Permisos e listas de control de acceso: ACLs, Eiciel

Nesta sección veremos como superar as limitacións do tradicional sistema de permisos de UNIX. Veranse as Listas de Control de Acceso (ACLs), que permiten indicar que conxunto de grupos e/ou usuarios poden acceder a un arquivo e/ou carpeta , sen estar limitados a Usuario Propietario, Grupo Propietario e Outros.

As ACLs aplicadas sobre un sistema arquivos exportado por NFS ou CIFS tamén son efectivas nos clientes.

Sumario

Introdución ás ACLs

Como xa vimos, o sistema tradicional de permisos de GNU/Linux só permite afinar os permisos dunha carpeta ou arquivo para o usuario e o grupo propietario dos mesmos. Para os demais queda un grupo chamado *outros* que non permite facer ningún axuste fino.

Permisos básicos

Existen tres permisos independentes, chamados **permisos básicos**, que poden ser permitidos (estado 1) ou denegados (estado 0) a un arquivo e/ou directorio.

- **r - lectura**
- **w - escritura**
- **x - execución**

O significado destes tres permisos resúmese na seguinte táboa:

Permiso	Arquivo	Directorio
Lectura	Ver o contido do arquivo.	Ver o nome dos arquivos dentro do directorio (pero sen poder saber nada máis sobre eles como: o tipo de arquivo, tamaño, propietario, permisos, etc.)
Escritura	Modificar ou eliminar el arquivo.	Agregar, eliminar e renomear arquivos do directorio
Execución	Executar o arquivo.	Percorrer o súa árbore para acceder arquivos e subdirectorios, pero non velos arquivos dentro do directorio (excepto que se lle dea o permiso de lectura)

Clases de usuarios

Os permisos de sistemas UNIX divídense en catro *clases*, coñecidas como *usuario*, *grupo* e *outros* (con frecuencia abreviado *UGO* polas súas siglas en inglés).

Por lo tanto, as clases de usuarios ás cales se lles poden asignar os **permisos básicos** anteriormente mencionados son:

- **u ? dono**: dono do arquivo ou directorio
- **g ? grupo**: grupo ó que pertence ó arquivo
- **o ? outros**: todos os demais usuarios que non son o dono nin do grupo

Os *permisos efectivos* aplicados a un determinado usuario en relación a un arquivo determínanse nunha orde lóxica de precedencia. Por exemplo, o usuario propietario do arquivo terá os permisos efectivos dados á clase de usuario, sen importar os asignados a clase do grupo ou á clase de outros.

ACLs

O sistema de permisos básicos so permite afinar permisos sobre obxectos para o usuario propietario e grupo propietario, os demais usuarios e grupos van nun *feixe* todos xuntos na clase *outros*.

As Listas de Control de Acceso (ACLs) permiten un acceso máis granular ao sistema de arquivos de GNU/Linux, permitindo indicar que varios usuarios e varios grupos teñan acceso en modo lectura a unha carpeta, por exemplo. Tamén se pode indicar cales deses usuarios ou grupos se deben herdar para os obxectos secundarios (ficheiros e subcarpetas) que se creen dentro desa carpeta.

As acls por agora non veñen de *serie* coa instalación de GNU/Linux. Haberá que instalar o paquete e configurar o sistema de arquivos sobre o que se desexan aplicar.

Tomemos por exemplo unha carpeta chamada **dam1** que alberga as carpetas persoais de todo o alumnado dese grupo. Interesa que a esa carpeta só entre o root (control total) e os alumnos de *dam1*, para logo que cada quen acceda á súa carpeta particular. Finalmente os profesores dese curso poden entrar na carpeta *dam1*, e en todas as subcarpetas e ficheiros (actuais e futuros) en modo lectura.

Para iso esa carpeta *dam1*, terá a seguinte acl:

```
# file: dam1                    -- carpeta/ficheiro sobre o que hai unha acl.
# owner: root                   -- usuario propietario: root
# group: root                   -- grupo propietario: root
user::rwx                       -- Permisos do usuario propietario: rwx
group::---                      -- Permisos do grupo propietario: ningún
group:g-dam1-alum:r-x           -- Grupos que hai na acl: g-dam1-alum:rx . Este grupo non se herdará a
                                -- obxectos secundarios, porque non aparece en default.
group:g-dam1-profes:r-x         -- Grupos que hai na acl: g-dam1-profes:r . Este grupo herdarase a obxectos
                                -- secundarios, porque SI aparece en default.

mask::rwx                       -- limita os permisos efectivos que se conceden aos usuarios e grupos enumerados.
                                -- Os permisos do dono e de "others" non se ven afectados por mask
other::---                      -- Permisos da clase outros: ningún
                                -- Nas seguintes substitúase default por herdar e entenderase mellor.
                                   E pénsese nun obxecto secundario (subcarpeta ou ficheiro)
                                   que se cre dentro da da carpeta dam1.

default:user::rwx               -- Permisos que herdará o usuario propietario: rwx
default:group::---              -- Permisos que herdará o grupo propietario: ningún
default:group:g-dam1-profes:r-x         -- Entrada que herdarán obxectos secundarios futuros: g-dam1-profes: rx.
                                -- PERO OLLO: aquí só aparecen as entradas herdables, pero polo feito de estar
                                   como herdable só afecta ós obxectos secundarios que se creen nun futuro, non
                                   son permisos que ten a carpeta principal. Para asignar os mesmos permisos á
                                   carpeta principal débese indicar explicitamente como na entrada superior:
                                   group:g-dam1-profes:r--


default:mask::r--
default:other::---              -- Permisos que herdará a clase outros: ningún
```

Observar:

- Que o grupo g-dam1-profes aparece dúas veces, unha para os permisos da propia carpeta **dam1** e outra para que os permisos sexan propagados ás subcarpetas e arquivos futuros que se creen dentro de **dam1**.
- Pola contra, o grupo g-dam1-alum só ten permisos de lectura en dam1 e non aparece a maiores nas entradas **default**, por tanto, esa entrada non será propagada aos obxectos secundarios que se creen dentro de **dam1**.

Comandos que se usaran nesta sección

Os dous primeiros son os que xa utilizamos ata agora para manipular os permisos, e os seguintes nos permitirán manipular as ACLs.

chown

- **Descrición**: permite cambiar o usuario e grupo propietario dun ficheiro ou carpeta (change owner).

