



File System

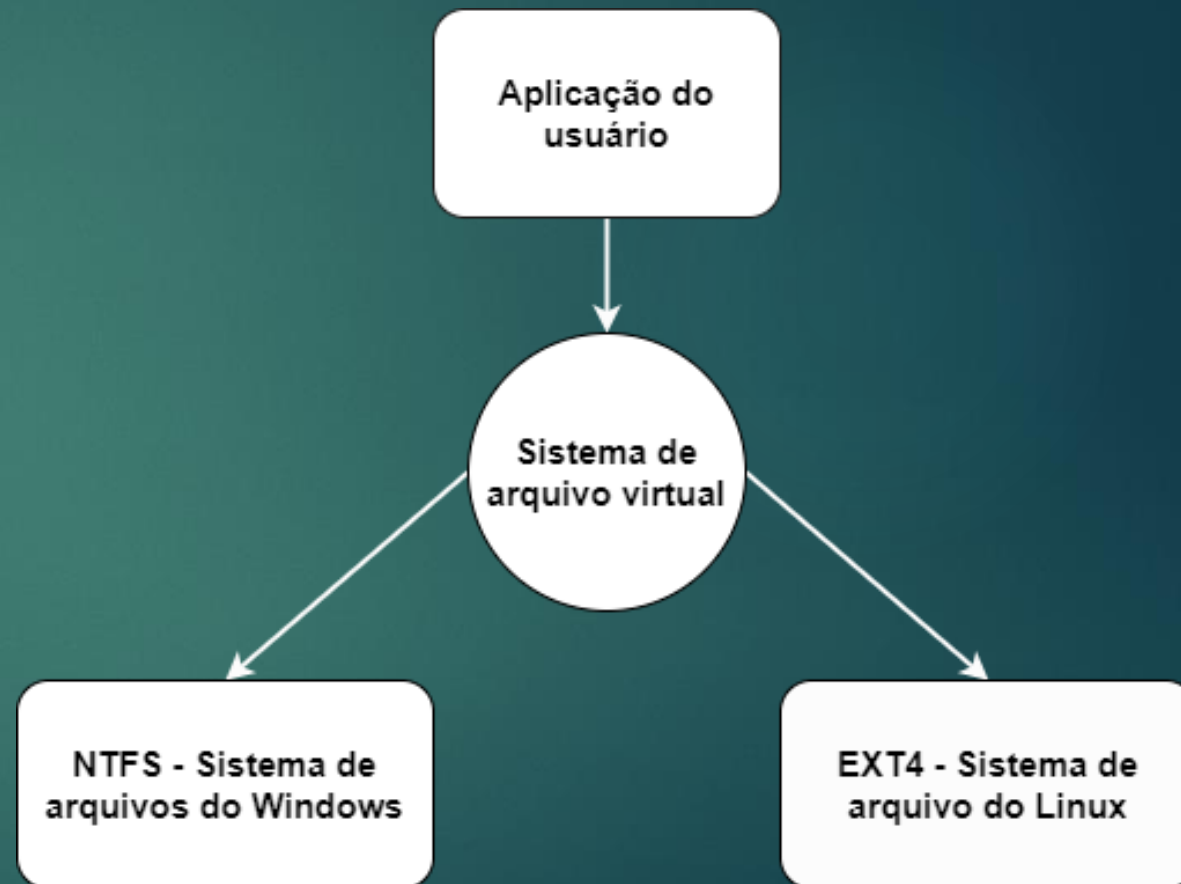
OZÉIAS SOUZA E PEDRO VINÍCIUS

O que é um sistema de arquivos?

- ▶ Segundo Robert Love em seu livro Desenvolvimento do Kernel Linux um sistema de arquivos é o armazenamento de dados de forma hierárquica aderente a uma estrutura específica.
- ▶ Esse conceito abrange banco de dados, sistemas de arquivos virtuais, entre vários outros e até o Git se abstrairmos um pouco.

Aonde podemos utilizar?

- ▶ A função de um sistema de arquivos virtual é permitir que aplicações acessem diferentes formas de sistemas de arquivos concretos de forma uniforme.
- ▶ Por exemplo, em sistemas de dual boot utiliza-se um sistema de arquivos virtual para ler um arquivo em ambos sistemas de forma uniforme.

