

MÒDUL 0374 – Administració avançada de sistemes operatius

Administració de sistemes operatius

Activitat pràctica 1
Bash scripting

RA7: Utilitza llenguatges de guions en sistemes operatius, descrivint la seva aplicació i administrant serveis del sistema operatiu.

Observacions

Responen en un color diferenciador.

L'objectiu d'aquesta activitat és la de definir scripts en bash (Linux) que permetin automatitzar i operar sistemes Linux de forma de desatesa, controlant els errors que es produeixin. El codi cal que estigui documentat de forma correcta.

Els punts que es treballaran en aquesta activitat són:

- Variables d'entorn
- Pas de paràmetres a funcions
- Estructures iteratives
- Lectura i escriptura de/a fitxers

RECOMANABLE: Per cada exercici, cal posar el script generat i l'execució del mateix on es vegi el resultat. Servirà d'apunts per a vosaltres.

Generar un fitxer .sh per cada exercici de nom exerciciX.sh on la X és el número d'exercici.

Variables d'entorn

1.	<p>Substituint nom_d'usuari pel valor de la variable \$USER, escriviu una ordre que mostri la frase següent:</p> <p>"El \$USER: nom d'usuari està llegint el llibre " _el llibre que estiguen llegint o que hàgiu llegit_"</p>
	

2.

Creeu un un script que mostri:

Hola món, son les %hora% del dia %DiaMesAny% i soc l'usuari %nomusuari%

```
isard@ubuntu-24:~$ source exercici2.sh
Hola món, son les 10 del dia 08/01/2026 i soc l'usuari Adam
isard@ubuntu-24:~$
```

GNU nano 7.2

exercici2.sh

```
#!/bin/bash
nomusuari=Adam
hora=$(/usr/bin/date +%H)
DiaMesAny=$(/usr/bin/date +%d/%m/%Y)
echo "Hola món, son les $hora del dia $DiaMesAny i soc l'usuari $nomusuari"
```

Creeu un directori ~/scripts per guardar els vostres programes, moveu el fitxer generat al punt anterior al directori ~/scripts i afegiu el directori al contingut de la variable PATH amb la següent comanda:

3.

```
mkdir ~/scripts
mv ~/llibre.sh ~/scripts
export PATH="$PATH:~/scripts"
```

Cal provar que ara l'execució de l'script funciona sense haver d'accedir al directori.



isard@ubuntu-24: ~

GNU nano 7.2

exercici3.sh

```
#!/bin/bash
mkdir ~/scripts
mv ~/exercici2.sh ~/scripts
export PATH="$PATH:~/scripts"
```

