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

«Процессы в ОС UNIX»

Выполнил студент группы 9091

\_\_\_\_\_\_\_\_\_\_\_\_\_/Малинин Никита Валерьевич/

Подпись ФИО

Принял преподаватель

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

Подпись ФИО

Великий Новгород

2021

**Цель лабораторной работы**

Цель работы: познакомиться с механизмом создания новых процессов в операционной системе UNIX.

**Исходный текст программы**

|  |
| --- |
| processmaker.cpp |
| #include <stdio.h>  #include <sys/shm.h>  #include <stdlib.h>  #include <unistd.h>  #include <sys/types.h>  #include <sys/wait.h>  #include "time.h"  void \*allocateSharedMemory(size\_t memSize, int &memId)  {  memId = shmget(IPC\_PRIVATE, memSize, 0600|IPC\_CREAT|IPC\_EXCL);  if (memId <= 0)  {  perror("error with memId");  return NULL;  }  void \*mem = shmat(memId, 0, 0);    if (NULL == mem)  {  perror("error with shmat");  }    return mem;  }  void printArray(int\* arr)  {  for (int i = 0; i < 20; i++)  printf("%i ", \*(arr + i));  printf("\n");  }  int compareValue(const void\* a, const void\* b)  {  return \*((int\*) a) - \*((int\*) b);  }  void childMainCode(int\* sharedMem)  {  qsort(sharedMem, 20, 4, compareValue);  printArray(sharedMem);  exit(0);  }  int main() {  int memId;  int \*sharedMem = (int \*)allocateSharedMemory(1024, memId);  printf("memId = %d\n", memId);  printf("starting child process...\n");  srand(time(NULL));  for(int i = 0; i < 20; i++)  \*(sharedMem + i) = rand()%100;    printArray(sharedMem);  pid\_t childId = fork();    if (childId < 0)  {  perror("error with fork()\n");  }  else if (childId > 0)  {  waitpid(childId, NULL, 0);  }  else  {  childMainCode(sharedMem);  }    char sharedMemDel[124];  sprintf(sharedMemDel, "ipcrm -m %i", memId);  system(sharedMemDel);  return 0;  } |

**Результат выполнения программы**

|  |
| --- |
| Вывод в терминале |
| >> make  g++ processmaker.cpp -o processmaker  ./processmaker  memId = 32801  starting child process...  46 98 19 11 29 46 37 1 72 5 19 0 49 1 68 82 57 70 69 97  0 1 1 5 11 19 19 29 37 46 46 49 57 68 69 70 72 82 97 98 |

**Вывод**

Вывод: выполняя лабораторную работу, я познакомился с механизмов создания новых процессов в операционной системе UNIX.