



به نام خدای رنگین کمان

گزارش کار آزمایش ۴م

امیرفاضل کوزه گر کالجی

علیرضا زارع زین آبادی

۹۹۳۱۰۹۹

۹۹۳۱۰۲۲

تمرین اول

در این تمرین، باید یک فضای حافظه اشتراکی ایجاد کنیم. و بتوانیم از آن بخوانیم و بنویسیم

برای این منظور دو فایل `reader.c` , `writer.c` را ایجاد میکنیم
کد آنها به صورت زیر میباشد:

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

#define SHMSIZE 50

char *strrev(char *str);

int main(){

    key_t id = 6000;

    int shm_id = shmget(id, SHMSIZE, 0666);
    if (shm_id<0){
        perror("Reader error: Access denied");
        return 0;
    }

    char *shm = (shm = shmat(shm_id, NULL, 0));

    if(shm=="-1")
    {
        perror("Reader Error: Problem in attaching");
        return 0;
    }

    printf("\nReader Read: ");
    char *s = shm;
    char buff[100];
    int count=0;
    for (s = shm; *s!='\0'; s++)
    {
        char c = *s;
        putchar(c);
        buff[count]=c;
        count++;
    }
}
```

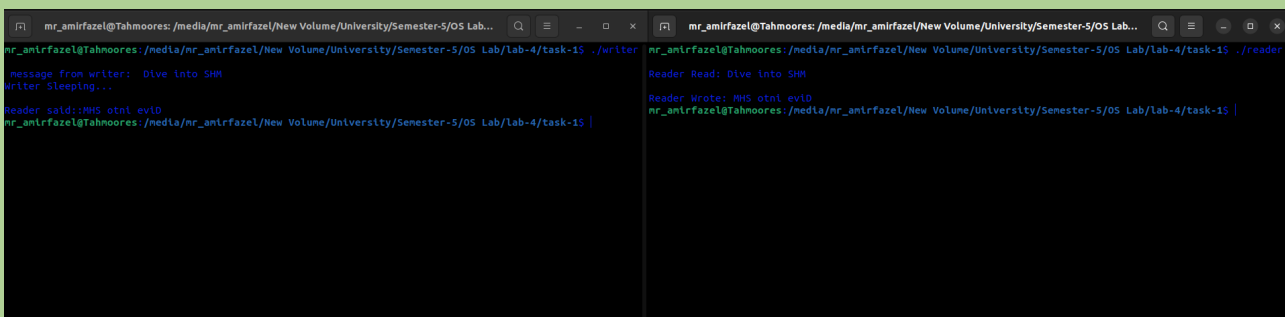
```
42     putchar('\n');
43
44     char *revs = strrev(buff);
45     printf("\nReader Wrote: %s\n",buff);
46
47     s = shm;
48     *s++;
49     for (int i=0;i<strlen(buff);i++)
50     {
51         char c = buff[i];
52         *s++ = c;
53     }
54
55     *s = '\0';
56     *shm = '*';
57
58
59     return 0;
60 }
61
62 char *strrev(char *str)
63 {
64     char *p1, *p2;
65
66     if (! str || ! *str)
67         return str;
68     for (p1 = str, p2 = str + strlen(str) - 1; p2 > p1; ++p1, --p2)
69     {
70         *p1 ^= *p2;
71         *p2 ^= *p1;
72         *p1 ^= *p2;
73     }
74     return str;
75 }
```

reader.c

```
1 #include<stdio.h>
2 #include <sys/ipc.h>
3 #include <sys/shm.h>
4 #include<string.h>
5 #include <stdlib.h>
6 #include <unistd.h>
7
8 #define SHMSIZE 50
9
10 int main (){
11
12     key_t id = 6000;
13
14     int shm_id = shmget(id, SHMSIZE, IPC_CREAT | 0666);
15
16     if (shm_id<0){
17         perror("Writer error, access denied");
18         return 0;
19     }
20
21     char *shm = shmat(shm_id, NULL, 0);
22
23     if (shm=="-1"){
24         perror("Writer error, problem in attaching shared memory");
25         return 0;
26     }
27
28     char msg[100] = "Dive into SHM";
29     char *s = shm;
30     printf("\n message from writer: ");
31     for (int i = 0; i <strlen(msg); i++)
32     {
33         char c = msg[i];
34         putchar(c);
35         *s++ = c;
36     }
37
38     *s = '\0';
39
40     printf("\nWriter Sleeping...\n");
41     while (*shm!='*')
42     {
43         sleep(1);
44     }
45
46     printf("\nReader said:");
47     for (s = shm+1; *s!= '\0'; s++)
48         putchar(*s);
49     putchar('\n');
50
51     return 0;
52 }
53 }
```

writer.c

حال برای اجرای برنامه ها و اطمینان از کارکرد حافظه اشتراکی، دو ترمینال جداگانه همزمان باز کرده و هردو برنامه را اجرا میکنیم. (نویسنده را اول اجرا میکنیم). خواهیم داشت:



```
mr_anfazel@Tahmoore: /media/mr_anfazel/New Volume/University/Semester-5/OS Lab...
mr_anfazel@Tahmoore: /media/mr_anfazel/New Volume/University/Semester-5/OS Lab/Lab-4/task-15 ./writer
Message from writer: Dive into SHM
Writer Sleeping...
Reader said: MMS otnl eviD
mr_anfazel@Tahmoore: /media/mr_anfazel/New Volume/University/Semester-5/OS Lab/Lab-4/task-15 |

mr_anfazel@Tahmoore: /media/mr_anfazel/New Volume/University/Semester-5/OS Lab...
mr_anfazel@Tahmoore: /media/mr_anfazel/New Volume/University/Semester-5/OS Lab/Lab-4/task-15 ./reader
Reader Read: Dive into SHM
Reader Wrote: MMS otnl eviD
mr_anfazel@Tahmoore: /media/mr_anfazel/New Volume/University/Semester-5/OS Lab/Lab-4/task-15 |
```

همانگونه که مشخص است، ابتدا نویسنه یک فضای حافظه اشتراکی ایجاد میکند و در آن یک رشته قرار میدهد. حال هرگاه که برنامه خواننده را اجرا کنیم رشته نوشته شده در SHM را خوانده و آن را چاپ میکند.

تمرین دوم:

در این تمرین باید با socket یک چت روم ایجاد کنیم:
سرور این چت روم به شرح زیر است:
که بعد از اینکه یک کاربر به آن socket متصل شد و یوزرنیم خود را ارسال کرد به آن اجازه چت کردن را میدهد.

```
1
2 import socket
3 from _thread import *
4
5
6 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
8
9 IP_address = '127.0.0.1'
10
11 Port = 8080
12
13 server.bind((IP_address, Port))
14
15 server.listen(100)
16
17 list_of_clients = []
18 list_of_messages = []
19
20 def clientthread(conn, addr):
21     username = conn.recv(2048).decode('utf-8')
22
23     for x in list_of_messages:
24         conn.send(x.encode('utf-8'))
25
26     conn.send('\n--end--/'.encode('utf-8'))
27     print('----<' + username + '> join the chat---')
28     broadcast('----<' + username + '> join the chat---', conn)
```

```
29 while True:
30     try:
31         message = conn.recv(2048).decode('utf-8')
32         if message.__contains__('exit') and len(message) == 5:
33             t = "----<" + username + "> left the chat----"
34             print(t)
35             broadcast(t, conn)
36             add_message(t)
37             remove(conn)
38             return
39         elif message:
40
41             print("<" + username + "> " + message)
42
43
44             message_to_send = "<" + username + "> " + message
45             broadcast(message_to_send, conn)
46             add_message(message_to_send)
47
48         else:
49
50             remove(conn)
51             t = "----<" + username + "> left the chat----"
52             print(t)
53             add_message(t)
54             return
55
56     except:
57         continue
58
59
60 def broadcast(message, connection):
61     for clients in list_of_clients:
62         if clients!=connection:
63             try:
64                 clients.send(message.encode('utf-8'))
65             except:
66                 clients.close()
67
68
69         remove(clients)
70
71 def remove(connection):
72     if connection in list_of_clients:
73         list_of_clients.remove(connection)
74
```

```
74
75 def add_message(message):
76     list_of_messages.append(message)
77
78 while True:
79
80     conn, addr = server.accept()
81
82     list_of_clients.append(conn)
83     start_new_thread(clientthread, (conn, addr))
84
85 conn.close()
86 server.close()
87
```

و کد کلاینت آن به شرح زیر است:


```
1
2 import socket
3 import select
4 import sys
5
6 username = input("username: ")
7 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8 IP_address = '127.0.0.1'
9 Port = 8080
10 server.connect((IP_address, Port))
11 server.send(username.encode('utf-8'))
12
13 history = []
14 while True:
15     x = server.recv(2048).decode('utf-8')
16     if x.__contains__('--end--/'):
17         break
18     history.append(x)
19
20 for x in history:
21     print(x , end='\n\n')
22
23 del history
24
25 while True:
26
27     sockets_list = [sys.stdin, server]
28
29
30
31     read_sockets,write_socket, error_socket = select.select(sockets_list,[],[])
32
33     for socks in read_sockets:
34         if socks == server:
35             message = socks.recv(2048).decode('utf-8')
36             print (message)
37         else:
38             message = sys.stdin.readline()
39             if message.__contains__('exit') and len(message) == 5:
40                 server.send(message.encode('utf-8'))
41                 server.close()
42                 sys.exit(0)
43             server.send(message.encode('utf-8'))
44             sys.stdout.write("<You>")
45             sys.stdout.write(message)
46             sys.stdout.flush()
47     server.close()
48
```

