

Υποβολή 2ης Άσκησης



Ερώτημα 1

Λίστα Τροποποιηθέντων Αρχείων

`/usr/src/servers/vfs/open.c`

Τροποποιήσεις

`/usr/src/servers/vfs/open.c`

Line	Code
611	<code>printf("New Dir -> %s, %o\n", fullpath, dirmode);</code>

Screenshots Αποτελεσμάτων

```
minix 3.2 - VMware Player (Non-commercial use only)
File Virtual Machine Help

#
# mkdir dir
New Dir -> dir 755
#
_
```

Ερώτημα 2

Λίστα Τροποποιηθέντων Αρχείων

/usr/src/lib/libsys/sys_fork.c

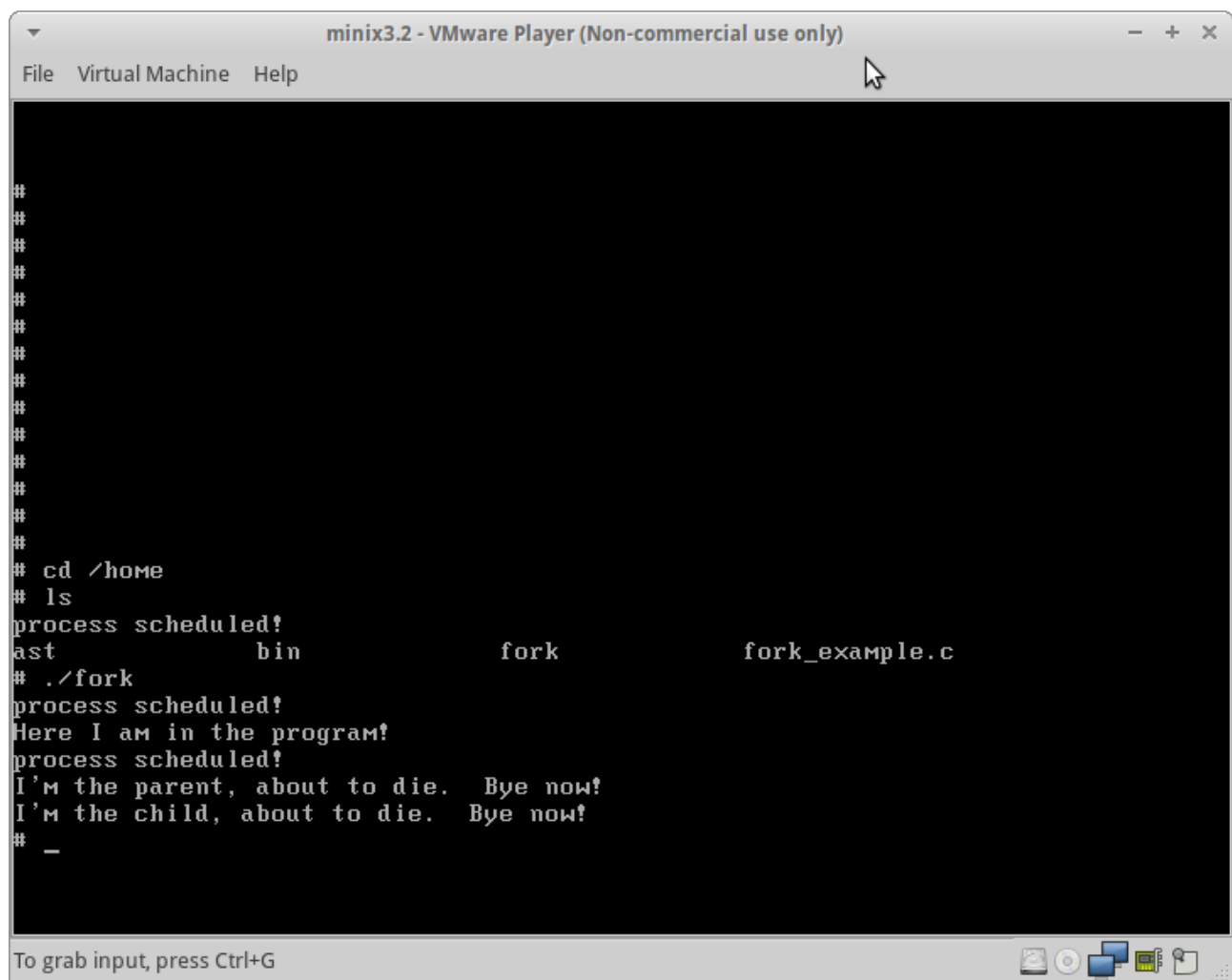
Τροποποιήσεις

/usr/src/lib/libsys/sys_fork.c

Line	Code
21	<code>printf("Process scheduled!\n");</code>

Screenshots Αποτελεσμάτων

Για να ελέγξουμε αν η αλλαγή που κάναμε ήταν σωστή τρέχουμε ένα πρόγραμμα που να περιέχει την εντολή `fork()` έτσι ώστε να δημιουργηθεί μια νέα διεργασία και ο kernel να ενημερώσει το δικό του process table.



```
minix3.2 - VMware Player (Non-commercial use only)
File Virtual Machine Help

#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
# cd /home
# ls
process scheduled!
ast          bin          fork          fork_example.c
# ./fork
process scheduled!
Here I am in the program!
process scheduled!
I'm the parent, about to die. Bye now!
I'm the child, about to die. Bye now!
# _

To grab input, press Ctrl+G
```

