



# **Trabalho 1**

## ***Gestão de armazenamento:***

### ***Monitorização do espaço ocupado***

**Professor:**

Nuno Lau ([nunolau@ua.pt](mailto:nunolau@ua.pt))

Guilherme Campos ([guilherme.campos@ua.pt](mailto:guilherme.campos@ua.pt))

**Bruno Meixedo, 113372**  
**Diogo Costa, 112714**

# Índice

<b>Introdução .....</b>	<b>3</b>
<b>Explicação do código.....</b>	<b>4</b>
Definição da variável .....	4
Função calculate_folder_sizes() .....	5
Estrutura getopt .....	6
Cabeçalho .....	7
Bloco condicional .....	7
<b>Resultado dos testes .....</b>	<b>8</b>
Teste de Filtragem por Expressão Regular: .....	8
Teste de Ordenação em Ordem Inversa (-r): .....	8
Teste de Limitação de Saída (-l): .....	9
Teste de Ordenação por Nome (-a): .....	9
Teste de Filtragem por Tamanho (-s): .....	9
Teste de Filtragem por Data (-d): .....	10
Lista de Testes (completos) .....	11
<b>Explicação do código.....</b>	<b>12</b>
Definição de variável .....	12
Estrutura getopt .....	13
Validação de número de argumentos .....	13
Leitura de arquivos .....	14
Declaração de arrays e armazenamento dos dados dos ficheiros .....	14
Iteração nos arrays e cálculos da diferença .....	15
Bloco condicional para ordenação .....	16
<b>Resultado dos testes .....</b>	<b>16</b>
Ficheiros de exemplo: .....	16
<b>Conclusão .....</b>	<b>18</b>
<b>Bibliografia .....</b>	<b>19</b>

## **Introdução**

O primeiro ficheiro, intitulado por `spacecheck.sh`, tem como objetivo fornecer uma visão completa do espaço ocupado pelos arquivos em diretórios específicos e as suas subpastas. Usa expressões regulares, data de modificação e tamanho mínimo como critérios de seleção, o ficheiro também oferece opções para classificar e limitar o número de linhas exibidas. Isso proporciona uma abordagem flexível para analisar e entender a distribuição do espaço em disco.

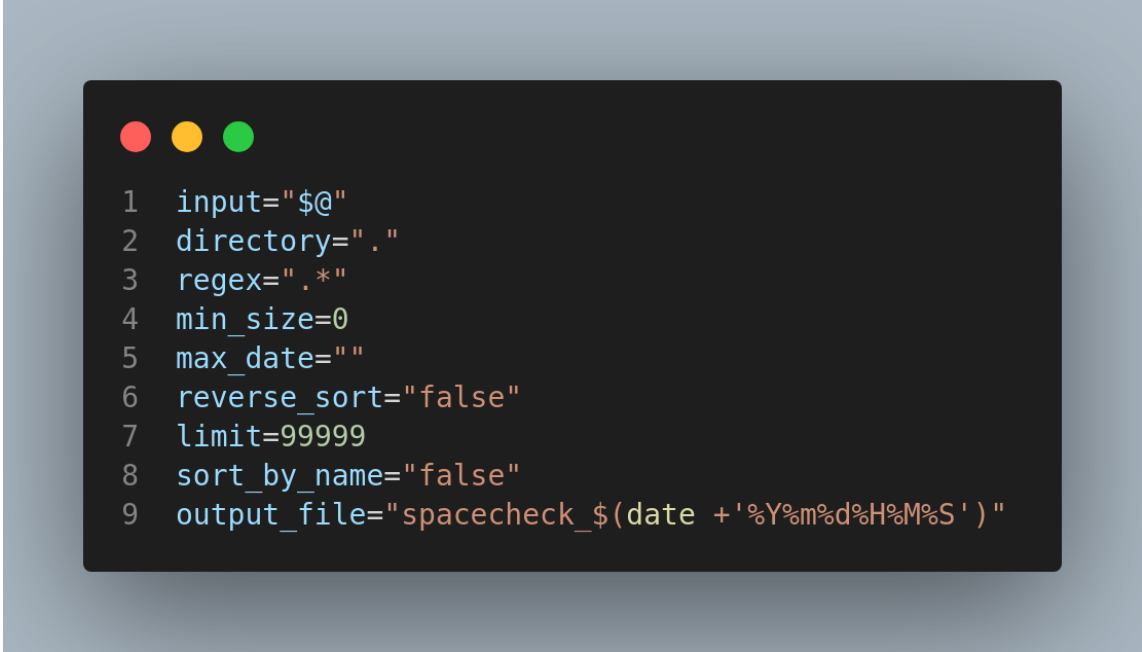
O segundo ficheiro, denominado por `spacerate.sh`, tem a função de comparar as saídas geradas pelo `spacecheck.sh` em dois momentos diferentes. Este script destaca as diferenças no espaço ocupado entre os dois momentos, identificando novos arquivos, pastas removidas e alterações no espaço ocupado. As opções de classificação disponíveis nesse roteiro ampliam a capacidade de análise e facilitam a identificação de padrões e tendências no uso do espaço em disco ao longo do tempo.

Ambos os ficheiros são desenvolvidos para lidar com desafios comuns encontrados na gestão de armazenamento, como presença de espaços nos nomes dos arquivos e pastas, além da gestão adequada de situações onde o acesso a informações específicas é restrito. Será fornecido um detalhamento completo das estratégias adotadas para atender aos requisitos específicos de cada ficheiro. Além disso, serão realizados testes extensivos para validar a precisão e eficácia das soluções propostas.

