

Факультет «Информатика и системы управления»
Кафедра «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №2 по курсу:

«Операционные системы»

«Файлы и каталоги»

Студент группы ИУ7-63Б: Фурдик Н. О.
(Фамилия И.О.)

Преподаватель: Рязанова Н.Ю.
(Фамилия И.О.)

Оглавление

Задание	2
Постановка задачи	2
Листинг кода	2
Результат работы	6
Список литературы	9

Задание

Постановка задачи

Реализовать загружаемый модуль ядра, выводящий информацию о процессах.

Листинг кода

Листинг 1: Листинг кода заголовочного файла

```
#ifndef STACK_H
#define STACK_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define NMAX 100000

struct elem
{
    char *filename;
    int len;
};

struct stack
{
    struct elem elems[NMAX];
    int top;
};

void init(struct stack *stk);
void push(struct stack *stk, char *filename, int len);
struct elem pop(struct stack *stk);
int isempty(struct stack *stk);

#endif
```

Листинг 2: Листинг кода программы

```
#include <sys/stat.h>
#include <dirent.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <string.h>
#include <errno.h>
#include "stack.h"

#define FTW_F 1
#define FTW_D 2
#define FTW_DNR 3
#define FTW_NS 4

typedef int AnyType(const char *, int, int);
```

```

static AnyType myFunc;
static int myFtw(char *, AnyType *);
static int doPath(AnyType *, char*, int);
static struct stack stk;

void init(struct stack *stk)
{
    stk->top = 0;
}

void push(struct stack *stk, char *filename, int len)
{
    if (stk->top < NMAX)
    {
        stk->elems[stk->top].filename = calloc(strlen(filename), sizeof(char));
        strcpy(stk->elems[stk->top].filename, filename);
        stk->elems[stk->top].len = len;
        stk->top++;
    }
    else
        printf("Стек полон!\n");
}

struct elem pop(struct stack *stk)
{
    struct elem el;
    if ((stk->top) > 0)
    {
        stk->top--;
        el.filename = calloc(strlen(stk->elems[stk->top].filename), sizeof(char));
        strcpy(el.filename, stk->elems[stk->top].filename);
        free(stk->elems[stk->top].filename);
        el.len = stk->elems[stk->top].len;

        return el;
    }
    else
    {
        printf("Стек пуст!\n");
        return el;
    }
}

int isempty(struct stack *stk)
{
    if (stk->top == 0)
        return 1;
    else
        return 0;
}

static int myFtw(char *pathname, AnyType *func)

```

```

{
    if (chdir(pathname) == -1)
    {
        printf("Ошибка вызова функции chdir %s\n", pathname);
        return 1;
    }

    init(&stk);
    push(&stk, ".", 0);

    while (!isempty(&stk))
    {
        struct elem el = pop(&stk);
        doPath(func, el.filename, el.len);
    }

    return 0;
}

static int doPath(AnyType* func, char *fullpath, int len)
{
    if (len < 0)
    {
        if (chdir(fullpath) == -1)
        {
            printf("Каталога не существует!");
            return 1;
        }

        return 0;
    }

    struct stat statbuf;
    struct dirent *dirp;
    DIR *dp;

    if (lstat(fullpath, &statbuf) == -1)
    {
        return 1;
    }

    if (S_ISDIR(statbuf.st_mode) == 0)
    {
        return 1;
    }

    func(fullpath, FTW_D, len);

    if ((dp = opendir(fullpath)) == NULL)
    {
        return 1;
    }
}

