

Name : Gourav Kumar Shaw

Enrollment Id. : 2020CSB010

Section: Gx

Subject : Operating Systems Laboratory

Qn. Independently run C programs using shared variables.

Answer:

teacher.c

```
// 2020CSB010 GOURAV KUMAR SHAW

#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h> /* for shmget() shmctl() */
#include <unistd.h>
#include <sys/shm.h> /* for shmget() shmctl() */
#include <signal.h> /* for signal(2) kill(2) */
#include <time.h>

typedef void (*sighandler_t)(int);
typedef struct att
{
    int roll;
    time_t tm;
} att;
int shmids;
int n;
/*
The value of n is the number of students. It is obtained from the first
argument passed to the program at runtime. The line n=atoi(argv[1]); converts
the string passed as the first argument to an integer and stores it in n. This
value is then used to determine the size of the shared memory to be created
and the number of students' records to be stored in it.
*/
typedef void (*sighandler_t)(int);
```

```
/*
releaseSHM is a function in the given code that is installed as a signal
handler for the SIGINT signal (which is generated when the user presses Ctrl-
C).
```

When the SIGINT signal is received, releaseSHM is called and performs the following actions:

- It prints the roll numbers of all students who have logged in (where a student has logged in if their roll value is not equal to -1).

- It removes the shared memory segment by calling shmctl with the IPC_RMID command.

- It terminates the process by calling kill with the SIGKILL signal.

In this way, the function releaseSHM provides a way to clean up shared memory and terminate the process when the user presses Ctrl-C.

```
*/
void releaseSHM(int signum)
{
    /*
    signum in the releaseSHM function is an integer argument representing the
    signal number that was caught by the signal handler. It is used to determine
    which signal triggered the handler. In this case, the signal being handled is
    SIGINT (the interrupt signal sent by pressing Ctrl + C), and signum is equal
    to SIGINT when this signal is caught.
    */
    int status;
    printf(" ctrl+c is pressed and the students who have logged in are : \n");
    int i = 0;
    att *shm;
    shm = shmat(shmid, NULL, 0);
    char buffer[100];
    for (i = 0; i < n; i++)
    {
        if ((shm + i)->roll != -1)
        {
            time_t t;
            t = (shm + i)->tm;
            struct tm *tm = localtime(&t);
            strftime(buffer, 80, "%B %d, %Y, %T", tm);
            /*
            strftime is a function in C to format time and date information.
            In the code, the strftime function is used to format the time stored in the tm
            structure and store the formatted string in the buffer character array.

            The first argument, buffer, is a character array to store the
            formatted string.
```

The second argument, 80, is the size of the buffer character array.

The third argument, "%B %d, %Y, %T", is a format string that specifies how to format the time stored in the tm structure.

%B represents the full month name, %d represents the day of the month as a decimal number, %Y represents the year with century as a decimal number and %T represents the time in 24-hour notation HH:MM:SS.

The fourth argument, tm, is a pointer to a tm structure that holds the time information.

After the function call, the buffer character array will contain a formatted string of the time and date information stored in the tm structure.

```
*/
    printf("\n Student with Roll Number:  %d attended at %s ", (shm +
i)->roll, buffer);
    }
    else
    {
        break;
    }
}
printf("\n");
// int shmctl(int shmid, int cmd, struct shmctl_ds *buf);
status = shmctl(shmid, IPC_RMID, NULL); /* IPC_RMID is the command for
destroying the shared memory*/
if (status == 0)
{
    fprintf(stderr, "Remove shared memory with id = %d.\n", shmid);
}
else if (status == -1)
{
    fprintf(stderr, "Cannot remove shared memory with id = %d.\n", shmid);
}
else
{
    fprintf(stderr, "shmctl() returned wrong value while removing shared
memory with id = %d.\n", shmid);
}

// int kill(pid_t pid, int sig);
/*
the signal SIGKILL to the process with ID 0, which is typically the init
process. The kill function returns 0 if the signal was successfully sent, and
-1 if an error occurred.
```

The SIGKILL signal cannot be caught or ignored, and it immediately terminates the process it is sent to. This makes it a forceful and extreme way to end a process, but it is also a last resort when other signals or methods have failed to stop a process.

In this code snippet, 0 refers to the process ID (PID) of the process to be killed. A PID of 0 typically means the process group ID of the caller process, which means that the signal will be sent to all processes in the same process group as the caller.

