0.

Objetivos del aprendizaje

- Describir los elementos fundamentales de organización del sistema operativo GNU/-Linux.
- Identificar la importancia de los ficheros en GNU/Linux ("*si algo no es un fichero, entonces es un proceso*").
- Describir en líneas generales como se organiza el sistema de ficheros y qué son los nodos-i.
- Definir qué papel tienen el usuario y el grupo propietario de un fichero y cómo pueden modificarse.
- Interpretar una cadena de permisos de GNU/Linux.
- Cambiar los permisos de un fichero, tanto en modo simbólico como en modo absoluto.
- Explicar el objetivo de los permisos especiales (*sticky bit*, `suid` y `sgid`).
- Utilizar la máscara `umask` para cambiar los permisos por defecto con los que se crean los archivos y directorios.
- Explicar cómo se aplican los *bits* de permisos a la hora de decidir si un usuario puede realizar una determinada acción sobre un fichero.
- Distinguir todos los tipos de ficheros que se pueden utilizar en GNU/Linux y diferenciar claramente entre enlaces simbólicos y enlace físicos.
- Definir el rol de los procesos en GNU/Linux, cuáles son sus atributos y qué tipos de procesos existen.
- Explicar cómo se represan los dispositivos en GNU/Linux.
- Enumerar las carpetas fundamentales de la estructura genérica del sistema de ficheros en GNU/Linux.
- Diferenciar entre contenidos estáticos y dinámicos y contenidos compartibles y no compartibles.

Contenidos

2.1. Ficheros.

  2.1.1. Sistema de ficheros.

  2.1.2. Propietarios y permisos.

    2.1.2.1. Usuario y grupo propietario.

    2.1.2.2. Permisos de lectura, escritura y ejecución para ficheros y directorios.

    2.1.2.3. Permisos especiales (*sticky bit*, `suid` y `sgid`).

Evaluación

- Cuestionarios objetivos.
- Pruebas de respuesta libre.
- Tareas de administración.

1. Ficheros

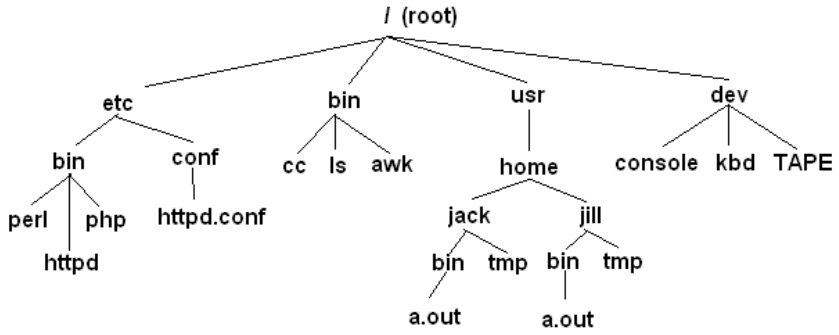
1.1. Sistema de ficheros

Ficheros

- En GNU/Linux, **todo** son ficheros ("*si algo no es un fichero, entonces es un proceso*"):
  - Los programas u órdenes son ficheros: `/bin/ls`, `/usr/bin/find`...
  - Los dispositivos I/O son ficheros: `/dev/sda`, `/dev/fd0`, `/dev/tty0`...
  - Comunicación entre procesos: *sockets* o tuberías (*pipes*).
  - Directorios, ficheros de datos, ficheros de configuración...
  - El propio núcleo del sistema operativo (*kernel*).
- GNU/Linux tiene una estructura jerárquica de directorios, conocida como sistema de archivos:
  - / $\rightarrow$ directorio raíz.
  - Puede estar compuesto por varias particiones pertenecientes a varios dispositivos (discos duros, CDs, DVDs...).
  - Todos disponibles desde la jerarquía de directorios.

Sistema de ficheros

- Sistema de ficheros:
  - Guarda los ficheros del sistema.
  - Se organiza de manera jerárquica, en directorios.
  - No hay unidades.



