Artix Linux

Guía de Usuario

1. Compartir audio

Al iniciar sesión se crea un *micrófono virtual* con el que podemos combinar el audio de nuestro micrófono con el audio de las aplicaciones. De esta forma podemos compartir junto con nuestra pantalla, el audio de dicha retransmisión en aplicaciones que no tienen dicha funcionalidad.

Para esto tendremos que elegir *my-virtualmic* como nuestro micrófono predeterminado. Podemos añadir el audio de nuestro micrófono real y nuestras aplicaiones mediante el script `pipewire-virtualmic-select`.

2. Cronie y scripts útiles

El repositorio incluye una variedad de scripts para automatizar tareas. Por defecto no se usan, pero podemos hacer que se ejecuten de forma automática con crond.

2.1. convert-2m4a y convert-2mp3

Estos scripts convierten toda la música del directorio que damos como primer argumento a formato *.m4a* o *.mp3* en un mirror que replica al estructura de los archivos en el directorio original.

Si ejecutamos:

```
convert-2mp3 /musica/biblioteca /musica/mp3
```

Toda la música de `/musica/biblioteca` se convertira en mp3 en la carpeta `/musica/mp3`

- El script puede usarse con el flag *-p* para convertir varios archivos de forma paralela, lo que reduce el tiempo necesario para la conversión de archivos, pero consume más recursos y probablemente ocupe todo el tiempo de CPU hasta que el script se termine de ejecutar.
- El script también puede usarse con el flag *-l* lo que hará que además de convertir las canciones a otro formato, se busque la letra de la canción y se incluya dentro del archivo de audio.

2.2. corruption-check

Este script comprueba que no haya archivos corruptos en nuestra biblioteca de música, corrige falsos positivos de corrupción y nos escribe una lista con los archivos que no se pueden reproducir correctamente en `/tmp/corruption.log`. Necesita como argumento el directorio cuyos archivos de audio queremos analizar.

2.3. exif-remove

Este script necesita borra toda la información EXIF de las imágenes que contiene el directorio que se le da como argumento.

2.4. wake

Este script comprueba si hay alguna máquina virtual en ejecución, y si no encuentra ninguna, suspende nuestro equipo y lo reanuda a las 7 de la mañana del día siguiente. Útil para ahorrar energía y no tener que preocuparte por suspender tu equipo, ni de encenderlo por las mañanas.

El script te avisa de que el sistema se va suspender y pasados 10 minutos desde dicho aviso, suspende el sistema. *Si el script se ejecuta pasándole el argumento "now", entonces el sistema se supenderá inmediatamente*.

2.5. wakeme

Este script funciona como un despertador, hace sonar el archivo de audio especificado hasta que le damos al un botón que apaga nuestra alarma.

2.6. compressed-backup

Este script crea un fichero comprimido `tar.gz` con una copia de seguridad del directorio que se le da por primer argumento en el directorio que se le da por segundo argumento. Además se encarga de borrar las copias de seguridad que tienen mas de un mes automáticamente.

3. VFIO GPU passthrough

Con VFIO (Virtual Function I/O) GPU passthrough podemos pasarle una tarjeta gráfica física a una máquina virtual. Lo que nos permite tener gráficos acelerados dentro de dicha máquina virtual.

3.1. Pasos Iniciales

- Primero activamos VT-d o AMD-v *dependiendo de si tenemos un procesador Intel o AMD*.
- Debemos de tener IOMMU activado. En la mayoría de placas bases activar VT-d o AMD-v, también activa IOMMU.
- Debemos tener desactivado CSM (Compatibility Support Module) en los ajustes de arranque de nuestra placa base.

3.2. Pre-configurar el gestor de arranque

El gestor de arranque GNU GRUB, es el programa que se encarga de cargar el kernel de nuestro sistema operativo.

Para usar VFIO necesitamos configurar el arranque del kernel. Las opciones de configuración globales de GRUB están en `/etc/default/grub`

Este archivo contiene las opciones de arranque del kernel en: *GRUB_CMDLINE_LINUX*.

```
GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3 quiet"
GRUB_CMDLINE_LINUX=""
```

Tendremos que añadir las siguientes opciones:

- `iommu=pt`
- `amd_iommu=on` o `intel_iommu=on`, dependiendo de si tenemos una cpu Intel o AMD.

