

EXPLICAÇÃO BREVE EXERCÍCIO 02

- Um segmento de memória compartilhada é criado com o tamanho necessário para armazenar n inteiros (onde n é o número de elementos da sequência de Fibonacci que queremos gerar).
- O ponteiro `shared_memory` é usado para acessar essa memória compartilhada.

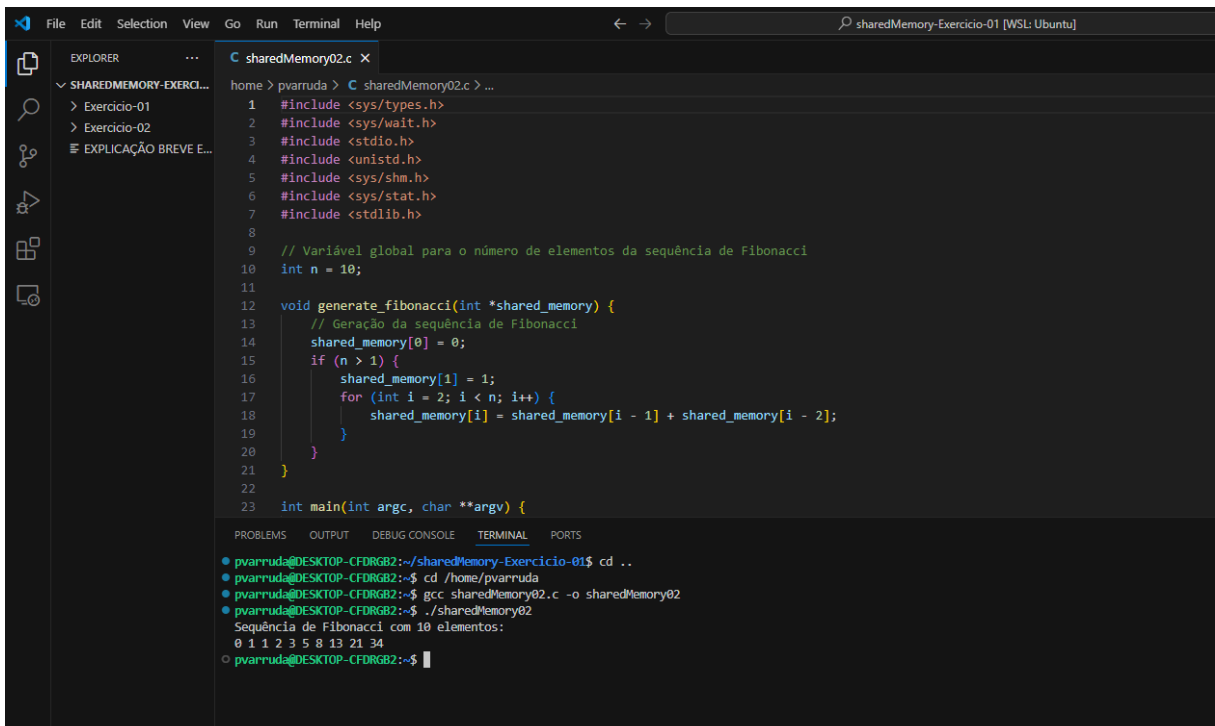
Processo filho:

- O processo filho gera a sequência de Fibonacci e armazena cada valor na memória compartilhada. A função `generate_fibonacci` é chamada para isso.
- Após gerar a sequência, o processo filho desanexa a memória compartilhada e termina sua execução.

Processo pai:

- O processo pai espera o filho terminar (`wait(NULL)`), e então imprime a sequência de Fibonacci que foi armazenada na memória compartilhada.
- Depois de imprimir, o pai desanexa e remove a memória compartilhada.

Teste com 10 elementos



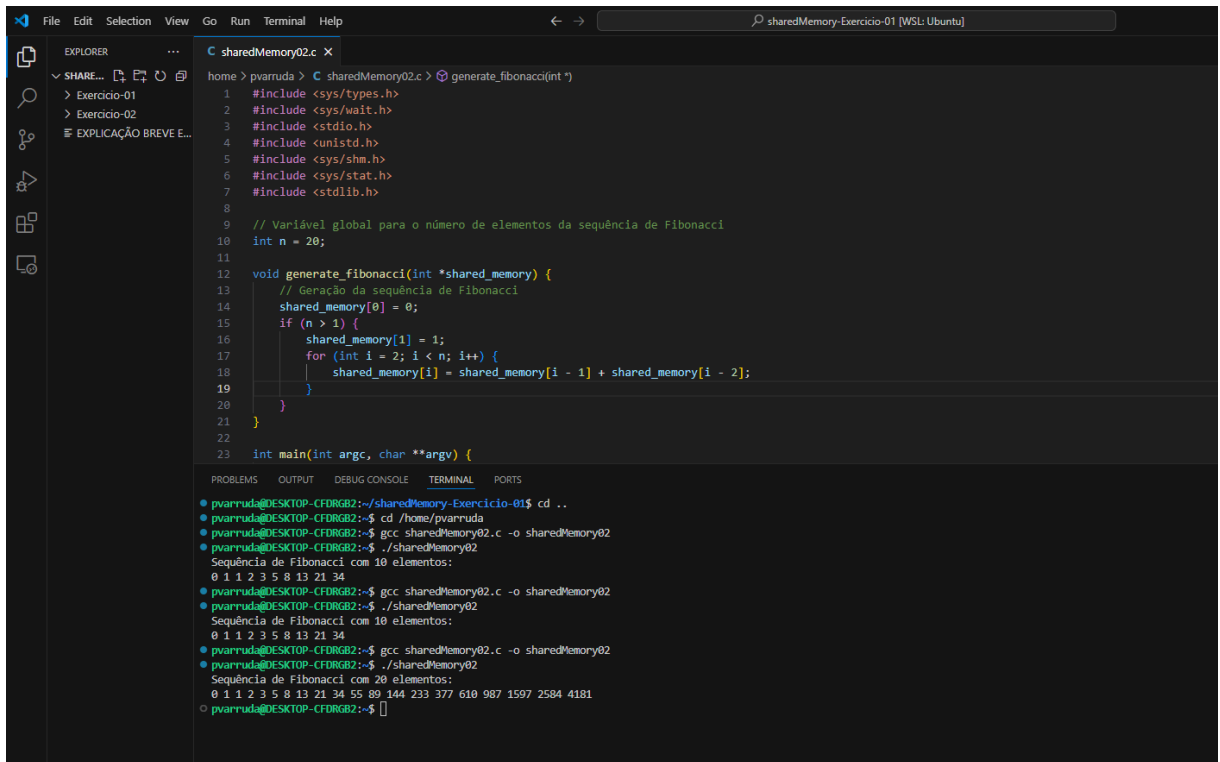
```
File Edit Selection View Go Run Terminal Help
sharedMemory-Exercicio-01 [WSL: Ubuntu]

EXPLORER
SHAREDMEMORY-EXERCIO...
  Exercício-01
  Exercício-02
  EXPLICAÇÃO BREVE E...

C sharedMemory02.c X
home > pvarruda > C sharedMemory02.c > ...
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 // Variável global para o número de elementos da sequência de Fibonacci
10 int n = 10;
11
12 void generate_fibonacci(int *shared_memory) {
13     // Geração da sequência de Fibonacci
14     shared_memory[0] = 0;
15     if (n > 1) {
16         shared_memory[1] = 1;
17         for (int i = 2; i < n; i++) {
18             shared_memory[i] = shared_memory[i - 1] + shared_memory[i - 2];
19         }
20     }
21 }
22
23 int main(int argc, char **argv) {
24
25     PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
26
27     pvarruda@DESKTOP-CFDRG82:~/sharedMemory-Exercicio-01$ cd ..
28     pvarruda@DESKTOP-CFDRG82:~$ cd /home/pvarruda
29     pvarruda@DESKTOP-CFDRG82:~$ gcc sharedMemory02.c -o sharedMemory02
30     pvarruda@DESKTOP-CFDRG82:~$ ./sharedMemory02
31     Sequência de Fibonacci com 10 elementos:
32     0 1 1 2 3 5 8 13 21 34
33     pvarruda@DESKTOP-CFDRG82:~$
```

Mais...

Teste com 20 elementos



The image shows a Visual Studio Code editor window with a C program named `sharedMemory02.c` and its execution output in the terminal.

Code:

```
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 // Variável global para o número de elementos da sequência de Fibonacci
10 int n = 20;
11
12 void generate_fibonacci(int *shared_memory) {
13     // Geração da sequência de Fibonacci
14     shared_memory[0] = 0;
15     if (n > 1) {
16         shared_memory[1] = 1;
17         for (int i = 2; i < n; i++) {
18             shared_memory[i] = shared_memory[i - 1] + shared_memory[i - 2];
19         }
20     }
21 }
22
23 int main(int argc, char **argv) {
```

Terminal Output:

```
pvarruda@DESKTOP-CFDRGB2:~/sharedMemory-Exercicio-01$ cd ..
pvarruda@DESKTOP-CFDRGB2:~$ cd /home/pvarruda
pvarruda@DESKTOP-CFDRGB2:~$ gcc sharedMemory02.c -o sharedMemory02
pvarruda@DESKTOP-CFDRGB2:~$ ./sharedMemory02
Sequência de Fibonacci com 10 elementos:
0 1 1 2 3 5 8 13 21 34
pvarruda@DESKTOP-CFDRGB2:~$ gcc sharedMemory02.c -o sharedMemory02
pvarruda@DESKTOP-CFDRGB2:~$ ./sharedMemory02
Sequência de Fibonacci com 10 elementos:
0 1 1 2 3 5 8 13 21 34
pvarruda@DESKTOP-CFDRGB2:~$ gcc sharedMemory02.c -o sharedMemory02
pvarruda@DESKTOP-CFDRGB2:~$ ./sharedMemory02
Sequência de Fibonacci com 20 elementos:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
pvarruda@DESKTOP-CFDRGB2:~$
```