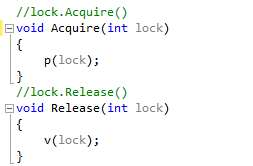
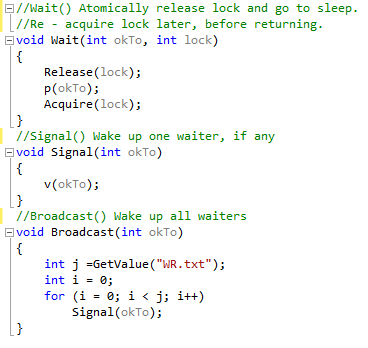
Reader & Writer Problem

20123420 컴퓨터공학부 차경준

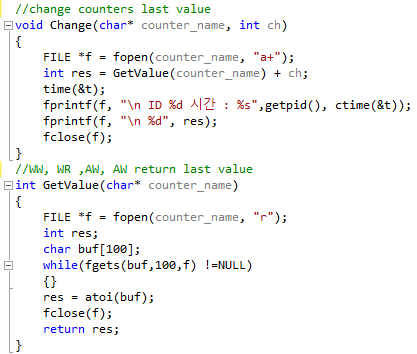
1. Lock 구현하기 // 초기값인 1인 세마포를 이용하여 구현하였다. Acquire(), Release()



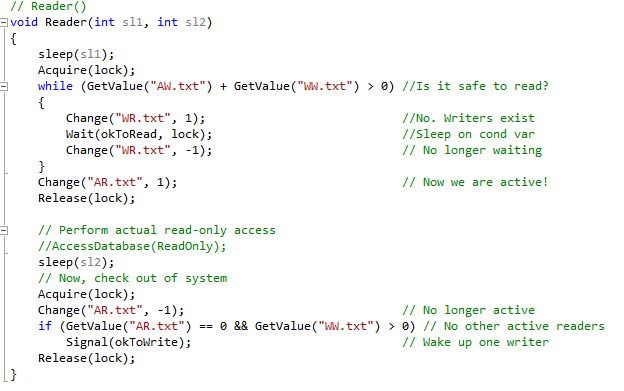
1. Condition Variable // Wait() Signal() BroadCast()