- video=efifb:off

```
GRUB_CMDLINE_LINUX="intel_iommu=on iommu=pt video=efifb:off"
GRUB_CMDLINE_LINUX="amd_iommu=on iommu=pt video=efifb:off"
```

Una vez editado el archivo, debemos actualizar nuestra configuración de GRUB con:

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

3.3. Identificar el ID de nuestra gráfica y los grupos IOMMU

Después de esto ya podemos asignar el driver VFIO a nuestra tarjeta gráfica para poder usarla en nuestra máquina virtual.

Necesitamos conocer el identificador de nuestra gráfica, podemos obtener esta informaciónejecutando en BASH:

```
shopt -s nullglob
for g in /sys/kernel/iommu_groups/*; do
    echo "IOMMU Group ${g##*/}:"
    for d in $g/devices/*; do
        echo -e "\t$(lspci -nns ${d##*/})"
    done;
done;
```

```
IOMMU Group 15:
  08:00.0 VGA compatible controller [0300]: NVIDIA TU116 [GeForce GTX 1660] [10de:21c4] (rev a1)
  08:00.1 Audio device [0403]: NVIDIA TU116 High Definition Audio Controller [10de:1aeb] (rev a1)
  08:00.2 USB controller [0c03]: NVIDIA Device [10de:1aec] (rev a1)
  08:00.3 Serial bus controller [0c80]: NVIDIA TU116 [GeForce GTX 1650] [10de:1aed] (rev a1)
```

Debemos añadir cada uno de los IDs de los dispositivos que se encuentran en el mismo grupo que nuestra gráfica (en nuestro ejemplo: `10de:21c4`, `10de:1aeb`, `10de:1aec` y `10de:1aed`) a nuestro archivo de configuración de GRUB, diciéndole que le asigne a estos dispositivos el driver vfio.

```
GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3 quiet"
GRUB_CMDLINE_LINUX="amd_iommu=on iommu=pt video=efifb:off
vfio-pci.ids=10de:21c4,10de:1aeb,10de:1aec,10de:1aed"
```

Ejecutamos de nuevo:

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

3.4. Módulos del kernel

Ahora cada vez que iniciemos nuestro ordenador el kernel intentará asignarle a esos dispositivos PCI el driver VFIO, pero el driver VFIO por defecto no se carga durante el arranque del sistema.

Debemos de configurar nuestro kernel para que nuestra imagen de arranque incluya el driver VFIO, para poder asignarselo a nuestra gráfica.

Para esto debemos editar `/etc/mkinitcpio.conf` y añadir los módulos del kernel que vamos a usar.

```
MODULES=( vfio_pci vfio vfio_iommu_type1 )
```

Una vez editado nuestro archivo debemos regenerar el initramfs con:

```
# mkinitcpio -P
```

Para comprobar que hemos seguido los pasos correctamente reiniciamos nuestro ordenador y cuando arranque de nuevo, ejecutamos:

```
# dmesg | grep -i vfio
```

Si tenemos una salida parecida a:

```
[3.416692] vfio_pci: add [10de:21c4[ffffffff:ffffffff]] class 0x000000/00000000
[3.433353] vfio_pci: add [10de:1aeb[ffffffff:ffffffff]] class 0x000000/00000000
[3.450019] vfio_pci: add [10de:1aec[ffffffff:ffffffff]] class 0x000000/00000000
[3.466953] vfio_pci: add [10de:1aed[ffffffff:ffffffff]] class 0x000000/00000000
```

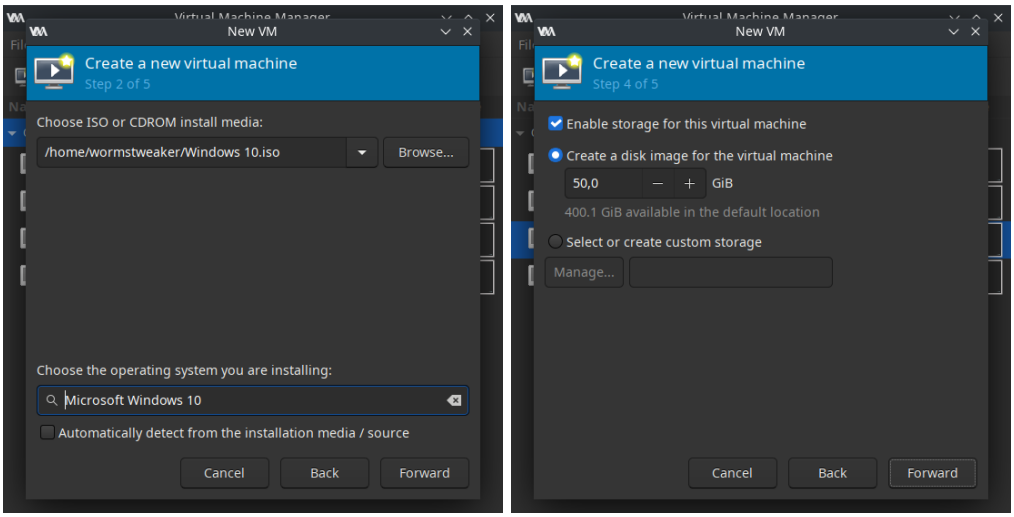
donde podemos ver que el driver `vfio_pci` se ha cargado correctamente para nuestros dispositivos, entonces hemos realizado correctamente todos los pasos.