Sistema de ficheros: nodos-i

- Aunque a nivel lógico, el sistema de ficheros parece un árbol, en realidad los ficheros se almacenan desorganizados por el disco duro.
- Un fichero puede tener sectores a lo largo de toda la superficie.
- Los nodos-i son metadatos sobre los ficheros que nos proporcionan información sobre aspectos como su tamaño, sus permisos, la posición de sus sectores, número de enlaces... *¿nombre?*
- Cada fichero tiene un nodo-i.
- Todos están localizados en un área del disco duro, que está limitada (nº máximo de nodos-i).

1.2. Propietarios y permisos

Gestión del acceso: propietarios y permisos

- El acceso a los ficheros se gestiona de la siguiente forma:
  - Propietarios:
    - Cada fichero tiene dos propietarios: usuario y grupo.
    - `chown` → cambia el usuario propietario (se necesitan privilegios de **root**).

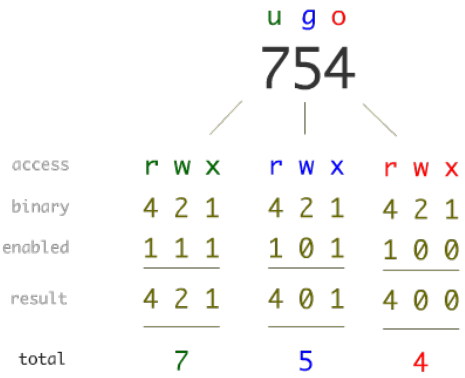
```
chown pagutierrez fichero
chown pagutierrez.profesores fichero
chown -R pagutierrez directorio
```

    - `chgrp` → cambia el grupo propietario (puede hacerlo el propietario del fichero, el que pertenezca al grupo, o **root**).

```
chgrp profesores fichero
chgrp -R profesores directorio
```

Gestión del acceso: propietarios y permisos

```
pedroa@pagutierrezLaptop:~/tmp$ ls -la
total 36
drwxrwxr-x  4 pedroa pedroa  4096 feb 17 17:52 .
drwxr-xr-x 84 pedroa pedroa 20480 feb 17 18:14 ..
-rw-r--r-- 1 pedroa pedroa     0 feb 17 14:22 fichero1
-rw-r--r-- 1 pedroa pedroa     0 feb 17 14:21 fichero2
```



Gestión del acceso: propietarios y permisos

- El acceso a los ficheros se gestiona de la siguiente forma:

Acceso	Fichero	Directorio
`r`	Ver el contenido	Listar el contenido
`w`	Modificar el contenido	Crear/eliminar ficheros
`x`	Ejecutar el fichero	Entrar en el directorio

- Se establecen independientemente para: el usuario propietario (`u`), usuarios del grupo propietario (`g`) y resto de usuarios (`o`).

```
chmod u+r fichero                # Modo simbólico
chmod -R u+rwx,go-rwx directorio # Modo simbólico
chmod 740 fichero                # Modo absoluto u+rwx,g+r,g-wx,o-rwx
```

- Otros comandos: `adduser`, `addgroup`...

Gestión del acceso: propietarios y permisos

- Ejercicio: Establezca permisos de escritura para el fichero `ejemplo` sólo para el usuario propietario, de lectura para todos y de ejecución para el usuario y grupo propietarios.

Gestión del acceso: propietarios y permisos

Actual	chmod	Resultado	Descripción
`rw-------`	`a+x`	`rwx--x--x`	Agregar a todos (all) permiso de escritura.
`rwx--x--x`	`go-x`	`rwx------`	Se elimina permiso de ejecución para grupo y otros.
`rwxr-xr-x`	`u-x,go-r`	`rw---x--x`	Al usuario se le quita ejecución, al grupo y a otros se les quita lectura.
`rwxrwxrwx`	`u-x,go-rwx`	`rw-------`	Al usuario se le elimina ejecución, al grupo y a otros se les eliminan todos los permisos.
`r--------`	`a+r,u+w`	`rw-r--r--`	A todos se les agrega lectura, al usuario se le agrega escritura.
`rw-r-----`	`u-rw,g+w,o+x`	`---rw---x`	Al usuario se le eliminan lectura y escritura, al grupo se le agrega escritura y a otros se les agrega ejecución.

