

POSIX APIs

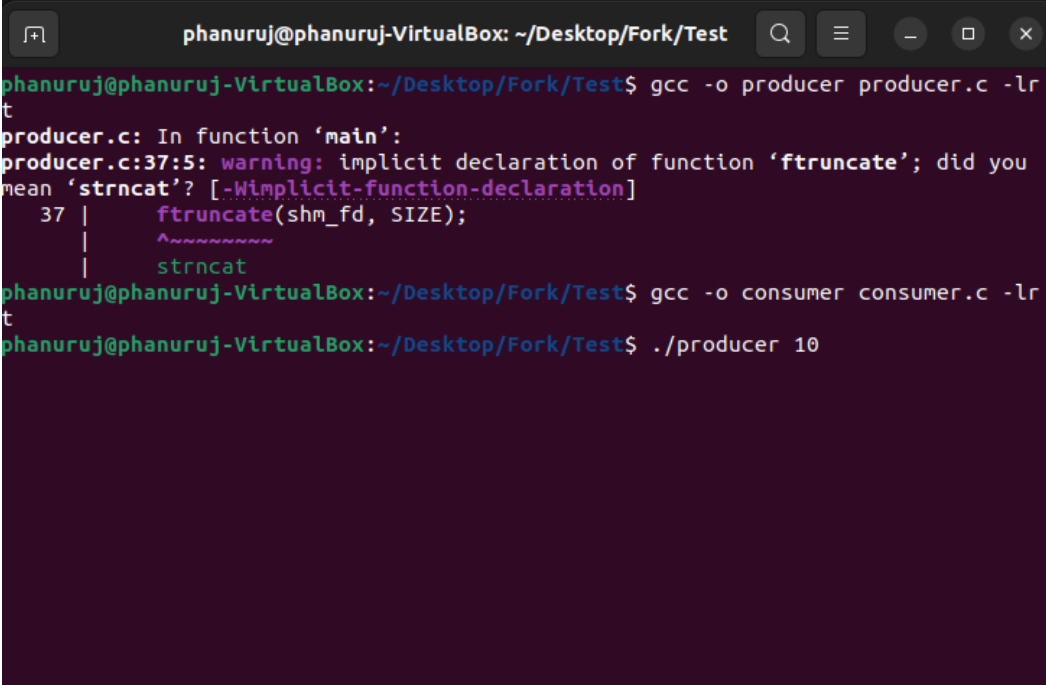
The code consists of two programs: `producer.c` and `consumer.c`. The producer program takes an integer as an input, calculates the sum of integers from 1 to that input, and then stores the result in shared memory. The consumer program reads the result from shared memory and displays it on the screen. If there is an error, the producer program stores an error message in shared memory instead of the result.

To run the program, first compile the `producer.c` and `consumer.c` programs separately by running the following commands in the terminal:

```
gcc -o producer producer.c -lrt
gcc -o consumer consumer.c -lrt
```

Then, run the producer program with an integer argument, for example:

```
./producer 10
```

A terminal window titled 'phanuruj@phanuruj-VirtualBox: ~/Desktop/Fork/Test' with standard window controls. It shows the compilation of 'producer.c' with a warning about an implicit declaration of 'ftruncate'. Then, it shows the compilation of 'consumer.c' and the execution of './producer 10'.

```
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ gcc -o producer producer.c -lrt
producer.c: In function 'main':
producer.c:37:5: warning: implicit declaration of function 'ftruncate'; did you
mean 'strncat'? [-Wimplicit-function-declaration]
   37 |     ftruncate(shm_fd, SIZE);
      |     ^~~~~~
      |     strncat
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ gcc -o consumer consumer.c -lrt
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ ./producer 10
```

This will calculate the sum of integers from 1 to 10 and store the result in shared memory.

Next, run the consumer program:

`./consumer`

```
phanuruj@phanuruj-VirtualBox: ~/Desktop/Fork/Test
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ gcc -o producer producer.c -lrt
producer.c: In function 'main':
producer.c:37:5: warning: implicit declaration of function 'ftruncate'; did you mean 'strncat'? [-Wimplicit-function-declaration]
   37 |     ftruncate(shm_fd, SIZE);
      |     ^~~~~~
   37 |     strncat
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ gcc -o consumer consumer.c -lrt
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ ./producer 10
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ ./consumer
The result is 55
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$
```

This will read the result from shared memory and display it on the screen.

If the input to the producer program is not a valid positive integer, it will store an error message in shared memory instead of the result. When the consumer program reads from shared memory, it will detect the error message and display it on the screen.

But if we Input -5 (error case) it will display:

```
phanuruj@phanuruj-VirtualBox: ~/Desktop/Fork/Test
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ gcc -o producer producer.c -lrt
producer.c: In function 'main':
producer.c:37:5: warning: implicit declaration of function 'ftruncate'; did you mean 'strncat'? [-Wimplicit-function-declaration]
   37 |     ftruncate(shm_fd, SIZE);
      |     ^~~~~~
   37 |     strncat
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ gcc -o consumer consumer.c -lrt
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ ./producer 10
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ ./consumer
The result is 55
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ ./producer -5
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$ ./consumer
Error: Invalid argument.
phanuruj@phanuruj-VirtualBox:~/Desktop/Fork/Test$
```