Departamento de Computação

SISTEMAS OPERACIONAIS 1
21270 A

Departamento de Computação
Prof. Kelen Cristiane Teixeira Vivaldini

Departamento de Computação

Lab. Sistema de Arquivos

Lab. Sistema de Arquivos

pwd –	Exibe o diretório atual
cd	Navegando entre diretórios
ls	Listar arquivos (-l detalhado ; -a lista arquivos ocultos; -h exibe o tamanho num formato legível; -R lista também os subdiretórios encontrados)
mkdir	Cria um diretório
rmdir	Remove um diretório vazio
cp	Cópia de arquivos e diretórios
mv	Move arquivos e diretórios
rm	Remove arquivos e diretórios
find	Procura arquivos
cat	Exibe o conteúdo de um arquivo
head, tail	Mostra o começo e fim do arquivo
more, less	Visualiza arquivos por páginas
file	Indica o tipo de arquivo

Departamento de Computação

Lab. Sistema de Arquivos

- Arquivos:
  - Nomes;
  - Estrutura;
  - Tipos;
  - Acessos;
  - Atributos;
  - Operações;

Lab. Sistema de Arquivos

Comando Stat

stat serve para apresentar as informações de status de um arquivo ou sistema de arquivos. Ele apresenta uma série de informações sobre o arquivo que você informar como argumento. Dentre as informações estão o *Tamanho*, *Blocos*, *Permissões de Acesso*, *Data e Hora de último acesso*, *Data e Hora de última modificação*, etc.

stat <caminho_do_arquivo_ou_sistema_de_arquivos>.

Lab. Sistema de Arquivos

Comando make

O comando make faz uma "leitura" de um arquivo chamado Makefile

make

Departamento de Computação

Lab. Sistema de Arquivos
Operações em arquivos

- Diferentes sistemas provêm diferentes operações que permitem armazenar e recuperar arquivos;
- Operações mais comuns (*system calls*):
  - `Create`; `Delete`;
  - `Open`; `Close`;
  - `Read`; `Write`; `Append`;
  - `Seek`;
  - `Get attributes`; `Set attributes`;
  - `Rename`;

- Veja struct file operations em
  linux-4.X.Y/include/linux/fs.h

Lab. Sistema de Arquivos

Streams e File Descriptors

File descriptors provide a primitive, low-level interface to input and output operations. [...]
The main advantage of using the stream interface is that the set of functions for performing actual input and output operations (as opposed to control operations) on streams is much richer and more powerful than the corresponding facilities for file descriptors. The file descriptor interface provides only simple functions for transferring blocks of characters, but the stream interface also provides powerful formatted input and output functions (printf and scanf) as well as functions for character- and line-oriented input and output.

https://www.gnu.org/software/libc/manual/html_node/Streams-and-File-Descriptors.html

Lab. Sistema de Arquivos

Streams

```
int fprintf(FILE *stream, const char *format, ...);

int fscanf(FILE *stream, const char *format, ...);

FILE *fopen(const char *path, const char *mode);

int fclose(FILE *stream);
```

Veja os exemplos fscanf.c e fscanf2.c

Sistema de Arquivos
Diretórios – Operações

- `Create; Delete;`
- `Opendir; Closedir;`
- `Readdir;`
- `Rename;`
- `Link` (um arquivo pode aparecer em mais de um diretório);
- `Unlink;`

- Veja struct inode operations em linux-4.X.Y/include/linux/fs.h
- Veja linux-4.X.Y/include/linux/dirent.h
- Veja http://pubs.opengroup.org/onlinepubs/7908799/xsh/dirent.h.html
- Veja o código dir.c

Lab. Sistema de Arquivos

LINK SIMBÓLICO E HARDLINK

- O link é um mecanismo que faz referência a outro arquivo ou diretório em outra localização. Os links são arquivos especiais e podem ser identificados com um "l" quando executado o comando: "ls -la".

