



1 Introdução

A introdução ao projeto foi feita nos enunciados dos anteriores miniprojetos. Por isso a sua leitura prévia é fundamental e deve ser considerada como introdução a este enunciado.

O objetivo geral desta fase é a interação com o sistema de ficheiros e a utilização das funções de tempo. Particularmente nesta fase pretende-se melhorar o ficheiro `makefile` com vista à gestão e execução do `SOinstala`.

2 Objetivos do miniprojeto 4

De uma forma geral deve ser configurado o ficheiro `makefile` utilizando programação em *Shell script*. De seguida são apresentados os objetivos específicos.

2.1 Limpar o projeto

A “limpeza” do projeto consiste na eliminação de todos os ficheiros desnecessários.

Acrescentar ao `makefile` um alvo `clean` que permita fazer a limpeza do projeto. Esse `clean` deve apagar o ficheiro executável, todos os ficheiros objeto e todos os ficheiros temporários (de todas as pastas em que tal faça sentido). Além disso, também se deve apagar o ficheiro `core`¹.

2.2 Criar cópias de segurança (backup)

Quando se implementa um projeto de programação, há muita coisa que pode correr mal: podemos eliminar um ficheiro sem querer, o disco rígido ou outra componente do computador pode falhar num momento crítico, o sistema pode ser infetado por um vírus, ou então o programador pode introduzir *bugs* no programa sem perceber. Para se proteger contra estes eventos inesperados o programador deve criar periodicamente cópias de segurança do seu projeto. No caso de algo correr mal, este tem a possibilidade de recorrer a estes *backups* e assim tentar resolver o problema.

Criar um *shell script* `criar_backup.sh` que crie uma cópia de segurança de todos os ficheiros fonte e de todos os cabeçalhos. Este *backup* deve consistir num único ficheiro comprimido que mantenha a estrutura de diretórios. O nome do ficheiro deve ser:

```
backup_AAAA-MM-DD_HH-MM-SS.tar.gz
```

Sendo AAAA-MM-DD o dia (ano, mês, dia) e HH-MM-SS a hora (hora, minutos, segundos) em que o ficheiro é criado. Apenas o dono do ficheiro deve ter permissão de escrita e leitura do mesmo.

Acrescentar ao `makefile` um alvo `backup` que corra o *script* `criar_backup.sh`.

2.3 Executar e testar o projeto

Escrever um *shell-script* `testar.sh` que:

¹ Em Linux, o ficheiro `core` é criado quando um programa termina de forma abrupta, devido a um *bug* ou a uma violação dos mecanismos de proteção do sistema operativo ou do hardware. Este ficheiro inclui informação que pode permitir perceber qual a causa do erro.

- a) Execute o programa **soinstala** original, passando como primeiro argumento um ficheiro de configuração (que está em `testes/in`). O segundo argumento é um ficheiro de resultado (este argumento é opcional).
- b) Execute o programa **soinstala** desenvolvido pelo grupo, passando como argumento o ficheiro de teste de entrada (que estará em `testes/in`). O segundo argumento é um ficheiro de resultado (este argumento é opcional).
- c) Ordene (`sort`) as linhas de cada ficheiro de resultados (em `testes/out`), gerados em a) e b), e compare (`diff`) os dois ficheiros ordenados (sem os alterar). No final, o programa deve enviar uma mensagem ao utilizador informando-o se os ficheiros de resultados são iguais ou diferentes, após a ordenação.

Acrescentar ao `makefile` um alvo `testar` que corra o *shell-script* `testar.sh`.

3 Entrega

Os ficheiros `makefile`, `criar_backup.sh` e `testar.sh` devem ser entregues até às **20h** do dia **31 de Maio de 2015 (domingo)**, colocando-os no diretório **TRAB4** (maiúsculas) na respetiva área de grupos.

Não serão aceites trabalhos por e-mail nem por qualquer outro meio não definido nesta secção. Se não se verificar algum destes requisitos o trabalho é considerado como não entregue.