

## EJERCICIO DE CLASE

Para este ejercicio, ejecutarás las acciones que aquí se muestran y entregarás las capturas de pantalla de tu ambiente Linux, similares a las que se muestran en el documento. Todo se entregará en un PDF con tu nombre, empezando por apellido paterno.

Vamos a probar los IDs asociados a un usuario. Para reproducir las acciones de este ejercicio vamos a entrar a nuestro sistema Linux en 2 terminales. En una terminal se requiere que estés *logueado* como usuario root y en la otra terminal que estés *logueado* como usuario “no-root”. En este ejemplo, el usuario “no-root”, “no-admin” o “sin privilegios” es el usuario *croman* (imagen 1):

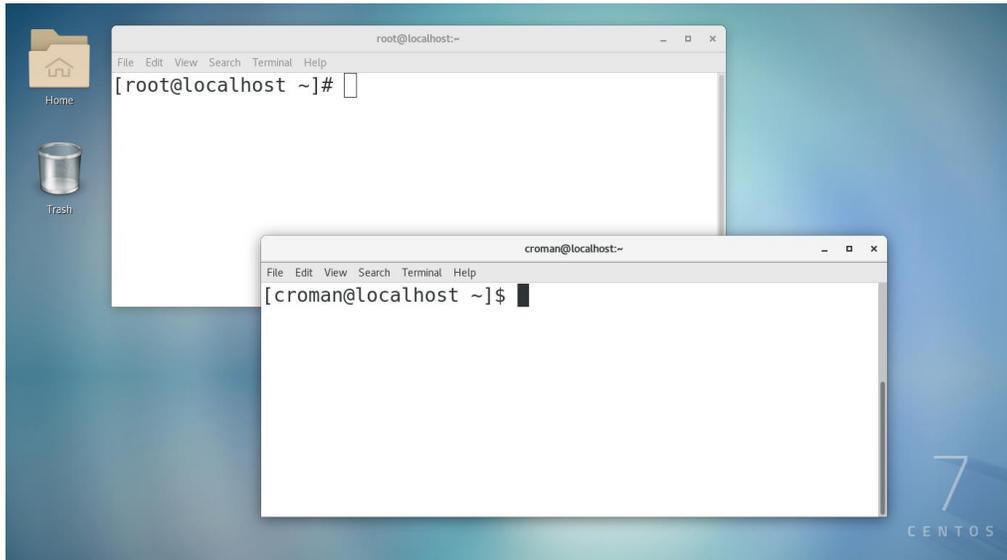


Imagen 1: Abrir 2 terminales, en una como usuario root y en la otra como usuario “no-root”

**En ocasiones** el nombre de usuario (o username) no se muestra en el prompt de la terminal, entonces ¿Cómo saber qué usuario soy? Una manera de comprobarlo consiste en ejecutar el comando “whoami” en ambas terminales (imagen 2):

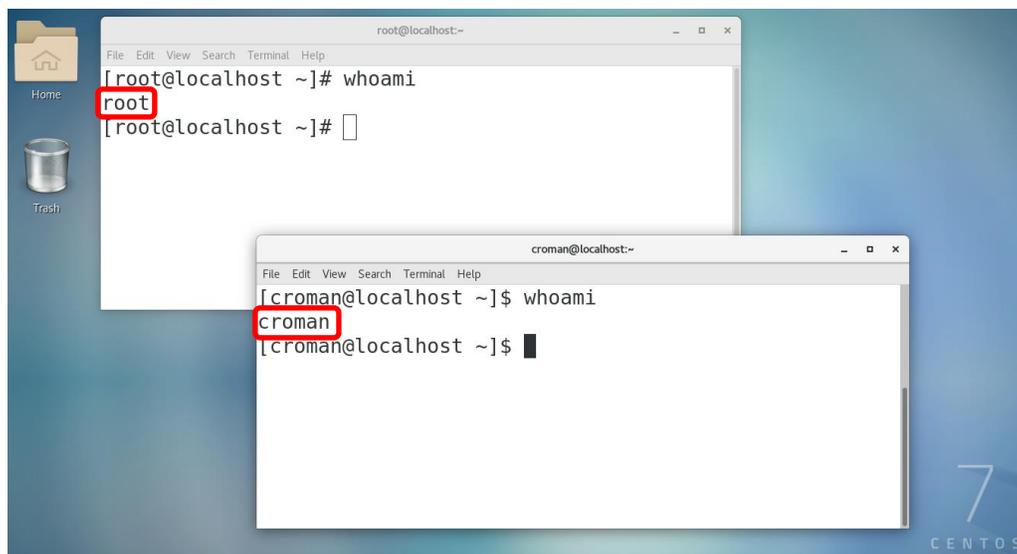


Imagen 2: Ejecución del comando whoami

El comando “whoami” proviene de la pregunta en inglés “Who am I?” que se traduciría como “¿Quién soy yo?” en español.

