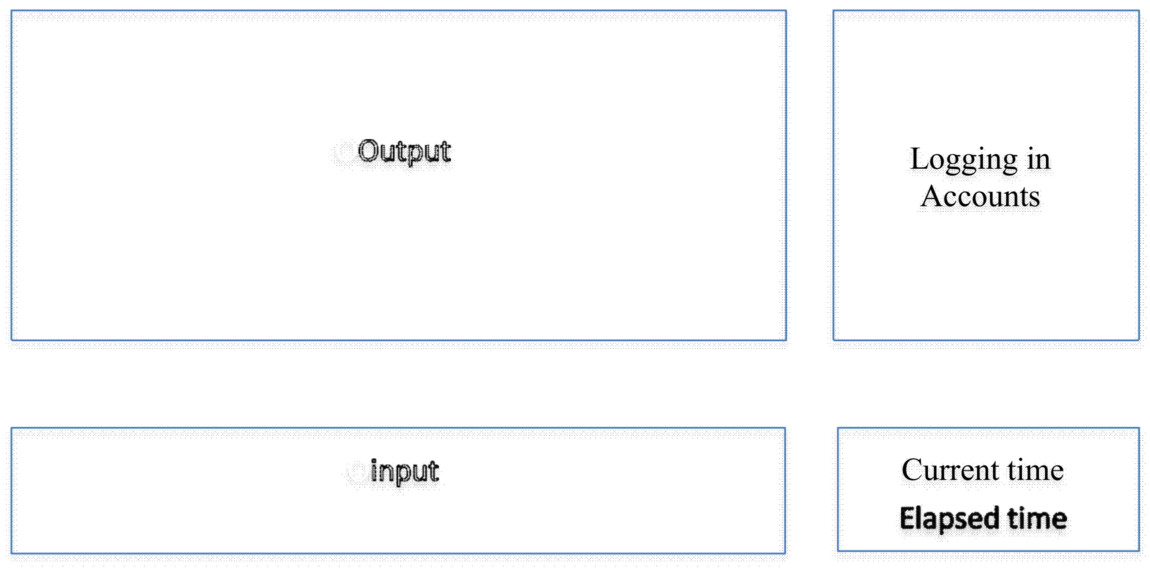
Take-Home Midterm Exam

School ID: 201624548 Name: 이재윤

The goal of take-home midterm exam is to design and implement the chatting app that you developed in homework 3 with thread synchronizations



[Guide line]

1. You must use named semaphore with sem_open(), sem_wait() and sem_post() to synchronize the shared memory (once or more times)
2. FetchMessageFromShmThread() fetches the message from the shared memory. DisplayMessageThread() prints out the fetched message on the output window. To synchronize these two threads, you must use the followings: pthread_cond_wait(), pthread_cond_post(), pthread_mutex_lock(), and pthread_mutex_unlock()
3. If you use the other mutex locks, just use them
4. You make sure that there are no synchronization problems in your chatting app

[1] Use ncurses to divide the main screen into 4 sub-windows

[2] Each sub-windows has box line.

- Terminal size is 80 X 24
- Output window size is 60 X 20
- Input window size is 60 X 4

- “Logging in Accounts” window size is 20 X 20

- “Time” window size is 20 X 4

[3] Display the current time and elapsed time (time will be refreshed in every 500ms)

[4] When a person logs in, his/her account will be appeared in the “Logging in Accounts” window.

When he/she type “/bye”, his/her account will be disappeared in the “Logging in Accounts” window.

(Question: do we need another shared memory?)

[5] Show that scrolling in the output window works correctly

How to test your app:

- Open the first terminal and run chat Jico to write text messages automatically and randomly between 1 sec and 2secs. Chatting Messages looks like:

  Hello!! My name is Jico I love to sing any song-1

  Hello!! My name is Jico I love to sing any song-2

  Hello!! My name is Jico I love to sing any song-3

  ....

- Open the second terminal and run chat Izzy to write text messages automatically and randomly between 1 sec and 2secs. Chatting Messages looks like:

  Hi!! I am Issy I like to play on the stage. Ho-1

  Hi!! I am Issy I like to play on the stage. Ho-2

  Hi!! I am Issy I like to play on the stage. Ho-3

  ......

- Open the third terminal and run chat GD to write text messages

- Open the fourth terminal and run chat IU to write text messages

  - Chat sequences are randomly and GD by IU.

Take-Home Midterm Exam

School ID: Name:

==

1. Submit your source file to the plato system

2020-05-04 Upload Complete

2. Put your program source as here (Do not put the screen shot of your source code!! If you insist, you will get zero point)

1) Makefile

```
#201624548_leejaeyoon
CC=gcc
CFLAGES=-g -Wall
all :201624548-Lee
201624548-Lee: 201624548-Lee.o
	$(CC) $(CFLAGES) -o $@ $^ -lncurses -lpthread
	201624548-Lee.o: 201624548-Lee.c
clean:
	rm -f *.o
	rm -f chat
```

