

Introdução à Ciência da Computação

Shell Script – parte V

Professor Iago Augusto de Carvalho
iago.carvalho@unifal-mg.edu.br

Comando read

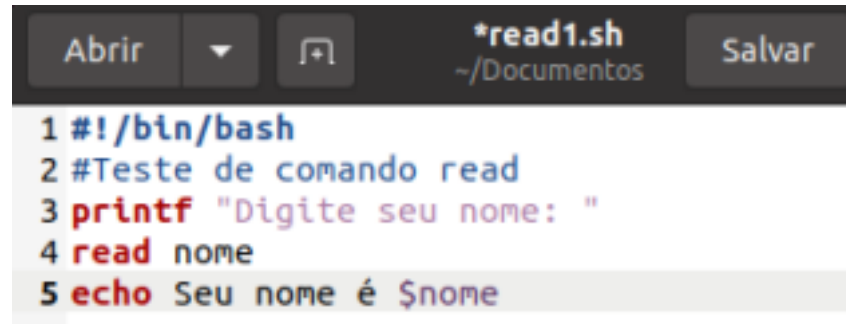
O comando read aceita entrada de dados da entrada padrão ou de outro descritor de arquivos.

Após receber

a entrada, o

comando

read coloca os dados em uma variável padrão.



```
1 #!/bin/bash
2 #Teste de comando read
3 printf "Digite seu nome: "
4 read nome
5 echo "Seu nome é $nome"
```

Exemplo:

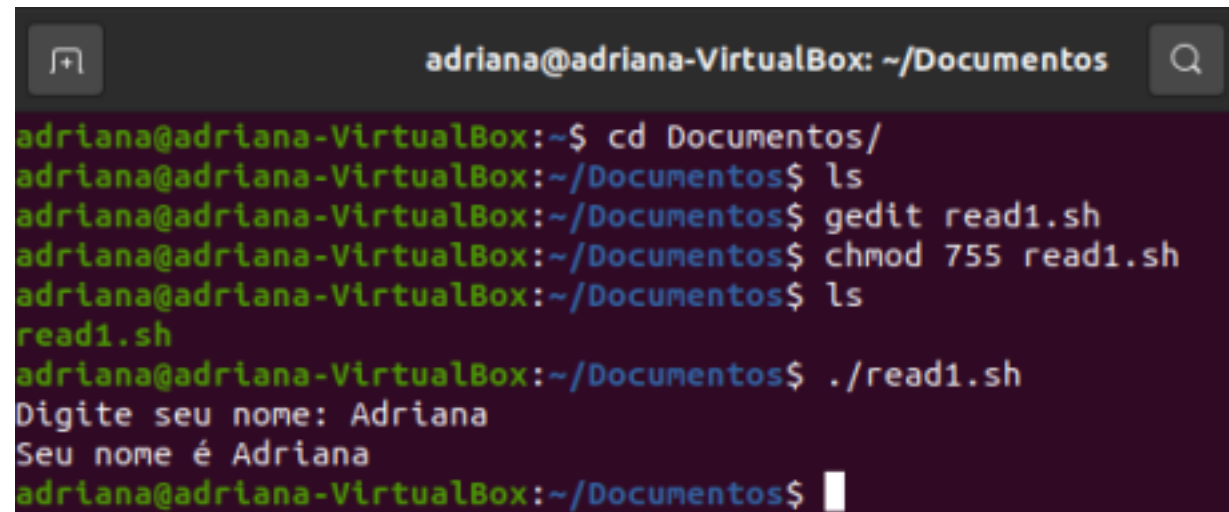
```
#!/bin/bash
```

```
#Teste de comando read
```

```
printf "Digite seu nome: "
```

```
read nome
```

```
echo "Seu nome é $nome"
```

