HW3 보고서

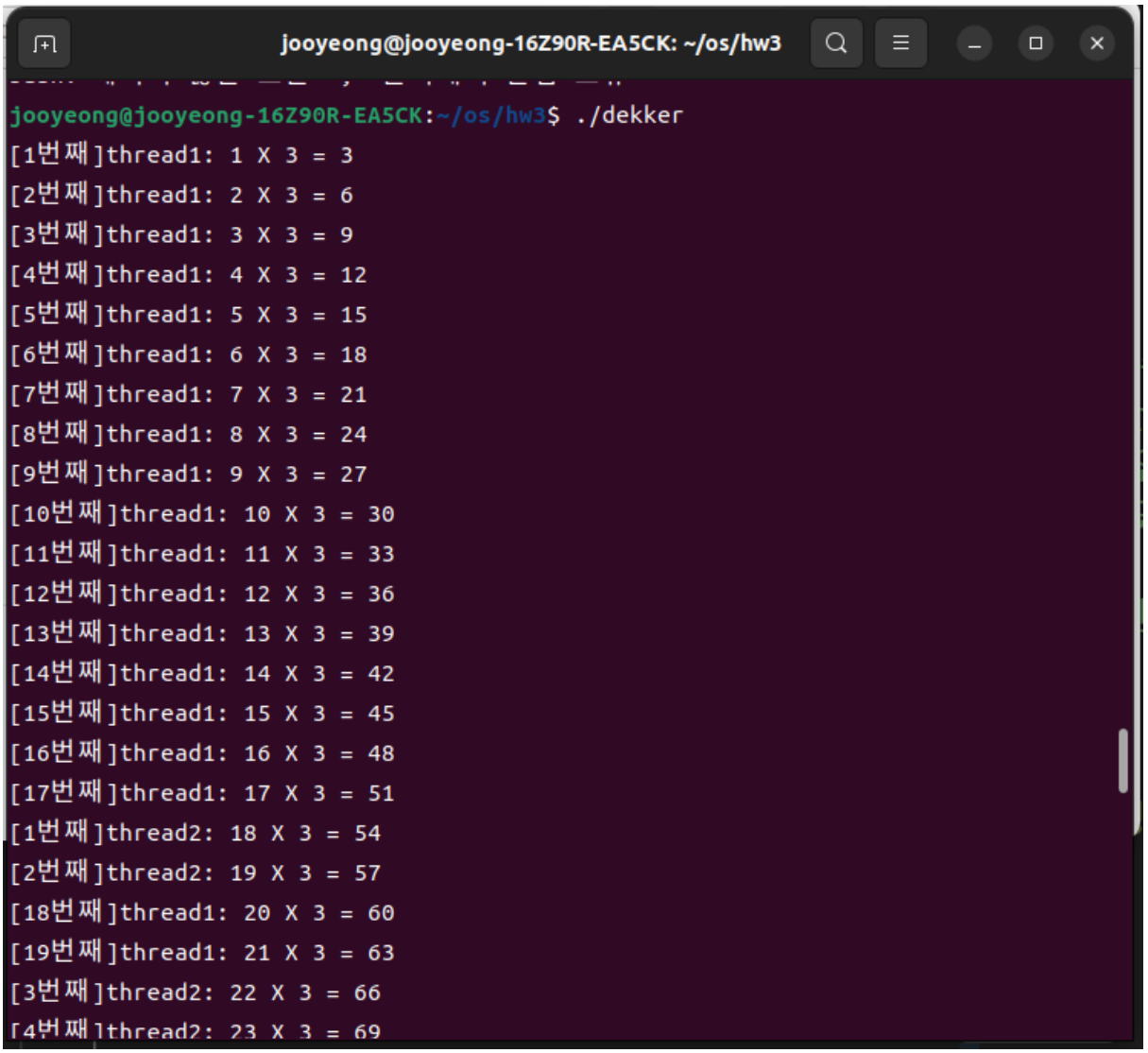
이름: 정주영

학번: 2024317

운영체제 01분반

과제 : mutual exclusion (상호배제) 구현 학습

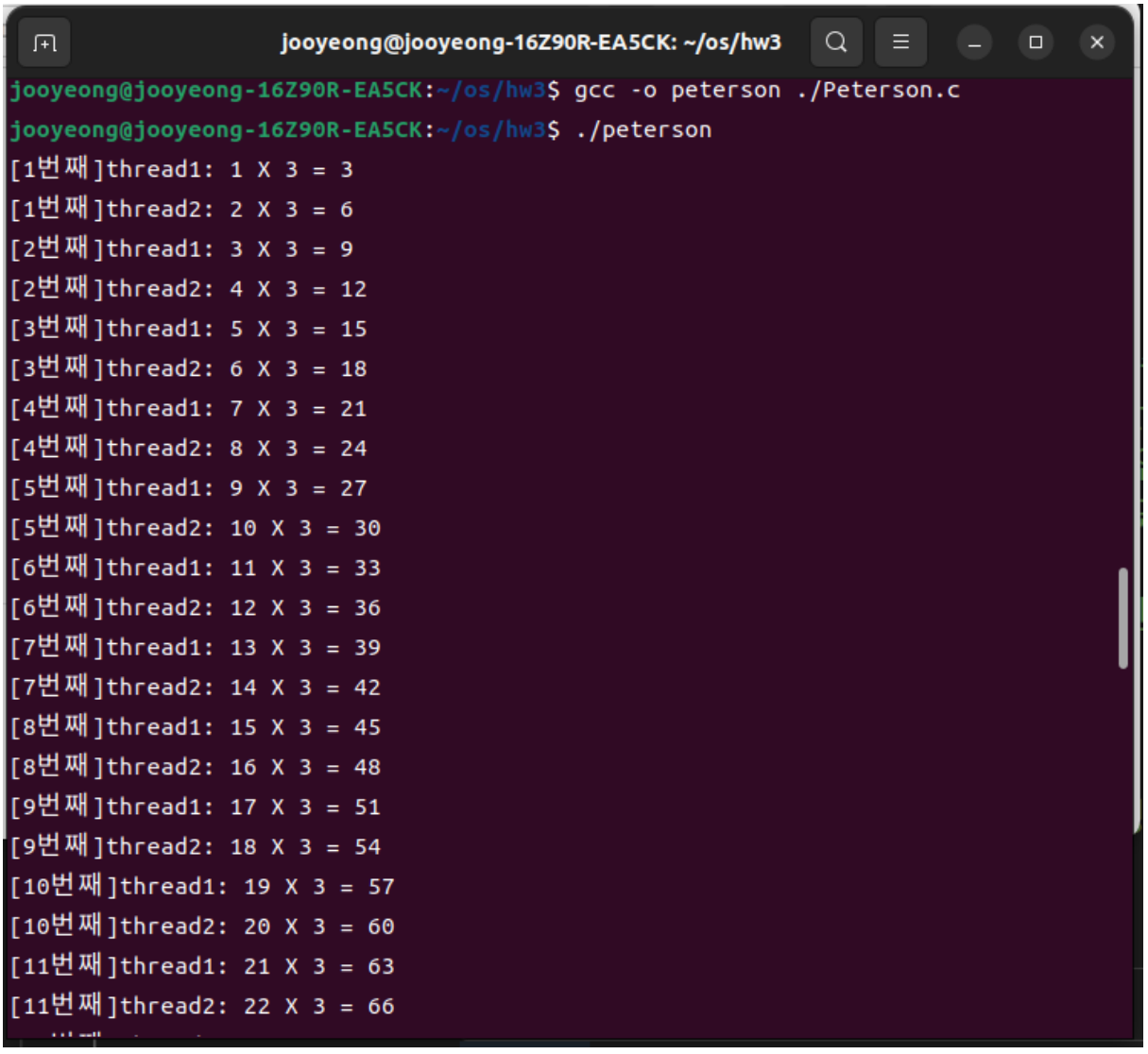
1. Dekker’s algorithm 구현



1번 쓰레드실행 함수와 2번 쓰레드 실행 함수 에서 각각 임계 영역(전역 변수)에 있는 변수(var)를 접근 하여 값을 변화 시킨다. 동일한 양의 숫자 출력을 위해 for문을 사용하여 반복을 진행했다. 매 접근 시에 상호배제를 진행 하다보니 2개의 쓰레드가 비슷한 처리 속도를 보

