

**Índex**

[1. Introducció 3](#_heading=h.gjdgxs)

[1.1. Objectius 3](#_heading=h.30j0zll)

[2. Profile i entorn d'usuari 3](#_heading=h.1fob9te)

[3. Creació manual d'usuaris 4](#_heading=h.3znysh7)

[4. Creació automàtica d'usuaris 10](#_heading=h.kxh7ojfjmwl6)

[5. Connexió remota d'usuaris 15](#_heading=h.3dy6vkm)

[6. Eliminació i desactivació d'usuaris 22](#_heading=h.1t3h5sf)

[7. Usuari especial asosh 28](#_heading=h.4d34og8)

[8. Sudo i control d'execució d'aplicacions 30](#_heading=h.2s8eyo1)

# 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 é*s 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 el usuaris.

## Objectius

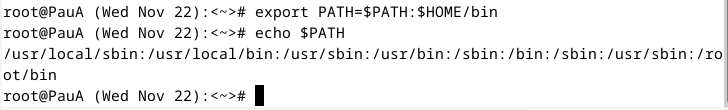
Gestionar els usuaris del sistema: realitzar l'alta i baixa d'usuaris i modificar les propietats dels comptes d'usuari.

# 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**).



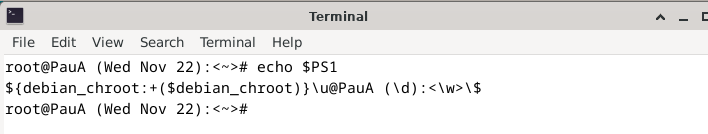
Volem que el *prompt* sigui el *username* seguit de la data actual i finalment “> “ (per exemple, el de l'usuari xavim seria "xavim (Tue April 10) >” )





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.



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

# 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 introduirem 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:



(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 |
| *Username* | usuari1 | usuari2 |
| Directori home | /home/usuari1 | /home/usuari2 |
| *Shell* | /bin/bash | /bin/bash |
| Grups | admin | admin |

Quina és la diferencia en usar **vipw** o editar directament el fitxer de passwd amb vi? (pista: obriu dos vipw en sessions diferents)

Al usar **vipw**, 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 **vipw** 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:



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:



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.



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:



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.

1. 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: 

Al acabar reactivarem el compte amb: 

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 el fitxers que estiguin dintre del directori.



**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:***



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)**.

1. **La segona comanda** gestiona els permisos de l’arxiu .shh 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?*



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?



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’advertència 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.

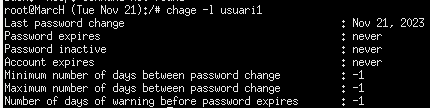
Per exemple:

**sudo chage -d YYYY-MM-DD nom\_usuari**: estableix la data de l’ultim 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:



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:

# 

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



L’estructura és **chsh -s /ruta/del/shell nom\_usuari**

# 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

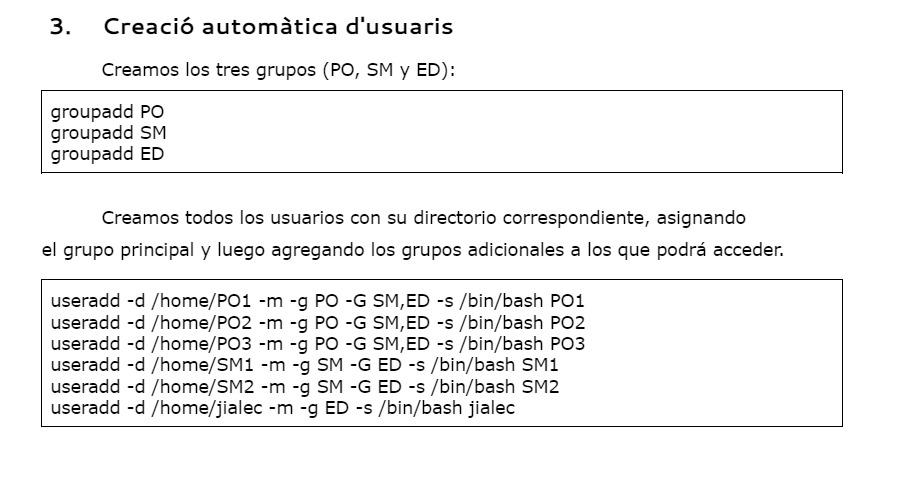


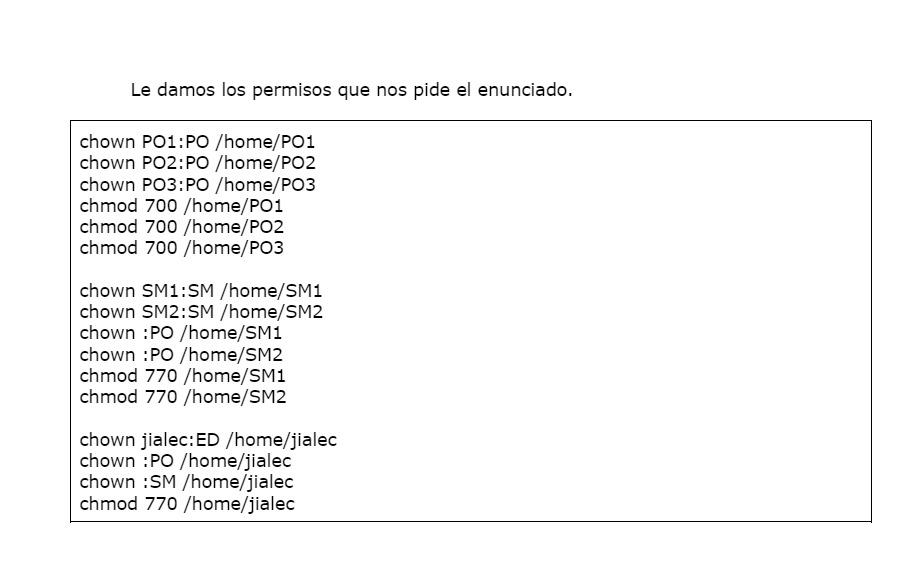
Els posarem a aquest directori per a tenir-los endreçats i agrupats a la ubicació estàndard. A part, afegir els usuaris dins el directori de **/home** ens proporciona:

* **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.

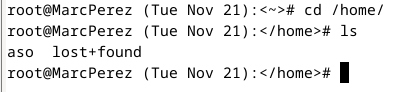
Els permisos de cadascun d'aquests grups d'usuaris (POs, SMs i ED) venen definits de la següent forma:

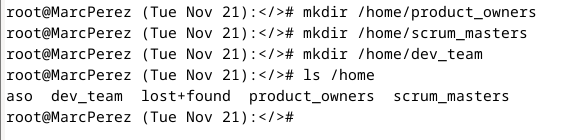






Mostra tot el procés de creació indicant pas a pas que s'ha fet





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

Comanda: **sudo groupadd nombreGrupo**

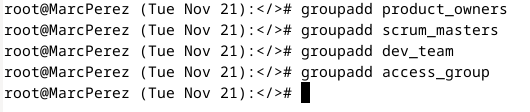
sudo groupadd product\_owners

sudo groupadd scrum\_masters

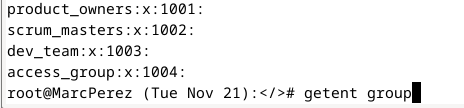
sudo groupadd dev\_team

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.



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



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

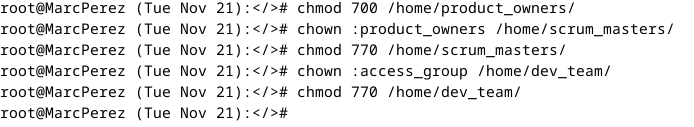
sudo chown :product\_owners /home/scrum\_masters

sudo chmod 770 /home/scrum\_masters

sudo chown :access\_group /home/dev\_team

sudo chmod 770 /home/dev\_team



****

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/PO1 -G product\_owners,access\_group PO1

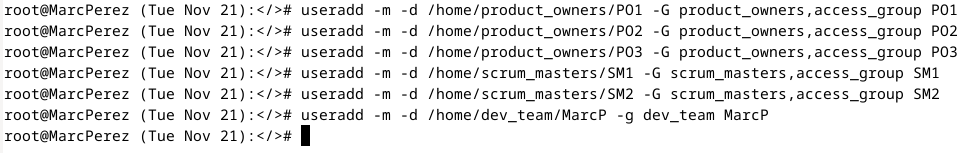
sudo useradd -m -d /home/product\_owners/PO2 -G product\_owners,access\_group PO2

sudo useradd -m -d /home/product\_owners/PO3 -G product\_owners,access\_group PO3

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

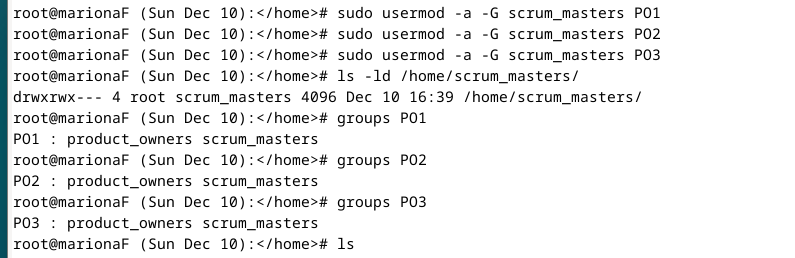
sudo useradd -m -d /home/dev\_team/MarcP -g dev\_team MarcP

****

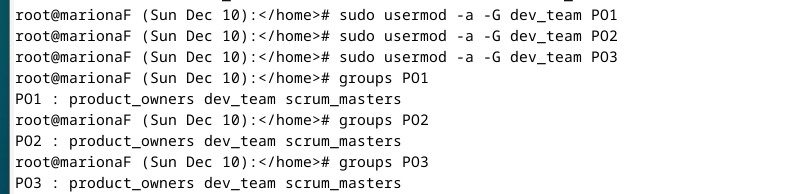
**POSSIBLE CANVI: POS ACONSEGUIEXIN ENTRAR A SCRUM\_MASTERS**

**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)**

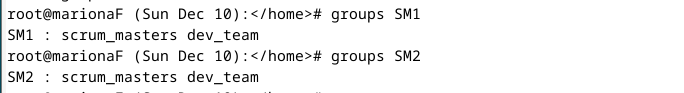


**PO DINS DE DEV TEAM:gg**

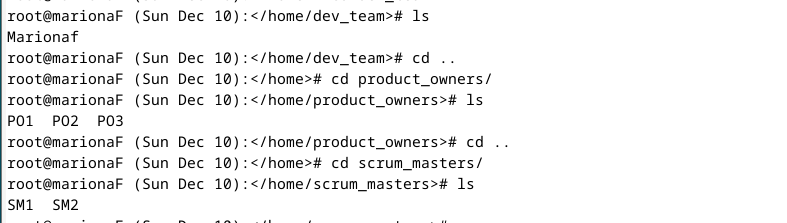
****

**SM A DINS DE DEV TEAM**

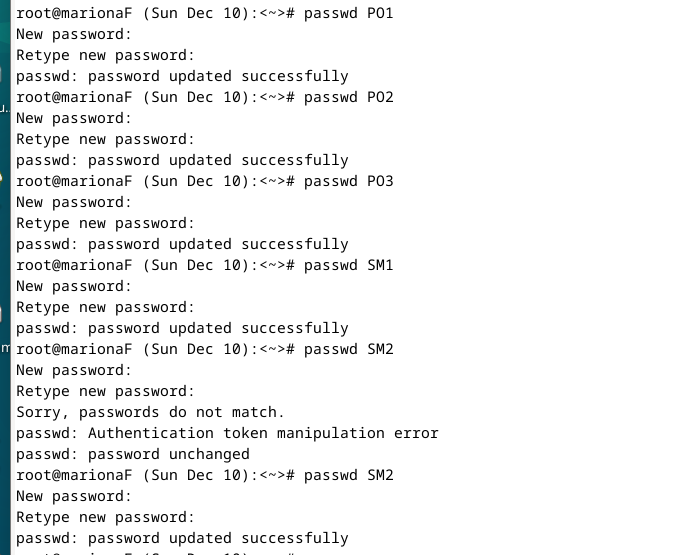




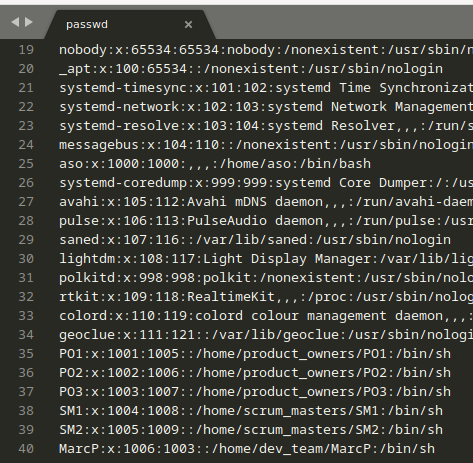
**ELS USUARIS HAN DE QUEDAR AIXI:**

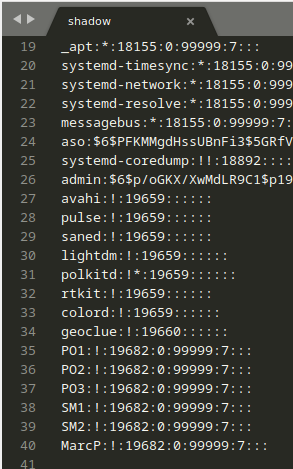
****

**POSARLIS CONTRASENYES! canviara el /etc/shadow**



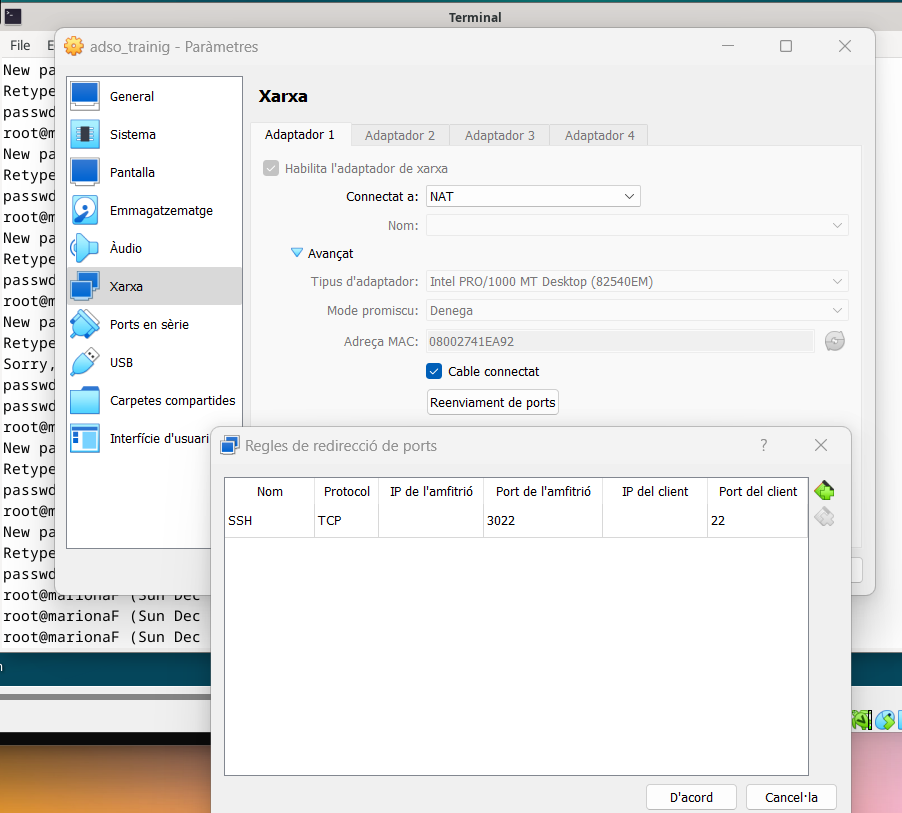
Podemos ponernos comprovar que se han añadido correctamente mirando en **/etc/passwd** o bien en **/etc/shadow**

****



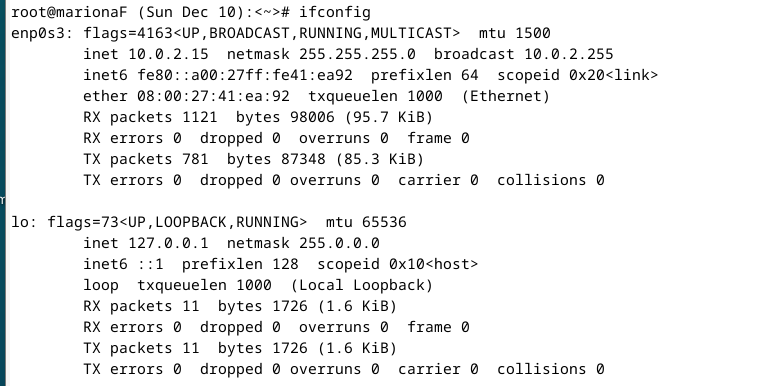
**COMPROVACIÓ!!! (microhakaton4)**

**ABANS CANVIAR per ssh (descarregar openssh)**

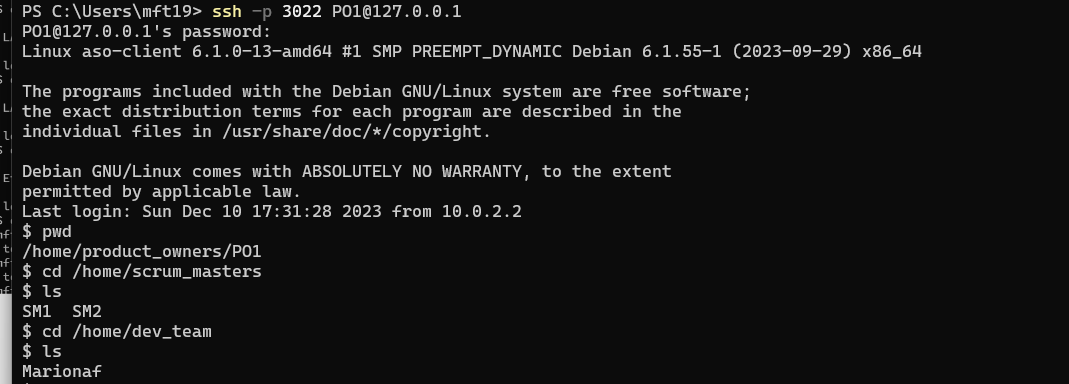
****

**Des de ternimal de WINDOWS posar el ssh -p 3022 USER\_A\_PROVAR@ip de la maquina virtual**

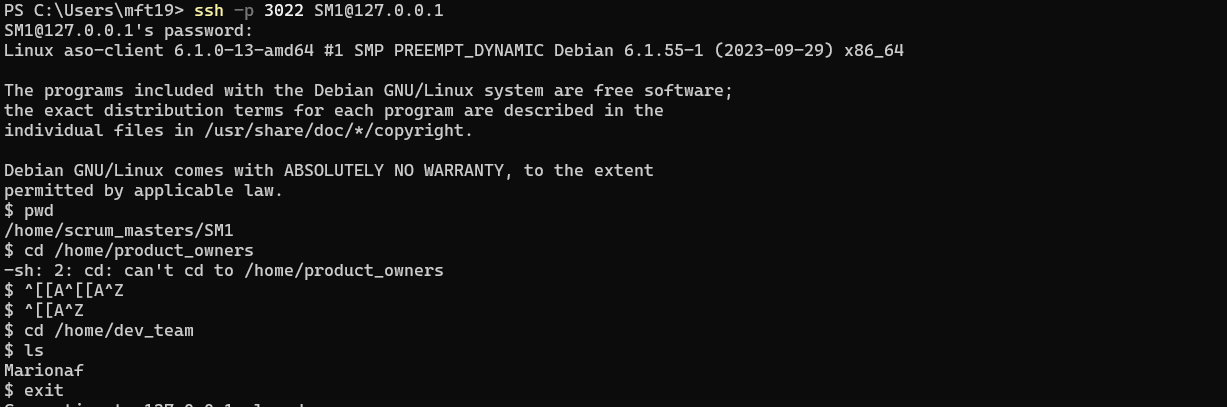
**ip VM → ifconfig**

****

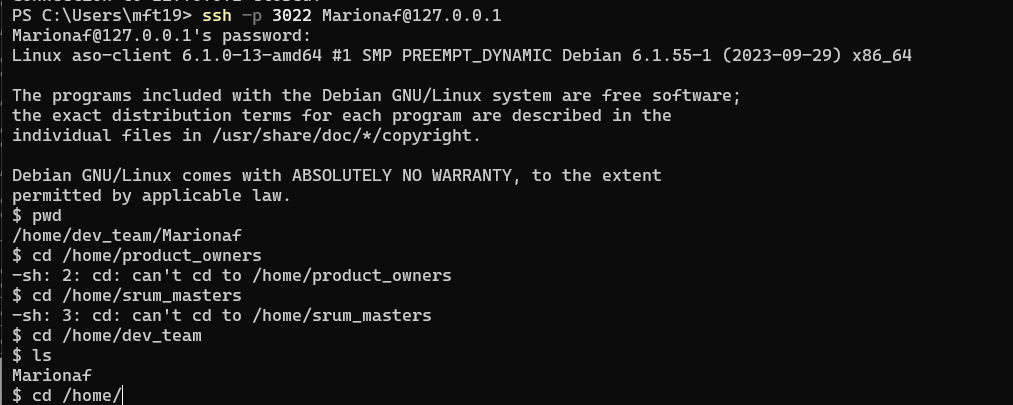
**PO a altres carpetas**

****

**SM a altres carpetas**



**ED a altes carpetas**

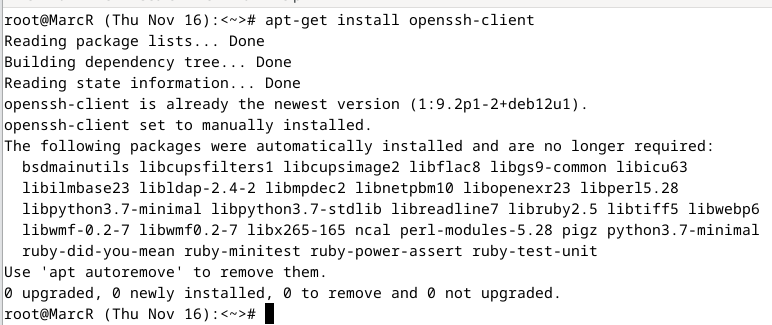
****

# Connexió remota d'usuaris

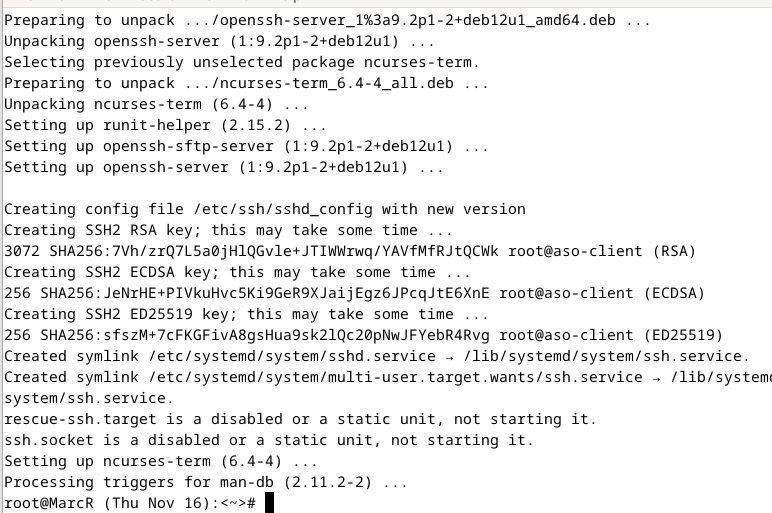
Els usuaris de la nostra màquina han de tenir l'opció de poder connectar-se en remot de una forma segura.

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:



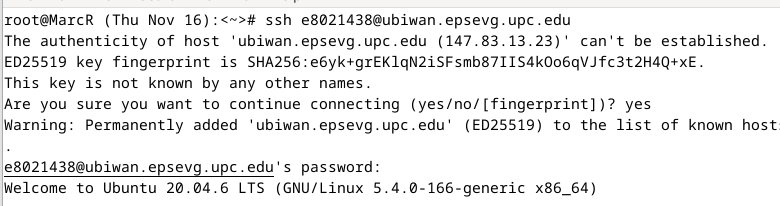
Com podem veure en el meu cas ja el tenìa instal·lat, aleshores procedirem a instal·lar el server amb la comanda apt-get install openssh-server:

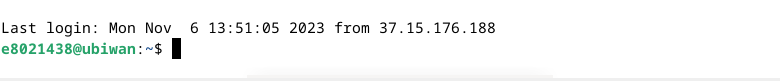


Comproveu que us podeu connectar remotament a un altra màquina.

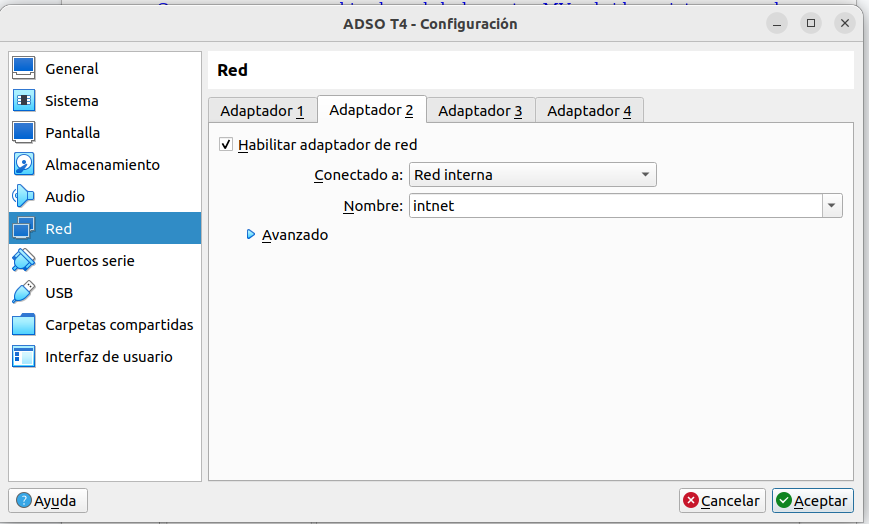
Una vegada que ja tenim instal·lat els openssh comprovem que ens podem connectar i que es poden connectar a nosaltres:

Connexió desde màquina virtual a un altre màquina:



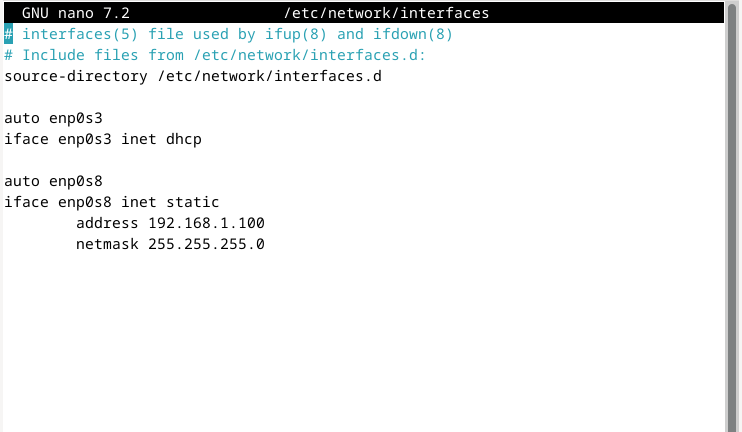


Començarem per afegir la red a la nostre MV en mode bridge o interna, en el meu cas la faré interna en les dues màquines:

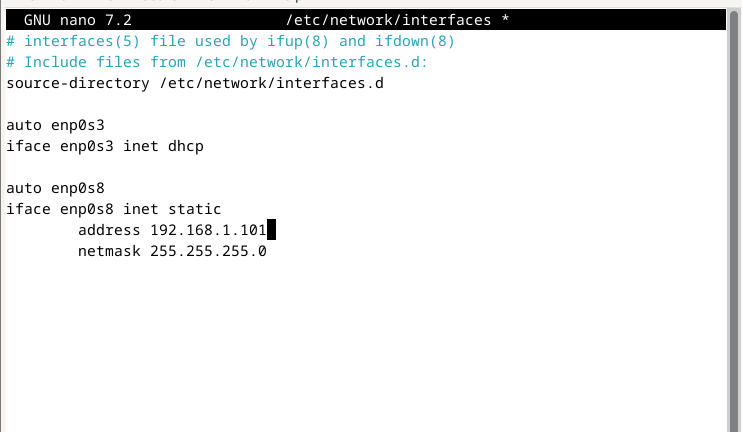


A continuació entrarem a les nostres màquines i ens anirem a l’arxiu /etc/network/interfaces per a afegir la nova interfície de la següent manera:

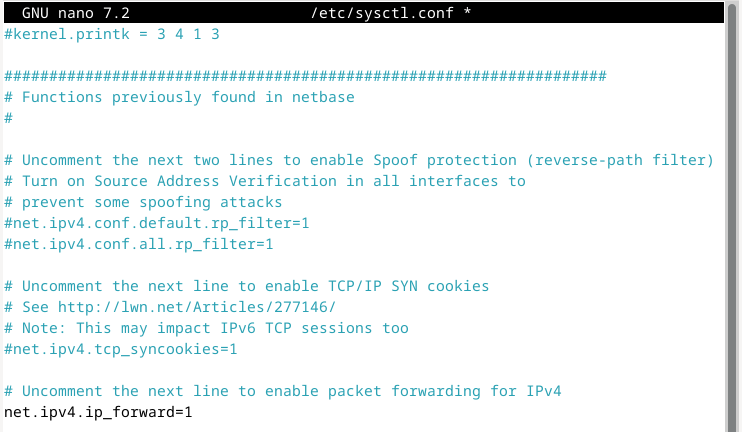
Maquina 1:



Maquina 2:



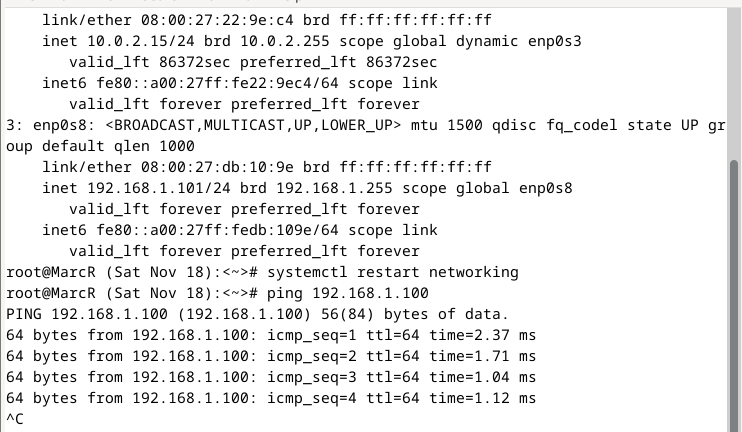
Habilitem l'enrutament descomentant la línia de l’arxiu /etc/sysctl.conf com en la imatge:



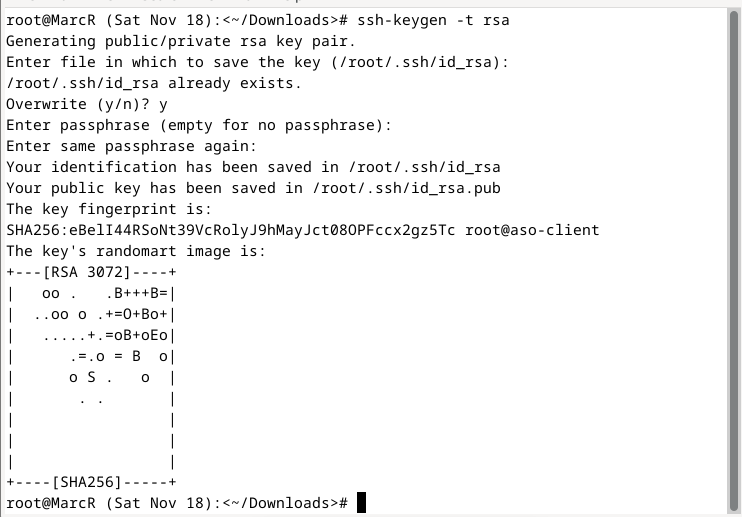
enp0s3 es la nostra NAT, mentre que enp0s8 es la nostre xarxa interna.

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

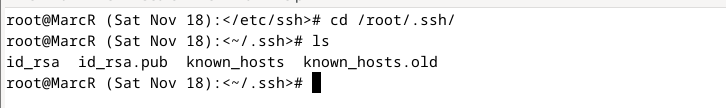
Ping desde la màquina 2(192.168.1.101) a la màquina 1(192.168.1.100):



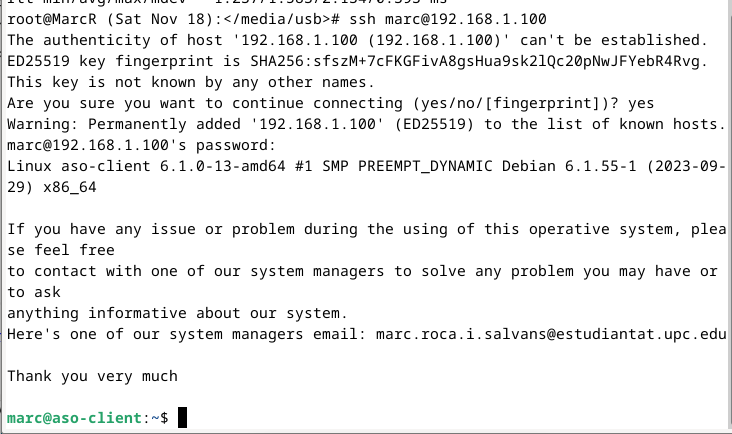
Una vegada modificada la red procedirem a crear unes claus desde la màquina la qual ens volem connectar amb la comanda ssh-keygen -t rsa, aixó ens crearà una clau privada i una pública:



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



A continuació copiarem la clau id\_rsa.pub de la nostre 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:



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

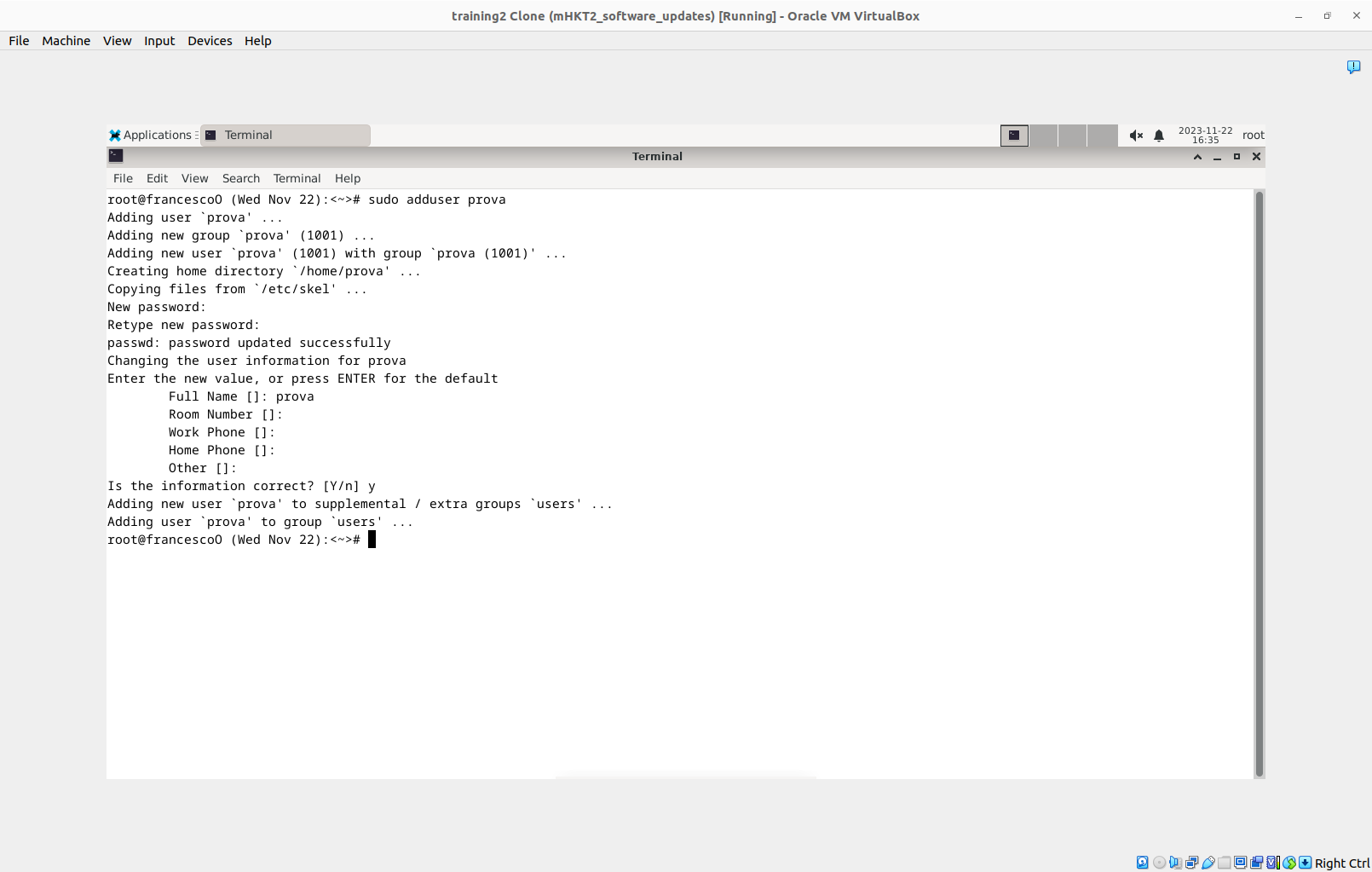
# Eliminació i desactivació d'usuaris

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**



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

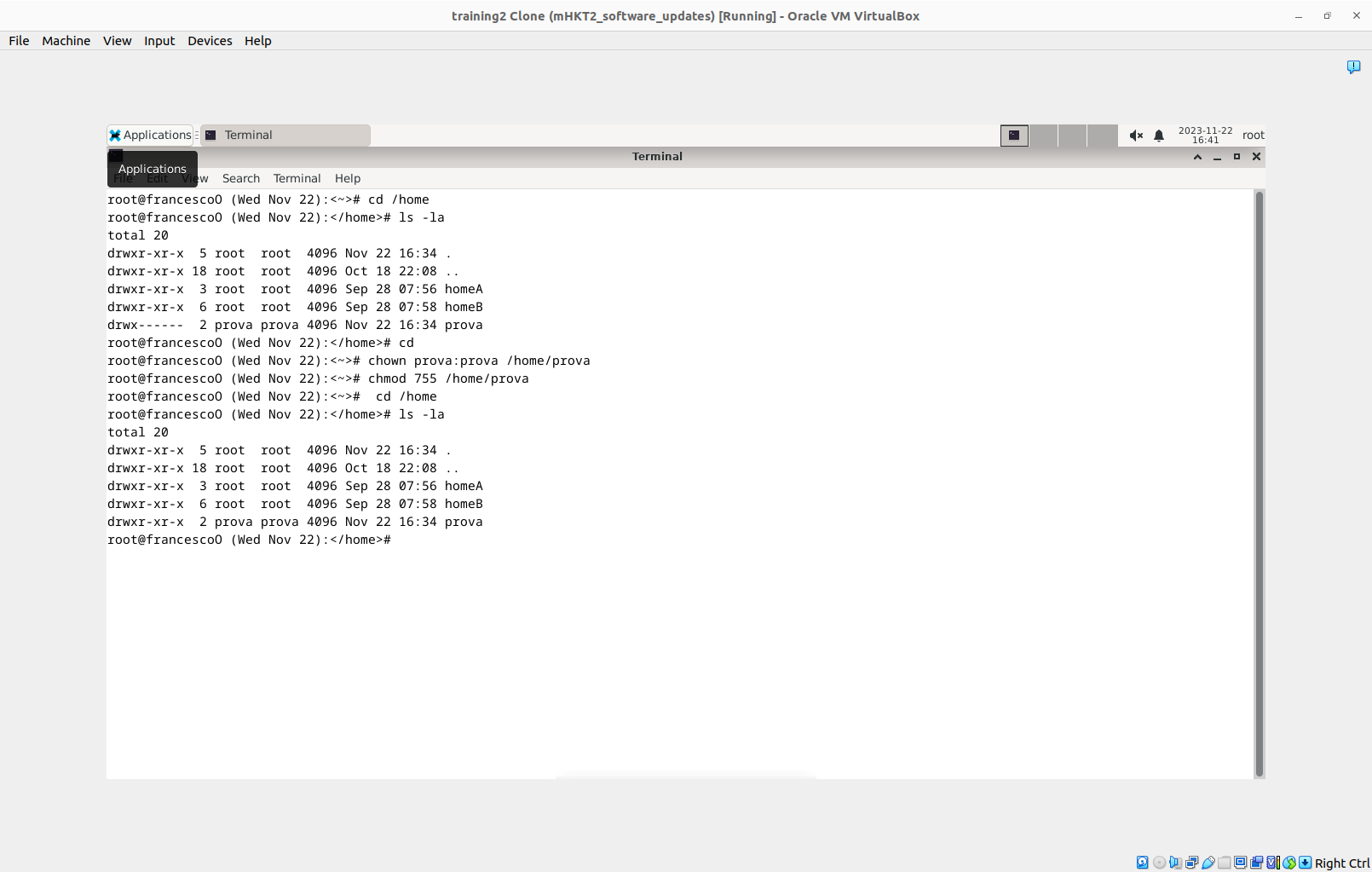
Si no es crea el directori directament, el podem crear manualment amb les següents instruccions:

**sudo mkdir /home/prova**

Donar privilegis d’usuari als directori creat:

**sudo chown user:user /home/user**

**sudo chmod 755 /home/prova**

****

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 dóna 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:



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 interpertar-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

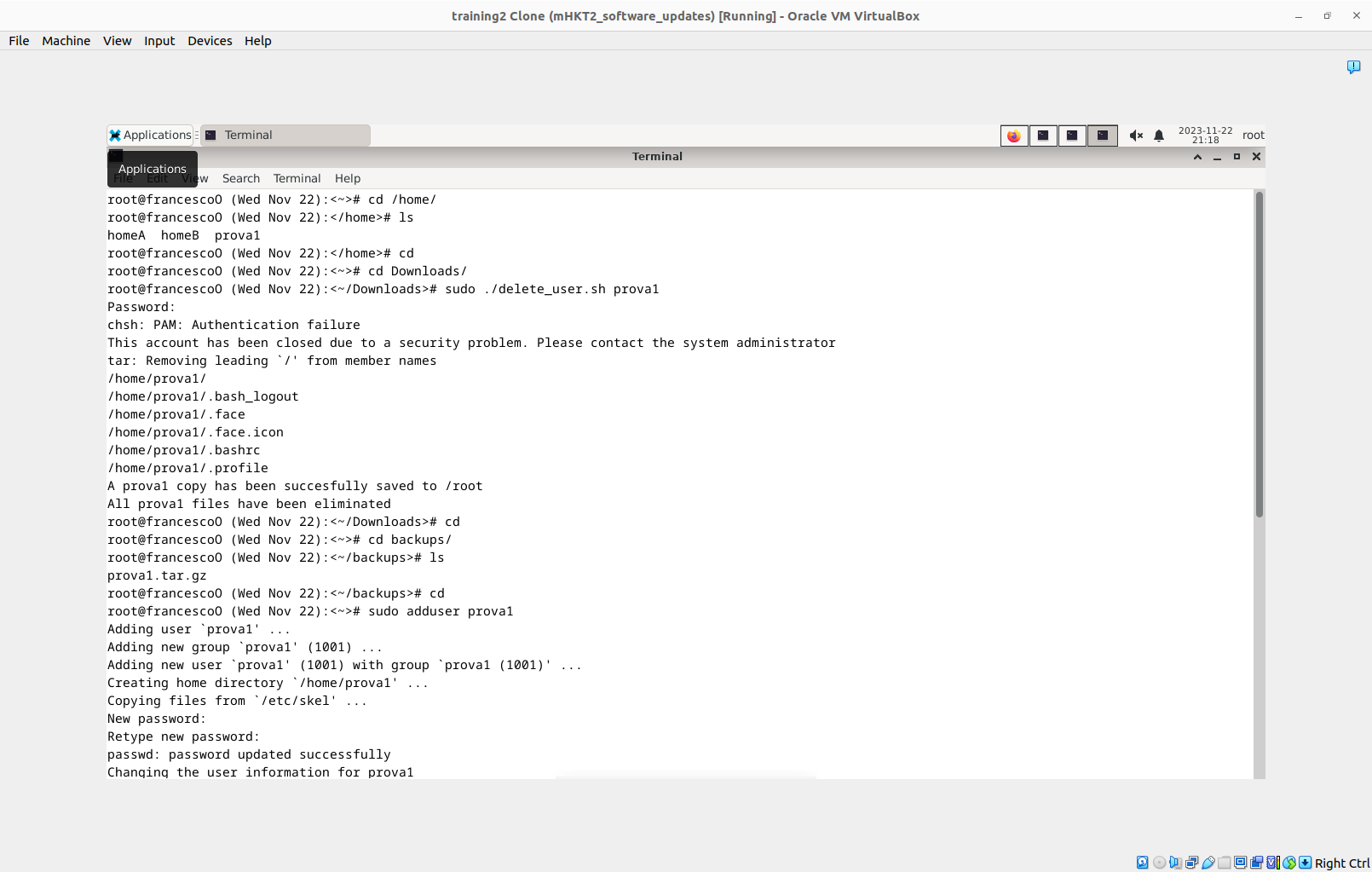
find / -user $1 -exec rm -r "{}" \; 2> /dev/null

echo "All $1 files have been eliminated"

userdel $1



Comprova que s'ha fet correctament



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.

# **Usuari especial asosh**

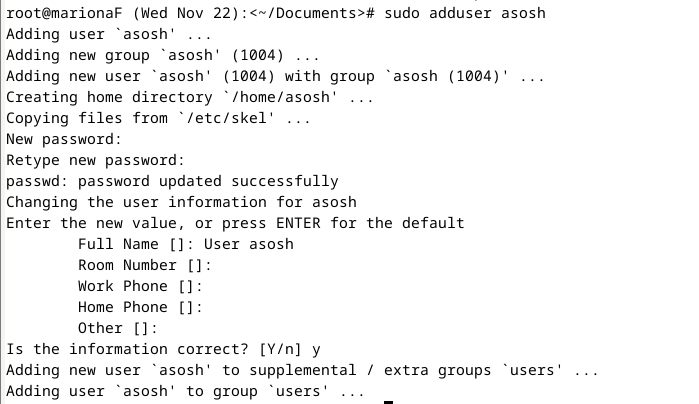
A Unix hi ha comandes com el **shutdown** per apagar la màquina que només pot executar l'usuari root. En moltes ocasions pot ser interessant que algun altre usuari pugui apagar també la màquina però sense que tingui accés als privilegis de root.

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:

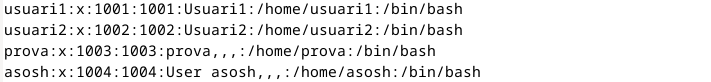
**sudo adduser asosh**

****

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:

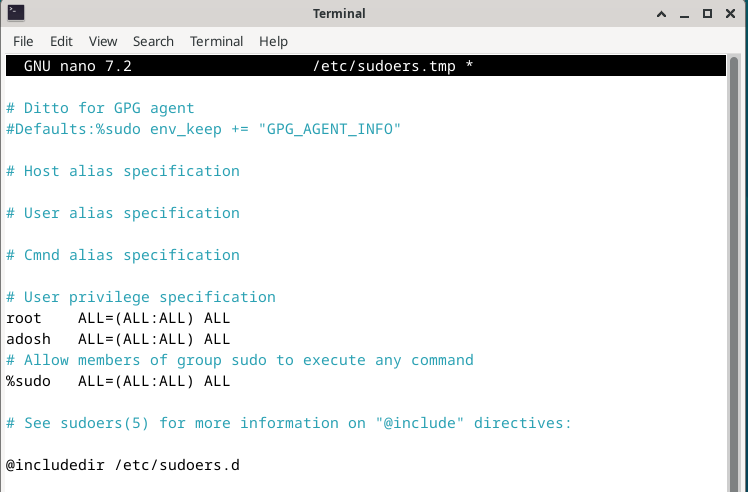


Donar privilegis de superusuari, entrar al /etc/sudoers.tmp o fent servir

**sudo visudo**

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

**asosh ALL=(ALL:ALL) ALL**

****

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’haura de guardar el fitxer asosh.sh

**cd /home/asosh**

****

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**

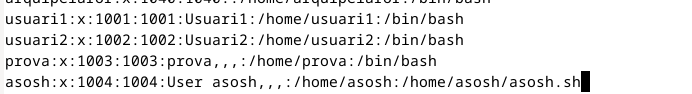


****

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**

****

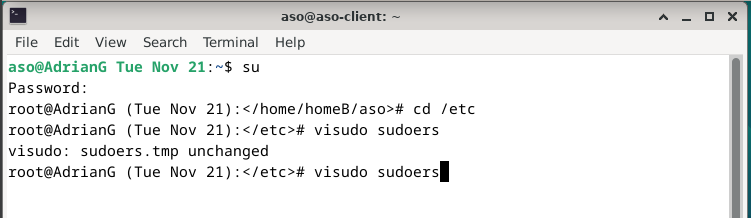
En l’inici del login de user asosh, es voldrà executar el fitxer asosh.sh però no es podrà executar ja que no té els privilegis correctes.

# **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.**

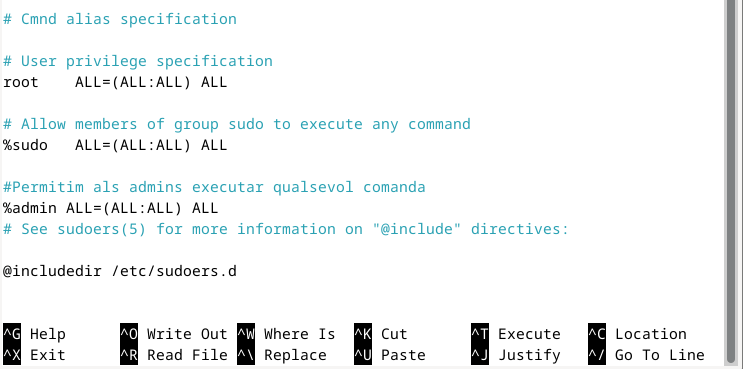
Feu els canvis necessaris perquè els membres del grup admin puguin executar qualsevol comanda amb privilegis de superusuari.

Entrem a la carpeta /etc amb superusuari. I dintre d’aquesta utilitzem la comanda “visudo sudoers”.



