Министерство образования Республики Беларусь

Учреждение Образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа №1

по дисциплине «Операционные системы и системное программирование»

на тему «Знакомство с Linux/Unix и средой программирования. POSIX-совместимая файловая система.»

|  |  |  |
| --- | --- | --- |
| Выполнил  студент гр. 250505  Рудько Т.А. |  | Проверил  Басак Д.В. |

Минск 2024

# УСЛОВИЕ ЛАБОРАТОРНОЙ РАБОТЫ

## Освоить эффективную работу с файлами в оболочке и mc.

Разработать программу dirwalk, сканирующую файловую систему и выводящую в stdout

информацию в соответствии с опциями программы.

Формат вывода аналогичен формату вывода утилиты find.

dirwalk [dir] [options]

dir ‒ начальный каталог. Если опущен, текущий (./).

options ‒ опции.

-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 раздельно, так и вместе (-l -d, -ld ).

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

### **ОПИСАНИЕ АЛГОРИТМОВ И РЕШЕНИЙ**

В данной программе реализованы следующие алгоритмы и решения:

1. Сканирование каталога:
   * Используется рекурсивный обход файлов и подкаталогов в указанном каталоге.
   * Для сканирования используется функция scan\_file, которая принимает параметры, включая путь к текущему каталогу, и рекурсивно вызывает саму себя для обработки вложенных элементов.
2. Параметры командной строки:
   * Программа обрабатывает параметры командной строки, заданные при запуске.
   * Используется библиотека getopt для анализа опций командной строки.
   * Параметры включают в себя:
     + -l: Включение символических ссылок в результате.
     + -d: Включение директорий в результате.
     + -f: Включение обычных файлов в результате.
     + -s: Сортировка результатов в алфавитном порядке.
3. Сортировка результатов:
   * Результаты сканирования сортируются в алфавитном порядке при необходимости.
   * Используется стандартная функция сортировки qsort с собственной функцией сравнения строк compare\_strings.
4. Структуры данных:
   * Программа использует структуры данных для хранения информации о каждом элементе (файле/каталоге).
   * item\_t - структура, представляющая элемент с полями path (путь к элементу) и type (тип элемента: файл, символическая ссылка, директория).
   * parameters\_t - структура, представляющая параметры командной строки с полями path (путь к каталогу), symlinks, catalogs, files и sort.
5. Освобождение памяти:
   * Программа заботится об освобождении выделенной памяти для предотвращения утечек.
   * Функция free\_items освобождает память, выделенную для структур item\_t.

Эти алгоритмы и решения обеспечивают эффективное сканирование каталога с учетом заданных параметров и предоставляют удобный интерфейс для работы с результатами.

**ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОЕКТА**

Проект лабораторной работы организован с учетом модульного подхода, что позволяет легко поддерживать, расширять и изменять код. В нем выделяются следующие модули:

1. main.c:
   * Главный модуль программы, отвечающий за взаимодействие с пользователем, обработку параметров командной строки и вызов основных функций.
2. utils.c/.h:
   * Модуль, содержащий вспомогательные функции, такие как выделение и освобождение памяти, форматированный вывод ошибок и другие утилиты.
3. parameters\_t.h:
   * Заголовочный файл, описывающий структуру parameters\_t, представляющую параметры командной строки.
4. item\_t.h/.c:
   * Модуль, отвечающий за структуру item\_t, представляющую информацию о каждом элементе (файле/каталоге) и функции для работы с этой структурой.
5. item\_t\_funcs.h/.c:
   * Модуль, содержащий функции для создания, печати и освобождения элементов.
6. Makefile:
   * Файл для автоматизации сборки проекта. Содержит правила для компиляции и линковки различных модулей, а также цели для запуска и очистки проекта.
7. headers/:
   * Каталог, содержащий заголовочные файлы с описаниями структур и функций для обеспечения модульности и читаемости кода.
8. src/:
   * Каталог, содержащий исходные файлы программы.

Такая структура проекта способствует легкости поддержки, тестирования и расширения функциональности программы. Каждый модуль выполняет свою конкретную задачу, что делает код более читаемым и понятным для разработчиков.

# Сборка проекта и использование

# Сборка проекта

1. Системные требования:

Для успешной сборки и запуска проекта требуется наличие компилятора GCC и установленной утилиты make.

1. Клонирование репозитория:

Склонируйте репозиторий на ваш компьютер:

git clone https://github.com/TimoxGagarin/dirwalk

1. Переход в каталог с проектом:

Перейдите в каталог с загруженным репозиторием:

cd <название каталога>

* Сборка проекта:
* Выберите режим сборки (debug или release) и выполните команду make:

make MODE=debug # для режима отладки

make MODE=release # для режима релиза

* Запуск программы:

После успешной сборки запустите программу, указав необходимые параметры командной строки:

./dirwalk [-l] [-d] [-f] [-s] [каталог]

# Описание параметров командной строки

* -l: Включить символические ссылки в результате.
* -d: Включить директории в результате.
* -f: Включить обычные файлы в результате.
* -s: Отсортировать результат в алфавитном порядке.

Если каталог не указан, будет выполнено сканирование текущего каталога.

# Примеры использования:

Сканирование текущего каталога с включенными директориями и отсортированным результатом:

./dirwalk -d -s

* Сканирование определенного каталога с включенными символическими ссылками и обычными файлами:

./dirwalk -l -f /путь/к/каталогу

Пожалуйста, обратите внимание, что параметры могут быть указаны в любом порядке.

# Метод тестирования и результаты

# Метод тестирования

Для тестирования программы были использованы следующие подходы:

1. Положительные тесты:
   * Запуск программы с различными комбинациями параметров для проверки корректности работы с различными типами файлов и директорий.
   * Сравнение результатов работы программы с ожидаемыми результатами в соответствии с указанными параметрами.
2. Негативные тесты:
   * Запуск программы с некорректными параметрами командной строки для проверки обработки ошибок и вывода соответствующих сообщений.
   * Попытка запуска программы с некорректным или несуществующим путем каталога для проверки обработки таких ситуаций.

# Результаты тестирования

По результатам тестирования программы были выявлены следующие аспекты:

1. Положительные результаты:
   * Программа успешно сканирует указанный каталог и выводит список файлов и директорий в соответствии с указанными параметрами.
   * Результаты работы программы соответствуют ожидаемым результатам для различных комбинаций параметров.
2. Негативные результаты:
   * Программа корректно обрабатывает некорректные параметры командной строки и выводит соответствующие сообщения об ошибке.
   * При попытке запуска программы с некорректным путем каталога выводится сообщение об ошибке и программа завершает свою работу.

# Общий вывод

Программа успешно прошла тестирование и демонстрирует корректное поведение при различных сценариях использования. Все функциональные возможности программы работают ожидаемым образом, а обработка ошибок и вывод сообщений пользователю реализованы адекватно.