

## تمرین سوم آزمایشگاه سیستم‌های عامل

با دستور `cat > oslabfile1.txt` فایل را ایجاد کرده، محتوا دلخواه را نوشته و با `Ctrl+D` آن را ذخیره می‌کنیم (از Windows Subsystem استفاده شده).

با اجرای دستور `ls -li oslabfile1.txt` می‌توانیم مشخصات فایل را مشاهده کنیم، با دستور `ln` آن را به فایل دوم لینک می‌کنیم

```
javad@Xdeveloper:~$ cat > oslabfile1.txt
Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.
javad@Xdeveloper:~$ ls -li oslabfile1.txt
1608 -rw-r--r-- 1 javad javad 138 Dec 15 18:34 oslabfile1.txt
javad@Xdeveloper:~$ ln oslabfile1.txt oslabfile2.txt
```

سوال (1) با فراخوانی مجدد دستور `ln`، مقدار inode number برای هر دو فایل نشان داده می‌شود که برابر 1608 است

```
javad@Xdeveloper:~$ ls -li oslabfile1.txt
1608 -rw-r--r-- 2 javad javad 138 Dec 15 18:34 oslabfile1.txt
javad@Xdeveloper:~$ ls -li oslabfile2.txt
1608 -rw-r--r-- 2 javad javad 138 Dec 15 18:34 oslabfile2.txt
```

سوال (2) بله

سوال (3) با دستور `cat` محتوای هر دو فایل را مشاهده می‌کنیم که محتوای آن یکسان اند

```
javad@Xdeveloper:~$ cat oslabfile1.txt
Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.
javad@Xdeveloper:~$ cat oslabfile2.txt
Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.
```

سوال (4) بله، با ویرایش فایل اول یک خط جدید به آن اضافه می‌کنیم، و با استفاده مجدد از دستور `cat` محتوای هر دو فایل را مشاهده می‌کنیم که دوباره یکسان اند

```
javad@Xdeveloper:~$ echo 'yet another line' >> oslabfile1.txt
javad@Xdeveloper:~$ cat oslabfile1.txt
Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.
yet another line
javad@Xdeveloper:~$ cat oslabfile2.txt
Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.
yet another line
javad@Xdeveloper:~$ |
```

سوال (5) بله

```
javad@Xdeveloper:~$ rm oslabfile1.txt
javad@Xdeveloper:~$ cat oslabfile2.txt
Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.
yet another line
```

سوال 6)

**Strace** یک ابزار CLI است که برای بررسی خطاهای موجود در سیستم استفاده می شود پس از امکان نظارت بر مکالمات سیستمی را فراهم می کند توسط یک برنامه خاص و تمام سیگنالهایی که دریافت می کند استفاده می شود. نرم افزار به کاربر اجازه می دهد تا نظارت و) از نسخه 4.15 (در روند تعامل بین برنامه و هسته مداخله کنید از جمله تماس های مداوم سیستم ، پنجره های بازشو و تغییرات وضعیت روند

عملکرد آن توسط ویژگی هسته linux به نام ptrace امکان پذیر است. این برنامه مشابه برنامه خرابایی موجود در سایر سیستم های یونیکس است. برنامه Cygwin ابزار مشابهی را ارائه می دهد. متداولترین کاربرد آن شروع آن همراه با برنامه ردیابی شده است ، که لیستی از سیستمی را که اجرا می کند چاپ می کند.

**مفید است که علت خرابی یک برنامه خاص را بیابید** یا به دلیل اینکه موقعیت هایی را گزارش می کند که مثلاً برنامه سعی دارد به فایلی دسترسی پیدا کند یا برای آن مجوز خواندن ندارد.

از ویژگی های برجسته **Strace** ، موارد زیر برجسته است:

- بتوانید فیلتر نام syscall را برای ردیابی تعیین کنید) با استفاده از گزینه e trace - (= با نام ، مانند کلون ، چنگال ، vfork؛ استفاده از یکی از گروههای از پیش تعریف شده ، مانند %ipc.%file. (از نسخه 4.17) با استفاده از نحو بیان منظم ، مانند.\*\_clock
- لیستی از مسیرهای ردیابی را مشخص کنید) به عنوان مثال.(P /etc/ld.so.cache -
- لیستی از توضیحات پرونده را مشخص کنید که ورودی / خروجی آنها باید ریخته شود
- زمان اعدام و شمارش syscall را بشمارید
- چاپ مهرهای زمانی نسبی یا مطلق
- کد بازگشت و خطای تماسهای سیستم مشخص شده را تغییر داده و سیگنالها را پس از اجرا تزریق کنید
- استخراج اطلاعات در توصیفگرهای پرونده (از جمله سوکت).
- چاپ ردیابی پشته ، شامل (از نسخه 4.21) نماد تقاضا.(k-)
- فیلتر کردن با وضعیت بازگشت syscall

