

**Министерство образования и науки Российской Федерации**

**Федеральное агентство по образованию**

**Государственное образовательное учреждение высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Робототехника и комплексная автоматизация» (РК)**

**Кафедра «Системы автоматизированного проектирования» (РК6)**

****

**Отчет по лабораторной работе №2 по курсу**

**«Операционные системы»**

**Студент: Сергеева Диана**

**Группа:** РК6-56Б

**Преподаватель:** Грошев С.В.

Проверил:

Дата:

2021 год

**13 вариант**

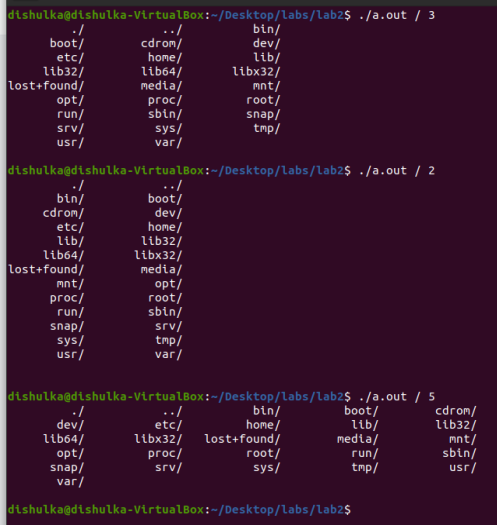
**Задание «Обработка каталогов файловых систем»:**

Требуется разработать программу для визуального просмотра оглавлений каталогов файловых систем OS UNIX, которая позволяет определить состав любого доступного каталога. Имя каталога должно передаваться программе через аргумент командной строки её вызова. В результате обработки оглавления заданного каталога программа должна обеспечивать стандартный вывод списков имён подкаталогов и файлов, которые содержаться в нём. Имена файлов и подкаталогов обоих списков должны перечисляться в словарном порядке в соответствие в соответствие с кодами ASCII, образующих из символов. Стандартный вывод оглавления заданного каталога должен завершаться информационным сообщением, которое идентифицирует число подкаталогов и файлов в нём. Это информационное сообщение должно отображаться через протокол стандартный диагностики. При разработке программы необходимо использовать библиотечные функции просмотра каталогов opendir, readdir, closedir и getewd, а также системные вызовы chdir, sbrk, statm write и exit.

Реализовать вывод оглавления заданного каталога в указанное число колонок, в которых имена файлов и подкаталогов выровнены по правой границе.

**Исходный код:**

[#include](https://vk.com/im?sel=63386677&st=%23include) <sys/types.h>  
[#include](https://vk.com/im?sel=63386677&st=%23include) <dirent.h>  
[#include](https://vk.com/im?sel=63386677&st=%23include) <unistd.h>  
[#include](https://vk.com/im?sel=63386677&st=%23include) <errno.h>  
[#include](https://vk.com/im?sel=63386677&st=%23include) <stdio.h>  
[#include](https://vk.com/im?sel=63386677&st=%23include) <stdlib.h>  
[#include](https://vk.com/im?sel=63386677&st=%23include) <string.h>  
[#include](https://vk.com/im?sel=63386677&st=%23include) <sys/stat.h>  
int scanfolder(int, char\*);  
int diag(char\*);  
int insersort(int);  
/\*  
typedef int (\*FCMP)(const void\*, const void\*);  
int compare(const void\*, const void\*);  
int compare(const void\* s1, const void\* s2) {  
return(strcmp((const char\*) s1, (const char\*) s2));  
}  
\*/  
static char\* list[512];  
static int col;  
int insertsort(int num) {  
int i=0;  
int j;  
char\* p;  
while(++i < num) {  
j = i - 1;  
p = list[i];  
while(strcmp(p, list[j]) < 0) {  
list[j+1] = list[j];  
j--;  
}  
list[j+1] = p;  
}  
return(0);  
}  
int diag(char\* mes) {  
write(1, mes, strlen(mes));  
write(1, ": ", 2);  
write(1, sys\_errlist[errno], strlen(sys\_errlist[errno]));  
write(1, "\n", 1);  
exit(errno);  
}  
int scanfolder(int type, char\* suf) {  
DIR\* fdir;  
struct dirent\* folder;  
struct stat sbuf[1];  
char\* start;  
char\* s;  
int count=0;  
int len;  
int sum=0;  
if((fdir = opendir(".")) == NULL)  
return(errno);  
list[count] = "\n";  
count++;  
start = sbrk(0);  
while((folder = readdir(fdir)) != NULL) {  
/\* sbuf->st\_mode = 0; \*/  
stat(folder->d\_name, sbuf);  
if(((sbuf->st\_mode) & S\_IFMT) != type)  
continue;  
sum=sum+sbuf->st\_size;  
len = strlen(folder->d\_name);  
s = sbrk(len+1);  
memcpy(s, folder->d\_name, len);  
list[count] = s;  
count++;  
list[count] = NULL;  
}  
closedir(fdir);  
/\*qsort(start, i, 256, (FCMP) compare); \*/  
insertsort(count);  
count=1;  
len = strlen(list[count]);  
int end\_count = 1;  
while(list[count] != NULL) {  
if ((count) < len){  
count++;  
continue;  
}  
if (end\_count == col){  
printf("%10s%s\n", list[count], suf);  
end\_count = 1;  
}  
else{  
printf("%10s%s ", list[count], suf);  
end\_count++;  
}  
count++;  
}  
putchar('\n');  
brk(start);  
return(0);  
}  
int main(int argc, char\* argv[]) {  
col = atoi(argv[2]);  
if(chdir(argv[1]) < 0)  
return(diag(argv[1]));  
if(scanfolder(S\_IFDIR, "/") > 0)  
return(diag(argv[1]));  
scanfolder(S\_IFREG, "");  
return(0);  
}

**Результат работы программы:**