Операционные системы
ДЗ 7. Отчёт
Работа на 10 баллов

Фролов-Буканов Виктор Дмитриевич БПИ-228

29 марта 2024

1 Результаты работы программ

Первый способ

```
frolovbuk@LAPTOP-K0RFAB5O:~/hw7/first_way$ ./client.out
write a random number 383
write a random number 886
write a random number 777
write a random number 915
stowrite a random number 793
p
Was read: stop

The client was completed his work
```

```
frolovbuk@LAPTOP-K0RFAB5O:~/hw7/first_way$ ./server.out
Memory size set and = 8
0
0
0
383
886
777
915
793
The server has completed his work
```

Второй способ

```
frolovbuk@LAPTOP-K0RFAB5O:~/hw7/second_way$ ./client.out
Enter the server pid: 34885
addr = 0x7fbe97364000
write a random number 782
write a random number 129
write a random number 13
write a random number 359
swrite a random number 419
topwrite a random number 378

Was read: stop

Signal was sent

The client was completed his work
```

```
frolovbuk@LAPTOP-K0RFAB5O:~/hw7/second_way$ ./server.out
My pid is 34885
Memory size set and = 4
0
0
0
0
0
782
129
13
359
419
378
Signal was received
EXIT
```

Третий способ

```
frolovbuk@LAPTOP-K0RFAB5O:~/hw7/third_way$ ./client.out
addr = 0x7f485bae7000
write a random number 343
write a random number 969
write a random number 279
write a random number 674
stwrite a random number 434
op
Was read: stop

The client was completed his work
```

```
frolovbuk@LAPTOP-K0RFAB5O:~/hw7/third_way$ ./server.out
Memory size set and = 4
969
279
674
434
434
The server has completed his work
EXIT
```

2 Детали реализации каждого из способов

Реализации такие же, как и в ДЗ. В каждом из способов остановка работы программ осуществляется посредством введения в консоли клиента произвольной строки до 256 символов. Отличия между способами заключается лишь в взаимодействии между клиент-процессом и сервер-процессом. Вся очистка общих для двух процессов ресурсов производится на сервере (это удаление разделяемой памяти, а также *unlink* именнованного канала в 3 способе)

2.1 Первый способ

Структурой данных, которая является разделяемой для обеих процессов, теперь называется *message_t* и хранит в себе число, которое, собственно и генерируется клиентом, а позже передаётся серверу, а также текущий статус: *STOP* или *CONTINUE*. Если клиентом была прочитана строка, то статус устанавливается в *STOP*, что потом обрабатывается сервером и в зависимости от прочитанного флага принимается решение: продолжить выполнение цикла или выполнить *break*

2.2 Второй способ

Теперь индикатором остановки выполнения цикла на сервере является посылаемый сигнал *SIGTERM*. Если была прочитана строка на клиенте, то процессу-серверу посылается сигнал, в результате обработки которого происходит завершение выполнения цикла. Отмечу, что клиент при этом должен знать pid процесса-сервера, чтобы знать, куда отправлять сигнал, так что сервер выводит свой pid, а в клиенте перед началом работы необходимо ввести выведенный сервером pid

2.3 Третий способ

Теперь индикатором остановки выполнения цикла на сервере является посылаемое сообщение в именнованный канал *fifo*, к которому имеют доступ оба процесса. Если строка не была введена, то посылается сообщение *continue*, если же была введена, то посылается сообщение *stop*. В зависимости от полученного сообщения, сервером принимается решение о продолжении выполнения цикла

3 Замечание

В 1 и 3 способе критически важно сначала запускать сервер и только потом клиент, так как сервер выполняет инициализацию объекта разделяемой памяти *POSIX* посредством метода *shm_open*. Во 2 способе неважно, в каком порядке запускать программы, так как клиент не начнет свою работу, пока не получит *pid* сервера