|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образованияРоссийской Федерации | | |
| Федеральное государственное бюджетное образовательное  учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | path817.png |  |
| Кафедра теоретической и прикладной информатики | | |
|  | | |
| Лабораторная работа № 2 **Структура системы управления вводом-выводом в ОС UNIX** | | |
| по дисциплине «Управление ресурсами в вычислительных системах» | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| эмблема_светлая.png | Факультет: | ПМИ |
| Группа: | ПМ-72 |
| Вариант: | 8 |
| Студенты: | Антонов С.С. |
|  |  |
| Преподаватели: | Стасышин В.М.  Сивак М.А. |
|  |  |

|  |
| --- |
| Новосибирск |
| 2020 |

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. **Вариант**

Разработать программу, которая выводит на экран содержимое текущего каталога в алфавитном порядке. Каталоги не выводить.

1. **Описание метода решения задачи**

Программа написана на языке С, может быть скомпилирована компилятором языка С, с помощью make-файлов, либо без них.

Запуск программы осуществляется командой:

./hello [каталог]

По умолчанию вывод происходит на экран, но может быть легко перенаправлен в файл стандартными средствами интерпретаторов.

**Используемые средства:**

Strlen(char \*str) определение длины строки

DIR \*opendir(char\*path) открытие каталога

Strcmp(char \*str1, char \*str2) сравнение строк

Getcwd(char \*str, int size) нахождение пути к текущему каталогу

Struct dirent \*readdir(DIR \*dir) чтение записей каталога

Struct stat fs() структура со свойствами файла

Chdir(DIR \*dir) смена текущего каталога

Closedir (DIR \*dirptr) закрытие каталога

1. **Алгоритм**
2. Читаем текущий каталог
3. Проверяем является ли файл каталогом
4. Если да – пропускаем, если нет – записываем в структуру
5. По окончанию отбора сортируем имена файлов в структуре (массиве) по алфавиту
6. Выводим содержимое массива на экран
7. **Текст программы**

#include <stdio.h>

#include <errno.h>

#include <dirent.h>

#include <sys/stat.h>

#include <sys/types.h>

struct filelist

{

char \*name; // имя файла

};

void csort(struct filelist flist[256], int n) //сортировка файлов по алфавиту

{

int i, j, min;

struct filelist tmplist;

for (i=0;i<n-1;i++) // цикл по записям

{

tmplist=flist[i];

min=i;

for (j=i;j<n;j++)

{

if (strcmp(flist[j].name,tmplist.name)<0) //оператор сравнения строк

{

tmplist=flist[j];

min=j;

}

}

flist[min]=flist[i];

flist[i]=tmplist;

}

}

void print(struct filelist flist[256], int n, char \*type) //вывод имён файлов на экран

{

int i;

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

{

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

}

}

int main()

{

char cwd[1024];

DIR \*cd;

int n=0, m=0;

struct filelist flist[256];

struct dirent \* cf; // имя файла и номер индексного дескриптора

struct stat fs; // основные параметры файла

getcwd (cwd, sizeof(cwd));

errno = 0;

cd=opendir(cwd);

if (cd != NULL) //оператор открытия каталога opendir

{

while(cf=readdir(cd)) //оператор чтения каталого readdir

{

stat(cf->d\_name, &fs); //оператор получения информации о файле stat

if (fs.st\_mode&S\_IFREG) //записываем в массив файл {

flist[n].name=cf->d\_name;

n++;

}

}

closedir(cd); //закрытие каталога

}

else {

perror(“error: не удалось открыть”);// не удалось открыть каталог

return 0;

}

csort(flist,n); // сортировка

printf ("name \t\n");

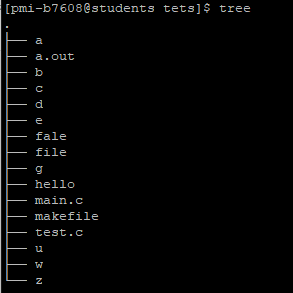
print(flist,n," "); // список файлов

return 0;

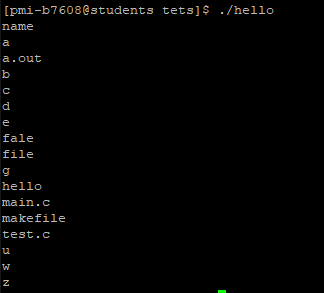
}

1. **Тесты**

Содержимое каталога:



Результат выполнения программы:



Тест с ошибкой

