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

\_\_\_\_\_\_\_\_\_\_\_\_\_/Лопатин Даниил Михайлович/

Подпись ФИО

Принял преподаватель

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

Подпись ФИО

Великий Новгород

2021

**Цель лабораторной 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 "time.h"  #include "sys/types.h"  #include "sys/wait.h"  #include <cstdlib>  #include <stdlib.h>  void \*allocateSharedMemory(size\_t mem\_size, int& mem\_id)  {  mem\_id = shmget(IPC\_PRIVATE, mem\_size, 0600 | IPC\_CREAT | IPC\_EXCL);  if (mem\_id <= 0)  {  perror("error with memId");  return NULL;  }    void \*mem = shmat(mem\_id, 0, 0);  if (NULL == mem)  {  perror("error with shmat");  }    return mem;  }  void print\_arr\_int(int\* arr\_ptr)  {  for (int i = 0; i < 20; i++)  printf("%i ", \*(arr\_ptr + i));  printf("\n");  }  int compare\_int(const void\* a, const void\* b)  {  return \*((int\*) a) - \*((int\*) b);  }  void child\_main\_code(int\* shared\_mem\_ptr)  {  qsort(shared\_mem\_ptr, 20, 4, compare\_int);  print\_arr\_int(shared\_mem\_ptr);  exit(0);  }  int main(void)  {  int mem\_id;  int\* shared\_mem\_ptr = (int\*) allocateSharedMemory(80, mem\_id);  printf("mem\_id = %d\n", mem\_id);    srand(time(NULL));  for (int i = 0; i < 20; i++)  {  \*(shared\_mem\_ptr + i) = rand() % 100;  }  print\_arr\_int(shared\_mem\_ptr);    pid\_t child\_process\_id = fork();    if (child\_process\_id == -1)  {  perror("error with fork() - process 1\n");  }  else if (child\_process\_id == 0)  {  child\_main\_code(shared\_mem\_ptr);  }  else  {  waitpid(child\_process\_id, NULL, 0);  }    char shared\_mem\_del\_com[124];  sprintf(shared\_mem\_del\_com, "ipcrm -m %i", mem\_id);  system(shared\_mem\_del\_com);    return 0;  } |

**Результат выполнения программы**

|  |
| --- |
| Вывод в терминале |
| >> make  g++ processmaker.cpp -o processmaker  ./processmaker  mem\_id = 2  7 80 35 92 31 28 44 87 8 32 83 83 93 33 68 24 8 37 50 60  7 8 8 24 28 31 32 33 35 37 44 50 60 68 80 83 83 87 92 93 |

**Вывод:**

Выполняя лабораторную работу, я познакомился с механизмов создания новых процессов в операционной системе UNIX.