Gestión del acceso: propietarios y permisos

- Permisos especiales:
  - `t`: *sticky bit*, `chmod o+t fichero`.
    - El comando `ls` lo representa como una `t` en el noveno bit (según mayúscula/minúscula, `t`→ `o+x`, `T`→ `o-x`).
    - Para ejecutables → mantener la imagen del fichero en la memoria de intercambio después de finalizar la ejecución del mismo (*en desuso*).
    - Para directorios → solo `root` o el propietario de un fichero (o de la carpeta) pueden borrar o renombrar el fichero, aunque tengan permiso de escritura en la carpeta.

```
1 pagutierrez@TOSHIBA:~$ ls -la fichero
2 -rw-r--r-- 1 pagutierrez pagutierrez 0 2017-02-07 13:31 fichero
3 pagutierrez@TOSHIBA:~$ chmod o+w+t fichero
4 pagutierrez@TOSHIBA:~$ ls -la fichero
5 -rw-r--rwT 1 pagutierrez pagutierrez 0 2017-02-07 13:31 fichero
6 pagutierrez@TOSHIBA:~$ ls -la /tmp
7 drwxrwxrwt 17 root        root          4096 2017-02-07 13:27 .
8 ...
```

Gestión del acceso: propietarios y permisos

- Permisos especiales:
  - `s`: para usuarios, `suid`, `chmod u+s fichero`.
    - El comando `ls` lo representa como una `s` en el tercer bit (según mayúscula/minúscula, `s`→ `u+x`, `S`→ `u-x`).
    - Para ejecutables → cambio de dominio a nivel de usuario. Durante la ejecución, el usuario efectivo del proceso es el propietario del fichero y no el usuario que lo ejecutó.
    - Para directorios → Ignorado.

```
* El ejecutable "gestorbd" lee el fichero "basedatos":
-rwxr--r-x root root /opt/bin/gestorbd
-rwx------ root root /opt/datos/basedatos
* El usuario pagutierrez puede ejecutar "gestorbd", pero ese programa NO podrá leer "
    basedatos"
* El programa si podrá leer "basedatos" si "gestorbd" tiene los permisos:
-rwsr--r-x root root /opt/bin/gestorbd
```

Gestión del acceso: propietarios y permisos

- Permisos especiales:
  - `s`: para grupos, `sgid`, `chmod g+s fichero`.
    - El comando `ls` lo representa como una `s` en el sexto bit (según mayúscula/-minúscula, `s`→ `g+x`, `S`→ `g-x`).
    - Para ejecutables → cambio de dominio a nivel de grupo. Durante la ejecución, el grupo efectivo del proceso es el grupo propietario del fichero y no el del usuario que lo ejecutó.
    - Para directorios → al crear un fichero en su interior, el grupo propietario del nuevo fichero es el grupo del directorio y no del usuario que ejecuta la orden.

Gestión del acceso: propietarios y permisos

```
* El ejecutable "gestorbd" lee el fichero "basedatos":
-rwxr-xr-x root root /opt/bin/gestorbd
-rwxr----- root root /opt/datos/basedatos
* Grupo "alumnos": pueden ejecutar "gestorbd" pero NO leer "basedatos"
* El programa si podrá leer "basedatos" si "gestorbd" tiene los permisos:
-rwxr-Sr-x root root /opt/bin/gestorbd
=====================
* El usuario "pagutierrez" sólo pertenece al grupo "profesores"
* Se tiene el directorio "drwxr-sr-x pagutierrez alumnos /practicas"
* Si "pagutierrez" ejecuta "cp tema2.pdf /practicas" entonces el fichero copiado pertenecerá
    al grupo "alumnos":
-rw-r--r-- pagutierrez alumnos /practicas/tema2.pdf
```

Máscara de permisos (`umask`)

