

UNITAT 3. ACTIVITATS 3.2.

SISTEMA OPERATIU

LINUX

Alumne: David Peñalver Navarro

Professor: Manuel Enguñados

Assignatura: Sistemes Informàtics

Índex

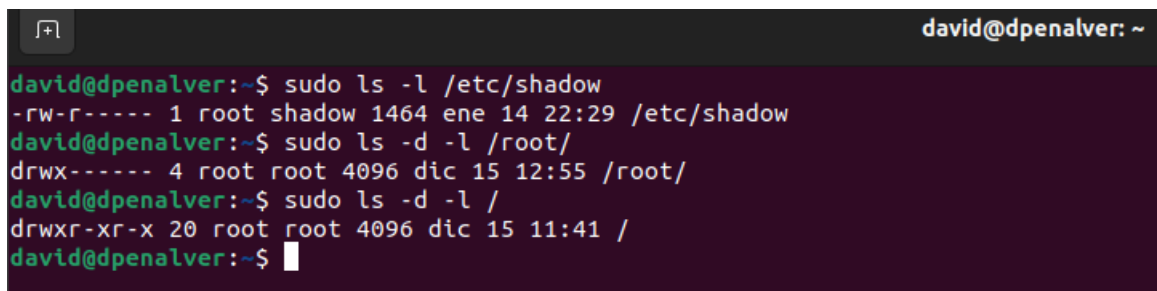
Activitats resoltes	3
1- Activitat resolta 3.17	3
2- Activitat resolta 3.18	4
3- Activitat resolta 3.19	5
4- Activitat resolta 3.20	8
5- Activitat resolta 3.21	9
6- Activitat resolta 3.22	12
7- Activitat resolta 3.23	13
8- Activitat resolta 3.24	15
9- Activitat resolta 3.25	16
10- Activitat resolta 3.26	17
11- Activitat resolta 3.27	18
12- Activitat resolta 3.28	19
13- Activitat resolta 3.29	20
Pràctica 4	21
1- Exercici 1	21
2- Exercici 2	22
3- Exercici 3	23
4- Exercici 4	24
5- Exercici 5	27
6- Exercici 6	28
7- Exercici 7	29
8- Exercici 8	30
9- Exercici 9	32

Activitats resoltes

1- Activitat resolta 3.17

Per a poder executar estes comandes amb éxit, necessitem fer ús de “sudo”. Així mateix, a l'hora d'utilitzar “ls” ficarem “-l” per a que ho faça de manera llarga, apareixent els permissos d'usuari, de grup i dels altres. Per últim, fem el “-d” per a “/” i “/root/” al ser directoris.

Així doncs, ens mostra en primer lloc un guió o una “d” si és fitxer o directori i a continuació els permissos associats a usuari, grup i altres, respectivament. Si el té, apareix la lletra i si no ho té apareix un guió.



```
david@dpenalver: ~  
david@dpenalver:~$ sudo ls -l /etc/shadow  
-rw-r----- 1 root shadow 1464 ene 14 22:29 /etc/shadow  
david@dpenalver:~$ sudo ls -d -l /root/  
drwx----- 4 root root 4096 dic 15 12:55 /root/  
david@dpenalver:~$ sudo ls -d -l /  
drwxr-xr-x 20 root root 4096 dic 15 11:41 /  
david@dpenalver:~$
```

Figura 1. Comprovant els permissos.

2- Activitat resolta 3.18

Per a que l'usuari tinga tots els privilegis menys execució, fiquem un 1. Per a que la resta no tinga, fiquem un 7 a cadascú (amaguem tots) (Figura 2).

A terminal window with a dark background. The title bar shows a window icon and the text 'david@dpنالver: ~'. The terminal content shows the prompt 'david@dpنالver:~\$' followed by the command 'umask 177' and a new prompt 'david@dpنالver:~\$' on the next line.

```
david@dpنالver:~$ umask 177
david@dpنالver:~$
```

Figura 2. Configurant el primer umask.

De cara a que tinga tots els privilegis, fique un 0; per a que el grup no tinga escrutura, fique el 2; per a que els altres no tinguen cap, fique un 7 (Figura 4).

A terminal window with a dark background. The title bar shows a window icon and the text 'david@dpنالver: ~'. The terminal content shows the prompt 'david@dpنالver:~\$' followed by the command 'umask 027' and a new prompt 'david@dpنالver:~\$' on the next line.

```
david@dpنالver:~$ umask 027
david@dpنالver:~$
```

Figura 4. Modificant el umask.

3- Activitat resolta 3.19

Creem una carpeta en Windows en primer lloc. Per exemple, en Documents (Figura 5) i l'anomenem “compartida”.

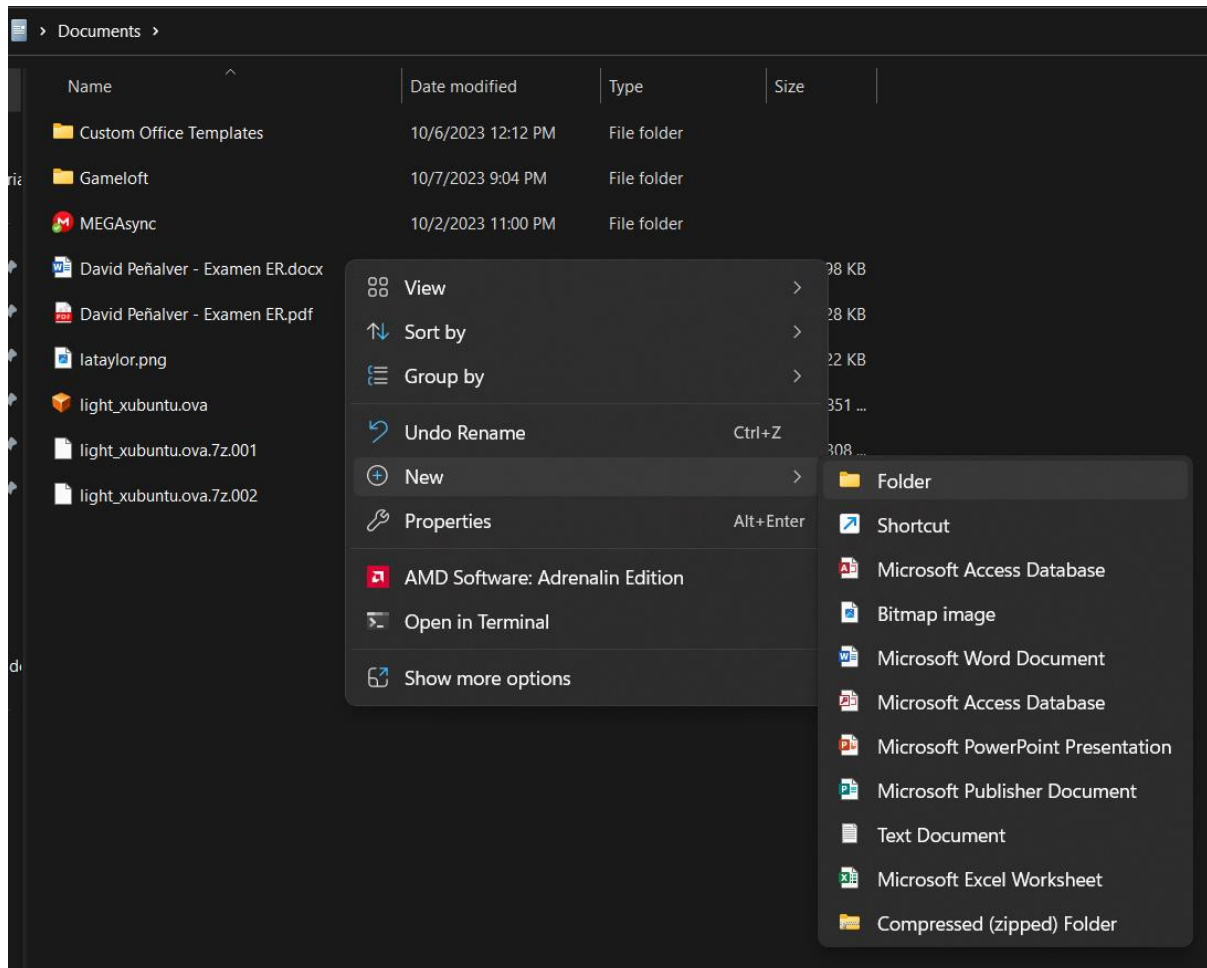


Figura 5. Creem la carpeta en Windows.

Posteriorment, en Virtual Box anem a “Devices -> Shared Folders -> Shared Folders Settings” (Figura 6).

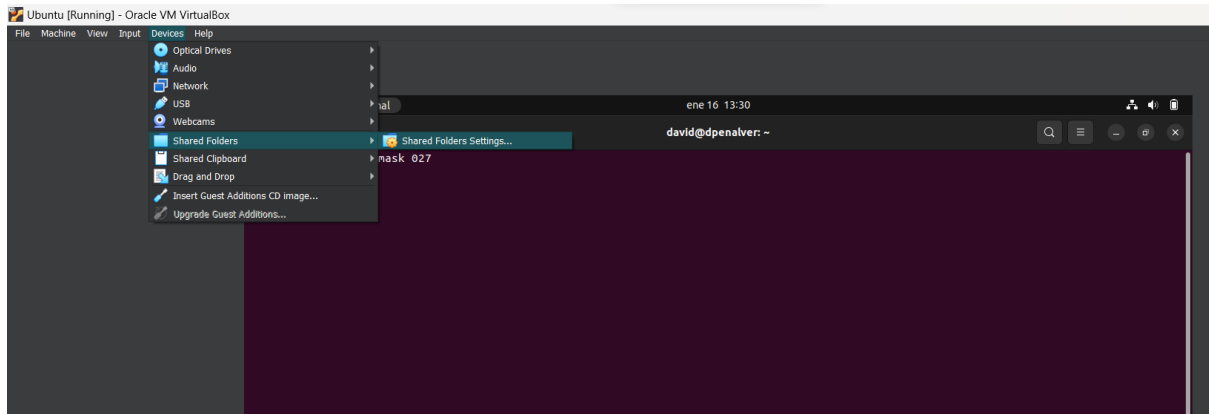


Figura 6. Primers passos en la màquina virtual amb Ubuntu.

