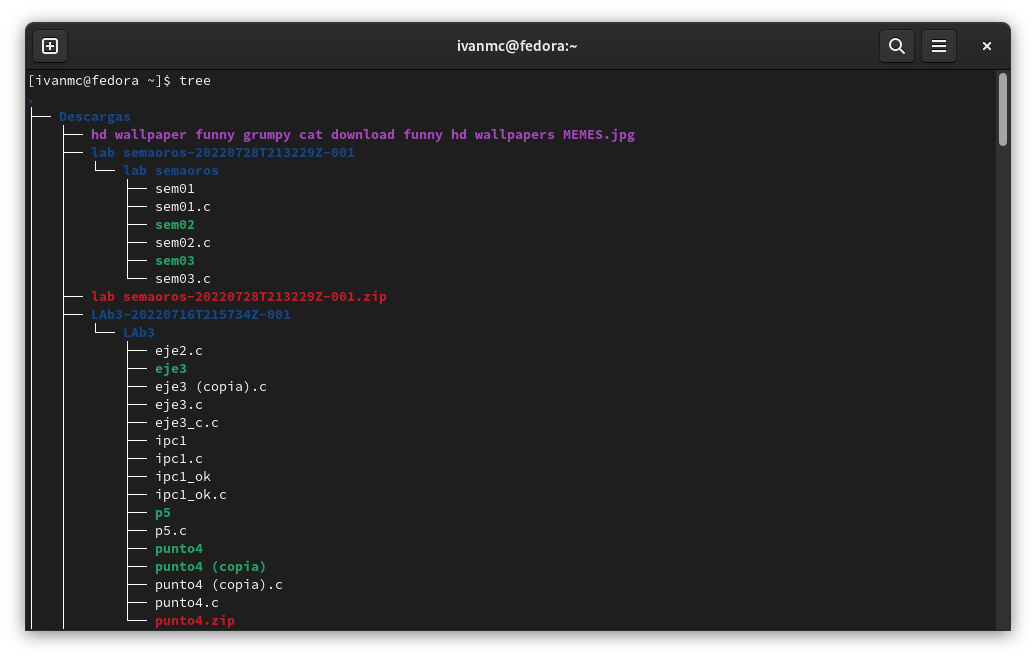
## 1.3 Ejercicios

1. En una terminal (bash) examine el árbol de directorio actual y determine la diferencia entre rutas absolutas y relativas. Adjunte las imágenes de los resultados obtenidos.

**Desarrollo:**

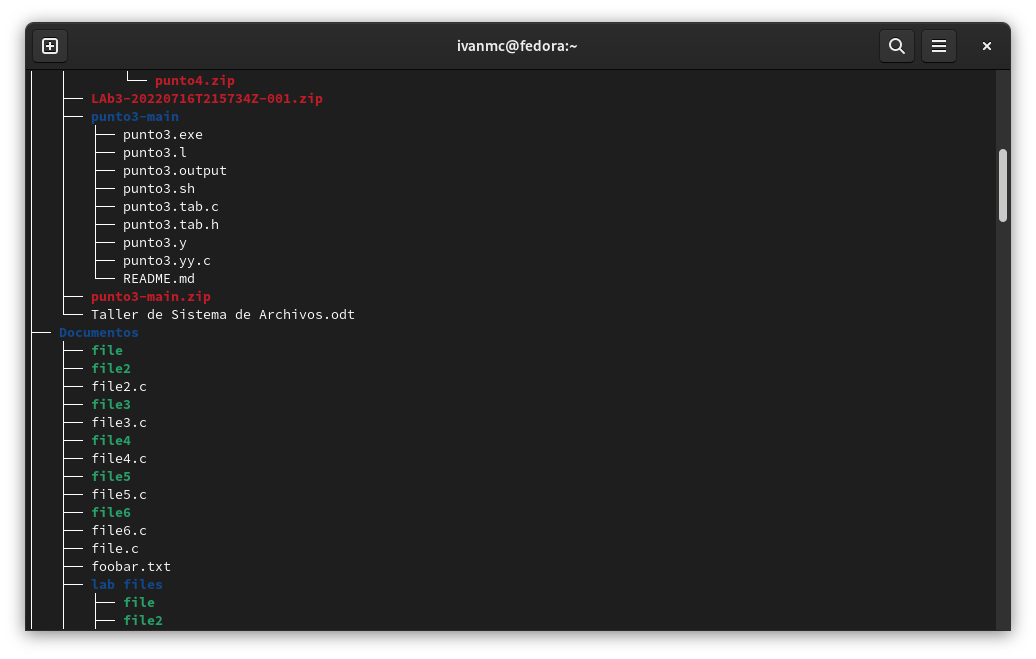
Para este ejercicio, ejecutamos el comando tree, para así poder ver la estructura y jerarquización de las directorios y archivos que se tiene en la carpeta personal, de esta forma, dando como resultado, el podemos visualizar el nivel de cada directorio y darnos un primer vistazo al tema de las rutas (tanto absolutas y relativas), la ejecución del comando tree, la podemos ver a continuación en las siguientes imágenes:

**Figura. Ejecución del comando tree parte 1**



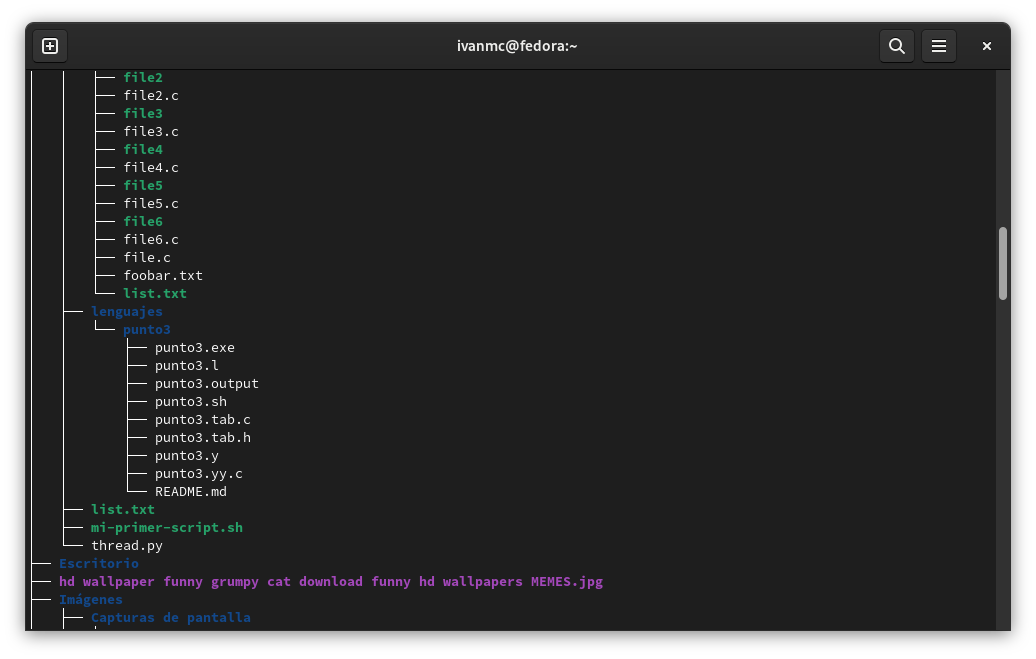
*Nota: autoría propia*

**Figura. Ejecución del comando tree parte 2**



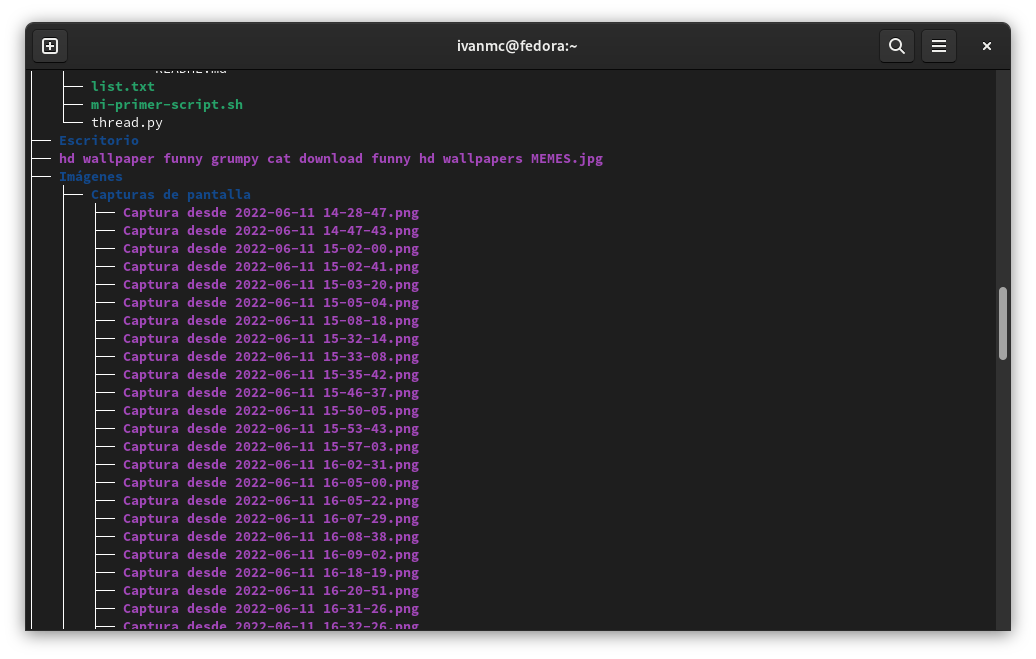
*Nota: autoría propia*

**Figura. Ejecución del comando tree parte 3**



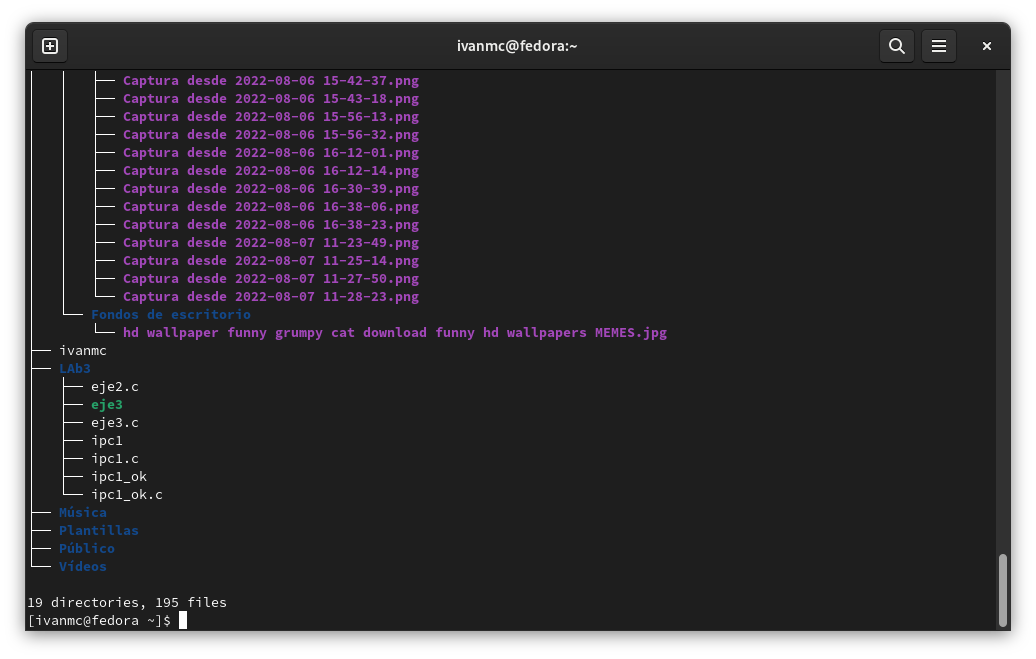
*Nota: autoría propia*

**Figura. Ejecución del comando tree parte 4**



*Nota: autoría propia*

**Figura. Ejecución del comando tree parte 5**



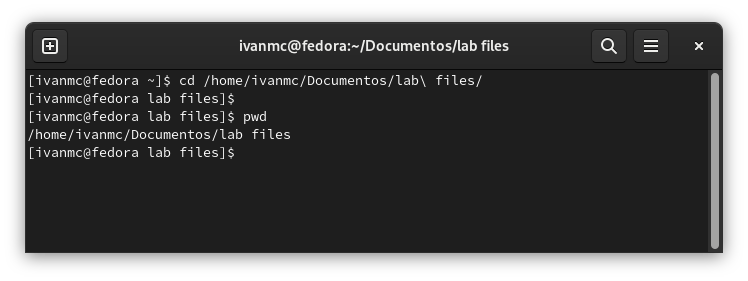
*Nota: autoría propia*

Como notamos, vemos una jerarquía entre los directorio y la forma de navegar entre ellos, por ejemplo, para el caso en el cual, queremos llegar a la carpeta “lab files”, por medio de una ruta absoluta tendríamos que hacerlo por medio del comando:

cd /home/ivanmc/Documentos/lab \files/

Como notamos, estamos iniciando las ruta desde su origen hasta el destino que queremos llegar, pasando por cada uno de los directorios que contienen dicha carpetas necesarias para llegar al punto de destino, como lo podemos notar en la siguiente figura

