

# GNU/Linux

A large, faint, light-colored illustration of Tux the penguin is centered in the background of the slide. Tux is shown from the chest up, facing forward, with his characteristic black body, white belly, and large eyes.

GNU/Linux è un sistema:

- **Multitasking:** riesce a gestire più processi contemporaneamente.
- **Memoria Protetta:** ogni processo in corso utilizza una ben determinata area di memoria per il proprio funzionamento.
- **Multiutente:** diverse persone possono collegarsi al sistema contemporaneamente e gestire le proprie attività.

# Multiutenza



Questo è l'aspetto che a noi interessa di più e andremo ad approfondire cosa vuol dire e cosa comporta...

# Utenti



Esistono principalmente 2 tipi di utenti:

**Amministratore:** può modificare ogni cosa del sistema. Nel caso di linux questo è l'utente "root".

**Utente normale:** ha accesso limitato al sistema e può operare sotto alcune regole che verranno spiegate di seguito.

# Utenti

Informazioni sugli **utenti** in **/etc/passwd**.

Le **password** in **/etc/shadow**.

I file personali degli utenti si trovano in  
**/home/<Nome Utente>**

# Gruppi



Se abbiamo diversi utenti accomunati da un comportamento possiamo associare quel comportamento ad un Gruppo e mettere gli utenti in quel gruppo.

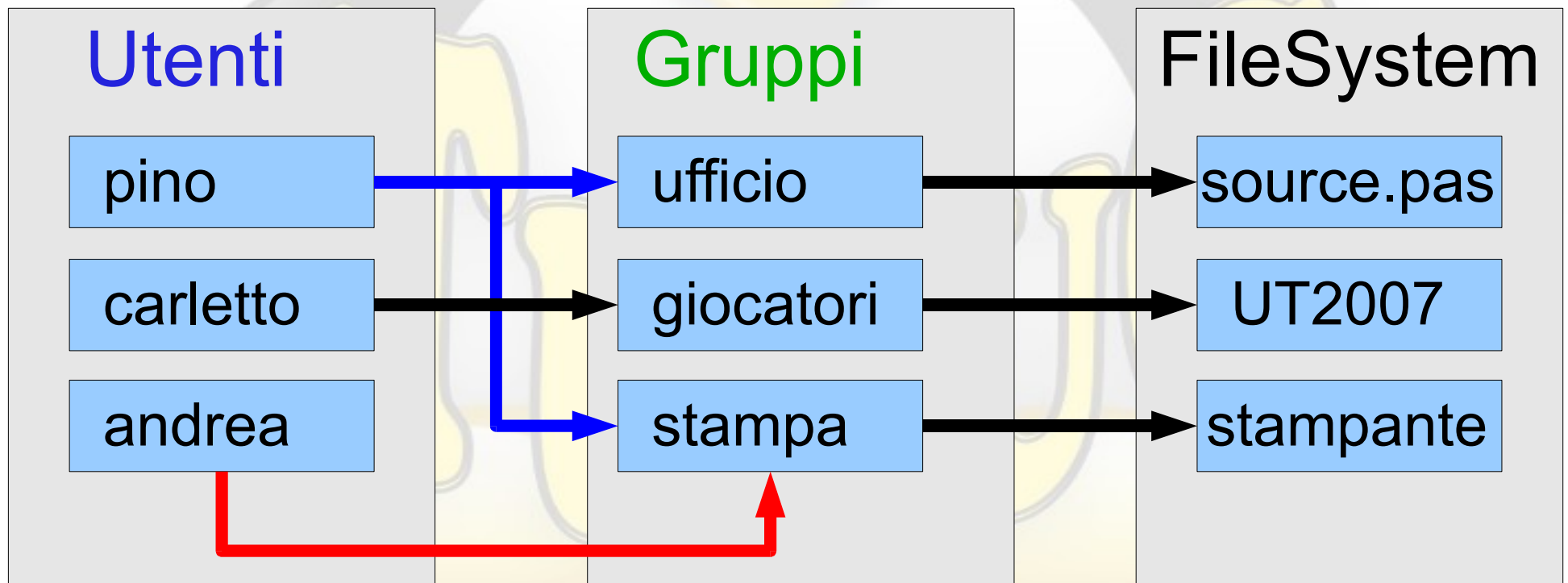
# Gruppi

Il file **/etc/group** permette di **raccogliere le notizie sui gruppi** e in particolare di stabilire la possibile appartenenza da parte di un utente a più gruppi.

Più Utenti possono far parte dello stesso gruppo.  
Un Utente può far parte di più gruppi.

# Esempio

- Nel caso in esempio
- Pino è un Programmatore di una nota società
- Carletto è il figlio del capo
- Andrea è il segretario della società



# Permessi

Un sistema multiutente ha la necessità di proteggere i file per garantire la sicurezza delle attività degli utenti come quella del sistema stesso.

