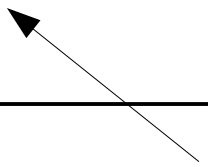
Accesso a Linux

E Mumolo - DIA

Login

- Esempio:

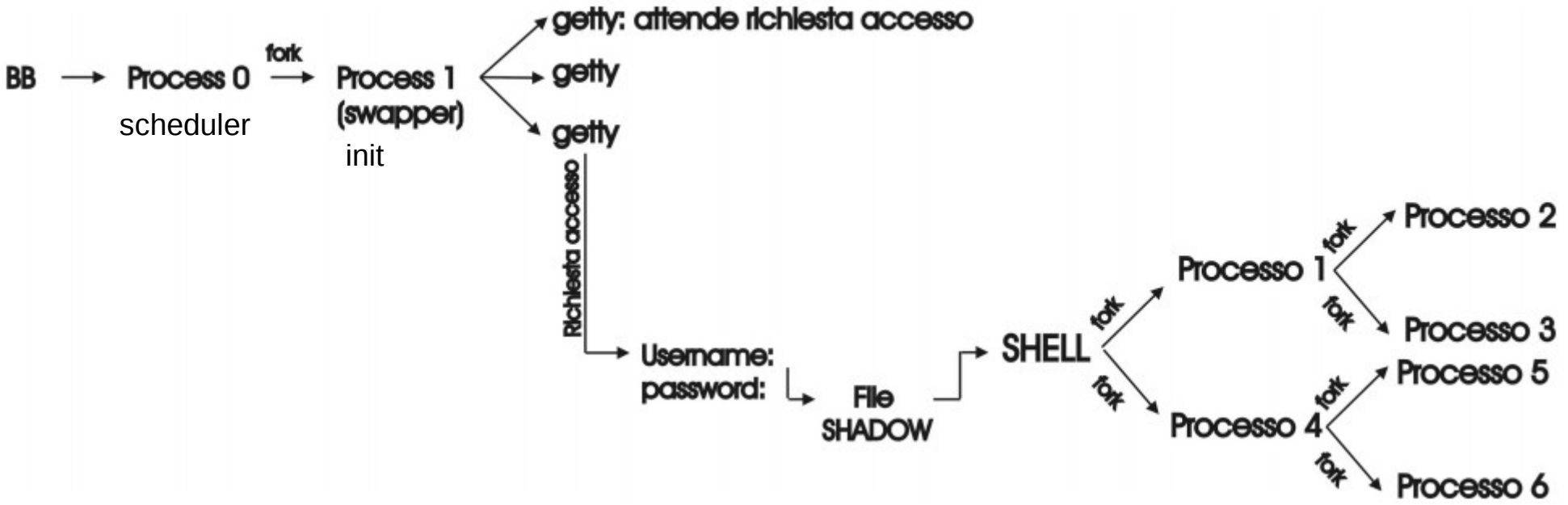
```
Ubuntu 10.04l4 LTS tt2
login: utente
Password:
$
```



