Пермский филиал федерального государственного автономного образовательного учреждения высшего образования   
«Национальный исследовательский университет   
«Высшая школа экономики»

*Факультет информационных технологий в бизнесе*

Можегова Анна Сергеевна

**Лабораторная работа №4 «testfs» по дисциплине**

**«Операционные системы»**

студента образовательной программы «Программная инженерия»

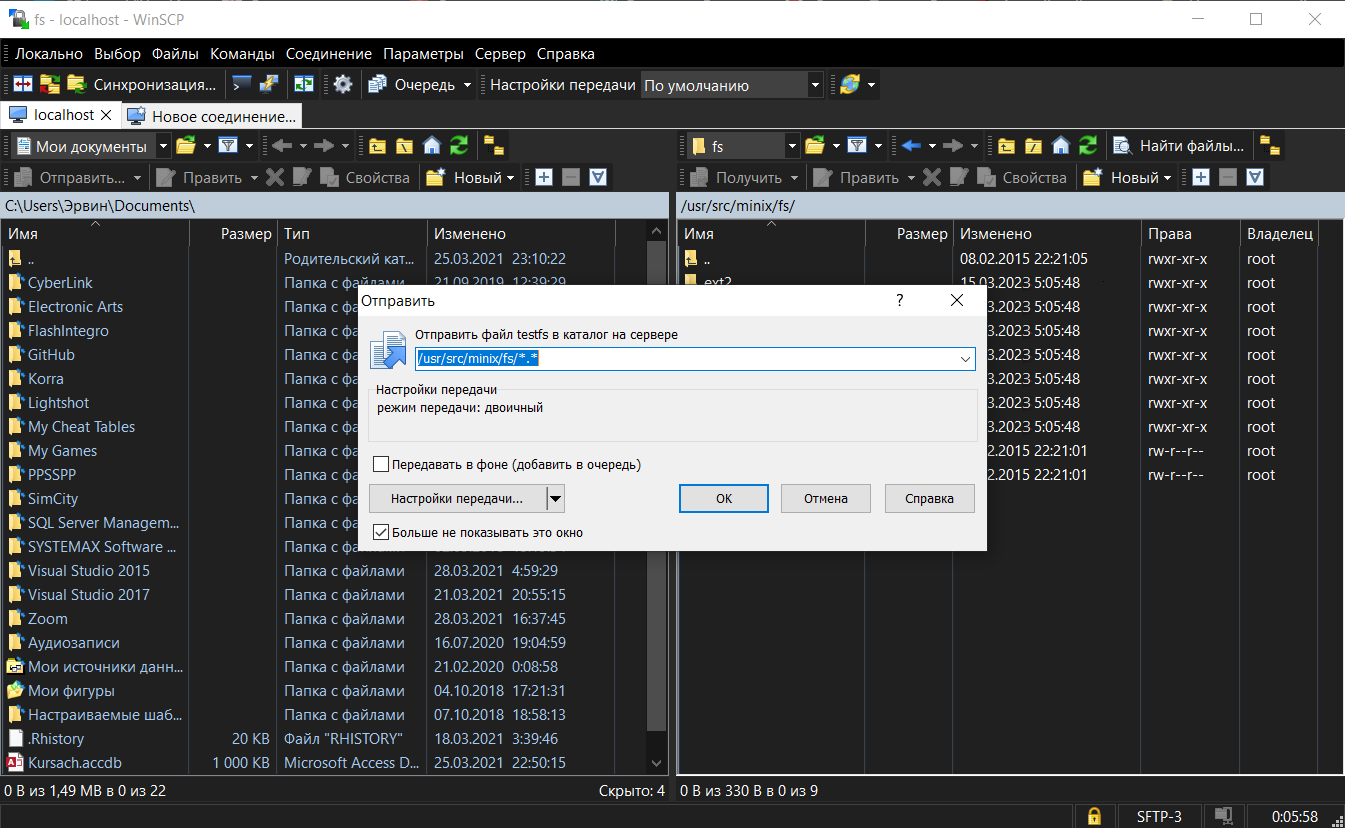
по направлению подготовки *09.03.04 Программная инженерия*

