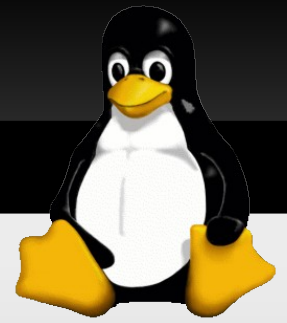


Shell Script - Básico

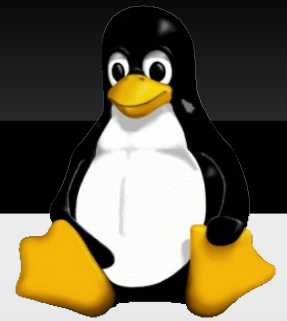
*Thomaz de Oliveira dos Reis - **thommy@globo.com***



“Pra programar Shell não é preciso Texaco”
- *Julio Cezar Neves*

Shell Script - Básico

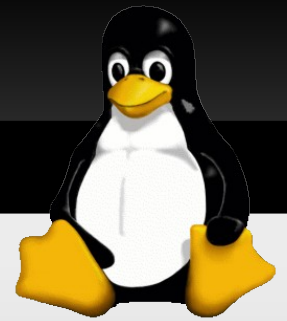
*Thomaz de Oliveira dos Reis - **thommy@globo.com***



- Como o Shell funciona
- Comandos interessantes
- Programando em shell

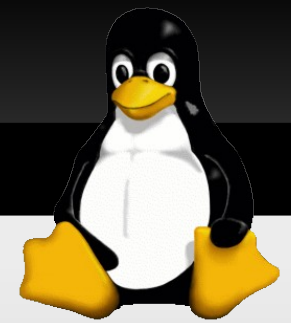
Shell Script - Básico

*Thomaz de Oliveira dos Reis - **thommy@globo.com***



- Como o Shell funciona
- Comandos interessantes
- Programando em shell

Shell Script - Básico



Thomaz de Oliveira dos Reis - ***thommy@globo.com***

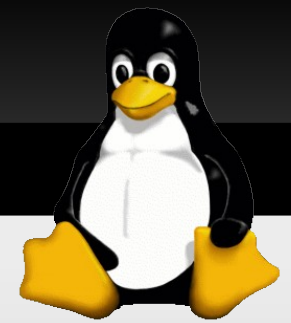
Como o Shell funciona?

“In computing, a shell is a piece of software that provides an interface for users.”

- Wikipedia

Tradução Livre: Na computação, um shell é um pedaço de programa que provê uma interface com o usuário.

Shell Script - Básico



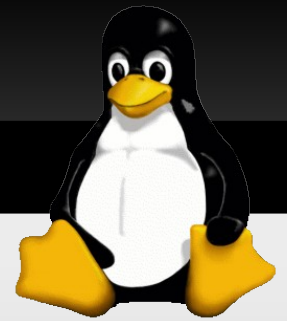
*Thomaz de Oliveira dos Reis - **thommy@globo.com***

Como o Shell funciona?

1. Exame da linha de comando recebida
2. Resolução de Redirecionamentos
3. Substituição de variáveis
4. Substituição de meta caracteres
5. Passa o comando para o kernel

Shell Script - Básico

Thomaz de Oliveira dos Reis - thommy@globo.com

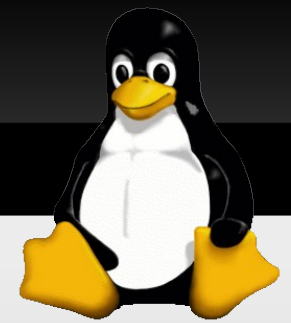


1. Exame da linha de comando recebida

Comando: Separado por espaço em branco, sendo o primeiro o comando e o restante os parâmetros **OBS: Remove todos os espaços em brancos repetidos!**

