Introdução ao

Script

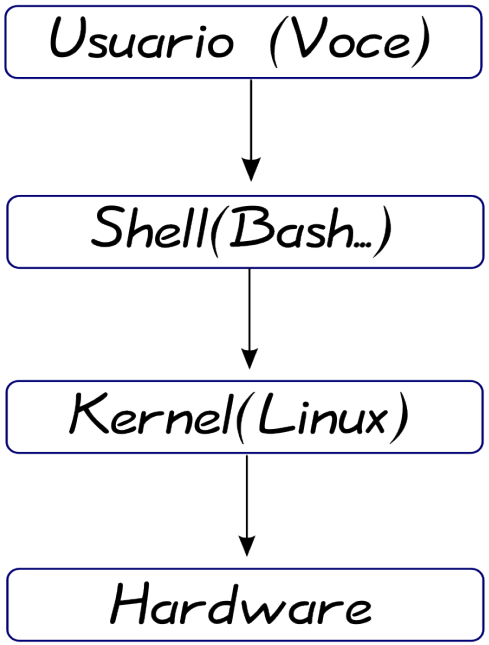
O que é “Bash Script”?

O que é “Bash” + “Script”?

O que é “Bash”?

O que é “Bash”?

- Ele é um interpretador de comandos.
- E um shell.
  - Shell é a interface com o sistema operacional.



Os primeiros Passo

- Qual a primeira linha em um bash script?
  - **#!/bin/bash**
- Se você quer que ele execute...
  - chmod u+x programa.sh

Agora é só escrever comandos!

A primeira vez.

- Abra um terminal.
- Escreva um programa que mostre informações do sistema.
  - Use o “ free -m ”
  - Use “ df -h / ”
  - Use o “ uname -s ”

On The Road!!

BASH

Bizu:

Terminal | Terminal | Terminal | Terminal

```
11:04 - psycho : [ debianFestival ] $cat primeiraVez.sh
#!/bin/bash
free -m
df -h /
uname -s
11:04 - psycho : [ debianFestival ] $./primeiravez.sh
bash: ./primeiravez.sh: Arquivo ou diretório não encontrado
11:04 - psycho : [ debianFestival ] $./primeiraVez.sh
bash: ./primeiraVez.sh: Permissão negada
11:05 - psycho : [ debianFestival ] $chmod u+x primeiraVez.sh
11:05 - psycho : [ debianFestival ] $./primeiraVez.sh
             total       used       free     shared    buffers     cached
Mem:           973        960         12          0         84        473
-/+ buffers/cache:        402        570
Swap:          956          6        950
Sist. Arq.             Tam   Usad Disp  Uso% Montado em
/dev/sda2              5,6G  4,9G  745M  87% /
Linux
11:05 - psycho : [ debianFestival ] $
```

Analise do primeiraVez.sh

- Não foi muito dinâmico.
  - Seu programa não tomou nenhuma decisão!
- Código confuso!!
- Saída muito Feia!!

Como melhorar a saída?

- A saída esta muito crua.
- Pouco informativa.
  - Só mostra um monte de coisas na tela.
  - Se você não tivesse escrito, não entenderia.
- **Temos que informar o que vamos fazer!**
  - Quase sempre....

Como melhorar a saída?

- Para isto existe o comando “**echo**”
- Imprime algo na saída padrão.
- Como funciona:

```
echo ola mundo
echo Semana Da Computação
```

A segunda vez.

- Abra um terminal.
- Escreva um programa que mostre informações do sistema.
  - Use o “ free -m ”
  - Use “ df -h / ”
  - Use o “ uname -s ”
- Agora, use o echo para melhorar o programa.
  - Tente, invente! Deixe mais amigável!!

On The Road 2 – A volta!!