PROMPT

- Verifica accesso:
  - File /etc/passwd
  - File /etc/shadow

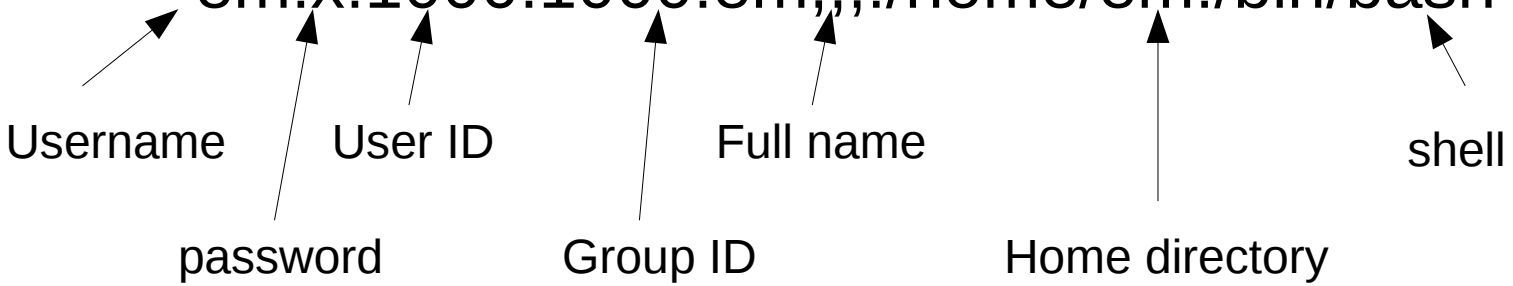
Login


BB
Process 0
scheduler
fork
Process 1
(swapper)
init
getty: attende richiesta accesso
getty
getty
Richiesta accesso
Username:
password:
File
SHADOW
SHELL
fork
fork
Processo 1
Processo 4
fork
fork
fork
fork
Processo 2
Processo 3
Processo 5
Processo 6

File password/shadow

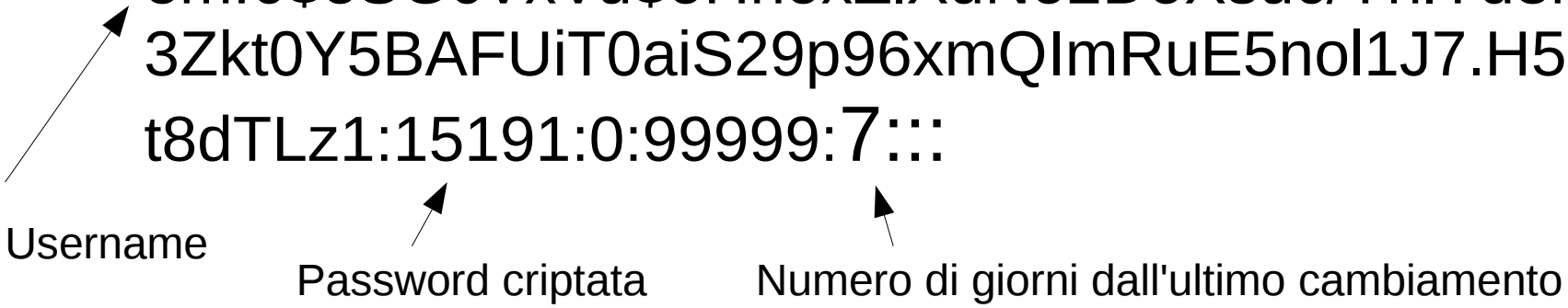
- La verifica dell'accesso è realizzata mediante i file password e shadow
- Esempio file password:

em:x:1000:1000:em,,,:/home/em:/bin/bash



- Le password sono gestite dal file shadow. Esempio di file shadow:

em:6$eSGcVxVu$oHhoxZlXuNo2DoXsuc/YhlYd3n5AT7Gh3Zkt0Y5BAFUiT0aiS29p96xmQImRuE5nol1J7.H5zMRH4uLt8dTLz1:15191:0:99999:7:::



Identificatori Utente

- Ottenibili con il comando id:
  - Es: **`$id`**

    `uid=1000(em) gid=1000(em)...`
- User Identifier (`id -u username`)
  - Es: **`$id -u em`**

    `1000`
- Group Identifier (`id -g username`)
  - Es: **`$id -g em`**

    `1000`
- Gruppi di utenti (`id -G username` fornisce i gruppi ai quali username appartiene)
  - Es: **`$id -G`**

    `1000 4 7 20 24 29 46 105 119 122`

    `$id`

    `uid=1000(em) gid=1000(em)` groups = 4(adm),7(lp),20(dialout), 24(cdrom), 29(audio), 46(plugdev), 105(lpadmin),119(admin), 122(sambashare), 1000(em)

