1. **¿Cuáles son las prioridades mínima y máxima de un proceso en Linux? ¿Cuál es la prioridad por defecto de un proceso en Linux?**
   La prioridad máxima que puede tener un proceso en Linux es -20 y la mínima es 19.
   La prioridad por defecto de un proceso en Linux es 0.

2. **Con el comando `nice`, ¿Cuál es el valor por defecto de *niceness*, si no lo indicamos explícitamente?¿Cuál es la máxima prioridad que puede imponer un usuario que no es root con el comando `nice`?¿Cuál es la diferencia entre `nice` y `renice`?¿Como se utiliza cada uno?**
   El valor por defecto de *niceness* con el comando `nice` es 10.
   La máxima prioridad que puede imponer un usuario que no es root con el comando `nice` es 0.
   La diferencia entre `nice` y `renice` es que `nice` se utiliza para arrancar un nuevo proceso con una prioridad específica, mientras que `renice` se utiliza para cambiar la prioridad de un proceso existente. En `nice` indicamos el nombre de la utilidad que queremos arrancar, mientras que en `renice` indicamos el PID del proceso concreto en ejecución.
   `nice [-n niceness] utilidad`, por ejemplo: `nice -n 12 yes` (ejecuta yes con prioridad 12)
   `renice [prioridad] -p PID`, por ejemplo: `renice 5 -p 2365` (cambia a 5 la prioridad del proceso con PID 2365)

3. **¿Qué usuarios pueden terminar un proceso?¿Cuál es la diferencia entre `Kill [PID]`, `kill -9 [PID]` y `kill -15 [PID]`?**
   Cualquier usuario puede terminar un proceso que sea propiedad de ese usuario. Sin embargo, solo el usuario root puede terminar cualquier proceso en el sistema.
   Kill [PID] envía la señal SIGTERM al proceso con el PID especificado, lo que le indica que termine su ejecución de forma ordenada. kill -9 [PID] envía la señal SIGKILL, que termina el proceso inmediatamente sin permitir que realice ningún proceso de limpieza. kill -15 [PID] es equivalente a kill [PID], ya que SIGTERM es la señal predeterminada enviada por kill.

4. **¿Qué hace el comando `ps -efl`?**
   El comando ps -efl muestra una lista detallada de todos los procesos en el sistema, incluidos los detalles de la jerarquía de procesos, los ID de usuario, los ID de grupo, las prioridades, etc.

5. **En la máquina virtual Ubu_*Iniciales* que creaste para la actividad, realiza los siguientes pasos:**
   - **Crea un grupo de usuarios *alumni***
     `[sudo] addgroup alumni`
   - **Crea un usuario *alumno_Iniciales* incluido en el grupo *alumni*, con contraseña "siiuser01".**
     `[sudo] adduser alumno_XYZ --ingroup alumni`
     Nos pide que indiquemos la contraseña (dos veces) y los datos del usuario.
   - **Cambia la contraseña de alumno_Iniciales por "siiuser02".**
     `[sudo] passwd nombre_usuario`
     Nos pedirá que introduzcamos dos veces la nueva contraseña.
   - **Ejecuta el siguiente comando: `id alumno_Iniciales > alumnos.txt` (se creará un archivo de texto con la respuesta al comando `id alumno` en la carpeta en la que estés).**
   - **Ejecuta `ls -l` para mostrar el contenido de la carpeta con información acerca de permisos, usuarios propietarios y grupos propietarios.**
     ```
     -rw-rw-r-- 1 vboxuser vboxuser  184 may  9 19:33 alumnos.txt
     ```
   - **Cambia el usuario y grupos propietarios del archivo alumnos.txt por *alumno_Iniciales* y *Alumni*, respectivamente.**
     `[sudo] chown alumno_XYZ:alumni alumnos.txt`
   - **Ejecuta `ls -l`, nuevamente, y comprueba que los propietarios han cambiado para el archivo alumnos.txt.**
     ```
     -rw-rw-r-- 1 alumno_epv alumni    184 may  9 19:33 alumnos.txt
     ```
6. **Crea un grupo de usuarios *apololsobresainte*. Añade a *alumno_Iniciales* al grupo recién creado. Comprueba que se ha añadido ejecutando el comando `groups alumno_Iniciales`.**
   `[sudo] addgroup apolosobresainte`
   `[sudo] adduser alumno_XYZ apolosobresainte`
   `groups alumno_XYZ`