2) chat.c

```
/*
 * chat.c
 * Mid-Term Exam Make Simple Chat Ncurses
 * Using semaphore, Mutex
 * Lee Jae Yoon(이재윤), 201624548
 * UNIX Programing(CP33357-059)
 * Revise HW #3
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ncurses.h>
#include <semaphore.h>
#include <fcntl.h>
#include <sys/ipc.h>
#include <sys/shm.h>

#define BUFFSIZE 1024
#define MAXACCOUNT 20
#define NAMESIZE 20

typedef struct message_buffer {
	char name[NAMESIZE];
	char msg[BUFFSIZE];
	char account[MAXACCOUNT][NAMESIZE];
	int id;
} MSG_BUFF;

// Save the Message By using buffer (In this Program I dont use Queue)
// Because I thought it could be implemented without using queue
MSG_BUFF buf_msg;
// Share the Message By using shm
MSG_BUFF * msg_buff;

int is_running;
int account_cnt;

// INPUT (Using semaphore, mutex)
void *get_input();
// Automatically Send message (Using semaphore, mutex)
void *autoSendMessage();
// Display Message From buf_msg (Using )
// Using pthread_cond_wait, pthread_mutex_unlock
void *DisplayMessageThread();
// Check new message from shm ( Using semaphore )
// Using pthread_cond_signal To DisplayMessageThread
void *FetchMessageFromShmThread();
// show current time, elapse time
void *show_time();
```

```
// show accounts who join this chat
void *show_account();

// ncurses setting and start chat threads
void chat();
// cleanup memory and other things
void cleanup();
void die(char * msg);

void setbox();

void refreshing();

int shmid;
char userID[NAMESIZE];
void * shmaddr = (void*) 0;

WINDOW *terminal_scr;
WINDOW *output_scr;
WINDOW *input_scr;
WINDOW *account_scr;
WINDOW *time_scr;

// Mutex (For Every Thread)
pthread_mutex_t thread_lock;
// Conditional Mutex (For Chat-Fetch Thread)
pthread_cond_t chat_cond;

// Semaphore Object
sem_t *sem;
int main(int argc, char* argv[])
{

	if (argc < 2) {
		fprintf(stderr, "[Useage]: ./chat UserID \n");
		exit(-1);
	}

	memset(userID , 0, NAMESIZE);
	strcpy(userID, (const char*)argv[1]);
	// attach shm
	key_t key;
	key = ftok("namedSemShm.c",'R');
	shmid = shmget(key, sizeof(MSG_BUFF), 0644|IPC_CREAT|IPC_EXCL);
	if (shmid < 0) {
		shmid = shmget(key, sizeof(MSG_BUFF), 0644);
		shmaddr = shmat(shmid, (void*)0, 0666);
		if (shmaddr < (void*) 0) {
			perror("shmat attach is failed : ");
			exit(-1);
		}
	}
	else {
		shmaddr = shmat(shmid, (void*)0, 0666);
	}
	if (shmaddr == (int*) (-1))
		perror("shmat: ");

	msg_buff = (MSG_BUFF *) shmaddr;

	// Mutex Initialize
	pthread_mutex_init(&thread_lock, NULL);
	pthread_cond_init(&chat_cond, NULL);

	// Clear Shared Memory (For Testing)
//	for (int i = 0; i < MAXACCOUNT; i++) {memset(msg_buff->account[i] , 0, NAMESIZE);}
//	return 0;

	for (int i = 0; i < MAXACCOUNT; i++) {
		if (msg_buff->account[i][0] == '\0') {
			strcpy(msg_buff->account[i], (const char*)argv[1]);
			account_cnt = i;
			break;
		}

	}

	// Attach named-semaphore
	sem = (sem_t*) malloc(sizeof(sem_t));
	// if not exist (= first person join chat)
	if (account_cnt == 0) {
		sem_unlink("201624548_Sem");
		sem = sem_open("201624548_Sem",O_CREAT|O_EXCL,0666,1);
	}
	// if exist
	else {
		sem = sem_open("201624548_Sem",0,0666,1);
	}

	if (sem == NULL) {
		perror("sem_open Failed : ");
		cleanup();
		exit(-1);
	}
	initscr();
	chat();
	endwin();
	return 0;
}


void chat()

{

	curs_set(0);

	scrollok(input_scr, TRUE);

	terminal_scr = newwin(24, 80, 0, 0);

	output_scr = subwin(terminal_scr, 20, 60, 0, 0);
	input_scr = subwin(terminal_scr, 4, 60, 20, 0);
	account_scr = subwin(terminal_scr, 20, 20, 0, 60);
	time_scr = subwin(terminal_scr, 4, 20, 20, 60);
	mvwhline(output_scr, 1, 1, 68, 0);
	wprintw(output_scr, "	***** Type /bye to quit !! ***** \n\n");
	setbox();
	refreshing();

	is_running = 1;

	//thread Start
	pthread_t thread[6];
	pthread_create(&thread[0], NULL, get_input, NULL);
	pthread_create(&thread[1], NULL, DisplayMessageThread, NULL);
	pthread_create(&thread[2], NULL, show_account, NULL);
	pthread_create(&thread[3], NULL, show_time, NULL);
	pthread_create(&thread[4], NULL, FetchMessageFromShmThread, NULL);
	pthread_create(&thread[5], NULL, autoSendMessage, NULL);

	for (int i = 0; i < 6; i++)
	{
		pthread_join(thread[i], NULL);
	}



	delwin(terminal_scr);
	delwin(output_scr);
	delwin(input_scr);
	delwin(account_scr);
	delwin(time_scr);


}

void *autoSendMessage() {
	int account_num = account_cnt;
```

```
        // This Thread Only For 0, 1 (Jico, Izzy) Person
        if (account_num > 1) {
                return NULL;
        }
        // Make Chat Message
        char tmp[BUFFSIZE] = "";
        if (account_num == 0) {
                strcat(tmp,"Hello! My name is ");
                strcat(tmp, userID);
                strcat(tmp, " I love to sing any song");
        }
        else if (account_num == 1) {
                strcat(tmp,"Hi!! i am ");
                strcat(tmp, userID);
                strcat(tmp, " I like to play on the stage. Ho");
        }

        // Send Message (same with get_input())
        srand(0);
        int send_id = 1;
        while (is_running) {
                pthread_mutex_lock(&thread_lock);
                sem_wait(sem);
                sprintf(msg_buff->msg, " %s-%d\n", tmp,send_id);
                strcpy(msg_buff->name, userID);
                msg_buff->id = send_id;
                sem_post(sem);
                wprintw(output_scr, " [Send : %d] > %s", send_id++, msg_buff->msg);
                setbox();
                refreshing();
                pthread_mutex_unlock(&thread_lock);
                sleep((rand()%2+1));
        }
}
```

```
void *show_account()
{
	int cnt;
	while (is_running) {
		pthread_mutex_lock(&thread_lock);
		werase(account_scr);
		cnt = 1;
		for (int i = 0; i < MAXACCOUNT; i++) {
			if (msg_buff->account[i][0] != '\0') {
				mvwprintw(account_scr, cnt++, 2,"%s", msg_buff->account[i]);
			}
		}
		box(account_scr, ACS_VLINE, ACS_HLINE);
		wrefresh(account_scr);
		pthread_mutex_unlock(&thread_lock);
		usleep(500);

	}
}

void *show_time()
{
	time_t start,now;
	start = time(0);
	struct tm ts;
	char buf[80];
	while(is_running){
		//시간 출력
		pthread_mutex_lock(&thread_lock);
		werase(time_scr);
		time(&now);
		ts = *localtime(&now);
		strftime(buf, sizeof(buf), "%H-%M-%S",&ts);
		mvwprintw(time_scr, 1,2,"%s",buf);
```

```
                now = time(0);
                time_t tmp = now-start;
                ts = *localtime(&tmp);
                // Match with our time
                ts.tm_hour -= 9;
                strftime(buf, sizeof(buf), "%H-%M-%S",&ts);
                mvwprintw(time_scr, 2,2,"%s",buf);
                box(time_scr, ACS_VLINE, ACS_HLINE);
                wrefresh(time_scr);
                pthread_mutex_unlock(&thread_lock);
                usleep(500000);
        }
}

void *get_input()
{
        char tmp[BUFFSIZE];
        int send_id = 1;
        while (is_running) {
                mvwgetstr(input_scr, 1, 1, tmp);

                pthread_mutex_lock(&thread_lock);
                sem_wait(sem);
                sprintf(msg_buff->msg, " %s\n", tmp);
                strcpy(msg_buff->name, userID);
                msg_buff->id = send_id;
                sem_post(sem);
                if (strcmp (msg_buff->msg, " /bye\n") == 0 ) {
                        die("exit");
                }
                wprintw(output_scr, " [Send : %d] > %s", send_id++, msg_buff->msg);
                werase(input_scr);
                setbox();
                refreshing();
                pthread_mutex_unlock(&thread_lock);
```

