SHARED MEMORY:

#include<stdio.h>

#include<sys/ipc.h>

#include<sys/types.h>

#include<string.h>

#include<errno.h>

#include<stdlib.h>

#define BUF\_SIZE 1024

#define SHM\_KEY 0x1234

struct shmseg {

int cnt;

int complete;

char buf[BUF\_SIZE];

};

int main(int argc, char \*argv[]) {

int shmid;

struct shmseg \*shmp;

shmid = shmget(SHM\_KEY, sizeof(struct shmseg), 0644|IPC\_CREAT);

if (shmid == -1) {

perror("Shared memory");

return 1;

}

// Attach to the segment to get a pointer to it.

shmp = shmat(shmid, NULL, 0);

if (shmp == (void \*) -1) {

perror("Shared memory attach");

return 1;

}

/\* Transfer blocks of data from shared memory to stdout\*/

while (shmp->complete != 1) {

printf("segment contains : \n\"%s\"\n", shmp->buf);

if (shmp->cnt == -1) {

perror("read");

return 1;

}

printf("Reading Process: Shared Memory: Read %d bytes\n", shmp->cnt);

sleep(3);

}

printf("Reading Process: Reading Done, Detaching Shared Memory\n");

if (shmdt(shmp) == -1) {

perror("shmdt");

return 1;

}

printf("Reading Process: Complete\n");

return 0;

}

OUTPUT:

Writing Process: Shared Memory Write: Wrote 1023 bytes

Writing Process: Shared Memory Write: Wrote 1023 bytes

Writing Process: Shared Memory Write: Wrote 1023 bytes

Writing Process: Shared Memory Write: Wrote 1023 bytes

Writing Process: Shared Memory Write: Wrote 1023 bytes

Writing Process: Wrote 5 times

Writing Process: Complete