A continuació, donem a l'icona d'afegir, en la finestra en l'opció de seleccionar la carpeta i busquem la carpeta "compartida" que tenim en "Documents".

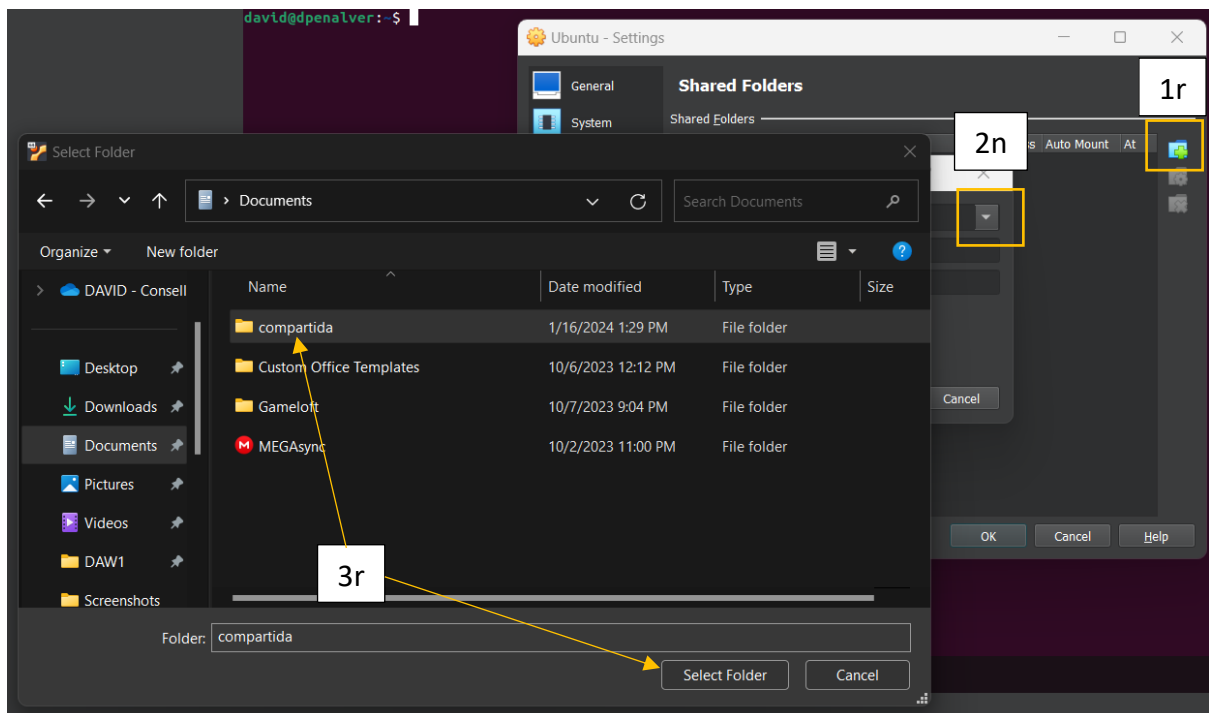


Figura 7. Seleccionant la carpeta "compartida".

Finalment, li fiquem un nom (pot ser el mateix o diferent) i marquem que s'automunte i siga permanent (Figura 8).

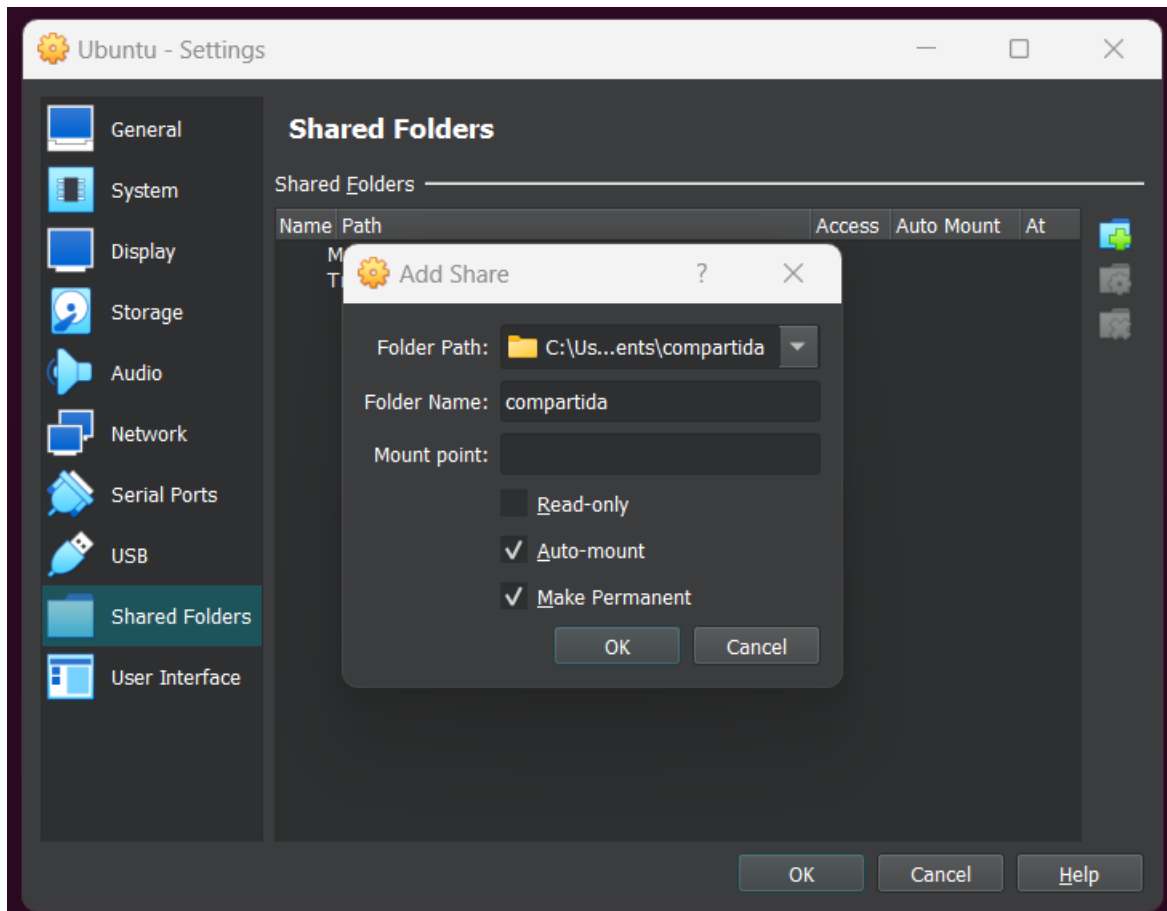


Figura 8. Aplicant la configuració de la carpeta compartida.

4- Activitat resolta 3.20

Farem 'sleep 120 &'. D'esta manera, programarem que en 120 segons hiberne la màquina i que l'acció es produïska en segon pla. A continuació, mostrem els processos amb 'jobs'. Finalment, matem el programa per a evitar que es complete el procés.

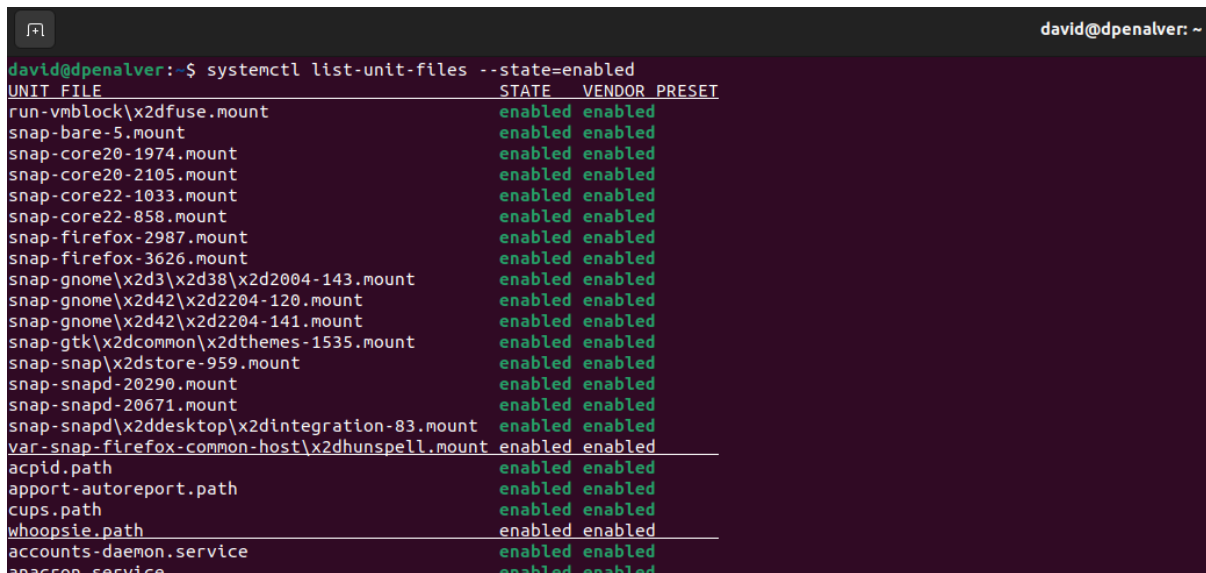


```
david@dpenalver:~$ sleep 120 &
[1] 4443
david@dpenalver:~$ jobs
[1]+  Running                  sleep 120 &
david@dpenalver:~$ kill 4443
david@dpenalver:~$ jobs
[1]+  Terminated              sleep 120
david@dpenalver:~$
```

Figura 9. Comandes per a fer el sleep en segon pla i matar el procés.