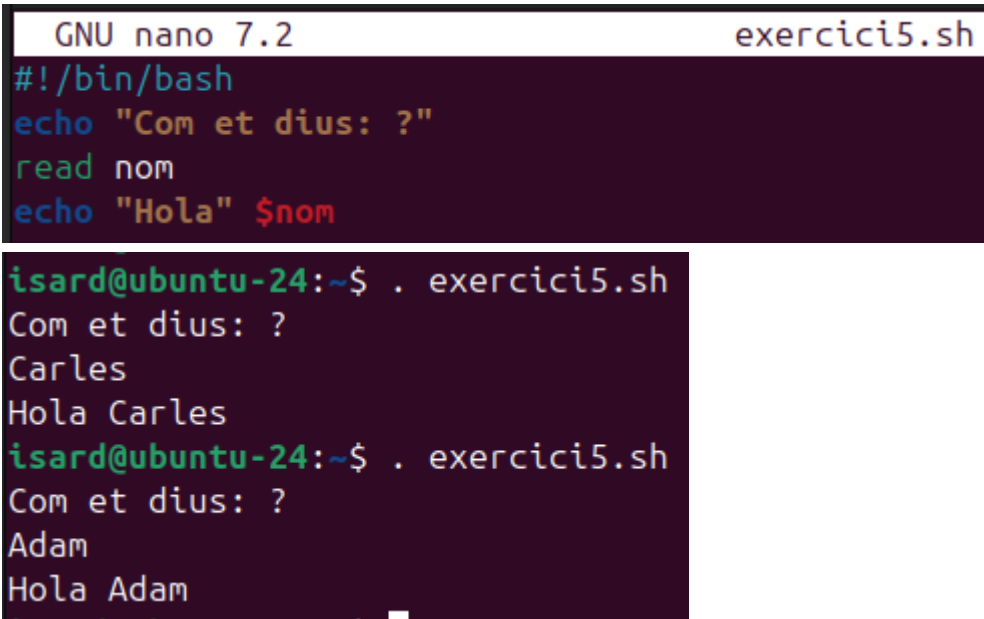
```
isard@ubuntu-24:~$ ls
Baixades  Escriptori  exercici2.sh  Imatges  Plantilles  snap
Documents exercici1.sh exercici3.sh  Música  Públic      Vídeos
isard@ubuntu-24:~$ . exercici3.sh
isard@ubuntu-24:~$ ls
Baixades  Escriptori  exercici3.sh  Música  Públic  snap
Documents exercici1.sh Imatges      Plantilles scripts  Vídeos
isard@ubuntu-24:~$ ls scripts
exercici2.sh
isard@ubuntu-24:~$
```

```
isard@ubuntu-24:~$ . exercici2.sh
Hola món, son les 10 del dia 09/01/2026 i soc l'usuari Adam
```

Pas de paràmetres

És molt habitual que els scripts rebin paràmetres que utilitzaran per a aplicar les diferents accions sobre els paràmetres rebuts.

4.	<p>Fer un script que rebi 3 paràmetres a l'hora d'executar: blau , verd, vermell</p> <p>La sortida del programa ha de ser la següent similar a la següent:</p> <p>Nom de l'script: ./exercici4.sh</p> <p>Valor del primer paràmetre de l'script: blau</p> <p>Valor del segon paràmetre de l'script: verd</p> <p>Valor del tercer paràmetre de l'script: vermell</p>
	<pre>GNU nano 7.2 exercici4.sh * #!/bin/bash echo "Nom de l'script: " \$0 echo "valor del primer paràmetre de l'script: " \$1 echo "valor del segon paràmetre de l'script: " \$2 echo "valor del tercer paràmetre de l'script: " \$3 isard@ubuntu-24:~\$./exercici4.sh blau verd vermell Nom de l'script: ./exercici4.sh valor del primer paràmetre de l'script: blau valor del segon paràmetre de l'script: verd valor del tercer paràmetre de l'script: vermell</pre>

5.	<p>Fer un script que us demani el vostre nom al terminal , l'imprimeixi per pantalla , generant la sortida següent:</p> <p>Com et dius: ? Carles Hola Carles</p>
	 <pre>GNU nano 7.2 exercici5.sh #!/bin/bash echo "Com et dius: ?" read nom echo "Hola" \$nom isard@ubuntu-24:~\$. exercici5.sh Com et dius: ? Carles Hola Carles isard@ubuntu-24:~\$. exercici5.sh Com et dius: ? Adam Hola Adam</pre>

6. Fer un script que demani 2 nombres i mostri la seva suma, resta, multiplicació i divisió.

```
isard@ubuntu-24:~$ . exercici6.sh
1º nombre: ?
8
2n nombre: ?
2
suma del primer numero i el segon= 10
resta del primer numero i el segon= 6
multiplicació del primer numero pel segon= 16
divisió del primer numero entre el segon= 4
isard@ubuntu-24:~$ . exercici6.sh
1º nombre: ?
500
2n nombre: ?
282
suma del primer numero i el segon= 782
resta del primer numero i el segon= 218
multiplicació del primer numero pel segon= 141000
divisió del primer numero entre el segon= 1
```

GNU nano 7.2

exercici6.sh

```
#!/bin/bash

echo "1º nombre: ?"
read num1
echo "2n nombre: ?"
read num2
let "z = num1 + num2"
let "x = num1 - num2"
let "c = num1 * num2"
let "v = num1 / num2"
echo "suma del primer numero i el segon= " $z
echo "resta del primer numero i el segon= "$x
echo "multiplicació del primer numero pel segon= "$c
echo "divisió del primer numero entre el segon= "$v
```

7.

