

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра теоретической и прикладной информатики

Лабораторная работа №1 по дисциплине «Управление ресурсами в вычислительных системах»

Факультет:	ПМИ
Группа:	ПМ-92
Вариант:	8
Студенты:	Иванов В., Кутузов И.
Преподаватели:	Стасышин В.М., Сивак М.А.

Новосибирск

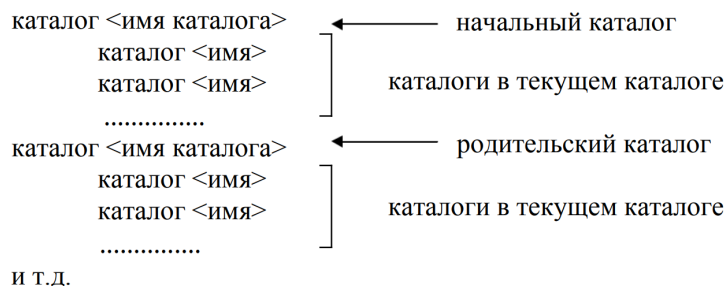
2022

1. Цель работы

Ознакомиться с устройством файловой системы ОС UNIX, механизмами ее функционирования, программными средствами для работы с ней (командный язык Shell, язык Си).

2. Задание

Разработать две программы на языках Си и Shell, реализующие просмотр каталога, имя которого указано параметром программы, и вывод имен встретившихся каталогов. После этого должен осуществляться переход в родительский каталог, который становится текущим; действия повторяются до тех пор, пока текущим каталогом не станет корневой каталог.



3. Описание используемых структур

DIR *opendir(char *dirname)

Открывает поток каталога и возвращает указатель на структуру типа DIR, которая содержит информацию о каталоге.

struct dirent *readdir(DIR *ptr)

Возвращает название следующего файла в каталоге. Иными словами, функция readdir() читает оглавление каталога по одному файлу за раз. Параметр ptr должен указывать на поток каталога, открытый с помощью opendir().

int closedir(DIR *ptr)

Закрывает поток каталога, на который указывает ptr. Возвращает 1 в случае успеха и 0 в противном случае. При неудаче переменная errno устанавливается равной EBADF (недействительный каталог).

char *getcwd(char *buf, size_t size)

Копирует абсолютный путь к текущему рабочему каталогу в массиве, на который указывает buf, имеющий длину size. Если текущий абсолютный путь требует буфера,

длина которого превышает size, то возвращается NULL, а errno принимает значение ERANGE.

int chdir(const char *path)

Изменяет текущий каталог на path. В случае успеха возвращается ноль, в противном случае возвращается -1.

4. Спецификация

Программа разработана и протестирована на компьютере с операционной системой Linux. В качестве компилятора используется GCC версии 11.2.0.

Директория, содержащая файл: **/home/lenferdetroud/Repos/university-tasks/Unix Resource Management/Unix Filesystem**

Название файла: **unix_filesystem[.c / .sh]**

Компиляция программы на языке Си: **gcc -o <результат компиляции> unix_filesystem.c**

Запуск программы на языке Си: **./<результат компиляции> <аргумент>**

Запуск сценария на языке Shell: **sh unix_filesystem.sh <аргумент>**

5. Описание алгоритма на языке Си

1. Передаем строку (название каталога) в качестве аргумента программы и проверяем, не является ли argv[1], содержащий переданную строку, пустым
2. Открываем каталог по переданной строке
3. Проверяем, существует ли каталог с указанным названием в текущем каталоге
4. Выводим этот каталог как начальный
5. Считываем и выводим на экран содержимое начального каталога
6. Открываем текущий каталог
7. Получаем и выводим путь текущего каталога
8. Получаем и выводим содержимое текущего каталога
9. Переходим в родительский каталог и закрываем текущий
10. Повторяем п.п. 7-10 до тех пор, пока путь каталога отличается от "/"

Для вывода содержимого каталога (п.п 6 и 9) используется цикл while, в котором поочередно выводятся значения d_name типа "директория" (d_type == DT_DIR), не являющиеся скрытыми каталогами, то есть не начинающиеся с точки (d_name[0] != '.').

6. Описание алгоритма на языке Shell

1. Передаем название каталога и проверяем, существует ли каталог, соответствующий переменной \$1
2. Проверяем, указан ли аргумент при вызове скрипта
3. Проверяем не является ли каталог файлом (флаг -f)

4. Переходим в каталог с переданным названием
5. Выводим название каталога, форматируя вывод команды pwd
6. Выводим содержимое каталога: при помощи команды ls получаем вертикальный список (-l) файлов (всех типов), в котором каталоги оканчиваются "/" (-F); передаем его командам grep (для выделения только каталогов по символу "/") и sed (для удаления "/"); полученные строки выводим в цикле while.
7. Переходим в родительский каталог
8. Повторяем п.п. 5-7 до тех пор, пока результат команды pwd не станет равным строке "/"
9. Переходим в корневой каталог и аналогично выводим его содержимое