Este relatório não regista apenas o código-fonte, mas também o raciocínio por trás das decisões tomadas e as considerações que fundamentam a implementação de cada funcionalidade

## Explicação do código

### Definição da variável

A terminal window with a dark background and light-colored text. It contains nine lines of code, each preceded by a line number from 1 to 9. The code defines various variables for a script, including input, directory, regex, min\_size, max\_date, reverse\_sort, limit, sort\_by\_name, and output\_file. The output\_file line uses a date format string. The window has three colored circles (red, yellow, green) in the top left corner, typical of a macOS-style window.

```
1 input="$@"
2 directory="."
3 regex=".*"
4 min_size=0
5 max_date=""
6 reverse_sort="false"
7 limit=99999
8 sort_by_name="false"
9 output_file="spacecheck_$(date +%Y%m%d%H%M%S) "
```

Figura 1- Declaração de variáveis auxiliares

“**input**”: Armazena todos os argumentos fornecidos ao script.

“**directory**”: Define o diretório inicial como o diretório atual.

“**regex**”: Inicializa uma expressão regular padrão.

“**min\_size**”: Define o tamanho mínimo para considerar pastas.

“**max\_date**”: Define a data máxima de modificação para arquivos.

“**reverse\_sort**”: Indica se a ordenação será na ordem inversa.

“**limit**”: Limita o número de linhas.

“**sort\_by\_name**”: Indica se a ordenação será pelo nome.

“**output\_file**”: Nome do arquivo de saída.

A configuração de parâmetros iniciais desempenha um papel fundamental na flexibilidade e adaptabilidade do script. Ao definir variáveis como **input**, **directory**, **regex**, e outros, o script ganha a capacidade de ser personalizado de acordo com as necessidades do usuário.

## Função calculate\_folder\_sizes()

```

1 calculate_folder_sizes() {
2     local search_directory="$1" # Diretório para pesquisar
3
4     # Encontre todas as pastas no diretório
5     find "$search_directory" -type d 2>/dev/null |
6     while read -r folder; # Para cada pasta
7     do
8         # Verifica a permissão de leitura na pasta
9         if [ -r "$folder" ]; then
10            # Filtrar os arquivos que correspondem ao regex e foram modificados antes da data máxima
11            local matching_files=$(find "$folder" -type f -mtime +$( ( $(date +%s) - $(date -d "$max_date" +%s) ) / 86400 ) 2>/dev/null | grep -E "$regex")
12
13            if [ -n "$matching_files" ]; then
14                folder_size=0
15
16                # Calcular o tamanho da pasta considerando a expressão regular
17                while IFS= read -r file; do
18                    size=$(du -b "$file" 2>/dev/null | awk '{print $1}') # Obter o tamanho do arquivo
19                    folder_size=$((folder_size + size)) # Somar o tamanho do arquivo ao tamanho da pasta
20                done <<< "$matching_files" # Passar os arquivos correspondentes para o loop
21
22                if [ "$folder_size" -ge "$min_size" ]; then
23                    echo "$folder_size $folder" # Imprimir o tamanho e o nome da pasta
24                fi
25            else
26                # Se não houver arquivos que atendam aos critérios, defina o tamanho da pasta como 0
27                if [ "$min_size" -eq 0 ]; then
28                    echo "0 $folder"
29                fi
30            fi
31        else
32            echo "NA $folder" # Permissão recusada
33        fi
34    done
35 }

```

Figura 2- Função calculate\_folder\_sizes()

A função **calculate\_folder\_sizes** é uma parte fundamental do script, desempenha o papel de calcular os tamanhos das pastas dentro de um diretório específico, seguindo critérios estabelecidos pelo user, como tamanho mínimo, data de modificação e expressão regular.

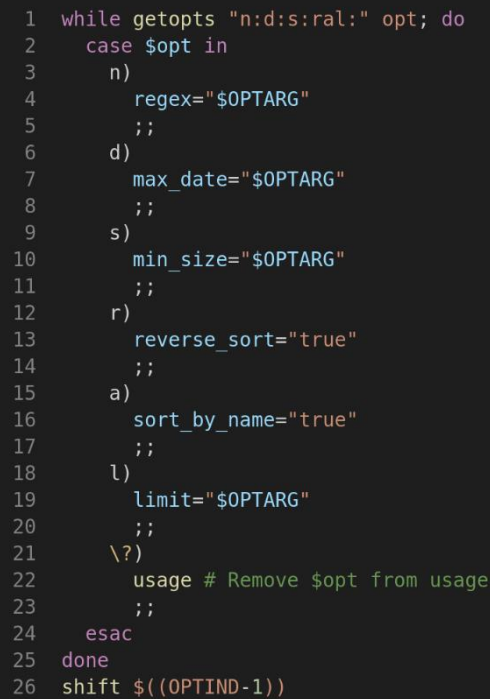
A função ao ser chamada requer um único parâmetro, **search\_directory**, que serve como o diretório base para o cálculo dos tamanhos das pastas (O user ao compilar o ficheiro tem de colocar sempre o diretório). Utilizando o comando **find**, a função realiza uma busca abrangente, identificando todas as pastas dentro do diretório fornecido, que está representado por \$search\_directory. Em seguida, a função inicia um loop de leitura, iterando sobre cada pasta identificada pelo comando find.

A filtragem de arquivos é uma **etapa crucial**. Novamente utilizando o comando find, a função encontra arquivos dentro da pasta que correspondem à expressão regular fornecida (**regex**) e que foram modificados antes da data máxima especificada. Essa filtragem é essencial para considerar apenas os arquivos relevantes no cálculo dos tamanhos.

O cálculo do tamanho total da pasta é realizado utilizando o comando **du**. Este comando soma os tamanhos individuais de todos os arquivos encontrados na pasta, fornecendo uma medida precisa do espaço ocupado. O comando **awk** está a ser usado para realizar uma operação de processamento de texto específica.

Por último, a função considera o caso em que a permissão da leitura na pasta não está concedida. Nesse cenário, a função indica essa falta de permissão, imprimindo **"NA"** seguido do caminho da pasta.

