

Gli ACL ci permettono di espandere il controllo in termini di possibilità di accesso in modalità scrittura, lettura ed esecuzione ai vari file. Chiaramente vogliamo gestire i permessi a grana più fine. Noi potremmo avere un utente all'interno di un gruppo che non è il nostro, a cui vogliamo dare dei permessi, con lo schema delle triplettine dovevamo dare i permessi a tutto quel gruppo, nella sezione G di UGO. Noi vogliamo la granularità. Noi vogliamo dare i permessi soltanto a quello specifico utente. Questa cosa ci è permessa, nel momento in cui noi lavoriamo con le ACL. Tramite gli I-NODES noi possiamo specificare dei permessi di accesso che sono a grana molto più fine.

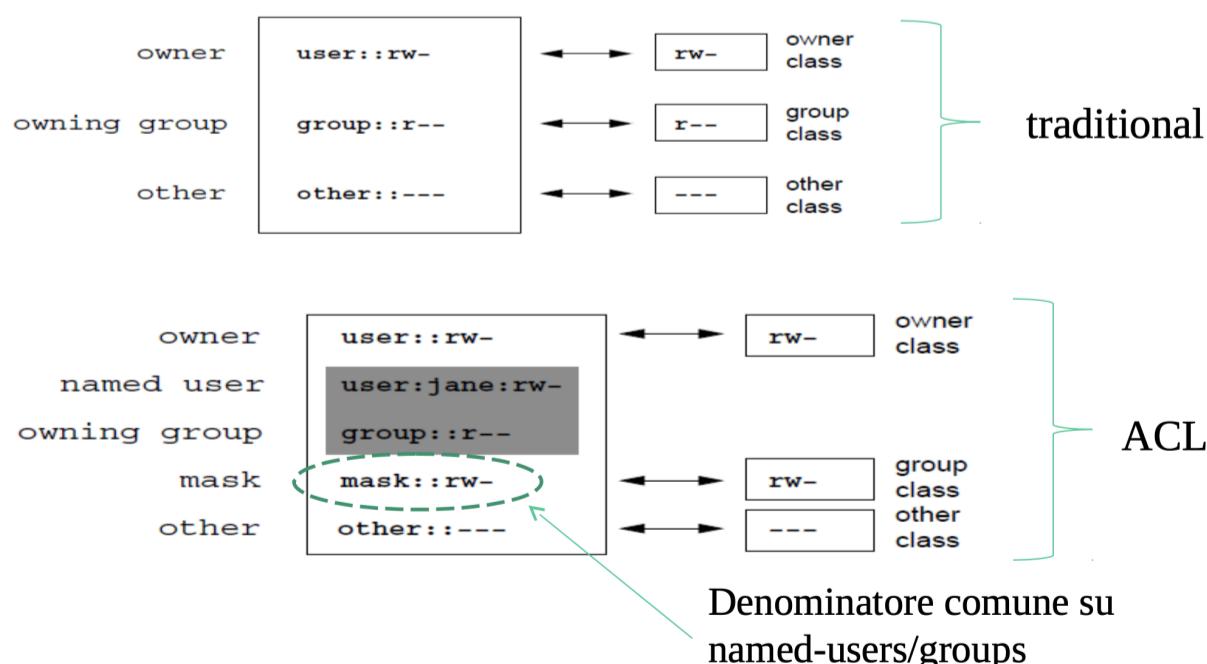
- Permette di avere i-nodes in cui specifichiamo a grana molto più fine i permessi di accesso e gestione dei file
- Una ACL è realizzata tramite un **file shadow** (associato al file originale)
- Quindi anche tramite un **i-node shadow**
- L'i-node originale in tal caso ha un indice che identifica l'i-node shadow
- Con ACL si può per ogni file specificare i permessi di accesso per ogni singolo utente o gruppo del sistema
- I comandi basici di shell per gestire ACL sono **getfacl** e **setfacl**

Quando noi abbiamo un ACL associato ad un file, associamo a questo file un cosiddetto FILE SHADOW, e all'interno di questo file shadow andiamo a specificare delle cose in più a riguardo di cosa può avvenire su questo file (permessi).

La rappresentazione di ACL all'interno di un file system è fatta in modo tale per cui, se noi abbiamo un file F e per questo file F vogliamo avere un ACL, dobbiamo associare un file SHADOW (un file ombra che non è accessibile come contenuto utilizzabile dalle applicazioni, ma accessibile come informazioni di sistema) e così abbiamo un I-NODE per il file F e abbiamo un I-NODE associato al file SHADOW di questo file F. All'interno di questo file shadow andiamo a specificare delle informazioni con una grana più fine per quanto riguarda ciò che può avvenire su questo file F: AD ESEMPIO QUALI SONO I THREAD CHE POSSONO LEGGERE O SCRIVERE SUL FILE, E i thread sono associati a codici numerici. Anche le ACL sono codici numerici. Quindi andiamo a specificare quali sono le utenze che possono lavorare sul contenuto di questo file.

