

Universidad Católica de Temuco

FACULTAD DE INGENIERÍA
INGENIERÍA CIVIL EN INFORMÁTICA

COMPARATIVA ENTRE LOS SISTEMA OPERATIVO ARCHLINUX Y OPENBSD

Informe de mantención de sistemas

Oscar Mellado Bustos
Ayudante de laboratorio
Estudiante de segundo semestre

Agosto 2022

Índice

1. Resumen	2
2. Introducción	2
3. Marco teórico	3
3.1. Partición de disco	3
3.1.1. Esquema de partición MBR	3
3.1.2. Tipos de particiones	5
3.1.3. Sistemas de archivos	5
3.2. Kernel	6
3.3. Arranque del sistema	6
4. Configuración e instalación de los diferentes sistemas operativos	7
4.1. Sistema archlinux	7
4.1.1. Requerimientos mínimos, recomendados y usados	7
4.1.2. Descripción del proceso básico de la instalación	7
4.1.3. Pre-instalación	8
4.1.4. Instalación	13
4.1.5. Configuración	13
4.1.6. Administración de usuarios	16
4.1.7. Instalación de la interfaz gráfica	18
4.1.8. Administrador de tareas	20
4.1.9. Controlador de dispositivos	21
4.1.10. Interfaz de comandos	21
4.1.11. Visor de sucesos	22
4.1.12. Directorios importantes del sistema	22
4.2. Sistema openBSD	23
4.2.1. Requerimientos mínimos, recomendados y usados	23
4.2.2. Descripción del proceso básico de instalación	23
4.2.3. Pre-instalación	24
4.2.4. Instalación	25
4.2.5. Creación de un usuario con permisos de administrador	26
4.2.6. Administrador de tareas	27
4.2.7. Controlador de dispositivos	27
4.2.8. Interfaz de comandos	28
4.2.9. Visor de sucesos	28
4.2.10. Directorios importantes del sistema	29
5. Conclusiones	30

1. Resumen

En este informe se comparan los sistemas operativos archlinux que es un sistema de tipo UNIX orientado usuarios avanzados y el sistema openBSD que está orientado a servidores por ser altamente eficiente con el uso de los recursos. En el desarrollo del informe se define un marco teórico con información sobre las particiones de disco, el esquema de particiones MBR, los tipos de particiones, y los sistemas de archivos. También se indaga en la definición de kernel, además del arranque de los sistemas operativos. Se explica de forma detallada la instalación de los sistemas archlinux, y openBSD, dando los requerimientos mínimos, recomendados, y usados de los sistemas, agregando una guía paso a paso de la instalación y configuración de cada sistema. Explorando los, en diferentes programas necesarios para su administración, donde se van a ir comparando.

2. Introducción

Un sistema operativo es el programa que actúa como intermediario entre la comunicación de los servicios o programas con los que interactúa directamente el usuario y el hardware [Stallings, 2005]. Gestiona los recursos de la máquina como las diferentes memorias presentes en una computadora, la velocidad reloj de la CPU o el volage que con el que funciona. En la actualidad existen diferentes sistemas operativos, orientados a diferentes trabajos y especializados en diferentes áreas, como los sistemas orientados a una estación de trabajo o pc personal y los sistemas orientados a los servidores.

Dentro de los SO (Sistemas Operativos) orientados a las estaciones de trabajo está el sistema archlinux. Es una distribución Linux para computadoras x86-64¹, arquitecturas ARM y I686 orientada a usuarios avanzados. Se compone en su mayor parte de software libre y de código abierto que apoya la participación comunitaria. Su modelo de desarrollo es de tipo "rolling release" esto quiere decir que está en constante lanzamiento de nuevas actualizaciones haciendo que los programas que se instalen a través del gestor de paquetes estén siempre en sus últimas versiones. Archlinux nació inspirado en la distribución minimalista CRUX² y el programador que lo desarrolló fue Judd Vinet en marzo de 2002.

Del lado de los SO dedicados a servidores está openBSD. Un sistema completo, libre, multiplataforma de tipo UNIX basado en BSD³. Sus esfuerzos están enfatizados a la portabilidad, estandarización, la correctitud⁴, la seguridad proactiva y la integración de la criptografía [Equipo de openBSD, 2022]. Los desarrolladores dan como ejemplo el programa ampliamente usado openSSH. OpenBSD mantiene versiones portables en muchos subsistemas como paquetes para otros sistemas operativos. Debido a que el sistema se desarrolla con la licencia BSD, muchos de sus componentes se reutilizan en proyectos de software propietarios y patrocinados por empresas. El código del cortafuegos de macOS de Apple se basa en el código del cortafuegos PF de OpenBSD, la biblioteca estándar Bionic C de Android se basa en el código de OpenBSD, LLVM utiliza la biblioteca de expresiones regulares de OpenBSD y Windows 10 utiliza OpenSSH (OpenBSD Secure Shell) con LibreSSL.

¹x64 (también conocido como x86_64 y AMD64) es la versión de 64 bits del conjunto de instrucciones x86. Soporta una cantidad mucho mayor de memoria virtual y memoria física de lo que le es posible a sus predecesores, permitiendo a los programas almacenar grandes cantidades de datos en la memoria. x86-64 también provee registros de uso general de 64 bits y muchas otras mejoras [AMD, 2012].

²CRUX es una distribución GNU/Linux liviana, simple y optimizada para la arquitectura X86-64 orientada a usuarios experimentados de Linux.

³BSD fue un sistema operativo derivado de Unix que nace a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

⁴La correctitud corresponde a una propiedad que distingue a un algoritmo de un procedimiento efectivo. Un algoritmo es correcto si: resuelve el problema computacional para el cual fue diseñado, para cada entrada, produce la salida deseada, termina en un tiempo de ejecución finito.

3. Marco teórico

3.1. Partición de disco

Una partición de disco, en mantenimiento, es el nombre genérico que recibe cada división presente en una sola unidad física de almacenamiento de datos. Toda partición tiene su propio sistema de archivos, generalmente, casi cualquier sistema operativo interpreta, utiliza y manipula cada partición como un disco físico independiente, a pesar de que dichas particiones estén en un solo disco físico [Levi, 2002].

Una partición de un disco duro es una división lógica en una unidad de almacenamiento, en la cual se alojan y organizan los archivos mediante un sistema de archivos. Existen distintos esquemas de particiones para la distribución de particiones en un disco. Los más conocidos y difundidos son MBR (Master Boot Record) y GPT (GUID Partition Table).

Una partición necesita de un sistema de archivos para poder almacenar datos. Existen múltiples sistemas de archivos con diferentes capacidades como: FAT, NTFS, FAT32, EXT2, EXT3, EXT4, Btrfs, ReiserFS, Reiser4 u otros.

En sistemas UNIX y UNIX-like, las particiones de datos son montadas en un mismo y único árbol jerárquico, en el cual se montan a través de una carpeta, proceso que solo el superusuario (root) puede realizar.

Es común que en los sistemas basados en UNIX se usen hasta 3 particiones. La principal que esta montada en el directorio raíz (/), la segunda que se usa para montar el directorio **/home**, en donde se guardan las configuraciones de los usuarios, y la tercera partición llamada swap, que se usa para la memoria virtual temporal. Las particiones mínimas que necesita el sistema operativo son dos (/ y swap), debido a que la partición que se monta en el directorio **/home** se puede simplemente obviar, y los datos de los usuarios quedarían guardados en la misma partición que se monta en la raíz. A la partición del swap no se le asigna un directorio porque se usa para guardar ciertas replicas de la memoria RAM, para liberar y optimizar el espacio de la RAM, dándole la prioridad a tareas en primer plano, guardando las tareas de segundo plano en la partición de intercambio. En la actualidad las capacidades de las memorias RAM modernas hace que tener una partición de intercambio sea más por tradición que por alguna razón practica, sin embargo la partición de swap no se a llegado a eliminar, sino más bien el volumen de la partición se va reduciendo cada vez más.

3.1.1. Esquema de partición MBR

El **MBR** del acrónimo *Master Boot Record* y en español Registro de Arranque Maestro, es un tipo especial de sector de arranque ubicado al comienzo de las particiones de la computadora. El MBR contiene la información sobre cómo se dividen los sectores del disco en particiones, donde cada partición teóricamente contiene un sistema de archivos. El MBR tiene también código ejecutable con la función de cargar el SO instalado, usualmente pasando el control a la segunda etapa del cargador, o junto con el registro de arranque de volumen (VBR)⁵ de cada partición. Este código del MBR se conoce como cargador de arranque. La organización de la tabla de particiones en el MBR limita el espacio máximo direccionable de un disco particionado a 2TiB [Han, 2022].

En la práctica, el MBR se refiere al sector de arranque de 512 bytes. Que se divide de la siguiente forma: 446 bytes para el código de máquina para arrancar el SO o otros programas, 64 bytes que guarda la información de las particiones (solo se pueden definir 4 particiones primarias, porque solo hay 4 registros para las particiones), osea que para una partición solo hay 16 bytes en los que se guarda el

⁵Es un tipo de sector de arranque introducido por IBM. Se puede encontrarse en un dispositivo de almacenamiento de datos particionado, como un disco duro, o en un dispositivo no particionado como un disquete, y contiene código de máquina para programas de arranque almacenados en otras partes del dispositivo.

estado de la partición, el cilindro, cabezal, primer sector de la partición, el cilindro, cabezal, último sector de la partición, el tipo de la partición, el LBA⁶, y la longitud de la partición en sectores. Como últimos 2 bytes del MBR son su firma.

Address	Description		Size (bytes)
0x0000 (0)	Bootstrap code area		446
0x01BE (446)	Partition entry №1	Partition table (for primary partitions)	16
0x01CE (462)	Partition entry №2		16
0x01DE (478)	Partition entry №3		16
0x01EE (494)	Partition entry №4		16
0x01FE (510)	0x55	Boot signature ^[a]	2
0x01FF (511)	0xAA		
Total size: 446 + 4×16 + 2			512

⁶Logical Block Addressing es un método muy común usado para especificar la localización de los bloques de datos en los sistemas de almacenamiento, principalmente el almacenamiento secundario de una computadora. El término LBA puede referirse también a la «dirección del bloque» al que enlaza.