- Cuando un fichero nuevo se crea, se le asignan permisos.
- Los permisos se deciden aplicando una máscara de permisos a los permisos base (que se puede consultar/modificar utilizando el comando `umask`):

```
pedroa@pagutierrezLaptop:~$ umask
0022
```

- La máscara de bits indica con un 1 aquellos bits que deberán ser 0 en la cadena de permisos, es decir, indica qué permisos está restringidos.
- Los permisos base para directorios son 777; para ficheros, 666.
- *¿Podremos especificar una máscara que permita crear ficheros con permisos de ejecución?*

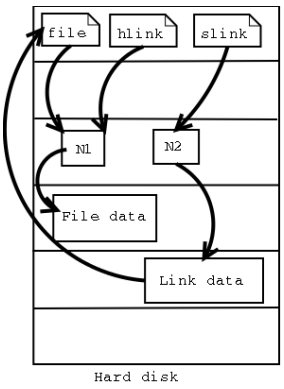
1.3. Tipos de ficheros

Tipos de ficheros (`ls -l`)

- Normal.
- Directorio (`d`): son ficheros que contienen enlaces a otros ficheros (ya sean directorios o archivos normales).
- Especial de bloque (`b`): fichero especial para interactuar con un dispositivo basado en bloques.
- Especial de carácter (`c`): fichero especial para interactuar con un dispositivo basado en caracteres.
- *Named Pipes* (`p`): tubería FIFO con nombre (comunicación de procesos de diferentes usuarios con tuberías).
- *Socket* (`s`): como los *pipes* pero con comunicación *duplex* (ambos sentidos, ej. `/tmp/.X11-unix/X0`).
- Enlace físico.
- Enlace simbólico (`l`).

Tipos de ficheros: enlaces

- Enlaces
  - Archivos especiales que permiten que varios nombres (enlaces) se asocien a un único e idéntico archivo.
  - Varias instancias de un mismo archivo en diversos lugares de la estructura jerárquica sin necesidad de copiarlos.
  - Ayuda a asegurar la coherencia y ahorrar espacio en el disco.
  - Grupo de personas trabajando sobre un mismo fichero (modificaciones centralizadas).



Tipos de ficheros: enlaces

- Enlaces
  - *Enlaces físicos* (`ln archivo-real enlace-físico`):
    - Representan un nombre alternativo para un archivo (dos nombres de fichero apuntando al mismo nodo-i).
    - Si eliminamos un enlace físico, no eliminamos el archivo original. Mientras quede *al menos un* enlace físico, el archivo no se elimina.
    - Sólo es posible entre ficheros que estén en la misma partición.
    - No se pueden realizar a directorios.

Tipos de ficheros: enlaces

- Enlaces
  - *Enlaces simbólicos* (`ln -s archivo-real enlace-simb`):
    - Es un puntero virtual al archivo real.
    - Fichero de texto (con su nodo-i independiente) que contiene la ruta del archivo al que apunta.
    - Si se elimina el enlace simbólico, no se elimina el fichero original.
    - Observad conteo de referencias (tercera columna, después de permisos).

```
1 pagutierrez@TOSHIBA:~$ ln prueba enlace.fisico
2 pagutierrez@TOSHIBA:~$ ls -li prueba enlace.fisico
3 6292999 -rw-r--r-- 2 pagutierrez pagutierrez 6 2017-02-07 19:28 enlace.fisico
4 6292999 -rw-r--r-- 2 pagutierrez pagutierrez 6 2017-02-07 19:28 prueba
5 pagutierrez@TOSHIBA:~$ ln -s prueba enlace.simbolico
6 pagutierrez@TOSHIBA:~$ ls -li prueba enlace.simbolico
7 6292993 lrwxrwxrwx 1 pagutierrez pagutierrez 6 2017-02-07 19:29 enlace.simbolico -> prueba
8 6292999 -rw-r--r-- 2 pagutierrez pagutierrez 6 2017-02-07 19:28 prueba
```

