

Лабораторная работа os_06. Вариант № 1 – 2007

Задача: создать программу *bnmtree* на языке C, имеющую следующую форму запуска:

```
bnmtree startdir string,
```

где

startdir – начальный каталог;

string – начало имени файла.

Программа должна просмотреть всю иерархию каталогов начиная с *startdir*, и вывести информацию о всех файлах, в имени которых первые N символов совпадают со строкой *string* (N – число символов в *string*). Считать, что пустая строка *string* соответствует любому имени файла.

Информация о файле должна включать имя файла с путем, тип, размер, атрибуты управления доступом к файлу. Для проверки типа файла использовать все макросы вида *S_ISDIR*, *S_IS**, определенные в заголовочном файле *sys/stat.h*. Информацию о режиме доступа выводить в виде *rwxrwxrwx*, причем наличие буквы свидетельствует о наличии соответствующего разрешения, а символ ‘-’ обозначает отсутствие разрешения. Заходить во все доступные каталоги, а каталоги, в которые нельзя войти или в которых нельзя получить информацию о содержимом, игнорировать. Не заходить в каталоги, указанные символическими ссылками.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

При выходе из программы ее текущий каталог должен быть таким же, как и в момент запуска.

Для операций с каталогами и другими файловыми объектами использовать функции *opendir*, *closedir*, *readdir*, *chdir*, *getcwd*, *stat*, *lstat* библиотеки языка C, обеспечивающие доступ к системным вызовам операционной системы.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке. Программа не должна завершаться аварийно.

Программа должна быть снабжена комментариями, а также содержать название лабораторной работы и сведения об авторе программы.

Лабораторная работа os_06. Вариант № 2 – 2007

Задача: создать программу *suftree* на языке C, имеющую следующую форму запуска:

```
suftree startdir suff,
```

где

startdir – начальный каталог;

suff – последний суффикс имени файла.

Программа должна просмотреть всю иерархию каталогов начиная с *startdir*, и вывести информацию о всех файлах, в имени которых последний из суффиксов совпадает с *suff*.

Информация о файле должна включать имя файла с путем, тип, размер, атрибуты управления доступом к файлу. Для проверки типа файла использовать все макросы вида *S_ISDIR*, *S_IS**, определенные в заголовочном файле *sys/stat.h*. Информацию о режиме доступа выводить в виде *rwXrwxrwx*, причем наличие буквы свидетельствует о наличии соответствующего разрешения, а символ ‘-’ обозначает отсутствие разрешения. Заходить во все доступные каталоги, а каталоги, в которые нельзя войти или в которых нельзя получить информацию о содержимом, игнорировать. Не заходить в каталоги, указанные символическими ссылками.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

При выходе из программы ее текущий каталог должен быть таким же, как и в момент запуска.

Для операций с каталогами и другими файловыми объектами использовать функции *opendir*, *closedir*, *readdir*, *chdir*, *getcwd*, *stat*, *lstat* библиотеки языка C, обеспечивающие доступ к системным вызовам операционной системы.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке. Программа не должна завершаться аварийно.

Программа должна быть снабжена комментариями, а также содержать название лабораторной работы и сведения об авторе программы.

Лабораторная работа os_06. Вариант № 3 – 2007

Задача: создать программу *uidtree* на языке C, имеющую следующую форму запуска:

```
uidtree startdir UID,
```

где

startdir – начальный каталог;

UID – числовой идентификатор владельца файла.

Программа должна просмотреть всю иерархию каталогов начиная с *startdir*, и вывести информацию о всех файлах, владельцем которого является пользователь с идентификатором UID.

Информация о файле должна включать имя файла с путем, тип, размер, атрибуты управления доступом к файлу. Для проверки типа файла использовать все макросы вида *S_ISDIR*, *S_IS**, определенные в заголовочном файле *sys/stat.h*. Информацию о режиме доступа выводить в виде *rwXrwxrwx*, причем наличие буквы свидетельствует о наличии соответствующего разрешения, а символ ‘-’ обозначает отсутствие разрешения. Заходить во все доступные каталоги, а каталоги, в которые нельзя войти или в которых нельзя получить информацию о содержимом, игнорировать. Не заходить в каталоги, указанные символическими ссылками.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

При выходе из программы ее текущий каталог должен быть таким же, как и в момент запуска.

