Отчет по лабораторной работе №2

«Бездисковая файловая система»

Выполнил: студент группы РИС-22-1б

Поважный Виталий Евгеньевич

Вариант 11

**Задание**

Необходимо:

1. В рамках сервера testfs реализовать обработчик по варианту;

2. Выявить назначение обработчика;

3. откомпилировать, проверить работоспособность.

Варианты:

1. lookup\_hook

2. write\_hook

3. trunc\_hook

4. mknod\_hook

5. unlink\_hook

6. slink\_hook

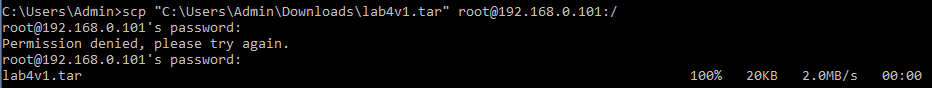
7. rdlink\_hook

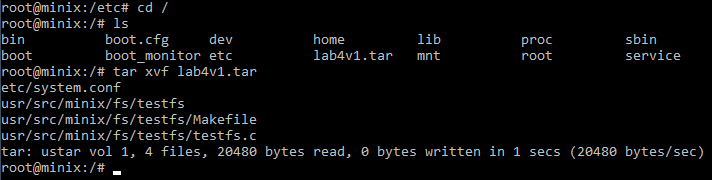
8. chstat\_hook

11 % 8 = 3 – мой вариант.

**Ход работы**

Скачиваем файл lab4v1.tar и через scp перекидываем его на minix. Далее распаковываем его.





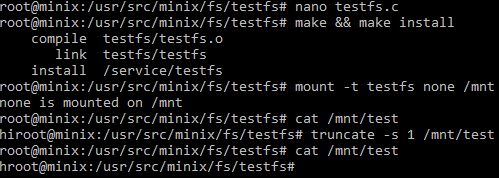
Собираем образ через makefile и монтируем его как /mnt



Командой cat /mnt/test выводятся “hihi” – все работает (хук write всегда выводит hihi).



Далее размонтируем, назначаем обработчик в testfs.c, добавляя свой хук в my\_hooks, собираем и монтируем обратно. Проводим проверку работы:



Код с хуком my\_trunk\_hook:

#include <minix/drivers.h>

#include <minix/vtreefs.h>

#include <sys/stat.h>

#include <time.h>

#include <assert.h>

#include <string.h>

#define MAX\_FILE\_SIZE 1024   
static char data[MAX\_FILE\_SIZE] = "hi";

static size\_t data\_size = 2;

static void my\_init\_hook(void) {

struct inode\_stat file\_stat;

struct inode \*inode;

file\_stat.mode = S\_IFREG | 0644;

file\_stat.uid = 0;

file\_stat.gid = 0;

file\_stat.size = 3;

file\_stat.dev = NO\_DEV;

inode = add\_inode(get\_root\_inode(), "test", NO\_INDEX, &file\_stat, 0, (cbdata\_t)1);

assert(inode != NULL);

}

static int my\_trunc\_hook(struct inode \*inode, off\_t new\_size, cbdata\_t cbdata) {

assert((int)cbdata == 1);

if (new\_size < 0 || new\_size > MAX\_FILE\_SIZE)

return EINVAL;   
  
 struct inode\_stat stat;

get\_inode\_stat(inode, &stat);

if (new\_size < data\_size) {

memset(data + new\_size, 0, data\_size - new\_size);

}

else {

memset(data + data\_size, 0, new\_size - data\_size);

}

data\_size = new\_size;

stat.size = data\_size;

set\_inode\_stat(inode, &stat);

return OK;

}

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

assert((int)cbdata == 1);

size\_t data\_len = strlen(data);

if (offset > data\_len) return 0;

if (len > data\_len - offset)

len = data\_len - offset;

memmove(ptr, data + offset, len);

return len;

}

struct fs\_hooks my\_hooks = {

.init\_hook = my\_init\_hook,

.read\_hook = my\_read\_hook,

.trunc\_hook = my\_trunc\_hook,

.lookup\_hook = NULL,

.getdents\_hook = NULL,

.message\_hook = NULL

};

int main(void) {

struct inode\_stat root\_stat = {

.mode = S\_IFDIR | 0555,

.uid = 0,

.gid = 0,

.size = 0,

.dev = NO\_DEV

};

run\_vtreefs(&my\_hooks, 10, 0, &root\_stat, 0, 0);

return 0;

}