Ερώτημα 3

Λίστα Τροποποιηθέντων Αρχείων

```
/usr/src/include/minix/callnr.h
/usr/src/servers/vfs/table.c
/usr/src/servers/vfs/proto.h
/usr/src/servers/vfs/open.c
/usr/src/sys/sys/stat.h
/usr/src/lib/libc/sys-minix/exmkdir.c Νέο Αρχείο
/usr/src/lib/libc/sys-minix/Makefile.inc
/usr/src/bin/mkdir/mkdir.c
```

Τροποποιήσεις

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

Line	Code
69	#define EXMKDIR 69

/usr/src/servers/vfs/table.c

Line	Code
87	do_exmkdir, /* 69 = exmkdir */

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

Line	Code
169	int do_exmkdir(void);

/usr/src/servers/vfs/open.c

Line	Code
621	int do_exmkdir()
.	.
.	.
653	printf("New Dir -> %s, %o\n",fullpath,dirmode);
.	.
.	.

658

Ο κώδικας της συνάρτησης `_exmkdir` είναι ίδιος με αυτόν του πρώτου ερωτήματος διότι παρότι έγινε προσπάθεια δεν βρέθηκε μεταβλητή τέτοια που να περιέχει το `fullpath` που ζητάει το ερώτημα

/usr/src/sys/sys/stat.h

Line	Code
229	<code>int exmkdir(const char *, mode_t);</code>

/usr/src/lib/libc/sys-minix/exmkdir.c

Εδώ περιέχεται ο κώδικας της βοηθητικής συνάρτησης με την οποία καλούμε `syscall_EXMKDIR`

Line	Code
12	<code>int mkdir(const char *name, mode_t mode)</code>
19	<code>return(_syscall(VFS_PROC_NR, EXMKDIR, &m));</code>

