



ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΑΣΚΗΣΗ 4

ΤΟΥΡΛΙΔΑ ΒΑΓΙΑ Α.Μ: 6233

email: tourlida@ceid.upatras.gr

ΕΡΩΤΗΜΑ Α

Λίστα τροποποιηθέντων αρχείων

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

Τροποποιήσεις

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

LINE	CODE
82	printf("chmod: %s %o\n",fullpath,new_mode);

```
-bash-4.2# chmod 644 protect.c
process forked
chmod: protect.c 644
-bash-4.2# _
```

ΕΡΩΤΗΜΑ Β

Λίστα τροποποιηθέντων αρχείων

/usr/src/kernel/system/do_fork.c

Τροποποιήσεις

/usr/src/kernel/system/do_fork.c

LINE	CODE
60	printf("process forked\n");

Screenshots αποτελεσμάτων

```
-bash-4.2# ls !grep patch
process forked
process forked
create_patch
-bash-4.2# _
```

Ερώτημα C

Λίστα τροποποιηθέντων αρχείων

/usr/src/servers/pm/table.c
/usr/src/include/minix/callnr.h
/usr/src/servers/pm/proto.h
/usr/src/servers/pm/do_countProc.c
/usr/src/servers/pm/Makefile

Τροποποιήσεις

/usr/src/servers/pm/table.c

Line	Code
83	do_countProc, /* 69 = do_countProc */

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

Line	Code
68	#define COUNTPROC 69

/usr/src/servers/pm/proto.h

Line	Code
60	int do_countProc(void);

/usr/src/servers/pm/Makefile

Line	Code
------	------

7	profile.c schedule.c do_procExists.c do_countProc.c
---	---

/usr/src/servers/pm/do_countProc.c

Line	Code
1	#include <stdio.h>
2	#include "signal.h"
3	#include "pm.h"
4	
5	int do_countProc()
6	{
7	
8	return procs_in_use;
9	
10	}

Η κλήση του system call έγινε μέσα στο αρχείο

/usr/src/servers/pm/main.c

Line	Code
61	message m;
62	_syscall(PM_PROC_NR,COUNTPROC,&m)

Screenshots αποτελεσμάτων

```
-bash-4.2# ./a.out
process forked
-bash-4.2#
```

Λόγω του ότι η συνάρτηση μας δεν εκτυπώνει κάτι δεν έχουμε κάτι να εκτυπώσουμε.Γιαυτό το λόγο δείχνουμε μόνο πως έγινε το system call στη main.

```
/*=====
 *                               main                               *
 *=====*/
int main()
{
/* Main routine of the process manager. */
    int result;
    message m;
    _syscall(PM_PROC_NR,COUNTPROC,&m);
```

Ερώτημα D

Λίστα τροποποιηθέντων αρχείων

/usr/src/servers/pm/table.c
/usr/src/include/minix/callnr.h
/usr/src/servers/pm/proto.h
/usr/src/servers/pm/do_countProc.c
/usr/src/servers/pm/sys_main.c
/usr/src/servers/pm/Makefile

Τροποποιήσεις

/usr/src/servers/pm/table.c

Line	Code
84	do_procExists, /* 70 = do_procExists */

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

Line	Code
69	#define PROC EXISTS 70

/usr/src/servers/pm/proto.h

Line	Code
59	int do_procExists(void);

/usr/src/servers/pm/Makefile

Line	Code
7	profile.c schedule.c do_procExists.c do_countProc.c

/usr/src/servers/pm/do_procExists.c

Line	Code
1	#include <stdio.h>
2	#include "pm.h"
3	#include "mproc.h"
4	
5	int do_procExists()

6	{
7	int id=m_in.m1_i1;
8	int flag=0;
9	struct mproc *mp;
10	for(int i=0;i<NR_PROCS;i++)
11	{
12	mp=&mproc[i];
13	if(mp->mp_pid==id)
14	{
15	flag=1;
16	break;
17	}
18	}
19	
20	if(flag==1) return 1;
21	else return 0;
22	}

Screenshot αποτελεσμάτων:

```
-bash-4.2# ./a.out
process forked
-bash-4.2#
```

Τα αποτελέσματα πάρθηκαν χρησιμοποιώντας την παρακάτω συνάρτηση sys_main.c που δημιουργήσαμε για καλέσουμε το system call:

```
#include <lib.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char **argv)
{
    if(argc<2)
        exit(1);
    int I=atoi(argv[1]);
    message m;
    m.m1_i1;

    int val=_syscall(PM_PROC_NR, PROCEXISTS, &m);

}
```