1. Descubra para que servem os comandos abaixo e liste suas opções de uso mais comuns. Exemplo: \* ls função: serve para listar os arquivos do diretório corrente uso mais comum: $ ls -la, que lista todos os arquivos do diretório corrente, incluindo arquivos ocultos, se houver, no formato de lista.
   1. Cal

Texto

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

“Mostra um calendário e a data da Páscoa.”

* 1. Date

Texto

Descrição gerada automaticamente com confiança baixa



“Mostra ou muda a data e o horário do sistema.”

* 1. Clear

Texto

Descrição gerada automaticamente com confiança baixa

“Limpa a tela do terminal.”

* 1. Exit

Texto

Descrição gerada automaticamente

“Fecha o terminal.”

* 1. Uname

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa



“Mostra no terminal as informações do sistema que está sendo utilizado.”

1. Usando o comando adequado:

**Por razões de acesso, fiz algumas coisas na pasta corrente.**

* 1. descubra qual o seu diretório corrente ao logar em um terminal Linux



* 1. Navegue até a pasta /home



* 1. liste todos os arquivos da pasta /home



* 1. navegue até a pasta /bin



* 1. certifique-se de qual diretório você se encontra



* 1. a partir do diretório corrente, liste o conteúdo da pasta /etc

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

* 1. agora liste os arquivos da pasta corrente

Texto

Descrição gerada automaticamente

* 1. volte ao diretório home



* 1. crie uma pasta com o seu nome

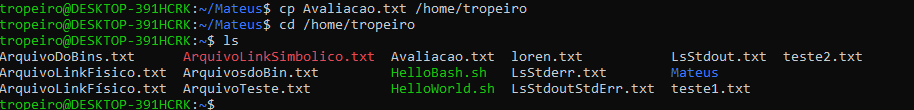


* 1. crie um arquivo texto dentro da pasta que você criou com o seu nome

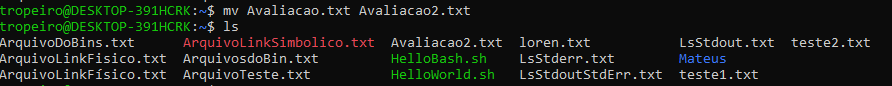
Texto

Descrição gerada automaticamente

* 1. copie este arquivo para a pasta /home.



* 1. renomeie o arquivo criado na pasta home



* 1. mova este arquivo que está na pasta que você criou com o seu nome para a pasta home

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

* 1. exclua o diretório que você criou com o seu nome

Tela de computador com texto preto sobre fundo verde

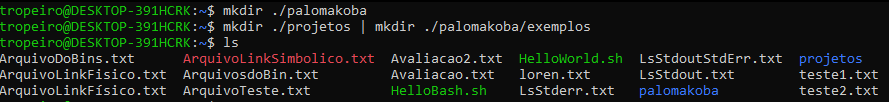
Descrição gerada automaticamente

1. Criar a seguinte estrutura de diretórios dentro do /home/seu\_usuario:

./palomakoba

./projetos

./palomakoba/exemplos



1. Entrar na pasta palomakoba.



1. Criar um arquivo chamado “numeros.txt”, usando o comando cat, contendo os seguintes números: 10 100 50 25 1 2

Texto

Descrição gerada automaticamente

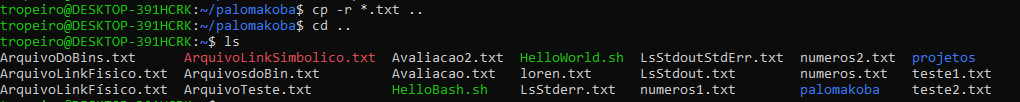


1. Duplicar o arquivo numeros.txt para numeros1.txt e numeros2.txt

Texto

Descrição gerada automaticamente

1. Copiar os arquivos com extensão .txt para a pasta trabalho no home do usuário



1. Exibir todos os arquivos com seus detalhes ( permissões de acesso, data, hora de criação, tamanho)

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

1. Mudar a permissão de acesso do arquivo numeros.txt para –rwxr-xr-x



1. Mudar a permissão de acesso do arquivo numeros.txt para -rw-r—r—



1. Crie a pasta palomakobabackup no home do usuário e copie o conteúdo da pasta palomakoba, para dentro do /home/seu\_usuario/palomakobabackup