7. Код программы на языке Си

```
#include <unistd.h>
#include <dirent.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    DIR *d; указатель на дескриптор каталога
    char cwd[512]; буфер, содержащий путь каталога
    struct dirent *dir;

    if (argv[1] == NULL) проверка на наличие аргумента
    {
        printf("Error: Directory not specified!\n");
        exit(0);
    }

    d = opendir(argv[1]); открытие начального каталога

    if (!d) проверка на существование каталога
    {
        printf("Error: No such directory!\n");
        exit(0);
    }

    printf("\n  Start directory (/s)\n", argv[1]); вывод начального каталога

    while ((dir = readdir(d)) != NULL) чтение и вывод содержимого начального каталога
        if (dir->d_type == DT_DIR && dir->d_name[0] != '.')
            printf("      %s\n", dir->d_name);

    do Вывод остальных каталогов
    {
        d = opendir("."); открытие текущего каталога

        if (getcwd(cwd, sizeof(cwd)) != NULL) запись и вывод пути каталога
            fprintf(stdout, "\n  %s\n", cwd);

        if (d) чтение и вывод содержимого текущего каталога
        {
            while ((dir = readdir(d)) != NULL)
```

```

        if (dir->d_type == DT_DIR && dir->d_name[0] != '.')
            printf("    %s\n", dir->d_name);

        chdir(".."); переход в родительский каталог
        closedir(d); закрытие каталога
    }

    } while ((int)cwd[1] != 0); пока не дойдет до корневого каталога

return 0;
}

```

8. Код программы на языке Shell

```

#!/bin/bash

if [ ! -e $1 ] проверка на существование каталога
then
    echo "Error: Directory does not exist."
    exit 1
fi

if [ ! $# -eq 1 ] проверка на наличие аргумента
then
    echo "Error: Directory name not specified or incorrect."
    exit 1
fi

if [ -f $1 ] проверка на то, является ли аргумент файлом
then
    echo "Error: Not a directory."
    exit 1
fi

cd $1 переход в начальную директорию

until [ "`pwd`" = "/" ] пока не дойдет до корневого каталога
do
    CURRENT=`pwd` текущий путь
    echo -e "\n $(basename "$CURRENT")" форматирование и вывод названия каталога
    ls -1F | grep '/$' | sed 's/\/$//' | форматирование содержимого каталога
    while read Name построчный вывод содержимого каталога
    do
        echo "          $Name"
    done
    cd .. переход в родительскую директорию
done

cd .. переход в корневой каталог
echo -e "\n /"
ls -1FS | grep '/$' | sed 's/\/$//' |
while read Name
do
    echo "          $Name"
done

```

9. Тестирование

```
~/Repos/university-tasks/Unix Resource Management/Unix Filesystem/test master ± ls ; cd .. ; ls
test1 test2 test3
a.out test unix_filesystem.c unix_filesystem.sh
~/Repos/university-tasks/Unix Resource Management/Unix Filesystem master ± gcc -o a.out unix_filesystem.c
~/Repos/university-tasks/Unix Resource Management/Unix Filesystem master ± ./a.out test
```

Start directory (/test)

```
test1
test3
test2
```

```
/home/lenferdetroud/Repos/university-tasks/Unix Resource Management/Unix Filesystem
test
```

```
/home/lenferdetroud/Repos/university-tasks/Unix Resource Management
Unix Filesystem
Unix Process Control
Process Synchronization
```

```
/home/lenferdetroud/Repos/university-tasks
Open Multi-Processing
Standard Template Library
Windows Resource Management
Mathematical Physics Equations
Numerical Methods
```

```
/home/lenferdetroud
Packages
Pictures
Downloads
Misc
Videos
Music
Repos
Documents
```

```
/home
lost+found
lenferdetroud
vladislav
```

```
/
lost+found
home
root
srv
var
sys
mnt
opt
proc
etc
run
usr
boot
dev
tmp
```

```
~/Repos/university-tasks/Unix Resource Management/Unix Filesystem ➤ master ± ➤ sh unix_filesystem.sh test
```

```
test
  test1
  test2
  test3

Unix Filesystem
  test

Unix Resource Management
  Process Synchronization
  Unix Filesystem
  Unix Process Control

university-tasks
  Assembly Language
  Computer Architecture
  Computer Networks
  Computer Translation
  Cryptography
  Exoskeleton Interface
  Logic Programming
  Mathematical Physics Equations
  Message Passing Interface
  Numerical Computations
  Numerical Methods

lenferdetroud
  Documents
  Downloads
  Misc
  Music
  Packages
  Pictures
  Repos
  Videos

home
  lenferdetroud
  lost+found
  vladislav

/
  lost+found
  boot
  etc
  home
  mnt
  opt
  root
  srv
  usr
  var
  dev
  run
  tmp
  proc
  sys
```

```
~/Repos/university-tasks/Unix Resource Management/Unix Filesystem ➤ master ± ➤ ./a.out
```

```
Error: Directory not specified!
```

```
~/Repos/university-tasks/Unix Resource Management/Unix Filesystem ➤ master ± ➤ ./a.out somedir
```

```
Error: No such directory!
```

```
~/Repos/university-tasks/Unix Resource Management/Unix Filesystem ➤ master ± ➤ sh unix_filesystem.sh
```

```
Error: Directory name not specified or incorrect.
```

```
✖ ~/Repos/university-tasks/Unix Resource Management/Unix Filesystem ➤ master ± ➤ sh unix_filesystem.sh somedir
```

```
Error: Directory does not exist.
```