5- Activitat resolta 3.21

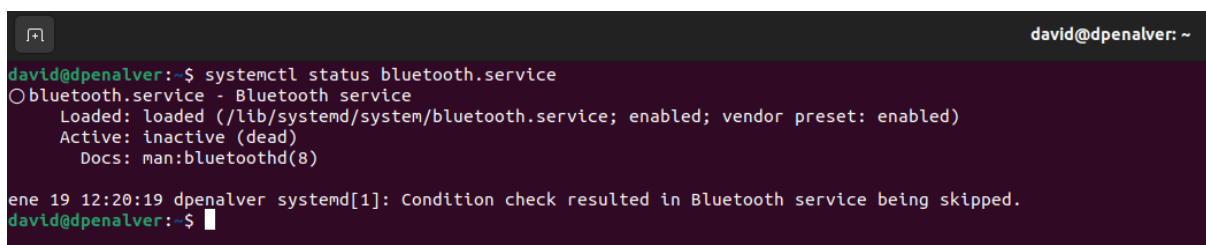
Per a mostrar tots els processos que s'estan executant, fem “systemctl list-unit-files --state=enabled” (Figura 10). Amb això, entren en el assistent dels processos. A continuació, listem tots els processos i especifiquen que siguin els que estan actius (Figura 10)



```
david@dpenalver:~$ systemctl list-unit-files --state=enabled
UNIT FILE                                STATE    VENDOR PRESET
run-vmblock\x2dfuse.mount               enabled enabled
snap-bare-5.mount                       enabled enabled
snap-core20-1974.mount                  enabled enabled
snap-core20-2105.mount                  enabled enabled
snap-core22-1033.mount                  enabled enabled
snap-core22-858.mount                   enabled enabled
snap-firefox-2987.mount                 enabled enabled
snap-firefox-3626.mount                 enabled enabled
snap-gnome\x2d3\x2d38\x2d2004-143.mount enabled enabled
snap-gnome\x2d42\x2d2204-120.mount      enabled enabled
snap-gnome\x2d42\x2d2204-141.mount      enabled enabled
snap-gtk\x2dcommon\x2dthemes-1535.mount enabled enabled
snap-snap\x2dstore-959.mount            enabled enabled
snap-snapd-20290.mount                 enabled enabled
snap-snapd-20671.mount                 enabled enabled
snap-snapd\x2ddesktop\x2dintegration-83.mount enabled enabled
var-snap-firefox-common-host\x2dhunspell.mount enabled enabled
acpid.path                             enabled enabled
apport-autoreport.path                 enabled enabled
cups.path                             enabled enabled
whoopsie.path                         enabled enabled
accounts-daemon.service               enabled enabled
anacron.service                       enabled enabled
```

Figura 10. Llistant tots els processos en execució.

En les següents captures, comprovem concretament si estan actius el Bluetooth (Figura 11), el cron (Figura 12), el cups (Figura 13), el Network Manager (Figura 14) i el snapd (Figura 15):



```
david@dpenalver:~$ systemctl status bluetooth.service
○ bluetooth.service - Bluetooth service
   Loaded: loaded (/lib/systemd/system/bluetooth.service; enabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:bluetoothd(8)

ene 19 12:20:19 dpenalver systemd[1]: Condition check resulted in Bluetooth service being skipped.
david@dpenalver:~$
```

Figura 11. Bluetooth.

```

david@dpenalver: ~
david@dpenalver:~$ systemctl status cron.service
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-01-19 12:20:17 CET; 38min ago
     Docs: man:cron(8)
    Main PID: 609 (cron)
      Tasks: 1 (limit: 9429)
     Memory: 448.0K
        CPU: 23ms
    CGroup: /system.slice/cron.service
            └─609 /usr/sbin/cron -f -P

ene 19 12:20:17 dpenalver systemd[1]: Started Regular background program processing daemon.
ene 19 12:20:17 dpenalver cron[609]: (CRON) INFO (pidfile fd = 3)
ene 19 12:20:17 dpenalver cron[609]: (CRON) INFO (Running @reboot jobs)
ene 19 12:30:01 dpenalver CRON[3907]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
ene 19 12:30:01 dpenalver CRON[3908]: (root) CMD ([ -x /etc/init.d/anacron ] && if [ ! -d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron start >/dev/null; fi)
ene 19 12:30:01 dpenalver CRON[3907]: pam_unix(cron:session): session closed for user root
david@dpenalver:~$

```

Figura 12. Cron.

```

david@dpenalver: ~
david@dpenalver:~$ systemctl status cups.service
● cups.service - CUPS Scheduler
   Loaded: loaded (/lib/systemd/system/cups.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-01-19 12:20:17 CET; 39min ago
 TriggeredBy: ● cups.socket
               ● cups.path
     Docs: man:cupsd(8)
    Main PID: 711 (cupsd)
    Status: "Scheduler is running..."
      Tasks: 1 (limit: 9429)
     Memory: 8.5M
        CPU: 55ms
    CGroup: /system.slice/cups.service
            └─711 /usr/sbin/cupsd -l

ene 19 12:20:17 dpenalver systemd[1]: Starting CUPS Scheduler...
ene 19 12:20:17 dpenalver systemd[1]: Started CUPS Scheduler.
david@dpenalver:~$

```

Figura 13. Cups.

```

david@dpenalver: ~
david@dpenalver:~$ systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-01-19 12:20:17 CET; 40min ago
     Docs: man:NetworkManager(8)
    Main PID: 612 (NetworkManager)
      Tasks: 3 (limit: 9429)
     Memory: 11.6M
        CPU: 634ms
    CGroup: /system.slice/NetworkManager.service
            └─612 /usr/sbin/NetworkManager --no-daemon

ene 19 12:20:17 dpenalver NetworkManager[612]: <info> [1705663217.9541] manager: NetworkManager state is now CONNECTED_LOCAL
ene 19 12:20:17 dpenalver NetworkManager[612]: <info> [1705663217.9542] manager: NetworkManager state is now CONNECTED_SITE
ene 19 12:20:17 dpenalver NetworkManager[612]: <info> [1705663217.9543] policy: set 'Wired connection 1' (enp0s3) as default for IPv4 routing and DNS
ene 19 12:20:17 dpenalver NetworkManager[612]: <info> [1705663217.9545] device (enp0s3): Activation: successful, device activated.
ene 19 12:20:17 dpenalver NetworkManager[612]: <info> [1705663217.9548] manager: startup complete
ene 19 12:20:18 dpenalver NetworkManager[612]: <info> [1705663218.0176] modem-manager: ModemManager not available
ene 19 12:20:18 dpenalver NetworkManager[612]: <info> [1705663218.0668] modem-manager: ModemManager now available
ene 19 12:20:18 dpenalver NetworkManager[612]: <info> [1705663218.1753] manager: NetworkManager state is now CONNECTED_GLOBAL
ene 19 12:20:23 dpenalver NetworkManager[612]: <info> [1705663223.7408] agent-manager: agent[26011c6543cb68f1,;1.43/org.gnome.Shell.NetworkAgent/1000]: agent registered
ene 19 12:40:02 dpenalver NetworkManager[612]: <info> [1705664402.9481] agent-manager: agent[1a50999ff19aaff8,;1.43/org.gnome.Shell.NetworkAgent/1000]: agent registered
david@dpenalver:~$

```

Figura 14. Network Manager.



```
david@dpنالver: ~  
david@dpنالver: $ systemctl status snapd.service  
● snapd.service - Snap Daemon  
   Loaded: loaded (/lib/systemd/system/snapd.service; enabled; vendor preset: enabled)  
   Active: active (running) since Fri 2024-01-19 12:20:18 CET; 41min ago  
 TriggeredBy: ● snapd.socket  
    Main PID: 630 (snapd)  
      Tasks: 12 (limit: 9429)  
     Memory: 52.3M  
        CPU: 29.251s  
    CGroup: /system.slice/snapd.service  
            └─630 /usr/lib/snapd/snapd  
  
ene 19 12:20:17 dpنالver systemd[1]: Starting Snap Daemon...  
ene 19 12:20:18 dpنالver snapd[630]: overlord.go:271: Acquiring state lock file  
ene 19 12:20:18 dpنالver snapd[630]: overlord.go:270: Acquired state lock file  
ene 19 12:20:18 dpنالver snapd[630]: daemon.go:247: started snapd/2.61.1 (series 16; classic) ubuntu/22.04 (amd64) linux/6.2.0-39-generic.  
ene 19 12:20:18 dpنالver snapd[630]: daemon.go:340: adjusting startup timeout by 1m20s (pessimistic estimate of 30s plus 5s per snap)  
ene 19 12:20:18 dpنالver snapd[630]: backends.go:58: AppArmor status: apparmor is enabled and all features are available (using snapd provided apparmor_parser)  
ene 19 12:20:18 dpنالver systemd[1]: Started Snap Daemon.  
david@dpنالver: $
```

Figura 15. Snapd.

Com no estic utilitzant ratolí, teclat ni res extern, tinc el Bluetooth desactivat, tal i com apareix en la Figura 11. La resta, com apareix amb el verd i el text, estan actius. És notable mencionar que el Network Manager hi ha que escriure-lo en majúscules.

6- Activitat resolta 3.22

Amb `'tar -czvf misdocumentos.tgz Documents/'` creem (-c) un arxiu comprimit (-z) amb nom (-f) `misdocumentos.tgz`. En eixe fitxer estaran els documents inclosos en Documents. Posteriorment, amb -t en compte de -c llistem els documents que estan en el arxiu comprimit. El -v servix per a fer-ho verbòs.

```
david@dpenalver:~$ tar -czvf misdocumentos.tgz Documents/  
Documents/  
Documents/lataylor.png  
david@dpenalver:~$ tar -tzvf misdocumentos.tgz Documents/  
drwxr-xr-x david/david    0 2024-01-19 13:26 Documents/  
-rwxrwx--- david/david 534221 2023-10-10 11:36 Documents/lataylor.png  
david@dpenalver:~$
```

Figura 16. Creant la còpia del seguretat.