Fer un script que indiqui si un nom donat com a paràmetre és un fitxer regular. L'script ha de controlar que se li passi només un paràmetre. Si no és així ha de mostrar un missatge d'error a l'usuari que li indiqui com cal executar el script, i retornar un codi d'error 1.

GNU nano 7.2

exercici7.sh

```
#!/bin/bash
#Adam Rkaini
test -f $1
parametre=$?
if [ $# -ne 1 ];then
echo "només s'accepta un parametre"
exit
elif [ $parametre -eq 0 ];then
echo "aquest parametre es un fitxer!"
else
echo "aquest parametre NO es un fitxer!"
fi
```

```
isard@ubuntu-24:~$ ls
1          exercici1.sh  exercici6.sh      Imatges      scripts
Baixades   exercici3.sh  exercici7.sh      Música       snap
Documents  exercici4.sh  exercici7.sh.save Plantilles   Videos
Escriptori exercici5.sh  fitxer            Públic

isard@ubuntu-24:~$ ./exercici7.sh hola
aquest parametre NO es un fitxer!
isard@ubuntu-24:~$ ./exercici7.sh exercici3.sh
aquest parametre es un fitxer!
isard@ubuntu-24:~$ ./exercici7.sh exercici3.sh hola
només s'accepta un parametre
```

8.

Fer un script que rebí un argument de manera que:

- Si és un directori mostri el missatge "El contingut del directori <nom_directori> és:" i llisti el seu contingut.
- Si l'argument no és cap directori ha de donar un missatge informatiu.

L'script ha de controlar que se li passi només un paràmetre. Si no és així ha de mostrar un missatge d'error a l'usuari i que li indiqui com cal executar el script.

```
GNU nano 7.2 exercici8.sh
#!/bin/bash
#Adam Rkaini
test -d $1
parametre=$?
if [ $# -ne 1 ];then
echo "només s'accepta un parametre"
exit
elif [ $parametre -eq 0 ];then
echo "el contingut del directori $1 és:"
ls $1
else
echo "aquest parametre NO es un directori!"
fi
```

```
isard@ubuntu-24:~$ ./exercici8.sh Imatges haoia
només s'accepta un parametre
isard@ubuntu-24:~$ ./exercici8.sh Imatges
el contingut del directori Imatges és:
isard@ubuntu-24:~$ ls Imatges/
isard@ubuntu-24:~$ ./exercici8.sh ~
el contingut del directori /home/isard és:
1          exercici1.sh exercici6.sh      fitxer          Plantilles  Vídeos
Baixades   exercici3.sh  exercici7.sh      Imatges         Públic
Documents  exercici4.sh  exercici7.sh.save Música          scripts
Escriptori exercici5.sh  exercici8.sh      nano.2270.save  snap
isard@ubuntu-24:~$
```


Estructures de control

El llenguatge bash disposa de diferents estructures de control que ens permeten controlar el fluxe i les iteracions de les execucions.

