**12171651 컴퓨터공학과 오윤석**

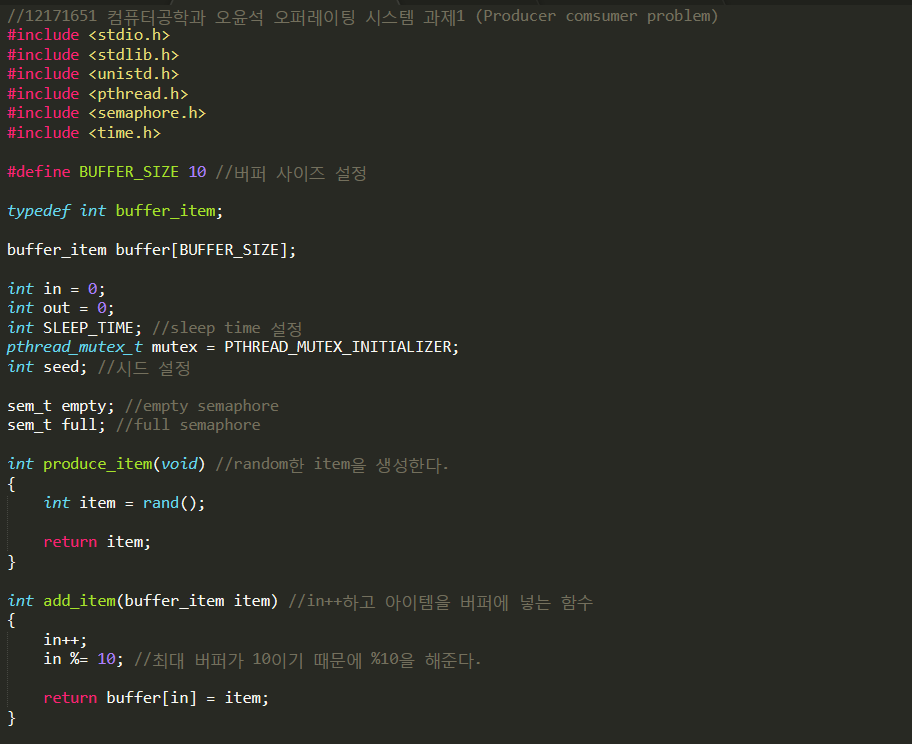
**과제1: Producer consumer problem**

1. 이해한 내용

Producer consumer problem이란 여러 개의 프로세스를 어떻게 동기화 할 것이라는 문제이다. 우리의 문제에서는 mutex를 사용하여 구현을 하였고, 또한 Entry section, critical section, Exit section을 직접 코딩을 통해 구현하고 또한 producer thread와 consumer thread를 직접 코드로 생성함으로써 문제를 해결하였다.

여러 명의 Producer는 유한한 개수의 item을 임시로 버퍼에 저장을 해놓고, 여러 명의 consumer는 producer가 생성을 한 item을 소비하는 것을 해결하는 문제이다. 여기서 문제점이 발생할 수 있는데 만약 producer가 저장할 공간이 없을 때, consumer가 소비할 item이 없을 때로 나누게 된다. 이 문제점들은 Empty와 full로 해결을 할 수 있다. Producer는 item을 생성하기전 empty를 사용해 버퍼에 빈 공간이 있나 확인을 한다. 또한 consumer는 full을 사용해 버퍼에 가지고 갈 item이 있는지 확인할 수 있다. 그리고 동시에 critical section에 진입할 수 있는 문제점도 있는데 이것은 mutex를 사용해 mutual exclusion과 progress를 만족시켜준다.

