

Configuración de Artix OpenRC

aleister888

29 de marzo de 2024

1 Introducción

Este documento pretende ser una guía para terminar de configurar nuestra instalación de Artix, pues hay cosas que no son factibles auto-configurar. Así como puede ser una ayuda para terceros que se han animado a usar mi instalador y que ahora se encuentran ante la tarea de aprender a usar un entorno de trabajo que no es el suyo.

Nota

Cualquier texto que veas coloreado en naranja es un hiper-vínculo (*exceptuando los títulos*), si haces doble clic en él, te llevará a la página web a la que hace referencia o a una donde se explica de que estoy hablando.

2 Atajos de teclado

Dwm es el programa que se encarga de repartir el espacio de trabajo entre nuestras ventanas (*aplicaciones gráficas*). Hay una cantidad enorme de [videos sobre dwm](#), como funciona y sus ventajas. Aquí veremos simplemente los atajos de teclado que tengo configurados:

- Alt Izq. + Ctrl + F1: Abrir este documento
- Alt Izq. + P: Abrir lanzador de comandos
- Alt Izq. + Shift + P: Abrir lanzador de aplicaciones
- Alt Izq. + Shift + Intro: Abrir terminal
 - Alt Izq. + C: Copiar texto
 - Alt Izq. + V: Pegar texto
- Alt Izq. + F1 / Alt + Shift + F1: Configurar pantallas
- Alt Izq. + F2: Abrir Navegador (*Firefox*)
- Alt Izq. + F3: Abrir Administrador de archivos
 - Espacio: Seleccionar archivos
 - Shift + S: Borrar archivos
 - Shift + D: Mover archivos a la papelera
 - Ctrl + D: Vaciar papelera
 - Alt Izq. + D: Restaurar papelera
 - Shift + P: Mirar el tamaño de una carpeta

- Ctrl + Z: Permitir arrastrar archivos a otra ventana
 - D: Cortar archivos
 - Y: Copiar archivos
 - P: Pegar archivos
 - Shift + E: Extraer archivo
 - Ctrl + E: Comprimir contenidos de la carpeta actual
 - Ctrl + R: Renombrar los contenidos de la carpeta actual
 - R: Renombrar archivo
 - S: Abrir shell
-
- Alt Izq. + F4: Abrir el reproductor de música
 - Alt Izq. + F5: Montar dispositivo android en el árbol de ficheros
 - Alt Izq. + Shift + F5: Desmontar dispositivo android del árbol de ficheros
 - Alt Izq. + F11: Abrir gestión de energía
 - Alt Izq. + Shift + F11: Reiniciar dwm
 - Alt Izq. + F12: Abrir mezclador de sonido
 - Alt Izq. + Z: Canción anterior
 - Alt Izq. + X: Canción siguiente
 - Alt Izq. + Shift + Z/X: Pausar/reanudar la reproducción
 - Alt Izq. + N: Bajar Volumen
 - Alt Izq. + M: Subir Volumen
 - Alt Izq. + Shift + N: Establecer volumen al 50 %
 - Alt Izq. + Shift + M: Establecer volumen al 100 %
 - Alt Izq. + Ctrl + N/M: Silenciar/des-silenciar el audio
 - Alt Izq. + O: Captura de pantalla al portapapeles
 - Alt Izq. + Ctrl + O: Guardar captura de pantalla
 - Alt Izq. + Shift + O: Captura de un área de la pantalla al porta-papeles
 - Alt Izq. + Ctrl + Shift + O: Guardar captura de un área de la pantalla
 - Alt Izq. + B: Ocultar/Mostrar la barra de estado
 - Alt Izq. + ,: Mover el foco a la posición anterior
 - Alt Izq. + Shift + ,: Mover la ventana a la posición anterior
 - Alt Izq. + .: Mover el foco a la posición siguiente
 - Alt Izq. + Shift + .: Mover la ventana a la posición siguiente
 - Alt Izq. + Q: Moverse al espacio anterior
 - Alt Izq. + W: Moverse al espacio siguiente
 - Alt Izq. + 1-9: Moverse al espacio 1-9
 - Alt Izq. + Shift + 1-9: Mover ventana al espacio 1-9
 - Alt Izq. + Shift + Q: Cerrar ventana
 - Alt Izq. + {: Mover el foco al monitor anterior
 - Alt Izq. + }: Mover el foco al monitor siguiente
 - Alt Izq. + Shift + {: Mover ventana al monitor anterior
 - Alt Izq. + Shift + }: Mover ventana al monitor siguiente

3 Compartir audio en Discord

Si has instalado el cliente de discord te darás cuenta de una funcionalidad que esta en la versión de Windows y no en la de Linux. En Linux, no puedes compartir el audio cuando compartes tu escritorio o video.

