INTER PROCESS COMMUNICATION USING SHARED MEMORY

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <unistd.h>
#include <sys/wait.h>
#define SHM_SIZE 1024
int main() {
    key_t key;
    int shmid;
    char *str;
    key = ftok("shmfile", 65);
    shmid = shmget(key, SHM_SIZE, 0666 | IPC_CREAT);
    if (shmid == -1) {
        perror("shmget");
        exit(1);
    }
    pid_t pid = fork();
    if (pid < 0) {
        perror("fork");
        exit(1);
    }
    if (pid > 0) {
        str = (char *) shmat(shmid, NULL, 0);
        if (str == (char *) -1) {
            perror("shmat");
            exit(1);
        }
        strcpy(str, "Hello from Parent Process using Shared Memory!");
        printf("Parent: Data written to shared memory.\n");
        shmdt(str);
        wait(NULL);
        shmctl(shmid, IPC_RMID, NULL);
    } else {
        sleep(1); // Ensure parent writes first
        str = (char *) shmat(shmid, NULL, 0);
        if (str == (char *) -1) {
            perror("shmat");
            exit(1);
        }
        printf("Child: Data read from shared memory -> %s\n", str);
        shmdt(str);
    }
    return 0;
}
```