

Exercício 5

Sistemas Operativos 2014-15
LEIC-A / LEIC-T / LETI
IST

Resumo

Este exercício consiste em fazer uma versão integrada dos programas LEITOR e ESCRITOR com um monitor, em que o utilizador pode ir indicando quais os ficheiros a verificar.

1 Leitor Paralelos com Monitor Integrado

Pretende-se desenvolver uma nova variante a partir do LEITOR com sincronização resultante do Exercício 3. O LEITOR terá um fio de execução (*thread*) principal que cria vários fios de execução (*threads* filhas) cuja tarefa é verificar a coerência dos ficheiros (ver Fig.1). No entanto, nesta versão, os filhos não verificam apenas um único ficheiro seleccionado aleatoriamente. Pelo contrário, ficam em ciclo infinito a receber nomes de ficheiros do pai que depois verificam imprimindo o resultado no `stdout`. Quando um filho acaba de verificar um ficheiro, bloqueia-se até receber um novo nome de ficheiro enviado do pai. A comunicação entre o pai e os filhos é feita por um *buffer* que é partilhado por todos os fios de execução: o pai coloca nomes no *buffer* partilhado que são posteriormente retirados pelos filhos. Os nomes que o pai passa aos filhos não são gerados aleatoriamente. Pelo contrário, são lidos do `stdin` (usando a chamada de sistema *read*) de forma a que o utilizador do programa possa escolher quais os ficheiros a validar. Este LEITOR pode ser usado a par com qualquer dos ESCRITORES desenvolvidos nas fases anteriores do projecto.

Os alunos devem usar os mecanismos de sincronização associados aos fios de execução, para controlar o acesso ao *buffer* partilhado.

2 ESCRITOR Paralelo

O ESCRITOR paralelo usado nesta fase do projecto deverá ser desenvolvido a partir do resultado da terceira parte do projecto. Esse ESCRITOR paralelo deverá ser alterado de forma a realizar o seu paralelismo com fios de execução (*threads*) e não com processos e a suportar as seguintes funcionalidades:

- O ESCRITOR não termina ao fim de um número limitado de iterações, mas continua a escrever ficheiros em ciclo infinito.
- O ESCRITOR processa *signals* que lhe sejam enviados por outros processos da seguinte forma:

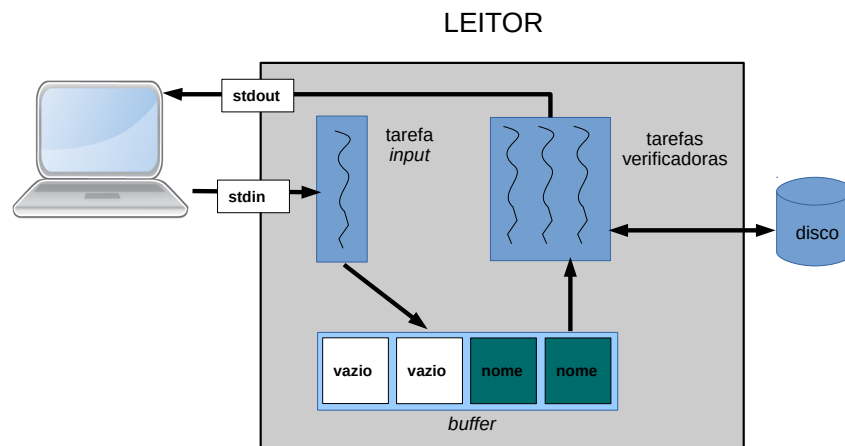


Figura 1: O leitor paralelo com monitor integrado.

- Se receber um **SIGUSR1**, **inverte a utilização de trincos de ficheiro** (*file locks*) exclusivos pelo ESCRITOR. Se estiver/não estiver a usar trincos deixa de os usar/passa a usá-los, quando recebe o *signal*.
- Se receber um **SIGUSR2**, **inverte a escrita de erros** pelo ESCRITOR. Se o ESCRITOR estiver/não estiver a escrever ficheiros sem erros, passa a/deixa de incluir erros nos ficheiros, quando recebe o *signal*. **Os erros introduzidos deverão ser uma letra errada, a cada linha alternada.**
- Se receber um **SIGSTOP**, **completa a escrita de ficheiros que tiver em curso** e seguidamente **termina a sua execução**.