9.	<p>Fer un script que demani el valor d'una nota (un nombre enter) i ens digui si la nota és una D (0, 1, 2), una C- (3, 4), una C+ (5, 6), una B (7, 8) o una A (9,10). Cal comprovar que els valors que s'informen són entre 0 i 10, sinó donar un error.</p> <p>Utilitzar alguna estructura de tipus if-else</p>
----	---

```
GNU nano 7.2 exercici9.sh
#!/bin/bash
#Adam Rkaini

echo "Quin es el valor de la nota?"
read nota

if [ $nota -lt 0 ] || [ $nota -gt 10 ];then
echo "La nota ha de ser entre 1 i 10"
exit 1
elif [ $nota -ge 0 ] && [ $nota -le 2 ];then
echo "La nota és una D"
elif [ $nota -ge 3 ] && [ $nota -le 4 ];then
echo "La nota es una C-"
elif [ $nota -ge 5 ] && [ $nota -le 6 ];then
echo "La nota es una C+"
elif [ $nota -ge 7 ] && [ $nota -le 8 ];then
echo "La nota es una B"
elif [ $nota -ge 9 ] && [ $nota -le 10 ];then
echo "La nota es una A"
fi
```

```
isard@ubuntu-24:~$ ./exercici9.sh
Quin es el valor de la nota?
0
La nota és una D
isard@ubuntu-24:~$ ./exercici9.sh
Quin es el valor de la nota?
10
La nota es una A
isard@ubuntu-24:~$ ./exercici9.sh
Quin es el valor de la nota?
11
La nota ha de ser entre 1 i 10
isard@ubuntu-24:~$ ./exercici9.sh
Quin es el valor de la nota?
-2
La nota ha de ser entre 1 i 10
isard@ubuntu-24:~$ ./exercici9.sh
Quin es el valor de la nota?
8
La nota es una B
isard@ubuntu-24:~$ ./exercici9.sh
Quin es el valor de la nota?
2
La nota és una D
```

10. Fer el mateix script que implementat amb una escala if-else, però ara utilitzant una estructura **case**

```
Quina es la teva nota
3
La nota es una C-
isard@ubuntu-24:~$ ./exercici10.sh
Quina es la teva nota
2
La nota es una D
isard@ubuntu-24:~$ ./exercici10.sh
Quina es la teva nota
1
La nota es una D
isard@ubuntu-24:~$ ./exercici10.sh
Quina es la teva nota
0
La nota es una D
isard@ubuntu-24:~$ ./exercici10.sh
Quina es la teva nota
11
La nota ha de ser entre 0 i 10 exit 1
isard@ubuntu-24:~$ ./exercici10.sh
Quina es la teva nota
-1
La nota ha de ser entre 0 i 10 exit 1
isard@ubuntu-24:~$
```

GNU nano 7.2

exercici10.sh

```
#!/bin/bash
#Adam Rkaini

echo "Quina es la teva nota"
read nota
case "$nota" in
[0-2]) echo "La nota es una D" ;;
[3-4]) echo "La nota es una C-" ;;
[5-6]) echo "La nota es una C+" ;;
[7-8]) echo "La nota es una B" ;;
9|10) echo "La nota es una A";;
*) echo "La nota ha de ser entre 0 i 10" exit 1 ;;
esac
```

Scripts de control de sistema

11.	Fer un script que calculi l'ocupació de disc de cada directori de /home. Podeu obtenir la llista d'aquests directoris executant l'ordre ls dins del directori /home. Per calcular l'ocupació del directori podeu utilitzar l'ordre du (disk usage).
-----	---

```
isard@ubuntu-24:~$ ./exercici11.sh
Aquests son els directoris dins del directori home
isard
Aquesta es el l'ocupacio dels directoris
8      /home/isard/scripts
4      /home/isard/Públic
4      /home/isard/Vídeos
4      /home/isard/Imatges
4      /home/isard/Música
4      /home/isard/Plantilles
4      /home/isard/.ssh
4      /home/isard/Documents
12     /home/isard/.gnupg
12     /home/isard/.cache/update-manager-core
0      /home/isard/.cache/ubuntu-software
```

GNU nano 7.2

exercici11.sh

```
#!/bin/bash
#Adam Rkaini

echo "Aquests son els directoris dins del directori home"
ls /home
echo "Aquesta es el l'ocupacio dels directoris"
du /home
```

12.