3.5. Instalación del sistema operativo

Ya tenemos lista nuestra gráfica para ser usada por nuestra máquina virtual, queda instalar nuestra máquina virtual y configurarla para usarla cómodamente.

Descargamos la ISO de Windows 10 desde https://www.microsoft.com, y los drivers que necesitará Windows desde https://fedorapeople.org.

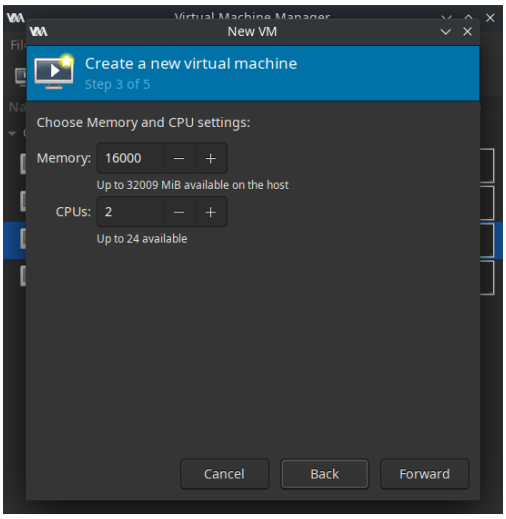
Una vez descargadas las ISO, abrimos *virt-manager* y creamos una máquina virtual:

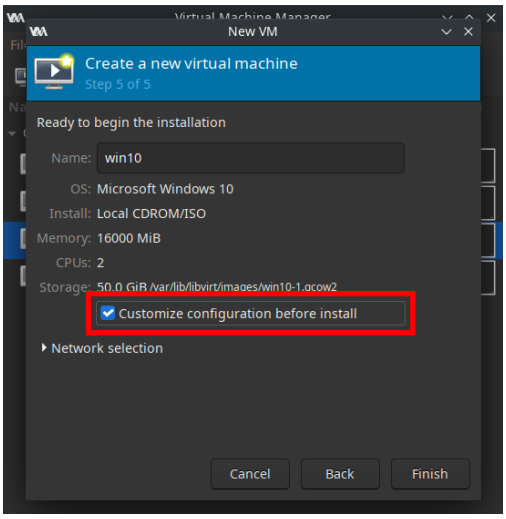


Asignamos la memoria RAM *(8GB como mínimo)* a nuestra máquina virtual. Y en el último paso, elegimos *customizar nuestra máquina virtual* antes de la instalación.

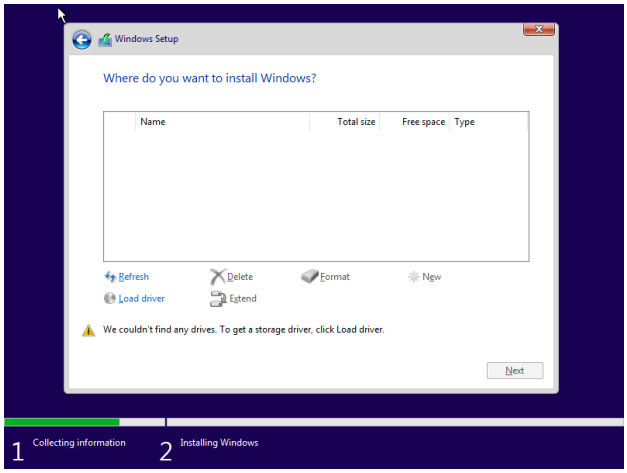
Ahora deberás hacer las siguientes modificaciones:

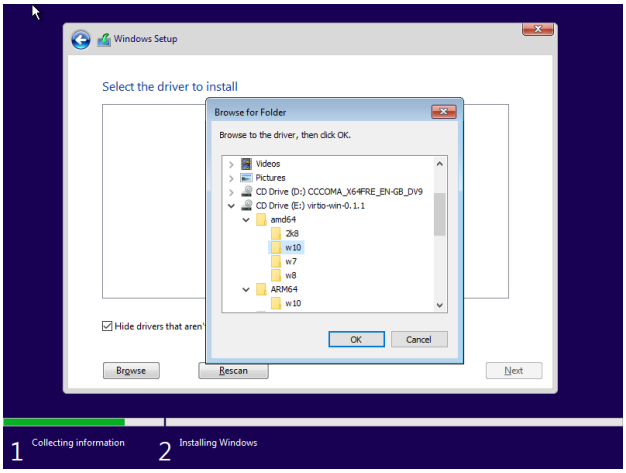
- Cambiar el chipset a *Q35*
- Elegir el firmware `UEFI:x86_64:/usr/share/edk2/x64/OVMF_CODE.fd`
- En la pestaña *CPUs* activa el recuadro *Copiar configuración de la CPU del anfitrión*, despliega el menú *Topología* y ajusta la topología de la forma siguiente:
  - 1 *Socket*, Tantos *Centros* comos núcleos tenga tu procesador y tantos *Hilos* como hilos tenga tu procesador por núcleo. Por ejemplo para un procesador con 8 núcleos y 16 hilos, elegiríamos: *1 Socket, 8 Centros, 2 Hilos*. Esto es, si queremos asignar todos los núcleos a nuestra máquina virtual.





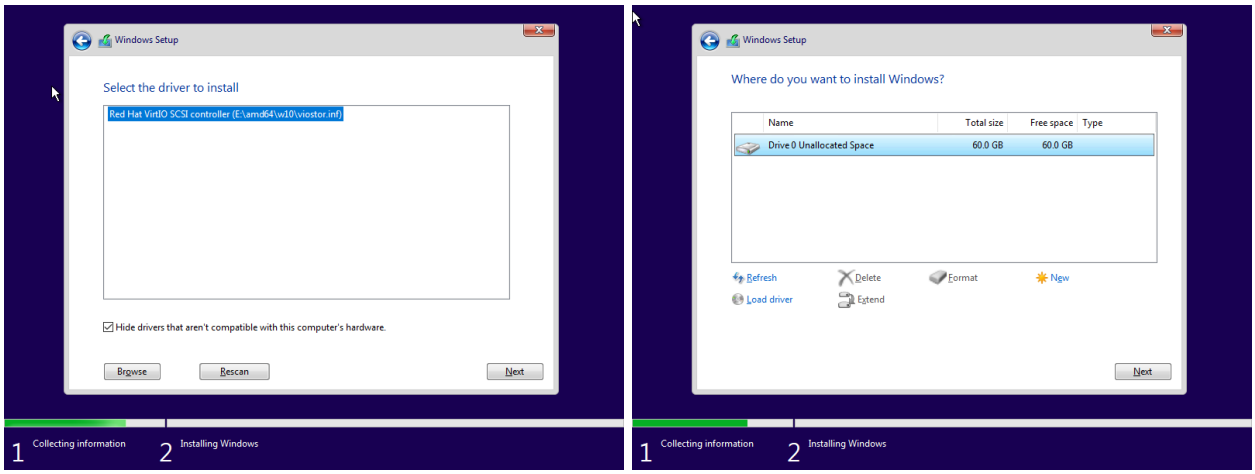
- Nos dirigimos a *SATA Disk 1* y cambiamos el bus a VirtIO, de forma que ahora aparezca *VirtIO Disk 1*. Abrimos las opciones avanzadas, y cambiamos el modo de caché a *writeback*.
- En los ajustes del *NIC* cambiamos el modelo de dispositivo a *virtio*
- Por último dale a *Añadir hardware* y añade un almacenamiento de tipo de dispositivo CDROM y escribe al lado de *Administrar...* la localización de el ISO con los drivers virtio, por ejemplo: */home/usuario/Downloads/virtio-win.iso*.





Dale a *Iniciar la instalación* y cuando veas la selección de disco del instalador de windows verás que está vacío. Primero debes darle a *Cargar driver* o *Load driver*, en inglés.

Selecciona el CD con los drivers virtio y la carpeta que contiene los drivers para Windows 10. Entonces aparecerá tu disco duro virtual y podrás continuar la instalación de Windows. Cuando esta finalize, apaga la máquina virtual.