Així doncs, anant a la nostra carpeta personal podem trobar el nostre arxiu comprimit que conté els fitxers que es mencionen en el procés de la Figura 16:

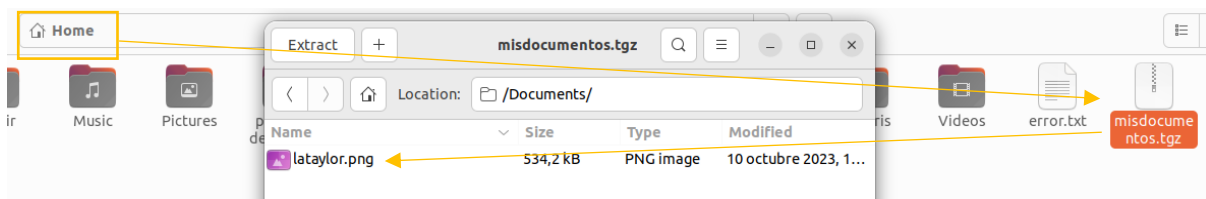
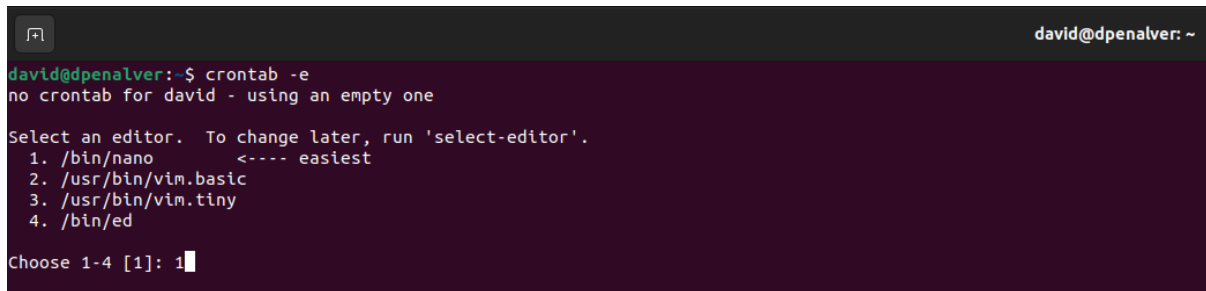


Figura 17. Comprovació de que el procés s'ha efectuat correctament.

7- Activitat resolta 3.23

Obrim terminal i executem “crontab -e” per a crear una nova:

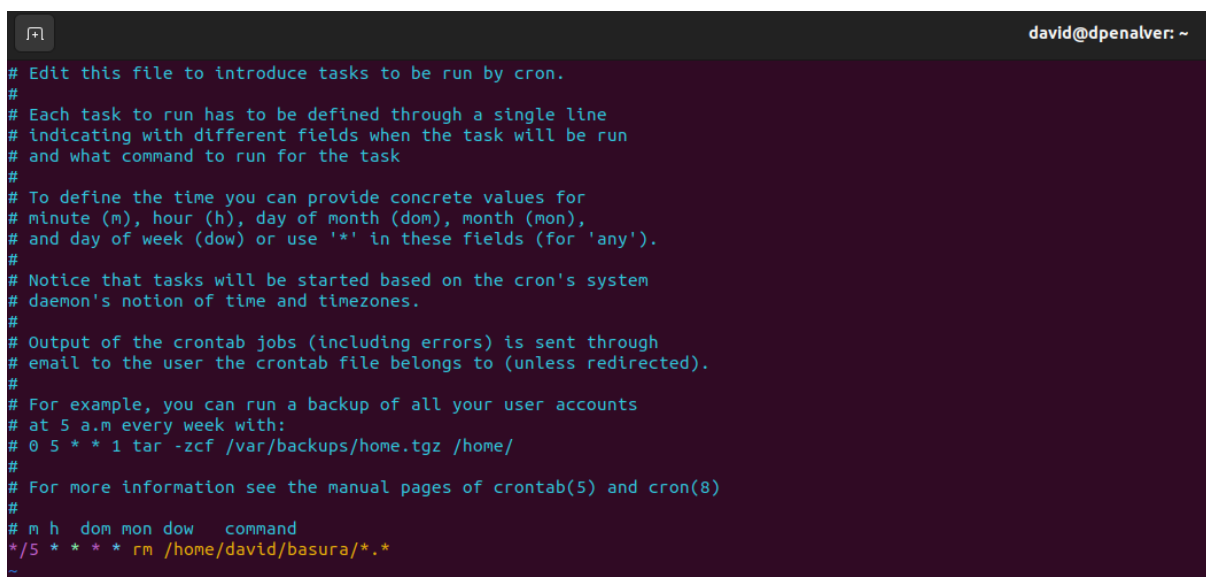


```
david@dpenalver: ~
david@dpenalver:~$ crontab -e
no crontab for david - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed
Choose 1-4 [1]: 1
```

Figura 18. Obrint l’editor per a establir una tarea programada.

A continuació, amb el Nano o el Vim, editem el final del fitxer per a afegir el moment en el qual s’executa i l’acció que es desenvoluparà:



```
david@dpenalver: ~
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/5 * * * * rm /home/david/basura/*.*
~
```

Figura 19. Establint la tarea programada.

Amb ‘contrab -l’ ens eixirà la tarea programada que acabem de crear. Tot allò que està precedit per “#” són comentaris que no s’executaran.


```
crontab: installing new crontab
david@dpenalver:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/5 * * * * rm /home/david/basura/*.*
david@dpenalver:~$
```

Figura 20. Comprovant que s'ha programat la tarea.

8- Activitat resolta 3.24

Obrim una terminal en la qual escrivim “journalctl -f” i ens mostrarà els processos de log en temps real. A continuació, obrim una nova terminal, on escriurem “su david”, però la contrasenya l’esciurem malament. Així doncs, en la primera terminal ens eixirà un missatge d’intent de sessió i d’error.

The image shows two terminal windows side-by-side. The left window displays the output of 'journalctl -f', showing real-time system logs. The right window shows the execution of 'su david', which fails due to an incorrect password.

```
david@dpenalver: ~
david@dpenalver:~$ journalctl -f
ene 23 13:35:34 dpenalver systemd[1]: apt-daily-upgrade.service: Consumed 3min 21.271s CPU time.
ene 23 13:36:30 dpenalver systemd[1183]: vte-spawn-d85e0508-8cb9-41cc-be4b-b3a9f190c7f5.scope: Consumed 2.041s CPU time.
ene 23 13:36:30 dpenalver gnome-shell[1370]: JS ERROR: TypeError: this.actor is null
    _syncEnabled@resource:///org/gnome/shell/ui/windowManager.js:138:25
    onStopped@resource:///org/gnome/shell/ui/windowManager.js:150:35
    _makeEaseCallback/@resource:///org/gnome/shell/ui/environment.js:151:22
    _easeActorProperty/@resource:///org/gnome/shell/ui/environment.js:317:60
    _destroyWindowDone@resource:///org/gnome/shell/ui/windowManager.js:1596:21
    onStopped@resource:///org/gnome/shell/ui/windowManager.js:1564:39
    _makeEaseCallback/@resource:///org/gnome/shell/ui/environment.js:151:22
    _easeActor/@resource:///org/gnome/shell/ui/environment.js:240:64
ene 23 13:36:30 dpenalver systemd[1183]: gnome-terminal-server.service: Consumed 0.847s CPU time.
ene 23 13:36:31 dpenalver systemd[1183]: Started Application launched by gsd-media-keys.
ene 23 13:36:31 dpenalver dbus-daemon[1227]: [session uid=1000 pid=1227] Activating via systemd: service name='org.gnome.termin
omn="/usr/bin/gnome-terminal.real --wait " label="unconfined")
ene 23 13:36:31 dpenalver systemd[1183]: Starting GNOME Terminal Server...
ene 23 13:36:31 dpenalver dbus-daemon[1227]: [session uid=1000 pid=1227] Successfully activated service 'org.gnome.Termin
ene 23 13:36:31 dpenalver systemd[1183]: Started GNOME Terminal Server.
ene 23 13:36:31 dpenalver systemd[1183]: Started VTE child process 21854 launched by gnome-terminal-server process 21836.
ene 23 13:36:42 dpenalver systemd[1183]: Started Application launched by gnome-shell.
ene 23 13:36:42 dpenalver systemd[1183]: Started VTE child process 21869 launched by gnome-terminal-server process 21836.
ene 23 13:36:56 dpenalver su[21876]: pam_unix(su:auth): authentication failure; logname= uid=1000 euid=0 tty=/dev/pts/1 r
ene 23 13:36:58 dpenalver su[21876]: FAILED SU (to david) david on pts/1
```

Figura 21. Comprovant el funcionament de “journalctl”.

9- Activitat resolta 3.25

En primer lloc, he creat un directori en el qual ficar els diferents scripts a realitzar. En segon lloc, cree l'script i entre en ell:



```
david@dpenalver: ~/scripts
david@dpenalver:~$ mkdir scripts
david@dpenalver:~$ cd scripts/
david@dpenalver:~/scripts$ touch clearanddate.sh
david@dpenalver:~/scripts$ vim clearanddate.sh
```

Figura 22. Creant i entrant en el nou script.

