

# **Organizzazione di Sistemi Operativi e Reti**

**Corso di Laurea Triennale in Ingegneria Informatica**

**Sara Lioba Volpi** ([sara.volpi@iet.unipi.it](mailto:sara.volpi@iet.unipi.it))

# **Seconda Esercitazione**

- Editor
  - vi,
  - emacs,
- Utenti e gruppi,
- Permessi,
- Gestione utenti (prima parte).

**Editor di testo**

**Esistono vari editor di testo:**

- **vi,**
- **emacs,**
- **...**

- **Per creare o aprire un file già esistente:**
  - `vi nomeFile`
  
- **Esistono due modalità di funzionamento per vi:**
  - 1.modalità comandi:** permette di inserire comandi e scegliere quale azione compiere;
  - 2.modalità editing:** permette di inserire e cancellare testo (come se fosse un editor di testo quale blocco note).

# Editor vi: comandi (1 di 2)



## Elenco di alcuni comandi dell'editor vi

<code>esc</code>	<b>Passa in modalità comandi.</b>
<code>i</code>	<b>Passa in modalità inserimento nella posizione corrente.</b>
<code>o</code>	<b>Apri in inserimento una nuova linea sotto la posizione corrente.</b>
<code>x</code>	<b>Cancella un carattere.</b>
<code>u</code>	<b>Undo.</b>
<code>r?</code>	<b>Sostituisce con ? il carattere su cui si trova il cursore.</b>
<code>dd</code>	<b>Taglia la riga corrente.</b>
<code>ndd</code>	<b>Taglia n righe.</b>
<code>yy</code>	<b>Copia una riga.</b>
<code>nyy</code>	<b>Copia n righe.</b>

# Editor vi: comandi (2 di 2)



## Elenco di alcuni comandi dell'editor vi

<b>p</b>	<b>Incolla la selezione nella riga sotto il cursore.</b>
<b>/word</b>	<b>Ricerca nel testo la parola word</b>
<b>n</b>	<b>Si posiziona sull'occorrenza successiva (nella ricerca).</b>
<b>N</b>	<b>Si posiziona sull'occorrenza precedente (nella ricerca).</b>
<b>:q</b>	<b>Esce (solo se non si sono fatte modifiche).</b>
<b>:w</b>	<b>Salva.</b>
<b>:wq</b>	<b>Salva ed esce.</b>
<b>:q!</b>	<b>Esce senza salvare.</b>



# Interfaccia grafica



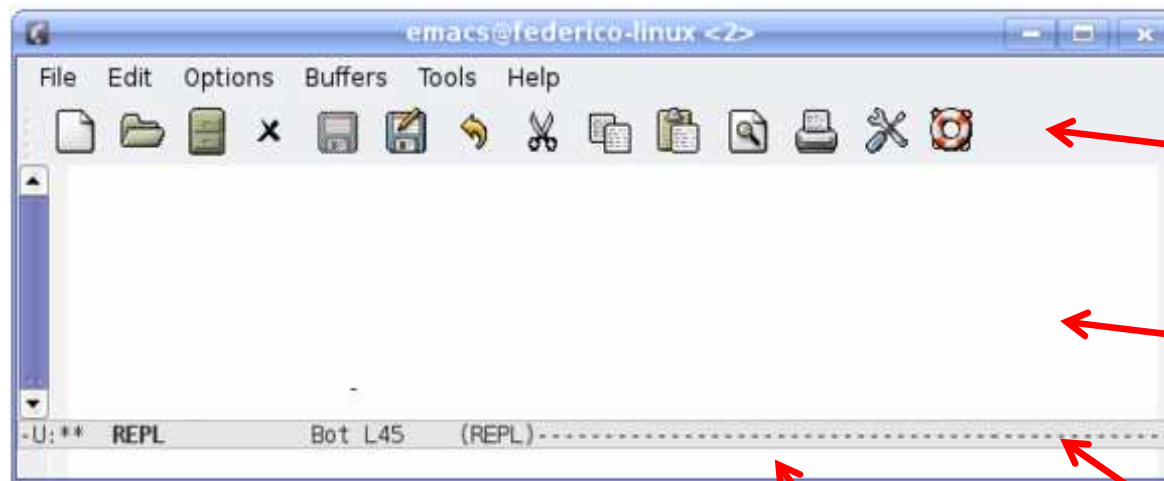
1. **Creare nella propria home un file con nome:**
  - `.xinitrc`
2. **Nel file `.xinitrc` scrivere:**
  - `wmaker`
3. **Chiudere e salvare il file `.xinitrc`**
4. **Digitare il comando**
  - `startx`