Для операций с каталогами и другими файловыми объектами использовать функции *opendir*, *closedir*, *readdir*, *chdir*, *getcwd*, *stat*, *lstat* библиотеки языка C, обеспечивающие доступ к системным вызовам операционной системы.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке. Программа не должна завершаться аварийно.

Программа должна быть снабжена комментариями, а также содержать название лабораторной работы и сведения об авторе программы.

Лабораторная работа os_06. Вариант № 4 – 2007

Задача: создать программу *gidtree* на языке C, имеющую следующую форму запуска:

```
gidtree startdir GID,
```

где

startdir – начальный каталог;

GID – числовой идентификатор группы файла.

Программа должна просмотреть всю иерархию каталогов начиная с *startdir*, и вывести информацию о всех файлах, идентификатор группы которых равен *GID*.

Информация о файле должна включать имя файла с путем, тип, размер, атрибуты управления доступом к файлу. Для проверки типа файла использовать все макросы вида *S_ISDIR*, *S_IS**, определенные в заголовочном файле *sys/stat.h*. Информацию о режиме доступа выводить в виде *rwXrwxrwx*, причем наличие буквы свидетельствует о наличии соответствующего разрешения, а символ ‘-’ обозначает отсутствие разрешения. Заходить во все доступные каталоги, а каталоги, в которые нельзя войти или в которых нельзя получить информацию о содержимом, игнорировать. Не заходить в каталоги, указанные символическими ссылками.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

При выходе из программы ее текущий каталог должен быть таким же, как и в момент запуска.

Для операций с каталогами и другими файловыми объектами использовать функции *opendir*, *closedir*, *readdir*, *chdir*, *getcwd*, *stat*, *lstat* библиотеки языка C, обеспечивающие доступ к системным вызовам операционной системы.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке. Программа не должна завершаться аварийно.

Программа должна быть снабжена комментариями, а также содержать название лабораторной работы и сведения об авторе программы.

Лабораторная работа os_06. Вариант № 5 – 2007

Задача: создать программу *sizetree* на языке C, имеющую следующую форму запуска:

```
sizetree startdir size,
```

где

startdir – начальный каталог;

size – минимальный размер файла.

Программа должна просмотреть всю иерархию каталогов начиная с *startdir*, и вывести информацию о всех файлах, размер которых не превышает *size*.

Информация о файле должна включать имя файла с путем, тип, размер, атрибуты управления доступом к файлу. Для проверки типа файла использовать все макросы вида *S_ISDIR*, *S_IS**, определенные в заголовочном файле *sys/stat.h*. Информацию о режиме доступа выводить в виде *rwXrwxrwx*, причем наличие буквы свидетельствует о наличии соответствующего разрешения, а символ ‘-’ обозначает отсутствие разрешения. Заходить во все доступные каталоги, а каталоги, в которые нельзя войти или в которых нельзя получить информацию о содержимом, игнорировать. Не заходить в каталоги, указанные символическими ссылками.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

При выходе из программы ее текущий каталог должен быть таким же, как и в момент запуска.

Для операций с каталогами и другими файловыми объектами использовать функции *opendir*, *closedir*, *readdir*, *chdir*, *getcwd*, *stat*, *lstat* библиотеки языка C, обеспечивающие доступ к системным вызовам операционной системы.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке. Программа не должна завершаться аварийно.

Программа должна быть снабжена комментариями, а также содержать название лабораторной работы и сведения об авторе программы.

Лабораторная работа os_06. Вариант № 6 – 2007

Задача: создать программу *acctree* на языке C, имеющую следующую форму запуска:

```
acctree startdir hour,
```

где

startdir – начальный каталог;

hour – интервал времени в часах.

Программа должна просмотреть всю иерархию каталогов начиная с *startdir*, и вывести информацию о всех файлах, к которым не было доступа за последние *hour* часов.

