

## Laboratório 6: Shell Script - Parte 1

Professor: Diego da Silva de Medeiros  
Baseado no material do Prof. Eraldo.

diegomedeiros@ifsc.edu.br

## 1 Objetivos

1. comando for;
2. comando while para ler linhas de arquivos

## 2 Comando de repetição for

**Nota:** Os comandos de repetição do bash permitem a repetição controlada de uma sequência de instruções, permitindo a execução automática de tarefas complexas. Para detalhes ver <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO-7.html>

No comando for abaixo, o código dentro do *loop* (entre *do* - *done*) é executado para cada valor assumido pela variável *i*.

1. Chame o um editor de texto qualquer e crie um arquivo para ser o seu primeiro script:

```
gedit meu_script.sh
```

2. Entre com o seguinte texto, salve e saia do editor:

```
#!/bin/bash

echo
echo Primeiro FOR

for i in alfa beta gama teste
do
    echo item $i
done

echo
echo Segundo FOR

for i in "alfa beta" "gama teste"
do
    echo item $i
done
```

3. Atribua direito de execução:

```
chmod u+x ./meu_script.sh
```

4. Teste o script e analise-o com o professor:

```
./meu_script.sh
```

### 3 Substituindo a lista de strings pela saída de um comando

1. Chame o um editor de texto qualquer e crie um arquivo para ser o seu primeiro script:

```
gedit meu_script.sh
```

2. Entre com o seguinte texto, salve e saia do editor:

```
#!/bin/bash

for i in `ls`; do
    echo ARQUIVO: $i
done
```

3. Atribua direito de execução:

```
chmod u+x ./meu_script.sh
```

4. Teste o script:

```
./meu_script.sh
```

**Nota:** Observe que neste exemplo, inicialmente o comando `ls` é executado. Na sequência a variável `i` assume o valor da saída do comando `ls`. Para cada cadeia de caracteres (separadas por espaço em branco) os comandos internos ao `for` são executados.

5. Analise e teste o seguinte script (execute com `sudo`):

```
#!/bin/bash

DIR_ALVO=$HOME/confs

mkdir $DIR_ALVO
for i in `ls /etc/*.conf`; do
    cp $i $DIR_ALVO
done
```

6. DESAFIO: Criar um script para colocar um arquivo padrão a ser passado como parâmetro em todos os diretórios de usuários que estão no */home*

### 4 Acessando dados dentro de um arquivo

1. Use um editor de textos para colocar dados dentro de um arquivo `alfa.txt`

```
delta
epson
beta
gama
```

2. Execute o seguinte script:

```
#!/bin/bash

for i in `cat alfa.txt`; do
    echo Palavra $i
done
```

3. DESAFIO: Baseando-se no exemplo anterior, criar um script para criar um arquivo beta.txt que substitua palavras do alfa.txt que coincidam com uma palavra passada como parâmetro por uma segunda palavra passada como parâmetro.

DICA 1: use um comando if [ “S1” == “S2” ] ... dentro do comando for.

DICA 2: gere o outro arquivo simplesmente fazendo echo e redirecionando a saída padrão. O arquivo deve ser criado previamente.

## 5 Lendo campos em linhas com o comando while

1. Edite e salve um arquivo *users.txt* com os seguintes dados:

```
eraldo projX
maria projY
jose projY
lara projX
joaquim projY
```

2. Edite e execute o seguinte shell script:

```
#!/bin/bash

CONT=1
while read usuario projeto;do
    echo $CONT $usuario $projeto
    let CONT=CONT+1
done < users.txt
```

**Nota:** Observe que o comando while também implementa um loop e com o apoio do comando read permite ler linhas de um arquivo. Neste exemplo também mostramos o uso de uma variável aritmética (CONT).

3. DESAFIO: criar um script para contar quantos usuários do projeto projX estão presentes no arquivo. Use um comando if dentro do loop para testar a igualdade de projeto.