

Homework #1

정보컴퓨터공학부

201424423

김영진

1. Put your program source as here

<chatshm.h>

```
#ifndef __CHAT_SHM_H__  
#define __CHAT_SHM_H__
```

```
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/ipc.h>  
#include <sys/shm.h>  
#include <unistd.h>  
#include <time.h>
```

```
// Declare struct for saving chat information
```

```
typedef struct chatInfo {
```

```
    char userID[20];  
    char messageTime[10];  
    char message[40];  
    int messageID; // I add messageID variable to distinguish each message
```

```
} CHAT_INFO;
```

```
// Define for checking current time
```

```
time_t now;  
struct tm t;
```

```
// Define function that print chat message
```

```
void printchat(CHAT_INFO *chatInfo, int i);
```

```
#endif // __CHAT_SHM_H__
```

<chat.c>

```
#include "chatshm.h"
```

```
int main(int argc, char* argv[]) {
```

```
    int shmID; // Define for saving shared memory ID
```

```
    char userID[20]; // Define for saving user ID
```

```
    // Declare instance of CHAT_INFO struct and allocate memory 10 times of checkInfo struct
```

```
    CHAT_INFO *chatInfo = (CHAT_INFO *)malloc(sizeof(struct checkInfo) * 10);
```

```
    // Declare shared memory address for 0 and type casting int to void*
```

```
    void *shmaddr = (void *) 0;
```

```
    // If user doesn't type userID, messenger print error message
```

```
    if(argc < 2){
```

```
        fprintf(stderr, "[Usage]: ./chat UserID Wn");
```

```
        exit(-1);
```

```
    }
```

```
    // Save user ID in userID variable
```

```
    strcpy(userID, argv[1]);
```

```
    // Create 1 shared memory, its key is 20200406
```

```
    shmID = shmget((key_t)20200406, sizeof(CHAT_INFO), 0666|IPC_CREAT|IPC_EXCL);
```

```
    // If there already exists shared memory which has key 20200406, then Dont' create sharedmemory
```

```
    // and just reference shared memory ID
```

```
    if(shmID < 0){
```

```
        shmID = shmget((key_t)20200406, sizeof(CHAT_INFO), 0666);
```

```
    }
```

```
    // attach shared memory into process
```

```
    shmaddr = shmat(shmID, (void *)0, 0000);
```

```
    // type casting from void* to CHAT_INFO*
```

```
    chatInfo = (CHAT_INFO *) shmaddr;
```

```

printf("----- Chat Message -----\\n");
while(1){
    // Define variables for saving message and current time
    char inputstr[40];
    char time_now[50];

    // get input message from user and save in inputstr
    printf("Enter your message : ");
    fgets(inputstr, 40, stdin);

    // get current time
    time(&now);
    t = *localtime(&now);

    // Change form to what we want
    strftime(time_now, sizeof(time_now), "%H:%M:%S", &t);

    // if user type "..", print all message except user's own message
    if(strcmp(inputstr, "..\\n")==0){
        int i = 0;
        for(i=0; i<chatInfo->messageID; i++){
            if(strcmp(chatInfo[i].userID, userID) != 0){
                printchat(chatInfo, i);
            }
        }
    }
    // if user type /bye, break loop
    else if(strcmp(inputstr, "/bye\\n")==0){
        break;
    }
    // If user type any message, save each message in chatInfo shared memory
    else{
        strcpy(chatInfo[chatInfo->messageID].userID, userID);
        strcpy(chatInfo[chatInfo->messageID].messageTime, time_now);
        strcpy(chatInfo[chatInfo->messageID].message, inputstr);

        // Print user's own message echoed

```

```

        printchat(chatInfo, chatInfo->messageID);
        // Increase messageID for distinguish each message
        chatInfo->messageID++;
    }
}

return 0;
}

```

<printchat.c>

```

#include "chatshm.h"

// Print userID, messageTime, message
void printchat(CHAT_INFO *chatInfo, int i){
    printf("[%s] %s : %s\n", chatInfo[i].userID, chatInfo[i].messageTime, chatInfo[i].message);
}

```

<shmremove.c>

```

#include "chatshm.h"

int main() {

    // Define for saving shared memory Id
    int shmID;

    // Get shared memory ID which has key 20200406
    shmID = shmget((key_t)20200406, sizeof(CHAT_INFO), 0666);

    // If there's no shared memory ID which has key 20200406, then print error message
    if(shmID < 0){
        perror ("shmget failed : ");
        exit(-1);
    }

    // Delete shared memory
    // If there's no shared memory to delete, print error message
    if(shmctl(shmID, IPC_RMID, 0) < 0) {
        printf("Failed to delete shared memory\n");
    }
}

```

```

        exit(-1);
    }
    // If deleting is finished successfully, print success message
    else {
        printf("Successfully delete shared memory\n");
    }

    return 0;
}

```

<Makefile>

```

CC = gcc
CFLAGS = -g -Wall
TARGET = messenger
OBJECTS = chat.o printchat.o

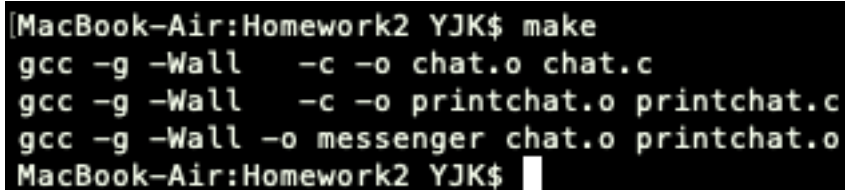
all : $(TARGET)

$(TARGET): $(OBJECTS)
    $(CC) $(CFLAGS) -o $@ $^

clean :
    rm *.o messenger
    ./shmremove

```

2. You must show the building result after compiling and linking your source codes. You must show no warnings and errors.