- **Sintaxe**:

```
Emprego: chown [OPCIÓN]... [DONO][:[GRUPO]]... FICHEIRO...
      O DONO e o GRUPO poden ser numéricos ou simbólicos.

 Opcións máis comúns
  -R, --recursive: opera sobre ficheiros e directorios recursivamente .

Exemplos:
  chown root /u                 Muda o dono de /u para "root".
  chown root:persoal /u         Igualmente, mais muda tamén o seu grupo para "persoal".
  chown -R root /u      Muda o dono de /u e os ficheiros e subcarpetas para "root".
```

chmod

- **Descrición**: permite cambiar os permisos dunha carpeta ou ficheiro usuario e grupo propietario dun ficheiro ou carpeta (change mod).

- **IMPORTANTE**: este comando só o pode executar o usuario root (directamente ou con *sudo chmod*) ou o usuario propietario da carpeta ou ficheiro.

- **Sintaxe**:

```
Emprego: chmod [OPCIÓNS]... permisos ... FICHEIRO/CARPETA...

 Opcións máis comúns
  -R, --recursive: opera sobre ficheiros e directorios recursivamente .

Exemplos:
  chmod 750 /u          Sobre a carpeta /u o usuario ten permisos de (rwx), o grupo (r-x) e os demais ningún.
```

getfacl

- **Descrición**: amosa a lista de control de acceso dunha carpeta ou ficheiro. Este comando ben co paquete acl.

- **Sintaxe**:

```
Emprego: getfacl [OPCIÓNS]... FICHEIRO/CARPETA

 Opcións máis comúns
  -R, --recursive: opera sobre ficheiros e directorios recursivamente .
  -d,  amosa só as entradas herdables da lista de control de acceso.

Exemplos:
  getfacl -R /u         Amosa os permisos básicos e estendidos do directorio /u e do seu contido recursivamente.
```

setfacl

- **Descrición**: introduce, elimina ou modifica entradas da lista de control de acceso.

- **Sintaxe**:

```
Emprego: setfacl [OPCIÓNS] usuario/grupo:permisos FICHEIRO/CARPETA

 Opcións máis comúns
  -b ?- borra tódalas entradas das acl.
```

```
  -k ?- borra tódalas entradas herdables da acl.
  -d  -- modificador que afecta ás entradas herdables.
  -R, --recursive: opera sobre ficheiros e directorios recursivamente .

  -m, --modifica a acl.
  -x,  -- elimina unha entrada da acl.

  Usuario: u:usuario
  grupo: g:grupo

  permisos: r | w | x

Exemplos:
  setfacl -Rm g:users:rx /u    Introduce na acl de /u de tódolos
                               seus subdirectorios e ficheiros permisos
                               de lectura e escritura para o grupo users.
                               Que o faga recursivo só afecta ás carpetas/ficheiros actuais e non ás futuras.

  setfacl -Rdm g:users:rx /u   Igual que caso anterior, pero agora cando
                               se cree unha carpeta/ficheiro dentro de /u
                               tamén vai herdar a entrada da acl onde se
                               indica que os membros do grupo users podes ler e executar.

  setfacl -dx g:users /u       Borra a entrada anterior da acl.
```

Instalación e configuración ACLs

Instalar o paquete **acl** en *dserver00*.

```
sudo apt-get install acl
```

Agora queda activar no ficheiro */etc/fstab* en que sistemas de arquivos se quere que estean activas as ACLs (vaise escoller en /home/iescalquera e /comun). Editar o ficheiro /**etc**/**fstab** e engadir o parámetro **acl** ao punto de montaxe /**home**/**iescalquera** e /**comun**. **Ollo:** Non copiar as seguintes liñas!!!! Cada quen ten un UUID ou un dispositivo de disco/partición distinto.

```
#Partición montada facendo uso de UUID
#/dev/sdb1: LABEL="Usuarios"
UUID=f7d9a85b-5847-449a-9f98-29dfecf4239e       /home/iescalquera       ext4    defaults,acl 0 0

#Partición montada facendo uso do dispositivo /dev/sdb*
#/dev/sdb2: LABEL="Comun" UUID="726f54f6-960b-4ad0-9ec2-35eace42290a"
/dev/sdb2                                       /comun                  ext4    defaults,acl 0 0
```

Para facer efectivo o cambio sen reiniciar o servidor: volver a remontar o sistema de arquivos de /**home**/**iescalquera** e /**comun** para que collan o novo parámetro (acl).

```
sudo mount /home/iescalquera -o remount
sudo mount /comun -o remount
```

Exemplos de manexo das ACLs

Os únicos usuarios que poden cambiar os permisos dunha carpeta ou dun ficheiro son o usuario propietario do obxecto ou o usuario root. Vista a introdución e os comandos *chmod*, *getfacl* e *setfacl* enriba explicados, vaise estudar con distintos exemplos a inserción, modificación e borrado de acls así como a propagación de permisos.

- No servidor, comprobamos os permisos da carpeta /**comun**:

```
root@dserver00:~$ ls / -l

...
drwxr-x--- 5 root g-usuarios 4096 Mai 12 02:35 comun
...
```

- Obter a acl de */comun*, observar a información anterior disposta doutra forma.

```
root@dserver00:~$ getfacl /comun
```

```
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/
# owner: root
# group: g-usuarios
user::rwx
group::r-x
other::---
```

- Imos cambiar os permisos de */comun* para que o grupo propietario sexa *root*, e afinar o que se desexe con acls en lugar de cos permisos de Linux:

```
root@dserver00:~$ chown root:root /comun
root@dserver00:~$ ls / -l
...
drwxr-x--- 5 root root 4096 Mai 12 02:35 comun
...
```

- Obter a acl de */comun* unha vez cambiados os permisos, observar a información anterior disposta doutra forma. Fixarse como agora o grupo propietario da carpeta é *root*.

```
root@dserver00:~$ getfacl /comun
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/
# owner: root
# group: root
user::rwx
group::r-x
other::---
```

- Permitir ao grupo **g-profes** que poida ler e acceder (r-x) á carpeta */comun*, só a esa carpeta. (-m)

```
root@dserver00:~$ setfacl -m g:g-profes:rx /comun

root@dserver00:~$ getfacl /comun
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/
# owner: root
# group: root
user::rwx
group::r-x
group:g-profes:r-x
mask::r-x
other::---
```

Observar como aparece o grupo **g-profes** con permisos r-x.