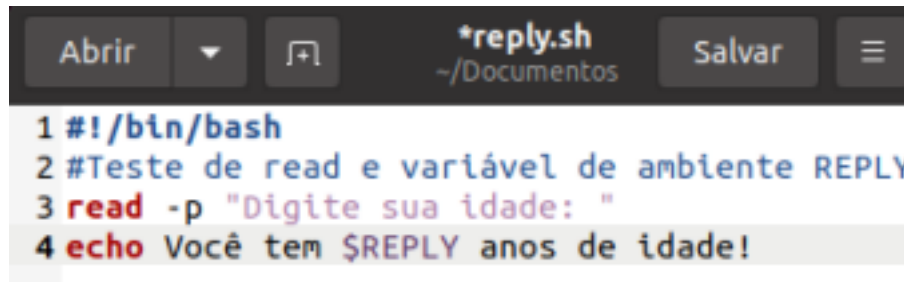


```
adriana@adriana-VirtualBox: ~$ cd Documentos/
adriana@adriana-VirtualBox: ~/Documentos$ ls
adriana@adriana-VirtualBox: ~/Documentos$ gedit read1.sh
adriana@adriana-VirtualBox: ~/Documentos$ chmod 755 read1.sh
adriana@adriana-VirtualBox: ~/Documentos$ ls
read1.sh
adriana@adriana-VirtualBox: ~/Documentos$ ./read1.sh
Digite seu nome: Adriana
Seu nome é Adriana
adriana@adriana-VirtualBox: ~/Documentos$
```

Comando read

Se não for especificada nenhuma variável para receber os dados no comando read, então o valor lido será colocado na variável de ambiente REPLY:

```
#!/bin/bash
#Teste de read e variável de ambiente
REPLY
read -p "Digite sua idade: "
echo Você tem $REPLY anos de
idade!
```

A screenshot of a terminal window with a dark background. The window title bar shows 'Abrir', a dropdown arrow, a file icon, and the filename '*reply.sh' with the path '~/Documentos'. The terminal content shows four lines of a script: line 1 is '#!/bin/bash' in blue; line 2 is '#Teste de read e variável de ambiente REPLY' in blue; line 3 is 'read -p "Digite sua idade: "' in red and purple; line 4 is 'echo Você tem \$REPLY anos de idade!' in red and purple. The lines are numbered 1 through 4 on the left.

```
1 #!/bin/bash
2 #Teste de read e variável de ambiente REPLY
3 read -p "Digite sua idade: "
4 echo Você tem $REPLY anos de idade!
```

```
adriana@adriana-VirtualBox: ~/Documentos
adriana@adriana-VirtualBox:~/Documentos$ gedit reply.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod a+x reply.sh
adriana@adriana-VirtualBox:~/Documentos$ ./reply.sh
Digite sua idade: 26
Você tem 26 anos de idade!
adriana@adriana-VirtualBox:~/Documentos$
```

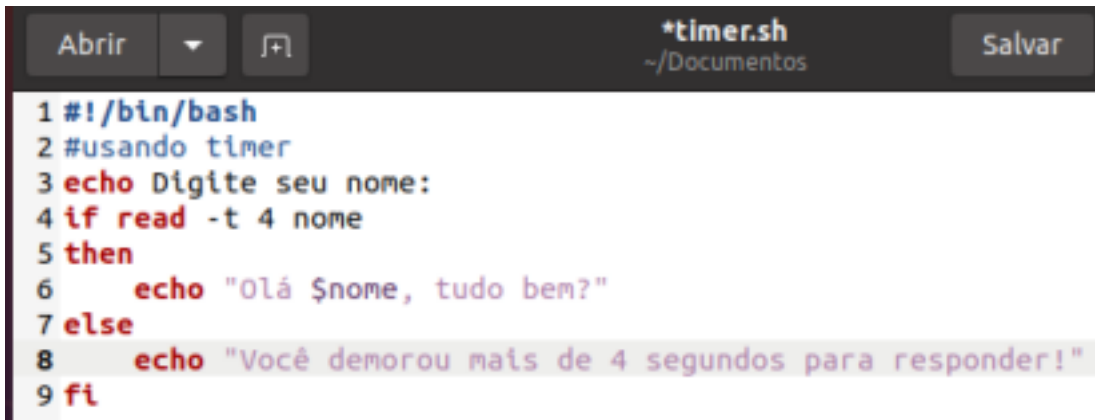
Comando read com timer

É possível especificar um valor de timeout para o comando read. Para isso, usamos a opção `-t` para passar o número de segundos que o comando aguardará uma entrada do usuário. Quando o timer expira, o read retorna status de saída não-zero.

```
#!/bin/bash
#Usando timer
echo Digite seu nome:
if read -t 4 nome
then
```

```
echo Olá $nome, tudo bem?  
else  
    echo Você demorou mais de 4 segundos para responder!  
fi
```