Информация о файле должна включать имя файла с путем, тип, размер, атрибуты управления доступом к файлу. Для проверки типа файла использовать все макросы вида *S_ISDIR*, *S_IS**, определенные в заголовочном файле *sys/stat.h*. Информацию о режиме доступа выводить в виде *rwxxrwxrwx*, причем наличие буквы свидетельствует о наличии соответствующего разрешения, а символ ‘-’ обозначает отсутствие разрешения. Заходить во все доступные каталоги, а каталоги, в которые нельзя войти или в которых нельзя получить информацию о содержимом, игнорировать. Не заходить в каталоги, указанные символическими ссылками.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

При выходе из программы ее текущий каталог должен быть таким же, как и в момент запуска.

Для операций с каталогами и другими файловыми объектами использовать функции *opendir*, *closedir*, *readdir*, *chdir*, *getcwd*, *stat*, *lstat* библиотеки языка C, обеспечивающие доступ к системным вызовам операционной системы.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке. Программа не должна завершаться аварийно.

Программа должна быть снабжена комментариями, а также содержать название лабораторной работы и сведения об авторе программы.

Лабораторная работа os_06. Вариант № 7 – 2007

Задача: создать программу *modtree* на языке C, имеющую следующую форму запуска:

```
modtree startdir hour,
```

где

startdir – начальный каталог;

hour – интервал времени в часах.

Программа должна просмотреть всю иерархию каталогов начиная с *startdir*, и вывести информацию о всех файлах, которые не изменялись за последние *hour* часов.

Информация о файле должна включать имя файла с путем, тип, размер, атрибуты управления доступом к файлу. Для проверки типа файла использовать все макросы вида *S_ISDIR*, *S_IS**, определенные в заголовочном файле *sys/stat.h*. Информацию о режиме доступа выводить в виде *rwxxrwxrwx*, причем наличие буквы свидетельствует о наличии соответствующего разрешения, а символ ‘-’ обозначает отсутствие разрешения. Заходить во все доступные каталоги, а каталоги, в которые нельзя войти или в которых нельзя получить информацию о содержимом, игнорировать. Не заходить в каталоги, указанные символическими ссылками.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

При выходе из программы ее текущий каталог должен быть таким же, как и в момент запуска.

Для операций с каталогами и другими файловыми объектами использовать функции *opendir*, *closedir*, *readdir*, *chdir*, *getcwd*, *stat*, *lstat* библиотеки языка C, обеспечивающие доступ к системным вызовам операционной системы.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке. Программа не должна завершаться аварийно.

Программа должна быть снабжена комментариями, а также содержать название лабораторной работы и сведения об авторе программы.

Лабораторная работа os_06. Вариант № 8 – 2007

Задача: создать программу *linktree* на языке C, имеющую следующую форму запуска:

```
linktree startdir links,
```

где

startdir – начальный каталог;

links – минимальное количество ссылок на файл.

Программа должна просмотреть всю иерархию каталогов начиная с *startdir*, и вывести информацию о всех файлах, на которые существует не менее *links* жёстких ссылок.

Информация о файле должна включать имя файла с путем, тип, размер, атрибуты управления доступом к файлу. Для проверки типа файла использовать все макросы вида *S_ISDIR*, *S_IS**, определенные в заголовочном файле *sys/stat.h*. Информацию о режиме доступа выводить в виде *rwxxrwxrwx*, причем наличие буквы свидетельствует о наличии соответствующего разрешения, а символ ‘-’ обозначает отсутствие разрешения. Заходить во все доступные каталоги, а каталоги, в которые нельзя войти или в которых нельзя получить информацию о содержимом, игнорировать. Не заходить в каталоги, указанные символическими ссылками.

Если первый аргумент задан как “-h”, или указано недостаточное количество аргументов, вывести краткую справку о программе и правильных аргументах командной строки.

При выходе из программы ее текущий каталог должен быть таким же, как и в момент запуска.

Для операций с каталогами и другими файловыми объектами использовать функции *opendir*, *closedir*, *readdir*, *chdir*, *getcwd*, *stat*, *lstat* библиотеки языка C, обеспечивающие доступ к системным вызовам операционной системы.

В случае возникновения ошибки завершать свою работу с ненулевым кодом возврата и системным сообщением об ошибке. Программа не должна завершаться аварийно.

Программа должна быть снабжена комментариями, а также содержать название лабораторной работы и сведения об авторе программы.