Logotipo

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

**CURSO TEÓRICO-PRÁCTICO:**

**EPIDEMIOLOGÍA GENÓMICA Y DESCUBRIMIENTO DE PATÓGENOS MEDIANTE SECUENCIACIÓN DE PRÓXIMA GENERACIÓN**

**/ GENOMIC EPIDEMIOLOGY AND PATHOGEN DISCOVERY BY NEXT GENERATION SEQUENCING**

**Noviembre 8 - 12, 2021**

**Introducción a la línea de comandos de Linux**

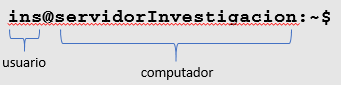
Los comandos se encuentran en letra Courier dentro de cajas.

1. **Identifique su nombre de usuario y nombre del equipo en la terminal**

Comandos: whoami

hostname

Este comando nos devuelve el nombre de usuario





1. **¿En qué carpeta estamos trabajandor?**

Con el comando pwd podemos ver en qué directorio estamos ubicados y conocer su PATH completo.



1. **Cree un directorio nuevo**

mkdir crea directorios. Podemos crear un directorio para el taller

mkdir taller



También podemos intentar crear subdirectorios

mkdir taller/prueba/punto1



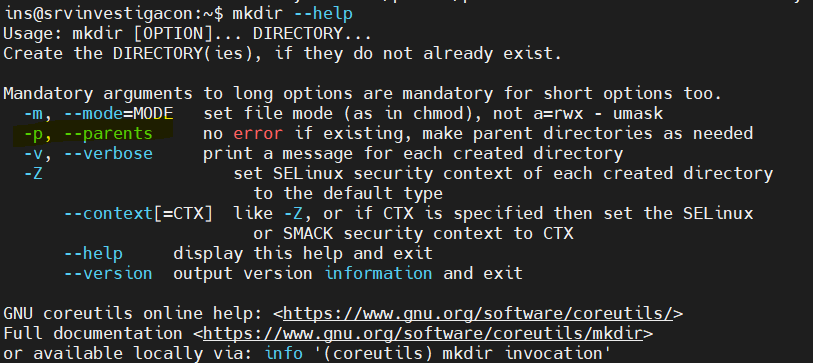
Pero el comando no nos permite crear el directorio *punto1* dentro del directorio inexistente *prueba*. **Veamos la ayuda del comando**

1. **¿Cómo veo la ayuda o instrucciones de los comandos?**

Opcion en los commandos: --help

Cada comando tiene una opción para desplegar la ayuda. La mayoría tiene la opción ayuda agregando --help, o -h.

mkdir --help



Viendo la ayuda del comando encontramos las opciones que podemos usar con el comando mkdir. Entre estas vemos la opción –p que se asegura de crear directorios parentales si no existen.

mkdir –p /taller/prueba/punto1



1. **Ingrese al directorio *taller* que acabamos de crear**

Para entrar a los directorios en nuestra ubicación usamos el comando cd.

cd taller



Vemos que nuestra ubicación en la consola ha cambiado a ~/taller

El símbolo ~ representa el directorio home.

También podemos ingresar al directorio que queramos definiendo el PATH completo.

cd /home/mi\_usuario/taller

EL nombre de usuario lo podemos ver en la terminal, o con el comando whoami



1. **Revise qué contiene el directorio actual**

Ahora que estamos dentro del directorio queremos ver su contenido. El comando ls nos muestra una lista del contenido



1. **Revise los permisos, tamaño y fecha de modificación de los archivos**

Comando: ls -lh

Con la opción --help podemos ver las opciones para visualizar esta lista.

Si queremos activar varias opciones al tiempo podemos usar un solo guion y las letras correspondientes.



Pregunta: ¿Qué nos muestra el comando con las opciones –lh?

Pregunta: ¿Cómo podemos ver archivos ocultos?

1. **Sacar en pantalla un texto**

Con echo podemos ver en pantalla el valor de variables o mostrar en el stdout.