Comando read com timer

A screenshot of a code editor window. The title bar shows the filename as *timer.sh and the location as ~/Documentos. The editor contains a shell script with the following lines:

```
1 #!/bin/bash  
2 #usando timer  
3 echo Digite seu nome:  
4 if read -t 4 nome  
5 then  
6     echo "Olá $nome, tudo bem?"  
7 else  
8     echo "Você demorou mais de 4 segundos para responder!"  
9 fi
```

```
adriana@adriana-VirtualBox: ~/Documentos
adriana@adriana-VirtualBox:~/Documentos$ gedit timer.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod 755 timer.sh
adriana@adriana-VirtualBox:~/Documentos$ ./timer.sh
Digite seu nome:
Você demorou mais de 4 segundos para responder!
adriana@adriana-VirtualBox:~/Documentos$
```

```
adriana@adriana-VirtualBox: ~/Documentos
adriana@adriana-VirtualBox:~/Documentos$ ./timer.sh
Digite seu nome:
Adriana
Olá Adriana, tudo bem?
adriana@adriana-VirtualBox:~/Documentos$
```

Comando read

Limitando número de caracteres digitados

É possível também fazer com que o comando read conte o número de caracteres de entrada, e saia imediatamente quando um número predefinido for digitado, atribuindo os dados recolhidos a uma variável.

```
#!/bin/bash
```

```
#Lendo um caracter (-n1 significa um caracter lido)
echo Deseja prosseguir com a operação [s/n]?
read -n1 resposta
case $resposta in
    S | s ) echo
                echo Você escolheu continuar;;
    N | n ) echo
                echo Operação interrompida;;

    * ) echo
                echo Opção inválida. Digite apenas s ou n;;
esac
```

Comando read

Limitando número de caracteres digitados

```
Abrir  ▾  [+]
```

*read2.sh
~/Documentos

Salvar

```
1 #!/bin/bash
2 #Lendo um caracter (-n1 significa um caracter lido)
3 echo "Deseja prosseguir com a operação [s/n]?"
4 read -n1 resposta
5 case $resposta in
6     S | s) echo
7             echo Você escolheu continuar;;
8     N | n) echo
9             echo Operação interrompida;;
10    *) echo
11        echo Opção inválida. Digite apenas s ou n;;
12 esac
```

```
[+]  adriana@adriana-VirtualBox: ~/Documentos  🔍
```

```
adriana@adriana-VirtualBox:~/Documentos$ gedit read2.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod a+x read2.sh
adriana@adriana-VirtualBox:~/Documentos$ ./read2.sh
Deseja prosseguir com a operação [s/n]?
s
Você escolheu continuar
adriana@adriana-VirtualBox:~/Documentos$ ./read2.sh
Deseja prosseguir com a operação [s/n]?
n
Operação interrompida
adriana@adriana-VirtualBox:~/Documentos$ ./read2.sh
Deseja prosseguir com a operação [s/n]?
y
Opção inválida. Digite apenas s ou n
adriana@adriana-VirtualBox:~/Documentos$
```


Comando read

Leitura silenciosa de dados

É possível ler o que o usuário digita, sem que apareçam os caracteres na tela. Esse recurso é útil, por exemplo, para capturarmos senhas. Para isso, usamos a opção -s:

```
#!/bin/bash
```

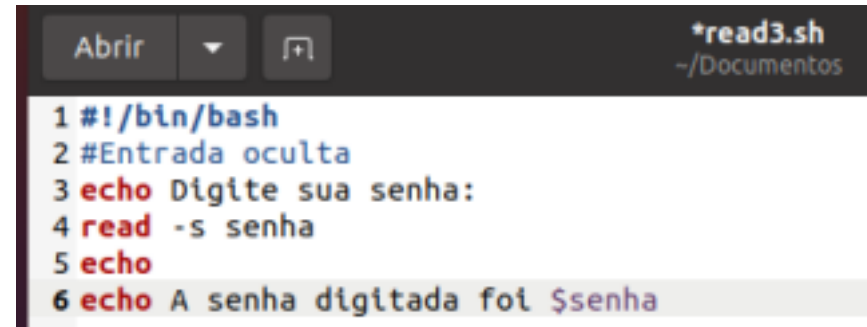
```
#Entrada oculta
```

```
echo Digite sua senha
```