Atribuição: Separado por “=” sem espaços em branco

Shell Script - Básico



Thomaz de Oliveira dos Reis - thommy@globo.com

2. Resolução de Redirecionamentos

">" e ">>": Redireciona saída padrão para um arquivo

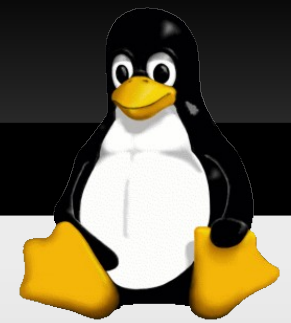
"2>" e "2>>": Redireciona saída de erro para um arquivo

"|": Redireciona a saída padrão para a entrada de outro programa

"tee": Redireciona o pipe ("|") para a saída padrão

"<" e "<<": Redireciona a entrada padrão

Shell Script - Básico



Thomaz de Oliveira dos Reis - thommy@globo.com

3. Substituição de variáveis

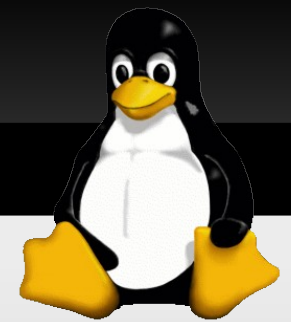
substitui todos os \$Variavel pelo conteúdo da variável.

ex:

```
echo $PATH  
echo \ $PATH  
echo '$PATH'  
echo "$PATH"
```


Shell Script - Básico

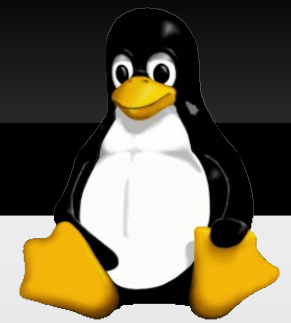
*Thomaz de Oliveira dos Reis - **thommy@globo.com***



Uma paradinha para explicar \, ' e “:

- \ : Ignora o próximo character
- ': ignora o conteúdo entre dois '
- “: considera tudo como apenas um parametro

Shell Script - Básico



*Thomaz de Oliveira dos Reis - **thommy@globo.com***

4. Substituição de meta caracteres

Para arquivos:

*: número qualquer de letras, e qualquer letra

?: apenas uma letra qualquer

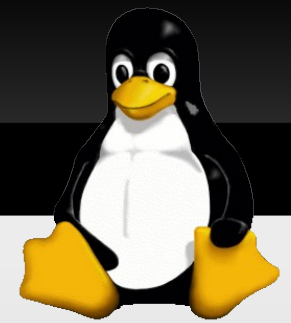
. : Pasta atual

.. : Pasta anterior

~: Home do usuário

`: Abre um shell filho, e executa o comando que está entre `.

Shell Script - Básico



Thomaz de Oliveira dos Reis - thommy@globo.com

5. Passa o comando para o kernel

!!!ATENÇÃO:!!!

!!!Próximas informações apenas para geeks!!!

O Shell envia o comando para o kernel, que por sua vez executa um fork, criando um processo filho na memória, e então executa o programa determinado.

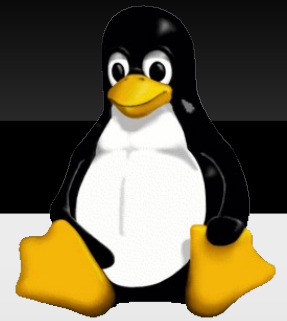
Para os não (muito) geeks:

O Shell manda o SO executar o comando

OBS: Caso o comando seja um script, todas as modificações a variáveis feitas dentro dele não alteram o shell principal.

