Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №10**

«Система команд и файловая структура ОС UNIX»

дисциплина «Операционные системы и оболочки»

Выполнил:

студент группы 313.1 Козявин М.С.

Проверил:

к.т.н., доцент кафедры ПОиАИС Кривонос А.В.

Курск, 2023

**Цели работы:**

Изучить команды ОС Fedora для работы с файлами, каталогами,дисками; текстовый редактор joe и программу для мониторинга процессов htop. Написать программу на языке C++ работающую с текстовыми файлами.

**Задание:**

1. Изучить теоретическую часть лабораторной работы.
2. В консольном режиме, используя команды терминала, создать в домашней папке подкаталог /номер\_группы/ФИО\_студента, где в дальнейшем будут храниться все файлы студента. Перейти в корневой каталог и вывести его содержимое, используя команды dir и ls –all, проанализировать различия.
3. Проверить действие команд ps, ps –x, top, htop. Используя команду man, найти в справочной системе справку по функциям fprintf, fputc и команде ls.
4. В текстовом редакторе joe (вызов: joe 1.cpp) написать программу 1.cpp,

выводящую на экран простое сообщение. Компилировать

полученную программу компилятором g++, запустить полученный файл на выполнение.

**Вариант 4**

Необходимо для чтения или записи файла использовать функции посимвольного ввода – вывода fgetc(), fputc( ) или getc(), putc(). Должен быть контроль ошибок открытия, закрытия, чтения и записи файла или каталога. Вывод сообщений об ошибках должен производиться в стандартный поток вывода сообщений об ошибках (stderr) в следующем виде: имя\_модуля: текст\_сообщения. Имя модуля берется из аргументов командной строки.

4. Программа подсчета числа отображаемых символов в строках текстового файла. Результаты подсчета записываются во второй текстовый файл (имена файлов передаются в качестве аргументов командной строки при запуске программы). Пример работы программы: исходный текстовый файл из трех строк:

**QWER**

**REEEt**

**WEEEEEEERSIIIONN**

файл, полученный в результате работы программы:

**1. 4**

**2. 5**

**3. 16**

**итого: 3 строки 35 символов**

**Выполнение:**

1. Ознакомился с теоретический частью и установил ОС Fedora на базе ядра Linux.
2. Используя команды терминала создал папку /313/Kozyavin Maxim, вывел содержимое корневого каталога.

Команда dir выводит только имена каталогов, а команда ls –all выводит подробную информацию о каталогах и отображает скрытые каталоги.

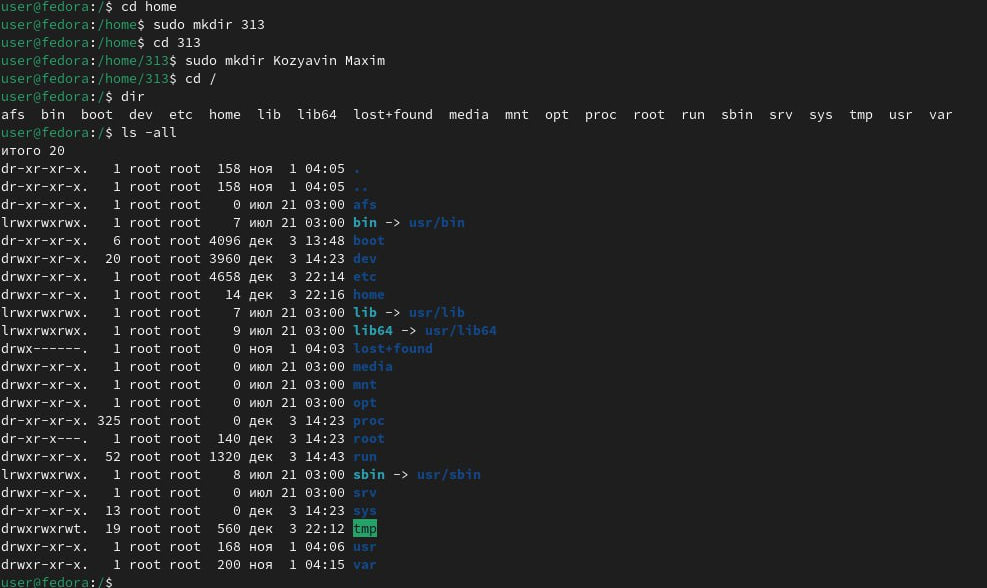


Рисунок 1 – Создание и просмотр директорий

1. Протестировал команды ps, top, htop и вывел справочную информацию при помощи команды man. Программу htop необходимо установить перед использованием с помощью команды sudo dnf install htop.