3.1.2. Tipos de particiones

El formato o sistema de archivos de las particiones no debe ser confundido con el tipo de partición ya que en realidad no tienen directamente mucho que ver. Independientemente del sistema de archivos de una partición (FAT, ext4, NTFS, etc.), existen 3 tipos diferentes de particiones [Lissot, 2005]:

- **Partición primaria:** Son las divisiones crudas o primarias del disco, solo puede haber 4 de éstas o 3 primarias y una extendida. Un disco físico completamente formateado consiste, en realidad, de una partición primaria que ocupa todo el espacio del disco y posee un sistema de archivos. A este tipo de particiones, prácticamente cualquier sistema operativo puede detectarlas y asignarles una unidad, siempre y cuando el sistema operativo reconozca su formato.
- **Partición extendida:** También conocida como partición secundaria es otro tipo de partición que actúa como una partición primaria, sirve para contener múltiples unidades lógicas en su interior. Fue ideada para romper la limitación de 4 particiones primarias en un solo disco físico. Solo puede existir una partición de este tipo por disco, y solo sirve para contener particiones lógicas. Por lo tanto, es el único tipo de partición que no soporta un sistema de archivos directamente.
- **Partición lógica:** Ocupa una porción de la partición extendida o la totalidad de la misma, la cual se ha formateado con un tipo específico de sistema de archivos y se le ha asignado una unidad, así el sistema operativo reconoce las particiones lógicas o su sistema de archivos. Puede haber un máximo de 23 particiones lógicas en una partición extendida. Linux impone un máximo de 15, incluyendo las 4 primarias, en discos SCSI y en discos IDE 8963.

3.1.3. Sistemas de archivos

Se puede definir el sistema de archivos de un sistema operativo como aquellas estructuras lógicas y sus correspondientes métodos que utiliza el propio sistema para organizar los archivos en disco. Un disco duro puede poseer una o más particiones, y cada una de esas particiones requieren de un formato lógico mediante un sistema de archivos específico. Estos sistemas de archivos estructuran la información para poder mostrarla, tanto de forma gráfica, como de forma textual mediante los gestores de archivos [Juncos, 2008].

El sistema de archivos debe permitir acceder de manera conocida a la información almacenada en la partición. Para ello, los sistemas de archivos GNU/Linux, poseen una estructura jerárquica o de "árbol". Es decir, el sistema contiene directorios, que a su vez podrían contener más subdirectorios, que asocian características de archivos con los archivos guardados en la partición.

La relación descrita entre los archivos y la forma de localizarlos se realiza, en los sistemas GNU/Linux, mediante una tabla de inodo. Un inodo contiene los parámetros característicos del objeto referenciado, permisos, fechas, ubicación, etc. Este objeto puede ser tanto un archivo, un directorio, un enlace simbólico, y por generalizar, cualquier objeto puede ser entendido por el sistema de archivos.

Cada inodo está identificado por un número entero único en el sistema de archivos, y los directorios son los responsables de guardar ternas de números de inodo y nombre identificativo del archivo. El inodo no guarda el nombre del archivo, no forma parte de su estructura. Así, cada archivo posee un único inodo, pero puede tener varios nombres que lo identifiquen.

Con esta estructura de sistema de archivos planteada, y desde punto de vista del usuario, para poder referenciar en archivo se puede utilizar la cadena de texto denominada "ruta". La ruta es el resultado de la concatenación de los nombres identificativos de directorios y subdirectorios que dibujan la estructura arbórea hasta llegar al archivo, más el nombre del propio archivo. Además, la sintaxis que sigue es estricta: la ruta empezará por el directorio superior, añadiendo a continuación los subdirectorios y por

último al final el nombre del archivo, y todo ello dividido por caracteres especiales, en el caso de los UNIX se usa el slash "/".

3.2. Kernel

Kernel es un software que constituye una parte fundamental en el SO, y se define como la parte que se ejecuta en modo privilegiado⁷ [Bellevue Linux Users Group, 2006]. Es la parte del SO que siempre está presente en la memoria, y es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. En muchos sistemas, el kernel es uno de los primeros programas en ser cargados en el arranque, después del gestor de arranque. Maneja el resto del inicio así como la memoria, los periféricos, y las peticiones de E/S del software, traduciendo en instrucciones de procesamiento de datos para la CPU.

El código crítico del kernel generalmente se carga en un área de la memoria separado, que está protegido contra el acceso por parte del software de aplicaciones u otras partes menos críticas del sistema operativo. El kernel realiza tareas como, ejecutar procesos, administrar el hardware como los discos o la CPU, y maneja las interrupciones, en este espacio protegido del kernel. En contraste, los programas como un navegador de internet, el reproductor de audios o videos, o un editor de texto usan un espacio en la memoria separado que es el espacio de los usuarios. Esta separación evita que los datos de los usuarios y del kernel interfieran entre sí y causen inestabilidad y lentitud, así como también evitar que las aplicaciones que funcionen mal afecten a otras aplicaciones o bloqueen todo el SO.

3.3. Arranque del sistema

En cada plataforma hay pequeñas diferencias en cuanto al proceso de arranque del sistema (boot), sin embargo los pasos siguientes que representan ese mismo proceso, son la generalidad de los sistemas UNIX. Al encender la computadora, la BIOS⁸ ejecuta la prueba de auto arranque o POST⁹ junto con el descubrimiento e inicialización de los dispositivos iniciales, ya que en el proceso de inicio del SO puede requerir el acceso a los discos, pantallas, teclados, entre otras. Como paso siguiente se lee el primer sector del disco de inicio, el MBR se coloca en una ubicación fija de memoria y se ejecuta. Este sector tiene un pequeño programa que carga un programa independiente llamado **boot** del dispositivo de inicio. El programa boot se copia a sí mismo en una dirección libre de la parte superior de la memoria, para liberar la parte inferior de la memoria para el sistema operativo.

Una vez que se traslada, boot lee el directorio raíz del dispositivo de inicio. Para ello tiene que comprender el sistema de archivos y el formato de los directorios, lo cual es el caso con algunos cargadores de inicio como GRUB. Después el programa boot lee el kernel del SO, lo coloca en memoria y salta a él. En este punto termina su trabajo y el kernel ya está en ejecución [Tanenbaum, 2003].

⁷En el modo privilegiado, la CPU puede realizar cualquier operación permitida por su arquitectura.

⁸BIOS es el sistema básico de entrada y salida (Basic Input Output System).

⁹POST, siglas en inglés de power-on self-test o autoprueba de arranque, es un proceso de verificación e inicialización de los componentes de entrada y salida en un sistema computacional que se encarga de configurar y diagnosticar el estado del hardware.

4. Configuración e instalación de los diferentes sistemas operativos

4.1. Sistema archlinux

4.1.1. Requerimientos mínimos, recomendados y usados

En los requerimientos mínimos para la configuración básica se necesita lo siguiente:

- Procesador con la arquitectura x86_64.
- 512MiB de memoria RAM.
- 2GiB de memoria de disco.

Como requerimientos recomendados serían los siguientes:

- Procesador con la arquitectura x86_64.
- 1024MiB de memoria RAM.
- 4GiB de memoria de disco.

Las especificaciones de hardware de la máquina virtual que se esta usando son las siguientes:

- Procesador con la arquitectura x86_64.
- 2048MiB de memoria RAM.
- 20GiB de memoria de disco.

4.1.2. Descripción del proceso básico de la instalación

Archlinux es una distribución de GNU/Linux dedicada a usuarios avanzados, por lo que su forma de instalar es totalmente manual. La instalación se inicia particionando el disco en dos particiones que serian la partición del directorio raíz y la partición de intercambio o swap. En el caso de la partición del directorio raíz es necesario montarlo en el directorio **/mnt** de la unidad booteable y activar la partición de intercambio. Cuando la partición de la raíz está montada y tiene sistema de archivos, se puede instalar los paquetes base del sistema con el kernel y el firmware. Siguiendo con la configuración del sistema, se hace el archivo **fstab** que guarda información sobre las particiones como sus sistema de archivos, y el punto de montaje, también define cómo deben montarse en el sistema de archivos las particiones de disco, además de varios otros dispositivos de bloque o sistemas de archivos remotos [Equipo archlinux, 2022a]. Después se hace una cambio de directorio raíz dentro del directorio **/mnt**. Dentro de lo que va a ser el sistema final se configura la zona horaria y se sincroniza el reloj del hardware desde el reloj del sistema. Se configura la distribución del teclado, editando los archivos **/etc/locale.gen**, y **/etc/vconsole.conf**. Se le asigna el hostname a la máquina, la contraseña del usuario root y se procede a la instalación del programa *bootloader* en este caso GRUB. GRUB está en los repositorios oficiales de archlinux, por lo que se descarga con el gestor de paquetes predeterminado, pacman. Luego de descargar el paquete GRUB, se tiene que instalar asignándole el dispositivo de almacenamiento. Posteriormente se configura el GRUB. En este punto el sistema ya está instalado en su configuración más básica, por lo que se tiene que salir del entorno que genera el cambio de raíz. Por último se desmontan todas las particiones que se encuentran en el directorio **/mnt**. Se termina reiniciando el sistema [Equipo archlinux, 2022b].

4.1.3. Pre-instalación

En el primer comando se configura la distribución de teclado en español.