Para solucionar este problema, cuando inicias sesión se crea un *micrófono virtual* al que puedes añadir tu micrófono real y el audio de las aplicaciones que desees. Para que la gente pueda escuchar el audio de las aplicaciones tendrás que elegir en discord el micrófono como *my-virtualmic*, y, recomendable; desactivar al reducción de ruido. Pues esta detectará el audio de las aplicaciones como ruido.

Para añadir tu micrófono, haz click en una parte vacía del escritorio y en la pestaña *Compar. audio* haz clic en *Ajustes Mic..* Aparecerá un menu para añadir o quitar dispositivos de audio del micrófono virtual.

También verás que en la pestaña *Compar. audio* puedes hacer clic en las diferentes opciones que hay para compartir o dejar de compartir el audio de las aplicaciones

4 Cronie y script útiles

Mi instalación viene con varios scripts para automatizar tareas pero que por defecto no se usan para nada.

Con tu instalación base deberías de tener activado el servicio **crond**. Este servicio se encarga de ejecutar comandos de forma automática. El archivo de configuración se encuentra en */etc/crontab* y tal como se ha configurado, debería verse como:

```
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

0 8 * * * root reflector --verbose --latest 10 --sort rate --save /etc/pacman.d/mirrorlist-arch
@reboot usuario syncthing --no-browser --no-default-folder
```

- `"0 8 * * *"` es la parte de donde definimos cuando se ejecutara el comando, si quieres investigar hay un montón de guías en youtube o cualquier otra plataforma sobre cual es la sintaxis para configurar la ejecución de los comandos. Puedes también comprobar si tu sintaxis es correcta en crontab.guru.
- `root` es el usuario que ejecuta el comando, y `reflector --verbose --latest 10 --sort rate --save /etc/pacman.d/mirrorlist-arch`, el comando.

4.1. convert-2m4a y convert-2mp3

Este script coge toda la música del directorio que damos como primer argumento y nos hace un mirror en la carpeta que damos como segundo argumento en formato *m4a* o *mp3*.

Si ejecutamos:

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

esto nos convertirá toda la música de */musica/biblioteca* a la carpeta */musica/mp3*, en formato mp3. Lo que nos puede resultar de interés porque los archivos mp3 ocupan menos que su equivalente en flac u otros formatos sin pérdida de calidad. Personalmente, uso este script para convertir toda mi música a una carpeta que se sincroniza con mi móvil, que no tiene almacenamiento suficiente para guardar mi biblioteca música en su calidad original.

Podemos automatizar este proceso, para que el usuario *usuario1* convierta a mp3 su música, todos los días a las 8:30, añadiendo esta línea a */etc/crontab*:

```
30 8 * * * usuario1 convert-2mp3 /musica/biblioteca /musica/mp3
```

4.2. 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 realmente están corruptos y no se pueden reproducir correctamente en */tmp/corruption.log*. Solo necesita como argumento el directorio cuyos archivos de audio queremos comprobar.

Podemos automatizar esta tarea añadiendo el comando a */etc/crontab*, aquí un ejemplo:

```
15 7 * * * usuario1 corruption-check /musica/biblioteca
```

4.3. exif-remove

Este script necesita como argumento un directorio y borrará toda la información EXIF de las imágenes que contiene el mismo.

Los metadatos EXIF sirven para identificar el usuario que tomó la fotografía, por ejemplo, si tomas una fotografía con tu teléfono, el teléfono guarda como información EXIF datos como; el teléfono desde el cual la fotografía fue tomada, o desde que coordenadas geográficas fue tomada la foto. Esto es útil para compartir imágenes guardadas en tu ordenador reduciendo la huella digital que dejas cuando las compartes.

Para borrar automáticamente los metadatos de una carpeta, puedes añadir a */etc/crontab* una línea parecida a:

```
0 17 * * * usuario1 exif-remove /fotografias
```

4.4. 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.