2. Procesos

Procesos

- Procesos: son programas en ejecución.
- Los atributos de un proceso son:
  - PID ⇒ identificador del proceso.
  - PPID ⇒ identificador del proceso padre.
  - Nice number ⇒ prioridad asignada al ejecutarlo.
  - TTY ⇒ terminal en el que se está ejecutando.
  - RUID ⇒ identificador del usuario real, el que lo ejecutó.
  - EUID ⇒ identificador del usuario efectivo, si hay cambio de dominio se refleja aquí (permiso **suid**).
  - RGID ⇒ identificador del grupo real, el grupo del usuario que lo ejecutó.
  - EGID ⇒ identificador del grupo efectivo, si hay cambio de dominio se refleja aquí (permiso **sgid**).

Procesos

- Atributos de un proceso: `ps -Fl PID`
- Tipos de procesos:
  - Interactivos: hay alguien conectado al sistema que los inicia (primer o segundo plano `&`).
  - Encolados: procesos que se mandan a un *buffer* para ser ejecutados (en una fecha concreta o cuando la carga del sistema sea baja).
  - Demonios: programas ejecutados en segundo plano durante el arranque, que esperan de forma continua un determinado evento.

3. Dispositivos

Dispositivos

- Los dispositivos se representan/manejan como ficheros:
  - Ficheros especiales de caracteres: representan a dispositivos de caracteres (cinta magnética, puerto paralelo, puerto serie...)
  - Ficheros especiales de bloques: representan a dispositivos de bloques (disquete, partición de un disco duro o un pendrive...)
  - Escribir/leer en un dispositivo se convierte en escribir/leer en el fichero correspondiente.
- Esos ficheros se almacenan en el directorio `/dev`:
  - `/dev/fd0` ⇒ disquete de la primera disquetera.
  - `/dev/sda` ⇒ primer disco duro (sin considerar particiones).
  - `/dev/sda1` ⇒ primera partición del primer disco.
  - `/dev/sdb` ⇒ segundo disco duro.
  - `/dev/sdc` ⇒ disco USB (primer nombre de dipositivo libre).
  - `/dev/tty1` ⇒ primera terminal de consola (`tty2` segunda).
  - `/dev/lp0` ⇒ primer puerto paralelo.

4. Estructura genérica del sistema de ficheros

Estructura genérica del sistema de ficheros

- *Filesystem Hierarchy Standard*: Jerarquía Estándar del Sistema de Ficheros.
- Especificación estándar para sistemas tipo Unix.
- Fruto del consenso entre la comunidad (desarrolladores, administradores...).

- Versión 3.0 (2015), especificación disponible en la URL: `http://wiki.linuxfoundation.org/en/FHS`
- En Linux, disponible como página de manual:

```
1 man hier
```

Estructura genérica del sistema de ficheros

- Existen dos tipos de distinciones cuando hablamos del tipo de contenido de un directorio: estáticos/dinámicos y compartibles/no compartibles.
  - Estáticos: Contiene binarios, bibliotecas, documentación y otros ficheros que no cambian sin intervención del administrador. Pueden estar en dispositivos de solo lectura (*read-only*) y no necesitan que se hagan copias de seguridad tan a menudo como los ficheros dinámicos.
  - Dinámicos: Contiene ficheros que no son estáticos. Deben de encontrase en dispositivos de lectura-escritura (*read-write*). Necesitan que se hagan copias de seguridad a menudo.
  - Compartibles: Contiene ficheros que se pueden encontrar en un ordenador y utilizarse en otro.
  - No compartibles: Contiene ficheros que no podemos utilizar en distintas máquinas.

Estructura genérica del sistema de ficheros

- `/bin` ⇒ ficheros ejecutables básicos compartidos (`mv`, `cp`).
- `/dev` ⇒ ficheros especiales de dispositivos.
- `/etc` ⇒ la mayoría de los ficheros de configuración locales del sistema (solo archivos de texto).
- `/root` ⇒ directorio HOME del administrador.
- `/sbin` ⇒ ficheros ejecutables que, normalmente, sólo el administrador puede ejecutar.
- `/home` ⇒ los directorios de trabajo de los usuarios.
- `/lost+found` ⇒ contiene "referencias" a los ficheros marcados como erróneos al chequear el sistema de ficheros.
- `/lib` ⇒ librerías necesarias para ejecutar los archivos.

Estructura genérica del sistema de ficheros

- `/proc` y `/sys` ⇒ sistemas de ficheros virtuales, contienen información sobre procesos, núcleo, módulos cargados, dispositivos, sucesos...
- `/tmp` ⇒ ficheros temporales. Tiene el permiso `t` activo.
- `/var` ⇒ ficheros variables: colas de datos (`spool`) de impresión, `e-mail`..., ficheros del `cron`, `atd`, ficheros de `log`...
- `/boot` ⇒ núcleo y ficheros necesarios para cargar el núcleo y ficheros de configuración del gestor de arranque.
- `/mnt`, `/mount` ó `/media` ⇒ montaje de otros sistemas de ficheros: disquetes, cdroms...
  - `/mnt/floppy` ó `/media/floppy`
  - `/mnt/cdrom` ó `/media/cdrom`
- `/opt`: paquetes de aplicaciones estáticas (no actualizables).

Estructura genérica del sistema de ficheros

- `/usr` ⇒ contiene subdirectorios de solo lectura, que no deben ser específicos de la máquina que los usa (*Unix system resources*):
  - `/usr/bin` ⇒ ficheros ejecutables por todos los usuarios.
  - `/usr/sbin` ⇒ ficheros ejecutables de administración.
  - `/usr/include` ⇒ ficheros cabecera de cabecera estándar para compilación.
  - `/usr/lib` ⇒ librerías binarias.
  - `/usr/local` ⇒ *software* local específico.
  - `/usr/share` ⇒ datos compartidos (independientes de la arquitectura: imágenes, ficheros de texto...).
    - `/usr/share/man`.
    - `/usr/share/doc`.
  - `/usr/src` ⇒ código fuente, como el del kernel...

Estructura genérica del sistema de ficheros

- Estáticos: `/bin`, `/sbin`, `/opt`, `/boot`, `/usr/bin`...
- Dinámicos: `/var/mail`, `/var/spool`, `/var/run`, `/var/lock`, `/home`
- Compartibles: `/usr/bin`, `/opt`...
- No compartibles: `/etc`, `/boot`, `/var/run`, `/var/lock`...

5. Referencias

Referencias

Referencias

[Nemeth et al., 2010] Evi Nemeth, Garth Snyder, Trent R. Hein y Ben Whaley Unix and Linux system administration handbook.
Capítulo 6. *The filesystem*. Prentice Hall. Cuarta edición. 2010.

[Frisch, 2002] Aeleen Frisch. Essential system administration.
Capítulo 2. *The Unix Way*. O'Reilly and Associates. Tercera edición. 2002.

Algunos ejercicios

⋆ Crear un archivo con el contenido `HOLA!`, utilizando la orden `echo`, y asignarle permisos para que solo puede ser consultado por su propietario y por los miembros del grupo.
⋆ Marque todos los tipos de enlace que pueden establecerse en los siguientes casos (físico, simbólico o ambos).

- enlace hacia un archivo en el mismo directorio.
- enlace hacia un archivo en otro sistema de archivos.
- enlace hacia un directorio en el mismo sistema de archivos.
- enlace hacia un directorio en otro sistema de archivos.

Algunos ejercicios (permisos)

- Transformar los siguientes permisos simbólicos en absolutos:

```
1  rwxr-xr-x    r-xr--r--    rw-r-----    r-x--x--x
```

- Transformar los siguientes permisos absolutos en simbólicos:

```
1  644    755    610    631
```

- Fijar, en modo simbólico, los permisos de `arch1` en modo `754`.
- Fijar, en modo absoluto, los permisos de `arch1` en modo `rwxr-x--x`.
- Fijar los permisos del directorio `dir1` de modo que todos lo puedan leer y entrar, pero sólo el dueño pueda modificar sus archivos: 1) en modo simbólico; 2) en modo absoluto.
- Modificar para que el grupo también pueda modificar archivos.
- Fijar en modo simbólico los permisos del archivo ejecutable `exec.tar` para que sea ejecutable por todos, legible por el dueño y el grupo, y modificable solo por el dueño. Repetir en modo absoluto.

