

ADSO Training 4

Gestió d'usuaris

Índex

1. Introducció	3
1.1. Objectius	3
2. Profile i entorn d'usuari	3
3. Creació manual d'usuaris	4
4. Creació automàtica d'usuaris	10
5. Connexió remota d'usuaris	15
6. Eliminació i desactivació d'usuaris	22
7. Usuari especial asosh	28
8. Sudo i control d'execució d'aplicacions	30

1. Introducció

Al sistema cada usuari té un compte associat. Un compte són tots els fitxers, recursos i informació que pertanyen a cada usuari. Els comptes d'usuari permeten al sistema diferenciar les dades i processos de cada usuari i permeten als usuaris protegir la seva informació.

Per al kernel els usuaris s'identifiquen amb un nombre enter conegut com l'identificador d'usuari (*user identifier* o *UID*). A més hi ha una base de dades que associa el UID amb un nom textual: el *username*. Aquest *username* és l'utilitzat per l'usuari per fer *login*. La base de dades d'usuaris inclou altra informació relativa a l'usuari com la ruta del directori *home*, el nom complet de l'usuari i l'interpret de comandes (shell).

La creació de un nou usuari inclou l'assignació d'un UID i la modificació de la base de dades d'usuaris per assignar els paràmetres propis de l'usuari. A més és necessari associar almenys un grup a l'usuari i finalment copiar els fitxers de configuració i personalització al directori *home* de cada usuari.

Opcionalment es pot assignar l'usuari a més d'un grup, la qual cosa permet a l'administrador del sistema dividir els usuaris en grups amb diferents permisos i privilegis. D'aquesta manera podem mantenir un millor control sobre què poden fer els usuaris.

1.1. Objectius

Gestionar els usuaris del sistema: realitzar l'alta i baixa d'usuaris i modificar les propietats dels comptes d'usuari.

2. Profile i entorn d'usuari

Quant s'inicia un *login* interactiu, el *shell* automàticament executa un o més fitxers predefinits. Cada *shell* executa fitxers diferents. El shell **bash** executa el fitxer */etc/profile* i a més a més executa el fitxer *.profile*, *.bash_profile* o *.bashrc* del *home* de cada usuari. El fitxer */etc/profile* permet a l'administrador del sistema definir un entorn comú per a tots els usuaris, especialment definint la variable **PATH**. Per altra banda *.bash_profile* o *.bashrc* permet a cada usuari definir el seu propi entorn adequant el *PATH*, el *prompt*, etc.

Quan es crea el directori *home* d'un usuari s'han de copiar els fitxers del directori */etc/skel*. L'administrador del sistema pot posar fitxers a */etc/skel* que donin un entorn inicial pels usuaris. Per exemple, com administradors creeu un fitxer */etc/skel/.bashrc* (si no està ja creat) amb unes definicions bàsiques que després l'usuari podria canviar.

Comproveu que al PATH de tots els usuaris hi sigui el directori `/usr/local/bin` i, si cal, feu que el `.bashrc` modifiqui el PATH per incloure un directori bin situat en el directori *home* de cada usuari (**`$HOME/bin`**).

```
root@PauA (Wed Nov 22):<~># export PATH=$PATH:$HOME/bin
root@PauA (Wed Nov 22):<~># echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/sbin:/usr/sbin:/root/bin
root@PauA (Wed Nov 22):<~>#
```

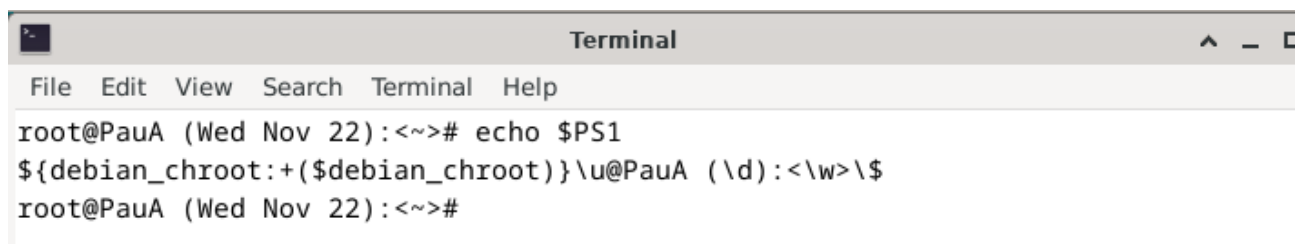
Volem que el *prompt* sigui el *username* seguit de la data actual i finalment "> " (per exemple, el de l'usuari xavim seria "xavim (Tue April 10) > ")

```
root@PauA (Wed Nov 22):<~># export PS1="\u (\d) > "
```

```
root (Wed Nov 22) >
```

Quina variable d'entorn té la definició del prompt?

La variable d'entorn que defineix el prompt en la majoria dels shells de Unix com Bash és PS1. Aquesta és la variable primària que s'utilitza per definir l'aparença del prompt de la línia d'ordres quan s'utilitza un shell interactiu.



```
Terminal
File Edit View Search Terminal Help
root@PauA (Wed Nov 22):<~># echo $PS1
${debian_chroot:+($debian_chroot)}\u@PauA (\d):<\w>\$
root@PauA (Wed Nov 22):<~>#
```

Per verificar la configuració actual del prompt, s'utilitza el comandament `echo $PS1` en el terminal.

3. Creació manual d'usuaris

Ara volem donar d'alta un compte d'usuari per a dos usuaris. Abans de començar trieu els paràmetres de cada usuari. Els usuaris han de formar part del grup *admin*.

Omple la següent taula:

Editeu la base de dades d'usuaris per afegir els nous usuaris. Utilitzeu la comanda **`vipw`** per editar aquest fitxer.

Abans de començar a emplenar la taula, podem comprovar que l'UID de l'usuari que

introduïrem no existeix ja dins el nostre sistema. Això ho farem amb aquesta comanda:

`grep ":<uid>" /etc/passwd` , on uid és el uid que volem comprovar. Si no hi ha cap sortida per pantalla al executar la comanda, voldrà dir que aquest UID està lliure:

```
root@March (Tue Nov 21):~# grep ":<1000>" /etc/passwd
root@March (Tue Nov 21):~# grep ":<1001>" /etc/passwd
root@March (Tue Nov 21):~# grep ":<1002>" /etc/passwd
```

(També podem obrir el fitxer `/etc/passwd` per fer aquesta comprovació, però en casos on tinguem ja molts usuaris serà més còmoda la consulta directa amb `"grep"`).

En el nostre cas, utilitzarem, per exemple, els UID 1001 i 1002.

paràmetres /Usuari	Usuari 1	Usuari 2
UID	1001	1002
<i>Username</i>	usuari1	usuari2
Directori home	/home/usuari1	/home/usuari2
<i>Shell</i>	/bin/bash	/bin/bash
Grups	admin	admin

Quina és la diferencia en usar **vi** o editar directament el fitxer de passwd amb **vi**?

(pista: obriu dos **vi** en sessions diferents)

Al usar **vi**, es bloqueja temporalment l'accés al fitxer `/etc/passwd` per altres processos mentre s'està editant. Això evita conflictes entre usuaris, en cas de concurrència a l'hora de canviar un fitxer.

Si simplement usem l'editor **vi** sense `'pw'`, no es bloqueja l'accés a altres processos, la qual cosa pot provocar que es sobreescriguin diverses edicions del fitxer i tinguem problemes.

Llavors, en el cas que intentem editar el fitxer `passwd` amb la comanda **vi** des de dues sessions diferents, en una d'aquestes sessions no hi podrem accedir. En el cas de **vi**, podem accedir-hi de forma simultània amb el risc que això comporta.

PRIMER PAS PER LA CREACIÓ DE L'USUARI:

1. usar la comanda: **sudo vipw**
2. s'obrirà el fitxer passwd, i afegirem les següents dues línies al final:

```
usuari1:x:1001:1001:Usuari 1:/home/usuari1:/bin/bash
usuari2:x:1002:1002:Usuari 2:/home/usuari2:/bin/bash
```

format:

nom_usuari:contrassenya_encriptada(x):UID:GID:nom_complet:directori_home:shell

De la mateixa manera, utilitzeu la comanda **vigr** per crear un grup per a cada usuari i definir els altres grups que siguin necessaris.

Per afegir qualsevol grup o comprovar els grups que tenim, usem la comanda **vigr**. Veurem tots els grups, i si volem afegir-ne un només hem d'escriure'l a una nova línia amb el format: **nom_grup:x:ID_GRUP**

En el nostre cas, hem afegit el grup "admin" amb gid=1000:

```
colord:x:119:
_ssh:x:120:
geoclue:x:121:
admin:x:1000:
```

Un cop creat el nou grup, podem afegir els usuaris 1 i 2 a aquest amb la comanda:

sudo usermod -aG admin nom_usuari

-a: afegeix a l'usuari als grups especificats sense eliminar-lo dels seus grups actuals.