/usr/src/lib/libc/sys-minix/Makefile.inc

Προσθέτουμε στο Makefile το αρχείο που δημιουργήσαμε

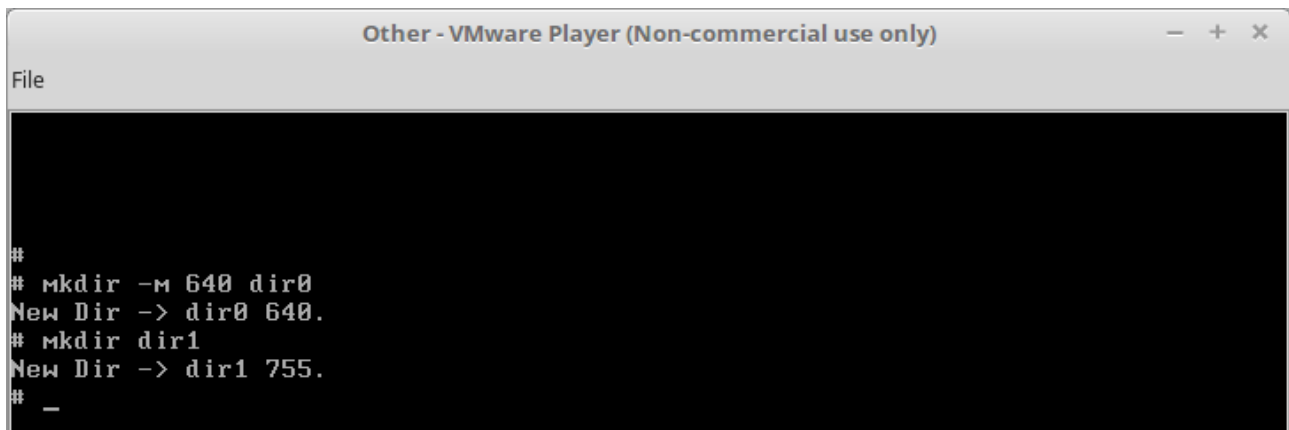
Line	Code
5	<code>connect.c dup.c dup2.c execve.c exmkdir.c fcntl.c flock.c fpathconf.c fork.c \</code>

/usr/src/bin/mkdir/mkdir.c

Τροποποιούμε το command `mkdir` ώστε να καλεί την βοηθητική συνάρτηση που φτιάξαμε.

Line	Code
124	<code>if (exmkdir(*argv, mode) < 0) {</code>
169	<code>rv = exmkdir(path, done ? mode : dir_mode);</code>

Screenshots Αποτελεσμάτων



The screenshot shows a VMware Player window titled "Other - VMware Player (Non-commercial use only)". The window contains a terminal window with a black background and white text. The terminal output shows the following commands and their results:

```
#  
# mkdir -m 640 dir0  
New Dir -> dir0 640.  
# mkdir dir1  
New Dir -> dir1 755.  
# _
```

Ερώτημα 4

Λίστα Τροποποιηθέντων Αρχείων

```
/usr/src/include/minix/callnr.h
/usr/src/servers/pm/table.c
/usr/src/servers/pm/proto.h
/usr/src/servers/pm/misc.c
/usr/src/sys/sys/stat.h
/usr/src/lib/libc/sys-minix/erotima4.c Νέο αρχείο
/usr/src/lib/libc/sys-minix/Makefile.inc
/usr/src/commands/Makefile
/usr/src/commands/erotima4/erotima4.c Νέο αρχείο
/usr/src/commands/erotima4/Makefile Νέο αρχείο
```

Τροποποιήσεις

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

Line	Code
59	int do_erotima4(void);

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

Line	Code
84	do_erotima4, /* 70 = erotima4 */

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

Line	Code
173	int do_exmkdir(void);

```
/usr/src/servers/pm/misc.c
```

Line	Code
499	<pre>/ *===== ** * do_erotima4</pre>

```

*
*=====
==*/
int do_erotima4()
{
    int i, sum_in_seconds, counter=0;
    clock_t clock_usr=0;        //user time
    clock_t clock_sys=0;        //sys time
    clock_t sum;                //_sinolikos xronos

    /*pername ola ta slots tou pinaka tw n diergasiwn */
    for (i=0; i<NR_PROCS; i++)
    {
        /* Theloume na metrisoume mono tis energes diergasies.
        Oi energes diergasies tha xaraktirizonte apo ta flags IN_USE,
        WAITING kai PAUSED
        */
        if (mproc[i].mp_flags % 11)
        {
            counter++;          //Afksisi tw n arithmwn tw n diergasiwn pou
            einai energes
            clock_usr+=mproc[i].mp_child_utime;
            clock_sys+=mproc[i].mp_child_stime;
        }
    }

    printf("Number of active processes: %d\n",counter);

    sum=clock_usr + clock_sys;
    sum_in_seconds = (int)sum / (int)CLOCKS_PER_SEC;

    printf("(usr+sys) time of active processes %d second(s).\n",
    sum_in_seconds);

    return 0;
}