SIGKILL is a signal in the Unix operating system that terminates a process immediately, regardless of whether the process has any pending operations or not. The kill function sends a signal to a process specified by its PID. In this case, the signal sent is SIGKILL, which means the process identified by PID 0 will be terminated immediately.

```
*/
status = kill(0, SIGKILL);
if (status == 0)
{
    fprintf(stderr, "kill susccesful.\n");
}
else if (status == -1)
{
    perror("kill failed.\n");
    fprintf(stderr, "Cannot remove shared memory with id = %d.\n", shmid);
}
else
{
    fprintf(stderr, "kill(2) returned wrong value.\n");
}
/*
```

fprintf is used in this code to print error messages to the standard error stream (stderr). The function fprintf is similar to printf, but it allows you to specify a file stream (in this case, stderr) as the first argument, to which the formatted output will be written.

By using fprintf instead of printf, the error messages will be displayed on the screen and will not be mixed with the standard output. This can be useful in cases where you want to differentiate between error messages and normal output.

In this code, fprintf is used to print error messages depending on the return value of the kill function. If kill returns 0, it means the signal was successfully sent to the process and the message "kill successful" is printed to stderr. If kill returns -1, it means the signal was not sent successfully, and the message "kill failed" is printed to stderr along with a description of the error using the perror function. Finally, if kill returns any value other than 0 or -1, the message "kill(2) returned wrong value" is printed to stderr.

```

    */
}
int main(int argc, char *argv[])
{
    sighandler_t shandler;

    // sighandler_t signal(int signum, sighandler_t handler);
    shandler = signal(SIGINT, releaseSHM); // SIGINT is for ctrl+c
    if (argc < 2)
    {
        exit(1);
    }
    n = atoi(argv[1]);
    int projid = 1;
    key_t shmkey = ftok("./file.txt", projid);
    shm = shmget(shmkey, n * sizeof(att), IPC_CREAT | 0666);
    if (shm < 0)
    {
        perror("shmget");
        return 1;
    }
    att *shm;
    shm = shmat(shm, NULL, 0);
    int i;
    for (i = 0; i < n; i++)
    {
        (shm + i)->roll = -1;
    }
    while (1)
    {
        sleep(600); // sleep for 600 sec
    }
    shmdt(shm);
}

```

student.c

```
// 2020CSB010 GOURAV KUMAR SHAW

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h> /* for shmget() shmctl() */
#include <sys/shm.h> /* for shmget() shmctl() */
#include <signal.h> /* for signal(2) kill(2) */
#include <time.h>

typedef void (*sighandler_t)(int);
typedef struct att
{
    int roll;
    time_t tm;
} att;

int shmids;

int main(int argc, char *argv[])
{
    int projid=1;
    int n=atoi(argv[1]);
    key_t shmkey=ftok("./file.txt",projid);
    shmids=shmget(shmkey,sizeof(att), 0666);
    printf("%d shmids\n",shmids);
    if(shmids<0)
    {
        perror("shmget");
        return 1;
    }
    att * shm;
    shm=shmat(shmids,NULL,0);
    while(1)
    {
        if(shm->roll==n)
        {
            shm->roll=n; // here n is number of students
            shm->tm=time(NULL);
            break;
        }
        shm=shm+1;
    }
    shmdt(shm);
    return 0;
}
```

```
}
```

Output:

```
gourav @ LAPTOP-868QQ3N0: ~/Assingment_5$ master &
^C ctrl+c is pressed and the students who have logged in are :

Student with Roll Number: 7 attended at February 07, 2023, 11:19:46
Student with Roll Number: 75 attended at February 07, 2023, 11:19:53
Student with Roll Number: 56 attended at February 07, 2023, 11:20:00
Student with Roll Number: 5 attended at February 07, 2023, 11:20:03
Remove shared memory with id = 6.
Killed
```

```
gourav @ LAPTOP-868QQ3N0: ~/Assingment_5$ master &
6 shmid

gourav @ LAPTOP-868QQ3N0: ~/Assingment_5$ master &
6 shmid

gourav @ LAPTOP-868QQ3N0: ~/Assingment_5$ master &
6 shmid

gourav @ LAPTOP-868QQ3N0: ~/Assingment_5$ master &
6 shmid
```