# Editor emacs



- Per creare o aprire un file già esistente:
  - `emacs nomeFile`
- Usare emacs da interfaccia grafica (consigliato).
- Ambiente grafico, si possono usare i menù a tendina.
- Lavora su un buffer e non sul disco.
- Al salvataggio il file viene aggiornato sul disco.
- I comandi sono espressi come combinazioni di tasti.
  - Abbreviazioni usate di seguito:
    - ⇒ **C** = CTRL,
    - ⇒ **M** = Alt.

# Editor emacs: composizione interfaccia



Barra degli strumenti

Buffer

Barra di stato

Mini-buffer

# Editor emacs: comandi (1 di 4)



## Elenco di alcuni comandi dell'editor emacs

### Comandi di base

C+h + es. a	<b>Help.</b> a(apropos) : apropos cerca nella documentazione in linea di Emacs l'espressione regolare digitata.
C+x, u	<b>Annulla operazione precedente.</b>
C+x, C+g	<b>Esce dal comando corrente.</b>
C+x, C+c	<b>Esce e chiede di salvare</b>
C+x, C+s	<b>Salva il file.</b>
C+x, C+w	<b>Salva con nome.</b>

### File

C+x, C+f	<b>Apri un file nel buffer corrente.</b>
C+x, C+v	<b>Cerca e legge un file alternativo.</b>
C+x, i	<b>Inserisce un file nella posizione corrente.</b>

# Editor emacs: comandi (2 di 4)



## Elenco di alcuni comandi dell'editor emacs

### Movimenti del cursore

<code>C+p</code>	Linea precedente.
<code>C+n</code>	Linea successiva.
<code>C+a</code>	Inizio linea.
<code>C+e</code>	Fine linea.
<code>C+v</code>	Sfoggia avanti.
<code>M+v</code>	Sfoggia indietro.
<code>M+f</code>	Avanti di una parola.
<code>M+b</code>	Indietro di una parola.
<code>M+&lt;</code>	Inizio buffer.
<code>M+&gt;</code>	Fine buffer.
<code>M+x goto-line &lt;invio&gt; n</code>	Muove il cursore alla linea n.

# Editor emacs: comandi (3 di 4)



## Elenco di alcuni comandi dell'editor emacs

### Ricerca

C+s	Ricerca in avanti.
C+r	Ricerca al rovescio.
M-x replace-string <invio> stringa <invio> nuovastringa <invio>	Cerca e rimpiazza una stringa.

### Cancellazione testo

C+d	Cancella un carattere.
M+d	Cancella dal cursore fino a fine parola.
C+k	Cancella dal cursore fino a fine linea.

### Trasformazione testo

M+u	Maiuscolo dal carattere corrente fino a fine parola.
M+l	Minuscolo dal carattere corrente fino a fine parola.

# Editor emacs: comandi (4 di 4)



## Elenco di alcuni comandi dell'editor emacs

### Copia, Taglia, Incolla

<code>C+&lt;space&gt;</code>	Per marcare una posizione.
<code>M+w</code>	Copia la parte evidenziata o dalla marcatura al cursore.
<code>C+w</code>	Taglia/cancella la parte evidenziata o dalla marcatura al cursore.
<code>C+y</code>	Incolla.

### Dividere l'ambiente in più finestre

<code>C+x, n</code>	Divide l'ambiente in 2 finestre ( $n = 2$ o $3$ ).
<code>C+x, 1</code>	Cancella tutte le finestre ad eccezione di quella corrente.
<code>C+x, o</code>	Si sposta sull'altra finestra.

# Editor emacs: programmazione (1 di 2)



- emacs supporta le modalità di un'ampia gamma di linguaggi di programmazione come C e C++.

## Elenco di alcuni comandi dell'editor emacs

### Supporto alla programmazione

M-x [language]-mode	Passa ad una determinata modalità di linguaggio (per il C: M-x c-mode).
M-x [language]-indent-region	Per indentare con il tasto TAB.
M-x font-lock-mode	Attiva la colorazione della sintassi.



# Editor emacs: programmazione (2 di 2)



## Elenco di alcuni comandi dell'editor emacs

### Supporto alla programmazione: compilazione

`M+x compile` Chiama il compilatore al quale bisogna passare i comandi per compilare, es.: `gcc -o main main.c`

- Chiama compilatori esterni.
- Mostra il risultato in un buffer.
  - **Compilation finished:** se la compilazione va a buon fine.
  - **Compilation exited abnormally:** se la compilazione non va a buon fine. Nel buffer dedicato ai messaggi del compilatore viene visualizzato l'elenco dei messaggi di errore con indicazioni sulla linea del sorgente.

# **Utenti e gruppi**

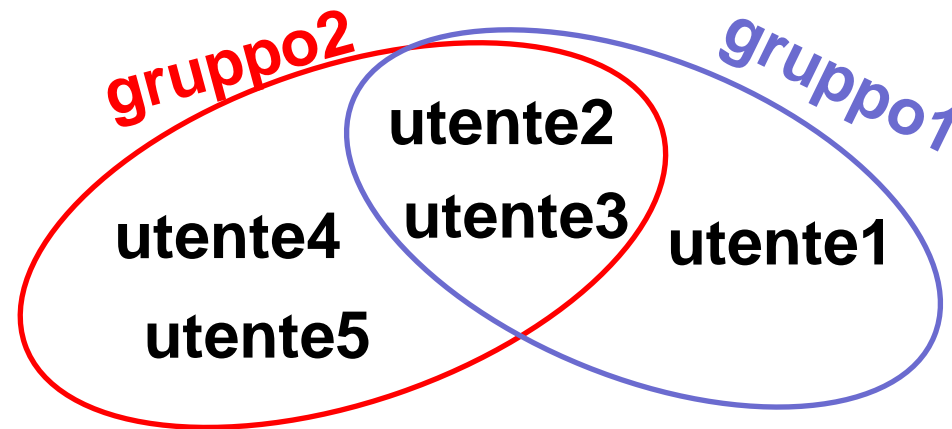
- **Ogni utente è identificato da:**
  - **un username,**
  - **un UID (user ID) numerico.**
- **Username e UID sono pubblici.**
- **Ogni utente può appartenere ad uno o più gruppi.**

# Gruppi (1 di 2)



- Ogni gruppo è identificato da:
  - un **groupname**,
  - un **GUID** (group ID) numerico.
- L'utente **root** appartiene di default al gruppo **wheel**.

# Gruppi (2 di 2)



- **utente1** appartiene al gruppo **gruppo1**,
- **utente2** appartiene ai gruppi **gruppo1** e **gruppo2**,
- **utente3** appartiene ai gruppi **gruppo1** e **gruppo2**,
- **utente4** appartiene al gruppo **gruppo2**,
- **utente5** appartiene al gruppo **gruppo2**.

**Permessi**

# Gestione dei permessi



I permessi possono essere di:

- lettura,
- scrittura,
- esecuzione.

I permessi da applicare vengono differenziati in base alla natura dell'utente che vuole “operare” sul file, directory, ... :

- utente proprietario del file: **owner**,
- gruppo proprietario: **group owner**,
- gli altri utenti: **others**.

# Gestione dei permessi: file



Attributo	Significato
Lettura	Permette di leggere il contenuto del file
Scrittura	Permette di modificare il contenuto del file
Esecuzione	Permette di eseguire un file (binario o script)

- **Il permesso di scrittura**
  - non incide sulla possibilità di cancellare un file,
  - bisogna considerare i permessi della directory.