Fer el mateix script, però rebent per paràmetre el nom dels directoris dels quals volem conèixer l'espai que ocupen al disc.
Cal comprovar que el directori facilitat és correcte i existeix.

```
isard@ubuntu-24:~$ ./exercici12.sh
quin directori vols comprovar
hola
Aquest directori no es valid o no existeix
isard@ubuntu-24:~$ ./exercici12.sh
quin directori vols comprovar
Baixades
Aquest es l'espai que ocupa al disc
224K    Baixades
isard@ubuntu-24:~$
```

```
GNU nano 7.2                                exercici12.sh
#!/bin/bash
#Adam Rkaini

echo "quin directori vols comprovar"
read directori
test -d $directori
if [ $? -ne 0 ];then
echo "Aquest directori no es valid o no existeix"
exit 1
else
echo "Aquest es l'espai que ocupa al disc"
du -h $directori
fi
```

13.

Fer un script que mostri els usuaris del sistema que poden iniciar sessió i permeti accedir al sistema amb l'usuari seleccionat. Si l'usuari seleccionat no existeix té que mostrar el missatge de "L'usuari no existeix"

```
isard@ubuntu-24:~$ ./exercici13.sh
Aquests son els usuaris del sistema que poden iniciar sessió:
root
isard
amb quin usuari desitjes iniciar sessió?
caca
L'usuari no existeix
isard@ubuntu-24:~$ ./exercici13.sh
Aquests son els usuaris del sistema que poden iniciar sessió:
root
isard
amb quin usuari desitjes iniciar sessió?
isard
Contrasenya:
isard@ubuntu-24:~$
```

GNU nano 7.2

exercici13.sh

```
#!/bin/bash
#Adam Rkaini
echo "Aquests son els usuaris del sistema que poden iniciar sessió:"

cat /etc/passwd | grep /bin/bash | cut -d ":" -f 1

echo "amb quin usuari desitjes iniciar sessió?"

read usuario
cat /etc/passwd | grep $usuario >> /dev/null
if [ $? -ne 0 ];then
echo "L'usuari no existeix"
exit
else
su $usuario
fi
```

Lectura de fitxers CSV

Una tasca molt habitual és la de llegir fitxers en format CSV (separat per ,) on hi ha informació com per exemple usuaris o directoris / fitxers a crear.

14.	<p>A partir d'un fitxer CSV (es passarà el nom del fitxer com a argument a l'script) que contindrà el següent format:</p> <p><i>Nom d'usuari, shell, home, password, email</i> → ex ("jcasas,/bin/bash,/home/jordi,jordi.casas@itb.cat")</p> <p>Fer un script que llegeixi cada línia i faci les següents operacions:</p> <ul style="list-style-type: none"> • Comprova que l'usuari no existeixi al sistema. Si existeix indica "Usuari \$nom_usuari ja existent al sistema" • Si no existeix crea l'usuari al sistema amb les dades indicades al fitxer. Un cop creat es traurà el missatge de "Usuari \$nom_usuari donat d'alta al sistema" <p>L'script ha de tenir el control dels paràmetres correctes i verificar que el fitxer existeix. Així mateix tot missatge que es genera de l'script, a part de sortir per pantalla, s'ha de registrar en un fitxer logs.txt</p>
	<pre>isard@ubuntu-24:~\$ cat exercici14.csv ramon,/bin/bash,/home/ramon,lola,ramon.casas@itb.cat Adam,/bin/bash,/home/Adam,paco,adam.casas@itb.cat isard,/bin/bash,/home/isard,pirineus,isard.casas@itb.cat Dani,/bin/bash,/home/Dani,nana,Dani.casas@itb.cat tete,/bin/bash,/home/tete,jefe,tete.casas@itb.cat</pre> <pre>isard@ubuntu-24:~\$./exercici14.sh exercici14.csv [sudo] contrasenya per a isard: CONTRASENYA DOLENTA: La contrasenya és inferior als 8 caràcters Usuari ramon donat d'alta al sistema CONTRASENYA DOLENTA: La contrasenya és inferior als 8 caràcters Usuari Adam donat d'alta al sistema Usuari isard ja existent al sistema CONTRASENYA DOLENTA: La contrasenya és inferior als 8 caràcters Usuari Dani donat d'alta al sistema CONTRASENYA DOLENTA: La contrasenya és inferior als 8 caràcters Usuari tete donat d'alta al sistema useradd: el directori personal «» no és vàlid chpasswd: (user) pam_chauthtok() failed, error: Error de manipulació del testimoni d'autenticació chpasswd: (línia 1, usuari) contrasenya sense canviar Usuari donat d'alta al sistema</pre>

```
GNU nano 7.2 exercici14.sh
#!/bin/bash
#Adam Rkaini

#comprova que el fitxer existeix si no existeix codi d'error 1
test -f $1

if [ $? -ne 0 ];then
echo "Aquest arxiu no existeix"
exit 1
fi
#si existeix comença un bucle on mentre hi hagi una coma llegeix els
#diferents parametres
while IFS=, read -r usuari shell home pass email
do
#comprova que l'usuari de la primera coma existeix al sistema
id "$usuari" &> /dev/null
if [ $? -eq 0 ]; then
echo "Usuari $usuari ja existent al sistema" | tee -a logs.txt
else
sudo useradd -m -d "$home" -s "$shell" "$usuari"

echo "$usuari:$pass" | sudo chpasswd
echo "Usuari $usuari donat d'alta al sistema" | tee -a logs.txt
fi
done < "$1"
```

```
isard@ubuntu-24:~$ cat logs.txt
Usuari ramon donat d'alta al sistema
Usuari Adam donat d'alta al sistema
Usuari isard ja existent al sistema
Usuari Dani donat d'alta al sistema
Usuari tete donat d'alta al sistema
Usuari donat d'alta al sistema
```


