16.1 Gestione del registro del sistema

«

Un sistema operativo complesso, quale può essere un sistema Unix, richiede l'annotazione di alcuni eventi importanti in un registro,

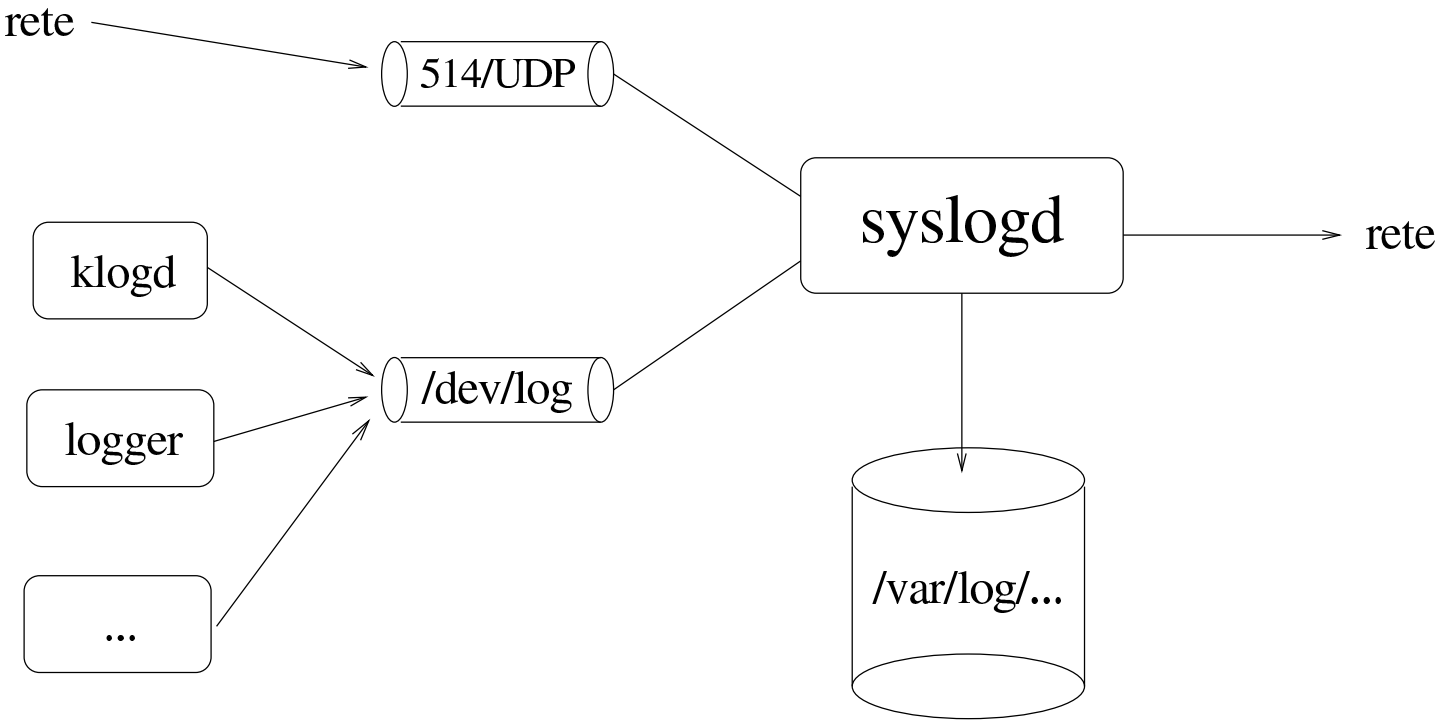
composto da uno o più file specifici. Il sistema che si occupa della compilazione di questo registro, può intervenire solo localmente, oppure può ricevere queste informazioni anche da elaboratori remoti, attraverso la rete; nello stesso modo, può rinviare la registrazione a un elaboratore remoto. Tuttavia, la questione riguardante la rete viene accennata, contando sulla conoscenza delle problematiche essenziali legate ai protocolli TCP/IP; al riguardo si vedano eventualmente il capitoli a partire da 32.

16.1.1 Registro del sistema

Il registro del sistema (*system log*, o anche *syslog*) è la procedura di registrazione degli eventi importanti all'interno di un cosiddetto file di *log*, ovvero un file delle registrazioni, o più semplicemente «registro». Questa procedura è gestita principalmente dal demone '**syslogd**', il quale viene configurato attraverso '`/etc/syslog.conf`'. Altri programmi o demoni possono aggiungere annotazioni al registro inviando messaggi a '**syslogd**'.

Anche se potrebbe sembrare che la conoscenza di questo metodo di registrazione sia uno strumento utile principalmente per chi ha già esperienza di sistemi Unix, la consultazione dei file delle registrazioni può essere di aiuto al principiante che si trova in difficoltà e non sa quale sia la causa del mancato funzionamento di qualcosa.

Figura 16.1. Schema di massima dei flussi di dati che coinvolgono il demone '**syslogd**'.



16.1.1.1 Utilizzo di «syslogd»

«

Il programma '**syslogd**' è il demone che si occupa delle annotazioni nel registro del sistema.[1]

```
syslogd [opzioni]
```

Di norma viene avviato durante la procedura di avvio del sistema. Utilizza un file di configurazione che di solito è '`/etc/syslog.conf`'. Questo file viene letto nel momento in cui '**syslogd**' si avvia e, per fare in modo che venga riletto (per esempio dopo una modifica), occorre inviare al processo di '**syslogd**' un segnale di aggancio ('**SIGHUP**').

```
kill -HUP pid_di_syslogd
```

Tabella 16.2. Alcune opzioni.

Opzione	Descrizione
`-f` ***file_di_configurazione***	Specifica un file di configurazione diverso da quello predefinito.
`-m` ***minuti***	Stabilisce l'intervallo espresso in minuti tra i messaggi di marcatura. Il valore predefinito è 20.
`-p` ***log_socket***	Specifica un socket di dominio Unix diverso da quello predefinito che è '`/dev/log`'.
`-a` ***log_socket***	Specifica un socket di dominio Internet addizionale, per la ricezione; quello predefinito è di norma 514/UDP.

16.1.1.2 File «/etc/syslog.conf»

Il file '`/etc/syslog.conf`' contiene la configurazione di '**`syslogd`**' che definisce in che modo devono essere gestiti i messaggi da registrare. Se si vogliono apportare modifiche a questo file è necessario fare in modo che venga riletto da '**`syslogd`**'. Per fare questo è possibile mandare a '**`syslogd`**' il segnale '**`SIGHUP`**':

```
kill -HUP pid_di_syslogd
```

Tuttavia, in certi casi, questo segnale può anche provocare la conclusione del funzionamento del programma. Se necessario si può riavviare semplicemente:

```
# syslogd [Invio]
```

La sintassi per l'utilizzo di questo file di configurazione è relativamente semplice. Le righe vuote e quelle che iniziano con il simbolo '**#**' sono ignorate. Le altre sono record composti da due campi: il primo definisce la selezione, il secondo l'azione.

Il campo che definisce la selezione, serve a indicare per quali eventi effettuare un'annotazione attraverso l'azione indicata nel secondo campo. Questo primo campo si divide in due sottocampi, uniti da un punto singolo ('**.**'), i quali si riferiscono ai servizi e alle priorità. I servizi sono rappresentati da parole chiave che individuano una possibile origine di messaggi, mentre le priorità sono altre parole chiave che identificano il livello di gravità dell'informazione.

Le parole chiave riferite ai servizi possono essere: '**auth**'; '**authpriv**'; '**cron**'; '**daemon**'; '**kern**'; '**lpr**'; '**mail**'; '**news**'; '**syslog**'; '**user**'; '**uucp**'; da '**local0**' a '**local7**'.

Volendo identificare tutti i servizi si può usare l'asterisco ('*****'), mentre per indicarne un gruppo se ne può inserire un elenco separato da virgole ('**,**').

Le parole chiave riferite alle priorità possono essere quelle seguenti, elencate in ordine di importanza crescente, per cui l'ultima è quella che rappresenta un evento più importante: '**debug**'; '**info**'; '**notice**'; '**warning**'; '**err**'; '**crit**'; '**alert**'; '**emerg**'.

In linea di massima, l'indicazione di una parola chiave che rappresenta una priorità implica l'inclusione dei messaggi che si riferiscono a quel livello, insieme a tutti quelli dei livelli superiori. Per indicare esclusivamente un livello di priorità, occorre fare precedere la parola chiave corrispondente dal simbolo '**=**'. Si possono indicare

assieme più gruppi di servizi e priorità, in un solo campo, unendoli attraverso un punto e virgola (‘`;`’). Si possono escludere delle priorità ponendo anteriormente un punto esclamativo (‘`!`’).

Il secondo campo, quello che definisce l’azione, serve a indicare la destinazione dei messaggi riferiti a un certo gruppo di servizi e priorità, come definito dal primo campo. Può trattarsi di un file o di altro, a seconda del primo carattere utilizzato per identificarlo. Segue l’elenco.

Tabella 16.3. Destinazione dei messaggi.

Primo carattere	Descrizione
`/`	Se il primo carattere è una barra obliqua normale, si intende che si tratti dell’indicazione di un percorso assoluto di un file destinatario dei messaggi. Può trattarsi anche di un file di dispositivo opportuno, come quello di una console virtuale.
`|`	Se il primo carattere è una barra verticale, si intende che la parte restante sia l’indicazione del percorso assoluto di un file FIFO (*pipe* con nome), generato attraverso ‘**`mkfifo`**’ (20.15.1).
`@`	Se il primo carattere è il simbolo ‘`@`’, si intende che la parte restante sia l’indicazione di un elaboratore remoto, che ricevendo tali messaggi li inserisce nel proprio sistema di registrazione.
utente `[,…]`	Se il primo carattere non è scelto tra quelli elencati fino a questo punto, si intende che si tratti di un elenco di utenti (separati da virgole) a cui inviare i messaggi sullo schermo del terminale, se questi stanno accedendo in quel momento.

Primo carattere	Descrizione
`*`	Se il primo e unico carattere è un asterisco (‘**`*`**’), si intende che i messaggi debbano essere inviati sullo schermo del terminale di tutti gli utenti connessi in quel momento.

È importante osservare che gli stessi messaggi possono essere inviati anche a destinazioni differenti, attraverso più record in cui si definiscono le stesse coppie di servizi e priorità, oppure coppie differenti che però si sovrappongono. Per un approfondimento si veda anche la pagina di manuale *syslog.conf(5)*.

Segue la descrizione di alcuni esempi.

- ```
  *.*                     /var/log/syslog
  ```
  Invia tutti i messaggi nel file ‘`/var/log/syslog`’.

- ```
  kern.*                  /dev/console
  ```
  I messaggi del servizio ‘**`kern`**’, a qualunque livello di priorità appartengano, vengono inviati al dispositivo corrispondente alla console. In pratica vengono scritti sullo schermo della console.

- ```
  mail.*                  /var/log/maillog
  ```
  I messaggi riferiti alla gestione della posta elettronica sono memorizzati nel file ‘`/var/log/maillog`’.

- ```
  *.warning               @dinkel.brot.dg
  ```
  I messaggi la cui priorità raggiunge o supera il livello ‘**`warning`**’, vengono inviati all’elaboratore *dinkel.brot.dg*.

- ```
  *.*                         @dinkel.brot.dg
  *.=debug                    /var/log/debug
  *.=info;*.=notice           /var/log/messages
  *.warning                   /var/log/syslog
  ```

  Invia tutti i messaggi all'elaboratore *dinkel.brot.dg*; inoltre invia i messaggi '**debug**' nel file '`/var/log/debug`', i messaggi '**info**' e '**notice**' nel file '`/var/log/messages`', infine i messaggi da '**warning**' in su nel file '`/var/log/syslog`'.

- ```
  *.=info;*.=notice           /var/log/messages
  *.warning                   /var/log/syslog
  *.=debug;*.=info            /dev/tty9
  *.=notice;*.=warning        /dev/tty10
  *.=err;*.=crit              /dev/tty11
  *.=alert;*.=emerg           /dev/tty12
  ```

  Invia i messaggi '**info**' e '**notice**' nel file '`/var/log/messages`', i messaggi da '**warning**' in su nel file '`/var/log/syslog`', quindi suddivide nuovamente i livelli di priorità e li invia a quattro diverse console virtuali, da '`/dev/tty9`' a '`/dev/tty12`'.

16.1.1.3 Archiviazione dei file delle registrazioni del sistema

Per archiviare i file generati da '**syslogd**', se la propria distribuzione GNU/Linux non gestisce già questo problema, si possono spostare i file delle registrazioni altrove, dove poi eventualmente possono anche essere compressi comprimendoli, ripristinando i file originali vuoti e riavviando i servizi che li aggiornano.

Supponendo di dovere gestire il file '`/var/log/syslog`', prodotto da '**syslogd**'. Si potrebbe procedere secondo la modalità seguente:

```
# mv /var/log/syslog /var/log/syslog.`date +%Y%m%d`[Invio]

# touch /var/log/syslog[Invio]

# killall -HUP syslogd[Invio]

# gzip -9 /var/log/syslog.`date +%Y%m%d`[Invio]
```

In pratica, in questo modo, il file '`/var/log/syslog`' verrebbe archiviato in un file del tipo '`/var/log/syslog.`***aaaammgg***`.gz`', dove ***aaaammgg*** rappresenta la data di archiviazione.

16.1.1.4 Riservatezza delle informazioni

Le informazioni che vengono memorizzate nel registro del sistema potrebbero essere delicate, sia per la sicurezza del sistema, sia per i singoli utenti. Per questo, è bene ricordare che i file che compongono il registro del sistema non dovrebbero essere accessibili in lettura agli utenti comuni.

16.1.1.5 Utilizzo di «logger»

Il programma '**`logger`**' permette di aggiungere delle annotazioni all'interno del registro del sistema locale.[2] Se non vengono forniti argomenti, il messaggio da registrare viene atteso dallo standard input. Se si utilizza la tastiera, per concludere è necessario utilizzare il codice di EOF che di norma si ottiene con la combinazione [*Ctrl d*].

`logger` [***opzioni***] [***messaggio***]

Opzione	Descrizione
`-f` ***file***	Permette di includere il file indicato all'interno del registro del sistema.

16.1.1.6 Utilizzo di «klogd»

Il programma '**klogd**' è il demone specifico per l'intercettazione e la registrazione dei messaggi del kernel Linux. [3] Di norma viene avviato dalla procedura di inizializzazione del sistema, subito dopo '**syslogd**'.

```
klogd [opzioni]
```

Il demone '**klogd**', oltre a inviare i messaggi del kernel al registro, visualizza sulla console i messaggi più importanti. Il livello di importanza dei messaggi da inviare anche sulla console dipende dall'opzione '**-c**'. Normalmente, il valore predefinito associato a questa opzione è quattro; per ridurre la quantità di messaggi che si ricevono sulla console basta portare questo valore a tre. Per modificare questo valore, di norma è necessario intervenire nello script della procedura di inizializzazione del sistema che si occupa del suo avvio.

Tabella 16.11. Alcune opzioni.

Opzione	Descrizione
`-f` ***file_delle_registrazioni***	Specifica un file particolare per le registrazioni, invece di dirigere i messaggi direttamente al demone della gestione del registro del sistema, cioè '**syslogd**'.

Opzione	Descrizione
`-c` ***n***	Specifica il livello di priorità dei messaggi da non inviare alla console. In pratica, normalmente è predefinito il livello quattro, che comporta la visualizzazione dei messaggi da zero a tre (che sono più importanti).

16.1.2 Rotazione dei file

«

I file utilizzati per annotare ciò che accade nel sistema possono essere generati da '**`syslogd`**', o da un programma analogo, ma nel sistema si aggiungono normalmente altri file generati specificatamente per il controllo di altri programmi. L'unico punto in comune dei vari programmi che generano file di questo tipo è la directory di partenza, all'interno della quale questi file vengono collocati: '`/var/log/`'. A parte questo, il problema che si incontra normalmente sta nel sistemare una procedura di rotazione dei file, che includa tutto ciò di cui c'è bisogno.

Può essere necessario conoscere la struttura del sistema di rotazione dei file delle registrazioni della propria distribuzione, nel caso il proprio utilizzo del sistema implichi l'obbligo di conservare questi dati per un certo tempo. Infatti, di solito il sistema automatico di archiviazione si occupa di mantenere solo pochi giorni di informazioni.

16.1.2.1 Rotazione dei file delle registrazioni di sistema nelle distribuzioni Debian

Le distribuzioni GNU/Linux Debian gestiscono un proprio sistema per la rotazione dei file delle registrazioni generati dalla configurazione del file '`/etc/syslog.conf`'.

Per la precisione, il meccanismo si appoggia su due script avviati periodicamente dal sistema Cron: '`/etc/cron.daily/sysklogd`' e '`/etc/cron.weekly/sysklogd`'. Come si può comprendere, il primo viene avviato ogni giorno e il secondo ogni settimana.

Questi script si avvalgono di due programmi: '**`syslogd-listfiles`**'[4] e '**`savelog`**'.[5] Il primo di questi due programmi serve a estrapolare dal file '`/etc/syslog.conf`' l'elenco dei file delle registrazioni utilizzati effettivamente, distinguendo in qualche modo tra quelli che vanno ruotati giornalmente e quelli che invece richiedono un ciclo settimanale. In base all'elenco ottenuto, viene poi usato '**`savelog`**' che si occupa effettivamente di creare il ciclo di file.

Per esempio, se nel file '`/etc/syslog.conf`' esiste una riga come quella seguente, '**`savelog`**' viene utilizzato giornalmente per salvare il file '`/var/log/registro`':

```
*.*                                     /var/log/registro
```

Per la precisione, analizzando lo script '`/etc/cron.daily/sysklogd`' si potrebbe leggere un ciclo come quello seguente:

```
cd /var/log
for LOG in `syslogd-listfiles`
do
   if [ -s $LOG ]; then
      savelog -g adm -m 640 -u root -c 3 $LOG >/dev/null
   fi
done
```

Si può osservare in questo caso che '**savelog**' viene avviato con l'opzione '**-c 3**', che richiede una rotazione in tre file differenti, generando in pratica i file:

File	Descrizione
'/var/log/registro'	file corrente;
'/var/log/registro.0'	ultima archiviazione del file (non compressa);
'/var/log/registro.1.gz'	penultima archiviazione del file (compressa);
'/var/log/registro.2.gz'	terzultima archiviazione del file (compressa).

