

Actividad 1 Fortran

La historia de linux empieza en 1991 por Linus Torvalds. Es un sistema operativo, al principio se distribuyó como un código fuente y luego como un par de imágenes de disquete descargables , una de arranque y que contiene el kernel de Linux, y la otra con Un conjunto de utilidades y herramientas de GNU para configurar un sistema de archivos.

LINUX (GNU/LINUX) es un Sistema Operativo como Windows. Es decir, Linux es el software para que el ordenador te permita utilizar programas como: editores de texto, juegos, navegadores de Internet, etc. Linux puede usarse mediante un interfaz gráfico al igual que Windows o MacOS, pero también puede usarse mediante línea de comandos como DOS.

Las distribuciones tempranas incluyeron lo siguiente:

- "Boot-root" de HJ Lu, el par de imágenes de disco antes mencionado con el núcleo y las herramientas mínimas absolutas para comenzar, a fines de 1991
- MCC Interim Linux , se puso a disposición del público para su descarga en febrero de 1992
- Softlanding Linux System (SLS), lanzado en 1992, fue la distribución más completa por un corto tiempo, incluido el sistema X Window
- Yggdrasil Linux / GNU / X , una distribución comercial lanzada por primera vez en diciembre de 1992

En términos generales, las distribuciones de Linux pueden ser:

- Comercial o no comercial
- Diseñado para usuarios empresariales, usuarios avanzados o usuarios domésticos.
- Compatible con múltiples tipos de hardware, o específicos de la plataforma
- Diseñado para servidores, equipos de escritorio o dispositivos integrados.
- Propósito general o altamente especializado hacia funcionalidades específicas de la máquina
- Dirigido a grupos de usuarios específicos, por ejemplo a través de la internacionalización y localización de idiomas , o mediante la inclusión de muchos paquetes de producción musical o computación científica
- Construido principalmente para seguridad , usabilidad , portabilidad o exhaustividad
- Lanzamiento estándar o lanzamiento rodante.

Linux tiene más de 600 variantes muchas de ellas conocidas como android que se utiliza en celulares. Además este sistema operativo predomina en las supercomputadoras más potentes del mundo.

Bitácora de notas Actividad 1

Tutorial 1 “La línea de comando ”:

Línea de comando o terminal: es una interfaz de texto basada en el sistema, donde se pueden ingresar comandos para ciertas acciones. Por lo general, un comando es siempre lo primero que escribe. Después de eso tenemos lo que se conoce como argumentos de línea de comando (-l / home / nombre). Es importante saber que estos van separados por espacios (también debe haber un espacio entre el comando y el primer argumento de la línea de comando). El primer argumento de línea de comando (-l) también se conoce como una opción.

Las opciones se usan generalmente para modificar el comportamiento del comando. Las opciones se enumeran antes que otros argumentos y generalmente comienzan con un guión (-).

Muchos comandos te muestran el resultado de su acción pero algunos solo la realizará y no se mostrará nada en pantalla .

Abrir una terminal:

Para sistema operativo linux: Aplicaciones -> Sistema o Aplicaciones -> Utilidades . Alternativamente, hacer clic con el botón derecho en el escritorio y puede haber una opción 'Abrir en terminal'.

The shell bash:

Shell: dentro de una terminal hay lo que se conoce como un shell. Es una parte del sistema operativo que define cómo se comportará la terminal y que aspecto tiene después de ejecutar

Atajo:

Cuando ingresa comandos, se almacenan en un historial. Se puede recorrer este historial con las teclas de flecha arriba y abajo. Por lo tanto, no es necesario volver a escribir los comandos que ingresó anteriormente, generalmente puede presionar la flecha hacia arriba varias veces. También puede editar estos comandos con las teclas de flecha izquierda y derecha para mover el cursor donde desee.

Tutorial 2 “Navegación Básica”:

Entonces, dónde estamos?:

Pwd = imprimir directorio de trabajo, nos dice cuál es nuestro directorio actual

¿Que hay en nuestra ubicación actual?:

ls = Solo hará una lista de nuestra ubicación actual.

argumento de línea de comando (/ etc). Cuando hacemos esto, le dice a ls que no enumere nuestro directorio actual, sino que enumera el contenido de ese directorio.

Caminos:

Siempre que nos referimos a un archivo o directorio en la línea de comando, de hecho nos estamos refiriendo a una ruta. es decir. Una ruta es un medio para llegar a un archivo o directorio particular en el sistema.