Script de còpia de seguretat

15.	<p>Realitzar un script final que farà una còpia de seguretat d'una carpeta que es passi per paràmetre. El segon paràmetre serà el directori on cal guardar el fitxer de la còpia. A tenir en compte:</p> <ul style="list-style-type: none">• La còpia cal fer-la amb la comanda tar i que comprimeixi el contingut.• El nom del fitxer ha de ser backup_nom_directori_YYYYMMDD.tar.gz• L'script cal que controli primer que no existeixi ja un fitxer prèviament amb el mateix nom. Si és així, l'script donarà error i parará, retornant codi 2• Si no existeix el directori a fer-ne la còpia, l'script donarà error i parará, retornant codi 1
	<pre>isard@ubuntu-24:~\$./exercici15.sh scripts Escriptori Còpia realitzada correctament: Escriptori/backup_scripts_20260115.tar.gz isard@ubuntu-24:~\$ ls Escriptori/ backup_scripts_20260115.tar.gz isard@ubuntu-24:~\$ isard@ubuntu-24:~\$./exercici15.sh scripts/ Escriptori/ Error: El fitxer de còpia Escriptori//backup_scripts_20260115.tar.gz ja existeix.</pre>

```
isard@ubuntu-24:~$ ./exercici15.sh scripts
Ús: ./exercici15.sh <directori_origen> <directori_desti>
```

```
GNU nano 7.2 exercici15.sh
#!/bin/bash
#Adam Rkaini

# Exercici 15: Backup amb control d'errors i codis de sortida
# Ús: ./exercici15.sh /ruta/origen /ruta/desti

# 1. Validem que rebem 2 paràmetres
if [ $# -ne 2 ]; then
    echo "Ús: $0 <directori_origen> <directori_desti>"
    exit 1
fi

ORIGEN=$1
DESTI=$2

# Obtenim el nom de la carpeta sense la ruta (ex: /home/jordi -> jordi)
NOM_CARPETA=$(basename "$ORIGEN")
DATE=$(date +%Y%m%d)

# Construïm el nom final del fitxer
NOM_FITXER="backup_${NOM_CARPETA}_${DATE}.tar.gz"
RUTA_FINAL="$DESTI/$NOM_FITXER"

# 2. CHECK 1: Comprovar que no existeixi ja el fitxer (Codi 2)
if [ -f "$RUTA_FINAL" ]; then
    echo "Error: El fitxer de còpia $RUTA_FINAL ja existeix."
    exit 2
fi

# 3. CHECK 2: Comprovar que existeix el directori origen (Codi 1)
if [ ! -d "$ORIGEN" ]; then
    echo "Error: El directori d'origen $ORIGEN no existeix."
    exit 1
fi

# 4. Creació de la còpia
# c=create, z=gzip, f=file
# Redirigim errors a /dev/null per neteja, o ho traïem si vols veure errors del tar
tar -czf "$RUTA_FINAL" "$ORIGEN" 2>/dev/null

if [ $? -eq 0 ]; then
    echo "Còpia realitzada correctament: $RUTA_FINAL"
else
    echo "Hi ha hagut un error en crear el tar."
    exit 1
fi
```

