Lista de Exercícios – Bash

**ATENÇÃO**: Lembre-se de colocar no início de todos os seus scrips

#!/bin/bash

Lembre-se também de conceder permissão de execução ao seu script

sudo chmod 755 <nome do seu script>

1. Crie um script bash chamado questao01.sh. Ele deve conter uma variável cujo valor é o seu nome. O script deve ecoar a expressão “Olá, ” junto com o valor da variável que você criou.
2. Crie um script bash chamado questao02.sh. Ele deve receber um nome como parâmetro na linha de comando. O script deve imprimir a expressão “Olá, ” junto com o parâmetro passado.
3. Crie um script bash chamado questao03.sh. Ele deve receber um nome como parâmetro na linha de comando. O script deve criar uma pasta com o nome passado e ecoar a frase : “Pasta ”<nome da pasta> “ criada com sucesso!”

1. Observe o script abaixo:

#!/bin/bash

meu\_numero\_favorito=13

echo “Meu número favorito é: $meu\_numero\_favorito”

meu\_numero\_favorito=15

echo “Meu número favorito é: $meu\_numero\_favorito”

Qual é a saída da execução desse script? Salve a saída em um arquivo questao04.txt.

1. Observe o script abaixo:

#!/bin/bash

meu\_numero\_favorito=13

nome\_qualquer=“Jose”

echo “Meu número favorito é: $meu\_numero\_favorito”

meu\_numero\_favorito=$nome\_qualquer

echo “Meu número favorito é: $meu\_numero\_favorito”

Qual é a saída da execução desse script? Salve a saída em um arquivo questao05.txt.

1. O comando “cat” exibe na tela o conteúdo de qualquer arquivo que o usuário passar como parâmetro. Crie um script bash chamado questao06.sh que:
   1. Baixe o arquivo em <https://archive.org/stream/alicesadventures19033gut/19033.txt> usando o comando wget
   2. Exiba o conteúdo do arquivo baixado com : cat 19033.txt
2. Quando desejamos ver um arquivo muito longo, usar o cat pode ser um problema, pois dificulta a leitura. Para esses casos, use o comando less. Execute no terminal: less 19033.txt. Clique “q” quando desejar sair. Explique a diferença para o resultado com o comando cat. Escreva a sua resposta em um arquivo questao07.txt
3. O comando lspci mostra todos os dispositivos conectados ao computador pelo barramento PCI. Execute-o no terminal e identifique quais dispositivos o sistema reconhece. Salve a saída em um arquivo questao08.txt.
4. O operador “>” indica redirecionamento. É possível escrever um fluxo de dados (stream) para alguma saída específica. Por exemplo, o resultado de qualquer comando pode ser redirecionado para algum arquivo, para a saída padrão (tela) ou para a impressora. Já o operador “>>” evita sobrescrever o arquivo, acrescentando os dados a um arquivo já existente. Execute no terminal o comando:

lspci > questao09.txt

Veja o conteúdo do arquivo resultante usando o comando cat

1. Salve o script da questão 4) , execute-o redirecionando a sua saída para o arquivo questao10.txt usando o operador “>” . Verifique o conteúdo do questao10.txt usando o comando cat
2. O comando “grep” é extremamente poderoso para realizar busca de conteúdo dentro de arquivos. Por exemplo a linha a seguir procura por ocorrências de “olá, mundo” dentro dos arquivos menu.h e main.c ignorando a diferença entre letras maiúsculas e minúsculas.

grep -i "olá, mundo" menu.h main.c

Crie um script questao11.sh que seja capaz de exibir as ocorrências da palavra “wonderland” (ou maiúsculo ou minúsculo) no arquivo 19033.txt que você baixou na questão 6.

1. O operador “|” (lê-se pipe) é muito usado em shell script quando se deseja encadear mais de um comando em uma só linha. Por exemplo, a saída de um comando é usada como parâmetro para outro comando. Assim, podemos identificar se no diretório atual existe por exemplo a pasta “Downloads”, associando os comandos ls e grep da sequinte forma:

ls | grep Downloads

Use o encadeamento do “|” para descobrir se o arquivo 19033.txt se encontra no diretório atual. Redirecione o resultado para “questao12.txt”

1. ~~Resultados de comandos podem ser armazenados em variáveis. Basta que você atribua o comando a uma variável. Porém, lembre-se de cercar o comando com acentos graves (``). Por exemplo. Se você colocar no shell script a seguinte linha:~~

~~var=~~**~~`~~**~~ls | grep Downloads~~**~~`~~**

~~O resultado do comando “ls | grep Downloads” será armazenado em var.~~

~~Crie um script “questao13.sh” que verifique se o diretório “Downloads” está presente na pasta corrente. Se sim, deve ecoar a frase “Diretório existe!”; caso contrário deve ecoar “Diretório inexistente”.~~

1. Crie um script “questao14.sh” que seja capaz de baixar o arquivo 19033.txt apenas se ele não existir na pasta atual. Descubra na internet como verificar a existência de um arquivo dentro de uma pasta. Caso ele já exista, ecoe a mensagem “Arquivo já baixado – economizando banda”.
2. Execute o comando ifconfig para descobrir quantas interfaces de rede o seu computador tem. Redirecione o resultado do comando para questao15.txt.
3. O uso de wildcards torna o gerenciamento de arquivos mais prático. Caso você queira listar os arquivos que você gerou até agora, digite no terminal:

ls questao\*

Caso você queira apenas listar os arquivos txt que você criou até agora, digite:

ls questao\*.txt

Liste os arquivos cujos nomes começam com “questao” e terminam com “5.txt” e redirecione a saída para o arquivo “questao16.txt”

1. A localização de arquivos pode ser feita no terminal com ajuda do comando find. Por exemplo, caso eu queira verificar no sistema inteiro onde está um arquivo ou pasta chamada “minhapasta” o comando deve ser executado da seguinte forma:

sudo find / -name minhapasta

Dessa vez, usamos o “sudo” para ter permissão de administrador e entrar em todas as pastas que forem necessárias. O “/” é o nome da pasta raiz do linux, ou seja equivale ao “[C:\](file:///C:/)” do windows. Caso o usuário não se lembre do nome exato, pode usar um wildcard:

sudo find / -name minha\*

Todo sistema linux possui um arquivo chamado “profile”. Realize a busca e coloque o resultado em um arquivo “questao17.txt”

1. O comando “top” lista os processos que estão rodando no sistema em tempo real. Abra o navegador firefox e verifique que ele aparece entre os processos. Para sair, use as teclas “Ctrl+C”. Agora, combine os comandos top e grep usando o operador “|” (pipe) para filtrar o processo “gnome-terminal” . Coloque o resultado no arquivo “questao18.txt”
2. Use o comando **history** para ver os 1000 últimos comandos já passados para o terminal. Coloque o resultado no arquivo “questao19.txt”
3. Crie uma pasta com o seu nome e mova todos os arquivos que começam com “questao” para ela (use o comando mv). Zipe essa pasta, gerando um arquivo tar.gz com o comando tar e envie o anexo para o google classroom:

tar cvfz seunome.tar.gz seunome/