1. 스크린샷과 분석포인트



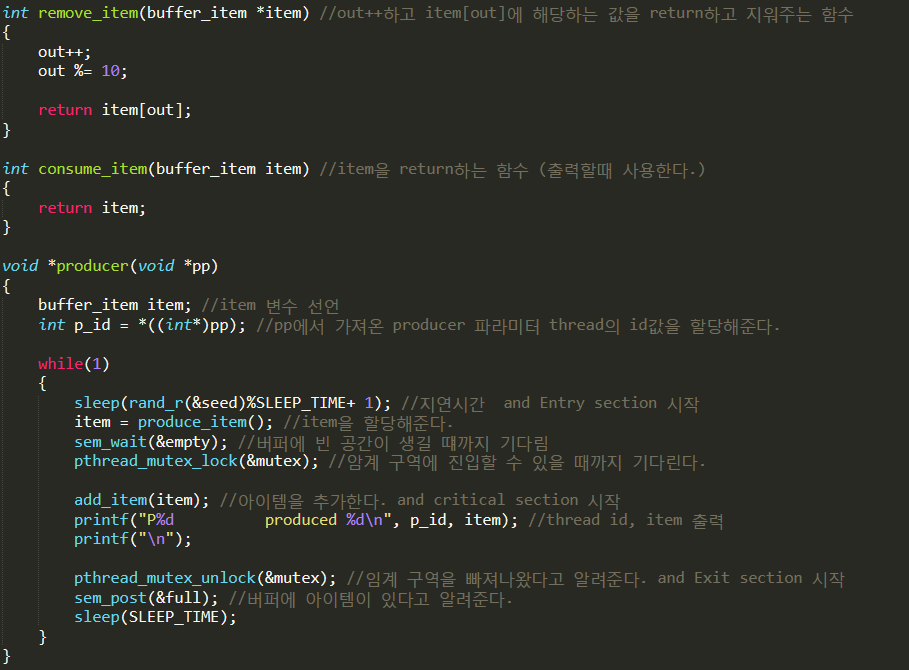
우선 기본 라이브러리 생성과 이 프로그램에서 핵심적인 역할을 하는 pthread.h, semaphore.h 라이브러리를 삽입한다.

그리고 우리의 item저장 공간인 buffer의 size를 10으로 설정해준다. 그리고 여기서 in은 item을 add할 때, out은 item을 remove할 때 사용하는 일종의 count역할과 위치역할을 해주는 변수이다. 그리고 sem\_t empty와 sem\_t full은 생산자와 소비자의 함수에서 중요한 역할을 해주는 변수이다. 뒤에서 설명하겠다.

우선 produce\_item 함수를 살펴보면 rand()의 호출을 통해 random한 item을 생성하고 return해준다.

그리고 add\_item함수는 in의 count를 1증가시켜주고 %10을 해주는 이유는 버퍼의 최대 크기가 10이기 때문에 최댓값을 넘어가지 않으려고 해주는 것이다.

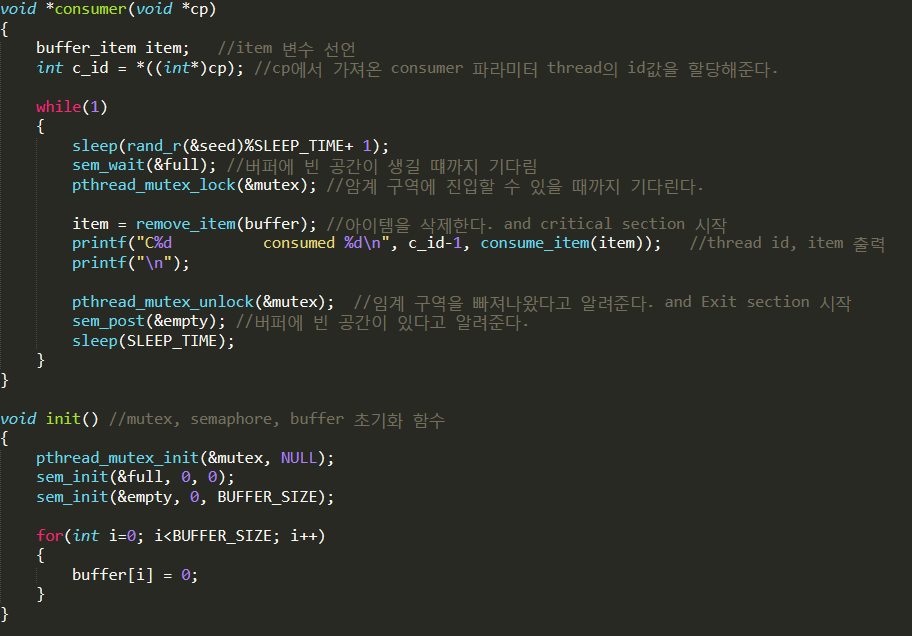
그 다음에는 buffer[in]에 해당하는 아이템을 return 해준다.



