



Universidade do Minho

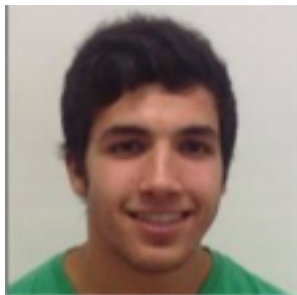
Mestrado Integrado em Engenharia Informática

Sistemas Operativos

Relatório

Trabalho Prático | Backup eficiente

GRUPO 38



André Gonçalves
A75625



Inês Sampaio
A72626



José Bastos
A74696

20 de Maio de 2016

Índice

1 - Introdução	3
2 - Como funciona	4
3 - Makefile	6
4 - Conclusão	7

1.Introdução

No âmbito da unidade curricular de Sistemas Operativos, e com o objetivo principal de fazer aplicar os conhecimentos adquiridos até à data, foi proposto aos alunos o desenvolvimento de um sistema de cópias de segurança (backup), para salvaguardar ficheiros de um utilizador. Esta ferramenta teria, não só que fazer **backup** - criar copia de ficheiros e **restore** - repor o backup no local original, mas também que ser eficiente, minimizando ao máximo o espaço ocupado em disco pelo backup, e apresentar algumas garantias relativamente à privacidade do utilizador.

Ao longo deste relatório, serão descritos e discutidos os pressupostos do problema apresentado, e tudo aquilo que fizemos para o solucionar.

2.Como funciona

A aplicação é constituída por dois programas principais, o programa Servidor sobusrv e o programa Cliente sobucli.

O servidor é iniciado com o comando

```
$ sobusrv
```

Após a sua inicialização, o servidor aguarda instruções do cliente para executar uma operação de cópia ou restore.

Isto acontece porque, aquando da inicialização do programa, há um *fork* para o servidor ficar a correr em segundo plano, sendo que o processo “pai” termina mas o “filho” continua a correr. Neste existe um ciclo infinito que vai fazer com que o servidor corra indefinidamente. Posteriormente é aberto o *pipe* e é lida a informação contida nele.

Usamos uma variável para calcular a quantidade de informação que é lida e, no caso de não ser nula, é criado outro *fork* (com o objetivo de que clientes diferentes tenham acesso ao programa). Dentro deste *fork* estão criados dois *pipes*, um de ida e um de volta do servidor para o cliente. De seguida é enviado um sinal ao cliente para que este envie o ficheiro, se necessário (não é necessário no caso de o ficheiro já existir).

Quando é recebido o ficheiro do cliente, por um *pipe*, o servidor cria os ficheiros respetivos na memória e são eliminados os dois *pipes* auxiliares. O *pipe* principal é fechado e o programa começa um novo ciclo à espera de informação.

O programa Cliente sobucli realiza o backup dos ficheiros que se pretende, restaurando os mesmo através do comando ‘restore’. Também há a possibilidade de eliminar um ficheiro do servidor onde este se encontra fazendo assim a exclusão das entradas que não estejam alinhadas a nenhum ficheiro.

Foi criada uma estrutura Ficheiro que serve como canal de comunicação entre o servidor e o cliente. Esta estrutura contém as informações:

pid - pid que requisitou a operação

operação - array onde fica guardado o comando que se pretende efetuar

nome - nome do ficheiro

sha1sum - é o código que identifica cada um dos ficheiros compactados sendo único para cada ficheiro com conteúdo diferente

```
typedef struct ficheiro{  
    int pid;  
    char operacao[BUFF];  
    char nome[BUFF];  
    char sha1sum[BUFF];  
    char pipe_server[BUFF];  
}*Fich;
```

pipe_server - pipe para comunicar entre o servidor e o cliente

Assim, todas as funções apenas necessitam de trabalhar com esta estrutura, facilitando assim a comunicação do servidor com o cliente e um bom aproveitamento das informações que a mesma contém.

3.Makefile

MAKE

O Comando make tem como função compilar o programa gerando assim os executáveis do servidor e do cliente.

MAKE CLEAN

Com este comando são removidos os executáveis presentes onde foram gerados.

MAKE INSTALL

Este comando corre o comando all. De seguida mete os programas na bin.

MAKE UNINSTALL

Este comando é o inverso do install.

MAKE KILL

Comando que mata os processos criados.

4. Conclusão

Temos de concluir que tivemos que passar por alguma dificuldades que este trabalho nos apresentou, devido a ser um projeto bastante desafiante para o grupo mas que se demonstrou bastante gratificante no final.

Concretizada a oportunidade de colocar em prática todo o conhecimento adquirido ao longo das aulas da unidade curricular de Sistemas Operativos, podemos afirmar que o objetivo principal deste trabalho prático foi alcançado, apesar de ser num contexto diferente e completamente mais complexo.

Toda a experiência adquirida ao longo do desenvolvimento da aplicação de backup, foi fundamental para aprofundar os nossos conhecimentos em linguagem C.