

**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 №1 по курсу**

**«Операционные системы»**

**Студент:** Петраков Станислав

**Группа:** РК6-56Б

**Преподаватель:** Грошев С.

Проверил:

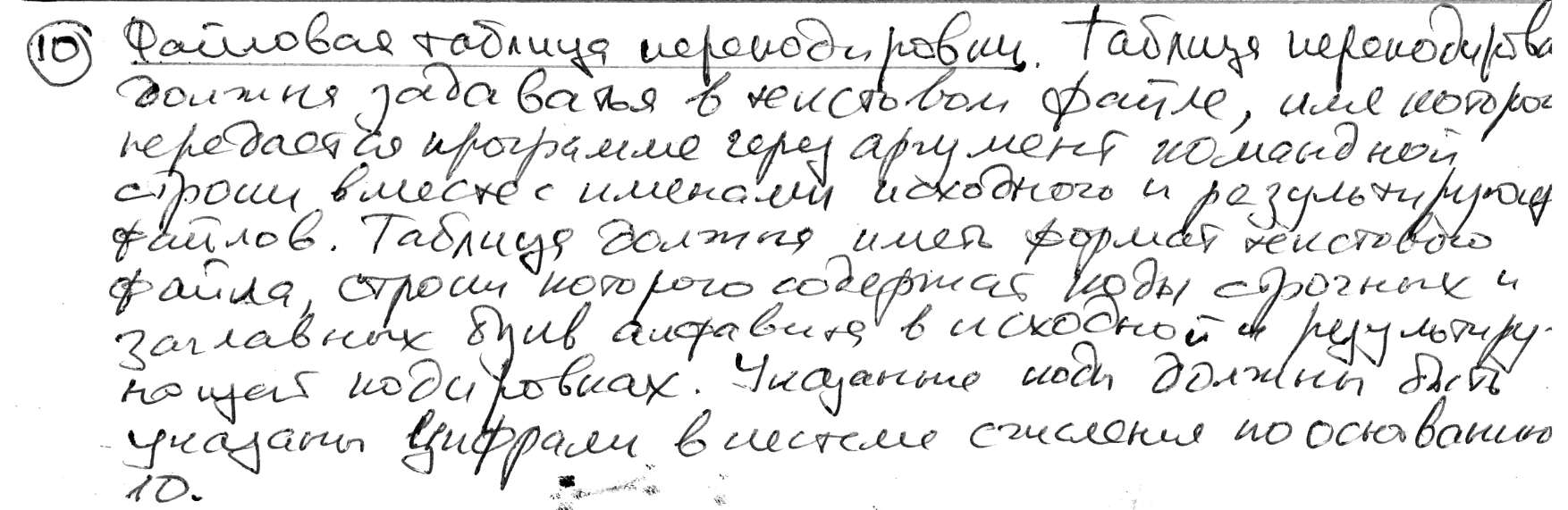
Дата:

2021 год

**10 вариант**

**Задание:**

Таблица перекодировки в текстовом файле и 10-ой системе счисления.

****

**Исходный код:**

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <stdio.h>

//

#include <stdlib.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <sys/types.h>

//

int diag(char \*, const char \*, int);

unsigned char \*recode(int);

static unsigned char buf[BUFSIZ]; //буфер-чтения - записи

static char tab[32];

void generate(int);

int oct\_to\_dec(int);

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

{

int fds; //файловый дискриптор исходного КОИ-8 текста

int fdt;

int fdtable;

int num;

if (argc < 2)

{

exit(diag(argv[0], "Source & target file names?", 127));

}

if (argc < 3)

{

exit(diag(argv[0], "Target file names?", 63));

}

if (argc < 4)

{

exit(diag(argv[0], "Code table?", 65));

}

if ((fds = open(argv[1], 0)) < 0)

{

exit(diag(argv[1], sys\_errlist[errno], errno));

}

if ((fdtable = open(argv[3], 0)) < 0)

{

exit(diag(argv[1], sys\_errlist[errno], errno));

}

if (access(argv[2], 0) == 0)

exit(diag(argv[2], "Target file already exist", 255));

errno = 0;

if ((fdt = creat(argv[2], 0644)) < 0)

exit(diag(argv[2], sys\_errlist[errno], errno));

//------------------------------------

// read code table

while ((num = read(fdtable, buf, BUFSIZ)) > 0)

generate(num);

//------------------------------------

while ((num = read(fds, buf, BUFSIZ)) > 0)

write(fdt, recode(num), num);

close(fdt);

close(fds);

exit(0);

}

int diag(char \*name, const char \*mes, int code)

{

int len;

strcpy(buf, name);

strncat(buf, ": ", 2);

strcat(buf, mes);

len = strlen(buf);

buf[len] = '\n';

write(2, buf, len + 1);

return (code);

}

unsigned char \*recode(int n)

{

unsigned char c;

int i = 0;

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

{

c = buf[i];

if (c < 192)

{

continue;

}

if (c < 224)

{

buf[i] = tab[c - 192] + 224;

}

else

{

buf[i] = tab[c - 224] + 192;

}

}

return (buf);

}

void generate(int n)

{

int cur;

int i = 0;

int temp = 0;

int tablePlace = 0;

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

{

if (buf[i] == ' ')

{

tab[tablePlace] = temp;

temp = 0;

tablePlace++;

continue;

}

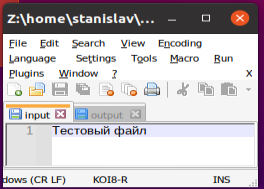
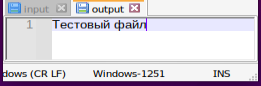
cur = buf[i] - '0';

temp=temp\*10+cur;

}

}

**Результат работы программы:**

****