- Listar carpetas con acls: Observar o caracter **+**, indica que esa carpeta ten unha acl.

```
root@dserver00:~$ ls / -l
...
drwxr-x---+ 5 root root 4096 Mai 12 02:35 comun
...
```

- Crear unha subcarpeta en */comun*: /**comun**/**exames**, e obter as acls de esa carpeta.

```
root@dserver00:~$ mkdir /comun/exames
```

```
root@dserver00:~$ getfacl /comun/exames
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/exames/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

Observar como ao crear a carpeta **exames** non se herdou **g-profes** de **comun**.

- Engadir unha acl para **g-alum** con permisos (r-x), que se propaguen polas subcarpetas e arquivos existentes en */comun*. (-R)

```
root@dserver00:~$ setfacl -Rm g:g-alum:rx /comun

root@dserver00:~$ getfacl /comun
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/
# owner: root
# group: root
user::rwx
group::r-x
group:g-profes:r-x
group:g-alum:r-x
mask::r-x
other::---

root@dserver00:~$ getfacl /comun/exames
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/exames/
# owner: root
# group: root
user::rwx
group::r-x
group:g-alum:r-x
mask::r-x
other::r-x
```

O grupo **g-alum:r-x** está na acl de comun e na da subcarpeta **exames**.

- Finalmente engadir unha entrada á acl, grupo **g-profes:rwx**, que sexa herdable, de xeito que cando no futuro se cre un ficheiro ou subcarpeta-ficheiro en */comun* herde esa entrada automaticamente (-d).

```
root@dserver00:~$ setfacl -dm g:g-profes:rwx /comun
root@dserver00:~$ mkdir /comun/practicas
root@dserver00:~$ getfacl /comun
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/
# owner: root
# group: root
user::rwx
group::r-x
group:g-profes:r-x
group:g-alum:r-x
mask::r-x
other::---
default:user::rwx
default:group::r-x
default:group:g-profes:rwx
default:mask::rwx
default:other::---

root@dserver00:~$ getfacl /comun/exames
```

```
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/exames/
# owner: root
# group: root
user::rwx
group::r-x
group:g-alum:r-x
mask::r-x
other::r-x

root@dserver00:~$ getfacl /comun/practicas
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/practicas/
# owner: root
# group: root
user::rwx
group::r-x
group:g-profes:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:g-profes:rwx
default:mask::rwx
default:other::---
```

- Observar:
  - **o grupo g-profes**: non ten permisos de escritura sobre /**comun** pois só está como herdable para futuros obxectos secundarios (neste caso *practicas*). Co cal un usuario do grupo **g-profes** pode escribir en */comun/practicas*, pero non en /**comun**.
  - **o grupo g-profes**: non está en /**comun**/**exames**, pero si na carpeta **practicas** que se creou despois de introducir a acl en */comun*. **default** indica que esa entrada é herdable. **g-profes** non aparece en *exames*, porque esa subcarpeta xa estaba creada antes de meter a entrada na acl.

- Para que un usuario de **g-profes** poida escribir a carpeta /**comun** é preciso modificar a súa ACL:

```
root@dserver00:~$ setfacl -m g:g-profes:rwx /comun

root@dserver00:~$ getfacl /comun
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/
# owner: root
# group: root
user::rwx
group::r-x
group:g-profes:rwx
group:g-alum:r-x
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:g-profes:rwx
default:mask::rwx
default:other::---
```

Agora si, un usuario de **g-profes** pode escribir en */comun* e nas subcarpetas nas que teña permisos.

- Como borrar unha acl?: eliminarase o grupo **g-alum** da acl. Primeiro de */comun* e logo recursivamente de todas as subcarpetas. (-x)

```
root@dserver00:~$ setfacl -x g:g-alum /comun

root@dserver00:~$ getfacl /comun
getfacl: Removing leading '/' from absolute path names
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/
```

```
# owner: root
# group: root
user::rwx
group::r-x
group:g-profes:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:g-profes:rwx
default:mask::rwx
default:other::---

root@dserver00:~$ getfacl /comun/exames
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/exames/
# owner: root
# group: root
user::rwx
group::r-x
group:g-alum:r-x
mask::r-x
other::r-x

root@dserver00:~$ getfacl /comun/practicas
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: srv/samba/comun/practicas/
# owner: root
# group: root
user::rwx
group::r-x
group:g-profes:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:g-profes:rwx
default:mask::rwx
default:other::---
```

Observar que **g-alum** está eliminado da acl de */comun* pero non da subcarpeta *exames*.

- Para eliminalo de */comun* e de todas as súas subcarpetas e arquivos hai que facelo recursivamente: (-Rx)

```
root@dserver00:~$ setfacl -Rx g:g-alum /comun

root@dserver00:~$ getfacl /comun
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/
# owner: root
# group: root
user::rwx
group::r-x
group:g-profes:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:g-profes:rwx
default:mask::rwx
default:other::---

root@dserver00:~$ getfacl /comun/exames
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/exames/
# owner: root
# group: root
user::rwx
group::r-x
```

