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

Ковалев А.Д\_\_\_\_\_\_\_

«\_\_».\_\_\_\_\_.2021г.

Проверил:

Ананьев В.В\_\_\_\_\_\_

«\_\_».\_\_\_\_\_.2021г.

**Великий Новгород**

**2021**

**Цель лабораторной работы**

Цель работы: познакомиться с механизмом обмена данными между потоками процессов - разделяемой памяться. Познакомться с потоками thread в соответствии стандарта POSIX (pthread).

**Задание на разделяемую память**

**Исходный текст программ**

Sender.c

#include "stdio.h"

#include "sys/shm.h"

#include "stdlib.h"

#include "sys/ipc.h"

#include "time.h"

int main(void)

{

const size\_t memSize = 80;

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

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

if (memId <= 0)

{

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

return -1;

}

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

if (NULL == mem)

{

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

return -2;

}

srand(time(NULL));

for (int i = 0; i < 20; i++)

\*(mem + i) = rand() % 100;

char callbuf[1024];

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

system(callbuf);

return 0;

}

Reciver.c

#include "stdio.h"

#include "sys/shm.h"

#include "stdlib.h"

void print\_array\_int(int\* arr)

{

for (int i = 0; i < 20; i++)

printf("%i ", \*(arr + i));

printf("\n");

}

int compare\_int\_value(const void\* a, const void\* b)

{

return \*((int\*) a) - \*((int\*) b);

}

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);

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

if (NULL == mem)

{

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

return -3;

}

printf("receiving next array:\n");

print\_array\_int(mem);

qsort(mem, 20, 4, compare\_int\_value);

printf("sorted array:\n");

print\_array\_int(mem);

return 0;

}

**Результат выполнения программ**

>> make

gcc receiver.c -o receiver

gcc sender.c -o sender

./sender

5356 4211 5614 3697 5639 3965 8099 8894 5938 8267 7457 8978 2629 8268 8065 5923 2834 6387 9282 5025

2629 2834 3697 3965 4211 5025 5356 5614 5639 5923 5938 6387 7457 8065 8099 8267 8268 8894 8978 9282

**Задание на потоки**

**Исходный текст программ**

**Threader.c**

#include "stdio.h"

#include "stdlib.h"

#include "pthread.h"

void\* print\_message\_function1()

{

for (int i = 1; i < 11; i++)

printf("Hello Thread(%i)\n", i), sleep(1);

return NULL;

}

void\* print\_message\_function2()

{

for (int i = 1; i < 13; i++)

printf("This is iteration %i\n", i), sleep(2);

return NULL;

}

int main(void)

{

pthread\_t thread1, thread2;

int res1 = pthread\_create(&thread1, NULL, print\_message\_function1, NULL);

int res2 = pthread\_create(&thread2, NULL, print\_message\_function2, NULL);

int iret1, iret2;

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

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

return 0;}

**Результат выполнения программы**

>> make

gcc threader.c -lpthreader -o threader

./threader

Hello Threads (1), This is iteration 1, Hello Threads (2), Hello Threads (3), This is iteration 2, Hello Threads (4), This is iteration 3, Hello Threads (5), Hello Threads (6), This is iteration 4, Hello Threads (7), Hello Threads (8), This is iteration 5, Hello Threads (9), Hello Threads (10), This is iteration 6, This is iteration 7, This is iteration 8, This is iteration 9, This is iteration 10, This is iteration 11, This is iteration 12,

**Вывод**

Вывод: выполняя лабораторную работу, я познакомился с разделяемой памятью и с потоками в соответствии стандарта POSIX (pthread).