

2019 년 2 학기 운영체제 과제 #2

2019/10/04(금)

1. 과제 내용

운영체제 다단계 큐 스케줄링(Multilevel Queue Scheduling) 알고리즘을 시뮬레이션하는 프로그램을 구현하시오.

2. 요구 사항

- A. C 언어와 기본 C 라이브러리(glibc) API만 이용하고, 소스 파일 'mutisched.c' 한 개에 모든 기능을 구현합니다. 실행파일 이름은 'multisched' 입니다.
- B. 텍스트 형식의 프로세스 스케줄링 목록 입력 파일을 프로그램 첫 번째 인수로 받아들여 처리합니다.

- 한 행에 스케줄링할 한 개의 프로세스 스케줄링 정보가 기술되어 있습니다. 예를 들어 다음과 같습니다.

```
# Process Scheduling Information  
  
P1 H 0 8 3  
P2 L 1 4 2  
P3 M 2 9 4  
P4 H 3 5 1
```

- 한 행은 "**id process-type arrive-time service-time priority**" 형식으로 구성되며 각 항목은 다음과 같습니다.
 - **id** : 알파벳 대문자와 숫자로 구성된 2 자리 프로세스 ID (중복 허용 안 함)
 - **process-type** : H(High process), M(Mid-process) 그리고 L(Low-process)
 - **arrive-time** : 0 이상 30 이하 정수로 명시된 프로세스 도착 시각 (밀리초)
 - **service-time** : 1 이상의 30 이하 정수로 명시된 프로세스 서비스 시간 (밀리초)
 - **priority** : 1 이상 10 이하 정수로 명시된 프로세스 우선 순위 (작은 값이 높은 우선 순위)
- 도착 시간은 첫 번째 행이 가장 빠르고 다음 행부터는 같거나 나중이어야 합니다.
- 각 항목은 공백[space] 한 문자로 구분됩니다.
- 행의 첫 문자가 '#' 일 경우 주석이므로 무시합니다. 공백만 있거나 비어있는 행도 무시합니다.

- 개행 문자는 유닉스 표준 문자 '\n'만 사용하고, 마지막 행은 개행 문자가 없어도 됩니다.
- 지정할 수 있는 프로세스 갯수는 최대 알파벳 대소문자(26*2)*숫자(10)이다.

C. 프로세스 스케줄링 목록 정보를 기반으로 다단계 큐 스케줄링(Multilevel Queue Scheduling)

시뮬레이션 결과를 표준입력(stdout) 장치로 출력합니다.

- 각 큐가 지원 해야 하는 알고리즘 이름과 종류는 다음과 같습니다.
 - **H** : Priority Scheduling (preemptive)
 - **M** : Shortest-Job-First Scheduling
 - **L** : First-Come, First-Served Scheduling
- 각 큐는 CPU 시간의 일정량을 받아서 자기 큐에 있는 프로세스를 스케줄링 합니다.
 - **H** : CPU시간의 60%
 - **M** : CPU시간의 40%
- SJF, FCFS 스케줄링 알고리즘에서는 우선순위 항목을 사용하지 않습니다.
- 각 스케줄링 알고리즘별 스케줄링 결과는 간트(Gantt) 차트 형태로 출력합니다. 예를 들어 다음과 같습니다.

- [Multilevel Queue Scheduling]

```

P1 ***          *****      *
P2                      *****
P3          *****      ***** *
P4      ***      **
```

CPU TIME: 26

AVERAGE TURNAROUND TIME: 19

AVERAGE WAITING TIME: 12.75

- '*' 글자 하나는 1 밀리초를 의미합니다.
- CPU 시간[CPU TIME]은 스케줄링할 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')
- 부정확한 형식의 프로세스 정보가 있을 경우 (예: invalid format in line 3, ignored)
- id가 형식에 맞지 않을 경우 (예: invalid process id 'Abc' in line 4, ignored)
- 이미 존재하는 id를 다시 사용했을 경우 (예: duplicate process id 'P1' in line 4, ignored)
- 프로세스 타입(process-type)을 지정하지 않거나, 없는 타입을 지정한 경우 (예: invalid format 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)

4. 참고 자료

- ✓ 과제 압축 파일 안에 포함된 소스 코드와 설정 파일
- ✓ 가능한 **우분투(Ubuntu) 최신 버전** 리눅스 배포판 사용
- ✓ gcc 컴파일러 및 개발환경: 터미널에서 'sudo apt-get install build-essential' 실행

5. 채점 기준

- ✓ 보고서 (10 점) : 아래 목차를 따라 작성합니다.
 1. 소개
 2. 관련 연구 (소스 분석 / 사용한 시스템 콜, API 함수 설명) (4)
 3. 문제해결방법 구현 방법 (4)
 4. 결론(2)
- ✓ 구현 (10 점)
 - B 구현(4 점)

- C 구현(3 점)
- D 구현(3 점)

6. 제출 방법

- ✓ 마감시간: **2019/10/18 오후 6 시 (2 주)**, 늦게 제출할 경우 1 일 10 점 감점
- ✓ 제출 방법
 - 보고서 파일과 소스 파일, 테스트 스케줄링파일(*.txt) 압축.
 - 파일 이름은 '반_이름_과제번호.zip' 형식의 이름을 사용 (예: 가_홍길동_hw1.zip)
 - vanillo@realtime.ssu.ac.kr 메일 계정으로 제출.

6. 기타

- ✓ 만일 복사본이 발견되면 복사를 한 학생과 더불어 원본 제공자 또한 **한 학기 과제 점수를 전체 0 점** 처리합니다.