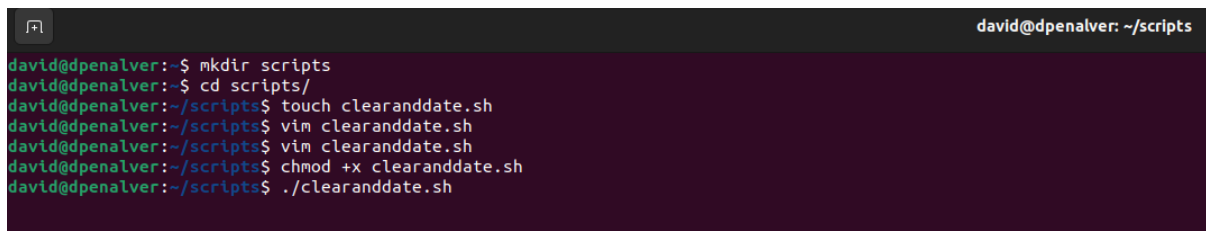
En tercer lloc, escric que és un script del shell Bash, el comentari de què fa i allò que vullc que faça:



```
david@dpenalver: ~/scripts
#!/bin/bash
clear
echo "Hoy es: $(date)"
#Este script borra la pantalla i dona la data
~
```

Figura 23. Creant l'script.

Done permís i escric l'execució:



```
david@dpenalver: ~/scripts
david@dpenalver:~$ mkdir scripts
david@dpenalver:~$ cd scripts/
david@dpenalver:~/scripts$ touch clearanddate.sh
david@dpenalver:~/scripts$ vim clearanddate.sh
david@dpenalver:~/scripts$ vim clearanddate.sh
david@dpenalver:~/scripts$ chmod +x clearanddate.sh
david@dpenalver:~/scripts$ ./clearanddate.sh
```

Figura 24. Executant l'script

Executat:

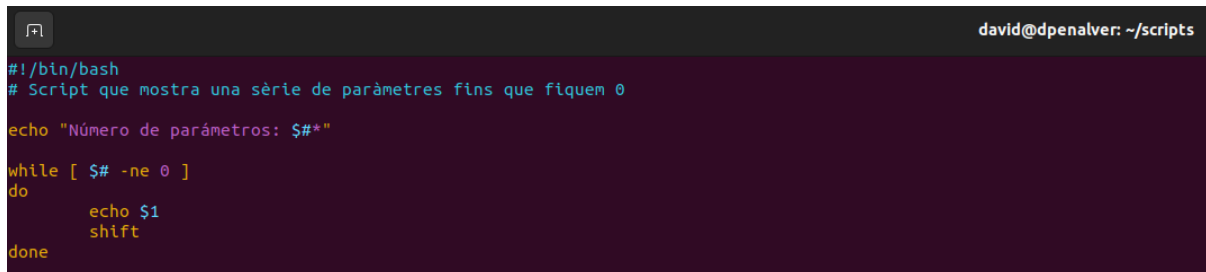


```
david@dpenalver: ~/scripts
Hoy es: vie 26 ene 2024 12:17:41 CET
david@dpenalver:~/scripts$
```

Figura 25. Script executat.

10- Activitat resolta 3.26

Creem el fitxer “total.sh” seguint els passos de la Figura 22. A continuació, ficarem els paràmetres que ens demana (Figura 26).



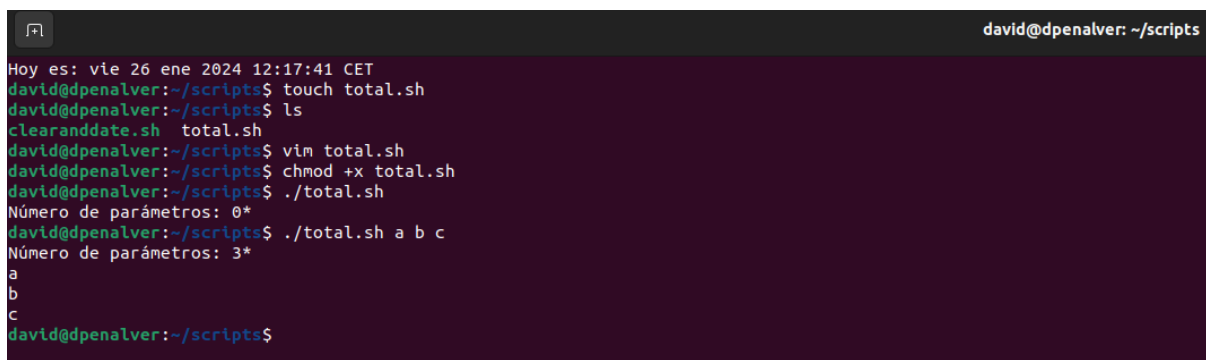
```
#!/bin/bash
# Script que mostra una sèrie de paràmetres fins que fiquem 0

echo "Número de paràmetres: $#*"

while [ $# -ne 0 ]
do
    echo $1
    shift
done
```

Figura 26. El·laborem l'script.

Donem permís d'execució i executem:

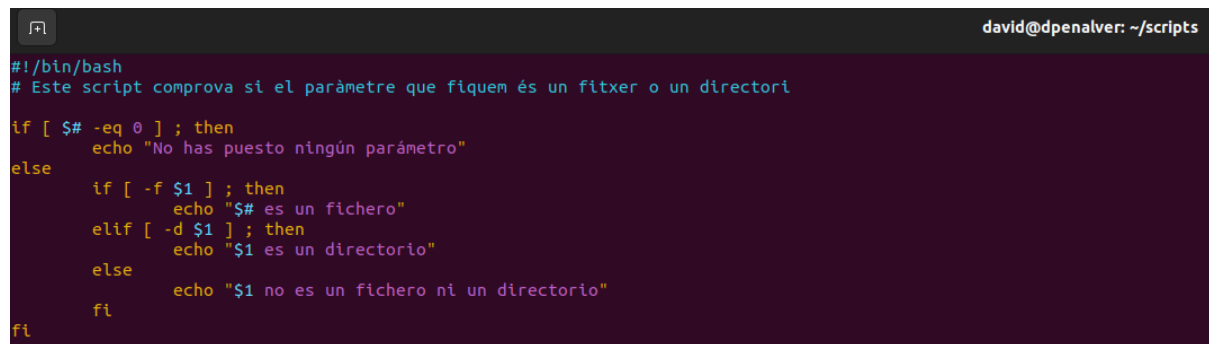


```
Hoy es: vie 26 ene 2024 12:17:41 CET
david@dpنالver:~/scripts$ touch total.sh
david@dpنالver:~/scripts$ ls
clearanddate.sh  total.sh
david@dpنالver:~/scripts$ vim total.sh
david@dpنالver:~/scripts$ chmod +x total.sh
david@dpنالver:~/scripts$ ./total.sh
Número de parámetros: 0*
david@dpنالver:~/scripts$ ./total.sh a b c
Número de parámetros: 3*
a
b
c
david@dpنالver:~/scripts$
```

Figura 27. Executant total.sh.

11- Activitat resolta 3.27

De nou, creem l'script seguint els passos de la Figura 25. Tindrà un paràmetre rebut i comprovarà si compleix alguna de les condicions. El codi, per tant, serà el següent:



```
#!/bin/bash
# Este script comprueba si el parámetro que fiquem és un fitxer o un directori

if [ $# -eq 0 ] ; then
    echo "No has puesto ningún parámetro"
else
    if [ -f $1 ] ; then
        echo "$# es un fichero"
    elif [ -d $1 ] ; then
        echo "$1 es un directorio"
    else
        echo "$1 no es un fichero ni un directorio"
    fi
fi
```

Figura 28. Codi de tipo.sh.

Quan donem permís d'exució i executem, trobarem el següent:



```
david@dpنالver:~/scripts$ chmod +x tipo.sh
david@dpنالver:~/scripts$ ./tipo.sh
No has puesto ningún parámetro
david@dpنالver:~/scripts$ ./tipo.sh .
. es un directorio
david@dpنالver:~/scripts$ ./tipo.sh tipo.sh
1 es un fichero
david@dpنالver:~/scripts$ ./tipo.sh hola
hola no es un fichero ni un directorio
david@dpنالver:~/scripts$
```

Figura 29. Provant tipo.sh.

12- Activitat resolta 3.28

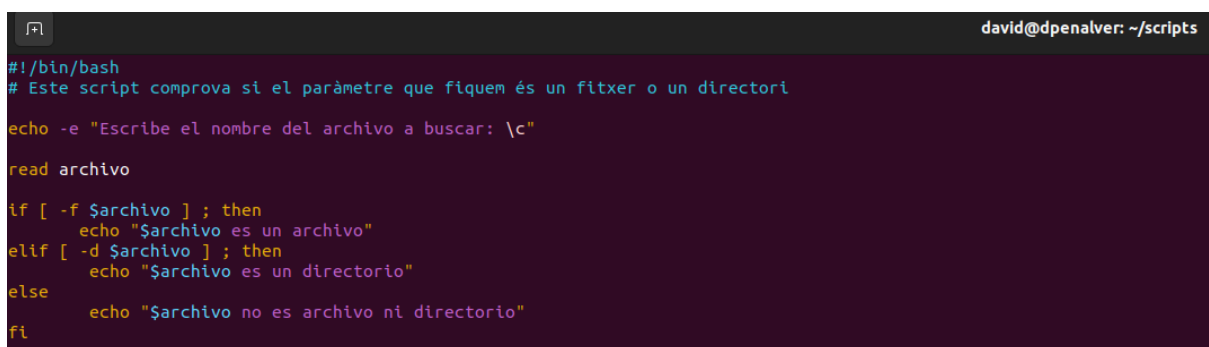
En este cas, començarem copiant el fitxer canviant-li el nom:



```
david@dpenalver: ~/scripts
david@dpenalver:~/scripts$ cp tipo.sh pregunta.sh
david@dpenalver:~/scripts$ ls
clearanddate.sh  pregunta.sh  tipo.sh  total.sh
david@dpenalver:~/scripts$
```

Figura 30. Creant l'script.

