

Scripts Bash

1. Fes un script que ens indiqui si la tecla que hem polsat és una lletra, un número o un altre caràcter.

```
#!/bin/bash

echo "Entra un caracter:"

read car

if [[ $car =~ ^[A-Za-z]$ ]]
then
    echo "Es una letra"
else
    echo "Es un numero"
fi
```

2. Fes un script que quan l'executem ens digui quin dia de la setmana es (dilluns, dimarts, dimecres, dijous, divendres, dissabte o diumenge)

```
#!/bin/bash

dia=$(date +%A)

cat diaSemana.txt | grep $dia | awk '{print $2}'
```

3. Fes un script que ens doni la hora escrita segons el següent format: "dotze hores i vint-i-tres minuts"

```
#!/bin/bash

hora=$(date | awk '{print $4}' | awk -F ":" '{print $1}')
#echo $hora

minuts=$(date | awk '{print $4}' | awk -F ":" '{print $2}')
#echo $minuts

hori=$(cat nums.txt | awk -v lhora=$hora '{print $lhora}')

mini=$(cat nums.txt | awk -v lmini=$minuts '{print $lmini}')

if [[ $mini = 0 ]]; then
    echo "son les $hori en punt"
fi
echo "son les $hori i $mini"
```

4. Fes un script que ens calculi el factorial d'un nombre

```
#!/bin/bash

read -p "Dame un numero para el factorial: " numInput

count=1
total=1

while [[ $count -lt $numInput ]]
do
    total=$((total * $count))
    ((count++))
done
echo $total
```

5. Fes un script que ens digui si un nombre és primer

```
#!/bin/bash

read -p "Dame un numero para comprobar: " numInput

count=$((numInput-1))

while [[ $count -gt 0 ]]
do
    #echo $count
    remn=$((numInput % $count))
    if [[ $remn -eq 0 && $count -ne 1 ]];
    then
        noes=1
    fi

    count=$((count-1))
done

if [[ $noes -ne 1 ]]; then
    echo "Es primo"
fi
```

6. Fes un script que ens faci una llista del nombres primers. Per fer aquest script aprofita l'anterior, i dissenya una funció que ens digui si un nombre és o no primer.

```
#!/bin/bash

numInput=1

while [[ numInput -lt 100 ]]
do
    count=$((numInput-1))
    noes=0
    rem=1

    while [[ $count -gt 0 ]]
    do
        #echo $count
        remn=$((numInput % $count))
        if [[ $remn -eq 0 && $count -ne 1 ]];
        then
            noes=1
        fi

        count=$((count-1))
    done

    if [[ $noes -ne 1 ]]; then
        echo $numInput >> numPrimos.txt
    fi
    numInput=$((numInput+1))
done
```

7. Fes un script que li digui a l'usuari que l'executa quin és el seu "login" i quin és el seu "home"

```
#!/bin/bash

nom=$(whoami)

echo "El teu login es:" $nom

directory=$(cat /etc/passwd | grep $nom | awk -F ":" '{print $6}')

echo "El teu home directory es:" $directory
```

8. Fes un script que ens mostri per pantalla quin és el "UID" de cadascun dels usuaris (només dels usuaris que no son de sistema) de la màquina ("login" "UID")

```
#!/bin/bash
echo ""
echo "#####"
echo -e "\tLogin" ---- "UID"
echo "#####"
cat /etc/passwd | grep -v "bash" | awk -F ":" '{print $1,"--",$3}'
echo ""
```

9. Fes un script que ens faci un llistat de tots els subdirectoris que hi ha en un determinat directori entrat per línia de comandament. Si el directori no existeix, surt del script amb un missatge d'error

```
#!/bin/bash

read -p "Donam un directori a buscar: " directory

ls -R $directory 2>/dev/null

if [[ $? -ne 0 ]]; then
    echo "El directori seleccionat no existeix"
fi
```

10. Per el script anterior, fes el llistat de tots els subdirectoris amb dos nivells de profunditat.

```
#!/bin/bash

read -p "Donam un directori a buscar: " directory
# canviat una mica ja que requereix profunditat
find $directory -mindepth 2 -maxdepth 2 -type d
# afegit com a variable per que per algun motiu avegades fallaba, així va bé
correct=$(echo $?)
if [[ $correct -ne 0 ]]; then
    echo "El directori seleccionat no existeix"
fi
```

11. Fes un script que ens busqui un fitxer dins d'un directori. En cas que existeixi ens digui el propietari del mateix. El script ens demanarà un nou directori, en cas que el que fiquem no existeixi

```
#!/bin/bash

noestabe=0
while [[ noestabe -eq 0 ]];
do
    read -p "Donam un directori a buscar: " directory
    read -p "Digam el fitxer a buscar: " fitxer

    # trobar en directori arxiu que contingui $fitxer
    find ${directory}/*${fitxer}* 2>/dev/null

    # si dona codi correcte sortir del loop
    correct=$(echo $?)
    if [[ $correct -eq 0 ]]; then
        noestabe=1
    fi
done
```

12. Fes el mateix script que l'anterior, però amb dos nivells de profunditat (directori i subdirectori)

```
#!/bin/bash
noestabe=0
while [[ noestabe -eq 0 ]];
do
    read -p "Donam un directori a buscar: " directory
    read -p "Digam el fitxer a buscar: " fitxer

    # trobar en directori arxiu que contingui $fitxer
    find ${directory} -maxdepth 2 -name ${fitxer} -type f 2>/dev/null

    # si dona codi correcte sortir del loop
    correct=$(echo $? )
    if [[ $correct -eq 0 ]]; then
        noestabe=1
    fi
done
```

13. Executa tres pings a diferents ips, en background i redirigint la seva sortida a /dev/null. Fes un script que ens mostri una llista amb el PID associat a cada ping ("numero" "PID" "pagina a la que fa ping"), i posteriorment et preguntí de quin numero de la llista vols aturar el ping.

```
#!/bin/bash

ping -c 1 8.8.8.8 >/dev/null &
ping -c 1 4.4.4.4 >/dev/null &
ping -c 1 www.google.es >/dev/null &

echo " "
ps aux | grep "ping" | grep -v "grep" | awk '{print NR,$2,$14}'
echo " "
```

14. Fes un script que faci un llista dels usuaris que han fet login en el sistema. El script s'executarà cada minut, la sortida ha de ser de la forma: "data" "hora" "usuaris logejats" i s'ha d'escriure sobre un fitxer. (pots utilitzar CRON)

```
echo -e "Username\t" " " "Data\t" "Hora Login" " " "N. Users\t" | tee
users_log.txt
num_user=$(who | awk '{print $1}' | sort -u | wc -l)
who | awk -v num=$num_user '{print $1, "\t\t", $3, " \t ", $4, "\t ",
num}' | sort -u -k 1 | tee -a users_log.txt

# per a que repeteixi cada minut fariam servir crontab -e i hi escriuriem "
* * * * * Ejercici14.sh" per a repetir cada minut cada dia cada mes i
setmana
```

15. Fes un script que guardi en un fitxer els logs de quan hem fet un login al sistema i quan hem sortit del sistema (Has d'utilitzar els `.bashrc` i `.bash_logout`)

Podem afegir dins de `bashrc` i dins de `bash_logout` un redireccionament creant un arxiu, com aquests arxius s'executen al loguejar (`bashrc`) i al logout (`bash_logout`) aquests guardaran els "logs"

En `bashrc`:

```
Echo $(whoami) s'ha loguejat $(date)
```

En `bash_logout`:

```
Echo $(whoami) s'ha desloguejat $(date)
```