Para suspender tu equipo automáticamente todos los días a las 11:00 puedes añadir a tu *crontab*:

```
0 23 * * * root wake
```

Y para notificar al usuario *usuario1* de que el sistema se suspenderá con 10 minutos de antelación:

```
50 22 * * * usuario1 DISPLAY=:0 XDG_RUNTIME_DIR=/run/usr/$(id -u) notify-send
-u critical -i system-error "El sistema se suspenderá en 10min"
5 7 usuario1 dunstctl close-all
0 23 * * * root wake
```

Para que este script funcione correctamente, es recomendable establecer el reloj como TSC. Puedes mirar como se hace esto en ??

4.5. wakeme

Este script es un despertador, hace sonar un archivo de audio hasta que le damos al un botón que apaga nuestra alarma. Necesita como único argumento el archivo de audio que queremos usar como despertador. Si queremos que este despertador suene todas las mañanas a las 7:30, podemos añadir una línea parecida a esta a */etc/crontab*:

```
30 7 * * * usuario1 DISPLAY=:0 XDG_RUNTIME_DIR=/run/usr/$(id -u) wakeme ~/musica/alarma.mp3
```

4.6. Mantenimiento btrfs

Si elegiste particionar tu disco con BTRFS es recomendable que se ejecuten tareas de mantenimiento cada *X* tiempo. Añadir esto a tu */etc/crontab* bastará:

```
0 14 * * * 5 root btrfs balance start -dusage=95 / && btrfs balance start -usage=95 /
```

Esto realizará diversas tareas de mantenimiento en tu disco principal todos los viernes a las 14:00.

Si tienes un disco diferente para tu partición home, también formateado en BTRFS, puede interesarte hacer lo mismo para tu disco */home*

```
0 14 * * * 5 root btrfs balance start -dusage=95 /home && btrfs balance start -usage=95 /home
```

4.7. Limpiar cache periódicamente

Puede interesarte limpiar periódicamente archivos de registro y cache antigua de tu ordenador. Para borrarla de forma periódica para el usuario *usuario1* añade a tu */etc/crontab*:

```
30 7 * * */2 usuario1 find ~/.cache -mtime +2 -delete
30 7 * * */2 usuario1 find ~/ -name "*.log" -mtime +4 -delete
```

4.8. Reorganizar plugins

Si elegiste instalar herramientas de producción musical, deberás tener instalado **yabridge**, una aplicación para usar plugins **VST** de windows en linux. Es recomendable configurar el escaneo de nuevos plugins y el mantenimiento de tu librería de plugins para que se haga automáticamente.

```
15 23 * * * usuario1 yabridgectl sync --prune
```

4.9. 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.

Por ejemplo, añadir esta línea a nuestro */etc/crontab*:

```
0 16 * * * 3 usuario1 compressed-backup ~/Música /Resguardos/Música
```

Nos crearía todos los miércoles a las 16:00 una copia de seguridad de la carpeta */Música* en un archivo comprimido en */Resguardos/Música*

4.10. Copias de seguridad

Podemos usar el comando *rsync* para crear copias de seguridad tipo espejo de nuestro directorio */home* en otro disco duro.

Por ejemplo, añadir esta línea a nuestro */etc/crontab*:

```
0 16 * * */2 usuario1 rsync -av --delete /home/${whoami}/ /mnt/Mirror1
0 16 * * */7 usuario1 rsync -av --delete /home/${whoami}/ /mnt/Mirror2
```

Hará que se cree un espejo de nuestro directorio */home* en */mnt/Mirror1* cada 2 días y otro en */mnt/Mirror2* cada semana.

5 VFIO GPU passthrough

VFIO (Virtual Function I/O) GPU passthrough es una forma de asignar una tarjeta gráfica física a una máquina virtual en un entorno de hipervisor.

Esto es de nuestro interés cuando queremos tener gráficos acelerados dentro de una máquina virtual. Si afirmó que iba a usar máquinas virtuales a la hora de la instalación, *virt-manager* debería haberse instalado y configurado para usar máquinas virtuales con QEMU (Quick EMUlator) y KVM (Kernel-based Virtual Machine).

Sin embargo, crear una máquina virtual y auto-configurar el uso de una gráfica dedicada a través de VFIO es un proceso más complejo, que prefiero dejar en manos del usuario.