3 Monitor Independente

Os alunos devem desenvolver um **programa monitor**, que **gere de forma integrada o ESCRITOR paralelo e o LEITOR paralelo** (ver Fig. 2). O monitor independente deve ter as seguintes funcionalidades:

1. Deve **lançar um processo filho que executa o ESCRITOR paralelo** descrito acima.
2. Deve **lançar um outro filho que executa o LEITOR paralelo**. A **ligação** entre o monitor e o LEITOR será feita por um **pipe** ligado ao stdin do LEITOR.
3. Deve **aceitar nomes de ficheiros a verificar do teclado**, enviando estes nomes para o LEITOR através do **pipe** criado para esse efeito.
4. Deve terminar a sua execução e a dos LEITOR e ESCRITOR paralelos, quando recebe um comando especial pelo teclado. **Quando o nome do ficheiro a verificar for "sair", o monitor inicia a terminação da aplicação.** O LEITOR é terminado **fechando o pipe** que

liga o monitor ao LEITOR. O ESCRITOR é terminado com o envio de um signal SIGSTP. O LEITOR e ESCRITOR devem terminar de processar o ficheiro que estejam a processar quando recebem a instrução de terminação. O monitor deve esperar pela terminação do LEITOR e do ESCRITOR antes de terminar a sua própria execução.

5. Para além dos nomes dos ficheiros a serem processados pelo LEITOR e do comando "sair", o monitor recebe dois outros comandos destinados ao ESCRITOR:

- *il* - Este comando desencadeia o envio de um SIGUSR1 ao ESCRITOR.
- *ie* - Este comando desencadeia o envio de um SIGUSR2 ao ESCRITOR.

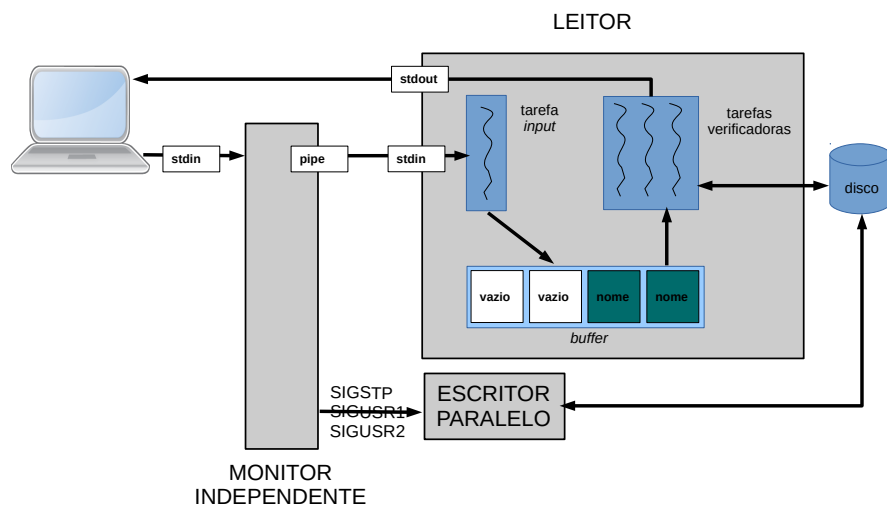


Figura 2: O sistema completo: leitor paralelo, escritor paralelo e monitor independente.

4 Submissão

Os alunos devem submeter um ficheiro no formato zip com o código fonte, uma Makefile, e os executáveis através do sistema Fénix. O exercício deve obrigatoriamente compilar e executar nos computadores dos laboratórios.

A submissão pode ser feita até às 23:59 do dia 5 de Dezembro.

5 Cooperação entre Grupos

Os alunos são livres de discutir com outros colegas soluções alternativas para o exercício. No entanto, *em caso algum*, os alunos podem copiar ou deixar copiar o código do exercício. Caso duas soluções sejam cópias, ambos os grupos reprovarão à disciplina.