

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
MIEIC - 2013/2014
SISTEMAS OPERATIVOS
Trabalhos Práticos

TRABALHO Nº 1

Detetor de palavras em tempo real

Objetivos do trabalho

Proporcionar a familiarização com a programação de sistema, em ambiente Unix/Linux, envolvendo, principalmente, a manipulação de ficheiros e diretórios, o desenvolvimento de aplicações multiprocesso e a utilização de sinais e *pipes* como mecanismos de sincronização/comunicação entre processos.

Especificação do trabalho

Pretende-se desenvolver uma aplicação (**monitor**) que monitorize, em tempo real, um conjunto de ficheiros e alerte sempre que uma determinada palavra seja acrescentada no final de qualquer um desses ficheiros.

A invocação do programa deve seguir a seguinte sintaxe:

```
./monitor tempo palavra file1.txt file2.txt ... fileN.txt
```

Se após a invocação do monitor algum dos ficheiros monitorizados for alterado (da forma referida), deve ser exibida na consola uma mensagem contendo a data e hora do evento, o nome do ficheiro e a linha que contém a palavra, como se ilustra a seguir, para uma situação em que a **palavra** era "palavra" e os ficheiros eram **fileX.txt**, **fileY.txt** e **fileZ.txt**:

```
2014-03-11T23:01:03 - fileX.txt - "Não posso usar esta palavra porque é monitorizada"
```

```
2014-03-11T23:10:12 - fileY.txt - "As palavras são detetadas"
```

```
2014-03-12T00:01:09 - fileZ.txt - "Assim como um palavrao mal escrito"
```

Para detetar novas linhas deve ser usado o utilitário **tail**, com as opções **-f** (aguarda a escrita de novas linhas) e **-n 0** (só exibe novas linhas).

Para verificar se a nova linha contém a palavra pretendida deve ser usado o utilitário **grep**, com a opção **--line-buffered** para garantir que escreve os resultados linha a linha, em vez de usar um *buffer* de grande dimensão.

Quer o **grep** quer o **tail** devem ser lançados através de chamadas independentes ao sistema **exec**. O resultado do **tail** dever ser direcionado para a entrada do **grep** e a saída deste deve ser processada por um processo-filho de **monitor** que, por cada linha recebida do **grep**, deve compor a mensagem a enviar para a consola, no formato acima ilustrado.

A aplicação **monitor** deve continuar a funcionar durante o **tempo** (em segundos) indicado como argumento da linha de comando ou até que todos os ficheiros a monitorizar tenham sido removidos. Deve também garantir que não há processos filhos no estado *zombie*.

Como o processo **tail** continua a executar mesmo que o ficheiro tenha sido removido, deve haver um processo, filho de **monitor**, dedicado a verificar (de 5 em 5 segundos) se algum dos ficheiros monitorizados foi removido e enviar um sinal SIGUSR1 ao(s) processo(s) associados especificamente à monitorização desse ficheiro. Sugestão: investigue a possibilidade de utilização da chamada **setpggrp()** para criar um grupo de processos e da chamada **kill()** para enviar um sinal a um grupo de processos.

Quando o tempo se esgotar ou todos os ficheiros sob monitorização tiverem sido removidos, o processo pai deve também terminar, mas não sem antes terminar todos os outros processos filhos que possa haver, nomeadamente o processo, acima referido, que monitoriza a remoção de ficheiros.

Entrega do trabalho

- Data limite para a entrega do trabalho: 11/Abr/2014, às 23:55h.
- Oportunamente serão publicadas algumas regras para submissão do trabalho na página de "Sistemas Operativos", no Moodle da Universidade do Porto.