**Problem Statement: Inter-Process Communication (IPC) using Pipes, Shared Memory, and Message Queues**

**Design and implement efficient and reliable inter-process communication (IPC) mechanisms using pipes, shared memory, and message queues in C to facilitate data exchange and synchronization between multiple processes within a single system.**

**Specific Requirements:**

**Pipe: Create and manage unidirectional and bidirectional pipes for simple data transfer between related processes.**

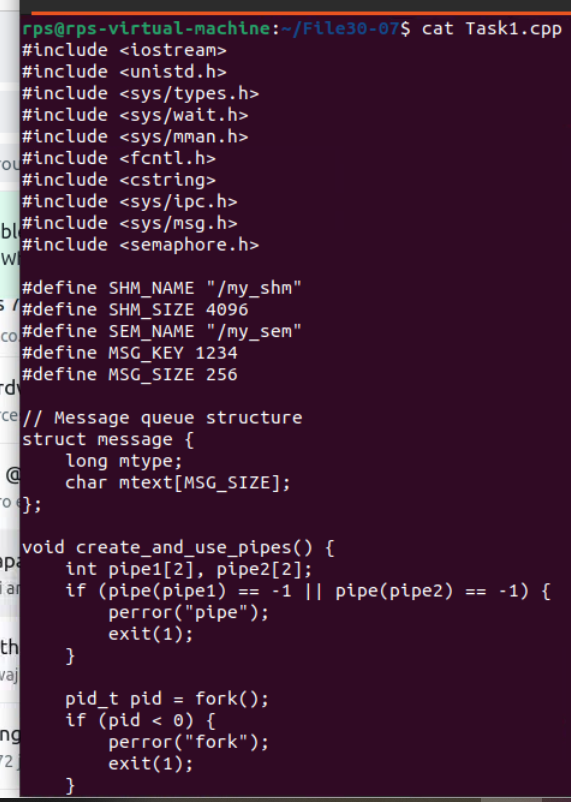
**Shared Memory: Allocate and manage shared memory segments for efficient data sharing between multiple processes.**

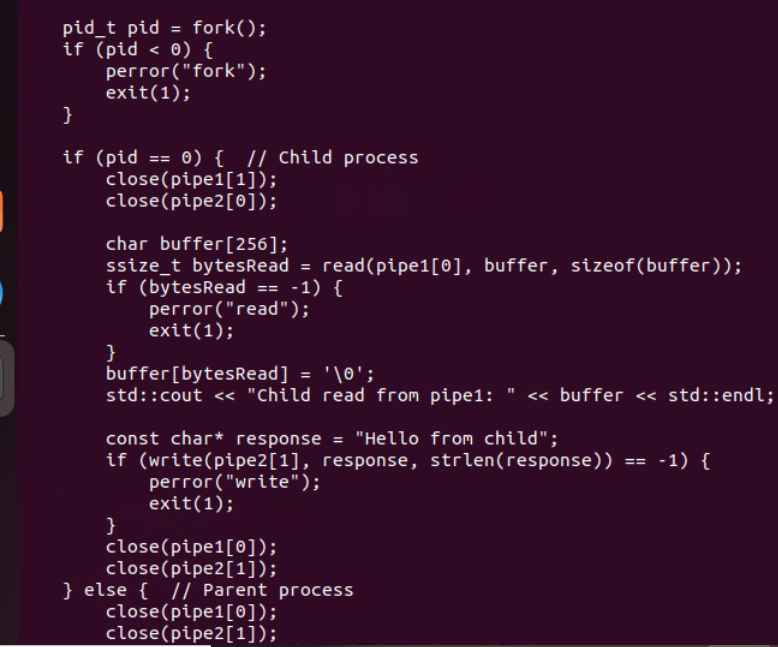
**Message Queues: Create and utilize message queues for asynchronous communication and data exchange with message prioritization.**

**Synchronization: Implement appropriate synchronization mechanisms (e.g., semaphores, mutexes) to coordinate access to shared resources and prevent race conditions.**