Come si può intendere, aumentando il valore dell'opzione '**-c**' aumenta di conseguenza la quantità di archivi precedenti del file che viene ruotato. In questo caso, volendo eventualmente conservare un anno di file delle registrazioni, nello script '/etc/cron.daily/sysklogd' occorrerebbe usare l'opzione '**-c 365**' e nello script '/etc/cron.weekly/sysklogd' occorrerebbe usare l'opzione '**-c 53**'.

16.1.2.2 Logrotate

Logrotate [6] è un sistema di archiviazione dei file delle registrazioni, con un sistema di configurazione che consente l'inclusione e l'eliminazione di file, senza creare troppe complicazioni. Tutto quanto si basa sul programma eseguibile '**logrotate**', a cui si associa un file di configurazione, generalmente '`/etc/logrotate.conf`', che normalmente incorpora altre porzioni di configurazione contenute nella directory '`/etc/logrotate.d/`'. Generalmente, il programma '**logrotate**' viene avviato giornalmente dal sistema Cron.

Il file di configurazione ha una struttura abbastanza intuitiva; quello che segue è l'esempio di una distribuzione Debian:

```
# see "man logrotate" for details
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# uncomment this if you want your log files compressed
#compress
# packages drop log rotation information into this directory
include /etc/logrotate.d
# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 1
}
/var/log/btmp {
    missingok
```

```
    monthly
    create 0664 root utmp
    rotate 1
}
# system-specific logs may be configured here
```

Come si può osservare le direttive possono essere generali, oppure inserite all'interno di sezioni, corrispondenti al nome di un file da ruotare periodicamente. Le direttive generali valgono fino a dove non vengono modificate; pertanto diventano direttive predefinite.

Nell'esempio appare inizialmente la direttiva '**`weekly`**', che richiede una rotazione settimanale, quindi, con la direttiva '**`rotate 4`**' viene specificata la quantità di cicli, che in questo caso sono quattro, pari a quattro settimane.

Diventa molto importante la direttiva '**`include /etc/logrotate.d`**', che richiede espressamente l'inclusione di tutti i file contenuti nella directory '`/etc/logrotate.d/`'; file che potrebbero essere simili a quello seguente, riferito alle esigenze di PostgreSQL:

```
/var/log/postgresql/postgres.log {
        daily
        rotate 10
        copytruncate
        delaycompress
        compress
        notifempty
        create 640 postgres postgres
}
```

Generalmente, se la propria distribuzione GNU utilizza Logrotate, è molto probabile che tutto sia già predisposto correttamente; tuttavia,

nel caso si intenda conservare le informazioni dei file delle registrazioni per un tempo diverso da quello previsto in modo predefinito da chi ha organizzato i pacchetti applicativi, diventa necessario intervenire nel file di configurazione generale e nelle singole sezioni, soprattutto nei file inclusi. Infatti, come si vede dagli esempi già apparsi, la configurazione generale del periodo di rotazione e della quantità di file conservati viene stabilito in modo generale, ma poi, quasi ogni sezione modifica i tempi e la grandezza del ciclo.

Per approfondire l'uso e la configurazione di Logrotate si può leggere la pagina di manuale *logrotate(8)*.

16.1.3 Console-log

In questo capitolo è già descritto in che modo è possibile inviare i messaggi destinati normalmente ai file delle registrazioni su una o più console di un sistema GNU/Linux, attraverso la configurazione del file '`/etc/syslog.conf`'. Il pacchetto Console-log[7] consente di raggiungere questo risultato in modo più pratico, consentendo di ottenere un testo che può anche essere fatto scorrere sullo schermo.

Il tutto parte da uno script della procedura di inizializzazione del sistema, che potrebbe essere precisamente '`/etc/init.d/console-log`', il quale legge un file di configurazione e si comporta di conseguenza.

Il file di configurazione dovrebbe essere '`/etc/console-log.conf`' e il suo contenuto standard è il seguente:

```
# /etc/console-log.conf -- configuration file for console-log.
# See console-log.conf(5) for details
tty 9
# uncomment next line if you want to chvt to the syslog on startup
#chvt yes
file /var/log/syslog
group adm
tty 8
file /var/log/exim4/mainlog /var/log/exim/mainlog /var/log/mail.log
group adm
```

Intuitivamente si comprende che in questo modo si vuole inviare alla console numero nove una copia del file '`/var/log/syslog`', mentre alla console numero otto si vuole inviare una copia del file '`/var/log/exim4/mainlog`'. In modo simile si può estendere il file di configurazione per includere altri file da visualizzare su altrettante console. Si veda comunque la pagina di manuale *console-log.conf(5)* per la spiegazione dettagliata delle direttive utilizzabili nella configurazione.

Una volta configurato correttamente Console-log e riavviata la sua funzione attraverso lo script relativo della procedura di inizializzazione del sistema, sui terminali predisposti si ottiene la visualizzazione di questi file, ma è sufficiente premere la combinazione [*Ctrl c*] per interrompere la visualizzazione normale e passare al controllo di Less, cosa che consente di scorrere all'indietro e anche orizzontalmente il testo.

Quando si scorre il testo attraverso Less, non si ricevono altri dati dal file originale, pertanto, alla fine è bene ritornare allo stato precedente chiudendo il funzionamento di Less, con la pressione del tasto [*q*].

16.2 Controllo degli accessi

In ogni sistema operativo multiutente c'è la necessità di controllare gli accessi. Nei sistemi Unix un utente che può accedere ha un *account*: letteralmente si tratta di un conto, o in altri termini un «accredito», o meglio ancora una specie di contratto di utenza che gli permette di esistere nel sistema in qualità di «utente logico».

16.2.1 Identità reale o efficace

La gestione delle utenze in un sistema Unix comporta l'associazione ai processi elaborativi di identità virtuali, con le quali è possibile definire dei privilegi, per il controllo dell'uso del sistema operativo. Un processo elaborativo di un sistema Unix dispone almeno di due identità fondamentali: l'utente e il gruppo a cui questo appartiene. Ognuna delle identità fondamentali si distingue in almeno tre possibilità: identità reale, identità salvata e identità efficace.

L'identità reale di un processo elaborativo è quella con cui è stato creato; l'identità efficace è quella che viene considerata per il controllo effettivo dei privilegi di cui il processo dispone; l'identità salvata è quella che il processo aveva prima di acquisire l'identità efficace attuale.

Un processo elaborativo di un sistema Unix, per poter cambiare la propria identità efficace deve utilizzare la chiamata di sistema stan-

dard ***setuid()***, con la quale si tiene conto dell'identità reale originale ed eventualmente dell'identità salvata precedentemente.

Ai fini pratici, per l'utente di un sistema Unix che non si interessi della programmazione, è importante sapere qual è l'identità efficace di un processo, tanto che spesso si sottintende questo fatto, quando si fa riferimento a un'identità senza altre specificazioni.

16.2.2 Login, ovvero la procedura di accesso

«

Il *login*[8] è la procedura di accesso attraverso la quale un utente, registrato precedentemente, viene riconosciuto e gli viene concesso di utilizzare il sistema. Il concetto è simile a quello di una firma di ingresso. Quando un utente conclude la sua attività con il sistema, esegue un *logout*. Il concetto del *logout* è simile a quello di una firma di uscita.

La procedura di accesso è controllata normalmente dal programma '**`login`**', il quale si prende cura di verificare la parola d'ordine fornita, prima di consentire l'ingresso dell'utente. Tuttavia, i programmi '**`login`**' non sono uguali tutti i sistemi Unix e ognuno può essere stato predisposto per una politica differente. A titolo di esempio, un programma '**`login`**' potrebbe accettare l'accesso da parte di utenti per i quali non sia stata definita una parola d'ordine, mentre un altro potrebbe escluderlo. In queste sezioni si affronta il problema in modo superficiale, cercando di fare riferimento alle consuetudini che sembrano consolidate; il lettore deve tenere presente che l'unica documentazione certa sul funzionamento di '**`login`**' è quella fornita assieme al proprio sistema operativo: la pagina di manuale *login(1)*.

16.2.2.1 Utilizzo di «login»

Il programma '**login**' permette l'accesso dell'utente al sistema. Di solito non si usa direttamente, anzi, ciò dovrebbe essere impossibile: è compito del programma di gestione del terminale, Getty o simili, di avviarlo dopo aver ottenuto il nominativo-utente.

```
login [utente]
```

Ogni utente registrato nel sistema, cioè ogni utente che (teoricamente) può accedere al sistema, ha una directory personale, o directory *home*, all'interno della quale si trova posizionato al momento dell'accesso. Questa directory contiene dei file riguardanti la configurazione particolare dell'utente a cui appartiene. La directory personale è collocata normalmente in '`/home/`***nome_utente***`/`' e questa, se la shell lo consente, viene abbreviata utilizzando il simbolo tilde ('**~**'). La directory personale dell'utente '**root**' è speciale e dovrebbe trovarsi in '`/root/`'. Durante un accesso normale da parte di un utente qualunque, compreso '**root**', vengono richiesti il nome dell'utente (se non è già stato fornito nella riga di comando) e la parola d'ordine. Quindi vengono visualizzati:

- la data e l'ora dell'ultimo accesso;
- l'avviso della presenza di posta (se esistono messaggi di posta elettronica non ancora letti);
- il messaggio del giorno.

Se si tratta di un utente al quale è associata una parola d'ordine, questa viene richiesta e controllata. Se risulta errata, vengono consentiti

un numero limitato di tentativi. Generalmente, gli errori vengono riportati all'interno del registro del sistema. Se l'utente che chiede di accedere non è '**root**' e se esiste il file '`/etc/nologin`', ne viene visualizzato il contenuto sullo schermo e non viene consentito l'accesso. Ciò serve per impedire l'accesso al sistema, tipicamente quando si intende chiuderlo. Perché l'accesso possa essere effettuato come utente '**root**', occorre che il terminale (TTY) da cui si intende accedere sia elencato all'interno di '`/etc/securetty`'. I tentativi di questo tipo che provengono da terminali non ammessi, vengono annotati all'interno del registro del sistema. Se esiste il file '`~/.hushlogin`', viene eseguito un accesso silenzioso, nel senso che vengono disattivati:

- il controllo per la presenza di messaggi di posta elettronica;
- la visualizzazione della data e dell'ora dell'ultimo accesso effettuato da parte di quell'utente;
- la visualizzazione del messaggio del giorno.

Se esiste il file '`/var/log/lastlog`', viene visualizzata la data e l'ora dell'ultimo accesso e ne viene registrato quello in corso. Al termine della procedura di accesso viene avviata la shell dell'utente. Se all'interno del file '`/etc/passwd`' non è indicata la shell da associare all'utente che accede, viene utilizzato '`/bin/sh`'. Se all'interno del file '`/etc/passwd`' non è indicata la directory personale dell'utente, o se quella indicata non è raggiungibile, viene utilizzata la directory radice ('**/**').

Quanto affermato dovrebbe essere sufficiente per capire che la semplice rimozione dell'indicazione della shell o della directory personale da un record del file '`/etc/passwd`', non è sufficiente per impedire l'accesso a un utente.

16.2.2.2 File «/etc/passwd»

Il file '`/etc/passwd`' è un elenco di utenti, parole d'ordine, directory *home* (directory personali nel caso di utenti umani), shell e altre informazioni personali utilizzate da Finger (36.6.3). La struttura dei record (le righe) di questo file è molto semplice:

utente : ***parola_d'ordine_cifrata*** : ***uid*** : ***gid*** : ***dati_personali*** : ***directory_home*** : ***shell***

Segue la descrizione dei campi.

1. ***utente***
   È il nome utilizzato per identificare l'utente logico che accede al sistema.
2. ***parola_d'ordine_cifrata***
   È la parola d'ordine cifrata. In condizioni normali, se questa indicazione manca, l'utente può accedere senza indicare alcuna parola d'ordine.

Se questo campo contiene un asterisco (‘`*`’) l’utente non può accedere al sistema. Con questa tecnica è possibile impedire temporaneamente l’accesso, con la possibilità di ripristinarlo successivamente con la stessa parola d’ordine, togliendo semplicemente l’asterisco.

3. ***uid***

   È il numero identificativo dell’utente (*User ID*).

4. ***gid***

   È il numero identificativo del gruppo a cui appartiene l’utente (*Group ID*).

5. ***dati_personali***

   Di solito, questo campo contiene solo l’indicazione del nominativo completo dell’utente (nome e cognome), ma può contenere anche altre informazioni che di solito sono inserite attraverso ‘**`chfn`**’ (16.4.6).

6. ***directory_home***

   La directory assegnata all’utente.

7. ***shell***

   La shell assegnata all’utente.

Segue la descrizione di alcuni esempi.

- `tizio:724AD9dGbG25k:502:502:Tizio Tizi,,,,:/home/tizio:/bin/bash`

  L’utente ‘**`tizio`**’ corrisponde al numero UID 502 e al numero GID 502; si chiama Tizio Tizi; la sua directory personale è

‘/home/tizio/’; la sua shell è ‘/bin/bash’. Di questo utente, personalmente, non si conosce niente altro che il nome e il cognome. Il fatto che UID e GID corrispondano dipende da una scelta organizzativa dell’amministratore del sistema.

- `tizio:*:502:502:Tizio Tizi,,,,:/home/tizio:/bin/bash`

  Questo esempio mostra una situazione simile a quella precedente, ma l’utente ‘**tizio**’ non può accedere, perché al posto della parola d’ordine cifrata appare un asterisco.

16.2.2.3 File «/etc/group»

È l’elenco dei gruppi di utenti. La struttura delle righe di questo file è molto semplice.

gruppo*:*parola_d’ordine_cifrata*:*gid*:*lista_di_utenti

Segue la descrizione dei campi.

1. ***gruppo***

   È il nome utilizzato per identificare il gruppo.

2. ***parola_d’ordine_cifrata***

   È la parola d’ordine cifrata. Di solito non viene utilizzata e di conseguenza non viene inserita. Se è presente una parola d’ordine, questa dovrebbe essere richiesta quando un utente tenta di cambiare gruppo attraverso ‘**newgrp**’ (16.2.3.2).

3. ***gid***

   È il numero identificativo del gruppo.

4. ***lista_di_utenti***

   È la lista degli utenti che appartengono al gruppo, anche se questo non risulta dal file '`/etc/passwd`'. Si tratta di un elenco di nomi di utente separati da virgole.

Segue la descrizione di alcuni esempi.

- `tizio::502:`

  Si tratta di un caso molto semplice in cui il gruppo '**`tizio`**' non ha alcuna parola d'ordine e a esso non appartiene alcun utente aggiuntivo, oltre a quanto già specificato nel file '`/etc/passwd`'.

- `users::100:tizio,caio,sempronio`

  In questo caso, gli utenti '**`tizio`**', '**`caio`**' e '**`sempronio`**' appartengono al gruppo '**`users`**'.

16.2.2.4 File «/etc/shadow»

«

Il file '`/etc/shadow`' appare in quei sistemi in cui è attivata la gestione delle parole d'ordine oscurate (*shadow password*). Serve a contenere le parole d'ordine cifrate, togliendole dal file '`/etc/passwd`'. Così facendo, è possibile inibire la maggior parte dei permessi di accesso a questo file ('`/etc/shadow`'), proteggendo le parole d'ordine che contiene. Al contrario, non è possibile impedire l'accesso in lettura del file '`/etc/passwd`' che fornisce una quantità di informazioni sugli utenti, indispensabili a molti programmi. Il problema è descritto nella sezione 16.8.

16.2.2.5 File «/var/run/utmp»

Il file ‘`/var/run/utmp`’ contiene l’elenco degli accessi in essere nel sistema. Non è un file di testo normale e per l’estrazione delle informazioni in esso contenute si usano dei programmi di servizio appositi. Tuttavia, è possibile che gli utenti presenti effettivamente nel sistema siano in numero maggiore, a causa del fatto che non tutti i programmi usano il metodo di registrazione fornito attraverso questo file.

Se il file non esiste, conviene crearlo manualmente in uno dei due modi seguenti:

```
# cp /dev/null /var/run/utmp[Invio]

# touch /var/run/utmp[Invio]
```

Solitamente, è la procedura di inizializzazione del sistema a prendersi cura di questo file, azzerandolo o ricreandolo, a seconda della necessità.

16.2.2.6 File «/var/log/wtmp»

Il file ‘`/var/log/wtmp`’ ha una struttura analoga a quella di ‘`/var/run/utmp`’ e serve per conservare la registrazione degli accessi e della loro conclusione (*login-logout*). Questo file non viene creato automaticamente; se manca, la conservazione delle registrazioni all’interno del sistema non viene effettuata. Viene aggiornato da Init e anche dal programma che si occupa di gestire la procedura di accesso al sistema (‘**login**’).

Il formato di questo file non è quello di un file di testo normale, quindi non è leggibile o stampabile direttamente.

Se questo file non esiste, conviene crearlo manualmente in uno dei due modi seguenti:

```
# cp /dev/null /var/log/wtmp[Invio]

# touch /var/log/wtmp[Invio]
```

16.2.2.7 File «/etc/motd»

Il contenuto del file '`/etc/motd`' viene visualizzato da '**`login`**' al termine della procedura di accesso, prima dell'avvio della shell associata all'utente. Questo file contiene, o dovrebbe contenere, il cosiddetto messaggio del giorno (*Message of the day*).

16.2.2.8 File «/etc/nologin»

Se esiste il file '`/etc/nologin`', '**`login`**' non accetta nuovi accessi al sistema, visualizzando il suo contenuto a ogni tentativo.

Se si desidera fermare il sistema è possibile creare questo file scrivendoci all'interno il motivo, o una breve spiegazione di ciò che sta avvenendo.

16.2.2.9 File «/etc/securetty»

Il file '`/etc/securetty`' contiene l'elenco dei ***terminali sicuri***, cioè di quelli da cui si permette l'accesso all'utente '**`root`**'. I nomi dei terminali vengono indicati facendo riferimento ai file di dispositivo relativi, senza l'indicazione del prefisso '`/dev/`'. L'esempio seguente mostra un elenco di terminali che comprende la console vera e propria, le sei console virtuali standard, quattro terminali seriali e quattro pseudo-terminali che accedono dalla rete locale oppure dal sistema grafico X.

```
console
tty1
tty2
tty3
tty4
tty5
tty6
ttyS0
ttyS1
ttyS2
ttyS3
ttyp0
ttyp1
ttyp2
ttyp3
```

A seconda di come è organizzato il sistema di file di dispositivo, può essere necessario modificare di conseguenza questo file.

