

Операционные системы
ДЗ 6. Отчёт
Работа на 10 баллов

Фролов-Буканов Виктор Дмитриевич БПИ-228

21 марта 2024

1 Результаты работы программ

Первый способ

```
frolovbuk@LAPTOP-K0RFAB50:~/hw6/first_way$ ./client.out
shm_id = 13
share = 0x7f3af5dba000
write a random number 234
write a random number 856
write a random number 413
dsfwrite a random number 63
sdf
Was read: dsfsdf

The client was completed his work

frolovbuk@LAPTOP-K0RFAB50:~/hw6/first_way$ ./server.out
0
0
0
0
0
234
856
413
63
The server has completed his work
```

Второй способ

```
frolovbuk@LAPTOP-K0RFAB50:~/hw6/second_way$ ./client.out
Enter the server pid: 50166
shm_id = 17
share = 0x7fdc02100000
write a random number 759
write a random number 451
write a random number 494
write a random number 421
sdfsdf
Was read: sdfsdf

Signal was sent

The client was completed his work
```

```
frolovbuk@LAPTOP-KØRFAB50:~/hw6/second_way$ ./server.out
My pid is 50166
0
0
0
0
0
0
0
0
0
0
759
451
494
421
Signal was received
EXIT
```

Третий способ

```
frolovbuk@LAPTOP-KØRFAB50:~/hw6/third_way$ ./client.out
shm_id = 16
share = 0x7fec2a29e000
write a random number 109
write a random number 187
write a random number 258
sdfsdwrite a random number 334

Was read: sdfsd

The client was completed his work

frolovbuk@LAPTOP-KØRFAB50:~/hw6/third_way$ ./server.out
109
187
258
334
The server has completed his work
```

2 Детали реализации каждого из способов

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 обоих 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 именованный 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