A continuació, modifiquem el codi per a complir amb les indicacions de l'activitat:



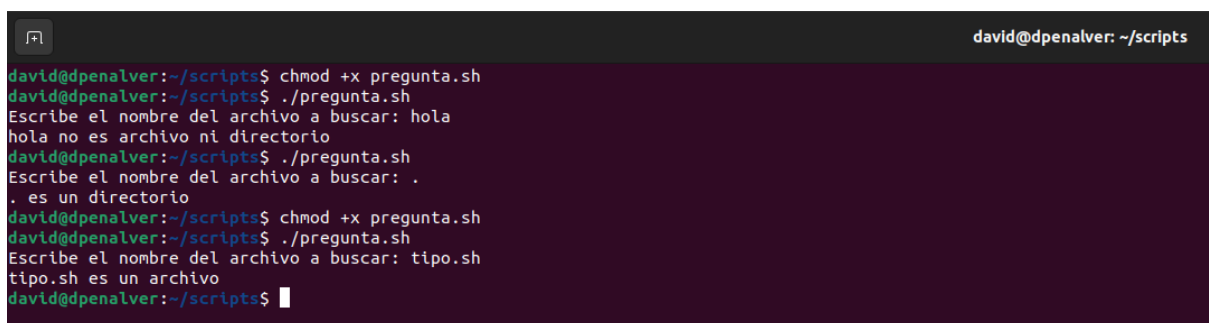
```
david@dpenalver: ~/scripts
#!/bin/bash
# Este script comprueba si el parámetro que fiquem és un fitxer o un directori

echo -e "Escribe el nombre del archivo a buscar: \c"
read archivo

if [ -f $archivo ] ; then
    echo "$archivo es un archivo"
elif [ -d $archivo ] ; then
    echo "$archivo es un directorio"
else
    echo "$archivo no es archivo ni directorio"
fi
```

Figura 31. Codi de pregunta.sh.

Donem permís d'execució i executem:



```
david@dpenalver: ~/scripts
david@dpenalver:~/scripts$ chmod +x pregunta.sh
david@dpenalver:~/scripts$ ./pregunta.sh
Escribe el nombre del archivo a buscar: hola
hola no es archivo ni directorio
david@dpenalver:~/scripts$ ./pregunta.sh
Escribe el nombre del archivo a buscar: .
. es un directorio
david@dpenalver:~/scripts$ chmod +x pregunta.sh
david@dpenalver:~/scripts$ ./pregunta.sh
Escribe el nombre del archivo a buscar: tipo.sh
tipo.sh es un archivo
david@dpenalver:~/scripts$
```

Figura 32. Executant pregunta.sh.

13- Activitat resolta 3.29

Instal·lem el “cowsay”, creem l'script i entrem en ell:

```
david@dpenalver: ~/scripts
david@dpenalver:~/scripts$ sudo apt install cowsay
[sudo] password for david:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  filters cowsay-off
The following NEW packages will be installed:
  cowsay
0 upgraded, 1 newly installed, 0 to remove and 38 not upgraded.
Need to get 18,6 kB of archives.
After this operation, 93,2 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 cowsay all 3.03+dfsg2-8 [18,6 kB]
Fetched 18,6 kB in 0s (111 kB/s)
Selecting previously unselected package cowsay.
(Reading database ... 207761 files and directories currently installed.)
Preparing to unpack .../cowsay_3.03+dfsg2-8_all.deb ...
Unpacking cowsay (3.03+dfsg2-8) ...
Setting up cowsay (3.03+dfsg2-8) ...
Processing triggers for man-db (2.10.2-1) ...
david@dpenalver:~/scripts$ touch saludo.sh
david@dpenalver:~/scripts$ vim saludo.sh
```

Figura 33. Instal·lació de cowsay i creació de saludo.sh.

A continuació, escribim el codi pertinent:

```
david@dpenalver: ~/scripts
#!/bin/bash
clear
echo "----- \c ¡Hola, $LOGNAME! Son las $(date +%R). \c -----" | cowsay
~
```

Figura 34. Codi de saludo.sh.

Finalment, donem permís d'execució i executem:

```
david@dpenalver: ~/scripts
david@dpenalver:~/scripts$ chmod +x saludo.sh
david@dpenalver:~/scripts$ ./saludo.sh

/ ----- \c ¡Hola, david! Son \
\ las 13:10. \c ----- /
-----
      ^ ^
      (oo)\_____ )\
      (__)\       )\/\
           ||----w |
           ||     ||
david@dpenalver:~/scripts$
```

Figura 35. Executant saludo.sh.

Pràctica 4

1- Exercici 1

- a) El comandament que ficaria per a apagar l'ordinador hui a les 16:00 és el següent:
"at 16:00 shutdown". Es possible que tingan que instal·lar-ho prèviament (sudo apt install at).
- b) "crontab -e" per a crear-ho/editar. Una vegada dins, fiquem al final del fitxer el següent:
"@reboot usuari /home/usuari/arranque.sh".
- c) En este cas, el codique ficaríem al final del fitxer seria este:
"30 1 * * 1-5 usuari /home/usuari/copia-pega.sh".

2- Exercici 2

'ps': mostra tots els processos.

'ps aux': mostra informació detallada de tots el processos actius.

'ps -e': mostra una llista de tots els processos del sistema.

'ps -ef': mostra tots els processos amb informació detallada.

'ps -eH': mostra tots els processos com una jerarquia.

'ps -u usuari': mostra tots els processos d'un usuari específic.

'ps -o': permet personalitzar l'eixida especificant les columnas que es volen vore.

'top': oferix una visualització en temps real dels processos actius i dels recursos del sistema.

'htop': similar a l'anterior, pero gràficament més amigable i amb funcions addicionals.

3- Exercici 3

'sudo apt update && sudo apt upgrade': per a fer-ho tot alhora.

Es pot separar també en 'sudo apt update' i a continuació en 'sudo apt upgrade'.

Finalment, per a configurar les actualitzacions de manera automàtica, podem fer:
'sudo dpkg-reconfigure -plow unattended-upgrades'.

4- Exercici 4

Anem a realitzar el procés amb Déjà Dup. Obrim l'aplicació i li donem al botó verd.

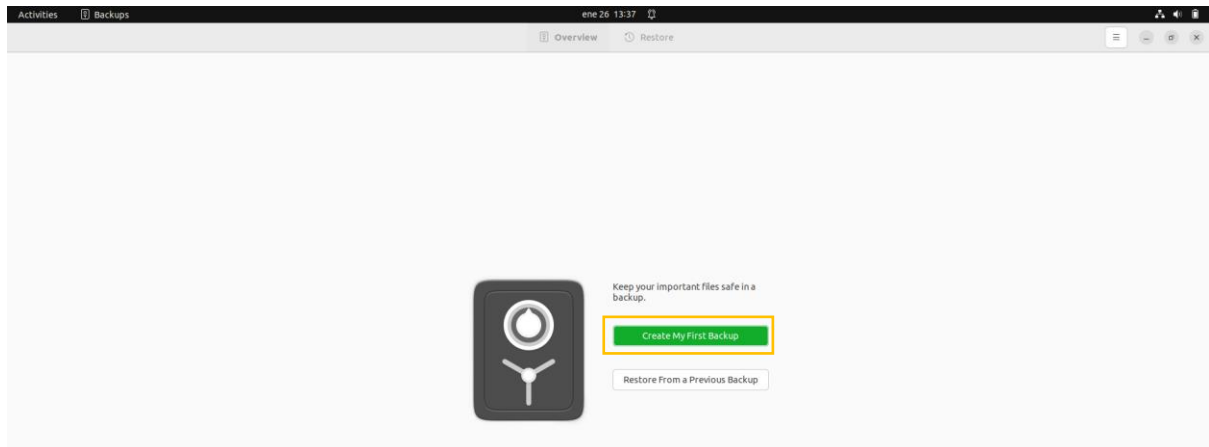


Figura 36. Cliquem en el botó verd.

A continuació, seleccionem quines opcions volem i quines volem ignorar:

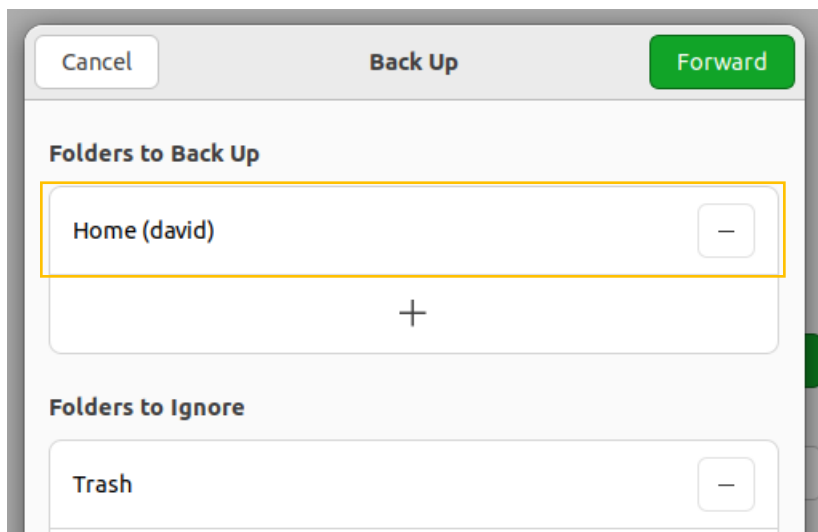


Figura 37. Seleccionant què volem fer backup.

