

운영체제 과제 1

염선욱 물리학과 201921786

단계	완료여부
List 자료구조 파악: 예제 수행	O
list_for_each_entry()를 이용한 순회	O
list_for_each_safe_reverse()를 이용한 순회	O
과제 1-1 완료 (JOTA 확인)	O
Idle process 구현	O
push_job() 및 push_ct() 구현	O
과제 1-2 완료 (JOTA 확인)	O
210 clock context-switching 전까지 구현	O
과제 1-3 완료 (JOTA 확인)	X

예제를 실행함으로써 linux list library 기반 연결리스트를 유용하게 활용할 수 있는 법을 알았다. 과제 1-1을 수행할 때는 그냥 예제처럼 쉽게 접근했다. 하지만, 연결 리스트의 장점인 구조체끼리도 연결할 수 있다는 점을 놓쳐서 조금 헤맸다. 포인터를 활용해서 process에 code_tuple들을 연결시킬 수 있다는 점을 생각하면서 예제를 참고했더니 쉽게 해결할 수 있었다. process 구조체 멤버로 리스트 헤드를 하나 두고, 그 head에다가 code_tuple들을 연결하면 되었다. 처음에는 main 함수 안에 모든 코드를 작성했다. 추후에 함수를 만들어서 개념적으로 job queue에 push하는 기능의 함수와 process 하나에 code_tuple을 push하는 기능의 함수를 활용하고, 처음 코드는 주석 처리했다.

과제 1-2를 늦게 시작해서 반성하고 있지만, 풀긴 했다. 초반에는 무한 루프를 한 번 반복할 때마다 clock이 1씩 증가하도록 풀어서 vscode 내에서는 정답이 나와서 JOTA에 제출했지만, 시간초과가 났다. 그래서 무한루프를 한 번 돌 때 clock이 1씩 증가하는 게 아니라는 생각을 하게 되었다. 분명 과제 안내에는 1씩 돌고 있다고 가정했었다. CPU나 IO 작업을 만났을 때 clock이 한꺼번에 증가하도록 만들어서 시작했다. 역시 어려웠다. 이상하게 풀고 있다고 느껴서 다시 원래 풀이로 돌아왔다. 아마도 처음에 JOTA에 제출한 코드에는 시간을 많이 소요하는 다른 이유가 있을 거라고 생각한다. 과제 1-2에서는 1-1에서 만든 함수를 활용했다. Idle process를 마지막에 push하는 작업이 있어서 중복 코드를 줄일 수 있었다. 변수를 추가해서 현재 프로세스의 상태를 관찰하는 게 중요했다. psize는 fread로 읽어올 프로세스의 소스 크기이다. csize는 fread로 읽어올 code_tuple의 크기이다. cnt_ps는 프로그램의 총 프로세스 개수(idle 포함)다. 그리고 무한루프를 돌면서 필요한 변수에는 clocks(현재 시간), flag(다음 프로세스를 실행할 시간), idle(프로세서가 idle 상태일 때의 총 시간), load_cnt(job_q에 넣게 된 프로세스의 누적 개수), left_cnt(현재 job_q에 있는 프로세스 개수), all_loaded(load_cnt가 cnt_ps가 되면 1을 반환)가 있다.

무한루프는 먼저 프로세스 3개와 idle이 다 들어올 때까지 도착시간 순서대로 적재한다. 그 다음에는 크게 두 개의 반복문을 중첩하고 있다. 먼저 ready_q를 for-each 형식으로 반복문을 돌면서 각 프로세스를 FCFS인 순차적으로 반복한다. 여기서 idle process는 제외하고, 어차피 flag 시간이 돼야 다음 code를 실행하