```

```

    if (chdir(fullpath) == -1)
    {
        printf("Ошибка вызова функции chdir %s\n", fullpath);
        return 1;
    }

    len += 5;

    push(&stk, "..", -1);
    while ((dirp = readdir(dp)) != NULL)
    {
        if (strcmp(dirp->d_name, ".") != 0 && strcmp(dirp->d_name, "..") != 0 && strcmp(dirp->d_name, ".git") != 0)
        {
            push(&stk, dirp->d_name, len);
        }
    }

    if (closedir(dp) == -1)
        printf("Невозможно закрыть каталог %s\n", fullpath);

    return 0;
}

static int myFunc(const char *pathname, int type, int len)
{
    if (type == FTW_D)
    {
        for (int i = 0; i < len; i++)
        {
            if (i % 5 == 1 && i / 5 != 0)
                printf("|");
            else
                printf(" ");
        }

        printf(" |- %s\n", pathname);
    }

    return 0;
}

int main(int argc, char *argv[])
{
    if (argc != 2)
    {
        printf("Не указан <начальный_каталог!>\n");
        return 1;
    }

    return myFtw(argv[1], myFunc);
}

```

Результат работы

1. Скомпилируем программу и продемонстрируем вывод дерева каталогов, задав начальным каталог "~/"

```
schoolboychik@schoolboychik-VirtualBox:~/Рабочий стол/os/semestr2/lab2$ ./main.exe ~/
|- .
|   |- .ssh
|   |- Рабочий стол
|   |   |- os
|   |   |   |- semestr2
|   |   |   |   |- lab3
|   |   |   |   |   |- images
|   |   |   |   |   |- task1
|   |   |   |   |   |- task2
|   |   |   |   |- lab8
|   |   |   |   |   |- images
|   |   |   |   |- lab9
|   |   |   |   |   |- tasklet
|   |   |   |   |   |- images
|   |   |   |   |   |- workqueue
|   |   |   |   |- lab1
|   |   |   |   |- lab2
|   |   |   |   |- lab4
|   |   |   |   |   |- images
|   |   |   |   |- lab6
|   |   |   |   |   |- server
|   |   |   |   |   |- images
|   |   |   |   |   |- unix
|   |   |- Документы
|   |   |- .config
|   |   |   |- yelp
|   |   |   |- ibus
|   |   |   |   |- bus
|   |   |   |- nautilus
|   |   |   |- dconf
|   |   |- Общедоступные
|   |   |- snap
|   |   |   |- gnome-system-monitor
|   |   |   |- common
```

```

|   |   |   |- common
|   |   |   |   |- .cache
|   |   |   |   |   |- gio-modules
|   |   |   |   |   |- fontconfig
|   |   |   |   |   |- immodules
|   |   |   |   |- 127
|   |   |   |   |   |- .config
|   |   |   |   |   |   |- gtk-2.0
|   |   |   |   |   |   |- ibus
|   |   |   |   |   |   |- fontconfig
|   |   |   |   |   |   |- dconf
|   |   |   |   |   |   |- gtk-3.0
|   |   |   |   |   |- .local
|   |   |   |   |   |   |- share
|   |   |   |   |   |   |   |- icons
|   |   |   |   |   |   |   |- glib-2.0
|   |   |   |   |   |   |   |   |- schemas
|   |   |   |   |   |- 145
|   |   |   |   |   |   |- .config
|   |   |   |   |   |   |   |- gtk-2.0
|   |   |   |   |   |   |   |- ibus
|   |   |   |   |   |   |   |- fontconfig
|   |   |   |   |   |   |   |- dconf
|   |   |   |   |   |   |   |- gtk-3.0
|   |   |   |   |   |   |- .local
|   |   |   |   |   |   |   |- share
|   |   |   |   |   |   |   |   |- icons
|   |   |   |   |   |   |   |   |- glib-2.0
|   |   |   |   |   |   |   |   |   |- schemas
|   |   |   |   |- .cache
|   |   |   |   |   |- gnome-software
|   |   |   |   |   |   |- fwupd
|   |   |   |   |   |   |   |- remotes.d
|   |   |   |   |   |   |   |   |- lvfs
|   |   |   |   |   |- shell-extensions
|   |   |   |   |   |- odrs
|   |   |   |   |   |- icons
|   |   |   |   |   |- screenshots
```

```
| | | | | screenshots  
| | | | |- 624x351  
| | | | |- 112x63  
| | | | cssresource  
- yelp  
| | | WebKitCache  
| | | Version 16  
| | | Blobs  
- obexd  
- thumbnails  
| | large  
| | fail  
| | gnome-thumbnail-factory  
- ubuntu-report  
- gnome-control-center  
| | backgrounds  
| | wacom  
- ibus-table  
- ibus  
| | bus  
- update-manager-core  
- evolution  
| | sources  
| | trash  
| | tasks  
| | trash  
| | calendar  
| | trash  
| | mail  
| | trash  
| | memos  
| | trash  
| | addressbook  
| | trash  
- fontconfig  
- gstreamer-1.0  
- wallpaper  
- libgweather  
- Шаблоны  
- Загрузки
```

```

|- Изображения
|- Видео
|- Музыка
|- .gnupg
  |- private-keys-v1.d
|- .local
  |- share
    |- keyrings
    |- applications
    |- app-info
      |- xmls
    |- icc
    |- xorg
    |- ibus-table
    |- nautilus
      |- scripts
    |- gvfs-metadatas
    |- evolution
      |- tasks
        |- trash
        |- calendar
        |- trash
        |- system
      |- mail
        |- trash
      |- memos
        |- trash
      |- addressbook
        |- trash
        |- system
        |- photos
    |- Trash
      |- info
      |- expunged
      |- files
    |- flatpak
      |- db
    |- webkitgtk
      |- deviceidhashsalts
        |- 1