Shell Script - Básico

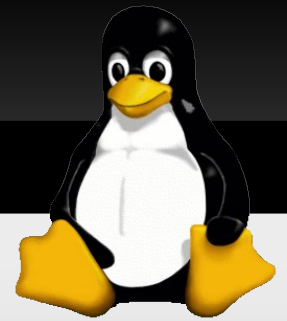
*Thomaz de Oliveira dos Reis - **thommy@globo.com***



- Como o Shell funciona
- Comandos interessantes
- Programando em shell

Shell Script - Básico

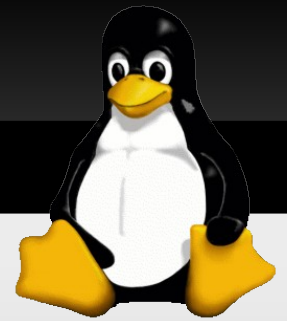
*Thomaz de Oliveira dos Reis - **thommy@globo.com***



- ★ Como o Shell funciona
- ➡ Comandos interessantes
- Programando em shell

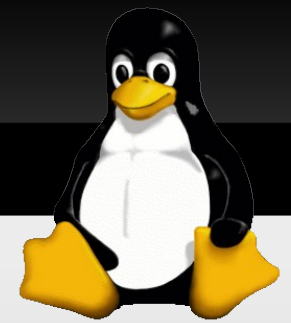
Shell Script - Básico

Thomaz de Oliveira dos Reis - ***thommy@globo.com***



- 1.cat, more e less
- 2.wc
- 3.sort
- 4.head e tail
- 5.find
- 6.grep
- 7.sed
- 8.cut e paste
- 9.tr
- 10.dialog

Shell Script - Básico



Thomaz de Oliveira dos Reis - thommy@globo.com

1. **cat, more, less**

Serve para mostrar o conteúdo de arquivos

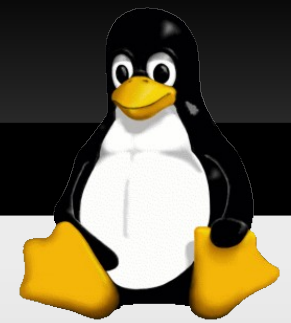
cat: Joga todo o conteúdo do arquivo para a saída padrão

more: Joga o conteúdo do arquivo até preencher 1 tela, e espera a tecla enter para ir para a próxima linha. Possui recursos de busca e outros estilo “vi”.

less: O mesmo do more só que pode voltar...

Shell Script - Básico

Thomaz de Oliveira dos Reis - thommy@globo.com



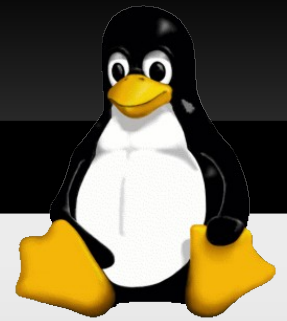
2. **wc**

Mostra o número de linhas, palavras e bytes de um arquivo.

- l:** mostra apenas o número de linhas
- c:** mostra apenas o número de bytes
- m:** mostra apenas o número de letra
- w:** mostra apenas o número de palavras

Shell Script - Básico

*Thomaz de Oliveira dos Reis - **thommy@globocom***

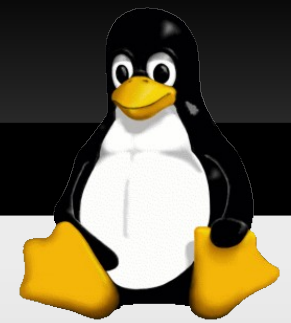


3. sort

Mostra o conteúdo de um arquivo em ordem crescente ou alfabética

Shell Script - Básico

Thomaz de Oliveira dos Reis - thommy@globo.com



4. head e tail

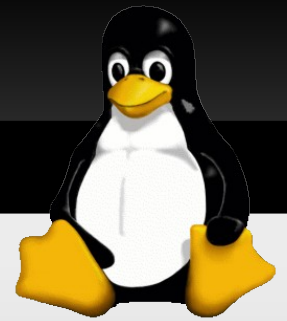
Mostra as primeiras ou ultimas linhas de um arquivo

head -x: mostra as x primeiras linhas do arquivo

tail -x: mostra as x ultimas linhas do arquivo

Shell Script - Básico

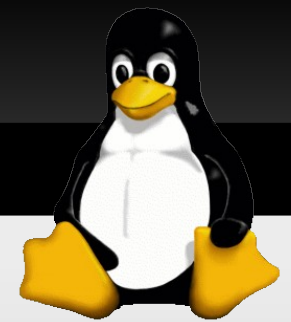
*Thomaz de Oliveira dos Reis - **thommy@globo.com***



Os três próximos comandos (find, grep e sed), são EXTREMAMENTE poderosos, e vale o aprofundamento do estudo deles.

