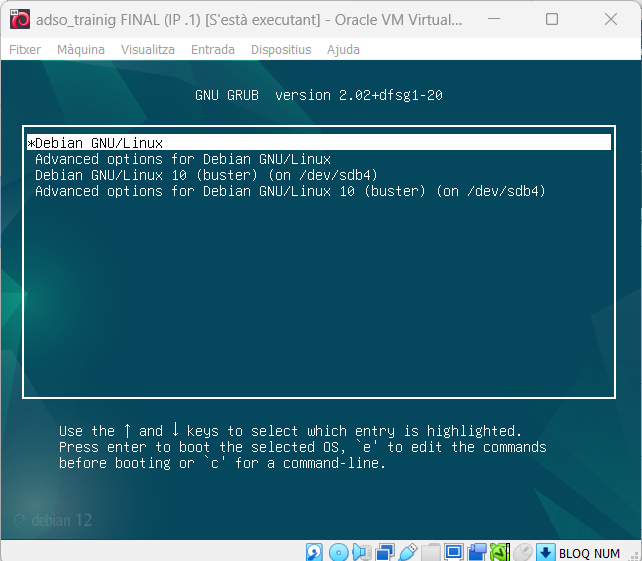


**Mariona Farré Tapias**

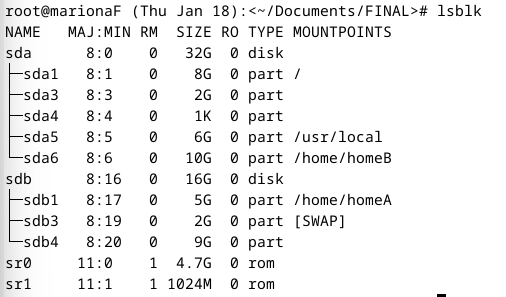
1. (3**,5 punts**) Es vol substituir el sistema operatiu instal·lat al disc virtual (VHD) de la màquina de la mHKT 1 seguint les següents especificacions:
   * + Utilitzarem **la versió més actualitzada** de Arch Linux que podem trobar a <https://archlinux.org/>
     + Els directoris /*home/homeA, /usr/local, /var, /tmp i el swap* hauran d’estar en particions separades
     + *El sistema de fitxer es muntarà a l’engegar el sistema*
     + La maquina es dirà **<el teu\_nom>-mhkt24 (ha d'aparèixer al prompt)**
       - * **Ex.:** 
     + la xarxa es detectarà automàticament i l’assignació d'adreces serà per dhcp. Comprova abans de fer cap acció si ja tens connexió a internet.
     + Es permetrà accés amb usuari root.
     + Es crearà un usuari amb user **aso** i password **aso,** amb homea **/home/homeA** de forma manual (modificant les bases de dades d’usuari)
     + S’ha d'instal·lar grub <https://wiki.archlinux.org/title/GRUB_(Espa%C3%B1ol)>. Si l’aplicació grub no existeix, instal·la-la <https://man.archlinux.org/man/pacman.8> (utilitza l’operació Synchronize packages)
   1. (**0,5 punt**) Indica, el mes detallat possible, els passos (algoritme) que seguiràs per fer la instal·lació i que es compleixin totes les especificacions.

* Descarregar la imatge de Arch Linux més actualitzada
* Posar la imatge dins de la màquina virtual dins des del disc òptic
* Inicialitzar la màquina amb l’Archlinux
* Modificar les particions dins del fstab
* I Mirar com es mouen els directoris dits anteriorment:
  + /home/homeA
  + /usr/local
  + /var
  + /tmp
  + el swap
* Creem l’usuari aso i root
* Inicialitzar la màquina modificada
* Dins d’aquesta cambiar configuració com amb nom donat: Mariona-mhkt24
  1. (**0,5 punts**) Mostra l’estat actual del teu sistema abans de fer la nova instal·lació (particions, usuaris, fstab, SO instal·lats . Indica les comandes que fas servir.

El grub inicial és:

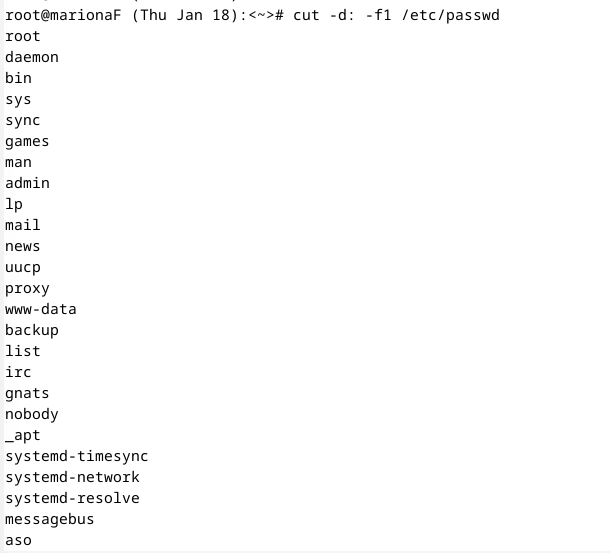


Les particions es poden veure amb la comanda: lsblk

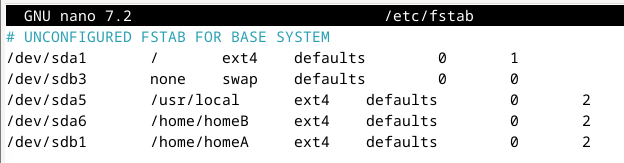


On podem veure el home/homeA i el Swap en l’antic lloc

Els usuaris de dins amb la llista d'usuaris: cut -d: -f1 /etc/passwd

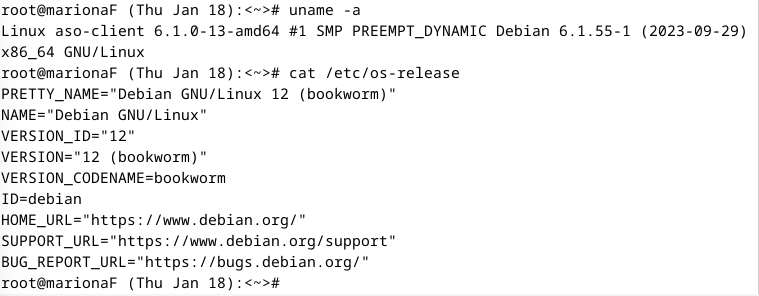


Amb l’arxiu de fstab sent: nano /etc/fstab



El Sistema operatiu anterior per mirar la màquina i la versió: uname -a

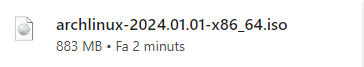
I per una informació més detallada de la versió: cat /etc/os-release



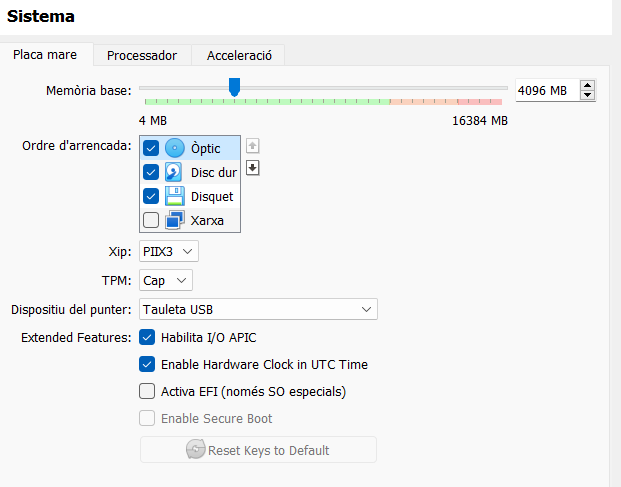
* 1. (**0,5 punt**) Descarrega la imatge de **ArchLinux** i prepara la MV per bootar des d’aquesta imatge. Indica quin es el mode d'instal·lació que has escollit. Mostra la versió a instal·la-lar.

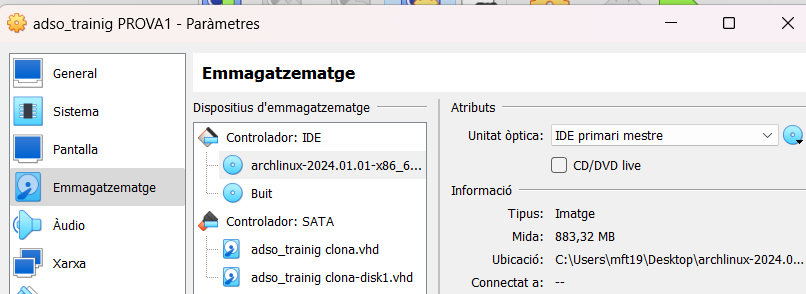
Descarregarse la imatge de Archlinux més actualitzada des de la web oficial:





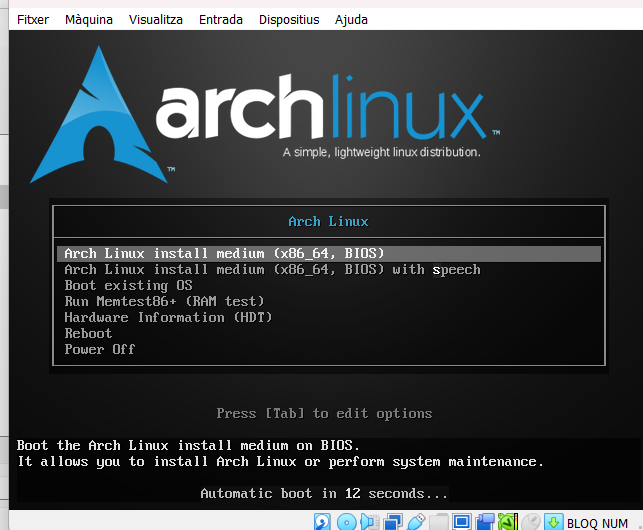
Guardar en l’ordinador i pujar en el disc óptic com a primera opció:



