Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

Студент	Гусаров Аркадий Андреевич
Группа	РК6-53Б
Тип задания	Лабораторная работа №2
Тема лабораторной работы	Обработка каталогов файловых систем

Студент _____ **_Гусаров А.А._**
подпись, дата *фамилия, и.о.*

Оценка _____

Москва, 2021 г.

Задание на лабораторную работу

Разработать программу для визуального просмотра оглавлений каталогов файловых систем ОХ UNIX. Отобразить список файлов заданного каталога, размер которых превышает заданную величину в байтах.

Теоретическая часть. Описание алгоритма

При запуске программа проверяет входные данные на ошибки, после чего последовательно “проходит” по указанному пути, параллельно считая кол-во вложенных файлов и папок в указанной директории. При этом происходит проверка размера файла, чтобы соответствовать заданному условию.

Код программы

```
#include <sys/types.h>
#include <dirent.h>
#include <unistd.h>
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
// Для int64_t
#include <stdint.h>
// Для правильного вывода int64_t в printf
#include <inttypes.h>

int scanfolder(int, char *);
int insersort(int);

static char *list[512];
int mainCount = -4;
// Путь к указанной директории
char path[64];
// Переданное макс количество байт
int bytes;

// Функция получения размера файла
int64_t getFileSize(const char *file_name)
{
    int64_t _file_size = 0;
    FILE *fd = fopen(file_name, "rb");
```

```
    if (fd == NULL)
        _file_size = -1;
    else
    {
        while (getc(fd) != EOF)
            _file_size++;
        fclose(fd);
    }

    return _file_size;
}

// Функция конкатинирует две строки
char *concat(const char *s1, const char *s2)
{
    // +1 для null - окончание строки
    char *result = malloc(strlen(s1) + strlen(s2) + 1);
    strcpy(result, s1);
    strcat(result, s2);

    return result;
}

// Сортировка файлов по именам
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 scanfolder(int type, char *suf)
{
    DIR *fdir;
    struct dirent *folder;
    struct stat sbuf[1];
    char *start;
    char *s;
```

```
    int count = 0;
    int len;

    if ((fdir = opendir(".")) == NULL)
        return (errno);

    list[count] = "\n";
    count++;
    // Функция sbrk() увеличивает / уменьшает память на величину amount
байт
    start = sbrk(0);

    // Итерируемся по папкам
    while ((folder = readdir(fdir)) != NULL)
    {
        // Получаем путь к каждому элементу в каталоге
        // folder->d_name - строка - имя файла / каталога
        char *currPath = concat(path, folder->d_name);
        // ! printf("YYYYY: %s\n", currPath);
        // Получаем размер каждого файла
        int64_t file_size = getFileSize(currPath);
        // ! printf("File size: %" PRId64 "\n", file_size);

        // Проверка, что файл больше заданного объема в байтах
        if (file_size >= bytes)
            continue;

        sbuf->st_mode = 0;
        stat(folder->d_name, sbuf);

        if (((sbuf->st_mode) & S_IFMT) != type)
            continue;

        len = strlen(folder->d_name);
        s = sbrk(len + 1);
        memcpy(s, folder->d_name, len);
        list[count] = s;
        count++;
        list[count] = NULL;
    }

    closedir(fdir);
    insertsort(count);

    count = 1;

    // Итерируемся по файлам
    while (list[count] != NULL)
    {
        len = strlen(list[count]);
```

```
        write(1, list[count], len);
        write(1, suf, 2);
        count++;
    }

    brk(start);

    mainCount += count;

    return (0);
}

int main(int argc, char *argv[])
{
    if (argc != 3)
    {
        printf("Wrong input\n");
        exit(10);
    }

    bytes = atoi(argv[2]);
    // В path записываем указанный путь
    strcpy(path, argv[1]);

    // chdir устанавливает в качестве текущего каталога - argv[1]. При
ошибке => -1
    if (chdir(argv[1]) < 0)
    {
        printf("Wrong path\n");
        exit(101);
    }
    // Проверка каталог - каталог
    if (scanfolder(S_IFDIR, "/\n") > 0)
    {
        printf("No folder\n");
        exit(102);
    }
    // Проверка папки на файлы. S_IFREG - обычный файл
    scanfolder(S_IFREG, " \n");

    printf("\nЧисло подкаталогов и файлов: %d\n", mainCount);
    return (0);
}
```

Результат работы программы

Вызов:

./a.out /home/arcady/Downloads/02/ 18000

Результат:

./

../

a.out

main.c

Число подкаталогов и файлов: 2