```
mask::r-x
other::r-x

root@dserver00:~$ getfacl /comun/practicas
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/practicas/
# owner: root
# group: root
user::rwx
group::r-x
group:g-profes:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:g-profes:rwx
default:mask::rwx
default:other::---
```

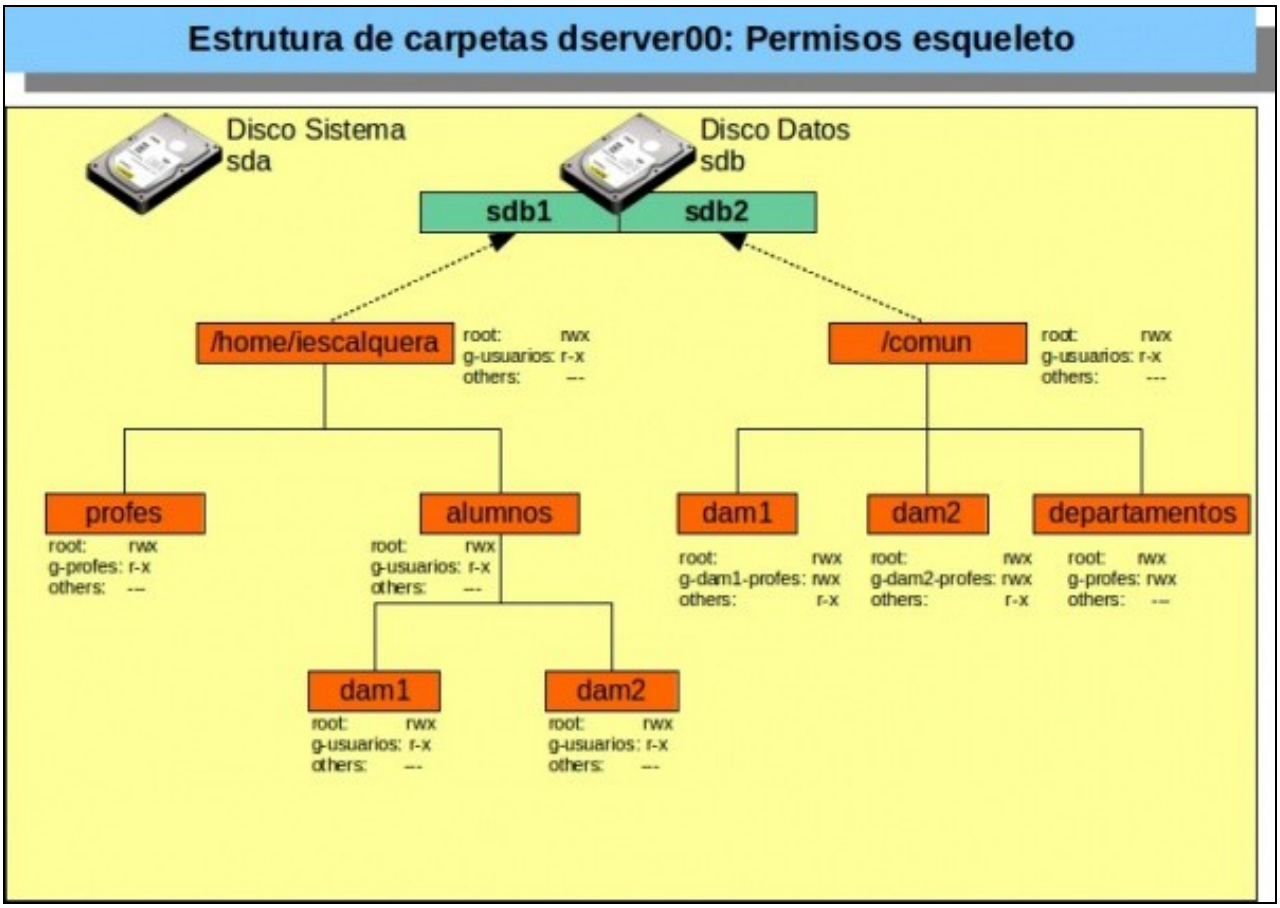
- Por último, e como estas ACLs que configuramos son simplemente de proba e non son as que nos interesan para o noso esqueleto de carpetas, imos borrar todas as ACLs de */comun* e das súas subcarpetas:

```
root@dserver00:~$ setfacl -bR /comun

root@dserver00:~$ getfacl /comun
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/
# owner: root
# group: root
user::rwx
group::r-x
other::---

root@dserver00:~# getfacl /comun/exames/
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/exames/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x