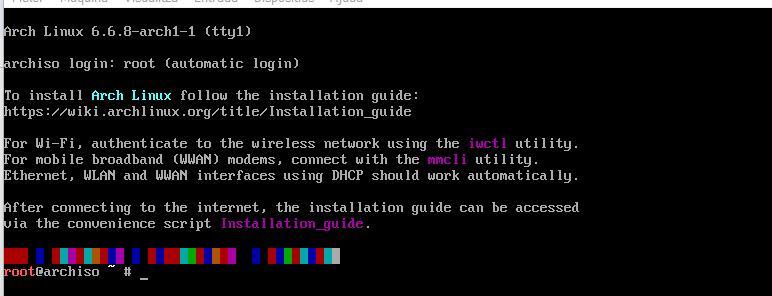


* 1. (1 **punts**) Mostra els passos que vas seguint durant la instal·lació.

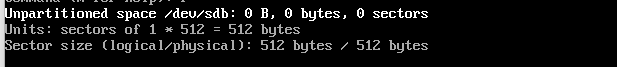
Ara inicialitzar la màquina amb els passos anteriors fets:



Trobarem la pantalla principal:



Podem entrar en el fdisk /dev /sdb per veure la actual partció:



* 1. (**1 punt**) Acaba la instal·lació. Entra al nou sistema amb l’usuari aso i mostra, des de terminal, els resultats (prompt, particions, punts de muntatge i versió del kernel)

1. **(3,5 punts)** Es vol crear un mecanisme per gestionar usuaris dins del sistema de tal forma que un administrador de sistemes **novell** (**sense permisos de root**) s’encarregui d’aquesta gestió. Aquest administrador s’encarregarà de modificar un arxiu de text, anomenat **usuaris.txt**, a on ficarà informació dels nous usuaris a donar d’alta o dels usuaris que s’han de donar de baixa.

El nostre mecanisme utilitzarà, de forma periòdica, aquest fitxer per crear/eliminar usuaris. S’ha de complir les següents condicions:

* + - Els usuaris del sistema estan organitzats en equips de treball i, per tant, el seu grup serà el nom de l’equip. (veure adjunt: «**equips\_usuaris.xls**»)
    - L’username de cada usuari correspon al seu cognom. La repetició de cognoms ha d’estar solucionada
    - Cada equip tindrà un directori propi, penjat del **/home**, amb el nom de l’equip i a dins cada usuari tindrà el seu directori propi
    - Cap usuari pot accedir al directori de qualsevol altre usuari
    - El directori de l’equip s’esborrarà quan no hi hagi cap usuari a l’equip
    - Per tal de compartir informació, dins de cada directori d’equip hi hauran dos directoris afegits:
* el directori «grup» (**/home/nom\_equip/grup**) que serà accessible únicament per a tots els membres de l’equip
* el directori «public» (**/home/nom\_equip/public**) que serà accessible per a tothom

L’actualització d’usuaris es farà automàticament el dilluns de cada setmana a les 6:30h, excepte els mesos de juliol i agost

* 1. (**0,5 punts**) Explica com crearàs aquest mecanisme
* Entendre correctament els permisos de cada usuari, cada grup i els directoris que necessiten cadascun:

Des de NO ROOT

crear/eliminar users des de usuaris.txt (cognom,nom,equip)

Usuaris entrar per EQUIPS (3 paràmetre)