Remove\_item에서는 out의 count를 증가시켜주고, out%10을 하는 이유는 add\_item에서의 이유와 같다. 그리고 item[out]에 해당되는 값을 반환해준다.

Consume\_item에서는 우리 코드에서는 오로지 item을 retrun하기 위해 사용했다.

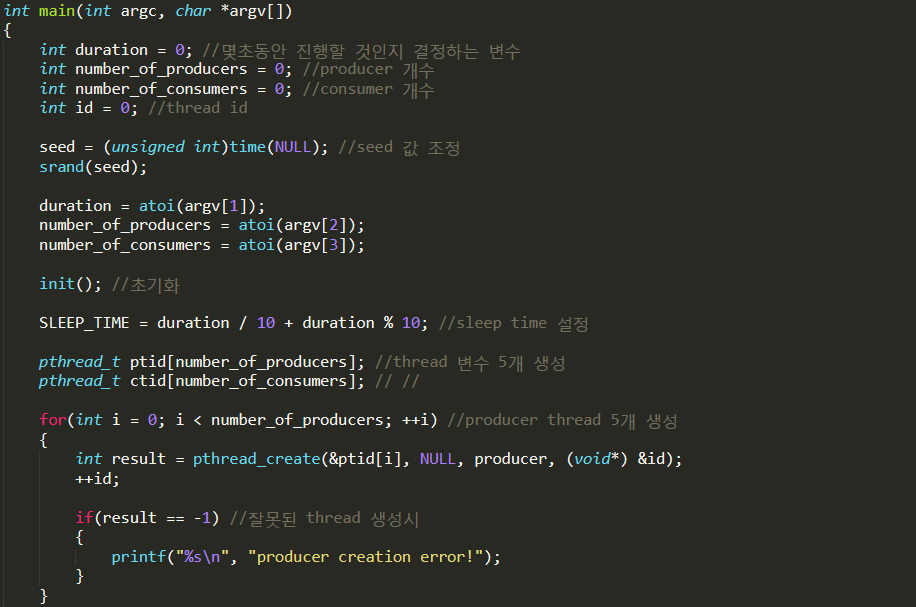
producer함수에서는 우선 아이템을 생산하기전 item변수를 선언을 해주고, p\_id를 통해 해당 thread의 id를 할당해준다. 그리고 while문을 통해 duration 시간 동안 produce를 진행한다. Sleep을 통해 빠르게 생성되는 것을 막아준다. Entry section에서는 Empty를 통해 버퍼에 빈 공간이 생길 때까지 기다린다. 일종의 체크이다. mutex를 통해 critical section 진입 전, 진입할 수 있는지 확인한다. 그리고 critical section에서는 Add\_item 함수 호출을 통해 버퍼에 item을 추가시켜주고, mutex를 통해 critical section을 빠져나왔다고 알려준다. 그리고 exit section이 시작된다. Full을 사용해 버퍼에 아이템이 있다는 것을 알려주고 produce함수를 빠져나온다.



consumer함수도 producer함수와 비슷한 원리로 작동된다. 중복되는 내용은 제외하고 while위는 producer랑 기능이 같다. Entry section의 시작인 full로 버퍼의 빈 공간이 있는 지 체크하고 있으면 진행한다. 그 다음 mutex로 producer와 똑같이 임계구역 진입을 기다린다.

그 다음 critical section에 진입하게 되고 remove\_item을 통해 아이템을 삭제한다. 그 다음 임계구역을 빠져나왔다고 알려준후 exit section이 시작된다. Empty를 통해 버퍼에 빈 공간이 있다고 알려준다.

init함수는 mutex, full empty, buffer를 초기화 해주는 함수이다.



그 다음은 main함수이다. 우선 duration은 몇 초 동안 진행할 것인지, number\_of\_producer는 producer의 계수, number\_of\_consumers는 consumer의 개수이다 id는 thread id 변수이다.

init으로 모든 것을 초기화 시켜주고 for문을 통해 producer와 consumer를 각각 생산한다.



모든 thread가 생산된 후 duration 시간이 지나면 각 thread를 종료 시켜주고, mutex, empty, full또한 해체시켜주고 프로그램은 종료하게 된다.