**Figura. uso de una ruta absoluta**



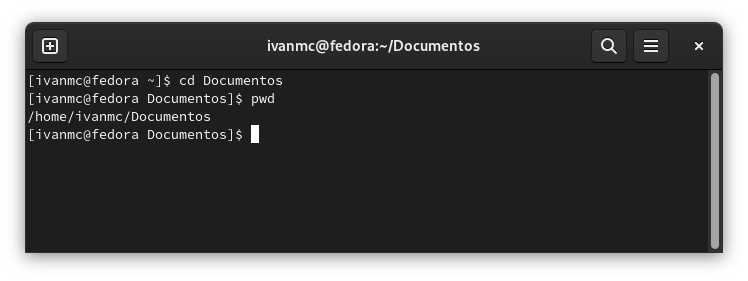
Nota: autoría propia

En el otro caso, si queremos llegar a la carpeta de “Documentos”, por medio de una ruta absoluta, en este caso únicamente ejecutamos el siguiente comando:

cd Documentos

Donde notamos que ya no es necesario especificar todas las carpetas por las que tiene que pasar desde la raíz, para llegar al directorio objetivo, como lo podemos ver en la siguiente figura:

**Figura. uso de una ruta relativa**



Nota: autoría propia

Entonces, podemos diferenciar los tipos de rutas de la siguiente manera:

**Ruta absoluta:** es una estructura jerárquica que tiene una raíz, que se indica con la barra (/). La ruta absoluta a un directorio indica todos los directorios por los que hay que pasar empezando desde la raíz del sistema de archivos (/)

**Ruta relativa:** La ruta relativa a un directorio indica el camino hacia el directorio al que queremos cambiar o alcanzar, pero basándose en el directorio desde el que se ejecuta el comando, es decir, desde el directorio actual o de trabajo.

Donde la principal diferencia entre estos dos tipos de rutas radica en que **las rutas absolutas suelen ser más largas que las rutas relativas, pero tienen la ventaja de que siempre funcionan, independientemente del lugar del árbol de directorios desde el que se ejecute el comando.**

2. Una forma de utilizar la asignación contigua del disco y evitar sectores de huecos (holes) es compactar el disco cada vez que se elimina un archivo. Dado que todos los archivos son contiguos, la copia de un archivo requiere un tiempo de búsqueda, un retraso de rotación para leer el archivo y la velocidad de transferencia. Este mismo procedimiento se utiliza en la escritura de archivos. Suponga:

- Un tiempo de búsqueda de 5 ms

- Un retraso de rotación de 4 ms

- Una velocidad de transferencia de 8 Mbps

- Un tamaño de archivo promedio de 8 KB

¿Cuánto tiempo lleva leer un archivo en la memoria principal y luego volver a escribirlo al disco en una nueva ubicación?.

