

Εργαστήριο Λειτουργικών Συστημάτων

1η Άσκηση

Αλέξιος Ζαμάνης

03115010

Μιχάηλ Μεγγίσογλου

03115014

13 Ιανουαρίου 2019

1 Δοκιμασίες

0. Πρόβλημα:

```
$ strace ./riddle
openat(AT_FDCWD, ".hello_there", O_RDONLY) = -1
    ENOENT (No such file or directory)
```

Λύση:

```
$ touch .hello_there
```

Σχόλια: Δεν υπάρχει το αρχείο, ούτε το ανοίγουμε με O_CREAT, οπότε το δημιουργούμε.

1. Πρόβλημα:

```
$ strace ./riddle
openat(AT_FDCWD, ".hello_there", O_WRONLY) = 4
```

Λύση:

```
$ chmod -w .hello_there
```

Σχόλια: Αφαιρούμε το δικαίωμα εγγραφής από το αρχείο.

2. Πρόβλημα:

```
$ strace ./riddle
rt_sigaction(SIGALRM, {sa_handler=0x55e2e8605d40,
    sa_mask=[ALRM], sa_flags=SA_RESTORER|SA_RESTART,
    sa_restorer=0x7f0e02f31f20}, {sa_handler=SIG_DFL,
    sa_mask=[], sa_flags=0}, 8) = 0
rt_sigaction(SIGCONT, {sa_handler=0x55e2e8605d40,
    sa_mask=[CONT], sa_flags=SA_RESTORER|SA_RESTART,
    sa_restorer=0x7f0e02f31f20}, {sa_handler=SIG_DFL,
    sa_mask=[], sa_flags=0}, 8) = 0
alarm(10) = 0
pause() = ? ERESTARTNOHAND (To be restarted if no
    handler)
--- SIGALRM {si_signo=SIGALRM, si_code=SI_KERNEL}
---
rt_sigreturn({mask=[]}) = -1 EINTR (Interrupted
    system call)
```

Λύση:

```
$ ./riddle
```

```
$ pkill -SIGCONT riddle
```

Σχόλια: Ορίζονται δύο handlers, ένας για SIGALRM και ένας για SIGCONT. Μέσα σε 10 δευτερόλεπτα στέλνουμε SIGCONT στη διεργασία, πριν τη σκοτώσει ο handler της SIGALRM.

3. Πρόβλημα:

```
$ ltrace ./riddle  
getenv("ANSWER") = nil
```

Λύση:

```
$ export ANSWER=42
```

Σχόλια: Ορίζουμε τη μεταβλητή περιβάλλοντος ANSWER και της δίνουμε την τιμή 42 (προφανώς).

4. Πρόβλημα:

```
$ strace ./riddle  
openat(AT_FDCWD, "magic_mirror", O_RDWR) = -1 ENOENT  
      (No such file or directory)
```

Λύση:

```
$ mkfifo magic_mirror
```

Σχόλια: Δημιουργούμε ένα named pipe με το όνομα magic_mirror, ώστε η διεργασία να μπορεί με FIFO τρόπο να διαβάζει ό,τι γράφει.

5. Πρόβλημα:

```
$ strace ./riddle  
fcntl(99, F_GETFD) = -1 EBADF (Bad file descriptor)
```

Λύση:

```
$ exec 99<>foo
```

Σχόλια: Δημιουργούμε και ανοίγουμε ένα αρχείο ονόματι foo με περιγραφητή 99. Αυτός κληρονομείται από τη riddle.

6. Πρόβλημα:

```
$ strace -f ./riddle
[pid 545] read(33, <unfinished ...>
[pid 545] <... read resumed> 0x7ffe758cbcdc, 4) = -1
        EBADF (Bad file descriptor)
[pid 544] write(34, "\0\0\0\0", 4 <unfinished ...>
[pid 544] <... write resumed> ) = -1 EBADF (Bad file
        descriptor)
$ strace -f ./challenge6
[pid 623] read(33, <unfinished ...>
[pid 623] <... read resumed> 0x7ffe758cbcdc, 4) = -1
        EBADF (Bad file descriptor)
[pid 622] write(34, "\0\0\0\0", 4 <unfinished ...>
[pid 622] <... write resumed> ) = -1 EBADF (Bad file
        descriptor)
[pid 622] read(53, <unfinished ...>
[pid 622] <... read resumed> 0x7ffa6953b4c, 4) = -1
        EBADF (Bad file descriptor)
[pid 623] write(54, "\1\0\0\0", 4) = -1 EBADF (Bad
        file descriptor)
```

Λύση:

```
$ ./challenge6
```

Σχόλια: Ανοίγουμε 2 pipes και αντιγράφουμε τα άκρα τους στους ζητούμενους περιγραφητές, ήτοι 33, 34, 53 και 54.

7. Πρόβλημα:

```
$ strace ./riddle
lstat(".hey_there", 0x7fff5fed1390) = -1 ENOENT (No
        such file or directory)
```

Λύση:

```
$ ln .hello_there .hey_there
```

Σχόλια: Δημιουργούμε ένα σύνδεσμο στο αρχείο .hello_there ονόματι .hey_there.

8. Πρόβλημα:

```
$ strace ./riddle
openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
        such file or directory)
```

```

openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "bf00", O_RDONLY) = -1 ENOENT (No
such file or directory)

```

Λύση:

```

$ for i in {0..9}
do
    ln -s /dev/urandom bf0${i}
done

```

Σχόλια: Δημιουργούμε συμβολικούς συνδέσμους στο urandom ονόματι .bf0{0..9}, οπότε αυτό επιστρέφει τυχαία δεδομένα για ανάγνωση.