## Estrutura getopt



```

1 while getopt "n:d:s:ral:" opt; do
2     case $opt in
3         n)
4             regex="$OPTARG"
5             ;;
6         d)
7             max_date="$OPTARG"
8             ;;
9         s)
10            min_size="$OPTARG"
11            ;;
12        r)
13            reverse_sort="true"
14            ;;
15        a)
16            sort_by_name="true"
17            ;;
18        l)
19            limit="$OPTARG"
20            ;;
21        \?)
22            usage # Remove $opt from usage
23            ;;
24    esac
25 done
26 shift $((OPTIND-1))

```

Figura 3- estrutura getopt

Utilizamos a estrutura **getopts** para analisar os argumentos da linha de comando, permitindo a passagem de opções e argumentos ao script.

As opções definidas são:

- **n**: Define a expressão regular.
- **d**: Define a data máxima de modificação.
- **s**: Define o tamanho mínimo.
- **r**: Ativa a ordenação reversa.
- **a**: Ativa a ordenação pelo nome.
- **l**: Define o limite de linhas a serem exibidas.

Dentro do bloco case, as variáveis são ajustadas de acordo com as opções fornecidas pelo usuário.

O bloco **shift \$((OPTIND-1))** remove os argumentos da linha de comando que já foram processados pela estrutura **getopts**.

## Cabeçalho

```
1 printf "SIZE NAME %s %s\n" "$build_date" "$input"
```

Figura 5-comando para impressão do cabeçalho

```
1 build_date=$(date +%Y%m%d')
```

Figura 4- comando que obtém a data

Imprime um **cabeçalho** que inclui a data de construção do script e os argumentos de entrada fornecidos.

**\$(date +%Y%m%d')** executa o comando **date** para obter a data atual e, em seguida, formata essa data de acordo com o especificador **'%Y%m%d'**.

## Bloco condicional

```
1 # Tamanhos das pastas
2 if [ "$sort_by_name" == "true" ]; then # Se a ordem for alfabética
3     calculate_folder_sizes "$directory" | # Calcular os tamanhos das pastas
4     sort -k2 | # Ordenar por nome
5     head -n "$limit" | # Obter as primeiras linhas
6     tee "$output_file" # Salvar a saída em um arquivo
7 else
8     if [ "$reverse_sort" == "true" ]; then # Se a ordem for inversa
9         calculate_folder_sizes "$directory" | # Calcular os tamanhos das pastas
10        sort -n | # Ordenar por tamanho
11        head -n "$limit" | # Obter as primeiras linhas
12        tee "$output_file" # Salvar a saída em um arquivo
13    else
14        calculate_folder_sizes "$directory" | # Calcular os tamanhos das pastas
15        sort -n -r | # Ordenar por tamanho em ordem inversa
16        head -n "$limit" | # Obter as primeiras linhas
17        tee "$output_file" # Salvar a saída em um arquivo
18    fi
19 fi
```

Figura 6- bloco condicional

Este bloco condicional no script é responsável por determinar como os resultados do cálculo de tamanhos de pastas devem ser apresentados. Ele considera dois fatores principais: a **ordenação por nome** e a **ordenação por tamanho**.

- Se a variável `$sort_by_name` estiver configurada como "true", os resultados serão ordenados alfabeticamente pelo nome das pastas. Se `$sort_by_name` não for "true", o script verifica se `$reverse_sort` é "true", e se for, os resultados são ordenados por ordem inversa.
- Os resultados são apresentados no terminal e também salvos em um arquivo de saída especificado por `$output_file`.

## Resultado dos testes

Para compilar o ficheiro, primeiro é preciso que o arquivo tenha permissões de execução, para conceder essas permissões é preciso escrever no terminal (**chmod +x spacecheck**).

Agora para monitorizar o espaço do disco usa-se (**./spacecheck.sh**), se o user quiser usar algumas das opções disponíveis e escolher um diretório, tem de digitar no terminal (**./spacecheck |opções | /caminho do diretório** ).

Agora vou listar alguns dos testes que fizemos para verificar se o programa estava a compilar de forma correta.

### Teste de Filtragem por Expressão Regular:

- Cenário: Utilização da opção `-n` para especificar uma expressão regular.
- Objetivo: Confirmar que o script filtra corretamente as pastas com base na expressão regular fornecida.
- Resultados: As pastas são filtradas de acordo com a expressão regular. A saída reflete apenas as pastas que correspondem à expressão especificada.

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -n '.*sh' /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -n .*sh /home/brunotm/Desktop/ua/2ano/so
1337864 /home/brunotm/Desktop/ua/2ano/so
788285 /home/brunotm/Desktop/ua/2ano/so/aula2
542175 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto
245116 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git
96306 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1
66508 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1
23442 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/hooks
9544 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1/rrr
2925 /home/brunotm/Desktop/ua/2ano/so/aula3
2776 /home/brunotm/Desktop/ua/2ano/so/aula4
1703 /home/brunotm/Desktop/ua/2ano/so/aula1
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$
```

Figura 7- teste de filtragem por expressão regular

### Teste de Ordenação em Ordem Inversa (-r):

- Cenário: Utilização da opção `-r` para ordenar os resultados na ordem inversa.
- Objetivo: Verificar se a ordenação por tamanho está correta, com as pastas maiores exibidas primeiro.
- Resultados: As pastas são ordenadas corretamente por tamanho em ordem inversa.

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -r -n '.*sh' /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -r -n .*sh /home/brunotm/Desktop/ua/2ano/so
1703 /home/brunotm/Desktop/ua/2ano/so/aula1
2776 /home/brunotm/Desktop/ua/2ano/so/aula4
2925 /home/brunotm/Desktop/ua/2ano/so/aula3
9544 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1/rrr
23442 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/hooks
66508 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1
96886 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1
245116 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git
542755 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto
788285 /home/brunotm/Desktop/ua/2ano/so/aula2
```

Figura 8- teste de Ordenação em Ordem Inversa



**Teste de Limitação de Saída (-l):**

- Cenário: Utilização da opção -l para limitar a saída a um número específico de linhas.
- Objetivo: Confirmar que o script respeita o limite especificado na saída.
- Resultados: O número de linhas na saída é restrito de acordo com o valor definido pela opção -l.