3.6. Looking Glass en el Host

Para poder usar nuestra máquina virtual de forma cómoda vamos a configurar Looking Glass. Una aplicación que nos permite ver la pantalla conectada a nuestra gráfica a través de una ventana y poder controlar nuestra máquina virtual sin necesidad de conectar un teclado y ratón sólo para nuestra máquina virtual. El único problema es que para usar looking glass sin necesidad de dos monitores, necesitamos comprar un display dummy. Por lo menos, su precio suele rondar los 5-7€.

Looking Glass nos muestra una imagen de lo que esta haciendo nuestra gráfica con una latencia muy baja, suficiente para poder jugar a juegos o editar vídeo *(y otros usos posibles)* sin problema en nuestra máquina virtual. Logra esto mandando la información a través de un archivo compartido entre máquina virtual y host.

Primero necesitamos configurar tmpfilesd para que cada vez que iniciamos el sistema, cree el archivo que compartirán máquina virtual y host para enviarse información. Tenemos que ejecutar el siguiente comando *(sustituyendo usuario por el nombre de tu usuario)*:

```
echo "f /dev/shm/looking-glass 0660 usuario kvm -" | \
doas tee -a /etc/tmpfiles.d/looking-glass.conf
```

Tenemos que instalar el servicio que se encarga de crear estos archivos con:

```
doas pacman -Sy --noconfirm etmpfiles
```

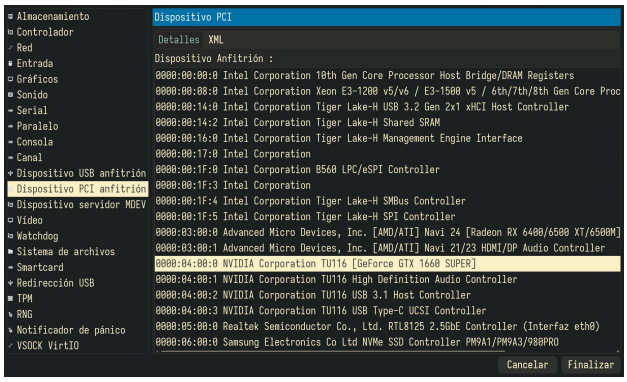
3.7. Añadir Gráfica y Looking Glass en el Guest

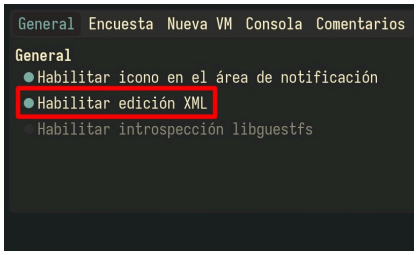
Ahora ya tenemos casi todo listo, queda instalar los drivers de vídeo e instalar Looking Glass en nuestra máquina virtual *(a la hora de la instalación debió de haberse instalado ya en nuestro linux)*.

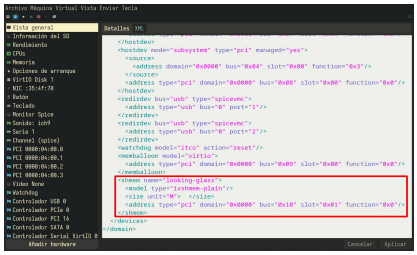
Primero tenemos que añadir nuestra tarjeta gráfica a nuestra máquina virtual. Para ello tenemos que ir a los ajustes de nuestra máquina virtual y en *"Añadir Dispositivo"*, busca la pestaña para añadir dispositivos PCI y añade tu gráfica.

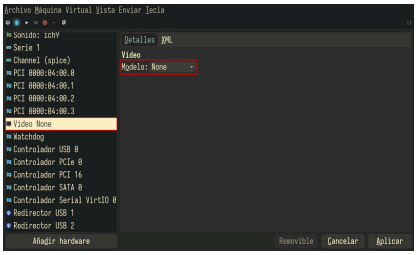
Después debemos de configurar el servidor Spice para que use el puerto 6000 en vez de asignar un puerto automáticamente

Ahora tenemos que activar la edición de XML en Virt-Manager (Editar → Preferencias) y editar el código XML de nuestra máquina virtual. Añadiremos estas lineas de código al final para que el archivo acabe viéndose así:









Iniciamos nuestra máquina virtual e instalamos nuestros drivers de NVIDIA o AMD y Looking Glass Host.