- ~ (tilde): este es un acceso directo para su directorio de inicio. por ejemplo, si su directorio de inicio es / home / ryan, entonces puede consultar los documentos del directorio con la ruta / home / ryan / Documents o ~ / Documents
- . (punto): esta es una referencia a su directorio actual. Por ejemplo, en el ejemplo anterior nos referimos a Documentos en la línea 4 con una ruta relativa. También podría escribirse como ./Documents (Normalmente, este bit adicional no es necesario, pero en las secciones posteriores veremos dónde resulta útil).
- .. (dotdot): esta es una referencia al directorio principal. Puede usar esto varias veces en una ruta para seguir ascendiendo en la jerarquía. por ejemplo, si estuviera en la ruta / home / ryan, podría ejecutar el comando ls ../../ y esto haría una lista del directorio raíz.

Vamos a movernos un poco:

cd = Cambiar directorio

Si ejecuta el comando cd sin ningún argumento, siempre lo llevará de regreso a su directorio de inicio.

Camino relativo

Una ubicación de archivo o directorio relativa a donde estamos actualmente en el sistema de archivos.

Camino absoluto

Una ubicación de archivo o directorio en relación con la raíz del sistema de archivos.

Ocupaciones:

- / var / log: almacena archivos de registro para varios programas del sistema. (Es posible que no tenga permiso para ver todo en este directorio. Sin embargo, no deje que eso le impida explorar. Algunos mensajes de error nunca perjudican a nadie).
- / bin: la ubicación de varios programas de uso común (algunos de los cuales aprenderemos en el resto de este tutorial).
- / usr / bin: otra ubicación para los programas en el sistema.

Tutorial 3 “Más sobre archivos”:

Todo es un archivo:

Un archivo de texto es un archivo, un directorio es un archivo, su teclado es un archivo (uno del que solo lee el sistema), su monitor es un archivo (uno en el que solo escribe el sistema) etc. Para empezar, esto ganó No afecta demasiado lo que hacemos, pero tenlo en cuenta, ya que ayuda a comprender el comportamiento de Linux a medida que administramos archivos y directorios.

Linux es un sistema sin extensión:

Una extensión de archivo normalmente es un conjunto de 2 a 4 caracteres después de una parada completa al final de un archivo, que denota qué tipo de archivo es. Las siguientes son extensiones comunes:

- file.exe: un archivo ejecutable o programa.
- file.txt: un archivo de texto sin formato.
- file.png, file.gif, file.jpg: una imagen.

Linux es sensible a mayúsculas y minúsculas:

Linux como tal, es posible tener dos o más archivos y directorios con el mismo nombre pero letras de mayúsculas y minúsculas.

Espacios en los nombres:

Tener cuidado con los espacios en los nombres de los archivos, porque la terminal lo puede identificar como dos archivos de nombre diferente.

Citas:

El primer enfoque implica el uso de citas alrededor de todo el elemento. Puede usar comillas simples o dobles. Cualquier cosa entre comillas se considera un solo elemento.

Personajes de escape:

Otro método es usar lo que se llama un carácter de escape, que es una barra invertida (\). Lo que hace la barra invertida es escapar (o anular) el significado especial del siguiente personaje.

Archivos y directorios ocultos:

Si el nombre del archivo o directorio comienza con a. (punto final), entonces se considera oculto.

Los archivos y directorios pueden estar ocultos por varias razones. Los archivos de configuración para un usuario en particular (que normalmente se almacenan en su directorio de inicio) están ocultos, por ejemplo, para que no se interpongan en el camino del usuario en sus tareas cotidianas.

Para ocultar un archivo o directorio, todo lo que necesita hacer es crear el archivo o directorio con su nombre comenzando con a. o renombrarlo para que sea como tal. Del mismo modo, puede cambiar el nombre de un archivo oculto para eliminar el. y se volverá oculto. El comando ls que hemos visto en la sección anterior no enumerará los archivos y directorios ocultos por defecto. Podemos modificarlo incluyendo la opción de línea de comando -a para que muestre archivos y directorios ocultos.

Tutorial 4 “Páginas del manual”:

¿Qué son exactamente?:

Son un conjunto de páginas que explican cada comando disponible en su sistema

man <comando para mirar hacia arriba>.