Shell Script - Básico

Thomaz de Oliveira dos Reis - thommy@globo.com



5. find

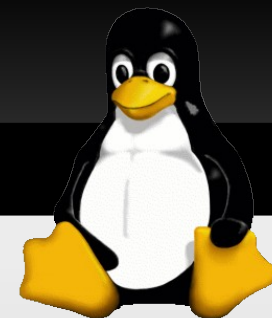
Procura arquivos por características
find caminho expressão ação

expressões:

- name** nome-do-arquivo
- user** nome-do-usuário
- group** nome-do-grupo
- type** tipo-do-arquivo
(*d: diretório, f: arquivo, l: link, p: pipe*)

Shell Script - Básico

Thomaz de Oliveira dos Reis - thommy@globo.com



5. find

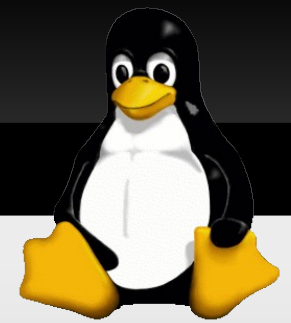
Procura arquivos por características
find caminho expressão ação

ações:

- print** (ação padrão)
- exec** comando {} \;
- ok** comando {} \;

Shell Script - Básico

Thomaz de Oliveira dos Reis - thommy@globo.com

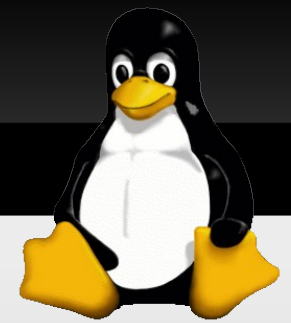


6. grep

Mostra as linhas dos arquivos que casam com uma expressão regular

- i** ignora maiúsculas e minúsculas
- v** inverte o resultado
- R** procura recursivamente

Shell Script - Básico



Thomaz de Oliveira dos Reis - thommy@globo.com

7. sed

*Editor de arquivos completo
Muito usado para substituir cadeias de
caracteres*

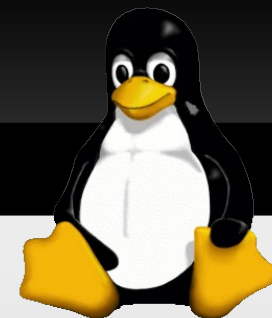
ex:

sed 's/original/final/g' arquivo

Troca todas as cadeias de caracteres iguais a
original por final

Shell Script - Básico

Thomaz de Oliveira dos Reis - thommy@globo.com



8. cut e paste

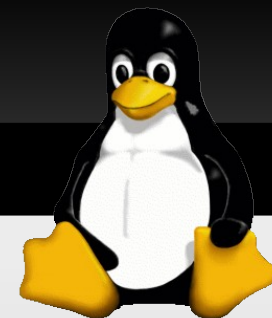
cut: separa o arquivos em partes

paste: junta arquivos

- d:** qual o caractere que separa as partes
- f:** qual das partes deve ser mostrada

