МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

ИНСТИТУТ ЦИФРОВЫХ ТЕХНОЛОГИЙ, ЭЛЕКТРОНИКИ И ФИЗИКИ (ИЦТЭФ)

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ЭЛЕКТРОНИКИ (ВТиЭ)

**Отчет по лабораторной работе № 2**

по курсу “Системное программное обеспечение”

**Вариант №10**

Выполнил студент 506 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В. Осипенко

Проверил: преп. кафедры ВТиЭ.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Сеулеков

Лабораторная работа защищена

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2023

# Цель работы:

Получение навыков работы с процессами в Linux.

# **Задачи:**

Создать директорию lab1, в ней создать файл программы, две директории: input; output. Программа должна считать содержимое файлов, находящихся в директории input, провести необходимые вычисления и записать результат в директорию output. Родительский процесс запускает программы из лабораторной № 1, затем программу для данной лабораторной работы.

# Задание:

Вариант 10 – Найти среди входных файлов файл, содержащий минимальное количество пробельных символов (пробелов и табуляций), записать в выходной файл это количество и название этого файла. Если таких файлов несколько, записать все их имена.

# **Выполнение работы:**

В директорию input помещаем 3 файла:

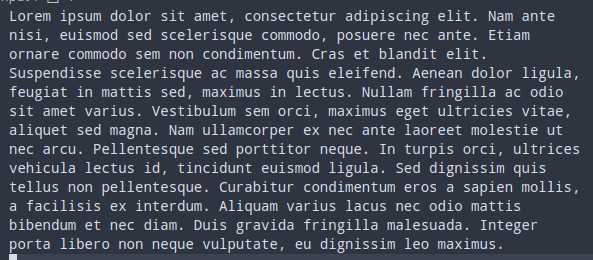


Рис 1.1 Содержимое файла 1.

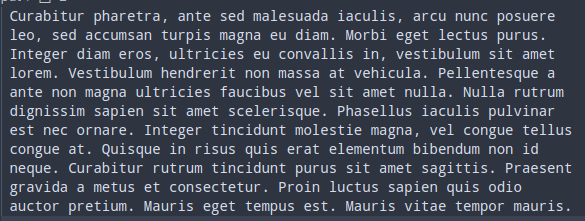


Рис 1.2 Содержимое файла 2.

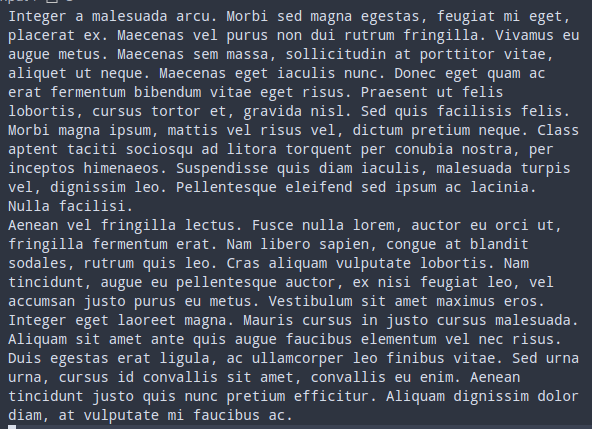


Рис 1.3 Содержимое файла 3.

Далее происходит разработка программы на языке ZIG. Для доступа к директориям, файлам и создания дочерних процессов используются функции из стандартной библиотеки, которые являются обертками системный вызовов определенной операционной системы.

Далее приведен код родительской программы, создающий вначале дочерний процесс на основе программы из 1 лабораторной работы, затем из текущей:

const std = @import("std");

pub fn main() !void {

var arg1 = [\_][]const u8{"./../lab1/lab1"}; // набор параметров для 1 процесса

var arg2 = [\_][]const u8{"./lab2"}; // набор параметров для 2 процесса

// создания 1 и 2 процесса

var proc1 = std.process.Child.init(&arg1, std.heap.page\_allocator);

var proc2 = std.process.Child.init(&arg2, std.heap.page\_allocator);

// вызов и ожидание выполнения 1 процесса из лаб. 1

try proc1.spawn();

\_ = try proc1.wait();

// вызов и ожидание выполнения 2 процесса из лаб. 2

try proc2.spawn();

\_ = try proc2.wait();

}

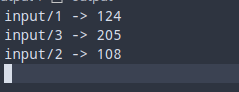


Рис. 1.4 Результат работы 1 процесса.



Рис. 1.5 Результат работы 2 процесса.

Далее приведен код программы из текущей лабораторной, который читает результат работы программы из 1 лабораторной, находит файл с наименьшим количеством пробелов и табуляций и записывает результат в новый выходной файл:

const std = @import("std");

const fs = std.fs;

const string = []const u8;

pub fn main() !void {

var out1 = try fs.cwd().openFile("output/output", .{ .mode = .read\_only });

var out2 = try fs.cwd().openFile("output/output2", .{ .mode = .write\_only });

defer out1.close();

defer out2.close();

var data = try out1.readToEndAlloc(std.heap.page\_allocator, 65536);

var it = std.mem.split(u8, data, "\n");

var min: i32 = -1;

var nameList = std.ArrayList(string).init(std.heap.page\_allocator);

while (it.next()) |row| {

if (row.len == 0) break;

var towIt = std.mem.split(u8, row, " -> ");

var name = towIt.next().?;

var value = try std.fmt.parseInt(i32, towIt.next().?, 10);

if (min == -1 or min > value) {

min = value;

nameList.clearAndFree();

try nameList.append(name);

} else if (min == value) {

try nameList.append(name);

}

}

try std.fmt.format(out2.writer(), "{d} -> {s}\n", .{ min, nameList.items });

}

# Вывод:

В ходе выполнения лабораторной работы были получили навыки работы с процессами в Linux.