Sens obrira un fitxer, i ditnre d’aquest afegirem la següent linia de codi

%admin ALL=(ALL) ALL



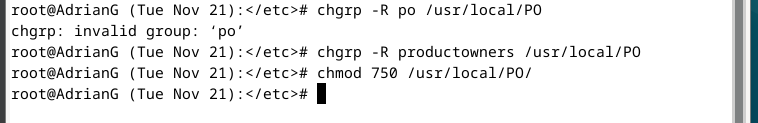
Feu els canvis necessaris perquè els usuaris PO puguin executar l'script per esborrar els usuaris que heu creat abans i tots els binaris que siguin al directori /usr/local/PO/bin.

Creem la carpeta /usr/local/PO/bin

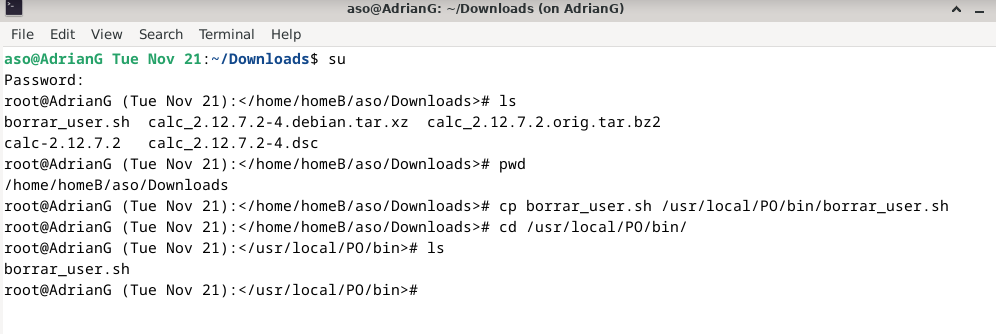


Fem que el grup product owners sigui propietari de la carpeta amb “chgrp -R productowners /usr/local/PO”

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



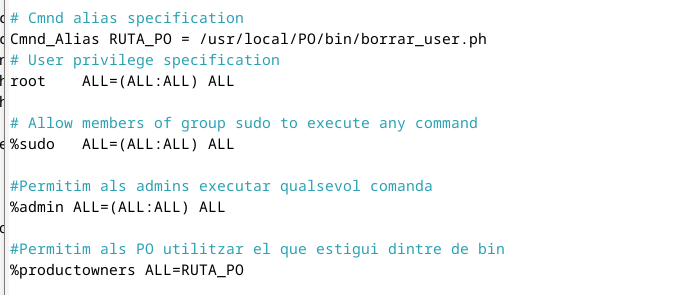
Fiquem l’script dintre de la carpeta /PO/bin



Afegim dos liníes més a la carpeta sudoers utilitzant “visudo /etc/sudoers”

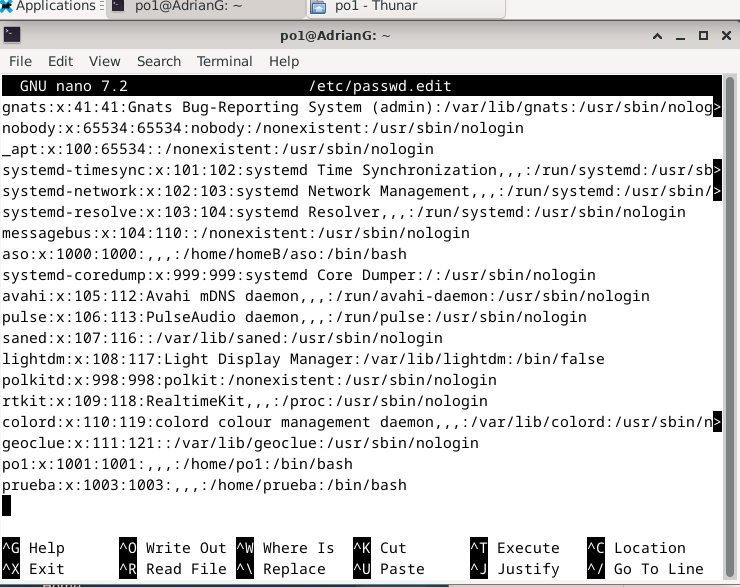
Cmnd\_Alias RUTA\_PO = /usr/local/PO/bin/borrar\_user.ph

%productowners ALL=RUTA\_PO

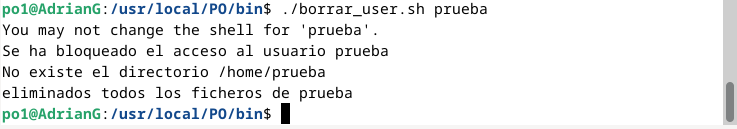


Comproveu que això funciona executant la comanda **vipw.**

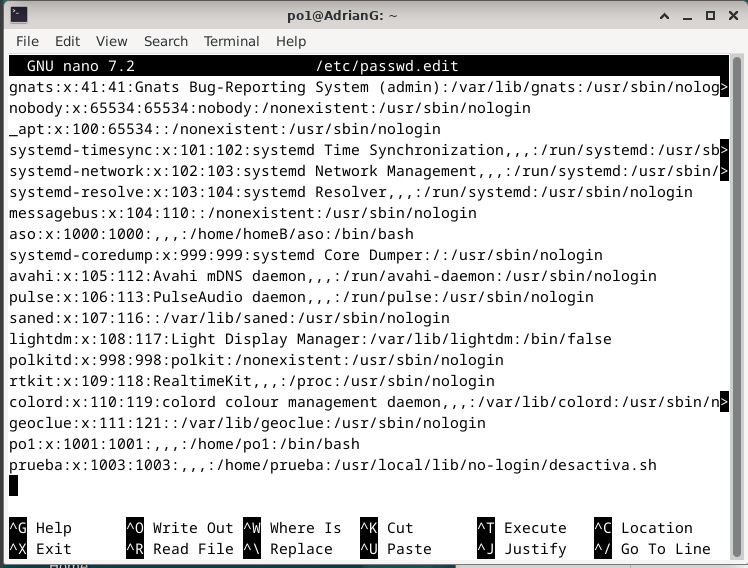
Abans de fer servir l’script si utilitzem vipw l’usuari surt activat

****

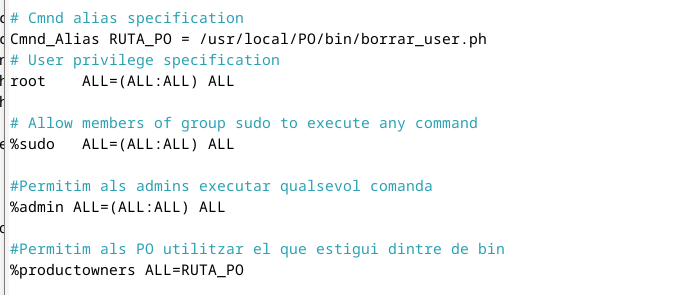
Utilitzem l’script (en aquest cas acababa de borrar l’usuari però podem veure com funciona).



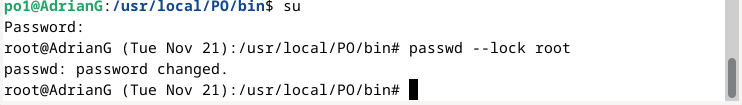
Despres d’utilitzar l’script

****

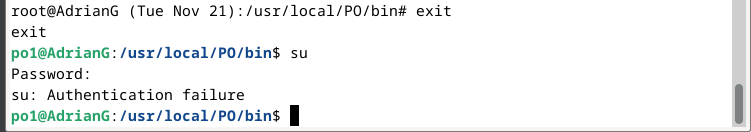
Quins canvis heu fet al fitxer /etc/sudoers per activar els controls anteriors?

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”

****

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

****

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