Quindi se io eseguo una setfacl0 dico che voglio modificare - o creare ora, se non esiste - questo file shadow in cui sto andando a scrivere qualche informazione di protezione in più con grana più fine per questo oggetto di I/O.
Getfacl0 va a prelevare le informazioni.

ACL example



Originariamente noi avevamo la possibilità di distinguere i permessi sull'owner, del suo gruppo di appartenenza e delle altre utenze.

Con un ACL noi possiamo "estendere" questo modo di proteggere le informazioni.

Possiamo dire cosa deve succedere per le attività dell'owner (user).

Dire che cosa deve succedere nel resto del mondo (other).

Dire che cosa deve succedere nel gruppo di appartenenza (group).

Dire per SPECIFICHE utenze (JANE) quali sono per lei le possibilità di operare.

Con la maschera resettiamo e blocchiamo le utenze e le operazioni che a loro sono garantite. La maschera è il rilessio di tutto ciò che abbiamo al gruppo e alle utenze nominali specifiche.

ESEMPIO

The terminal window shows a Makefile named 'Makefile' with the following content:

```
getACL:
    getfacl pippo

setACL:
    setfacl -m u:8765:r pippo
    setfacl -m g:8765:r pippo

maskACL:
    chmod 000 pippo

resetACL:
    setfacl -b pippo
```

Below the Makefile, the terminal command is run:

```
> make setACL pippo
```

The output of the command is:

```
PLES/VIRTUAL-FILE-SYSTEM/UNIX/ACL> make getACL
getfacl pippo
# file: pippo
# owner: francesco
# group: users
user::rw-
user:8765:r--
group::r--
group:8765:r--
mask::r--
other::r--
```

E per sapere l'i-node associato a ciascuno degli elementi?

```
PLES/VIRTUAL-FILE-SYSTEM/UNIX/ACL> ls -i
11797263 Makefile 11797598 pippo
```

Abbiamo l'i-node con l'indice 11797263 per il make file e l'altro i-node per pippo.

```
PLES/VIRTUAL-FILE-SYSTEM/UNIX/ACL> ls -lai
total 12
11797262 drwxr-xr-x  2 francesco users  4096 Apr 14 12:24 .
11797250 drwxr-xr-x 10 francesco users  4096 Apr 14 10:50 ..
11797263 -rw-r--r--  1 francesco users   146 Apr 19 2020 Makefile
11797598 -rw-r--r--+ 1 francesco users     0 Apr 14 12:24 pippo
```

Per reperire altre informazioni in generale su questo file system usiamo il comando df

```
PLES/VIRTUAL-FILE-SYSTEM/UNIX/ACL> df
Filesystem      1K-blocks    Used Available Use% Mounted on
devtmpfs          8071320      0   8071320   0% /dev
tmpfs            8080056  118756  7961300   2% /dev/shm
tmpfs            8080056   67256  8012800   1% /run
tmpfs            8080056      0  8080056   0% /sys/fs/cgroup
/dev/nvme0n1p2  41943040 17522628 24245820 42% /
/dev/nvme0n1p2  41943040 17522628 24245820 42% /boot/grub2/i386-pc
/dev/nvme0n1p2  41943040 17522628 24245820 42% /var
/dev/nvme0n1p2  41943040 17522628 24245820 42% /tmp
/dev/nvme0n1p2  41943040 17522628 24245820 42% /usr/local
/dev/nvme0n1p2  41943040 17522628 24245820 42% /root
/dev/nvme0n1p2  41943040 17522628 24245820 42% /.snapshots
/dev/nvme0n1p2  41943040 17522628 24245820 42% /boot/grub2/x86_64-e
/dev/nvme0n1p2  41943040 17522628 24245820 42% /opt
/dev/nvme0n1p2  41943040 17522628 24245820 42% /srv
/dev/nvme0n1p1    511720    5112   506608   1% /boot/efi
/dev/nvme0n1p3 433511304 310135344 101285056 76% /home
tmpfs           1616008     20  1615988   1% /run/user/1000
```

Questo ci va a listare tutti i file system che sono correntemente

gestitul

| Filesystem | Type | Size | Used | Avail | Use% | Mounted on |
|----------------|----------|------|------|-------|------|------------------------|
| devtmpfs | devtmpfs | 7.7G | 0 | 7.7G | 0% | /dev |
| tmpfs | tmpfs | 7.8G | 116M | 7.6G | 2% | /dev/shm |
| tmpfs | tmpfs | 7.8G | 66M | 7.7G | 1% | /run |
| tmpfs | tmpfs | 7.8G | 0 | 7.8G | 0% | /sys/fs/cgroup |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | / |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | /boot/grub2/i386-pc |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | /var |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | /tmp |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | /usr/local |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | /root |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | /.snapshots |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | /boot/grub2/x86_64-efi |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | /opt |
| /dev/nvme0n1p2 | btrfs | 40G | 17G | 24G | 42% | /srv |
| /dev/nvme0n1p1 | vfat | 500M | 5.0M | 495M | 1% | /boot/efi |
| /dev/nvme0n1p3 | ext4 | 414G | 296G | 97G | 76% | /home |
| tmpfs | tmpfs | 1.6G | 20K | 1.6G | 1% | /run/user/1000 |