En este cas, deixarem fora la paperera i continuem amb el botó verd de nou. Guardem en local per a no ficar el compte de Google i continuem:

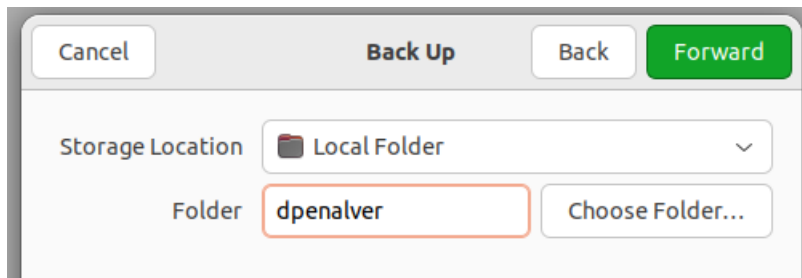


Figura 38. Selecciónant a on volem la còpia de seguretat.

Selecciónem si volem contrasenya, no en este cas d'exemple:

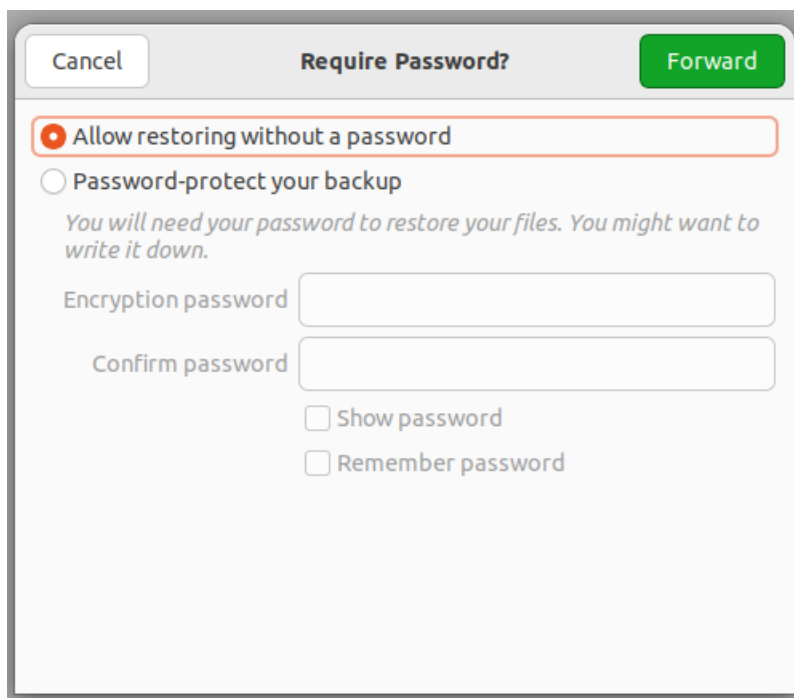


Figura 39. Contrasenya de la còpia de seguretat.

Es crearà la còpia de seguretat:

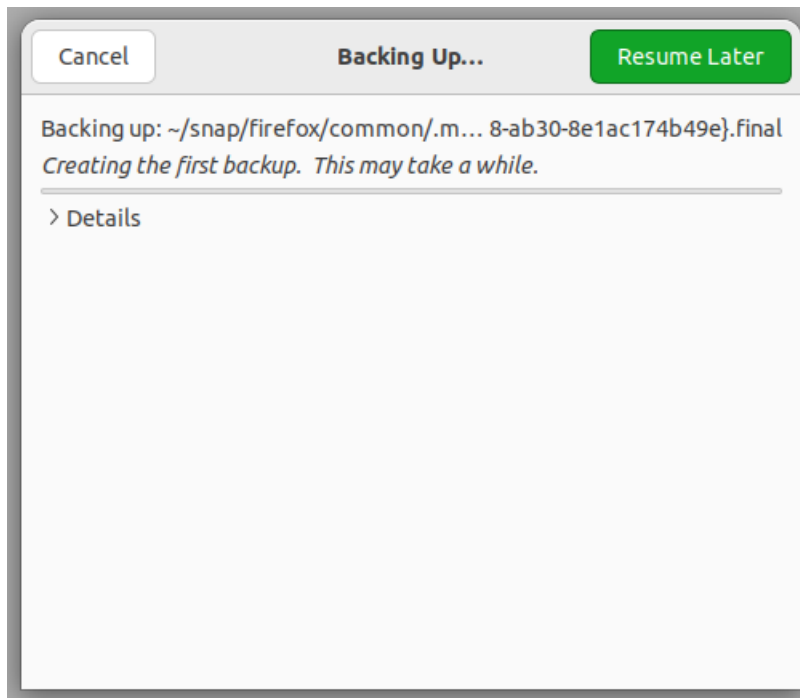


Figura 40. Finalitzant la còpia de seguretat.

Quan acabe, es tancarà i tornarà al menú d'inici de l'aplicació.

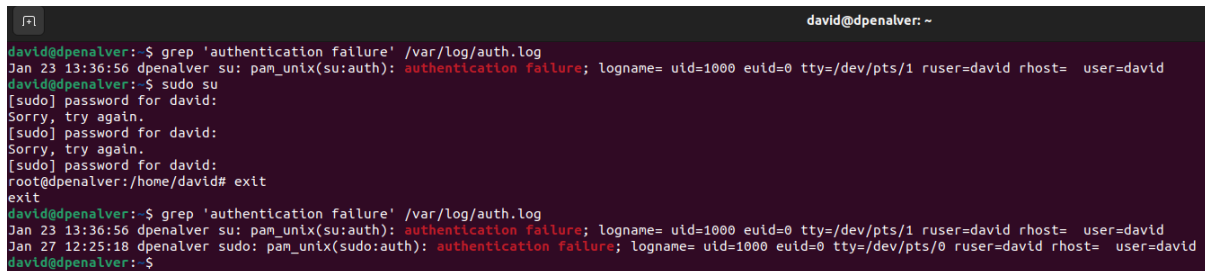
5- Exercici 5

Revisar i Instal·lar: utilitzar l'aplicació "Software" o "Gestor de Paquets Synaptic" per revisar i instal·lar actualitzacions gràficament.

Configurar actualitzacions automàtiques: es poden configurar actualitzacions automàtiques a través de "Configuració del Sistema" -> "Software i Actualitzacions" -> "Pestanya Actualitzacions".

6- Exercici 6

Si un usuari no es valida correctament, a l'hora de fer una tarea d'administració, es poden veure els intents fallits en “/var/log/auth.log”. Si fem la prova, podrem veure-lo de la següent forma per exemple:



```

david@dpenalver: ~
david@dpenalver:~$ grep 'authentication failure' /var/log/auth.log
Jan 23 13:36:56 dpenalver su: pam_unix(su:auth): authentication failure; logname= uid=1000 euid=0 tty=/dev/pts/1 ruser=david rhost= user=david
david@dpenalver:~$ sudo su
[sudo] password for david:
Sorry, try again.
[sudo] password for david:
Sorry, try again.
[sudo] password for david:
root@dpenalver:/home/david# exit
exit
david@dpenalver:~$ grep 'authentication failure' /var/log/auth.log
Jan 23 13:36:56 dpenalver su: pam_unix(su:auth): authentication failure; logname= uid=1000 euid=0 tty=/dev/pts/1 ruser=david rhost= user=david
Jan 27 12:25:18 dpenalver sudo: pam_unix(sudo:auth): authentication failure; logname= uid=1000 euid=0 tty=/dev/pts/0 ruser=david rhost= user=david
david@dpenalver:~$
  
```

Figura 36. Exemple de mostra de l'error d'autenticació.

En la Figura 36 podem veure com al inicialitzar el comandament trobem un intent fallit de fa quatre dies. A continuació, entre en “sudo su” ficant malament la contrasenya a propòsit per a generar un nou error d'autenticació. Finalment, entre i isc correctament i torne a executar el primer comandament, apareixent el nou error d'autenticació.

7- Exercici 7

Creem l'arxiu i entrem per a editar-ho:



```
david@dpenalver: ~/scripts
david@dpenalver:~$ cd scripts/
david@dpenalver:~/scripts$ touch ejercicio7.sh
david@dpenalver:~/scripts$ vim ejercicio7.sh
david@dpenalver:~/scripts$
```

Figura 41. Creant l'script i entrant.

Fem el codi necessari:



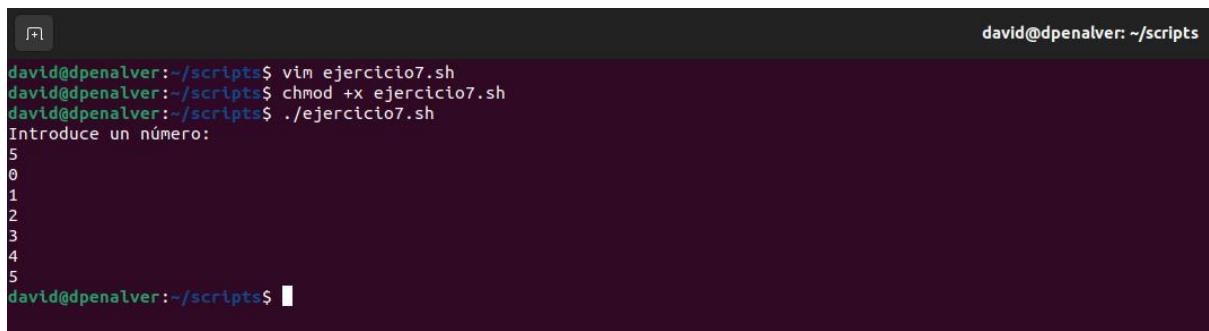
```
david@dpenalver: ~/scripts
#!/bin/bash
# Ejercicio 7

echo "Introduce un número:"
read numero

for (( i=0; i<=numero; i++ )); do
    echo $i
done
```

Figura 42. Codi de l'script.