- loadkeys es

```
Arch Linux 5.18.16-arch1-1 (tty1)
archiso login: root (automatic login)

To install Arch Linux follow the installation guide:
https://wiki.archlinux.org/title/Installation_guide

For Wi-Fi, authenticate to the wireless network using the iwctl utility.
For mobile broadband (WWAN) modems, connect with the mmcli utility.
Ethernet, WLAN and WWAN interfaces using DHCP should work automatically.

After connecting to the internet, the installation guide can be accessed
via the convenience script Installation_guide.

root@archiso ~ # loadkeys es
root@archiso ~ # _
```

Con la siguiente instrucción se comprueba la conexión a internet de la máquina. La cual al ser una máquina virtualizada no debería haber problemas de conexión. El comando ping se puede realizar a cualquier url.

- ping google.cl

```
root@archiso ~ # ping google.cl
PING google.cl (142.251.0.94) 56(84) bytes of data.
64 bytes from cj-in-f94.1e100.net (142.251.0.94): icmp_seq=1 ttl=63 time=17.4 ms
64 bytes from cj-in-f94.1e100.net (142.251.0.94): icmp_seq=2 ttl=63 time=17.8 ms
^C
--- google.cl ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 17.362/17.595/17.829/0.233 ms
root@archiso ~ # _
```

Para sincronizar el reloj del sistema con la zona horaria desde internet se utiliza el siguiente comando:

- timedate set-ntp true

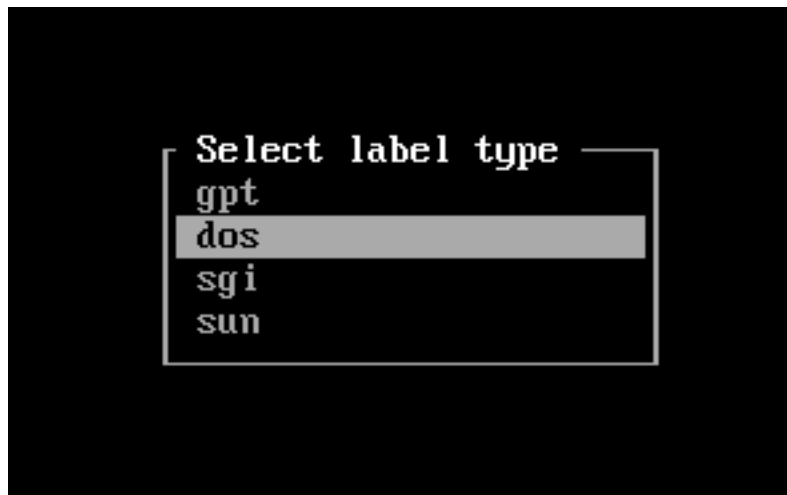
```
root@archiso ~ # timedatectl set-ntp true
root@archiso ~ # _
```

Ahora se particiona el disco con el comando:

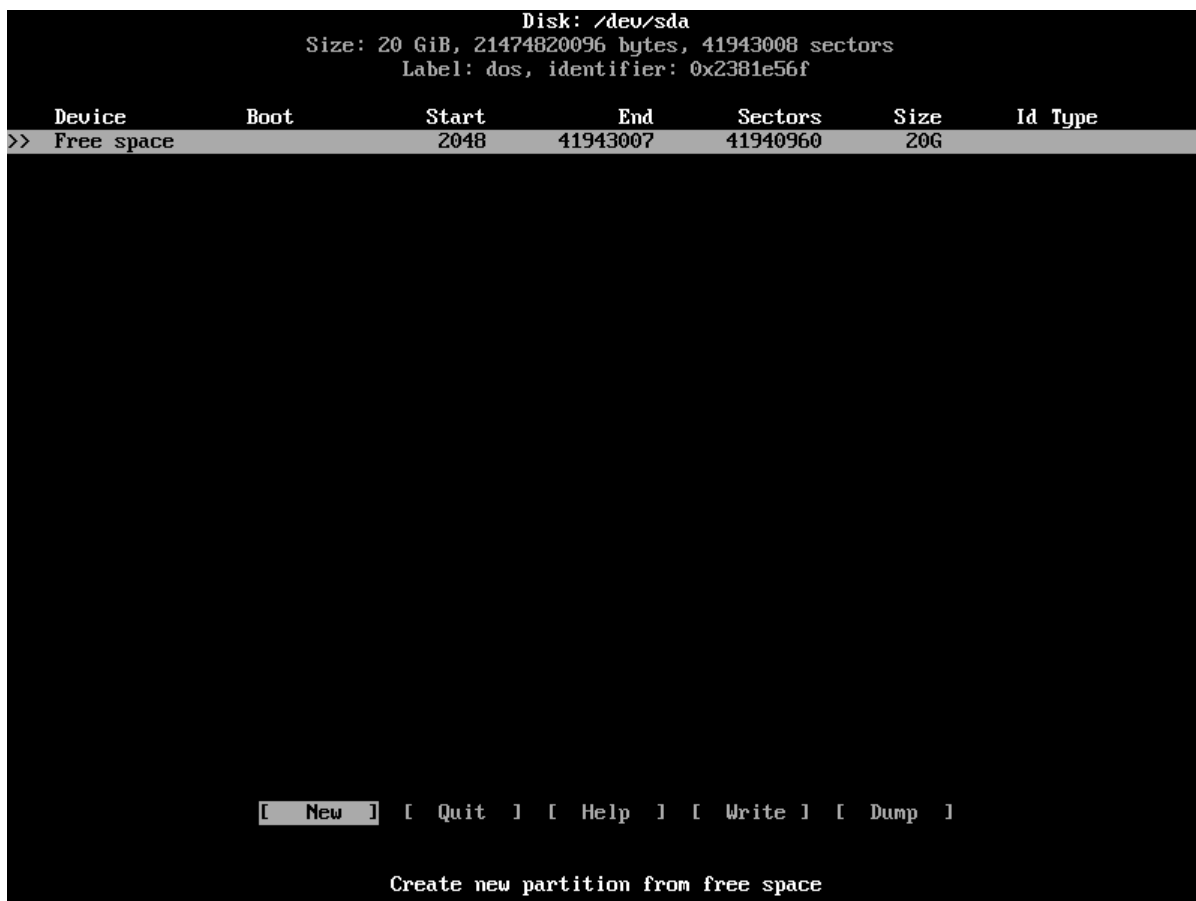
- cfdisk

```
root@archiso ~ # cfdisk_
```

Al ejecutar el comando anterior se muestra un menú de opciones para seleccionar el esquema de particiones. Se selecciona con las flechas del teclado el opción "dos".



Dentro de esa opción se despliega otro menú para manipular las particiones del dispositivo de almacenamiento. Como es una máquina virtual nueva no tiene particiones creadas.



En la parte inferior de la pantalla se ubican las opciones de operación cuando se selecciona el espacio libre. Como se necesitan 2 particiones mínimas se usa la opción "New", en donde se va a definir el volumen de la partición, y el tipo de la partición. La primera partición va a tener un volumen de 15GB.

```

Disk: /dev/sda
Size: 20 GiB, 21474820096 bytes, 41943008 sectors
Label: dos, identifier: 0x2381e56f

>>  Device      Boot      Start         End      Sectors      Size      Id Type
    Free space                2048         41943007      41940960      20G

Partition size: 15G_
```

Y va a ser de tipo primaria.

```

[ primary] [extended]
```

Esta es la partición que va a tener el MBR por lo que se marca con la opción "bootable".

```

Disk: /dev/sda
Size: 20 GiB, 21474820096 bytes, 41943008 sectors
Label: dos, identifier: 0x2381e56f

  Device      Boot      Start         End      Sectors      Size      Id Type
>> /dev/sda1                2048         31459327    31457280      15G       83 Linux
    Free space              31459328    41943007    10483680        5G

Partition type: Linux (83)

[Bootable] [ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ] [ Dump ]
```

Con el resto de espacio que queda en el disco se hace otra partición de la misma forma que la anterior, con la única diferencia de que no va a ser booteable porque va a ser la partición de intercambio, así que se tiene que cambiar su "type" a "Linux swap / Solaris".

```

81 Minix / old Linux
82 Linux swap / Solaris
83 Linux
```

Siguiendo todas las instrucciones anteriores el programa cfdisk se debe ver de la siguiente forma:

```

Disk: /dev/sda
Size: 20 GiB, 21474820096 bytes, 41943008 sectors
Label: dos, identifier: 0x2381e56f

  Device      Boot      Start         End      Sectors      Size      Id Type
>> /dev/sda1                2048         31459327    31457280      15G       83 Linux
    /dev/sda2              31459328    41943007    10483680        5G       82 Linux swap / Solaris

Partition type: Linux swap / Solaris (82)

[Bootable] [ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ] [ Dump ]

Changed type of partition 2.
```

Para terminar con el particionado en el programa cfdisk se selecciona la opción "write" y se escribe "yes" para confirmar el particionado, y se sale con la opción "quit".