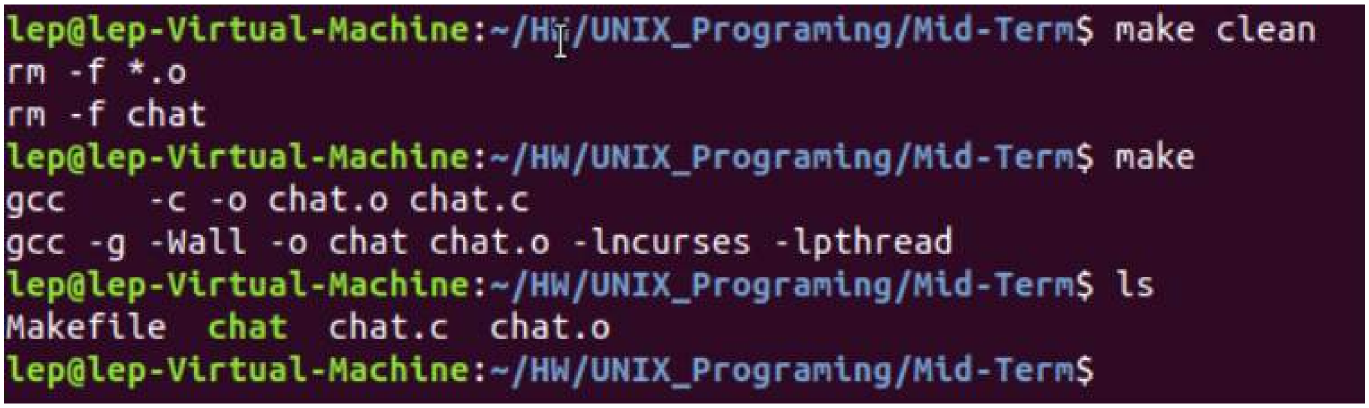
```
            usleep(300);
        }
}

// shm에서 데이터를 메세지 버퍼에 저장
void * FetchMessageFromShmThread() {
        strcpy(buf_msg.msg, msg_buff->msg);
        strcpy(buf_msg.name, msg_buff->name);
        buf_msg.id = msg_buff->id;
        int i = 0;
        while (is_running) {
                sem_wait(sem);
                if (strcmp(msg_buff->msg, " .exit\n") == 0) {
                        fprintf(stderr, "Chat is closed\n");
                        is_running = 0;
                }
                else {
                        if ( (   strcmp(msg_buff->msg, buf_msg.msg) != 0 ||
                                strcmp(msg_buff->name, buf_msg.name) != 0 ||
                                buf_msg.id != msg_buff->id ) &&
                                strcmp(msg_buff->name,userID) != 0 ) {

                                buf_msg.id = msg_buff->id;
                                strcpy(buf_msg.msg, msg_buff->msg);
                                strcpy(buf_msg.name, msg_buff->name);
                                pthread_cond_signal(&chat_cond);
                        }
                }
                sem_post(sem);
                usleep(500);
        }
}

// FetchMessageFromShmThread 에서 시그널을 보내면 메세지 버퍼에 저장된 값을 출력
void *DisplayMessageThread()
```

```
{
	int myrecv = 1;
	while (is_running) {
		pthread_cond_wait(&chat_cond,&thread_lock);
		if (strcmp(buf_msg.msg, " .exit\n") == 0) {
			fprintf(stderr, "Chat is closed\n");
			is_running = 0;
			cleanup();
		}
		else {
			if (strcmp(buf_msg.msg, " /bye\n") == 0) {
				wprintw(output_scr, " [Exit %s] Goodbye %s !\n",
					buf_msg.name, buf_msg.name);
			}
			else {
				wprintw(output_scr, " [Recv #%d By %s] > %s",
					myrecv++, buf_msg.name, buf_msg.msg);
			}
			box(output_scr, ACS_VLINE, ACS_HLINE);
			wrefresh(output_scr);
		}
		pthread_mutex_unlock(&thread_lock);
	}
}

void setbox() {
	box(output_scr, ACS_VLINE, ACS_HLINE);
	box(input_scr, ACS_VLINE, ACS_HLINE);
	box(account_scr, ACS_VLINE, ACS_HLINE);
	box(time_scr, ACS_VLINE, ACS_HLINE);
	wrefresh(input_scr);
}

void refreshing() {
	wrefresh(output_scr);
```