-G admin: especifica el grup al qual volem afegir l'usuari. Es poden afegir més grups a la vegada, separant-los amb comes.

```
root@March (Tue Nov 21):~# usermod -aG admin usuari1
root@March (Tue Nov 21):~# usermod -aG admin usuari2
root@March (Tue Nov 21):~#
```

Amb la comanda **id nom_usuari** podem comprovar tant l'ID de l'usuari com dels grups als que pertany, per comprovar que tot ha anat correctament:

```
root@March (Tue Nov 21):~# id usuari1
uid=1001(usuari1) gid=1001 groups=1001,1000(aso)
root@March (Tue Nov 21):~# id usuari2
uid=1002(usuari2) gid=1002 groups=1002,1000(aso)
root@March (Tue Nov 21):~#
```

Com es pot desactivar un compte de forma que l'usuari no pugui fer *login*?

Tenim dues opcions per fer-ho:

1. `sudo usermod -L nom_usuari`:

bloqueja el compte i l'usuari simplement veurà un missatge dient que té el compte bloquejat.

2. `sudo usermod -s /usr/sbin/nologin nom_usuari`:

l'usuari veurà un missatge indicant que l'accés està prohibit, ja que li hem modificat el shell d'accés perquè no pugui fer res.

Desactiveu els comptes nous fins que no hagi finalitzat de donar d'alta els usuaris.

Desactivem el compte: `root@March (Tue Nov 21):~# usermod -L usuari2`

Al acabar reactivarem el compte amb: `root@March (Tue Nov 21):~# usermod -U usuari2`

Creeu el directori *home* de cada usuari, copieu els fitxers que estiguin a */etc/skel* i assigneu el propietari i permisos adequats per al directori *home* i per a tots els fitxers que estiguin dintre del directori.

```
root@March (Tue Nov 21):~# mkdir /home/usuari1
root@March (Tue Nov 21):~# cp -r /etc/skel/. /home/usuari1
root@March (Tue Nov 21):~# chown -R usuari1:admin /home/usuari1
```

La primera comanda crea el directori.

La segona, copia */etc/skel* a aquest.

La tercera fa que l'usuari1 sigui el "propietari" del seu fitxer home.

Ho repetirem amb l'usuari2.

Ara hem de donar permisos:

```
root@March (Tue Nov 21):~# chmod 755 /home/usuari1
root@March (Tue Nov 21):~# chmod -R 700 /home/usuari1/.ssh
```

1. **La primera comanda** gestiona els permisos de lectura, escriptura i execució per al propietari, el grup i la resta d'usuaris. (755 = 7 - 5 - 5), respectivament:

El **propietari(usuari1)** té permisos lectura, escriptura i execució del directori */home/usuari1*. **(7)**

El **grup admin** té permisos de lectura i execució, però no d'escriptura. (5)

Els **altres usuaris** que no siguin usuari1 ni grup admin, tenen permisos de lectura i execució, però no d'escriptura. (5).

2. **La segona comanda** gestiona els permisos de l'arxiu .ssh per assegurar les claus d'autenticació ssh. El conjunt 700 és comú a aquest directori i a altres fitxers importants per una connexió segura.

Ara assigneu una clau (password) per a cada usuari nou.

Per raons de seguretat la clau no es posa directament al fitxer /etc/passwd. Per això hi ha un altre fitxer anomenat /etc/shadow que només té permisos de lectura per al superusuari. En aquest fitxer es posa la clau xifrada i altres paràmetres associats a la vigència de la clau.

Amb quina comanda es pot editar de manera segura el fitxer de *shadow*?

```
root@March (Tue Nov 21):/# passwd usuari1
New password:
Retype new password:
passwd: password updated successfully
```

Amb aquesta comanda (**passwd nom_usuari**) modificarem la password canviant el fitxer /etc/shadow evitant la possibilitat d'errors editant aquest fitxer.

Quin es el significat dels altres paràmetres que es poden definir al fitxer de shadow?

```
usuari1:$y$j9T$HmoL1Bj2K1ESAfwg0dJ7r,$MvqE1t1XMok,a6FfKCGuDtnSQNvSSg0J4%arveyd7f6:19682:::::::
```

Com podem veure per el número de caràcters ':' dins d'una línia qualsevol al fitxer shadow, hi ha un màxim de 9 paràmetres, els quals son, respectivament:

1. Nom d'usuari
2. Contrasenya xifrada
3. Dies des de l'últim canvi de contrasenya realitzat
4. Dies fins que l'usuari haurà de fer un canvi de contrasenya.
5. Dies abans que l'usuari pugui tornar a canviar la contrasenya.
6. Dies d'avertència abans de l'expiració de la contrasenya.
7. Dies fina a l'expiració de la contrasenya.
8. Data d'expiració de l'usuari.
9. Camp reservat per a ús futur, que pot contenir info adicional.

Amb quina comanda es poden modificar aquests paràmetres?

Tots aquests paràmetres es poden modificar amb la comanda **chage**.

Per exemple:

sudo chage -d YYYY-MM-DD nom_usuari: estableix la data de l'últim canvi de contrasenya.

sudo chage -E YYYY-MM-DD: estableix la data d'expiració de l'usuari.

sudo chage -M nombre_de_dies nom_usuari: estableix el màxim nombre de dies entre canvis de contrasenya.

I per acabar, amb la comanda **sudo chage -l nom_usuari** podem veure la informació actual de l'usuari, incloent els detalls relacionats amb les contrasenyes:

```
root@March (Tue Nov 21):/# chage -l usuari1
Last password change           : Nov 21, 2023
Password expires               : never
Password inactive              : never
Account expires                : never
Minimum number of days between password change : -1
Maximum number of days between password change : -1
Number of days of warning before password expires : -1
```

Per editar altres paràmetres del compte d'usuari es poden utilitzar les comandes: **chfn** i **chsh**. Utilitzeu aquestes comandes per assignar valors adequats als comptes creats.

Amb la comanda **chfn nom_usuari** veiem que podem afegir informació adicional de l'usuari, com el seu nom real, el telèfon i alguna nota extra:

```
root@March (Tue Nov 21):/# chfn usuari1
Changing the user information for usuari1
Enter the new value, or press ENTER for the default
  Full Name [Usuari 1]: paquito
   Room Number []:
  Work Phone []: 645645454
  Home Phone []: 978787878
   Other []: res
```

La comanda **chsh** serveix per canviar l'interpret de comandes associat a un usuari. En el cas que veurem a continuació, hem canviat l'interpret a **/bin/bash**:

```
root@March (Tue Nov 21):/# chsh -s /bin/bash usuari2
```

L'estructura és **chsh -s /ruta/del/shell nom_usuari**

4. Creació automàtica d'usuaris

La majoria de les distribucions de Linux inclouen programes per automatitzar les tasques de creació i modificació de dades d'usuaris. Unes d'aquestes aplicacions son **useradd** i **adduser**, que permeten crear usuaris i assignar els diferents paràmetres necessaris per donar d'alta cada compte.

Utilitzeu aquestes comandes per donar d'alta els usuaris següents:

- Product Owners: PO1, PO2, PO3
- Scrum Master: SM1, SM2
- Equip de Desenvolupament (ED): El nom d'usuari del compte serà: nomX, on nom és el vostre nom i X la primera lletra del vostre cognom en minúscules.

Trieu i justifiqueu el lloc més adequat per als home de tots els usuaris.

La ubicació estàndard on afegir els usuaris es a **/home** pero podem afegir-los on vulguem, per exemple podem crear un directori per agrupar els usuaris allà o agruparlos per directori del grup al que pertanyen.

```
mkdir /home/product_owners
mkdir /home/scrum_masters
mkdir /home/dev_team
```

Els posarem a aquest directori per a tenir-los endreçats i agrupats a la ubicació estàndard. A part, afegir els usuaris dins el directori de **/home** ens proporciona:

- **Seguretat i Organització:** Manté els directoris personals dels usuaris separats dels directoris del sistema ubicats a /, /usr, /bin, etc.
- **Gestió de Permisos:** Facilita l'assignació i la gestió de permisos.
- **Backups i Manteniment:** Simplifica els processos de suport i manteniment, ja que els directoris dels usuaris estan tots en un lloc centralitzat.

Els permisos de cadascun d'aquests grups d'usuaris (POs, SMs i ED) venen definits de la següent forma:

Els POs tindran control d'accés a nivell de grup a tots els fitxers de tots els usuaris definits. És a dir: l'accés dels POs a fitxers i directoris dels altres usuaris vindrà determinat pels permisos de grup d'aquests fitxers i directoris. No tindrà accés als altres PO

Els SMs tindran control d'accés, a nivell de grup, a tots els fitxers de tots els usuaris, exceptuant els dels usuaris POs i els altres SM.

Els membres del ED NO tindran accés, a nivell de grup, als fitxers dels POs, ni dels SMs, ni dels altres membres del ED.

