Writer Process (writer.c)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#define SHM\_SIZE 1024 // Size of the shared memory segment

int main() {

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

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

if (shmid == -1) {

perror("shmget failed");

exit(1);

}

char \*str = (char \*)shmat(shmid, (void \*)0, 0); // Attach to the shared memory

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

perror("shmat failed");

exit(1);

}

printf("Write Data: ");

fgets(str, SHM\_SIZE, stdin); // Get input from the user and write to shared memory

printf("Data written to shared memory: %s\n", str);

if (shmdt(str) == -1) { // Detach from shared memory

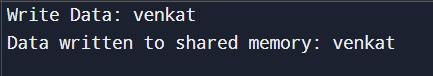
perror("shmdt failed");

exit(1);

}

return 0;

}



Reader Process (reader.c)

#include <stdio.h>

#include <stdlib.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#define SHM\_SIZE 1024 // Size of the shared memory segment

int main() {

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

int shmid = shmget(key, SHM\_SIZE, 0666); // Locate the shared memory segment

if (shmid == -1) {

perror("shmget failed");

exit(1);

}

char \*str = (char \*)shmat(shmid, (void \*)0, 0); // Attach to the shared memory

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

perror("shmat failed");

exit(1);

}

printf("Data read from shared memory: %s\n", str);

if (shmdt(str) == -1) { // Detach from shared memory

perror("shmdt failed");

exit(1);

}

if (shmctl(shmid, IPC\_RMID, NULL) == -1) { // Destroy the shared memory segment

perror("shmctl failed");

exit(1);

}

return 0;

}