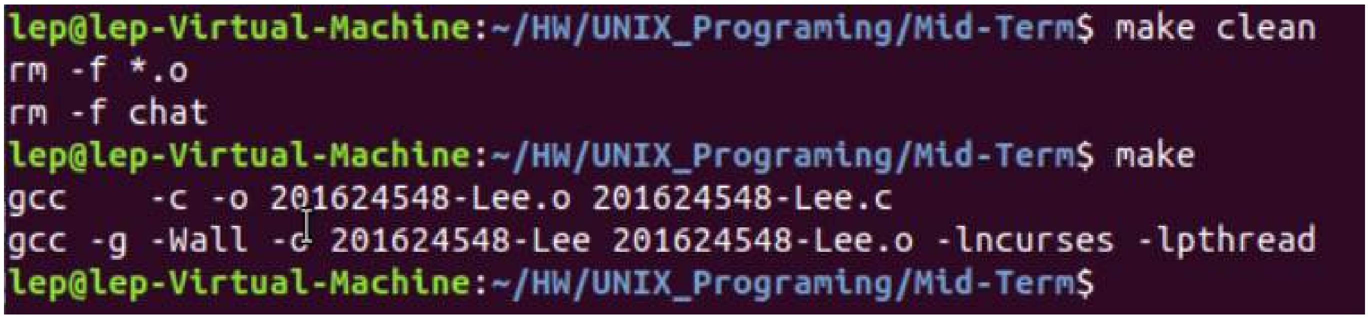
```
    wrefresh(input_scr);

    wrefresh(account_scr);

    wrefresh(time_scr);

}


void cleanup()

{

    memset(msg_buff->account[account_cnt] , 0, NAMESIZE);

    delwin(terminal_scr);

    delwin(output_scr);

    delwin(input_scr);

    delwin(account_scr);

    delwin(time_scr);

    endwin();

    sem_close(sem);

    shmdt(shmaddr);

}


void die(char * msg)

{

    cleanup();

    perror(msg);

    exit(-1);

}
```

3. You must show the building result after compiling and linking your source codes. You must show no warnings and errors (Use gcc -Wall option).

(Put a screen shot of your C debugging output)



```
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ make clean
rm -f *.o
rm -f chat
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ make
gcc    -c -o chat.o chat.c
gcc -g -Wall -o chat chat.o -lncurses -lpthread
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ ls
Makefile  chat  chat.c  chat.o
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$
```



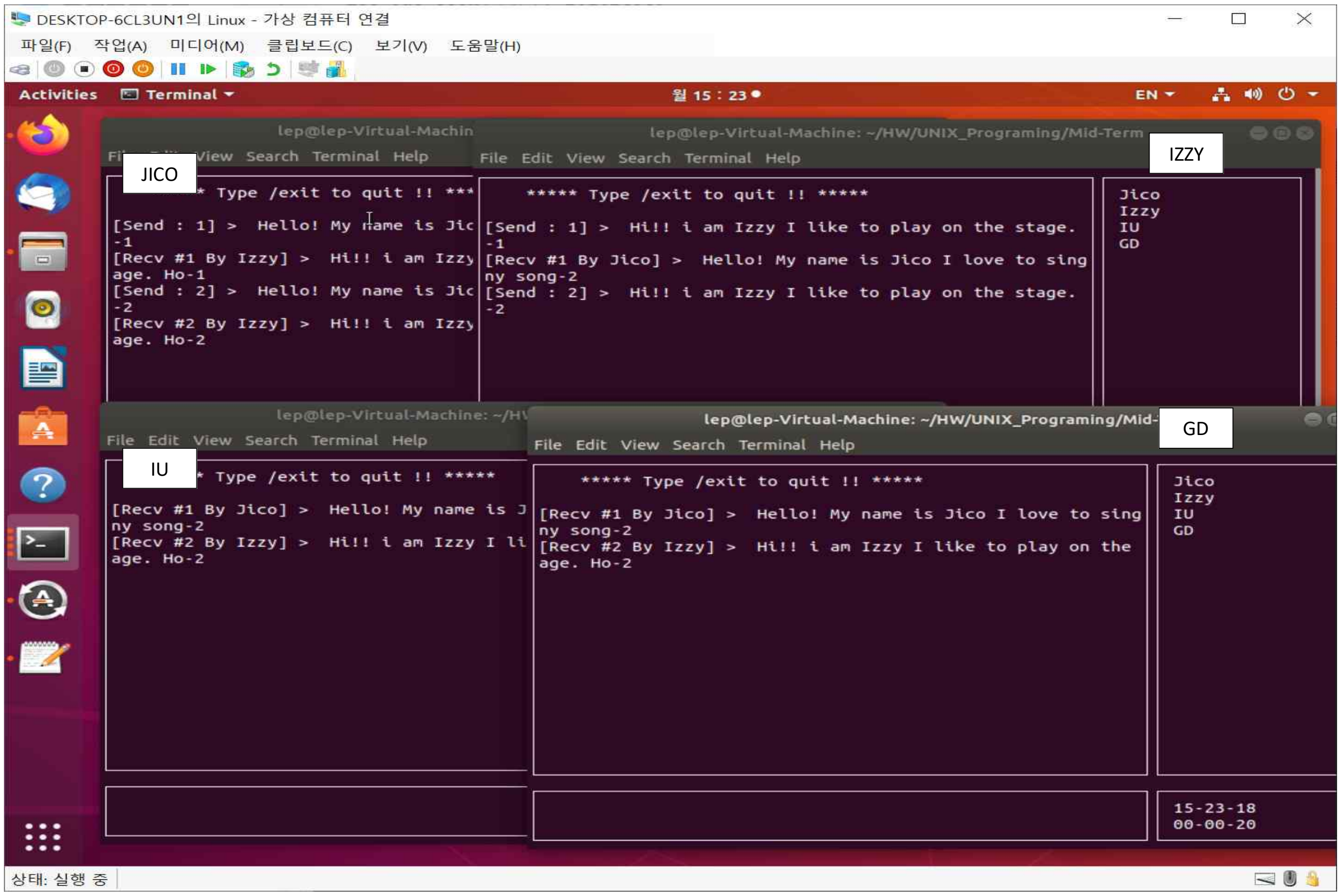
```
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ make clean
rm -f *.o
rm -f chat
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ make
gcc    -c -o 201624548-Lee.o 201624548-Lee.c
gcc -g -Wall -o 201624548-Lee 201624548-Lee.o -lncurses -lpthread
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$
```

