

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Операционные системы и системное программирование

ОТЧЕТ

по лабораторной работе №1

на тему

“Знакомство с Linux/Unix и средой программирования. POSIX-совместимая
файловая система.”

Выполнил:

студ. гр. 350501 Чехович Д. С.

Проверил:

ст.пр. Поденок Л. П.

МИНСК 2025

СОДЕРЖАНИЕ

1.ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ.....	3
2.ВЫПОЛНЕНИЕ РАБОТЫ.....	4
2.1 Описание алгоритма выполнения работы.....	4
2.2 Описание основных функций.....	4
2.3 Описание порядка сборки и использования.....	5
3.РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ.....	6
4.ВЫВОД.....	7

1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ

Целью работы является разработка программы `dirwalk`, которая сканирует файловую систему и выводит в `stdout` информацию в соответствии с опциями программы.

Исходные данные к работе:

Формат вывода аналогичен формату вывода утилиты `find`.

`dirwalk [dir] [options]` – `dir` начальный каталог.

Опции [options]:

- `l` только символические ссылки (`-type l`)
- `d` только каталоги (`-type d`)
- `f` только файлы (`-type f`)
- `s` сортировать выход в соответствии с – `LC_COLLATE`

Опции могут быть указаны как перед каталогом, так и после.

Опции могут быть указаны как отдельно, так и вместе (`-l -d, -ld`).

Если опции `ldf` опущены, выводятся каталоги, файлы и ссылки.

Для обработки опций рекомендуется использовать `getopt(3)`.

Программа должна быть переносимой (возможности `linux` не используются, только `POSIX`).

2 ВЫПОЛНЕНИЕ РАБОТЫ

2.1 Описание алгоритма выполнения работы

Программа предназначена для рекурсивного обхода директорий и вывода списка файлов, директорий и символических ссылок в зависимости от заданных параметров. Она начинается с инициализации локали для корректной сортировки строк. Затем программа анализирует переданные аргументы командной строки, чтобы определить, какие типы файлов нужно выводить (символические ссылки, директории, файлы) и нужно ли сортировать результат. После этого программа рекурсивно обходит указанную директорию, проверяя тип каждого найденного элемента с помощью функции `match_type`. Если элемент соответствует заданным критериям, его путь сохраняется в массиве. По завершении обхода программа сортирует массив (если это указано) и выводит результаты на экран.

2.2 Описание основных функций

Программа состоит из нескольких подпрограмм (частей программы), представляющих собой некоторые функции. К ним относятся функции:

1. Проверка файлов по типу
2. Обход директории
3. Вывод файлов
4. Реализация отсутствия обязательного порядка опций и пути

Функция `match_type()`

Данная функция предназначена для фильтрации файлов и директорий в зависимости от их типа и установленных флагов `show_links`, `show_dirs`, `show_files`.

С помощью макросов функция проверяет установлены ли флаги на 1 и является ли файл тем или иным типом. Если бы одно условие истинно, функция возвращает 1, указывая, что данный файл соответствует выбранному типу.

Если ни один из флагов не установлен, то функция вернет 1 и укажет на дальнейший вывод всех файлов.

Функция `dirwalk()`

Данная функция выполняет обход указанного каталога и собирает список файлов и папок, удовлетворяющих условиям, заданным флагами. Результаты сохраняются в массив `files`, а количество найденных файлов в переменную `count`.

1. Функция `opendir()` открывает указанную директорию, если директорию открыть не удалось, то выводится сообщение об ошибке.
2. Функция `readdir()` читает содержимое директории.
3. Пропускаются родительские директории.
4. Функция `sprint()` формирует полный путь к файлу.
5. Функция `stat()` получает информацию о файле.
6. Фильтруются файлы по типу.
7. Рекурсивно обходятся вложенные директории.
8. Закрытие переданной директории.

Функция `compare ()`

Функция сравнения для сортировки строк с использованием `strcoll`. Она используется в `qsort` для сортировки путей файлов.

Функция `main ()`

Основная функция программы. Обработывает аргументы командной строки, инициализирует обход директории, сортирует результаты (если нужно) и выводит их на экран.

2.3 Описание порядка сборки и использования

Для сборки проекта используется утилита `make`, которая автоматически компилирует исходный код и создает исполняемый файл. Исполняемый файл помещается в отдельную директорию `build`, что упрощает организацию проекта и поддержание порядка. Если директория `build` отсутствует, она создается автоматически в процессе сборки.

Для удаления скомпилированного файла и очистки директории `build` предусмотрена команда `make clean`. Это позволяет поддерживать рабочую среду в чистоте и гарантировать корректность последующих сборок.

3 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

```
dimin@dimin-Vivobook-ASUSLaptop-M6500QC-M6500QC:~/term_4/OSSP/
lab01$ dirwalk -d
./src
./test
./.git
./.git/refs
./.git/refs/tags
./.git/refs/heads
./.git/branches
./.git/objects
./.git/objects/dc
./.git/objects/pack
./.git/objects/07
./.git/objects/54
./.git/objects/c5
./.git/objects/73
./.git/objects/72
./.git/objects/info
./.git/info
./.git/logs
./.git/logs/refs
./.git/logs/refs/heads
./.git/hooks
```

Рисунок 1 – параметры d

```
dimin@dimin-Vivobook-ASUSLaptop-M6500QC-M6500QC:~/term_4/OSSP/
lab01$ dirwalk /home/dimin/term_4/OSSP/lab01/ -ds
/home/dimin/term_4/OSSP/lab01/build
/home/dimin/term_4/OSSP/lab01/.git
/home/dimin/term_4/OSSP/lab01/.git/branches
/home/dimin/term_4/OSSP/lab01/.git/hooks
/home/dimin/term_4/OSSP/lab01/.git/info
/home/dimin/term_4/OSSP/lab01/.git/logs
/home/dimin/term_4/OSSP/lab01/.git/logs/refs
/home/dimin/term_4/OSSP/lab01/.git/logs/refs/heads
/home/dimin/term_4/OSSP/lab01/.git/objects
/home/dimin/term_4/OSSP/lab01/.git/objects/07
/home/dimin/term_4/OSSP/lab01/.git/objects/54
/home/dimin/term_4/OSSP/lab01/.git/objects/72
/home/dimin/term_4/OSSP/lab01/.git/objects/73
/home/dimin/term_4/OSSP/lab01/.git/objects/c5
/home/dimin/term_4/OSSP/lab01/.git/objects/dc
/home/dimin/term_4/OSSP/lab01/.git/objects/info
/home/dimin/term_4/OSSP/lab01/.git/objects/pack
/home/dimin/term_4/OSSP/lab01/.git/refs
/home/dimin/term_4/OSSP/lab01/.git/refs/heads
/home/dimin/term_4/OSSP/lab01/.git/refs/tags
/home/dimin/term_4/OSSP/lab01/src
```

Рисунок 2 – пример использования двух параметров сразу и параметра s

ВЫВОД

При выполнении лабораторной работы была изучена работа с файлами в операционной системе Linux.

В ходе выполнения работы была разработана программа, которая способна сканировать файлы и выводить в поток вывода информацию в соответствии с опциями программы.