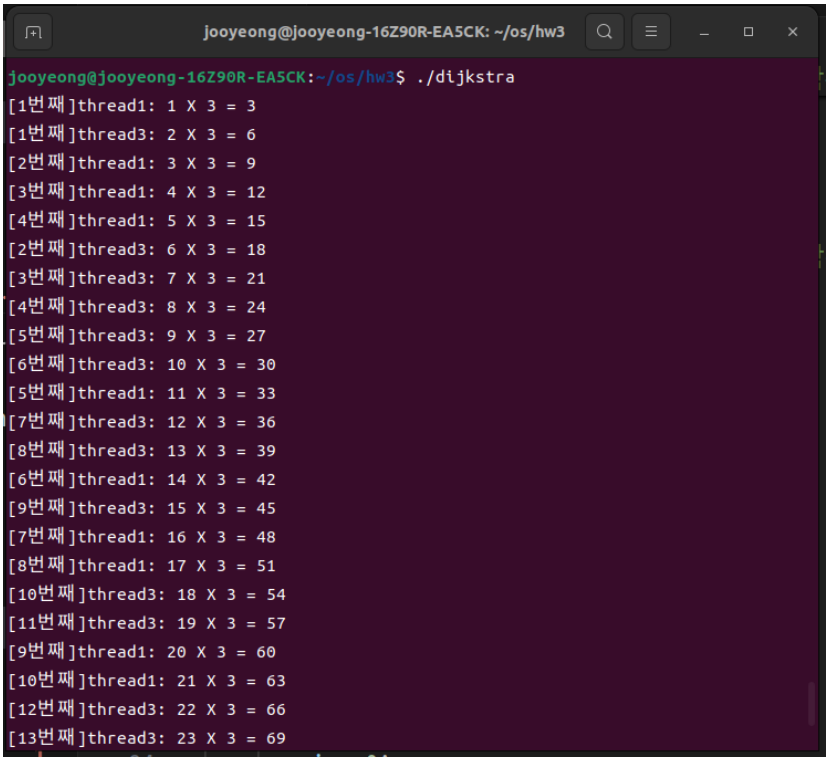
인다. 시작시에 깃발을 들어 임계 영역에 진입의사를 표하고 1번이 들고 있을 경우 turn이 1번이면 깃발을 내려 진입 순서를 양보한다.

