

01 - Na estrutura de controle do arquivo (mode no Unix) tem um campo que é o número de links. Todas as vezes que é criado um link de diretório para um arquivo, esse campo é incrementado e, quando esse arquivo é apagado de um diretório, esse campo é decrementado.

Quando o número de links está zerado, o SO fica sabendo que o arquivo não está sendo referenciado por nenhum diretório, portanto, o arquivo pode ser, de fato, apagado do disco. ✓

03 - a) São cópias de uma entrada do sistema de arquivos. As duas entradas contêm nomes diferentes, mas apontam para o mesmo local físico do disco (inode).

Elas compartilham, além do mesmo conteúdo, as mesmas permissões. Se o arquivo verdadeiro for apagado, o hard link continua apontando para o mesmo local físico sendo, portanto, acessível da mesma forma.

O arquivo e o hard link que aponta para ele devem obrigatoriamente estar localizados no mesmo sistema de arquivos, já que o hard link aponta para um endereço físico (inode) e não se pode garantir que estes endereços sejam únicos em vários sistemas de arquivos, ou seja, em volumes diferentes. ✓

pag 15

S02-PT- Luiz Henrique Jucá da Cruz

Não recomendo
/ mente no
raiz

03. b) O Link Simbólico é um arquivo que foi criado no Diretório raiz que tem como conteúdo o caminho do arquivo que se deseja acessar em outro

Diretório que está em outro volume. Ele é um arquivo especial assim como o Diretório. Todo o arquivo simbólico é ~~simb~~ simbolizado com a flag *. O SO trata esse arquivo de forma diferente.

Se apagar o arquivo original, o Link Simbólico fica "preso no ar". Não é igual ao link que continua funcionando enquanto tiver link simbolizado no HOSX.

Então, se tentar usar o arquivo, dará erro de arquivo não encontrado. É isso é igual ao link normal, a diferença é quando se apaga. ✓

Q2- A Tabela de Página do Processo é grande demais e fica armazenada na memória sendo consultada pelo HW HW para descobrir o endereço real de uma variável ou instrução. Para isso, há 2 ações a memória:

1) Consulta a tabela de páginas;

2) Acesso ao endereço.

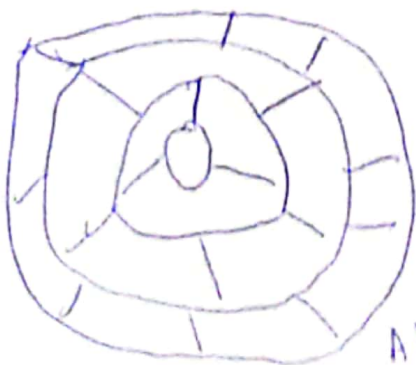
Isso torna a paginação 2x mais lenta que um HW sem paginação. Para otimizar esse processo, a TLB foi criada para ser consultada pelo HW HW antes de acessar a Tabela de Páginas.

A TLB guarda um resumo da Tabela de Páginas. Se você estiver usando uma página cuja conversão esteja na TLB, vai ser mais rápido do que acessar a Tabela de Página do processo. **Em início na primeira**

Existe outro componente de HW que tem TLB "alternativa". Essa TLB tem o campo num.-processo. Por exemplo, a TLB estava com os dados do Processo 1. Mas ele foi bloqueado e o SO escalonou o Processo 2. Nesse caso, não precisamos limpar a TLB nem invalidar o conteúdo dos campos da TLB por causa do campo num.-processo que diz de qual tabela que veio o dado que está na TLB. Quando P2 iniciar de fato, pega uma TLB vezia. ~~TLB~~
Caso P2 seja bloqueado, o SO volta a executar P1. ✓

04- ~~(falta)~~ falamos ter um setor interno que tem um tamanho pequeno e, por outro lado, tem os setores mais brilhantes mais externos que tem um tamanho maior. Diante desse contexto, temos uma diferença grande de comprimentos. E aí temos um problema: ~~(a grande é diferente)~~ considerando o setor A maior que o setor B, então tem muito mais superfície para guardar informação no setor A do que no setor B. Só que, na construção histórica de discos, a mesma capacidade de bytes é guardada para os dois casos, mesmo que o setor B seja menor que o A. Nesse caso, existe um desperdício de capacidade de armazenamento dos setores que ficaram em trilhas mais externas mas, nos dois casos, se guardava 1 Kb tanto no A quanto no B.

Só que hoje em dia não é assim, porém, para que isso fosse possível, teve que mudar um elemento importante, que foi o conversor analógico-digital. Ou seja, a famosa fatia de pizza acabou morrendo, mas, desse fato, não tem mais desperdício de capacidade de armazenamento e hoje os setores tem aproximadamente o mesmo tamanho.



A tal "fatia de pizza", como era antigamente

pag 515

PF-SO2- Luiz Henrique Soares da Cruz

OS - Isso tem a ver com o Working Set. A ideia do SO é de manter o US estabilizado. Caso tenhamos uma qtd de páginas físicas maior que o US, o SO não permite que um novo processo seja executado.

X