

# Introdução à **bash**

## Programação

José Pedro Oliveira  
(jpo@di.uminho.pt)

Grupo de Sistemas Distribuídos  
Departamento de Informática  
Escola de Engenharia  
Universidade do Minho

Sistemas Operativos 2005-2006

# Conteúdo

- 1 Programação
  - Script shell
  - Variáveis
  - Parâmetros
- 2 Estruturas de controlo
  - if
  - case
- 3 Ciclos
  - for
  - while
  - until

# Script shell

## Descrição

- A primeira linha deve conter **#!** seguido do nome de um interpretador (por exemplo: **#!/bin/sh**)
- As restantes linhas devem conter instruções válidas para esse interpretador
- O ficheiro deve ter o atributo **x**. Este atributo pode ser activado através do comando **chmod** (por exemplo: **chmod a+x nome\_script**)

## Exemplo de um script shell

### Exemplo de um script shell

```
1  #!/bin/sh
2
3  # Comentário
4
5  date -l
6  uname -s -r
```

# Exemplo de um script perl

## Exemplo de um script perl

```
1 #!/usr/bin/perl -w
2 use strict;
3
4 my $cadeira = 'CSI';
5
6 print " Cadeira :_ $cadeira\n";
```

## Permissão de execução

Alteração de permissões

Alternativa 1: modo simbólico

```
$ chmod a+x nome_ficheiro
```

Alternativa 2: modo numérico (octal)

```
$ chmod 755 nome_ficheiro
```

## Permissão de execução

Alteração de permissões

Alternativa 1: modo simbólico

```
$ chmod a+x nome_ficheiro
```

Alternativa 2: modo numérico (octal)

```
$ chmod 755 nome_ficheiro
```

# Variáveis

## Variáveis

- para guardar um valor numa variável utiliza-se o símbolo = (não utilizar espaços antes e depois do sinal de atribuição)
- para mostrar o valor de uma variável utiliza-se o símbolo \$

## Exemplo

```
1 #!/bin/sh
2
3 ABC=123
4
5 echo Valor da variável ABC: $ABC
```



## Variáveis: exemplo

### Exemplo

```
1 #!/bin/sh
2
3 DATA=`date -I`
4
5 echo Data: $DATA
```

# Parâmetros

## Variáveis

`$#` - número de parâmetros

`$*` - lista de parâmetros

`$1` - parâmetro número 1

`$2` - parâmetro número 2

...

`$n` - parâmetro número  $n$

## Parâmetros: exemplo

### Script exemplo03.sh

```

1  #!/bin/sh
2
3  echo "Número de parâmetros: $#"
```

4	echo "Lista de parâmetros : \$*"
5	echo "          Primeiro : \$1"
6	echo "          Segundo : \$2"

```
$ ./exemplo03.sh abc 123
```

```

Número de parâmetros: 2
Lista de parâmetros : abc 123
          Primeiro : abc
          Segundo : 123
```

# Conteúdo

- 1 Programação
  - Script shell
  - Variáveis
  - Parâmetros
- 2 Estruturas de controlo
  - if
  - case
- 3 Ciclos
  - for
  - while
  - until

# Instrução if

## Sintaxe 1

```
if [condição]  
then  
    comando...  
fi
```

## Sintaxe 2

```
if [condição]; then  
    comando...  
fi
```

# Instrução if

## Sintaxe 3

```
if [condição1]  
then  
    comando...  
elif [condição2]  
then  
    comando...  
else  
    comando...  
fi
```

## Exemplo

## Script shell

```
#!/bin/sh
```

```
if [ $# != 3 ]; then
```

```
    echo "Erro: este script precisa de 3 parâmetros"
```

```
    exit 1
```

```
fi
```

```
cat $1 $2 | sort | nl -b a > $3
```

# Instrução case

## Sintaxe

```
case "$variável" in
    condição1)
        comando...
        ;;
    condição2)
        comando...
        ;;
esac
```



# Exemplo

## Extracto de um script shell

```
...
case "$1" in
  start)
    echo "start"
    ;;
  stop)
    echo "stop"
    ;;
  *)
    echo "default"
    ;;
esac
...
```

# Conteúdo

- 1 Programação
  - Script shell
  - Variáveis
  - Parâmetros
- 2 Estruturas de controlo
  - if
  - case
- 3 Ciclos
  - for
  - while
  - until

# Ciclo for

## Sintaxe 1

```
for arg in [lista]  
do  
    comando...  
done
```

## Sintaxe 2

```
for arg in [lista]; do  
    comando...  
done
```

# Exemplos

## Listar argumentos (linha de comando)

```
#!/bin/sh

for arg
do
    echo $arg
done
```

## Operação sobre uma lista de ficheiros

```
#!/bin/sh

for fic in *.sh; do
    echo $fic
done
```

# Exemplo

## Criar backup de um conjunto de ficheiros

```
#!/bin/sh

for f in *.c; do
    cp -p $f $f.bak
done
```

## Recorrendo apenas à linha de comando

```
$ for f in *.c; do cp -p $f $f.bak; done
```

# Ciclo **while**

## Sintaxe 1

```
while [condição]  
do  
    comando...  
done
```

## Sintaxe 2

```
while [condição]; do  
    comando...  
done
```

# Ciclo **until**

## Sintaxe 1

```
until [condição]  
do  
    comando...  
done
```

## Sintaxe 2

```
until [condição]; do  
    comando...  
done
```

# Conteúdo

## 4 Referências



# Referências

## Documentação

- **GNU Bash**

<http://www.gnu.org/software/bash/manual/bash.html>

## Bibliografia

- **Learning the bash Shell (segunda edição)**

<http://www.oreilly.com/catalog/bash2/>

- **Advanced Bash-Scripting Guide**

<http://www.tldp.org/LDP/abs/html/index.html>