

# Department of Computer Science and Engineering

S.G.Shivanirudh , 185001146, Semester IV

17 February 2020

---

## UCS1411 - Operating Systems Laboratory

---

### Exercise – 5-InterProcess Communications using Shared Memory

#### *Objective:*

Develop the following applications that uses interprocess communication concepts using shared memory.

#### *Code:*

Q1.Develop an application for getting a name in parent and convert it into uppercase in child using shared memory.

```
1 #include <sys/ipc.h>
2 #define NULL 0
3 #include <sys/shm.h>
4 #include <sys/types.h>
```

```

5 #include<unistd.h>
6 #include<stdio.h>
7 #include<stdlib.h>
8 #include<string.h>
9 #include <sys/wait.h>
10 #include <stdio_ext.h>
11 #include<ctype.h>
12
13
14 int main(){
15     int pid;
16
17     //Input buffer in parent
18     char *parent_ip;
19     parent_ip=(char*)malloc(100*sizeof(char));
20
21     //Output buffer in child
22     char *child_op;
23     child_op=(char*)malloc(100*sizeof(char));
24
25     //Creating shared memory location
26     int shmемid=shmget(146,100,IPC_CREAT|00666);
27
28     pid=fork();
29
30     //Parent process
31     if(pid>0){
32         printf("\nParent Process:\n");
33
34         //Attaching memory location
35         parent_ip=shmat(shmemid,NULL,0);
36
37         printf("\nEnter a string: \n");scanf(" %[^\n]",
parent_ip);
38         wait(NULL);
39         shmdt(parent_ip);
40     }
41     //Child Process
42     else{
43         sleep(5);
44
45         printf("\nChild Process:\n");
46
47         //Attaching memory location

```

```

49         child_op=shmat(shmemid,NULL,0);
50
51         printf("\nEntered string: %s\n",child_op);
52
53         printf("\nString in upper case: ");
54         int i=0;
55         while(child_op[i]){
56             putchar(toupper(child_op[i]));
57             i++;
58         }
59         printf("\n");
60         shmdt(child_op);
61     }
62     shmctl(shmemid,IPC_RMID,NULL);
63 }

```

### ***Output:***

```

1 Parent Process:
2
3 Enter a string:
4 hello
5
6 Child Process:
7
8 Entered string: hello
9
10 String in upper case: HELLO

```

---

Q2.Develop an client / server application for file transfer using shared memory.

```

1 #include <sys/ipc.h>
2 #define NULL 0
3 #include <sys/shm.h>
4 #include <sys/types.h>
5 #include<unistd.h>
6 #include<stdio.h>
7 #include<stdlib.h>
8 #include<string.h>
9 #include <sys/wait.h>
10 #include <stdio_ext.h>
11 #include<fcntl.h>

```

```

12 #include <ctype.h>
13
14 int main(){
15
16     printf("\nClient: \n");
17     //Creating shared memory location
18     int shmемid=shmget(146,1000,IPC_CREAT|00666);
19
20     //Buffer to read from/write onto the shared memory
    location
21     char *file_read=(char*)calloc(1000,sizeof(char));
22
23     //Attaching to shared memory
24     file_read=shmat(shmemid,NULL,0);
25
26     printf("\nEnter file name: ");
27     scanf(" %s",file_read);
28
29     //Sleeping to get contents of file from server
30     sleep(5);
31     printf("\nFile Contents:\n");
32     printf("\n%s\n",file_read);
33
34     //Detaching from shared memory
35     shmdt(file_read);
36
37     //Deallocating the acquired memory location
38     shmctl(shmemid,IPC_RMID,NULL);
39 }

```

```

1
2 #include <sys/ipc.h>
3 #define NULL 0
4 #include <sys/shm.h>
5 #include <sys/types.h>
6 #include <unistd.h>
7 #include <stdio.h>
8 #include <stdlib.h>
9 #include <string.h>
10 #include <sys/wait.h>
11 #include <stdio_ext.h>
12 #include <fcntl.h>
13 #include <ctype.h>
14
15 int main(){
16