Operadors

16. Escriu les ordres necessàries per cercar el nom d'un usuari anomenat "messi" al fitxer /etc/passwd i donar missatges diferents segons si es troba o no al fitxer d'usuaris..

```
isard@ubuntu-24:~$ ./exercici16.sh
L'usuari 'messi' NO s'ha trobat al fitxer d'usuaris.
```

```
GNU nano 7.2 exercici16.sh
#!/bin/bash
#Adam Rkaini

# Busquem la línia que comenci per messi seguit de : per evitar falsos positius (ex: messi10)
grep "^messi:" /etc/passwd > /dev/null

# Comprovem el resultat de l'última comanda ($?)
if [ $? -eq 0 ]; then
    echo "L'usuari 'messi' EXISTEIX al sistema."
else
    echo "L'usuari 'messi' NO s'ha trobat al fitxer d'usuaris."
fi
```

17. Escriu les ordres necessàries per assegurar-te que el superusuari root és qui està executant el guió de shell i donar un missatge depenent de si ho és o no.

```
isard@ubuntu-24:~$ ./exercici17.sh
No ets root. Si us plau, executa l'script com a administrador (sudo).
isard@ubuntu-24:~$ sudo ./exercici17.sh
Ets l'usuari ROOT. Tens permís per continuar.
```

```
GNU nano 7.2 exercici17.sh
#!/bin/bash
#Adam Rkaini

# id -u ens retorna l'ID de l'usuari actual. Si és 0, és root.
if [ "$(id -u)" -eq 0 ]; then
    echo "Ets l'usuari ROOT. Tens permís per continuar."
else
    echo "No ets root. Si us plau, executa l'script com a administrador (sudo)."
fi
```

AWS CLI

Es poden automatitzar les activitats amb els elements d'AWS utilitzant el CLI que ofereix AWS.

Recordar que el CLI cal configurar-lo amb les credencials correctes posant-les al fitxer `./aws/configure`.

18.	<p>Defineix un script que faci un bolcat de totes les EC2 que estan operatives en un entorn d'AWS i en generi un CSV, indicant la següent informació:</p> <ul style="list-style-type: none"> • InstanceID • Estat de la instància (running, stopped, ...) • Type (t2.small, t2.micro, ...) • Nom de la màquina
-----	--

```
eee_W_5489378@runweb204927:~$ cat exercici18.sh
#!/bin/bash
#Adam Rkaini

# Exercici 18: Generar CSV d'instàncies EC2 operatives

FITXER_SORTIDA="instancies_aws.csv"

# Capçalera del CSV
echo "InstanceID,Estat,Type,Nom" > "$FITXER_SORTIDA"

# Comanda AWS:
# 1. describe-instances: llista màquines
# 2. --filters: només les que no estiguin terminades (o només running si vols operatives 100%)
# 3. --query: extraiem només els camps que demana l'enunciat.
#    Nota: El nom sol estar al tag "Name".
# 4. --output text: format text per poder processar-lo fàcilment
# 5. tr: canviem els tabuladors que dona AWS per comes (format CSV)

aws ec2 describe-instances \
  --filters "Name=instance-state-name,Values=running" \
  --query 'Reservations[*].Instances[*].[InstanceId, State.Name, InstanceType, Tags[?Key==`Name`].Value | [0]]' \
  --output text | tr '\t' ',' >> "$FITXER_SORTIDA"

echo "Bolcat realitzat. Resultat a $FITXER_SORTIDA"

eee_W_5489378@runweb204934:~$ echo "Bolcat realitzat. Resultat a $FITXER_SORTIDA"
Bolcat realitzat. Resultat a instancies_aws.csv

eee_W_5489378@runweb204934:~$ cat instancies_aws.csv
InstanceID,Estat,Type,Nom
i-05c2f40404d20515f,running,t3.micro,Ubuntu Server
i-088c7f59aac283e5,running,t3.micro,WebSrv2
i-0a55ddcb946fe7afb,running,t3.micro,WebSrv3
i-0a55909d0bd08fd36,running,t3.micro,WebSrv1
i-06d2d9ed2663f1f4a,running,t3.micro,Gestio d'incidencia
i-00247b8ab7b4a92c3,running,t3.micro,Node Ansible
i-071d96b225e808bce,running,t3.small,Windows Server
eee_W_5489378@runweb204934:~$
```