Hay variables del sistema que podemos ver con echo, como la ubicación actual o el id del usuario $UID

echo $UID



1. **Guarde la salida (stdout) en un archivo**

La salida que veríamos en el stdout la podemos guardar con el símbolo >

echo “hello world” > hello.txt



Podemos agregar líneas al archivo hello.txt usando el símbolo >>

echo “hola mundo” >> hello.txt

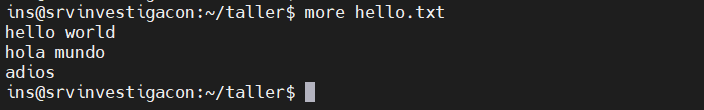
echo “adios” >> hello.txt



1. **Ver el contenido de archivos**

Los comandos cat, more, less, head, tail nos muestran el contenido de archivos de texto de diferentes formas.

more hello.txt



cat también puede ser usado para concatenar archivos de texto

con head podemos ver las primeras líneas de un archivo

con tail podemos ver las líneas finales

1. **Clonar repositorios de GitHub**

Para empezar, debemos descargar los archivos que usaremos en el taller. Los archivos están depositados en un repositorio de GitHub, desde donde los copiaremos

git clone https://github.com/dfbautista/Curso-Epidemiologia-Genomica-Practica-Bioinformatica/tree/main/Intro\_LineaComandos\_Linux

el contenido de este repositorio será clonado al directorio donde estemos ubicados.

1. **Descargar archivos**

Podemos usar el comando wget para descargar archivos.

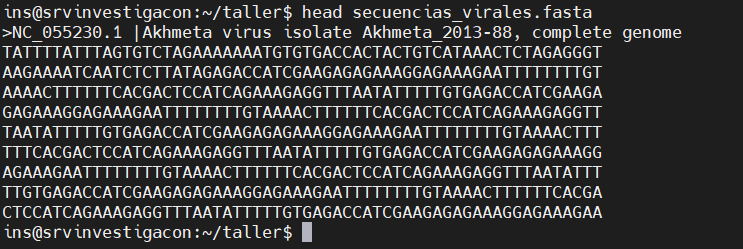
wget https://github.com/dfbautista/Curso-Epidemiologia-Genomica-Practica-Bioinformatica/raw/main/Intro\_LineaComandos\_Linux/secuencias\_virales.fasta

En la carpeta que descargamos hay un archivo fasta llamado **secuencias\_virales.fasta**.

1. **Pregunta: ¿Cuántas secuencias hay en este archivo?**

Primero veamos cómo es el archivo:

head secuencias\_virales.fasta



1. **grep – Hacer una búsqueda dentro de un texto**

Busca un patrón dado en el archivo o texto que indiquemos.

Podemos buscar en nuestro archivo hello.txt todas las veces que aparezca la letra h.

grep “h” hello.txt



Este comando puede ser útil para explorar archivos de texto, por ejemplo archivos fasta.

La manera más fácil de ver cuantas secuencias hay sería buscando el símbolo “>” de cada header en el fasta y contándolos

grep -c “>” secuencias\_virales.fasta



1. **Escribir scripts y archivos de texto con nano**

Podemos crear archivos de texto con diferentes herramientas como nano o vim. nano es sencilla de usar. Llamamos el comando nano y el nombre del archivo que vamos a crear o editar.

nano archivo.txt

Acá podemos escribir el contenido de nuestro archivo.

Vamos a escribir el encabezado de un script de bash

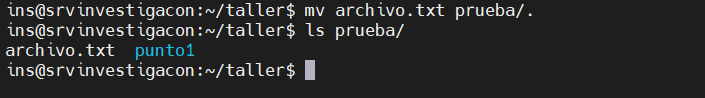
#!/bin/bash

Para salir damos Ctrl+x, luego confirmamos que queremos guardar los cambios

1. Mover archivos y cambiarles el nombre

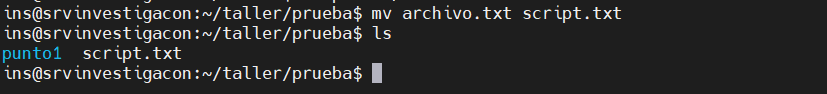
Estas dos acciones las podemos realizar con el comando mv

Sí queremos cambiar el archivo de ubicación es como si reescribieramos su PATH



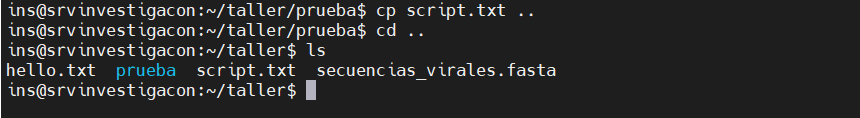
Acá hemos movido el archivo al directorio prueba/

También podemos cambiar el nombre, o si quisiéramos realizar las dos acciones simultáneamente.