Рисунок 2 – Реакция на команды ps

Команда ps выводит список активных процессов пользователя, а ps –x выводит список всех процессов данного пользователя.

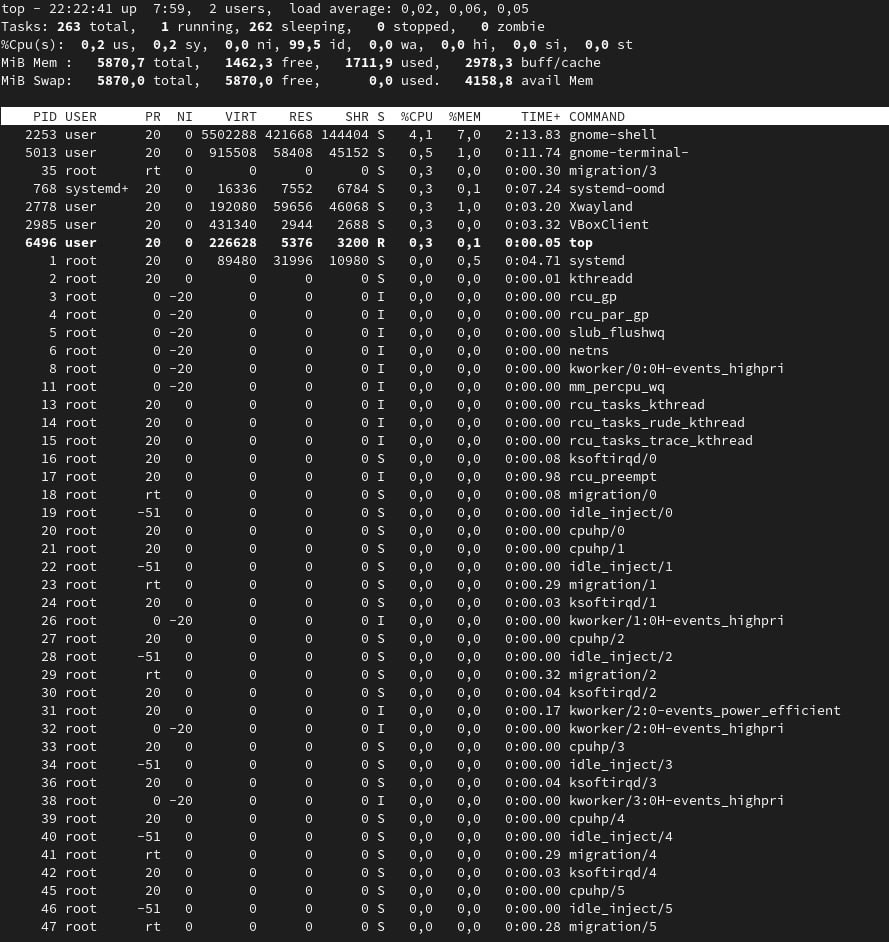


Рисунок 3 – Реакция на команду top

Команда top выводит полный список процессов.

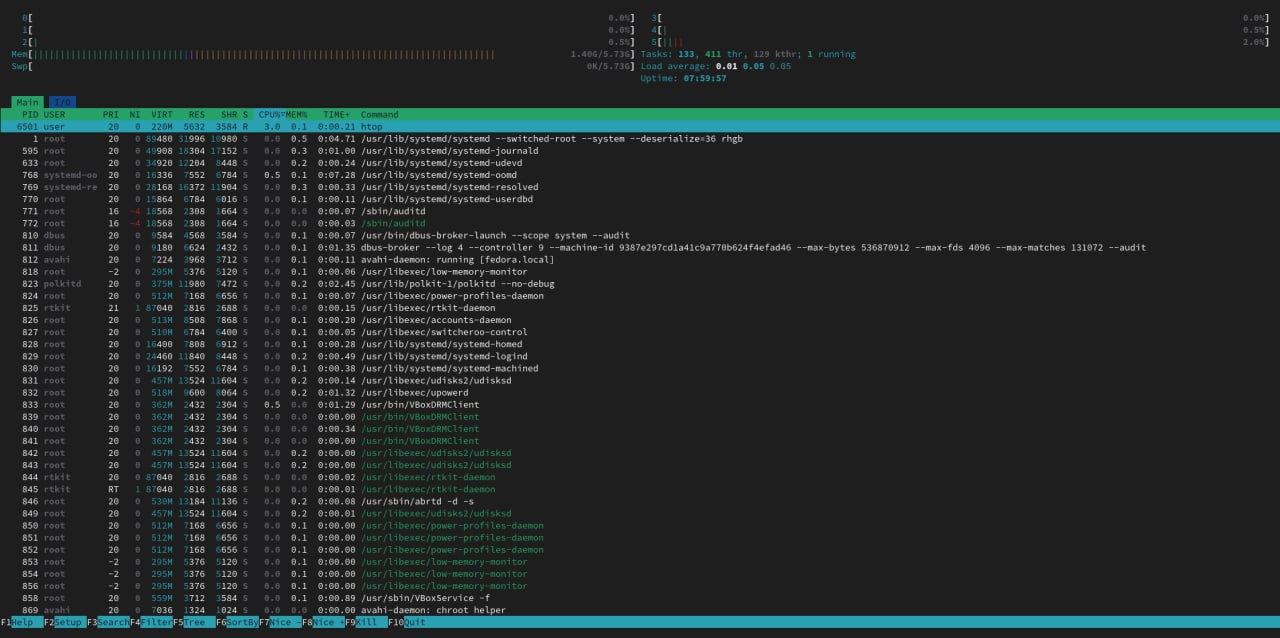


Рисунок 4 – Реакция на команду htop

Команда htop выводит открывает интерактивное приложение для терминала для просмотра информации о процессах.

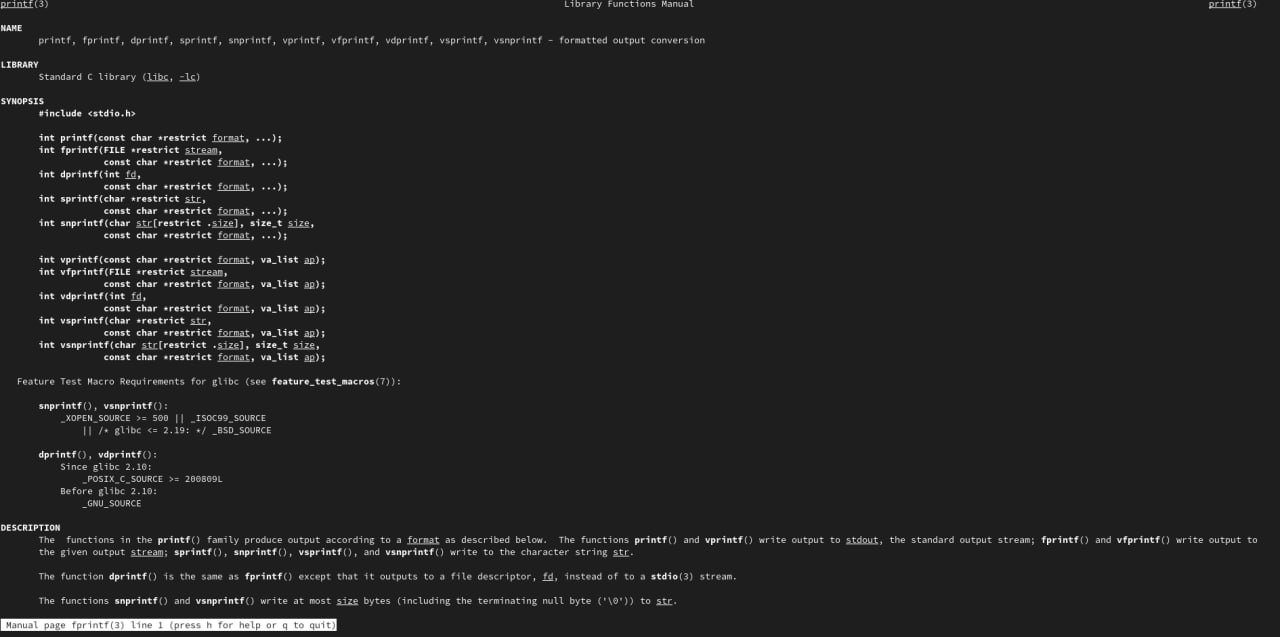


Рисунок 5 – Реакция на команду man fprintf

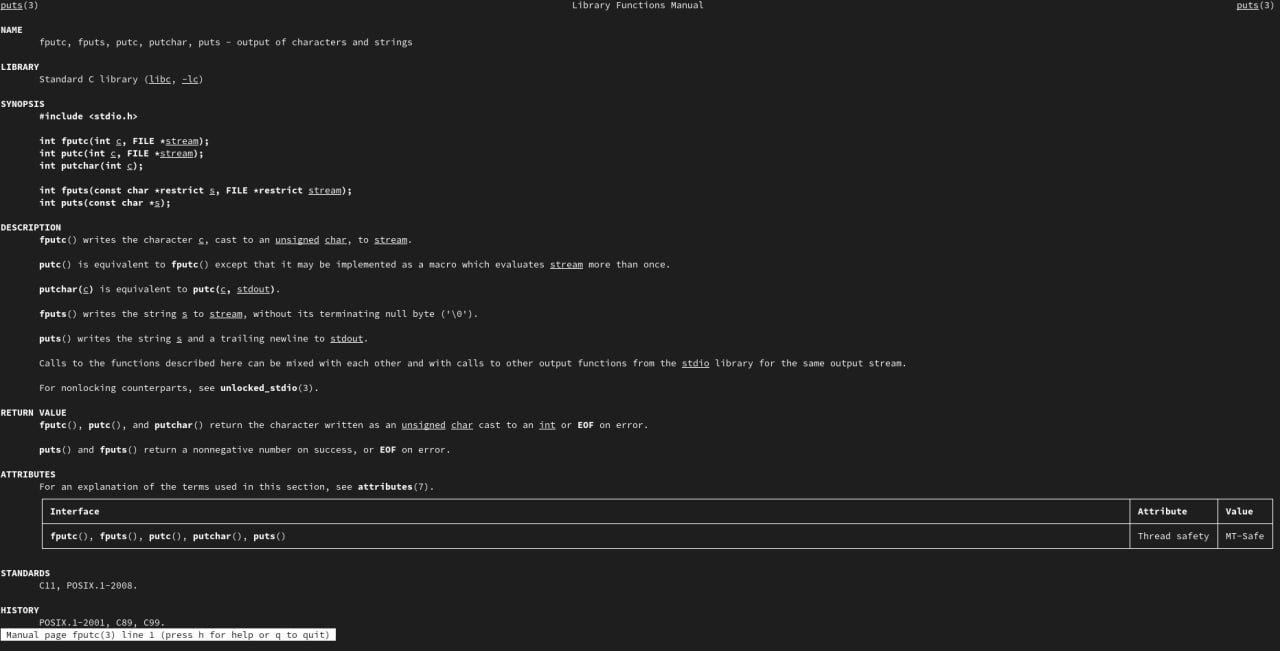


Рисунок 6 – Реакция на команду man fputc

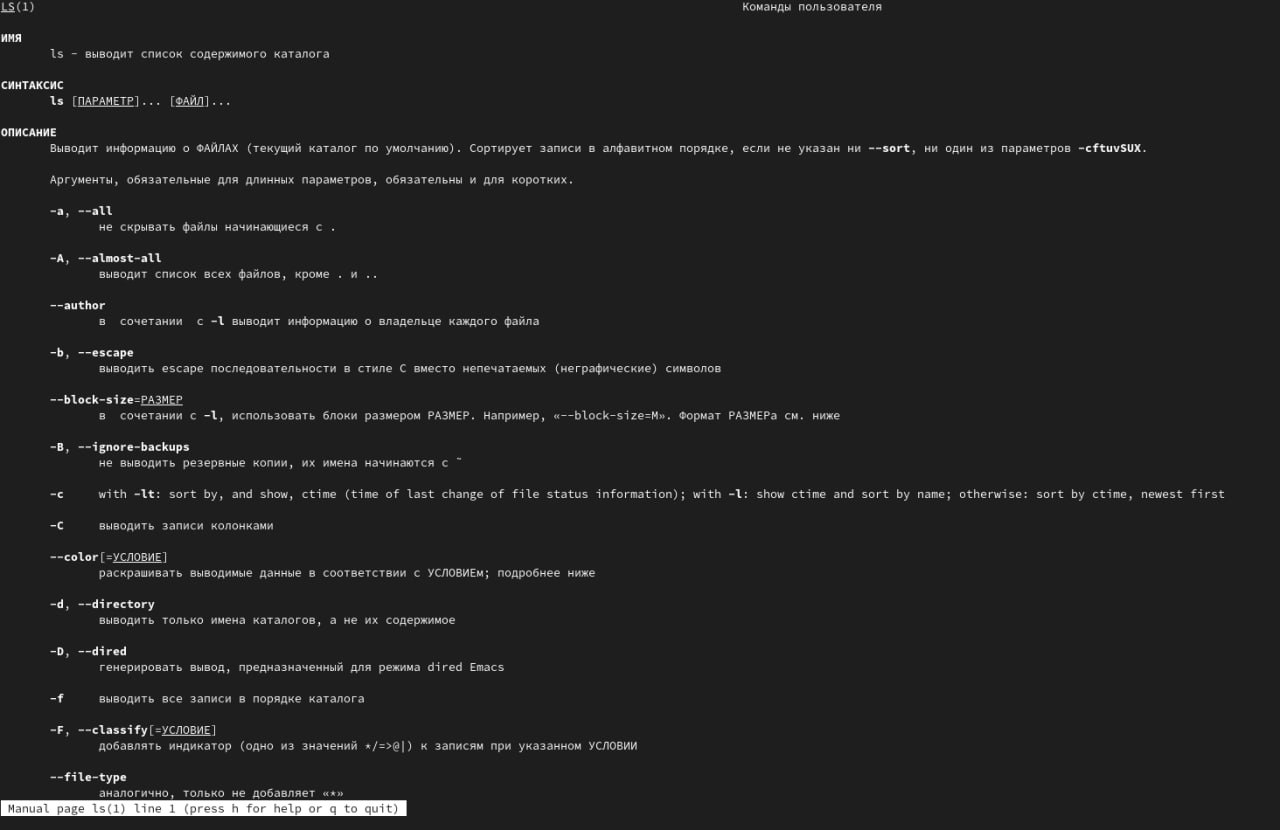


Рисунок 7 – Реакция на команду man ls

1. Написал программу на языке C++ для вывода простого сообщения на экран с использованием текстового редактора joe.



Рисунок 8 – Компиляция и вывод программы

**Индивидуальное задание:**

Написал программу на языке C++ с использованием текстового редактора nano.



Рисунок 9 – интерфейс программы nano

Запустил и протестировал программу с вводом и выводом в файл.

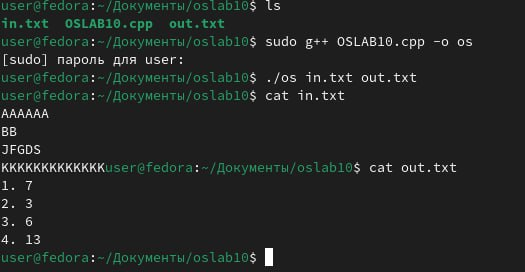


Рисунок 10 – интерфейс программы nano

**Исходный код пограммы:**

#include <stdio.h>

#include <stdbool.h>

#include <stdarg.h>

#define BUF\_SIZE 900

using namespace std;

int myfprintf(FILE \*f, const char \*fmt, ...) {

va\_list ptr;

va\_start(ptr, fmt);

char str[BUF\_SIZE];

int n = vsnprintf(str, BUF\_SIZE, fmt, ptr);

for (size\_t i=0; i < BUF\_SIZE; i++) {

if (str[i] == 0) break;

fputc(str[i], f);

}

va\_end(ptr);

return n;

}

int main(int argc, char\* argv[]) {

if (argc < 3) {