root@dserver00:~# getfacl /comun/practicas/
getfacl: Eliminando '/' iniciais en nomes de ruta absolutos
# file: comun/practicas/
# owner: root
# group: root
user::rwx
group::r-x
other::---
```

- Á vista dos exemplos anteriores:
    - Para traballar con usuarios no canto de grupos, usar: **u:usuario**
    - Non confundir recursividade (-R) con herdanza (-d). O primeiro afecta a carpeta e subcarpetas/arquivos existentes, o segundo non afecta á carpeta senón ás subcarpetas e arquivos que se creen nun futuro.
    - Para modificar os permisos do usuario propietario, pódese usar chmod ou: **setfacl -opcións u::permisos /comun**. Para o grupo propietario sería semellante pero g:: no canto de u::.

Axustar permisos do noso esqueleto

O seguinte esquema representa a estrutura actual de permisos establecido no esqueleto de carpetas que deseñamos para o noso dominio:



Isto podería ser suficiente en moitos casos, pero imos presentar algunhas situacións nas que o esquema de permisos de Linux non nos permite acadar unha solución e si poderíamos resolver co uso de ACLs:

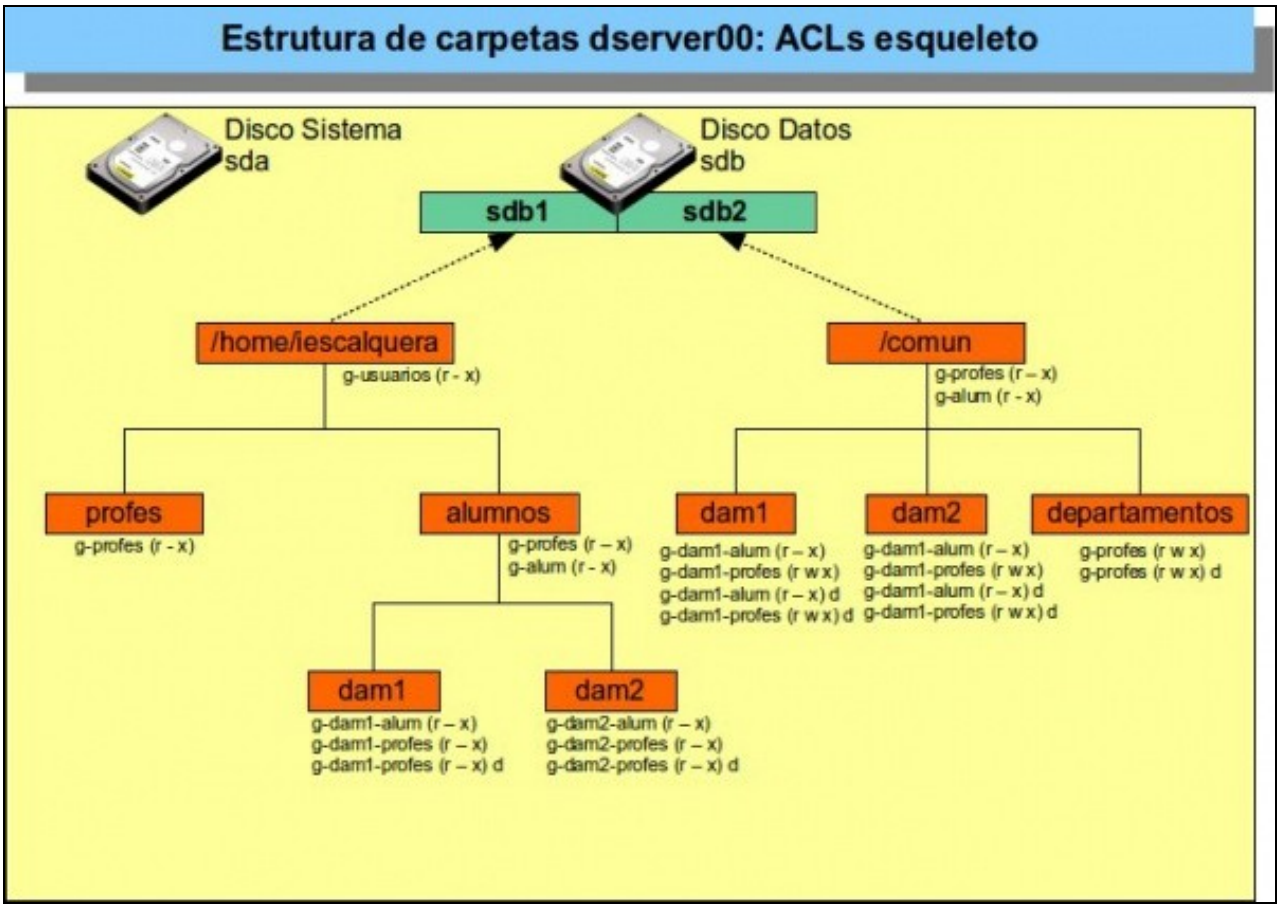
- Coa estrutura actual de permisos, calquera usuario do grupo *g-usuarios* podería entrar dentro da carpeta de /home/iescalquera/alumnos e ver o seu contido. Se houbese usuarios dentro deste grupo que non fosen profes nin alumnos isto non nos interesaría, e para evitalo teríamos que permitir a entrada á carpeta ao grupo *g-profes* e ao grupo *g-alum*, e denegar o acceso ao resto de usuarios.

- Un alumno de *dam1* pode ver o contido da carpeta */home/iescalquera/alumnos/dam2* (aínda que non poida ver o contido das súas subcarpetas), cousa que non nos interesaría. O ideal é que nesa carpeta só poidan acceder os usuarios dos grupos '*g-dam2-profes* e *g-dam2-alum*.

- Máis importante é o que ocorre na carpeta de */comun/dam2*, xa que un alumno de *dam1* podería acceder ao contido desta carpeta. Sería desexable que só puidesen entrar para ler e escribir os usuarios de *g-dam2-profes* e os de *g-dam2-alum* só para ler, mentres que o resto dos usuarios non debería poder acceder a esta carpeta.

- Por último, utilizando permisos herdables poderíamos configurar os permisos por defecto adecuados para as carpetas persoais dos usuarios sen necesidade dun script que os modifique expresamente.

O seguinte esquema de ACLs solventaría estes aspectos:



Para establecer as ACLs do esquema, podemos crear o script co seguinte contido.

SCRIPT: 02_axustar_acls_esqueleto.sh

```
#!/bin/bash

#Chamar ao scrip de variables
. ./00_variables.sh # Tamén podería ser: source ./00_variables.sh

#Cartafol de usuarios
chown root:root $DIR_HOME_LDAP
chmod 750 $DIR_HOME_LDAP
setfacl -m g:g-usuarios:rx $DIR_HOME_LDAP

#Cartafol profes
chown root:root $DIR_HOME_LDAP/profes
chmod 750 $DIR_HOME_LDAP/profes
setfacl -m g:g-profes:rx $DIR_HOME_LDAP/profes

#Cartafol alumnos
chown root:root $DIR_HOME_LDAP/alumnos
chmod 750 $DIR_HOME_LDAP/alumnos
setfacl -m g:g-profes:rx $DIR_HOME_LDAP/alumnos
setfacl -m g:g-alum:rx $DIR_HOME_LDAP/alumnos

#Cartafoles cursos
for CURSO in $(cat f00_cursos.txt)
do
        chown root:root $DIR_HOME_LDAP/alumnos/$CURSO
        chmod 750 $DIR_HOME_LDAP/alumnos/$CURSO
        setfacl -m g:g-$CURSO-alum:rx $DIR_HOME_LDAP/alumnos/$CURSO
        setfacl -Rm g:g-$CURSO-profes:rx $DIR_HOME_LDAP/alumnos/$CURSO
        setfacl -dm g:g-$CURSO-profes:rx $DIR_HOME_LDAP/alumnos/$CURSO
done