Con el siguiente comando se listan las particiones que están en el sistema:

- lsblk

```
root@archiso ~ # lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0       7:0    0   680M  1 loop /run/archiso/airootfs
sda         8:0    0   20G   0 disk
├─sda1      8:1    0   15G   0 part
└─sda2      8:2    0    5G   0 part
sr0         11:0   1 786.3M  0 rom  /run/archiso/bootmnt
root@archiso ~ # _
```

Para configurar la partición de la raíz el sistema de archivos se hace con el siguiente comando:

- mkfs.ext4 /dev/sda1

```
root@archiso ~ # mkfs.ext4 /dev/sda1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 3932160 4k blocks and 983040 inodes
Filesystem UUID: 39884a8f-4d5e-4645-a63d-e1959564e776
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
root@archiso ~ #
```

Para crear la partición de intercambio se usa el siguiente comando:

- mkswap /dev/sda2

```
root@archiso ~ # mkswap /dev/sda2
Setting up swapspace version 1, size = 5 GiB (5367640064 bytes)
no label, UUID=53725e81-e2fa-4a38-bcc3-c31bb25b7cd6
root@archiso ~ # _
```

Para activar la partición de intercambio se usa el comando:

- swapon /dev/sda2

```
root@archiso ~ # swapon /dev/sda2
root@archiso ~ # _
```

Para montar la partición de la raíz se usa el comando:

- mount /dev/sda1 /mnt

```
root@archiso ~ # mount /dev/sda1 /mnt
root@archiso ~ #
```

4.1.4. Instalación

El comando para instalar los paquetes base del sistema, el kernel, y el firmware es el siguiente:

- `pacstrap /mnt base linux linux-firmware`

Si bien se pueden incluir otros paquetes en este mismo comando como un editor de texto, un navegador web o un gestor de archivos. Mantenerlo es su versión más fundamental ayuda al orden, y al entendimiento de cada instrucción.

```
root@archiso ~ # pacstrap /mnt base linux linux-firmware _
```

4.1.5. Configuración

Se usa el siguiente comando para generar el archivo `/etc/fstab`:

- `genfstab -U /mnt >> /mnt/etc/fstab`

```
root@archiso ~ # genfstab -U /mnt >> /mnt/etc/fstab
root@archiso ~ # _
```

Se hace un cambio de raíz con el comando:

- `arch-chroot /mnt`

```
root@archiso ~ # arch-chroot /mnt
[root@archiso /]# _
```

Se configura la zona horaria local en America/Santiago con el comando:

- `ln -sf /usr/share/zoneinfo/America/Santiago /etc/localtime`

```
[root@archiso /]# ln -sf /usr/share/zoneinfo/America/Santiago /etc/localtime
[root@archiso /]#
```

Para sincronizar el reloj del hardware desde el reloj del sistema se usa el comando:

- `hwclock --systohc`

```
[root@archiso /]# hwclock --systohc
[root@archiso /]#
```

Para editar el archivo `/etc/locale.gen` se necesita instalar un editor de texto que funcione en la terminal. Por lo que se usa el comando:

- `pacman -S vim`

```
[root@archiso /]# pacman -S vim _
```

Con el siguiente comando se puede editar el archivo `/etc/locale.gen` con el programa vim:

- `vim /etc/locale.gen`

```
[root@archiso /]# vim /etc/locale.gen _
```

Dentro del archivo **/etc/locale.gen** se tienen que descomentar las líneas que salen en la imagen, se guarda el archivo, y se sale.

```
#en_SG ISO-8859-1
en_US.UTF-8 UTF-8
#en_US ISO-8859-1
#en_ZA.UTF-8 UTF-8
#en_ZA ISO-8859-1
#en_ZM UTF-8
#en_ZW.UTF-8 UTF-8
#en_ZW ISO-8859-1
#eo UTF-8
#es_AR.UTF-8 UTF-8
#es_AR ISO-8859-1
#es_BO.UTF-8 UTF-8
#es_BO ISO-8859-1
es_CL.UTF-8 UTF-8
#es_CL ISO-8859-1
#es_CO.UTF-8 UTF-8
```

Con el siguiente comando se genera la configuración para la codificación de los caracteres dependiendo del archivo anteriormente editado:

- locale-gen

```
[root@archiso /]# locale-gen
Generating locales...
  en_US.UTF-8... done
  es_CL.UTF-8... done
Generation complete.
[root@archiso /]# _
```

Con el siguiente comando se crea y escribe en el archivo **/etc/locale.conf** la variable LANG:

- echo "LANG=es_CL.UTF-8"> /etc/locale.conf

El comando anterior es interesante porque en una sola línea se crea y escribe en el archivo, redirigiendo la salida estándar del comando **echo** al archivo.

```
[root@archiso /]# echo "LANG=es_CL.UTF-8" > /etc/locale.conf
[root@archiso /]# _
```

Con el siguiente comando de la misma forma que el anterior se crea y escribe en el archivo **/etc/vconsole.conf** la distribución del teclado.

- echo "KEYMAP=es"> /etc/vconsole.conf

```
[root@archiso /]# echo "KEYMAP=es" > /etc/vconsole.conf
[root@archiso /]# _
```

De la misma forma que los comandos anteriores se crea y escribe en el archivo **/etc/hostname**.

- echo "mellado"> /etc/hostname

```
[root@archiso /]# echo "mellado" > /etc/hostname
[root@archiso /]#
```

Con el siguiente comando se le asigna una contraseña al usuario root:

- passwd

```
[root@archiso /]# passwd
New password:
Retype new password:
passwd: password updated successfully
[root@archiso /]# _
```

Se descarga el bootloader, en este caso el programa GRUB:

- pacman -S grub

```
[root@archiso /]# pacman -S grub
resolving dependencies...
looking for conflicting packages...

Packages (1) grub-2:2.06.r322.gd9b4638c5-4

Total Download Size:    6.78 MiB
Total Installed Size:  33.02 MiB

:: Proceed with installation? [Y/n]
```

Se instala el bootloader:

- grub-install --target=i386-pc /dev/sda

```
[root@archiso /]# grub-install --target=i386-pc /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
[root@archiso /]#
```

Se genera la configuración del bootloader:

- grub-mkconfig -o /dev/sda

```
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot:  initramfs-linux-fallback.img
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
[root@archiso /]#
```


Instalación del servicio de gestión de redes:

- `pacman -S networkmanager`

```
[root@archiso /]# pacman -S networkmanager
resolving dependencies...
looking for conflicting packages...

Packages (1) networkmanager-1.40.0-1

Total Installed Size: 16.38 MiB

:: Proceed with installation? [Y/n]
```

Activación del servicio de gestión de redes:

- `systemctl enable NetworkManager`

```
[root@archiso /]# systemctl enable NetworkManager
Created symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service → /usr/lib/systemd/system/NetworkManager.service.
Created symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service → /usr/lib/systemd/system/NetworkManager-dispatcher.service.
Created symlink /etc/systemd/system/network-online.target.wants/NetworkManager-wait-online.service → /usr/lib/systemd/system/NetworkManager-wait-online.service.
[root@archiso /]#
```

4.1.6. Administración de usuarios

Creación de un usuario con directorio home:

- `useradd -m oscar`

```
[root@archiso /]# useradd -m oscar
[root@archiso /]#
```

Asignación de una contraseña a un usuario:

- `passwd oscar`

```
[root@archiso /]# passwd oscar
New password:
Retype new password:
passwd: password updated successfully
[root@archiso /]#
```

Configuración de grupos de un usuario con permisos de administrador, audio, video, y almacenamiento:

- `usermod -aG wheel,audio,video,storage oscar`

```
[root@archiso /]# usermod -aG wheel,audio,video,storage oscar
[root@archiso /]# _
```

Instalación del paquete sudo:

- `pacman -S sudo`

```
[root@archiso /]# pacman -S sudo
resolving dependencies...
looking for conflicting packages...

Packages (1) sudo-1.9.11.p3-1

Total Download Size:   1.69 MiB
Total Installed Size:  6.98 MiB

:: Proceed with installation? [Y/n] _
```

Configuración de archivo `/etc/sudoers`, para el grupo wheel. Se descomenta la línea de la imagen siguiente:

```
##
## User privilege specification
##
root ALL=(ALL:ALL) ALL

## Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL:ALL) ALL
```

Se sale del entorno del cambio de raíz:

- `exit`

```
[root@archiso /]# exit
exit
arch-chroot /mnt  46.51s user 18.99s system 0% cpu 2:48:43.55 total
root@archiso ~ #
```

Se desmontan las particiones que están en el directorio `/mnt`:

- `umount -R /mnt`

```
root@archiso ~ # umount -R /mnt
root@archiso ~ # _
```

Por último se reinicia el sistema con el comando **reboot**.

4.1.7. Instalación de la interfaz gráfica

Cuando se reinicia la máquina y se inicia en el sistema instalado, se autentica el usuario para seguir con la instalación de la interfaz gráfica. Se parte la instalando el X server, junto con el git, el paquete base-devel que trae distintos programas para la compilación de aplicaciones, el paquete make que también trae otro programas para la compilación de aplicaciones, la terminal alacritty, el lanzador de aplicaciones dmenu y el administrador de tareas htop.

- `sudo pacman -S xorg git base-devel make alacritty dmenu htop`

```
[oscar@mellado ~]$ sudo pacman -S xorg git base-devel make alacritty dmenu htop
```

Se crea un directorio oculto llamado **.config** en el que se van a guardar todas las configuraciones del usuario.

- `mkdir .config`

```
[oscar@mellado ~]$ mkdir .config
[oscar@mellado ~]$
```

Dentro del directorio **.config/** se copia el repositorio de el gestor de ventanas dwm, que es minimalista, y para configurarlo se tiene que editar el código fuente para después compilarlo.

- `git clone https://git.suckless.org/dwm`

```
[oscar@mellado .config]$ git clone https://git.suckless.org/dwm
Clonando en 'dwm'...
remote: Enumerating objects: 6481, done.
remote: Counting objects: 100% (6481/6481), done.
remote: Compressing objects: 100% (3193/3193), done.
remote: Total 6481 (delta 3730), reused 5922 (delta 3287), pack-reused 0
Recibiendo objetos: 100% (6481/6481), 6.06 MiB | 1.99 MiB/s, listo.
Resolviendo deltas: 100% (3730/3730), listo.
[oscar@mellado .config]$
```

Dentro del directorio **dwm/** que se creó se va a editar el archivo `config.def.h`, y se va a eliminar la línea seleccionada en la imagen.

```
static const char *dmenucmd[] = { "dmenu_run", "-fn", dmenufont, "-nb", col_gray1, "-nf", col_gray3,
"-sb", col_cyan, "-sf", col_gray4, NULL };
static const char *termcmd[] = { "st", NULL };

static const Key keys[] = {
    /* modifier      key          function        argument */
    { MODKEY,         XK_p,        spawn,          {.u = dmenucmd } },
    { MODKEY|ShiftMask, XK_Return,    spawn,          {.u = termcmd } },
    { MODKEY,         XK_b,        togglebar,      {0} },

```

También se va a modificar la siguiente línea.

```
{ MODKEY,           XK_Return, zoom,           {0} },
```

A la siguiente:

```
{ MODKEY,           XK_Return, spawn,          SHCMD("alacritty") },
```

Después de editar el archivo config.def.h se compila el gestor de ventanas.