1. Deletar os arquivos com extensão .txt

Texto

Descrição gerada automaticamente

1. Apagar a pasta textos que está dentro de palomakoba

Texto

Descrição gerada automaticamente

1. Entrar na pasta /home/seu\_usuario/palomakoba



1. Renomear o arquivo numeros.txt para sequencia.txt



1. Listar todos os arquivos da pasta /bin e guardar essa lista em um arquivo chamado “listabin.txt”

Texto

Descrição gerada automaticamente

1. Qual é o comando que apaga uma pasta vazia?

R= rmdir

1. Identifique a data atual e salve esta informação no diretório data em um arquivo chamado “data-atual.txt”;

Texto

Descrição gerada automaticamente

1. Crie um diretório usando seu primeiro nome como nome do diretório;



1. Renomeie o diretório com seu primeiro nome para seu nome-sobrenome;



1. Dentro do diretório nome-sobrenome, crie 3 diretórios: documentos, imagens e musicas; \*Use o comando mkdir, porém, estruture o comando para criar os 3 diretórios ao mesmo tempo.



1. Liste o conteúdo do diretório raíz do Linux (o “/“) em forma de lista vertical e, após, salve estas informações em um arquivo chamado “ls-root.txt”;

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

1. Copie os arquivos **/etc/passswd, /etc/group** e **/etc/protocols** para o diretório documentos;
2. Conte o número de linhas e palavras do arquivo passwd;

Tela de celular

Descrição gerada automaticamente com confiança baixa

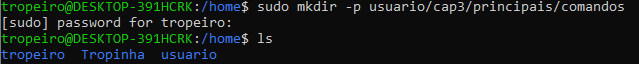
1. Identifique o tipo do arquivo passwd; *\*Use o comando* ***file****.*

**

1. Identifique somente o usuário root no arquivo passwd; *\*Use o comando* ***cat*** *e, junto dele, o comando* ***grep****.*

**

1. Liste o conteúdo do diretório exercicio-so-1 em forma de lista, incluindo seus subdiretórios e, após, salve estas informações em um arquivo chamado “ls-exercicio-so-1.txt”; *\*Use o comando* ***ls -lhR****.*
2. A seguinte estrutura de diretórios deve ser criada: “/ -> home -> usuario -> cap3 -> principais -> comandos”. Como podemos criar tal estrutura com apenas um comando?



1. Qual comando irá exibir as últimas linhas de um arquivo

/var/log/cups/access\_log?

**Não possuo este diretório, usei um diretório alternativo**

Texto

Descrição gerada automaticamente

1. Escreva em um arquivo a quantidade de arquivos com formato pdf salvos em um diretório.

Uma imagem contendo Texto

Descrição gerada automaticamente

1. Especifique o comando que é utilizado para dar permissão de leitura, escrita e execução de um arquivo a um usuário. E se fosse apenas leitura e escrita, qual seria o comando?

O comando para dar permissão de leitura, escrita e execução a um usuário é:

chmod u+rwx NomeDoArquivo

para apenas leitura e escrita é:

chmod u+rw NomeDoArquivo

1. Faça um bash shell script chamado *vars* para experimentar com variáveis.

Explique o funcionamento do script.