Tingueu en compte que les condicions anteriors estan especificant els nivells d'accés. El nivell d'accés només indica a quin nivell es miren els privilegis sobre un fitxer o directori determinat (user, group, other).

3. Creació automàtica d'usuaris

Creemos los tres grupos (PO, SM y ED):

```
groupadd PO
groupadd SM
groupadd ED
```

Creemos todos los usuarios con su directorio correspondiente, asignando el grupo principal y luego agregando los grupos adicionales a los que podrá acceder.

```
useradd -d /home/PO1 -m -g PO -G SM,ED -s /bin/bash PO1
useradd -d /home/PO2 -m -g PO -G SM,ED -s /bin/bash PO2
useradd -d /home/PO3 -m -g PO -G SM,ED -s /bin/bash PO3
useradd -d /home/SM1 -m -g SM -G ED -s /bin/bash SM1
useradd -d /home/SM2 -m -g SM -G ED -s /bin/bash SM2
useradd -d /home/jialec -m -g ED -s /bin/bash jialec
```

Le damos los permisos que nos pide el enunciado.

```
chown PO1:PO /home/PO1
chown PO2:PO /home/PO2
chown PO3:PO /home/PO3
chmod 700 /home/PO1
chmod 700 /home/PO2
chmod 700 /home/PO3

chown SM1:SM /home/SM1
chown SM2:SM /home/SM2
chown :PO /home/SM1
chown :PO /home/SM2
chmod 770 /home/SM1
chmod 770 /home/SM2

chown jialec:ED /home/jialec
chown :PO /home/jialec
chown :SM /home/jialec
chmod 770 /home/jialec
```

Mostra tot el procés de creació indicant pas a pas que s'ha fet

```
root@MarcPerez (Tue Nov 21):<~># cd /home/  
root@MarcPerez (Tue Nov 21):</home># ls  
aso lost+found  
root@MarcPerez (Tue Nov 21):</home># █
```

```
root@MarcPerez (Tue Nov 21):</># mkdir /home/product_owners  
root@MarcPerez (Tue Nov 21):</># mkdir /home/scrum_masters  
root@MarcPerez (Tue Nov 21):</># mkdir /home/dev_team  
root@MarcPerez (Tue Nov 21):</># ls /home  
aso dev_team lost+found product_owners scrum_masters  
root@MarcPerez (Tue Nov 21):</>#
```

Lo primer que farem serà crear els grups i donar els permisos:

Comanda: **sudo groupadd nombreGrupo**

```
sudo groupadd product_owners
```

```
sudo groupadd scrum_masters
```

```
sudo groupadd dev_team
```

```
sudo groupadd access_group
```

Aquest últim grup es un grup de permisos, el qual està format pels product_owners y els scrum_masters.

L'hem creat perquè així podem assignar privilegis sobre els dev_team.

```
root@MarcPerez (Tue Nov 21):</># groupadd product_owners  
root@MarcPerez (Tue Nov 21):</># groupadd scrum_masters  
root@MarcPerez (Tue Nov 21):</># groupadd dev_team  
root@MarcPerez (Tue Nov 21):</># groupadd access_group  
root@MarcPerez (Tue Nov 21):</># █
```

Comprovem que ha funcionat, podem fer-ho amb la comanda **getent group** o obrint el fitxer **/etc/group**

```
product_owners:x:1001:
scrum_masters:x:1002:
dev_team:x:1003:
access_group:x:1004:
root@MarcPerez (Tue Nov 21):</># getent group
```



```
52  sgx:x:111:
53  avahi:x:112:
54  pulse:x:113:
55  pulse-access:x:114:
56  scanner:x:115:saned
57  saned:x:116:
58  lightdm:x:117:
59  polkitd:x:998:
60  rtkit:x:118:
61  colord:x:119:
62  _ssh:x:120:
63  geoclue:x:121:
64  product_owners:x:1001:
65  scrum_masters:x:1002:
66  dev_team:x:1003:
67  access_group:x:1004:
68
```

A continuació donarem permisos

Comanda: **sudo chmod XXX /path/group**

Primer Dígit (7): Aquest dígit especifica els permisos per al propietari del fitxer o directori. Al sistema octal, el número 7 és una suma de:

4: Permís de lectura (read).

2: Permís d'escriptura (write).

1: Permís d'execució (execute).

Per tant, 7 significa que el propietari té permisos complets (lectura, escriptura i execució).

Segon Dígit (7): Aquest dígit defineix els permisos per al grup al qual pertany el fitxer o directori. Com el primer dígit, 7 aquí significa que els membres del grup també tenen permisos complets.

Tercer Dígit (0): Aquest dígit estableix els permisos per a "altres" usuaris, és a dir, tots els altres usuaris del sistema que no són ni el propietari ni part del grup propietari. Un valor de 0 significa que altres usuaris no tenen permís d'accés (ni lectura, escriptura ni execució).

```
sudo chmod 700 /home/product_owners
```

```
sudo chown :product_owners /home/scrum_masters
sudo chmod 770 /home/scrum_masters
```

```
sudo chown :access_group /home/dev_team
sudo chmod 770 /home/dev_team
```

```
root@MarcPerez (Tue Nov 21):</># chmod 700 /home/product_owners/
root@MarcPerez (Tue Nov 21):</># chown :product_owners /home/scrum_masters/
root@MarcPerez (Tue Nov 21):</># chmod 770 /home/scrum_masters/
root@MarcPerez (Tue Nov 21):</># chown :access_group /home/dev_team/
root@MarcPerez (Tue Nov 21):</># chmod 770 /home/dev_team/
root@MarcPerez (Tue Nov 21):</>#
```

El siguiente paso será crear los usuarios

Comanda: **sudo useradd -m -d /home/grupo/nombreDirectorioUsuario -g nombreDelGrupo nombreX**

```
sudo useradd -m -d /home/product_owners/P01 -G product_owners,access_group P01
sudo useradd -m -d /home/product_owners/P02 -G product_owners,access_group P02
sudo useradd -m -d /home/product_owners/P03 -G product_owners,access_group P03
```

```
sudo useradd -m -d /home/scrum_masters/SM1 -G scrum_masters,access_group SM1
sudo useradd -m -d /home/scrum_masters/SM2 -G scrum_masters,access_group SM2
```

```
sudo useradd -m -d /home/dev_team/MarcP -g dev_team MarcP
```

```
root@MarcPerez (Tue Nov 21):</># useradd -m -d /home/product_owners/P01 -G product_owners,access_group P01
root@MarcPerez (Tue Nov 21):</># useradd -m -d /home/product_owners/P02 -G product_owners,access_group P02
root@MarcPerez (Tue Nov 21):</># useradd -m -d /home/product_owners/P03 -G product_owners,access_group P03
root@MarcPerez (Tue Nov 21):</># useradd -m -d /home/scrum_masters/SM1 -G scrum_masters,access_group SM1
root@MarcPerez (Tue Nov 21):</># useradd -m -d /home/scrum_masters/SM2 -G scrum_masters,access_group SM2
root@MarcPerez (Tue Nov 21):</># useradd -m -d /home/dev_team/MarcP -g dev_team MarcP
root@MarcPerez (Tue Nov 21):</># █
```

POSSIBLE CANVI: POS ACONSEGUIEXIN ENTRAR A SCRUM_MASTERS

sudo usermod -a -G scrum_masters NOMPO

(no els entra com a users de scrum sino q els hi dona els permisos necessaris del po)

```

root@marionaF (Sun Dec 10):</home># sudo usermod -a -G scrum_masters P01
root@marionaF (Sun Dec 10):</home># sudo usermod -a -G scrum_masters P02
root@marionaF (Sun Dec 10):</home># sudo usermod -a -G scrum_masters P03
root@marionaF (Sun Dec 10):</home># ls -ld /home/scrum_masters/
drwxrwx--- 4 root scrum_masters 4096 Dec 10 16:39 /home/scrum_masters/
root@marionaF (Sun Dec 10):</home># groups P01
P01 : product_owners scrum_masters
root@marionaF (Sun Dec 10):</home># groups P02
P02 : product_owners scrum_masters
root@marionaF (Sun Dec 10):</home># groups P03
P03 : product_owners scrum_masters
root@marionaF (Sun Dec 10):</home># ls

```

PO DINS DE DEV TEAM:gg

```

root@marionaF (Sun Dec 10):</home># sudo usermod -a -G dev_team P01
root@marionaF (Sun Dec 10):</home># sudo usermod -a -G dev_team P02
root@marionaF (Sun Dec 10):</home># sudo usermod -a -G dev_team P03
root@marionaF (Sun Dec 10):</home># groups P01
P01 : product_owners dev_team scrum_masters
root@marionaF (Sun Dec 10):</home># groups P02
P02 : product_owners dev_team scrum_masters
root@marionaF (Sun Dec 10):</home># groups P03
P03 : product_owners dev_team scrum_masters