Terminal | Terminal | Terminal

```
14:00 - psycho : [ debianFestival ] $cat ./segundaVez.sh
#!/bin/bash
#Lalalaalla, não serei executado, sou um comentário
#Fale o que você vai mostrar agora:
echo Informações sobre a memoria usada:
free -m
#vamos mostrar as infos do diretório raiz
echo
echo Infomações sobre o tamanho da partição principal:
echo
df -h /
echo
echo O meu sistema operacional é:
uname -s
14:00 - psycho : [ debianFestival ] $./segundaVez.sh
Informações sobre a memoria usada:
             total       used       free     shared    buffers     cached
Mem:           973        952         21          0         54        544
-/+ buffers/cache:        352        620
Swap:          956         38        918

Infomações sobre o tamanho da partição principal:

Sist. Arq.            Tam   Usad Disp  Uso% Montado em
/dev/sda2             5,6G  4,9G  744M  87% /

O meu sistema operacional é:
Linux
14:00 - psycho : [ debianFestival ] $
```

Variáveis em Bash Script

Variáveis em Bash Script

- O que são variáveis?
  - É uma posição de memoria que armazena uma informação que pode ser alterada ou consultada pelo programa.
- O que são variáveis em Bash script?
  - É um lugar da memoria que grava uma string de dados que você pode acessar para alterar ou consultar.
- A principal diferença:
  - Em bash script, todas as variáveis não tem tipos.
  - Elas são strings, para todos os efeitos.

- Como criar uma variável?
  - Basta atribuir uma string a ela.

- Então, como atribuir um valor a uma variável?
  - Variável=“Valor”

    ou
  - Variável=

- **E agora, como eu acesso o conteúdo?**
  - ${Variável}

- Exemplo:

```
X=“ola mundo”
echo ${X}
```

Expansão de Variáveis

- E se eu quiser imprimir na tela:
  - “\O/” “Hello” “world” “\O/”
- Primeira tentativa:
  - echo "\O/" "Hello" "world" "\O/"
  - Deu certo? ? ?

Expansão de Variáveis

- O bash interpreta como uma única string caracteres agrupados com aspas duplas.
  - “Semana da Computação”
    - String única.
  - Semana da Computação
    - Três strings
  - Semana da Computação
    - Três strings.
  - Para as aspas duplas serem consideradas aspas duplas, devemos “escapa-las”.
    - \"

Expansão de Variáveis

- O bash interpreta como uma unica string caracteres agrupados com aspas duplas.
  - Para as aspas duplas serem consideradas aspas duplas, devemos “escapa-las”.
- Segunda tentativa:
  - echo "\\O/\" \"Hello\" \"world\" \"\\O/\"
  - Deu certo? ? ?
    - Não ...
  - Ficou simples?
    - Não, muitos “escapes” ... Isso não deveria ser simples?

Expansão de Variáveis

- Para simplificar a vida, existem a aspa simples
- O que elas fazem?
  - Tudo dentro das aspas simples não são especiais.
  - Não e necessário “escapar” nada!!
- Terceira tentativa:
  - echo '"\O/" "Hello" "world" "\O/"'

On The Road III – A expansão

- Abra um terminal.
- Escreva um programa que:
  - Atribua a uma variável a string:
    - 'Ola Mundo \O/'
  - Mostre na tela o resultado sem aspas
    - ${variavel}
  - Mostre na tela o resultado com aspas simples.
    - ' ${variavel} '
  - Mostre na tela o resultado com aspas duplas.
    - “${variavel}”

Resultados:

Terminal | Terminal | Terminal

```
02:10 - psycho : [ debianFestival ] $cat expansao.sh
#!/bin/bash

VAR='Ola       Mundo         \O/'

echo ${VAR}
echo '${VAR}'
echo "${VAR}"
02:11 - psycho : [ debianFestival ] $./expansao.sh
Ola Mundo \O/
${VAR}
Ola       Mundo         \O/
02:11 - psycho : [ debianFestival ] $
```

Como podemos interagir com o usuário?

