Shared Memory Communication between Parent and Child Processes

This program demonstrates shared memory communication between a parent process and a child process using the System V IPC mechanism. The parent process sends input text to the child process through shared memory, and the child process processes the input and returns suggestions back to the parent process.

Compilation and Execution

Compile Both the child.cpp file and textwriter.cpp file and run only textwriter.exe

- g++ child.cpp -o child
- g++ textwriter.cpp -o parent
- ./parent

Input Text:

Enter the text when prompted by the parent process in a single line.

Shared Memory Communication:

The parent process writes the input text to shared memory, and the child process reads from the shared memory, processes the input, and writes the suggestions back to shared memory.

Output:

The parent process displays suggestions received from the child process.

Key Components

- **Parent Process**:
  - Reads input text from the user.
  - Writes the input text to shared memory.
  - Waits for the child process to finish.
  - Reads suggestions from shared memory and displays them.
- **Child Process**:
  - Reads input text from shared memory.
  - Processes the input (e.g., spell check).
  - Write suggestions back to shared memory.
  - Exits.

Shared Memory Implementation

- Shared memory is established using the System V IPC mechanism.
- The `shmget()` function is used to create a shared memory segment.
- The `shmat()` function is used to attach the shared memory segment to the process's address space.
- The parent and child processes use the shared memory segment to exchange data.