Shell Script - Básico

Thomaz de Oliveira dos Reis - ***thommy@globo.com***



9. tr

substitui caracteres

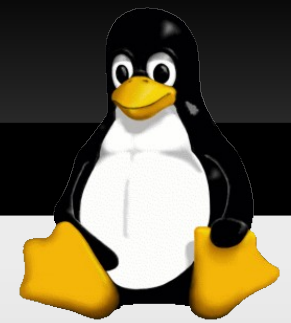
`[:upper:]` letras em maiusculo

`[:lower:]` letras em minusculo

`[:digit:]` digitos

Shell Script - Básico

Thomaz de Oliveira dos Reis - thommy@globo.com



10. dialog

Mostra uma “janela”, com mensagens, entradas e outros

dialog –tipo “texto” X Y

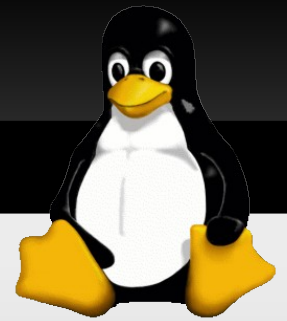
onde tipo pode ser: msgbox, inputbox e outros

X e Y é o tamanho da “janela”

tem também o Xdialog (100% compatível e gráfico), gdialog (usa gtk) e kdialog (usa qt, e possui diversos recursos diferentes)

Shell Script - Básico

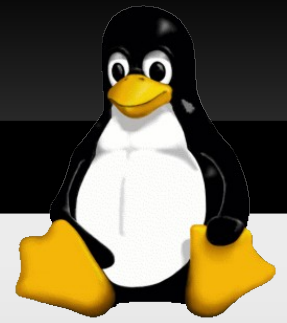
*Thomaz de Oliveira dos Reis - **thommy@globo.com***



- ★ Como o Shell funciona
- ➦ Comandos interessantes
- Programando em shell

Shell Script - Básico

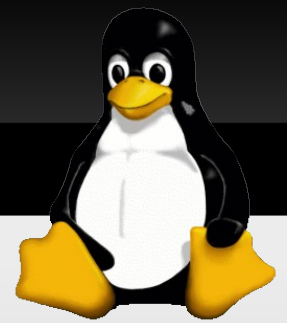
*Thomaz de Oliveira dos Reis - **thommy@globo.com***



- ★ Como o Shell funciona
- ★ Comandos interessantes
- ➔ Programando em shell

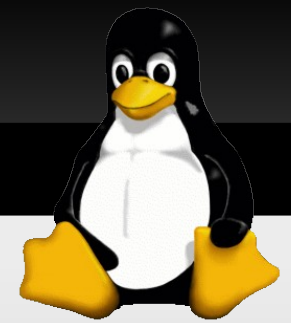
Shell Script - Básico

*Thomaz de Oliveira dos Reis - **thommy@globo.com***



1. “Hello World” em Shell Script
2. Cabeçalho
3. Comentários
4. Condicional
5. Pegando informação do teclado
6. Estruturas de repetição

Shell Script - Básico



Thomaz de Oliveira dos Reis - ***thommy@globo.com***

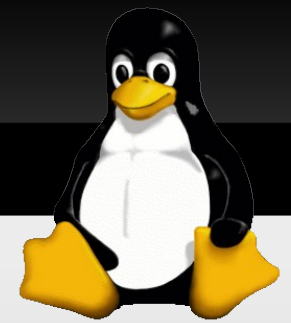
1. Hello World em Shell Script

```
$ [vi|mcedit|kwrite|editor preferido] hello.sh
```

```
1 echo Hello World
```

```
$ chmod +x hello.sh  
$ ./hello.sh
```

Shell Script - Básico



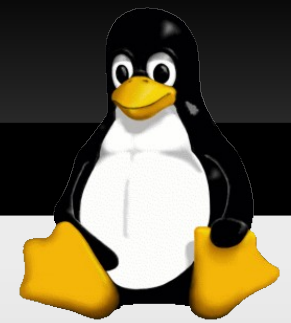
*Thomaz de Oliveira dos Reis - **thommy@globo.com***

2. Cabeçalho

O cabeçalho serve para escolher o interpretador de comandos que será utilizado. Tem que estar na primeira linha e ser iniciado com #!

```
1 #!/bin/sh
2 echo Hello World
```