- Como podemos “pegar” uma informação?
- Para isto existe o comando “read”
  - Armazena numa variável o que o usuário digitou.
- Como funciona:

```
read entrada
echo “${entrada}”
```

On The Read 4

- Abra um terminal.
- Escreva um programa que:
  - Leia o nome do usuário e coloque na variável nome
  - Mostre na tela o nome “O nome do usuário é ” seguido do nome do usuário

Resultados:

Terminal | Terminal | Terminal

```
02:30 - psycho : [ debianFestival ] $cat read.sh
#!/bin/bash

#esse echo é apenas informativo.
echo 'Digite o nome do usuario:'
read nome
echo 'O nome do usuario é '"${nome}"
02:30 - psycho : [ debianFestival ] $./read.sh
Digite o nome do usuario:
Psycho Mantys  Da Silva
O nome do usuario é Psycho Mantys  Da Silva
02:31 - psycho : [ debianFestival ] $
02:31 - psycho : [ debianFestival ] $
```

Execução Condicional

O comando “test”

BASH

- Testa uma condição, retornando erro se falso.
  - **test CONDIÇÃO**
- **Exemplos:**
  - **test “” #retorna erro, falso. string vazia é falso.**
  - **test “qualque coisa” #verdade. Não esta vazio.**
  - test “${x}” = “${x}” #verdade.

Execução Condicional - if

```
if COMANDO
then
    comando1
    comando2
else
    comando3
    comando4
fi
```

test com if

```
if test “${x}” = “${x}”
then
    echo igual
else
    echo diferente
fi
```

On The Road 5 – A comparação

- Abra um terminal.
- Escreva um programa que:
  - Leia o nome de duas pessoas.
  - Compare os nome e diga se são iguais ou diferentes

Resultados:

Terminal | Terminal | Terminal

```
04:36 - psycho : [ debianFestival ] $cat if.sh
#!/bin/bash

echo "Digite o primeiro nome:"
read nome1

echo "Digite o segundo nome:"
read nome2

if test "${nome1}" = "${nome2}" ; then
        echo "iguais"
else
        echo "diferentes"
fi
04:36 - psycho : [ debianFestival ] $./if.sh
Digite o primeiro nome:
Jose
Digite o segundo nome:
jose
diferentes
04:36 - psycho : [ debianFestival ] $./if.sh
Digite o primeiro nome:
Jose
Digite o segundo nome:
Jose
iguais
04:36 - psycho : [ debianFestival ] $
04:36 - psycho : [ debianFestival ] $
```

Execução Condicional - for

```
for VAR in LISTA
do
    comando1
    echo “${VAR}”
    comando2
done
```

Exemplo for

```
for var in 1 2 3 4 5
do
    echo “${var}”
done
```

Execução Condicional - while

```
while COMANDO
do
    comandos
done
```

- COMANDO as mesmas regras do if.
  - test CONDIÇÃO

On The Road 6 – Enquanto...

- Abra um terminal.
- Escreva um programa que:
  - Leia um nome.
  - E só pare quando for o seu nome digitado.

Resultados:

Terminal | Terminal | Terminal

```
05:07 - psycho : [ debianFestival ] $cat while.sh
#!/bin/bash

nome=

while test "${nome}" != "Psycho Mantys" ; do
        echo digite seu nome:
        read nome
done
05:07 - psycho : [ debianFestival ] $while.sh
digite seu nome:
Psycho
digite seu nome:
Mantys
digite seu nome:
Psycho mantys
digite seu nome:
Psycho    Mantys
digite seu nome:
Psycho Mantys
05:08 - psycho : [ debianFestival ] $
```

Aprendendo a Contar

Expressões Aritméticas

- Como fazer uma conta em bash script?
  - **$((expressão))**
    - **echo $((1 + 1))**
    - **echo $((x + x)) # x é uma variável**
  - expr “expressão”
    - expr 1 + 1
    - expr “${x}” + “${x}”

Fatorial

- Abra um terminal.
- Escreva um programa que:
  - Leia o um numero.
  - Imprima o fatorial daquele numero.

