|  |
| --- |
| [506489] 시스템프로그래밍 |
| **실습 #12 문제 및 보고서** |

|  |  |
| --- | --- |
| **이름** | 곽영주 |
| **학번** | 20175105 |
| **소속**  **학과/대학** | 빅데이터 |
| **분반** | 01 (담당교수: 김태운) |

## <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 작성 한 경우, 해당 파일의 이름도 적어주세요.
* SmartLEAD 제출 데드라인:
  + 월요일 분반: 다음 실습 시간 전날까지(일요일 까지)
  + 수요일 분반: 다음 실습 시간 전날까지 (화요일 까지)
  + 데드라인을 지나서 제출하면 24시간 단위로 20%씩 감점(5일 경과 시 0점)
  + 주말/휴일/학교행사 등 모든 날짜 카운트 함
  + 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0점 처리함
  + 예외 없음
* SmartLEAD 에 아래의 파일을 제출해 주세요
  + 보고서(PDF 파일로 변환 후 제출 권장하나, WORD 형식으로 제출도 가능)
  + 보고서 파일명에 이름과 학번을 입력해 주세요.
  + 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (zip 파일로 압축하여 하나의 파일로 제출)

## <개요>

이번 과제는 프로세스간 상호 통신(IPC, Inter-Process Communication)에 대한 내용입니다.

## <실습 과제>

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

답변: 직접 실습을 해보면서 파이프, 메시지 큐, 공유 메모리에 대해 개념을 잡을 수 있었습니다. 또한, 다운로드와 채팅 시스템을 간단하게 만들어보면서 스레드와 소켓을 같이 사용해보면서 복습을 할 수 있었습니다.

|  |
| --- |
| **[Q 1] PIPE(=이름없는 파이프) [20점]**  ‘pipe’ 함수를 사용해서 $ ls / | grep s 의 동작을 동일하게 구현하는 프로그램을 작성하세요. 최종 결과는 터미널에 출력하세요.  [Task 1] 구현한 코드를 실행하고, 터미널 출력 결과를 캡처해서 본 문서에 첨부하고, 소스코드도 제출하세요.  [Task 2] $ ls / | grep s 명령을 터미널에서 실행하고, 터미널 출력 결과를 캡처해서 본 문서에 첨부하세요. |

답변 1:

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 2:

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 2] FIFO(=이름있는 파이프) [20점]**  FIFO를 사용해서 클라이언트 프로세스가 서버 프로세스로부터 파일을 다운로드하는 프로그램을 작성하세요. 아래와 같이 동작합니다.   1. 클라이언트 터미널에서 사용자가 <GET> hello.txt 라고 입력 2. 서버는 <GET> hello.txt 라는 요청 메시지를 받으면:    1. hello.txt 파일이 현재 디렉토리에 있으면 해당 파일을 클라이언트에게 전송    2. 만약 파일이 없으면 오류 메시지(또는 오류 코드)를 전송 3. 클라이언트는    1. hello.txt 파일을 수신한 경우, 파일의 내용을 화면(터미널)에 출력하고 파일을 현재 디렉토리에 download\_hello.txt 라고 저장.    2. 오류 메시지(또는 오류 코드)를 받으면 “File does not exist”라는 메시지를 터미널에 출력 4. 클라이언트 터미널에서 사용자가 <EXIT> 이라고 입력하면 해당 메시지를 서버에 전송하고 클라이언트 프로그램 종료 5. 서버는 <EXIT> 명령어를 수신하면 서버 프로그램 종료   (서버와 클라이언트 프로그램을 동일한 폴더 내에서 실행하세요.)  [Task 1] hello.txt 파일을 만들고, 파일 안에 “Hello World!” 라는 텍스트를 입력하세요. 클라이언트에서 hello.txt 파일을 다운받으세요. 터미널 출력 결과를 캡처해서 본 문서에 첨부하세요.  [Task 2] 클라이언트에서, 서버에 존재하지 않는 nosuchfile.txt 파일을 다운받으려고 시도하세요. 터미널 출력 결과를 캡처해서 본 문서에 첨부하세요.  소스코드도 제출하세요. |

답변 [Task 1] :

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 [Task 2] :

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 3] Message Q [20점]**  메시지 큐를 사용해서 1:1 채팅 프로그램을 만드세요. 채팅 프로그램은 비동기 채팅을 지원해야 합니다. 즉, A => B => A => B => … 순서로만 채팅을 하는 것이 아니라, 사용자는 언제든 메시지를 전송할 수 있고 전송한 메시지는 즉시 상대방에게 전송되어야 합니다. 내가 입력한 메시지와 상대방이 입력한 메시지가 시간 순서에 맞게 터미널에 출력 되어야 합니다.  [Task] 1:1 채팅을 시작하고, 채팅을 하는 사용자 A와 B가 다음의 순서대로 메시지를 전송하세요. A => B => B => A => A => A.  터미널 출력 결과를 캡처해서 본 문서에 첨부하세요. 소스코드도 제출하세요. |

답변:

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 4] Shared Memory + Semaphore [30점]**  아래와 같이 동작하는 프로그램을 작성하세요. 서버-클라이언트 구조로 동작합니다. 클라이언트 아래의 2가지 동작을 할 수 있습니다.   1. 채팅: 서버와 클라이언트는 공유 메모리를 통해 채팅을 할 수 있습니다. 2. 파일 다운로드: 클라이언트는 공유 메모리를 통해 서버로부터 파일을 다운받을 수 있습니다.   공유 메모리 크기를 128바이트로 설정하도록 프로그램을 작성하세요.  [Task 1] 클라이언트가 서버와 채팅을 합니다. 각자 최소한 한번씩 메시지를 입력하세요. 본인의 메시지 뿐 아니라 상대방의 메시지가 모두 터미널에 출력되야 합니다.  [Task 2] 클라이언트가 hello.file 이라는 파일을 다운받습니다. 단, hello.file 파일의 크기는 128바이트를 초과하는 크기여야 합니다.  터미널 출력 결과를 캡처해서 본 문서에 첨부하세요. 소스코드도 제출하세요. |

답변 [Task 1]:

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 [Task 2]:

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**끝! 수고하셨습니다 ☺**