

Homework #2

School ID: 201824607

Name: 편경찬

1. Submit your program source / makefile with detailed description (comments)

(1) csechatshm.h

해당 헤더파일에는 유저의 정보를 담는 구조체를 선언하고 유저의 아이디, 메시지 시간, 메시지 내용과 read_flag를 두어 유저의 입력을 읽어야 하는지 판단 할 수 있는 변수를 두었습니다.

(2) csechatwrite.c

해당 소스파일에서는 'shmget'과 'shmat' 등을 이용해 공유메모리 영역을 생성하며 구조체에 유저의 정보와 입력된 메시지들을 저장하도록 했습니다. 입력은 40자 까지 받을 수 있도록 했습니다. 이 과정들은 while(1)을 통해 항상 실행되고 있기에 입력이 있을 때만 chatinfo->read_flag = 1로 설정해 차후 csechatread.c에서 해당 flag를 보고 중복되지 않은 메시지임을 알 수 있도록 했습니다.

(3) csechatread.c

해당 파일에서는 공유메모리 영역을 생성하며 ncurses 기능을 이용해 화면을 분할 했습니다. 서브 윈도우 4개를 2x2 배열로 생성했으며 subwin1에는 메시지 출력, subwin2에는 로그인 된 유저목록, subwin3에는 입력된 메시지 내용, subwin4에는 채팅 프로그램의 이름을 화면에 보이도록 했습니다. 'box'와 'mvprintw'를 이용해 화면간 테두리를 설정하고 출력될 정확한 위치를 조정했습니다.

서브 윈도우 1에서는 read_flag가 1이라면 메시지를 읽고 메시지를 담는 포인터 배열인 'messageArray'를 이용해 메시지들을 출력했습니다. 이때 메시지 개수가 10개를 초과하면 오래된 메시지를 제거하고 새로운 메시지가 보이도록 배열간 순서를 조정했습니다. 메시지를 출력할 때 snprintf를 통해 적절한 포맷을 유지하도록 했습니다. 메시지를 읽은 후에는 read_flag를 0으로 설정해 차후 또 다른 입력이 들어올 수 있도록 만들었습니다.

서브 윈도우 2에서는 로그인된 유저들의 목록을 저장하는 'usrlist' 포인터 배열을 만들고 전체 유저수가 3명을 넘지 않게 유지하면서 신규 유저가 채팅에 입장하면 목록에 추가했습니다. strcmp를 통해 공유메모리의 userID가 유저리스트 목록에 있는지 비교하며 중복되지 않도록 했습니다.

서브 윈도우 3에서는 input의 의의에 맞게 최근에 입력된 메시지 내용을 화면에 출력되도록 했습니다. 하나의 inputline만 허용하기에 매번 'wclear(subwin3)'을 통해 서브 윈도우의 내용을 지운 후 필요한 내용을 출력하고 'wrefresh' 시켰습니다.

서브 윈도우 4에서는 채팅 프로그램 이름 'unlucky kakaotalk'이 (5, 5) 위치에 출력되도록 했습니다.

