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-12

Вариант: 10

Студенты: Курдюков И. Омельницкая Е.

Преподаватели: Стасышин В.М., Сивак М.А.

Новосибирск

2024

# Цель работы

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

# Задание

Разработать две программы на языках Си и Shell, программы выводят имена тех каталогов в каталоге, которые в себе не содержат каталогов. Имя каталога задано параметром программы.

# 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).

# Спецификация

Программа разработана и протестирована на компьютере с Cygwin. Cygwin — UNIX-подобная среда и интерфейс командной строки для Microsoft Windows. В качестве компилятора используется GCC версии 11.2.0.

Директория, содержащая файлC:\cygwin64\home\Ivan

Название файла: **pgoram.sh ; third\_ver.c ; third\_ver.exe**

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

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

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

# Описание алгоритма на языке Си

1. Передаем строку (название каталога) в качестве аргумента функции listDirectories
2. Проверяем открывается ли каталог, если нет – выводим ошибку.
3. Считываем запись в каталоге
4. Если запись соответствует типу директория то записываем ее путь
5. Проверяем эту директорию на наличие других директорий
6. Для каждой найденной директории проверяем их поддиректории на наличие директорий
7. Те поддиректории у которых не оказалось директорий выводим.
8. Повторяем с пункта 3.

Для вывода содержимого каталога (цикл с п.3 по п.8) 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] != '.').

# 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 является ли аргумент директорий (флаг -d)
4. Цикл: для каждой директории в $1
5. Если находится директория внутри выбранной (метод find) то продолжаем
6. Иначе выводим имя директории(внутри которой директорий нет)
7. Повторяем п.п. 4-6 до тех пор, пока не кончатся директории в $1

# Код программы на языке Си

#include <stdio.h>

#include <dirent.h>

#include <sys/stat.h>

#include <string.h>

void listDirectories(const char \*path)

{

    DIR \*dir;

    struct dirent \*entry;

    struct stat statbuf;

    dir = opendir(path);

//     Открытие и закрытие каталога выполняется системными вызовами opendir()

// и closedir(). При успешном открытии каталога системный вызов возвращает

// указатель на переменную типа DIR, являющуюся дескриптором каталога,

// определенную в файле <dirent.h> и используемую при чтении и записи в каталог.

// При неудачном вызове возвращается значение NULL.

    if (dir == NULL)

    {

        perror("Ошибка открытия каталога");

        return;

    }

    while ((entry = readdir(dir)) != NULL) //Функция readdir() возвращает указатель на следующую запись каталога в структуре dirent, прочитанную из потока каталога

    {

        if (entry->d\_type == DT\_DIR) // в поле d\_type хранится тип, DT\_DIR соответствует типу Директория

        {

            char fullpath[PATH\_MAX];

            snprintf(fullpath, PATH\_MAX, "%s/%s", path, entry->d\_name); //записываем в fullpath путь проверяемой директории. Например var/log

            // Проверяем, не содержит ли каталог других каталогов

            if (stat(fullpath, &statbuf) == 0 && S\_ISDIR(statbuf.st\_mode)) // The following macros will test whether a file is of the specified type. The value m supplied to the macros is the value of st\_mode from a stat structure. The macro evaluates to a non-zero value if the test is true, 0 if the test is false.

            { // иными словами S\_ISDIR - это макрос которому на вход подается содержимое флага из поля st\_mode, которое находится в statbuf. При успешном заполнении структоры функция stat возваращает 0

              //Функция stat() вносит в структуру, на которую указывает statbuf, информацию, содержащуюся в файле, связанном с указателем filename. Структура stat определена в sys\stat.h.

                DIR \*subdir = opendir(fullpath); // делаем все то же самое для поддиректории

                if (subdir != NULL)

                {

                    struct dirent \*subentry;

                    int contains\_subdirs = 0;

                    while ((subentry = readdir(subdir)) != NULL) // Просматриваем содержимое поддиректории, проверяем, есть ли другие директории. Закрываем поддиректорию.

                    {

                        if (subentry->d\_type == DT\_DIR && strcmp(subentry->d\_name, ".") != 0 && strcmp(subentry->d\_name, "..") != 0) // Игнорируем текущий и родительский каталоги

                        {

                            contains\_subdirs = 1;

                            break;

                        }

                    }

                    closedir(subdir);

                    // Если каталог не содержит других каталогов, выводим его имя

                    if (!contains\_subdirs)

                    {

                        printf("%s/%s\n", path, entry->d\_name);

                    }

                }

            }

        }

    }

    closedir(dir);

}

int main()

{

    const char \*directory\_path = "var";

    listDirectories(directory\_path);

    return 0;

}

# Код программы на языке Shell

#!/bin/bash

a="var"

# Проверяем, что передан аргумент

if [ -z "$a" ]

then

  echo "Usage: $0 directory\_name"

  exit 1

fi

# Проверяем, является ли аргументом директорией

if [ ! -d "$a" ]

then

  echo "$a is not a directory"

  exit 1

fi

# Ищем все поддиректории в указанной директории и проверяем, не содержат ли они своих поддиректорий

echo "Каталоги в $a которые не содержат других каталогов:"

for dir in "$a"/\*/

do

  if [[ -n $(find "$dir" -mindepth 1 -type d) ]]

  then

    continue

  else

    echo "${dir%/}"  # Удаляем завершающий слеш из имени директории перед выводом

  fi

done

1. **Тестирование**