Finalmente cambiamos el driver virtual de video de *QXL* a *None* y borramos el dispositivo de tableta táctil. La próxima vez que iniciemos la máquina virtual solo tenemos que iniciarla en Virt-Manager e ínteractuaremos con nuestra máquina virtual abriendo Looking Glass *(Podemos abrirlo seleccionando “Looking Glass“ en el menú que aparece cuando hacemos clic derecho en alguna parte vacía del Escritorio)*.

Para que esto funcione debemos reiniciar nuestro ordenador o ejecutar el comando.

```
doas install -g kvm -o $(whoami) -m 0660 /dev/null /dev/shm/looking-glass
```

Nota

Para más información sobre como configurar un VFIO passthought puede consultar:

- https://qqq.ninja/blog/post/vfio-on-arch/
- https://gitlab.com/risingprismtv/single-gpu-passthrough/-/wikis/home
- https://wiki.archlinux.org/title/PCI_passthrough_via_OVMF

4. VirtioFS

Podemos usar virtiofs para compartir directorios del host con nuestra máquina virtual. Para ello, necesitamos añadir los directorios deseados desde Virt-Manager, seleccionando: Añadir hardware, Sistema de archivos, eligiendo la Ruta de origen (la carpeta que queremos compartir) y la Ruta objetivo (el nombre con el que aparecerá en nuestra máquina invitada).

También necesitamos instalar los controladores de VirtIO en el host. Si todavía tenemos montado el disco que contiene estos controladores, encontraremos en el CD un instalador (virtio-win-gt-x64.msi) para los controladores, así como otro instalador *opcional* para añadir la integración con el host (virtio-win-guest-tools.exe), que permite funciones como compartir el portapapeles, entre otras.

Después de instalar los controladores, debemos instalar WinFSP para poder montar sistemas de archivos VirtioFS. Y seguidamente ejecutar en el cmd de windows:

```
"C:\Program Files (x86)\WinFsp\bin\fsreg.bat" virtiofs \
"C:\Program Files\Virtio-Win\VioFS\virtiofs.exe" "-t %1 -m %2"
```

Una vez que tenemos todo instalado, podemos crear un script '.bat' que monte nuestras carpetas en la máquina invitada. Por ejemplo, si tenemos dos carpetas cuyas rutas objetivo son "documentos" y "videos", y queremos montarlas en "Y:" y "Z:", respectivamente, podemos crear un script para montar ambas de la siguiente manera:

```
"C:\Program Files (x86)\WinFsp\bin\launchctl-x64.exe" start virtiofs documentos documentos Y:
"C:\Program Files (x86)\WinFsp\bin\launchctl-x64.exe" start virtiofs videos videos Z:
```

5. Conexión bridge y RDP

Adicionalmente, podemos configurar nuestra máquina virtual para, a efectos prácticos, comportarse como un ordenador distinto en la red. En vez de usar un NAT y que nuestro host se encargue del re-direccionamiento al guest, podemos usar una conexión puente y hacer que nuestro router reconozca nuestra máquina virtual como un dipositivo distinto y le asigne su propia IP. Esto es útil para abrir servicios desde nuestro guest y poder alcanzarlos desde el exterior de nuestra red privada.

5.1. Creación de nuestra red puente

Para crear nuestra conexión de puente, utilizaremos *NetworkManager*. Podemos configurar nuestras conexiones a través de la terminal usando `nmtui`. Al ejecutar `nmtui`, seleccionaremos *Modificar una conexión* y luego <*Añadir*>. En el tipo de red, escogeremos *Puente*.

Una vez que seleccionemos el dispositivo tipo *Puente*, aparecerán las distintas opciones de configuración de nuestra red. Procederemos a configurar el dispositivo que queremos puentear; para ello, en la sección *Puente, ports*, seleccionamos <*Añadir*> y elegimos <*Ethernet*> (Nota: esto solo funciona correctamente para dispositivos conectados mediante Ethernet; no es posible utilizar conexiones puenteadas a través de Wi-Fi). Luego, seleccionamos <*Crear*> y finalmente <*Aceptar*>. Por último, confirmaremos los cambios para agregar nuestra conexión de puente.

Finalmente, debemos eliminar la conexión Ethernet de la lista de conexiones en *NetworkManager* (no te preocupes, tu host seguirá teniendo acceso a internet; tanto la máquina anfitriona como la máquina invitada usarán la conexión de puente para acceder a internet). Para ello, en `nmtui` seleccionamos *Modificar una conexión* y eliminamos la conexión correspondiente.