```

SM A DINS DE DEV TEAM

```

root@marionaF (Sun Dec 10):</home># sudo usermod -a -G dev_team SM1
root@marionaF (Sun Dec 10):</home># sudo usermod -a -G dev_team SM2

```

```

root@marionaF (Sun Dec 10):</home># groups SM1
SM1 : scrum_masters dev_team
root@marionaF (Sun Dec 10):</home># groups SM2
SM2 : scrum_masters dev_team

```

ELS USUARIS HAN DE QUEDAR AIXI:

```

root@marionaF (Sun Dec 10):</home/dev_team># ls
Marionaf
root@marionaF (Sun Dec 10):</home/dev_team># cd ..
root@marionaF (Sun Dec 10):</home># cd product_owners/
root@marionaF (Sun Dec 10):</home/product_owners># ls
P01 P02 P03
root@marionaF (Sun Dec 10):</home/product_owners># cd ..
root@marionaF (Sun Dec 10):</home># cd scrum_masters/
root@marionaF (Sun Dec 10):</home/scrum_masters># ls
SM1 SM2

```

POSARLIS CONTRASENYES! canviara el /etc/shadow

```
root@marionaF (Sun Dec 10):<~># passwd P01
New password:
Retype new password:
passwd: password updated successfully
root@marionaF (Sun Dec 10):<~># passwd P02
New password:
Retype new password:
passwd: password updated successfully
root@marionaF (Sun Dec 10):<~># passwd P03
New password:
Retype new password:
passwd: password updated successfully
root@marionaF (Sun Dec 10):<~># passwd SM1
New password:
Retype new password:
passwd: password updated successfully
root@marionaF (Sun Dec 10):<~># passwd SM2
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
root@marionaF (Sun Dec 10):<~># passwd SM2
New password:
Retype new password:
passwd: password updated successfully
```

Podemos ponernos a comprobar que se han añadido correctamente mirando en **/etc/passwd** o bien en **/etc/shadow**


```

19 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/n
20 _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
21 systemd-timesync:x:101:102:systemd Time Synchronizat
22 systemd-network:x:102:103:systemd Network Management
23 systemd-resolve:x:103:104:systemd Resolver,,,:/run/s
24 messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
25 aso:x:1000:1000:,,,:/home/aso:/bin/bash
26 systemd-coredump:x:999:999:systemd Core Dumper:/usr
27 avahi:x:105:112:Avahi mDNS daemon,,,:/run/avahi-daem
28 pulse:x:106:113:PulseAudio daemon,,,:/run/pulse:/usr
29 saned:x:107:116::/var/lib/saned:/usr/sbin/nologin
30 lightdm:x:108:117:Light Display Manager:/var/lib/lig
31 polkitd:x:998:998:polkit:/nonexistent:/usr/sbin/nolo
32 rtkit:x:109:118:RealtimeKit,,,:/proc:/usr/sbin/nolog
33 colord:x:110:119:colord colour management daemon,,:
34 geoclue:x:111:121::/var/lib/geoclue:/usr/sbin/nologi
35 P01:x:1001:1005::/home/product_owners/P01:/bin/sh
36 P02:x:1002:1006::/home/product_owners/P02:/bin/sh
37 P03:x:1003:1007::/home/product_owners/P03:/bin/sh
38 SM1:x:1004:1008::/home/scrum_masters/SM1:/bin/sh
39 SM2:x:1005:1009::/home/scrum_masters/SM2:/bin/sh
40 MarcP:x:1006:1003::/home/dev_team/MarcP:/bin/sh

```

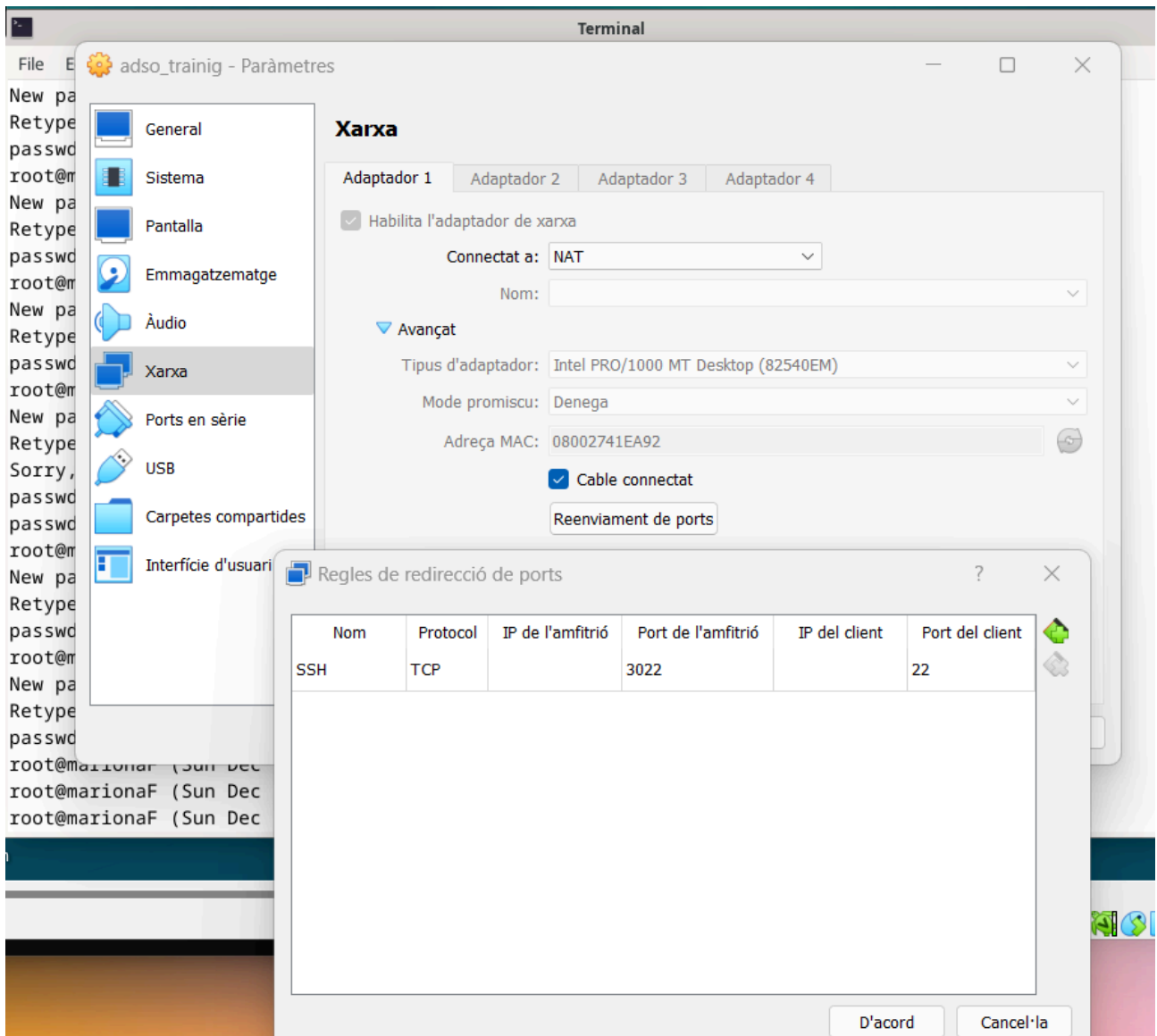
```

19 _apt:!:18155:0:99999:7:::
20 systemd-timesync:!:18155:0:99
21 systemd-network:!:18155:0:999
22 systemd-resolve:!:18155:0:999
23 messagebus:!:18155:0:99999:7:
24 aso:$6$P$FKMMgdHssUBnFi3$5GRfV
25 systemd-coredump:!!:18892:::
26 admin:$6$p/oGKX/XwMdLR9C1$p19
27 avahi:!:19659:~::~:
28 pulse:!:19659:~::~:
29 saned:!:19659:~::~:
30 lightdm:!:19659:~::~:
31 polkitd:!*:19659:~::~:
32 rtkit:!:19659:~::~:
33 colord:!:19659:~::~:
34 geoclue:!:19660:~::~:
35 P01:!:19682:0:99999:7:::
36 P02:!:19682:0:99999:7:::
37 P03:!:19682:0:99999:7:::
38 SM1:!:19682:0:99999:7:::
39 SM2:!:19682:0:99999:7:::
40 MarcP:!:19682:0:99999:7:::
41

```

COMPROVACIÓ!!! (microhakaton4)

ABANS CANVIAR per ssh (descarregar openssh)



Des de terminal de WINDOWS posar el `ssh -p 3022 USER_A_PROVAR@ip de la maquina virtual`

ip VM → ifconfig

PO a altres carpetas

```

PS C:\Users\mft19> ssh -p 3022 P01@127.0.0.1
P01@127.0.0.1's password:
Linux aso-client 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Dec 10 17:31:28 2023 from 10.0.2.2
$ pwd
/home/product_owners/P01
$ cd /home/scrum_masters
$ ls
SM1 SM2
$ cd /home/dev_team
$ ls
Marionaf