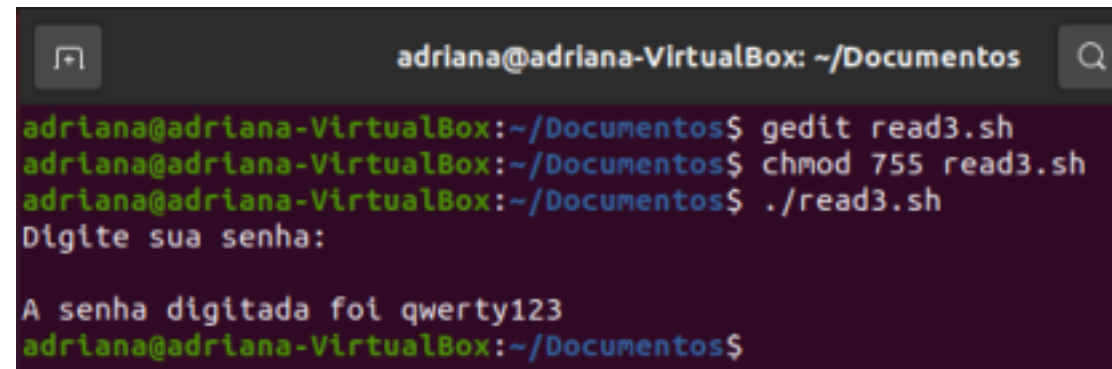
```
read -s senha
```

```
echo
```

```
echo A senha digitada foi $senha
```

A screenshot of a code editor window titled '*read3.sh' in the top right corner, with the file path '~/Documentos' below it. The editor contains six lines of code: 1 '#!/bin/bash', 2 '#Entrada oculta', 3 'echo Digite sua senha:', 4 'read -s senha', 5 'echo', and 6 'echo A senha digitada foi \$senha'. The lines are numbered on the left side of the editor.

```
1 #!/bin/bash
2 #Entrada oculta
3 echo Digite sua senha:
4 read -s senha
5 echo
6 echo A senha digitada foi $senha
```

A screenshot of a terminal window with the title 'adriana@adriana-VirtualBox: ~/Documentos'. The terminal shows the execution of the script: the user runs 'gedit read3.sh', then 'chmod 755 read3.sh', and finally './read3.sh'. The script's output is shown: 'Digite sua senha:' followed by a prompt where 'qwerty123' was entered (characters are not visible), and then 'A senha digitada foi qwerty123'. The prompt returns to the shell.

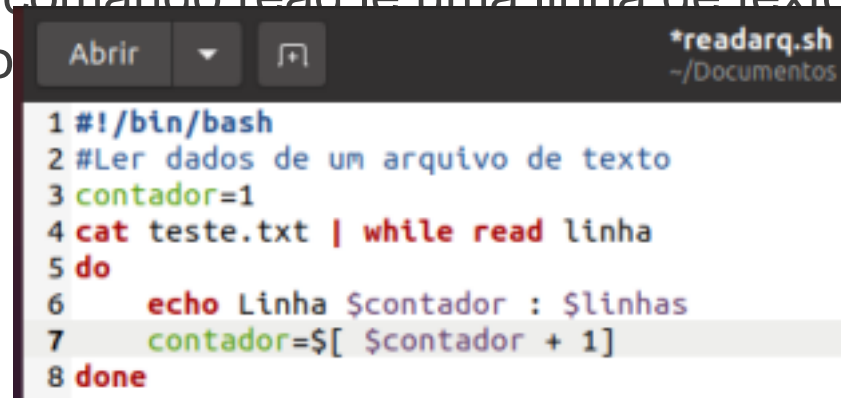
```
adriana@adriana-VirtualBox:~/Documentos$ gedit read3.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod 755 read3.sh
adriana@adriana-VirtualBox:~/Documentos$ ./read3.sh
Digite sua senha:
A senha digitada foi qwerty123
adriana@adriana-VirtualBox:~/Documentos$
```

Comando

read Ler dados de um arquivo

Podemos usar o comando read para ler dados armazenados em um arquivo no computador. Cada chamada ao comando read lê uma linha de texto. Quando não houverem mais linhas para ler, o comando saída zero.