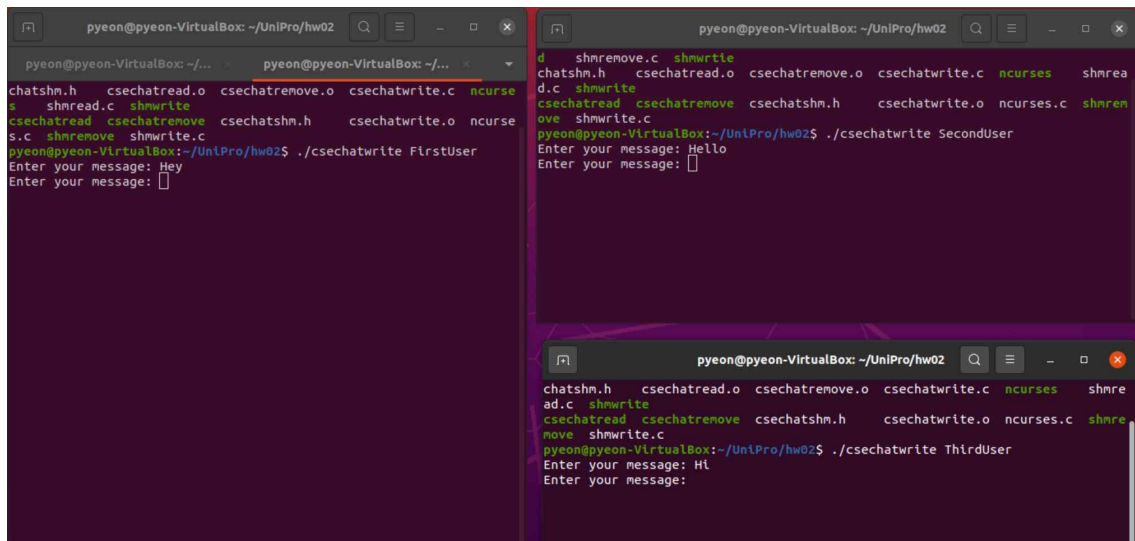
(4) csechatremove.c

'if (shmctl(shmid, IPC_RMID, 0) < 0)' 조건을 통해 공유 메모리가 삭제되도록 만들었습니다.

(5) Makefile

LIBS = -lnurses를 통해 ncurses 라이브러리를 사용할 수 있도록 했고 target은 'csechatread', 'csechatwrite', 'csechatremove'로 설정했습니다. Dependency는 'csechatshm.h'로 설정했습니다. 'make clean' 명령어를 통해 생성된 실행파일들을 지울 수 있도록 clean rule도 추가했습니다.

2. Put a screen shot of output generated by your program as here.



The image shows three terminal windows from a VirtualBox environment. The top-left window shows the compilation of the chat program using gcc and the creation of the chatshm.h header file. The top-right window shows the execution of the program with two users, 'FirstUser' and 'SecondUser', sending messages. The bottom window shows the execution of the program with three users, 'FirstUser', 'SecondUser', and 'ThirdUser', sending messages.

```
pyeon@pyeon-VirtualBox: ~/UniPro/hw02
pyeon@pyeon-VirtualBox: ~/UniPro/hw02
chatshm.h csechatread.o csechatremove.o csechatwrite.o ncurses
s shmread.c shmwrite
csechatread csechatremove csechatshm.h csechatwrite.o ncurses.c shmread.c shmwrite
s.c shmremove shmwrite.c
pyeon@pyeon-VirtualBox:~/UniPro/hw02$ ./csechatwrite FirstUser
Enter your message: Hey
Enter your message:

pyeon@pyeon-VirtualBox:~/UniPro/hw02$ ./csechatwrite SecondUser
Enter your message: Hello
Enter your message:

pyeon@pyeon-VirtualBox:~/UniPro/hw02$ ./csechatwrite ThirdUser
Enter your message: Hi
Enter your message:
```

3명의 유저('FirstUser', 'SecondUser', 'ThirdUser')가 각각 메시지를 입력한 모습입니다.



The image shows a screenshot of the chat application interface. It is divided into four panels: 'Chatting..' (top-left), 'Logged in Users' (top-right), 'input line' (bottom-left), and 'Chat Info' (bottom-right). The 'Chatting..' panel shows the chat history: [FirstUser]1: Hey, [SecondUser]2: Hello, [ThirdUser]3: Hi. The 'Logged in Users' panel shows the list of users: FirstUser, SecondUser, ThirdUser. The 'input line' panel shows the input text: Hi. The 'Chat Info' panel shows the chat info: unlucky kakaotalk.

