

Clases 1, 2 y 3

1. Defina una arquitectura cualquiera de una PC y explique las funciones:

La arquitectura de una PC típica comprende varios componentes clave, incluyendo la CPU (Unidad Central de Procesamiento), la memoria RAM (Memoria de Acceso Aleatorio), el disco duro (o unidades de estado sólido), la tarjeta madre (o placa base), tarjetas de expansión (como tarjetas gráficas o de red), y dispositivos de entrada y salida (como teclado, ratón y monitor). La CPU es responsable de ejecutar instrucciones y coordinar el funcionamiento de todos los componentes. La memoria RAM almacena temporalmente datos y programas que están en uso. El disco duro almacena permanentemente datos y programas incluso cuando la PC está apagada. La tarjeta madre conecta todos los componentes entre sí y proporciona comunicación entre ellos. Las tarjetas de expansión agregan funcionalidad adicional, como gráficos mejorados o conectividad de red. Los dispositivos de entrada y salida permiten la interacción con la computadora.

2. Qué es un sistema operativo:

Un sistema operativo es un conjunto de programas y rutinas que coordina el uso del hardware de una computadora y proporciona servicios comunes para programas de aplicación. Actúa como un intermediario entre el hardware de la computadora y el software de la aplicación.

- **Gestión de procesos:** Coordina la ejecución de programas (procesos), gestiona la multitarea y la sincronización entre procesos.
- **Gestión de memoria:** Se ocupa del uso eficiente de la memoria, incluida la memoria virtual.
- **Sistemas de archivos:** Proporciona una forma de almacenar, organizar y acceder a archivos en dispositivos de almacenamiento.

El sistema operativo se divide en dos partes: Kernel e interfaz de usuario.

3. Define Kernel e interfaces de usuario:

El kernel es la parte central del sistema operativo que gestiona los recursos del sistema, como la memoria, los dispositivos de entrada/salida y el procesamiento del CPU.

Hay varios tipos de kernel:

- **Monolítico:** Todo el núcleo está integrado en un solo bloque.
- **Microkernel:** Solo las funciones básicas están en el núcleo; otras funciones corren fuera de él.

- Híbrido: Una mezcla de monolítico y microkernel.

Las interfaces de usuario son las formas en que los usuarios interactúan con el sistema operativo y los programas, como la interfaz de línea de comandos (CLI) o la interfaz gráfica de usuario (GUI).

4.Cuál es la mayor diferencia entre las arquitecturas de 32 bits y 64 bits:

La principal diferencia radica en la cantidad de memoria que pueden direccionar. Los sistemas de 32 bits pueden direccionar hasta 4 GB de memoria RAM, mientras que los sistemas de 64 bits pueden direccionar una cantidad mucho mayor, teóricamente hasta 16 exabytes de memoria.

32 bits:

- Los sistemas de 32 bits pueden manejar 2^{32} direcciones de memoria, lo que equivale a 4 GB de memoria RAM.
- Esta arquitectura es suficiente para muchas aplicaciones, pero tiene limitaciones, especialmente con el uso de aplicaciones más modernas que requieren más memoria.
- La mayoría de los sistemas operativos antiguos y las versiones antiguas de software funcionan en 32 bits.

64 bits:

- Los sistemas de 64 bits pueden manejar 2^{64} direcciones de memoria, lo que permite un acceso teórico a 16 exabytes de RAM (aunque en la práctica, los límites son mucho más bajos).
- Esta arquitectura es necesaria para ejecutar aplicaciones que requieren grandes cantidades de memoria, como simulaciones científicas, modelado 3D o servidores de bases de datos.
- Además de poder usar más memoria, los procesadores de 64 bits pueden realizar cálculos más rápidos y manejar más datos en un solo ciclo de reloj en comparación con los de 32 bits.
- Los sistemas operativos modernos y la mayoría del software actual están diseñados para arquitecturas de 64 bits.

5. Define procesos e hilos:

Un proceso es una instancia en ejecución de un programa en un sistema operativo, que incluye el código del programa, los datos asociados y el estado del recurso del sistema. Un hilo es una unidad básica de ejecución dentro de un proceso, que puede ejecutar tareas independientes simultáneamente con otros hilos en el mismo proceso.

```
int main()
{
    int i;
    printf("Inicio\n");

    for(i=0;i<2;i++)
        printf("Hola %d\n",i);

    printf("Fin");
}
```

Un proceso está
compuesto por uno o
más hilos.

6. Dentro de un proceso que son los datos, códigos y pilas, como se relacionan con los hilos del proceso:

Los datos son variables y estructuras de datos utilizadas por el programa. El código es la secuencia de instrucciones ejecutables del programa. La pila es una región de memoria utilizada para almacenar datos temporales y contexto de ejecución de funciones. Los hilos comparten el mismo espacio de código y datos, pero tienen su propia pila de ejecución.

7. Poner ejemplo de cómo el sistema operativo organiza los procesos:

El sistema operativo organiza los procesos utilizando tablas de procesos, que contienen información sobre cada proceso en ejecución, como su identificador único, estado, recursos asignados y prioridad de ejecución como:

FIFO (First-In, First-Out):

Este método asigna prioridad a los procesos en función de su tiempo de llegada, es decir, el primero que llega es el primero en ser atendido.

Round-Robin:

En este método, cada proceso recibe una pequeña y misma cantidad de tiempo de CPU, llamado cuántum, y luego se pasa al siguiente proceso en la cola aun cuando este no haya terminado. Si un proceso no ha terminado durante su cuántum asignado, se pone al final de la cola para esperar su próximo turno.

