Josep Fayos

  Organizador de Cluster

Killergame

Contenido

[1. Plataformar equipos 3](#_Toc137507729)

[a. Instalar ISO 3](#_Toc137507730)

[b. Instalar paquete de openjdk 19 3](#_Toc137507731)

[c. Ip estática 4](#_Toc137507732)

[2. Ordenador de control 4](#_Toc137507733)

[a. Configurar Cluster SSH 4](#_Toc137507734)

[b. Carpeta compartida Samba 4](#_Toc137507735)

[c. Comando SCP 5](#_Toc137507736)

[d. Script xrandr.sh 5](#_Toc137507737)

[3. Configurar router 5](#_Toc137507738)

[a. DHCP móviles 5](#_Toc137507739)

[4. RAC 5](#_Toc137507740)

[5. Extras 5](#_Toc137507741)

# Plataformar equipos

## Instalar ISO

La primera decisión que tomar es el sistema operativo que vamos a utilizar ya que los recursos de los portátiles son limitados, principalmente la memoria RAM y la CPU.

Se ha elegido un sistema operativo basado en Unix ya que las distribuciones que se encuentran basadas en este sistema suelen ser bastante ligeras y de código abierto, para los portátiles nos va perfecto.

Teníamos varios partidarios para poder instalar entre ellos:

* Xubuntu
* Linux Mint
* Lubuntu
* Lxle
* Ubuntu

Pero tras investigar sobre el entorno gráfico que tienen y las características necesarias en cada uno de ellos nos quedamos con **Lubuntu 22.04**

Una vez instalado en sistema operativo en uno de los ordenadores, me surge la duda de cómo preparar el resto y se piensa en clonar un equipo listo para el resto y así únicamente se debería configurar con las características propias del portátil, pero no del sistema operativo. Esto último no fue viable porque los USB que tienen los portátiles son 2.0 y la transferencia de archivos sea muy lenta, hasta el punto de que es más rápido plataformarlos manualmente.

Para la instalación del sistema no tiene mucho misterio, lo único importante es poner que el teclado es en español para evitar confusiones cuando vayamos a insertar comandos manualmente y el usuario.

El usuario administrador es “dam” y la contraseña P4s5w0rD y otro como “alumnat” sin contraseña y sin permisos de sudo, así tendremos un usuario donde podremos realizar todo el proyecto sin capar por completo los portátiles para otros usuarios.

Con ayuda de varios compañeros de clase, se preparan los equipos, en los siguientes puntos se explicarán las configuraciones realizadas.

## Instalar paquete de openjdk 19

Una vez que los ordenadores ya tienen el sistema operativo instalado, necesitamos algunas configuraciones extra, en este caso vamos a instalar el kit de desarrollo que nos va a permitir arrancar las aplicaciones desarrolladas en Java dentro de nuestro ordenador, así nuestros compañeros cuando nos pasen el archivo .jar nuestro ordenador podrá ejecutar dicho programa en el entorno gráfico.

Este paquete no lo tendremos nada más instalar el sistema, tendremos que conectar a una red con internet para actualizar el repositorio. El comando es el siguiente: **sudo apt update**

Esperamos a que actualice completamente el repositorio local y procedemos a instalar opensdk con el comando: **sudo apt install openjdk-19-jdk**

## Ip estática

Al realizar el proyecto en una red LAN (Local Area Network), necesitaremos tener un control de los equipos conectados para evitar que el DNS (Domain Name System) nos asigne ips dinámicas a los equipos conectados. Para ello vamos a utilizar las ips estáticas en cada uno de los equipos que vayan a interactuar en el proyecto y así saber exactamente a que equipo estamos apuntando cuando enviamos algún paquete a una ip.

**Antes de nada, revisar que se ha quitado la conexión a internet que podamos tener de cuando actualizamos el repositorio local para evitar que nos influya asignando ips.**

En este caso, lo podemos realizar manualmente haciendo clic derecho encima del icono de conexión de la barra del sistema y seleccionando propiedades de las conexiones. Dentro de los ajustes avanzados podemos seleccionar que en vez de tener ip dinámica podemos poner una estática. Para que todas estén en una misma subred utilizamos una red 255.255.255.0, donde se van a utilizar los tres primeros números para nombrar la red y los últimos 255 para los ordenadores. Se usa 172.168.1.x donde la x varía según el número que se le haya asignado al ordenador.

# Ordenador de control

El ordenador de control tendrá exactamente las mismas características que los ordenadores anteriores, pero con algunos extras para poder controlar el resto de los equipos sin necesidad de hacerlo uno a uno. En este caso es únicamente uno y tiene asignado el número 50.

## Configurar Cluster SSH

Se utiliza el comando de cssh para poder conectarnos por ssh a todos los equipos a la vez y poder ejecutar los mismos comandos en todos sin necesidad de conectarnos individualmente.

Primero debemos instalar Cluster SSH, utilizamos el siguiente comando: **sudo apt install clusterssh**

Una vez haya acabado, configuramos el grupo de ips a las que se conectará. El archivo que editar es el que se encuentra en /etc/clusters poniendo el nombre del grupo primero. Un ejemplo es el siguiente: **pantallas dam@192.168.1.1 dam@192.168.1.2 dam@192.168.1.3**

Una vez configurado, ejecutamos el comando: cssh pantallas

Este comando buscará el grupo de pantallas en el archivo e intentará conectarse por ssh a todas las ips puestas con el usuario escrito antes de la @. Nos abre un cuadro donde veremos que si escribimos la contraseña la cogerá en todas las ventanas.

A partir de aquí ya tenemos la opción de poner comandos a la vez en todos los equipos.

## Carpeta compartida Samba

Adjunto un enlace explicativo de como se comparte la carpeta en Samba, donde hemos utilizado como host el ordenador de control:

<https://www.linuxbabe.com/ubuntu/install-samba-server-file-share>

En nuestro caso la carpeta compartida se encuentra en /home/samba/shared/

En el caso de los equipos, gracias al script monta esta carpeta compartida en red en /home/dam/mountedFolder

## Comando SCP

Existe un comando para transferir archivos a través de ssh, es el comando SCP.

La nomenclatura para utilizarlo es la siguiente:

**scp [OPTION] [user@]SRC\_HOST:]file1 [user@]DEST\_HOST:]file2**

Un ejemplo práctico sería el siguiente:

**scp file.txt remote\_username@10.10.0.2:/remote/directory**

Podemos utilizar tanto el origen como el destino el equipo remoto para copiar un archivo de otro equipo al local o a la inversa.

## Script xrandr.sh

Creamos un script para configurar la resolución de las pantallas y aprovechamos para añadir a este mismo todos los elementos necesarios a ejecutar antes de arrancar la aplicación de java para que la configuración que se muestre por pantalla sea correcta y no se haya desconfigurado nada desde el último inicio.

A continuación, se adjunta el script y se explica el funcionamiento:

#!/usr/bin/env sh

# Montar carpeta compartida

sudo mount -t cifs -o username-anonymous,password= //192.168.1.50/public /home/dam/mountedFolder/

# Seleccionar monitor

export DISPLAV=:0

# Configurar resolución de monitor

echo "--DELMODE"

xrandr --delmode VGA1 1024x768

echo "--NEWMODE"

xrandr —newmode "1024x768" 85.25 1368 1440 1576 1784 768 771 781 798 -sync

echo "--ADDMODE"

xrandr --addmode VGA1 1024x768

echo "--OUTPUT"

xrandr --output VGA1 --mode 1024x768

# Ejecutar JAR

cd /home/dam/java/

java -jar /home/dam/mountedFolder/P2P.jar

La mayoría vienen comentadas por el propio script, pero al principio se monta la carpeta compartida montada en Samba en la carpeta local para poder acceder a todo lo que tenga el pc de control. Luego se selecciona que los comandos que vayamos a ejecutar se efectúen sobre la pantalla que está como display 0, es decir, la que está conectada por VGA que es la que nos interesa que tenga buena resolución porque es donde se ejecuta el juego y tras tener la pantalla que sabemos que queremos aplicar los cambios, se borra la posible config que tenga y se añade la resolución, añade a la configuración de la pantalla y se le aplica.

Finalmente, pasamos a la carpeta local de /home/dam/java donde tenemos lo archivos de configuración para que arranque correctamente la aplicación y desde ahí llamamos el .jar que tenemos en la carpeta compartida de Samba. En nuestro caso llamamos siempre a la que tiene el nombre de P2P.jar y vamos borrando y poniendo de nuevo con las versiones que nos pasan nuestros compañeros. Se puede cambiar al nombre que resulte más cómodo.

Dicho script se encuentra en la carpeta de /home/dam/java también de forma local para evitar que se pierda en el caso de no poder acceder a la carpeta compartida donde también se encuentra.

# Configurar router

## DHCP móviles

Se configura para que los dispositivos (móviles principalmente) que se conecten nuevos cojan la ip a partir de la 100 para evitar que interfieran en los dispositivos que tenemos la ip fija.

Para realizar esta configuración nos tenemos que conectar al router y configurarlo poniendo en el navegador la ip del propio router, en nuestro caso 172.168.1.0

# RAC

Este ha sido el apartado más físico, donde hemos preparado el armario RAC donde irán todos los equipos conectados con las pantallas. Se realizan agujeros para poder pasar los cables vga, los cargadores y los ethernet. Todo esto midiendo que quede correcto para que se pueda organizar bien los equipos dentro y probando en el proceso por si se estropeaba algún cable antes de dejarlo atado y ordenado.

Finalmente se estudia como poder refrigerar de mejor forma el RAC pero no conseguimos llegar a una solución, queda pendiente optimizar el aire para una mejora de rendimiento tras estar tiempo encendido ya que sube bastante la temperatura.

# Extras

Como subtareas que se probaron però que finalmente se descartaron tras no conseguir implementarlas fueron las siguientes:

* On login

Intentamos editar un archivo para que al hacer login en el usuario se ejecutara directamente un script y así iniciar el juego nada más acceder, pero no conseguimos configurarlo correctamente porque el on login no tiene interfaz grafica si iniciamos el equipo a través de SSH.

* Wake on lan

Revisamos a ver si con la configuración de la BIOS de los equipos podíamos lanzar algún comando a través de la LAN conectándonos al wifi para iniciar los equipos sin necesidad de tener que hacerlo físicamente, pero no se podía realizar sino se dejaban los equipos encendidos (en suspensión) y eso tampoco nos interesaba ya que seguiría consumiendo. La única opción era con algún enchufe inteligente pero no disponíamos de este material para realizarlo.

* Crear imagen para clonar

Como se comenta a lo largo del documento, al ser los USB 2.0 no salía rentable generar esta imagen ya que se debería traspasar por USB y tarda mucho tiempo, no nos es optimo porque manualmente se tarda mucho menos. Se podría estudiar de hacer a través de la red pero no tuvimos tiempo de revisar como poder hacerlo.