Buscando:

Es posible hacer una búsqueda de palabras clave en las páginas del Manual. Esto puede ser útil si no está seguro de qué comando puede querer usar pero sabe lo que quiere lograr. Para ser efectivo con este enfoque, es posible que necesite algunas pruebas.

man-k<Término de búsqueda>

Si desea buscar dentro de una página de manual, esto también es posible. Para hacer esto, mientras se encuentra en la página del manual en particular que desea buscar, presione la barra diagonal '/' seguido del término que desea buscar y presione 'enter'. Si el término aparece varias veces, puede desplazarse por ellos. presionando el botón 'n' para el siguiente.

Tutorial 5 “Manipulación de archivos”:

Hacer un directorio:

Mkdir [opciones] <Directorio>

- mkdir / inicio / ryan / foo
- mkdir ./blah
- mkdir ../dir1
- mkdir ~ / linuxtutorialwork / dir2

Hay algunas opciones útiles para mkdir:

-p: le dice a mkdir que cree directorios principales según sea necesario

-v: hace que mkdir nos diga que es lo que está haciendo

Eliminar un directorio:

rmdir [opciones] <Directorio>

Crear un archivo en blanco:

Toque [opciones] <nombre de archivo>

Touch es un comando que podemos usar para modificar los tiempos de acceso y modificación en un archivo .

Copiar un archivo o directorio:

cp [opciones] <source> <destination>

Tenga en cuenta que tanto el origen como el destino son caminos. Esto significa que podemos referirnos a ellos utilizando rutas tanto absolutas como relativas. Aquí están algunos ejemplos:

- `cp / home / ryan / linuxtutorialwork / example2 example3`
- `cp ejemplo2 ../../backups`
- `cp ejemplo2 ../../backups/example4`
- `cp / home / ryan / linuxtutorialwork / example2 / otherdir / foo / example5`

Cuando usamos `cp`, el destino puede ser una ruta a un archivo o directorio. Si se trata de un archivo, creará una copia de la fuente, nombrará la copia con el nombre de archivo especificado en el destino. Si proporcionamos un directorio como destino, copiará el archivo en ese directorio y la copia tendrá el mismo nombre que la fuente.

En su comportamiento predeterminado, `cp` solo copiará un archivo. Usando la opción `-r`, que significa recursivo, podemos copiar directorios. Recursivo significa que queremos mirar un directorio y todos los archivos y directorios dentro de él, y para subdirectorios, entrar en ellos y hacer lo mismo y seguir haciendo esto.

Mover un archivo o directorio:

Para mover un archivo usamos el comando `mv` que es la abreviatura de mover. Funciona de manera similar a `cp`. Una ventaja es que podemos mover directorios sin tener que proporcionar la opción `-r`.

```
mv [opciones] <source> <destination>
```

Eliminar un archivo y directorios no vacíos:

Al igual que con `rmdir`. El comando para eliminar o eliminar un archivo es `rm`, que significa eliminar.

```
rm [opciones] <archivo>
```

Eliminar directorios no vacíos:

Al igual que otros comandos, `rm` tiene varias opciones. Similar a `cp` significa recursivo. Cuando `rm` se ejecuta con la opción `-r`, nos permite eliminar directorios y todos los archivos y directorios que contiene.

Tutorial 6 “Editor de texto vi”:

Un editor de línea de comando:

Vi es un editor de texto de línea de comando, es un editor de texto sin formato,

Tiene dos modos: inserción y edición, En el modo de inserción, se puede ingresar contenido en el archivo. En el modo de edición, puede moverse por el archivo, realizar acciones como eliminar, copiar, buscar y reemplazar, guardar, etc.

Vi <Archivo>

Al usar vi siempre inicia en modo edición, al presionar la tecla “i” cambia al modo de inserción.

Guardar y salir:

- **ZZ** - Guardar y salir
- **:q!** - descartar todos los cambios, desde el último guardado, y salir
- **:w** - guardar archivo pero no salir
- **:wq** - nuevamente, guarde y salga

Si un comando al inicio tiene “:” significa que tiene que usar enter para que funcione.

Otras formas de ver archivos:

Cat = sirve para ver archivos

gato <archivo>

Para archivos más grandes hay un comando para archivos más grandes

menos <archivo>

Permite moverse hacia arriba y hacia abajo dentro de un archivo con las teclas de flecha. Puede avanzar una página completa con la barra espaciadora o retroceder una página presionando b.

