

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет безопасности информационных технологий

Дисциплина:

«Основы системного программирования»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1.1

«Работа с файлами и каталогами»

Вариант 7

Выполнили:

Бардышев Артём,
студент группы N3246

(подпись)

Проверил:

Грозов В. А.,
преподаватель практики

(отметка о выполнении)

(подпись)

Санкт-Петербург 2025 г.

ВВЕДЕНИЕ

Разработать на языке C для ОС Linux программу, позволяющую выполнять

рекурсивный поиск файлов, начиная с указанного каталога, в соответствии с условием из

Табл. 3 и вариантом из Табл. 4.

Программа должна представлять собой консольную утилиту, настройка работы

которой осуществляется путем передачи аргументов в строке запуска и/или с помощью

переменных окружения (опции необязательны, аргументы каталог и цель_поиска — обязательны):

lab11abcNXXXXX [опции] каталог цель_поиска

Программа должна выполнять рекурсивный поиск файлов, отвечающих критерию,

который задается аргументом цель_поиска в соответствии с условием из Табл. 3. При

обнаружении файла, отвечающего заданным критериям поиска, программа должна вывести в

стандартный поток вывода полный путь к этому файлу.

При указании опций -h или -v (или их "длинных" аналогов --help или --version)

выполняется вывод информации, заданной опцией, и работа программы завершается. Опции,

которые должны поддерживаться программой, приведены в Табл. 1.

При определении переменной окружения LAB11DEBUG в стандартный поток ошибок

должна выводиться информация о том, что и в каком месте файла нашлось (чтобы было

легче понять, почему файл отвечает критериям поиска), а также может выводиться любая

дополнительная отладочная информация. Переменные окружения, которые должны поддерживаться программой, приведены в Табл. 2.

Имя программы должно начинаться на `lab11`, далее должен следовать уникальный для

варианта суффикс. Уникальный суффикс составляется из первых букв имени, отчества (если

есть) и фамилии студента, выполняющего лабораторную работу. Далее следует номер

группы студента. Используются строчные латинские буквы и арабские (в традиционном

понимании, т. е. 0..9) цифры. Например, если студента, выполняющего лабораторную, зовут

Петр Сергеевич Иванов, его группа — N32451, то имя программы должно быть

`lab11psiN32451`.

Проект (исходные коды, заголовочные файлы, `Makefile` и прочие файлы, необходимые для сборки) должен содержаться в отдельном каталоге с именем, совпадающим с названием

программы (`lab11abcNXXXXXX`) и собираться с помощью стандартной утилиты `make`.

Исходные файлы программы на языке C должны компилироваться с помощью `gcc`. `Makefile`

должен поддерживать как минимум цели `all` и `clean`. Если для сборки проекта требуется что-

то большее, чем `make all`, или для запуска и проверки проекта требуются какие-либо

нетривиальные или неочевидные действия, то инструкции по сборке и запуску проекта

следует добавить в файл `README.txt` в формате `plain text` и разместить его в каталоге проекта.

РАЗРАБОТКА ПРОГРАММЫ

Опция	Назначение
<code>-v, --version</code>	Вывод версии программы и информации о программе (ФИО исполнителя, номер группы, номер варианта лабораторной).
<code>-h, --help</code>	Вывод справки по опциям.

Переменная	Назначение
<code>LAB11DEBUG</code>	Включение вывода отладочной информации.

Количество букв в фамилии студента, выполняющего работу, четное.	Выполняется поиск заданной последовательности байтов. Аргумент <code>цель_поиска</code> имеет формат строки в кодировке UTF-8. <i>Пример запуска программы:</i> <code>./lab11psiN32451 /home "лавандовый раф"</code> (Начиная с каталога <code>/home</code> , рекурсивно выполняется поиск файлов, содержащих последовательность байтов <code>0xd0 0xbb</code> ('л'), <code>0xd0 0xb0</code> ('а'), ..., <code>0xd1 0x84</code> ('ф').)
--	--

<code>opendir()/readdir()</code> с рекурсией
--

1.7.1 Makefile

```
# Указываем компилятор и флаги компиляции
CC = gcc
CFLAGS = -Wall -Wextra -Werror -O3

# Имя исполняемого файла
TARGET = lab11aabN3246

# Все цели, которые нужно собрать
all: $(TARGET)

# Правило для сборки исполняемого файла
$(TARGET): lab11aabN3246.c
    $(CC) $(CFLAGS) -o $(TARGET) lab11aabN3246.c

# Очистка проекта (удаление временных и исполняемых файлов)
clean:
    rm -f $(TARGET)

# Запуск программы под valgrind
valgrind: $(TARGET)
valgrind --leak-check=full --log-file=valgrind.txt ./$(TARGET)
```

ТЕСТИРОВАНИЕ



```
(kali@kali)-[~/Desktop/Labs/lab11aabN3246]
$ ./lab11aabN3246 ~/Desktop "Iron Man"
Found: /home/kali/Desktop/TEST/3.txt
```

valgrind.txt :

```
==56602== Memcheck, a memory error detector
==56602== Copyright (C) 2002-2024, and GNU GPL'd, by Julian Seward et al.
==56602== Using Valgrind-3.24.0 and LibVEX; rerun with -h for copyright info
==56602== Command: ./lab11aabN3246
==56602== Parent PID: 56601
==56602==
==56602==
==56602== HEAP SUMMARY:
==56602==      in use at exit: 0 bytes in 0 blocks
==56602==    total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==56602==
==56602== All heap blocks were freed -- no leaks are possible
==56602==
==56602== For lists of detected and suppressed errors, rerun with: -s
==56602== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

ЗАКЛЮЧЕНИЕ

В рамках лабораторной работы была разработана консольная утилита на C для Linux, выполняющая рекурсивный поиск файлов по заданному критерию. Реализована поддержка обязательных аргументов, опций `--help` и `--version`, а также переменной окружения `LAB11DEBUG` для вывода отладочной информации. Проект собирается с помощью `make`, реализованы цели `all` и `clean`. Программа успешно проходит тестирование и соответствует заданным требованиям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Керниган, Брайан У., Ритчи, Деннис М.: Язык программирования С, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс» – 2009. – 304 с.: ил. – Парал. тит. англ. – URL: [Керниган, Ритчи. Язык программирования С.pdf - Google Диск](#)
2. Гирик А. В.: Основы системного программирования. Язык С. Сборка и отладка программ. Системные вызовы. Обработка ошибок. – Университет ИТМО – 2025. – URL: [01. ОСП. Язык С. Сборка и отладка программ. Системные вызовы. Обработка ошибок.pdf - Google Диск](#)
3. Гирик А. В.: Основы системного программирования. Файлы и каталоги. Управление памятью – Университет ИТМО – 2025. – URL: [02. ОСП. Файлы и каталоги. Управление памятью.pdf - Google Диск](#)