USERNAME: cognom

si cognom igual no crear

EQUIP: directori propi /home/nom\_equip

/home/nom\_equip/grup → NOMÉS MEMBRES EQUIP

/home/nom\_equip/public → per tothom

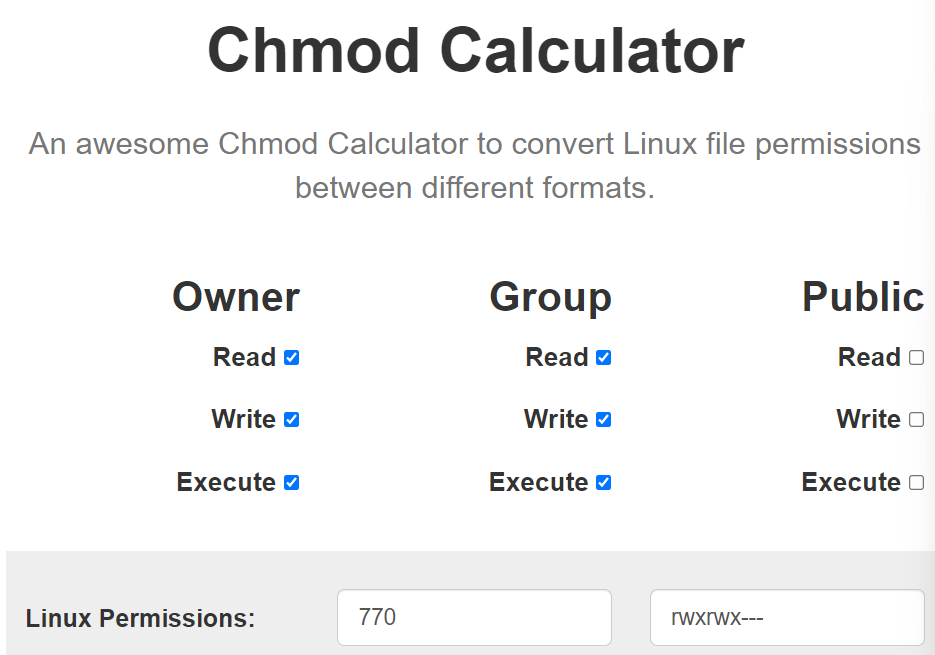
dins cada user amb directori /cognom

usuari accedir qualsevol usuari

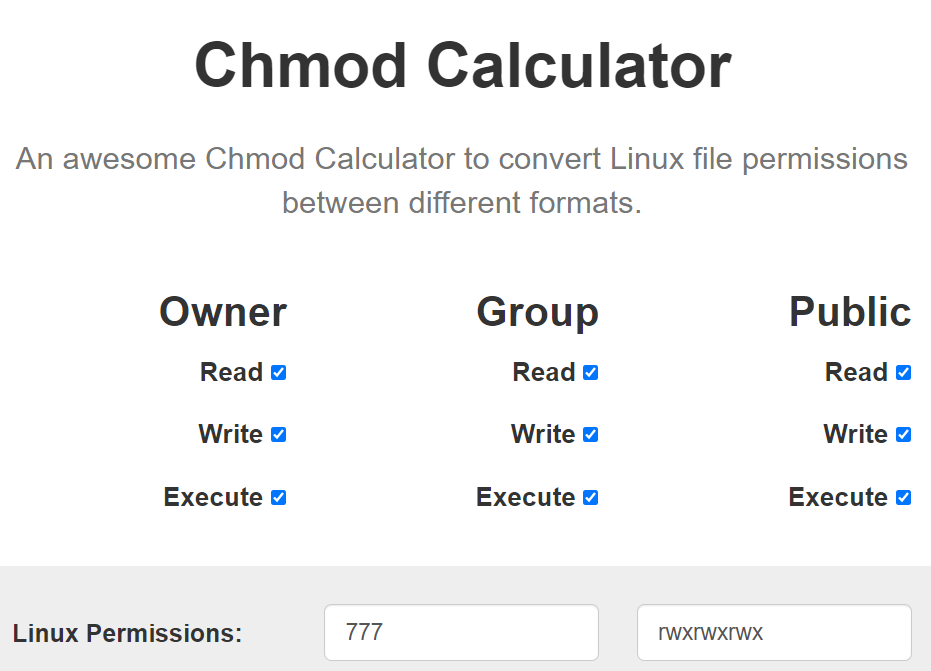
Si equip buit esborrar directori /home/nom\_equip

* Primer s’ha convertir el equip\_usuaris.xls en usuaris.txt per millor entrada al script
* Fer l’script amb els grups i usuaris que li entren per usuaris.txt passat per paràmetre:
  + Crear directoris necessaris
  + Donat un usuari mirar:
    - si ja existeix no crear-lo de nou
    - si cognom no crear i eliminar usuari
    - si no existeix crear usuari
    - Amb el grup d’usuari mirar:
      * si ja existeix entrar usuari directament
      * si no existeix crear grup amb els directoris necessaris
* Finalment executar el fitxer gestio.sh usuaris.txt

Els permisos hauran de ser:



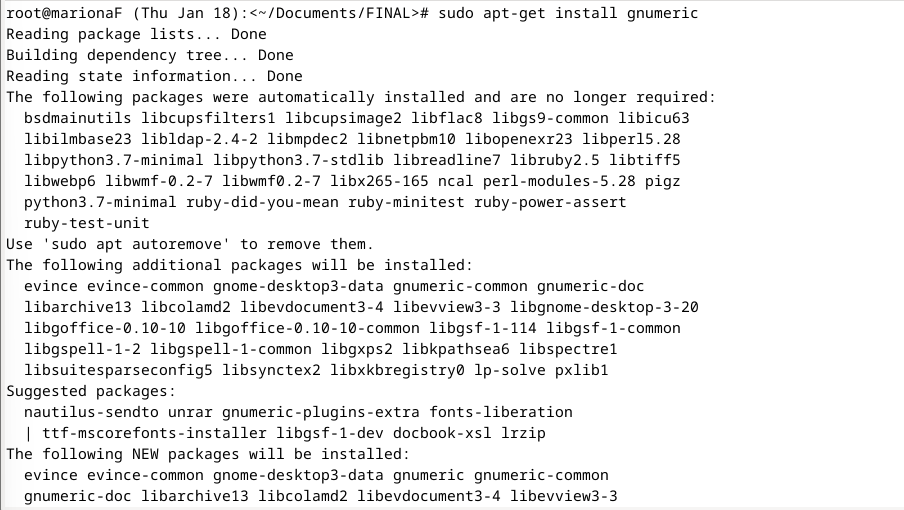
Per cada usuari de l'equip i només la carpeta privada de l'equip