```

532

/usr/src/sys/sys/stat.h

Line	Code
231	int erotima4(int);

/usr/src/lib/libc/sys-minix/erotima4.c

Line	Code
1	<pre> #include <sys/cdefs.h> #include "namespace.h" #include <lib.h> #include <sys/stat.h> #include <string.h> int erotima4() </pre>

14	<pre> { message m; return(_syscall(PM_PROC_NR, EROTIMA4, &m)); } </pre>
----	--

/usr/src/lib/libc/sys-minix/Makefile.inc

Line	Code
5	<pre> connect.c dup.c dup2.c erotima4.c erotima5.c execve.c exmkdir.c fcntl.c flock.c fpathconf.c fork.c \ </pre>

/usr/src/commands/Makefile

Line	Code
11	<pre> eject env erotima4 erotima5 expand factor fbdctl \ </pre>

Με αυτή τη προσθήκη στο συγκεκριμένο μπορούμε να κάνουμε χρήση του νέου system call που δημιουργήσαμε

/usr/src/commands/erotima4/erotima4.c

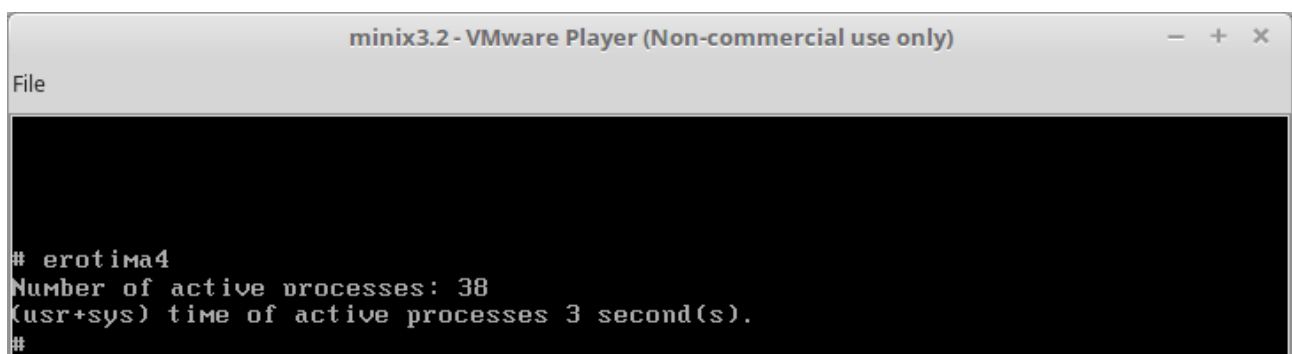
Line	Code
1	<pre> #include <sys/cdefs.h> #ifndef lint #endif #ifndef lint #if 0 static char sccsid[] = "@(#)erotima4.c; #else __RCSID(""); #endif #endif /* not lint */ #include <sys/param.h> #include <sys/stat.h> #include <sys/types.h> #include <err.h> #include <errno.h> #include <locale.h> #include <stdio.h> #include <stdlib.h> #include <string.h> #include <unistd.h> void usage(void); int main(int, char *[]); int main(int argc, char *argv[]) { if (argc==1) { erotima4(); //Kaleitai i voithitiki sinartisi } else usage(); } </pre>

	<pre> return 0; } void usage(void) { (void)fprintf(stderr, "usage: %s [NoArgument] \n", getprogname()); exit(0); } </pre>
45	}

/usr/src/commands/erotima4/Makefile

Line	Code
1	<pre> PROG= erotima4 MAN= </pre>
4	<pre> .include <bsd.prog.mk> </pre>