Navegando un archivo en vi:

- Teclas de flecha : mueva el cursor
- j, k, h, l - mueve el cursor hacia abajo, arriba, izquierda y derecha (similar a las teclas de flecha)
- ^ (caret) - mueve el cursor al comienzo de la línea actual
- \$ - mueve el cursor al final de la línea actual
- nG - moverse a la n ésima línea (por ejemplo 5G mueve a quinta línea)
- G - moverse a la última línea
- w - ir al comienzo de la siguiente palabra
- nw - avanzan palabra (por ejemplo, 2w mueve dos palabras hacia adelante)
- b - ir al comienzo de la palabra anterior
- nb - retroceder n palabra
- { - retroceder un párrafo
- } - avanzar un párrafo

Eliminar contenido:

Comandos para eliminar dentro de vi:

- **x** : eliminar un solo carácter
- **nx** : elimina n caracteres (por ejemplo, 5x elimina cinco caracteres)
- **dd** : elimina la línea actual
- **dn** - d seguido de un comando de movimiento. Eliminar a donde el comando de movimiento te habría llevado. (por ejemplo, d5w significa eliminar 5 palabras)

Ruina:

Comandos para hacer cambios:

- **u** - Deshace la última acción (puede seguir presionando u para continuar deshaciendo)
- **U (Nota: capital)** : deshace todos los cambios en la línea actual

Tutorial 7 “Comodines”:

Entonces, ¿Qué son?:

Los comodines son un conjunto de bloques de construcción que le permiten crear un patrón que define un conjunto de archivos o directorios

comandos:

- * - representa cero o más caracteres
- ? - representa un solo personaje
- [] - representa un rango de caracteres

Bajo el capó:

Bash (el programa que proporciona la interfaz de línea de comando) hace la traducción por nosotros. Cuando le ofrecemos este comando, ve que hemos usado comodines y, por lo tanto, antes de ejecutar el comando (en este caso, ls) reemplaza el patrón con cada archivo o directorio (es decir, ruta) que coincida con ese patrón. Emitimos el comando:

- ls b *

sistema traduce esto a:

- ls barry.txt blah.txt bob

y luego ejecuta el programa. El programa nunca ve los comodines y no tiene idea de que los usamos.

Tutorial 8 “Permisos”:

Entonces ¿Qué son?:

Los permisos de Linux son 3 cosas que puede hacer con un archivo: leer, escribir y ejecutar.

Se hace referencia a ellos en Linux con una sola letra cada uno.

- r leer: puede ver el contenido del archivo.
- w escribir: puede cambiar el contenido del archivo.
- x execute: puede ejecutar o ejecutar el archivo si es un programa o script.

Para cada archivo definimos 3 conjuntos de personas para quienes podemos especificar permisos.

- propietario : una sola persona propietaria del archivo. (normalmente la persona que creó el archivo, pero ciertos usuarios pueden otorgarle la propiedad a otra persona)
- grupo : cada archivo pertenece a un solo grupo.
- otros : todos los demás que no están en el grupo o el propietario.

Ver permisos:

`ls -l [ruta]`

los primeros 10 caracteres de la salida son lo que observamos para identificar los permisos.

- El primer carácter identifica el tipo de archivo. Si es un guión (-), entonces es un archivo normal. Si es un anuncio, entonces es un directorio.
- Los siguientes 3 caracteres representan los permisos para el propietario. Una letra representa la presencia de un permiso y un guión (-) representa la ausencia de un permiso. En este ejemplo, el propietario tiene todos los permisos (leer, escribir y ejecutar).
- Los siguientes 3 caracteres representan los permisos para el grupo.
- Finalmente, los últimos 3 caracteres representan los permisos para otros (o para todos los demás). En este ejemplo tienen el permiso de ejecución y nada más.

Cambiar permisos:

Chmod: Cambiar los permisos de un archivo o un directorio.

