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

«Организация ввода-вывода в Unix. Файлы устройств»

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 огранизацией ввод-вывода в UNIX и файлами устройств.

**Исходный текст программы**

|  |
| --- |
| main.c |
| #include "stdlib.h"  #include "stdio.h"  int main(int argv, char\* argc[])  {  if (argv <= 2)  {  printf("Error! Not enough arguments! Required (2): file name, count (in bytes)\n");  return -1;  }  const char\* file\_name = argc[1];  size\_t bytes\_to\_read = atoi(argc[2]);  FILE\* file\_in = fopen(file\_name, "r");  if (!file\_in)  {  printf("Error! Cant open this file!\n");  return -1;  }  char\* buffer = malloc(bytes\_to\_read);  int result = fread(buffer, sizeof(char), bytes\_to\_read, file\_in);  if (!result)  printf("Cant read this file!\n");  else  {  for (size\_t i = 0; i < bytes\_to\_read; i++)  printf("int = %i \t char = %c \t (0x%x)\n", buffer[i], buffer[i], buffer[i]);  printf("\n");  }  free(buffer);  fclose(file\_in);  return 0;  } |

**Результат выполнения программы**

|  |
| --- |
| Вывод в терминале |
| >> make  gcc main.c -o main  ./main /dev/null 10  Cant read this file!  ./main /dev/zero 10  int = 0 char = (0x0)  int = 0 char = (0x0)  int = 0 char = (0x0)  int = 0 char = (0x0)  int = 0 char = (0x0)  int = 0 char = (0x0)  int = 0 char = (0x0)  int = 0 char = (0x0)  int = 0 char = (0x0)  int = 0 char = (0x0)  ./main /dev/urandom 10  int = -99 char = � (0xffffff9d)  int = 39 char = ' (0x27)  int = 86 char = V (0x56)  int = -66 char = � (0xffffffbe)  int = 87 char = W (0x57)  int = -69 char = � (0xffffffbb)  int = -48 char = � (0xffffffd0)  int = -62 char = � (0xffffffc2)  int = -64 char = � (0xffffffc0)  int = 99 char = c (0x63) |

|  |
| --- |
| Чтение первых 20 байт загрузочного раздела |
| >> sudo ./main *dev*/sda1 20  int = -21 char = � (0xffffffeb)  int = 88 char = X (0x58)  int = -112 char = � (0xffffff90)  int = 109 char = m (0x6d)  int = 107 char = k (0x6b)  int = 102 char = f (0x66)  int = 115 char = s (0x73)  int = 46 char = . (0x2e)  int = 102 char = f (0x66)  int = 97 char = a (0x61)  int = 116 char = t (0x74)  int = 0 char = (0x0)  int = 2 char = (0x2)  int = 8 char = (0x8)  int = 32 char = (0x20)  int = 0 char = (0x0)  int = 2 char = (0x2)  int = 0 char = (0x0)  int = 0 char = (0x0)  int = 0 char = (0x0) |

**Вывод**

Вывод: выполняя лабораторную работу, я познакомился с огранизацией ввод-вывода в UNIX и файлами устройств.