A continuación te acompañare en el proceso:

5.1. Pasos Iniciales

- Antes de usar máquinas virtuales necesitas tener **VT-d o AMD-v** activado *dependiendo de si tienes un procesador Intel o AMD*, lo que nos permite la ejecución de máquinas virtuales.
- Debes de tener **IOMMU (Unidad de administración de memoria de entrada/salida)** activado, lo mas probable es que en tu placa base, activar VT-d o AMD-v active también IOMMU. Incluyo este requisito para los raros casos en los que activarlo es un ajuste por separado.
- Debes tener desactivado **CSM (Compatibility Support Module)**. Esto normalmente se hace en los ajustes de arranque de tu placa base.

5.2. Pre-configurar el gestor de arranque

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

Para poder usar VFIO necesitamos configurar como arranca nuestro sistema operativo. Las opciones de configuración globales de GRUB están en */etc/default/grub*

En este archivo, tendremos que localizar la línea donde están las opciones de arranque, la podemos identificar por *GRUB_CMDLINE_LINUX* y el archivo es algo parecido a:

```
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`

Al finalizar, nuestro archivo debería verse así:

```
GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3 quiet"
GRUB_CMDLINE_LINUX="amd_iommu=on iommu=pt video=efifb:off"
...
```

Una vez editado el archivo, debemos actualizar nuestra configuración de GRUB, ejecutando el siguiente comando con permisos de administrador:

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

Después de esto vamos a reiniciar nuestro ordenador para avanzar al siguiente paso.

5.3. 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, que es una cadena de caracteres que identifica inequívocamente un dispositivo PCI (*en nuestro caso nuestra tarjeta gráfica*). Para poder obtener esta información, deberás ejecutar esta función con 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;
```

Una vez ejecutes este bloque de comandos, deberás obtener una salida parecida a la siguiente:

```
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)
```

Debes fijarte en que grupo se encuentra tu tarjeta gráfica, en este caso, se trata de una MSI GTX 1660.

Ahora que tienes identificado en que grupo esta tu tarjeta gráfica, nos tenemos que fijar en las ID, que están contenidas por corchetes, para nuestro ejemplo serían:

- `[10de:21c4]` Tarjeta de video (GPU) compatible con VGA, se correspondería con el procesador gráfico.
- `[10de:1aeb]` Es el dispositivo de audio de nuestra gráfica, se encarga por ejemplo, de darnos audio através de HDMI

- `[10de:1aec]` Este es el controlador USB de nuestra gráfica
- `[10de:1aed]` Este es el controlador de bus serie de nuestra gráfica

Ahora debemos añadir cada uno de estos ID (*sólo el texto contenido por los corchetes, sin incluir a estos últimos*), separados por comas a la opción de GRUB; `vfio-pci.ids=`

Si añadimos esta opción a nuestro archivo de configuración de GRUB, este quedaría como:

```
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
```

Esto lo que hace es decirle al kernel que utilice el driver VFIO para todos esos dispositivos PCI, por el contrario, si no hiciésemos esto, el kernel le asignaría el driver *nvidia* o el correspondiente para un uso normal.

5.4. Módulos del kernel

Ahora cada vez que iniciemos nuestro ordenador el kernel intentará asignarle a esos dispositivos PCI el driver VFIO, *pero hay un problema*. El driver VFIO por defecto, no se carga en memoria a la hora del arranque del sistema (*cosa que tiene sentido, poca es la gente que utiliza esta función del sistema operativo y cargarla en memoria por defecto sería un desperdicio de memoria*).

Debemos entonces configurar que cuando arranquemos el ordenador, se nos cargue el driver VFIO, para poder asignárselo a nuestra gráfica.

Para esto debemos editar `/etc/mkinitcpio.conf` y añadir los módulos del kernel que vamos a usar. El archivo se vería algo así:

```
MODULES=()
BINARIES=()
FILES=()
HOOKS=(base udev autodetect modconf block filesystems keyboard fsck)
```

Esto es una simplificación, y el archivo real tendría un montón de líneas de comentario, encabezadas por #, pero nos sirve para ilustrarnos.

Debemos localizar la línea con `MODULES=()` y añadir los módulos que queremos cargar con el arranque del ordenador. Si antes los paréntesis no contenían nada, ahora la línea pasaría a verse así:

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

Una vez editado nuestro archivo debemos regenerar el **initramfs (Initial RAM File System)** ejecutando con permisos de administrador el comando:

```
mkinitcpio -P
```

Vamos a comprobar que los cambios se han realizado correctamente. Para esto vamos a reiniciar nuestro ordenador y cuando arranque el ordenador de nuevo, vamos a ejecutar: