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

\_\_\_\_\_\_\_\_\_\_\_\_\_/Алентьев Александр Григорьевич/

Подпись ФИО

Принял преподаватель

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

Подпись ФИО

Великий Новгород

2021

**Цель лабораторной работ:**

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

**Задание:**

1. Объединить программы первой части предыдущей лабораторной в одну программу.
2. Вначале программа должна сгенерировать 20 случайных чисел, разместить их в разделяемой памяти и вывести на экран.
3. Затем требуется породить новый процесс, который выполнит сортировку самостоятельно, без запуска других программ (в отличие от предыдущей лабораторной).
4. Второй процесс должен отсортировать числа и также вывести их на экран.
5. Первый процесс должен дождаться завершения работы второго, после чего самостоятельно выполнить освобождение выделенной разделяемой памяти и всех прочих выделенных ресурсов (при наличии таковых).

**Ход работы:**

**Исходный текст программы:**

|  |
| --- |
| processmaker.cpp |
| #include "stdio.h"  #include "sys/shm.h"  #include "stdlib.h"  #include "sys/ipc.h"  #include "unistd.h"  #include "time.h"  #include <sys/types.h>  #include <sys/wait.h>  void\* allocate\_shared\_memory(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;  }  int compare\_int\_value(const void\* a, const void\* b)  {  return \*((int\*) a) - \*((int\*) b);  }  void printArray(int\* sharedMem)  {  for(int i = 0; i < 20; i++)  printf("%i ", \*(sharedMem +i));  printf("\n");  }  void childMainCode(int\* sharedMem)  {  qsort(sharedMem, 20, 4, compare\_int\_value);  printArray(sharedMem);  }  int main(void)  {  int memId;  int\* sharedMem = (int\*) allocate\_shared\_memory(128, memId);  printf("mem\_id = %d\n", memId);  srand(time(NULL));  for (int i = 0; i < 20; i++)  \*(sharedMem + i) = rand() % 50;  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 callbuf[128];  sprintf(callbuf, "ipcrm -m %i", memId);  system(callbuf);  return 0;  } |

|  |
| --- |
| Output Terminal |
| make  g++ processmaker.cpp -o processmaker  ./processmaker  mem\_id = 33  40 28 2 24 8 36 12 14 25 22 11 3 16 38 35 27 11 40 45 47  2 3 8 11 11 12 14 16 22 24 25 27 28 35 36 38 40 40 45 47 |

**Вывод:**

Входе выполнения лабораторной работы, я познакомился с созданием процессов в операционной системе UNIX.