- sudo make clean install

```
[oscar@mellado dwm]$ sudo make clean install_
```

Se crea el directorio **/usr/share/xsessions**.

- sudo mkdir /usr/share/xsessions

```
[oscar@mellado ~]$ sudo mkdir /usr/share/xsessions
[oscar@mellado ~]$ _
```

Se edita el archivo **/usr/share/xsessions/dwm.desktop** con el siguiente contenido.

```
[Desktop Entry]
Encoding=UTF-8
Name=Dwm
Comment=Dynamic window manager
Exec=dwm
Icon=dwm
Type=XSession
```

Se instala el gestor de inicio de sesión.

- sudo pacman -S lightdm lightdm-gtk-greeter

```
[oscar@mellado ~]$ sudo pacman -S lightdm lightdm-gtk-greeter_
```

Se inicia el servicio del gestor de inicio de sesión.

- sudo systemctl enable lightdm

```
[oscar@mellado ~]$ sudo systemctl enable lightdm
Created symlink /etc/systemd/system/display-manager.service → /usr/lib/systemd/system/lightdm.service.
[oscar@mellado ~]$ _
```

La interfaz gráfica está instalada solo queda reiniciar la máquina.

4.1.8. Administrador de tareas

El administrador de tareas se ejecuta desde una terminal con el comando **htop**.

```
1 2 3 4 5 6 7 8 9 []= Alacritty dwm-6.3

CPU[|||||] 17.0% Tasks: 16, 14 thr, 69 kthr; 1 running
Mem[|||||] 217M/1.93G Load average: 0.00 0.06 0.07
Swp[ ] 0K/5.00G Uptime: 00:10:25

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
352 oscar 20 0 673M 112M 50908 S 10.5 5.7 0:05.37 alacritty
376 oscar 20 0 10764 4388 3596 R 1.3 0.2 0:00.24 htop
286 root 20 0 286M 59364 37296 S 0.7 2.9 0:02.55 /usr/lib/Xorg :0 -sea
1 root 20 0 98M 11508 8996 S 0.0 0.6 0:02.76 /sbin/init
181 root 20 0 48156 18352 15356 S 0.0 0.9 0:00.74 /usr/lib/systemd/syst
194 root 20 0 30980 8648 7400 S 0.0 0.4 0:00.55 /usr/lib/systemd/syst
224 dbus 20 0 8632 4876 4076 S 0.0 0.2 0:00.55 /usr/bin/dbus-daemon
225 root 20 0 24004 7940 6952 S 0.0 0.4 0:00.46 /usr/lib/systemd/syst
226 root 20 0 326M 23916 17084 S 0.0 1.2 0:00.80 /usr/bin/NetworkManag
227 root 20 0 326M 23916 17084 S 0.0 1.2 0:00.03 /usr/bin/NetworkManag
228 root 20 0 326M 23916 17084 S 0.0 1.2 0:00.09 /usr/bin/NetworkManag
232 root 20 0 298M 7448 6332 S 0.0 0.4 0:00.15 /usr/bin/lightdm
235 root 20 0 298M 7448 6332 S 0.0 0.4 0:00.00 /usr/bin/lightdm
238 root 20 0 298M 7448 6332 S 0.0 0.4 0:00.02 /usr/bin/lightdm
292 root 20 0 286M 59364 37296 S 0.0 2.9 0:00.11 /usr/lib/Xorg :0 -sea
331 root 20 0 158M 9380 8100 S 0.0 0.5 0:00.18 lightdm --session-chi
336 root 20 0 158M 9380 8100 S 0.0 0.5 0:00.00 lightdm --session-chi
337 root 20 0 158M 9380 8100 S 0.0 0.5 0:00.00 lightdm --session-chi
340 oscar 20 0 17652 10288 8596 S 0.0 0.5 0:00.75 /usr/lib/systemd/syst
341 oscar 20 0 22076 2856 0 S 0.0 0.1 0:00.00 (sd-pam)
347 oscar 20 0 17824 8416 7172 S 0.0 0.4 0:00.21 dwm
351 oscar 20 0 8176 3976 3600 S 0.0 0.2 0:00.02 /usr/bin/dbus-daemon
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

El administrador de tareas tiene la función de listar, y matar los procesos que estén en ejecución dentro del sistema. Si bien desde un programa como **htop** se puede perfectamente hacer la gestión de los procesos, también está la opción de usar los comando **ps** para listar los procesos con sus pid y el comando **kill -9** para detener la ejecución del proceso con el pid que se ingresa por parámetro.

4.1.9. Controlador de dispositivos

Un controlador de dispositivos es un programa que se encarga de controlar un dispositivo de hardware en específico. Los controladores de dispositivos son necesarios para el correcto funcionamiento del sistema. Lo que hace es conectar un dispositivo externo con el núcleo del sistema. Con el comando **lsmod** se listan todos los controladores que el sistema tiene cargados [Gillis, 2020].

```
1 2 3 4 5 6 7 8 9 []= Alacritty dwm-6.3
[oscar@mellado ~]$ lsmod
Module                Size  Used by
intel_rapl_msr         20480  0
intel_rapl_common      32768  1 intel_rapl_msr
snd_intel8x0           53248  0
snd_ac97_codec         180224 1 snd_intel8x0
crct10dif_pclmul       16384  1
crc32_pclmul           16384  0
ac97_bus               16384  1 snd_ac97_codec
ghash_clmulni_intel    16384  0
snd_pcm                172032 2 snd_intel8x0,snd_ac97_codec
aesni_intel            380928 0
vmwgfx                 393216 2
snd_timer              49152  1 snd_pcm
crypto_simd            16384  1 aesni_intel
joydev                 28672  0
snd                    126976 4 snd_intel8x0,snd_timer,snd_ac97_codec,snd_pcm
cryptd                 28672  2 crypto_simd,ghash_clmulni_intel
drm_ttm_helper         16384  1 vmwgfx
psmouse               212992 0
mousedev              24576  0
intel_agp              24576  0
e1000                 172032 0
rfkill                32768  1
ttm                   94208  2 vmwgfx,drm_ttm_helper
vboxguest             49152  0
pcspkr                16384  0
soundcore             16384  1 snd
i2c_piix4             36864  0
intel_gtt             28672  1 intel_agp
```

4.1.10. Interfaz de comandos

La interfaz de comando o interprete de comando es una aplicación para la interacción con el SO. Es un programa en el cual se pueden ejecutar comandos del sistema o lanzar aplicaciones de interfaz gráfica. En el caso del sistema archlinux, se instaló el emulador de terminal alacritty, y de manera predeterminada la shell es el programa intérprete bash.

```
1 2 3 4 5 6 7 8 9 []= Alacritty dwm-6.3
[oscar@mellado ~]$ echo $SHELL
/bin/bash
[oscar@mellado ~]$
```

4.1.11. Visor de sucesos

El visor de sucesos son los reportes que entrega el sistema para saber del estado del mismo sistema. En el caso de archlinux los reportes de todos los programas se encuentran en el directorio **/var/log**, y cada log se puede ver con el comando **cat** o el comando **tail -f** para que actualice la salida cuando ocurran cambios.

```
1 2 3 4 5 6 7 8 9 []= Alacritty dwm-6.3
[oscar@mellado ~]$ tail -f /var/log/Xorg.0.log
[ 19.937] (**) ImExPS/2 Generic Explorer Mouse: (accel) acceleration factor: 2.000
[ 19.937] (**) ImExPS/2 Generic Explorer Mouse: (accel) acceleration threshold: 4
[ 19.942] (II) event7 - ImExPS/2 Generic Explorer Mouse: is tagged by udev as: Mouse
[ 19.943] (II) event7 - ImExPS/2 Generic Explorer Mouse: device is a pointer
[ 19.947] (II) config/udev: Adding input device ImExPS/2 Generic Explorer Mouse (/dev
/input/mouse2)
[ 19.947] (II) No input driver specified, ignoring this device.
[ 19.947] (II) This device may have been added with another device file.
[ 19.949] (II) config/udev: Adding input device PC Speaker (/dev/input/event6)
[ 19.949] (II) No input driver specified, ignoring this device.
[ 19.949] (II) This device may have been added with another device file.
```

4.1.12. Directorios importantes del sistema

Dentro del sistema, se encuentran distintos directorios, pero los más importantes están ubicados en el directorio raíz **/**. En donde se haya el directorio **/boot**, que guarda la configuración del programa bootloader, necesario para el arranque del sistema. El directorio **/bin** que es donde se guardan todos los binarios del sistema que los usuarios pueden ejecutar. En el directorio **/sbin** se guardan los binarios que solo los puede usar el administrador del sistema. El directorio **/etc** es en donde se guardan las configuraciones de los programas instalados en el sistema.