Shell Script - Básico



Thomaz de Oliveira dos Reis - ***thommy@globo.com***

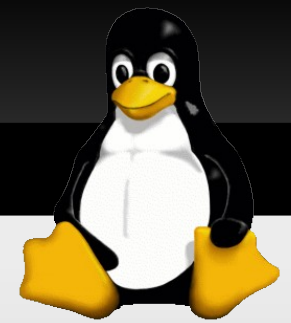
3. Comentários

Comentário é toda texto contido no script (ou código fonte) que não faz parte do código.

Os comentários em shell são iniciados com #

```
1 #!/bin/sh
2
3 #Isto é um comentário :)
4 echo Hello World
```


Shell Script - Básico



Thomaz de Oliveira dos Reis - **thommy@globo.com**

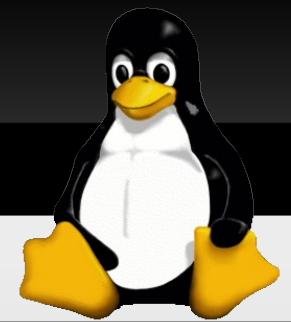
4. Condicional

A condicional serve para ver se uma sentença é verdadeira ou falsa
Em shell, a condicional é o if, e retorna verdadeiro se o comando terminar com sucesso, e falso se der erro no comando

```
1  #!/bin/sh
2
3  if ls
4  then
5      echo o comando deu certo
6  else
7      echo o comando deu errado
8  fi
```

Experimente mudar o “ls” para um comando inexistente ou usar parâmetros errados no ls.

Shell Script - Básico



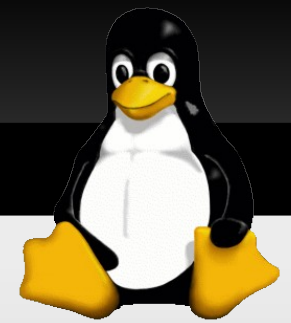
Thomaz de Oliveira dos Reis - **thommy@globo.com**

```
1  #!/bin/sh
2
3  #Comparação de textos
4  if [ isso = aquilo ]
5  then
6      echo são iguais
7  else
8      echo são diferentes
9  fi
10
11 #O arquivo existe?
12 if [ -f arquivo ]
13 then
14     echo arquivo existe
15 else
16     echo arquivo não existe
17 fi
18
19 #o diretório existe?
20 if [ -d diretorio ]
21 then
22     echo diretório existe
23 else
24     echo diretório não existe
25 fi
```

```
26
27 #negando o if
28 if ! [ isso = aquilo ]
29 then
30     echo são diferentes
31 fi
32
33 #Conjunção E
34 if [ isso = isso ] && [ -f arquivo ]
35 then
36     echo são iguais E arquivo existe
37 fi
38
39 #conjunção OU
40 if [ isso = isso ] || [ -f arquivo ]
41 then
42     echo são iguais OU arquivo existe
43 fi
44
45 #comparador matemático
46 if (( 5 < 10 ))
47 then
48     echo 5 é menor que 10
49 fi
```

Shell Script - Básico

Thomaz de Oliveira dos Reis - ***thommy@globo.com***



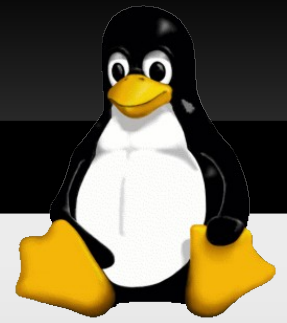
5. Pegando informação do teclado

O comando `read` serve para pegar uma entrada do teclado, e armazena-la na variável informada

```
1  #!/bin/sh
2
3  echo Qual é o seu nome?
4
5  read NOME
6
7  echo Olá $NOME\! Tenha um bom-dia\!
```

Shell Script - Básico

Thomaz de Oliveira dos Reis - ***thommy@globo.com***



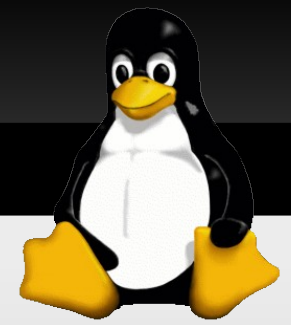
Desafio!

