칙에 따른다(먼저 생성된 프로세스의 priority가 더 높음).
- 마지막으로 Queue4는 priority가 31인 일반 프로세스가 배정된다.
- 스케줄링은 일반 프로세스에서 priority 가 가장 높은 프로세스가 선택되어 CPU를 배정받게 되며, 시간할당량(time quantum)을 20이며, 20이내에 종료하지 못하면 priority가  $\text{기존값} + 10$ 으로 변경되어 다음 우선순위 Queue에 연결된다. 이 프로세스는 나중에 순서가 되면 CPU를 재할당 받아 나머지 작업을 처리한다.
- 만일 변경된 priority가 30을 초과하게 되면 priority 값은 31로 고정되게 되며(즉 변경된 priority의 값은 31을 넘지 못한다), 4번째 우선 순위 큐인 Queue4에 배정된다. Queue4는 우선순위가 같으므로 round-robin 방식의 스케줄링 알고리즘이 적용된다.
- 만일 실시간 프로세스가 생성되는 경우, 현재 수행중인 일반프로세스의 시간할당량이 완료된 후, 우선적으로 CPU를 배정한다. 실시간 프로세스는 시간할당량에 관계 없이 자신의 computing\_time동안 CPU를 점유하여 작업을 수행한다.

### (4) 프로그램의 출력

- 프로그램의 출력 필드는 Process\_id, Queue\_id, priority, computing\_time, turn\_around time의 5가지 필드를 가진다.
- Process\_id는 입력 받은 프로세스의 id를 출력한다.

- Queue\_id는 일반 프로세스의 경우 프로세스의 수행이 종료되기 직전에 연결되어 있던 Queue의 id를 출력한다. 실시간 프로세스는 real\_time이라고 출력한다.
- priority는 프로세스가 종료되기 직전의 변경된 priority를 출력한다.
- computing\_time은 입력시 지정된 computing\_time을 출력한다.
- turn\_around time은 프로세스가 종료된 시점에서 프로세스가 입력된 시점의 상대시간을 뺀 값을 출력한다.

(5) 프로그램 입력 예제

```
- 입력
0      1      25      80
0      2      15      40
0      3      8       30
1      0      0       0
0      4      12      10
0      5      -2      30
0      6      28      50
1      0      0       0
0      7      5       20
0      8      -3      40
0      9      13      60
1      0      0       0
0      10     24      45
-1
```

(6) 프로그램 출력 예제

Process_id	Queue_id	priority	computing_time	turn_around time
5	real_time	-2	30	30
4	2	12	10	40
8	real_time	-3	40	40
7	1	5	20	60
3	2	18	30	170
2	3	25	40	250
9	4	31	60	190
10	4	31	45	235
6	4	31	50	335
1	4	31	80	405

6. 관련 스케줄링 알고리즘

- 우선순위 기반 스케줄링 알고리즘
- Multi-level Feedback Queue 스케줄링 알고리즘
- round-robin 스케줄링 알고리즘