7. **Teniendo en cuenta las capturas que se incluyen a continuación, ¿Qué podemos saber del usuario agd?**

   ```
   agd:x:1002:1003:Aquel Guicho Dacola,,,:/home/agd:/bin/bash
   ```
   (fila de la respuesta al comando `cat /etc/passwd`)
   ```
   alumni:x:1001:
   malaxente:x:1002:
   festeiros:x:1003:
   repunantes:x:1004:
   aghonias:x:1005:
   estraghados:x:1006:
   ```
   (filas de la respuesta al comando `cat /etc/group`)

   Sabemos que agd es el usuario con UID 1002, está activo, pertenece al grupo de *festeiros*, se llama Aquel Guicho Dacola, no nos ha dado su teléfono y su directorio $HOME es /home/agd

8. **Siguiendo con el usuario agd de la pregunta anterior, ¿Qué comando tendríamos que utilizar para que `cat /etc/passwd` muestre el resultado de la imagen a continuación?**

```
agd:x:1002:1006:Aquel Guicho Dacola,,,:/home/agd:/bin/bash
```

Nuestro amigo agd ha pasado del grupo de festeiros al grupo de estraghados, por tanto, el comando usado ha sido: `[sudo] usermod agd -g estraghados`

9. **Elimina el usuarios Alumno_Iniciales y el grupo Alumni creados en el ejercicio 5, así como el directorio $HOME correspondiente a dicho usuario.**

```
sudo userdel -r alumno_XYZ
sudo groupdel alumni
```

10. **¿Puede un usuario tener identificador de grupo (GID) y usuario (UID) a cero? ¿De qué usuario se trataría?**

Sí, el superusuario root (se puede ver si ejecutamos cat /etc/passwd, por ejemplo)

11. **¿Qué comando usarías para formatear la segunda partición lógica del tercer disco duro?**

mkfs /dev/sdc6

12. **Indica la nomenclatura que tendrán en Ubuntu las siguientes particiones:**
    - **Primera partición primaria del segundo disco duro: sdb1**
    - **Primera partición lógica del segundo disco duro: sdb5**
    - **Tercera partición lógica del primer disco duro: sda7**
    - **Tercera partición primaria del primer disco duro: sda3**

13. **¿Qué comando podrías utilizar para montar una partición en el punto de montaje */mnt/carpeta*?¿Y para desmontarlo de dicho punto de montaje?¿A qué nos referimos cuando hablamos de "punto de montaje"?**

Siendo x=disco duro e y=partición:
- **Para montarlo en "/mnt/carpeta": `(sudo) mount /dev/sdxy /mnt/carpeta`**
- **Para desmontarlo: `(sudo) umount /mnt/carpeta`**

`Sudo` es necesario si no tenemos permisos de root.

Un "punto de montaje" es, en el contexto de Linux, una dirección en la que se acopla el dispositivo o partición, haciéndolo accesible en el sistema de archivos.

14. **¿Qué comando podrías utilizar para montar un CD en el punto de montaje */mnt/carpeta*?¿Y para desmontarlo de dicho punto de montaje?**
    - **Para montarlo en"/mnt/carpeta": `(sudo) mount /dev/sr0 /mnt/carpeta`**
    - **Para desmontarlo: `(sudo) umount /mnt/carpeta`**

`Sudo` es necesario si no tenemos permisos de root.

