

# Estruturas

## Percorrer ficheiros

```
while read LINE
do
...
done < file
```

```
cat file | while read LINE
do
...
done
```

```
#!/bin/bash

while read LINE
do
    USUARIO=$(echo ${LINE} | cut -d ':' -f1)
    USUARIO_ID=$(echo ${LINE} | cut -d ':' -f3)
    GRUPO_ID=$(echo ${LINE} | cut -d ':' -f4)
    DIRECTORIO=$(echo ${LINE} | cut -d ':' -f6)
    CONSOLA=$(echo ${LINE} | cut -d ':' -f7)
    echo -e "USER=${USUARIO}\tUID=${USUARIO_ID}\tGID=${GRUPO_ID}\tHOME=${DIRECTORIO}\tSHELL=${CONSOLA}"
    sleep 1
done < /etc/passwd
```

## Control de fluxo

### if then fi

```
if test CONDICION;then ...;fi
equivale a
if [ CONDICION ];then ...;fi
equivale a
[ CONDICION ] && comando
```

### if then else fi

```
if test CONDICION;then ...;else ...;fi
equivale a
if [ CONDICION ];then ...;else ...;fi
equivale a
[ CONDICION ] && comando1 || comando2
```

```
#!/bin/bash
##-----
if test -f /etc/passwd; then
    echo Existe
fi
##-----
if [ -f /etc/passwd ]; then
    echo Existe
fi
##-----
[ -f /etc/passwd ] && echo Existe
```

```
#!/bin/bash
##-----
if test ! -f /etc/passwd; then
    echo Non existe
else
    echo Existe
fi
##-----
if [ ! -f /etc/passwd ]; then
    echo Non existe
else
    echo Existe
fi
##-----
[ ! -f /etc/passwd ] && echo Non existe || echo Existe
```

```
#!/bin/bash
##-----
CADEA='string'
##-----
if test -n ${CADEA}; then
    echo Lonx. non nula
fi
##-----
if [ -n ${CADEA} ]; then
    echo Lonx. non nula
fi
##-----
[ -n ${CADEA} ] && echo Lonx. non nula
```

```
#!/bin/bash
##-----
CADEA='string'
##-----
if test ! -z ${CADEA}; then
    echo Lonx. non nula
else
    echo Lonx. nula
fi
##-----
if [ ! -z ${CADEA} ]; then
    echo Lonx. non nula
else
    echo Lonx. nula
fi
##-----
[ ! -z ${CADEA} ] && echo Lonx. non nula || echo Lonx. nula
```

Menú

echo read case ... esac

```
echo Opción 1...
echo Opción 2...
echo Opción 3...
read -p 'Elixé opción: 1,2,3? ' opcion
case $opcion in
    1) comandos
    ;;
    2) comandos
    ;;
    3) comandos
    ;;
    *) echo Opción non correcta
    ;;
esac
```

select do case ... esac done

```
PS3='Elixé opción:1,2,3? '
OPCION1='Texto1'
OPCION2='Texto2'
OPCION3='Texto3'
select opcion in "${OPCION1}" "${OPCION2}" "${OPCION3}"
do
    case ${opcion} in
        ${OPCION1}) comandos
        ;;
        ${OPCION2}) comandos
        ;;
        ${OPCION3}) comandos
        ;;
        *) echo Opción non correcta
        ;;
    esac
done
```

```
#!/bin/bash

echo Opcion1. Ver directorio actual
echo Opcion2. Ler /tmp
echo Opcion3. Sair
read -p 'Elixé opción: 1,2,3? ' opcion
case $opcion in
    1) pwd
    ;;
    2) ls
    ;;
    3) exit
    ;;
    *) echo Non elixiches nin 1,2,3
    ;;
esac
```

```
#!/bin/bash

PS3='Elixé opción:1,2,3? '
OPCION1='Ver directorio actual'
OPCION2='Ler /tmp'
OPCION3='Sair'

select opcion in "${OPCION1}" "${OPCION2}" "${OPCION3}"
do
    case ${opcion} in
        ${OPCION1}) pwd
        ;;
        ${OPCION2}) ls
        ;;
        ${OPCION3}) exit
        ;;
        *) echo Non elixiches nin 1,2,3
        ;;
    esac
done
```

array: select do case esac done

```
PS3='Opción? '
options=("Texto1" "Texto2" "Texto3")
select opcion in "${options[@]}"
do
    case $opcion in
        "Texto1") comandos
        ;;
        "Texto2") comandos
        ;;
        "Texto3") break
        ;;
        *) echo "Non elixiches ningunha opción válida"
        ;;
    esac
done
```

```
#!/bin/bash

PS3='Opción? '
options=("Ver directorio actual" "Ler /tmp" "Sair")
select opcion in "${options[@]}"
do
    case $opcion in
        "Ver directorio actual") pwd
        ;;
        "Ler /tmp") ls /tmp
        ;;
        "Sair") break
        ;;
        *) echo "Non elixiches ningunha opción válida";;
    esac
done
```

# Funcións

## Definir e Invocar

```
function f_name() {
  comandos
}

f_name
```

## Menú

```
function f_op1() { comandos;}
function f_op2() { comandos;}
function f_menu(){
  echo Opcion1. Texto1
  echo Opcion2. Texto1
  read -p 'Elixe opcion:1,2? ' opcion
  case $opcion in
    1) comandos;;
    2) comandos;;
    *) echo Opción non correcta && f_menu;;
  ;;
  esac
}
function f_main(){
  f_menu
}
f_main
```

```
#!/bin/bash

function f_suma() {
  read -p 'Introduce número: ' n1
  read -p 'Outro número: ' n2
  echo Suma: $n1 + $n2 = $((n1+n2))
}

f_suma
```

```
#!/bin/bash

function f_op1(){ pwd;}
function f_op2(){ ls /tmp;}
function f_op3(){ exit;}

function f_menu() {
  echo Opcion1. Ver directorio actual
  echo Opcion2. Ler /tmp
  echo Opcion3. Sair
  read -p 'Elixe opcion:1,2,3? ' opcion
  case $opcion in
    1) f_op1;;
    2) f_op2;;
    3) f_op3;;
    *) echo Non elixiches nin 1,2,3 && f_menu;;
  esac
}

function f_main() {
  f_menu
}

f_main
```

## Invocar parámetros

```
function f_help() {
  echo "Exemplo execución: \ $1 \ $2" && exit
}
function f_parametros() {
  [ $# -ne 2 ] && f_help
}
function f_main() {
  f_parametros $*
}
f_main $*
```

```
#!/bin/bash

function f_help() {
  echo "Exemplo execución: 4 17"
  exit
}
function f_parametros() {
  [ $# -ne 2 ] && f_help
}
function f_suma() {
  echo A suma de $1 + $2 é: $((n1+n2))
}
function f_main() {
  f_parametros $*
  f_suma $*
}

f_main $*
```

## Errorlevel: \$?

```
comando
[ $? -eq 0 ] && echo OK || echo KO
```

```
#!/bin/bash

ping -c2 127.0.0.1
[ $? -eq 0 ] && echo OK || echo KO

nc -vz 127.0.0.1 80
[ $? -eq 0 ] && echo OK || echo KO

read -p 'Número: ' n1
[[ $n1 =~ ^-?[0-9]+$ ]] && echo SI || echo NON
```

## Arrays

```
array_name=(1 2 3)
for i in "${array_name[@]}"
do
    echo $i
done
```

```
#!/bin/bash

array_ports_TCP=(21 22 23 80 443 445)
array_IPs=(127.0.0.1 127.127.127.127)

function f_port() {
    for i in "${array_IPs[@]}"
    do
        nc -vz $i "${array_ports_TCP[@]}"
    done
}

f port
```

## Contadores

```
for i in $(seq 1 10)
do
...
done
```

```
for ((i=1;i<=10;i++))
do
...
done
```

```
#!/bin/bash

for i in $(seq 1 10)
do
    echo Valor de i: $i
done
```

```
#!/bin/bash

for ((i=1;i<=10;i++))
do
    echo Valor de i: $i
done
```