Donem permís d'exemució i executem:



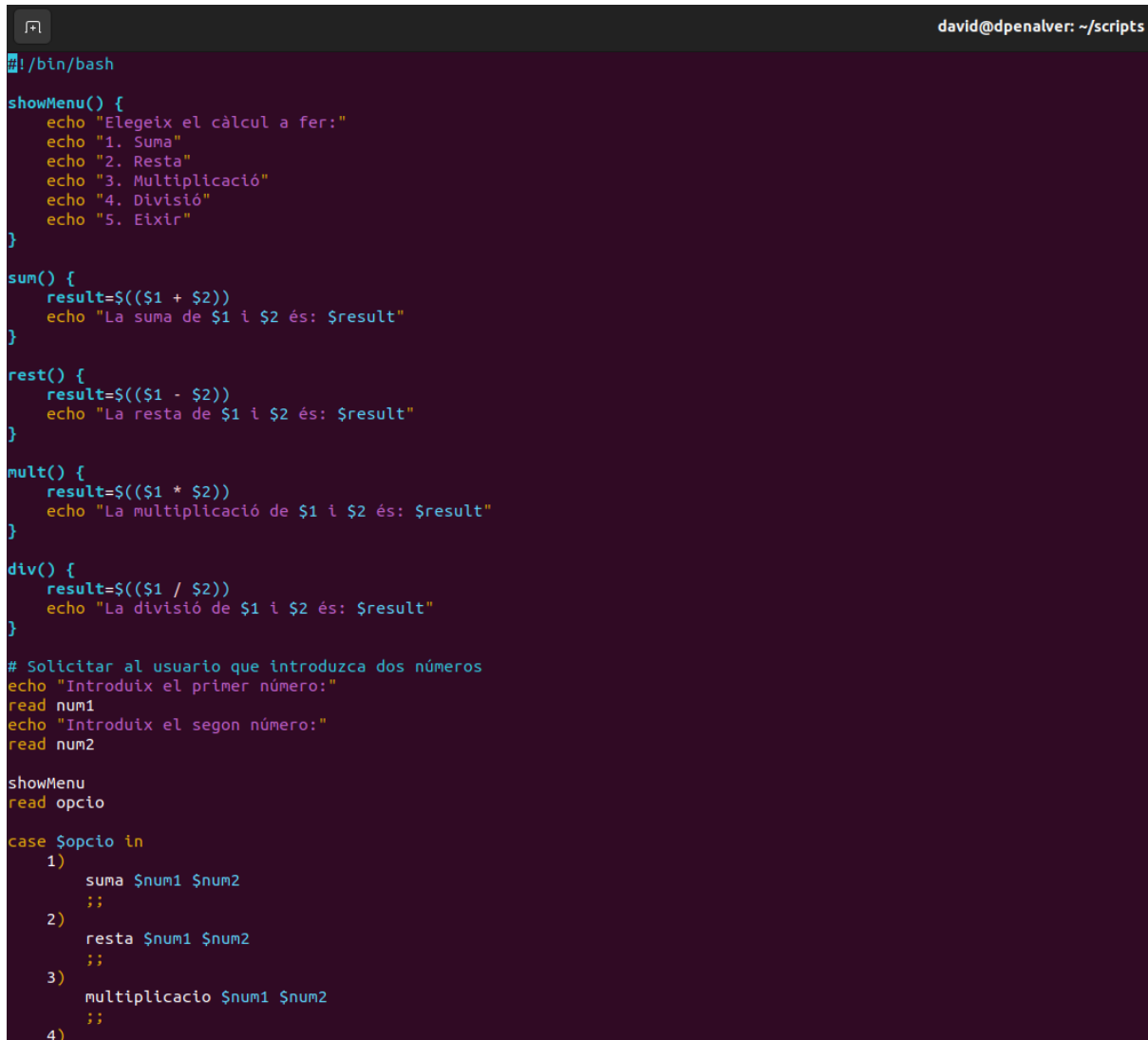
```
david@dpenalver:~/scripts$ vim ejercicio7.sh
david@dpenalver:~/scripts$ chmod +x ejercicio7.sh
david@dpenalver:~/scripts$ ./ejercicio7.sh
Introduce un número:
5
0
1
2
3
4
5
david@dpenalver:~/scripts$
```

Figura 43. Executant el codi.

8- Exercici 8

Creem l'arxiu i entrem per a editar-ho com en la Figura 41.

Fem el codi necessari:



```
#!/bin/bash

showMenu() {
    echo "Elegeix el càlcul a fer:"
    echo "1. Suma"
    echo "2. Resta"
    echo "3. Multiplicació"
    echo "4. Divisió"
    echo "5. Eixir"
}

sum() {
    result=$(( $1 + $2 ))
    echo "La suma de $1 i $2 és: $result"
}

rest() {
    result=$(( $1 - $2 ))
    echo "La resta de $1 i $2 és: $result"
}

mult() {
    result=$(( $1 * $2 ))
    echo "La multiplicació de $1 i $2 és: $result"
}

div() {
    result=$(( $1 / $2 ))
    echo "La divisió de $1 i $2 és: $result"
}

# Solicitar al usuario que introduzca dos números
echo "Introduix el primer número:"
read num1
echo "Introduix el segon número:"
read num2

showMenu
read opcio

case $opcio in
    1)
        suma $num1 $num2
        ;;
    2)
        resta $num1 $num2
        ;;
    3)
        multiplicacio $num1 $num2
        ;;
    4)
```

Figura 44. Codi de l'script.

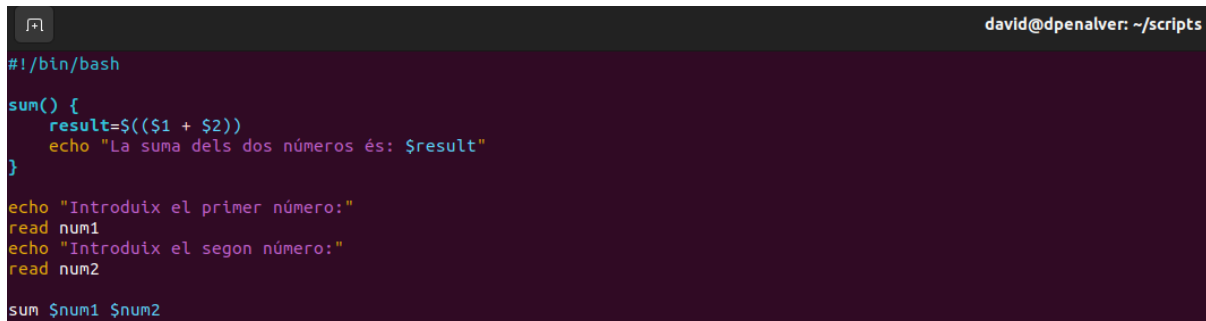
Donem permís d'execució i executem:

```
┌───┐ david@dpenalver: ~/scripts
Introduix el primer número:
10
Introduix el segon número:
2
Elegeix el càlcul a fer:
1. Suma
2. Resta
3. Multiplicació
4. Divisió
5. Eixir
2
La resta de 10 i 2 és: 8
david@dpenalver:~/scripts$ ./ejercicio8.sh
Introduix el primer número:
10
Introduix el segon número:
5
Elegeix el càlcul a fer:
1. Suma
2. Resta
3. Multiplicació
4. Divisió
5. Eixir
3
La multiplicació de 10 i 5 és: 50
david@dpenalver:~/scripts$ ./ejercicio8.sh
Introduix el primer número:
10
Introduix el segon número:
5
Elegeix el càlcul a fer:
1. Suma
2. Resta
3. Multiplicació
4. Divisió
5. Eixir
4
La divisió de 10 i 5 és: 2
david@dpenalver:~/scripts$ ./ejercicio8.sh
Introduix el primer número:
10
Introduix el segon número:
5
Elegeix el càlcul a fer:
1. Suma
2. Resta
3. Multiplicació
4. Divisió
5. Eixir
5
Adéu!
david@dpenalver:~/scripts$
```

Figura 45. Executant el codi.

9- Exercici 9

Creem l'arxiu i entrem per a editar-ho com en la Figura 41. A continuació, escribim el codi necessari:



```
david@dpنالver: ~/scripts
#!/bin/bash

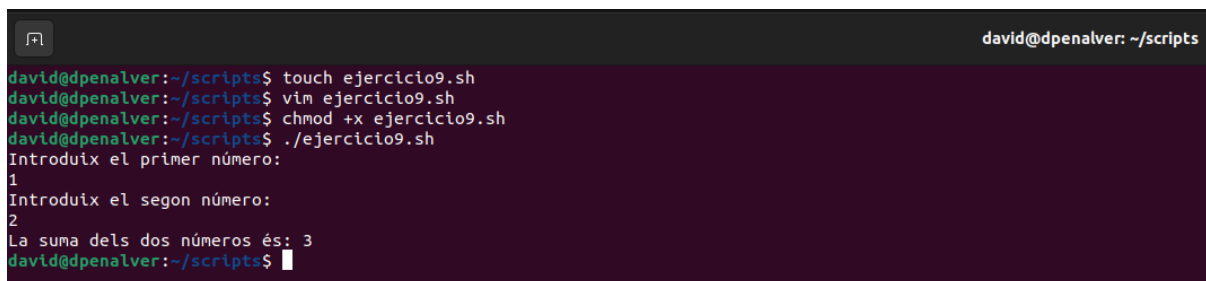
sum() {
    result=$(( $1 + $2 ))
    echo "La suma dels dos números és: $result"
}

echo "Introduix el primer número:"
read num1
echo "Introduix el segon número:"
read num2

sum $num1 $num2
```

Figura 46. Codi de l'script.

Per últim, donem permís d'execució i executem:



```
david@dpنالver: ~/scripts$ touch ejercicio9.sh
david@dpنالver: ~/scripts$ vim ejercicio9.sh
david@dpنالver: ~/scripts$ chmod +x ejercicio9.sh
david@dpنالver: ~/scripts$ ./ejercicio9.sh
Introduix el primer número:
1
Introduix el segon número:
2
La suma dels dos números és: 3
david@dpنالver: ~/scripts$
```

Figura 47. Executant el codi.