19.	<p>Defineix un script que a partir d'un CSV que conté -com a mínim- aquesta informació, creï les màquines a AWS de forma automàtica:</p> <ul style="list-style-type: none">• Tipus d'instància• Tipus d'AMI• Security group• Nom de la instància (etiqueta)
-----	--

```
eee_W_5489378@runweb204935:~$ cat noves_maquines.csv
Type,AMI,SecurityGroup,Name
t2.micro,ami-07ff62358b87c7116,sg-0e308add29e3bd73a,ServidorWeb
t2.small,ami-07ff62358b87c7116,sg-0e308add29e3bd73a,BaseDades
eee_W_5489378@runweb204935:~$ . exercici19.sh noves_maquines.csv
```

```
-----
Processant màquina: ServidorWeb
- Tipus: t2.micro
- AMI: ami-07ff62358b87c7116
- Security Group: sg-0e308add29e3bd73a
[OK] Instància creada correctament.
```

```
-----
Processant màquina: BaseDades
- Tipus: t2.small
- AMI: ami-07ff62358b87c7116
- Security Group: sg-0e308add29e3bd73a
[OK] Instància creada correctament.
```

```
eee_W_5489378@runweb204935:~$ cat exercici19.sh
#!/bin/bash
#Crear instàncies des de CSV
#Adam Rkaini

# 1. Validar arguments
if [ -z "$1" ]; then
    echo "Error: Has d'indicar el fitxer CSV."
    echo "Ús: $0 fitxer.csv"
    exit 1
fi

FITXER_CSV=$1

# 2. Validar que el fitxer existeix
if [ ! -f "$FITXER_CSV" ]; then
    echo "Error: El fitxer $FITXER_CSV no existeix."
    exit 1
fi

# 3. Llegir el fitxer línia a línia
# IFS=, indica que la coma separa les columnes
while IFS=, read -r tipus ami seguretat nom
do
    # --- NETEJA DE DADES (IMPORTANT) ---
    # Eliminem el caràcter de retorn de carro '\r' per si el CSV ve de Windows
    nom=$(echo "$nom" | tr -d '\r')
    tipus=$(echo "$tipus" | tr -d '\r')

    # Saltem la línia si és la capçalera o està buida
    if [ "$tipus" == "Type" ] || [ -z "$tipus" ]; then
        continue
    fi

    echo "-----"
    echo "Processant màquina: $nom"
    echo "  - Tipus: $tipus"
    echo "  - AMI: $ami"
    echo "  - Security Group: $seguretat"

    # 4. Executar la comanda AWS
    aws ec2 run-instances \
        --image-id "$ami" \
        --instance-type "$tipus" \
        --security-group-ids "$seguretat" \
```

```
--tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=$nom}]" \  
--output text --query 'Instances[0].InstanceId' > /dev/null  
  
# 5. Comprovar errors  
if [ $? -eq 0 ]; then  
    echo " [OK] Instància creada correctament."  
else  
    echo " [ERROR] No s'ha pogut crear la instància."  
fi  
  
done < "$FITXER_CSV"
```