#Cartafol comun
chown -R root:root $DIR_COMUN
chmod -R 750 $DIR_COMUN
```

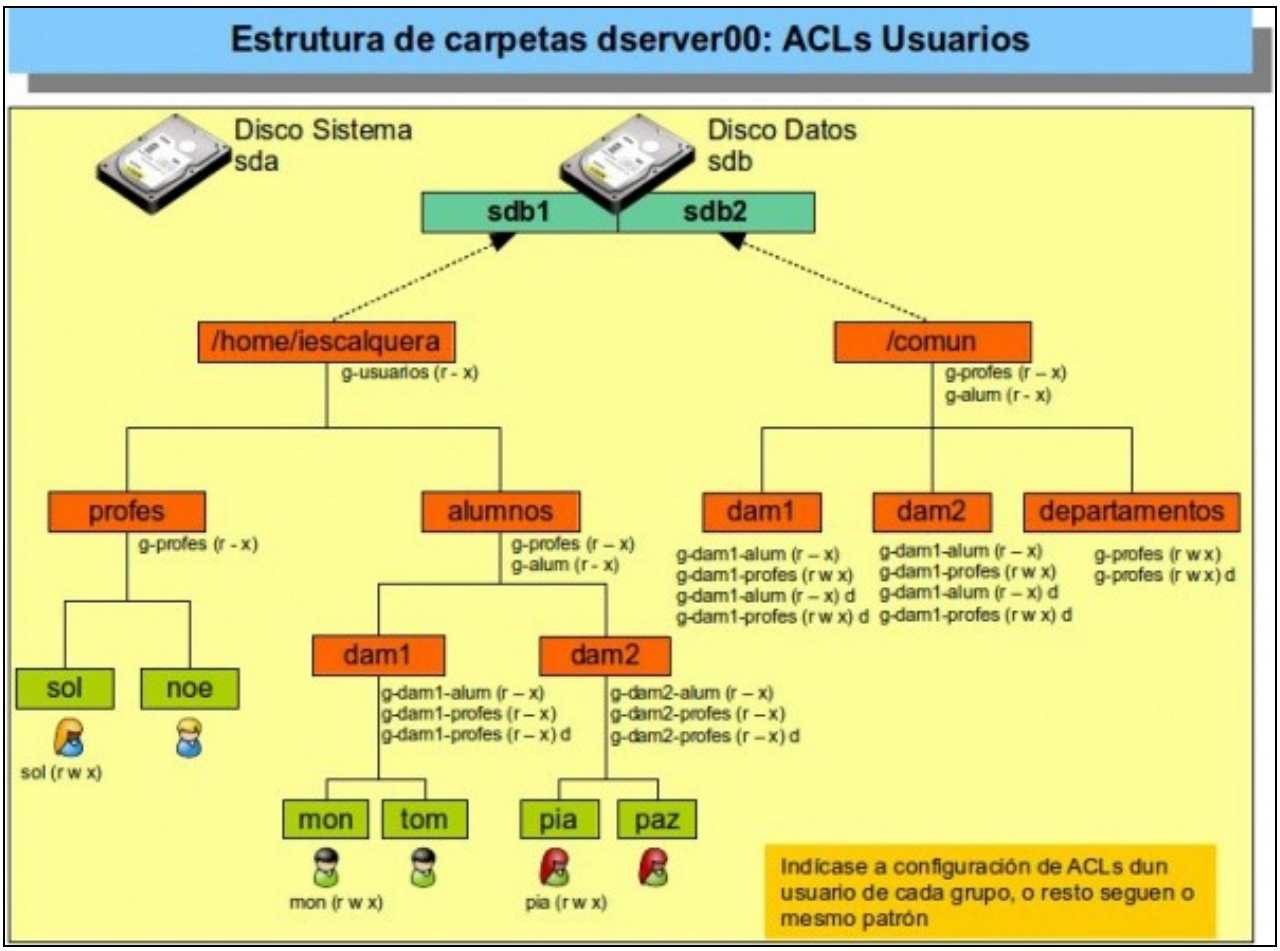
```
setfacl -m g:g-alum:rx $DIR_COMUN
setfacl -m g:g-profes:rx $DIR_COMUN

#Subcartafol departamentos
setfacl -Rm g:g-profes:rwx $DIR_COMUN/departamentos
setfacl -dm g:g-profes:rwx $DIR_COMUN/departamentos


#Subcartafoles cursos
# O participante no curso á vista do esquema de permisos
# do exemplo de arriba debe ser quen de axustar
# os permisos de /comun/cursos
```

- Executamos o script

```
sh 02_axustar_acls_esqueleto.sh
```

Permisos para as carpetas persoais dos usuarios

Con respecto ás carpetas persoais dos usuarios, agora o único necesario é que cada usuario teña permisos de lectura e escritura sobre a súa carpeta (e isto pódese facer cos permisos de Linux ou con ACLs). O esquema sería o seguinte:



Así que modificaríamos o script que crea as carpetas persoais para deixalo como se mostra a continuación.

SCRIPT: 03_crear_home_usuarios_axustar_acls.sh

```
#!/bin/bash


#Lembrar que  cada usuario ten o seguinte formato
# Un/unha profe  -> sol:x:10000:10000:Profe - Sol Lua:/home/iescalquera/profes/sol:/bin/bash
# Un/unha alumna -> mon:x:10002:10000:DAM1 Mon Mon:/home/iescalquera/alumnos/dam1/mon:/bin/bash

# Observar que posición ocupan os campos e que están separados por :

# Imos etraer con awk dos usuarios con ID (campo 3) entre 10000 e 60000 os campos
# Usuario (campo 1) e home (campo 6)
# Deste último campo (home) imos extraer o grupo ao que pertence o usuario
# Neste caso o separador de campos é /, e o grupo está no 4º campo.


#Volcamos tódolos usuarios (locais e ldap) do sistema a un ficheiro
getent passwd>usuarios.txt


#Extraemos os campos anteriores
for USUARIO in $( awk -F: '$3>=10000 && $3<60000  {print $1":"$6}' usuarios.txt )
do
#USUARIO vai ter o seguinte formato
# sol:/home/iescalquera/profes/sol

NOME_USUARIO=$( echo $USUARIO | awk -F: '{print $1}')
HOME_USUARIO=$( echo $USUARIO | awk -F: '{print $2}')

#Creamos a carpeta persoal do usuario/a
test -d $HOME_USUARIO || mkdir -p $HOME_USUARIO

#Copiamos o contido de skel_ubuntu (ocultos incluídos, -a) á carpeta persoal do usuario/a
        cp -a skel_ubuntu/. $HOME_USUARIO

        #Con estes dous comandos estableceríamos os permisos de Linux
#chown -R $NOME_USUARIO $HOME_USUARIO
#chmod -R 700 $HOME_USUARIO

        #Estes comandos establecen os permisos por ACLs
        chmod -R 750 $HOME_USUARIO
        setfacl -m u:$NOME_USUARIO:rwx $HOME_USUARIO
done

rm usuarios.txt
```

Uso de ACLs a través de NFS

Se se exporta unha carpeta con ACLs a través de NFS, ó proceso é transparente nas actuais versións de NFS.

Antes de continuar no cliente débese instalar o paquete acl do mesmo xeito que se fixo do servidor:

```
sudo apt-get install acl
```

Exportar /comun

Engadir en server00 ó ficheiro **/etc/exports** unha nova exportación.