Lab. Sistema de Arquivos

TIPO SIMBÓLICO

O link tipo simbólico é um arquivo especial de disco do tipo link, que tem como conteúdo o caminho para chegar até o arquivo alvo.

Características:

- Pode-se fazer links simbólicos em arquivos e diretórios;
- O link simbólico e o arquivo alvo não precisam estar na mesma partição de disco;
- Se o link simbólico for apagado/movido. Somente o link será apagado/movido;
- Qualquer usuário pode criar/desfazer um link simbólico (respeitando as permissões).

Lab. Sistema de Arquivos

TIPO HARDLINK

No link tipo *hardlink*, o link é apontado para o mesmo inode do arquivo alvo, sendo assim, os dois arquivos serão o mesmo.

Características:

- Não é possível fazer um *hardlink* para um diretório;
- Somente é possível fazer *hardlink* em arquivos que estejam em uma mesma partição de disco;
- Se o *hardlink* for apagado/movido, você estará apagando/movendo o arquivo alvo;
- Somente o usuário root pode criar/desfazer *hardlinks*.

Lab. Sistema de Arquivos

CRIANDO LINKS

O comando ln é utilizado para criar links entre dois arquivos ou para um diretório.

Sintaxe:

ln [OPÇÕES]... [-T] ALVO NOME_LINK (1a forma)
ln [OPÇÕES]... ALVO (2a forma)
ln [OPÇÕES]... ALVO... DIRETÓRIO (3a forma)
ln [OPÇÕES]... -t DIRETÓRIO ALVO... (4a forma)

Explicando:

- ALVO: Diretório ou arquivo de onde será feito o link;
- NOME_LINK: Nome do link que será criado;

OPÇÕES:

- -s

  Cria um link simbólico.

- -v

  Modo verbose.

Lab. Sistema de Arquivos

Criando um hardlink chamado “*.txt" apontando para o arquivo “*.txt":

Crie um arquivo texto qualquer exemplo arquivo.txt

ln arquivo.txt outroarquivo.txt

Note que o arquivo os dois arquivos possuem o mesmo Inode e o mesmo Device. Portanto, as modificações realizadas em qualquer um dos arquivos refletirá no outro.

Use o comando **stat**

Lab. Sistema de Arquivos
Diretórios – Caminho (*path name*)

- O método hierárquico requer métodos pelos quais os arquivos são acessados;
- Dois métodos diferentes:
  - Caminho absoluto (*absolute path name*);
  - Caminho relativo (*relative path name*);

Lab. Sistema de Arquivos
Diretórios – Caminho (*path name*)

- Crie um diretório contendo

  um arquivo regular
  um hardlink para este arquivo
  um link com caminho relativo para este arquivo
  um link com caminho absoluto para este relativo

- Copie este diretório com cp –r
- Como ficou o resultado?

Lab. Sistema de Arquivos
pipe()

int pipe (int FILEDES[2])

The 'pipe' function creates a pipe and puts the file descriptors for the reading and writing ends of the pipe (respectively) into 'FILEDES[0]' and 'FILEDES[1]'.

Veja o código: mypipe.c

http://man7.org/linux/man-pages/man2/pipe.2.html

Lab. Sistema de Arquivos
pipe()

int dup2(int oldfd, int newfd);

dup2 makes newfd be the copy of oldfd, closing newfd first if necessary. After successful return of dup or dup2, the old and new descriptors may be used interchangeably.

Veja o código: mypipe2.c

http://man7.org/linux/man-pages/man2/pipe.2.html

Lab. Sistema de Arquivos
popen()

FILE *popen(const char *command, const char *type);

int pclose(FILE *stream);

> *The popen() function opens a process by creating a pipe, forking, and invoking the shell. Since a pipe is by definition unidirectional, the type argument may specify only reading or writing, not both; the resulting stream is correspondingly read-only or write-only.*

Veja o código: mypopen.c e mypopen2.c