2. Peterson's algorithm 구현



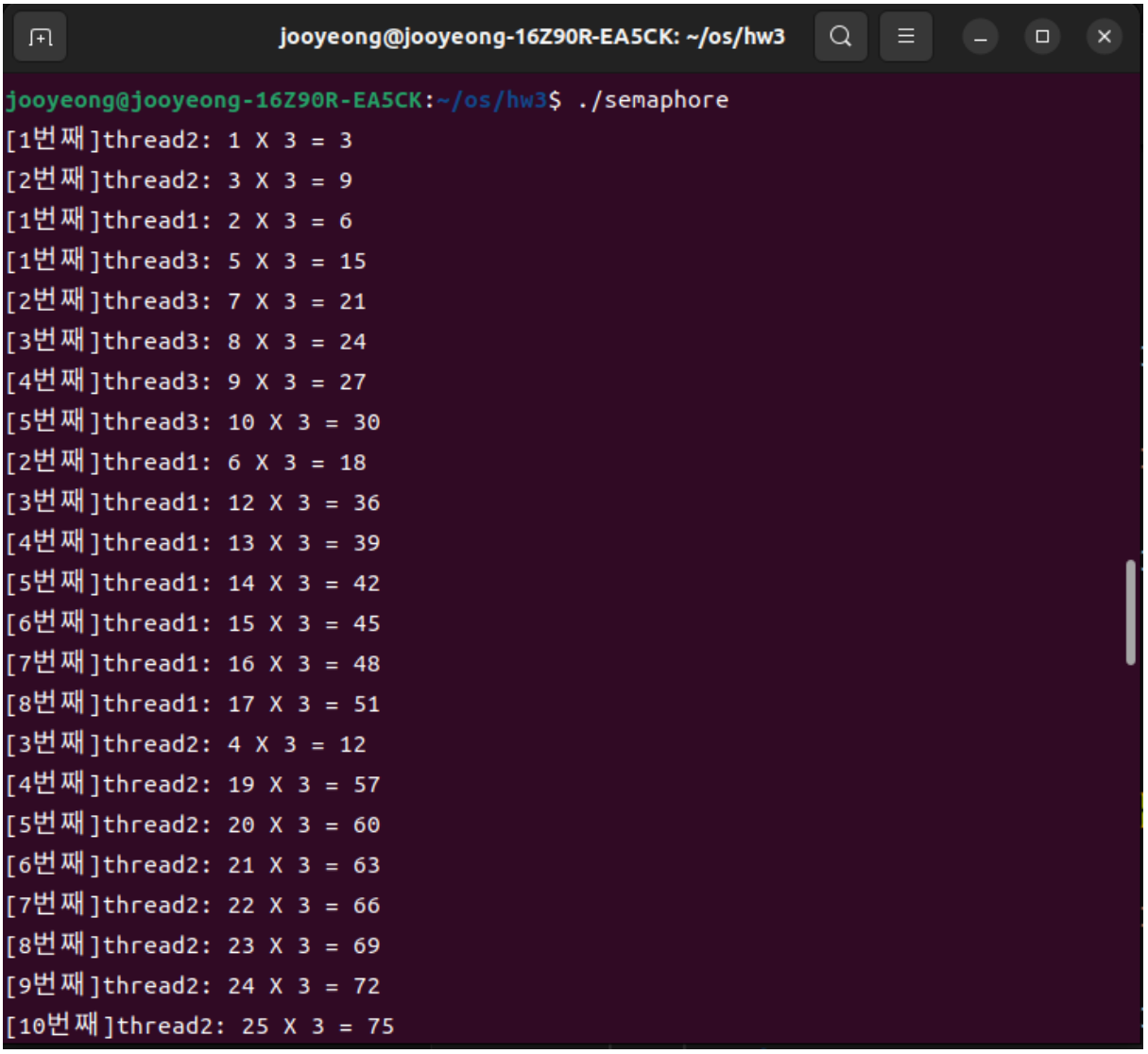
마찬가지로 1번 쓰레드실행 함수와 2번 쓰레드 실행 함수 에서 각각 임계 영역(전역 변수)에 있는 변수(var)를 접근 하여 값을 변화 시킨다. 동일한 양의 숫자 출력을 위해 for문을 사용하여 반복을 진행했다. 시작과 함께 깃발을 들어 임계 영역 진입 의사를 표하지만 turn 을 상대한테 넘겨 가장 늦게 양보한 프로세스가 진입하게 된다.

3. Dijkstra's algorithm (N process) 구현



Dijkstra 알고리즘은 N개의 프로세스의 상호배제 문제를 해결하는 알고리즘이다. 2단계의 임계 지역 진입시도를 위해 wantIn 단계와 inCS 단계를 구현했다. N 개의 프로세스를 처리하다 보니 처리 속도때문에 쓰레드가 골고루 사용되지는 못하는 것으로 보인다.

4. Semaphore 구현



semaphore(세마포) 는 semaphore.h 헤더파일을 이용하여 쉽게 구현이 가능하다. 세마포는 단일 프로세서와 다중 프로세서 모든 상황에서 사용이 가능하다. 이번 구현은 초기값을 1로 설정했기 때문에 임계영역에서 작업할 수 있는 쓰레드는 1개 뿐이다. 따라서 각 쓰레드의 처리 속도에 따라 쓰레드가 일하는 부분이 조금 뭉쳐져 보인다. 세마포는 임계 영역에 들어갈 수 있는 갯수를 num 에 지정하고 num 의 갯수가 0이 아닌 자리가 남아있으면 들어간 후 후처리를 해주는 과정으로 이루어 진다.