

Aula 04

- Programação bash

Linha de comando e script

- O bash interpreta comandos em linha de comando e salvos em arquivos similarmente. Podemos colocar em um script bash os comandos na sequencia que digitamos.

```
> echo Hello
```

- No arquivo hello.sh

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

```
echo Hello
```

- (comando1 ; comando2) isola os comandos
- No caso do arquivo, verificar permissão de execução e PATH

Variáveis

- Para utilizar variáveis no script, apenas atribua um valor a ela
#!/bin/bash
VAR=valor
echo VAR = \$VAR
- Variáveis do bash
 - Parâmetros: \$0, \$1, \$2
 - Todos os parâmetros: @\$
 - Quantidade de parâmetros: \$#

Variáveis

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

```
echo Quantidade de parametros $#
```

```
echo Todos os parametros "$@"
```

```
echo Parametro0 $0
```

```
echo Parametro1 $1
```

```
echo Parametro2 $2
```

```
echo Parametro3 $3
```

```
echo Parametro4 $4
```

Variáveis

- \$? saída do ultimo comando
 - > ls
 - > echo \$?
 - > ls zzzz
 - > echo \$?
- \$! ultimo processo em background
- \$IFS separador de linhas

Variáveis

- Variáveis
 - echo \$var
 - echo \${var}
- Operações matemáticas
 - let x=10
 - let y=\$x*2+1
 - echo \$x e \$y
 - c=\$((\$x*\$y))

Loops

- Loop for executa comandos percorrendo lista de itens separado por espaço (padrão \$IFS, mas pode alterar)

```
> for arg in [list]
```

```
do
```

```
...
```

```
done
```

loops

- Separador de variáveis é o IFS
- Arquivos
 - for var in “*”
- Parâmetros
 - for var in “\$@”
- Variáveis
 - for var in \$PATH (Tem que alterar \$IFS)
- Sub-shell
 - for var in \$(comando)

Condições de testes

- Condições de testes estão documentadas no
 - `man test`
 - `info test`
 - `help [`
- `[-z "$VAR"]`
- `["$VAR" = "texto"]`
- `["$VAR" -eq 10]`
- `[-f "$arquivo"]`
- `(($VAR == 20))`
- `(($# > 3))`

Comando if

- help if
- if: if COMMANDS; then COMMANDS; [elif COMMANDS; then COMMANDS;]... [else COMMANDS;] fi
 > if [-f arquivo]
 then ...
 elif [-d arquivo]
 then ...
 else
 fi

Comando while

- help while
- while: while COMMANDS; do COMMANDS; done

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

```
count=1
```

```
while [ "$count" -le "10" ]
```

```
do
```

```
    echo $count
```

```
    let count++
```

```
done
```

Comando while

```
#!/bin/bash
cat file | while read line
do
    echo $line
done

#!/bin/bash
while read line
do
    echo $line
done
```

Dicas de Bash

- Ctrl+X Ctrl+E Editor de linha de comando. Configurar \$EDITOR
- `var=$(comando)`
- `comando $(comando)`
- `VAR=valor comando`
- `VAR=valor ; comando`
- `comando_para_shell ; echo !! >> script.sh (!n)`
- `comando arg ; comando $_ (ou primeiro !^, ou !comando:n)`

Exercícios

1) Escreva um programa tarefa.sh na pasta <nome>/aula04/exercicios

Nesse programa, faça um teste usando o if teste de quantos parâmetros são passados.

Se nenhum é passado exiba a ajuda do programa:

tarefa.sh <parametro>

Se uma opção é passada imprima

Execução \$0 \$1

Exercícios

2) Acrescente no final do arquivo tarefa.sh :

Uma rotina de teste se a variável passada como parâmetro \$1 é um arquivo ou diretório

Imprima

Arquivo <nome>

Diretorio <nome>

Exercícios

3) Modifique o tarefa.sh para:

parâmetro é um arquivo

- execute o programa: file \$1

parâmetro é um diretório

- faça um loop for linha in \$1/* do....
- echo \$linha

4) Modifique tarefa.sh para contar quantidade de arquivos e diretórios listados.