```
python3 sv.py
<ali> salam
----<mamad> join the chat---
<mamad> salam
<fazel> chetorin
khobam
<ali> khobam
----<fazel> left the chat----
----<mamad> left the chat----
----<ali> left the chat----
----<kiarash> join the chat---
<kiarash> salam
<kiarash> chetorin
<kiarash> hahaha
<kiarash> hahaha
----<kiarash> left the chat---
$ python3 cl.py
username: kiarash
<fazel> salam
<ali> salam
<mamad> salam
<fazel> chetorin
<ali> khobam
----<fazel> left the chat----
----<mamad> left the chat----
salam
<You>salam
chetorin
<You>chetorin
hahaha
<You>hahaha
hahaha
<You>hahaha
exit

azare242@ahora-pc ~/Desktop/os/chat_room
$ python3 cl.py
username: ali
----<fazel> join the chat---
<fazel> salam
salam
<You>salam
----<mamad> join the chat---
<mamad> salam
<fazel> chetorin
khobam
<You>khobam
----<fazel> left the chat----
----<mamad> left the chat----
exit
azare242@ahora-pc ~/Desktop/os/chat_room
$

azare242@ahora-pc ~/Desktop/os/chat_room
$ python3 cl.py
username: mamad
salam
<You>salam
<fazel> chetorin
<ali> khobam
----<fazel> left the chat----
exit
azare242@ahora-pc ~/Desktop/os/chat_room
$
```

تمرین سوم:

در این تمرین باید با ایجاد یک خط لوله یا پایپ باعث ایجاد ارتباط بین دو فرایند شویم فرایند ها را به صورت والد-فرزندی ایجاد میکنیم. همچنین رشته ارسال شده توسط فرزند باید کاراکتر های کاپیتال را کوچک و کاراکتر های کوچک را کاپیتال کند. کد برنامه به صورت زیر می باشد:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <sys/types.h>
6 #include <sys/wait.h>
7
8 #define SIZE 200
9
10
11 void reverse_string(char str[]);
12
13 int main(int argc, char* argv[]) {
14     int fd[2];
15     if (pipe(fd) == -1) {
16         return 1;
17     }
18
19     int pid = fork();
20     if (pid == -1) {
21         perror("Some error occurred in creating child process");
22         return -1;
23     }
24
25     if (pid == 0) {
26         close(fd[0]);
27
28         char str[SIZE];
29         printf("Input string: ");
30         fgets(str, SIZE, stdin);
31         str[strlen(str) - 1] = '\0';
32
33         int n = strlen(str) + 1;
34         if (write(fd[1], &n, sizeof(int)) < 0) {
35             return -1;
36         }
37
38         if (write(fd[1], &str, sizeof(char) * n) < 0) {
39             return -1;
40         }
41
42         close(fd[1]);
43     } else {
44         close(fd[1]);
45         char str[SIZE];
46         int n;
47
48         if (read(fd[0], &n, sizeof(int)) < 0) {
49             return -1;
50         }
51         if (read(fd[0], &str, sizeof(int) * n) < 0) {
52             return -1;
53         }
54         reverse_string(str);
55
56         close(fd[0]);
57         wait(NULL);
58     }
59
60     return 0;
61 }
```

```
59
60     return 0;
61 }
62
63
64 void reverse_string(char str[SIZE]){
65     for (size_t i = 0; i < strlen(str); i++)
66     {
67         char c = str[i];
68         if(c >= 'a' && c <= 'z') {
69             str[i] = str[i] - 32;
70             continue;
71         }
72         if(c >= 'A' && c <= 'Z') {
73             str[i] = str[i] + 32;
74             continue;
75         }
76     }
77     printf("%s\n", str);
78
79
80 }
```

حال نوبت به اجرای برنامه و مشاهده نتیجه است. داریم:

```
mr_amirfazel@Tahmoores:/media/mr_amirfazel/New Volume/University/Semester-5/OS Lab/lab-4/task-3$ ./pipe
Input string: HI this IS a TESt message
hI THIS is A test MESSAGE
mr_amirfazel@Tahmoores:/media/mr_amirfazel/New Volume/University/Semester-5/OS Lab/lab-4/task-3$ |
```