



Operating Systems

Sistemas Operativos

Shell

Prof. Amine Berqia

Email :
bamine@ualg.pt

Bash as an example of a shell

A function is a set of instructions, allowing you to perform multiple tasks with different input parameters.

You are not limited in the number of functions. However, they should not have the same name.

You can reuse the global variables initialized in your script inside or outside your function.

You can declare variables local to your functions.

A good function is a function that handles a specific, recurring element in your script.

Bash as an example of a shell

In Bash, there are two ways to declare a function:

```
myFunction ()
```

```
{  
instructions  
}
```

```
myFunction
```

```
function myFunction
```

```
{  
instructions  
}
```

```
myFunction
```

Bash as an example of a shell

Function (sum)

```
#!/bin/bash
# Exemplo de uma função (soma)

ARG=2
WRONG_ARGS=-1
if [ $# -ne $ARG ]; then
echo "`basename $0` numero1 numero2"
return $WRONG_ARGS
fi
```

```
sum ()
{
local number1=$1
local number2=$2
result=`expr $number1 + $number2`
return $result
}

sum $1 $2
echo "$1 + $2 = $?."
return 0
```

Bash as an example of a shell

Arrays

To create one or more empty classical arrays, you usually use the `-a` option of the `declare` command

`declare -a array1`

As in C, the index of a classical array always starts at 0 and its maximum value is that of the largest positive integer that can be represented.

Bash as an example of a shell

To designate an element of a classic array, use the syntax: `array1[index]`

```
$ declare -a tab                                tab=(val0 val1 ...)  
  
$ read tab[1] tab[3]                            tab=( [indice]=val ... )  
ola bonjour  
$ tab[0]=hello  
$
```

-a option of the read or readonly command. Each inserted word becomes an element of the classic array:

```
$ read -a tab  
ola bonjour hello  
$  
  
declare -ra tab?  
Declare -p ?
```

Bash as an example of a shell

We got the value of an element from an array using the syntax : `${tab[indice]}`

```
$ echo ${tab[1]}
```

```
ola
```

```
$ echo ${tab[1**2+1]}
```

```
hello
```

qualquer expressão aritmética válida para
calcular o índice de um elemento

Bash as an example of a shell

a variable not previously defined as an array can be interpreted as a classical array :

```
$ var=hello
$ echo ${var[0]}
hello
$ var=( hi "${var[0]}" )
$ echo ${var[1]}
????
```

To get the length of an array element `${# tab[indice]}`

```
$ echo ${#var[0]}
```

```
5
```


Bash as an example of a shell

To delete one or more classic arrays: **unset** tab ...

To add one or more elements to a classic array:

```
tab+=( val1 val2 ... )  
tab=( val0 val1 ... "${tab[@]}" )
```

....