Operating Systems Lab

IPC - Shared Memory

By: Muhammad Ahsan

1. Shared Memory

- Region of Memory that is shared by cooperating processes
- Processes exchange Data by reading/writing to the shared region

Example 1:

shareMemory_server.c

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/ipc.h>
#include<sys/shm.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
#define SHMSZ 1024
void main()
{
        key_t key = 12345;
        int shmid = shmget(key, SHMSZ, 0777 | IPC_CREAT );
        char *shm = shmat(shmid, NULL, 0 );
        memcpy(shm, "Hello Pakistan\n", SHMSZ );
}
```

sharedMemory_client.c

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/ipc.h>
#include<sys/shm.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
#define SHMSZ 1024
void main()
{
        key_t key = 12345;
        int shmid = shmget(key, SHMSZ, 0777 | IPC_EXCL);
        char *shm = shmat(shmid, NULL, 0);
```

```
        printf("shm = %s\n", shm);
        shmdt(shm);
        shmctl(shmid, IPC_RMID, NULL);
}
```

Example 2:

sharedMemory.c

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/ipc.h>
#include<sys/shm.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
#define SHMSZ 1024
void main()
{
        key_t k = 12345;
        int x = fork();
        if(x > 0)
        {
                //parent - Server Side - write
                int shmid ;
                // shared memory id
                char *shm; // char pointer that points to the shm
                char *c;
                // creating shared memory and checking for error
                shmid = shmget (k, 1024, 0777 | IPC_CREAT);
                if (shmid < 0 ){
                        perror("shmget");
                        exit(1);
                } // end if
                // attaching data to the shared mem and checking for error
                shm = shmat(shmid, NULL, 0);
                if (shm == (char *) -1){
                        perror("shmat");
                        exit(1);
                } // end if
                memcpy(shm,"Hello world", 11);
        } // end if x > 0
        else if ( x==0 )
        {
```

```
                //child - client side - read
                int shmid ;
                // shared memory id
                char *shm; // char pointer that points to the shm
                // to creating shared memory and checking for error
                shmid = shmget (k, 1024, 0777 | IPC_EXCL);
                if (shmid < 0 ){
                        perror("shmget");
                        exit(1);
                } // end if
                // attaching data to the shm and checking for error
                shm = shmat(shmid, NULL, 0);
                if (shm == (char *) -1){
                        perror("shmat");
                        exit(1);
                } // end if
                printf("shm = %s \n",shm);
                // to de-attach data from the shared mem
                shmdt(shm);
                // destroying shared memory
                shmctl(shmid, IPC_RMID, NULL);
        } // end else if
}
```

Example 3:

shm_server.c

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdio.h>
#define SHMSZ 27
int main()
{
        char c;
        int shmid;
        key_t key;
        char *shm, *s;
        key = 5678;
        if ((shmid = shmget(key, SHMSZ, IPC_CREAT | 0666)) < 0) {
                perror("shmget");
                exit(1);
        }
        if ((shm = shmat(shmid, NULL, 0)) == (char *) -1) {
                perror("shmat");
                exit(1);
        }
        s = shm;
        for (c = 'a'; c <= 'z'; c++)
```

```
                *s++ = c;
        *s = NULL;
        while (*shm != '*'){
                sleep(1);
        }
        exit(0);
}
```

shm_client.c

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdio.h>
#define SHMSZ 27
void main()
{
        int shmid;
        key_t key;
        char *shm, *s;
        key = 5678;
        if ((shmid = shmget(key, SHMSZ, 0666)) < 0)
        {
                perror("shmget");
                exit(1);
        }
        if ((shm = shmat(shmid, NULL, 0)) == (char *) -1) {
                perror("shmat");
                exit(1);
        }
        for (s = shm; *s != NULL; s++){
                putchar(*s);
                putchar('\n');
        }
        *shm = '*';
        shmdt(shm);
        shmctl(shmid,IPC_RMID,NULL);
        exit(0);
}
```