5.2. Invitado con conexión puenteada

Ahora tenemos que decirle a virt-manager que queremos usar nuestra conexión puenteada para la máquina virtual, podemos incluso borrar la red NAT *'default"* si no tenemos ninguna otra máquina virtual que queramos conectar a internet.

Para usar la conexión puenteada en nuestra máquina virtual; en las propiedades de nuestra máquina virtual, dentro de la configuración del **NIC** vamos a elegir en la *'Fuente de red"* nuestra conexión puenteada.

Adicionalmente, podemos ajustar la dirección MAC y establecer una IP estática asociada a esa dirección MAC en nuestro router. Lo cual facilitará confiurar servicios y acceder a nuestra máquina invitada desde la red.

5.3. Configurar Escritorio remoto

Ahora podemos configurar nuestra máquina invitada para acceder remotamente a ella mediante RDP, para ello en Windows *(Sólo si tenemos Windows 10/11 Pro)* en Ajustes/Escritorio Remoto, activaremos el escritorio remoto.

Es recomendable configurar el firewall de nuestro guest para permitir las conexiones entrantes de RDP y ICMP, para ello ejecutaremos estos comandos en la Consola de Comandos como Administrador:

```
netsh advfirewall firewall add rule name="Allow ICMPv4-In" protocol=icmpv4:any,any dir=in action=allow
netsh advfirewall firewall add rule name="Allow ICMPv6-In" protocol=icmpv6:any,any dir=in action=allow
netsh advfirewall firewall add rule name="Allow RDP" protocol=TCP dir=in localport=3389 action=allow
```

Podemos conectarnos ya a nuestra máquina usando freerdp con el comando

```
xfreerdp3 /u:USUARIO /p:'CONTRASEÑA' /v:IP /auth-pkg-list:'!kerberos' \
        /sound:sys:alsa /size:1920x1080 /smart-sizing:1920x1080 +clipboard
```

O puedes usar el script `rdp-connect`

6. SSH

Si queremos conectarnos a un equipo remoto a través de internet debemos configurar OpenSSH. Por defecto incluyo un script que configura SSH de forma muy básica, configurando SSH para poder acceder a tu equipo remoto con una contraseña. Para activar SSH simplemente ejecuta:

```
ssh-configure
```

Mi recomendación es no conformarse con esta configuración básica. Desactiva el login con el usuario root, desactiva el login por contraseña y usa claves públicas para conectarte. Si no sabes de lo que estoy hablando, es mejor que no uses SSH y no te expongas a abrir la puerta a tu ordenador al internet, aunque sea con candado.

6.1. VNC através de SSH

Podemos usar SSH para acceder a nuestro entorno gráfico de forma remota y usar nuestro ordenador sin estar necesariamente delante de el físicamente. Por defecto dwm inicia un servidor VNC para que puedas conectarte remotamente a una interfaz gráfica. Para poder hacer uso del servidor VNC tienes que usar tunneling. Para conectarte y poder usar VNC, ejecuta:

```
ssh usuario@255.255.255.255 -L 5900:localhost:5900
```

Sustituye *usuario* por el usuario de tu máquina, y *255.255.255.255* por la dirección IP de tu máquina. Para conectarte al equipo remotamente debes tener instalado algún cliente VNC, mi recomendación es remmina.

7. Firewall

Instalar y configurar un firewall no es posible dentro de un chroot. Por eso configurar el firewall debe hacerse una vez reiniciemos y hallamos arrancado nuestro sistema operativo. Para instalar el firewall ejecuta:

```
doas sh -c 'pacman -Sy ufw-openrc; rc-update add ufw default; ufw enable'
```

Ahora, la próxima vez que inicies tu equipo tendras un firewall ya instalado. Para configurarlo puedes ejecutar los siguientes comandos, que establecen unas reglas de filtrado de paquetes bastante permisivas.

```
doas ufw limit 22/tcp; doas ufw allow 80/tcp
doas ufw allow 443/tcp; doas ufw allow syncthing
doas ufw default deny incoming; doas ufw default allow outgoing
```

Nota

Si ejecutaste *ssh-configure* el firewall ya se configuró automaticamente

8. Syncthing

Syncthing es un software para sincronizar archivos entre dispositivos, es lo que uso para poder trabajar en todos mis equipos sin necesidad de estar intercambiando pen-drives como un troglodita.