```

SM a altres carpetas

```

PS C:\Users\mft19> ssh -p 3022 SM1@127.0.0.1
SM1@127.0.0.1's password:
Linux aso-client 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
$ pwd
/home/scrum_masters/SM1
$ cd /home/product_owners
-sh: 2: cd: can't cd to /home/product_owners
$ ^[[A^[[A^Z
$ ^[[A^Z
$ cd /home/dev_team
$ ls
Marionaf
$ exit

```

ED a altes carpetas

```

PS C:\Users\mft19> ssh -p 3022 Marionaf@127.0.0.1
Marionaf@127.0.0.1's password:
Linux aso-client 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
$ pwd
/home/dev_team/Marionaf
$ cd /home/product_owners
-sh: 2: cd: can't cd to /home/product_owners
$ cd /home/srum_masters
-sh: 3: cd: can't cd to /home/srum_masters
$ cd /home/dev_team
$ ls
Marionaf
$ cd /home/|

```

5. Connexió remota d'usuaris

Els usuaris de la nostra màquina han de tenir l'opció de poder connectar-se en remot de una forma segura.

instal·leu el paquet **openssh-server** i **openssh-client** (si cal)

Per a començar provarem a instal·lar el client per a veure si el tenim ja instal·lat, utilitzarem la comanda `apt-get install openssh-client`:

```
root@MarcR (Thu Nov 16):<~># apt-get install openssh-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-client is already the newest version (1:9.2p1-2+deb12u1).
openssh-client set to manually installed.
The following packages were automatically installed and are no longer required:
  bsdmainutils libcupsfilters1 libcupsimage2 libflac8 libgs9-common libicu63
  libilmbase23 libldap-2.4-2 libmpdec2 libnetpbm10 libopenexr23 libperl5.28
  libpython3.7-minimal libpython3.7-stdlib libreadline7 libruby2.5 libtiff5 libwebp6
  libwmf-0.2-7 libwmf0.2-7 libx265-165 ncal perl-modules-5.28 pigz python3.7-minimal
  ruby-did-you-mean ruby-minitest ruby-power-assert ruby-test-unit
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@MarcR (Thu Nov 16):<~># █
```

Com podem veure en el meu cas ja el tenia instal·lat, aleshores procedirem a instal·lar el server amb la comanda `apt-get install openssh-server`:

```

Preparing to unpack .../openssh-server_1%3a9.2p1-2+deb12u1_amd64.deb ...
Unpacking openssh-server (1:9.2p1-2+deb12u1) ...
Selecting previously unselected package ncurses-term.
Preparing to unpack .../ncurses-term_6.4-4_all.deb ...
Unpacking ncurses-term (6.4-4) ...
Setting up runit-helper (2.15.2) ...
Setting up openssh-sftp-server (1:9.2p1-2+deb12u1) ...
Setting up openssh-server (1:9.2p1-2+deb12u1) ...

Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:7Vh/zrQ7L5a0jHlQGvle+JTIWWrwq/YAVfMfRJtQCWk root@aso-client (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:JeNrHE+PIVkuHvc5Ki9GeR9XJaijEgz6JPcqJtE6XnE root@aso-client (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:sfszM+7cFKGFivA8gsHua9sk2lQc20pNwJFYebR4Rvg root@aso-client (ED25519)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
ssh.socket is a disabled or a static unit, not starting it.
Setting up ncurses-term (6.4-4) ...
Processing triggers for man-db (2.11.2-2) ...
root@MarcR (Thu Nov 16):<~># █

```

Comproveu que us podeu connectar remotament a un altra màquina.

Una vegada que ja tenim instal·lat els openssh comprovem que ens podem connectar i que es poden connectar a nosaltres:

Connexió desde màquina virtual a un altre màquina:

```

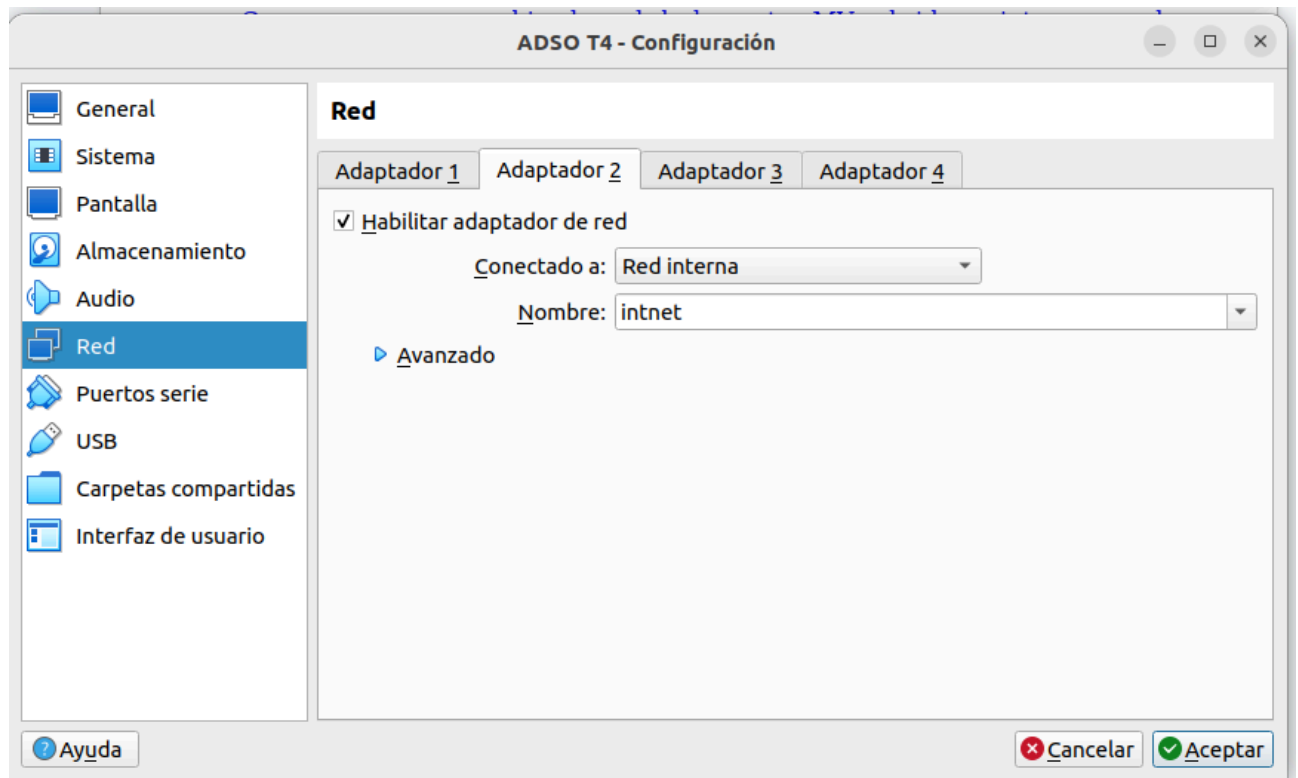
root@MarcR (Thu Nov 16):<~># ssh e8021438@ubiwan.epsevg.upc.edu
The authenticity of host 'ubiwan.epsevg.upc.edu (147.83.13.23)' can't be established.
ED25519 key fingerprint is SHA256:e6yk+grEKlqN2iSFsmb87IIS4k0o6qVJfc3t2H4Q+xE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ubiwan.epsevg.upc.edu' (ED25519) to the list of known hosts.
e8021438@ubiwan.epsevg.upc.edu's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-166-generic x86_64)

Last login: Mon Nov  6 13:51:05 2023 from 37.15.176.188
e8021438@ubiwan:~$ █

```

Començarem per afegir la red a la nostre MV en mode bridge o interna, en el meu cas la

faré interna en les dues màquines:



A continuació entrarem a les nostres màquines i ens anirem a l'arxiu /etc/network/interfaces per a afegir la nova interfície de la següent manera:

Maquina 1:

```
GNU nano 7.2 /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto enp0s3
iface enp0s3 inet dhcp

auto enp0s8
iface enp0s8 inet static
    address 192.168.1.100
    netmask 255.255.255.0
```

Maquina 2:

```
GNU nano 7.2 /etc/network/interfaces *
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto enp0s3
iface enp0s3 inet dhcp

auto enp0s8
iface enp0s8 inet static
    address 192.168.1.101
    netmask 255.255.255.0
```

Habilitem l'enrutament descomentant la línia de l'arxiu `/etc/sysctl.conf` com en la imatge:

```

GNU nano 7.2 /etc/sysctl.conf *
#kernel.printk = 3 4 1 3

#####
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

```

enp0s3 es la nostra NAT, mentre que enp0s8 es la nostre xarxa interna.

Podem comprovar que es poden veure entre elles amb un ping:

Ping desde la màquina 2(192.168.1.101) a la màquina 1(192.168.1.100):

```

link/ether 08:00:27:22:9e:c4 brd ff:ff:ff:ff:ff:ff
inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
    valid_lft 86372sec preferred_lft 86372sec
inet6 fe80::a00:27ff:fe22:9ec4/64 scope link
    valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:db:10:9e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.101/24 brd 192.168.1.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fedb:109e/64 scope link
        valid_lft forever preferred_lft forever
root@MarcR (Sat Nov 18):<~># systemctl restart networking
root@MarcR (Sat Nov 18):<~># ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=2.37 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=1.71 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=1.04 ms
64 bytes from 192.168.1.100: icmp_seq=4 ttl=64 time=1.12 ms
^C

```

Una vegada modificada la red procedirem a crear unes claus desde la màquina la qual

ens volem connectar amb la comanda `ssh-keygen -t rsa`, això ens crearà una clau privada i una pública:

```
root@MarcR (Sat Nov 18):<~/Downloads># ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:eBelI44RSoNt39VcRolyJ9hMayJct080PFccx2gz5Tc root@aso-client
The key's randomart image is:
+---[RSA 3072]-----+
|  oo . .B+++B=|
| ..oo o .+=0+Bo+|
| .....+. =oB+oEo|
|   .o = B  o|
|   o S .  o |
|   . .      |
|             |
|             |
|             |
+-----[SHA256]-----+
root@MarcR (Sat Nov 18):<~/Downloads>#
```

Per a trobar les nostres claus ens anirem a la carpeta personal de l'usuari que tenim `/.ssh`:

```
root@MarcR (Sat Nov 18):</etc/ssh># cd /root/.ssh/
root@MarcR (Sat Nov 18):<~/ssh># ls
id_rsa id_rsa.pub known_hosts known_hosts.old
root@MarcR (Sat Nov 18):<~/ssh>#
```

A continuació copiarem la clau `id_rsa.pub` de la nostra màquina i la afegirem en la màquina on ens volem connectar a la carpeta `~/ssh/` amb el nom de `authorized_keys` si no volem posar contrasenya quan ens connectem, si ja teniem un altre clau copiada senzillament obrim l'arxiu i l'enganxem a sota de la que ja existeix. Si no ens importa la contrasenya ja podem executar la comanda `ssh` amb la ip de la màquina:

```

root@MarcR (Sat Nov 18):</media/usb># ssh marc@192.168.1.100
The authenticity of host '192.168.1.100 (192.168.1.100)' can't be established.
ED25519 key fingerprint is SHA256:sfszM+7cFKGFivA8gsHua9sk2lQc20pNwJFYebR4Rvg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.100' (ED25519) to the list of known hosts.
marc@192.168.1.100's password:
Linux aso-client 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

If you have any issue or problem during the using of this operative system, please feel free
to contact with one of our system managers to solve any problem you may have or
to ask
anything informative about our system.
Here's one of our system managers email: marc.roca.i.salvans@estudiantat.upc.edu

Thank you very much

marc@aso-client:~$ █

```

I ja ens podriem moure per la màquina sense problemes, òbviament amb les limitacions de l'usuari.nano

6. Eliminació i desactivació d'usuaris

Per donar de baixa un usuari és necessari eliminar tots els seus fitxers, les bústies de correu, treballs d'impressió, treballs **cron** i **at** i totes les referències a l'usuari. Després d'això es poden esborrar les línies associades a l'usuari al fitxer de passwd i de grups. Com un usuari pot tenir fitxers fora del seu directori home és necessari buscar per tot l'arbre de directoris els fitxers que pertanyen l'usuari i esborrar-los.

Crea un usuari de prova (o escolleix un existent) i afegeix fitxers al seu home.

Crearem un usuari de prova amb tots els possibles directoris que pot utilitzar:

sudo adduser prova

```
root@francesco0 (Wed Nov 22):<~># sudo adduser prova
Adding user `prova' ...
Adding new group `prova' (1001) ...
Adding new user `prova' (1001) with group `prova (1001)' ...
Creating home directory `/home/prova' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for prova
Enter the new value, or press ENTER for the default
    Full Name []: prova
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
Adding new user `prova' to supplemental / extra groups `users' ...
Adding user `prova' to group `users' ...
root@francesco0 (Wed Nov 22):<~># █
```

Automàticament et dirà de crear una contrasenya, per la prova utilitzarem la contrasenya:
1234

Anar donant a ENTER per no especificar cap dada de telèfons o altres

Si no es crea el directori directament, el podem crear manualment amb les següents instruccions:

sudo mkdir /home/prova

Donar privilegis d'usuari als directori creat:

sudo chown user:user /home/user

sudo chmod 755 /home/prova

```

root@francesco0 (Wed Nov 22):<~># cd /home
root@francesco0 (Wed Nov 22):</home># ls -la
total 20
drwxr-xr-x  5 root  root  4096 Nov 22 16:34 .
drwxr-xr-x 18 root  root  4096 Oct 18 22:08 ..
drwxr-xr-x  3 root  root  4096 Sep 28 07:56 homeA
drwxr-xr-x  6 root  root  4096 Sep 28 07:58 homeB
drwx----- 2 prova prova 4096 Nov 22 16:34 prova
root@francesco0 (Wed Nov 22):</home># cd
root@francesco0 (Wed Nov 22):<~># chown prova:prova /home/prova
root@francesco0 (Wed Nov 22):<~># chmod 755 /home/prova
root@francesco0 (Wed Nov 22):<~># cd /home
root@francesco0 (Wed Nov 22):</home># ls -la
total 20
drwxr-xr-x  5 root  root  4096 Nov 22 16:34 .
drwxr-xr-x 18 root  root  4096 Oct 18 22:08 ..
drwxr-xr-x  3 root  root  4096 Sep 28 07:56 homeA
drwxr-xr-x  6 root  root  4096 Sep 28 07:58 homeB
drwxr-xr-x  2 prova prova 4096 Nov 22 16:34 prova
root@francesco0 (Wed Nov 22):</home>#

```

És una bona pràctica de seguretat primer desactivar el compte de l'usuari abans de començar el procés de donar-lo de baixa.

Una manera de desactivar un compte, a banda d'invalidar el password, consisteix en canviar el *shell* de l'usuari per un un programa senzill que només escriu a la pantalla un missatge i dona informació a l'usuari de les raons per les quals el seu compte d'usuari ha estat desactivat. Per això es pot crear un 'tail script'. Per exemple:

```
#!/usr/bin/tail -n 2

This account has been closed due to a security problem. Please contact the system
administrator.
```

Aquest script es pot posar com shell de l'usuari usant la comanda **chsh** i es pot guardar en un directori separat, per exemple **/usr/local/lib/no-login**.

Utilitzeu la comanda `chsh` per posar un *tail script* per desactivar el compte de l'usuari creat .

Com es pot comprovar que el compte ha quedat desactivat?

Per comprovar si un compte ha quedat desactivat, podem fer-ho de 2 maneres:

Iniciem sessió, si podem entrar-hi vol dir que no ha quedat desactivat, altrament el compte ha quedat desactivat. L'altre manera es accedir a l'arxiu `/etc/passwd` i mirar si el compte està desactivat o no.

Fes un backup amb tots els fitxers de l'usuari (tingueu en compte que pot ser una llista molt llarga de fitxers. Pista: feu servir **xargs**)

Per fe el backup de tots els fitxers d'usuari utilitzarem la comanda **xargs** perquè copia arxius utilitzant un filtre. En el nostre cas el filtre són tots els arxius de l'usuari indicat.

Quin problema hi ha amb els fitxers que tinguin espais al seu nom? Com es pot resoldre això? (veure les opcions de la comanda **xargs** o la opció `-exec` de **find**)

Com que hi ha fitxers que tenen espais al seu nom, en funció del programa/comanda utilitzats generen problemes, perquè quan escrivim comandes els espais s'utilitzen per introduir nous paràmetres. Per solucionar aquest problema sense canviar el nom del fitxer, utilitzem una comanda que representa cada espai amb el caràcter `\`. Per exemple, si tenim un fitxer que es diu `Training 4`, a la línia de comandes, per poder interpretar-lo escriurem `Training\4`.

Busca tots els fitxers de l'usuari i esborrar-los.

Ara crea un script que donat el nom d'usuari, faci un backup del seu directori home, esborri tots el fitxers que l'usuari tingui al sistema i canviï el shell per un *tail script* que avisi a l'usuari que el seu compte ha estat esborrat.