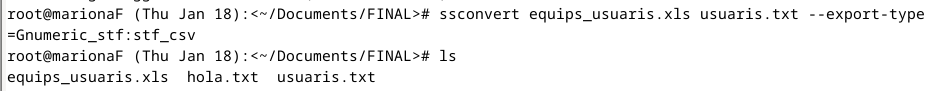
Per les altres carpetes que tothom pot entrar

* 1. (**0,5 punts**) Crea l’arxiu **usuaris.txt** a partir del fitxer de dades adjunt i fes que pugui ser modificat per l’administrador novell. Escriu les instruccions que ha de seguir l’administrador novell per a introduir/eliminar usuaris de l’arxiu **usuaris.txt**

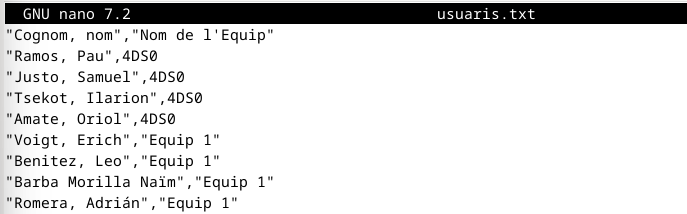
Instalar gnumeric per fer traducció dels fitxers:



Executar següent instrucció per obtenir .txt:



El .txt encara t´he problemes de la primera línea dient les columnes que hi han i els noms estan entre cometes “”:



Eliminar la primera fila i les cometes executant:

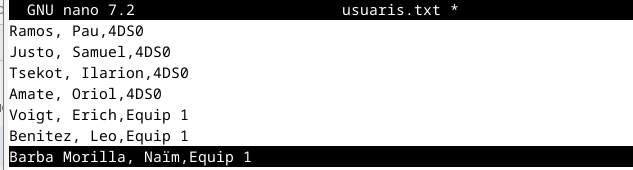
I si queden algunes cometes és perque estan directament després dels noms, donem per suposat que totes les cometes s’han de borrar ja que només serveixen per fer de separadors:

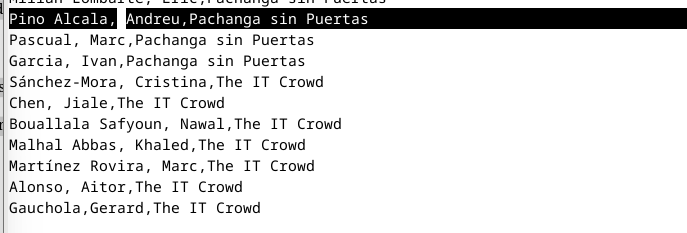


Ara si mirem el usuaris.txt estarà net de cometes i sense la primera línea:



Error en la fila marcada falta una cometa, posada manualment:





* 1. (**1,5 punts**) Posa en marxa (amb una temporització puntual) el mecanisme i comprova la creació/eliminació d’usuaris, permisos d'accés a directoris i altres usuaris.

Crear un script amb totes les especificacions anteriors i que li entri per paràmetre el fitxer usuaris.txt:

(Eliminar els usuaris de .txt si es passa per parametre /dev/null)

Per comprovar users: cut -d: -f1 /etc/passwd

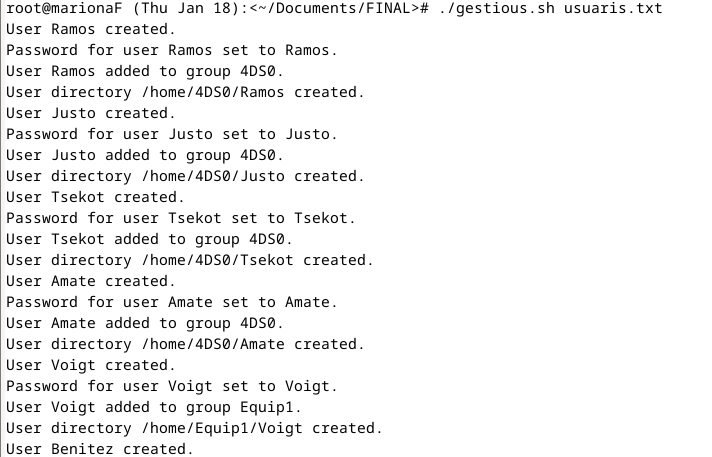
Equips: 4DSO, Equip1,GPTeamm noNameSO, Pachangasinpuertas,The it crowd

| #!/bin/bash  # Check if filename is provided as a parameter  if [ $# -eq 0 ]; then  echo "No filename provided. Usage: ./scriptname.sh usuaris.txt"  exit 1  fi  filename=$1  # If /dev/null is passed as a parameter, delete users  if [ "$2" == "/dev/null" ]; then  while IFS=',' read -r cognom nom equip  do  # Remove spaces  cognom=${cognom// /}  nom=${nom// /}  equip=${equip// /}  # Check if user exists  if id -u "$cognom" >/dev/null 2>&1; then  # Delete user  userdel "$cognom"  echo "User $cognom deleted."  # Delete user's directory  if [ -d "/home/$equip/$cognom" ]; then  rm -r "/home/$equip/$cognom"  echo "User directory /home/$equip/$cognom deleted."  fi  # Delete team directory if it's empty  if [ ! "$(ls -A /home/$equip)" ]; then  rm -r "/home/$equip"  echo "Team directory /home/$equip deleted."  fi  else  echo "User $cognom does not exist. Skipping."  fi  done < "$filename"  else  declare -A equip\_members  while IFS=',' read -r cognom nom equip  do  # Remove spaces  cognom=${cognom// /}  nom=${nom// /}  equip=${equip// /}  # Check if user already exists  if id -u "$cognom" >/dev/null 2>&1; then  echo "User $cognom already exists. Skipping."  else  # Create user without a home directory  useradd "$cognom"  echo "User $cognom created."  # Set user password to cognom  echo "$cognom:$cognom" | chpasswd  echo "Password for user $cognom set to $cognom."  # Create team directory if it doesn't exist  if [ ! -d "/home/$equip" ]; then  mkdir "/home/$equip"  mkdir "/home/$equip/grup"  mkdir "/home/$equip/public"  echo "Directory /home/$equip created."  fi  # Add user to team group  groupadd -f "$equip"  usermod -a -G "$equip" "$cognom"  echo "User $cognom added to group $equip."  # Set permissions for team directory  chown :"$equip" "/home/$equip/grup"  chmod 770 "/home/$equip/grup"  chmod 777 "/home/$equip/public"  # Create user directory inside team directory  mkdir "/home/$equip/$cognom"  chown "$cognom":"$cognom" "/home/$equip/$cognom"  chmod 700 "/home/$equip/$cognom"  echo "User directory /home/$equip/$cognom created."  # Add user to equip\_members array  equip\_members["$equip"]="$cognom ${equip\_members[$equip]}"  fi  done < "$filename"  # Delete team directories if they are empty  for equip in "${!equip\_members[@]}"; do  if [ -z "${equip\_members[$equip]}" ]; then  rm -r "/home/$equip"  echo "Directory /home/$equip deleted."  fi  done  fi |
| --- |

Donar permisos necessaris per executar .sh:

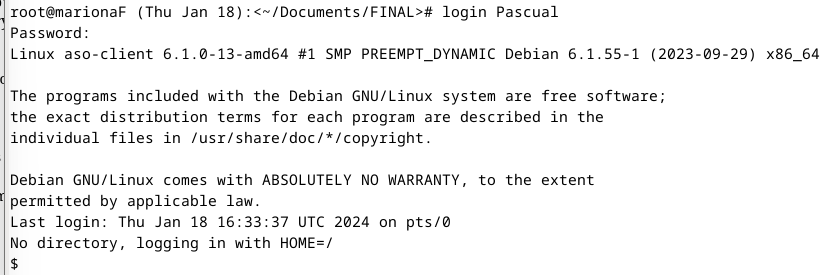


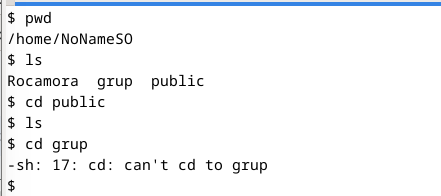
Executar script:



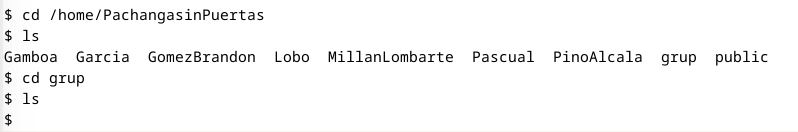
Comprovacions:

login de un usuari:



Pot entrar a altres equips(4dso) però si al seu i mirar les carpetes dels seus companys però no a la carpeta principal del equip (grup)

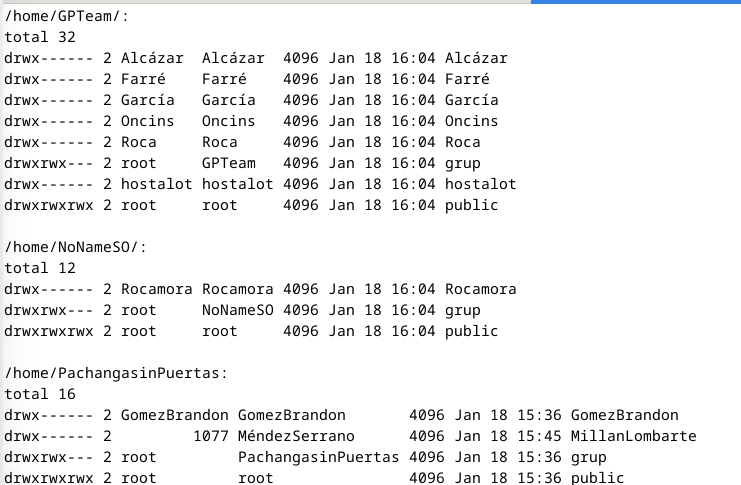
Però si que pot entrar a la carpeta principal del seu equip:

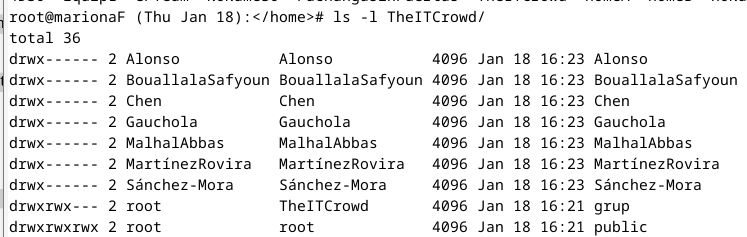


Els directoris dels equips i els seus usuaris en general:

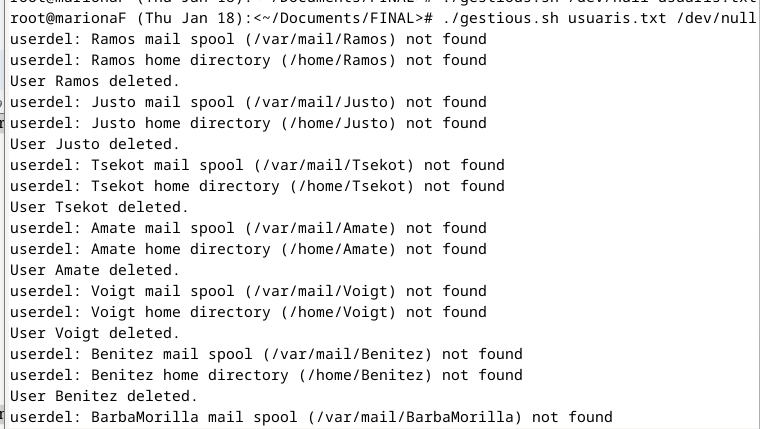








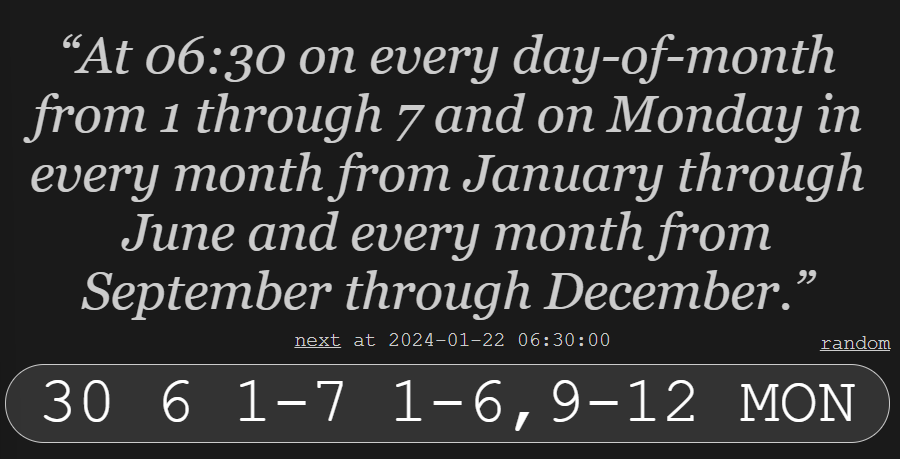
Podem veure que esborra correctament els usuaris:



* 1. (**1 punt**) Temporitza la tasca

Per temporitzar la tasca s’ha de fer servir el crontab, com que pot ser una mica complicat trobar el codi per el temps i la data que es volen, hi ha una web on t’hajuda a trobar-ho: <https://crontab.guru/>

Aquí entrem l’especificació donada



tenint el codi següent per la temporització:

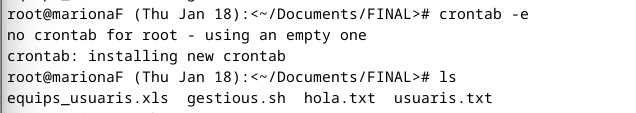
30 6 1-7 1-6,9-12 MON

Després des del fitxer crontab enllaçar el codi de la temporització amb l’script dels usuaris:

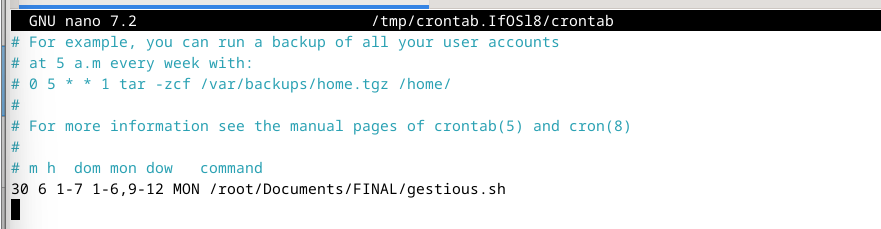
Mirar on esta el fitxer:



I Crear un crontab :



Dins d’aquest posar el codi que s’ha tret anteriorment:



1. **(3 punts)** La comanda **inxi** ens dona informació del nostre sistema. Utilitzant aquesta informació, volem crear un script de protecció del nostre sistema que doni un senyal d’avís quan la temperatura exterior (punt geogràfic a on estem treballant) sigui mes gran que un valor escollit per vosaltres.
   * + Aquest avís ha de contenir informació del nombre de processos en execució i el % de memòria utilitzada.
     + S’ha d’enviar un missatge al terminal de tots els usuaris connectats, tant en local com en remot via ssh. (<https://man7.org/linux/man-pages/man1/wall.1.html>)
     + **Nota: tota la informació s’ha d’extreure de la comanda inxi**
   1. (**0,5 punts**) Explica en format pseudocodi el funcionament del script

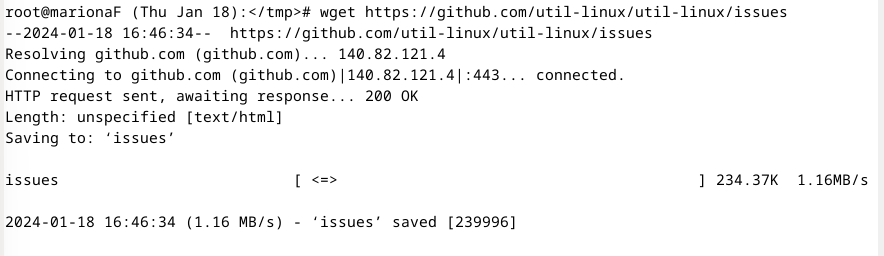
* Aconseguir la temperatura exterior amb la comanda inxi
* Aconseguir els processos de la màquina actuals en execució i la memòria que utilitzen també amb inxi.
* Tenir un interval de temperatura al exterior
* Crear un script que miri la temperatura exterior i si esta dins d’aquest inserval
* Posar en el script els missatges necessaris:
  + temperatura
  + número de treballs
  + memòria utlitzada
* Enviar per wall els missatges a tots els usuaris que estiguin connectats amb els ssh
  1. (**0,5 punts**) descarrega el fitxer precompilat (.**deb**) de inxi amb la comanda **wget** directament al directori **/tmp** (utilitza opcions de wget). Si tens instal·lat inxi, desinstal·la-ho abans de començar.

Instalar wget:



Entrar a /tmp i fer el wget del github donat:

wget <https://github.com/util-linux/util-linux/issues>



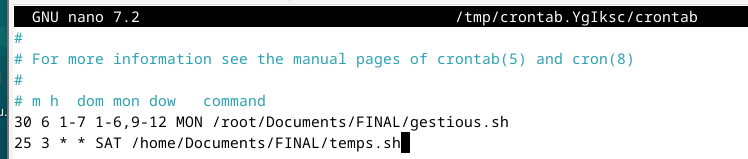
I instalarho amb el dpkg -i

* 1. (**0,**5 **punts**)Temporitza la protecció segons el teu criteri d’administrador del sistema.

Amb el crometab posar una data límit per l’administrador

Executar cada dissabte a les 3:25





* 1. (**1,5 punts**) Realitza les proves que consideris adients per tal que quedi demostrat el correcte funcionament del script (diferents valor de temperatura, enviament d’avis als usuaris connectats en local i en remot via ssh). Mostra captures de pantalla per a cada cas.

Crear el script de .sh i donar-li permisos d’execució:



Entrega:

* **mHKT individual 23-24\_final.odt (o .doc)** amb les captures de pantalla inserides a cadascun dels apartats de l’exercici, i els comentaris que creguis convenient per tal de mostrar la resolució dels exercicis
* tots els codis **font.**