```
[oscar@mellado /]$ ls -l
total 52
lrwxrwxrwx 1 root root 7 dic 6 2021 bin -> usr/bin
drwxr-xr-x 3 root root 4096 sep 5 21:39 boot
drwxr-xr-x 19 root root 3740 sep 6 14:28 dev
drwxr-xr-x 48 root root 4096 sep 6 14:28 etc
drwxr-xr-x 3 root root 4096 sep 5 22:05 home
lrwxrwxrwx 1 root root 7 dic 6 2021 lib -> usr/lib
lrwxrwxrwx 1 root root 7 dic 6 2021 lib64 -> usr/lib
drwx----- 2 root root 16384 sep 5 18:52 lost+found
drwxr-xr-x 2 root root 4096 dic 6 2021 mnt
drwxr-xr-x 2 root root 4096 dic 6 2021 opt
dr-xr-xr-x 146 root root 0 sep 6 14:28 proc
drwxr-x--- 4 root root 4096 sep 5 23:39 root
drwxr-xr-x 18 root root 440 sep 6 14:28 run
lrwxrwxrwx 1 root root 7 dic 6 2021 sbin -> usr/bin
drwxr-xr-x 4 root root 4096 sep 5 19:16 srv
dr-xr-xr-x 13 root root 0 sep 6 14:28 sys
drwxrwxrwt 7 root root 160 sep 6 14:29 tmp
drwxr-xr-x 10 root root 4096 sep 5 23:43 usr
drwxr-xr-x 12 root root 4096 sep 5 23:52 var
[oscar@mellado /]$
```

4.2. Sistema openBSD

4.2.1. Requerimientos mínimos, recomendados y usados

En los requerimientos mínimos para el sistema se necesita lo siguiente:

- Procesador con la arquitectura x86_64.
- 256MiB de memoria RAM
- 4GiB de memoria de disco

Como requerimientos recomendados son los siguientes:

- Procesador con la arquitectura x86_64.
- 512MiB de memoria RAM
- 8GiB de memoria de disco

En la máquina donde se instaló se usaron los siguientes requerimientos:

- Procesador con la arquitectura x86_64.
- 512MiB de memoria RAM
- 4GiB de memoria de disco

4.2.2. Descripción del proceso básico de instalación

OpenBSD es un fork del proyecto BSD orientado a la seguridad informática. A diferencia de el sistema archlinux, openBSD trae un instalador que le pregunta a usuario sobre las especificaciones la instalación y configuración del sistema. El instalador inicia preguntando el modo de la instalación, siguiendo con la distribución del teclado, datos de la máquina, configuración de red. Se configura la contraseña del usuario root, y la activación de servicios como el servidor ssh, el X server, y la interfaz gráfica. Pregunta sobre la creación de un usuario con menos privilegios dentro del sistema, se configura la zona horaria. Luego hace preguntas sobre el dispositivo de almacenamiento, el esquema de particiones, y la forma en la que se va a particionar el disco. Se instala el sistema seleccionando los sets de paquetes para luego confirmar la verificación de los mismo. Se corrobora el reloj en el sistema y se reinicia la máquina.

4.2.3. Pre-instalación

En la primera pregunta se solicita el modo de instalación. Se va a seleccionar la opción "install".

```
Welcome to the OpenBSD/amd64 7.1 installation program.  
(I)nstall, (U)pgrade, (A)utoinstall or (S)hell? i_
```

En la siguiente pregunta se selecciona la opción "la", que es la distribución de teclado correspondiente a latino america.

```
Choose your keyboard layout ('?' or 'L' for list) [default] la
```

Se define el hostname de la máquina.

```
System hostname? (short form, e.g. 'foo') mellado_
```

Se selecciona la interfaz de red para configurarla.

```
Which network interface do you wish to configure? (or 'done') [em0]
```

En la configuración de la dirección ipv4 se selecciona la opción "autoconf".

```
IPv4 address for em0? (or 'autoconf' or 'none') [autoconf]
```

En la configuración de la dirección ipv6 se selecciona la opción "none", porque para esta instalación no va a ser necesaria.

```
IPv6 address for em0? (or 'autoconf' or 'none') [none]
```

Vuelve a aparecer la pregunta anterior sobre seleccionar la interfaz red, en este caso se selecciona la opción "done", porque ya se termino de configurar la interfaz de red.

```
Which network interface do you wish to configure? (or 'done') [done]
```

Se le asigna la contraseña al usuario root:

```
Password for root account? (will not echo)  
Password for root account? (again)
```

Se desactiva el servicio de sshd, porque para esta instalación no va a ser necesario.

```
Start sshd(8) by default? [yes] no
```

Se activa el servicio del X server, para después instalar interfaz gráfica.

```
Do you expect to run the X Window System? [yes]
```

Se activa el servicio del gestor de sesiones que trae por defecto el sistema.

```
Do you want the X Window System to be started by xenodm(1)? [no] yes
```

En las opciones de los usuarios se selecciona la opción "no", porque posteriormente se va a crear un usuario de forma manual.

```
Setup a user? (enter a lower-case loginname, or 'no') [no]
```

Se configura la zona horaria en America/Santiago.

```
What timezone are you in? ('?' for list) [America/Santiago] _
```

Se selecciona el dispositivo de almacenamiento.

```
Available disks are: wd0.  
Which disk is the root disk? ('?' for details) [wd0] _
```

Se selecciona el esquema de particiones, en este caso se usa el esquema MBR.

```
Use (W)hole disk MBR, whole disk (G)PT or (E)dit? [whole] _
```

Se selecciona la opción "a", para que haga las particiones de forma automática. También en esta pregunta se muestran las particiones que se van a crear con la opción automática.

```
The auto-allocated layout for wd0 is:  
#          size      offset  fstype  [fsize  bsize   cppl  
a:         883.0M          64   4.2BSD   2048 16384    1 # /  
b:         246.0M     1808416    swap  
c:        4096.0M          0   unused  
d:        2594.8M     2312224   4.2BSD   2048 16384    1 # /usr  
e:         372.2M     7626304   4.2BSD   2048 16384    1 # /home  
Use (A)uto layout, (E)dit auto layout, or create (C)ustom layout? [a] _
```

4.2.4. Instalación

En esta parte se va a instalar el sistema, seleccionando los sets de paquetes para ello. Se instala el kernel junto con otros paquetes que sirven para la compilación de programas, manuales de uso de algunos de los comandos, y en general aplicaciones base del sistema [openbsdhandbook,]. Después pregunta si se continúa la instalación sin la verificación de la autenticidad de los paquetes a lo que se responde con la opción "yes", porque al ser una máquina virtual de prueba y no un sistema que vaya a estar en producción, no hay necesidad de verificar la autenticidad de los paquetes.

```
Let's install the sets!  
Location of sets? (cd0 disk http nfs or 'done') [cd0]  
Pathname to the sets? (or 'done') [/7.1/amd64]  
  
Select sets by entering a set name, a file name pattern or 'all'. De-select  
sets by prepending a '-', e.g.: '-game*'. Selected sets are labelled '[X]'.  
[X] bsd          [X] comp71.tgz      [X] xbase71.tgz   [X] xserv71.tgz  
[X] bsd.rd       [X] man71.tgz       [X] xshare71.tgz  
[X] base71.tgz   [X] game71.tgz      [X] xfont71.tgz  
Set name(s)? (or 'abort' or 'done') [done]  
Directory does not contain SHA256.sig. Continue without verification? [no] yes
```

Cuando se termina la instalación del sistema se vuelve a repetir una de las preguntas anteriores, a la cual se escribe la opción "done".

```
Location of sets? (cd0 disk http nfs or 'done') [done] _
```

A continuación pregunta sobre si la zona horaria está correcta a lo que se responde con la opción "yes".

```
Time appears wrong. Set to 'Tue Sep 6 21:52:37 -03 2022'? [yes]
```

El sistema ya ha sido instalado correctamente. Se reinicia la máquina.

```
CONGRATULATIONS! Your OpenBSD install has been successfully completed!

When you login to your new system the first time, please read your mail
using the 'mail' command.

Exit to (S)hell, (H)alt or (R)eboot? [reboot]
```

4.2.5. Creación de un usuario con permisos de administrador

De forma parecida a la creación de usuarios en el sistema archlinux. Se hace en el sistema openBSD. Se usan los mismo comando, diferenciándose en la parametrización del comando **usermod**, y exigiendo una longitud mínima en la contraseña del usuario.

Con los siguientes comando se crea un usuario con directorio home, y con el grupo wheel.

- `useradd -m oscar`
- `usermod -G wheel oscar`

```
mellado# useradd -m oscar
mellado# usermod -G wheel oscar
mellado# █
```

Se le asigna una contraseña al usuario con el comando:

- `passwd oscar`

```
mellado# passwd oscar
Changing password for oscar.
New password:
Retype new password:
passwd: password updated successfully
mellado# █
```

De manera predeterminada el sistema openBSD viene con la implementación del programa **doas**, que comparando lo con el sistema archlinux sería el equivalente al programa **sudo**, no obstante al igual que el arch, en el openBSD se puede instalar y configurar el comando **sudo**.

El comando **doas** se configura de la siguiente forma:

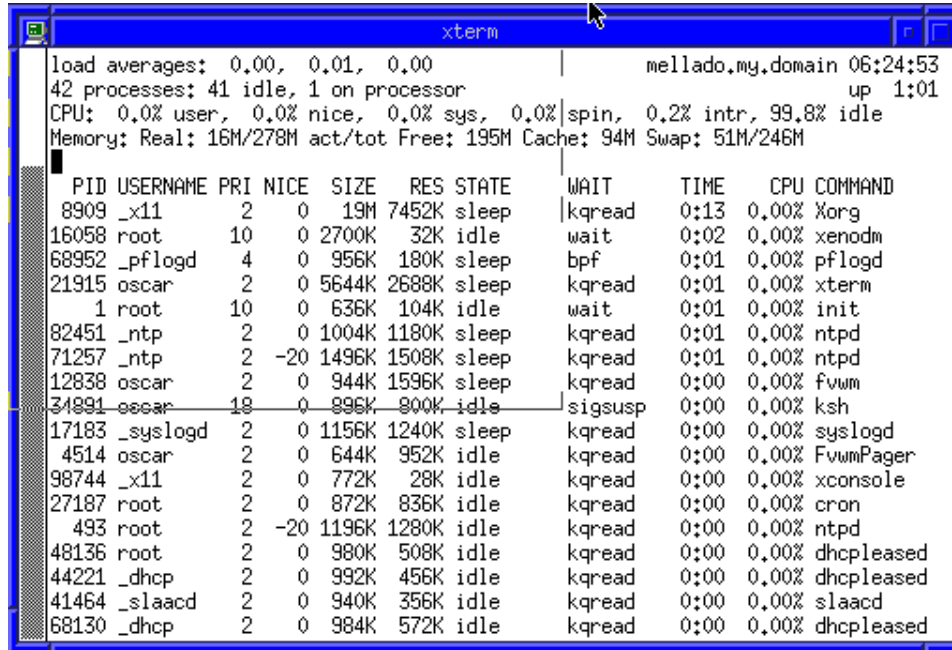
- `echo "permit :wheel"> /etc/doas.conf`

En el archivo **/etc/doas.conf** están las normas de configuración relativas a los permisos de los grupos. En el comando anterior se está permitiendo que el grupo wheel pueda usar el comando **doas**.

```
mellado# echo "permit :wheel" > /etc/doas.conf
mellado# █
```

4.2.6. Administrador de tareas

En el sistema openBSD el administrador de tareas se ejecuta con el comando **top**.



```
xterm
load averages: 0.00, 0.01, 0.00      mellado.my.domain 06:24:53
42 processes: 41 idle, 1 on processor      up 1:01
CPU: 0.0% user, 0.0% nice, 0.0% sys, 0.0% spin, 0.2% intr, 99.8% idle
Memory: Real: 16M/278M act/tot Free: 195M Cache: 94M Swap: 51M/246M

  PID USERNAME PRI NICE  SIZE  RES  STATE  WAIT  TIME  CPU COMMAND
  8909  _x11      2   0   19M 7452K sleep  kqread 0:13 0.00% Xorg
 16058  root      10   0  2700K  32K idle   wait    0:02 0.00% xenodm
 68952  _pflogd   4   0   956K  180K sleep  bpf     0:01 0.00% pflogd
 21915  oscar     2   0  5644K 2688K sleep  kqread  0:01 0.00% xterm
    1 root      10   0   636K  104K idle   wait    0:01 0.00% init
 82451  _ntp      2   0  1004K 1180K sleep  kqread  0:01 0.00% ntpd
 71257  _ntp      2  -20 1496K 1508K sleep  kqread  0:01 0.00% ntpd
 12838  oscar     2   0   944K 1596K sleep  kqread  0:00 0.00% fvwm
 34891  oscar    18   0   896K  800K idle   sigsusp 0:00 0.00% ksh
 17183  _syslogd  2   0  1156K 1240K sleep  kqread  0:00 0.00% syslogd
  4514  oscar     2   0   644K  952K idle   kqread  0:00 0.00% FvwmPager
 98744  _x11      2   0   772K  28K idle   kqread  0:00 0.00% xconsole
 27187  root      2   0   872K  836K idle   kqread  0:00 0.00% cron
  493 root      2  -20 1196K 1280K idle   kqread  0:00 0.00% ntpd
 48136  root      2   0   980K  508K idle   kqread  0:00 0.00% dhcpleased
 44221  _dhcp     2   0   992K  456K idle   kqread  0:00 0.00% dhcpleased
 41464  _slaacd   2   0   940K  356K idle   kqread  0:00 0.00% slaacd
 68130  _dhcp     2   0   984K  572K idle   kqread  0:00 0.00% dhcpleased
```

Al igual que en arch, en openBSD se pueden usar los comando **ps**, y **kill** para el manejo de los procesos.

4.2.7. Controlador de dispositivos

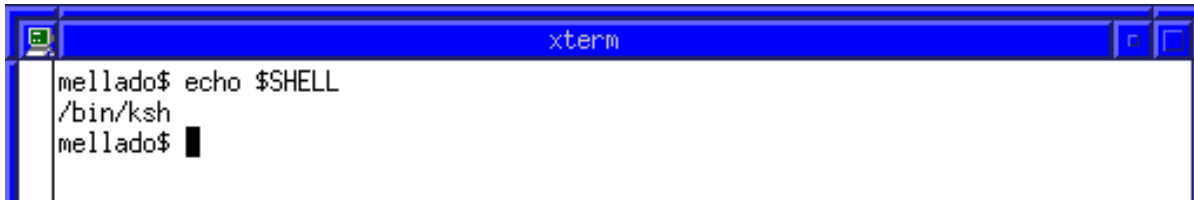
Por medio del comando **dmesg** (diagnostic messages), se muestran los mensajes que están en el buffer del kernel al momento de ejecutar lo. En el output del comando también se encuentran los mensajes de los controladores de dispositivos que están cargados en el sistema.



```
xterm
mellado$ dmesg
OpenBSD 7.1 (GENERIC) #443: Mon Apr 11 17:55:15 MDT 2022
  deraadt@amd64.openbsd.org:/usr/src/sys/arch/amd64/compile/GENERIC
real mem = 520028160 (495MB)
avail mem = 487141376 (464MB)
random: good seed from bootblocks
mpath0 at root
scsibus0 at mpath0: 256 targets
mainbus0 at root
bios0 at mainbus0: SMBIOS rev. 2.5 @ 0xe1000 (10 entries)
bios0: vendor innotek GmbH version "VirtualBox" date 12/01/2006
bios0: innotek GmbH VirtualBox
acpi0 at bios0: ACPI 4.0
acpi0: sleep states S0 S5
acpi0: tables DSDT FACP APIC SSDT
acpi0: wakeup devices
acpitimer0 at acpi0: 3579545 Hz, 32 bits
acpinadt0 at acpi0 addr 0xfef00000: PC-AT compat
cpu0 at mainbus0: apid 0 (boot processor)
cpu0: AMD Ryzen 9 5900HS with Radeon Graphics, 3305.76 MHz, 19-50-00
cpu0: FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE36,CF
LUSH,MMX,FXSR,SSE,SSE2,HTT,SSE3,PCLMUL,MWAIT,SSSE3,CX16,SSE4.1,SSE4.2,MOVBE,POPC
NT,AES,XSAVE,AVX,RDRAND,NXE,MMX,FXSR,RTSC,LONG,LAHF,AMCR8,ABM,SSE4A,MASSSE,3D
NOWP,ITSC,FSGSBASE,AVX2,INVPCID,RDSEED,CLFLUSHOPT,SSBDNR
```

4.2.8. Interfaz de comandos

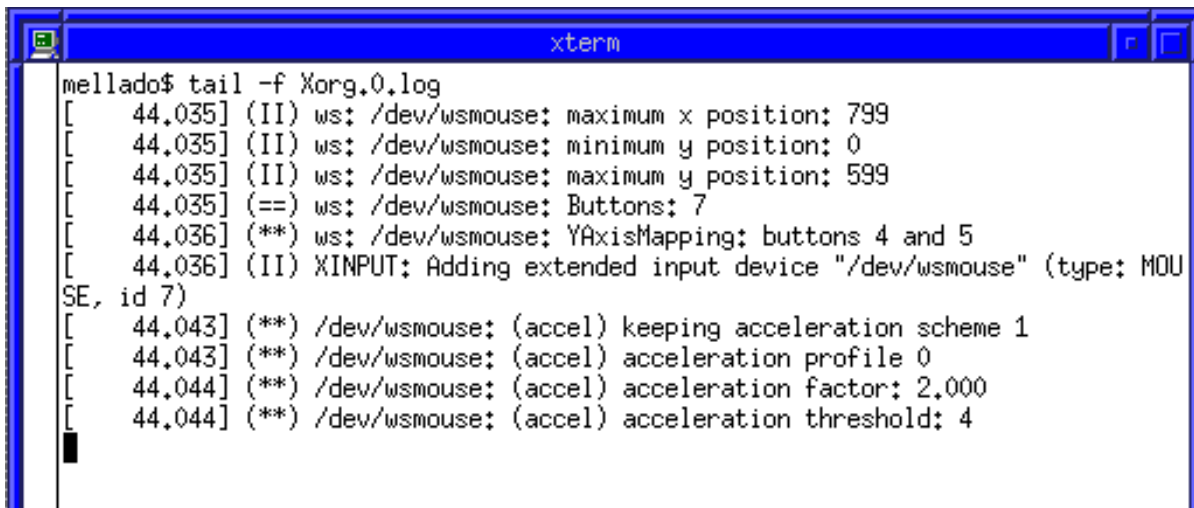
De igual forma que en el arch, en el sistema openBSD existe una terminal, además de un intérprete de comandos. En openBSD el emulador de terminal es el programa xterm, y el intérprete de comando es **ksh**.