- strace از رمزگشایی آرگومانهای برخی کلاسهای دستور ioctl مانند BTRFS\_\* ، V4L2\_\* ، DM\_\* ، NSFS\_\* ، MEM\_\* ، EVIO\_\* ، KVM\_\* و چندین مورد دیگر پشتیبانی می کند.

همانطور که **strace** فقط جزئیات سیستم را فرا می خواند ، نمی تواند به اندازه مشکلات اشکال زدایی کد مانند GNU Debugger (gdb) شناسایی شود. با این حال ، استفاده از آن از اشکال زدایی کد آسان تر است و یک ابزار بسیار مفید برای مدیران سیستم است. همچنین توسط محققان برای ایجاد ردیابی فراخوانی سیستم برای پخش بعدی استفاده می شود.

سوال (7)

UNIX select(2) system call استفاده شده، در واقع دستور rm از `unlink` برای سیستم کال اصلی در کنار سایر کال سیستم کال ها مانند `execve` استفاده کرده، `strace` میتواند سایر عملیات ها را نمایش دهد:

```
javad@XZdevleper:~$ strace rm oslabfile2.txt
execve("/usr/bin/rm", ["rm", "oslabfile2.txt"], 0x7ffeca2bb838 /* 24 vars */) = 0
brk(NULL)                               = 0x55a5d2527000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe56b46680) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8de03e1000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=14731, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 14731, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8de03dd000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0P\237\2\0\0\0\0\0"... , 832) = 832
pread64(3, "\6\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0..." , 784, 64) = 784
pread64(3, "\4\0\0\0\0\0\0\0\0\0\0GNU\0\2\0\0\300\4\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0..." , 48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\360\352,\223\340"."... , 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2216304, ...}, AT_EMPTY_PATH) = 0
pread64(3, "\6\0\0\0\4\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0..." , 784, 64) = 784
mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f8de01b5000
mmap(0x7f8de01dd000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f8de01dd000
mmap(0x7f8de0372000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f8de0372000
mmap(0x7f8de03ca000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7f8de03ca000
mmap(0x7f8de03d0000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f8de03d0000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8de01b2000
arch_prctl(ARCH_SET_FS, 0x7f8de01b2740) = 0
set_tid_address(0x7f8de01b2a10)         = 84
set_robust_list(0x7f8de01b2a20, 24)     = 0
rseq(0x7f8de01b30e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f8de03ca000, 16384, PROT_READ) = 0
mprotect(0x55a5d05a6000, 4096, PROT_READ) = 0
mprotect(0x7f8de041b000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f8de03dd000, 14731)           = 0
getrandom("\x2b\xfa\x6d\x6c\xd\x6c\x0c\x30", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0x55a5d2527000
brk(0x55a5d2548000)                     = 0x55a5d2548000
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/share/locale/locale.alias", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2996, ...}, AT_EMPTY_PATH) = 0
```

سوال 8) با دستور `cat > oslabfile3.txt` فایل را ایجاد کرده، محتوا دلخواه را نوشته و آن را ذخیره می‌کنیم. حال با دستور `ln -s` یک لینک نرم بین آن دو ایجاد می‌کنیم، با دستور `ls` مشاهده می‌کنیم مقدار `inode` آنها متفاوت است

```
javad@Xdeveloper:~$ cat > oslabfile3.txt
Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature
from 45 BC
javad@Xdeveloper:~$ ln -s oslabfile3.txt oslabfile4.txt
javad@Xdeveloper:~$ ls -li oslabfile*.txt
1608 -rw-r--r-- 1 javad javad 132 Dec 15 21:44 oslabfile3.txt
1615 lrwxrwxrwx 1 javad javad 14 Dec 15 21:45 oslabfile4.txt -> oslabfile3.txt
```