Resultados:

Sem título | Sem título

```
21:47 - psycho : [ samples ] $cat fatorial.sh
#!/bin/bash
x=
resultado=1

read x

while [ ${x} != 0 ] ; do
        resultado=$(( resultado*x ))
        x=$((x-1))
done

echo ${resultado}
21:48 - psycho : [ samples ] $./fatorial.sh
3
6
21:48 - psycho : [ samples ] $./fatorial.sh
5
120
21:48 - psycho : [ samples ] $
```

Redirecionamento e substituição

Redirecionamento e substituição

- Redirecionamentos desviam um fluxo de dados do programa para um outra caminho.
  - O exemplo mais simples é a saida para a tela.
- No Bash, você pode redirecionar:
  - Processo-processo (via “pipe”).
  - Arquivo para processo.
  - Processo para um arquivo.
  - E algumas outras formas.

Processo para um arquivo

BASH

- Como fazer isso?
  - Comando > arquivo
    - Apaga “arquivo” é coloca saída de “Comando” em “arquivo”
  - Comando >> arquivo
    - Coloca a saida de “Comando” no final de “arquivo”
- O que aconteceu?
  - A saída normal do comando sera colocada em “arquivo”.
  - Por exemplo, se Comando é “echo teste”, em “arquivo” estaria escrito “teste”.

On The Road 7 – Dando novas direções

- Abra um terminal.
- Modifique o segundo programa, criando outro:
  - Este outro programa deve salvar toda a saída em um arquivo chamado log.txt para consulta posterior.
  - Depois grave toda a saída não apagando o conteúdo anterior do arquivo.

Resultados:


Terminal
Terminal
Terminal
06:01 - psycho : [debianFestival] $cp segundaVez.sh redirecionar.sh
06:01 - psycho : [debianFestival] $vim redirecionar.sh
06:02 - psycho : [debianFestival] $cat redirecionar.sh
#!/bin/bash
#Lalalaalla, não serei executado, sou um comentário
#Fale o que você vai mostrar agora:
echo Informações sobre a memoria usada: > log.txt
free -m >> log.txt
#vamos mostrar as infos do diretório raiz
echo >> log.txt
echo Infomações sobre o tamanho da partição principal: >> log.txt
echo >> log.txt
df -h / >> log.txt
echo >> log.txt
echo O meu sistema operacional é: >> log.txt
uname -s >> log.txt
06:02 - psycho : [debianFestival] $./redirecionar.sh
06:02 - psycho : [debianFestival] $cat log.txt
Informações sobre a memoria usada:
total used free shared buffers cached
Mem: 973 957 16 0 137 409
-/+ buffers/cache: 409 563
Swap: 956 61 895
Infomações sobre o tamanho da partição principal:
Sist. Arq. Tam Usad Disp Uso% Montado em
/dev/sda2 5,6G 4,9G 745M 87% /
O meu sistema operacional é:
Linux
06:02 - psycho : [debianFestival] $

Processo para um arquivo

- Como fazer isso?
  - Comando | Comando2
    - Toda a saída de Comando alimenta a entrada de Comando2.
- O que aconteceu?
  - E como se você digitasse a saída de Comando em Comando2!!

Comando para um Comando

- Exemplo:
  - Executamos primeiraVez.sh.
  - Através do pipe, comando less pega o que vem pela entrada padrão e mostra.

O Comando:
./primeiraVez.sh | less

Links

- www.shellscript.com.br
  - Site do livro muito bom sobre shell.
- http://aurelio.net/shell/
  - Site obrigatorio de um dos maiores programadores de shell do brasil.
- http://twiki.softwarelivre.org/TWikiBar/WebHome
  - Site do “Pai do Shell” Brasileiro, Júlio Neves.
- http://thobias.org/
  - Site com muitas dicas uteis, artigos e programas.

Fim!

[psycho@localhost ~]# logout

psycho.mantys@gmail.com