4. **Put a screen shot of output generated by your program. Your output screen shot must be readable for me to verify your chat program. You need to explain your outcomes in English**

First. I wanted to send (Jico, Izzy)'s message automatically. (random 1~2 seconds)

But I find it is too fast to capture screen, so I replace it to 15~16 seconds to capture screenshot.

Four persons join the Chat


DESKTOP-6CL3UN1의 Linux - 가상 컴퓨터 연결
파일(F) 작업(A) 미디어(M) 클립보드(C) 보기(V) 도움말(H)
Activities
Terminal
월 15 : 23
EN
lep@lep-Virtual-Machine: ~/HW/UNIX_Programing/Mid-Term
File Edit View Search Terminal Help
JICO
IZZY
IU
GD
상태: 실행 중

JICO:

```
* Type /exit to quit !! ***

[Send : 1] >  Hello! My name is Jic
-1
[Recv #1 By Izzy] >  Hi!! i am Izzy
age. Ho-1
[Send : 2] >  Hello! My name is Jic
-2
[Recv #2 By Izzy] >  Hi!! i am Izzy
age. Ho-2
```

IZZY:

```
    ***** Type /exit to quit !! *****

[Send : 1] >  Hi!! i am Izzy I like to play on the stage.
-1
[Recv #1 By Jico] >  Hello! My name is Jico I love to sing
ny song-2
[Send : 2] >  Hi!! i am Izzy I like to play on the stage.
-2
```

```
Jico
Izzy
IU
GD
```

IU:

```
* Type /exit to quit !! *****

[Recv #1 By Jico] >  Hello! My name is J
ny song-2
[Recv #2 By Izzy] >  Hi!! i am Izzy I li
age. Ho-2
```

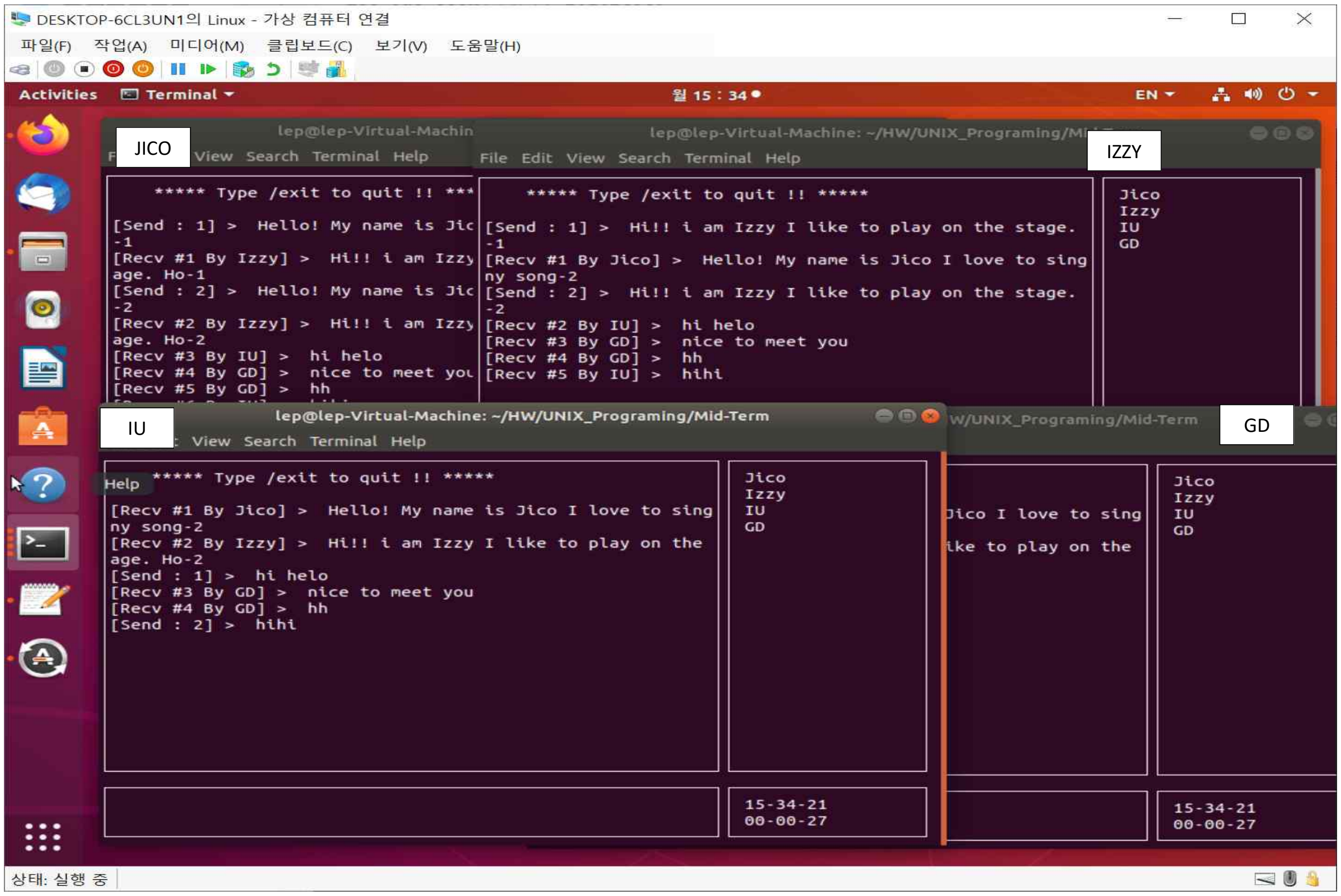
GD:

```
    ***** Type /exit to quit !! *****

[Recv #1 By Jico] >  Hello! My name is Jico I love to sing
ny song-2
[Recv #2 By Izzy] >  Hi!! i am Izzy I like to play on the
age. Ho-2
```

```
Jico
Izzy
IU
GD
```

```
15-23-18
00-00-20
```