16.2.2.10 Casella di posta elettronica

Il file il cui percorso si trova contenuto nella variabile di ambiente ***MAIL*** (a volte corrisponde a '`/var/mail/`***nome_utente***', altre volte a '`/var/spool/mail/`***nome_utente***', oppure anche a un file contenuto nella directory personale dell'utente stesso), viene usato normalmente per accumulare i messaggi di posta elettronica a lui diretti.

Il programma '**`login`**', dopo la visualizzazione del messaggio contenuto in '`/etc/motd`', se trova che c'è posta per l'utente, visualizza un messaggio di avvertimento in tal senso.

La collocazione dei file che rappresentano le caselle postali degli utenti, dipende dalla configurazione e dalla filosofia del sistema di gestione della posta elettronica. Sulla base di tale configurazione, i processi devono ottenere una variabile di ambiente ***MAIL*** con il valore necessario a raggiungere la casella della posta in ingresso.

16.2.2.11 File «~/.hushlogin»

«

Se esiste il file '`.hushlogin`' all'interno della directory personale di un certo utente, quando quell'utente accede, '**`login`**' non visualizza alcun messaggio introduttivo.

16.2.2.12 File «/var/log/lastlog»

«

Il file '`/var/log/lastlog`', se esiste, viene utilizzato da '**`login`**' per registrare gli ultimi accessi al sistema e per poter visualizzare la data e l'ora dell'ultimo accesso. Se questo file non esiste, conviene crearlo manualmente in uno dei due modi seguenti:

```
# cp /dev/null /var/log/lastlog[Invio]

# touch /var/log/lastlog[Invio]
```

16.2.3 Cambiamento di identità

«

Alcuni programmi consentono di ottenere i privilegi di un altro utente, come se si ripetesse la procedura di accesso. Questa possibilità rappresenta generalmente un problema di sicurezza. Per mezzo di questi programmi può capitare di riuscire a ottenere i privilegi dell'utente '**`root`**' anche quando si accede da un terminale che non

viene considerato sicuro, pertanto non risulta incluso nell'elenco di '`/etc/securetty`'.

16.2.3.1 Utilizzo di «su»

Il programma '**su**'[9] permette a un utente di diventare temporaneamente un altro, avviando una shell con i privilegi dell'utente indicato (questo vale anche per il gruppo o i gruppi a cui questo appartiene). Se non viene indicato un utente, '**su**' sottintende '**root**'. Prima di attivare la nuova shell, viene richiesta la parola d'ordine associata all'utente selezionato, a meno che '**su**' sia stato eseguito da chi sta già operando in qualità di utente '**root**'.

```
su [opzioni] [utente]
```

L'opzione più importante di '**su**' è data dal trattino singolo ('**-**'), con il quale si fa in modo che la nuova shell venga avviata come shell di *login*. In questo modo, si ha di fronte l'ambiente normale dell'utente che si va a impersonare, come se si facesse un accesso standard.

Per terminare l'attività in veste di questo nuovo utente, basta concludere l'esecuzione della shell con il comando '**exit**'. Segue la descrizione di alcuni esempi.

- `$` **su** [*Invio*]

  Utilizzando '**su**' senza argomenti, si intende implicitamente di voler acquisire i privilegi dell'utente '**root**'. Per questo viene richiesta la parola d'ordine.

- `$` **su caio** [*Invio*]

Volendo trasformarsi temporaneamente in un altro utente, basta indicarlo come argomento, come in questo caso. Viene richiesta la parola d'ordine.

- `# **su tizio**[*Invio*]`

  L'utente '**root**' può sempre fare quello che vuole; pertanto, se seleziona un altro utente perde dei privilegi, così non gli viene richiesta alcuna parola d'ordine.

- `$ **su - caio**[*Invio*]`

  Si acquista la personalità dell'utente '**caio**', con tutto l'ambiente normale, senza semplificazioni.

Il programma '**su**', per poter svolgere il suo compito, deve appartenere all'utente '**root**' e avere il bit SUID attivato (SUID-root). È in questo modo che un utente comune riesce a ottenere i privilegi di '**root**' o di un altro utente.

`$ **ls -l /bin/su**[*Invio*]`

```
-rwsr-xr-x 1 root root 27908 2009-05-22 17:03 /bin/su
```

Il programma '**su**' viene usato frequentemente dall'utente '**root**', o da un processo che ha già i privilegi dell'utente '**root**', per diventare temporaneamente un utente comune. In tal caso, dal momento che il processo che avvia '**su**' ha già i privilegi di '**root**', non c'è alcuna necessità della presenza del bit SUID attivo.

In generale, dal momento che '**su**' è molto importante per agevolare il lavoro dell'amministratore del sistema, se si temono problemi alla sicurezza, si può eliminare il bit SUID, per concedere praticamente il suo utilizzo solo all'utente '**root**':

```
# chmod u-s /bin/su [Invio]
```

Volendo calcare la mano, si possono togliere anche tutti i permessi per il gruppo proprietario e per gli altri utenti:

```
# chmod go-rwx /bin/su [Invio]
```

Se si toglie la funzione SUID-root all'eseguibile '**su**', si impedisce agli utenti comuni di elevarsi al livello di utente '**root**'. Questo fatto implica che, a meno di disporre di altri programmi che compiano funzioni simili, l'utente '**root**' può accedere solo nel modo «normale», pertanto si va così a confermare il rispetto del file '`/etc/securetty`' che altrimenti potrebbe essere raggirato. Tuttavia, è perfettamente ammissibile la presenza di un file '`/etc/securetty`' limitato, assieme a un programma '**su**' con i permessi SUID-root, quando si vuole consentire un accesso solo a chi dispone di un'utenza normale, lasciando salva poi la possibilità di dimostrare di essere anche l'amministratore.

Fino a questo punto è stato mostrato l'utilizzo di '**su**' per avviare una shell interattiva, ma in generale questo programma può essere usato per avviare un certo processo elaborativo, al termine del quale si vuole che tutto ritorni come prima. In tal caso la sintassi va modificata come segue:

```
su [opzioni] [utente [argomenti]]
```

In pratica, gli argomenti che appaiono alla fine della riga di comando sono ciò che si vuole avviare. Tuttavia, **si creano spesso delle complicazioni nel modo corretto di interpretare tali argomenti.**

Si vedano gli esempi seguenti, che in teoria dovrebbero produrre lo stesso effetto:

1. `# su tizio ls -l` [*Invio*]
2. `# su tizio "ls -l"` [*Invio*]
3. `# su -c "ls -l" tizio` [*Invio*]

In pratica, si vuole che il comando '**`ls -l`**' venga eseguito con i privilegi dell'utente '**`tizio`**', quando originariamente si hanno quelli dell'amministratore. Delle tre forme, è sicuro il funzionamento solo dell'ultima, dove ci si affida all'opzione '**`-c`**' e di conseguenza il comando da eseguire è passato in forma di stringa. Naturalmente, sarebbe ammissibile scrivere quel comando anche nel modo seguente:

```
# su tizio -c "ls -l" [Invio]
```

Inoltre, l'uso di questa forma, consente di scrivere comandi più complessi, come nell'esempio seguente:

```
# su tizio -c "cd ; ls -l | less" [Invio]
```

16.2.3.2 Utilizzo di «newgrp»

«

Il programma '**`newgrp`**'[10] permette di cambiare il gruppo a cui appartiene l'utente. L'utente non cambia, la directory personale nemmeno, cambia solo il GID. Un utente può cambiare gruppo se nel file '`/etc/group`' sono diversi i gruppi a cui può appartenere l'utente. In alternativa, se il gruppo ha una parola d'ordine, l'utente può «entrare» nel gruppo solo se la conosce.

```
newgrp [gruppo]
```

Il problema della gestione dei gruppi, specialmente per ciò che riguarda le parole d'ordine, è descritto meglio nella sezione 16.8.

16.2.4 Informazioni sugli accessi

Molti programmi permettono di avere informazioni sugli accessi e di conseguenza anche sugli utenti. In particolare sono importanti quelli che permettono di leggere il contenuto dei file '`/var/run/utmp`' e '`/var/log/wtmp`' il cui formato non è leggibile attraverso l'uso di un semplice '**`cat`**' (sezione 16.10).

16.2.4.1 Utilizzo di «users»

Il programma '**`users`**' [11] visualizza i nomi degli utenti che accedono attualmente all'elaboratore. Se un utente ha attivato più sessioni in corso, il suo nome appare più volte nell'elenco. Se il comando viene avviato senza l'indicazione di un file, i dati visualizzati vengono estratti da '`/etc/utmp`'. Esiste comunque la possibilità di visualizzare attraverso '**`users`**' il contenuto di '`/etc/wtmp`'.

```
users [file]
```

16.2.4.2 Utilizzo di «w»

Il programma '**`w`**'[12] visualizza i nomi degli utenti che accedono attualmente e varie informazioni sulla loro attività; in particolare l'uso della CPU:

```
w [opzioni] [utente]
```

Attraverso le opzioni è possibile controllare in qualche modo le informazioni che vengono visualizzate, mentre se si indica il nome di un utente alla fine della riga di comando, si ottengono informazioni solo su quello. L'esempio seguente mostra cosa si può ottenere con '**w**' usato senza argomenti:

```
$ w [Invio]

 16:50:46 up 15 min, 5 users, load average: 0,12, 0,14, 0,20
 USER  TTY   FROM       LOGIN@  IDLE  JCPU  PCPU WHAT
 tizio tty1  -          16:37  10:53  6.88s 6.75s ↩
↪/usr/bin/mc -P /tmp/mc4818-2783
 tizio tty5  -          16:50  23.00s 1.17s 0.12s -sh
 tizio tty6  -          16:38  40.00s 0.28s 0.12s ssh ↩
↪root@172.21.1.1
 root  pts/2 172.21.1.2 16:46  45.00s 0.07s 0.07s -bash
 tizio pts/3 :0.0       16:50   0.00s 0.07s 0.01s w
```

Come si può osservare, la prima riga che si ottiene è equivalente a quanto genera il programma '**uptime**', descritto nella sezione 10.3.

Le colonne identificate dalle sigle '**JCPU**' e '**PCPU**' indicano il tempo di utilizzo della CPU; nel primo caso si tratta di tutti i processi, ancora attivi, anche se sullo sfondo, mentre nel secondo si tratta esclusivamente del tempo utilizzato dal processo indicato nella colonna finale ('**WHAT**'). La colonna '**IDLE**' indica il tempo di funzionamento complessivo, anche se inattivo, del processo indicato nella colonna '**WHAT**'.[13]

Per l'uso delle opzioni che qui non vengono descritte, si veda la pagina di manuale *w(1)*.

16.2.4.3 Utilizzo di «who»

Il programma '**who**'[14] visualizza i nomi degli utenti che accedono attualmente e varie informazioni sulla loro attività. '**who**' trae normalmente le sue informazioni dal file '`/etc/utmp`', se non ne viene indicato un altro negli argomenti (per esempio '`/etc/wtmp`').

```
who [opzioni] [file] [am i]
```

Si veda il documento *info who*, oppure la pagina di manuale *who(1)*.

16.2.4.4 Utilizzo di «pinky»

Il programma '**pinky**'[15] visualizza l'elenco degli utenti che utilizzano l'elaboratore, oppure visualizza informazioni dettagliate su utenti particolari. Il suo funzionamento è simile a quello di Finger (36.6.3):

```
pinky [opzioni] [utente]...
```

Attraverso le opzioni si può controllare la quantità di informazioni che si vogliono ottenere.

Tabella 16.25. Alcune opzioni.

Opzione	Descrizione
`-s`	Produce un formato sintetico; questa modalità è predefinita.
`-l`	Questa opzione richiede l'indicazione di almeno un utente e serve a produrre informazioni dettagliate.

Segue la descrizione di alcuni esempi.

- `$` **`pinky`** [*Invio*]

```
Login  Name      TTY       Idle    When          Where
tizio            pts/125           Sep 21 10:23 host192-116:S.0
tizio            pts/126           Sep 21 10:23 host192-116:S.1
```

  Mostra l'elenco degli utenti che utilizzano il sistema.

- `$` **`pinky -l tizio`** [*Invio*]

```
Login name: tizio                       In real life:  Tizio Tizi
Directory: /home/tizio                  Shell:  /bin/sh
```

  Mostra le informazioni disponibili sull'utente '**`tizio`**'.

Si veda eventualmente la pagina di manuale *pinky(1)*.

16.2.4.5 Utilizzo di «whoami»

«

Il programma '**`whoami`**'[16] visualizza il nome dell'utente associato con l'attuale UID efficace. È equivalente a '**`id -un`**'.

```
whoami
```

Il nominativo-utente associato al numero UID efficace è in pratica l'identità con cui si sta lavorando. Per esempio, dopo l'utilizzo di '**`su`**' per diventare utenti '**`caio`**', il programma '**`whoami`**' restituisce esattamente il nome '**`caio`**'.

16.2.4.6 Utilizzo di «logname»

«

Il programma '**`logname`**'[17] emette il nome dell'utente, così come appare dal file '`/var/run/utmp`'.

```
logname
```

A titolo di esempio si può immaginare la situazione in cui l'utente '**tizio**' sia riuscito a ottenere i privilegi dell'utente '**root**' attraverso l'uso di '**su**'.

```
tizio$ su root [Invio]

Password: ******* [Invio]
```

Quello che si dovrebbe ottenere con '**logname**' è il nome dell'utente che è stato usato per accedere inizialmente al sistema.

```
root# logname [Invio]

  tizio
```

16.2.4.7 Utilizzo di «groups»

Il programma '**groups**'[18] visualizza i gruppi ai quali l'utente o gli utenti appartengono.

```
groups [utente…]
```

Il risultato è equivalente al comando seguente:

```
id -Gn [nome_utente]
```

16.2.4.8 Utilizzo di «id»

Il programma '**id**' visualizza il numero UID (*User ID*) e il numero GID (*Group ID*) reale ed efficace dell'utente selezionato o di quello corrente.[19]

`id` [*opzioni*] [*utente*]

Tabella 16.29. Alcune opzioni.

Opzione	Descrizione
`-u` `--user`	Emette solo il numero dell'utente (UID).
`-g` `--group`	Emette solo il numero del gruppo (GID).
`-G` `--groups`	Emette solo i numeri dei gruppi supplementari.
`-n` `--name`	Emette il nome dell'utente, del gruppo o dei gruppi, a seconda che sia usato insieme a '**`-u`**', '**`-g`**' o '**`-G`**'.
`-r` `--real`	Emette i numeri UID o GID reali invece di quelli efficaci (ammesso che ci sia differenza). Si usa insieme a '**`-u`**', '**`-g`**' o '**`-G`**'.

Usato senza argomenti, '**`id`**' fornisce l'identità dell'utente, il gruppo standard e l'elenco dei gruppi a cui l'utente è aggregato, come si vede dall'esempio seguente:

```
$ id[Invio]

uid=1001(tizio) gid=1001(tizio) gruppi=1001(tizio),6(disk),↩
↪7(lp),24(cdrom),25(floppy),29(audio)
```

Le opzioni servono in pratica a limitare le informazioni che si desiderano avere; eventualmente si può consultare il documento *info id*, oppure la pagina di manuale *id(1)* per maggiori dettagli su questo programma.

16.3 Parole d'ordine cifrate

In questo documento si accenna più volte al fatto che le parole d'ordine utilizzate per accedere vengono annotate in forma cifrata, nel file '`/etc/passwd`', oppure nel file '`/etc/shadow`'.

La cifratura genera una stringa che può essere usata per verificare la correttezza della parola d'ordine, mentre da sola, questa stringa non permette di determinare quale sia la parola d'ordine di origine. In pratica, data la parola d'ordine si può determinare la stringa cifrata, ma dalla stringa cifrata non si ottiene la parola d'ordine.

La verifica dell'identità avviene quindi attraverso la generazione della stringa cifrata corrispondente: se corrisponde a quanto annotato nel file '`/etc/passwd`', oppure nel file '`/etc/shadow`', la parola d'ordine è valida, altrimenti no.

16.3.1 Funzione «crypt()»

L'algoritmo usato per generare la parola d'ordine cifrata non è uguale in tutti i sistemi. Per quanto riguarda i sistemi GNU si distinguono due possibilità: l'algoritmo tradizionale DES, il quale accetta parole d'ordine con un massimo di **otto caratteri**, e l'algoritmo MD5, con cui non si pongono limitazioni.

La gestione dell'algoritmo di cifratura delle parole d'ordine è a carico della funzione ***crypt()*** (descritta in *crypt(3)*). Nelle distribuzioni

GNU in cui si può usare l'algoritmo MD5 dovrebbe essere possibile scegliere questo, o l'algoritmo precedente, attraverso un file di configurazione ('`/etc/login.defs`', descritto nella sezione ??capitolo shadow??).

Se la propria distribuzione non sembra predisposta per la cifratura MD5, è meglio non fare esperimenti, perché è importante che ogni componente del sistema di autenticazione e di gestione delle parole d'ordine sia aggiornato correttamente.

16.3.2 Trasferimento delle utenze

«

Il trasferimento, o la replicazione delle utenze si basa sulla riproduzione delle parole d'ordine cifrate, in modo tale da poter ignorare quale sia il loro valore di origine. Questa riproduzione può avvenire in modo manuale o automatico; cioè può essere l'amministratore del sistema che provvede a ricopiare le utenze, oppure può essere un servizio di rete, come il NIS (*Network information service*, noto anche come YP, *Yellow pages*).

In tutti i casi di riproduzione delle utenze, occorre che i sistemi coinvolti concordino nel funzionamento della funzione ***crypt()***, cioè generino le stesse stringhe cifrate a partire dalle parole d'ordine. Questo è il punto più delicato nella scelta di utilizzare o meno un algoritmo più sofisticato rispetto a quello tradizionale.

