Sistemas Operativos
"Echelinho"

Universidade do Minho
Licenciatura em Engenharia Informática

A60993 - Luís Fonseca
A60994 - André Santos
A61075 - Ricardo Branco

6 de Junho de 2012

Conteúdo

Resumo

Este projeto foi realizado no âmbito da Unidade Curricular de Sistemas Operativos, do 2º Semestre, do 2º Ano, da Licenciatura de Engenharia Informática, da Universidade do Minho.

Tem-se como objectivo a implementação de um serviço de monitorização de ficheiros log, na linguagem C. Este serviço deverá reportar em tempo real a ocorrência de *Strings* em tais ficheiros.
Foram aplicados em prática os conceitos:

- *Forks*
- *Pipes*
- *Sinais*
- *Paralelismo*

De forma externa ao que foi ensinado na Unidade Curricular foram usadas algumas das *System Calls* da API do *inotify*(Ferramenta que permite a monitorização de ficheiros).

1 Descrição

Este projeto é constituido por:

1. Um programa *monitor*
2. Um ou mais *daemon*

1.1 Monitor

O programa *monitor* funciona como um interpretador de comandos que pode ser uitilizado para administrar o serviço e reportar as ocorrências observadas.
Este permite fazer as seguintes operações:

- adicionar ou remover ficheiros log a monitorizar
- reportar a ocorrencia de strings não monitorizadas que foram acrescentadas no ultimo minuto
- mostrar estado do serviço

1.2 Daemon

O *daemon* é responsavel por acompanhar em tempo real a evolução dos ficheiros a monitorizar.
Este é criado sempre um ficheiro é monitorizado e sempre que são feitas alterações envia notificações ao *monitor* atraves de um *pipe*.
Cada *daemon* está preparado para reagir a modificação de ficheiros monitorizados, mas ao implementar o *inotify*, este passou a estar preparado para reagir aos seguintes acontecimentos:

- Leitura do ficheiro
- Alteração dos atributos do ficheiro
- Abertura de um ficheiro
- Eliminação de um ficheiro
- Alteração do directório do ficheiro

2 Execução do programa

O *monitor* já como foi dito na página 1 tem como uma das funcionalidades um interpretador de comandos. Os comandos que este aceita são os seguintes

```
NEWDAEMON <nome do ficheiro> <string a monitorizar>
ADDSTRING <nome do ficheiro> <string a monitorizar>
REMOVESTRING <nome do ficheiro> <string monitorizada>
ESTADO
CLOSEMONITOR
EXIT
```

Ao fazer *NEWDAEMON* é feito um fork. O processo filho fica responsavel por correr o daemon, que por sua vez este cria um *pipe* do *daemon* para o *monitor*, caso este nao exista e cria um ficheiro no guarda as strings monitorizadas

O comando *ADDSTRING* actualiza o ficheiro de string monitorizadas e avisa o daemon através de um Sinal. Caso o ficheiro passado como argumento não esteja a ser monitorizado pelo daemon imprime uma mensagem de erro no *stderr*.

O comando *REMOVESTRING* actualiza o ficheiro de string monitorizadas e avisa o daemon através de um Sinal. Caso o ficheiro passado como argumento não esteja a ser monitorizado pelo daemon ou a *string* não exista no ficheiro de *strings* imprime uma mensagem de erro no *stderr*.

No comando *CLOSEMONITOR* o *monitor* envia um sinal a cada *daemon* avisando que vai encerrar.É criado por sua vez um ficheiro que guarda os *pids* dos *daemon*. Quando o *monitor* for iniciado este vai imprimir os eventos que aconteceram após o último encerramento do *monitor*.

No comando *EXIT* o *monitor* vai percorrer a lista de *pids* dos *daemon* e vai mandar um sinal *SIGKILL* a cada *daemon*.

3 Enquadramento Teórico

Neste projeto foram aplicados em prática alguns conceitos ensinados nas aulas práticas e teoricas da unidade curricular de Sistemas Operativos tais como:

- Forks
- Pipes
- Sinais
- Paralelismo

Forks no sentido em que sempre o ficheiro novo é monitorizado, um *Fork* é criado.
Visto que o *monitor* e o *daemon* são processos independentes é necessário um *Pipe* para haver comunicação entre os processos.
Sinais na medida em que o *monitor* quer interagir com os *daemon*, um sinal é enviado, desta forma em vez de termos uma espera activa temos uma espera passiva.
Por último *Paralelismo*, uma vez que temos vários *daemon* a correr em simultaneo.

4 Conclusão

Todos/ou quase todos os conceitos adquiridos durante o Semestre na Unidade Curricular de Sistemas Operativos foram aplicados durante a realização deste projeto. Desta forma deu para preparar para o teste da mesma Unidade Curricular.
Os objectivos indicados no enunciado foram concluidos com exito, apesar de algumas dificuldades encontradas na sua realização. Conseguimos reconciliar um bom funcionamento do programa com uma interface amigavel.