Script `delete_user.sh` :

```
#!/bin/bash

usage="Usage: delete_user.sh [usuari]"

if [ $# -ne 1 ]; then
    echo $usage
```

```
        exit 1
fi

chsh -s /usr/local/lib/no-login/prova $1 #és un script que conté una linia que diu que el compte
s'ha tancat
echo "This account has been closed due to a security problem. Please contact the system
administrator"

if [ ! -d $HOME/backups ]; then
    mkdir $HOME/backups
    echo "Backups directory created"
fi

dir_home=`cat /etc/passwd | grep "^$1\>" | cut -d: -f6`

if [ -d $dir_home ]; then
    tar -cvzf $HOME/backups/$1.tar.gz $dir_home
    rm -r $dir_home
    echo "A $1 copy has been succesfully saved"
else
    echo "The directory $dir_home does not exist"
fi

find / -user $1 -exec rm -r "{}" \; 2> /dev/null

echo "All $1 files have been eliminated"

userdel $1
```

Comprova que s'ha fet correctament

```
root@francesco0 (Wed Nov 22):<~># cd /home/
root@francesco0 (Wed Nov 22):</home># ls
homeA homeB prova1
root@francesco0 (Wed Nov 22):</home># cd
root@francesco0 (Wed Nov 22):<~># cd Downloads/
root@francesco0 (Wed Nov 22):<~/Downloads># sudo ./delete_user.sh prova1
Password:
chsh: PAM: Authentication failure
This account has been closed due to a security problem. Please contact the system administrator
tar: Removing leading `/' from member names
/home/prova1/
/home/prova1/.bash_logout
/home/prova1/.face
/home/prova1/.face.icon
/home/prova1/.bashrc
/home/prova1/.profile
A prova1 copy has been succesfully saved to /root
All prova1 files have been eliminated
root@francesco0 (Wed Nov 22):<~/Downloads># cd
root@francesco0 (Wed Nov 22):<~># cd backups/
root@francesco0 (Wed Nov 22):<~/backups># ls
prova1.tar.gz
root@francesco0 (Wed Nov 22):<~/backups># cd
root@francesco0 (Wed Nov 22):<~># sudo adduser prova1
Adding user `prova1' ...
Adding new group `prova1' (1001) ...
Adding new user `prova1' (1001) with group `prova1 (1001)' ...
Creating home directory `/home/prova1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Chanqing the user information for prova1
```

Podem comprovar que l'usuari s'ha eliminat correctament ja que si intentem crear un nou usuari amb el mateix nom no ens dona cap problema. Si l'usuari no estigués ben eliminat diria que no es pot crear l'usuari prova1 perquè ja existeix.

7. Usuari especial asosh

A Unix hi ha comandes com el **shutdown** per apagar la màquina que només pot executar l'usuari root. En moltes ocasions pot ser interessant que algun altre usuari pugui apagar també la màquina però sense que tingui accés als privilegis de root.

Per aconseguir-ho es demana que creeu un compte especial que serveixi per executar un shell simplificat que permetrà fer **shutdown** i altres tasques especials amb permisos de superusuari.

L'username corresponent serà **asosh**, i el password que decidiu. Quan algú faci un login en aquest compte s'executarà l'script asosh que hauríeu de tenir instal·lat de la pràctica anterior d'aplicacions.

[Crear el superusuari asosh:](#)

sudo adduser asosh

```
root@marionaF (Wed Nov 22):<~/Documents># sudo adduser asosh
Adding user `asosh' ...
Adding new group `asosh' (1004) ...
Adding new user `asosh' (1004) with group `asosh (1004)' ...
Creating home directory `/home/asosh' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for asosh
Enter the new value, or press ENTER for the default
    Full Name []: User asosh
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
Adding new user `asosh' to supplemental / extra groups `users' ...
Adding user `asosh' to group `users' ... _
```

[La contrasenya serà igual a l'usuari creat asosh: asosh](#)

[Anar donant a ENTER per no especificar cap dada de telèfons o altres](#)

[Entrar en el fitxer /etc/passwd i veure si s'ha creat correctament l'usuari:](#)

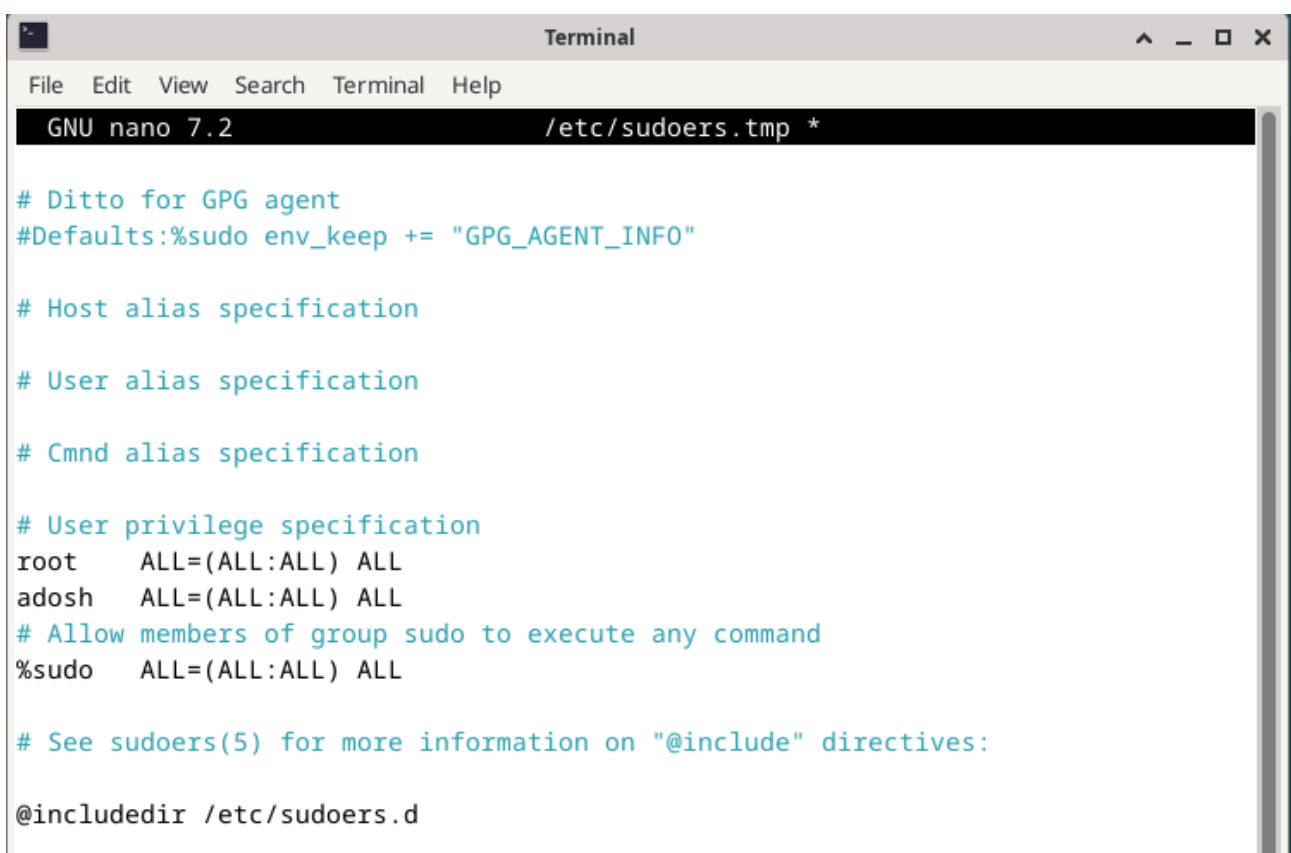

```
usuari1:x:1001:1001:Usuari1:/home/usuari1:/bin/bash
usuari2:x:1002:1002:Usuari2:/home/usuari2:/bin/bash
prova:x:1003:1003:prova,,,:/home/prova:/bin/bash
asosh:x:1004:1004:User asosh,,,:/home/asosh:/bin/bash
```

Donar privilegis de superusuari, entrar al `/etc/sudoers.tmp` o fent servir

sudo visudo

Crear una nova línia de codi per l'usuari asosh especificant que té privilegis de superusuari

asosh ALL=(ALL:ALL) ALL



```
Terminal
File Edit View Search Terminal Help
GNU nano 7.2 /etc/sudoers.tmp *

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
adosh   ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
```

Per raons de seguretat cal que us assegureu que quan s'entra amb aquest compte no s'executa cap shell script. Quins permisos posaríeu a aquesta aplicació perquè no pugui ser executat per cap usuari directament?

Dins del usuari asosh s'haurà de guardar el fitxer asosh.sh

cd /home/asosh

```
root@marionaF (Wed Nov 22):~/home/asosh# ls
asosh.sh
```

Denegar els permisos del fitxer asosh.sh fent que només l'usuari ROOT podrà llegir, escriure i executar el fitxer i cap altre usuari ho podrà fer

chmod 700 /home/asosh/asosh.sh

chown root:root /home/asosh/asosh.sh

```
root@marionaF (Wed Nov 22):</home/asosh># chmod 700 /home/asosh/asosh.sh
```

```
root@marionaF (Wed Nov 22):</home/asosh># chown root:root /home/asosh/asosh.sh
```

Com queda finalment l'entrada de la base de dades d'usuaris per a l'usuari **asosh**?