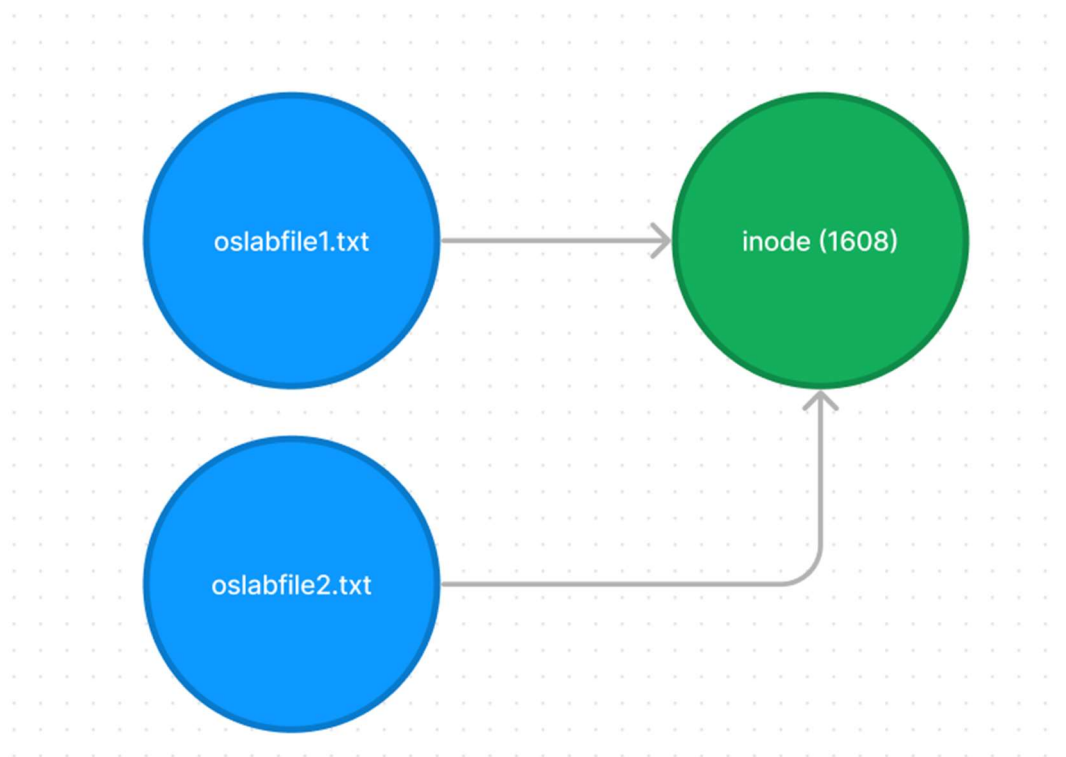
سوال 9) بله

```
javad@Xdeveloper:~$ echo 'This is a test' > oslabfile4.txt
javad@Xdeveloper:~$ cat oslabfile3.txt
This is a test
```

سوال 10) فایل 3 را حذف و فایل 4 را آپدیت می‌کنیم، اما مشاهده می‌کنیم مقدار هر دو فایل تغییر کرده، اما فایل 3 قبلاً حذف شده بود! برای اطمینان دوباره فایل 3 را حذف و مقدار آن را مشاهده می‌کنیم که به خطای عدم وجود فایل مواجه می‌شویم، سپس مقدار فایل 4 را دوباره آپدیت می‌کنیم، حال مجدداً خروجی می‌گیریم. با توجه به این موضوع به خاطر لینک نرم بین دو فایل ویرایش مقدار فایل 4 باعث آپدیت و در صورت عدم وجود ساختن فایل 3 خواهد شد.