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -l 2 /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -l 2 /home/brunotm/Desktop/ua/2ano/so
1339604 /home/brunotm/Desktop/ua/2ano/so
788285 /home/brunotm/Desktop/ua/2ano/so/aula2
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$
```

Figura 9- teste de limitação de saída

**Teste de Ordenação por Nome (-a):**

- Cenário: Utilização da opção -a para ordenar os resultados alfabeticamente.
- Objetivo: Verificar se a ordenação por nome está correta.
- Resultados: As pastas são ordenadas alfabeticamente.

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -a -n '*.sh' /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -a -n '*.sh' /home/brunotm/Desktop/ua/2ano/so
1339604 /home/brunotm/Desktop/ua/2ano/so
1703 /home/brunotm/Desktop/ua/2ano/so/aula1
788285 /home/brunotm/Desktop/ua/2ano/so/aula2
2925 /home/brunotm/Desktop/ua/2ano/so/aula3
2776 /home/brunotm/Desktop/ua/2ano/so/aula4
543335 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto
245116 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git
23442 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/hooks
97466 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1
66508 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1
9544 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1/rrr
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$
```

Figura 10- teste de ordenação por nome

**Teste de Filtragem por Tamanho (-s):**

- Cenário: Utilização da opção -s para especificar um tamanho mínimo.
- Objetivo: Confirmar que o script filtra corretamente as pastas com base no tamanho mínimo especificado.
- Resultados: As pastas são filtradas adequadamente, excluindo aquelas que não atendem ao tamanho mínimo especificado.

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -s 10000 /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -s 10000 /home/brunotm/Desktop/ua/2ano/so
1345306 /home/brunotm/Desktop/ua/2ano/so
788285 /home/brunotm/Desktop/ua/2ano/so/aula2
549617 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto
245116 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git
209124 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/objects
180327 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/objects/80
103748 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1
66508 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1
40368 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1/aaa
23442 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/hooks
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -s 45000 /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -s 45000 /home/brunotm/Desktop/ua/2ano/so
1345887 /home/brunotm/Desktop/ua/2ano/so
788285 /home/brunotm/Desktop/ua/2ano/so/aula2
550198 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto
245116 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git
209124 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/objects
180327 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/objects/80
104329 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1
66508 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$
```

Figura 11- teste de filtragem por tamanho

**Teste de Filtragem por Data (-d):**

- Cenário: Utilização da opção -d para especificar uma data máxima de modificação.
- Objetivo: Verificar se o script filtra corretamente as pastas com base na data máxima de modificação.
- Resultados: As pastas são filtradas com precisão, excluindo aquelas cujos arquivos foram modificados após a data especificada.

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -d 'Sep 9 10:10' /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -d Sep 9 10:10 /home/brunotm/Desktop/ua/2ano/so
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -d 'Oct 9 10:10' /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -d Oct 9 10:10 /home/brunotm/Desktop/ua/2ano/so
1346424 /home/brunotm/Desktop/ua/2ano/so
788285 /home/brunotm/Desktop/ua/2ano/so/aula2
1703 /home/brunotm/Desktop/ua/2ano/so/aula1
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -d 'Nov 9 10:10' /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -d Nov 9 10:10 /home/brunotm/Desktop/ua/2ano/so
1346555 /home/brunotm/Desktop/ua/2ano/so
788285 /home/brunotm/Desktop/ua/2ano/so/aula2
550866 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto
245116 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git
209124 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/objects
180327 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/objects/80
104997 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1
66508 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1
40368 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1/aaa
23442 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/hooks
9910 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/logs
9544 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1/rrr
7240 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/test_a1/test_a1/aaa/zzzz
6469 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/logs/refs
3028 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/logs/refs/remotes/origin
3028 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/logs/refs/remotes
2925 /home/brunotm/Desktop/ua/2ano/so/aula3
2776 /home/brunotm/Desktop/ua/2ano/so/aula4
2507 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/objects/pack
1879 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/objects/52
1703 /home/brunotm/Desktop/ua/2ano/so/aula1
1465 /home/brunotm/Desktop/ua/2ano/so/S0_Projeto/.git/objects/35
```

Figuras 12- testes de filtragem por data máxima de edição

Podemos usar **mais que uma opção** ao compilar, como neste exemplo:

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacecheck.sh -s 45000 -l 2 /home/brunotm/Desktop/ua/2ano/so
SIZE NAME 20231111 -s 45000 -l 2 /home/brunotm/Desktop/ua/2ano/so
1346337 /home/brunotm/Desktop/ua/2ano/so
788285 /home/brunotm/Desktop/ua/2ano/so/aula2
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$
```

Figura 13- teste de uso com mais que uma opção

## Lista de Testes (completos)

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_ai$ ./spacecheck.sh /home/brunotm/Desktop/ua/2ano/so_praticas/
SIZE NAME 20231113 /home/brunotm/Desktop/ua/2ano/so_praticas/
6027156 /home/brunotm/Desktop/ua/2ano/so_praticas/
5231467 /home/brunotm/Desktop/ua/2ano/so_praticas/Material das aulas teóricas-20230928
788285 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
2925 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
2776 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4
1703 /home/brunotm/Desktop/ua/2ano/so_praticas/aula1
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_ai$ ./spacecheck.sh -n '*.sh' /home/brunotm/Desktop/ua/2ano/so_praticas/
SIZE NAME 20231113 -n *.sh /home/brunotm/Desktop/ua/2ano/so_praticas/
14367 /home/brunotm/Desktop/ua/2ano/so_praticas/
6991 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
2897 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
2776 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4
1703 /home/brunotm/Desktop/ua/2ano/so_praticas/aula1
0 /home/brunotm/Desktop/ua/2ano/so_praticas/Material das aulas teóricas-20230928
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_ai$ ./spacecheck.sh -n '*.sh' -l 2 /home/brunotm/Desktop/ua/2ano/so_praticas/
SIZE NAME 20231113 -n *.sh -l 2 /home/brunotm/Desktop/ua/2ano/so_praticas/
14367 /home/brunotm/Desktop/ua/2ano/so_praticas/
6991 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_ai$ ./spacecheck.sh -n '*.sh' -s 1800 /home/brunotm/Desktop/ua/2ano/so_praticas/
SIZE NAME 20231113 -n *.sh -s 1800 /home/brunotm/Desktop/ua/2ano/so_praticas/
14367 /home/brunotm/Desktop/ua/2ano/so_praticas/
6991 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
2897 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
2776 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_ai$ ./spacecheck.sh -n '*.sh' -d 'Nov 12 10:00' /home/brunotm/Desktop/ua/2ano/so_praticas/
SIZE NAME 20231113 -n *.sh -d Nov 12 10:00 /home/brunotm/Desktop/ua/2ano/so_praticas/
14367 /home/brunotm/Desktop/ua/2ano/so_praticas/
6991 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
2897 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
2776 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4
1703 /home/brunotm/Desktop/ua/2ano/so_praticas/aula1
0 /home/brunotm/Desktop/ua/2ano/so_praticas/Material das aulas teóricas-20230928
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_ai$ ./spacecheck.sh -n '*.sh' -d 'Sep 01 10:00' /home/brunotm/Desktop/ua/2ano/so_praticas/
SIZE NAME 20231113 -n *.sh -d Sep 01 10:00 /home/brunotm/Desktop/ua/2ano/so_praticas/
0 /home/brunotm/Desktop/ua/2ano/so_praticas/Material das aulas teóricas-20230928
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula1
0 /home/brunotm/Desktop/ua/2ano/so_praticas/
```

Figura 14- teste de uso com mais que uma opção

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_ai$ ./spacecheck.sh -r -n '*.sh' /home/brunotm/Desktop/ua/2ano/so_praticas/
SIZE NAME 20231113 -r -n *.sh /home/brunotm/Desktop/ua/2ano/so_praticas/
0 /home/brunotm/Desktop/ua/2ano/so_praticas/Material das aulas teóricas-20230928
1703 /home/brunotm/Desktop/ua/2ano/so_praticas/aula1
2776 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4
2897 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
6991 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
14367 /home/brunotm/Desktop/ua/2ano/so_praticas/
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_ai$ ./spacecheck.sh -a -n '*.sh' /home/brunotm/Desktop/ua/2ano/so_praticas/
SIZE NAME 20231113 -a -n *.sh /home/brunotm/Desktop/ua/2ano/so_praticas/
14367 /home/brunotm/Desktop/ua/2ano/so_praticas/
1703 /home/brunotm/Desktop/ua/2ano/so_praticas/aula1
6991 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
2897 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
2776 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4
0 /home/brunotm/Desktop/ua/2ano/so_praticas/Material das aulas teóricas-20230928
```

Figura 15- teste de uso com mais que uma opção

## Spacerate.sh

O script **spacerate.sh** foi concebido com o objetivo de comparar 2 ficheiros obtidos a partir da execução do script **spacecheck.sh**, permitindo monitorizar a variação do espaço ocupado no disco ao longo do tempo, identificando as diferenças nos tamanhos dos ficheiros, se um diretório foi adicionado ou removido ou se teve o seu tamanho alterado. Para além disso, assim como o script **spacecheck.sh**, permite a ordenação dos seus resultados por ordem alfabética (**-a**) ou por ordem de tamanho de cada um dos diretórios, sendo que é possível inverter esta ordem com o uso de **-r** no momento de execução do script.

## Explicação do código

### Definição de variáveis

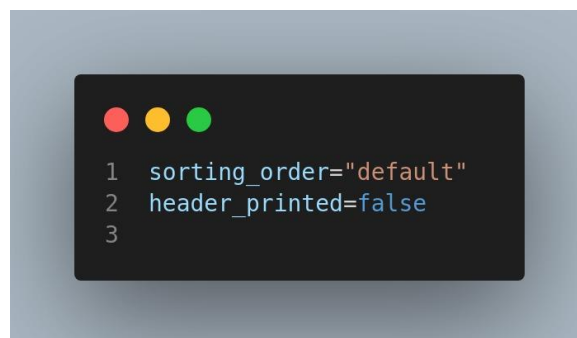
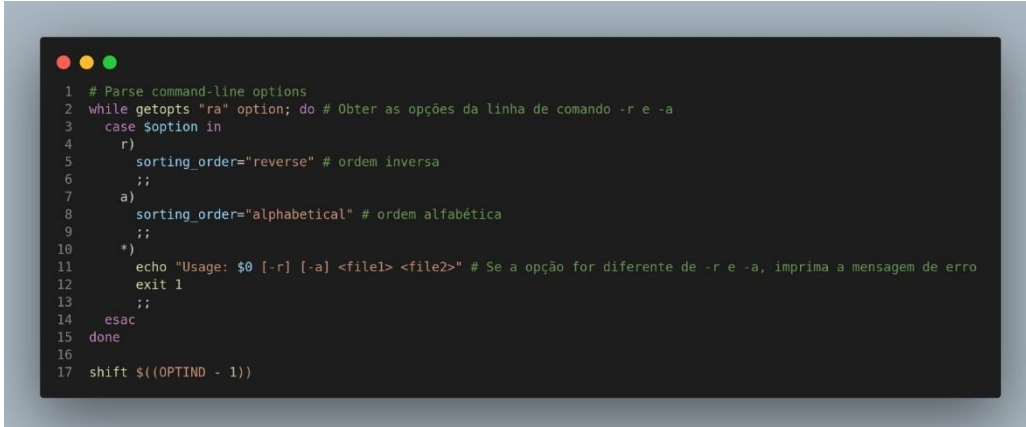


Figura 16- declaração de variável

Na fase inicial do programa, encontra-se a definição das variáveis **sorting\_order** e **header\_printed** que irão auxiliar a ordem da impressão dos resultados no terminal de acordo com as instruções dadas pelo utilizador.

## Estrutura getopt