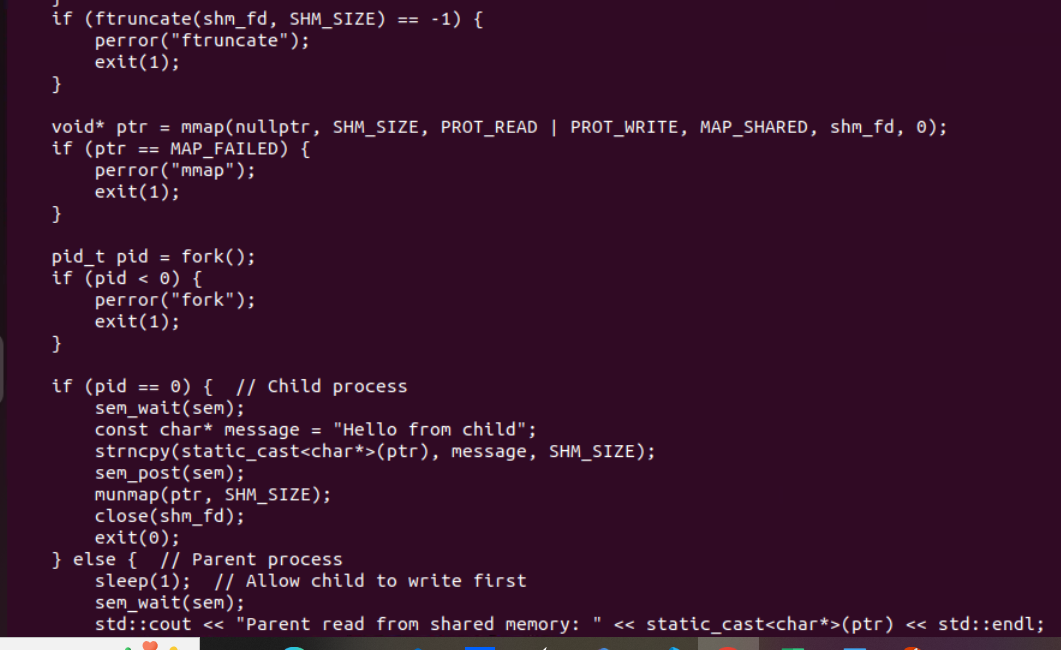
**Error Handling: Incorporate robust error handling to manage potential IPC failures and resource leaks.**