```
# /etc/exports
/comun  *(rw,async)
# O * indica que a exportación é para calquera rede IP.
```

Facer efectiva a exportación:

```
sudo exportfs -ra
```

Montar /comun no cliente

Crear no cliente /**mnt/comun**

```
sudo mkdir /mnt/comun
```

Editar o ficheiro /etc/fstab do cliente e engadir o punto de montaxe NFS:

```
#/etc/fstab do cliente
10.0.0.100:/comun /mnt/comun nfs defaults,vers=3        0       0
```

Montar os puntos de montaxe de /etc/fstab que non estean activos:

```
sudo mount -a
```

- Comprobar as ACLs exportadas.

```
administrador@cliente00:~$ getfacl -R /mnt/comun

getfacl: Removing leading '/' from absolute path names
# file: mnt/comun
# owner: root
# group: root
user::rwx
user:root:r-x
user:alfredo:r-x
user:xan:rwx
group::r-x
mask::rwx
other::---
default:user::rwx
default:user:xan:rwx
default:group::r-x
default:mask::rwx
default:other::---

getfacl: /mnt/comun: Permission denied
```

Observar como ó usuario administrador do cliente non lle deixa acceder as ACLs das subcarpetas de */mnt/comun*. Iso é porque o usuario administrador do cliente caería dentro da clase **outros**, que xustamente non ten ningún permiso na ACL de */mnt/comun*.

Xestionar as ACLs no cliente

Lembrar:

- que os permisos só poden ser cambiados polo usuario propietario ou polo usuario root.
- as exportacións no server fixéronse **Confiando en todo o mundo excepto root**: Desa forma o servidor NFS mapeará o UID do usuario do equipo cliente cun usuario do servidor excepto se se trata do usuario root, que será mapeado como usuario anónimo.

Por tanto no cliente só poden xestionar os permisos dunha exportación os usuarios propietarios do server que non sexan root.

Entón tense:

- No exemplo que se ten co punto de montaxe: /**mnt**/**comun** O usuario propietario dese **comun** é o root do servidor non o root do cliente, por tanto o usuario root do cliente non vai poder cambiar ningún permiso, nin tomar posesión, cambiar o propietario, etc, das carpetas e subcarpetas exportadas por un servidor.
- Os usuarios do servidor que sexan donos de carpetas, subcarpetas ou ficheiros exportadas si van poder modificar os permisos de aquelas nas que son donos.
- Se o usuario root desexa facer cambios, debe ser o root no servidor, ben a través de **ssh** ou directamente no servidor, ou facendo uso de **sudo**, pero sempre no servidor.