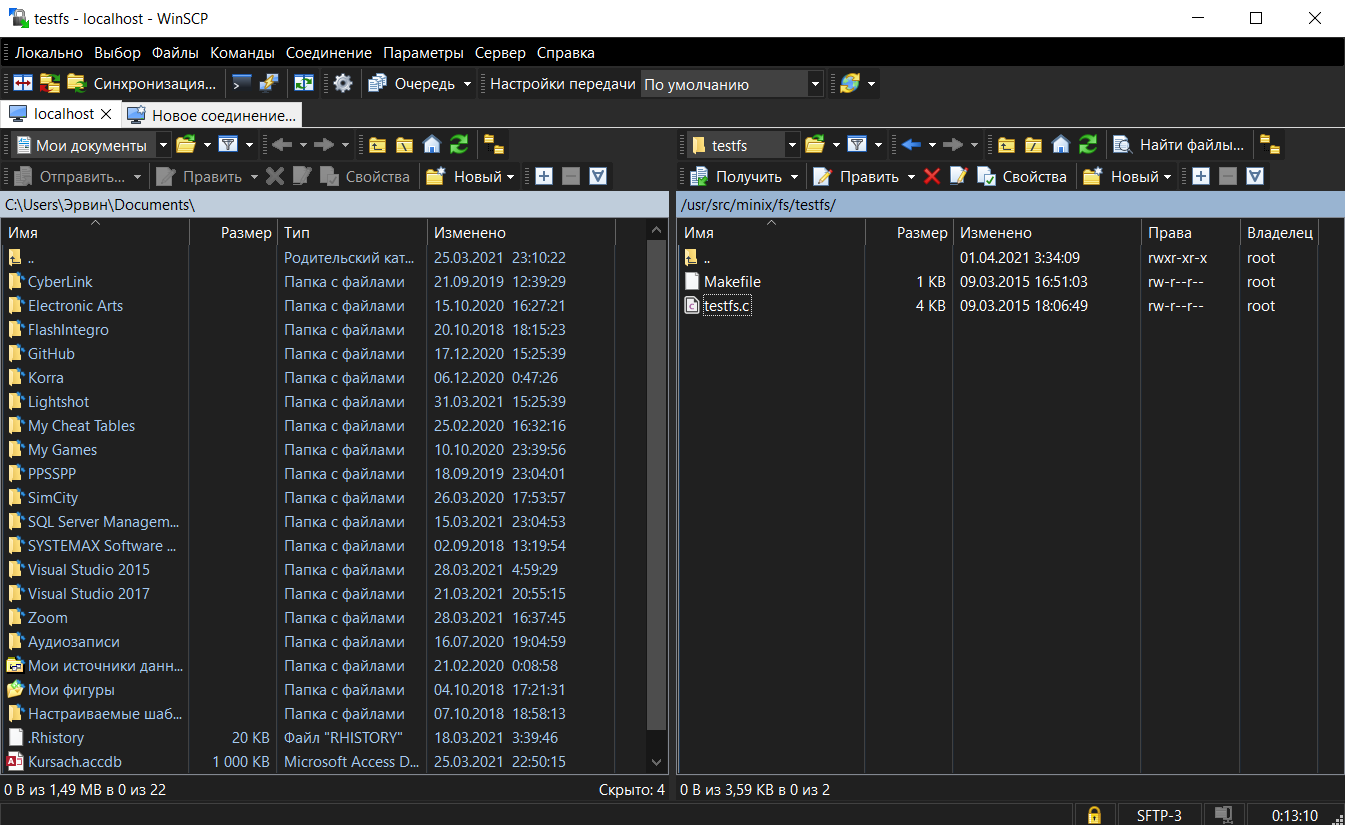
**Руководитель:**

Кузнецов Денис  
 Борисович

Пермь, 2023 год

**Скриншоты, иллюстрирующие порядок выполнения лабораторной**



****

/\* The table with callback hooks.