`chmod [permisos] [ruta]`

chmod tiene argumentos de permiso que se componen de 3 componentes

- ¿Para quién estamos cambiando el permiso? [ugo]: usuario (o propietario), grupo, otros, todos
- ¿Estamos otorgando o revocando el permiso, indicado con un signo más (+) o menos (-)
- ¿Qué permiso estamos estableciendo? - leer (r), escribir (w) o ejecutar (x)

Configuración de permisos abreviados:

Nuestro sistema numérico típico es decimal. Es un sistema de números de base 10 y como tal tiene 10 símbolos (0 - 9) utilizados. Otro sistema de números es octal, que es la base 8 (0-7). Ahora sucede que con 3 permisos y cada uno activado o desactivado, tenemos 8 combinaciones posibles (2^3). Ahora también podemos representar nuestros números usando binarios que solo tienen 2 símbolos (0 y 1). El mapeo de octal a binario está en la tabla a continuación.

Podemos representar los 8 valores octales con 3 bits binarios y que cada combinación posible de 1 y 0 está incluida en él. Entonces tenemos 3 bits y también tenemos 3 permisos. Si piensa que 1 representa activado y 0 desactivado, se puede usar un solo número octal para representar un conjunto de permisos para un conjunto de personas. Tres números y podemos especificar permisos para el usuario, grupo y otros.

Permisos para directorios:

Comandos:

- **r** : tiene la capacidad de leer el contenido del directorio (es decir, hacer un ls)
- **w** : tiene la capacidad de escribir en el directorio (es decir, crear archivos y directorios)
- **x** : tiene la capacidad de ingresar a ese directorio (es decir, cd)

Tutorial 9 “Filtros”:

Entonces, ¿Qué son?:

Los filtros es un programa que acepta datos textuales y luego lo transforma de una manera particular, son una forma de tomar datos sin procesar

Cabeza:

Head es un programa que imprime las primeras líneas de su entrada, por defecto imprimirá las primeras 10.

cabeza [-Número de líneas a imprimir] [ruta]

Cola:

Es lo opuesto a la cabeza, Tail imprime las últimas 10 líneas de la entrada.

cola [-número de líneas a imprimir] [ruta]

Ordenar:

Ordenará su entrada alfabéticamente

ordenar [-opciones] [ruta]

Nl:

Nl representa las líneas numéricas

Baño:

wc es recuento de palabras

wc [-opciones] [ruta]

Cortar:

Cut es un programa para usar si tu contenido está separado en campos (columnas) y solo quieres ciertos campos.

cortar [-opciones] [ruta]

Sed:

Stream Editor nos permite hacer una búsqueda y reemplazar nuestros datos.

```
sed <expresión> [ruta]
```

Una expresión básica tiene el siguiente formato:

- s / buscar / reemplazar / g

Uniq:

Uniq significa único y su trabajo es eliminar líneas duplicadas de los datos. Sin embargo, esas líneas deben ser adyacentes (es decir, una después de la otra).

```
uniq [opciones] [ruta]
```

Tac:

El programa tac es en realidad gato en reversa. Hace lo contrario de gato. Dada la información, imprimirá la última línea primero, hasta la primera línea.

```
tac [ruta]
```

Tutorial 10 “Grep y expresiones regulares”:

Entonces, ¿Qué son?:

Nos permiten crear un patrón. Los Re generalmente se usan para identificar y manipular datos específicos. Es posible que deseemos identificar cada línea que contenga una dirección de correo electrónico o una URL en un conjunto de datos.

eGrep:

Egrep es un programa que buscará un conjunto de datos dado e imprimirá cada línea que contenga un patrón dado. Es una extensión de un programa llamado grep. Tiene muchas opciones de línea de comando que modifican su comportamiento.

egrep [opciones de línea de comando] <patrón> [ruta]

Expresiones regulares:

Comandos:

- **.** (punto): un solo carácter.
- **?** - el carácter anterior coincide sólo 0 o 1 veces.
- ***** - el carácter anterior coincide con 0 o más veces.
- **+** : el carácter anterior coincide 1 o más veces.
- **{n}** : el carácter anterior coincide exactamente n veces.
- **{n, m}** : el carácter anterior coincide al menos n veces y no más de m veces.
- **[agd]** : el personaje es uno de los incluidos entre corchetes.
- **[^agd]** : el personaje no es uno de los incluidos entre corchetes.
- **[cf]** - el guión dentro de los corchetes funciona como un rango. En este caso significa las letras c, d, e o f.
- **()** : nos permite agrupar varios caracteres para que se comporten como uno solo.
- **EI |** (símbolo de tubería): la operación lógica OR.
- **^** : coincide con el comienzo de la línea.
- **\$** : coincide con el final de la línea.