

# Relatório - EP 4

## Sistemas Operacionais

10 de dezembro de 2014

- Victor Sanches Portella - N° USP: 7991152
- Vinícius Jorge Vendramini - N° USP: 7991103

### Parte 2 - Verificação de Arquivos

A parte 2 foi implementada com o uso de uma syscall nova, a *lsr*. Para chamá-la, basta passar o caminho para o arquivo (absoluto ou relativo) em uma string. Ela faz exatamente o que foi pedido: lista todos os processos que estão usando este arquivo, lista os blocos (vini: diretos e indiretos??) em que o arquivo está contido e imprime uma mensagem de erro.

### Parte 1 - Implementação de arquivos imediatos

A parte de implementar arquivos imediatos não foi completa, mas a estrutura geral está presente. A maior parte do código relacionado a isso está nos arquivos `open.c` e `read.c`, apesar de existirem outros arquivos pouco modificados também.

Quando se cria um arquivo, ele é criado inicialmente como imediato. Sempre que vamos fazer uma escrita, escrevemos no próprio iNode - a não ser que isso faça o arquivo ultrapassar 32 bytes. Nesse caso, copiamos o que está no iNode para um bloco separado (transformando então o arquivo em um arquivo normal) e então escrevemos o que precisávamos no arquivo normalmente. Para a leitura, checamos se o arquivo é imediato ou não. Se for, lemos do iNode, se não, dos blocos.

Para saber se um arquivo é ou não imediato, optamos por adicionar uma flag `I_IMMEDIATE` análoga a outras flags do sistema, como a `I_REGULAR`.

O valor da flag foi escolhido de modo a não interferir com o resto do sistema e, com isso, acabamos por não gastar nenhum espaço a mais no iNode.

Além disso, criamos também uma flag `O_CREAT` que usamos para criar arquivos imediatos na mão ao fazer uma chamada direta para a syscall `open` (declarada no arquivo `fcntl.h`) que ficaria (por exemplo) da forma `open("path/to/file", 01024, 0666)`, onde 01024 é o valor da flag em si e 0666 são as permissões do arquivo. (vini: talvez seja bom deixar um teste com uma chamada para essa syscall).

## Observações

Existe(m) arquivo(s) teste criado(s) durante o desenvolvimento que podem ser acessados na pasta `/usr/src/EP4Testes`. Um deles chama a lsr no arquivo `teste.c`, presente na mesma pasta. Para ver a função é necessário que o arquivo esteja aberto, o que pode ser feito (por exemplo) abrindo ele no vi ou no emacs em outra aba do Minix. O outro arquivo serve para testar diferentes operações com arquivos imediatos.