/\* 10 Inter Process Communication

Problem Statement: B. Inter-process Communication using Shared Memory using System V. Application to demonstrate:

Client and Server Programs in which server process creates a shared memory segment and writes the

message to the shared memory segment. Client process reads the message from the shared memory

segment and displays it to the screen.

Client.c \*/

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 27

void die(char \*s)

{

perror(s);

exit(1);

}

int main()

{

int shmid;

key\_t key;

char \*shm, \*s;

key = 5679;

if ((shmid = shmget(key, MAXSIZE, 0666)) < 0)

die("shmget");

if ((shm = shmat(shmid, NULL, 0)) == (char \*) -1)

die("shmat");

printf("\n\nClient started reading\n\n");

for (s = shm; \*s != '\0'; s++)

{

putchar(\*s);

printf("\t");

}

putchar('\n');

\*shm = '\*';

printf("\n\nClient Terminated!!!\n\n");

exit(0); }

/\* server.c\*/

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 27

void die(char \*s)

{

perror(s);

exit(1);

}

int main()

{

char c;

int shmid;

key\_t key;

char \*shm, \*s;

key = 5679;

if ((shmid = shmget(key, MAXSIZE, IPC\_CREAT | 0666)) < 0)

die("shmget");

if ((shm = shmat(shmid, NULL, 0)) == (char \*) -1)

die("shmat");

s = shm;

for (c = 'a'; c <= 'z'; c++)

\*s++ = c;

printf("\n\nServer is Ready!!\n\nNow data is available on shared memory!!\n");

/\*

\* Waiting for other process

\*/

while (\*shm != '\*')

sleep(1);

printf("\n\nServer Terminated!!!\n\n");

exit(0);

}