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

```
#include <stdio.h>
#include <sys/mman.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>

int main() {
    const char *name = "/my_shared_mem"; // shared memory name
    const size_t SIZE = 4096;            // size of shared memory

    // Create shared memory object
    int shm_fd = shm_open(name, O_CREAT | O_RDWR, 0666);
    if(shm_fd == -1) {
        perror("shm_open");
        return 1;
    }

    // Configure size
    ftruncate(shm_fd, SIZE);

    // Map shared memory
    char *ptr = mmap(0, SIZE, PROT_READ | PROT_WRITE, MAP_SHARED,
shm_fd, 0);
    if(ptr == MAP_FAILED) {
        perror("mmap");
        return 1;
    }

    pid_t pid = fork();

    if(pid == 0) {
        // Child process writes to shared memory
        sprintf(ptr, "Hello from Child Process (PID=%d)", getpid());
        printf("Child wrote message and exits.\n");
        return 0;
    } else {
        // Parent waits for child to complete
        wait(NULL);
        printf("Parent reads from shared memory: '%s'\n", ptr);

        // Cleanup
        munmap(ptr, SIZE);
        shm_unlink(name);
    }

    return 0;
}
```

OUPUT:

```
Child wrote message and exits.
Parent reads from shared memory: 'Hello from Child Process (PID=12345)'
```