# Gestione dei permessi: directory



Attributo	Significato
Lettura	Permette di leggere il contenuto (l'elenco dei file)
Scrittura	Permette di modificare il contenuto della cartella
Esecuzione	Permette di attraversare una cartella

- **Negare l'accesso in lettura impedisce l'esecuzione con successo del comando `ls`.**
- **Negare l'accesso in scrittura impedisce di creare, cancellare e rinominare file.**
- **Negare l'accesso in esecuzione impedisce di fare un `cd` sulla directory.**

# Gestione dei permessi: permessi speciali



Attributo	Significato
SUID	Durante l'esecuzione il comando acquisisce i privilegi del proprietario del file stesso
SGID	Durante l'esecuzione il comando acquisisce i privilegi del gruppo a cui il file appartiene
Sticky	Durante l'esecuzione di un comando salva l'immagine nella memoria virtuale

- **Attributo sticky su cartelle**
  - non consente ad un utente di cancellare o rinominare un file a meno che non ne sia il proprietario (anche se ha il permesso in scrittura sulla directory).

# Rappresentazione simbolica dei permessi (1 di 2)



I permessi di un file, directory,... possono essere visualizzati con il comando `ls -l`:

`d``rwxr-xr-x` ... prova

`t rwx rwx rwx`

others  
group  
owner

- `t` : tipo di file, `d` : directory, `-` : file, ...
- `r`: permesso di lettura,
- `w`: permesso di scrittura,
- `x`: permesso di esecuzione,
- `-` indica l'assenza del permesso corrispondente.

# Rappresentazione simbolica dei permessi (2)



- Il permesso di esecuzione in owner può essere anche:
  - **s** : SUID ed esecuzione permessi,
  - **S** : solo SUID.
- Il permesso di esecuzione in group owner può essere anche:
  - **s** : SGID ed esecuzione permessi,
  - **S** : solo SGID.
- Il permesso di esecuzione in others può essere anche:
  - **t** : sticky ed esecuzione permessi,
  - **T**: solo sticky.

# Rappresentazione ottale dei permessi



La rappresentazione ottale dei permessi consiste in un numero di 4 cifre in base 8.

- lettura: 4,
- scrittura: 2,
- esecuzione: 1,
  
- SUID: 4,
- SGID: 2,
- sticky: 1.



# Comando chmod (1 di 2)

chmod permette di modificare i permessi.

La sintassi generale di chmod è la seguente:

```
chmod [who][how][which]
```

- **who: a chi modificare i permessi: owner (u), group (g), others (o), all (a).**
- **how:**
  - '+' aggiunge un permesso,
  - '-' toglie un permesso,
  - '=' assegna un permesso.

# Comando chmod (2 di 2)



## Rappresentazione simbolica:

```
chmod a+rwx directory1
```

## Rappresentazione ottale:

```
chmod 0741 file
```

**opzione -R** : modifica ricorsivamente i permessi delle directory indicate e del loro contenuto.

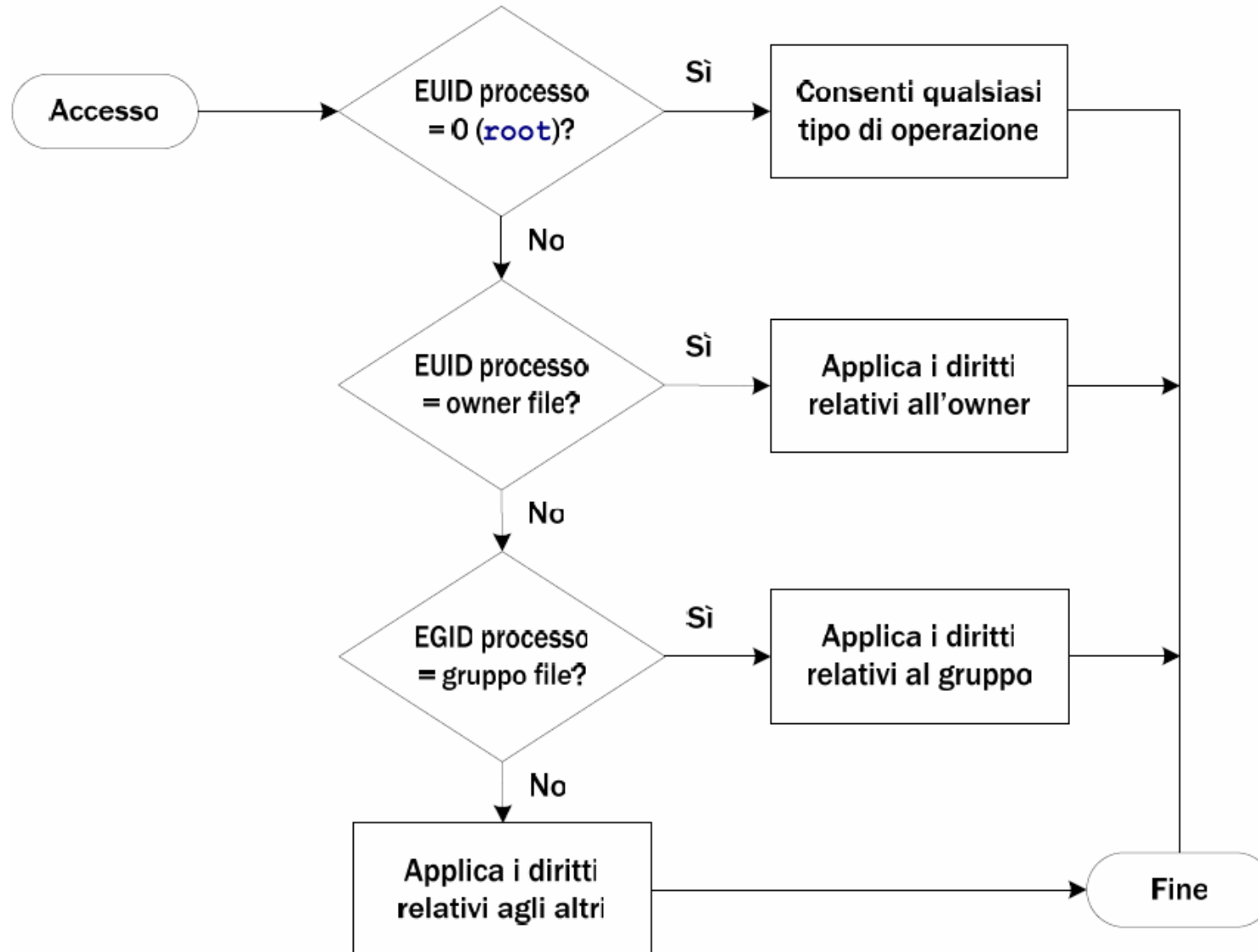
# Real e Effective User ID



Suffisso	Gruppo	Descrizione
UID	Real	Utente che ha mandato in esecuzione il programma
GID	Real	Gruppo principale dell'utente che ha mandato in esecuzione il programma
EUID	Effective	Utente usato nel controllo degli accessi (privilegi effettivi utente)
EGID	Effective	Gruppo usato nel controllo degli accessi (privilegi effettivi gruppo)



# Applicazione dei permessi



# Proprietari dei file



- Quando viene creato un file/directory/...,
  - l'owner è rappresentato dall' EUID,
  - il group owner è rappresentato dall' EGUID.

# **Gestione utenti (prima parte)**

# Comandi passwd, whoami, id, groups



- `passwd` : permette di cambiare la password.
- `whoami` : visualizza l'EUID.
- `id` : visualizza UID e *GID dell'utente selezionato o di quello corrente*.
- `groups` : visualizza i gruppi ai quali l'utente appartiene.

# Creazione e rimozione degli utenti



- `adduser` : **crea un nuovo utente** “nomeUtente”.
- `adduser nomeUtente`
  
- `rmuser` : **rimuove l'utente** “nomeUtente”.
- `rmuser nomeUtente`

# **Interazione fra gli utenti**

- Il comando `su` (switch user o substitute user) viene tipicamente usato da:
  - utenti non privilegiati per avviare una shell come root
    - ⇒ per poterlo eseguire l'utente deve appartenere anche al gruppo `wheel`,
  - root per avviare una shell come utente ordinario.
- Richiede autenticazione
  - password dell'utente che si vuole diventare.