Faça um script, que mostre o conteúdo do arquivo informado pelo usuário!

Atenção: se o arquivo não existir mostre a seguinte mensagem:

Que pena, o arquivo (nome do arquivo) infelizmente não foi encontrado...

Shell Script - Básico



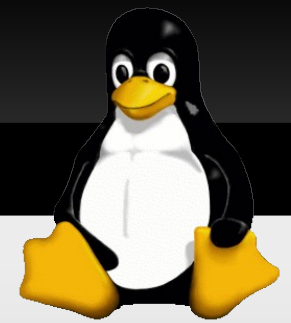
Thomaz de Oliveira dos Reis - **thommy@globo.com**

Solução simples

```
1  #!/bin/sh
2
3  echo Informe o nome do arquivo:
4  read ARQUIVO
5
6  if [ -f "$ARQUIVO" ]
7  then
8      cat "$ARQUIVO"
9  else
10     echo Que pena, o arquivo $ARQUIVO infelizmente não foi encontrado...
11 fi
```

Shell Script - Básico

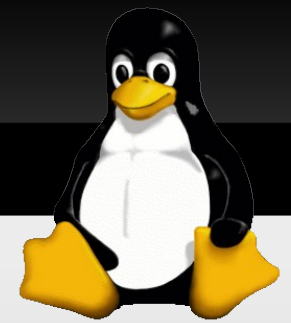
Thomaz de Oliveira dos Reis - ***thommy@globo.com***



Solução elegante

```
1  #!/bin/sh
2
3  echo Informe o nome do arquivo:
4  read $ARQUIVO
5  if ! cat "$ARQUIVO" 2> /dev/null
6  then
7      echo Que pena, o arquivo $ARQUIVO infelizmente não foi encontrado...
8  fi
```

Shell Script - Básico

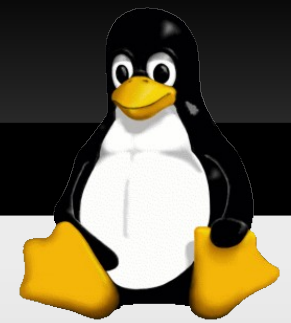


Thomaz de Oliveira dos Reis - **thommy@globo.com**

Solução gráfica

```
1  #!/bin/sh
2  DIALOG=dialog #você pode escolher o Xdialog ou o dialog
3  ARQTEMP=/tmp/script.$$ # $$ é trocado pelo número do processo do script
4
5  #pega o nome do arquivo e guarda em arqtemp
6  $DIALOG --fselect "$HOME/" 10 60 2>$ARQTEMP
7
8  #pega o conteúdo do arquivo ARQTEMP e bota em ARQUIVO
9  ARQUIVO=`cat $ARQTEMP`
10
11 rm -fr $ARQTEMP #remove o ARQTEMP
12
13 if [ -f $ARQUIVO ]
14 then
15     $DIALOG --textbox "$ARQUIVO" 0 0
16 else
17     $DIALOG --msgbox "Que pena, o arquivo $ARQUIVO infelizmente não foi
18 encontrado..." 0 0
19 fi
```

Shell Script - Básico



Thomaz de Oliveira dos Reis - **thommy@globo.com**

6. Estruturas de repetição

As estruturas de repetição servem para repetir uma série de comandos até que uma condição especial seja alcançada.