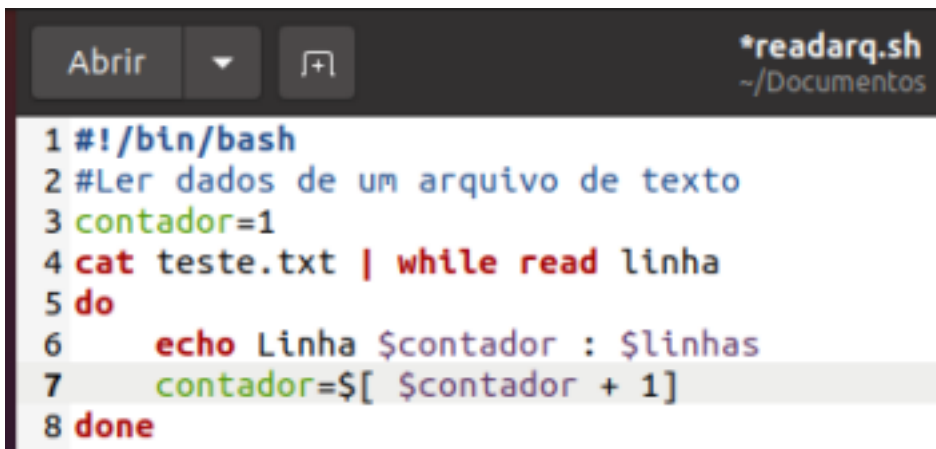
```
#!/bin/bash
#Ler dados de um arquivo de
texto:
contador=1
cat teste.txt | while read linha
do
    echo Linha $contador : $linha
    contador=$(( $contador + 1 ))
done
```



```
1 #!/bin/bash
2 #Ler dados de um arquivo de texto
3 contador=1
4 cat teste.txt | while read linha
5 do
6     echo Linha $contador : $linha
7     contador=$(( $contador + 1 ))
8 done
```

Comando read

Ler dados de um arquivo



```
1 #!/bin/bash
2 #Ler dados de um arquivo de texto
3 contador=1
4 cat teste.txt | while read linha
5 do
6     echo Linha $contador : $linhas
7     contador=$(( $contador + 1 ])
8 done
```

```
adriana@adriana-VirtualBox: ~/Documentos
adriana@adriana-VirtualBox:~/Documentos$ gedit readarq.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod 755 readarq.sh
adriana@adriana-VirtualBox:~/Documentos$ ./readarq.sh
Linha 1 : Castelo De Areia
Linha 2 : Num dia de verão, estava na praia,
Linha 3 : observando duas crianças brincando na
Linha 4 : areia. Elas trabalhavam muito,
Linha 5 : construindo um castelo de areia, com torres,
Linha 6 : passarelas e passagens internas.
Linha 7 : Quando estavam quase acabando, veio uma
Linha 8 : onda e destruiu tudo, reduzindo o
Linha 9 : castelo a um monte de areia e espuma.
Linha 10 : Achei que, depois de tanto esforço e cuidado,
Linha 11 : as crianças caíam no choro.
Linha 12 : Mas tive uma surpresa.
adriana@adriana-VirtualBox:~/Documentos$
```

Funções

```
function nome_função {  
  comandos da função }
```

1ª sintaxe: 2ª

```
nome_função() {  
  comandos da função }
```

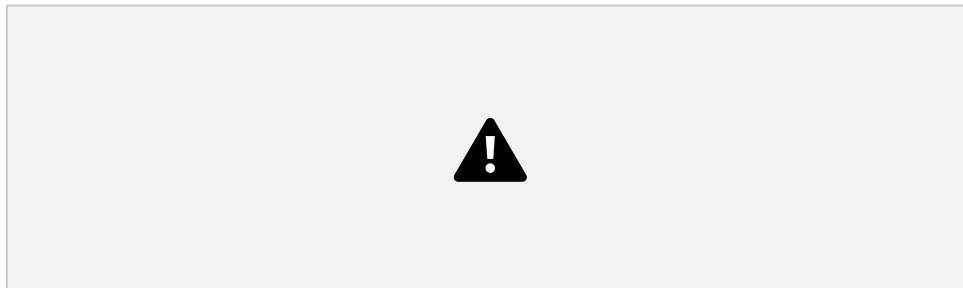
sintaxe:

Funções

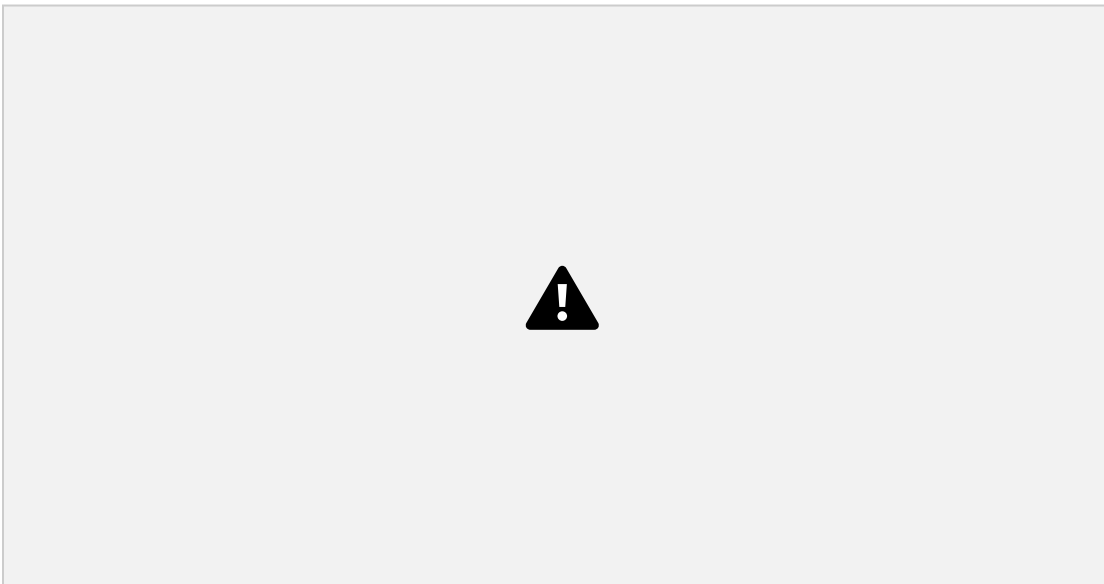


```
func.sh
1 #!/bin/bash
2 #Função em um script so shell
3
4 function mensagem {
5     echo "Aula ICC"
6 }
7
8 #Chamando a função dez vezes usando um contador:
9 contador=1
10 while [ $contador -le 10 ]
11 do
12     mensagem
13     contador=$(( contador + 1 ))
14 done
15
16 #Outro texto, indicando o fim do script:
17 echo "Função executada com sucesso!"
```

Funções



Nome da função entre crases (backtick`)



Caixas de diálogo com whiptail



Linhas

Colunas



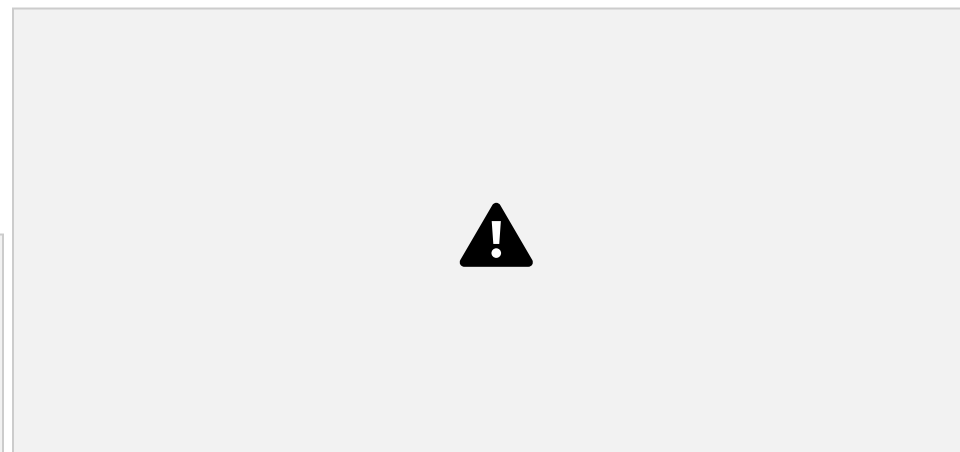
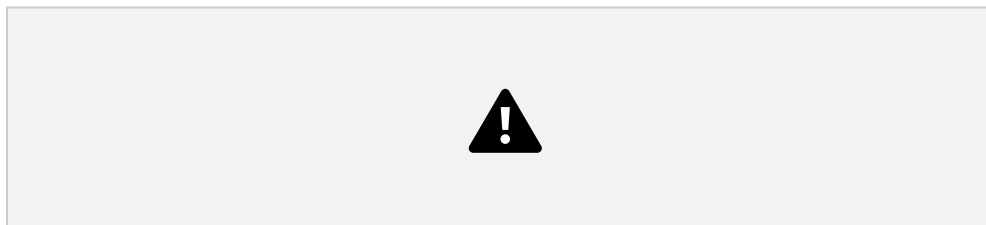
Pressione a tecla enter para voltar