Prioridad dinámica (SJF - Shortest Job First):

En este método, el proceso con el menor tiempo de ejecución se selecciona para ejecutarse primero. Esto minimiza el tiempo promedio de espera y maximiza la utilización del procesador.

Prioridad estática:

Cada proceso recibe una prioridad fija en función de su importancia relativa. Los procesos con mayor prioridad se ejecutan antes que los de menor prioridad. Puede llevar a problemas de inanición si los procesos de alta prioridad monopolizan los recursos.

8. Poner ejemplo de cómo el sistema operativo maneja la memoria RAM:

El sistema operativo administra la memoria RAM asignando y liberando espacio para procesos y datos según sea necesario. Utiliza técnicas como la paginación, la segmentación y el swapping para gestionar eficientemente la memoria disponible.

Paginación:

- La paginación es una técnica de gestión de memoria que divide la memoria física y virtual en bloques de tamaño fijo llamados "páginas".
- Tanto el espacio de direcciones físicas como virtuales se dividen en páginas del mismo tamaño.
- Cuando un proceso intenta acceder a la memoria, el sistema operativo traduce la dirección virtual del proceso a una dirección física utilizando una tabla de páginas.
- La paginación permite la asignación de memoria de forma flexible y eficiente, ya que los marcos de página pueden asignarse y liberarse individualmente.

Segmentación:

- La segmentación es otra técnica de gestión de memoria que divide la memoria en segmentos lógicos de diferentes tamaños.
- Cada segmento representa una región lógica del espacio de direcciones del proceso, como el código, los datos y la pila.

- Los segmentos pueden crecer y contraerse dinámicamente según las necesidades del proceso.
- El sistema operativo utiliza tablas de segmentos para realizar la traducción de direcciones lógicas a direcciones físicas.

Swapping:

- El swapping, o intercambio de memoria, es una técnica utilizada cuando la memoria física se agota y es necesario liberar espacio.
- Consiste en mover parte de un proceso de la memoria principal (RAM) a un espacio de almacenamiento secundario, como el disco duro.
- Cuando se necesita nuevamente la parte del proceso que se ha movido al almacenamiento secundario, se vuelve a cargar en la memoria principal.
- El swapping permite que el sistema operativo mantenga más procesos en ejecución que la capacidad física de la memoria, aunque con el costo de una menor velocidad de acceso a los datos que se han movido al disco.

9. Qué es la memoria virtual:

La memoria virtual es una técnica que utiliza el sistema operativo para proporcionar a cada proceso la ilusión de disponer de una memoria contigua y privada, independientemente de la cantidad de memoria física real disponible en el sistema. Esta técnica utiliza tanto la memoria RAM (memoria física) como el almacenamiento en disco (memoria secundaria) para administrar la memoria de manera eficiente.

La relación entre la memoria virtual y el swapping radica en cómo se gestiona la memoria cuando la cantidad de memoria física es limitada. Cuando un proceso necesita más memoria de la que está disponible en la RAM, el sistema operativo puede utilizar la memoria virtual para almacenar temporalmente partes menos utilizadas de la memoria en el disco. Este proceso se conoce como "swapping" que explicamos anteriormente.

10. Código abierto, software libre y software gratis:

Código Abierto (Open Source)

El código abierto es un modelo de desarrollo de software donde el código fuente está disponible para que cualquiera lo vea, lo modifique y lo distribuya. El objetivo es promover

la colaboración y mejora del software por parte de la comunidad de desarrolladores, con un enfoque en la eficiencia técnica y la innovación.

Características del Código Abierto:

1. Acceso al código fuente: El código está disponible públicamente para que los usuarios lo examinen, modifiquen y mejoren.
2. Colaboración: Cualquier persona puede contribuir, lo que resulta en mejoras continuas, correcciones de errores y nuevas funciones.

Ejemplo:

Linux es uno de los ejemplos más conocidos de código abierto. Su núcleo (kernel) está disponible para que cualquiera lo modifique o utilice en sus propias distribuciones, como Ubuntu o Fedora.

Software Libre (Free Software)

El software libre es un concepto más filosófico y ético, defendido por la Free Software Foundation y promovido por Richard Stallman. Se enfoca en la libertad de los usuarios para usar, estudiar, modificar y compartir el software. El término "libre" se refiere a libertad, no a costo.

Las Cuatro Libertades del Software Libre:

1. Libertad para ejecutar el software como se desee.
2. Libertad para estudiar el código fuente y adaptarlo.
3. Libertad para redistribuir copias del software.
4. Libertad para mejorar el software y compartir esas mejoras con la comunidad.

Diferencias con el Código Abierto:

El software libre garantiza las libertades del usuario y requiere que cualquier modificación que se distribuya mantenga esas mismas libertades (como es el caso de la licencia GNU GPL, que obliga a compartir las modificaciones bajo la misma licencia). En cambio, el código abierto se enfoca en la colaboración y puede ser más flexible en cuanto a licencias, permitiendo, en algunos casos, que las modificaciones no sean compartidas.

Ejemplo:

GNU y LibreOffice son ejemplos clásicos de software libre. Ambos garantizan las libertades de sus usuarios y fomentan la contribución y redistribución bajo licencias libres como la GPL.

Software Gratis (Free Software)

El término software gratis o "freeware" se refiere a software que se distribuye de manera gratuita, pero no necesariamente proporciona acceso al código fuente ni garantiza las libertades del software libre o de código abierto. En este caso, "free" hace referencia únicamente al costo, no a la libertad.

