Лабораторная работа №9 – Работа с разделенной памятью

Цель работы

Использование для обмена данными разделяемой памяти (shared memory) – самого быстрого средства межпроцессного взаимодействия в Linux.

Пункт 1

Скомпилируйте и выполните программу gener_shm.cpp демонстрирующую создание сегментов разделяемой памяти. Запустите программу несколько раз и после каждого ее завершения выполните команду ipcs -m . Поясните зависимость процедуры создания сегментов разделяемой памяти от используемых в вызове shmget() флагов.

```
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ make
g++ gener_shm.cpp -o gener_shm
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ./gener_shm
First memory identifire is 24
Second shared memory identifire is 25
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x00000000 13         wooffie    600        524288     2          dest
0x0000000f 24         wooffie    644        1000       0
0x00000000 25         wooffie    644        20         0

wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ./gener_shm
First memory identifire is 24
Second shared memory identifire is 26
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x00000000 13         wooffie    600        524288     2          dest
0x0000000f 24         wooffie    644        1000       0
0x00000000 25         wooffie    644        20         0
0x00000000 26         wooffie    644        20         0

wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ./gener_shm
First memory identifire is 24
Second shared memory identifire is 27
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x00000000 13         wooffie    600        524288     2          dest
0x0000000f 24         wooffie    644        1000       0
0x00000000 25         wooffie    644        20         0
0x00000000 26         wooffie    644        20         0
0x00000000 27         wooffie    644        20         0
```

Рис. 9-1 Многократное исполнение программы gener_shm.

Первый участок памяти создаётся по ключу 15. После, системный вызов shmget() лишь возвращает его идентификатор, найденный по ключу. Второй участок создаётся каждый раз так-как значение key равно **IPC_PRIVATE**.

Пункт 2

Удалите созданные на предыдущем шаге сегменты разделяемой памяти с помощью команды ipcrm с соответствующей опцией и значением id сегмента или ключа.

```
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x00000000 13         wooffie    600        524288     2          dest
0x0000000f 24         wooffie    644        1000       0
0x00000000 25         wooffie    644        20         0
0x00000000 26         wooffie    644        20         0
0x00000000 27         wooffie    644        20         0

wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ipcrm -M 0x0000000f
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ipcrm -m 25
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ipcrm -m 26
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ipcrm -m 27
wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x00000000 13         wooffie    600        524288     2          dest

wooffie@PC:~/spbstu-os-labs/labs/lab9/src1$
```

Рис. 9-2 Удаление сегментов разделяемой памяти.

Пункт 3

Скомпилируйте shmdemo.cpp , осуществляющую операции записи в разделяемую память без разделения доступа к этому общему ресурсу. Символы, записываемые в общую память, передаются в качестве параметра командной строки при запуске процесса shmdemo . Запуск этого процесса без параметров приводит к выводу на консоль текущего содержимого сегмента общей памяти.

```
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ make
g++ shmdemo.cpp -o shmdemo
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x00000000 13         wooffie    600        524288     2          dest
0x00000000 34         wooffie    600        524288     2          dest

wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ./shmdemo
segment contains: ""
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x00000000 13         wooffie    600        524288     2          dest
0x00000000 34         wooffie    600        524288     2          dest
0x5205808d 37         wooffie    644        1024       0

wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$
```

Рис. 9-3 Создание сегментов программой shmdemo.

Несложно заметить, что программа создаёт новый участок разделяемой памяти размером 1 килобайт.

Пункт 4

Запустите несколько раз процессы типа shmdemo с различными значениями параметров и проиллюстрируйте возможности чтения и записи в сегмент общей памяти независимо исполняемыми процессами. Затем удалите сегмент памяти командой ipcrm.

```
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ./shmdemo Hello!
writing to segment: "Hello!"
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ./shmdemo
segment contains: "Hello!"
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ./shmdemo Hello,_friends!
writing to segment: "Hello,_friends!"
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ./shmdemo Hello,Ubuntu!
writing to segment: "Hello,Ubuntu!"
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ./shmdemo
segment contains: "Hello,Ubuntu!"
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$
```

Рис. 9-4 Демонстрация работы shmdemo.