16.3.3 Debolezza del sistema di parole d'ordine cifrate

«

Questo sistema di autenticazione basato sulla conservazione di una parola d'ordine cifrata, ha una debolezza fondamentale: conoscendo la stringa cifrata e l'algoritmo che la genera, si può determinare la parola d'ordine originale per tentativi.[20]

Un sistema che consente l'utilizzo di parole d'ordine con un massimo di otto caratteri è molto debole ai giorni nostri, perché tutte le combinazioni possibili possono essere provate in tempi brevi, anche con un elaboratore di potenza media.

16.4 Utenti e gruppi

I nuovi utenti possono essere aggiunti solo da parte dell'utente '**root**', ma poi possono essere loro stessi a cambiare alcuni elementi della loro registrazione. Il più importante è naturalmente la parola d'ordine.

16.4.1 Utilizzo di «adduser» e «useradd»

Il programma per l'inserimento di un utente nuovo può avere due nomi alternativi: '**adduser**' o '**useradd**'. L'inserimento di un utente è consentito solo all'utente '**root**' e consiste normalmente nell'aggiunta delle voci necessarie ai file '`/etc/passwd`', '`/etc/group`' e '`/etc/shadow`', creando eventualmente anche la directory personale dell'utente stesso.

```
adduser [opzioni]
```

```
useradd [opzioni] utente
```

Se si vuole mantenere la massima compatibilità con qualunque programma che abbia qualcosa a che fare con il riconoscimento delle utente, il nome dell'utente non può superare gli otto caratteri. In ogni caso, è opportuno limitarsi all'uso di lettere non accentate e di

numeri; qualunque altro simbolo, compresi i segni di punteggiatura, potrebbero creare problemi di vario tipo.

Quando l'inserimento dell'utente implica la creazione automatica della sua directory personale, vengono copiati all'interno di questa alcuni file di configurazione standard contenuti nella directory '`/etc/skel/`'. Di conseguenza, basta porre all'interno di questa directory i file e le directory che si vogliono riprodurre nella directory personale di ogni nuovo utente.

Se nel proprio sistema GNU sono presenti entrambi questi programmi, molto probabilmente si comportano in maniera leggermente diversa. Nella distribuzione Debian, '**useradd**', senza l'indicazione di opzioni particolari, si comporta così:

```
# useradd mevio [Invio]
```

In pratica non si vede altro; l'utente '**mevio**' viene creato, inserendo dati predefiniti essenziali nel file '`/etc/passwd`' e '`/etc/shadow`', ma senza specificare la parola d'ordine e nemmeno la shell:

```
mevio:x:1002:100::/home/mevio:
```

Quello che si vede sopra è l'esempio di quanto viene aggiunto nel file '`/etc/passwd`'.

Al contrario, sempre nella distribuzione Debian, il programma '**adduser**' è più completo:

```
# adduser mevio [Invio]
```

```
  Adding user mevio...
  Adding new group mevio (1003).
  Adding new user mevio (1003) with group mevio.
  Creating home directory /home/mevio.
  Copying files from /etc/skel
Enter new UNIX password: ******* [Invio]

Retype new UNIX password: ******* [Invio]

  Changing the user information for mevio
  Enter the new value, or press ENTER for the default
Full Name []: Mevio Mevi [Invio]

Room Number []: [Invio]

Work Phone []: [Invio]

Home Phone []: [Invio]

Other []: [Invio]

Is the information correct? [y/n] y [Invio]
```

A proposito di '**adduser**' della distribuzione Debian si veda la sezione 16.8.4.

16.4.2 File «/etc/skel/*»

La directory '`/etc/skel/`' viene utilizzata normalmente come directory personale tipica per i nuovi utenti. In pratica, quando si aggiunge un nuovo utente e gli si prepara la sua directory personale, viene copiato all'interno di questa il contenuto di '`/etc/skel/`'.

Il nome *skel* sta per *skeleton*, cioè scheletro. In effetti rappresenta lo scheletro di una nuova directory personale.

È molto importante la preparazione di questa directory, in modo che ogni nuovo utente trovi subito i file di configurazione, necessari a utilizzare le shell previste nel sistema ed eventualmente altri programmi essenziali.

16.4.3 Utilizzo di «passwd»

«

Il programma '**passwd**' permette di cambiare la parola d'ordine registrata all'interno di '`/etc/passwd`' (oppure all'interno di '`/etc/shadow`', come viene mostrato in seguito).[21] Solo l'utente '**root**' può cambiare la parola d'ordine di un altro utente. Gli utenti comuni (tutti escluso '**root**') devono utilizzare una parola d'ordine non troppo breve, composta sia da maiuscole, sia da minuscole o simboli diversi. Alcune parole d'ordine simili al nome utilizzato per identificare l'utente, non sono valide.[22]

```
passwd [utente]
```

Se non si dispone di un mezzo per l'inserimento di un nuovo utente, come quello fornito da '**adduser**', è possibile aggiungere manualmente un record all'interno del file '`/etc/passwd`' senza l'indicazione della parola d'ordine che può essere specificata successivamente attraverso '**passwd**'.

Quando si inventa una nuova parola d'ordine bisogna essere sicuri di poterla introdurre in tutte le situazioni che si potrebbero presentare. Se si utilizzano lettere accentate (cosa sconsigliabile), potrebbe poi capitare di trovare un terminale che non permette il loro inserimento. In generale, conviene limitarsi a utilizzare i simboli che rientrano nella codifica ASCII a 7 bit, con la debita prudenza.

16.4.4 Utilizzo di «chsh»

Il programma '**chsh**' permette di cambiare la shell predefinita all'interno del file '`/etc/passwd`'.[23] È possibile indicare solo una shell esistente e possibilmente elencata all'interno di '`/etc/shells`'. Se la nuova shell non viene indicata tra gli argomenti, questa viene richiesta subito dopo l'avvio di '**chsh**'. Per conferma, viene richiesta anche la ripetizione della parola d'ordine.

`chsh` [*opzioni*] [*utente*]

Tabella 16.34. Alcune opzioni.

Opzione	Descrizione
`-s` ***shell*** `--shell` ***shell***	Permette di specificare la shell.
`-l` `--list-shells`	Emette un elenco delle shell disponibili in base al contenuto di '`/etc/shells`'.

16.4.5 File «/etc/shells»

«

Il file ‘`/etc/shells`’ contiene semplicemente un elenco di shell valide, cioè di quelle che sono esistenti e possono essere utilizzate. Segue un esempio di questo file.

```
/bin/sh
/bin/bash
/bin/tcsh
/bin/csh
/bin/ash
/bin/zsh
```

È molto importante che questo file sia preparato con cura e contenga solo le shell per le quali il sistema è predisposto. Ciò significa, quanto meno, che deve esistere una configurazione generalizzata per ognuna di queste e che nella directory ‘`/etc/skel/`’ devono essere stati predisposti tutti i file di configurazione personalizzabili che sono necessari. Quindi, un file ‘`/etc/shells`’ con un semplice elenco di tutte le shell disponibili non è sufficiente.

16.4.6 Utilizzo di «chfn»

«

Il programma ‘**`chfn`**’ consente di modificare le informazioni personali registrate all’interno del file ‘`/etc/passwd`’. [24] Si tratta in pratica del nome e cognome dell’utente, del numero dell’ufficio, del telefono dell’ufficio e del telefono di casa. Se non vengono specificate opzioni, i dati vengono inseriti in maniera interattiva, se non viene specificato l’utente, si intende quello che ha eseguito il comando. Solo l’utente ‘**`root`**’ può cambiare le informazioni di un altro utente.

```
chfn [opzioni] [utente]
```

Le informazioni indicate nel quinto capo dei record del file '`/etc/passwd`', sono strutturate solo in modo convenzionale, senza che esista una necessità effettiva.

L'esempio seguente mostra le azioni compiute da un utente per definire le proprie informazioni personali.

```
tizio$ chfn[Invio]

  Changing finger information for tizio
Password: ********[Invio]

Name [tizio]: Tizio Tizi[Invio]

Office []: Riparazioni[Invio]

Office Phone[]: 123456[Invio]

Home Phone[]: 9876543[Invio]

  Finger information changed.
```

Volendo verificare il risultato all'interno del file '`/etc/passwd`', si può trovare il record seguente, che appare suddiviso su due righe per la mancanza di spazio:

```
tizio:724AD9dGbG25k:502:502:↩
↪Tizio Tizi,Riparazioni,123456,987654:↩
↪/home/tizio:/bin/bash
```

Le informazioni personali possono essere delicate, specialmente quando si tratta di indicare il numero telefonico dell'abitazione di un utente. Per questo, quando si tratta di utenze presso elaboratori raggiungibili attraverso una rete estesa, come Internet, occorre prudenza.

16.5 Utenti e gruppi importanti

Osservando il file '`/etc/passwd`' si possono notare diversi utenti fittizi standard che hanno degli scopi particolari. Si tratta di ***utenti di sistema***, nel senso che servono al buon funzionamento del sistema operativo.

```
root:dxdFf9MvQ3s:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/lib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
nobody:*:99:99:Nobody:/:
```

Di conseguenza, anche '`/etc/group`' contiene l'indicazione di gruppi particolari (gruppi di sistema).

```
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:
disk::6:root
lp::7:daemon,lp
mem::8:
kmem::9:
wheel::10:root
mail::12:mail
news::13:news
uucp::14:uucp
man::15:
games::20:
gopher::30:
dip::40:
ftp::50:
nobody::99:
users::100:
```

I campi delle parole d'ordine di questi utenti speciali (tutti tranne '**root**') hanno un asterisco che di fatto impedisce qualunque accesso.

Le varie distribuzioni GNU si distinguono spesso nella quantità e nell'organizzazione degli utenti e dei gruppi fittizi. In questo caso, in particolare, l'utente fittizio '**nobody**' ha il numero UID 99, come definito nella distribuzione GNU/Linux Red Hat. In generale, questo utente potrebbe avere il numero −1, che applicandosi a un intero positivo rappresenta in pratica il numero più alto gestibile di UID, altre volte potrebbe essere il numero −2. Il numero massimo di UID dipende dalle caratteristiche del file system e dalle librerie utilizzate.

Tabella 16.41. Alcuni utenti e gruppi importanti.

Opzione	Descrizione
'**root**'	L'utente '**root**' è l'amministratore del sistema: ogni sistema Unix ha un utente '**root**'. L'utente '**root**' ha sempre il numero UID pari a zero.
'**bin**'	L'utente '**bin**' non esiste nella realtà. Si tratta di un nome fittizio definito per assegnare ai file eseguibili (*binary*) un proprietario diverso dall'utente '**root**'. Di solito, con GNU/Linux, questi eseguibili appartengono al gruppo '**bin**', mentre l'utente proprietario resta '**root**'.
'**tty**'	Di solito, al gruppo '**tty**' appartengono i file di dispositivo utilizzabili come canali per la connessione di un terminale.
'**disk**'	Di solito, al gruppo '**disk**' appartengono i file di dispositivo che si riferiscono a unità a dischi, compresi CD, DVD e assimilati.
'**floppy**'	Di solito, al gruppo '**floppy**' appartengono i file di dispositivo che si riferiscono alle unità a dischetti.

Opzione	Descrizione
‘**nobody**’	L’utente ‘**nobody**’ corrisponde in linea di massima a un utente generico, non identificato, senza privilegi particolari. Viene usato in particolare per evitare che un utente ‘**root**’ possa accedere a un file system di rete (NFS) mantenendo i suoi privilegi: quando ciò accade, l’elaboratore che offre il servizio NFS lo tratta come utente ‘**nobody**’. In generale, ‘**nobody**’ non deve essere utilizzabile per l’accesso umano. A seconda della distribuzione GNU che si utilizza, il gruppo abbinato a questo utente potrebbe chiamarsi ‘**nobody**’, oppure anche ‘**nogroup**’.

16.6 Eliminazione di un utente

L’eliminazione di un utente dal sistema non è gestibile attraverso un programma di servizio standard di uso generale: la particolare distribuzione GNU può fornire degli strumenti adatti, oppure si deve agire manualmente. In questa sezione si descrive come si può intervenire manualmente. Fondamentalmente si tratta di agire su due punti:

- l’eliminazione dell’utente dai file ‘`/etc/passwd`’ e ‘`/etc/group`’ (ed eventualmente anche da ‘`/etc/shadow`’);
- l’eliminazione dei file appartenenti a quell’utente.

I file di un utente possono trovarsi ovunque gli sia stato consentito di scriverli. In particolare:

- la directory personale;

- la directory delle caselle postali (‘`/var/mail/`’ o in certi casi ‘`/var/spool/mail/`’, a meno che questa non sia già inserita direttamente nelle directory personali);
- la directory ‘`/var/spool/cron/crontabs/`’ e ‘`/var/spool/cron/atjobs/`’ per eventuali applicazioni a esecuzione pianificata.

Per elencare tutti i file appartenenti a un certo utente, è possibile usare il programma Find in uno dei modi seguenti.

```
find / -uid numero_utente -print
```

```
find / -user utente -print
```

Volendo, si potrebbe costruire uno script per l’eliminazione automatica di tutti i file appartenenti a un utente determinato. L’esempio seguente, prima di eliminare i file, crea una copia compressa.

```
#!/bin/sh
# eliminautente

# Il nome dell'utente viene fornito come primo e unico
# argomento di questo script.
NOME_UTENTE="$1"

# Nome per un file temporaneo contenente l'elenco dei file
# appartenenti all'utente che si vuole eliminare.
ELENCO_FILE_UTENTE=`tempfile`

# Visualizza la sintassi corretta per l'utilizzo di questo
# script.
function sintassi () {
```

```
    echo ""
    echo "eliminautente <nome-utente>"
}

# Inizio: verifica la quantità di argomenti.
if [ $# != 1 ]
then
    # La quantità di argomenti è errata. Richiama la
    # funzione «sintassi» e termina l'esecuzione dello
    # script restituendo un valore corrispondente a «falso».
    sintassi
    exit 1
fi

# Verifica che l'utente sia root.
if [ $UID != 0 ]
then
    # Dal momento che l'utente non è root, avvisa
    # dell'errore e termina l'esecuzione restituendo un
    # valore corrispondente a «falso».
    printf "Questo script può essere utilizzato "
    printf "solo dall'utente root."
    printf "\n"
    exit 1
fi

# Crea un elenco di tutti i file appartenenti all'utente
# specificato. Si deve evitare che find cerchi di entrare
# nella directory /dev/.
find / -user $NOME_UTENTE -a \
    \( -path "/dev" -prune -o -print \) \
    > $ELENCO_FILE_UTENTE

# Comprime i file generando un file compresso con lo stesso
```

```
# nome dell'utente da eliminare e con estensione «.tgz».
# Si utilizza «tar» e in particolare:
# «z» permette di comprimere automaticamente l'archivio
# attraverso «gzip».
# In questo caso, l'archivio viene generato nella directory
# personale dell'amministratore del sistema.
if tar czvf ~/$NOME_UTENTE.tgz `cat $ELENCO_FILE_UTENTE`
then
    # Se è andato tutto bene elimina i file
    # (togliere il commento), quindi elimina l'elenco
    # temporaneo.
    #rm `cat $ELENCO_FILE_UTENTE`
    rm $ELENCO_FILE_UTENTE
fi
```

16.7 Trucchi per la gestione degli utenti

«

Alcuni accorgimenti nella gestione degli utenti e dei gruppi possono essere utili in situazioni particolari, anche se a volte si tratta di scelte discutibili. Nelle sezioni seguenti se ne descrivono alcuni.

16.7.1 Utente con funzione specifica

«

Un trucco che potrebbe rivelarsi comodo in certe situazioni è quello di creare un utente fittizio, con o senza parola d'ordine, al quale si associa un programma o uno script, al posto di una shell. La directory corrente nel momento in cui il programma o lo script viene eseguito è quella indicata come directory *home* (directory personale).

L'esempio seguente mostra un record del file '`/etc/passwd`' preparato in modo da permettere a chiunque di eseguire il programma (o lo script) '`/usr/local/bin/ciao`' partendo dalla posizione

della directory ‘`/tmp/`’. Il numero UID 505 e GID 100 sono solo un esempio.

```
ciao::505:100:Ciao a tutti:/tmp:/usr/local/bin/ciao
```

Naturalmente, il fatto di poter avere un utente (reale o fittizio) che possa accedere senza parola d’ordine, dipende dal sistema di autenticazione: il programma ‘**`login`**’, il quale potrebbe essere stato configurato (o predisposto all’atto della compilazione) per vietare un tale comportamento.

16.7.2 Gruppo di utenti con lo stesso UID

All’interno di un ambiente in cui esiste una certa fiducia nel comportamento reciproco, potrebbe essere conveniente creare un gruppo di utenti con lo stesso numero UID.

Ogni utente avrebbe un proprio nome e una parola d’ordine per accedere al sistema, ma poi, tutti i file apparterrebbero a un utente immaginario che rappresenta tutto il gruppo. Segue un esempio del file ‘`/etc/passwd`’.

```
tutti:*:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
alfa:34gdf6r123455:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
bravo:e445gsdfr2124:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
charlie:t654df7u72341:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
tutti:*:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
```

Si osservi che l’utente fittizio ‘**`tutti`**’ dell’esempio mostrato appare per primo e per ultimo, in modo da non avere differenze di comportamento in presenza di un sistema NIS per la condivisione delle utenze in rete.

Se esiste la necessità o l’utilità si possono assegnare anche directory personali e shell differenti.

16.7.3 Uno stesso UID e GID per più nominativi-utente

«

Un utente reale potrebbe avere bisogno di gestire diversi nominativi-utente per accedere allo stesso elaboratore e gestire attività differenti, pur mantenendo lo stesso numero UID e lo stesso numero GID. In questo modo, avrebbe a disposizione diverse directory personali, una per ogni progetto che conduce.

```
tizio:34gdf6r123455:1000:1000:Tizio Tizi:/home/tizio:/bin/sh
alfa:34gdf6r123455:1000:1000:Tizio Tizi prog. Alfa:/home/alfa:/bin/sh
bravo:34gdf6r123455:1000:1000:Tizio Tizi prog. Bravo:/home/bravo:/bin/sh
charlie:34gdf6r123455:1000:1000:Tizio Tizi prog. Charlie:/home/charlie:/bin/sh
tizio:34gdf6r123455:1000:1000:Tizio Tizi:/home/tizio:/bin/sh
```

Si osservi che la dichiarazione dell'utente '**tizio**' dell'esempio mostrato appare per primo e per ultimo, in modo da non avere differenze di comportamento in presenza di un sistema NIS per la condivisione delle utenze in rete.