struct fs\_hooks my\_hooks ={

my\_init\_hook,

NULL, /\* cleanup\_hook \*

NULL, /\* lookup\_hook \*

NULL, /\* getdents\_hook \*

my\_read\_hook,

my\_write\_hook,/\* write\_hook \*/

NULL, /\* redlink\_hook \*/

NULL, /\* redlink\_hook \*

NULl, /\* redlink\_hook \*

NULL, /\* redlink\_hook \*

NULL, /\* redlink\_hook \*/

NULL, /\* redlink\_hook \*/

NULL /\* message\_hook \*/

};

int main(void)

{

struct inode\_state root\_stat;

/\* /\* Fill in the details to be used for the root node. It will be

a

\* directory, readable and searchable by anyone, and owned by

root.

root\_stat.mode = S\_IF DIR | 0555;

root\_stat.uid = 0;

root\_stat.gid = 0;

root\_stat.size = 0;;

root\_stat.dev = NO\_DEV;

/\* /\* Now start Trees. Preallocate 10 inodes, which is more than

we'l1

\* \* need for this example No indexed entries are used

static char data[51];

static void my\_init\_hook(void)

{

struct inode\_state file\_stat;

struct inode \*inode;

file\_stat.mode = S\_IF REG | 0444;

file\_stat.uid = 0;

file\_stat.gid = 0;

file stat.size = 3;

file\_stat.dev = NO\_DEV;

inode = add\_node(get\_root\_inode(), "test", No\_INDEX,

&file\_stat, o,

(cbdata\_t) 1);

assert(inode != NULL);

}

static int my\_read\_hook(struct inode \*inode, char \*ptr, size\_t len, off\_t offset, cbdata\_cbdata)

{

assert((int) cbdata == 1);

if (offset > 50)

{

len = 0;

return 0;

}

for (int i = 0; i < (int)len; i++)

{

ptr[i] = data[i];

}

return (int) len;

}

static int my\_write\_hook(struct inode \*inode, char \*ptr, size\_t max, off\_t offset, cbdata\_cbdata)

{

for (int i = 0; i < (int)max; i++)

{

data[i] = ptr[i];

}

}

return (int) max;

/\* The table with callback hooks. \*

struct fs\_hooks my\_hooks = {

my\_init\_hook,

NULL, /\* cleanup\_hook \*/

NULL, /\* lookup\_hook \*/

NULL, /\* getdents\_hook \*/

my\_read\_hook,

my\_write\_hook, /\* write\_hook \*

NULL, /\* redlink\_hook \*

NULL, /\* redlink\_hook \*/

NULL, /\* redlink hook \*

NULL, /\* redlink\_hook \*/

NULL, /\* redlink\_hook \*/

NULL, /\* redlink hook \*/

NULL /\* message\_hook \*/

};

int main(void)

