

## ANOTAÇÕES

- baixou linux kernel 4.8
- pasta fs/minix
- copiou pasta minix para “pasta pessoal”
- terminal comandos de

**<https://www.ibm.com/developerworks/library/l-linux-filesystem/index.html>**

```
$ dd if=/dev/zero of=file.img bs=1k count=10000
```

```
$ losetup /dev/loop0 file.img
```

```
$ mkfs.minix -c /dev/loop0 10000
```

to be continued.....

pretenciosos garotos querem:

- compilar modulo novo com printk. **ok**
- tirar o minix.ko e subir o novo. **ok**
- executar ações que utilizem esse módulo
- ver se printk aparecem no log **ok**

**por hoje eh soh**

só salvando esses links aqui (ignorar):

<https://minix1.woodhull.com/faq/mxfromlx.html>

24/10

Para aparecer o *Volume 10MB*:

```
> losetup /dev/loop0 file.img
```

Desmontar/remover partição (apenas quando não utilizando):

```
> umount /media/gruposob/disk
```

```
> rmmod minix.ko
```

Lembrete: Ao clicar em *Volume 10MB* ele faz um mount sozinho, então é impossível removê-lo com o *rmmod*. É necessário fechá-lo antes.

Vamos no *inode.c* e no seu *init\_minix\_fs* colocamos um *printk* para teste de depuração.

No *Makefile* modificamos para `"(...) := minix_1.o"`.

```
> make
```

```
> insmod minix_1.ko
```

```
> dmesg
```

Nosso *printk* apareceu. Great Success. Parabéns. Muito bem, Paulinho.

Criamos um arquivo *.txt* para teste, para ver se realmente ele está sendo utilizado:

```
> cd /media/gruposob/disk
```

```
> nano teste.txt
```

```
> rmmod minix_1
```

Está sendo utilizado, como prevíamos. Confirmando nossa teoria. Removemos este arquivo *teste*.

Vamos em *dir.c* em *minix\_make\_empty* colocamos um *printk* para depuração.

Testamos:

```
> umount /media/gruposob/disk
```

```
> rmmod minix_1.ko
```

```
> make
```

```
> insmod minix_1.ko
```

```
> mount /media/gruposob/disk (OU SIMPLEMENTE ABRIR Volume 10MB)
```

```
> mkdir pasta1
```

```
> dmesg
```

Nosso *printk* apareceu. Great Success. Parabéns. Muito bem, Paulinho. Então entra na função *minix\_make\_empty* quando é criado algo no diretório.

Removemos a pasta criada:

```
> rmdir pasta1
```

Criamos um arquivo para teste:

```
> nano teste.txt
```

```
> dmesg
```

Nosso printk apareceu. Great Success. Parabéns. Muito bem, Paulinho. Então entra na função *minix\_make\_empty* quando é criado algo no diretório. Agora ainda mais.

Sempre que criamos pastas ou arquivos entra nessa função.

Vamos em *bitmap.c* em *minix\_new\_block* colocamos um printk para depuração.

Testamos:

```
> umount /media/gruposob/disk
```

```
> rmmod minix_1.ko
```

```
> make
```

```
> insmod minix_1.ko
```

```
> mount /media/gruposob/disk (OU SIMPLEMENTE ABRIR Volume 10MB)
```

Criamos uma pasta no diretório.

```
> dmesg
```

Nosso printk apareceu. Great Success. Parabéns. Muito bem, Paulinho.

Criamos um arquivo.txt na pasta.

Cada vez que criamos, escrevemos no arquivo e salvamos ou deletamos, um printk aparece.

Vamos em *namei.c* em *minix\_mkdir* e vemos que é ela quem chama as outras funções como:

*minix\_new\_inode*

*minix\_set\_inode*

*minix\_inc\_link\_count*

*minix\_make\_empty*

Teste com strace

Dentro do /media/gruposob/disk:

> strace -c mkdir teste

% time	seconds	usecs/call	calls	errors	syscall
0.00	0.000000	0	7		read
0.00	0.000000	0	8		open
0.00	0.000000	0	10		close
0.00	0.000000	0	8		fstat
0.00	0.000000	0	19		mmap
0.00	0.000000	0	12		mprotect
0.00	0.000000	0	1		munmap
0.00	0.000000	0	3		brk
0.00	0.000000	0	2		rt_sigaction
0.00	0.000000	0	1		rt_sigprocmask
0.00	0.000000	0	7		access
0.00	0.000000	0	1		execve
0.00	0.000000	0	1		mkdir
0.00	0.000000	0	1		getrlimit
0.00	0.000000	0	2		statfs
0.00	0.000000	0	1		arch_prctl
0.00	0.000000	0	1		set_tid_address
0.00	0.000000	0	1		set_robust_list
100.00	0.000000		86	9 total	

> strace -e open mkdir teste2

```
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
open("/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
open("/lib/x86_64-linux-gnu/libpcre.so.3", O_RDONLY|O_CLOEXEC) = 3
open("/lib/x86_64-linux-gnu/libdl.so.2", O_RDONLY|O_CLOEXEC) = 3
open("/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3
open("/proc/filesystems", O_RDONLY) = 3
open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
+++ exited with 0 +++
```