Características del Software Gratis:

1. Costo cero: El software se ofrece gratuitamente, pero esto no implica que sea de código abierto o libre.
2. Restricciones: A menudo, el software gratis tiene restricciones en cuanto a su uso, modificación o redistribución. No necesariamente se puede estudiar o cambiar.
3. Licencias: Las licencias de software gratis pueden ser propietarias, lo que significa que el desarrollador retiene todos los derechos sobre el software. Algunas versiones del software gratuito tienen funciones limitadas o muestran publicidad.

Ejemplo:

Un ejemplo común de software gratis es Skype. Gratuito para el usuario final, pero no permite el acceso al código fuente ni otorgan la libertad para modificarlos o redistribuirlos.

Resumen:

- Código Abierto (Open Source): Se enfoca en la colaboración y eficiencia técnica. El código está disponible para que cualquiera lo modifique y mejore, con diferentes tipos de licencias (algunas más permisivas que otras).
- Software Libre: Enfocado en las libertades de los usuarios, garantizando que puedan usar, estudiar, modificar y compartir el software sin restricciones. Se adhiere a un enfoque más ético y suele usar licencias como la GPL.
- Software Gratis (Freeware): Es software que se distribuye sin costo, pero sin garantizar acceso al código fuente ni las libertades del software libre o de código abierto. Puede tener restricciones en su uso o redistribución.

11. Poner ejemplo de Windows y Linux respecto al kernel y sus interfaces. Relaciónalo con el código abierto, libre y gratis:

Windows tiene un kernel cerrado y sus interfaces son principalmente gráficas, como el escritorio y las ventanas. Linux, por otro lado, tiene un kernel de código abierto y ofrece una variedad de interfaces, incluyendo la interfaz de línea de comandos y varias interfaces gráficas de usuario como GNOME y KDE.

Windows es un ejemplo clásico de software cerrado y propietario. Aunque algunos componentes de Windows pueden ser gratuitos (como versiones de prueba), el sistema operativo en su totalidad no es código abierto ni software libre. Los usuarios no tienen acceso al código fuente ni la posibilidad de modificarlo. Windows es, en general, un producto comercial que se compra, y aunque en algunas ocasiones puede estar disponible como software gratis (por ejemplo, versiones educativas o promociones), sigue siendo restrictivo en cuanto a libertades de modificación y redistribución.

Linux, en cambio, es un ejemplo perfecto de software libre y de código abierto. El código del kernel es accesible, modificable y redistribuible bajo los términos de la GPL. Además, la mayoría de las distribuciones de Linux (como Ubuntu, Debian, Fedora) son también software gratis, lo que significa que los usuarios pueden descargar e instalar el sistema sin ningún costo. Esta combinación de código abierto, software libre y software gratis ha hecho de Linux una opción popular, tanto en servidores como en entornos de desarrollo y usuarios comunes que valoran la libertad y personalización.

Ejemplo Comparativo:

En Windows, si un usuario encuentra un problema con el sistema, no puede acceder al kernel para modificar o corregir el código. Además, el sistema tiene una interfaz gráfica única y cerrada, diseñada por Microsoft, y cualquier cambio en la apariencia o funcionalidad depende de las actualizaciones oficiales.

En Linux, si un usuario avanzado encuentra un problema, puede acceder al código fuente del kernel y, si tiene los conocimientos necesarios, puede modificarlo para solucionar el problema. También puede elegir entre varias interfaces gráficas, como GNOME o KDE, o usar directamente la línea de comandos para un mayor control del sistema.

12. Estructura de directorios en Windows y Linux:

Conocer la estructura de directorios es importante por varias razones:

- Facilita la navegación y la administración del sistema operativo.
- Proporciona un marco organizativo para los archivos y directorios del sistema.
- Permite a los administradores de sistemas y usuarios comprender mejor cómo funciona el sistema operativo y dónde se almacenan los diferentes tipos de archivos.
- Facilita la localización y el acceso a archivos y recursos del sistema.
- Ayuda a comprender la funcionalidad y la integración de aplicaciones y servicios en el sistema operativo.
- Es fundamental para la solución de problemas y el mantenimiento del sistema.

Estructura de directorios de Windows

La estructura de directorios de Windows se organiza de manera jerárquica, utilizando carpetas (también conocidas como directorios) que contienen archivos y subcarpetas:

Raíz del Sistema (`C:\`)

El disco principal donde se instala Windows generalmente es la unidad C:\. Esta es la raíz del sistema, y de ella cuelgan todas las carpetas y archivos principales. Otros discos duros o particiones tienen asignadas letras como D:\, E:\, etc.

Carpetas Principales en la Unidad C:

En una instalación típica de Windows, encontrarás varias carpetas clave en la raíz del sistema, que incluyen:

- `C:\Windows`: La carpeta donde se almacenan los archivos del sistema operativo Windows. Aquí están todos los archivos necesarios para que el sistema funcione, como controladores, bibliotecas (DLL), configuraciones, y más.

Subcarpetas importantes dentro de `C:\Windows`:

- `System32`: Contiene archivos esenciales del sistema operativo, como ejecutables y bibliotecas dinámicas (.dll).

- `Fonts`: Almacena las fuentes instaladas en el sistema.

- `C:\Program Files`: Carpeta donde se instalan la mayoría de los programas de 64 bits. Los archivos de los programas instalados (ejecutables, bibliotecas, archivos de ayuda) se almacenan aquí.

- `C:\Program Files (x86)`: Carpeta equivalente a `Program Files`, pero específica para los programas de 32 bits en sistemas operativos de 64 bits.

- `C:\Users\NombreDeUsuario\Documents`: Carpeta de documentos personales del usuario.

- `C:\Users\NombreDeUsuario\Downloads`: Carpeta de descargas.

- `C:\Users\NombreDeUsuario\AppData`: Carpeta oculta que contiene datos de las aplicaciones específicas del usuario, como configuraciones o datos de caché.

- `C:\ProgramData`: Similar a `AppData`, pero almacena datos de aplicaciones a nivel global, accesibles por todos los usuarios del sistema. También suele estar oculta.

- `C:\Windows\Temp`: Carpeta donde se almacenan archivos temporales creados por Windows y aplicaciones.

Estructura de directorio de Linux

/ (Raíz):

Función: El directorio raíz del sistema de archivos de Linux.

Contenido: Todos los archivos y directorios del sistema.

Importancia: Conocer la estructura de directorios desde la raíz es fundamental para comprender cómo está organizado el sistema operativo Linux y acceder a sus recursos.

/bin (Binarios):

Función: Contiene programas ejecutables esenciales del sistema.

Contenido: Comandos básicos como `ls`, `cp`, `mv`, etc.

Importancia: Estos programas son necesarios para el funcionamiento básico del sistema, y acceder a ellos es crucial para realizar tareas cotidianas en Linux.

/boot (Arranque):

Función: Contiene archivos necesarios para el proceso de arranque del sistema.

Contenido: Núcleos del sistema y archivos de configuración del gestor de arranque (GRUB).

Importancia: Es esencial para el inicio del sistema y la gestión de múltiples sistemas operativos en una computadora.

/dev (Dispositivos):

Función: Contiene archivos de dispositivo que representan hardware y pseudo-dispositivos del sistema.

Contenido: Representaciones de dispositivos como discos duros, terminales, etc.

Importancia: Permite interactuar con el hardware del sistema a través de archivos en el sistema de archivos.

/etc (Configuración):

Función: Almacena archivos de configuración del sistema y de programas instalados.

Contenido: Archivos como `/etc/passwd`, `/etc/hosts`, etc.

Importancia: Configura el comportamiento del sistema y de las aplicaciones instaladas.

/home (Hogares de usuarios):

Función: Directorio base para los directorios de inicio de los usuarios.

Contenido: Directorios de inicio de usuarios individuales.

Importancia: Almacena los datos personales y configuraciones de los usuarios.

/lib (Bibliotecas):

Función: Contiene bibliotecas compartidas necesarias para programas en `/bin` y `/sbin`.

Contenido: Bibliotecas compartidas dinámicas (`*.so`).

Importancia: Proporciona funcionalidad compartida a programas en todo el sistema.

/media (Medios extraíbles):

Función: Punto de montaje para dispositivos extraíbles como USB, DVD, etc.

Contenido: Montajes temporales de dispositivos externos.

Importancia: Facilita el acceso a medios extraíbles y su uso en el sistema.

/mnt (Puntos de montaje):

Función: Directorio comúnmente utilizado para montar sistemas de archivos temporales o de red.

Contenido: Montajes temporales de sistemas de archivos.

Importancia: Facilita la gestión de sistemas de archivos temporales y montajes de red.

/opt (Opcionales):

Función: Contiene paquetes de software adicionales instalados en el sistema.

Contenido: Programas de terceros y aplicaciones.

Importancia: Proporciona un lugar para instalar software que no está disponible en los repositorios predeterminados del sistema.

/proc (Proceso):

Función: Contiene información sobre los procesos en ejecución y configuraciones del kernel.

Contenido: Archivos virtuales que representan procesos y configuraciones del kernel.

Importancia: Proporciona información en tiempo real sobre el estado del sistema y los procesos en ejecución.

/root (Raíz del sistema):

Función: Directorio de inicio del usuario raíz (administrador) del sistema.

Contenido: Directorio de inicio del usuario root.

Importancia: Proporciona acceso a los archivos y configuraciones del usuario raíz.

/run (Ejecutar):

Función: Contiene archivos de estado y archivos de comunicación temporal entre procesos.

Contenido: Archivos temporales necesarios para la ejecución de procesos.

Importancia: Facilita la comunicación y el intercambio de datos entre procesos en ejecución.

/sbin (Binarios del sistema):

Función: Contiene programas ejecutables esenciales para la administración del sistema.

Contenido: Comandos destinados principalmente para el uso del superusuario (root).

Importancia: Proporciona herramientas esenciales para la administración del sistema y la solución de problemas.

/srv (Servicio):

Función: Contiene datos de servicio específicos del sistema que son servidos por el sistema.

Contenido: Datos de servicios web, FTP, etc.

Importancia: Almacena datos relacionados con servicios que son accesibles a través de la red.

/sys (Sistema):

Función: Contiene información sobre el sistema, el hardware y los controladores del kernel.

Contenido: Archivos que representan la configuración y el estado del sistema.

Importancia: Proporciona una interfaz para configurar y manipular parámetros del kernel en tiempo de ejecución.

/tmp (Temporal):

Función: Contiene archivos temporales que se eliminan automáticamente al reiniciar el sistema.

Contenido: Archivos temporales creados por programas.

Importancia: Proporciona un lugar para almacenar datos temporales que no necesitan persistencia entre sesiones de usuario.

/usr (Usuario):

Función: Contiene archivos y directorios no esenciales para el funcionamiento del sistema.

Contenido: Software y archivos del usuario, como programas instalados, bibliotecas y documentación.

Importancia: Almacena la mayor parte del software del sistema y del usuario, facilitando su gestión y actualización.

/var (Variables):

Función: Contiene archivos variables como registros del sistema, archivos de correo, bases de datos, etc.

Contenido: Datos que pueden cambiar de tamaño o contenido con el tiempo.

Importancia: Almacena datos que varían durante la ejecución del sistema, como registros, correos electrónicos, datos de aplicaciones, etc.

/lost+found:

Función: Utilizado por el sistema de archivos para almacenar archivos recuperados durante la verificación del sistema de archivos.

Contenido: Archivos recuperados que no tienen una ubicación clara.

Importancia: Proporciona un lugar para almacenar archivos recuperados en caso de corrupción del sistema de archivos.

13. El particionamiento de discos

El particionamiento de discos es el proceso de dividir un disco físico en múltiples unidades lógicas, conocidas como particiones. Cada partición actúa como un disco separado y puede ser formateada con diferentes sistemas de archivos. Esta división ayuda a organizar los datos de manera eficiente y proporciona flexibilidad para la instalación de múltiples sistemas operativos o la segregación de datos.

Tipos de particiones:

Primaria: Puede contener un sistema operativo y solo pueden existir cuatro particiones primarias por disco.

Lógica: Se encuentra dentro de una partición extendida. Útil para crear más de cuatro particiones.

Extendida: Una partición especial que puede contener múltiples particiones lógicas. Esto permite superar la limitación de cuatro particiones.

Esquemas de particionamiento principales:

MBR (Master Boot Record): Utilizado en discos antiguos, permite hasta cuatro particiones primarias y un tamaño de disco máximo de 2 TB.

GPT (GUID Partition Table): Más moderno y utilizado en discos de gran capacidad (superiores a 2 TB). Permite un número casi ilimitado de particiones y es más seguro frente a la corrupción.

Formateo y sistemas de archivos

Una vez que las particiones están creadas, el siguiente paso es formatearlas. El formateo consiste en preparar la partición para almacenar datos mediante la aplicación de un sistema de archivos.

Sistemas de archivos básicos:

FAT32 (File Allocation Table):

Antiguo pero muy compatible.

Limitación de archivos de hasta 4 GB.

Compatible con Windows, Linux y macOS.

NTFS (New Technology File System):

Sistema de archivos predeterminado para Windows.

Compatible con archivos de gran tamaño.

Soporta características avanzadas como permisos de archivos, compresión y encriptación.

Ligeramente menos compatible con otros sistemas operativos de forma nativa.

exFAT (Extended File Allocation Table):

Una evolución de FAT32, sin las limitaciones de tamaño de archivo.

Ideal para dispositivos de almacenamiento como memorias USB y discos externos.

Compatible con Windows y macOS.

EXT., EXT4 (Fourth Extended File System):

Sistema de archivos predeterminado para Linux.

Soporta archivos de gran tamaño y tiene un rendimiento excelente en discos grandes.

HFS+ y APFS (Apple File System):

HFS+ es el sistema antiguo de macOS.

APFS es el sistema moderno y rápido introducido en macOS 10.13, con optimizaciones para SSDs y características avanzadas como instantáneas.

Beneficios del particionamiento

- **Organización:** Permite dividir datos del sistema operativo, aplicaciones y archivos personales.
- **Multisistema operativo:** Se pueden instalar varios sistemas operativos en particiones diferentes (dual-boot).
- **Seguridad:** Una partición dañada no afecta a las otras. Los datos críticos se pueden separar del sistema.
- **Mantenimiento:** Facilita la creación de copias de seguridad y la recuperación de particiones individuales sin afectar al disco completo.

Prácticas recomendadas

Uso de GPT: Siempre que sea posible, usa GPT en lugar de MBR para aprovechar su flexibilidad y seguridad.

Crear particiones separadas: Al menos una para el sistema operativo y otra para los datos personales.

Formato adecuado: Selecciona el sistema de archivos según el uso del dispositivo. Por ejemplo, NTFS o exFAT para dispositivos portátiles, EXT4 para servidores Linux, y APFS para equipos Apple.

14. Que significa LTS:

LTS significa "Long-Term Support" (Soporte a Largo Plazo) y se refiere a versiones de software que reciben soporte extendido y actualizaciones de seguridad durante un período prolongado, generalmente varios años.

15. Qué es la virtualización y cuál es su importancia:

La virtualización es la creación de una versión virtual de un recurso de hardware, como una computadora, un servidor, un sistema de almacenamiento o una red. Permite la ejecución de múltiples sistemas operativos y aplicaciones en una sola máquina física, lo que mejora la eficiencia, la flexibilidad y la utilización de recursos.

16. Quiénes son el Host y el Guest:

El host es la máquina física que ejecuta el software de virtualización, mientras que el guest es la máquina virtual que se ejecuta dentro del host.

17. Qué significa Hipervisor tipo1 y tipo 2:

Un hipervisor de tipo 1 se ejecuta directamente sobre el hardware físico y gestiona los sistemas operativos invitados, mientras que un hipervisor de tipo 2 se ejecuta como una aplicación dentro de un sistema operativo hospedador y proporciona un entorno para las máquinas virtuales.

Hipervisor de tipo 1, aplicaciones:

- VMware ESXi
- Microsoft Hyper-V
- Xen
- KVM (Kernel-based Virtual Machine)

Hipervisor de tipo 2, aplicaciones:

- VMware Workstation
- **Oracle VirtualBox**
- Parallels Desktop
- QEMU

18. Relate los pasos para crear una Máquina Virtual con VirtualBox:

Los pasos para crear una máquina virtual con VirtualBox incluyen instalar VirtualBox, crear una nueva máquina virtual, asignar recursos como CPU y RAM, configurar el almacenamiento, montar la imagen ISO del sistema operativo invitado, y finalmente arrancar la máquina virtual para instalar el sistema operativo.

19. Qué significa proporcionar recursos a la máquina virtual:

Significa asignar recursos físicos como CPU, memoria RAM, almacenamiento y periféricos a la máquina virtual para su uso exclusivo.

20. Qué es y para qué puede utilizar la clonación y el Snapshot en VirtualBox:

La clonación en VirtualBox es el proceso de crear una copia idéntica de una máquina virtual existente. Los snapshots son instantáneas del estado actual de una máquina virtual en un momento dado, lo que permite volver a ese estado en caso de problemas o cambios no deseados.

21. Qué significa el estado NAT en la asignación de red de la máquina virtual:

NAT (Network Address Translation) es un método que utiliza VirtualBox para asignar direcciones IP a las máquinas virtuales y permitirles acceder a la red externa a través del host, ocultando las direcciones IP internas de las máquinas virtuales.

22. Es posible Compartir carpetas entre el Host y el Guest:

Sí, es posible compartir carpetas entre el host y el guest mediante la configuración de carpetas compartidas en la configuración de la máquina virtual.

23. Diferencia entre ISO y Disco Virtual:

Un archivo ISO es una imagen de disco que contiene una copia exacta de los datos de un disco óptico, como un CD o un DVD. Un disco virtual, por otro lado, es un archivo que simula un disco duro o una unidad de almacenamiento y puede contener cualquier tipo de datos, incluidos sistemas operativos, aplicaciones y archivos de datos.

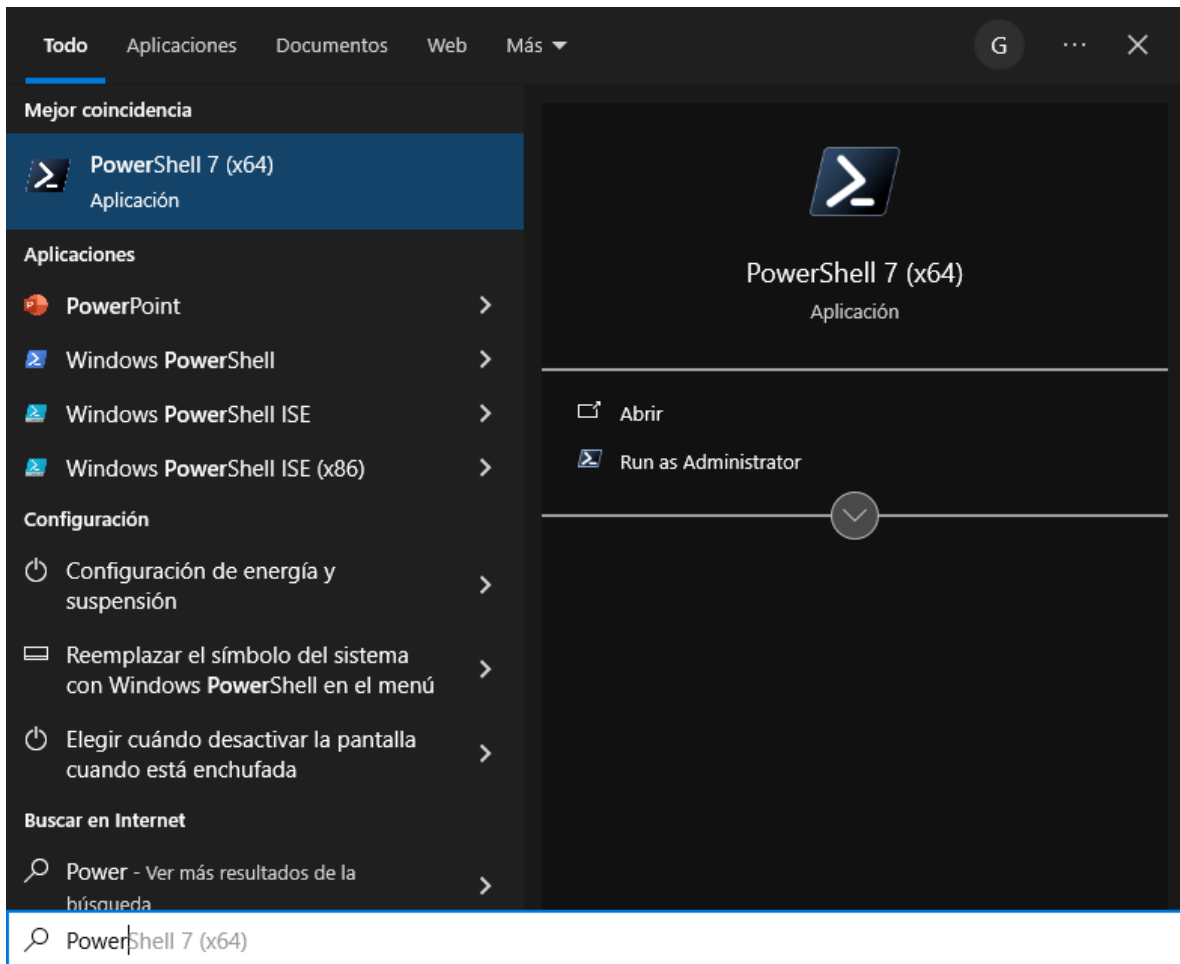
PARTE 2 Instalando en la práctica: Diversas formas de acceder a diferentes sistemas operativos

- 1. PowerShell**
- 2. WSL**
- 3. Virtualbox**
- 4. Chocolatey**
- 5. Vagrant**
- 6. Con el uso de GIT**

1 - PowerShell

PowerShell es una interfaz de línea de comandos (CLI) y un entorno de scripting desarrollado por Microsoft para la administración de sistemas y la automatización de tareas en sistemas operativos

PowerShell está diseñado para ser más potente que el símbolo del sistema tradicional de Windows (cmd.exe), ya que permite el acceso a una amplia gama de funciones del sistema y la manipulación de objetos y datos de manera más eficiente.



