

MAC0422 - Sistemas Operacionais
Bacharelado em Ciência da Computação
IME/USP – SEGUNDO SEMESTRE DE 2011
Quarto Exercício-Programa

1 Symbolic Links

1.1 Introdução

É comum na maioria dos SOs, a existência de “symbolic links” ou ligações simbólicas para arquivos comuns e diretórios. Eles também são chamados de “aliases” ou “soft links” para arquivos ou diretórios já existentes. Na maioria das variações Linux/UNIX/Mac OS X, o comando `ln -s source_file target_file` cria um arquivo chamado `target_file` no diretório atual que contém como conteúdo o nome `source_file`. Note que para o SO distinguir `target_file` com um soft link e não como um arquivo-texto normal, ele é atribuído um tipo diferente de arquivo.

O comportamento esperado de um link simbólico é servir de *proxy* para o arquivo ou diretório destino. Assim, as syscalls onde são passados ou retornados descritores de arquivos, como `open()`, `read()`, `write()`, `seek()`, `close()`, etc devem operar no arquivo/diretório destino, e não no arquivo do link simbólico. Em contraste, algumas operações, como por exemplo `unlink()`, irá operar no arquivo do link simbólico e não no arquivo/diretório apontado por ele.

A sua tarefa é implementar suporte à “soft links” no xv6. Como você irá perceber em breve, já existem “hard links” para *inodes* dentro do xv6. Estes podem ser criados através do comando `ln`, mas você alterará `ln.c` para comportar links simbólicos também.

1.2 Aquecimento

Antes de mais nada, leia o capítulo 5 da documentação do xv6. Você também deverá se familiarizar com os seguintes arquivos do kernel: `fs.h`, `fs.c`, `stat.h`, `sysfile.c` e `ln.c`. Entenda o funcionamento e a relação entre as diferentes camadas do sistema de arquivos. Você irá notar uma similaridade muito grande com o ext2 e ext3 visto em aula. Como sugestão, escreva um programa simples que crie um arquivo-texto, escreva algo, feche-o e leia-o após nova abertura. Use o `gdb` para acompanhar as chamadas ao sistema.

2 Implementação

2.1 Parte I (30%)

Para criar *symlinks* no xv6, você precisará definir um novo tipo de arquivo. Além dos tipos `T_FILE`, `T_DIR`, e `T_DEV`, deverá criar o tipo `T_SYMLINK`. Ache o arquivo que contém tais definições e faça esta adição. Você também modificará o arquivo `ln.c` para aceitar mais um argumento, o flag `-s` para identificar a criação de um link simbólico. Isto significa que a syscall `sys_link()` que é invocada dentro do programa `ln.c` também deverá ser alterada em seu número de argumentos. Execute tais modificações.

2.2 Parte II (70%)

Nesta parte você efetivamente construirá a base para o funcionamento dos links simbólicos dentro do SO. Para tanto, o primeiro passo é criá-los, e o segundo passo é fazer que o SO trate-os adequadamente tendo em vista as operações em arquivos existentes.

2.2.1 Criação (30%)

Para criar um link simbólico, você deverá primeiramente validar se o arquivo a ser apontado existe, caso contrário a syscall deve falhar e retornar erro. Você deve falhar também a syscall quando já existir um arquivo com mesmo nome do link pretendido. Se tudo funcionar até este instante, você deverá fazer as seguintes tarefas:

1. Criar uma entrada no diretório atual com o nome do link simbólico (`target_file`) fornecido pela syscall.
2. Alocar um bloco no dispositivo e gravar neste bloco o link `source_file`. A syscall deve falhar se o link não couber em um único bloco.
3. Alocar um *inode*, preenchê-lo adequadamente, indicando o novo tipo de arquivo (`T_SYMLINK`, e o bloco que o link foi gravado.

A partir deste momento, o seu link simbólico existe no sistema de arquivos. Se você executar um `ls`, você deverá vê-lo como um arquivo de tipo distinto dentro do sistema.

2.2.2 Manipulação de arquivos (20%)

Para que as funções de manipulação de arquivos funcionem normalmente seguindo os links simbólicos, a syscall `sys_open()` deve ser modificada para dar a opção de seguir ou não o link simbólico. Por default, a abertura do arquivo-link deverá abrir o link. No entanto, você deverá criar um flag chamado `O_NOFOLLOW` a ser passado na syscall, caso você esteja interessado no arquivo-link e não no arquivo apontado pelo link. O resto das syscalls de manipulação de arquivos deverão funcionar independente das suas mudanças.

É possível que existam links simbólicos para links simbólicos. Nesta configuração, podem aparecer ciclos onde um link aponta para um link que aponta para o link anterior. Neste caso, você deverá detectar o número de indireções (número de links seguidos) e falhar caso este número seja maior que 10 níveis.

É possível que você tenha que alterar outras funções dentro do kernel, dependendo do seu design. Seja conciso, tente adicionar ou mudar o mínimo possível.

2.2.3 Visualização (20%)

Para ter certeza que você realmente criou o link corretamente, você alterará o comando `ls.c` para lidar com os softlinks. Em particular, você deverá imprimir as informações normais do arquivo-link, como nome, tipo, inode e tamanho, mas também o conteúdo do link a la Linux. Veja abaixo um exemplo em resposta ao comando “ls” apropriadamente modificado:

```
ola.c 0 3 133
link 3 7 6 -> ola.c
ls 0 10 512
```

Aqui você provavelmente verá a utilidade do flag `O_NOFOLLOW` prescrito anteriormente.

Instruções de entrega

Teste as suas alterações criando links tanto para arquivos quanto diretórios. Experimente funções básicas como leitura e gravação em cima dos symlinks criados. Verifique que removendo o arquivo-link remove ele e somente ele (e não o arquivo apontado pelo link).

Você deve entregar todos os arquivos fonte que você modificou no xv6, e quaisquer outros arquivos adicionais que você criou. Lembre-se, se você modificou o Makefile, inclua-o também, senão não

poderemos testá-lo. Crie um arquivo compactado `.zip`, `.tar.gz`, `.tgz` ou `.bz2` contendo todos os arquivos pertencentes ao seu EP. Não inclua código executável, somente código fonte (`.h`, `.c`, `.S`) e arquivos textos contendo quaisquer descrições suas. Identifique o seu EP de acordo com o exemplo abaixo:

`EP4-Marcel-Henrique.zip`

Não esqueça de adicionar um arquivo `LEIAME.TXT` contendo os nomes dos participantes da equipe (máximo 2 pessoas), números USP e demais instruções necessárias para compilar o seu programa e entender o seu EP. Você é responsável pela clareza destas instruções. Entregue o seu EP eletronicamente no site da disciplina até a data final. Envie somente um EP para a sua equipe.