Eventualmente, per distinguere quale sia il nominativo-utente utilizzato effettivamente, si potrebbe modificare la stringa di definizione dell'invito della shell. Nel caso di Bash, si potrebbe utilizzare quella seguente:

```
PS1="$USER->\u@\h:\w\\$ "
export PS1
```

Il significato di questo viene approfondito nel capitolo dedicato alla shell POSIX, in cui vengono annotate anche alcune particolarità di Bash (17).

16.7.4 Un gruppo per ogni utente (gruppi privati)

«

Si tratta di una strategia di gestione degli utenti e dei gruppi con cui, ogni volta che si crea un nuovo utente, si crea anche un gruppo con lo stesso nome e, possibilmente, lo stesso numero (UID = GID). Questa

tecnica si combina con una maschera dei permessi 002_8. In pratica, i file vengono creati in modo predefinito con i permessi di lettura e scrittura, sia per l'utente proprietario, sia per il gruppo, mentre si esclude la scrittura per gli altri utenti.

Il motivo di tutto questo sta nella facilità con cui si può concedere a un altro utente di poter partecipare al proprio lavoro: basta aggiungere il suo nome nell'elenco degli utenti associati al proprio gruppo.

Volendo agire in maniera più elegante, si possono creare degli altri gruppi aggiuntivi, in base alle attività comuni e aggiungere a questi gruppi i nomi degli utenti che di volta in volta partecipano a quelle attività. Naturalmente, i file da condividere all'interno dei gruppi devono appartenere a questi stessi gruppi.[25]

A titolo di esempio, si mostra cosa sia necessario fare per gestire un gruppo di lavoro per un ipotetico progetto «alfa».

1. Si fa in modo che la maschera dei permessi predefiniti (*umask*) degli utenti che devono far parte del progetto, sia pari a 002_8, per consentire in modo normale ogni tipo di accesso agli utenti dei gruppi di cui si fa parte, ai file e alle directory che vengono create.
2. Si crea il gruppo '**alfa**' e a questo si abbinano tutti gli utenti che devono fare parte del progetto. Il record del gruppo '**alfa**', nel file '`/etc/group`', potrebbe essere simile a quello seguente:

   ```
   alfa::101:tizio,caio,sempronio
   ```

3. Si crea una sorta di directory *home* per i file del progetto, con eventuali ramificazioni.

```
# mkdir /home/progetti/alfa [Invio]

# mkdir /home/progetti/alfa/... [Invio]
```

4. Si assegna l'appartenenza di questa directory (ed eventuali sottodirectory) al gruppo di lavoro.

   ```
   # chown -R root:alfa /home/gruppi/alfa [Invio]
   ```

5. Si assegnano i permessi in modo che ciò che viene creato all'interno del gruppo di directory appartenga al gruppo delle directory stesse.

   ```
   # chmod -R 2775 /home/progetti/alfa [Invio]
   ```

   In questo modo tutte le directory del progetto ottengono l'attivazione del bit SGID, attraverso il quale, in modo predefinito, i file creati al loro interno vengono abbinati allo stesso gruppo delle directory stesse, cioè quello del progetto per cui sono state predisposte.

16.7.5 Directory personali controllate rigidamente dall'amministratore

«

Quando si gestiscono i gruppi privati, ovvero quando ogni utente ha un proprio gruppo personale, e quando il bit Sticky dei permessi sulle directory comporta l'impossibilità di cancellare ciò che non appartiene all'utente stesso, è possibile attuare un controllo abbastanza stretto sulle directory personali degli utenti. Il meccanismo si basa su un principio molto semplice: le directory personali appartengono all'utente '**root**' e solo al gruppo del vero utente destinatario, con la facoltà di lettura, scrittura e accesso per il gruppo, ma con il bit Sticky attivo. Per esempio, supponendo che la directory personale dell'utente '**tizio**' sia '`/home/tizio`':

```
# chown root /home/tizio[Invio]

# chmod ug+rwx /home/tizio[Invio]

# chmod +t /home/tizio[Invio]

# ls -ld /home/tizio[Invio]

  drwxrwxr-t 2 root tizio 4096 2009-12-15 18:20 /home/tizio
```

In queste condizioni, l'utente '**tizio**' può agire quasi normalmente nella propria directory personale, con la differenza che non può cancellare i file che non gli appartengono e non può cambiare i permessi della propria stessa directory personale.

Il primo vantaggio per l'amministratore consiste nel poter controllare i permessi nelle directory personali degli utenti; per esempio può impedire loro che condividere delle informazioni, togliendo il permesso di lettura o di attraversamento per gli altri utenti.

L'amministratore può inoltre creare dei file o delle sottodirectory, appartenenti a sé, sulle quali controllare ulteriormente i contenuti e i permessi di accesso. Per esempio potrebbe impedire ad alcuni utenti di pubblicare delle informazioni nella directory '`~/public_html/`', creandone una vuota, ma priva dei permessi di accesso al gruppo e agli altri utenti (gli utenti non avrebbero modo di eliminarla).

L'amministratore potrebbe predisporre una configurazione legata alle singole utenze, composta da directory e file, contenuti nelle directory personali degli stessi, togliendo agli utenti la facoltà di intervenirvi. Per esempio potrebbe modificare il file '`/etc/profile`' (della shell standard), per verificare la presenza e il contenuto di cer-

ti file nella directory personale dell'utente che accede, controllando di conseguenza delle azioni preventive.

16.7.6 Directory personali controllate da un utente «tutore»

«

Un meccanismo molto simile a quello descritto nella sezione precedente, consente di attribuire a un utente, definibile come «tutore», il ruolo di controllo delle directory personali di un certo insieme di altri utenti «pupilli». Per comprendere il senso si ciò si può immaginare che il tutore sia un insegnante, a cui viene affidato il controllo delle utenze che rappresentano gli studenti di propria competenza. Supponendo che l'utente '**tizio**' sia il pupillo e che l'utente '**martino**' sia il tutore della situazione, l'esempio seguente mostra il processo di acquisizione del controllo dell'utenza di '**tizio**' nel modo descritto:

```
# chown martino /home/tizio [Invio]

# chmod ug+rwx /home/tizio [Invio]

# chmod +t /home/tizio [Invio]

# ls -ld /home/tizio [Invio]

drwxrwxr-t 2 martino tizio 4096 2009-12-15 18:20 /home/tizio
```

In questo caso, il vantaggio più importante consiste nella facoltà che guadagna l'utente «tutore» di accedere alle directory personali dei propri «pupilli», anche quando si volesse impedire agli altri utenti di accedervi, oltre che di poter controllare i permessi e l'inamovibilità di certi file e directory.

È bene chiarire che gli utenti tutori possono ottenere questo ruolo particolare solo in base a un intervento dell'amministratore, senza

il quale non si potrebbero assegnare inizialmente le proprietà alle directory personali dei pupilli.

16.7.7 Creazione di un'utenza in più contesti

Può capitare la necessità di creare un proprio script per la creazione delle utenze, per esempio quando si vuole concedere una forma di accesso alternativa, che prevede però un proprio meccanismo di autenticazione e di memorizzazione delle parole d'ordine. Questo problema può capitare per esempio con Samba, anche se teoricamente è già previsto un meccanismo del genere al suo interno.

Volendo o dovendo creare un tale script, il problema che si incontra sta nell'assegnare o modificare la parola d'ordine dell'utente. Per farlo, si può cercare di controllare il programma '**passwd**' tramite lo script, ma bisogna dare il tempo a '**passwd**' di fare le domande:

```
# passwd tizio[Invio]

Enter new UNIX password: digitazione_all'oscuro[Invio]

Retype new UNIX password: digitazione_all'oscuro[Invio]

  passwd: password updated successfully.
```

Per poter inviare al programma '**passwd**' la parola d'ordine (si supponga che si tratti della parola «segreta»), non la si può inserire in un file del genere per poi inviarla attraverso lo standard input:

```
segreta
segreta
```

Supponendo che questo sia il file '/tmp/nuova' non serve a nulla fare così:

```
# cat /tmp/nuova | passwd tizio[Invio]
```

Ecco cosa si vedrebbe apparire:

```
Enter new UNIX password: Retype new UNIX password: Sorry, ↩
↪passwords do not match↩
↪passwd: Authentication information cannot be recovered
```

Per risolvere il problema, va inserita una pausa nel flusso dello standard input. Ecco come:

```
# ( sleep 1 ; echo "segreta" ; sleep 1 ; echo "segreta" ) ↩
↪| passwd tizio[Invio]

Enter new UNIX password: Retype new UNIX password: passwd: ↩
↪password updated successfully
```

16.8 Parole d'ordine oscurate

«

Il meccanismo delle parole d'ordine oscurate (*shadow password*) si basa su un principio molto semplice: nascondere le parole d'ordine cifrate ai processi che non hanno i privilegi dell'utente '**root**'. Infatti, nei sistemi in cui le parole d'ordine oscurate non sono attivate, è il file '`/etc/passwd`', leggibile a tutti i tipi di utenti, che contiene tali parole d'ordine cifrate.

Il problema nasce dal fatto che è possibile scoprire la parola d'ordine degli utenti attraverso programmi specializzati che scandiscono un vocabolario alla ricerca di una parola che possa corrispondere alla parola d'ordine cifrata.

L'utilizzo del sistema delle parole d'ordine oscurate richiede che alcuni programmi siano predisposti per questo. Nel capitolo si fa riferimento a strumenti standard che però si intende siano stati integrati nella distribuzione GNU che si utilizza. L'attivazione di parole d'ordine oscurate in una distribuzione che non sia stata predisposta, comporta delle difficoltà che rendono la cosa sconsigliabile.

16.8.1 Funzioni delle parole d'ordine oscurate

Con le parole d'ordine oscurate attivate si aggiunge il file '`/etc/shadow`' a fianco del consueto '`/etc/passwd`'. In questo secondo file vengono tolte le parole d'ordine cifrate e al loro posto viene inserita una '**x**', mentre nel file '`/etc/shadow`', oltre alle parole d'ordine cifrate, vengono inserite altre informazioni sulle utenze che permettono di aumentare la sicurezza.

Anche i gruppi possono avere delle parole d'ordine ed è possibile affiancare al file '`/etc/group`' il file '`/etc/gshadow`'.

16.8.1.1 File «/etc/shadow»

La presenza del file '`/etc/shadow`' indica l'attivazione delle parole d'ordine oscurate. I record di questo file (le righe) sono organizzati in campi, separati attraverso il simbolo due punti ('**:**'), secondo la sintassi seguente:

utente **:** ***parola_d'ordine_cifrata*** **:** ***modifica*** **:** ***valid_min*** **:** ***valid_max*** **:** ***preavviso*** **:** ↩
↪***tempo_riserva*** **:** ***termine*** **:** ***riservato***

I campi che rappresentano una data possono contenere un numero intero che indica la quantità di giorni trascorsi dal 1/1/1970, mentre quelli che rappresentano una durata, possono contenere un numero intero che esprime una quantità di giorni.

1. ***utente***

   Il nominativo dell'utente.

2. ***parola_d'ordine_cifrata***

   La parola d'ordine cifrata, tolta dal file '`/etc/passwd`'.

3. ***modifica***

   Data in cui è stata modificata la parola d'ordine per l'ultima volta.

4. ***validità_minima***

   Numero di giorni di validità minima della parola d'ordine; entro questo tempo, l'utente non può cambiare la parola d'ordine.

5. ***validità_massima***

   Numero di giorni di validità massima della parola d'ordine; prima che trascorra questo tempo, l'utente deve cambiare la parola d'ordine.

6. ***preavviso***

   Numero di giorni, prima della scadenza della parola d'ordine, durante i quali l'utente viene avvisato della necessità di modificarla.

7. ***tempo_di_riserva***

   Durata massima di validità dell'utenza dopo che la parola d'ordine è scaduta.

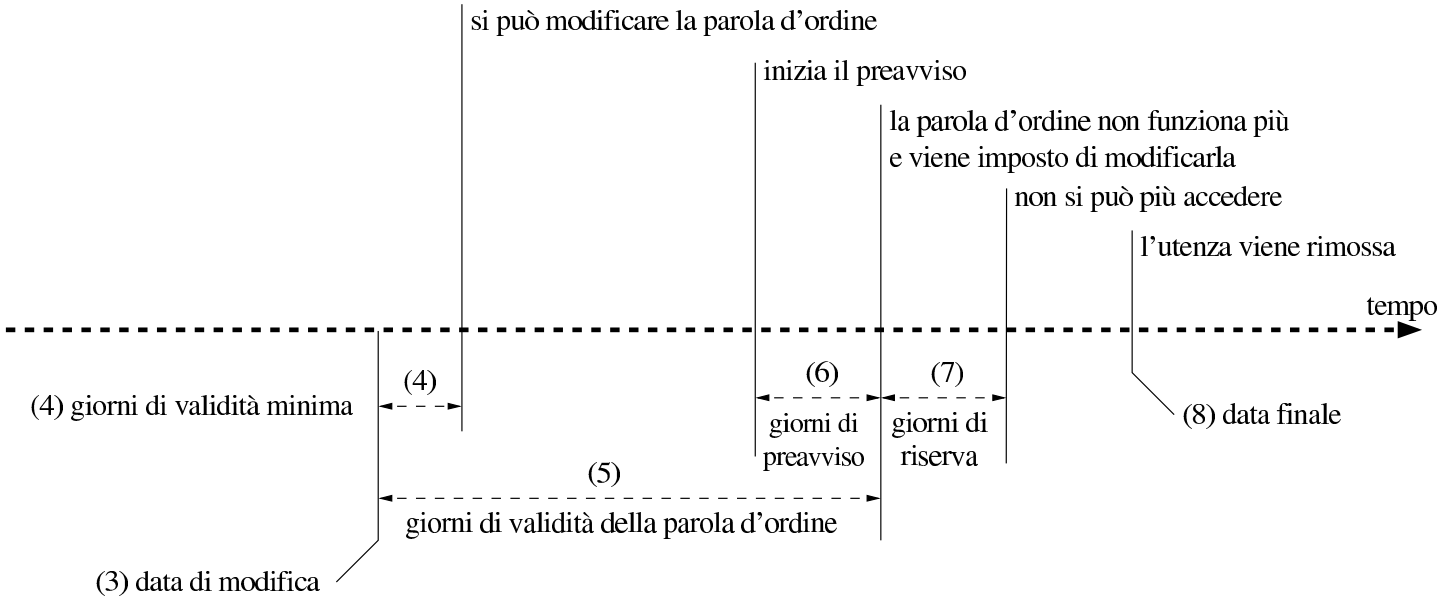
8. ***termine***

   Data di scadenza dell'utenza.

9. ***riservato***

   Riservato per usi futuri.

Figura 16.54. Schema dei tempi riferiti alle utenze. I numeri fanno riferimento ai campi del file '`/etc/shadow`'.



A titolo di esempio, viene mostrato un caso sufficientemente completo nella figura 16.55.

Figura 16.55. Esempio di un record del file '`/etc/shadow`'.

```
tizio:wsLHjp.FutW0sgtgeR67gdehk98:10459:0:30:7:10:10824:
 |     |                            |    | |  | |   |
 |     parola d'ordine              |    | |  | | termine
 |     cifrata                      |    | |  | | dell'utenza
 |                                  |    | |  | | il giorno
 utente                             |    | |  | | 21/08/1999
         la parola d'ordine è stata      | |  | |
         modificata il giorno            | |  | dieci giorni
         21/08/1988                      | |  | di riserva
                                         | |  |
          l'utente può cambiare password | |  sette giorni di
          in ogni momento                |  preavviso
                                         |
                                        trenta giorni di
                                        validità massima
```

Perché il sistema delle parole d'ordine oscurate possa dare la sicurezza che promette, è necessario che il file '`/etc/shadow`' appartenga all'utente '**`root`**' e abbia esclusivamente il permesso di lettura per il proprietario (0400_8).

16.8.1.2 File «/etc/passwd»

«

Quando è attivo il sistema delle parole d'ordine oscurate, il file '`/etc/passwd`' non dovrebbe contenere più le parole d'ordine cifrate. Al loro posto dovrebbe apparire una lettera '**`x`**' (minuscola), come nell'esempio seguente:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
…
tizio:x:1000:1000::/home/tizio:/bin/bash
…
```

Tuttavia, dovrebbe essere ammissibile la presenza di record contenenti la parola d'ordine cifrata dell'utente relativo, con la corrispondente assenza di un record nel file '`/etc/shadow`'. Per questi utenti, le funzionalità delle parole d'ordine oscurate sono ovviamente disattivate e non dovrebbero esserci altre conseguenze.

16.8.2 Amministrazione degli utenti

«

La presenza delle parole d'ordine oscurate richiede strumenti adeguati alla loro amministrazione. Le informazioni aggiuntive che richiede un'utenza quando sono attive le parole d'ordine oscurate, rendono utile la presenza di un file di configurazione contenente le ca-

ratteristiche predefinite che questa utenza dovrebbe avere. Il file in questione è '`/etc/login.defs`'.

16.8.2.1 File «/etc/login.defs»

Il file '`/etc/login.defs`' permette di stabilire alcune caratteristiche predefinite delle utenze che utilizzano le parole d'ordine oscurate. La sua presenza è importante soprattutto nel momento della creazione di un nuovo utente, ovvero della trasformazione di utenze normali in utenze munite di parole d'ordine oscurate, per definire i valori relativi alla validità e alla scadenza delle parole d'ordine.

Il file si compone di righe, in cui, ciò che inizia con il simbolo '**#**' viene considerato un commento, le righe vuote vengono ignorate e il resto compone le direttive di configurazione. La sintassi di queste è molto semplice: ogni direttiva occupa una sola riga e si compone di coppie ***nome valore***, spaziate, senza simboli di assegnamento.

I valori che possono essere attribuiti sono di tre tipi: stringa, numerico e logico (booleano). Le stringhe vengono indicate senza delimitatori di alcun tipo; i valori numerici possono essere di tipo decimale, ottale (e in tal caso iniziano con uno zero) ed esadecimale (quando iniziano con la sigla '**0x**'); i valori booleani sono indicati attraverso le costanti '**yes**' (*Vero*) e '**no**' (*Falso*).

Segue un estratto di esempio, derivante da una distribuzione GNU/Linux Debian.