```

1 # Parse command-line options
2 while getopt "ra" option; do # Obter as opções da linha de comando -r e -a
3     case $option in
4         r)
5             sorting_order="reverse" # ordem inversa
6             ;;
7         a)
8             sorting_order="alphabetical" # ordem alfabética
9             ;;
10        *)
11            echo "Usage: $0 [-r] [-a] <file1> <file2>" # Se a opção for diferente de -r e -a, imprima a mensagem de erro
12            exit 1
13            ;;
14        esac
15    done
16
17 shift $((OPTIND - 1))
  
```

Figura 17- estrutura getopt

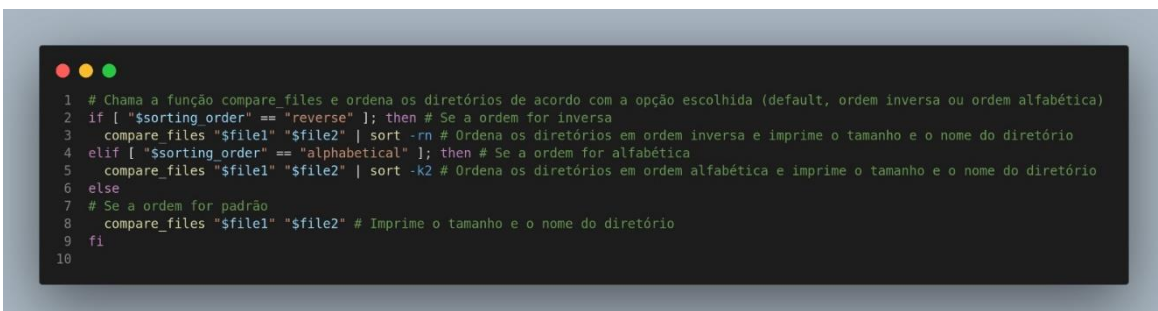
Esta estrutura analisa os argumentos introduzidos pelo utilizador e interpreta-os. Estes servirão de apoio à execução do script, uma vez que indicam a ordem pela qual os resultados serão apresentados no terminal.

As opções são:

- **r**: ativa a ordenação inversa, definindo a variável **sorting\_order** para “reverse”;
- **a**: ativa a ordenação alfabética, definindo a variável “**sorting\_order** para “alphabetical”.

O bloco **shift \$((OPTIND-1))** remove os argumentos da linha de comando que já foram processados pela estrutura **getopts**.

## Validação de número de argumentos