Vamos a probar un comando que requiere privilegios de root. Un ejemplo es el comando **fdisk**.  
**Investiga qué hace el comando fdisk:**

---

**Investiga la opción -l (guión y la letra “le”) del comando fdisk:**

---

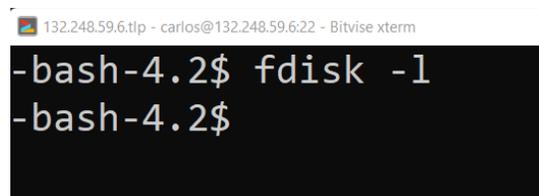
Ejecuta el comando `fdisk -l` con el usuario “no-root” para que te muestre el mensaje de error “Permiso denegado”, como se observa en la imagen 3:



```
croman@localhost:~/Documentos
Archivo Editar Ver Buscar Terminal Ayuda
[croman@localhost Documentos]$ fdisk -l
fdisk: no se puede abrir /dev/sda: Permiso denegado
[croman@localhost Documentos]$
```

Imagen 3: Ejecución del comando fdisk

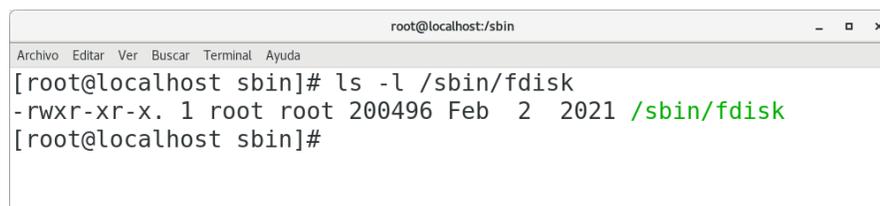
**IMPORTANTE:** Dependiendo de la distribución de Linux, puede mostrarse un mensaje de error o no. Por ejemplo, en la siguiente imagen, este usuario “no-root” en Fedora, no muestra mensaje de error, simplemente regresa el prompt sin mensaje, como se observa en la imagen 4:



```
132.248.59.6.tlp - carlos@132.248.59.6:22 - Bitvise xterm
-bash-4.2$ fdisk -l
-bash-4.2$
```

Imagen 4: Ejecución del comando fdisk

Ahora nos cambiamos a la terminal del usuario root y revisamos los permisos actuales del comando `fdisk`, como se observa en la imagen 5:



```
root@localhost:/sbin
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost sbin]# ls -l /sbin/fdisk
-rwxr-xr-x. 1 root root 200496 Feb  2 2021 /sbin/fdisk
[root@localhost sbin]#
```

Imagen 5: Permisos originales del comando fdisk

Ahora modificamos los permisos del comando `fdisk` modificando el bit “setuid”, como se observa en la imagen 6:

```
root@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@localhost ~]# chmod 4511 /sbin/fdisk  
[root@localhost ~]# █
```

Imagen 6: Ejecución del comando chmod para modificar el bit "setuid"

Y corroboramos que se cambiaron los permisos, como se observa en la imagen 7:

```
root@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@localhost ~]# ls -l /sbin/fdisk  
-r-s-x-x. 1 root root 200496 Feb  2 2021 /sbin/fdisk  
[root@localhost ~]# █
```

Imagen 7: Comprobación de la modificación del bit "setuid"

Regresamos a la terminal del usuario "no-root" y ejecutamos nuevamente el comando que hace un momento nos indicó "Permiso denegado" (Imagen 8):

```
croman@localhost:~/Documentos  
Archivo Editar Ver Buscar Terminal Ayuda  
[croman@localhost Documentos]$ fdisk -l  
  
Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 sectors  
Units = sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk label type: dos  
Identificador del disco: 0x0006b393  
  
Disposit. Inicio Comienzo Fin Bloques Id Sistema  
/dev/sda1 * 2048 616447 307200 83 Linux  
/dev/sda2 616448 4810751 2097152 82 Linux swap / Solaris  
/dev/sda3 4810752 41943039 18566144 83 Linux  
[croman@localhost Documentos]$ █
```

Imagen 8: Comprobación de la modificación del bit "setuid"

Ahora regresamos los permisos originales al fdisk, como se observa en las imágenes 9 y 10:

```
root@localhost:/sbin  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@localhost sbin]# chmod 755 /sbin/fdisk  
[root@localhost sbin]# █
```

Imagen 9: Regresando los permisos originales al comando fdisk

```
root@localhost:/sbin
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost sbin]# ls -l /sbin/fdisk
-rwxr-xr-x. 1 root root 200496 Feb  2 2021 /sbin/fdisk
[root@localhost sbin]#
```

Imagen 10: Comprobando los permisos originales del comando fdisk

## ACTIVIDAD# 2

Ahora escribe el programa 2 llamado ids.c

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main(void)
{
    printf("Real user ID: %d\n", getuid());
    printf("Effective user ID: %d\n", geteuid());
    printf("Real group ID: %d\n", getgid());
    printf("Effective group ID: %d\n", getegid());
    return 0;
}
```

Si lo compilas y ejecutas obtendrás que el usuario real y el usuario efectivo dan el mismo valor. ¿Porqué? Porque lo estás ejecutando con el mismo usuario que es el dueño del ejecutable.

Ahora haz que el programa sea de un usuario y trata de ejecutarlo como otro usuario.

Modifica el bit "setuid" para que veas diferencia entre el usuario real y el usuario efectivo.

```
croman@localhost:~
Archivo Editar Ver Buscar Terminal Ayuda
[croman@localhost ~]$ ./ids
Real user ID: 1000
Effective user ID: 0
Real group ID: 1000
Effective group ID: 1000
[croman@localhost ~]$
```