**2018008304 컴퓨터소프트웨어학부 박경하**

**운영 체제 HW#4**

**제출 일자 : 2020/04/06**

1. **과제 A**
2. **프로그램 설명**

텍스트이(가) 표시된 사진

자동 생성된 설명

먼저 lock 을 사용하지 않는 조건에서 result 변수의 값이 항상 같은 값을 같도록 하기 위해서 result 를 배열로 선언하여 스레드마다의 결과값을 더하는 형태로 구성하였습니다. 배열의 사이즈는 스레드의 수와 동일하게 했으며, 0으로 초기화하여 n을 더하여 결과값을 축적하는 형태입니다.

**텍스트이(가) 표시된 사진

자동 생성된 설명**

실행 시간이 감소하도록 하기 위해 멀티스레드를 이용하였으며, phtread\_t의 배열 형태로 멀티스레드를 생성하였습니다. 그리고 pthread\_create 함수를 이용하여 argument에 담은 ID로 각각의 ID 를 가진 멀티스레드를 threadID 배열 안에 생성하였습니다. 그 다음에, pthread\_join으로 스레드 종료를 기다리며 join이 되는 순간에 스레드는 모든 자원을 반납하게 됩니다. ThreadFunc 에서 number을 출력하고 result에 저장한 뒤에 result 들을 모두 더한 값을 마지막에 프린트하여 프로세스를 종료합니다.

1. **Time 결과값 설명**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

첫번째 사진은 멀티스레드를 적용하지 않은 결과이고, 두번째 사진은 멀티스레드를 적용한 결과입니다. Time 을 이용하여 나온 세 개의 시간은 각각 나타내는 의미가 다른데, 먼저 User 은 유저모드에서 프로세스가 소요한 시간이고, sys 은 커널모드에서 소요한 시간이며, real 은 프로그램을 돌리기 위해 소요한 총 시간입니다. 하지만 real = user + sys 가 아니며, 그 이유는 프로그램을 돌리는 시간 이외에 명령을 기다리는 시간이 포함되어 있기 때문입니다. 따라서 보통은 real >= user + sys 라고 할 수 있으며, 오늘의 실습 같은 멀티스레드의 경우에는 user와 sys타임을 thread를 생성한 수만큼 배로 계산하기 때문에 real 보다 더 큰 값을 갖습니다.

1. **시간 차이가 나는 이유 설명**

멀티스레드를 사용한 경우와 아닌 경우의 real time 을 비교하면 확연하게 차이나는 것을 확인할 수 있는데, 그 이유는 스레드를 여러 개 생성하게 되면 정해진 시간 동안 여러 작업을 동시에, 병렬로 처리하게 되면서 CPU의 속도를 개선할 수 있게 됩니다.

1. **과제 B**
2. **Mutex 예비레포트**

뮤텍스란 우리말로 상호 배제라고 하며, Mutual Exclusion의 줄임말이다. 예를 들어, 식당에 하나뿐인 화장실에 들어가려면 키가 있어야한다고 하면, 한 사람이 화장실에 있으면 다른 사람은 키가 없으므로 화장실에 가지 못한다. 즉, 먼저 들어가있는 사람이 화장실에서 나오고 키를 받아야만 화장실에 들어갈 수 있게 된다. 이 때, 화장실을 이용하는 사람은 프로세스 혹은 스레드이며, 화장실은 공유자원이고, 키는 공유 자원에 접근하기 위해 필요한 오브젝트이다. 이처럼 뮤텍스는, 크리티컬 섹션을 가진 스레드들의 러닝타임이 서로 겹치지 않도록 각각 단독으로 실행되기 하는 기술이라고 볼 수 있다. 즉, 뮤텍스는 스레드나 프로세스에 의해 소유될 수 있는 Key 를 기반으로 한 상호배제기법이다.