Relatório Trabalho 1 Monitorização de interfaces de rede em bash

Filipe Antão 103470

João Miguel Almeida Matos 103341

Para este trabalho, foi-nos pedido para realizar-mos um script que permitisse a visualização da quantidade de dados transmitidos e recebidos nas interfaces de rede selecionadas e as respetivas taxas de transferência para periodos de tempo pré-estabelecidos. O nosso script ***netifstat.sh*** cumpre todos os requisitos pedidos para o trabalho.

O nosso script está dividido em 5 partes:

- A função ***usage()***, que exibe os argumentos que são suportados pelo script;
- A função ***sortTable()***, usada quando o script é executado com um argumento de *sort*;
- A função ***filterTable()***, usada quando o script é executado com o argumento de filtragem das interfaces;
- As funções ***printTable()*** e ***printTableLoop()***, implementadas para dar apresentar a tabela contendo os dados após ser feita a respetiva recolha e manipulação dos mesmos;
- O corpo ***main*** do script, onde estão implementadas a seleção dos argumentos do script por parte do utilizador, bem como as rotinas associadas a cada argumento

Corpo *main*

O corpo main, como referido em cima, contém todas as rotinas associadas a cada argumento recebido na execução do script.

Em primeiro lugar, fazemos a análise das opções e argumentos que são passados na execução. Fazemos isso usando o comando da shell do Unix ***"getopts"***. Este comando reconhece as opções e argumentos passados para o script, criando *flags* quando as opções foram passadas. É possível passar várias opções, mas, no entanto, se essas opções forem mutualmente exclusivas (por exemplo, "-b" e "-m", são ambas utilizadas para representar a quantidade de dados mas em medidas diferentes), o programa chama a função ***usage()***, assim como quando o utilizador tenta utilizar uma opção não suportada.

Para além dessas opções, existe um argumento obrigatório, que é o número de segundos em que o o script monitoriza a transferência de dados nas interfaces. Caso o

argumento recebido seja um valor inválido o script, como com qualquer erro executa a função ***usage()***.

Após a análise das opções, o script faz a atribuição e cálculo dos valores de de dados que são recebidos e transferidos por cada valor de segundos passados. Para a recolha dos nomes das interfaces de rede utilizamos uma combinação dos comandos cut, tr, grep, e awk no output da função ifconfig, de modo a obtermos apenas os nomes das interfaces e os respectivos valores. Guardamos os valores em arrays para os podermos processar posteriormente.

As várias opções que podem ser executadas com o script são:

- **-b -k -m**
  Ao invocar um destes argumentos (mutualmente exclusivos), a flag correspondente toma o valor 1. Dependendo da *flag* ativa, os valores em bytes são (ou não) convertidos e kilobytes ou megabytes.

- **-l**

  Ao invocar este argumento, a *flag* correspondente toma o valor 1. Caso ativa, esta flag faz o *script* rodar dentro de um ciclo while que não acaba, mostrando uma nova tabela a cada intervalo de tempo especificado.

- **-p**
  Ao invocar este argumento, a *flag* correspondente toma o valor do argumento passado (um inteiro positivo), que limita o número de interfaces a serem mostradas na tabela.

- **-c**
  Ao invocar este argumento, a *flag* correspondente toma o valor de uma *substring* especificada. O script filtra as interfaces de modo a mostrar apenas aquelas que contêm a *substring*.

- **-t -r -T -R**
  Ao invocar um destes argumentos, a flag "$SORT" toma o valor da opção. Depois essa flag é considerada na função sortTable() para fazer a ordenação das linhas da tabela.

- **-v**

  Ao invocar este argumento, a *flag* correspondente toma o valor 1. Esse valor é considerado na função sortTable() para ordenar os arrays inversamente.

Função *sortTable()*

A função s*ortTable()* guarda todos os valores correspondentes as interfaces de rede numa variável v. Depois de ter essse array com os valores correpondentes a cada interface individual, ordenam-se os valores consoante o valor da variável ”$SORT” e o valor de “$OP_V”. Este valores das linhas são guardados no array SORTEDLINES.

Função filterTable()

A função *filterTable()* filtra o nome das interfaces no array SORTEDLINES, passando apenas aquelas que contêm no nome o argumento passado ao *script* para o array FILTEREDLINES, para que apenas essas sejam mostradas aquando da apresentação da tabela.

Função *printTable()*

A funão *printTable()* apenas formata os valores de FILTEREDLINES e imprime-os na consola. Para o caso de a opção loop ser passada como argumento, também existe a função *printTableLoop()*.

Função *usage()*

A função *usage()* é invocada aquando da incorreta utilização do script. Imprime como passar corretamente os argumentos ao script, e encerra o mesmo, retornando um valor de erro.

Alguns dos testes realizados:

- ./netifstat 10
- ./netifstat -l 3
- ./netifstat -k -t -v 5
- ./netifstat -m -R 5
- ./netifstat -p 1 5
- ./netifstat -c “w” -k -T -l 10