Viene instalado por defecto y se ejecuta cuando inicias tu ordenador. Para configurarlo tiene una interfaz gráfica a la que puedes acceder desde tu navegador.

Información

Aquí puedes encontrar documentación sobre como usar syncthing: https://docs.syncthing.net/

9. Monitores de altas tasas de refresco

Por defecto dwm actualiza el movimiento de las ventanas a 60 FPS. Si tienes un monitor de mas de 60 Hz esto puede hacer la experiencia de usar dwm insatisfactoria. Para cambiar este comportamiento, debes editar el archivo *dwm.c*. En el encontrarás varias funciones con esta misma linea de código:

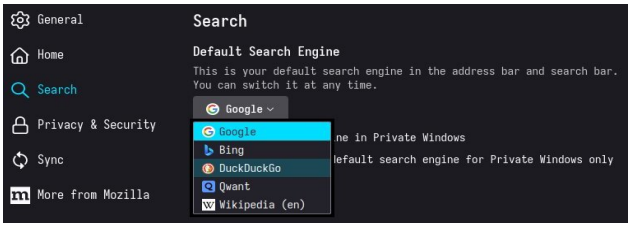
```
if ((ev.xmotion.time - lasttime) <= (1000 / 60))
```

Deberás cambiar el valor *60* por la tasa de refresco de tu monitor en cada aparición de esta linea. Por ejemplo, para un monitor de *144 Hz*, la líneas a cambiar deberían verse así: `if((ev.xmotion.time-lasttime)<=(1000/144))`

10. Firefox

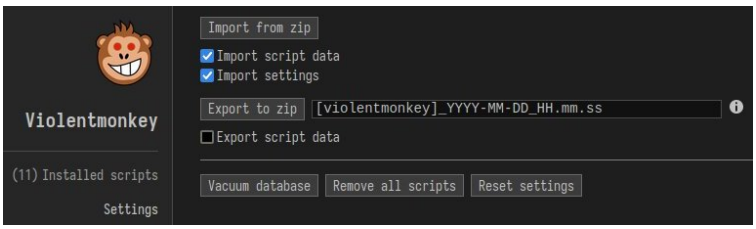
10.1. Cambiar buscador por defecto

Pese a usar arkenfox para mejorar algunos ajustes relacionados con la privacidad, el buscador por defecto sigue siendo Google. Es recomendable cambiarlo por alguno como Brave Search o DuckDuckGo.



10.2. Violentmonkey

Violentmonkey es una extensión para cargar pequeños scripts de javascript justop después de cargar la página web, para automatizar ciertas tareas o añadir funcionalidades. Si vas a los ajustes de la extensión, verás que hay una pestaña donde puedes importar tus ajustes. Podemos importar nuestros ajustes que están en el archivo:



`~/.dotfiles/assets/violentmonkey.zip`

10.3. Añadir buscadores útiles

Aquí tienes una lista de buscadores útiles, para añadirlos a tu buscador simplemente accede a cada página haciendo doble clic, y cuando estés en ella haz clic derecho en la barra de navegación y selecciona *Añadir "Buscador"*.

- Arch Wiki
- Gentoo Wiki
- Alpine Wiki
- YouTube
- Piped
- Invidious
- GitHub
- Yandex
- Nyaa
- Stack Overflow

Las Wikis de Gentoo y Alpine nos resultarán de utilidad, pues ambas distribuciones utilizan OpenRC en vez de SystemD. De hecho, si no encuentras un script de OpenRC para algún servicio en los repositorios de Artix, puedes copiar el script que utilizan Alpine o Gentoo.

11. Miscelánea

Configuraciones locales

Nuestra configuración de `zsh` tiene varios *alias* configurados en:

```
~/.config/zsh/aliasrc
```

Si deseamos añadir alias de forma no global, sin tener que añadir archivos al repositorio y de manera exclusiva para nuestro equipo, podemos crear un archivo con más funciones y abreviaciones en:

```
~/.config/useralias
```

Aviso de Uso

Este proyecto es para uso personal. Se comparte con la intención de que pueda ser útil para otros, pero se proporciona tal cual, sin ninguna garantía. No se asume responsabilidad por ninguna pérdida de datos o problemas que puedan surgir del uso de este proyecto.

No obstante, se aceptan propuestas de mejora razonables. Si alguien desea contribuir a hacer este setup más extensible y robusto, puede realizar un merge request sin inconveniente.