Operações em meta-dados

Nesta seção são apresentadas as principais funções para manipular meta-dados do sistema de arquivos, como: renomear arquivos, remover arquivos, consultar/alterar atributos e permissões.

Renomear e remover

```
#include <stdio.h>
int rename (const char *oldname, const char *newname)
```

Renomeia um arquivo do nome antigo `oldname` para o nome novo `newname`. Ambos os nome devem pertencer ao mesmo sistema de arquivos (mas não precisam estar no mesmo diretório). Retorna -1 em caso de erro.

```
#include <unistd.h>
int unlink (const char *filename)   /* versão Posix */

#include <stdio.h>
int remove (const char *filename)   /* versão ISO */
```

Removem o arquivo indicado.

Consultar atributos

As funções abaixo servem para consultar os atributos de um arquivo. Todas elas retornam os atributos em um buffer de tipo `struct stat`.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
int stat (const char *filename, struct stat *buf)
```

Consulta os atributos de um arquivo indicado por `filename`.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
int fstat (int filedes, struct stat *buf)
```

Idem, para o arquivo indicado pelo descritor filedes.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
int lstat (const char *filename, struct stat *buf)
```

Idem, mas não segue links simbólicos (se `filename` for um link, retorna os atributos do próprio link).

Todas as funções acima retornam um buffer do tipo `struct stat` (definido em `sys/stat.h`) que contém os seguintes campos:

- `mode_t st_mode`: modo do arquivo (tipo e permissões).
- `ino_t st_ino`: posição (*i-node*) do arquivo, número único no mesmo dispositivo.
- `dev_t st_dev`: identifica o dispostivo onde está o arquivo.
- `nlink_t st_nlink`: número de links físicos (*hard links*) para o arquivo.
- `uid_t st_uid`: *User ID* do proprietário.
- `gid_t st_gid`: *Group ID* do grupo proprietário.
- `off_t st_size`: tamanho do arquivo.
- `time_t st_atime`: data de último acesso.
- `unsigned long int st_atime_usec`: parte fracionária da data de último acesso.
- `time_t st_mtime`: data de última modificação de conteúdo.
- `unsigned long int st_mtime_usec`: parte fracionária da data de última modificação de conteúdo.
- `time_t st_ctime`: data de última modificação de atributos.
- `unsigned long int st_ctime_usec`: parte fracionária da data de última modificação de atributos.
- `blkcnt_t st_blocks`: espaço ocupado em disco, em blocos de 512 bytes.
- `unsigned int st_blksize`: o tamanho ideal de bloco para leituras/escritas nesse arquivo, em bytes.

As seguintes macros permitem verificar o tipo do arquivo, com base no campo `st_mode` da estrutura `stat` do arquivo. Elas retornam não-zero se o arquivo corresponder ao tipo especificado:

- `int S_ISDIR (mode_t st_mode)`: diretório.
- `int S_ISCHR (mode_t st_mode)`: dispositivo orientado a caracteres (terminal, porta serial).
- `int S_ISBLK (mode_t st_mode)`: dispositivo orientado a blocos (disco).
- `int S_ISREG (mode_t st_mode)`: arquivo normal.
- `int S_ISFIFO (mode_t st_mode)`: FIFO ou pipe.
- `int S_ISLNK (mode_t st_mode)`: link simbólico.
- `int S_ISSOCK (mode_t st_mode)`: socket.

As constantes a seguir, definidas em `sys/stat.h`, são usadas para testar as permissões de acesso a um arquivo:

- S_IRUSR: `r-- — —`
- S_IWUSR: `-w- — —`
- S_IXUSR: `--x — —`
- S_IRWXU: equivale a (`S_IRUSR | S_IWUSR | S_IXUSR`)
- S_IRGRP: `— r-- —`
- S_IWGRP: `— -w- —`
- S_IXGRP: `— --x —`
- S_IRWXG: equivale a (`S_IRGRP | S_IWGRP | S_IXGRP`)
- S_IROTH: `— — r--`
- S_IWOTH: `— — -w-`
- S_IXOTH: `— — --x`
- S_IRWXO: equivale a (`S_IROTH | S_IWOTH | S_IXOTH`)
- S_ISUID: `--s — —` (*set uid bit*)
- S_ISGID: `— --s —` (*set gid bit*)
- S_ISVTX: `— — --t` (*sticky bit*)

Para testar as permissões de um arquivo, basta realizar um AND entre a constante definindo a permissão a testar e o campo `st_mode` do `struct stat` referente ao arquivo:

```
if (buf.st_mode & S_IRUSR)
   printf ("File owner can read this file\n") ;
```

Obs: usar permissões em octal (0644) nos programas não é portável e prejudica sua legibilidade. Sempre que possível, devem ser usadas as permissões simbólicas.

Alterar permissões

```
#include <sys/types.h>
#include <sys/stat.h>
int chmod (const char *filename, mode_t mode)
```

Ajusta as permissões do arquivo `filename` para `mode` (que consiste em um OR binário das permissões desejadas). Se `filename` for um link simbólico, ajusta as permissões do arquivo apontado pelo link.

```
#include <sys/types.h>
#include <sys/stat.h>
int fchmod (int filedes, int mode)
```

Idem, sobre o arquivo aberto indicado pelo descritor `filedes`.

Links físicos e simbólicos

```
#include <unistd.h>
int link (const char *oldname, const char *newname)
```

Cria um link físico (*hard link*) para `oldname` usando o nome `newname`.

```
#include <unistd.h>
int symlink (const char *oldname, const char *newname)
```

Cria um link simbólico (*symbolic link*) para `oldname` usando o nome `newname`.

```
#include <unistd.h>
int readlink (const char *filename, char *buffer, size_t size)
```

Obtém o nome apontado pelo link simbólico `filename`. O resultado é copiado em buffer e a função retorna o número de caracteres copiado.

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=pua:operacoes_em_meta-dados

Last update: **2008/08/13 20:15**