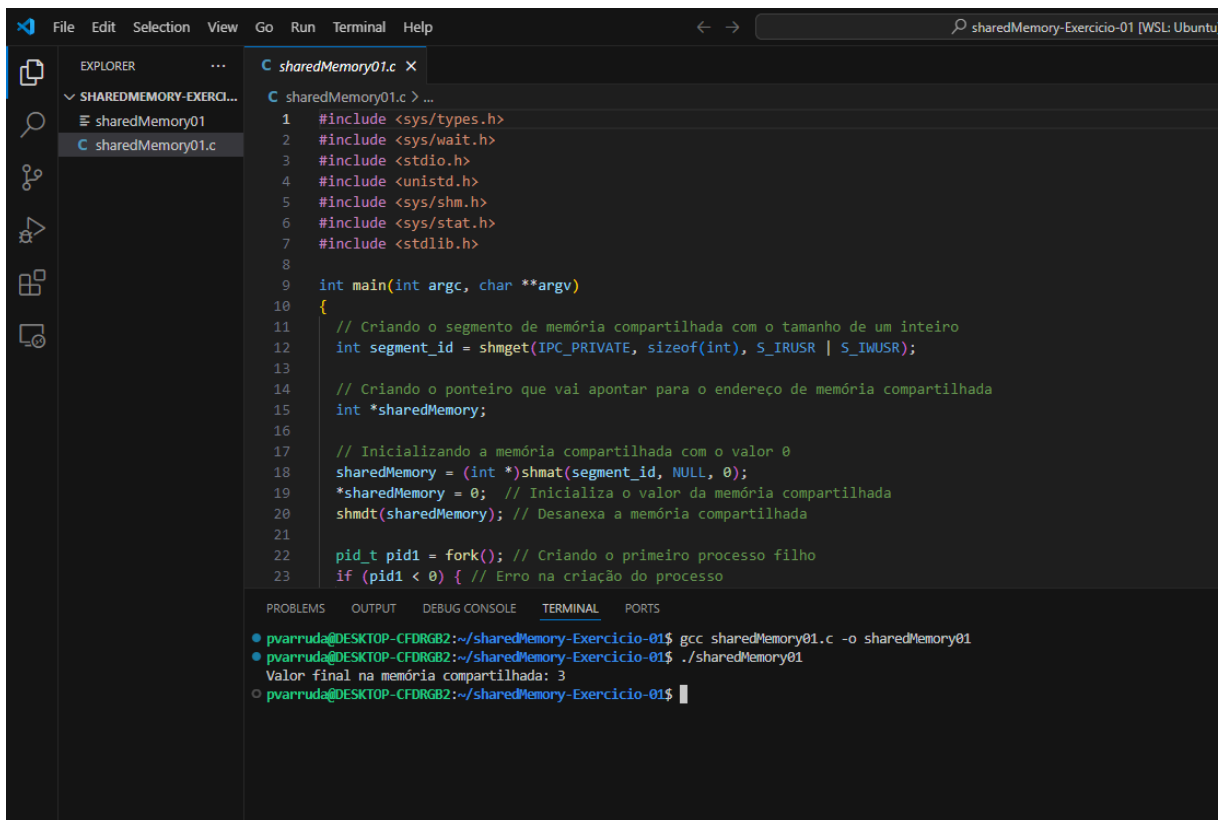


EXPLICAÇÃO BREVE EXERCÍCIO 01

- O segmento de memória compartilhada foi alterado para armazenar um inteiro em vez de uma string.
- Dois processos filhos são criados:
 - O primeiro filho incrementa o valor da memória compartilhada em 1.
 - O segundo filho incrementa o valor da memória compartilhada em 2.
- O processo pai espera que ambos os filhos terminem (wait(NULL) duas vezes), e então imprime o valor final armazenado na memória compartilhada.



The image shows a Visual Studio Code editor window with the file 'sharedMemory01.c' open. The code is in C and implements a shared memory exercise. It includes headers for `<sys/types.h>`, `<sys/wait.h>`, `<stdio.h>`, `<unistd.h>`, `<sys/shm.h>`, `<sys/stat.h>`, and `<stdlib.h>`. The `main` function creates a shared memory segment with `shmget(IPC_PRIVATE, sizeof(int), S_IRUSR | S_IWUSR)`, attaches it with `shmat(segment_id, NULL, 0)`, and initializes it to 0. It then forks two child processes. The first child increments the shared memory value by 1, and the second child increments it by 2. The parent process waits for both children to finish using `wait(NULL)` and then prints the final value of the shared memory.

```
1 #include <sys/types.h>
2 #include <sys/wait.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 #include <sys/shm.h>
6 #include <sys/stat.h>
7 #include <stdlib.h>
8
9 int main(int argc, char **argv)
10 {
11     // Criando o segmento de memória compartilhada com o tamanho de um inteiro
12     int segment_id = shmget(IPC_PRIVATE, sizeof(int), S_IRUSR | S_IWUSR);
13
14     // Criando o ponteiro que vai apontar para o endereço de memória compartilhada
15     int *sharedMemory;
16
17     // Inicializando a memória compartilhada com o valor 0
18     sharedMemory = (int *)shmat(segment_id, NULL, 0);
19     *sharedMemory = 0; // Inicializa o valor da memória compartilhada
20     shmdt(sharedMemory); // Desanexa a memória compartilhada
21
22     pid_t pid1 = fork(); // Criando o primeiro processo filho
23     if (pid1 < 0) { // Erro na criação do processo
```

The terminal output shows the compilation and execution of the program:

```
pvarruda@DESKTOP-CFDRGB2:~/sharedMemory-Exercicio-01$ gcc sharedMemory01.c -o sharedMemory01
pvarruda@DESKTOP-CFDRGB2:~/sharedMemory-Exercicio-01$ ./sharedMemory01
Valor final na memória compartilhada: 3
pvarruda@DESKTOP-CFDRGB2:~/sharedMemory-Exercicio-01$
```