```

```
| | | - 1
| | | - databases
| | | | - indexeddb
| | | | | - v1
| | | - localstorage
| - sounds
| - gnome-settings-daemon
| - gnome-shell
```


2. Покажем ошибочные ситуации: запуск без указания начального каталога и с указанием несуществующего каталога.

```
schoolboychik@schoolboychik-VirtualBox:~/Рабочий стол/os/semestr2/lab2$ ./main.exe
Не указан <начальный_каталог>!
```

```
schoolboychik@schoolboychik-VirtualBox:~/Рабочий стол/os/semestr2/lab2$ ./main.exe sadsadasd
Каталога не существует!
```

3. Комментарии к программе:

- После обработки всех файлов в каталоге вызывается функция `chdir(...)`, чтобы использовать короткие имена.
- Функция `stat()` заполняет структуру `struct stat` информацией об определенном файле; если при этом необходимо обнаруживать символьные ссылки, следует использовать `lstat()`
- Условиями выхода из рекурсии служат следующие ситуации:
 - 1) произошла ошибка вызова `stat`,
 - 2) тип файла не является каталогом,
 - 3) вызываемый каталог не доступен,
 - 4) ошибка вызова функции `chdir()`,
 - 5) а также возврат 0 в конце функции.
- Функция `opendir(const char *name)` открывает директорию для чтения с именем `name` и возвращает указатель на поток директории. Функция `closedir(DIR *dirp)` закрывает директорию для чтения.
- `chdir(const char *path)` - функция смены рабочего каталога. Она устанавливает текущий каталог, указанный в аргументе `path`.
- Функция `lstat(const char *file_name, struct stat *buf)` возвращает информацию о файле, заданном с помощью `file_name` и заполняет буфер `buf`. Она может возвращать следующие ошибки:
 - 1) `EBADF` (неверный файловый дескриптор `filedes`);
 - 2) `EFAULT` (некорректный адрес);
 - 3) `ENOENT` (компонент полного имени файла `file_name` не существует или полное имя является пустой строкой);
 - 4) `ELOOP` (при поиске файла встретилось слишком много символьных ссылок);
 - 5) `ENOTDIR` (компонент пути не является каталогом);
 - 6) `ENOMEM` (недостаточно памяти в системе);
 - 7) `EACCES` (доступ запрещен);
 - 8) `ENAMETOOLONG` (слишком длинное имя файла).

Литература

1. Рязанова Н.Ю. - Курс лекций по "Операционным системам"[Текст], Москва 2020 год.
2. У. Ричард Стивенс, Стивен А. Раго - UNIX. Профессиональное программирование. 3-е изд. - Москва: Питер, 2018. - 944 с.