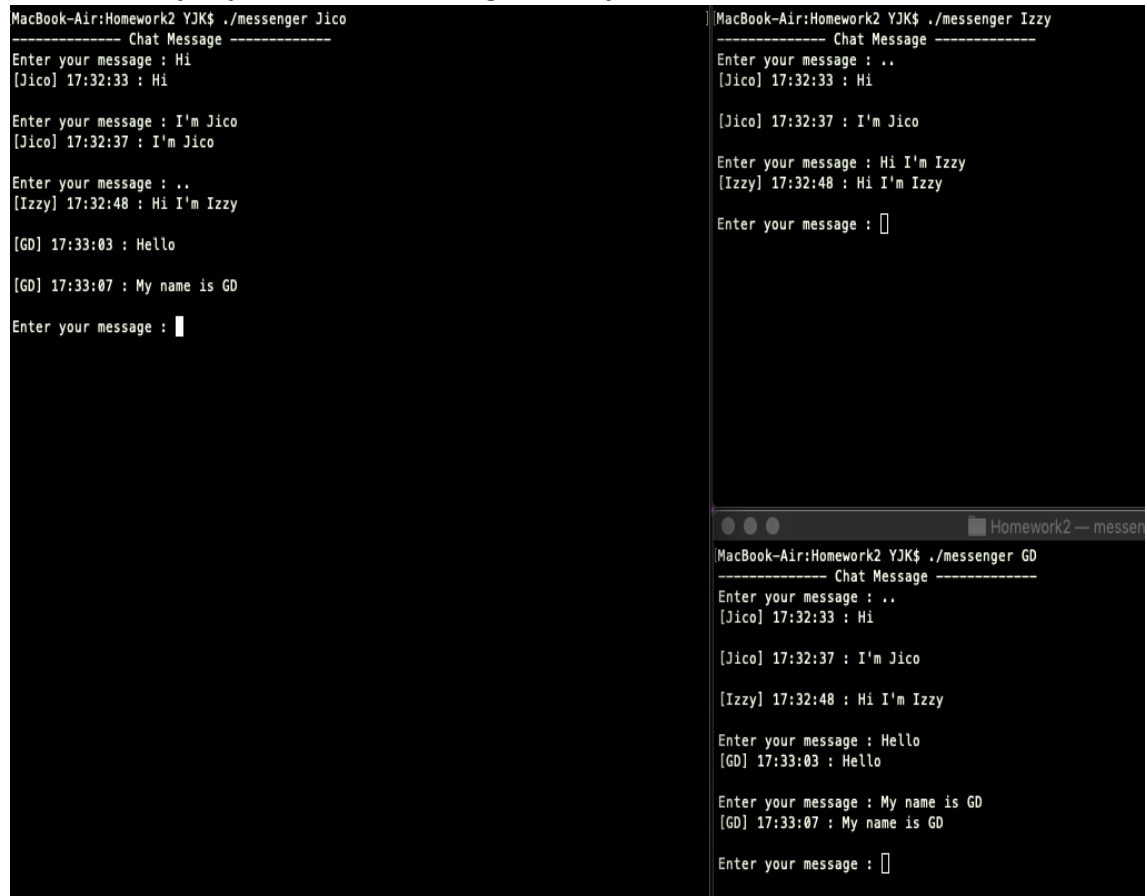


```

[MacBook-Air:Homework2 YJK$ make
gcc -g -Wall -c -o chat.o chat.c
gcc -g -Wall -c -o printchat.o printchat.c
gcc -g -Wall -o messenger chat.o printchat.o
MacBook-Air:Homework2 YJK$

```

3. Put a screen shot of output generated by your program as well as the Ubuntu system's clock to verify if your clock is working correctly or not



```
MacBook-Air:Homework2 YJK$ ./messenger Jico
----- Chat Message -----
Enter your message : Hi
[Jico] 17:32:33 : Hi

Enter your message : I'm Jico
[Jico] 17:32:37 : I'm Jico

Enter your message : ..
[Izzy] 17:32:48 : Hi I'm Izzy

[GD] 17:33:03 : Hello

[GD] 17:33:07 : My name is GD

Enter your message : █

MacBook-Air:Homework2 YJK$ ./messenger Izzy
----- Chat Message -----
Enter your message : ..
[Jico] 17:32:33 : Hi

[Jico] 17:32:37 : I'm Jico

[Izzy] 17:32:48 : Hi I'm Izzy

Enter your message : Hello
[GD] 17:33:03 : Hello

Enter your message : My name is GD
[GD] 17:33:07 : My name is GD

Enter your message : █

MacBook-Air:Homework2 YJK$ ./messenger GD
----- Chat Message -----
Enter your message : ..
[Jico] 17:32:33 : Hi

[Jico] 17:32:37 : I'm Jico

[Izzy] 17:32:48 : Hi I'm Izzy

Enter your message : Hello
[GD] 17:33:03 : Hello

Enter your message : My name is GD
[GD] 17:33:07 : My name is GD

Enter your message : █
```