

CAPÍTULO 7: DEPURACION DE SCRIPTS

7.1. Localice los errores del siguiente script.

```
#!/bin/bash
#Depurame.sh

if ( $(whoami) -ne root ) ; then
    echo "Debe ser root para ejecutar el script"
    exit 1
fi

echo "El root es todopoderoso"
```

Se han cometido varios errores al escribir el script:

1. La comparativa de valores se realiza entre corchetes y no entre paréntesis.
2. La comparación de cadena utiliza el operador `!=` en lugar de `-ne` (enteros).

El script correcto es el siguiente

```
#!/bin/bash
#DepurameOK.sh

if [ $(whoami) != root ] ; then
    echo "Debe ser root para ejecutar el script"
    exit 1
fi

echo "El root es todopoderoso"
```

7.2. Realice el trazado del siguiente script utilizando el depurador desarrollado en la unidad.

```
#!/bin/bash
echo "Comenzando..."

if [[ $1 -eq 10 ]] ; then
    if [[ $2 -eq 100 ]] ; then
        echo "Hacer algo"
    fi
fi
```

Para demostrar el funcionamiento del depurado del script se han

llevado cabo dos ejecuciones. En la primera (sin parámetros) se coloca un punto de ruptura en una línea inalcanzable por dichas condiciones, por lo que aún teniendo el punto de ruptura el script finaliza.

```
programacion@shell:~$ ./bashdb.sh depurar.sh
Parado en línea 1
bashbd>x
Traza activada
bashbd>p
0: : #!/bin/bash
1: :>echo "Comenzando..."
2: :
3: : if [[ $1 -eq 10 ]] ; then
4: :   if [[ $2 -eq 100 ]] ; then
5: :     echo "Hacer algo"
6: :   fi
7: : fi
bashbd>bp 5
bashbd>g
Comenzando...
+ :3:
programacion@shell:~$
```

En la segunda ejecución se le asignan los parámetros adecuados para alcanzar el punto de ruptura ubicado en el mismo lugar, con lo que se detiene en tal punto.

```
programacion@shell:~$ ./bashdb.sh depurar.sh 10 100
Parado en línea 1
bashbd>x
Traza activada
bashbd>p
0: : #!/bin/bash
1: :>echo "Comenzando..."
2: :
3: : if [[ $1 -eq 10 ]] ; then
4: :   if [[ $2 -eq 100 ]] ; then
5: :     echo "Hacer algo"
6: :   fi
7: : fi
bashbd>bp 5
bashbd>g
Comenzando...
+ :3:
+ :4:if [[ $1 -eq 10 ]] ; then
+ :5: if [[ $2 -eq 100 ]] ; then
Detenido en breakpoint en la línea 5
bashbd>p
0: : #!/bin/bash
1: : echo "Comenzando..."
2: :
3: : if [[ $1 -eq 10 ]] ; then
4: :   if [[ $2 -eq 100 ]] ; then
5: :>     echo "Hacer algo"
```

```
6: : fi
7: : fi
bashbd>g
Hacer algo
programacion@shell:~$
```

7.3. Trazado interno condicional. Ejecute el siguiente script con el valor de la variable `_DEBUG` en *on* y en *off* y compruebe que sucede.

```
#!/bin/bash
_DEBUG="on"
function DEBUG()
{
[ "$_DEBUG" == "on" ] && $@
}

DEBUG echo 'Depurando...'

DEBUG set -x
a=2
b=3
c=$(( $a + $b ))
DEBUG set +x
echo "$a + $b = $c"
```

Con el valor de la variable a *on*, se lleva a cabo una traza del script. Si dicha variable se establece a *off* el trazado se omite y el script se ejecuta normalmente.

```
programacion@shell:~$ ./debugScript.sh
Depurando...
+ a=2
+ b=3
+ c=5
+ DEBUG set +x
+ '[' on == on ']'
+ set +x
2 + 3 = 5
programacion@shell:~$ ./debugScript.sh
2 + 3 = 5
```