Los primeros comandos que vamos a estar utilizando

Acción	CMD (Windows)	PowerShell (Windows)	Linux/Unix
Listar Carpetas	<code>dir</code>	<code>Get-ChildItem</code>	<code>ls</code>
Moverse Entre Carpetas	<code>cd <nombre_de_la_carpeta></code>	<code>Set-Location <nombre_de_la_carpeta></code>	<code>cd <nombre_de_la_carpeta></code>
Crear Carpetas	<code>mkdir <nombre_de_la_carpeta></code>	<code>New-Item -ItemType Directory -Name <nombre_de_la_carpeta></code>	<code>mkdir <nombre_de_la_carpeta></code>

Eliminar Carpetas	rmdir <nombre_de_la_carpeta>	Remove-Item <nombre_de_la_carpeta>	rm -r <nombre_de_la_carpeta>
----------------------	---	---	--

Con el comando Get-Alias dir se puede tener el alias del comando en PowerShell

2- WSL

Los desarrolladores pueden acceder a la potencia de Windows y Linux al mismo tiempo en una máquina Windows. Subsistema de Windows para Linux (WSL) permite a los desarrolladores instalar una distribución de Linux (como Ubuntu, OpenSUSE, Kali, Debian, Arch Linux, etc.) y usar aplicaciones, utilidades y herramientas de línea de comandos de Bash directamente en Windows, sin modificar, sin la sobrecarga de una máquina virtual tradicional o una configuración de arranque dual.

wsl --install

```
PS D:\> wsl --install
Ubuntu ya está instalado.
Iniciando Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: 
```

Solicita usuario y contraseña para Ubuntu

Probar los comandos de la tabla

Podemos Cambiar la distribución predeterminada de Linux instalada

De forma predeterminada, la distribución de Linux instalada será Ubuntu. Se puede cambiar mediante la marca -d.

Para cambiar la distribución instalada, escriba: `wsl --install -d <Distribution Name>`. Reemplace <Distribution Name> por el nombre de la distribución que desea instalar.

Para ver una lista de las distribuciones de Linux disponibles para descargar a través de la tienda en línea, escriba `wsl --list --online` o `wsl -l -o`.

Para instalar distribuciones de Linux adicionales después de la instalación inicial, también puede usar el comando `wsl --install -d <Distribution Name>`.

```
Debian
yo
yo@Guille:~$ ls
yo@Guille:~$ cd..
cd..: command not found
yo@Guille:~$ mkdir mio
yo@Guille:~$ ls
mio
yo@Guille:~$ cd mio
yo@Guille:~/mio$ cd..
cd..: command not found
yo@Guille:~/mio$ cd
yo@Guille:~$ cd mio
yo@Guille:~/mio$ cd
yo@Guille:~$ wsl --list --online
Command 'wsl' not found, but can be installed with:
sudo apt install wsl
yo@Guille:~$ exit
logout
La operación se completó correctamente.
PS D:\> wsl --list --online
A continuación, se muestra una lista de las distribuciones válidas que se pueden instalar.
Instalar con "wsl.exe --install <Distro>".

NAME                                FRIENDLY NAME
Ubuntu                              Ubuntu
Debian                              Debian GNU/Linux
kali-linux                          Kali Linux Rolling
Ubuntu-18.04                         Ubuntu 18.04 LTS
Ubuntu-20.04                         Ubuntu 20.04 LTS
Ubuntu-22.04                         Ubuntu 22.04 LTS
Oraclelinux_7_9                      Oracle Linux 7.9
Oraclelinux_8_7                      Oracle Linux 8.7
Oraclelinux_9_1                      Oracle Linux 9.1
openSUSE-Leap-15.5                   openSUSE Leap 15.5
SUSE-Linux-Enterprise-Server-15-SP4  SUSE Linux Enterprise Server 15 SP4
SUSE-Linux-Enterprise-15-SP5         SUSE Linux Enterprise 15 SP5
openSUSE-Tumbleweed                  openSUSE Tumbleweed
PS D:\> wsl --install -d Debian
Instalando: Debian GNU/Linux.
Se ha instalado Debian GNU/Linux.
Iniciando Debian GNU/Linux...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: yo
New password:
```

Instalemos Debian y probemos todos los comandos

3- Virtualbox

Asegurarte de que puedas virtualizar (systeminfo) Ir a la Bios a habilitar

Descargar desde <https://www.virtualbox.org/wiki/Downloads>

Descargar ISO LTS

Crear y realizar las primeras configuraciones

Asignar recursos

Compartir red

Compartir carpetas entre el host y el guest

Clonar una Máquina

Crear Snapshot

Ir a la terminal y lanzar los primeros comandos

4 – Chocolatey

Chocolatey es una herramienta de gestión de paquetes para Windows que simplifica el proceso de instalación, actualización y administración de software en sistemas Windows. Funciona de manera similar a los gestores de paquetes en sistemas operativos basados en Unix, como apt en Debian/Ubuntu o yum en Red Hat/CentOS.