Caixas de diálogo com whiptail



Utilize as teclas e e pressione a tecla enter.

Caixas de diálogo com whiptail



A opção - **fb** permite o uso de botões completos (por padrão, o whiptail usa uma versão simplificada dos botões). Utilize as teclas e e pressione a tecla enter.

Caixas de diálogo com whiptail



Utilize as teclas e e pressione a tecla enter.

Caixas de diálogo com whiptail



Utilize as teclas e e pressione a tecla enter.

Caixas de diálogo com whiptail





Caixas de diálogo com whiptail





Utilize as teclas , e para mover.

Caixas de diálogo com whiptail



O que significa 3>&1 1>&2 2>&3 em um script do Shell no Linux?

3>&1 – Cria um novo descritor de arquivos e o redireciona para a saída

padrão (1) **1>&2** – Redireciona a saída padrão para o erro padrão (2)

2>&3 – Redireciona o erro padrão para o descritor criado, que agora irá redirecionar para a saída

padrão. <http://www.bosontreinamentos.com.br/shell-script/o-que-significa-31-12-23-em-um-script-do-shell-no-linux/>



com whiptail

Londres é tag
Inglaterra é item

As \ indicam que continua na linha debaixo.

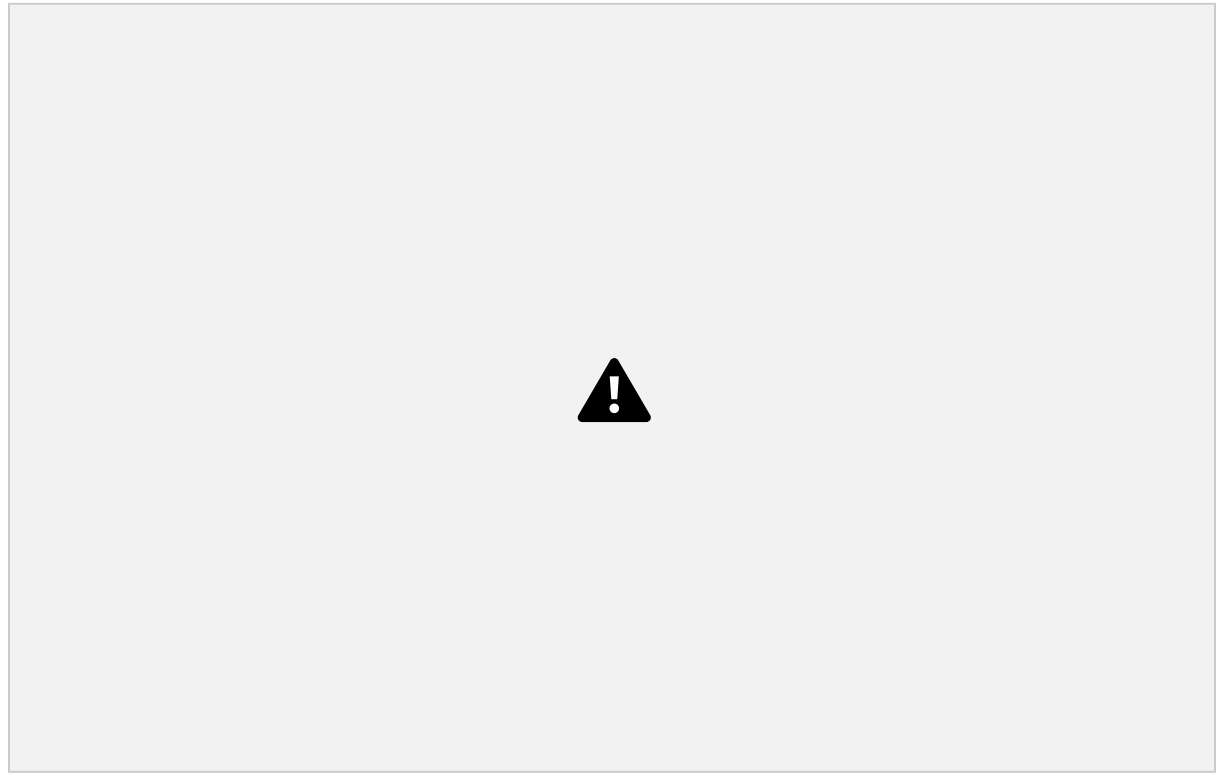
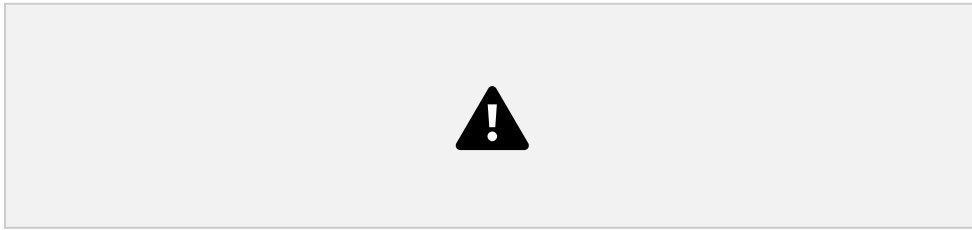