Xestión gráfica de ACLs: eiciel

Todo canto se fixo anteriormente pode ser realizado cunha ferramenta gráfica chamada Eiciel creada por un Catalán Roger Ferrer.

Esta ferramenta precisa dun entorno gráfico, por tanto, só se pode instalar nos clientes, e facilítalle ós usuarios *normais* a xestión de permisos, pois engade as solapas de propiedades dunha carpeta ou un ficheiro, unha nova lapela para xestionar graficamente a lista de control de acceso.

Eiciel, o que fai é traballar directamente coas acls, co cal estas poden ser modificadas tanto por consola como graficamente.

- Instalar Eiciel no cliente:

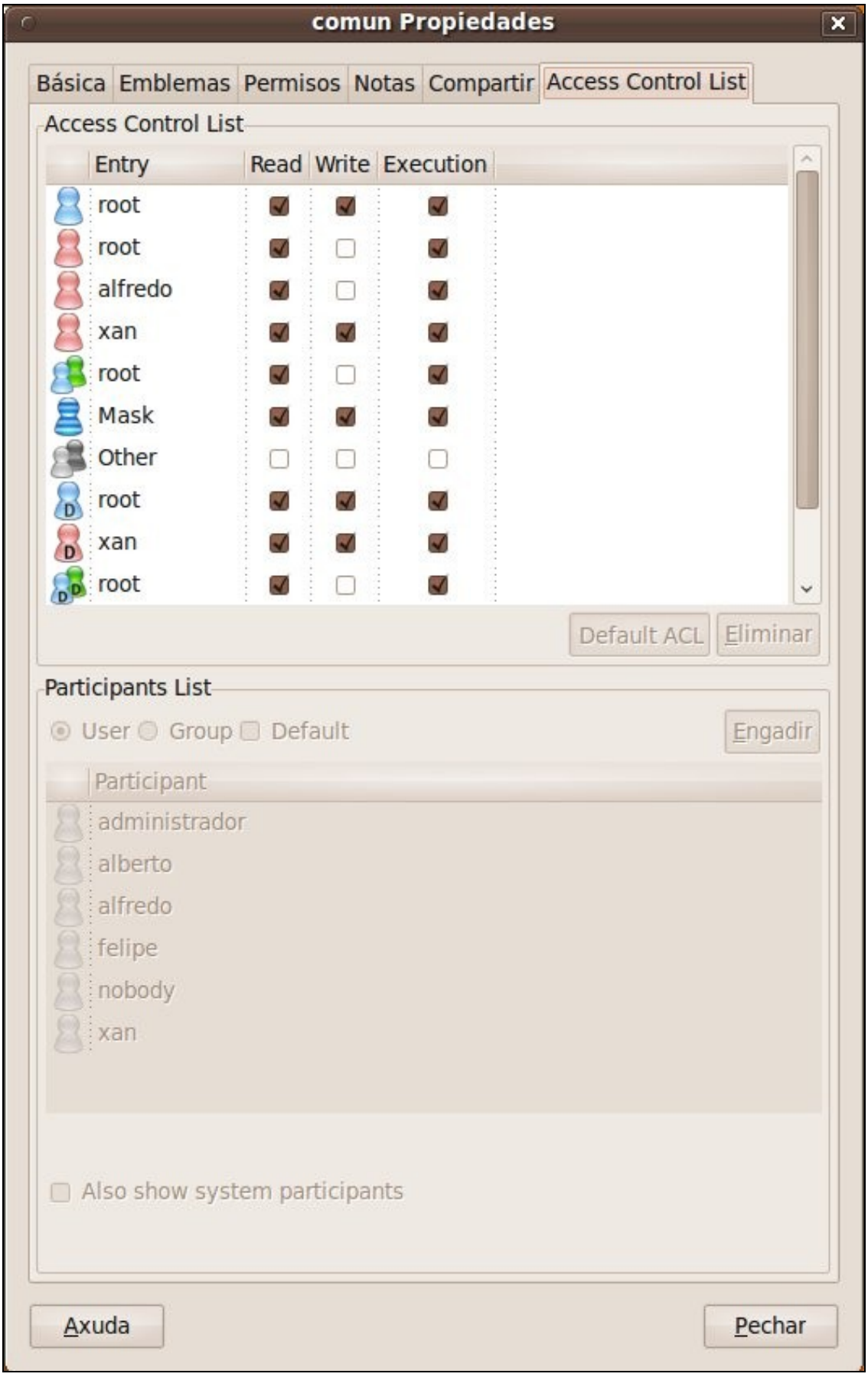
```
sudo apt-get install eiciel
```

- No menú do cliente: **Aplicativos->Ferramentas do sistema** hai unha entrada para Eiciel.

Pero eiciel nin as acls poden ser usados en ningún dos sistema de arquivos físicos do cliente, pois no */etc/fstab* do cliente non está o parámetro acl para ningún dos puntos de montaxe.

Para os puntos de montaxe nfs, si se poden usar as acls/eiciel pois as listas de control de acceso son transportadas por nfs de xeito transparente.

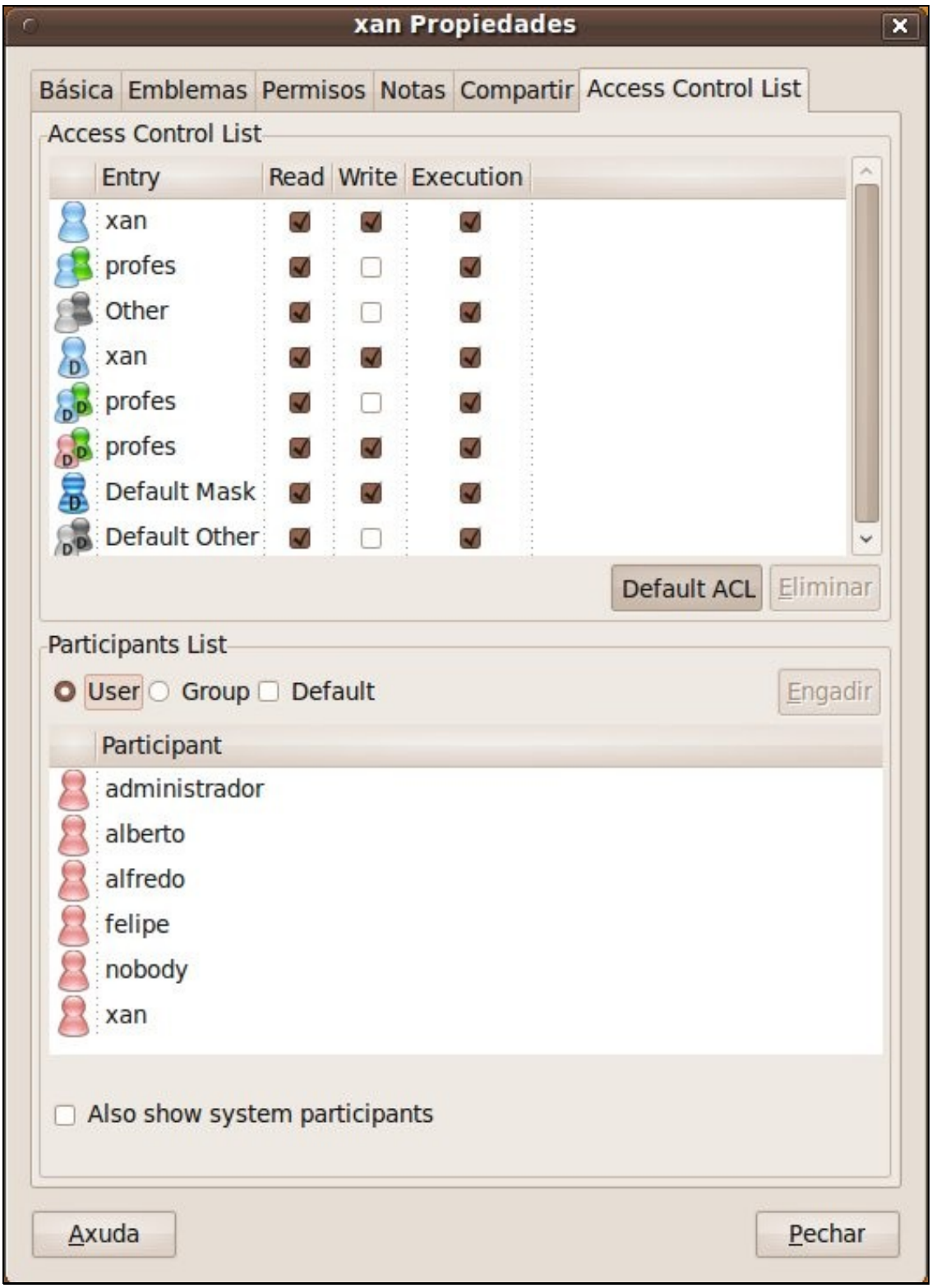
- Co usuario **administrador** do cliente usando nautilus ir a propiedades da carpeta **comun** que está en /**mnt** (**Ollo:** Despois de instalar o paquete eiciel, teremos que reiniciar a sesión para que a pestana de ACLs apareza).


comun Propiedades
Básica
Emblemas
Permisos
Notas
Compartir
Access Control List
Access Control List
Entry
Read
Write
Execution
root
root
alfredo
xan
root
Mask
Other
root
xan
root
Default ACL
Eliminar
Participants List
User
Group
Default
Engadir
Participant
administrador
alberto
alfredo
felipe
nobody
xan
Also show system participants
Axuda
Pechar

Observar como o usuario **administrador** do cliente só pode ver pero non cambiar nada.

Para familiarizarse coa interpretación das iconas, cores e formatos véxase a seguinte táboa 1 do enlace:
http://rofi.roger-ferrer.org/eiciel/doc/ar01s02.html#manipular-acl

- Pechar a sesión co cliente e agora entrar co usuario **xan** en contorno gráfico. Ir as propiedades da súa carpeta persoal.



Observar como agora xan pode engadir máis usuario, grupos, poñer usuarios e grupos por defecto (herdables), etc.

Agora **xan** pode ir a comun (/mnt/comun) do cliente e crear obxectos.

-- Antonio de Andrés Lema e Carlos Carrión Álvarez