Министерство науки и образования РФ  
Новосибирский государственный технический университет  
Кафедра ТПИ

Лабораторная работа №2

по дисциплине «Управление ресурсами в вычислительных системах»

Факультет: ПМИ

Группа: ПМИ-31

Студенты: Суслов А.В.

Эльвейн Н.Л.  
 Олимпиади М.Б.

Преподаватель: Хайленко Е.А.  
 Филиппова Е.В.

Вариант: 1

1. **Цель работы**

Ознакомиться с файловой системой ОС UNIX, механизмами ее функционирования, основными элементами файловой системы: суперблок, описатели файлов, типы файлов, список свободных описателей файлов, список свободных блоков.

1. **Задание**

Вариант №1: Разработать программу, которая осуществляет просмотр текущего каталога и выводит на экран его содержимое группами в порядке возрастания числа ссылок на файлы (в том числе имена каталогов). Группа представляет собой объединение файлов с одинаковым числом ссылок на них.

1. **Описание метода решения задачи**

Открываем указанную папку (если папка не указана, то открываем текущий каталог). Просматриваем содержимое каталога, игнорируя стандартные названия путей (. и ..). Для каждого найденного файла смотрим кол-во жестких ссылок на него и записываем в структуру его и имя файла. Для папок не смотрим кол-во ссылок и записываем в структуру только имя. Сортируем файлы в порядке убывания кол-ва на них ссылок. По окончании работы программы выводим всё на экран.

1. **Описание разработанного программного средства**

Программа написана на языке C. Для получения исполняемого файла исходный текст программы следует скомпилировать каким-либо компилятором языка C. Исполняемый файл может работать под любой UNIX-совместимой ОС, поддерживающей соответствующий формат исполняемого файла.

Запуск программы осуществляется командой:

./main [folder]

где folder – каталог, с которого начинается работа программы. Если он не указан, то программа будет работать в текущем каталоге.

1. **Текст программы**

#include <stdio.h>

#include <string.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <dirent.h>

#include <unistd.h>

struct element {

char name[NAME\_MAX];

int linkCount;

short type; //0 - файл, 1 - папка

};

void sort(element \*dirs, int n) {

int i, j;

element tmp;

for(i = 0; i < n - 1; i++)

for(j = 0; j < n - i - 1; j++)

if(dirs[j].linkCount < dirs[j + 1].linkCount || dirs[j].type == 1) {

tmp = dirs[j];

dirs[j] = dirs[j+1];

dirs[j+1] = tmp;

}

}

int main (int argc, char \*argv[]) {

DIR \*home;

if (argc > 1)

home = opendir (argv[1]);

else

home = opendir(".");

struct dirent \*curDir;

element allFiles[100];

struct stat curStat;

int n = 0, i;

while ((curDir = readdir(home)) != NULL) {

lstat (curDir->d\_name, &curStat);

if (curStat.st\_mode & S\_IFDIR) {

if (strcmp( curDir->d\_name, ".") != 0 && strcmp(curDir->d\_name, "..") != 0) {

strcpy(allFiles[n].name, curDir->d\_name);

allFiles[n].linkCount = 1;

allFiles[n].type = 1;

n++;

}

} else {

strcpy (allFiles[n].name, curDir->d\_name);

allFiles[n].linkCount = curStat.st\_nlink;

allFiles[n].type = 0;

n++;

}

}

closedir(home);

sort(allFiles, n);

if (n == 0)

printf ("Files not found");

else {

printf ("Link: %d\n", allFiles[i].linkCount);

printf ("\t%s\n", allFiles[0].name);

for (i = 1; i < n && allFiles[i].type != 1; i++) {

if (allFiles[i].linkCount != allFiles[i-1].linkCount) {

printf ("Link: %d\n", allFiles[i].linkCount);

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

} else {

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

}

}

if(i < n)

printf ("\nFolders: \n");

while (i < n) {

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

i++;

}

}

return 0;

}

1. **Тесты**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Выходные данные |  |
| 1 | $ ls -l  total 36  -rw-r--r-- 1 ne\_mat ne\_mat 1675 февр. 16 00:28 2lab\_upr.c  -rw-r--r-- 2 ne\_mat ne\_mat 395 февр. 15 20:09 aza.c  drwxr-xr-x 2 ne\_mat ne\_mat 4096 февр. 16 00:34 labs  -rwxr-xr-x 1 ne\_mat ne\_mat 13184 февр. 16 00:29 main  drwxr-xr-x 2 ne\_mat ne\_mat 4096 сент. 21 10:26 Сертификат | Link: 2  aza.c  Link: 1  main  2lab\_upr.c  Folders:  Сертификат  labs | Проверка работоспособности программы |
| 2 | $ ls -l  total 24  -rw-r--r-- 2 ne\_mat ne\_mat 395 февр. 15 20:09 1.c  -rw-r--r-- 1 ne\_mat ne\_mat 1568 февр. 15 22:24 2lab\_upr.c  lrwxrwxrwx 1 ne\_mat ne\_mat 40 февр. 15 19:36 azd -> /home/ne\_mat/Документы/labs/das  -rwxr-xr-x 1 ne\_mat ne\_mat 13184 февр. 16 00:29 main  lrwxrwxrwx 1 ne\_mat ne\_mat 41 февр. 15 19:44 Ссылка на main -> /home/ne\_mat/Документы/labs/main | Link: 2  1.c  Link: 1  azd  Ссылка на main  main  2lab\_upr.c | Запуск программы при отсутствии папок. |
| 3 | $ ls -l  total 24  -rw-r--r-- 2 ne\_mat ne\_mat 395 февр. 15 20:09 1.c  -rw-r--r-- 1 ne\_mat ne\_mat 1568 февр. 15 22:24 2lab\_upr.c  lrwxrwxrwx 1 ne\_mat ne\_mat 40 февр. 15 19:36 azd -> /home/ne\_mat/Документы/labs/das  -rwxr-xr-x 1 ne\_mat ne\_mat 13184 февр. 16 00:29 main  lrwxrwxrwx 1 ne\_mat ne\_mat 41 февр. 15 19:44 Ссылка на main -> /home/ne\_mat/Документы/labs/main | Link: 3  1.c  Link: 1  azd  Ссылка на main  main  2lab\_upr.c | Запуск при добавленной новой ссылке |