Screenshots Αποτελεσμάτων



Ερώτημα 5

Λίστα Τροποποιηθέντων Αρχείων

```
/usr/src/include/minix/callnr.h
/usr/src/servers/pm/table.c
/usr/src/servers/pm/proto.h
/usr/src/servers/pm/misc.c
/usr/src/sys/sys/stat.h
/usr/src/lib/libc/sys-minix/erotima5.c Νέο αρχείο
/usr/src/lib/libc/sys-minix/Makefile.inc
/usr/src/commands/Makefile
/usr/src/commands/erotima5/erotima5.c Νέο αρχείο
/usr/src/commands/erotima5/Makefile Νέο αρχείο
```

Τροποποιήσεις

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

Line	Code
59	<code>int do_erotima5(void);</code>

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

Line	Code
72	<code>do_erotima5,/* 58 = erotima5 */</code>

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

Line	Code
60	<code>int do_erotima5(void);</code>

```
/usr/src/servers/pm/misc.c
```

Line	Code
534	<pre>/ ===== ==* * * * ===== ==*/ int do_erotima5 ()</pre>

	<pre> { int i, flag=0; int var = m_in.m1_i1; //Diavazoume to tin metavliti tou minimatos pou periexei to id /*pername ola ta slots tou pinaka twv diergasiwn */ for (i=0; i<NR_PROCS; i++) { if (mproc[i].mp_pid == var) { flag=1; //An den vrethei process me afto to id, tote den iparxei afto to process if (mproc[i].mp_flags % 11) printf("Process is active.\n"); else printf("Process is inactive.\n"); } } if (flag==0) //Tipwnetai minima an den iparxei to process printf("erotima5: %d: no such process\n", var); return 0; } </pre>
562	}

/usr/src/sys/sys/stat.h

Line	Code
232	int erotima5(int);

/usr/src/lib/libc/sys-minix/erotima5.c

Line	Code
1	<pre> #include <sys/cdefs.h> #include "namespace.h" #include <lib.h> #include <sys/stat.h> #include <string.h> int erotima5(p_id) int p_id; { message m; m.m1_i1 = p_id; return(_syscall(PM_PROC_NR, EROTIMA5, &m)); } </pre>
17	}

/usr/src/lib/libc/sys-minix/Makefile.inc

Line	Code
5	connect.c dup.c dup2.c erotima4.c erotima5.c execve.c exmkdir.c fcntl.c flock.c fpathconf.c fork.c \

/usr/src/commands/Makefile

Line	Code
11	eject env erotima4 erotima5 expand factor fbdctl \

/usr/src/commands/erotima5/erotima5.c

Line	Code
1	<pre>#include <sys/cdefs.h> #ifndef lint #endif /* not lint */ #ifndef lint #if 0 static char sccsid[] = "@(#)erotima5.c"; #else __RCSID(""); #endif #endif /* not lint */ #include <sys/param.h> #include <sys/stat.h> #include <sys/types.h> #include <err.h> #include <errno.h> #include <locale.h> #include <stdio.h> #include <stdlib.h> #include <string.h> #include <unistd.h> void usage(void); int main(int, char *[]); int main(int argc, char *argv[]) { if (argc==2) { int id = atoi (argv[1]); //Pernoume to orisma pou edwse o xristis pifp (id); //Kaloume tin sinartisi erotima5() } else { usage(); } return 0; } void usage(void)</pre>

	{
	(void)fprintf(stderr, "usage: %s [pid] \n",
	getprogname());
	exit (0);
48	}

/usr/src/commands/erotima5/Makefile

Line	Code
1	PROG = erotima5 MAN =
4	.include <bsd.prog.mk>

Screenshots Αποτελεσμάτων

The screenshot shows a terminal window titled "minix3.2 - VMware Player (Non-commercial use only)". The terminal output is as follows:

```
# ps
PID TTY  TIME CMD
140  co   0:00 -sh
141  c1   0:00 getty
142  c2   0:00 getty
143  c3   0:00 getty
155  co   0:00 ps
# erotima5 140
Process is active.
# erotima5 130
erotima5: 130: no such process
#
```