Esecuzione di processi in Linux

- Se installati nel sistema, basta il nome
- Se disponibili come file contenenti codice eseguibile, <path>/nomefile
- Nella directory corrente: ./nomefile
- Comando per vedere I processi in esecuzione: **`$ps`**

`Esempio:`

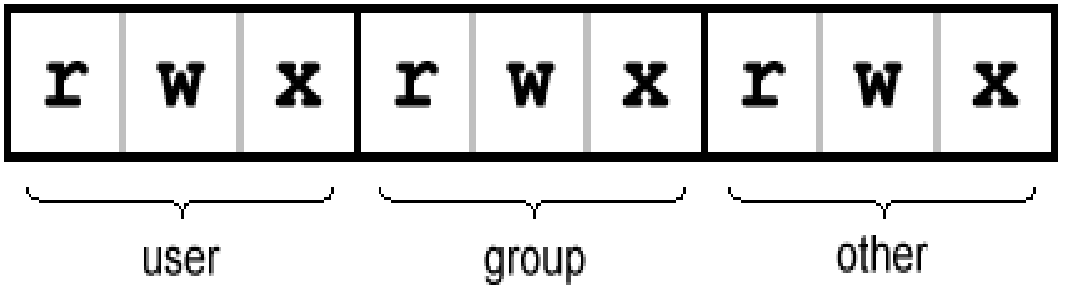
```
$ps
  PID TTY          TIME CMD
18027 pts/0    00:00:00 bash
18366 pts/0    00:00:00 ps
```

Identificatori dei processi

- Identificatori fondamentali
  - Process ID (PID)
  - Parent Process ID (PPID) del processo che l'ha generato
  - Process Group ID (PGID) del gruppo di processi al quale appartiene
- Altri identicatori dei processi:
  - REAL USER ID: il UID dell'utente che ha richiesto accesso
  - REAL GROUP ID: il GID dell'utente che ha richiesto accesso
  - EFFECTIVE USER ID: normalmente uguale al REAL USER ID, usato per la protezione dei file
  - EFFECTIVE GROUP ID: normalmente uguale al REAL GROUP ID, usato per la protezione dei file

Protezione dei file

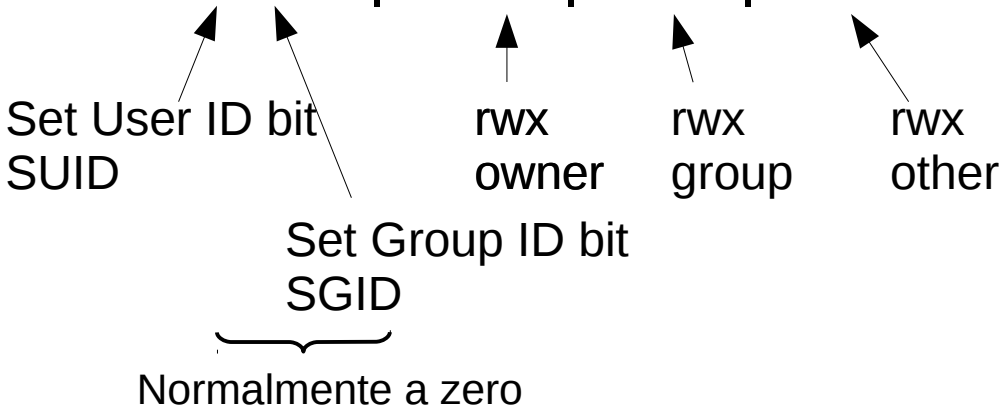
- Ogni file, nel campo file mode dell'Inode, ha i seguenti bit:



- Esempio: 111000000, 110110110 etc. dove:
  - il proprietario (indicato con U, user) e' l'unico utente che e' il proprietario (owner) del file, quello il cui User ID coincide con quello del file;
  - il gruppo (indicato con G, group) e' l'insieme di tutti gli utenti che appartengono al gruppo del proprietario, cioè hanno lo stesso Group ID del file;
  - gli altri (indicato con O, other) sono tutti gli altri utenti.