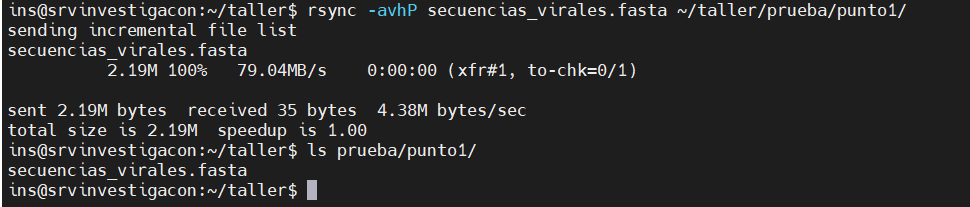
1. Copiar archivos con cp

Para copiar archivos, llamamos el comando y luego definimos el path del archivo que queremos copiar y luego el path de destino.



1. Copiar archivos con rsync (sincronización de directorios)

rsync es una herramienta que permite sincronizar el contenido de directorios. Esto quiere decir que podemos ir agregando archivos sin tener que copiar nuevamente todo un directorio. Además tiene opciones que nos permiten visualizar el proceso.



1. Eliminar archivos

Eliminar archivos en Linux es potencialmente peligroso.

El comando rm nos permite eliminar archivos sencillos.

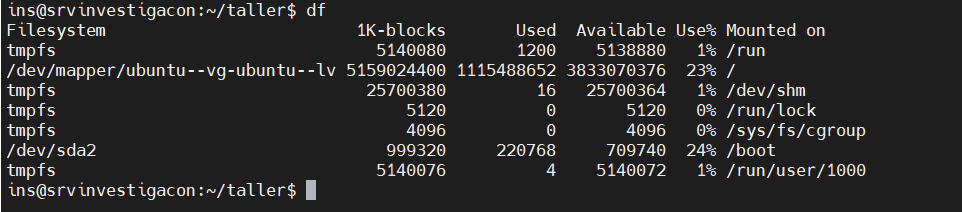
Si queremos eliminar directorios vacíos podemos usar rm –d

Para eliminar directorios y su contenido podemos usar rm –r

Linux no tiene papelera de reciclaje, así que al eliminar, esto se hace de manera permanente.

1. Ver el espacio disponible en disco

Comando: df



1. Ver el tamaño de un directorio

du –sh

¿Qué tamaño tienen los directorios que creamos?



1. Realizar un conteo de palabras o líneas de un archivo

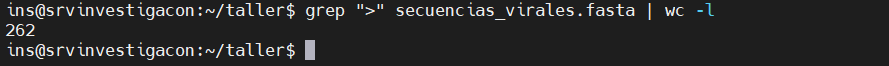
Podemos realizar un conteo del número de letras o palabras con el comando wc

Sí combinamos este comando con uno de búsqueda, por ejemplo grep, podemos obtener información de un archivo de texto.

Por ejemplo, ¿Cuántas bases tiene la secuencia del archivo secuencia\_nc\_055230\_1.fasta?

Para conectar la salida de un comando con la entrada de otro podemos usar el símbolo del pipeline |.

| wc –l



1. **Pausar la terminal por un tiempo**

A veces es útil hacer una pausa entre comandos. Para esto podemos usar el comando sleep, y definir el tiempo que queremos que espere el sistema en horas (h), minutos (m), o segundos (s).