Chat IU -> GD -> GD -> IU


DESKTOP-6CL3UN1의 Linux - 가상 컴퓨터 연결
파일(F) 작업(A) 미디어(M) 클립보드(C) 보기(V) 도움말(H)
Activities
Terminal
월 15 : 34
EN
JICO
IZZY
IU
GD
lep@lep-Virtual-Machine: ~/HW/UNIX_Programing/Mid-Term
File Edit View Search Terminal Help
상태: 실행 중

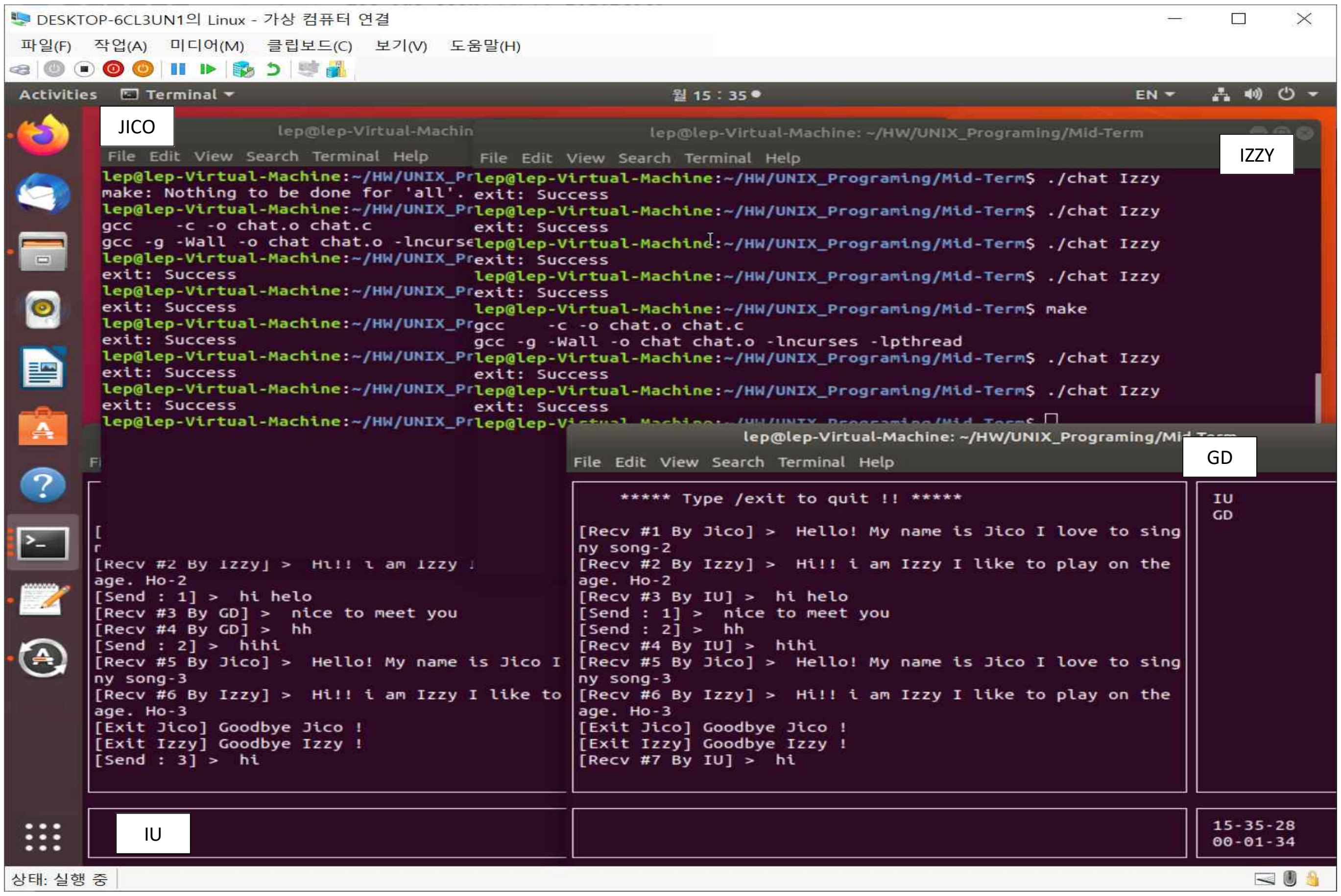
```
 ***** Type /exit to quit !! ***
[Send : 1] >  Hello! My name is Jic
-1
[Recv #1 By Izzy] >  Hi!! i am Izzy
age. Ho-1
[Send : 2] >  Hello! My name is Jic
-2
[Recv #2 By Izzy] >  Hi!! i am Izzy
age. Ho-2
[Recv #3 By IU] >  hi helo
[Recv #4 By GD] >  nice to meet you
[Recv #5 By GD] >  hh
```

```
 ***** Type /exit to quit !! *****
[Send : 1] >  Hi!! i am Izzy I like to play on the stage.
-1
[Recv #1 By Jico] >  Hello! My name is Jico I love to sing
ny song-2
[Send : 2] >  Hi!! i am Izzy I like to play on the stage.
-2
[Recv #2 By IU] >  hi helo
[Recv #3 By GD] >  nice to meet you
[Recv #4 By GD] >  hh
[Recv #5 By IU] >  hihi

Jico
Izzy
IU
GD
```

```
 ***** Type /exit to quit !! *****
[Recv #1 By Jico] >  Hello! My name is Jico I love to sing
ny song-2
[Recv #2 By Izzy] >  Hi!! i am Izzy I like to play on the
age. Ho-2
[Send : 1] >  hi helo
[Recv #3 By GD] >  nice to meet you
[Recv #4 By GD] >  hh
[Send : 2] >  hihi

Jico
Izzy
IU
GD

15-34-21
00-00-27
```

```
Jico I love to sing
ike to play on the

Jico
Izzy
IU
GD

15-34-21
00-00-27
```

