Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования "Новгородский государственный университет имени Ярослава Мудрого"

Кафедра «Информационных технологий и систем»

Дисциплина «Операционные системы»

Отчет по лабораторной работе

«Средства Systen V IPC. Организация работы с разделяемой памятью. Понятие потоков в UNIX»

Выполнил студент группы 9091

\_\_\_\_\_\_\_\_\_\_\_\_\_/Прокофьев Максим Юрьевич/

Подпись ФИО

Принял преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_/Ананьев Владислав Валерьевич/

Подпись ФИО

Великий Новгород

2021

**Цель лабораторной работы**

Цель работы: познакомиться с механизмом обмена данными между потоками процессов - разделяемой памятью. Познакомиться 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 <time.h>  const int NUMBER = 20;  int main()  {  srand((unsigned)(time(0)));  int memId = shmget(IPC\_PRIVATE, sizeof(int) \* NUMBER, 0600|IPC\_CREAT|IPC\_EXCL);  int\* array = (int\*)shmat(memId, 0, 0);  for (size\_t i = 0; i < NUMBER; i++)  {  array[i] = rand() % 300;  }  char callbuf[1024];  sprintf(callbuf, "./receiver %i %i", memId, NUMBER);  system(callbuf);    return 0;  } |

|  |
| --- |
| receiver.c |
| #include <stdio.h>  #include <sys/shm.h>  #include <stdlib.h>  int comparator(const void\* a, const void\* b)  {  return ( \*((int\*)a) - \*((int\*)b) );  }  int main (int argv, char\* argc[])  {  int memId = atoi(argc[1]);  const int NUMBER = atoi(argc[2]);  int\* mem = (int\*)shmat(memId, 0, 0);  for (size\_t i = 0; i < NUMBER; i++)  {  printf("%i ", mem[i]);  }  printf("\n");  qsort(mem, NUMBER, sizeof(int), comparator);  for (size\_t i = 0; i < NUMBER; i++)  {  printf("%i ", mem[i]);  }  printf("\n");    return 0;  } |

**Результат выполнения программ**

|  |
| --- |
| Вывод в терминале |
| >> make  gcc sender.c -o sender  gcc receiver.c -o receiver  ./sender  shmid = 54  receiving the memory data: shmid = 54  receiving next array:  10 84 225 213 3 169 221 120 63 43 23 146 83 183 70 46 17 251 61 45  3 10 17 23 43 45 46 61 63 70 83 120 146 169 183 213 221 224 251 |

**Задание на потоки**

**Исходный текст программ**

|  |
| --- |
| threader.c |
| #include <stdio.h>  #include <stdlib.h>  #include <pthread.h>  #include <unistd.h>  void \*mission1()  {  for (int i = 1; i <= 10; i++)  {  printf("Hello Threads(%i), ", i);  sleep(1);  }  }  void \*mission2()  {  for (int i = 1; i <= 12; i++)  {  printf("This is iteration %i, ", i);  sleep(2);  }  }  int main()  {  pthread\_t thread1, thread2;  int res1 = pthread\_create(&thread1, NULL, mission1, NULL);  int res2 = pthread\_create(&thread2, NULL, mission2, NULL);    pthread\_join(thread1, NULL);  pthread\_join(thread2, NULL);  printf("\n");    return 0;  } |

**Результат выполнения программы**

|  |
| --- |
| Вывод в терминале |
| >> make  gcc threader.c -lpthread -o threader  ./threader  Hello Thread(1)  This is iteration 1  Hello Thread(2)  This is iteration 2  Hello Thread(3)  Hello Thread(4)  This is iteration 3  Hello Thread(5)  Hello Thread(6)  This is iteration 4  Hello Thread(7)  Hello Thread(8)  This is iteration 5  Hello Thread(9)  Hello Thread(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).