

MÓDULO 6 - DEPURACIÓN DE TRABAJOS

AUTOR: DANIEL PÉREZ RUIZ

EJERCICIO 6.1

```
#!/bin/bash

if ! echo $PATH | /bin/egrep -q "(^|:)$1($|:)"
then
    if [[ $2 == "after" ]] #Faltaban dobles corchetes
    then
        PATH=$PATH:$1
    else
        PATH=$1:$PATH
    fi
else
    echo "$1 ya está en el path"
fi
```

Se tiene que usar "source", ya que si no se pone la variable "PATH" cambiará en el shell hijo y no en el padre.

EJERCICIO 6.2

Los guiones son correctos, y no ha sido necesario usar depuración, sino solo utilizar "echo" en algunas partes para controlar ciertas variables.

EJERCICIO 6.3

```
#!/bin/bash

#Hago lista
top -bn 1 > temporal

#Cuento las líneas válidas
lineas=`cat monoloco|wc -l`
lineas_validas=$(( $lineas - 8 ])

#Archivo con procesos de memoria(están ordenados, así que el último es el que más consume)
cat temporal|tail -n $lineas_validas|sort -k 10 > archivo
cat archivo|tail -n 1|tr -s " " "|cut -d " " -f 13

#Elimino basura generada
rm temporal
rm archivo
```

EJERCICIO 6.4

```
#!/bin/bash

i=1

while true
do
    printf "$i "
    sleep 1
    i=$((i+1))
done
```

- Para pararlo desde otro terminal podríamos usar:

```
$ kill -STOP $(ps -A -o pid,cmd | grep <nombre_guion> | head -n 1 | cut -d " " -f 1)
```

- Para renaudarlo:

```
$ kill -CONT $(ps -A -o pid,cmd | grep <nombre_guion> | head -n 1 | cut -d " " -f 1)
```

- Para finalizarlo:

```
$ kill -TERM $(ps -A -o pid,cmd | grep <nombre_guion> | head -n 1 | cut -d " " -f 1)
```

EJERCICIO 6.5

Sí que se puede: con la orden "jobs" obtenemos una lista de los procesos en segundo plano y su estado. Podemos escribir "kill -9 %<numero_proceso>", y se finalizará el proceso, o bien se podría buscar el PID desde otro terminal.

EJERCICIO 6.6

```
top -u nombre_usuario
```