

3ª Avaliação de Programação de Scripts

Redes de Computadores – Campus da UFC em Quixadá – 2019.2

Instruções: dentro da pasta *prova* no servidor, crie a pasta *prova03* e coloque os *scripts* das questões. Cada questão vale 2,5 pontos.

Entrega: 10/11/2019.

1ª Questão) Faça um *script awk* chamado *questao01.awk* que recebe um arquivo no seguinte formato:

```
0:0:1:1
1:1:4:5
2:1:0:0
1:2:3:4
```

Os dois primeiros números correspondem às coordenadas de um ponto e os dois segundos correspondem a coordenadas de um outro ponto. Calcule a distância entre os pontos. Exemplo de execução:

```
$ cat pontos.txt
0:0:1:1
1:1:4:5
2:1:0:0
1:2:3:4
$ awk -f questao01.awk pontos.txt
Distância entre (0,0) e (1,1) é 1.414214
Distância entre (1,1) e (4,5) é 5.000000
Distância entre (2,1) e (0,0) é 2.236068
Distância entre (1,2) e (3,4) é 2.828427
```

2ª Questão) Nesta questão, você irá escrever o *script* *questao02.sh*. A execução do *script* será da seguinte forma:

```
$ ./questao02.sh -u joaomarclo -k joaomarclo.pem -h scripts.joao.marclo.nom.br
```

Em outras palavras, o *script* recebe como parâmetros o nome de usuário, sua chave SSH e um endereço IP. A seguir, você deve criar as seguintes janelas com o uso do comando *dialog*:

1. Uma janela em que o usuário pode informar um pequeno programa na linguagem C.
2. Uma caixa de diálogo perguntando se o usuário deseja compilar remotamente o código.
3. Uma janela exibindo o resultado da compilação. Se a compilação falhar, o *script* finaliza.
4. Caso a compilação seja um sucesso, uma nova caixa de diálogo deve perguntar se o usuário deseja executar remotamente o código.
5. Uma janela com o resultado da execução. Seja a execução bem sucedida ou não, o *script* finaliza.

Entre os passos 2 e 3 entre os passos 4 e 5 existe comunicação entre sua máquina local e o servidor cujo IP foi fornecido como parâmetro. Faça a comunicação através do SSH/SCP. Como a execução é remota, o código em C fornecido deve ser bem simples, sem chamadas *scanf* ou recebendo valores como parâmetros. Se você não sabe C, procure na Google exemplos de programas simples, como um Olá Mundo ou imprimir os números de 0 a 10.

3ª Questão) O objetivo desta questão é desenvolver o *script* `questao03.sh`. Este *script* irá fazer buscas por docentes da UFC-Quixadá que tenham uma determinada área de interesse. Em <https://www.quixada.ufc.br/docente/>, você encontra *links* para o perfil completo dos professores do campus. Neste perfil há um campo **Áreas de Interesses**, no qual o professor descreve os assuntos que pesquisa. O seu *script* deve receber um conjunto de palavras chaves e retornar a lista de professores em cujas áreas de interesses aparece qualquer uma das palavras chaves. Exemplo hipotético (não corresponde à realidade):

```
$ ./questao03.sh -k "redes, peer-to-peer, sistemas distribuídos"
Marcos Dantas
João Marcelo
Jeandro Mesquita
...
```

As palavras chaves são informadas pela opção `-k`, separadas por vírgula. Não diferencie maiúsculas de minúsculas.

O seu *script* não deve assumir que a página já foi baixada e está salva em um arquivo ou diretório. Toda vez que executar, ele deve verificar se as páginas já foram baixadas, caso contrário, deve fazer o *download* para pesquisá-las.

Por último, você deve suportar a opção `-f`. Essa opção, quando presente, força o download da página independente se ela já tiver sido baixada ou não. Pense como a opção de recuperar a informação mais recente. Portanto, a execução:

```
$ ./questao03.sh -f -k "redes, peer-to-peer, sistemas distribuídos"
```

Deve buscar as páginas e fazer a busca, mesmo que já tenha uma versão da página em disco. Sinta-se à vontade de usar *awk*, caso ajude.

4ª Questão) Para o *script* `questao04.sh`, você irá analisar o arquivo `/home/compartilhado/auth.log`. Ao receber o arquivo, o *script* deve exibir, em ordem decrescente em tempo de uso, os usuários que acessaram o servidor. Em outras palavras, o primeiro da lista deve ser o usuário que passou mais tempo no servidor, seguido do usuário que passou a segunda maior quantidade de tempo etc. Portanto:

```
$ ./questao04.sh /home/compartilhado/auth.log
usuario1: 1h32m14s
usuario2: 0h47m13s
usuario3: 0h24m1s
...
```

Detalhes importantes:

1. Procure no *log* quando um usuário criou uma sessão e quando finalizou, calculando assim o tempo de uso do usuário.
2. Existe muito lixo no arquivo, de tentativas de invasões. Faça uma limpeza antes de analisar.
3. Cada nova sessão tem um número, você pode usá-lo para diferenciá-las.
4. Um usuário pode estar logado em mais de uma sessão ao mesmo tempo. Conte o tempo desde o *login* na primeira sessão e o *logout* na última sessão, ou seja, não conte intervalos duas vezes se o usuário estiver em duas sessões.
5. Use *awk* se achar necessário.