Data Science & Big Data

Infraestrutura Computacional
Parte 1: Linux e Shell

DSBD

Shell script

Bash Script

- Um script bash é um “texto” contendo uma sequência de comandos a serem executados por um terminal Bash
- Qualquer comando que pode ser executado num terminal, pode ser invocado por um script
- Para ser executado, o script precisa ter permissões de execução e ser invocado
- Para execução em modo DEBUG:
  - `set -x`

Exemplo de um script Bash

user@host: cat meuscript.sh #!/bin/bash # # Exemplo de um script bash echo "Arquivos no diretorio HOME" ls $HOME	Uso opcional da extensão .sh Identifica o interpretador Comentários: # Comando echo Comando ls
user@host: ls -l meuscript.sh **-rwx------ ... meuscript.sh** **user@host: ./meuscript.sh** Arquivos no diretorio HOME aula01.odp aula02.odp meuscript.sh	Permissão de execução Invocação. Por que **./**?

Variáveis

Atribuição

- **`nome=valor`**, sem espaços. Uso de aspas duplas no valor para incluir espaços
- Acesso à variável deve ser precedido do sinal **`$`**

```
var="Hello World"
echo $var
```

Variáveis especiais

- **`$0`**: nome do script
- **`$1-$9`**: parâmetros
- **`$#`**: número de parâmetros
- **`$*`**: todos os parâmetros

Variáveis e comandos

Resultado de comandos podem ser atribuidos a variáveis:

- var=$(comando)

```
user@host: cat variaveis.sh
#!/bin/bash

data=$(date)

echo "Hoje e dia $date"
echo "Meu nome e $0 e recebi $# argumentos"
echo "Os argumentos são $*"
```

Variáveis

Variáveis podem ter seu valor alterado por substituição de padrões

- **man bash** (/Parameter expansion)
- **${var/padrao/string}**: substitui padrão pela string
- **${var#padrao}**: remove padrao do inicio da variável
- **${var%padrao}**: remove padrao do final da variável

Variáveis

```
user@host: var=foto.JPG
user@host: echo ${var/JPG/jpg}
foto.jpg
user@host: echo ${var%JPG}
foto.
user@host: echo ${var#foto}
.JPG

user@host: fotos=$(ls *.JPG)
user@host: for f in $fotos; do
> $f ${f/JPG/jpg}
> done
user@host:
```

Substituir JPG → jpg

Exemplo 1

```
user@host: cat backup.sh
#!/bin/bash
# Salva o conteudo de um diretorio

data=$(date +%F)
dir=$1
destino=$HOME/backups/

mkdir -p $destino
cp -R $dir $destino/$dir-$data
echo "Backup do diretorio $dir completo"
```

Exemplo 2

```
user@host: cat backup2.sh
#!/bin/bash
# Salva o conteudo de um diretorio
data=$(date +%F)
dir=$1
destino=$HOME/backups/
if [ $# != 1 ]; then
  echo "Uso: um argumento que e o diretorio a ser salvo"
  exit
fi

if [ ! -d $dir ]; then
  echo "O diretorio $dir nao existe"
  exit
fi
```

Exemplo 2

```
…
# ja temos um diretorio de backup?
if [ -d $destino/$dir-$data ]; then
  echo "Este diretorio ja foi salvo hoje. Sobrescrever?"
  read resposta
  if [ $resposta != 's' ]; then
    exit
  fi
else
  mkdir -p $destino/$dir-$data
fi

cp -R $dir $destino/$dir-$data
echo "Backup do diretorio $dir completo"
```

Perguntas?

Editor de Textos vi

Editor vi

É um editor de linha de comando

- TUDO é feito pelo teclado
- vi x vim (*vi iMproved*)

Há dois modos no vi:

- **Inserção:** permite modificar o texto do arquivo
- **Edição:** permite movimentar no arquivo, efetuar comandos, etc.

Há dois tipos de pessoas no mundo:

- as que usam vi
- as que nunca serão!

Comandos básicos do vi

Comando	Significado
`i`	entra em modo de inserção
`ESC`	sai do modo de inserção / entra em edição
`:w`	salva o arquivo
`:q`	encerra o vi
`:NUM`	vai para a linha NUM
`r`	substitui um caractere
`R`	entra em modo de edição de sobrescrita
`J`	concatena a linha de baixo à atual
`dd`	apaga a linha atual
`$`	move o cursor para o final da linha
`u`	desfaz a última ação

Perguntas?

Referências

- Anatomy of the Linux kernel
- Linux OS Tutorial
- Introduction to UNIX
- Introduction to Linux
- Ryans Linux Tutorial
- Aurelio.net Shell
- Bash Guide for Beginners