Caixas de diálogo com whiptail



Utilize as teclas , e para mover. E espaço para selecionar.

Caixas de diálogo com whiptail



Utilize a tecla tab para mover.

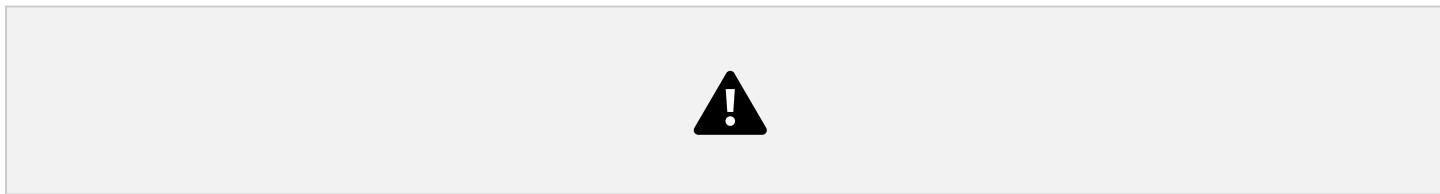
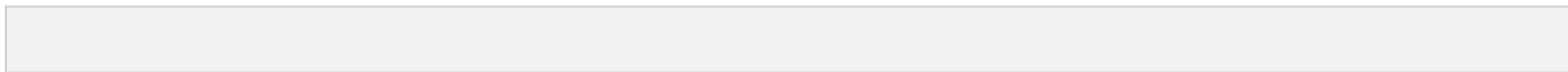
Caixas de

diálogo com
dialog

O comando `dialog` utiliza parâmetros de linha de comando para determinar que tipo de widget de janela deve ser criada. Um widget é um tipo de elemento de janela.

`inputmenu` Fornece um menu editável

Caixas de diálogo com dialog





Caixas de diálogo com dialog



Caixas de diálogo com dialog



Caixas de diálogo com dialog





Caixas de diálogo com dialog





30

Caixas de diálogo com dialog





Caixas de diálogo com dialog



Caixas de diálogo com dialog





Referências

PRITCHARD, S.; PESSANHA, B. G.; LANGFELDT, N.; STANGER, J.; DEAN, J. 2007. **Certificação Linux LPI Rápido e Prático. Guia de Referência nível 1: Exames 101 e 102.** 2ª Ed. Rio de Janeiro: Editora Alta Books.

Curso de Shell Scripting – Bóson Treinamentos

<http://www.bosontreinamentos.com.br/curso-de-shell-scripting/>