Questo è semplice da realizzare su GNU/Linux visto che su questo tipo di sistema ogni cosa è un file, con annesse le informazioni necessarie per la gestione.

# Esempio

```
$ ls -l
-rw-r--r-- 2 pino ufficio 0 Nov 23 01:54 sorgente
lrwxrwxrwx 1 carletto giocatori 4 Nov 23 01:55 link -> UT07
drwxr-xr-x 2 utente gruppo 4096 Nov 23 01:57 Directory
```

**Il comando ls** con l'opzione **-l** mostra i file e le directory con associati i vari attributi.

**I primi 10 caratteri in rosso** rappresentano proprio i permessi di cui ora andremo a parlare...

# Tipo di file

Il primo carattere dei permessi rappresenta proprio il tipo di file che stiamo visionando, esso può variare come segue:

- **-** un file comune
- **d** una directory
- **c** un character special file
- **p** un named pipe
- **s** un file socket
- **l** un link simbolico

# I Permessi

Le rimanenti 9 cifre rappresentano i permessi.

- **le prime 3** dell'utente proprietario del file.
- **le seconde 3** del gruppo a cui appartiene il file.
- **le restati 3** degli altri (gli utenti che non sono proprietari e che non appartengono al gruppo a cui appartiene il file).

Questo permette un ulteriore livello di sicurezza all'interno del sistema.

# Perché divisi in gruppi di 3?

I tre caratteri rappresentano i permessi fondamentali che si possono avere su un file.

Questi permessi possono essere indistintamente assegnati all'utente al gruppo o agli altri

**r w x**

- **r** sta per read (lettura)
- **w** sta per write (scrittura)
- **x** sta per execute (esecuzione)

# Attenzione al tipo di file



## File normale

- **r** permette di leggerne il contenuto
- **w** permette di modificarne il contenuto
- **x** permette di eseguire il file, ammesso che si tratti di un eseguibile binario o di uno script

# Attenzione al tipo di file

## Directory

- **r** permette di leggere il contenuto della directory e perciò conoscere l'elenco dei file in esso contenuti.
- **w** permette di modificarne il contenuto e quindi creare eliminare e rinominare i file.
- **x** permette di attraversare la directory.

# Rappresentazione dei permessi

Abbiamo visto che i permessi sono espressi attraverso le 3 **lettere r w x** esiste comunque un metodo per esprimere tali permessi come **Numeri** che vanno a 0 a 7.

$$x = 1$$

$$w = 2$$

$$r = 4$$

le combinazioni dei permessi possono essere create tramite la somma dei rispettivi numeri.

# Esempio

Attenzione: sono sempre 9 simboli, un numero ne rappresenta solo 3.

**x = 1**

**w = 2**

**r = 4**

NUMERICO

LETTERALE

100

-/--x/---/---

200

-/--w-/---/---

740

-/rwx/r--/---

755

-/rwx/r-x/r-x

# Permessi particolari

Ci sono 2 tipi di permessi aggiuntivi di cui daremo solo un accenno.

***SGID***, attiva il numero del gruppo (GID) durante l'esecuzione;

***SUID***, attiva il numero dell'utente (UID) durante l'esecuzione.

# Riassunto



- Abbiamo visto Utenti, Gruppi e permessi.
- Abbiamo capito che sono associati ai file.
- Abbiamo visto come i permessi possono essere rappresentati e il loro significato.

# In pratica

Come posso creare un nuovo utente?

**adduser** : il comando di riferimento.

*(“man adduser” per maggiori informazioni).*

```
adduser pino
```

aggiungiamo un nuovo utente di nome pino

# In pratica

Come si cambia proprietario e gruppo di un file?

**chown** : il comando di riferimento.

*(“man chown” per maggiori informazioni).*

```
chown pino sorgente.pas
```

Pino diventa il proprietario del file “sorgente.pas”

```
chown pino.ufficio sorgente.pas
```

Pino diventa il proprietario del file “sorgente.pas”

Ufficio diventa il gruppo del file “sorgente.pas”

# In pratica

Come posso cambiare i permessi di un file?

**chmod** : il comando di riferimento.

*(“man chmod” per maggiori informazioni).*

```
chmod 700 sorgente.pas
```

L'utente “pino” diventa l'unico oltre root a poter lavorare sul file

```
chmod 740 sorgente.pas
```

Tutti gli utenti appartenenti al gruppo ufficio possono ora accedere in lettura la file “sorgente.pas”

# In pratica

Come posso cambiare la password di un utente?

**passwd** : il comando di riferimento.

*(“man passwd” per maggiori informazioni).*

```
passwd pino
```

cambiamo la password dell'utente pino