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

«Очереди сообщений в UNIX и работа с ними»

Выполнил студент группы 9091

\_\_\_\_\_\_\_\_\_\_\_\_\_/Семенов Вячеслав Алексеевич/

Подпись ФИО

Принял преподаватель

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

Подпись ФИО

Великий Новгород

2021

**Цель лабораторной работы**

Цель работы: познакомиться с механизмом общения между процессами, используя очереди сообщений в UNIX.

**Исходный текст программы**

|  |
| --- |
| main.cpp |
| #include "stdio.h"  #include "stdlib.h"  #include "unistd.h"  #include "time.h"  #include "pthread.h"  #include "sys/ipc.h"  #include "sys/msg.h"  #include <algorithm>  typedef struct {  long mtype;  int msg\_data[4];  char is\_last;  } msg\_struct;  void swap\_values(int\* first, int\* second)  {  int temp = \*first;  \*first = \*second;  \*second = \*first;  }  void\* pthread\_work(void\* args)  {  msg\_struct pthread\_msg;  int msg\_id = \*((int \*) args);  ssize\_t msg\_len = msgrcv(msg\_id, &pthread\_msg, sizeof(pthread\_msg), 0, 0);  do  {  if (!std::next\_permutation(pthread\_msg.msg\_data, pthread\_msg.msg\_data + 4))  {  pthread\_msg.is\_last = 1;  msgsnd(msg\_id, &pthread\_msg, sizeof(pthread\_msg), 0);  break;  }  msgsnd(msg\_id, &pthread\_msg, sizeof(pthread\_msg), 0);  } while(1);  return 0;  }  void print\_received\_msg(msg\_struct\* msg)  {  for (int i = 0; i < 4; i++)  printf("%i ", msg->msg\_data[i]);  printf("\n");  }  int compare\_int\_value(const void\* a, const void\* b)  {  return \*((int\*) a) - \*((int\*) b);  }  int main(void)  {  pthread\_t thread;  int random\_numbers[4];    srand(time(NULL));  for (int i = 0; i < 4; i++)  random\_numbers[i] = rand() % 1000;  qsort(random\_numbers, 4, sizeof(int), compare\_int\_value);  printf("Random numbers is: ");  for (int i = 0; i < 4; i++)  printf("%i ", random\_numbers[i]);  printf("\n");  int msg\_id = msgget(IPC\_PRIVATE, 0600 | IPC\_CREAT);  if (msg\_id < 0)  {  perror("Error with msgget()!\n");  return -1;  }  printf("Message id = %i\n", msg\_id);  int res1 = pthread\_create(&thread, NULL, pthread\_work, &msg\_id);  // prepare to send values to thread  msg\_struct parent\_msg;  parent\_msg.mtype = 5;  parent\_msg.is\_last = 0;  for (int i = 0; i < 4; i++)  parent\_msg.msg\_data[i] = random\_numbers[i];  msgsnd(msg\_id, &parent\_msg, sizeof(parent\_msg), 0);  int count\_msg\_received = 0;  while (!parent\_msg.is\_last)  {  printf("=== Received next msg ===\n");  msgrcv(msg\_id, &parent\_msg, sizeof(parent\_msg), 0, 0);  print\_received\_msg(&parent\_msg);  count\_msg\_received++;  };  printf("========= RESULT =========\n");  printf("Msg count: %i\n", count\_msg\_received);  printf("==========================\n");  msgctl(msg\_id, IPC\_RMID, NULL);  return 0;  } |

**Результат выполнения программы**

|  |
| --- |
| Вывод в терминале |
| >> make  g++ main.cpp -lpthread -o main  ./main  Random numbers is: 461 547 724 784  Message id = 32831  === Received next msg ===  461 547 784 724  === Received next msg ===  461 724 547 784  === Received next msg ===  461 724 784 547  === Received next msg ===  461 784 547 724  === Received next msg ===  461 784 724 547  === Received next msg ===  547 461 724 784  === Received next msg ===  547 461 784 724  === Received next msg ===  547 724 461 784  === Received next msg ===  547 724 784 461  === Received next msg ===  547 784 461 724  === Received next msg ===  547 784 724 461  === Received next msg ===  724 461 547 784  === Received next msg ===  724 461 784 547  === Received next msg ===  724 547 461 784  === Received next msg ===  724 547 784 461  === Received next msg ===  724 784 461 547  === Received next msg ===  724 784 547 461  === Received next msg ===  784 461 547 724  === Received next msg ===  784 461 724 547  === Received next msg ===  784 547 461 724  === Received next msg ===  784 547 724 461  === Received next msg ===  784 724 461 547  === Received next msg ===  784 724 547 461  === Received next msg ===  461 547 724 784  ========= RESULT =========  Msg count: 24  ========================== |

**Вывод**

Вывод: выполняя лабораторную работу, я познакомился и научился работать с механизмом общения между процессами, используя очереди сообщений в UNIX.