sleep



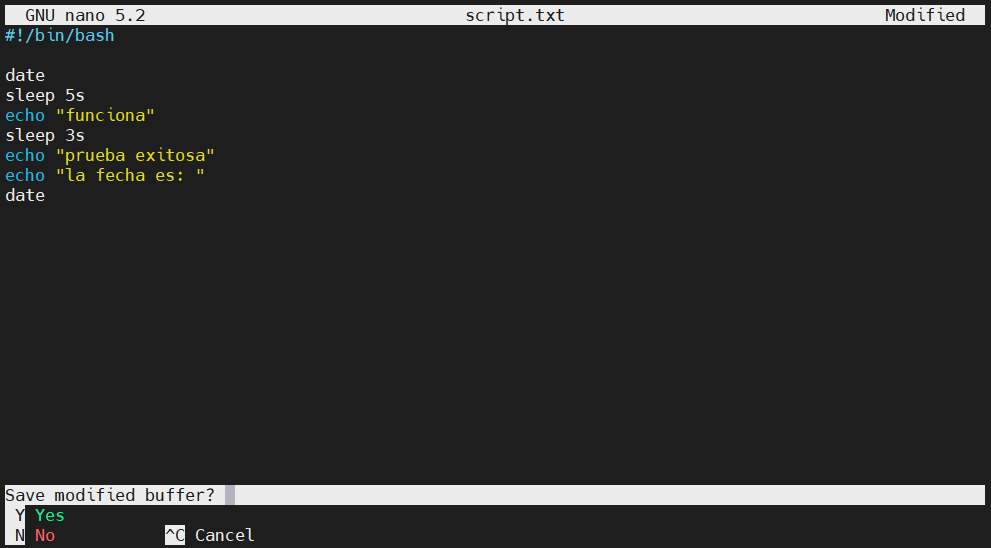
1. **Crear un script de bash con instrucciones**

En muchas ocasiones debemos realizar una tarea o ejecutar un comando repetidas veces para distintos archivos de entrada. Para realizar esto debemos saber cómo se estructuran los scripts de bash.

En el editor de texto podemos agregar en la primera el término #!/bin/bash

Esto le indicará al sistema que lo que está escrito en el archivo son instrucciones en bash.

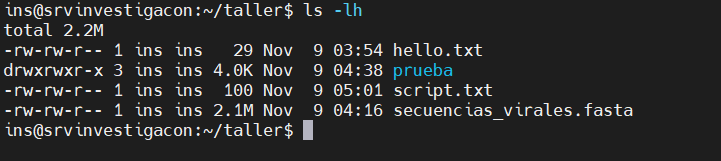
Probemos con el siguiente script:

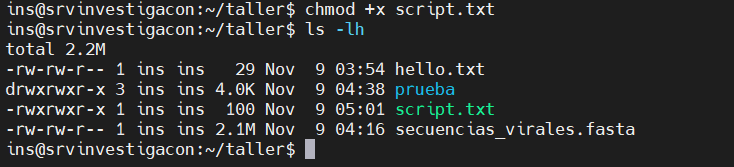


1. **Cambiar permisos de lectura, escritura y ejecución de archivos**

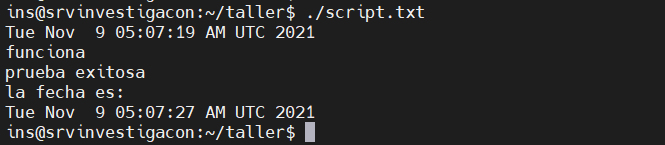
Para poder ejecutar nuestros scripts deben tener los permisos habilitados. Podemos usar ls –lh para revisar que permisos tenemos activos y el comando chmod para modificar los permisos.

chmod +x script.txt



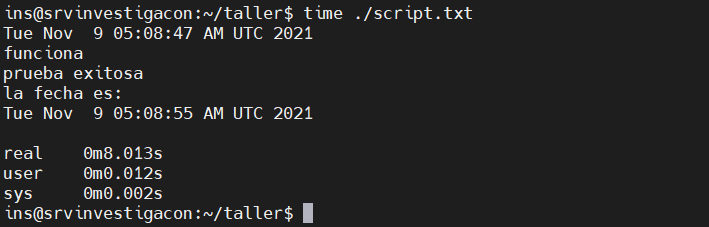


Para ejecutar nuestro script es necesario llamarlo con su path completo. Lo más sencillo si está en nuestra ubicación actual es usar el símbolo .