myfprintf(stderr, "Usage: %s <input\_file> <output\_file>\n", argv[0]);

return 1;

}

FILE \*input\_file = fopen(argv[1], "r");

if (input\_file == NULL) {

myfprintf(stderr, "%s:Couldn't open file\n", argv[1]);

return 2;

}

FILE \*output\_file = fopen(argv[2], "w");

if (output\_file == NULL) {

myfprintf(stderr, "%s:Couldn't open file\n", argv[2]);

fclose(input\_file);

return 2;

}

int line\_count = 1;

int line\_char\_count = 0;

int char\_count = 0;

int c;

c = getc(input\_file);

while (c != EOF)

{

if (c == '\n') {

myfprintf(output\_file, "%i. %i\n", line\_count, line\_char\_count);

line\_count++;

char\_count += line\_char\_count;

line\_char\_count = 0;

} else {

line\_char\_count++;

}

c = getc(input\_file);

}

myfprintf(output\_file, "%i. %i\n", line\_count, line\_char\_count);

char\_count += line\_char\_count;

myfprintf(output\_file, "In file %i lines and %i characters", line\_count, char\_count);

fclose(input\_file);

fclose(output\_file);

}

**Вывод**

В ходе лабораторной работы изучил команды ОС Fedora для работы с файлами, каталогами идисками; текстовые редакторы joe и nano и программу для мониторинга процессов htop. Написал программу на языке C++ работающую с текстовыми файлами.