로, clock이 flag와 동일할 때만 분기하도록 설계했다. ready_q 요소를 반복할 때 프로세스를 종료할 때 context-switching이 일어나도록(FCFS 구조이므로) 했다. 이때 PC(program counter)의 초기값은 -1로 두었다. code_bytes의 절반보다 1만큼 작으면 종료되는 것이다. 처음에는 PC의 초기값을 0으로 두고 설계해봤지만 쉽지 않아서 -1로 두고, 첫 code를 실행할 때 0, 두 번째 code를 실행할 때 1을 가리키도록 설계했다. 그리고 ready_q의 반복문 안에 pc라는 변수를 두어서 현재 프로세스의 PC를 추적하였다. 주의할 점은 프로그램을 종료할 때는 context-switching과 ticking(자연스러운 clock 증가)를 상쇄해줘야 한다는 것이었다. 마지막으로 동적으로 할당한 메모리는 과제 1-1처럼 연결 리스트 기반 함수를 활용해서 해제했다. 예제를 처음 봤을 때는 잘 못 느꼈지만, 과제 1-2를 풀면서 연결리스트의 강점을 확실히 느낄 수 있었다. Job_queue라는 헤드에 job(프로세스)들을 양방향으로 마음껏 연결할 수 있고, 그 상태에서는 그냥 바로 ready_q 헤드나 wait_q 헤드를 통해서 1, 2, 3번 프로세스 순서대로만이 아니라 원하는 프로세스를 원하는 순서대로 연결할 수 있다는 것이다. 하지만, 과제 1-2에서는 FCFS라서 job_q의 순서인 0, 100, 1, 2로 넣었다.

과제 1-3은 operation-processing과 context-switching이 중요해 보였다. 아쉽게 풀진 못했지만, 마지막에 가까운 210 clock에서 ps#1에서 ps#100으로 context-switching을 하는 부분이 계속 막혔다. 100번 프로세스가 if문에 의해 다르게 처리돼서 정상적으로 동작하지 않는 것 같다.

아래는 JOTA 제출 결과 캡처 사진입니다. 1-3은 틀렸지만 첨부했습니다.

DM::OJ

PROBLEMS

SUBMISSIONS

USERS

CONTESTS

ABOUT

Hello, os201921786.

os201921786의 운영체제 과제 1-1 제출

2024년 5월 2일 오후 6시 54분

C

View source

다시 제출

Execution Results

✓✓✓✓✓

➤ Test case #1: AC [0.003s, 1.01 MB] (2/2)

➤ Test case #2: AC [0.003s, 1.01 MB] (2/2)

➤ Test case #3: AC [0.003s, 1.01 MB] (2/2)

➤ Test case #4: AC [0.008s, 1.01 MB] (2/2)

➤ Test case #5: AC [0.003s, 1.01 MB] (2/2)

Resources: 0.020s, 1.01 MB

Maximum single-case runtime: 0.008s

Final score: 10/10 (10.0/10 points)

운영체제과제 - 57일 09:19:55

Windows 정품 인증

[설정]으로 이동하여 Windows를 정품 인증합니다.

proudly powered by DMOJ | 한국어 (ko)

DM::OJ

PROBLEMS

SUBMISSIONS

USERS

CONTESTS

ABOUT

Hello, os201921786.

os201921786의 운영체제 과제 1-2 제출

2024년 5월 3일 오전 12시 49분

C

View source

다시 제출

Execution Results

✓✓✓✓✓

➤ Test case #1: AC [0.003s, 1.01 MB] (2/2)

➤ Test case #2: AC [0.003s, 1.01 MB] (2/2)

➤ Test case #3: AC [0.003s, 1.01 MB] (2/2)

➤ Test case #4: AC [0.003s, 1.01 MB] (2/2)

➤ Test case #5: AC [0.003s, 1.01 MB] (2/2)

Resources: 0.016s, 1.01 MB

Maximum single-case runtime: 0.003s


Final score: 10/10 (10.0/10 points)

운영체제과제 - 57일 09:18:04

Windows 정품 인증

[설정]으로 이동하여 Windows를 정품 인증합니다.

proudly powered by DMOJ | 한국어 (ko)




PROBLEMS

SUBMISSIONS

USERS

CONTESTS

ABOUT

 Hello, os201921786.

os201921786의 운영체제 과제 1-3 제출

2024년 5월 3일 오후 11시 44분

View source

다시 제출

Execution Results

✖ ✖ ✖ ✖ ✖

> Test case #1: WA (0.004s, 1.02 MB) (0/2)

> Test case #2: WA (0.004s, 1.02 MB) (0/2)

> Test case #3: WA (0.007s, 1.02 MB) (0/2)

> Test case #4: WA (0.004s, 1.02 MB) (0/2)

> Test case #5: WA (0.004s, 1.02 MB) (0/2)

Resources: 0.022s, 1.02 MB

Final score: 0/10 (0.0/10 points)

운영체제과제 - 57일 09:15:28

Windows 정품 인증

[설정]으로 이동하여 Windows를 정품 인증합니다.

proudly powered by DMOJ | 한국어 (ko)