

동작시험 결과

수학과 202021218 김재영

1. Producer-Consumer

```
~/Desktop/3-2/운영 체제 /프로젝트 2/김재영-202021218/Producer_Consumer cd "/Users/kjy/Desktop/3-2/운영 체제 /프로젝트 2/김재영-202021218/Producer_Consumer/" && gcc PC_no_sync.c -o PC_no_sync && cd "/Users/kjy/Desktop/3-2/운영 체제 /프로젝트 2/김재영-202021218/Producer_Consumer/" && gcc PC_no_sync.c -o PC_no_sync
생산자 : Packet #0 전송
생산자 : Packet #0 전송
-----
소비자 : Packet #0 수신
-----
소비자 : Packet #0 수신
생산자 : Packet #1 전송
생산자 : Packet #1 전송
-----
소비자 : Packet #1 수신
-----
소비자 : Packet #1 수신
생산자 : Packet #2 전송
생산자 : Packet #2 전송
-----
소비자 : Packet #2 수신
-----
생산자 : Packet #3 전송
소비자 : Packet #2 수신
생산자 : Packet #3 전송
생산자 : Packet #4 전송
생산자 : Packet #4 전송
-----
소비자 : Packet #3 수신
-----
소비자 : Packet #3 수신
생산자 : Packet #5 전송
생산자 : Packet #5 전송
-----
소비자 : Packet #4 수신
-----
소비자 : Packet #5 수신
생산자 : Packet #6 전송
생산자 : Packet #6 전송
생산자 : Packet #7 전송
생산자 : Packet #7 전송
-----
소비자 : Packet #5 수신
-----
소비자 : Packet #5 수신
생산자 : Packet #8 전송
-----
소비자 : Packet #6 수신
생산자 : Packet #8 전송
-----
소비자 : Packet #6 수신
생산자 : Packet #9 전송
-----
소비자 : Packet #7 수신
생산자 : Packet #10 전송
```

```
생산자 : Packet #9 전송
-----
소비자 : Packet #7 수신
-----
소비자 : Packet #8 수신
-----
소비자 : Packet #8 수신
생산자 : Packet #11 전송
생산자 : Packet #10 전송
생산자 : Packet #12 전송
생산자 : Packet #11 전송
-----
소비자 : Packet #11 수신
-----
소비자 : Packet #10 수신
생산자 : Packet #13 전송
생산자 : Packet #12 전송
-----
소비자 : Packet #12 수신
-----
소비자 : Packet #11 수신
생산자 : Packet #14 전송
-----
소비자 : Packet #12 수신
생산자 : Packet #15 전송
생산자 : Packet #13 전송
-----
소비자 : Packet #13 수신
-----
소비자 : Packet #13 수신
생산자 : Packet #16 전송
생산자 : Packet #14 전송
-----
소비자 : Packet #14 수신
^C
```

~/Desktop/3-2/운영체제/프로젝트 2/김재영-202021218/Producer_Consumer

동기화 전 실행 결과

생산자가 Packet을 중복되게 전송하고, 소비자 또한 중복된 숫자의 Packet을 수신하는 결과가 도출되었다.

또한, race condition에 의해 종료 호출이 올바르게 되지 않아 강제로 종료해야 하는 결과가 발생하였다.

```

x ~/Desktop/3-2/운영 체제 /프로젝트 2/김재영-202021218/Producer_Consumer cd "/Users/kjy/Desktop/3-2/운영 체제 /프로젝트 2/김재
/" && gcc PC_sync.c -o PC_sync && "/Users/kjy/Desktop/3-2/운영 체제 /프로젝트 2/김재영-202021218/Producer_Consumer/"PC_sync
생산자 : Packet #0 전송
생산자 : Packet #1 전송
-----
소비자 : Packet #0 수신
-----
소비자 : Packet #1 수신
생산자 : Packet #2 전송
생산자 : Packet #3 전송
-----
소비자 : Packet #2 수신
-----
소비자 : Packet #3 수신
생산자 : Packet #4 전송
생산자 : Packet #5 전송
-----
소비자 : Packet #4 수신
-----
소비자 : Packet #5 수신
생산자 : Packet #6 전송
생산자 : Packet #7 전송
생산자 : Packet #8 전송
생산자 : Packet #9 전송
-----
소비자 : Packet #6 수신
-----
소비자 : Packet #7 수신
생산자 : Packet #10 전송
생산자 : Packet #11 전송
-----
소비자 : Packet #8 수신
생산자 : Packet #12 전송
-----
소비자 : Packet #9 수신
생산자 : Packet #13 전송
생산자 : Packet #14 전송
-----
소비자 : Packet #10 수신
생산자 : Packet #15 전송
-----
소비자 : Packet #11 수신
생산자 : Packet #16 전송
-----
소비자 : Packet #12 수신
-----
소비자 : Packet #13 수신
생산자 : Packet #17 전송
생산자 : Packet #18 전송
-----
소비자 : Packet #14 수신
-----
소비자 : Packet #15 수신
생산자 : Packet #19 전송
생산자 : Packet #20 전송

```

```

소비자 : Packet #16 수신
-----
소비자 : Packet #17 수신
생산자 : Packet #21 전송
생산자 : Packet #22 전송
-----
소비자 : Packet #18 수신
생산자 : Packet #23 전송
-----
소비자 : Packet #19 수신
생산자 : Packet #24 전송
-----
소비자 : Packet #20 수신
생산자 : Packet #25 전송
-----
소비자 : Packet #21 수신
생산자 : Packet #26 전송
-----
소비자 : Packet #22 수신
-----
소비자 : Packet #23 수신
생산자 : Packet #27 전송
생산자 : Packet #28 전송
-----
소비자 : Packet #24 수신
-----
소비자 : Packet #25 수신
생산자 : Packet #29 전송
생산자 : Packet #30 전송
-----
소비자 : Packet #26 수신
-----
소비자 : Packet #27 수신
생산자 : Packet #31 전송
생산자 : Packet #32 전송

```

동기화 후 실행결과

Semaphore의 lock에 의해 동기화가 올바르게 진행되어 중복된 Packet 넘버

가 송수신되지 않고 순차적으로 Packet 전송이 이루어진다.

2. Reader-Writer

```
~/Desktop/3-2/운영 체제 /프로젝트 2/김재영-202021218/Producer_Consumer cd "/Users/kjy/Desktop/3-2/운영 체제 /프로젝트 2/김재영-202021218/Reader_Writer/"
gcc RW_no_sync.c -o RW_no_sync && "/Users/kjy/Desktop/3-2/운영 체제 /프로젝트 2/김재영-202021218/Reader_Writer/"
Writer : 서버 데이터 추가 0

Reader 1 : 서버의 데이터 읽기 {}
Reader 3 : 서버의 데이터 읽기 {}
Reader 2 : 서버의 데이터 읽기 {}
Writer : 서버 데이터 추가 1

Reader 1 : 서버의 데이터 읽기 {0, 1}
Reader 2 : 서버의 데이터 읽기 {0, 1}
Writer : 서버 데이터 추가 1

Reader 1 : 서버의 데이터 읽기 {0, 1}
Reader 3 : 서버의 데이터 읽기 {0, 1, 1}
Writer : 서버 데이터 추가 2

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2}
Writer : 서버 데이터 추가 3

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3}
Writer : 서버 데이터 추가 5

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3}
Reader 3 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5}
Writer : 서버 데이터 추가 8

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8}
Writer : 서버 데이터 추가 13

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13}
Writer : 서버 데이터 추가 21

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21}
Reader 3 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21}
Writer : 서버 데이터 추가 34

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 3 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 3 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
```

동기화 전 실행결과

동기화가 올바르게 진행되지 않아서 Reader 실행 도중 Writer의 데이터 접근이 이루어진다. 이에 의해 Reader가 추가되기 전 데이터를 읽어서 출력한다.

```

~/Desktop/3-2/운영 체제 /프로젝트 2/김재영 -202021218/Reader_Writer cd "/Users/kjy/Desktop/3-2/운영 체제 /프로젝트 2/김재영 -2020
RW_sync.c -o RW_sync && "/Users/kjy/Desktop/3-2/운영 체제 /프로젝트 2/김재영 -202021218/Reader_Writer/"RW_sync
Writer : 서버 데이터 추가 0

Reader 1 : 서버의 데이터 읽기 {0}
Reader 2 : 서버의 데이터 읽기 {0}
Reader 3 : 서버의 데이터 읽기 {0}
Reader 1 : 서버의 데이터 읽기 {0}
Writer : 서버 데이터 추가 1

Writer : 서버 데이터 추가 1

Reader 2 : 서버의 데이터 읽기 {0, 1, 1}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1}
Reader 3 : 서버의 데이터 읽기 {0, 1, 1}
Writer : 서버 데이터 추가 2

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2}
Writer : 서버 데이터 추가 3

Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3}
Writer : 서버 데이터 추가 5

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5}
Reader 3 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5}
Writer : 서버 데이터 추가 8

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8}
Writer : 서버 데이터 추가 13

Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13}
Writer : 서버 데이터 추가 21

Reader 3 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21}
Writer : 서버 데이터 추가 34

Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 3 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 2 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 3 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Reader 1 : 서버의 데이터 읽기 {0, 1, 1, 2, 3, 5, 8, 13, 21, 34}

```

동기화 후 실행결과

동기화가 잘 적용되어 Reader의 실행 이후 Writer의 데이터 추가가 이루어져서 Reader의 실행 결과가 뒤섞이지 않는다.

3. 자가진단표

0. 프로그램 모델

모델 단계 (단계에 체크)	내용
10 <input checked="" type="checkbox"/>	Producer/Consumer와 Reader/Writer 모델을 기반으로 한 창의적인 프로그램
9 <input type="checkbox"/>	Producer/Consumer와 Reader/Writer 모델을 기반으로 응용 프로그램 제작
8 <input type="checkbox"/>	교재의 Producer/Consumer와 Reader/Writer 모델의 수정
7 <input type="checkbox"/>	교재의 Producer/Consumer와 Reader/Writer 모델의 최소한의 수정
6 <input type="checkbox"/>	교재의 Producer/Consumer와 Reader/Writer 모델을 그대로 구현

Producer-Consumer 모델을 기반으로 네트워크 송수신 패킷 프로그램을 제작하고, Reader-Writer 모델을 기반으로 피보나치 수열을 저장하는 데이터베이스 응용 프로그램을 제작하였다. 나름 배운 수준을 응용하여서 창의적인 프로그램을 제작하였다고 생각한다.

1. Producer/Consumer

구현 단계 (단계에 체크)	내용	주요 bug 및 프로그램의 문제점
10 <input checked="" type="checkbox"/>	동기화 기능 구현하여 문제점(race condition) 해결 확인	
9 <input type="checkbox"/>	동기화 기능 기본 동작 확인	
8 <input type="checkbox"/>	동기화 기능 구현	
7 <input type="checkbox"/>	동기화 제외한 프로그램의 문제점(race condition) 확인	
6 <input type="checkbox"/>	동기화를 제외한 프로그램 동작 확인	
5 <input type="checkbox"/>	동기화를 제외한 프로그램 일부 동작	
4 <input type="checkbox"/>	동기화를 제외한 프로그램 완성, 동작 불가	

생성 프로그램에 대한 동기화 구현 부분을 확인하여 네트워크 패킷 송수신 간의 race condition을 체크하고 해결하였다.

2. Reader/Writer

구현 단계 (단계에 체크)	내용	주요 bug 및 프로그램의 문제점
10 <input checked="" type="checkbox"/>	동기화 기능 구현하여 문제점(race condition) 해결 확인	
9 <input type="checkbox"/>	동기화 기능 기본 동작 확인	
8 <input type="checkbox"/>	동기화 기능 구현	
7 <input type="checkbox"/>	동기화 제외한 프로그램의 문제점(race condition) 확인	
6 <input type="checkbox"/>	동기화를 제외한 프로그램 동작 확인	
5 <input type="checkbox"/>	동기화를 제외한 프로그램 일부 동작	
4 <input type="checkbox"/>	동기화를 제외한 프로그램 완성, 동작 불가	

생성 프로그램에 대한 동기화 구현 부분을 확인하여 Reader-Writer 간의 피보나치 수열 데이터 추가 관련 race condition을 체크하고 해결하였다.

4. 느낀점

동기화가 제대로 이루어지지 않으면 제작한 프로그램이 원하는 대로 진행되지 않음을 실습을 통해 체감하게 되었다. 특히, Producer-Consumer 문제에서 통신 관련 프로그램처럼 네트워크 패킷 송수신 코드를 작성해보았는데 실제 상황에서 동기화가 제대로 이루어지지 않는다면 통신에 오류가 생겨 빅데이터 세상에서 큰 불편함을 느낄 것이라고 생각되었다. Reader-Writer 문제에서는 여러 명의 Reader-Writer를 보유하고 있는 데이터베이스 서버에서 공유 데이터의 접근이 동시에 이루어진다면 공유된 데이터의 손상이 야기되는 끔찍한 모습을 볼 수 있을 거라 생각된다. 따라서, 실생활에 크게 이용되는 프로그램을 직접 간단하게 구현해봄으로써 동기화 문제를 잘 해결해내는 것이 중요함을 알게 되었다.