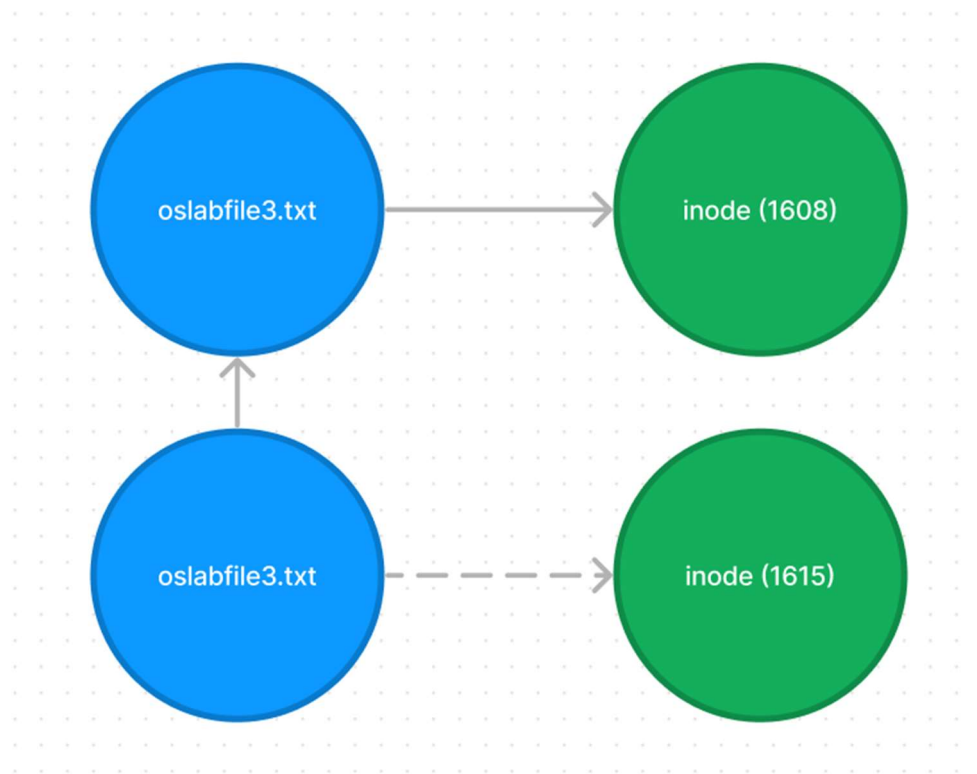
```
javad@Xdeveloper:~$ rm oslabfile3.txt
javad@Xdeveloper:~$ echo 'This is a test 2' > oslabfile4.txt
javad@Xdeveloper:~$ cat oslabfile4.txt
This is a test 2
javad@Xdeveloper:~$ cat oslabfile3.txt
This is a test 2
javad@Xdeveloper:~$ rm oslabfile3.txt
javad@Xdeveloper:~$ cat oslabfile3.txt
cat: oslabfile3.txt: No such file or directory
javad@Xdeveloper:~$ echo 'This is a test 3' > oslabfile4.txt
javad@Xdeveloper:~$ cat oslabfile3.txt
This is a test 3
javad@Xdeveloper:~$ |
```

سوال 11) ابتدا باید ارتباط مفهوم فایل و inode را متوجه شده تا بتوانیم سافت و هارد را تعریف کنیم. در حالت معمول کانسپتی که از فایل در ذهن ما ایجاد شده پیوند مستقیمی با محتوای آن در هارد دارد در حالی که اینگونه نیست. فایل را می توان به عنوان متا داده ای برای داده های فایل در نظر گرفت که در سیستم مدیریت فایل معرفی می شود. در این تمرین برای فایل 1 و 2 نمودار زیر را داریم:



در اینجا فایل 2 به طور مستقیم به محتوای فایل 1 لینک شده، ما حتی اگر فایل 1 را تغییر نام دهیم نیز تاثیری بر فایل 2 نخواهد داشت. ارتباط ما با دیتای اصلی فایل در حافظه است.

برای فایل های 3 و 4 نمودار زیر را خواهیم داشت:



لینک بین فایل 3 و 4 از نوع سافت است و به فایل لینک شده و نه داده اصلی، و این یعنی اگر فایل شما جابجا شود و یا تغییر نام بدهد اصلا مهم نیست و Link شما همیشه درست کار می کند زیرا اصلا ارتباطی به اسم فایل شما ندارد و به inode متصل شده است . در اینجا مثلا در صورت تغییر نام فایل 4، لینک بین 3 و 4 شکسته خواهد شد.

سوال 12) لینک سرعت دسترسی به فایل ها را بالا می برد، نیازی نیست برای ویرایش یک فایل خاص وارد پوشه اصلی آن شویم، به راحتی با لینک کردن آن می توان به آن دسترسی پیدا کرد. ایده کلی آن توسط Shortcut در ویندوز پیاده سازی شده، با توجه به جواب سوال 3 مشخصا Hard Link بر Soft Link برتری دارد پس دلیل استفاده از لینک نرم چیست؟ از معایب استفاده از Hard Link این است که نمی توان از آن در بین چند فایل سیستم یا پارتیشن استفاده کرد ، با توجه به اینکه در هر فایل سیستم ساختار inode ها متفاوت است و شماره گذاری ها نیز تفاوت دارد Hard Link ها فقط در یک فایل سیستم قابل پیاده سازی هستند.