

OS 실습 2 assignment 2

컴퓨터학과
2017320208
이연준

1. client에서 추가한 코드 설명

TODO 1에서의 while문은 receive, send 파이프가 존재하는 지 확인하는 부분으로 while문에서의 조건이 true이면 빠져나가지 않고 계속 기다리게 만든다.

그 뒤의 if 문에서는 client에서의 receive 파이프와 send파이프를 각각 읽고 쓰는 기능을 사용하게 open해주는 부분이다. 만약 open이 제대로 되지 않았을 경우에는 return -1;을 통해 비정상 종료를 시켜준다.

TODO 2에서는 write, sleep, read 세 부분으로 나뉘지는데 첫 번째 if문에서는 client에서 int값을 send 파이프를 통해 메시지로 보내는 부분이고 sleep은 보내고 난 뒤에 server에서 값을 계산하고 다시 보낼 때까지 시간이 걸리므로 기다리게 하는 부분이다. 마지막 if문에서는 server에서 보낸 결과 값을 receive 파이프를 통해 읽는 부분이다. 여기서도 마찬가지로 if문을 통해 제대로 보내지지 않거나 제대로 받아서 읽지 못한 경우에는 return -1;을 통해 비정상 종료를 시켜준다.

2. server에서 추가한 코드 설명

TODO 3에서 첫 번째 if문으로 NP_RECEIVE라는 이름을 가진 파이프가 이미 존재하고 있으면 삭제시키고 두 번째 if문으로 NP_RECEIVE라는 이름의 파이프를 다시 만들고 제대로 만들어지지 않는다면 return -1;을 통해 비정상 종료를 시켜준다. 그 후, NP_SEND에 대해서도 같은 작업을 해준다.

그 후 마지막 2개의 if문을 통해 NP_RECEIVE 파이프는 receive 파이프로서 읽는 작업만을 하도록 open시켜주고 NP_SEND 파이프는 send 파이프로서 쓰는(보내는) 작업만을 하도록 open시켜준다.

TODO 4에서는 read를 통해 client에서 온 메시지가 있으면 읽는 작업을 하고 없다면 return -1;을 통해 종료시켜주는 작업을 한다.

TODO 5에서는 server에서 계산한 값을 client로 write를 통해 메시지를 보낸다. 제대로 보내주지 않는다면 return -1;을 통해 종료 시켜주는 역할을 한다.

```
jot@ubuntu:~/Desktop/lpc_project_master/lpc_project_master/assignment/assignment
t_2$ chmod 777 run.sh
jot@ubuntu:~/Desktop/lpc_project_master/lpc_project_master/assignment/assignment
t_2$ ./run.sh
jot@ubuntu:~/Desktop/lpc_project_master/lpc_project_master/assignment/assignment
t_2$ client : send 12
server : receive 12
server : send 144
client : receive 144

client : send 13
server : receive 13
server : send 169
client : receive 169

client : send 14
server : receive 14
server : send 196
client : receive 196

client : send 15
server : receive 15
server : send 225
client : receive 225
```

<- 실행 결과