



# *Programação de sistema*

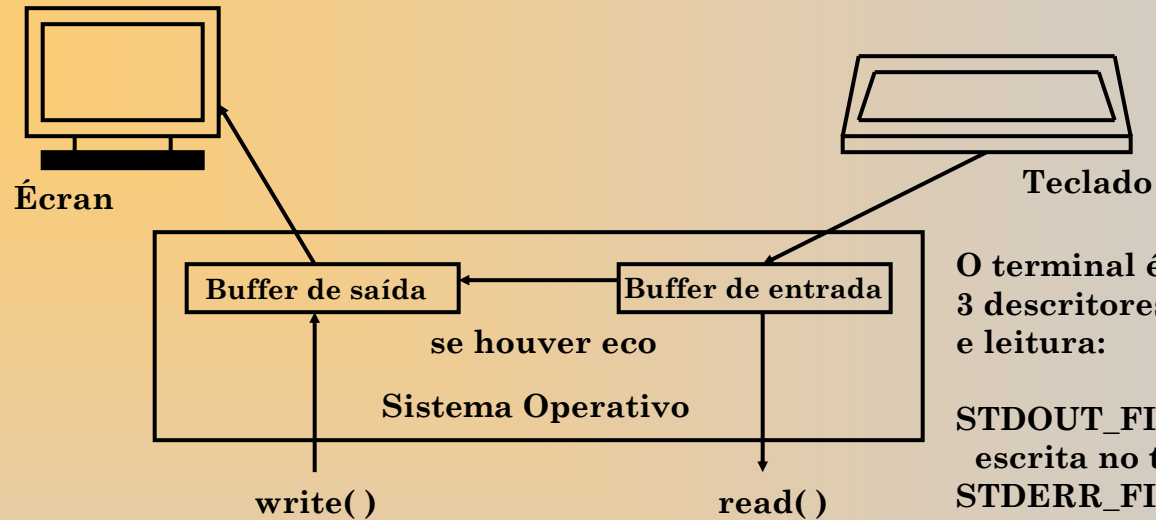
---

Terminal

Acesso a ficheiros

Acesso a diretórios

# Terminal de texto



O terminal é representado por 3 descritores usados para escrita e leitura:

**STDOUT\_FILENO**  
escrita no terminal (resultados)  
**STDERR\_FILENO**  
escrita no terminal (mensagens de erro)  
**STDIN\_FILENO**  
leitura do teclado

```
#include <unistd.h>
```

```
ssize_t read(int fildes, void *buffer, size_t nbytes);  
ssize_t write(int fildes, void *buffer, size_t nbytes);
```

Retorno: -1 erro  
0 fim (leitura)  
>0 número de bytes lidos



# Configuração do terminal

## Configuração da consola

```
#include <termios.h>
```

```
int tcgetattr(int filedes, struct termios *termpptr);  
int tcsetattr(int filedes, int opt, struct termios *termpptr);
```

```
struct termios {  
    tcflag_t c_iflag;    // input  
    tcflag_t c_oflag;    // output  
    tcflag_t c_cflag;    // control  
    tcflag_t c_lflag;    // local  
    cc_t c_cc[NCCS];     // special character definitions  
};
```

TCSANOW  
TCSADRAIN  
TCSAFLUSH

imediatamente  
após saída vazia  
esvazia entrada

Lê e guarda configuração atual

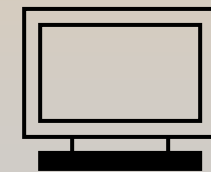
tcgetattr( ... )

Modifica configuração  
struct termios

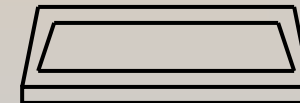
Escreve nova configuração  
tcsetattr( ... )

...

Repõe configuração primitiva  
tcsetattr( ... )