- Fijar en modo absoluto los permisos del directorio `dir1` de modo que sólo el dueño y el grupo lo puedan recorrer y leer, y sólo el dueño pueda grabar y borrar en él. Repetir en modo simbólico.

⋆: Sea la siguiente salida del comando `ls -l`:

```
-rwSr-xr-x 1 pagutierrez docentes 29024 ene 1  16:29  controlar
-rw-rw-r-- 1 pagutierrez docentes 2300  may 18 09:37  borrador.txt
-rw-r--r-- 1 pagutierrez docentes 5895  may 15 12:08  index.htm
-rwxr-xr-x 1 pagutierrez docentes 29024 ene 1  16:29  revisar
-rwxr--r-- 1 pagutierrez docentes 29024 ene 1  16:29  mostrar
drwxrwxrwt 2 pagutierrez tecnicos 1024  may 1  17:23  trabajos
drwxr-xr-x 2 pagutierrez tecnicos 1024  oct 16  1998  netscape3
drwxrwx--x 2 pagutierrez tecnicos 1024  may 11  7:29  finanzas
drwxrwxr-x 2 pagutierrez tecnicos 1024  jul 7   6:54  redes
drwxr-xr-x 2 jsanchezm   docentes 1024  jun 17 19:35  corporacion
```

El usuario `jsanchezm`, del grupo `docentes`, tiene acceso al presente directorio. Indicar, si los hay:

1. Archivos de los que puede mostrar contenido.
2. Archivos que puede ejecutar como programa.
3. Archivos en los que puede modificar contenido.
4. Subdirectorios en los que puede ingresar.
5. Subdirectorios en que puede crear/eliminar ficheros propios.
6. Subdirectorios en los que puede borrar archivos de otros.
7. Subdirectorios en los que puede entrar y ejecutar programas contenidos en ellos, pero no ver nombres de archivos.
8. Archivos que puede ejecutar como programa con permisos del usuario `pagutierrez`.

⋆: Sea la siguiente salida del comando `ls -l` (suponemos que `pagutierrez` pertenece a `staff`):

```
-rw-r--r--  1  root        root       33280 jun 12 19:40 Carta.doc
drwxrwxrwx 5  pagutierrez staff       1024 dic  4  1999 step
drwxrwxr-x 22 pagutierrez staff       1024 nov 20  1999 Office51
drwxr-x--- 6  pagutierrez staff       1024 may 7  16:43 argos
drwxrwxr-- 21 pagutierrez staff       1024 jul 11 18:22 bajados
-rw-rw---- 3  root        root         542 jul 13 11:26 boor.exe
drwxrwxrwt 3  pagutierrez staff       1024 may 25 10:02 borrador
-rwSrwSr-x 1  root        root        9218 jun 12 19:41 pph3
drwxrwx--x 2  pagutierrez pagutierrez 1024 may 7  16:47 cdir
-rw-rw-r-- 3  root        root         542 jul 13 11:26 mysql-doc
-rw-r-xr-- 3  pagutierrez staff       1084 ago 1  10:01 ver.exe
-rwxr-xr-x 3  pagutierrez staff       1084 ago 1  10:01 ver
drwxr-xr-x 7  pagutierrez pagutierrez 1024 jul 25 11:48 lit
```

Indicar, si los hay:

1. directorios públicos (todo el mundo puede entrar, listar y borrar archivos);
2. archivos que tienen enlaces hard o físicos;
3. archivos ejecutables por el usuario `pagutierrez`;
4. directorios navegables por todo el mundo;
5. directorios donde miembros del grupo `staff` puede borrar archivos;
6. archivos que son enlaces simbólicos.
7. archivos ejecutables que adquieren permisos de usuario `root`.