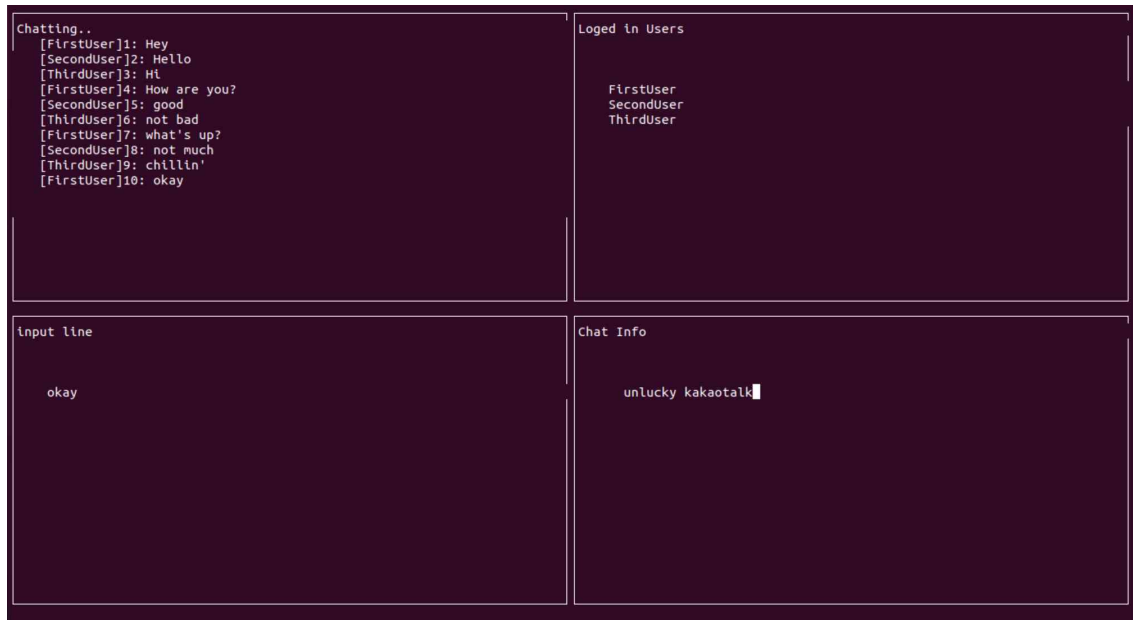
```
Chatting..
[FirstUser]1: Hey
[SecondUser]2: Hello
[ThirdUser]3: Hi

Logged in Users
FirstUser
SecondUser
ThirdUser

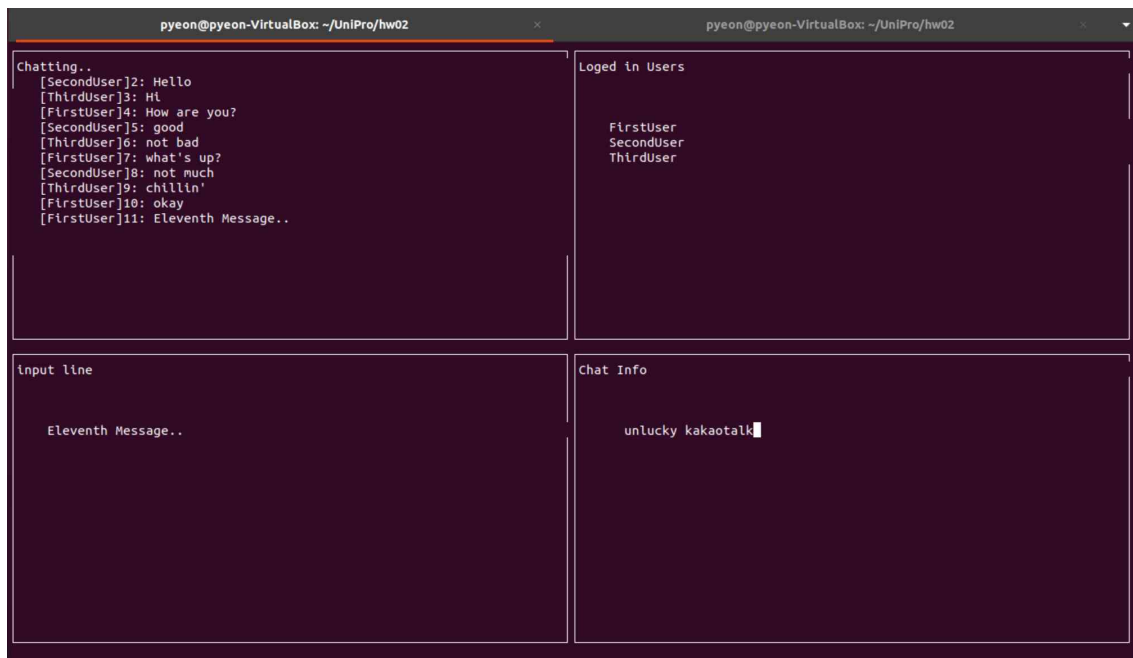
input line
Hi

Chat Info
unlucky kakaotalk
```

서브윈도우1에 입력 받은 메시지들이 출력 되며 서브윈도우2에는 입장한 유저들의 목록, 서브윈도우3에는 가장 최근에 input을 넣은 3번째 유저의 메시지가 출력된 모습입니다.



메시지가 10개까지 쌓인 모습입니다.



11번째 메시지가 들어오면 첫 메시지('Hey')는 사라지며 새로운 메시지('Eleventh Message..')가 최하단에 보이도록 '스크롤다운' 기능을 구현했습니다.