

Sistemas Operativos

Redireccionamento de Descritores de Ficheiros

Grupo de Sistemas Distribuídos
Universidade do Minho

1 Objectivos

Familiarizar-se e utilizar as chamadas ao sistema relativas à duplicação de descritores de ficheiros e a sua aplicação no redireccionamento, mais especificamente dos descritores *standard* de entrada, saída e de erros.

2 Chamadas ao sistema

```
#include <unistd.h>          /* chamadas ao sistema: defs e decls essenciais */

int dup(int fd);
int dup2(int fd1, int fd2);
```

3 Exercícios propostos

1. Escreva um programa que redireccione o descritor associado ao seu *standard input* para o ficheiro `/etc/passwd`, e o *standard output* e *error* respectivamente para `saida.txt` e `erros.txt`. Experimente ler uma linha a partir do seu *standard input* e repeti-la nos descritores de *output* e *error* usando directamente `read()` e `write()`. Experimente também realizar as mesmas operações através de funções da biblioteca de C, por exemplo, `fgets()`, `printf()` ou `fprintf()`. Em todas as situações, repare no conteúdo dos ficheiros criados. Note que depois de cada redireccionamento, os descritores dos ficheiros abertos deixam de ser necessários, e como tal é boa prática fechá-los de imediato.
2. Modifique o programa anterior de modo a que, depois de realizar os redireccionamentos, seja criado um novo processo que realize operações de leitura e escrita. Experimente ainda usar o comando `cat` no processo filho para copiar a entrada para a saída, tal como fez no Guião 1. Observe o conteúdo dos ficheiros. Repare que o processo filho “nasce” com as mesmas associações de descritores de ficheiros do processo pai.
3. Modifique novamente o programa inicial de modo a que seja executado o comando `wc`, sem argumentos, depois do redireccionamento dos descritores de entrada e saída. Note que, mais uma vez, as associações – e redireccionamentos – de descritores de ficheiros são preservados pela primitiva `exec()`.
4. Escreva um programa `redir` que permita executar um comando, opcionalmente redireccionando a entrada e/ou a saída. O programa poderá ser invocado, com:

```
redir [-i fich_entrada] [-o fich_saida] comando arg1 arg2 ...
```

4 Exercícios Adicionais

1. Acrescente ao interpretador de comandos proposto no guião anterior a possibilidade de redireccionar as entradas, saídas e erros dos comandos por ele executados. Considere os operadores de redireccionamento `<`, `>`, `>>`, `2>` e `2>>`.