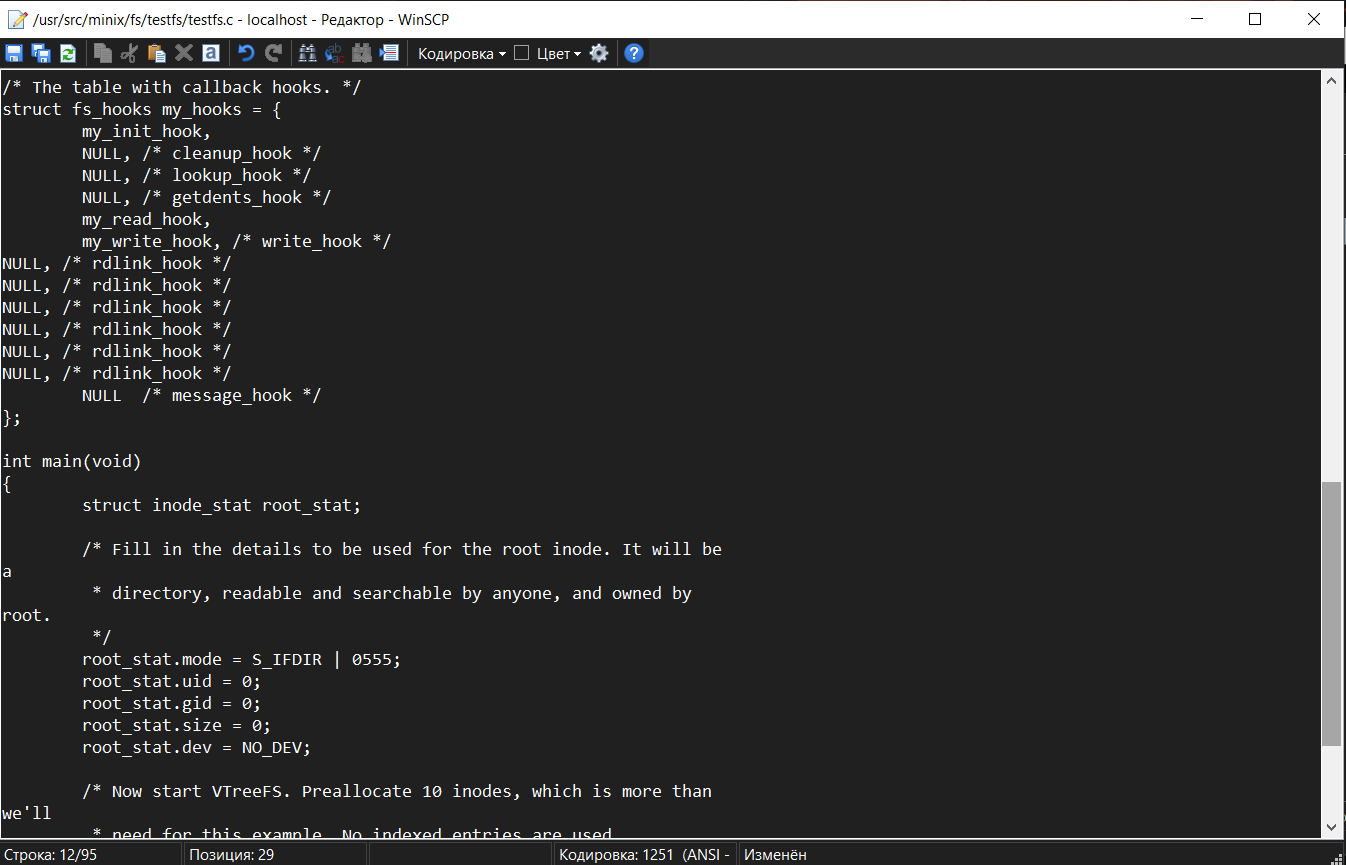
{