Protezione dei file

- File mode: b b b | b b b | b b b| b b b



- Quando SUID viene settato nei file che contengono codice eseguibile, l'EUID viene cambiato al proprietario del file per la durata del programma
- Quando SGID viene settato nei file che contengono codice eseguibile, l'EUID viene cambiato al proprietario del file per la durata del programma

Algoritmo di protezione dei file

- if(Effective_UID del processo == Owner_UID del file)
  usa i permessi di Owner;
  else
  if(Effective_GID del processo == Owner_GID del file)
  usa i permessi di Group;
  else
  usa i permessi di Altri;

Il Set-user-id (SUID) del file mode - esempio

- Assumiamo di avere
  - un file sensibile, FileCc,
  - un programma, ModCc, che modifica il file sensibile
- FileCc e ModCc appartengono all'utente ManagerCc
- Per dare a tutti gli altri utenti la possibilità di modifica dovrei dare a ModCc le protezioni `rwx---r-x` e al FileCc `rw----rw-`
- A questo punto però tutti potrebbero fare qualsiasi cosa col file, copiarlo o cancellarlo
- Mettendo il bit SUID del file a `1,` quando un utente esegue ModCc, l'EUID del suo processo ModCc diventa quello del proprietario di FileCc e ModCc, cioè ManagerCc
- Il file FileCc avrà allora le protezioni rwx------
- Il file FileCc può essere modificato SOLO dal programma ModCc
- Per tutti gli altri utenti che non siano il proprietario sarà un file nascosto

Qualche esempio di protezione file

Chiamiamo RUID, EUID “Real User ID” ed “Effective User ID”.
Si supponga che una directory di Linux contenga i seguenti file, dove l'unico file eseguibile è *All,* che è un programma per leggere il contenuto di un file:

Filename	Owner	Group	SetUID	Owner	Group	Other
a10.txt	15	99	s	rw-	r--	---
b10.txt	15	99	-	—--	rw-	rw-
c12.txt	75	75	s	rw-	rw-	---
c14.txt	25	25	—	rw-	r--	r--
dati.txt	25	25	s	—x	r-x	---
All	75	75	s	--x	--x	--x

Si supponga inoltre che l'utente 15 sia nel gruppo 99 e che l'utente 25 sia nel gruppo 25.
Rispondere alle seguenti domande:

a. Quali file può leggere un processo con RUID=15, EUID=15, EGID15?
b. Quali file può scrivere un processo con RUID=15, EUID=25, EGID=75?
c. Quali file può leggere un utente con UID=100 GID=100 che esegue il programma *All*?

Qualche esempio di protezione file

Chiamiamo RUID, EUID “Real User ID” ed “Effective User ID”.
Si supponga che una directory di Linux contenga i seguenti file, dove l'unico file eseguibile è *All,* che è un programma per leggere il contenuto di un file:

Filename	Owner	Group	SetUID	Owner	Group	Other
a10.txt	15	99	s	rw-	r--	---
b10.txt	15	99	-	---	rw-	rw-
c12.txt	75	75	s	rw-	rw-	---
c14.txt	25	25	—	rw-	r--	r--
dati.txt	25	25	s	--x	r-x	---
All	75	75	s	--x	--x	--x

Si supponga inoltre che l'utente 15 sia nel gruppo 99 e che l'utente 25 sia nel gruppo 25.
Rispondere alle seguenti domande:

a. Quali file può leggere un processo con RUID=15, EUID=15, EGID15?
b. Quali file può scrivere un processo con RUID=15, EUID=25, EGID=75?
c. Quali file può leggere un utente con UID=100 GID=100 che esegue il programma *All*?