El fitxer /etc/passwd hauria de ser canviat a:

asosh:x:1001:1001:Usuari asosh:/home/asosh:/home/asosh/asosh.sh

```
usuari1:x:1001:1001:Usuari1:/home/usuari1:/bin/bash
usuari2:x:1002:1002:Usuari2:/home/usuari2:/bin/bash
prova:x:1003:1003:prova,,,:/home/prova:/bin/bash
asosh:x:1004:1004:User asosh,,,:/home/asosh:/home/asosh/asosh.sh
```

En l'inici del login de user asosh, es voldrà executar el fitxer asosh.sh però no es podrà executar ja que no té els privilegis correctes.

8. Sudo i control d'execució d'aplicacions

Com el **shutdown** hi ha altres comandes d'administració que només poden ser executades per el superusuari. És una mala pràctica de seguretat utilitzar el compte del superusuari per executar aquestes comandes. Per resoldre això es pot utilitzar la comanda **sudo**. *Sudo* permet executar una comanda a un usuari autoritzat com superusuari o un altre usuari. L'especificació de quines aplicacions pot executar un determinat usuari es defineix al fitxer /etc/sudoers. Aquest fitxer es pot editar de forma segura fent servir la comanda **visudo**.

Feu els canvis necessaris perquè els membres del grup admin puguin executar qualsevol comanda amb privilegis de superusuari.

Entrem a la carpeta /etc amb superusuari. I dintre d'aquesta utilitzem la comanda "visudo sudoers".

```
aso@aso-client: ~  
File Edit View Search Terminal Help  
aso@AdrianG Tue Nov 21:~$ su  
Password:  
root@AdrianG (Tue Nov 21):</home/homeB/aso># cd /etc  
root@AdrianG (Tue Nov 21):</etc># visudo sudoers  
visudo: sudoers.tmp unchanged  
root@AdrianG (Tue Nov 21):</etc># visudo sudoers
```

Sens obrira un fitxer, i dintre d'aquest afegirem la següent línia de codi

%admin ALL=(ALL) ALL

```
# Cmnd alias specification  
  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
  
# Allow members of group sudo to execute any command  
%sudo    ALL=(ALL:ALL) ALL  
  
#Permitim als admins executar qualsevol comanda  
%admin ALL=(ALL:ALL) ALL  
# See sudoers(5) for more information on "@include" directives:  
  
@includedir /etc/sudoers.d  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Feu els canvis necessaris perquè els usuaris PO puguin executar l'script per esborrar els usuaris que heu creat abans i tots els binaris que siguin al directori /usr/local/PO/bin.

Creem la carpeta /usr/local/PO/bin

```
root@AdrianG (Tue Nov 21):</etc># mkdir -p /usr/local/PO/bin  
root@AdrianG (Tue Nov 21):</etc>#
```

Fem que el grup product owners sigui propietari de la carpeta amb "chgrp -R productowners /usr/local/PO"

Seguidament li donem permisos per executar binaris amb la comanda "chmod 750 /usr/local/PO".

```

root@AdrianG (Tue Nov 21):</etc># chgrp -R po /usr/local/PO
chgrp: invalid group: 'po'
root@AdrianG (Tue Nov 21):</etc># chgrp -R productowners /usr/local/PO
root@AdrianG (Tue Nov 21):</etc># chmod 750 /usr/local/PO/
root@AdrianG (Tue Nov 21):</etc># █

```

Fiquem l'script dintre de la carpeta /PO/bin

```

aso@AdrianG: ~/Downloads (on AdrianG)
File Edit View Search Terminal Help
aso@AdrianG Tue Nov 21:~/Downloads$ su
Password:
root@AdrianG (Tue Nov 21):</home/homeB/aso/Downloads># ls
borrar_user.sh  calc_2.12.7.2-4.debian.tar.xz  calc_2.12.7.2.orig.tar.bz2
calc-2.12.7.2  calc_2.12.7.2-4.dsc
root@AdrianG (Tue Nov 21):</home/homeB/aso/Downloads># pwd
/home/homeB/aso/Downloads
root@AdrianG (Tue Nov 21):</home/homeB/aso/Downloads># cp borrar_user.sh /usr/local/PO/bin/borrar_user.sh
root@AdrianG (Tue Nov 21):</home/homeB/aso/Downloads># cd /usr/local/PO/bin/
root@AdrianG (Tue Nov 21):</usr/local/PO/bin># ls
borrar_user.sh
root@AdrianG (Tue Nov 21):</usr/local/PO/bin>#

```

Afegim dos linies més a la carpeta sudoers utilitzant "visudo /etc/sudoers"

Cmdn_Alias RUTA_PO = /usr/local/PO/bin/borrar_user.ph

%productowners ALL=RUTA_PO

```

# Cmnd alias specification
Cmnd_Alias RUTA_PO = /usr/local/PO/bin/borrar_user.ph
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

#Permitim als admins executar qualsevol comanda
%admin  ALL=(ALL:ALL) ALL

#Permitim als PO utilitzar el que estigui dintre de bin
%productowners ALL=RUTA_PO

```

Comproveu que això funciona executant la comanda **vipw**.

Abans de fer servir l'script si utilitzem vipw l'usuari surt activat

```
Applications ▾ po1@AdrianG: ~ po1 - Thunar
po1@AdrianG: ~
File Edit View Search Terminal Help
GNU nano 7.2 /etc/passwd.edit
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nolog>
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sb>
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/>
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
aso:x:1000:1000:,,,:/home/homeB/aso:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:./usr/sbin/nologin
avahi:x:105:112:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
pulse:x:106:113:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
saned:x:107:116:./var/lib/saned:/usr/sbin/nologin
lightdm:x:108:117:Light Display Manager:/var/lib/lightdm:/bin/false
polkitd:x:998:998:polkit:/nonexistent:/usr/sbin/nologin
rtkit:x:109:118:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:110:119:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/n>
geoclue:x:111:121:./var/lib/geoclue:/usr/sbin/nologin
po1:x:1001:1001:,,,:/home/po1:/bin/bash
prueba:x:1003:1003:,,,:/home/prueba:/bin/bash
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Utilitzem l'script (en aquest cas acababa de borrar l'usuari però podem veure com funciona).

```
po1@AdrianG:/usr/local/P0/bin$ ./borrar_user.sh prueba
You may not change the shell for 'prueba'.
Se ha bloqueado el acceso al usuario prueba
No existe el directorio /home/prueba
eliminados todos los ficheros de prueba
po1@AdrianG:/usr/local/P0/bin$
```

Despres d'utilitzar l'script

```
pol@AdrianG: ~  
File Edit View Search Terminal Help  
GNU nano 7.2 /etc/passwd.edit  
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin  
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin  
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin  
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin  
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin  
aso:x:1000:1000:,,,:/home/homeB/aso:/bin/bash  
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin  
avahi:x:105:112:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin  
pulse:x:106:113:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin  
saned:x:107:116:/:/var/lib/saned:/usr/sbin/nologin  
lightdm:x:108:117:Light Display Manager:/var/lib/lightdm:/bin/false  
polkitd:x:998:998:polkit:/nonexistent:/usr/sbin/nologin  
rtkit:x:109:118:RealtimeKit,,,:/proc:/usr/sbin/nologin  
colord:x:110:119:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin  
geoclue:x:111:121:/:/var/lib/geoclue:/usr/sbin/nologin  
pol:x:1001:1001:,,,:/home/pol:/bin/bash  
prueba:x:1003:1003:,,,:/home/prueba:/usr/local/lib/no-login/desactiva.sh  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Quins canvis heu fet al fitxer /etc/sudoers per activar els controls anteriors?

```
# Cmnd alias specification  
Cmnd_Alias RUTA_PO = /usr/local/PO/bin/borrar_user.ph  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
  
# Allow members of group sudo to execute any command  
%sudo    ALL=(ALL:ALL) ALL  
  
#Permitim als admins executar qualsevol comanda  
%admin ALL=(ALL:ALL) ALL  
  
#Permitim als PO utilitzar el que estigui dintre de bin  
%productowners ALL=RUTA_PO
```

Finalment desactiveu el compte del root de tal forma no es pugui fer *login* com superusuari. Les comandes d'administració es podran fer només des dels comptes del grup admin o fent ús de l'usuari asosh. Assegureu-vos que podeu fer comandes des d'un usuari administrador abans de desactivar-ho.

Entrem al root i utilitzem la comanda "passwd -lock root"

```
po1@AdrianG:/usr/local/P0/bin$ su
Password:
root@AdrianG (Tue Nov 21):/usr/local/P0/bin# passwd --lock root
passwd: password changed.
root@AdrianG (Tue Nov 21):/usr/local/P0/bin#
```

Al sortir del root i intentar entrar una altra vegada veurem que no ens deixa

```
root@AdrianG (Tue Nov 21):/usr/local/P0/bin# exit
exit
po1@AdrianG:/usr/local/P0/bin$ su
Password:
su: Authentication failure
po1@AdrianG:/usr/local/P0/bin$
```

Si intentem entrar amb un usuari que no es admin veurem que ens surt error