```
# Enable logging and display of /var/log/faillog login failure info.
# This option conflicts with the pam_tally PAM module.
FAILLOG_ENAB            yes

# Enable display of unknown usernames when login failures are recorded.
LOG_UNKFAIL_ENAB        no
```

```
# Enable logging of successful logins
LOG_OK_LOGINS           no

# Enable "syslog" logging of su activity - in addition to sulog file logging.
# SYSLOG_SG_ENAB does the same for newgrp and sg.
SYSLOG_SU_ENAB          yes
SYSLOG_SG_ENAB          yes

# If defined, all su activity is logged to this file.
#SULOG_FILE     /var/log/sulog

# If defined, login failures will be logged here in a utmp format
# last, when invoked as lastb, will read /var/log/btmp, so...
FTMP_FILE       /var/log/btmp

# If defined, the command name to display when running "su -".  For
# example, if this is defined as "su" then a "ps" will display the
# command is "-su".  If not defined, then "ps" would display the
# name of the shell actually being run, e.g. something like "-sh".
SU_NAME         su

# If defined, file which inhibits all the usual chatter during the login
# sequence.  If a full pathname, then hushed mode will be enabled if the
# user's name or shell are found in the file.  If not a full pathname, then
# hushed mode will be enabled if the file exists in the user's home directory.
HUSHLOGIN_FILE  .hushlogin

# *REQUIRED*  The default minimal PATH settings, for superuser and normal users.
ENV_SUPATH      PATH=/etc/script:/usr/local/sbin:/usr/local/bin:↩
↪/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
ENV_PATH        PATH=/etc/script:/usr/local/bin:/usr/bin:/bin:↩
↪/usr/bin/X11:/usr/games

# Terminal permissions
#
#       TTYGROUP        Login tty will be assigned this group ownership.
#       TTYPERM         Login tty will be set to this permission.
#
# If you have a "write" program which is "setgid" to a special group
# which owns the terminals, define TTYGROUP to the group number and
# TTYPERM to 0620.  Otherwise leave TTYGROUP commented out and assign
# TTYPERM to either 622 or 600.
#
# In Debian /usr/bin/bsd-write or similar programs are setgid tty
```

```
# However, the default and recommended value for TTYPERM is still 0600
# to not allow anyone to write to anyone else console or terminal
# Users can still allow other people to write them by issuing
# the "mesg y" command.
TTYGROUP        tty
TTYPERM         0600

# Login configuration initializations:
#
#       ERASECHAR       Terminal ERASE character ('\010' = backspace).
#       KILLCHAR        Terminal KILL character ('\025' = CTRL/U).
#       UMASK           Default "umask" value.
#
# The ERASECHAR and KILLCHAR are used only on System V machines.
#
# UMASK usage is discouraged because it catches only some classes of user
# entries to system, in fact only those made through login(1), while setting
# umask in shell rc file will catch also logins through su, cron, ssh etc.
#
# At the same time, using shell rc to set umask won't catch entries which use
# non-shell executables in place of login shell, like /usr/sbin/pppd for "ppp"
# user and alike.
#
# Therefore the use of pam_umask is recommended as the solution which
# catches all these cases on PAM-enabled systems.
#
# This avoids the confusion created by having the umask set
# in two different places -- in login.defs and shell rc files (i.e.
# /etc/profile).
#
# For discussion, see #314539 and #248150 as well as the thread starting at
# http://lists.debian.org/debian-devel/2005/06/msg01598.html
#
# Prefix these values with "0" to get octal, "0x" to get hexadecimal.
ERASECHAR       0177
KILLCHAR        025
# 022 is the "historical" value in Debian for UMASK when it was used
# 027, or even 077, could be considered better for privacy.
# There is no One True Answer here: each sysadmin must make up his/her
# mind.
UMASK           022

# Password aging controls:
#
```

```
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7

# Min/max values for automatic uid selection in useradd
UID_MIN                 1000
UID_MAX                60000
# System accounts
#SYS_UID_MIN              100
#SYS_UID_MAX              999

# Min/max values for automatic gid selection in groupadd
GID_MIN                  100
GID_MAX                60000
# System accounts
#SYS_GID_MIN              100
#SYS_GID_MAX              999

# Max number of login retries if password is bad. This will most likely be
# overriden by PAM, since the default pam_unix module has it's own built
# in of 3 retries. However, this is a safe fallback in case you are using
# an authentication module that does not enforce PAM_MAXTRIES.
LOGIN_RETRIES           5

# Max time in seconds for login
LOGIN_TIMEOUT           60

# Which fields may be changed by regular users using chfn - use
# any combination of letters "frwh" (full name, room number, work
# phone, home phone).  If not defined, no changes are allowed.
# For backward compatibility, "yes" = "rwh" and "no" = "frwh".
CHFN_RESTRICT           rwh

# Should login be allowed if we can't cd to the home directory?
# Default in no.
DEFAULT_HOME    yes

# This enables userdel to remove user groups if no members exist.
# Other former uses of this variable such as setting the umask when
# user==primary group are not used in PAM environments, thus in Debian
USERGROUPS_ENAB yes
```

```
# If defined, either full pathname of a file containing device names or
# a ":" delimited list of device names.  Root logins will be allowed only
# upon these devices. This variable is used by login and su.
#CONSOLE        /etc/consoles
#CONSOLE        console:tty01:tty02:tty03:tty04

# List of groups to add to the user's supplementary group set
# when logging in on the console (as determined by the CONSOLE
# setting).  Default is none.
#
# Use with caution - it is possible for users to gain permanent
# access to these groups, even when not logged in on the console.
# How to do it is left as an exercise for the reader...
# This variable is used by login and su.
#CONSOLE_GROUPS         floppy:audio:cdrom

# If set to MD5 , MD5-based algorithm will be used for encrypting password
# If set to SHA256, SHA256-based algorithm will be used for encrypting password
# If set to SHA512, SHA512-based algorithm will be used for encrypting password
# If set to DES, DES-based algorithm will be used for encrypting password
#    (default)
#
# Note: It is recommended to use a value consistent with
# the PAM modules configuration.
#
#ENCRYPT_METHOD DES
```

Per quanto riguarda il problema particolare delle parole d'ordine oscurate, si possono osservare le direttive '**PASS_MAX_DAYS**', '**PASS_MIN_DAYS**' e '**PASS_WARN_AGE**'. La prima permette di stabilire la durata massima, predefinita, di validità di una parola d'ordine; la seconda serve a stabilire la durata minima; la terza il periodo di preavviso.

Tra una distribuzione GNU e l'altra, questo file può contenere o meno determinate direttive. In particolare, quando è attiva la gestione del sistema di autenticazione PAM, alcune direttive perdono di significato, perché riguardano aspetti che passano sotto il controllo della configurazione dei servizi di autenticazione PAM.

La descrizione dettagliata di alcune delle direttive può essere utile, anche se queste non hanno effetto in tutte le distribuzioni GNU.

Tabella 16.58. Alcune direttive.

Direttiva	Descrizione
`CHFN_RESTRICT` ↩ ↪ `[f] [r] [w] [h]`	Per consentire all'utente di modificare i propri dati personali, è necessario utilizzare questa direttiva. Attraverso la stringa che può contenere le lettere '**f**', '**r**', '**w**' e '**h**', si possono indicare quali elementi ha diritto di modificare l'utente: '**f**', *full name*, nome e cognome; '**r**', *room*, numero della stanza; '**w**', *work*, telefono dell'ufficio (di lavoro); '**h**', *home*, numero telefonico di casa.
`CONSOLE` ↩ ↪ { ***file*** ↩ ↪ | ***elenco_dispositivi_console*** }	Permette di definire quali siano i terminali da cui può accedere l'utente '**root**', attraverso l'indicazione di un file che solitamente è '`/etc/securetty`', oppure attraverso un elenco di nomi di file di dispositivo (senza l'indicazione della directory '`/dev/`'), separati da due punti verticali: '**console:tty01:tty02:tty03:tty04**'. Se è attiva la gestione del sistema di autenticazione PAM, questa direttiva non serve perché rimpiazzata dal modulo '`pam_securetty.so`'.

Direttiva	Descrizione
`DEFAULT_HOME {yes|no}`	Se si assegna il valore '**yes**', si intende permettere l'accesso anche se non risulta possibile entrare nella directory personale dell'utente (perché non esiste, perché i permessi non sono corretti, ecc.). Se non viene indicata questa direttiva, il valore predefinito è (o dovrebbe essere) '**no**'.
`GID_MIN` ***n_gid_minimo*** `GID_MAX` ***n_gid_massimo***	Queste due direttive permettono rispettivamente di definire il valore minimo e quello massimo per i numeri GID, cioè quelli che vengono utilizzati per distinguere i gruppi di utenti.
`UID_MIN` ***n_uid_minimo*** `UID_MAX` ***n_uid_massimo***	Queste due direttive permettono rispettivamente di definire il valore minimo e quello massimo per i numeri UID, cioè quelli che vengono utilizzati per distinguere gli utenti.
`LOGIN_RETRIES` ***n_tentativi***	Permette di definire un numero massimo di tentativi che possono essere compiuti dall'utente che cerca di accedere, a seguito di errori nella combinazione tra nominativo e parola d'ordine. Esauriti i tentativi a disposizione, il programma '**login**' dovrebbe terminare il suo funzionamento, anche se poi, di solito, viene riavviata una nuova copia del programma Getty.
`LOGIN_TIMEOUT` ***n_secondi***	Stabilisce un tempo massimo per completare la procedura di accesso, dopo il quale il programma '**login**' conclude il suo funzionamento.

Direttiva	Descrizione
`PASS_MIN_DAYS` ***n_giorni*** `PASS_MAX_DAYS` ***n_giorni***	Queste due direttive permettono di definire l'intervallo di validità delle parole d'ordine. Questi valori vengono utilizzati all'atto della registrazione di un nuovo utente, per il quale vengono presi come predefiniti. Per la precisione, '**`PASS_MIN_DAYS`**' stabilisce la durata minima di una parola d'ordine che quindi non può essere modificata con maggiore frequenza; '**`PASS_MAX_DAYS`**' stabilisce invece la durata massima di una parola d'ordine dopo la quale l'utenza viene bloccata.
`PASS_WARN_AGE` ***n_giorni***	Stabilisce il numero di giorni di preavviso per la scadenza delle parole d'ordine.
`TTYGROUP` $\{$ ***gruppo*** $\mid$ ***gid*** $\}$	Permette di definire il gruppo a cui attribuire il dispositivo corrispondente al terminale utilizzato dall'utente che accede. Di solito si tratta di '**`tty`**'. Ciò è utile in abbinamento alla direttiva '**`TTYPERM`**', in modo da consentire al programma '**`write`**' (abbinato allo stesso gruppo e impostato con il bit SGID) di scrivere su quel terminale.
`TTYPERM` ***permessi_numerici***	Permette di definire i permessi da attribuire al dispositivo corrispondente al terminale utilizzato per accedere. Di solito, se si utilizza l'abbinamento al gruppo '**`tty`**', si assegnano anche i permessi 0620_8. Il valore predefinito per questi è 0622_8, cosa che consentirebbe la scrittura a chiunque, mentre per motivi di sicurezza si potrebbe preferire 0600_8, in modo da escludere a priori l'uso di '**`write`**' e di qualunque altra interferenza simile.

16.8.2.2 Utilizzo di «pwconv»

Il programma '**pwconv**'[26] permette di convertire un file '`/etc/passwd`' normale in una coppia '`/etc/passwd`' e '`/etc/shadow`', togliendo dal primo le parole d'ordine cifrate. Il programma funziona anche se il file '`/etc/shadow`' esiste già; in tal caso serve per fare in modo che tutte le utenze siano registrate correttamente nel file '`/etc/shadow`' e le parole d'ordine siano tolte dal file '`/etc/passwd`'.

```
pwconv
```

Come si vede dalla sintassi indicata, questo programma non richiede argomenti: si avvale semplicemente della configurazione contenuta in '`/etc/login.defs`' per stabilire i periodi di validità delle parole d'ordine. In pratica, utilizza precisamente le informazioni delle direttive '**PASS_MAX_DAYS**', '**PASS_MIN_DAYS**' e '**PASS_WARN_AGE**'.

16.8.2.3 Utilizzo di «pwunconv»

A fianco di '**pwconv**', il programma '**pwunconv**'[27] svolge il compito inverso: quello di trasferire le parole d'ordine cifrate nel file '`/etc/passwd`', perdendo le informazioni aggiuntive contenute nel file '`/etc/shadow`'.

```
pwunconv
```

Anche questo programma è in grado di funzionare correttamente se parte delle utenze si trova già solo nel file '`/etc/passwd`'. In ogni caso, al termine viene eliminato il file '`/etc/shadow`'.

16.8.2.4 Utilizzo di «useradd»

«

Il programma '**useradd**'[28] permette di aggiungere un utente in un sistema in cui siano attive, o meno, le parole d'ordine oscurate.

```
useradd [opzioni] utente
```

```
useradd -D [opzioni]
```

Il funzionamento di '**useradd**' può essere configurato attraverso il file '`/etc/default/useradd`' e l'uso dell'opzione '**-D**' manifesta l'intenzione di visualizzare tale configurazione o di modificarla.

Dopo la creazione dell'utente, è necessario attribuirgli una parola d'ordine iniziale, attraverso il programma '**passwd**'; inoltre è opportuno creare la directory personale dell'utente.

Il funzionamento di '**useradd**' può essere controllato attraverso il file di configurazione '`/etc/default/useradd`', oppure attraverso opzioni della riga di comando. Queste opzioni possono essere utili quando si utilizza '**useradd**' attraverso uno script, mentre di solito si fa affidamento sulla configurazione memorizzata nel file.

Per questa ragione, nella tabella successiva vengono mostrate solo le opzioni valide in presenza dell'opzione '**-D**'. Quando questa opzione viene usata da sola, '**useradd**' visualizza semplicemente la configurazione attuale.

Tabella 16.59. Alcune opzioni di configurazione.

Opzione	Descrizione
`-D` […] `-b` ***directory_base***	Definisce la nuova directory predefinita di partenza per la creazione di directory personali. A questa viene aggiunta una directory con lo stesso nome dell'utente che si crea. Il valore normale è '`/home/`'. L'argomento di questa opzione viene annotato nella direttiva '**`HOME`**' del file '`/etc/default/useradd`'.
`-D` […] `-e` ***anno-mese-giorno***	Definisce la nuova data di scadenza predefinita delle utenze. La data va inserita nella forma '***aaaa-mm-gg***'. Il valore normale di questa data è indefinito. L'argomento di questa opzione viene annotato nella direttiva '**`EXPIRE`**' del file '`/etc/default/useradd`'.
`-D` […] `-f` ***giorni***	Definisce il numero di giorni predefinito in cui l'utenza rimane utilizzabile dopo la scadenza della validità della parola d'ordine. Il valore normale è −1, pari al numero più grande che possa essere gestito. L'argomento di questa opzione viene annotato nella direttiva '**`INACTIVE`**' del file '`/etc/default/useradd`'.

Opzione	Descrizione
`-D` […] `-g` ***gruppo*** | ***uid***	Definisce il gruppo predefinito a cui possono essere aggregati i nuovi utenti. Il valore normale è 100, pari al gruppo di utenti generico. L'argomento di questa opzione viene annotato nella direttiva '**`GROUP`**' del file '`/etc/default/useradd`'. Si osservi che possono esserci delle distribuzioni GNU in cui il programma '**`useradd`**' è modificato in modo che alla creazione di un nuovo utente, gli venga abbinato un gruppo privato. In tale condizione, questa opzione di configurazione risulterebbe non utilizzata in pratica.
`-D` […] `-s` ***shell***	Definisce la shell predefinita da assegnare ai nuovi utenti. Di solito si tratta di '`/bin/bash`'. L'argomento di questa opzione viene annotato nella direttiva '**`SHELL`**' del file '`/etc/default/useradd`'.

Segue la descrizione di alcuni esempi.

- `#` **`useradd caio`**[*Invio*]

  Crea l'utente '**`caio`**' secondo la configurazione stabilita nel file '`/etc/default/useradd`'.

- `#` **`useradd -D`**[*Invio*]

  Visualizza la configurazione attuale per la creazione di nuove utenze.

Nella distribuzione GNU/Linux Debian, è bene utilizzare sempre solo l'eseguibile '**adduser**', costituito da un programma Perl in grado di gestire correttamente sia '**useradd**' che '**groupadd**', in particolare per ciò che riguarda il problema dei gruppi privati. Per questo motivo, con la distribuzione GNU/Linux Debian non si deve toccare il file '`/etc/default/useradd`', ammesso che ci sia; inoltre non deve essere creato se questo non c'è.

16.8.2.5 File «/etc/default/useradd»

Il file '`/etc/default/useradd`' contiene la configurazione del programma '**useradd**'. Si tratta di direttive nella forma ***nome=valore*** e quasi tutto ciò che appare in questo file può essere modificato attraverso lo stesso '**useradd**', con l'opzione '**-D**'. Segue un esempio di questo file:

```
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

Il significato delle varie direttive è intuitivo; in ogni caso appare descritto nella sezione dedicata a '**useradd**'.

16.8.2.6 Utilizzo di «userdel»

Il programma '**userdel**' [29] permette di eliminare facilmente un'utenza dai file '`/etc/passwd`' e '`/etc/shadow`'. Eventualmente, se

si utilizza l'opzione '**-r**', viene eliminata anche la directory personale dell'utente cancellato, mente altri file che dovessero trovarsi al di fuori di quella gerarchia, possono essere tolti solo in modo manuale.

```
userdel [-r] utente
```

Se si utilizza la tecnica dei gruppi privati, potrebbe essere necessaria, o desiderabile, l'eliminazione del gruppo corrispondente. In tal caso, occorre intervenire manualmente nel file '`/etc/group`'.

Nella distribuzione GNU/Linux Debian, è bene utilizzare sempre solo l'eseguibile '**deluser**', costituito da un programma Perl in grado di gestire correttamente sia '**userdel**', sia '**groupdel**', occupandosi anche della directory personale dell'utente che viene rimosso.