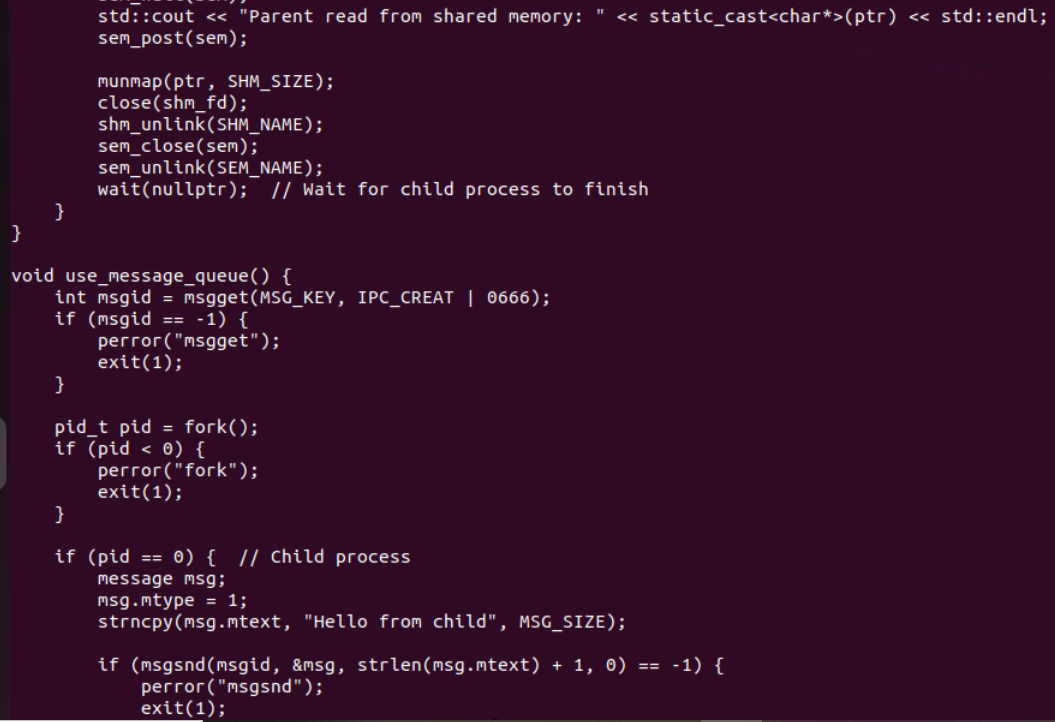
**CODE:**

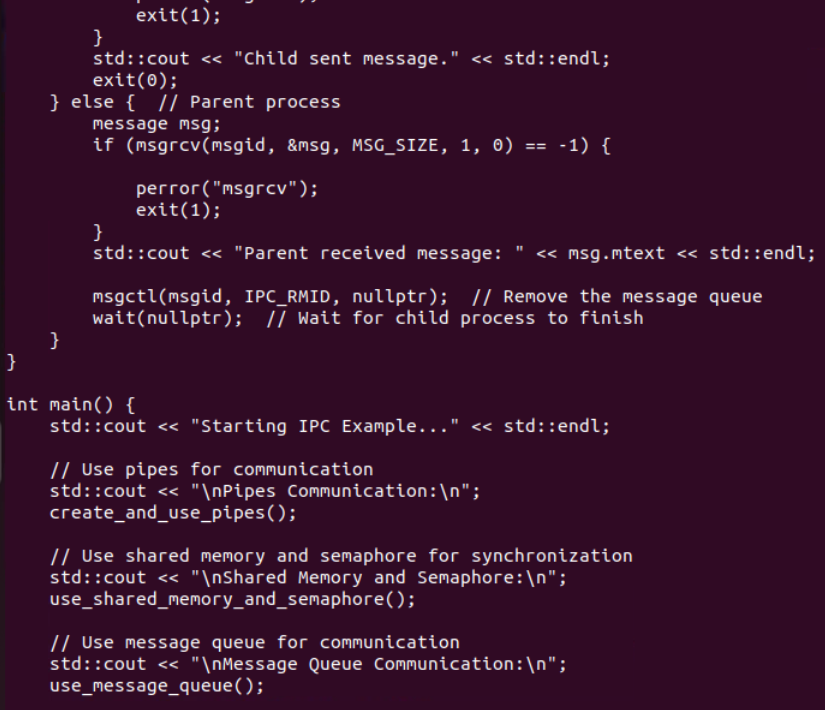
****

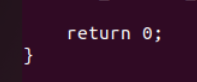
****

****

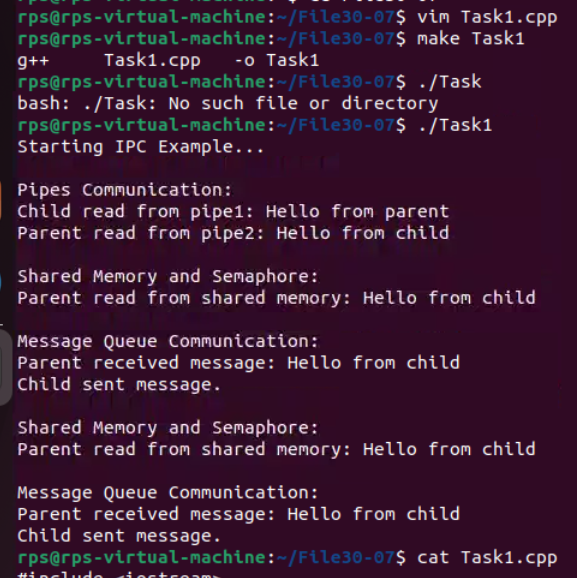
****

****

****

****

**OUTPUT:**

****