# Comandi `chown`, `chgrp`



Il comando `chown` permette di modificare il proprietario e/o il gruppo assegnato ad un file o directory.

Il comando `chgrp` permette di modificare il gruppo assegnato ad un file o directory.



# Esercizio

# Esercizio (1 di 3)



Supporre di essere l'utente utente.

1. Lavorare nella propria cartella home.
2. Creare una cartella con nome visibile e al suo interno una cartella con nome segreta.
3. Scrivere la stringa vero nel file notizia.txt all'interno di visibile.
4. Copiare notizia.txt all'interno di segreta assegnandole il nome cronaca.txt .
5.
  1. Togliere il permesso di esecuzione (proprietario) a visibile usando la rappresentazione simbolica.
  2. Rimettere il diritto di esecuzione (proprietario) a visibile usando la rappresentazione simbolica.
  3. Togliere di nuovo il diritto di esecuzione (proprietario) usando la rappresentazione ottale e lasciando invariati i diritti del gruppo e di others.

# Esercizio (2 di 3)



6. A questo punto:
  1. si riesce a vedere il contenuto di visibile?
  2. si riesce a vedere il file notizia.txt dentro visibile?
  3. si riesce a vedere il contenuto di segreta?
  4. si riesce a vedere il file cronaca.txt dentro segreta?
7. Creare un utente utente2 (login con root).
  1. utente riesce a vedere il contenuto della home di utente2?
  2. eventualmente cambiare i diritti in modo che utente non riesca a vedere il contenuto della home di utente2.
8. Controllare a quali gruppi appartiene l'utente root (login con utente).

# Esercizio (3 di 3)



- 9 . Creare un utente utente3.
10. Creare la cartella temp nella home di utente3.
11. Quali sono l'utente proprietario e il gruppo proprietario di temp?
12. Cambiare utente proprietario e gruppo proprietario di temp con utente3 e verificare che si avvenuto l'aggiornamento di tali campi.
13. Rimettere wheel come gruppo proprietario di temp.
14. Rimuovere utente2 ed utente3.

# Soluzione (1 di 3)



1. `cd`
2. `mkdir -p visibile/segreta`
3. `echo 'vero' > visibile/notizia.txt`
4. `cp visibile/notizia.txt  
visibile/segreta/cronaca.txt`
5.
  1. `chmod u-x visibile`
  2. `chmod u+x visibile`
  3. **Supponendo che i diritti del gruppo e di others siano solo di lettura ed esecuzione ma non di scrittura:**  
`chmod 655 visibile`

# Soluzione (2 di 3)



## 6. Risposte alle domande

1. Si riesce a vedere il contenuto di visibile (al proprietario è consentito l'accesso in lettura).
2. Non si può vedere il file notizia.txt dentro visibile (non si può attraversare visibile).
3. Non si può vedere il contenuto di segreta .
4. Non si può vedere il notizia.txt dentro segreta .

## 7. `adduser utente2`

1. supponendo che la maschera di default sia `rw-r-xr-x`, utente può vedere il contenuto della home di `utente2` .
2. supponendo che la home di `utente2` abbia un gruppo diverso da utente:

```
chmod o-r ~
```

# Soluzione (3 di 3)



8. `groups root`
9. `adduser utente3 (login root)`
10. `mkdir /home/utente3/temp (root)`
11. `ls -l /home/utente3/temp` (**l'utente proprietario è root e il gruppo proprietario è wheel**)
- 12.** `chown utente3:utente3 /home/utente3/temp`  
`ls -l`
13. `chgrp wheel /home/utente3/temp`
14. `rmuser utente2`  
`rmuser utente3`

# Comando umask



Quando viene creato un file:

- il file appartiene automaticamente all'utente che lo crea ed al gruppo principale dell'utente stesso,
- i permessi gli vengono attribuiti in base alla maschera dei permessi ( `umask` ) :
  - tale maschera rappresenta i permessi che **NON** vengono attribuiti,
  - il suo valore, in generale, è tale da non attribuire il permesso di scrittura né al gruppo proprietario, né agli altri utenti,
  - il valore di questa maschera può essere modificato (temporaneamente) con il comando: `umask`.