

```

//IPC using shared
memory

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

#define SHMSZ      27      // give the size for shared memory segment as needed.

int main()
{
    char c;
    int shmid,i;
    key_t key;
    char *shm, *s;

    /*
     * We'll name our shared memory segment
     * "5678".
     */
    key = 1234;

    /*
     * Create the segment.
     */
    if ((shmid = shmget(key, SHMSZ, IPC_CREAT | 0666)) < 0) {
        perror("shmget");
        //exit(1);
    }

    /*
     * Now we attach the segment to our data space.
     */
    if ((shm = shmat(shmid, NULL, 0)) == (char *) -1) {
        perror("shmat");
        //exit(1);
    }

    /*
     * Now put some things into the memory for the
     * other process to read.
     */
    s = shm;
    printf("Enter the message on shared memory for client");
    //scanf("%s",s);
    gets(s);

    /*
     * Finally, we wait until the other process
     * changes the first character of our memory
     * to '*', indicating that it has read what
     * we put there.
     */
    while (*shm != '*')
        sleep(10);

    /* detach the memory segment from address space of process*/
    i = shmdt(shm);
    if(i == -1) {
        perror("shmop: shmdt failed");
    }
}

```

```

    }
else
{
printf("segment detached successfully");
}

//exit(0);
}

//Output
[root@localhost root]# ./client
Enter the message on shared memory for client Hello
segment detached successfully[root@localhost root]#

```

SERVER

```

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

#define SHMSZ      27

int main()
{
    int shmid,i;
    key_t key;
    char *shm, *s;

    /*
     * We need to get the segment named
     * "5678", created by the server.
     */
    key = 1234;

    /*
     * Locate the segment.
     */
    if ((shmid=shmget(key,SHMSZ,0666)) < 0)
    {
        perror("shmget");

        // exit(1);
    }

    /*
     * Now we attach the segment to our data space.
     */
    if ((shm = shmat(shmid, NULL, 0)) == (char *) -1) {
        perror("shmat");
    }

    printf("\nattached successfully\n \n");
    /*
     * Now read what the server put in the shared memory segment.

```

```

    */
    for (s=shm; *s!=NULL; s++)
        // printf("%c",s);
        putchar(*s);
        putchar(T\nT);

    /*
     * Finally, change the first character of the
     * segment to T*T, indicating we have read
     * the segment.
     */
    *shm = T*T;

    /* detach the segment using shmdt function from address space of process*/
    i = shmdt(shm);
    if(i == -1) {
        perror("shmop: shmdt failed");
    }
    else
    {

printf("segment detached successfully");
    }

        //exit(0);
    }

//output
[root@localhost root]# gcc server.c -o server
server.c: In function :

[root@localhost root]# gcc client.c -o client[root@localhost
root]# ./server

attached successfully

Hello
segment detached successfully[root@localhost root]#

```