#!/bin/bash

# variaveis de ambiente echo $PATH

echo $USER $HOME

# pode-se ver todas as variáveis do ambiente com o comando env | less

#variaveis locais ola="bom dia" echo "$ola Paulo"

echo "$olaPaulo" $Texto Pegado a variavel ..

echo "${ola}Paulo" #proteger a variável com as chavetas mesg="$ola $USER"

echo $mesg

# input

echo "Introduza qualquer Coisa" ; read var echo "Introduziu $var"

# execução data=`date`

echo $data

info='echo $HOME ; echo " estamos em "; pwd' echo $info

# contas x=1

let x=x\*2+3 echo "x=$x" let x--

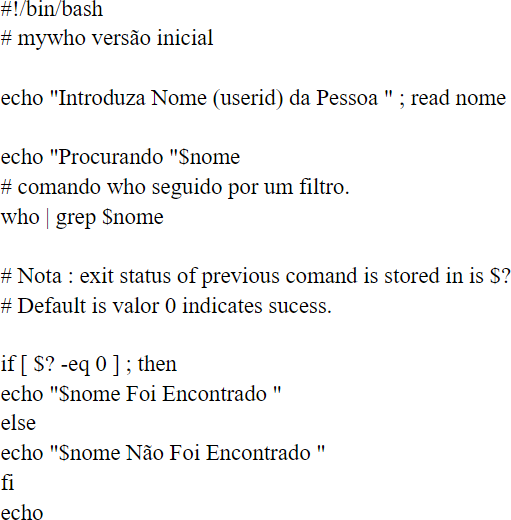
echo "x=$x"

#variaveis especiais

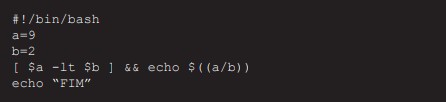
echo "Numero de Arguments para este script $#" echo "Todos os argumentos para este script $\*" echo "O primeiro $1 e segundo $2 argumentos"

echo "O nome deste ficheiro $0"echo "O Processo ID deste script $$" echo "Exit status do comando anterior $"”

R= Mostra o caminho para o path, usuário e home. Exibe um texto na tela dizendo ‘Olá, (nome do usuário)’, pede para introduzir um texto e exibe este texto logo em seguida. Busca a data e a exibe também. Realiza uma operação matemática e exibe o resultado. E busca por meio de variáveis especiais informações como numero de argumentos, o ID do arquivo .sh, o nome do arquivo.

1. Execute o script abaixo e explique o que ele faz. Qual a saída do programa?

R= Pede que escreva um nome, então ele procura o valor na variavel ‘nome’. Se o resultado ao buscar na variavel resultar em 0, é porque existe o nome. Senão, não existe o nome.

1. Faça um script que imprima quantos processos estão atualmente em execução na sua máquina. Use os comandos ps e wc para isso.
2. Crie um script para mostrar (cat) todos os usuários cadastrados no sistema (/etc/passwd/) ordenados em ordem alfabética.
3. Um dos parâmetros de cada linha do arquivo (/etc/passwd/) é o shell usado pelo usuário (o sétimo campo). Escreva um programa capaz de listar todos shells únicos existentes no passwd. O comando uniq pode ser útil.
4. Criar um programa que mostre o espaço utilizado pelos arquivos dentro de cada diretório da sua conta no sistema, colocando em ordem numérica o resultado. Use os comando du e sort.
5. Qual a saída do script abaixo? Explique o funcionamento!

R= São criadas duas variáveis, a e b, e então é questionado se o valor da esquerda ($a) é menor que o valor da direita ($b), se isto for verdade, então o código executa a operação matemática e printa “FIM”. Senão, ele não faz nada e apenas printa “FIM”.

1. Qual a saída do script abaixo? Explique o funcionamento!

#!/bin/bash x=0

while [ $x -le 2 ] do

echo valor: $x ((x++))

done

R= Inicializa uma variável como 0, enquanto x for menor que 2, printa o valor de x e incrementa +1 em X. E então, a saída é:

X = 0

X = 1

X = 2

1. Qual a saída do script abaixo? Explique o funcionamento!

#!/bin/bash

echo "Resultado..." for i in {0..12..2}

do a=$(($i%3)) if [ $a -eq 0 ] then

echo "Alo valor $i" ﬁ

done

R= Incrementa i, e pega o resto da divisão dos números. Se este resto for igual a 0, imprime o valor. A saida é:

Resultado...

Alo valor 0

Alo valor 6

Alo valor 12