Exit Test


DESKTOP-6CL3UN1의 Linux - 가상 컴퓨터 연결
파일(F) 작업(A) 미디어(M) 클립보드(C) 보기(V) 도움말(H)
Activities
Terminal
월 15 : 35
EN
JICO
IZZY
GD
IU
lep@lep-Virtual-Machine: ~/HW/UNIX_Programing/Mid-Term
File Edit View Search Terminal Help
상태: 실행 중

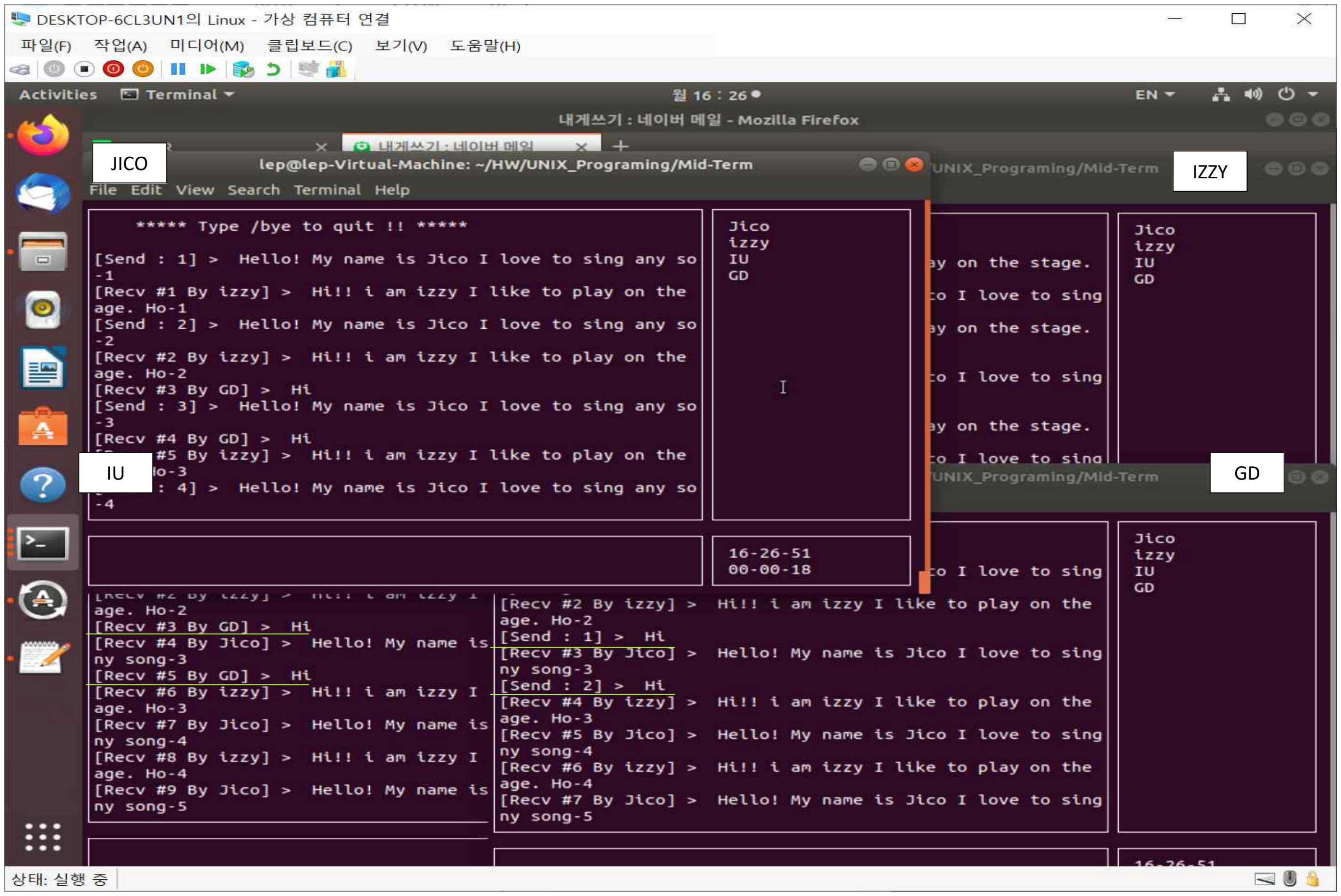
```
lep@lep-Virtual-Machine:~/HW/UNIX_Pr
make: Nothing to be done for 'all'.
lep@lep-Virtual-Machine:~/HW/UNIX_Pr
gcc    -c -o chat.o chat.c
gcc -g -Wall -o chat chat.o -lncurse
lep@lep-Virtual-Machine:~/HW/UNIX_Pr
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Pr
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Pr
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Pr
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Pr
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Pr
```

```
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ ./chat Izzy
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ ./chat Izzy
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ ./chat Izzy
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ ./chat Izzy
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ make
gcc    -c -o chat.o chat.c
gcc -g -Wall -o chat chat.o -lncurses -lpthread
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ ./chat Izzy
exit: Success
lep@lep-Virtual-Machine:~/HW/UNIX_Programing/Mid-Term$ ./chat Izzy
exit: Success
```

```
[
r
[Recv #2 By Izzy] >  Hi!! i am Izzy I
age. Ho-2
[Send : 1] >  hi helo
[Recv #3 By GD] >  nice to meet you
[Recv #4 By GD] >  hh
[Send : 2] >  hihi
[Recv #5 By Jico] >  Hello! My name is Jico I
ny song-3
[Recv #6 By Izzy] >  Hi!! i am Izzy I like to
age. Ho-3
[Exit Jico] Goodbye Jico !
[Exit Izzy] Goodbye Izzy !
[Send : 3] >  hi
```

```
 ***** Type /exit to quit !! *****
[Recv #1 By Jico] >  Hello! My name is Jico I love to sing
ny song-2
[Recv #2 By Izzy] >  Hi!! i am Izzy I like to play on the
age. Ho-2
[Recv #3 By IU] >  hi helo
[Send : 1] >  nice to meet you
[Send : 2] >  hh
[Recv #4 By IU] >  hihi
[Recv #5 By Jico] >  Hello! My name is Jico I love to sing
ny song-3
[Recv #6 By Izzy] >  Hi!! i am Izzy I like to play on the
age. Ho-3
[Exit Jico] Goodbye Jico !
[Exit Izzy] Goodbye Izzy !
[Recv #7 By IU] >  hi

IU
GD

15-35-28
00-01-34
```