---

a. a10.txt, c12.txt, c14.txt
b. a10.txt, b10.txt, c12.txt, c14.txt
c. a10.txt, b10.txt, c12.txt

Il file `lista` contiene il codice eseguibile di un programma che lista il contenutodi file testuali (l'uso del comando è quindi del tipo: `$lista <file>`).

Il file `lista` è dell'utente Giulio, con UID=1500 e GID=2000, ed ha le protezioni 550. Per vedere le caratteristiche del file `lista`, l'utente Giulio effettua il comando: `$ls –l > dati`

Facendo `$ls –l dati` Giulio vede che il file `dati` è ovviamente di Giulio ed ha le seguenti protezioni: 604.

a) Può Giulio vedere il contenuto del file dati con il comando `$lista dati` ?

b) Cosa succede se Giorgio del gruppo 2000 scrive il comando `$lista dati`?

c) Cosa succede se Giorgio del gruppo 2000 scrive il comando `$lista dati` dopo che Giulio setta il bit SetUID del file `dati`?

Il file lista contiene il codice eseguibile di un programma che lista il contenutodi file testuali (l'uso del comando è quindi del tipo: $lista <file>).

Il file lista è dell'utente Giulio, con UID=1500 e GID=2000, ed ha le protezioni 550. Per vedere le caratteristiche del file lista, l'utente Giulio effettua il comando: `$ls –l > dati`

Facendo `$ls –l` dati Giulio vede che il file dati è ovviamente di Giulio ed ha le seguenti protezioni: 604.

a) Può Giulio vedere il contenuto del file dati con il comando `$lista dati` ?

b) Cosa succede se Giorgio del gruppo 2000 scrive il comando `$lista dati`?

c) Cosa succede se Giorgio del gruppo 2000 scrive il comando `$lista dati` dopo che Giulio setta il bit SetUID del file dati?

File dati: 604 = 110 000 100 = rw- --- r-- appartiene a Giulio

File lista: 550 = 101 101 000 = r-x r-x --- appartiene a Giulio

Giulio: UID=1500, GID=2000

Giorgio: UID=?, GID=2000

a. si

b. può eseguire lista ma non può leggere dati

c. diventa il proprietario di dati e quindi lista può leggere dati

Si supponga che un sistema abbia accreditato l'utente Giuseppe, con `UID=110` e `GID=220`, e l'utente Mario, con `UID=105` e `GID=220.`

Nella *home directory* dell'utente Giulio, con `UID=100` e `GID=220` ci sono i file `data.txt` e `myprog` , entrambi di proprietà di Giuseppe, ed i file `data1.txt` e `data2.txt`, entrambi di proprietà di Mario.

Si supponga che gli 11 bit meno significativi del file mode (SUID|SGID|protezioni), dove SUID e SGID sono i bit SetUserID e SetGroupID, siano codificati in base 8 nel seguente modo:

```
myprog       4555
  data.txt    210
 data1.txt   1474
 data2.txt   2440
```

`myprog` è un programma eseguibile, creato da Giuseppe, che scrive determinate informazioni sul file fornito come argomento, e che viene eseguito da Giulio.

Su quali dei file `data.txt,data1.txt, data2.txt`, può scrivere il programma myprog?

```
        owner      group    prot
myprog  110        220      srw-rw-rw-
data.txt  110      220       -w---x---
data1.txt 105      220      or--rwxr--
data2.txt 105      220      gr--r-----
```

Giulio esegue $./myprog data.txt ,poi $./myprog data1.txt e infine $./myprog data2.txt

Il processo myprog ha real process uid=100 (Giulio) e real process gid=220. Visto che myprog ha il SUID settato, per data.txt che appartiene all'owner di myprog cambia l'effective uid a 110, quindi guarda I primi 3 bit e vede che può scriverci.

Per gli altri due file resta effective uid=100 e gid=220 quindi guarda I secondi 3 bit e vede che può scrivere solo su data1.txt

Quindi no si no.