

## Εργαστήριο Λειτουργικών Συστημάτων

Σειμένης Σπύρος	5070
Καλλιβωκάς Δημήτριος	4993

### Άσκηση 2 Ερώτημα 1

#### Λίστα τροποποιηθέντων αρχείων

/usr/src/servers/vfs/protect.c

#### Τροποποιήσεις

/usr/src/servers/vfs/protect.c

Line	Code
121	put_vnode(vp);
122	if ( r == OK)
123	printf("current owner = %d\n", uid);
124	return(r);
125	}

#### Screenshot αποτελέσματος

```
root@192 ~/oslab2> ls -l test
-rw-r--r-- 1 root  operator  0 Apr 27 23:14 test
root@192 ~/oslab2> chown user1 test
current owner=13
root@192 ~/oslab2> ls -l test
-rw-r--r-- 1 user1  operator  0 Apr 27 23:14 test
root@192 ~/oslab2> chown root test
current owner=0
root@192 ~/oslab2> ls -l test
-rw-r--r-- 1 root  operator  0 Apr 27 23:14 test
root@192 ~/oslab2> _
```

## Ερώτημα 2

Για να γίνει εκτύπωση μέσα σε συνάρτηση που καλεί το system call (π.χ do\_chown) απαιτείται η εκτέλεση από τον vfs μιας συνάρτησης βιβλιοθήκης προκειμένου να βρεθεί το όνομα του χρήστη( getpwuid() ). Η getpwuid λαμβάνει τις πληροφορίες της διαβάζοντας από το αρχείο /etc/passwd προφανώς εκτελώντας κάποιο system call στον vfs άρα δεν είναι δυνατόν ο ίδιος vfs ενώ εξυπηρετεί το νέο system call για chown να στείλει μήνυμα στον εαυτό του.

Για να ξεπεραστεί αυτό το πρόβλημα, δημιουργείται ένα νέο system call και μια νέα ρουτίνα εξυπηρέτησης στον vfs (69) με τις απαραίτητες τροποποιήσεις , καθώς και μία νέα συνάρτηση βιβλιοθήκης (mychownlib.h) για να καλεί το νέο system call, η οποία καλείται με την σειρά της από την εντολή chown. Η εκτύπωση γίνεται λοιπόν στην νέα βιβλιοθήκη και όχι σε επίπεδο system call.

### Λίστα τροποποιηθέντων αρχείων

/usr/src/commands/chown/chown.c

>>> Πλέον καλεί το νέο lib mychownlib.h κ όχι την \_chown.c

/usr/include/mychownlib.h

>>> Νέο lib βασισμένο στο /usr/src/lib/libc/posix/\_chown.c

/usr/src/include/minix/callnr.h

/usr/src/servers/vfs/table.h

/usr/src/servers/vfs/proto.h

/usr/src/servers/vfs/protect.c

>>> new system call copy of do\_chown

### Τροποποιήσεις

/usr/src/commands/chown/chown.c

Line	Code
25	#include <mychownlib.h>
:	:
135	If (mychown(file,oflag ? nuid : st.st_uid, gflag ? Ngid : st.st_gid)) {
136	perror(file);
137	error = 1;

/usr/include/mychownlib.h

Line	Code
1	#include <lib.h>
2	#define mychown _mychown
3	#include <string.h>
4	#include <unistd.h>
5	#include <pwd.h>
6	
7	PUBLIC int mychown(const char *name, uid_t owner, git_t grp)
8	{
9	message m;
10	struct passwd *pass;
11	int ret;
12	
13	m.m1_i1 = strlen(name) + 1;
14	m.m1_i2 = owner;
15	m.m1_i3 = grp;
16	m.m1_p1 = (char *) name;
17	
18	pass = getpwuid(owner);
19	ret = _syscall(FS, 69, &m); /* 69 is MYCHOWN system call */
20	if(!ret)
21	printf("current owner=%s\n",pass->pw_name);
22	
23	return ret;
24	}

/usr/src/include/minix/callnr.h

Line	Code
60	#define SETMCONTEXT           68
61	#define MYCHOWN               69
62	
63	/* Posix signal handling. */

/usr/src/servers/vfs/table.h

Line	Code
86	no_sys, /* 68 = unused */
87	do_mychown, /* 69 = do_mychown */
88	no_sys, /* 70 = unused */
89	no_sys, /* 71 = (sigaction) */

/usr/src/servers/vfs/proto.h

Line	Code
146	_PROTOTYPE ( int read_only, (struct vnode *vp) );
147	_PROTOTYPE ( int do_mychown, (void) );
148	
149	/* read.c */

/usr/src/servers/vfs/protect.c

Στο τέλος του αρχείου protect.c προσθέτω την νέα συνάρτηση do\_mychown ο κώδικας της οποίας είναι πανομοιότυπος με της do\_chown, που υπάρχει στο ίδιο αρχείο, με τις εξής αλλαγές.

Line	Code
76	PUBLIC int do_mychown()
:	:
86	if (call_nr == 69) { /* 69 is MYCHOWN code */
87	/* Temporarily open the file */

## Screenshots αποτελεσμάτων

```
root@192 ~/oslab2> ls -l test
-rw-r--r-- 1 root  operator  0 Apr 27 23:14 test
root@192 ~/oslab2> chown user1 test
current owner=user1
root@192 ~/oslab2> ls -l test
-rw-r--r-- 1 user1  operator  0 Apr 27 23:14 test
root@192 ~/oslab2> chown root test
current owner=root
root@192 ~/oslab2> ls -l test
-rw-r--r-- 1 root  operator  0 Apr 27 23:14 test
root@192 ~/oslab2> _
```