Écran



Teclado

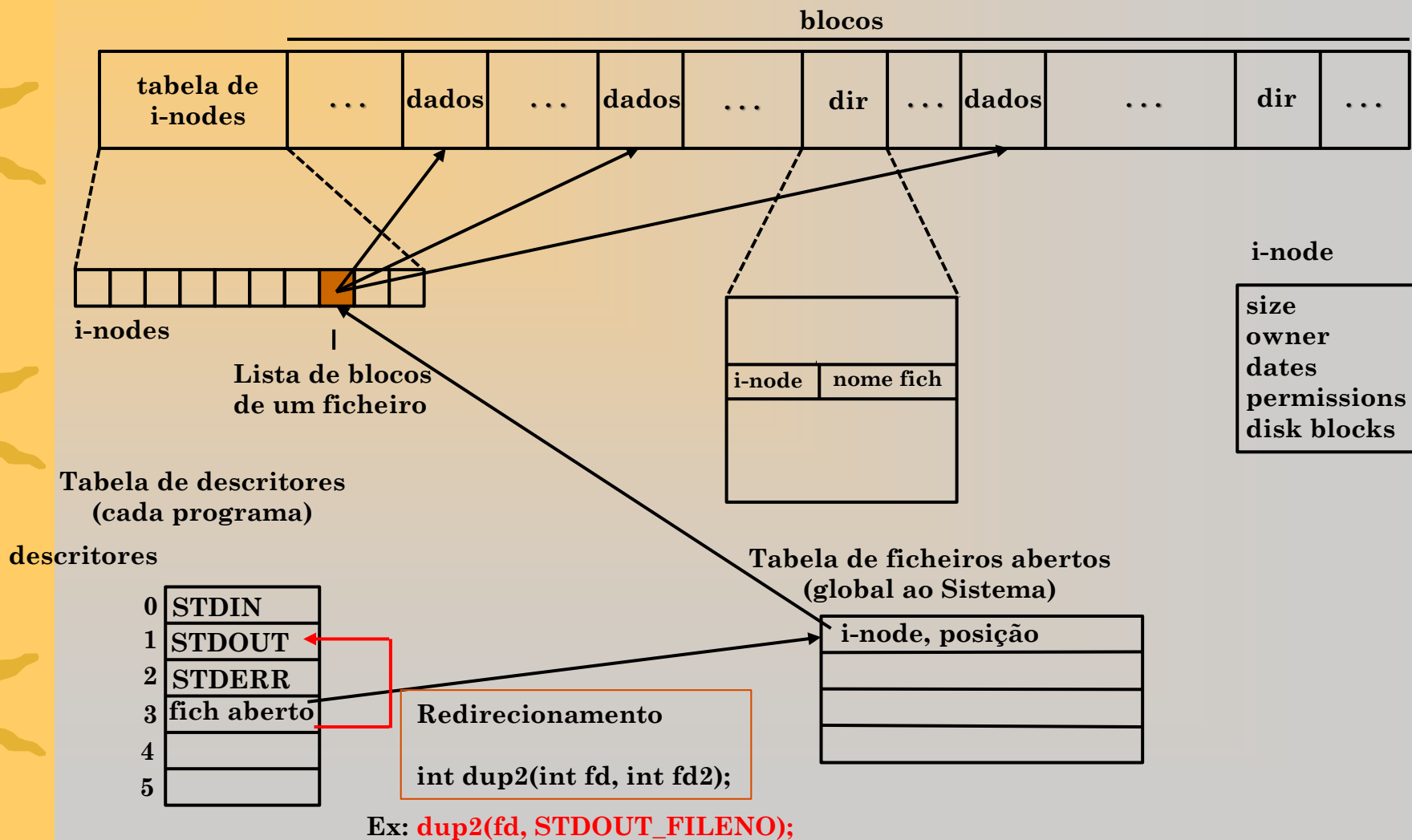
Buffer de saída

Buffer de entrada

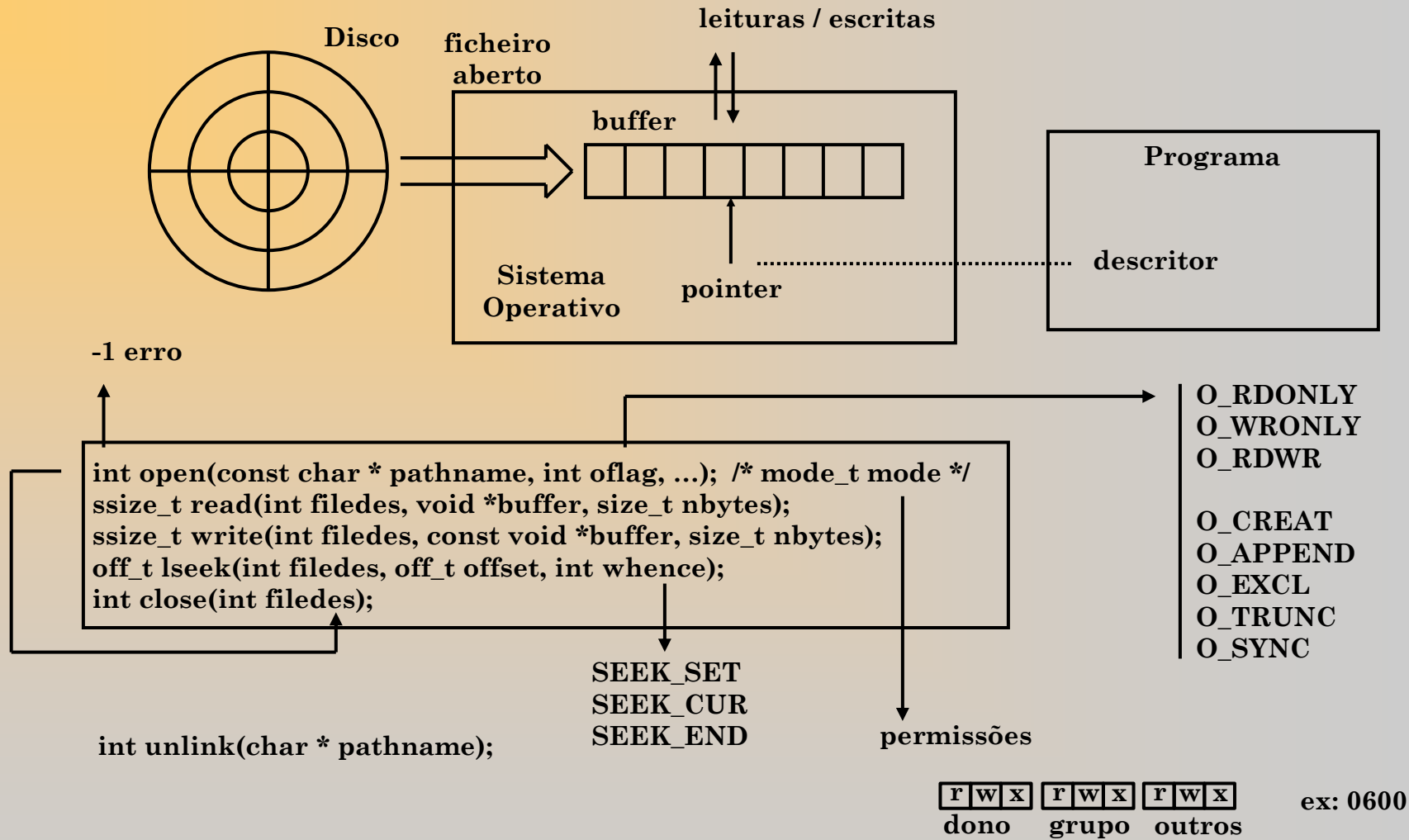


# File system

Disco – sequência de blocos; cada bloco tem em geral 512 bytes



# Serviços de acesso a ficheiros





# Informações

**Informação sobre uma entrada do sistema de ficheiros:  
(ficheiros, diretórios, links, ...)**

```
#include <sys/stat.h>
```

```
int stat(const char *pathname, struct stat *buf);  
int fstat(int filedes, struct stat *buf);  
int lstat(const char *pathname, struct stat *buf);
```

```
struct stat {  
    mode_t mode;           // permissões e tipo de entrada  
    ino_t st_ino;          // número do i-node  
    off_t st_size;         // tamanho em bytes  
    long st_blocks;       // número de blocos  
    ... (datas)           // ...  
    ...  
}
```

**Teste do tipo de entrada descrita numa struct stat**

```
S_ISREG(mode)    // ficheiro regular ?      (true / false)  
S_ISDIR(mode)    // diretório ?  
S_ISFIFO(mode)   // é um FIFO ?  
S_ISLNK(mode)    // é um link ?
```



# Acesso a diretórios

## Diretório corrente:

```
char *getcwd(char *buffer, size_t size);  
int chdir(const char *pathname);
```

## Criar e apagar diretórios:

```
int mkdir(const char *pathname, mode_t mode);  
int rmdir(const char *pathname);
```

## Nomes dos ficheiros contidos num diretório:

```
DIR * opendir(const char* pathname);
```

```
struct dirent *readdir(DIR *dp);  
void rewinddir(DIR *dp);  
  
int closedir(DIR *dp);
```

```
struct dirent {  
    ino_t d_ino;  
    char d_name[NAME_MAX+1];  
}
```