```

```

17     printf("\nServer: \n");
18     //Creating shared memory location
19     int shmемid=shmget(146,1000,IPC_CREAT|00666);
20
21     //Buffer to read from/write on the shared memory
22     char *file_read=(char*)calloc(1000,sizeof(char));
23
24     //Sleeping to accept file name as input from client
25     sleep(2);
26
27     //Attaching to shared memory
28     file_read=shmat(shmemid,NULL,0);
29     printf("\nFile to be read: %s\n",file_read);
30
31     //Opening file to be read
32     int sourcefd=open(file_read,O_RDONLY);
33     if(sourcefd==-1){
34         printf("No source file");
35     }
36     else{
37         //Reading file contents onto the buffer
38         int readfd=read(sourcefd,file_read,100);
39         file_read[readfd]='\0';
40         printf("\nFile read successfully\n");
41         wait(NULL);
42         close(sourcefd);
43         //Detaching buffer from shared memory
44         shmdt(file_read);
45     }
46
47 }

```

### ***Output:***

```

1 Client:
2
3 Enter file name: source.txt
4
5 File Contents:
6
7 asdfgf ;lkjhj
8 shiva
9 sharvan
10 shashu
11 siddharth

```

```
1 Server:
2
3 File to be read: source.txt
4
5 File read successfully
```

---

Q3. Develop an client/server chat application using shared memory.

```
1
2 #include <sys/ipc.h>
3 #define NULL 0
4 #include <sys/shm.h>
5 #include <sys/types.h>
6 #include <unistd.h>
7 #include <stdio.h>
8 #include <stdlib.h>
9 #include <string.h>
10 #include <sys/wait.h>
11 #include <stdio_ext.h>
12 #include <fcntl.h>
13 #include <ctype.h>
14
15 int main(){
16     //Creating shared memory location
17     int shmемid=shmget(146,1000,IPC_CREAT|00666);
18
19     //Buffer to read from/write onto the shared memory
20     location
21     char *file_read=(char*)calloc(1000,sizeof(char));
22
23     //Attaching to shared memory
24     file_read=shmat(shmemid,NULL,0);
25
26     while(strcmp(file_read,"end")!=0){
27
28         //Sleeping to get contents of file from server
29         printf("\nServer:%s\n",file_read);
30         printf("\nYou: ");scanf("%[^\n]",file_read);
31         sleep(5);
32         wait(NULL);
33     }
34     //Detaching from shared memory
35     shmdt(file_read);
```

```

35
36 //Deallocating the acquired memory location
37 shmctl(shmemid,IPC_RMID,NULL);
38 }

1
2 #include <sys/ipc.h>
3 #define NULL 0
4 #include <sys/shm.h>
5 #include <sys/types.h>
6 #include<unistd.h>
7 #include<stdio.h>
8 #include<stdlib.h>
9 #include<string.h>
10 #include <sys/wait.h>
11 #include <stdio_ext.h>
12 #include<fcntl.h>
13 #include<ctype.h>
14
15 int main(){
16 //Creating shared memory location
17 int shmemid=shmget(146,1000,IPC_CREAT|00666);
18
19 //Buffer to read from/write onto the shared memory
location
20 char *file_read=(char*)calloc(1000,sizeof(char));
21
22 //Attaching to shared memory
23 file_read=shmat(shmemid,NULL,0);
24
25 while(strcmp(file_read,"end")!=0){
26
27 //Sleeping to get contents of file from server
28 printf("\nClient:%s\n",file_read);
29 printf("\nYou: ");scanf("%[^\n]",file_read);
30 sleep(5);
31 wait(NULL);
32
33 }
34 //Detaching from shared memory
35 shmdt(file_read);
36
37 //Deallocating the acquired memory location
38 shmctl(shmemid,IPC_RMID,NULL);
39 }

```

### *Output:*

```
1 Client:
2
3 You: hi client
4
5 Client:hi server
6
7 You: hello client
8
9 Client:how r u
10
11 You: how u doinhhh
12
13 Client:ggg
14
15 You: how about u
16
17 Client:im fine
18
19 You: nice

1
2 Server:hi client
3
4 You: hi server
5
6 Server:hello client
7
8 You: how r u
9
10 Server:hhow u doingg
11
12 You: ggg
13
14 Server:how about u
15
16 You: im fine
17
18 Server:nice
19
20 You: end
```

---