|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Лабораторная работа № 1 | | |
| по дисциплине «Управление ресурсами в вычислительных системах» | | |
| **Программные средства для управления системными ресурсами. Файловая система OC UNIX.** | | |
|  | | |
|  | Бригада 7 | Грушев андрей |
| Группа ПМ-05 | болдырев сергей |
| Вариант 7 | хабарова анастасия |
|  |  |
|  |  |
|  |  |
| Преподаватели | стасышин владимир михайлович |
|  | сивак мария алексеевна |
| Новосибирск, 2023 | | |

1. **Условие**

Программа выводит содержимое каталога, имя которого указано  
параметром программы. При выводе сначала перечисляются имена каталогов, а затем в алфавитном порядке имена файлов с указанием их длин, даты последнего  
изменения и числа ссылок на них.

1. **Используемые программные средства**
2. **Алгоритм решения**
3. **Спецификации**
4. **Перечень тестов**

Для программы на языке **С:**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Назначение | Результаты работы программы |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |

Для скрипта на языке **Shell:**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Назначение | Результаты работы программы |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |

1. **Make-файлы**

main: main.c

    gcc main.c -o main.o

clean:

    rm main.o

1. **Листинг программы**

Код программы на языке **C**:

#include <stdio.h>

#include <errno.h>

#include <string.h>

#include <unistd.h>

#include <stdlib.h>

#include <dirent.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <time.h>

/\*\*

 \* Структура, которая содержит нужную по заданию

 \* информацию о файле

\*\*/

struct file\_info

{

   char\* name; //Имя файла.

   off\_t st\_size; // Размер в байтах.

   time\_t time; // Время последней модификации.

   nlink\_t st\_nlink; // Количество жёстких ссылок.

};

/\*\*

 \* Функция сортирует массив строк в алфавитном порядке

 \* при помощи сортировки пузырьком.

 \* @param array Массив, который нужно отсортировать.

 \* @param size Размер массива.

 \*\*/

void sortInfos(struct file\_info array[], int size)

{

   struct file\_info tmp;

   int i, j;

   for (i = 1; i < size; i++)

      for (j = 0; j < size - i; j++)

         if (strcmp(array[j].name, array[j + 1].name) > 0)

         {

            tmp = array[j];

            array[j] = array[j + 1];

            array[j + 1] = tmp;

         }

}

int main(int argc, char\*\* argv)

{

   DIR\* d;

   struct dirent\* dirElement;

   struct stat buff;

   char path[50];

   struct file\_info regularFiles[50];

   int filesCount = 0;

   int i;

   if (argc != 2)

   {

      fprintf(stderr, "Error! Wrong number of arguments (expected 1, given %d).\n", --argc);

      exit(1);

   }

   d = opendir(argv[1]);

   if (!d)

   {

      fprintf(stderr, "Error! %s cannot be opened.\n", argv[1]);

      closedir(d);

      exit(1);

   }

   while ((dirElement = readdir(d)) != NULL) // Чтение файлов из заданной директории.

      if (dirElement->d\_name[0] != '.') //Пропуск скрытых файлов.

      {

         sprintf(path, "%s/%s", argv[1], dirElement->d\_name); //Формируем путь к файлу, для функции stat().

         stat(path, &buff); //Получаем информацию о файле.

         if (S\_ISDIR(buff.st\_mode)) //Проверка является ли файл директорией.

            printf("Directory - %s\n", dirElement->d\_name); //Печать имени файла.

         else if (S\_ISREG(buff.st\_mode)) //Проверка является ли файл директорией.

         {

            regularFiles[filesCount].name = dirElement->d\_name;

            regularFiles[filesCount].st\_size = buff.st\_size;

            regularFiles[filesCount].time = buff.st\_mtime;

            regularFiles[filesCount].st\_nlink = buff.st\_nlink;

            filesCount++;

         }

      }

   sortInfos(regularFiles, filesCount); //Сортируем массив структур информациии о файле по алфавиту.

   printf("\n");

   for (i = 0; i < filesCount; i++)

   {

      printf("%s\n", regularFiles[i].name);

      printf("Size - %ld bytes\nTime -  %sLinks - %ju\n\n",

            regularFiles[i].st\_size,

            ctime(&regularFiles[i].time),

            regularFiles[i].st\_nlink); //Печатаем нужную информацию о файле.

   }

   closedir(d);

   return 0;

}

Код скрипта на языке **Shell**:

# !/bin/bash

if [ $# -eq 1 ]

then

    if [ -d $1 ]

    then

        if [ ${1: -1} == "/" ]

        then

            path="$1\*"

        else

            path="$1/\*"

        fi

        for file in $path

        do

            if [ -d $file ]

            then

                printf 'Directory - %s\n' "$(basename $file)"

            fi

        done

        printf '\n'

        for file in $path

        do

            if [ -f $file ]

            then

                printf 'File - %s\n' "$(basename $file)"

                printf '%s\n\n' "$(stat --printf="Size - %s bytes\nTime - %y\nLinks - %h" $file)"

            fi

        done

    else

        printf '%s cannot be opened.' "$1"

    fi

else

    printf 'Wrong number of arguments (expected 1, given %d).' "$#"

fi