15. **¿En qué directorio de Linux encontramos accesibles los dispositivos?**

En /dev/

16. **El comando chmod permite modificar los permisos de lectura, escritura y ejecución sobre archivos y directorios. Se muestra un ejemplo en la tabla para cada tipo de notación:**

Ejemplo	**Permisos de usuario: rwx**	**Permisos de grupo: r-x**	**Permisos del resto de usuarios: r--**	**Comando para establecer permisos de *archivo.txt***
Octal	**111 = 7**	**101=5**	**100=4**	`chmod 754 archivo.txt`
Simbólica	**rwx**	**rx**	**r**	`chmod u=rwx,g=rx,o=r archivo.txt`

Si, en lugar de establecer los permisos, queremos modificarlos, podemos usar + y - para cada uno de ellos. Para el archivo de ejemplo, supongamos que queremos eliminar el permiso de ejecución del grupo, y añadir el de escritura al grupo y al resto de usuarios:
`chmod g+w-x,o+w archivo.txt`

Escribe los comandos, tanto en notación octal como simbólica, que realicen las siguientes acciones:

- **Establecer los permisos de archivo.txt con las siguientes condiciones:**
  - **usuario: lectura, escritura y ejecución.**
  - **grupo: lectura y escritura.**
  - **resto: lectura.**
- **Modificar los permisos anteriores, quedando como sigue:**
  - **usuario: lectura y escritura.**
  - **grupo: lectura.**
  - **resto: ejecución.**

[Puedes hacer este ejercicio en la máquina virtual Ubuntu que has creado para la actividad, por ejemplo generando un archivo de texto a partir de un comando, como se hizo en el ejercicio 5. Para comprobar los permisos del archivo con `ls -l archivo.txt`]

Simbólica:

1. **chmod u=rwx,g=rw,o=r archivo.txt**
2. **chmod u-x,g-w,o+x-r archivo.txt**

Octal:

1. **chmod 764 archivo.txt**
2. **chmod 641 archivo.txt**

17. **Las contraseñas cifradas de los usuarios, ¿En qué fichero y cómo se guardan?¿Qué representa cada fila de /etc/passwd y /etc/group?**
Las contraseñas se guardan, encriptadas, en /etc/shadow.
Cada fila de /etc/passwd representa un usuario del sistema.
Cada fila de /etc/group representa un grupo de usuarios del sistema.

18. **¿Qué permisos son necesarios para poder copiar un archivo dentro de la misma carpeta?**
Permiso de lectura en archivo y permiso de escritura en carpeta.

19. **¿Qué usuarios pueden cambiar los permisos de un archivo?**
    Root y el propietario.

20. **¿Qué comando hay que ejecutar para ver el tamaño del directorio $HOME de root?¿Y para ver el de tu usuario?** [Puedes probar los comandos en la máquina ubuntu de la actividad]
    sudo du -sh /root
    du -sh /home/nombreusuario (siendo nombreusuario tu usuario)

21. **¿Qué hace `fdisk -l /dev/sda`? ¿Y `fdisk -l`?**
    `fdisk -l /dev/sda` muestra información de las particiones del primer disco duro.
    `fdisk -l` muestra información de todas las particiones en todos los discos.

22. **En la máquina ubuntu de la actividad, con un editor de texto, crea el siguiente script:**
    #!/bin/bash
    echo "Hello $USER!"
    Guárdalo como holita.sh en una carpeta de tu *home*.
    Intenta ejecutarlo con: `./holita.sh`
    ¿Qué es necesario para que puedas ejecutarlo?
    Que el usuario tenga permisos de ejecución sobre el script. Podemos dárselos con chmod
    `u=rwx holita.sh`

23. **¿Qué es el programa cron? ¿De qué partes consta?**
    Es un administrador que permite ejecutar tareas programadas o periódicas en Linux.
    Se compone de 2 elementos: cron daemon, que está corriendo en segundo plano todo el tiempo, y el archivo de configuración /etc/crontab, que contiene las tareas que se van a ejecutar.

24. **La línea de crontab: `59 23 * * * poweroff`, ¿Qué hace?**
    (Ten en cuenta el formato: `m h dom mon dow mon command`)
    Apagar el sistema todos los días a las 23:59 horas.

25. **La línea de crontab: `00 10 1 */3 *` /home/user/script.sh, ¿Qué hace?**
    (Ten en cuenta el formato: `m h dom mon dow mon command`)
    Ejecuta script.sh, cada tres meses, el día 1, a las 10:00.