```
xterm
mellado$ echo $SHELL
/bin/ksh
mellado$
```

4.2.9. Visor de sucesos

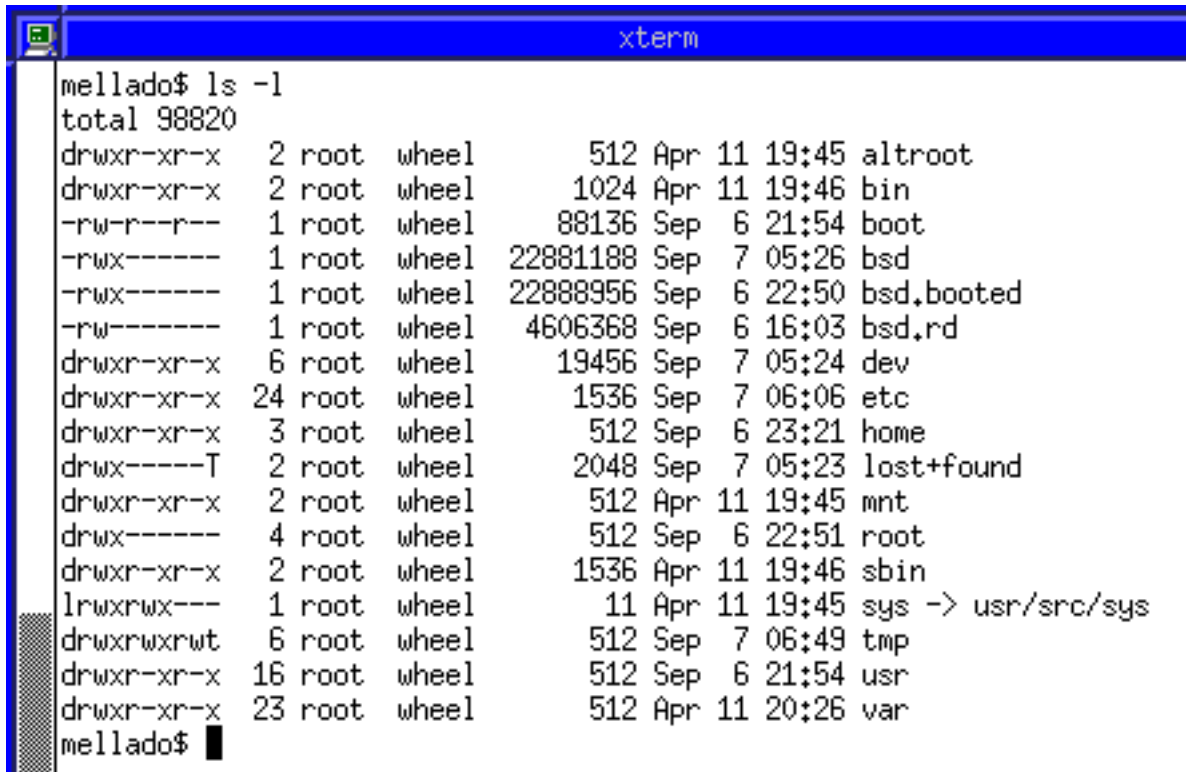
De la misma forma que en archlinux los reportes de sucesos se encuentran en el directorio **/var/log**, y al igual que en arch, se pueden leer los archivos con el comando **tail -f**.



```
xterm
mellado$ tail -f Xorg.0.log
[ 44.035] (II) ws: /dev/wsmouse: maximum x position: 799
[ 44.035] (II) ws: /dev/wsmouse: minimum y position: 0
[ 44.035] (II) ws: /dev/wsmouse: maximum y position: 599
[ 44.035] (==) ws: /dev/wsmouse: Buttons: 7
[ 44.036] (**) ws: /dev/wsmouse: YAxisMapping: buttons 4 and 5
[ 44.036] (II) XINPUT: Adding extended input device "/dev/wsmouse" (type: MOUSE, id 7)
[ 44.043] (**) /dev/wsmouse: (accel) keeping acceleration scheme 1
[ 44.043] (**) /dev/wsmouse: (accel) acceleration profile 0
[ 44.044] (**) /dev/wsmouse: (accel) acceleration factor: 2.000
[ 44.044] (**) /dev/wsmouse: (accel) acceleration threshold: 4
```

4.2.10. Directorios importantes del sistema

El directorio raíz "/" en el sistema openBSD es similar al del sistema arch. Todos los directorios son subdirectorios de la raíz. Se encuentra el directorio **/etc** con las configuraciones de los programas, el directorio **/var** donde se guarda la cache del sistema, además de los logs del mismo. El directorio **/bin** donde al igual que en arch, se encuentran los binarios o programas que pueden usar los usuarios, y el directorio **/sbin** donde se encuentran los programas o binarios que solo puede usar el usuario administrador o root.



```
xterm
mellado$ ls -l
total 98820
drwxr-xr-x  2 root  wheel   512 Apr 11 19:45 altroot
drwxr-xr-x  2 root  wheel  1024 Apr 11 19:46 bin
-rw-r--r--  1 root  wheel 88136 Sep  6 21:54 boot
-rwx-----  1 root  wheel 22881188 Sep  7 05:26 bsd
-rwx-----  1 root  wheel 22888956 Sep  6 22:50 bsd,booted
-rw-----  1 root  wheel 4606368 Sep  6 16:03 bsd.rd
drwxr-xr-x  6 root  wheel  19456 Sep  7 05:24 dev
drwxr-xr-x 24 root  wheel   1536 Sep  7 06:06 etc
drwxr-xr-x  3 root  wheel   512 Sep  6 23:21 home
drwx-----T 2 root  wheel  2048 Sep  7 05:23 lost+found
drwxr-xr-x  2 root  wheel   512 Apr 11 19:45 mnt
drwx-----  4 root  wheel   512 Sep  6 22:51 root
drwxr-xr-x  2 root  wheel  1536 Apr 11 19:46/sbin
lrwxrwx---  1 root  wheel   11 Apr 11 19:45 sys -> usr/src/sys
drwxrwxrwt  6 root  wheel   512 Sep  7 06:49 tmp
drwxr-xr-x 16 root  wheel   512 Sep  6 21:54 usr
drwxr-xr-x 23 root  wheel   512 Apr 11 20:26 var
mellado$
```

5. Conclusiones

Relativo a las formas de uso, se concluye que ambos sistemas son muy similares. Tanto en los comandos para desplazarse entre las distintas ramas de los directorios (que son exactamente los mismos en ambos sistemas), listar un directorio de manera simple o detallada, crear archivos, o directorios, además de ejecutar los programas probados en este informe. Los cuales si bien pueden tener diferencias en el comando que ejecuta el programa, los conceptos detrás de estos son exactamente los mismos. Son muy similares también debido a que ambos sistemas son de tipo UNIX, por lo que hay una cierta compatibilidad de los programas entre archlinux y openBSD. También al ser sistemas de tipo UNIX obedecen a las normas POSIX, que es un conjunto de estándares escritos por la IEEE para facilitar el desarrollo de software multiplataforma [Universidad de Indiana, 2017].

En base a los métodos de instalación y configuración, se concluye que el sistema archlinux es más complejo de instalar y configurar que el sistema openBSD. Esto debido a la orientación que tiene los distintos sistemas. En el caso de archlinux, está centrado para usuarios avanzados, con estaciones de trabajo personalizadas. El método de instalación y configuración es en su mayor parte manual, la imagen iso del sistema deja al usuario en una terminal donde tiene que tipear cada comando, desde particionar el disco, crear a los usuarios, montar las particiones, hasta el comando que descarga e instala los paquetes base y el kernel dentro del directorio `/mnt` de la unidad booteable. Sin embargo quien comprende cada comando en la instalación del sistema arch, comprende cuál es el proceso que sigue cualquier instalador gráfico o de scripts en los sistemas de tipo UNIX. OpenBSD está centrado a la seguridad del sistema, por lo que varias configuraciones las hace el instalador de manera automatizada. El método de instalación de openBSD a un nivel menor es similar al del arch, solo añadiendo un programa de instalación y configuración automatizada, delegando al instalador la tarea de ejecutar los comandos, dependiendo de las respuestas que de el usuario. Esto hace que la instalación del sistema sea bastante rápida, lo cual es bueno para que el SO esté en producción lo más antes posible.

En un entorno de producción donde surge la necesidad de un SO para un servidor, en mi opinión, y teniendo en cuenta lo anteriormente mencionado, tomaría la decisión de instalar el sistema openBSD, debido a la facilidad, además del poco tiempo que toma su instalación, en adición a la configuración, que por defecto deja el programa instalador, dedicada a la seguridad del sistema. Además de la gran similitud de uso con los sistemas GNU/Linux, lo cual es bueno porque la documentación de los sistemas GNU/Linux abunda por la red de internet, haciendo que cualquier persona se pueda documentar correctamente en lo esencial del sistema.

Referencias

- [AMD, 2012] AMD (2012). Amd64 architecture programmer's manual. 2(24593):13–14.
Consultado el 28 de agosto del 2022.
- [Bellevue Linux Users Group, 2006] Bellevue Linux Users Group (2006). Kernel definition.
<http://www.linfo.org/kernel.html>.
Consultado el 2 de septiembre del 2022.
- [Equipo archlinux, 2022a] Equipo archlinux (2022a). fstab.
<https://wiki.archlinux.org/title/Fstab>.
Consultado el 5 de septiembre del 2022.
- [Equipo archlinux, 2022b] Equipo archlinux (2022b). Installation guide.
https://wiki.archlinux.org/title/installation_guide.
Consultado el 5 de septiembre del 2022.
- [Equipo de openBSD, 2022] Equipo de openBSD (2022). Openbsd. <https://www.openbsd.org/>.
Consultado el 28 de agosto del 2022.
- [Gillis, 2020] Gillis, A. S. (2020). device driver.
<https://www.techtarget.com/searchenterprisedesktop/definition/device-driver>.
Consultado el 5 de septiembre del 2022.
- [Han, 2022] Han, D. (2022). Soporte de windows para discos duros de mas de 2tb.
<https://docs.microsoft.com/es-ES/troubleshoot/windows-server/backup-and-storage/support-for-hard-disks-exceeding-2-tb>.
Consultado el 3 de septiembre del 2022.
- [Juncos, 2008] Juncos, R. (2008). Sistema de ficheros gnu/linux.
<https://web.archive.org/web/20081214104329/http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=549>.
Consultado el 3 de septiembre del 2022.
- [Levi, 2002] Levi, B. (2002). *UNIX Administration*.
Consultado el 28 de agosto del 2022.
- [Lissot, 2005] Lissot, A. (2005). Partition types.
<https://tldp.org/HOWTO/Partition/partition-types.html>.
Consultado el 7 de septiembre del 2022.
- [openbsdhandbook,] openbsdhandbook. File sets.
https://www.openbsdhandbook.com/package_management/#file-sets.
Consultado el 7 de septiembre del 2022.
- [Stallings, 2005] Stallings, W. (2005). Operating systems. internals and design principles. 7.
Consultado el 7 de septiembre del 2022.
- [Tanenbaum, 2003] Tanenbaum, A. (2003). *Sistemas Operativos Modernos*, volume 3.
Consultado el 2 de septiembre del 2022.
- [Universidad de Indiana, 2017] Universidad de Indiana (2017). What is posix.
<https://web.archive.org/web/20180614172556/https://kb.iu.edu/d/agjv>.
Consultado el 7 de septiembre del 2022.