Nei sistemi Unix, sono presenti le astrazioni di **utenti** e **gruppi** (rispettivamente *user* e *groups*)
Un **utente** può corrispondere ad una persona umana o ad una entità usata per dichiarare chi utilizza uno specifico servizio di sistema (ad esempio l'*user* `mysql` esegue il servizio del database `mysql`).
Ogni utente è caratterizzato da una stringa di caratteri, chiamata ***username***, che contiene il nome dell'utente, e da un identificatore numerico chiamato ***userID***: entrambi sono univoci nel sistema. Ogni gruppo è similmente caratterizzato da una stringa di caratteri, chiamata ***groupname***, che contiene il nome del gruppo, e da un identificatore numerico chiamato ***groupID***: anche in questo caso sono entrambi univoci nel sistema.

Un utente appartiene ad un gruppo **primario** e può appartenere ad uno o più gruppi **secondari**:

- Gruppo **primario**: quando l'utente crea un file, viene associata l'appartenenza di tale file al gruppo primario a cui l'utente appartiene.
- Gruppo **secondario**: sono gruppi aggiuntivi a cui un utente può appartenere. Il fatto di appartenere a questi gruppi garantisce i permessi definiti per tali gruppi per quanto riguarda i file o le directory. Questa cosa permette di creare, ad esempio, directory e file condivisi tra più utenti.

Ogni file e directory ha un **utente proprietario** detto ***owner*** - di default è l'utente che ha *creato il file o la directory*. Ogni file e directory è associata ad **un solo gruppo**.

- Inizialmente il gruppo associato a un file/directory è il gruppo primario dell'utente che ha creato il file.
- Dopodiché il gruppo di appartenenza può essere cambiato dal proprietario del file, tramite il comando `chgrp`.
- L'utente proprietario di un file può inoltre cambiare il proprietario di tale file, tramite il comando `chown`: questo comando permette anche di cambiare l'*owner* e il gruppo di appartenenza in contemporanea.

```
# Supponiamo che il file ciao.txt ha come owner vulpi e come gruppo di
appartenenza utenti

chown chiavicembalo ciao.txt

# Ora l'owner del file ciao.txt è chiavicembalo
```

## I permessi

Un utente può tentare di interagire con un file o una directory di cui non è il proprietario: in questo caso tale utente viene chiamato ***effective user***, che evidenzia la differenza tra l'utente proprietario e l'utente che cerca di accedere al file.
Il proprietario del file può stabilire chi e come può interagire con il file: ciò è possibile tramite l'impostazione dei **permessi** di tale file, che si suddividono in 3 categorie:

- I permessi associati al proprietario del file (ossia sè stesso)
- I permessi associati al gruppo di appartenenza del file
- I permessi associati agli utenti esterni ai primi due canoni (tutti gli altri).

Il grado di permessi che una certa categoria possiede su un file o su una directory è determinato da un numero che va da 0 a 7, ottenuto dalla somma dei numeri che corrispondono ai seguenti permessi:

- Permesso di **lettura**: permette di leggere i contenuti di un file o di una directory. Ha come valore **4** e simbolo `r`.
- Permesso di **scrittura**: permette di modificare i contenuti di un file e di creare, rinominare ed eliminare file in una directory. Ha come valore **2** e come simbolo `w`
- Permesso di **esecuzione**: permette l'esecuzione di un file e l'accesso ad una directory (permette anche di visualizzarne le proprietà). Ha come valore **1** e come simbolo `x`.
  Al valore 6 sono associati i permessi `r` e `w`, al 3 `w` e `x`.
  E' possibile cambiare i permessi di un file se si è il proprietario, tramite il comando `chmod`:

```
chmod 735 ./ciao.txt

# I permessi per il file ciao.txt vengono impostati a:
# 7 (rwx) per il proprietario
# 3 (wx) per il gruppo di appartenenza
# 5 (rx) per gli altri utenti
```

E' possibile visualizzare i permessi associati ad una serie di file tramite il comando `ls`; di base `ls` mostra tutti i file presenti all'interno di una specifica directory (di *default* la directory in cui ci si trova attualmente). E' possibile usare il tag `-al` per visualizzare **i file nascosti** e **informazioni aggiuntive relative ai file**, tra cui i permessi.

```
# Supponiamo che nel Desktop ci sia un file main.exe
ls -al /home/vulpi/Desktop

# Output:
# -rwxrwxr-x 1 vulpi vulpi 8608 dic 17 2015 main.exe
```

La prima stringa di caratteri riporta **informazioni relative ai permessi**:

- Il primo carattere è il ***tipo del file***: `-` se è un file, `d` se è una directory, ecc...
- I primi 3 caratteri successivi al primo sono i ***permessi dell'owner*** del file.
- I secondi 3 caratteri sono ***i permessi del gruppo associato*** al file.
- Gli ultimi 3 caratteri sono ***i permessi per gli altri utenti***.

Esistono altri 3 permessi speciali associabili a file o directory:

- Permesso `setuid`: quando viene eseguito un file, l'effective user ottiene anche **i diritti dell'owner del file**. Rappresentato da `s` nei permessi utente al posto di `x`, impostabile con `4***`
- Permesso `setgid`: per i file è analogo al `setuid` ma per il gruppo di appartenenza; per le directory implica che tutte le sottodirectory create all'interno della directory stessa **ereditano il groupID del gruppo di appartenenza**. Rappresentato da `s` nei permessi del gruppo di appartenenza, impostabile con `2***`
- Permesso `sticky`: i file all'interno di una directory in cui il permesso `sticky` è impostato possono essere modificati, spostati o eliminati ***solo dal proprietario del file o della directory*** (oppure da root). Rappresentato da `t`, impostabile con `1***`
  NB: *al posto di ** *mettere i permessi da associare a owner, group e others.*