

# Sistemas Operativos

## Pipes Anónimos

Grupo de Sistemas Distribuídos  
Universidade do Minho

### 1 Objectivos

Familiarizar-se e utilizar as chamadas ao sistema relativas à comunicação entre processos por pipes anónimos e o seu uso conjugado com a técnica de redireccionamento de entradas e saídas.

### 2 Chamadas ao sistema

```
#include <unistd.h>          /* chamadas ao sistema: defs e decls essenciais */  
  
int pipe(pd[2]);
```

### 3 Exercícios propostos

1. Escreva um programa que crie um pipe anónimo e de seguida crie um processo filho. Relembre que as associações dos descritores de ficheiros são preservadas na criação do processo filho. Experimente o pai enviar uma linha de texto através do descritor de escrita do pipe, e o filho receber uma linha a partir do respectivo descritor de leitura. Note que a informação foi transferida do pai para o filho. Experimente de seguida provocar um atraso antes do pai enviar a linha de texto (p. ex., `sleep(5)`). Note agora que a leitura do filho bloqueia enquanto o pai não realizar a operação de escrita no pipe. Experimente agora inverter os papeis de modo à informação ser transmitida do filho para o pai.
2. Modifique o programa anterior de modo à leitura do pipe ser realizada enquanto não for detectada a situação de *end of file* no descritor respectivo. Repare que esta situação acontece apenas quando nenhum processo – neste caso, pai e filho – têm aberto o descritor de escrita do pipe.
3. Escreva um programa que execute o comando `wc` num processo filho. O processo pai deve enviar ao filho através de um pipe anónimo uma sequência de linhas de texto introduzidas pelo utilizador no seu *standard input*. Recorra à técnica de redireccionamento estudada no guião anterior de modo a associar o *standard input* do processo filho ao descritor de leitura do pipe anónimo criado pelo pai. Recorde a necessidade de fechar o(s) descritor(es) de escrita no pipe de modo a verificar-se a situação de *end of file*.
4. Escreva um programa que emule o funcionamento do interpretador de comandos na execução encadeada de `ls /etc | wc -l`.
5. Escreva um programa que emule o funcionamento do interpretador de comandos na execução encadeada de `grep -v ^# /etc/passwd | cut -f7 -d: | uniq | wc -l`.

## 4 Exercícios Adicionais

1. Acrescente ao interpretador de comandos proposto nos guiões anteriores a possibilidade de encadear as entradas e saídas de programas a executar através de pipes anónimos (operador `|`).
2. Implemente uma função semelhante ao `popen()`, mas que em vez de retornar um valor do tipo `FILE *`, retorne o descritor apropriado do descritor de um pipe anónimo. Nota: se desejar pode converter um descritor num endereço de uma estrutura `FILE` usando para o efeito a função `fdopen()`. Reflecta sobre a necessidade da função `pclose()` da biblioteca de C e como poderia proceder para suportar essa funcionalidade.