```

1 # Chama a função compare_files e ordena os diretórios de acordo com a opção escolhida (default, ordem inversa ou ordem alfabética)
2 if [ "$sorting_order" == "reverse" ]; then # Se a ordem for inversa
3     compare_files "$file1" "$file2" | sort -rn # Ordena os diretórios em ordem inversa e imprime o tamanho e o nome do diretório
4 elif [ "$sorting_order" == "alphabetical" ]; then # Se a ordem for alfabética
5     compare_files "$file1" "$file2" | sort -k2 # Ordena os diretórios em ordem alfabética e imprime o tamanho e o nome do diretório
6 else
7     # Se a ordem for padrão
8     compare_files "$file1" "$file2" # Imprime o tamanho e o nome do diretório
9 fi
10
  
```

Figura 18- bloco condicional que valida o número de argumentos

Este pequeno bloco condicional permite validar o número de argumentos que o utilizador introduziu ao chamar ao iniciar a execução do script, sendo este obtido com recurso a “\$#”. Caso este número seja diferente de 2, é impressa uma mensagem de erro e sai com um código de saída igual a 1, terminando e sua execução.



## Leitura de arquivos

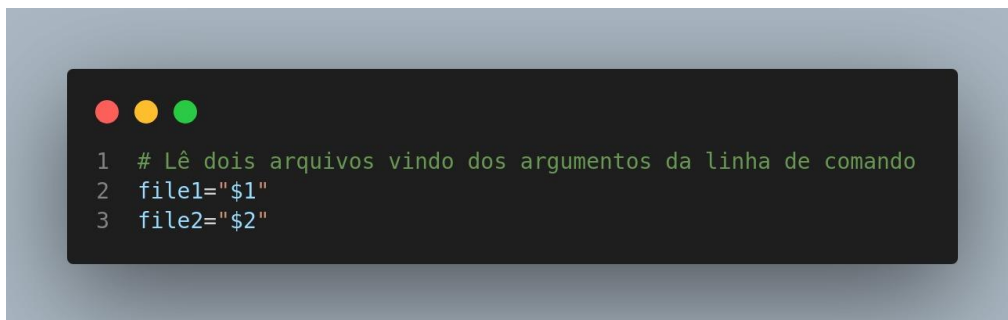


Figura 19- armazenamento dos arquivos passados nos argumentos

Neste excerto, são armazenados nas variáveis **file1** e **file2** os ficheiros passados como argumento, facilitando a sua utilização em partes posteriores do script.

## Função `compare_files`

Esta é a função principal do script. É nela onde é executada a comparação entre os diretórios presentes nos dois ficheiros, o objetivo deste script. Podemos dividi-la em três partes diferentes:

- Declaração de arrays e armazenamento dos dados dos ficheiros;
- Iteração nos arrays e cálculos da diferença;
- Bloco condicional para ordenação

## Declaração de arrays e armazenamento dos dados dos ficheiros

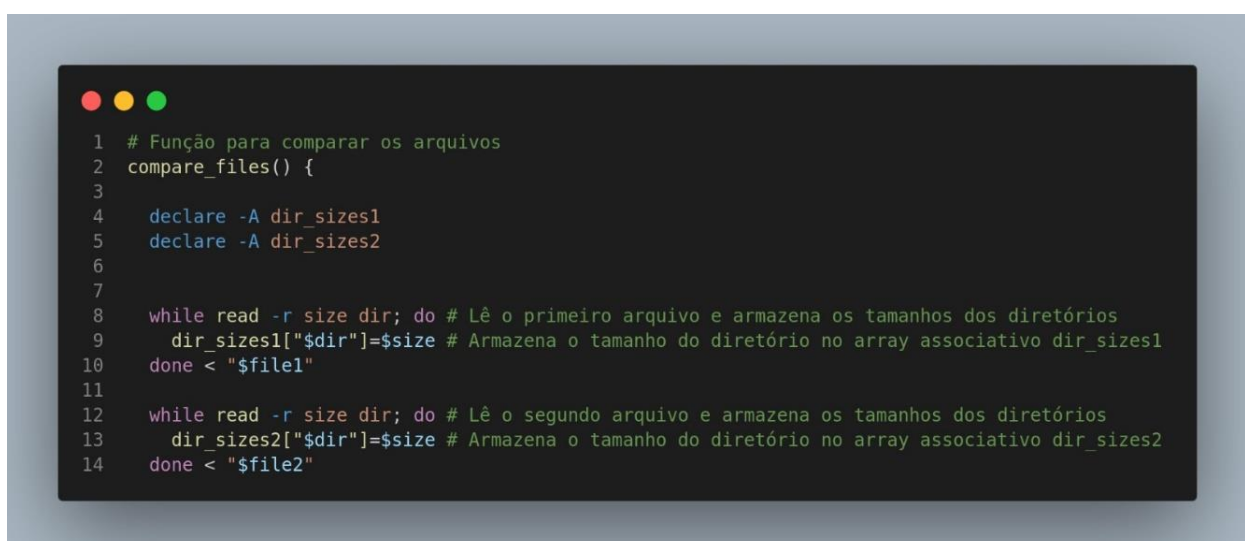


Figura 20- declaração de arrays e armazenamento dos dados dos ficheiros

Numa fase inicial, a função declara dois arrays, **dir\_sizes1** e **dir\_sizes2**. Posteriormente, realiza dois ciclos *while*, um para cada ficheiro (**file1** e **file2**), onde lê cada ficheiro linha a linha, dividindo-a em duas partes: **size** e **dir**. Para cada diretório (**dir**), atribui um valor do tamanho (**size**) e armazena estes pares no array respetivo (**dir\_sizes1** para **file1** e **dir\_size2** para **file2**, este encaminhamento é efetuado pela expressão “< fileX”, em que “X” é o número do ficheiro).

### Iteração nos arrays e cálculos da diferença

```

1  # Compara e imprime as diferenças nos tamanhos dos diretórios
2  for dir in "${!dir_sizes1[@]"; do
3      size1="${dir_sizes1[$dir]}"
4      size2="${dir_sizes2[$dir]}"
5
6      if [ -z "$size2" ]; then # Se o diretório não existe no segundo ficheiro
7          echo "-$size1 $dir REMOVED" # Se o diretório foi removido
8      else
9          size_diff=$((size2 - size1))
10         echo "$size_diff $dir"
11     fi
12 done
13
14 # Verifica diretórios adicionados no segundo ficheiro
15 for dir in "${!dir_sizes2[@]"; do
16     size1="${dir_sizes1[$dir]}"
17     size2="${dir_sizes2[$dir]}"
18
19     if [ -z "${dir_sizes1[$dir]}" ]; then # Se o diretório não existe no primeiro ficheiro
20         echo "${dir_sizes2[$dir]} $dir NEW" # Se o diretório foi adicionado
21     fi
22 done
23 }
```

Figura 21- iteração nos arrays

Esta fase, a mais complexa e importante do script, itera sobre os diretórios do array associativo **dir\_sizes1**, armazenando o nome em **dir** e o tamanho desse diretório do primeiro ficheiro em **size1** e do segundo ficheiro em **size2**. Posteriormente, verifica se **size2** é vazio (“if [ -z “\$size2” ]”). Daqui existem 2 caminhos:

- Se **size2** é vazio, apresenta no terminal que este foi removido;
- Se **size2** não é vazio, calcula a diferença dos dois tamanhos e apresenta essa informação no terminal.

Este processo repete-se, só que agora o array a ser iterado é **dir\_sizes2**, e se um dos diretórios exista em **dir\_size2** mas não em **dir\_size1**, é apresentada a informação de que este é novo.

## Bloco condicional para ordenação

```

1 # Chama a função compare_files e ordena os diretórios de acordo com a opção escolhida (default, ordem inversa ou ordem alfabética)
2 if [ "$sorting_order" == "reverse" ]; then # Se a ordem for inversa
3   compare_files "$file1" "$file2" | sort -rn # Ordena os diretórios em ordem inversa e imprime o tamanho e o nome do diretório
4 elif [ "$sorting_order" == "alphabetical" ]; then # Se a ordem for alfabética
5   compare_files "$file1" "$file2" | sort -k2 # Ordena os diretórios em ordem alfabética e imprime o tamanho e o nome do diretório
6 else
7   # Se a ordem for padrão
8   compare_files "$file1" "$file2" # Imprime o tamanho e o nome do diretório
9 fi
10

```

Figura 22- bloco condicional para ordenação

Este bloco condicional incide na variável **sorting\_order**, que consoante o valor que lhe foi atribuído, muda o modo como os resultados de **compare\_files** são apresentados:

- Se **sorting\_order == "reverse"**, acrescenta o parâmetro "**sort -rn**" à chamada de **compare\_files**, fazendo com que os resultados sejam apresentados na ordem inversa;
- Se **sorting\_order == "alphabetical"**, acrescenta o parâmetro "**sort -k2**" à chamada de **compare\_files**, fazendo com que os resultados sejam apresentados por ordem alfabética;
- Caso não se verifique nenhuma das condições, **compare\_files** é chamada normalmente.

## Resultado dos testes

### Ficheiros de exemplo:

- spacecheck\_20231113183151
  - 795689 /home/brunotm/Desktop/ua/2ano/so\_praticas/
  - 788285 /home/brunotm/Desktop/ua/2ano/so\_praticas/aula2
  - 2925 /home/brunotm/Desktop/ua/2ano/so\_praticas/aula3
  - 2776 /home/brunotm/Desktop/ua/2ano/so\_praticas/aula4
- spacecheck\_20231113183155
  - 795689 /home/brunotm/Desktop/ua/2ano/so\_praticas/
  - 788285 /home/brunotm/Desktop/ua/2ano/so\_praticas/aula2
  - 2925 /home/brunotm/Desktop/ua/2ano/so\_praticas/aula3
  - 1703 /home/brunotm/Desktop/ua/2ano/so\_praticas/aula1



```

brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacerate.sh spacecheck
spacecheck_20231113183151 spacecheck_20231113183155 spacecheck.sh
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacerate.sh spacecheck_20231113183151 spacecheck_20231113183155
SIZE NAME
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
-2776 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4 REMOVED
0 /home/brunotm/Desktop/ua/2ano/so_praticas/
1703 /home/brunotm/Desktop/ua/2ano/so_praticas/aula1 NEW
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ █

```

Figura 23- teste simples

Neste exemplo, comparamos os ficheiros spacecheck\_20231113183151 com o ficheiro spacecheck\_20231113183155. Quando compilamos desta maneira a ordem é default, e só aparecem duas pastas com algum “size”, isto acontece porque os ficheiros são bastante semelhantes.

Na pasta da aula 4, no final do caminho, aparece a palavra “REMOVED” pois este caminho só aparece no ficheiro 1 não no ficheiro 2. O size aparece em negativo porque ao compará-los calcula-se a diferença entre o arquivo 2 e o arquivo 1.

Quando aparece a mensagem “NEW” isso significa que o ficheiro está presente, neste caso, no spacecheck\_20231113183155 e não no spacecheck\_20231113183151.

```

brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacerate.sh -r spacecheck_20231113183151 spacecheck_20231113183155
SIZE NAME
1703 /home/brunotm/Desktop/ua/2ano/so_praticas/aula1 NEW
0 /home/brunotm/Desktop/ua/2ano/so_praticas/
-2776 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4 REMOVED
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ █

```

Figura 24- teste com uso de -r para ordenação em ordem inversa

Agora como usamos a opção -r, significa que a ordem vai ser alterada para uma ordem inversa à default.

```

brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ ./spacerate.sh -a spacecheck_20231113183151 spacecheck_20231113183155
SIZE NAME
0 /home/brunotm/Desktop/ua/2ano/so_praticas/
1703 /home/brunotm/Desktop/ua/2ano/so_praticas/aula1 NEW
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula2
0 /home/brunotm/Desktop/ua/2ano/so_praticas/aula3
-2776 /home/brunotm/Desktop/ua/2ano/so_praticas/aula4 REMOVED
brunotm@brunotavares:~/Desktop/ua/2ano/so/S0_Projeto/test_a1$ █

```

Figura 25- teste ordenação alfabética

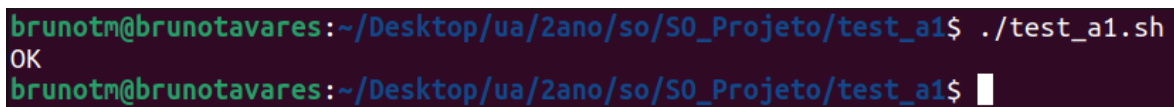
Neste caso, ao utilizarmos a opção -a, as pastas e subpastas são ordenadas por ordem alfabética.

## Conclusão

Em conclusão, os ficheiros **spacecheck.sh** e **spacerate.sh**, quando utilizados em conjunto, oferecem ao utilizador a possibilidade de gerir e analisar tanto o conteúdo do disco bem como a evolução da ocupação deste ao longo do tempo. A capacidade de personalização e as diferentes opções oferecidas, conferem versatilidade e qualidade aos scripts.

Apesar de aparecerem algumas dificuldades na redação do código consideramos que as conseguimos ultrapassar com alguma facilidade com alguma pesquisa, assim como muita entreaajuda na superação das dificuldades.

O teste que nos foi disponibilizado no e-learning foi realizado com sucesso:

A terminal window with a dark background. The prompt is 'brunotm@brunotavares:~/Desktop/ua/2ano/so/SO\_Projeto/test\_a1\$'. The user enters './test\_a1.sh'. The output is 'OK'. The prompt returns to 'brunotm@brunotavares:~/Desktop/ua/2ano/so/SO\_Projeto/test\_a1\$' followed by a cursor.

```
brunotm@brunotavares:~/Desktop/ua/2ano/so/SO_Projeto/test_a1$ ./test_a1.sh
OK
brunotm@brunotavares:~/Desktop/ua/2ano/so/SO_Projeto/test_a1$
```

Figura 21- resultado do teste disponibilizado

## **Bibliografia**

Para a realização do trabalho foi principalmente consultado os powerpoints relativos à programação em bash disponíveis na página do e-learning da cadeira. Foi também utilizado conteúdo obtido em sites para tirar dúvidas pontuais.

<https://stackoverflow.com>

<https://linuxhint.com>

<https://linuxhint.com/sort-bash-column-linux/>

<https://www.gnu.org/savannah-checkouts/gnu/bash/manual/bash.html>