

## 2015 년 2 학기 운영체제 과제 #2

2015/10/2 (금)

### 1. 과제 내용

운영체제 스케줄러 알고리즘을 시뮬레이션 하는 프로그램을 구현하시오.

### 2. 요구 사항

- A. C 언어와 기본 C 라이브러리(libc) API 만 이용하고, 소스 파일 'sched.c' 한 개에 모든 기능을 구현합니다. 실행파일 이름은 'sched' 입니다.
- B. 텍스트 형식의 프로세스 스케줄링 목록 입력 파일을 프로그램 첫 번째 인수로 받아들여 처리합니다.
- 한 행에 스케줄링 할 한 개의 프로세스 스케줄링 정보가 기술되어 있습니다. 예)
- ```
P1 0 3 1
P2 2 6 2
P3 4 4 1
P4 6 5 3
P5 8 2 1
```
- 한 행은 "**id arrive-time service-time priority**" 형식이며 각 항목은 다음과 같습니다.
- **id** : 알파벳 대문자와 숫자로 구성된 2 자리 프로세스 ID (중복 허용 안 함)
  - **arrive-time** : 0 이상 30 이하 정수로 명시된 프로세스 도착 시각 (초)
  - **service-time** : 1 이상의 30 이하 정수로 명시된 프로세스 서비스 시간 (초)
  - **priority** : 1 이상 10 이하 정수로 명시된 프로세스 우선 순위 (작은 값이 높은 우선 순위)
- 프로세스 도착 시간은 첫 번째 행이 가장 빠르고 다음 행부터는 이전 행의 프로세스 도착 시간과 같거나 나중이어야 합니다.
- 각 항목은 공백[space] 한 문자로 구분됩니다.
- 행의 첫 문자가 '#' 일 경우 주석이므로 무시합니다. 공백만 있거나 비어있는 행도 무시합니다.
- 개행문자는 유닉스 표준 문자 '\n'만 사용하고, 마지막 행은 개행문자가 없어도 됩니다.
- 지정할 수 있는 프로세스 개수는 최대 260 개입니다. (260 = 알파벳(26) \* 숫자(10))
- C. 프로세스 스케줄링 목록 정보를 기반으로 4 가지 스케줄링 알고리즘에 대한 시뮬레이션 결과를 표준입력(stdout) 장치로 출력합니다.

- 지원해야 하는 알고리즘 이름과 종류는 다음과 같습니다.
  - **SJF** : Shortest-Job-First Scheduling
  - **SRT** : Shortest-Remaining-Time-First Scheduling
  - **RR** : Round-Robin Scheduling
  - **PR** : Priority Scheduling (preemptive)
- SJF, SRT, RR 스케줄링 알고리즘에서는 우선순위 항목을 사용하지 않습니다.
- 각 스케줄링 알고리즘의 스케줄링 결과는 간트 차트(Gantt Chart) 형태로 출력합니다. 예를 들어 다음과 같습니다.

```
[SJF]
P1 ***
P2      *
P3      *
P4      *
P5      *
CPU TIME: 20
AVERAGE TURNAROUND TIME: 7.60
AVERAGE WAITING TIME: 3.60
```

```
[SRT]
P1 ***
P2      *
P3      *
P4      *
P5      *
CPU TIME: 20
AVERAGE TURNAROUND TIME: 7.20
AVERAGE WAITING TIME: 3.20
```

```
[RR]
P1 ***
P2      *
P3      *
P4      *
P5      *
CPU TIME: 20
AVERAGE TURNAROUND TIME: 10.60
AVERAGE WAITING TIME: 6.60
```

```
[PR]
P1 ***
P2      *
P3      *
P4      *
P5      *
CPU TIME: 20
AVERAGE TURNAROUND TIME: 7.20
AVERAGE WAITING TIME: 3.20
```

- `'''` 글자 하나는 1 밀리초를 의미합니다.
- CPU 시간[CPU TIME]은 스케줄링 할 프로세스가 없을 때까지 걸리는 시간을 표시합니다.
- 평균 완료시간[AVERAGE TURNAROUND TIME]은 각 프로세스의 완료시간(=종료시각[complete time] - 도착시각[arrival time])을 구한 다음 모든 프로세스의 평균값을 소수점 아래 둘째 자리까지 표시합니다.
- 평균 대기시간[AVERAGE WAITING TIME]은 각 프로세스의 대기시간(= 완료시간[turnaround time] - 실행 시간[service time])을 구한 다음 모든 프로세스의 평균값을 소수점 아래 둘째 자리까지 표시합니다.