16.8.2.7 Utilizzo di «usermod»

«

Il programma '**usermod**'[30] permette di modificare facilmente alcune caratteristiche di un'utenza. A seconda delle preferenze dell'amministratore del sistema, può darsi che si consideri più facile la modifica diretta dei file '`/etc/passwd`' e '`/etc/shadow`', tuttavia, se si intende indicare una data di scadenza per un'utenza, la conversione in giorni trascorsi dal 01/01/1970, necessaria per modificare direttamente il file '`/etc/shadow`', potrebbe essere un po' seccante.

```
usermod [opzioni] utente
```

Tabella 16.61. Alcune opzioni.

Opzione	Descrizione
`-e` ***anno-mese-giorno***	Definisce la data di scadenza dell'utenza. La data va inserita nella forma '***aaaa-mm-gg***'.
`-f` ***giorni***	Definisce il numero di giorni in cui l'utenza rimane utilizzabile dopo la scadenza della validità della parola d'ordine.
`[-m] -d` ***directory_home***	Modifica la posizione della directory personale dell'utente. Se viene usata anche l'opzione '**`-m`**' si ottiene lo spostamento della vecchia directory nella nuova collocazione, oppure, se manca, questa viene creata.

16.8.2.8 Utilizzo di «chage»

Il programma '**`chage`**'[31] consente di visualizzare o di modificare le informazioni relative alla validità della parola d'ordine di un utente, all'interno di un sistema in cui siano attive le parole d'ordine oscurate. A seconda dell'impostazione della propria distribuzione GNU, può darsi che sia consentito agli utenti comuni di utilizzare l'opzione '**`-l`**' per conoscere le proprie scadenze. Perché ciò avvenga, l'eseguibile '**`chage`**' deve essere SUID-root, oppure deve avere il bit SGID attivo, abbinato a un gruppo particolare che abbia accesso al file '`/etc/passwd`' in lettura.

```
chage [opzioni] utente
```

L'interrogazione di un'utenza come quella che appare nella figura 16.55 si traduce nel risultato seguente:

```
# chage -l tizio[Invio]

Minimum:        0
Maximum:        30
Warning:        7
Inactive:       10
Last Change:                Aug 21, 1998
Password Expires:           Sep 20, 1998
Password Inactive:          Sep 30, 1998
Account Expires:            Aug 21, 1999
```

Tabella 16.63. Alcune opzioni.

Opzione	Descrizione
`-d` ***data_modifica_parola_d'ordine***	Definisce la data in cui è stata modificata la parola d'ordine per l'ultima volta, corrispondente al terzo campo del file '`/etc/shadow`'. La data può essere inserita con un numero intero, corrispondente alla quantità di giorni trascorsi dal 01/01/1970, oppure secondo una forma differente, in base alla localizzazione (nella configurazione italiana, dovrebbe essere possibile scrivere la data come '***gg/mm/ssaa***').

Opzione	Descrizione
`-m` ***giorni_validità_minima***	Definisce il numero di giorni di validità minima della parola d'ordine, corrispondenti al valore inserito nel quarto campo del file '`/etc/shadow`'; entro questo tempo, l'utente non può cambiare la parola d'ordine. Se si indica il valore zero, si consente di cambiare la parola d'ordine in qualsiasi momento.
`-M` ***giorni_validità_massima***	Definisce il numero di giorni di validità massima della parola d'ordine, corrispondenti al valore inserito nel quinto campo del file '`/etc/shadow`'; prima che trascorra questo tempo, l'utente deve cambiare la parola d'ordine.
`-W` ***giorni_di_preavviso***	Definisce il numero di giorni, prima della scadenza della parola d'ordine, durante i quali l'utente viene avvisato della necessità di modificarla. L'informazione viene annotata nel sesto campo del file '`/etc/shadow`'.
`-I` ***giorni_di_riserva***	Definisce la quantità di giorni, dopo la scadenza della parola d'ordine, in cui è consentito accedere con l'obbligo di modificare la parola d'ordine. L'informazione viene annotata nel settimo campo del file '`/etc/shadow`'.

Opzione	Descrizione
`-E` ***data_termine***	Definisce la data di scadenza dell'utenza, corrispondente all'ottavo campo del file '`/etc/shadow`'. La data può essere inserita con un numero intero, corrispondente alla quantità di giorni trascorsi dal 01/01/1970, oppure secondo una forma differente, in base alla localizzazione (nella configurazione italiana, dovrebbe essere possibile scrivere la data come '***gg/mm/ssaa***').

16.8.3 Amministrazione dei gruppi

«

Anche i gruppi possono avere una parola d'ordine, per permettere agli utenti che non vi appartengono di potervisi inserire attraverso il comando '**`newgrp`**'.

Generalmente, per fare in modo che un utente possa partecipare a un gruppo del quale non fa già parte, basta aggiungere il suo nome nell'ultimo campo del record del gruppo in cui questo vuole essere inserito. Da quel momento, quell'utente ha la possibilità di utilizzare il comando '**`newgrp`** ***gruppo***' per agire con i privilegi concessi a quel gruppo.

L'idea di poter aggiungere una parola d'ordine ai gruppi, in modo che gli utenti estranei che la conoscono possano usare ugualmente '**`newgrp`**' per questo, è piuttosto discutibile. Infatti, una parola d'ordine è «sicura» solo se conosciuta da una sola persona; nel momento

in cui la stessa parola d'ordine è conosciuta da un gruppo di persone diventa incontrollabile la sua diffusione (a causa della natura umana).

Tuttavia, il problema esiste e vale la pena di analizzarne gli effetti in presenza di parole d'ordine oscurate.

16.8.3.1 File «/etc/gshadow»

La presenza del file '`/etc/gshadow`' indica l'attivazione delle parole d'ordine oscurate per i gruppi. I record di questo file sono organizzati in campi, separati attraverso due punti verticali ('`:`'), secondo la sintassi seguente:

gruppo : ***parola_d'ordine_cifrata*** : ***amministratori*** : ***utenti_membri***

1. ***gruppo***
   Il nome del gruppo.
2. ***parola_d'ordine_cifrata***
   La parola d'ordine cifrata (che normalmente è assente).
3. ***amministratori***
   Un elenco, separato da virgole, di utenti amministratori del gruppo.
4. ***utenti_membri***
   Un elenco, separato da virgole, di utenti che fanno parte del gruppo.

Gli amministratori del gruppo hanno la possibilità di aggiungere e togliere utenti membri; inoltre, possono cambiare la parola d'ordine.

16.8.3.2 Utilizzo di «grpconv»

Il programma '**grpconv**'[32] permette di convertire un file '`/etc/group`' normale in una coppia '`/etc/group`' e '`/etc/gshadow`', togliendo dal primo le eventuali parole d'ordine cifrate. Il programma funziona anche se il file '`/etc/gshadow`' esiste già: in tal caso serve per fare in modo che tutti i gruppi siano registrati correttamente nel file '`/etc/gshadow`' e le parole d'ordine siano tolte dal file '`/etc/group`'.

```
grpconv
```

16.8.3.3 Utilizzo di «grpunconv»

A fianco di '**grpconv**', il programma '**grpunconv**'[33] svolge il compito inverso: quello di trasferire le parole d'ordine cifrate nel file '`/etc/group`' perdendo le informazioni aggiuntive contenute nel file '`/etc/gshadow`'.

```
grpunconv
```

Anche questo programma è in grado di funzionare correttamente se parte delle utenze si trova solo nel file '`/etc/group`'. In ogni caso, al termine viene eliminato il file '`/etc/gshadow`'.

16.8.3.4 Utilizzo di «gpasswd»

Il programma '**gpasswd**',[34] come suggerisce il nome, serve a cambiare la parola d'ordine di un gruppo. Oltre a questo, però, permette anche di intervenire sugli altri campi del file '`/etc/gshadow`', inserendo o eliminando gli amministratori e i membri di un gruppo.

```
gpasswd [opzioni] gruppo
```

La presenza di una parola d'ordine in un gruppo, serve a permettere a utenti che non siano già membri di poterne fare parte utilizzando il comando '**newgrp**'. Tuttavia, il meccanismo potrebbe non funzionare, a dimostrazione dello scarso interesse verso questa possibilità. Infatti, la vera innovazione nell'introduzione del file '/etc/gshadow' sta nella possibilità di definire degli amministratori per i gruppi, competenti per l'aggregazione dei membri rispettivi.

Tabella 16.64. Alcune opzioni.

Opzione	Descrizione
`-A` ***amministratore*** [`,`…]	Permette all'utente '**root**' di definire uno o più amministratori per il gruppo. L'argomento dell'opzione è un elenco di uno o più utenti a cui viene attribuito il ruolo di amministratori del gruppo. L'elenco di amministratori va a sostituirsi a quanto impostato in precedenza.
`-M` ***membro*** [`,`…]	Permette all'utente '**root**' di definire uno o più membri del gruppo. L'argomento dell'opzione è un elenco di uno o più utenti membri del gruppo. L'elenco di membri va a sostituirsi a quanto impostato in precedenza.
`-a` ***membro***	Permette a un amministratore del gruppo di aggiungere un utente membro.

Opzione	Descrizione
`-d` ***membro***	Permette a un amministratore del gruppo di eliminare un utente membro.
`-r`	Permette a un amministratore del gruppo di eliminare la parola d'ordine.
`-R`	Permette a un amministratore del gruppo di rendere impossibile l'accesso attraverso la parola d'ordine.

16.8.3.5 Utilizzo di «groupadd»

Il programma '**groupadd**'[35] permette di aggiungere un gruppo in un sistema in cui siano attive, o meno, le parole d'ordine oscurate.

```
groupadd [opzioni] gruppo
```

16.8.3.6 Utilizzo di «groupdel»

Il programma '**groupdel**'[36] permette di eliminare un gruppo in un sistema in cui siano attive, o meno, le parole d'ordine oscurate.

```
groupdel gruppo
```

16.8.4 Caso particolare di «adduser» e «addgroup» nella distribuzione GNU/Linux Debian

La distribuzione GNU/Linux Debian, al posto del programma '**adduser**' tradizionale (quello che si usa di solito quando non si gestiscono le parole d'ordine oscurate), dispone di un programma

Perl creato appositamente per gestire simultaneamente la creazione degli utenti e dei gruppi privati relativi. Se si dispone di parole d'ordine oscurate, provvede a richiamare i programmi '**useradd**' e '**groupadd**', nel modo più opportuno.[37]

Con la distribuzione GNU/Linux Debian, i programmi '**useradd**' e '**groupadd**' non vanno usati direttamente; al loro posto si utilizzano '**adduser**' e '**addgroup**' (il secondo è solo un alias, in qualità di collegamento del primo) che si configurano attraverso il file '`/etc/adduser.conf`'. Senza approfondire la sintassi degli argomenti di '**adduser**' e di '**addgroup**', nella versione Debian, si può utilizzare il primo di questi due eseguibili indicando semplicemente il nome dell'utente che si vuole creare, affidandosi alla sua configurazione predefinita. Di seguito appare l'esempio standard del file '`/etc/adduser.conf`':

```
# /etc/adduser.conf: `adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

# The DSHELL variable specifies the default login shell on
# your system.
DSHELL=/bin/bash

# The DHOME variable specifies the directory containing
# users' home directories.
DHOME=/home

# If GROUPHOMES is "yes", then the home directories will be
# created as /home/groupname/user.
GROUPHOMES=no

# If LETTERHOMES is "yes", then the created home directories
# will have an extra directory - the first letter of the
# user name. For example:
```

```
# /home/u/user.
LETTERHOMES=no

# The SKEL variable specifies the directory containing
# "skeletal" user files; in other words, files such as a
# sample .profile that will be copied to the new user's home
# directory when it is created.
SKEL=/etc/skel

# FIRST_SYSTEM_UID to LAST_SYSTEM_UID inclusive is the range
# for UIDs for dynamically allocated administrative and
# system accounts.
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=199

# FIRST_UID to LAST_UID inclusive is the range of UIDs of
# dynamically allocated user accounts.
FIRST_UID=1000
LAST_UID=29999

# The USERGROUPS variable can be either "yes" or "no".
# If "yes" each created user will be given their own group
# to use as a default, and their home directories will be
# g+s.  If "no", each created user will be placed in the
# group whose gid is USERS_GID (see below).
USERGROUPS=yes

# If USERGROUPS is "no", then USERS_GID should be the GID of
# the group `users' (or the equivalent group) on your
# system.
USERS_GID=100

# If QUOTAUSER is set, a default quota will be set from that
# user with `edquota -p QUOTAUSER newuser'
```

```
QUOTAUSER=""

# If DIR_MODE is set, directories will be created with the
# specified mode. Otherwise the default mode 0755 will be
# used.
DIR_MODE=0755

# If SETGID_HOME is "yes" home directories for users with
# their own group the setgid bit will be set. This was the
# default for versions << 3.13 of adduser. Because it has
# some bad side effects we no longer do this per default.
# If you want it nevertheless you can still set it here.
SETGID_HOME=no
```

Come si può osservare, le direttive sono degli assegnamenti a variabili, dove le righe vuote e quelle bianche vengono ignorate, così come è ignorato il testo che segue il simbolo '**#**' fino alla fine della riga in cui appare.

Tabella 16.66. Alcune opzioni.

Opzione	Descrizione
`DSHELL=`***percorso_shell_standard***	Definisce la shell da attribuire agli utenti che vengono creati. In mancanza di questa indicazione, si utilizza '`/bin/bash`'.
`DHOME=`***radice_directory_personali***	Definisce la radice delle directory personali che vengono create. Il valore predefinito è '`/home/`'.
`SKEL=`***scheletro_directory_personali***	Definisce la directory da utilizzare come scheletro per la creazione delle directory personali. In modo predefinito si tratta di '`/etc/skel/`'.

Opzione	Descrizione
`FIRST_UID=`***n_uid_iniziale*** `LAST_UID=`***n_uid_finale***	Definiscono l'intervallo dei numeri UID che possono essere utilizzati per gli utenti. In modo predefinito, si tratta di 1000 e 29999 rispettivamente.
`USERGROUPS={yes|no}`	Serve a definire se gli utenti devono avere un gruppo privato. Se si attiva questa modalità, assegnando la parola chiave '**`yes`**', che è il valore predefinito.
`USERS_GID=`***n_gid***	Questa direttiva serve solo nel caso sia stata utilizzata '**`USERGROUPS=no`**', permettendo così di stabilire il numero GID del gruppo da abbinare agli utenti nuovi.

16.8.5 Verifiche di coerenza

La gestione delle utenze non è fatta solo di inserimenti, modifiche ed eliminazioni. Dal momento che le modifiche possono anche essere fatte direttamente sui file, è comodo se si dispone di qualche strumento di controllo di coerenza.

16.8.5.1 utilizzo di «pwck»

Il programma '**`pwck`**'[38] verifica la coerenza del file '`/etc/passwd`' e, se esiste, del file '`/etc/shadow`' (utilizzando anche il file '`/etc/group`' per il controllo dell'appartenenza ai gruppi). Il programma, previo consenso dell'utilizzatore (l'utente '**`root`**'), può risolvere da

solo alcuni tipi di problemi modificando i file. Tuttavia, se si utilizza l'opzione '**-r**', '**pwck**' si limita a segnalare i problemi.

```
pwck [-r] [file_passwd [file_shadow ] ]
```

Se necessario, si possono indicare espressamente i file che svolgono le funzioni di 'passwd' e 'shadow'.

16.8.5.2 Utilizzo di «grpck»

Il programma '**grpck**' [39] verifica la coerenza del file '/etc/group' e, se esiste, del file '/etc/gshadow' (utilizzando anche il file '/etc/passwd' per la verifica dell'aggregazione degli utenti). Il programma, previo consenso dell'utilizzatore (l'utente '**root**'), può risolvere da solo alcuni tipi di problemi modificando i file. Tuttavia, se si utilizza l'opzione '**-r**', '**grpck**' si limita a segnalare i problemi.

```
grpck [-r] [file_group [file_gshadow ] ]
```

Se necessario, si possono indicare espressamente i file che svolgono le funzioni di 'group' e 'gshadow'.

16.8.6 Copie di sicurezza

Quando si aggiunge, elimina, o si modifica un'utenza attraverso gli strumenti previsti, vengono generate delle copie di sicurezza dei file amministrativi coinvolti. Tipicamente può trattarsi di '/etc/passwd', '/etc/shadow', '/etc/group' e '/etc/gshadow'.

Queste copie di sicurezza si distinguono perché hanno gli stessi nomi dei file corrispondenti con l'aggiunta di un trattino finale. In pratica: '/etc/passwd-', '/etc/shadow-', '/etc/group-' e '/etc/

`gshadow-`'. È importante fare un minimo di attenzione anche a questi file, se si vuole evitare che informazioni importanti vengano conosciute da utenti che non ne hanno il diritto. Infatti, un file '`/etc/shadow-`' che per qualche motivo dovesse diventare leggibile a tutti gli utenti, costituirebbe un grosso buco nel sistema di sicurezza.

16.9 Moduli PAM

«

Secondo la tradizione dei sistemi Unix, l'identificazione di un utente avviene attraverso una parola d'ordine, annotata in modo cifrato nel file '`/etc/passwd`', oppure nel file '`/etc/shadow`' se è attiva la gestione delle parole d'ordine oscurate. L'introduzione delle parole d'ordine oscurate ha aggiunto la possibilità di maggiori controlli sull'utenza, in particolare definendo dei tempi di validità per le parole d'ordine e per le utenze stesse.

Tuttavia, il sistema delle parole d'ordine oscurate rimane ancorato alla gestione di parole d'ordine, mentre i metodi di identificazione potrebbero essere differenti. L'esigenza di definire una struttura generalizzata per la gestione dei metodi di autenticazione ha portato alla realizzazione del sistema PAM (*Pluggable authentication modules*), con il quale, attraverso le librerie PAM è possibile definire e configurare la politica di accesso in base al contesto.

Per mettere in pratica questo sistema, i programmi che in qualche modo si occupano di mettere a disposizione un ***servizio di autenticazione*** devono essere predisposti per l'utilizzo di tali librerie. In generale, un sistema Unix in cui è attivata la gestione PAM, configurato in modo standard, si comporta esattamente come un sistema che ne è sprovvisto. In altri termini, la configurazione standard del sistema PAM è quella che ricalca la tradizione dei sistemi Unix.