Chat Random Seconds (1~2) Chat Screenshot


DESKTOP-6CL3UN1의 Linux - 가상 컴퓨터 연결
파일(F) 작업(A) 미디어(M) 클립보드(C) 보기(V) 도움말(H)
Activities
Terminal
월 16 : 26
EN
내게쓰기 : 네이버 메일 - Mozilla Firefox
JICO
IZZY
IU
GD
lep@lep-Virtual-Machine: ~/HW/UNIX_Programing/Mid-Term
File Edit View Search Terminal Help
상태: 실행 중

```
    ***** Type /bye to quit !! *****

[Send : 1] >  Hello! My name is Jico I love to sing any so
-1
[Recv #1 By izzy] >  Hi!! i am izzy I like to play on the
age. Ho-1
[Send : 2] >  Hello! My name is Jico I love to sing any so
-2
[Recv #2 By izzy] >  Hi!! i am izzy I like to play on the
age. Ho-2
[Recv #3 By GD] >  Hi
[Send : 3] >  Hello! My name is Jico I love to sing any so
-3
[Recv #4 By GD] >  Hi
       #5 By izzy] >  Hi!! i am izzy I like to play on the
       Ho-3
       : 4] >  Hello! My name is Jico I love to sing any so
-4

Jico
izzy
IU
GD

16-26-51
00-00-18
```

```
ay on the stage.
co I love to sing
ay on the stage.
co I love to sing
ay on the stage.
co I love to sing

Jico
izzy
IU
GD
```

```
age. Ho-2
[Recv #3 By GD] >  Hi
[Recv #4 By Jico] >  Hello! My name is
ny song-3
[Recv #5 By GD] >  Hi
[Recv #6 By izzy] >  Hi!! i am izzy I
age. Ho-3
[Recv #7 By Jico] >  Hello! My name is
ny song-4
[Recv #8 By izzy] >  Hi!! i am izzy I
age. Ho-4
[Recv #9 By Jico] >  Hello! My name is
ny song-5
```

```
co I love to sing
[Recv #2 By izzy] >  Hi!! i am izzy I like to play on the
age. Ho-2
[Send : 1] >  Hi
[Recv #3 By Jico] >  Hello! My name is Jico I love to sing
ny song-3
[Send : 2] >  Hi
[Recv #4 By izzy] >  Hi!! i am izzy I like to play on the
age. Ho-3
[Recv #5 By Jico] >  Hello! My name is Jico I love to sing
ny song-4
[Recv #6 By izzy] >  Hi!! i am izzy I like to play on the
age. Ho-4
[Recv #7 By Jico] >  Hello! My name is Jico I love to sing
ny song-5

Jico
izzy
IU
GD

16-26-51
```

I painted Green line on IU and GE terminal.

I thought the exit method was / exit. But the exit String is /bye so I change it after taking all the screenshots.

I changed it late. but source code is