**Desarrollo:**

* Entonces, para iniciar la transferencia se requiere de 9 ms, que se trata sencillamente de la suma de los 5 ms de búsqueda y los 4 ms de retraso de rotación.
* Y para leer 213 bytes (8 KB) a una velocidad de transferencia de 223 bytes/s (8 MB/s) se requiere de 2-10 segundos, o más aproximadamente, alrededor de 0,977 ms.
* Por lo tanto, el tiempo total de búsqueda, rotación y transferencia es de 9.977 ms que es la suma del tiempo para iniciar la transferencia (los 9 ms) mas el tiempo para leer 8KB (los 0,977 ms), quedando ( 9ms + 0.977 ms)
* Por ende, la escritura de archivos lleva otros 9,977 ms. Y eso significa que al copiar un archivo promedio, toma alrededor de **19,954 ms**.

Usando estos datos ¿Cuánto tiempo tomaría compactar la mitad de un disco de 32 GB?

**Desarrollo:**

* Para compactar la mitad de un disco de 32 GB implicaría copiar 16 GB de almacenamiento
* Estas 16 GB de almacenamiento serian 16777216 kilobytes
* Al dividir los 16777216 kilobytes en 8 KB (que es el tamaño promedio de un archivo), obtenemos 2097152 archivos.
* Es decir, tenemos archivos
* Tenemos en cuenta que si para copiar se toma 19.954 mseg por archivo, entonces para obtener el tiempo para copiar archivos aplicamos una regla de tres

* Realizando la operación obtenemos: 41,846,571.008 ms
* Es decir 41846.57 segundos
* Es decir **11.624 horas**

# 3. Conclusiones

* Según algunas consultas que hicimos para el desarrollo de este laboratorio, pudimos notar que el sistema de archivos de igual forma tiene una importancia a nivel de la seguridad de un sistema, más específicamente en la protección ante posibles daños físicos de los datos o el acceso indebido a los mismos. Ya que si el sistema operativo no está en capacidad de identificar a los usuarios que acceden al sistema, de nada serviría el disponer de mecanismos de protección buenos.
* Como vemos en uno de los ejercicios del punto 1.3, más específicamente en el que se nos pide calcular el tiempo para compactar la mitad de un disco de 32 GB, notamos que no es muy buena idea compactar el disco después de cada eliminación de archivos no es una gran idea.
* De igual forma, notamos que un sistema de archivos permite funcionalidades como mecanismos para evitar la fragmentación, capacidad de enlaces simbólicos o duros, integridad del sistema de archivos (Journaling), soporte para archivos dispersos, soporte para cuotas de discos y soporte de crecimiento del sistema de archivos nativo
* En los sistemas de archivos jerárquicos, notamos que la estructura de los directorios funciona de una manera jerárquica, donde podemos dar la ubicación de un archivo o carpeta por medio de una ruta, que es una cadena de texto que consiste en una sucesión de nombres de directorios y subdirectorios, ordenados jerárquicamente de izquierda a derecha y separados por algún carácter especial, comúnmente la barra diagonal / o barra diagonal invertida \, y puede terminar en el nombre de un archivo presente en la última rama de directorios especificada.

# 

# Referencias

[1]. Go Linux Cloud. (2022). 10 basic & powerful commands to check file system type in Linux/Unix. Disponible en: <https://www.golinuxcloud.com/commands-linux-mounted-check-file-system-type/>.

[2]. Geeks for geeks. (2021). Input-output system calls in C | Create, Open, Close, Read, Write. Disponible en: <https://www.geeksforgeeks.org/input-output-system-calls-c-create-open-close-read-write/>

[3]. Silberschatz, Galvin and Gagne. A. Modern Operating Systems. 2018.

[4]. Tanenbaum, Modern Operating Systems, 2015.

[5]."Sistemas de archivos: qué son y cuáles son los más importantes". IONOS Digitalguide. https://www.ionos.es/digitalguide/servidores/know-how/sistemas-de-archivos/ (accedido el 6 de agosto de 2022).