

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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

«Корзина» для программ, использующих системный вызов `unlink()`

БГУИР КП 1-40 02 01 123 ПЗ

Студент:

Ткаченко И.Д.

Руководитель:

Старший преподаватель
Поденок Л.П.

Минск 2023

Учреждение образования
«Белорусский государственный университет информатики
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ

(подпись)

2023 г.

ЗАДАНИЕ
по курсовому проектированию

Студенту Ткаченко Илье Дмитриевичу

1. Тема проекта «Корзина» для программ, использующих системный вызов `unlink()`.
2. Срок сдачи студентом законченного проекта 25 мая 2023 г.
3. Исходные данные к проекту Язык программирования – Си
4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)
Введение. 1.Постановка задачи. 2.Обзор литературы. 2.Системное проектирование. 3.Функциональное проектирование. 4.Разработка программных модулей. 5.Программа и методика испытаний. 6.Руководство пользователя. Заключение. Литература
5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков) 1. Листинг кода.
6. Консультант по проекту (с обозначением разделов проекта) Л.П. Поденок
7. Дата выдачи задания 15 февраля 2023 г.
8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):
разделы 1,2 к 15 марта 2023 г. – 20 %;
разделы 3,4 к 15 апреля 2023 г. – 30 %;

разделы 5,6,7 к 15 мая 2023 г. – 30 %;

оформление пояснительной записки и графического материала к 25 мая 2023 г. – 20 %

Защита курсового проекта с 5 июня 2023 г. по 10 июня 2023 г.

РУКОВОДИТЕЛЬ

Л.П. Поденок

(подпись)

Задание принял к исполнению

И.Д. Ткаченко

(дата и подпись студента)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 ОБЗОР ЛИТЕРАТУРЫ.....	6
1.1 Обзор методов и алгоритмов решения поставленной задачи.....	6
1.2 Постановка задачи.....	7
2. СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ.....	7
3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ.....	7
4. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ.....	7
5. ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ.....	7
6. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	9
ЗАКЛЮЧЕНИЕ.....	9
СПИСОК ЛИТЕРАТУРЫ.....	10
ПРИЛОЖЕНИЕ А.....	11
ПРИЛОЖЕНИЕ Б.....	12
ПРИЛОЖЕНИЕ В.....	15

ВВЕДЕНИЕ

В наше время удаление файлов из компьютера может вызвать серьезные проблемы. Не всегда пользователь хочет окончательно избавиться от файла, но случайно нажатие кнопки "Удалить" может привести к потере важной информации. Для решения этой проблемы была создана функциональность "Корзина", которая позволяет пользователям перемещать файлы в специальную папку, где они будут храниться до момента окончательного удаления.

В данном курсовом проекте мы будем перехватывать системные вызовы `unlink()` и `unlinkat()` на языке программирования Си, для реализации функционала "Корзина". Мы будем создавать программу, которая будет перемещать файлы в корзину вместо их окончательного удаления, что позволит пользователю легко восстановить удаленные файлы в случае необходимости.

Наша программа будет включать в себя создание папки "Корзина" при помощи функции `mkdir()` в домашней директории пользователя, а также файл журнала, в котором будут храниться записи о перемещенных в корзину файлах при помощи функций `fopen()` и `fprintf()`. Также мы будем использовать системный вызов `rename()` для перемещения файлов в корзину.

Цель данного курсового проекта - написание программы "Корзина" на языке программирования Си, которая позволит пользователям безопасно удалять файлы, а также легко восстанавливать их в случае необходимости.

1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Обзор методов и алгоритмов решения поставленной задачи

Для решения поставленной задачи создания программы "Корзина" на языке программирования Си необходимо использовать следующие методы и алгоритмы:

Работа с файловой системой - для создания папки "Корзина", файла журнала, перемещения файлов в корзину, получения переменных среды необходимо использовать методы работы с файловой системой на языке Си. Для этого можно использовать стандартную библиотеку функций Си, такие как `mkdir()`, `rename()`, `getenv()`.

Обработка ошибок - при работе с файловой системой могут возникать различные ошибки, например, файл не найден или папка уже существует. Для обработки ошибок необходимо использовать стандартную библиотеку Си и проверять результат выполнения функций на наличие ошибок.

Работа со строками - для работы с путями к файлам, а также для записи данных в файл журнала, необходимо использовать функции работы со строками на языке Си, такие как `sprintf()` и `strcat()`, `strcpy()`.