D. 오류가 있을 경우 표준에러(stderr) 장치로 에러 메시지를 출력하고 해당 행은 무시합니다.

- 입력파일을 지정하지 않았을 경우 (예: input file must specified)
- 잘못된 입력파일을 지정한 경우 (예: failed to load input file '%s')
- 요구사항에 맞지 않는 형식의 프로세스 정보가 있을 경우 (예: invalid format in line 3, ignored)
- `id` 가 형식에 맞지 않을 경우 (예: invalid process id 'Abc' in line 4, ignored)
- 이미 존재하는 `id` 를 다시 사용했을 경우 (예: duplicate process id 'P1' in line 4, ignored)
- 잘못된 프로세스 도착 시간을 지정했을 때 (예: invalid arrive-time '-1' in line 22, ignored)
- 잘못된 프로세스 서비스 시간을 지정했을 때 (예: invalid service-time '0' in line 23, ignored)
- 잘못된 프로세스 우선순위를 지정했을 때 (예: invalid priority '99' in line 24, ignored)

### 3. 구현 방법

- ✓ 가능하다면 **우분투(Ubuntu) 15.04** 버전 리눅스 배포판 사용을 권장하지만, 본인이 익숙한 리눅스 배포판을 사용해도 무방합니다. PC 에 리눅스를 직접 설치하거나, VirtualBox / VMware 등과 같은 프로그램을 이용해도 상관없습니다.

### 4. 채점 기준

- ✓ 보고서 (20 점) : 아래 목차를 따라 작성합니다.
  1. 과제 목표
  2. 관련 연구 (스케줄링 알고리즘 / API 함수 조사 포함) (5)
  3. 문제 해결 방법 (5)
  4. 실행 결과 (3)

- 5. 추가 및 개선할 부분과 결론 (5)
- 6. 참고 자료 / 문헌 (2)
- 7. 소스 코드 (없을 경우 보고서 점수 0 점 처리)
- ✓ 구현 (30 점)
  - 요구사항 A, B 구현 (5 점)
  - 요구사항 C 구현 (15 점)
  - 요구사항 D 구현 (5 점)
  - 소스 코드 가독성(readability) 및 주석(comments) (5 점)

## 5. 제출 방법

- ✓ 마감시간 : **2015/10/16 오후 6 시 (2 주)**
- ✓ 감점 : 늦게 제출할 경우 1 일 10 점 감점
- ✓ 제출 방법
  - 보고서 파일(PDF)과 소스 파일, Makefile, README 파일 그리고 소프트 카피 포함에서 압축하고, 파일 이름은 '반\_이름\_과제번호.zip' 형식의 이름을 사용 (예: 가\_홍길동\_hw1.zip)
  - [vanillo@realtime.ssu.ac.kr](mailto:vanillo@realtime.ssu.ac.kr) 로 제출

## 6. 기타

- ✓ 과제 해결 방법에 대한 논의나 도움은 서로간에 주고 받을 수 있겠지만 과제 자체는 전적으로 본인이 직접 스스로 해야 합니다. 만일 복사본이 발견되면 복사를 한 학생과 더불어 원본 제공자 또한 해당 과제 점수를 전체 0 점 처리합니다. 참고로, 이미 검증된 소스 코드 분석[parsing] 프로그램을 이용해 과제 소스 코드 동일성 여부를 검사합니다.
- ✓ 과제 관련 궁금한 점이나 문의 사항은 수업 게시판을 이용하시기 바랍니다.  
<https://groups.google.com/forum/#!forum/ssu-smyang>