UCS 1411 - Operating Systems Lab Exercise 5 - Inter Process Communication Using Shared memory

Mahesh Bharadwaj K - 185001089

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

Program

```
#include<sys/ipc.h>
#define NULL 0
#include<sys/shm.h>
#include<sys/types.h>
#include<unistd.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<sys/wait.h>
void strupr(char *a)
{
    for(int i = 0; a[i] != '\0'; i++)
        a[i] -= (a[i] >= 'a' && a[i] <= 'z') ? 32 : 0;
}
int main()
{
    int pid, id;
    char *a, *b;
    id = shmget(111, 50, IPC_CREAT | 00666);
    pid = fork();
    if(pid > 0)
    {
        printf("Parent Process\n");
        a = shmat(id, NULL, 0);
        a[0] = '\0';
        printf("Enter a string: ");
        scanf("%[^\n]", a);
        wait(NULL);
        shmdt(a);
    }
    else
    {
        b = shmat(id, NULL, 0);
        while(b[0] == '\0');
        printf("\nChild Process:\n");
        strupr(b);
        printf("Uppercase: %s\n", b);
        shmdt(b);
    }
    shmctl(id, IPC_RMID, NULL);
}
```

Output

Parent Process
Enter a string: This is a test string

Child Process:
Uppercase: THIS IS A TEST STRING

---

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

Server Program

```
#include<sys/ipc.h>
#define NULL 0
#include<sys/shm.h>
#include<sys/types.h>
#include<unistd.h>
#include<fcntl.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<sys/wait.h>
int main()
{
    char *a;
    int fd;
    int id = shmget(101, 50, IPC_CREAT);
    a = shmat(id, NULL, 0);
    while(a[0] == '\0');
    fd = open(a, O_RDONLY);
    a[0] = '\0';
    if(fd < 0) {
        strcpy(a, "Does not exist!\n");
        shmdt(a);
        exit(0);
    }
    char r; int i = 0;
    while((read(fd, &r, 1)))
    {
        a[i] = r;
        i++;
    }
    printf("File Read!\n");
    shmdt(a);
    shmctl(id, IPC_RMID, NULL);
}
```

Client Program

```
#include<sys/ipc.h>
#define NULL 0
#include<sys/shm.h>
#include<sys/types.h>
#include<unistd.h>
#include<stdio.h>
#include<string.h>
```

```c
#include<stdlib.h>
#include<fcntl.h>
#include<sys/wait.h>
int main()
{
    int id;
    char *a;
    char filename[100];
    id = shmget(101, 50, IPC_CREAT | 00666);
    a = shmat(id, NULL, 0);
    a[0] = '\0';
    printf("Enter file name: ");
    scanf("%s",filename);
    strcpy(a,filename);
    strcat(filename,"-copy");
    sleep(3);
    while(a[0] == '\0');

    if(strcmp(a,"Does not exit!\n")){
        printf("%s", a);
        return 1;
    }

    int fd = open(filename,O_WRONLY | O_CREAT,S_IRWXU);
    printf("%s", a);
    write(fd,a,strlen(a));
    shmdt(a);
    shmctl(id, IPC_RMID, NULL);
    close(fd);
    return 0;
}
```

Output

Enter file name: file
Client Terminal
Sample file to test client server file transer.
Contains many lines.
Hello world
ABCD 123

Server Terminal
File Read!

---

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

User Program

```c
#include<sys/ipc.h>
#define NULL 0
#include<sys/shm.h>
#include<sys/types.h>
#include<unistd.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
```

```c
#include<sys/wait.h>
int main()
{
    char *a;
    int id = shmget(135, 500, IPC_CREAT | 00666);
    a = shmat(id, NULL, 0);
    while(1) {
        while(a[0] == '\0');
        printf("User: %s\n", a);
        if(strcmp(a,"bye") == 0) break;
        a[0] = '\0';
        printf("You: ");
        scanf("%[^\n]",a);
        getchar();
        if(strcmp(a,"bye") == 0) break;
        sleep(1);
    }
    printf("Connection Ended!\n");
    shmdt(a);
    shmctl(id, IPC_RMID, NULL);
}
```

Client Program

```c
#include<sys/ipc.h>
#define NULL 0
#include<sys/shm.h>
#include<sys/types.h>
#include<unistd.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<sys/wait.h>
int main()
{
    int id;
    char *a;
    id = shmget(135, 500, IPC_CREAT | 00666);
    a = shmat(id, NULL, 0);
    while(1) {
        a[0] = '\0';
        printf("You: ");
        scanf("%[^\n]",a);
        getchar();
        if(strcmp(a,"bye") == 0) break;
        sleep(1);
        while(a[0] == '\0');
        printf("Server: %s\n", a);
        if(strcmp(a,"bye") == 0) break;
    }
    printf("Connection Ended!\n");
    shmdt(a);
    shmctl(id, IPC_RMID, NULL);
}
```

Output

User Terminal
You: Hi
Server: Hello
You: How are you?

Server: Good and you?
You: Fine! Take care
Server: You too
You: bye
Connection Ended!

Server Terminal
User: Hi
You: Hello
User: How are you?
You: Good and you?
User: Fine! Take care
You: You too
User: bye
Connection Ended!