WHILE

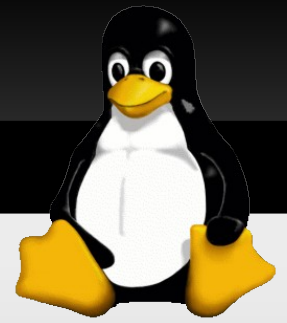
Enquanto a condição do while for verdadeira, as instruções entre do e done é executada.

O `$(())`, serve para executar operações matemáticas

```
1  #!/bin/sh
2
3  X=0
4  while (( X < 10 ))
5  do
6      echo $X
7      X=$((X+1))
8  done
```


Shell Script - Básico

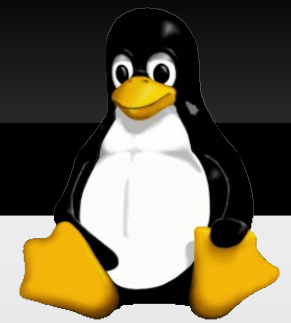
*Thomaz de Oliveira dos Reis - **thommy@globo.com***



Desafio!

Reescreva o programa anterior, para que toda vez que o usuário informar um nome de arquivo inválido, ele tenha que digitar o nome do arquivo novamente.

Shell Script - Básico



Thomaz de Oliveira dos Reis - **thommy@globo.com**

6. Estruturas de repetição

FOR

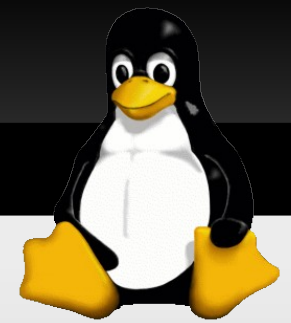
O for no shell se comporta diferente do que o for nas linguagens tradicionais. Em shell, o for percorre uma lista qualquer, e atribui a uma variável, cada valor dessa lista.

O comando `seq X Y Z`,
conta de X
até Z indo de Y em Y

```
1 #!/bin/sh
2
3 for X in `seq 1 1 10`
4 do
5     echo vamos contar: $X
6 done
7
```

Shell Script - Básico

Thomaz de Oliveira dos Reis - ***thommy@globo.com***



Desafio Final!

Faça um programa que redimensione todos as imagens png da pasta informada pela linha de comando para 50% do seu tamanho, e salve no formato jpg

Para modificar imagens pela linha de comando, use o comando convert, do image magick

para pegar os argumentos da linha de comando, use:

\$1 para o primeiro arguento, **\$2** para o segundo e assim por diante

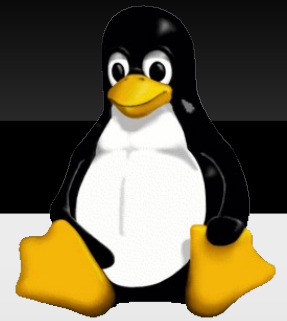
\$* retorna todos os argumentos da linha de comando

\$# retorna a quantidade de argumentos

shift 1 remove o primeiro parametro e o segundo passa a ser o primeiro, o terceiro o segundo e assim por diante

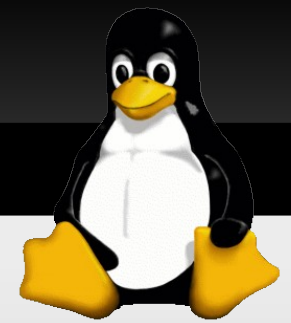
Shell Script - Básico

*Thomaz de Oliveira dos Reis - **thommy@globo.com***



- ★ Como o Shell funciona
- ★ Comandos interessantes
- ➔ Programando em shell

Shell Script - Básico



*Thomaz de Oliveira dos Reis - **thommy@globo.com***

- ★ Como o Shell funciona
- ★ Comandos interessantes
- ★ Programando em shell

Finalmente Chegamos ao Final!

Lista de Discussão: <http://br.groups.yahoo.com/group/shell-script/>
Links: www.thobias.org <http://aurelio.net/> www.julioneves.com
Livro: Programação Shell Linux – Julio Cezar Neves (ed. Brasport)