# Семинар ОС (04.05.2022)

sys-open()

Любая API функция приведёт к системному вызову, который переведёт систему из режима пользователя в режим ядра.

Первый параметр PATH\_NAME и это строка.  
strcpy-from-user()  
SYSCALL\_DEFINE3(<name>, …); мы этому syscall в поле <name> передадим open. В итоге будет вызван ksys\_open(filename, flags, mode); //filename <- user space

do\_sys\_open(AT\_FDCWD, filename, flags, mode);

struct open\_how

do\_sys\_openat2(dfd, filename, &how);

struct [file](https://elixir.bootlin.com/linux/latest/ident/file) \*f = [**do\_filp\_open**](https://elixir.bootlin.com/linux/latest/ident/do_filp_open)(dfd, tmp, &[op](https://elixir.bootlin.com/linux/latest/ident/op)); // dfd – дескриптор директории которая содержит открываемый файл, tmp – путь к файлу, op – флаги, а возвращает struct file

При этом в функции [**do\_filp\_open**](https://elixir.bootlin.com/linux/latest/ident/do_filp_open) три раза вызывается path\_openat:

в теле функции объявляется тип struct nameidata  
{struct path path;  
struct qstr last;  
struct path root;  
struct inode \*inode;  
unsigned int flags; …}  
Система – одно связано с другим

1. Быстрая проверка – быстрый проход по пути, при этом допускается пропуск…
2. Если быстрый проход не дал результата, то происходит обычный проход с проверками
3. Если и этот проход не дал результата, то выполняется медленный проход, в результате которого также происходит обновление устаревшей информации в кеше dentry

while (!(error = [**link\_path\_walk**](https://elixir.bootlin.com/linux/latest/ident/link_path_walk)(s, nd)) &&

(error = [**do\_last**](https://elixir.bootlin.com/linux/latest/ident/do_last)(nd, [**file**](https://elixir.bootlin.com/linux/latest/ident/file), [**op**](https://elixir.bootlin.com/linux/latest/ident/op))) > 0) {

nd->flags &= ~([**LOOKUP\_OPEN**](https://elixir.bootlin.com/linux/latest/ident/LOOKUP_OPEN)|[**LOOKUP\_CREATE**](https://elixir.bootlin.com/linux/latest/ident/LOOKUP_CREATE)|[**LOOKUP\_EXCL**](https://elixir.bootlin.com/linux/latest/ident/LOOKUP_EXCL));

s = [**trailing\_symlink**](https://elixir.bootlin.com/linux/latest/ident/trailing_symlink)(nd);

}

При этом обновляется struct nameidata

Если заглянуть в [**link\_path\_walk**](https://elixir.bootlin.com/linux/latest/ident/link_path_walk)(s, nd)) то увидими функции … которые кешируют dentry:

После того как мы дошли до конца файла, вызывается функция do\_last – выполняет множество проверок

error = [**lookup\_open**](https://elixir.bootlin.com/linux/latest/ident/lookup_open)(nd, &[**path**](https://elixir.bootlin.com/linux/latest/ident/path), [**file**](https://elixir.bootlin.com/linux/latest/ident/file), [**op**](https://elixir.bootlin.com/linux/latest/ident/op), got\_write); //самая важная строка. … а в случае отсутсвия inod’а если указан флаг O\_CREAT выполняет соотв. действия чтобы файл создать.

error = [**inode\_permission**](https://elixir.bootlin.com/linux/latest/ident/inode_permission)(**[dir](https://elixir.bootlin.com/linux/latest/ident/dir)**->**[dentry](https://elixir.bootlin.com/linux/latest/ident/dentry)**->**[d\_inode](https://elixir.bootlin.com/linux/latest/ident/d_inode)**, [**MAY\_WRITE**](https://elixir.bootlin.com/linux/latest/ident/MAY_WRITE) | [**MAY\_EXEC**](https://elixir.bootlin.com/linux/latest/ident/MAY_EXEC));

if (error)

[**return**](https://elixir.bootlin.com/linux/latest/ident/return) error;

[**return**](https://elixir.bootlin.com/linux/latest/ident/return) [**security\_inode\_create**](https://elixir.bootlin.com/linux/latest/ident/security_inode_create)(**[dir](https://elixir.bootlin.com/linux/latest/ident/dir)**->**[dentry](https://elixir.bootlin.com/linux/latest/ident/dentry)**->**[d\_inode](https://elixir.bootlin.com/linux/latest/ident/d_inode)**, [**dentry**](https://elixir.bootlin.com/linux/latest/ident/dentry), mode);

## ЛР7 VFS

Что нужно сделать в этой ЛР: