Fundamentos de Software - Prácticas
Módulo I. Órdenes UNIX y Shell Bash

Práctica 5: Programación del shell

Departamento de Lenguajes y Sistemas Informáticos – Universidad de Granada

UNIVERSIDAD DE GRANADA

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Conocer el comando `case-esac` como alternativa para construir instrucciones condicionales**

```
case expresion in
   patron1 )
            declaraciones ;;
   patron2 )
            declaraciones ;;
   * )
            declaraciones ;;
esac
```

Declaraciones a ejecutar si la expresión cumple con el patron1

Opcional para indicar cualquier otro patrón distinto a los mencionados previamente

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Conocer el comando `case-esac` como alternativa para construir instrucciones condicionales**

```
read opcion
case $opcion in
      s|S)
         echo "ha indicado la opción SI"
      ;;
      n|N)
         echo "ha indicado la opción NO"
      ;;
      *)
         echo "desconozco esa opción"
      ;;
esac
```

Variables numéricas o alfanuméricas

Posibilidad de indicar varios patrones (expresiones regulares, OR, AND…)

Posibilidad de incluir una o más sentencias

- Las estructura case-esac puede estar contenida dentro de otro, por ejemplo de un if o un while.
- Uso habitual para construir menús

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Lectura del teclado: comando `read`**
    - Detiene la ejecución del guión y espera a que el usuario teclee algo
    - El texto tecleado es asignado a la(s) variable(s) que acompañan a la orden.

```
read [ -ers ] [ -a array ] [ -d delim ] [ -i text ] [ -n nchars] [ -p
prompt] [ -t time out ] [ -ufd ] [ name … ]
```

```
printf "Introduzca el nombre del archivo:"
read ARCHIVO_INPUT
printf "\n El nombre del archivo es: %s" $ARCHIVO_INPUT
```

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Saber construir instrucciones de bucles: comandos `for`, `while` y `until`**
    - El bucle for permite repetir un cierto número de veces **conocido a priori** las declaraciones especificadas. Su sintaxis es:

```
for nombre [in lista]
do
   declaraciones que pueden usar $nombre
done
```

```
for archivo in $(ls)
for NUM in `seq 0 1 4`;
for (( CONTADOR=1; CONTADOR<10; CONTADOR++ )) ;
```

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Saber construir instrucciones de bucles: comandos `for`, `while` y `until`**
    - El bucle `while` permite repetir las declaraciones especificadas mientras se cumpla la condición. El numero de repeticiones no se conoce a priori y puede ser cero.

```
while expresión;
do
   declaraciones …
done
```

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Saber construir instrucciones de bucles: comandos `for`, `while` y `until`**
    - El bucle `while` permite repetir las declaraciones especificadas mientras se cumpla la condición. El numero de repeticiones no se conoce a priori y puede ser cero.

  - Combinación con `read` y sentencias `if`:

```
#!/bin/bash
printf "%s\n" "Introduzca los nombres de archivos o <control-d> para finalizar"
while read -p "Archivo ?" ARCHIVO ;
do
   if test -f "$ARCHIVO" ;
   then
      printf "%s\n" "El archivo existe"
   else
      printf "%s\n" "El archivo NO existe"
   fi
done
```

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Saber construir instrucciones de bucles: comandos `for`, `while` y `until`**
    - El bucle `while` permite repetir las declaraciones especificadas mientras se cumpla la condición. El numero de repeticiones no se conoce a priori y puede ser cero.
  - Bucles infinitos / variables booleanas. Sentencia `break`:

```
#!/bin/bash
printf "%s\n" "Introduzca los nombres de archivos o teclea FIN"
while true ;
do
   read -p "Archivo ?" ARCHIVO
   if [ "$ARCHIVO" = "FIN" ] ; then
      break
   elif test -f "$ARCHIVO" ; then
      printf "%s\n" "El archivo existe"
   else
      printf "%s\n" "El archivo NO existe"
```

```
#!/bin/bash
printf "%s\n" "Introduzca los nombres de archivos o teclea FIN"
sigue=true
while $sigue ;
do
   read -p "Archivo ?" ARCHIVO
   if [ "$ARCHIVO" = "FIN" ] ; then
      sigue=false
   elif test -f "$ARCHIVO" ; then
      printf "%s\n" "El archivo existe"
   else
      printf "%s\n" "El archivo NO existe"
   fi
done
```

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Saber construir instrucciones de bucles: comandos `for`, `while` y `until`**
    - El bucle `while` permite repetir las declaraciones especificadas mientras se cumpla la condición. El numero de repeticiones no se conoce a priori y puede ser cero.
  - Saltar instrucciones: uso de `continue`:

```
#!/bin/bash
for n in {1..9} ## Ver uso de llaves en sesiones anteriores
do
   x=$RANDOM
   [ $x -le 20000 ] && continue
   echo "n=$n x=$x"
done
```

  - Uso habitual para el desarrollo de menús.

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Saber construir instrucciones de bucles: comandos `for`, `while` y `until`**
    - El bucle `while` permite repetir las declaraciones especificadas mientras se cumpla la condición. El numero de repeticiones no se conoce a priori y puede ser cero.

  - Uso alternativo `until`: Repetir hasta que la condición sea cierta

```
until expresión;
do
   declaraciones …
done
```

```
#!/bin/bash
n=1
until [ $n -gt 10 ]
do
   echo "$n"
```

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Aprender a utilizar funciones**
    - Ejecución más rápida que llamar a un guión (se ejecutan dentro de la memoria del proceso bash que las utiliza)
    - Declaración (dos alternativas):

```
function nombre_fn  {
   declaraciones
}
```

```
nombre_fn() {
    declaraciones
}
```

    - Nomenclatura: Para evitar confusiones preceder el nombre de la función con un guión bajo `(_mifuncion ).`
    - Variables locales: `local f="$1"`
    - `Llamada a funciones y resultados:`

```
if ( si_existe_file "$1" )
```

```
resultado = `_dbl`
```

Práctica 5: Programación del shell

- OBJETIVOS:
  - **Practicar con ejemplos realistas de guiones**
    - Eliminar directorios vacíos
    - Mostrar información del sistema en una página html
    - Adaptar el guión al sistema operativo donde se ejecuta
    - Mostrar una raya girando mientras se ejecuta una orden

Práctica 5: Programación del shell

- COMANDOS:
  - `:`
  - `for`
  - `while`
  - `until`
  - `case`
  - `select`
  - `seq`
  - `read`
  - `break`
  - `true`
  - `continue`
  - `return`
  - `tar`
  - `gzip`
  - `cut`