Взаимодействие с пользователем - для обеспечения удобства использования программы, необходимо предусмотреть интерфейс, который будет взаимодействовать с пользователем. Для этого можно использовать стандартный ввод/вывод на языке Си.

Работа с датой и временем - для создания уникальных имен файлов в корзине и для записи времени удаления файлов в файл журнала, можно использовать функции работы с датой и временем, такие как `time()` и `localtime()`, `asctime()`.

Обработка команд пользователя - для обеспечения функциональности программы, необходимо обрабатывать команды пользователя, такие как перемещение файлов в корзину, восстановление файлов из корзины и удаление файлов из нее. Для этого можно использовать условные операторы и циклы на языке Си.

1.2 Постановка задачи

1. Создание папки "Корзина" в домашней директории пользователя.
2. Создание файла журнала, в котором будут храниться записи о перемещенных в корзину файлах.

3. Реализация функции перемещения файлов в корзину при помощи системного вызова `rename()`.
4. Реализация функции удаления файлов из корзины при помощи системного вызова `unlink()`.
5. Реализация функции восстановления файлов из корзины в их первоначальное место при помощи системного вызова `rename()`.
6. Добавление возможности просмотра содержимого корзины и списка перемещенных в нее файлов при помощи функции `ls()`.
7. Реализация функции очистки корзины, при которой все файлы будут окончательно удалены из системы.
8. Обработка ошибок, которые могут возникать при работе с файлами и папками, а также сообщение пользователю о всех действиях, производимых программой.
9. Цель данного проекта - создание полноценной программы "Корзина", которая будет обеспечивать безопасное удаление файлов и их легкое восстановление в случае необходимости. Программа должна быть надежной, удобной в использовании и иметь простой и понятный интерфейс для пользователя.

2. СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

4. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

5. ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

Работа с программой осуществляется следующим образом в терминале:

```
[safertao@safertao course-work]$ touch 1 2 3
[safertao@safertao course-work]$ LD_PRELOAD=./build/unlink.so rm
1 2 3
1 was succesfully moved /home/safertao/trash by unlinkat syscall
2 was succesfully moved /home/safertao/trash by unlinkat syscall
```

```
3 was succesfully moved /home/safertao/trash by unlinkat syscall
[safertao@safertao course-work]$ LD_PRELOAD=./build/unlink.so
unlink test

test was succesfully moved /home/safertao/trash by unlink
syscall
```

В процессе выполнения программы был создан каталог `trash` в домашнем каталоге пользователя, а также файл `trash.log` со следующим содержимым:

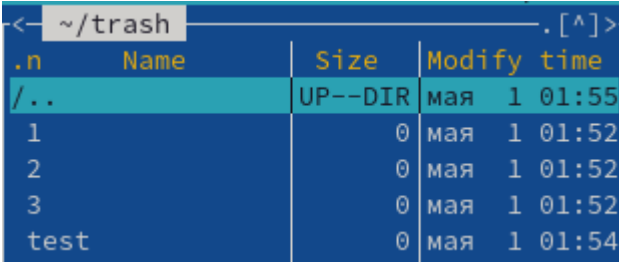
```
3 was moved to /home/safertao/trash by unlinkat syscall on Mon
May 1 01:53:07 2023

2 was moved to /home/safertao/trash by unlinkat syscall on Mon
May 1 01:53:07 2023

1 was moved to /home/safertao/trash by unlinkat syscall on Mon
May 1 01:53:07 2023

test was moved to /home/safertao/trash by unlink syscall on Mon
May 1 01:54:43 2023
```

Содержимое каталога `trash` представлено далее на рисунке 1.



.n	Name	Size	Modify time
/..	UP--DIR		мая 1 01:55
1		0	мая 1 01:52
2		0	мая 1 01:52
3		0	мая 1 01:52
test		0	мая 1 01:54

Рисунок 1 – Содержимое каталога `trash` после выполнения программы

6. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта были выполнены первоначально заданные цели, а именно была разработана и реализована функциональность корзины для программ, использующих системный вызов `unlink()`. Корзина представляет собой механизм сохранения удаленных файлов и возможность их восстановления при необходимости. Также были выполнены следующие задачи: был создан файл журнала `trash.log` для информации о времени удаления каждого файла, были обработаны ошибки, которые могут возникать при работе с файлами.