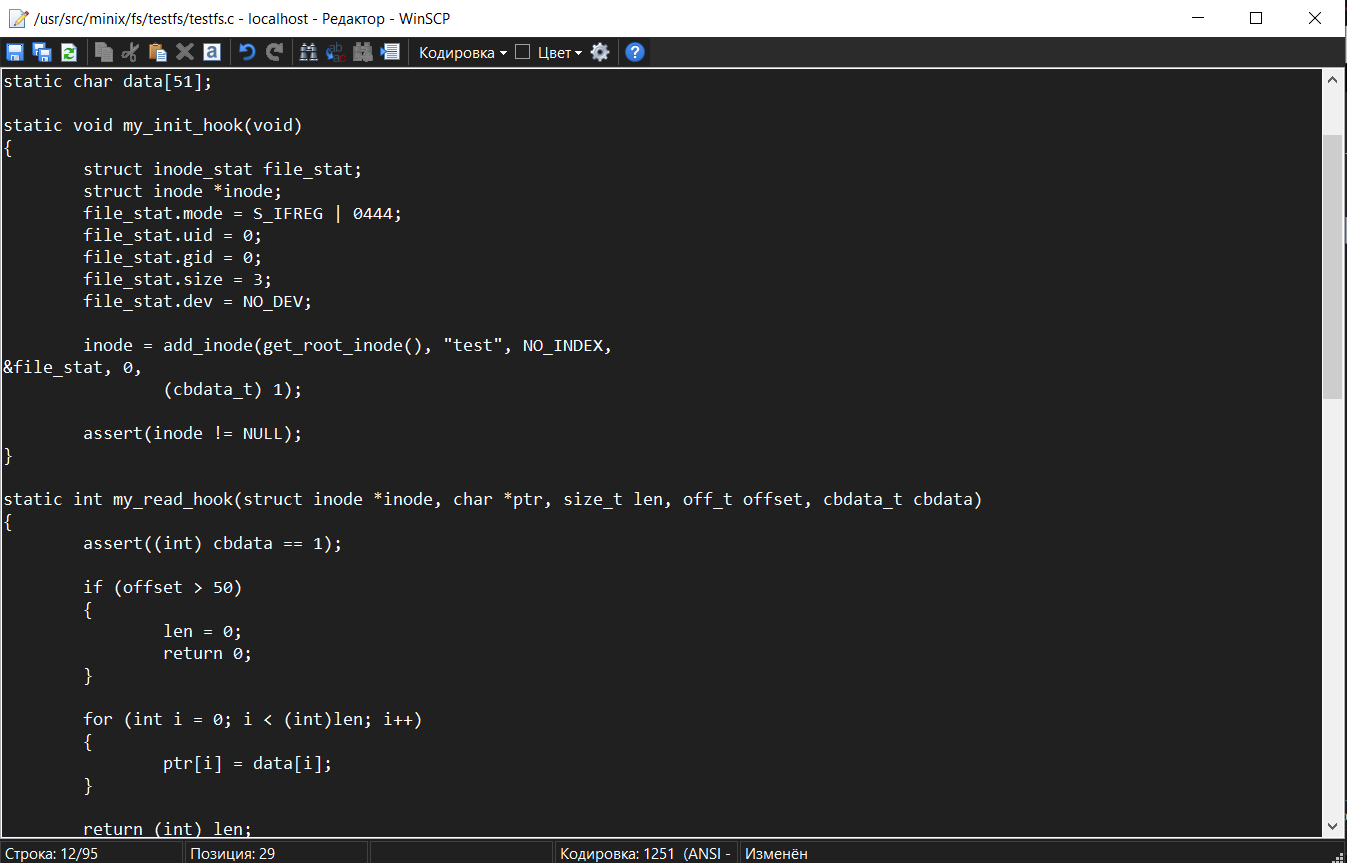
struct inode\_state root\_stat;