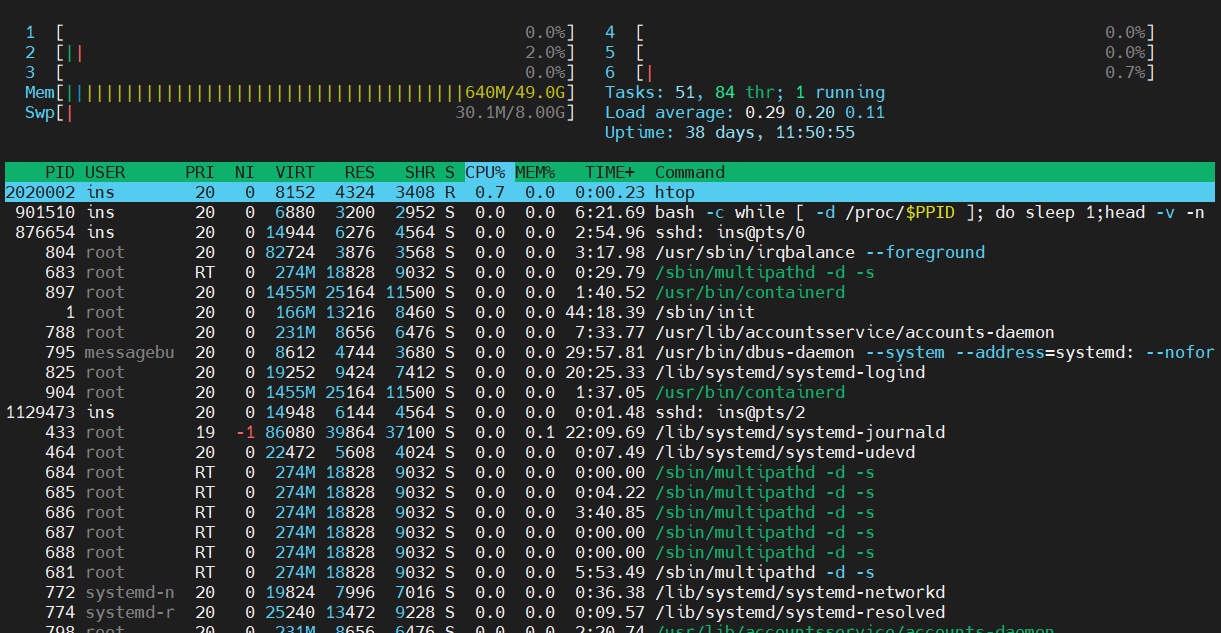
1. **¿Cuánto se demora un comando en correr?**

Si queremos saber cuánto tarda un comando en completar su función podemos agregar el termino time antes del comando o script. Al finalizar su ejecución obtendremos un reporte del tiempo que gastó.



1. **Ver qué procesos están corriendo**

htop



1. **Detener un comando en ejecución**

Si un comando o script no nos responde o simplemente queremos detenerlo podemos presionar las teclas Ctrl y la letra C simultáneamente para detener la ejecución.

1. **Buscar archivos**

Podemos usar el comando find para encontrar archivos con cierto nombre o extensión. El comando requiere saber donde estamos buscando y qué estamos buscando. Para esto tiene distintas opciones que nos permiten buscar por nombre, tipo de archivo y más.

¿Cuántos archivos fasta hay en la ubicación actual?

find . –name “\*.fasta”

¿Cuáles son los directorios en mi ubicación?

find . –type d

1. **Ejecutar un comando para cada archivo encontrado en una búsqueda**

Primero creemos un script que correr.

nano print.sh

#!/bin/bash

sam ="$1" # parametro archivo de entrada

printf "%b" "\n Mi archive sam ${sam} tiene este número de filas:"

wc –l $sam

Ahora debemos dar permiso de ejecución a nuestro script usando chmod

Finalmente, corramos nuestro commando de busqueda y ejecución haciendo un pipe del comando find al comando xargs

find . –name “\*.sam” | xargs print.sh

1. **Imprimir el historial de comandos que hemos ejecutado**

Si necesitamos un registro de los comandos que hemos ejecutado podemos recurrir al comando history

history

1. **Caracteres especiales de bash**

Linux es sensible a lo que escribamos en mayúscula y minúscula. Adicionalmente, algunos caracteres están reservados por el sistema y son de uso frecuente. Los comodines, atajos a directorio home y parental, y otros símbolos nos ayudan a ser más productivos.

|  |  |  |  |
| --- | --- | --- | --- |
| Caracter | Descripción | Caracter | Descripción |
| ~ | Directorio home | . | Directorio actual |
| .. | Directorio parental | # | Comentario |
| $ | Variable | & | Trabajo en background |
| \* | Comodín | ? | Comodín de una letra |
| ( | Inicio sub-shell (anidar comandos) | ) | Fin sub-shell (anidar comandos) |
| \ | Comillas al siguiente caracter | | | Pipe (el stdout será entrada del siguiente comando) |
| ; | Separador de comandos | ! | Negación (No lógico) |
| > | Redirigir stdout | >> | Agregar stdout a archivo |