При запуске программа получает идентификатор к сегменту разделяемой памяти по своему ключу. После происходит вызов shamat(), который подключается сегмент общей памяти с нашим идентификатором к адресному пространству вызывающего процесса. После в зависимости от ввода программа или записывает или выводит строку. В конце происходит отключение сегмента общей памяти от процесса.

```
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x5205808d 37         wooffie    644        1024       0
0x00000000 40         wooffie    600        524288     2          dest

wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ipcrm -m 37
wooffie@PC:~/spbstu-os-labs/labs/lab9/src2$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x00000000 40         wooffie    600        524288     2          dest
```

Рис. 9-5 Удаление сегмента.

Пункт 5

Скомпилируйте и выполните программу attach_shm.cpp , иллюстрирующую передачу символьной информации между двумя процессами (родственными) через сегмент общей памяти с модификацией этой информации. Проанализируйте значения выводимой информации о границах сегментов в системной памяти. За счет чего после завершения данной программы сегмент общей памяти уже не присутствует в системе?

```
wooffie@PC:~/spbstu-os-labs/labs/lab9/src3$ make
g++ attach_shm.cpp -o attach_shm
wooffie@PC:~/spbstu-os-labs/labs/lab9/src3$ ./attach_shm
Addresses in parent

shared mem: 0x7fd6e5635000
 program text (etext): 0x55af38d4b4d5
 initialized data (edata): 0x55af38d4e010
 uninitialized data (end): 0x55af38d4e018

In parent before fork, memory is : ABCDEFGHIJKLMNOPQRSTUVWXYZ
In child after fork, memory is : ABCDEFGHIJKLMNOPQRSTUVWXYZ

In parent after fork, memory is  : abcdefghijklmnopqrstuvwxyz
Parent removing shared memory
wooffie@PC:~/spbstu-os-labs/labs/lab9/src3$
```

Рис. 9-6 Выполнение программы из 5 пункта.

В ходе выполнения программы с сегмент общей памяти записывается английский алфавит, после процесс-потомок изменяет ее (переводя буквы в нижний регистр), а процесс-родитель выводит эти данные и удаляет сегмент памяти.

То есть оба потомка отстыковываются сегменты разделяемой памяти от процессов через вызов shmdt(). А процесс-родитель (который подождал 3 секунды для завершения процесса-потомка) при помощи shmctl() помечает сегмент как удалённый. И поэтому после выполнения программы мы не можем пронаблюдать сегмент разделённой памяти.

Пункт 6

Составьте программу, создающую три разделяемых сегмента памяти размером 1023 байта каждый. Укажите в вызове shmat() параметр shmaddr = 0 при привязке сегментов. Разместит ли система сегменты в последовательных участках? Позволит ли система ссылку или изменение 1024-го байта любого из этих участков?

```
wooffie@PC:~/spbstu-os-labs/labs/lab9/src4$ make
g++ three.cpp -o three
wooffie@PC:~/spbstu-os-labs/labs/lab9/src4$ ./three
wooffie@PC:~/spbstu-os-labs/labs/lab9/src4$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 32769      wooffie    600        524288     2          dest
0x00000000 4          wooffie    600        524288     2          dest
0x00000000 5          wooffie    600        524288     2          dest
0x00000000 8          wooffie    600        524288     2          dest
0x00000000 32777      wooffie    600        524288     2          dest
0x00000000 32778      wooffie    600        524288     2          dest
0x00000000 32784      wooffie    600        524288     2          dest
0x00000000 32794      wooffie    644        1023       0
0x00000000 32795      wooffie    644        1023       0
0x00000000 32796      wooffie    644        1023       0
0x00000000 32807      wooffie    600        524288     2          dest
0x00000000 32809      wooffie    644        1023       0
0x00000000 32810      wooffie    644        1023       0
0x00000000 32811      wooffie    644        1023       0
0x00000000 44         wooffie    600        524288     2          dest
```

Рис. 9-7 Создание трёх сегментов.

Сегменты действительно разместились “по соседству” в памяти. При обращении к 1024 биту любого из них получаем ошибку.

Вывод

В ходе работы мы познакомились с функциональностью использования разделяемой памяти. Создавали сегменты, записывали и считывали из них информацию. Наблюдали за передачей информации между двумя процссами через сегмент общей памяти.