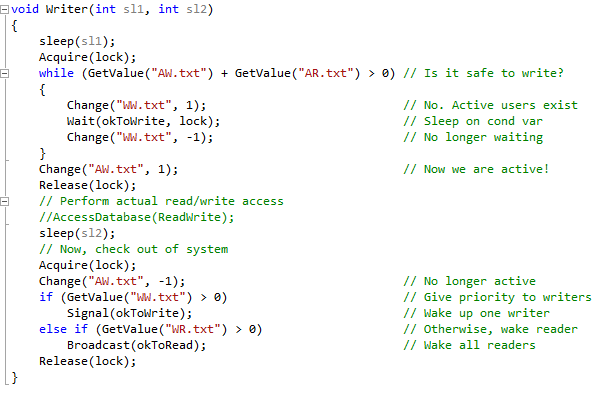


1. AW,AR,WW,WR 4개의 타운터를 파일에 기록한다. // Change(), GetValue() 함수들로 파일 에 접근한다.



1. Reader(), Writer() 구현





1. 실행 및 분석
   1. reader 1 5 &
   2. reader 2 5 &

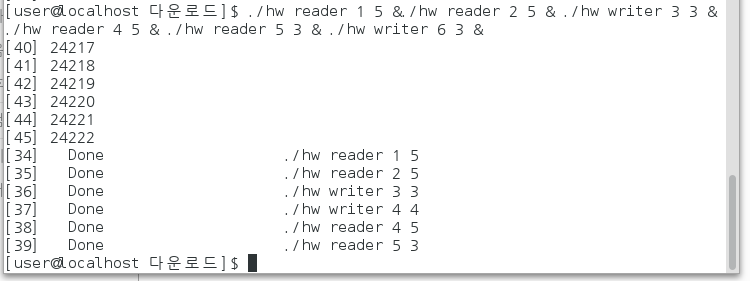
 C.  writer  3 3 &

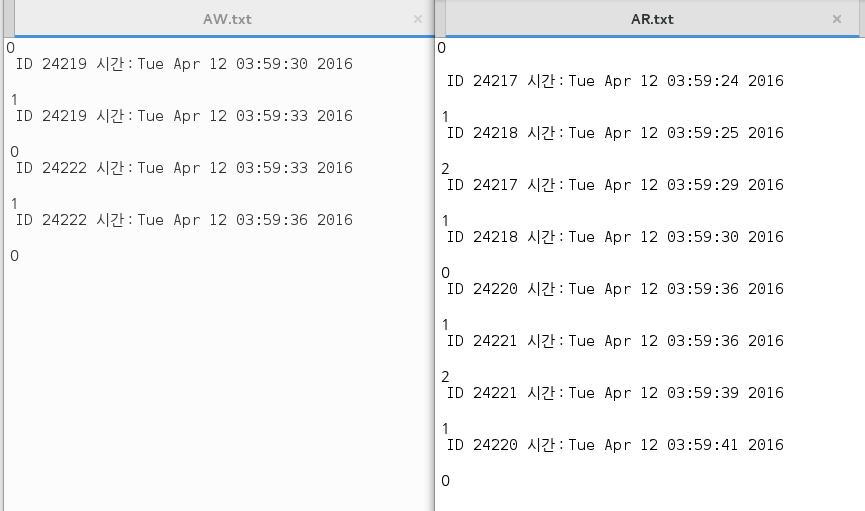
 D.  reader 4 5 &

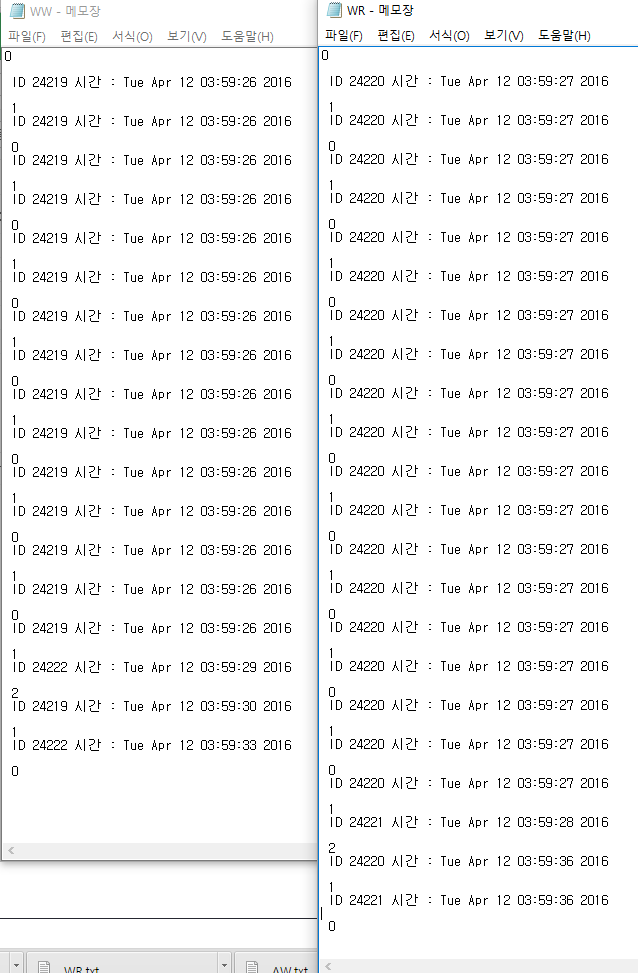
 E.  reader 5 3 &

 F.  writer  6 3 &

.[실행창]



결과값



위에 결과에서 Process ID에서 Thread를 추적할 수 있고 시간에 따른 카운터 값(AR,AW,WR,WW)의 변화를 볼 수 있다.

표의 가로의 한 칸을 ‘sleep(1)’ 이라고 할 때 카운터의 변경 값과 Active Reader/Writer의 값은 아래와 같다,

1~9

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A |  | AR=1 |  |  |  | AR=1 |  |  |  |
| B |  |  | AR=2 |  |  |  | AR=0 |  |  |
| C |  |  |  | WW=1 |  |  |  | WW=1  AW=1 |  |
| D |  |  |  |  | WR=1 |  |  |  |  |
| E |  |  |  |  |  | WR=2 |  |  |  |
| F |  |  |  |  |  |  | WW=2 |  |  |

10~18

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A |  | AR=1 |  |  |  | AR=1 |  |  |  |
| B |  |  | AR=2 |  |  |  | AR=0 |  |  |
| C | AW=0 |  |  |  |  |  |  | WW=1AW=1 |  |
| D |  |  |  |  | WR=0  AR=1 |  |  |  | AR=0 |
| E |  |  |  |  | WR=0  AR=2 |  | AR=1 |  |  |
| F |  | WW=0  AW=1 |  | AW=0 |  |  |  |  |  |

전체 실행 시간은 18초이며 가장 늦게 끝나는 것은 D reder 4 5 이다.