Con Chocolatey, los usuarios pueden instalar rápidamente una amplia variedad de software con solo un comando, en lugar de tener que buscar y descargar manualmente los instaladores individuales de cada aplicación. Además, Chocolatey facilita la gestión de las actualizaciones de software, ya que puede mantener automáticamente los programas instalados actualizados a las últimas versiones disponibles.

Algunas características clave de Chocolatey incluyen:

Fácil instalación de software: Con Chocolatey, puedes instalar software con un simple comando en la línea de comandos.

Administración centralizada de paquetes: Chocolatey proporciona una base de datos centralizada de paquetes de software, lo que facilita la búsqueda y la instalación de programas.

Actualizaciones automáticas: Chocolatey puede mantener tus programas instalados actualizados automáticamente, eliminando la necesidad de buscar manualmente actualizaciones.

Gestión de dependencias: Chocolatey maneja las dependencias entre paquetes de software, lo que garantiza que todas las dependencias necesarias se instalen correctamente.

En resumen, Chocolatey es una herramienta poderosa que simplifica la gestión de software en sistemas Windows, haciéndolo más eficiente y menos propenso a errores. Es

especialmente útil para administradores de sistemas y usuarios avanzados que desean automatizar y simplificar el proceso de instalación y mantenimiento de software en sus sistemas Windows.

Comando para instalar

Web: <https://chocolatey.org/install>

Verificar que estén las políticas activadas en power Shell ejecutar

```
PS C:\Windows\system32> Get-ExecutionPolicy
```

Restricted **(Si sale esto. Ejecutar entonces)**

```
PS C:\Windows\system32> Set-ExecutionPolicy AllSigned
```

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object  
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

Posteriormente con este comando se instalan las herramientas vagrant y git que estaremos utilizando desde chocolatey:

```
choco install -y git vagrant --log-file=c:\chocolatey_install.log
```

Importante con chocolatey Podemos instalar varias aplicaciones por ejemplo el propio virtual box

```
choco install -y virtualbox virtualbox-guest-additions-guest.install --log-  
file=c:\chocolatey_install.log
```

```
Administrador: Windows PowerShell

PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
Forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)
Getting latest version of the Chocolatey package for download.
Not using proxy.
Getting Chocolatey from https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2.
Downloading https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2 to C:\Users\Domoflow\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip
Not using proxy.
Extracting C:\Users\Domoflow\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip to C:\Users\Domoflow\AppData\Local\Temp\chocolatey\chocoInstall
Installing Chocolatey on the local machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
  Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
  before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
  (i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
  and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey folders if they do not already exist.

chocolatey.nupkg file not installed in lib.
  Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
ADVERTENCIA: Not setting tab completion: Profile file does not exist at
'C:\Users\Domoflow\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
  first prior to using choco.
Ensuring Chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\Windows\system32> choco install -y git sublimetext4 virtualbox virtualbox-guest-additions-guest.install vagrant mo
baxterm putty.portable winscp --log-file=c:\chocolatey_install.log
```

Generación automática de una VM y metodologías de conexión

Vagrant y GIT

Vagrant es una aplicación libre desarrollada en ruby que nos permite crear y personalizar entornos de desarrollo livianos, reproducibles y portables. Vagrant nos permite automatizar la creación y gestión de máquinas virtuales.

Git es un sistema de control de versiones distribuido, diseñado para rastrear cambios en archivos y coordinar el trabajo entre múltiples personas en proyectos de desarrollo de software.

Descargar el repositorio que está en

https://github.com/upszot/UTN-FRA_SO_Vagrant

Utilizar GIT para clonar

Crear en c carpeta

/c/so/repogit

Dentro de repogit dar clic derecho abri bash ejecutar

\$ git clone https://github.com/upszot/UTN-FRA_SO_Vagrant.git

\$ git pull para tener actualizado el directorio

probar los comandos:

Crear y dejar en running la/s VM definidas en el vagrantfile
vagrant up

vagrant ssh para entrar

Apagar y Eliminar sin pedir confirmación

vagrant destroy -f

Apagar la vm sin eliminarla (una forma de mantener persistencia de datos)

vagrant halt

Creemos nuestro primer vagrant file

vagrant init

vagrant init ubuntu/bionic64

Ansible en WSL otra forma de automatizar la creación de MV

Abre wsl y ejecuta:

sudo apt update

sudo apt install ansible

Verifica que Ansible esté instalado:

ansible --version

Instalar Hyper-V o VirtualBox en Windows

Habilitar Hyper-V: Ve al Panel de control > Programas > Activar o desactivar características de Windows y selecciona Hyper-V. Luego, reinicia tu máquina.



Instalar el módulo de Hyper-V para Ansible ejecuta el siguiente comando en la terminal de WSL:

ansible-galaxy collection install community.general

Crear un archivo Ansible Playbook

nano create_hyperv_vm.yml

Escribe el siguiente contenido en el archivo:

- name: Crear una máquina virtual en Hyper-V

hosts: localhost

tasks:

- **name: Crear una nueva máquina virtual en Hyper-V**

community.general.hyperv_vm:

name: "mi_maquina_virtual"

generation: 2

state: present

memory_mb: 2048

vcpu: 2

switch: "Default Switch" Cambia este nombre si usas otro switch virtual en

Hyper-V

disk_size_gb: 20

image_path: "C:\\HyperVImages\\ubuntu.vhdx" Ruta a la imagen VHDX de la

VM

os_type: Linux

Ejecutar el playbook de Ansible

ansible-playbook create_hyperv_vm.yml