|  |
| --- |
| **[514126] 컴퓨터네트워크** |
| **실습 #10 문제 및 보고서** |

|  |  |
| --- | --- |
| **이름** | 최영빈 |
| **학번** | 20175260 |
| **소속**  **학과/대학** | 전자공학과 |
| **분반** | 02 (담당교수: 박찬영) |

## <주의사항>

* 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
* 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.
  + 소스코드/스크립트 등을 작성한 경우, 해당 파일의 이름도 적어주세요.
* SmartLEAD 제출 데드라인:
  + 다음 다음 실습시간 전날 23:55까지 (2주간 진행하는 과제 입니다)
  + 데드라인을 지나서 제출하면 0점
  + 주말/휴일/학교행사 등으로 인한 데드라인 연장 없음
  + 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0점 처리함
* SmartLEAD 에 아래의 파일을 제출해 주세요
  + 보고서(PDF 파일로 변환 후 제출을 권장하나, WORD로 제출해도 됨)
  + 보고서 파일명에 이름과 학번을 입력해 주세요.
  + 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출(또는 본 문서에 소스코드 화면 캡처해서 붙여넣기)

## <개요>

이번 과제는 소켓 프로그래밍을 통한 통신 프로그램을 구현하는 내용으로 구성되어 있습니다.

\*\* 전체 소스코드를 압축하여 첨부파일로 제출하세요.

## <실습 과제>

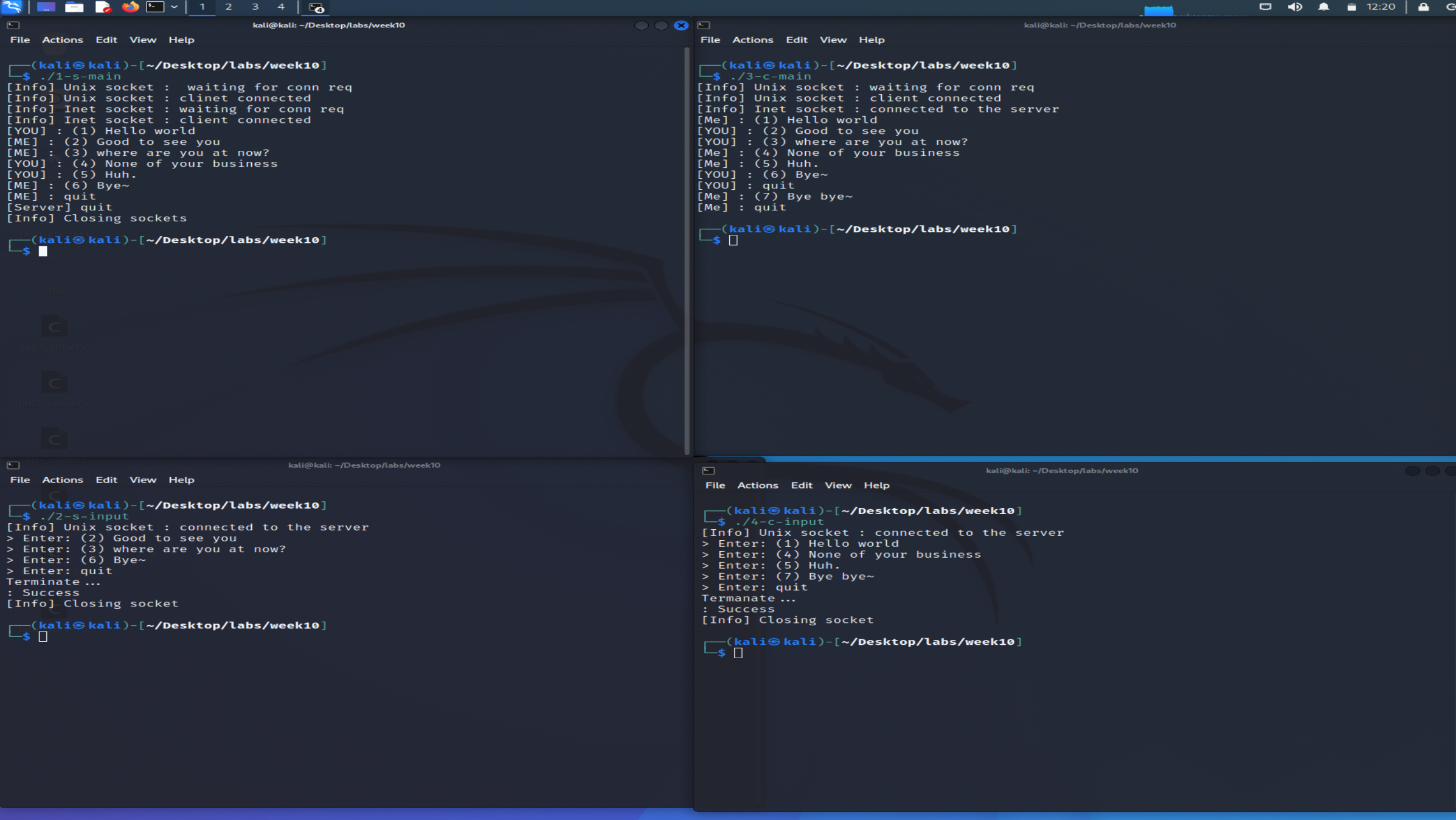
|  |
| --- |
| **[Q 0] 요약 [배점: 10]**  이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3문장으로 요약하세요. |

답변:

저번 실습에 배운 Unix, Inet 통신을 활용해 1:1 비동기 통신을 만들 수 있었다.

추가로 논블로킹을 할 수 있도록 fcntl을 이용해 파일 속성을 변경했다.

|  |
| --- |
| **[Q 1] 비동기형 1:1 채팅 프로그램 만들기 [배점: 90]**  (과제에 대한 자세한 설명은 강의자료 [참고자료]를 확인하세요)  1:1 채팅을 비동기 방식으로 구현하는 문제입니다. 동기형 1:1 채팅 프로그램에서는 A와 B 호스트가 채팅을 할 때, A 송신 >> B 수신 >> B 송신 >> A 수신 >> A 송신...의 방식으로, 즉 순차적으로 송신과 수신을 반복했습니다. 하지만, 비동기 방식의 1:1 채팅에서는 A와 B 호스트 모두, 언제든, 메시지를 주고 받을 수 있습니다. 강의자료 [참고자료]를 참고하여 4개의 소스코드로 구성된 비동기형 1:1 채팅 프로그램을 작성하세요. 그리고 아래와 같이 프로그램을 실행하고, 아래와 동일한 순서로, 동일한 메시지를 입력하고 그 화면을 아래와 같이 캡처하여 첨부하세요.  어떤 순서로 메시지를 입력했는지를 알려주기 위해, 각 채팅 메시지에 (1), (2), (3)... 과 같은 번호를 붙였습니다. 즉, (1), (2), (3), ... 등은 자동으로 붙는 번호가 아니고, 사용자가 직접 입력한 메시지 입니다. 예를 들어, 우측 하단의 4-c-input 프로그램 실행 화면에서 사용자는 (1) Hello world 라고 입력했습니다. 즉, (1)까지 직접 입력한 것입니다.  사용자가 quit 이라고 입력하면, 해당 사용자는 더 이상 상대편에게 메시지를 전송할 수 없습니다. 여기서, quit은 채팅 종료를 의미하는 제어 메시지 역할을 하지만, 동시에 채팅메시지로써 모두에게 전달도 됩니다. 두명의 사용자가 모두 quit을 입력하면 모든 프로그램이 종료됩니다. 아래의 예시 화면에서는, 왼쪽 하단 2-s-input 프로그램 터미널에서 사용자가 quit 이라고 입력했고, 이로 인해 해당 사용자의 터미널 및 왼쪽 상단 1-s-main 프로그램이 모두 종료했습니다. 하지만, 아직 우측의 두 프로그램은 종료하지 않았습니다. 다음으로, 우측 하단 4-c-input 프로그램 터미널에서 사용자가 quit 이라고 입력했고, 이로 인해 해당 프로그램 및 우측 상단의 3-c-main 프로그램이 모두 종료되었습니다.  그림 1  문제) 위와 같이 4개의 프로그램을 순서에 맞게(즉, 1-s-main => 2-s-input => 3-c-main => 4-c-input 순서로) 실행하고, 위와 같이 채팅 메시지를 입력하세요. 메시지 입력 순서는 아래와 같습니다.   1. [4-s-input] (1) Hello world 2. [2-s-input] (2) Good to see you 3. [2-s-input] (3) Where are you at now? 4. [4-s-input] (4) None of your business 5. [4-s-input] (5) Huh 6. [2-s-input] (6) Bye~ 7. [2-s-input] quit 8. [4-s-input] (7) Bye bye~ 9. [4-s-input] quit   4개의 터미널 화면을 위와 같이 동시에 캡처하여 아래에 첨부하고, 소스코드도 제출하세요. |

답변)

**끝! 수고하셨습니다** ☺