**EXPERIMENT 9**

9. Illustrate the concept of inter-process communication using shared memory with a C program.

#include <stdio.h>

#include <stdlib.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <unistd.h>

#include <string.h>

int main() {

key\_t key = ftok("shmfile", 65); // Generate unique key

int shmid = shmget(key, 1024, 0666|IPC\_CREAT); // Create shared memory segment

if (shmid == -1) {

perror("shmget failed");

exit(1);

}

pid\_t pid = fork(); // Create a child process

if (pid < 0) {

perror("Fork failed");

exit(1);

} else if (pid == 0) {

sleep(2); // Wait for parent to write

char \*str = (char\*) shmat(shmid, NULL, 0);

if (str == (char\*)(-1)) {

perror("shmat failed in child");

exit(1);

}

printf("Child reads from shared memory: \"%s\"\n", str);

shmdt(str); // Detach

} else {

char \*str = (char\*) shmat(shmid, NULL, 0);

if (str == (char\*)(-1)) {

perror("shmat failed in parent");

exit(1);

}

strcpy(str, "Hello from Parent to Child using Shared Memory!");

printf("Parent writes to shared memory: \"%s\"\n", str);

shmdt(str);

wait(NULL);

shmctl(shmid, IPC\_RMID, NULL);

}

return 0;

}

SAMPLE OUTPUT:

Parent writes to shared memory: "Hello from Parent to Child using Shared Memory!"

Child reads from shared memory: "Hello from Parent to Child using Shared Memory!"