9. Πρόβλημα:

```

$ strace ./riddle
connect(4, {sa_family=AF_INET, sin_port=htons(49842)
, sin_addr=inet_addr("127.0.0.1")}, 16) = -1
ECONNREFUSED (Connection refused)

```

Λύση:

```

$ nc -l localhost 49842
How much is 16054 + 1? 16055

$ ./riddle

```

Σχόλια: Ανοίγουμε μια σύνδεση TCP στο localhost με τη ζητούμενη θύρα και ακούμε σε αυτήν για την επρώτηση του προγράμματος. Έπειτα απαντάμε.

10. Πρόβλημα:

```
$ strace ./riddle
openat(AT_FDCWD, "secret_number", O_RDWR|O_CREAT|
      O_TRUNC, 0600) = 4
unlink("secret_number") = 0
write(4, "The number I am thinking of righ"... ,
      4096) = 4096
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_SHARED,
      4, 0) = 0x7f1c5d832000
close(4) = 0
```

Λύση:

```
$ touch secret_number
$ ./challenge10
```

Σχόλια: Δημιουργούμε το αρχείο secret_number. Έπειτα το πρόγραμμα μας το ανοίγει, ώστε να μη διαγραφεί με το unlink. Το riddle γράφει στο αρχείο. Εμείς το διαβάζουμε ύστερα από 1 δευτερόλεπτο και το κλείνουμε. Έπειτα γράφουμε την απάντηση.

11. Σχόλια: Ακριβώς όπως η προηγούμενη.

12. Πρόβλημα:

```
$ strace ./riddle
openat(AT_FDCWD, "/tmp/riddle-0j0oR8", O_RDWR|
      O_CREAT|O_EXCL, 0600) = 4
ftruncate(4, 4096) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_SHARED,
      4, 0) = 0x7fe7b419c000
```

Λύση:

```
$ ls /tmp | grep riddle
$ ./challenge12 A /tmp/riddle-0j0oR8
```

Σχόλια: Τρέχουμε το riddle. Μέσα σε λίγα δευτερόλεπτα βρίσκουμε το όνομα του αρχείου που δημιούργησε, τρέχουμε το πρόγραμμά μας με παραμέτρους το ζητούμενο γράμμα και το όνομα, και αυτό γράφει στο εν λόγω αρχείο το γράμμα στο σωστό offset.

13. Πρόβλημα:

```
$ strace ./riddle
openat(AT_FDCWD, ".hello_there", O_RDWR|O_CREAT,
      0600) = -1 EACCES (Permission denied)
$ strace ./riddle
ftruncate(4, 16384) = 0
```

Λύση:

```
$ chmod +w .hello_there
$ ./challenge13
```

Σχόλια: Επαναφέρουμε το δικαίωμα εγγραφής στο αρχείο .hello_there. Το πρόγραμμά μας τρέχει το riddle, που μικραίνει το μέγεθός του αρχείου, και το επαναφέρει στην αρχική τιμή του.

14. Πρόβλημα:

```
$ strace ./riddle
getpid() = 7291
```

Λύση:

```
$ ./challenge14
```

Σχόλια: Το πρόγραμμά μας κάνει fork το riddle μέχρι να πετύχει το σωστό pid.

2 Παράρτημα: Πηγαίος κώδικας

2.1 challenge6.c

```
#include <unistd.h>

int main()
{
    int pipefd1[2];
    int pipefd2[2];
    pipe(pipefd1);
    pipe(pipefd2);
    dup2(pipefd1[0], 33);
    dup2(pipefd1[1], 34);
    dup2(pipefd2[0], 53);
    dup2(pipefd2[1], 54);
```

```

    execl("riddle", "riddle", NULL);
}

```

2.2 challenge10.c

```

#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>

int main()
{
    int fd = open("secret_number", O_RDONLY);
    pid_t p = fork();
    if (p == 0)
        execl("riddle", "riddle", NULL);
    sleep(1);
    char buf[4096];
    read(fd, buf, 4096);
    FILE *stream = fopen("out", "w");
    fprintf(stream, "%s\n", buf);
    fclose(stream);
    close(fd);
    sleep(10);
    unlink("out");
}

```

2.3 challenge12.c

```

#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int fd = open(argv[2], O_WRONLY);
    lseek(fd, 0x6f, SEEK_SET);
    char buf[1];
    buf[0] = *argv[1];
    write(fd, buf, 1);
}

```



```
    close(fd);  
}
```

2.4 challenge13.c

```
#include <sys/types.h>  
#include <sys/stat.h>  
#include <fcntl.h>  
#include <unistd.h>  
  
int main()  
{  
    int fd = open(".hello_there", O_WRONLY);  
    pid_t p = fork();  
    if (p == 0)  
        execl("riddle", "riddle", NULL);  
    sleep(1);  
    ftruncate(fd, 32768);  
    close(fd);  
}
```

2.5 challenge14.c

```
#include <sys/types.h>  
#include <unistd.h>  
#include <stdlib.h>  
  
int main()  
{  
    pid_t p = getpid();  
    while (p != 32767)  
    {  
        p = fork();  
        if (p == 0)  
        {  
            if (getpid() == 32767)  
                execl("riddle", "riddle", NULL);  
            exit(0);  
        }  
    }  
}
```