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(){
      int pid;
15
16
      //Input buffer in parent
17
      char *parent_ip;
18
      parent_ip=(char*)malloc(100*sizeof(char));
19
20
      //Output buffer in child
21
      char *child_op;
22
      child_op=(char*)malloc(100*sizeof(char));
23
24
      //Creating shared memory location
      int shmemid=shmget(146,100,IPC_CREAT|00666);
26
      pid=fork();
28
      //Parent process
30
      if (pid>0) {
31
           printf("\nParent Process:\n");
32
           //Attaching memory location
34
           parent_ip=shmat(shmemid, NULL,0);
35
36
           printf("\nEnter a string: \n"); scanf(" %[^\n]",
37
     parent_ip);
           wait(NULL);
38
           shmdt(parent_ip);
39
40
41
      //Child Process
42
      else{
           sleep(5);
44
           printf("\nChild Process:\n");
46
           //Attaching memory location
```

```
child_op=shmat(shmemid,NULL,0);
49
          printf("\nEntered string: %s\n",child_op);
51
          printf("\nString in upper case: ");
          int i=0;
           while(child_op[i]){
55
               putchar(toupper(child_op[i]));
56
               i++;
          }
          printf("\n");
           shmdt(child_op);
60
      }
61
      shmctl(shmemid, IPC_RMID, NULL);
62
63 }
  Output:
1 Parent Process:
3 Enter a string:
4 hello
6 Child Process:
8 Entered string: hello
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>
14 int main(){
15
      printf("\nClient: \n");
16
      //Creating shared memory location
      int shmemid=shmget(146,1000,IPC_CREAT|00666);
18
      //Buffer to read from/write onto the shared memory
20
     location
      char *file_read=(char*)calloc(1000, sizeof(char));
2.1
      //Attaching to shared memory
23
      file_read=shmat(shmemid, NULL, 0);
      printf("\nEnter file name: ");
26
      scanf(" %s",file_read);
28
      //Sleeping to get contents of file from server
      sleep(5);
30
      printf("\nFile Contents:\n");
      printf("\n%s\n",file_read);
32
      //Detaching from shared memory
34
      shmdt(file_read);
36
      //Deallocating the acquired memory location
      shmctl(shmemid,IPC_RMID,NULL);
38
39 }
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>
#include <stdio_ext.h>
12 #include < fcntl.h>
13 #include < ctype . h >
15 int main(){
16
```

```
printf("\nServer: \n");
17
      //Creating shared memory location
18
      int shmemid=shmget(146,1000,IPC_CREAT|00666);
19
20
      //Buffer to read from/write on the shared memory
21
      char *file_read=(char*)calloc(1000, sizeof(char));
23
      //Sleeping to accept file name as input from client
24
      sleep(2);
25
      //Attaching to shared memory
27
      file_read=shmat(shmemid, NULL, 0);
      printf("\nFile to be read: %s\n",file_read);
29
30
      //Opening file to be read
31
      int sourcefd=open(file_read,O_RDONLY);
32
      if (sourcefd==-1) {
33
          printf("No source file");
34
      }
35
      else{
36
           //Reading file contents onto the buffer
           int readfd=read(sourcefd,file_read,100);
38
           file_read[readfd] = '\0';
          printf("\nFile read successfully\n");
40
          wait(NULL);
           close(sourcefd);
42
           //Detaching buffer from shared memory
           shmdt(file_read);
44
      }
45
46
47 }
  Output:
1 Client:
3 Enter file name: source.txt
5 File Contents:
7 asdfgf ; lkjhj
8 shiva
9 sharvan
10 shashu
11 siddharth
```

```
Server:

File to be read: source.txt

File read successfully
```

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

```
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>
15 int main(){
      //Creating shared memory location
      int shmemid=shmget(146,1000,IPC_CREAT|00666);
17
      //Buffer to read from/write onto the shared memory
19
     location
      char *file_read=(char*)calloc(1000, sizeof(char));
20
21
      //Attaching to shared memory
      file_read=shmat(shmemid, NULL, 0);
23
      while (strcmp(file_read, "end")!=0){
25
           //Sleeping to get contents of file from server
          printf("\nServer:%s\n",file_read);
          printf("\nYou: ");scanf(" %[^\n]",file_read);
29
           sleep(5);
           wait(NULL);
31
      }
      //Detaching from shared memory
33
      shmdt(file_read);
```

```
35
      //Deallocating the acquired memory location
36
      shmctl(shmemid,IPC_RMID,NULL);
37
38 }
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 >
15 int main(){
      //Creating shared memory location
      int shmemid=shmget(146,1000,IPC_CREAT|00666);
17
18
      //Buffer to read from/write onto the shared memory
19
     location
      char *file_read=(char*)calloc(1000, sizeof(char));
20
21
      //Attaching to shared memory
22
      file_read=shmat(shmemid, NULL, 0);
24
      while (strcmp(file_read, "end")!=0) {
26
           //Sleeping to get contents of file from server
           printf("\nClient:%s\n",file_read);
28
           printf("\nYou: ");scanf(" %[^\n]",file_read);
           sleep(5);
30
           wait(NULL);
31
32
33
      //Detaching from shared memory
34
      shmdt(file_read);
35
36
      //Deallocating the acquired memory location
37
      shmctl(shmemid, IPC_RMID, NULL);
38
39 }
```

Output:

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