В современном мире удаление файлов с компьютера может вызвать серьезные проблемы. Пользователи не всегда желают окончательно избавляться от файлов, но случайное нажатие кнопки "Удалить" может привести к потере важной информации. Для решения этой проблемы была создана функциональность "Корзина". Она предназначена для перемещения файлов в специальную папку, где они будут храниться до момента окончательного удаления или восстановления в случае ошибочного удаления.

СПИСОК ЛИТЕРАТУРЫ

- Брайан Керниган, Деннис Ритчи. Язык программирования Си. Издательство: «Вильямс», 2019 г.
- W. Richard Stevens, Stephen A. Rago. UNIX. Профессиональное программирование. Издательство: «Вильямс», 2017 г.
- Robert Love. Linux System Programming: Talking Directly to the Kernel and C Library. Издательство: O'Reilly Media, 2013 г.
- Андрей Робачевский. Программирование на языке Си в UNIX. Издательство: БХВ-Петербург, 2015 г.
- Андрей Алексеев. Программирование на языке Си в среде Linux. Издательство: БХВ-Петербург, 2015 г.
- Майкл Керниган, Брайан В. Керниган. UNIX. Руководство системного программиста. Издательство: «Вильямс», 2018 г.
- Ричард Стивенс, Стивен Раго. Разработка приложений для UNIX. Издательство: «Питер», 2011 г.
- Алан А. А. Донован, Брайан У. Керниган. Регулярные выражения. Издательство: «Вильямс», 2019 г.
- The C Programming Language. Издательство: Prentice Hall, 1988 г.

ПРИЛОЖЕНИЕ А
(обязательное)

Схема структурная

ПРИЛОЖЕНИЕ Б

(обязательное)

Листинг кода

```
#define _GNU_SOURCE
#define MAX_PATH_LEN 4096
#define MAX_FILENAME_LEN 256

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include <errno.h>
#include <time.h>

int find_last_slash(const char *s);

void logger(const char *home_path, const char *path_name, const
char *function);

int unlink(const char *path_name)
{
    char new_path[MAX_PATH_LEN];
    char *home = getenv("HOME");
    if(!home)
    {
        fprintf(stderr, "ERROR: can't get home environment\n");
        exit(1);
    }
    sprintf(new_path, "%s/trash", home);
    mkdir(new_path, 0755);
    // если каталог корзины существует, ничего не делать
    int index = find_last_slash(path_name);
    strcat(new_path, "/");
    strcat(new_path, path_name+index); // присоединяем имя файла
    if(rename(path_name, new_path))
    {
        perror("ERROR");
        exit(errno);
    }
    logger(home_path, path_name, __func__);
    return 0;
}

int unlinkat(int dirfd, const char *path_name, int flags)
```

```

{
    char new_path[MAX_PATH_LEN];
    char *home = getenv("HOME");
    if(!home)
    {
        fprintf(stderr, "ERROR: can't get home environment\n");
        exit(1);
    }
    sprintf(new_path, "%s/trash", home);
    mkdir(new_path, 0755);
    // если каталог корзины существует, ничего не делать
    int index = find_last_slash(s);
    strcat(new_path, "/");
    strcat(new_path, path_name+index); // присоединяем имя файла
    if(rename(path_name, new_path))
    {
        perror("ERROR");
        exit(errno);
    }
    logger(home_path, path_name, __func__);
    return 0;
}

void logger(const char *home_path, const char *path_name, const
char *function)
{
    time_t rawtime;
    struct tm * timeinfo;
    time(&rawtime);
    timeinfo = localtime(&rawtime);
    char log_path[MAX_FILENAME_LEN];
    sprintf(log_path, "%s/trash.log", home_path);
    FILE *log = fopen(log_path, "a");
    if(!log)
    {
        fprintf(stderr, "ERROR: %s can't be opened/created\n");
        exit(1);
    }
    fprintf(log, "%s was moved to %s/trash by %s syscall on %s",
path_name, home_path, function, asctime(timeinfo));
    printf("%s was succesfully moved %s/trash by %s syscall\n",
path_name, home_path, function);
}

int find_last_slash(const char *s)
{
    int index = 0;

```

```
    int len = strlen(s);  
    // идем с конца строки до первого '/' для получения имени  
    удаляемого файла  
    for(int i = len - 1; i > 0; i--)  
    {  
        if(s[i] == '/')  
        {  
            index = i + 1;  
            break;  
        }  
    }  
    return index;  
}
```

ПРИЛОЖЕНИЕ В
(обязательное)
Ведомость документов