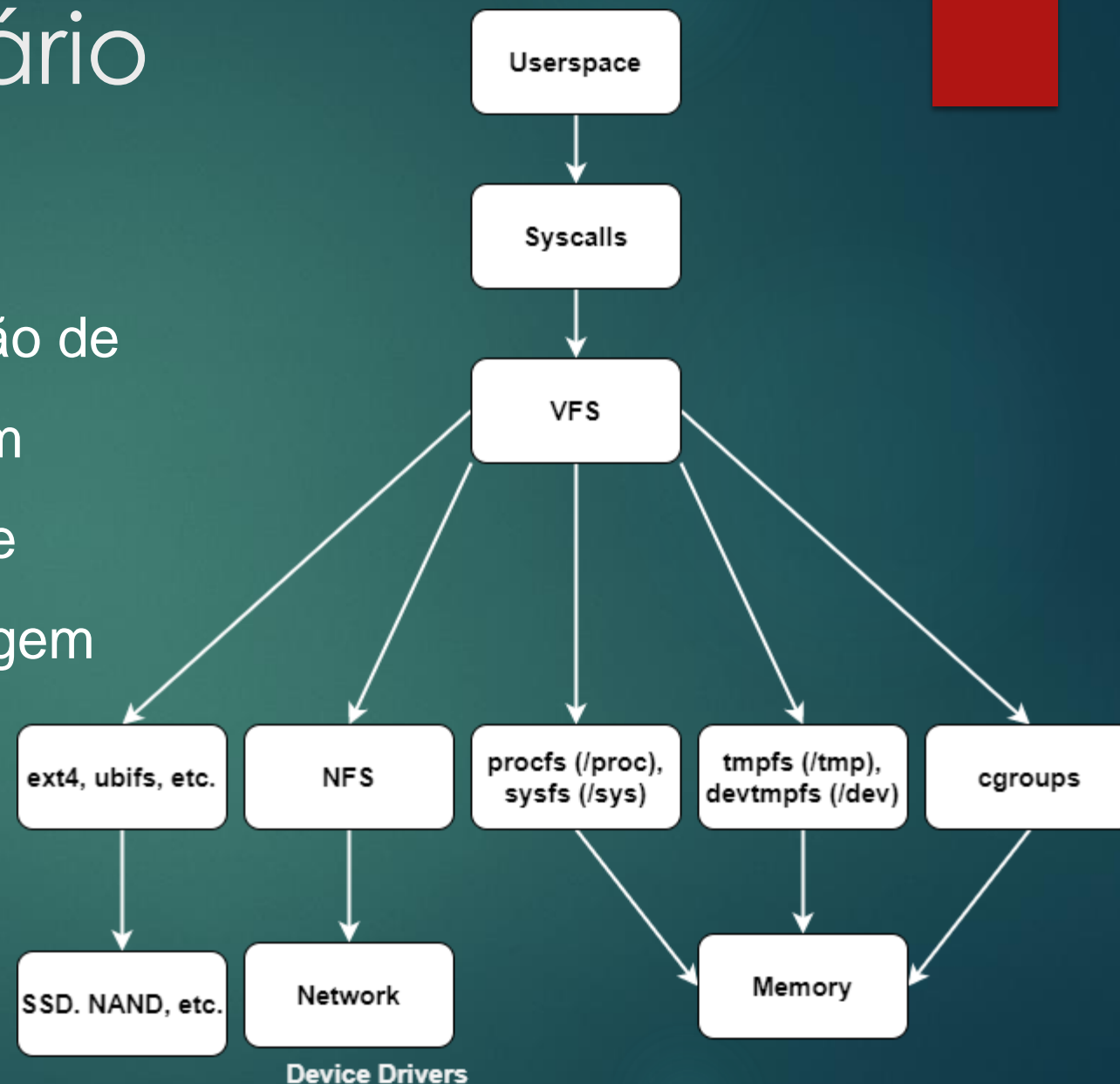


Mas é realmente importante?

- ▶ Essa virtualização permite ao usuário, e ao sistema operacional, mover arquivos entre sistemas e até abstrações diferentes sem se preocupar com o formato dos dados.
- ▶ Um Syscall pode copiar um arquivo com um método `read()` de um sistema de arquivos e depois usar o método `write()` de outro sistema de arquivos para realizar o output dos dados.
- ▶ Sistemas de arquivos virtuais são tão importantes que fazem parte do próprio boot de um sistema.

Interação do Usuário

- O diagrama de demonstração de como o usuário interage com diversos sistemas virtuais de arquivos e como eles interagem entre si.



Divisão de Disco

É o primeiro ou mais alto diretório em uma hierarquia, principalmente utilizados nos sistema de arquivos do Unix e de sistemas Unix-like.

São estruturas de dados que armazenam as propriedades de um arquivo

Aqui temos toda a informação de quanto espaço livre temos, isso pode ser armazenado como mapa de bits ou uma lista de ponteiros.



É utilizado para inicializar o computador e no final dele fica a tabela de partição onde ficam todos os endereços de cada partição.

É aonde contém os parâmetros para a inicialização do sistema de arquivos

Registrando um sistema de arquivo

- ▶ É a estrutura de dados básica que descreve um tipo de sistema de arquivos para o kernel
- ▶ Adiciona o sistema de arquivos passado a lista de sistemas de arquivos do qual o kernel está ciente para as montagens e outras chamadas de sistema.

```
static struct file_system_type fs_fs_type = {  
    .owner          = THIS_MODULE,  
    .name           = "fs",  
    .mount           = fs_get_sb,  
    .kill_sb        = kill_litter_super,  
};
```

```
static int __init init_fs_fs(void)  
{  
    return register_filesystem(&fs_fs_type);  
}
```


Registrando um sistema de arquivo

- ▶ A função `fs_make_inode` com o superbloco e o parâmetro `S_IFDIR` irá retorna um inode que descreve um diretório.
- ▶ Este inode de diretório deverá ser colocado em uma estrutura `Dentry` para que o VFS localize-o.
- ▶ Agora possuindo um diretório raiz inicializado no superbloco, podemos realizar operações predeterminadas

```
root = fs_make_inode(sb, S_IFDIR | 0755);  
if (!root)  
    goto out;  
root->i_op = &simple_dir_inode_operations;  
root->i_fop = &simple_dir_operations;  
root_dentry = d_alloc_root(root);  
if (!root_dentry)  
    goto out_input;  
sb->s_root = root_dentry;
```

```
static struct file_operations fs_file_ops = {  
    .open = fs_open,  
    .read = fs_read_file,  
    .write = fs_write_file,  
};
```


Montando o sistema de arquivos

- ▶ Criar dispositivo de loopback para montar o file system:

```
# dd if=/dev/zero of=rep bs=1k count=4
```

- ▶ Criar diretório para montar file system:

```
# mkdir -p mnt
```

- ▶ Montar o file system:

```
# mount -t lwnfs -o loop rep mnt
```

