



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

TASK 2

Matheus Santos Rosa Carneiro¹

1) Sabendo que a nona coluna delimitada por espaço em branco no arquivo access.log indica o status do pedido e que um status 404 significa que o arquivo pedido não foi encontrado, determine quantos arquivos foram pedidos e não encontrados no log.

```
R: bugger@bugger:~/Downloads$ cat access.log | cut -d " " -f 9 |  
grep "404" | uniq -c
```

2) Quantos dos arquivos da questão 1 foram pedidos no mês de Junho?

```
bugger@bugger:~/Downloads$ cat access.log | cut -d " " -f 4 |  
cut -d "/" -f 2 | grep "Jun" | grep "404" -c
```

3)

Utilizando pipe, crie uma linha de comando que:

- * cria uma lista longa com o conteúdo de /usr/bin
- * mostra somente os tamanhos e nomes dos arquivos encontrados, ordenados por tamanho

```
bugger@bugger:~$ ls -la /usr/bin | awk -F" " '{ print $5 ":" $9}' |  
sort -nr
```

4) Faça a mesma coisa da questão 3, porém utilizando um shell script que receba o diretório a ser listado como parâmetro.

```
#!/bin/bash
echo "Informe o diretório: "
read dir
ls -la $dir | awk -F" " '{ print $5 ":" $9}' | sort -nr
```

5) Escreva um shell script que receba exatamente dois números como parâmetro e mostre quem é maior entre eles.

```
R: #!/bin/bash
if [ $# != 2 ]
then
echo "Usa apenas dois argumentos!"
else

    if [ $1 -gt $2 ]
then

echo "O numero $1 é maior"
else
if [ $2 -gt $1 ]
then
echo "Numero $2 é o maior"
else
echo "Um é igual ao outro"
fi
fi
fi
```

6) O arquivo /etc/passwd mostra o login e o nome completo de cada usuário do sistema (campos 1 e 5) separados por um tab. Seu último campo mostra o shell utilizado por cada usuário. Escreva um script que mostre login, o nome do usuário e o shell utilizado para todos os usuários do sistema.

```
R: #!/bin/bash
```

```
cat /etc/passwd |grep -v "nologin" | grep -v "false" | cut -d ":" -f 1,5,7 >
texto.txt
for linha in $(cat texto.txt);
do
login=$(echo $linha | cut -d ":" -f 1)
nome=$(echo $linha | cut -d ":" -f 2 | cut -d "," -f 1)
shell=$(echo $linha | cut -d ":" -f 3)
echo "Login=$login"
echo "Nome=$nome"
echo "Shell=$shell"
echo ""
done
```

7) Escreva um shell script que receba duas palavras como parâmetro e verifique se a primeira palavra está contida dentro da segunda. Só mostre uma mensagem informativa em caso de sucesso; do contrário, não mostre nada.

R: #!/bin/bash

```
if [ $# == 2 ]
then
    echo $2 | grep -qs $1 && echo "$1 está contido em $2"
else
    echo "Precisa de dois argumentos"
fi
```

8) Escreva um shell script que conte quantos arquivos um usuário possui em seu diretório e informe esta quantidade. Este script deve ser executado sempre que um terminal for aberto pelo usuário.

R: #!/bin/bash

```
ls $1 | wc -l
```

Para aparecer toda vez que o usuário abrir o terminal, basta direcionar o caminho do script dentro do .bashrc:

~/Documentos/countdir.sh

9) Escreva um shell script que compacte todo o conteúdo do seu diretório trabalhos. Este script deve apagar os arquivos após a compactação. Este script deve, ainda, ser executado às 23h59 de todas as segundas feiras do mês. Dica: para compactar arquivos, use o comando `zip <nomedoarquivo.zip> <nomedo(s)arquivo(s)acomparar>`

R: `#!/bin/bash`

```
zip all.zip *
for i in $(ls | grep -v "all.zip");do
  rm -rf "$i"
done
echo "59, 23, *, *, 1 ~/Documentos/meu_novo_dir/crontab.sh" >>
/etc/crontab
```

10) Escreva um shell script que, a partir do arquivo access.log, que deve ser passado por parâmetro, calcule a quantidade total de bytes transferidas em um mês de acessos ao servidor. Dica: observe a coluna que conta os bytes transferidos. Use grep para selecionar o mês.

R: `cat access.log | cut -d " " -f 4,10 | cut -d "/" -f 2,3 | grep "Apr" | grep -v "-" | cut -d " " -f 2`