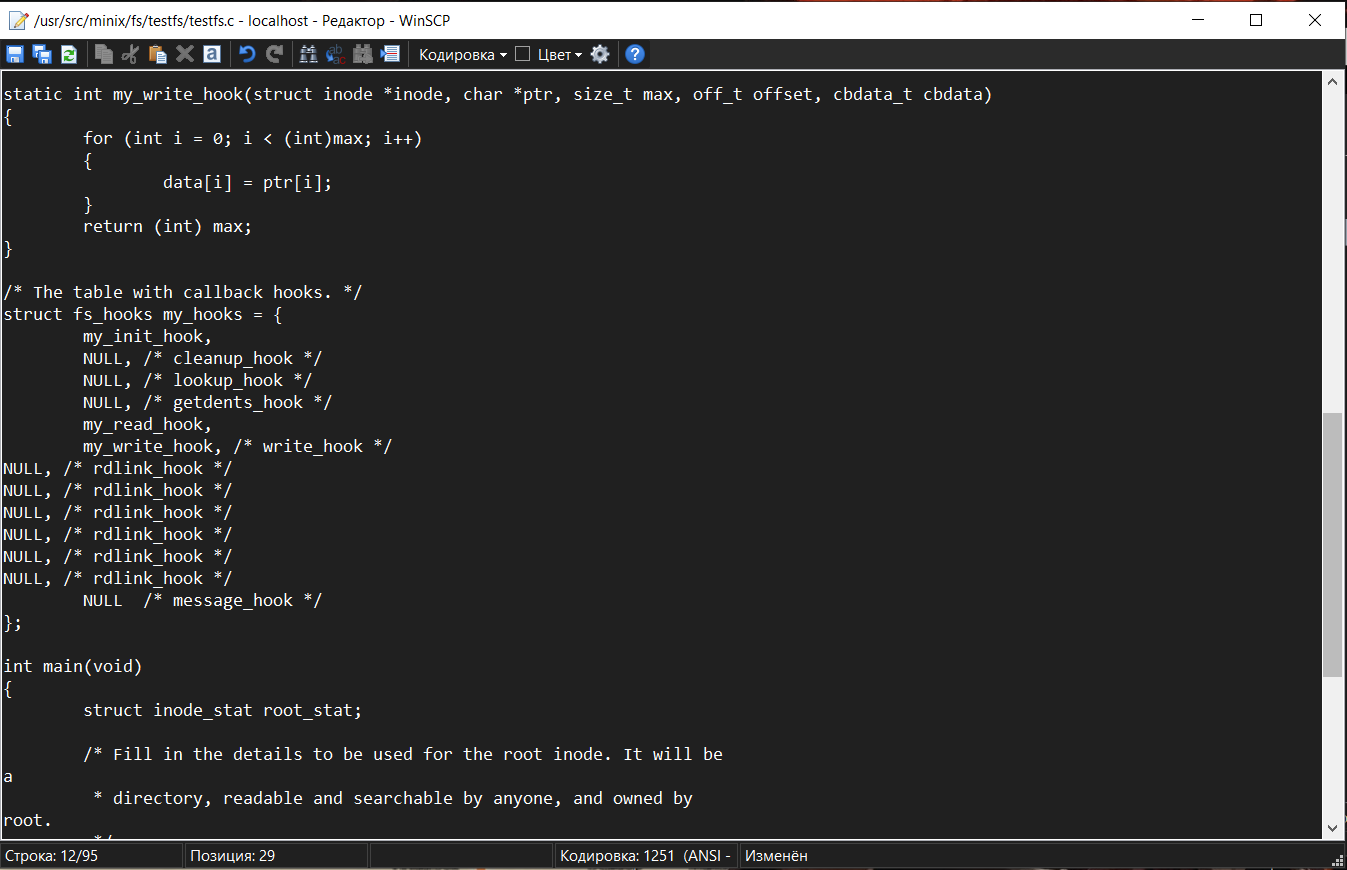
/\* /\* Fill in the details to be used for the root node. It will be

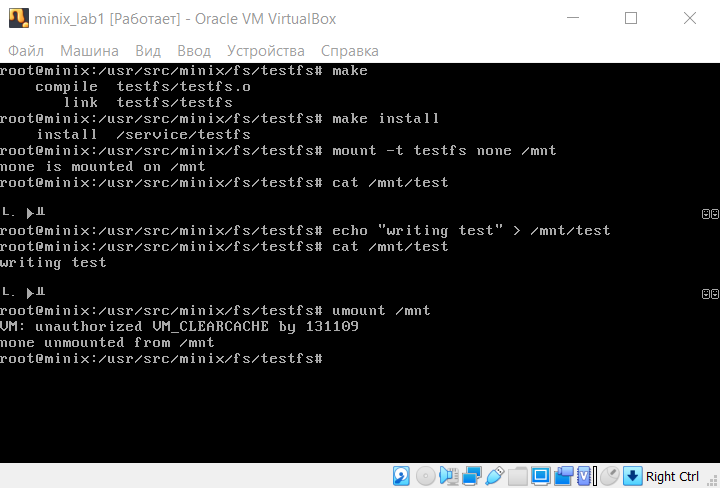
\* directory, readable and searchable by anyone, and owned by

root.







****