Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Кафедра информационных технологий и систем

**Отчет по лабораторной работе №3**

«Операционные системы»

Разработал:

Студент группы 9091

Чалый С.М\_\_\_\_\_\_\_

«\_\_».\_\_\_\_\_.2021г.

Проверил:

Ананьев В.В\_\_\_\_\_\_

«\_\_».\_\_\_\_\_.2021г.

**Великий Новгород**

**2021**

Цель работы : по заданию.

## Задание по разделяемой памяти

Написать две программы на C или C++.

Первая программа должна сформировать массив из 20 случайных чисел. Для массива должна быть выделена область в разделяемой памяти. Затем эта программа должна запустить вторую программу и передать ей ключ для доступа к общей области памяти в качестве параметра командной строки.

Вторая программа должна прочитать из общего массива все числа и вывести их на экран.

Затем она должна отсортировать их по возрастанию и вывести результат сортировки на экран.

## Задание по потокам

Написать многопоточную программу. Основной поток (который начинается в функции main) должен породить два новых потока, дождаться их завершения и закончить работу. Первый из новых потоков должен вывести на экран "Hello Threads (1)", "Hello Threads (2)" и т.д., всего 10 раз. Второй поток должен выводить в цикле строки "This is iteration 1", "This is iteration 2", ... "This is iteration 12".

Ход работы:

1) Создаю пустую папку :

mkdir lab03

cd lab03

2) Запустить редактор vim:

Разделяемая память:

1) Создаю файл sender.c :

#include <stdio.h>

#include <sys/shm.h>

#include <stdlib.h>

int main()

{

const size\_t memSize = 1024;

int memId = shmget(IPC\_PRIVATE, memSize, 0600|IPC\_CREAT|IPC\_EXCL);

printf("shmid = %i\n", memId);

if (memId <= 0)

{

printf("error with shmget()\n");

return -1;

}

char \*mem = (char \*)shmat(memId, 0, 0);

if (NULL == mem)

{

printf("error with shmat()\n");

return -2;

}

sprintf(mem, "string in shared memory\n");

char callbuf[1024];

sprintf(callbuf, "./receiver %i", memId);

system(callbuf);

return 0;

}

2) Создаю файл receiver.c:

#include <stdio.h>

#include <sys/shm.h>

#include <stdlib.h>

int main(int argv, char \*argc[])

{

if (argv <= 1)

{

printf("not enough params\n");

return -1;

}

char \*paramStr = argc[1];

int memId = atoi(paramStr);

if (memId == 0)

{

printf("incorrect parameter string: %s\n", paramStr);

return -2;

}

printf("receiving the memory data: shmid = %i\n", memId);

char \*mem = (char \*)shmat(memId, 0, 0);

if (NULL == mem)

{

printf("error with shmat()\n");

return -3;

}

printf("received string is: %s\n", mem);

return 0;

}

3) Компилирую файлы:

all:

gcc sender.c -o sender

gcc receiver.c -o receiver

4) Запустить отправителя:

./sender

5) Создать файл threader.c со следующим содержимым:

Добавляю функции вывода строк.

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

void \*print\_message\_function(void \*ptr)

{

int retval = 4;

char \*message = (char \*) ptr;

printf("%s\n", message);

return (void \*)(++retval);

}

int main()

{

pthread\_t thread1, thread2;

const char \*message1 = "Thread 1";

const char \*message2 = "Thread 2";

int res1 = pthread\_create(&thread1, NULL, print\_message\_function, (void\*) message1);

int res2 = pthread\_create(&thread2, NULL, print\_message\_function, (void\*) message2);

int iret1, iret2;

pthread\_join(thread1, (void \*\*)&iret1);

pthread\_join(thread2, (void \*\*)&iret2);

printf("Thread 1 returns: %d\n",iret1);

printf("Thread 2 returns: %d\n",iret2);

return 0;

}

6) Компилирую созданный файл

all:

gcc threader.c -lpthread -o threader

7) Запускаю программу:

./threader

Вывод: таким образом я научился работать с потоками, нитями и разделяемой памятью.