In questo capitolo si fa riferimento alla realizzazione del sistema PAM per GNU/Linux, ovvero Linux-PAM.[40]

16.9.1 Servizi PAM

I programmi che si avvalgono del sistema PAM sono stati compilati in modo da utilizzare la libreria LibPAM, costituita generalmente dalla libreria dinamica '`/lib/libpam.so`'. Lo si può verificare facilmente con '**`ldd`**':

```
# ldd /bin/login[Invio]

        ...
        libpam.so.0 => /lib/libpam.so.0 (0xb7f17000)
        ...
```

Ogni programma che utilizza il sistema PAM (attraverso la libreria LibPAM), viene associato a un ***servizio***, il cui nome corrisponde generalmente a quello del programma. Per esempio, si associa il servizio '**`login`**' al programma '**`login`**', ma succede anche che si associ il servizio '**`ssh`**' al programma '**`sshd`**'. L'associazione tra programmi e servizi dipende dal modo in cui i programmi sono compilati, pertanto non si può modificare in fase di amministrazione del sistema, a meno di intervenire direttamente nei sorgenti degli stessi.

Ogni servizio PAM rappresenta una situazione in cui si richiede l'autenticazione degli utenti, o comunque qualcosa di strettamente correlato. Teoricamente, programmi differenti possono condividere lo stesso servizio PAM.

Per ogni servizio PAM viene predisposta una configurazione particolareggiata, la quale può risiedere complessivamente nel file '`/etc/pam.conf`', oppure in file separati all'interno della directory '`/etc/`

pam.d/'. Generalmente il file '/etc/pam.conf' non viene più usato e i file contenuti nella directory '/etc/pam.d/' hanno nomi corrispondenti al servizio PAM a cui si riferiscono.

16.9.2 File di configurazione e moduli

«

L'aspetto più importante del sistema PAM è la modularità, costituita da diversi file di libreria, oltre a quello principale già descritto (LibPAM). Quando si gestisce il sistema PAM, questi file sono indispensabili al funzionamento del sistema, pertanto non possono essere collocati al di sotto della directory '/usr/'.[41] La collocazione normale di questi file è così la directory '/lib/security'. A titolo di esempio, nella directory si potrebbero vedere i file seguenti:

```
pam_access.so      pam_mail.so           pam_time.so
pam_deny.so        pam_mkhomedir.so      pam_unix.so
pam_env.so         pam_motd.so           pam_unix_acct.so
pam_filter.so      pam_nologin.so        pam_unix_auth.so
pam_ftp.so         pam_permit.so         pam_unix_passwd.so
pam_group.so       pam_rhosts_auth.so    pam_unix_session.so
pam_issue.so       pam_rootok.so         pam_userdb.so
pam_lastlog.so     pam_securetty.so      pam_warn.so
pam_ldap.so        pam_shells.so         pam_wheel.so
pam_limits.so      pam_stress.so
pam_listfile.so    pam_tally.so
```

Come già accennato, la configurazione del sistema PAM dipendeva originariamente dal file '/etc/pam.conf'. Attualmente la configurazione è suddivisa in più file (ognuno competente per un servizio PAM specifico), contenuti nella directory '/etc/pam.d/' e il file '/etc/pam.conf' rimane vuoto o commentato completamente. In tal modo, ogni servizio PAM ha un proprio file nella directory '/etc/pam.d/', facilitando anche la gestione dei pacchetti di appli-

cazioni, i quali non sono costretti a condividere la configurazione in un solo file.

I file di configurazione contenuti nella directory '`/etc/pam.d/`' sono file di testo normali, in cui le righe vuote e quelle bianche sono ignorate, così come sono ignorate quelle che iniziano con il simbolo '**#**'. Per il resto, si tratta di direttive con la struttura seguente:

tipo ***livello_di_controllo*** ***modulo_pam*** [***opzioni_del_modulo***]

Oppure, si può trattare di direttive di inclusione di altri file:

`@include` ***nome_file_da_includere***

Il listato 16.69 rappresenta l'esempio di come potrebbe apparire il file '`/etc/pam.d/login`' che serve a configurare il servizio di autenticazione attraverso il programma '**`login`**'.

Listato 16.69. Esempio del file '`/etc/pam.d/login`'.

```
auth       requisite  pam_securetty.so
auth       required   pam_nologin.so
auth       required   pam_env.so
auth       required   pam_unix.so nullok
account    required   pam_unix.so
session    required   pam_unix.so
session    optional   pam_lastlog.so
session    optional   pam_motd.so
session    optional   pam_mail.so standard noenv
password   required   pam_unix.so nullok obscure min=4 max=8
```

Il primo campo serve a definire la fase di autenticazione a cui si fa riferimento, attraverso alcune parole chiave ben definite, descritto nell'elenco successivo.

Parola chiave	Descrizione
`auth`	Verifica l'identità dell'utente, eventualmente attraverso la richiesta di una parola d'ordine oppure attraverso altri metodi di autenticazione.
`account`	Verifica lo stato dell'utenza, la quale può risultare attiva, scaduta, inattiva o disabilitata, solitamente secondo le informazioni provenienti dal file '`/etc/shadow`'.
`password`	Si utilizza per controllare la modifica della parola d'ordine, per esempio per imporre che questa non sia troppo banale.
`session`	Questo contesto permette di definire delle azioni da compiere nel momento dell'accesso, oppure al momento della sua conclusione. Osservando l'esempio che è stato proposto, si può intendere intuitivamente la richiesta di informare l'utente sull'ultimo accesso, di mostrare il messaggio del giorno e di informare sulla presenza di messaggi di posta elettronica.

Ogni modulo PAM può essere visto come una funzione che restituisce un valore. Sono possibili diversi casi, tra cui si distingue un successo completo o un insuccesso, con tante sfumature intermedie che, eventualmente, possono essere verificate nel dettaglio. A questo proposito, il secondo campo delle direttive di configurazione consente di definire come deve essere preso in considerazione l'esito della verifica fatta dal modulo corrispondente. Anche in questo caso si usa una parola chiave, come descritto nel prossimo elenco.

Parola chiave	Descrizione
`required`	Rappresenta un controllo indispensabile, attraverso il quale si pretende di ottenere un esito soddisfacente. Tuttavia, l'insuccesso nell'autenticazione, che porta comunque a un risultato finale negativo, non conclude immediatamente la procedura, in modo da non consentire all'utente di comprendere dove si sia verificato il problema.
`requisite`	Rappresenta un controllo indispensabile, attraverso il quale si pretende di ottenere un esito soddisfacente. A differenza di '**`required`**', in questo caso l'insuccesso fa concludere immediatamente la procedura (logicamente con un risultato finale negativo).
`sufficient`	Rappresenta un controllo sufficiente. In pratica, se prima di questo controllo non si sono verificati problemi, un esito soddisfacente conclude la procedura con un'autenticazione corretta. Al contrario, un esito non soddisfacente determina solo un risultato temporaneamente indeterminato. Naturalmente, se si tratta dell'ultimo risultato disponibile, ciò corrisponde a un risultato negativo (in quanto non positivo).
`optional`	Rappresenta un controllo opzionale, nel quale un esito non soddisfacente dà soltanto un risultato indeterminato. Si usa questa modalità di controllo quando il modulo PAM serve per compiere delle operazioni che possono fallire senza però pregiudicare la sicurezza.

Le direttive di ogni servizio PAM vengono analizzate nell'ordine in cui appaiono; pertanto, è evidente che il risultato finale dipenda dalla sequenza in cui vengono trovate nel file di configurazione relativo.

Le parole chiave appena descritte (quelle del secondo campo) possono essere dettagliate in modo completo, sostituendole con un'espressione tra parentesi quadre (pertanto le parentesi quadre vanno inserite effettivamente). A titolo di esempio, la tabella successiva descrive la traduzione delle parole chiave già descritte. Naturalmente, per poter usare tali espressioni occorre conoscere perfettamente le opzioni relative.

Parola chiave	Espressione equivalente
`required`	`[success=ok new_authok_reqd=ok ↩ ↪ignore=ignore default=bad]`
`requisite`	`[success=ok new_authok_reqd=ok ↩ ↪ignore=ignore default=die]`
`sufficient`	`[success=done new_authok_reqd=done ↩ ↪default=ignore]`
`optional`	`[success=ok new_authok_reqd=ok ↩ ↪default=ignore]`

Il terzo campo rappresenta il nome del file di libreria che costituisce il modulo relativo. Questo nome può essere completo di percorso assoluto, oppure può essere indicato senza tale informazione, se la sua collocazione è quella predefinita. Il quarto campo è costituito dalle opzioni da passare al modulo, separate tra loro da uno o più spazi.

Tra i vari file di configurazione è importante definirne uno denominato '`other`', che viene utilizzato quando per quel particolare servizio di autenticazione non ne è stato previsto uno specifico. L'esempio che si vede nel listato 16.73 rappresenta il contenuto di questo

file quando si vuole garantire un sistema minimo di autenticazione, secondo i canoni tradizionali.

Listato 16.73. Un file ‘`/etc/pam.d/other`’ per consentire l’accesso in mancanza di altro.

```
auth     required       pam_unix.so
account  required       pam_unix.so
password required       pam_unix.so
session  required       pam_unix.so
```

Per verificare che ciò sia vero, si può provare a spostare temporaneamente gli altri file di configurazione della directory ‘`/etc/pam.d/`’ in un’altra collocazione, lasciando al suo posto il file ‘`/etc/pam.d/other`’.[42]

In alternativa, si può fare in modo che non ci siano altre possibilità di autenticazione, al di fuori di quando definito dai file di configurazione specifici. Per questo, basta che il file ‘`/etc/pam.d/other`’ contenga le righe che si vedono nel listato 16.74.

Listato 16.74. Un file ‘`/etc/pam.d/other`’ per impedire l’accesso quando manca una configurazione specifica.

```
auth     required       pam_deny.so
account  required       pam_deny.so
password required       pam_deny.so
session  required       pam_deny.so
```

16.9.3 Verifica nel registro del sistema

A seconda delle circostanze, alcuni moduli possono annotare nel registro del sistema l’esito della loro verifica. Spesso è prevista l’opzione ‘**`debug`**’ per abilitare queste annotazioni, a meno che ciò sia implicito. Queste annotazioni possono aiutare l’amministratore a

comprendere dove ci possono essere dei problemi di configurazione. A titolo di esempio, si può osservare l'estratto seguente:

```
Oct 21 18:07:30 dinkel PAM_unix[591]: check pass; ↩
↪user unknown
Oct 21 18:07:30 dinkel PAM_unix[591]: authentication ↩
↪failure; LOGIN(uid=0) -> tizio for login service
Oct 21 18:07:33 dinkel login[591]: FAILED LOGIN (1) ↩
↪on 'tty3' FOR 'UNKNOWN', Authentication service ↩
↪cannot retrieve authentication info.
```

In questo caso si può osservare che l'utente '**tizio**' ha tentato di accedere attraverso il servizio di autenticazione '**login**', senza che per lui sia prevista un'utenza, pertanto, già il nominativo-utente '**tizio**' risulta sconosciuto.

16.9.4 Configurazione particolareggiata dei moduli

«

Oltre alle opzioni fornite nelle direttive dei file di configurazione dei servizi di autenticazione, nella directory '`/etc/pam.d/`', alcuni moduli possono richiedere una configurazione particolare. Questi file ulteriori hanno solitamente un nome corrispondente a quello dei moduli, senza il prefisso '`pam_`' e senza l'estensione '`.so`', con l'aggiunta dell'estensione '`.conf`', collocati nella directory '`/etc/security/`'. Per esempio, il file '`/etc/security/access.conf`' si riferisce al modulo '`pam_access.so`'.

In condizioni normali, tali file di configurazione ulteriori, sono vuoti, oppure sono commentati completamente, rimanendo a disposizione per la definizione di funzionalità particolari.

16.9.5 Descrizione di alcuni moduli

Per poter mettere mano, concretamente, alla configurazione del sistema PAM, occorre conoscere i moduli e il loro utilizzo. I moduli principali sono descritti nel documento *The Linux-PAM system administrators' guide*, annotato alla fine del capitolo, ma altri moduli possono aggiungersi per scopi specifici. Dovrebbero essere disponibili anche delle pagine di manuale, corrispondenti ai nomi dei moduli (senza l'estensione '`.so`'); per esempio *pam_unix(8)*, *pam_deny(8)*,... Nelle sezioni successive vengono descritti solo alcuni tra i moduli più semplici.

Bisogna ricordare che la modifica della configurazione del sistema PAM è sempre molto delicata, perché si corre il rischio di impedire l'accesso o di consentirlo indiscriminatamente, o comunque di gestirlo secondo criteri non desiderati.

16.9.5.1 Modulo «pam_warn.so»

Il modulo costituito dal file di libreria '`pam_warn.so`' consente di eseguire un'annotazione nel registro di sistema e può essere associato a tutte le fasi ('**`auth`**', '**`account`**', '**`password`**' e '**`session`**'). Per esempio si potrebbe utilizzare una direttiva come quella seguente:

```
auth  required  pam_warn.so
```

Quando la direttiva viene presa in considerazione, si ottiene un'annotazione nel registro di sistema, simile a quella seguente:

```
Nov  5 20:00:39 127 login[3702]: ↩
↪pam_warn(login:auth):function=[pam_sm_authenticate] ↩
↪service=[login] ↩
↪terminal=[tty4] user=[tizio] ruser=[<unknown>] ↩
↪rhost=[]
```

Naturalmente, altri moduli trasmettono già informazioni sufficienti nel registro di sistema e rendono normalmente inutile l'uso di 'pam_warn.so'.

16.9.5.2 Modulo «pam_permit.so»

«

Il modulo costituito dal file di libreria 'pam_permit.so' consente l'accesso in ogni circostanza; pertanto va usato solo per fare degli esperimenti. Può essere associato a tutte le fasi ('**auth**', '**account**', '**password**' e '**session**'). Per esempio, all'inizio del file '/etc/pam.d/login', prima delle altre direttive, potrebbe apparire quella seguente:

```
auth  sufficient  pam_permit.so
```

Così facendo, dai terminali comuni, sarebbe consentito l'accesso a qualunque utente, senza la richiesta di alcuna parola d'ordine (in altri termini, il programma '**login**' non procederebbe con alcuna richiesta di parola d'ordine). Ovviamente, una direttiva del genere, può essere utile concretamente solo quando si vuole specificare che la presenza di risultati indeterminati, precedenti, devono portare comunque a un risultato finale positivo.

16.9.5.3 Modulo «pam_deny.so»

Il modulo relativo al file ‘`pam_deny.so`’ è l’opposto di ‘**`pam_permit.so`**’, in quanto dà sempre un esito negativo. Anche questo modulo può essere usato per tutte le fasi (‘**`auth`**’, ‘**`account`**’, ‘**`password`**’ e ‘**`session`**’). Il suo uso più probabile riguarda il file ‘`/etc/pam.d/other`’, quando si vogliono escludere gli accessi che non siano stati previsti espressamente attraverso altri file:

```
auth       required  pam_deny.so
account    required  pam_deny.so
password   required  pam_deny.so
session    required  pam_deny.so
```

16.9.5.4 Modulo «pam_exec.so»

Il modulo relativo al file ‘`pam_exec.so`’ consente di eseguire un programma e di tenere conto dell’esito dello stesso. Può essere usato per tutte le fasi (‘**`auth`**’, ‘**`account`**’, ‘**`password`**’ e ‘**`session`**’). Nella documentazione del modulo si fa l’esempio seguente che è particolarmente significativo:

```
password  optional  pam_exec.so seteuid make -C /var/yp
```

In questo caso, quando viene presa in considerazione la direttiva si tratta di una modifica di una parola d’ordine, quindi l’azione associata al modulo consiste nell’eseguire il comando ‘**`make -C /var/yp`**’. In pratica, in questo modo, quando si cambia la parola d’ordine vengono aggiornati anche i file del NIS, presumibilmente per la condivisione delle utenze attraverso la rete.

Si osservi nell’esempio l’uso della parola chiave ‘**`optional`**’, per garantire che il risultato del comando eseguito non abbia effet-

to sul processo di modifica della parola d'ordine. Inoltre, l'opzione '**seteuid**' serve sostanzialmente a far sì che il comando ('**make -C /var/yp**') venga eseguito con i privilegi dell'utente '**root**'.

16.9.5.5 Modulo «pam_unix.so»

Il modulo relativo al file '`pam_unix.so`' consente di riprodurre il sistema di verifica e autenticazione tradizionale dei sistemi Unix, con i file '`/etc/passwd`' e '`/etc/shadow`' (se la gestione delle parole d'ordine oscurate è attiva). Può essere usato per tutte le fasi ('**auth**', '**account**', '**password**' e '**session**').

A titolo di esempio, un file '`/etc/pam.d/login`' ridotto all'osso potrebbe avere il contenuto seguente:

```
auth       required       pam_unix.so
account    required       pam_unix.so
password   required       pam_unix.so
session    required       pam_unix.so
```

Questo modulo prevede l'uso di diverse opzioni e di norma ne vengono usate alcune, soprattutto per garantire che le parole d'ordine siano presenti e rispettino alcuni criteri minimi di sicurezza. A ogni modo, si veda la pagina di manuale *pam_unix(8)*.

16.10 Contabilità dell'utilizzo di risorse del sistema

Il problema della registrazione dell'utilizzo di risorse è nato proprio per misurare e fare pagare i servizi utilizzati dagli utenti. In questo senso si spiega l'enfasi «contabile» che si dà al problema.

Alla base della contabilità dell'utilizzo delle risorse del sistema sta il file '`/var/log/wtmp`', che deve esistere perché tali registrazioni

avvengano effettivamente. Per motivi storici, non si tratta di un file di testo normale, così che per leggerlo si usa generalmente il programma '**last**', al quale si aggiungono eventualmente altri programmi più raffinati.

Oltre alla contabilità basata sul file '`/var/log/wtmp`' si aggiunge quella legata ai processi, derivata da BSD (*BSD process accounting*). Mentre il file '`/var/log/wtmp`' (e anche '`/var/run/utmp`') è gestito generalmente da Init, dalla procedura di accesso tradizionale ('**login**'), dalla serie dei programmi Getty e da altri programmi che sono legati al sistema di autenticazione degli utenti, la contabilità dei processi in stile BSD è gestita direttamente dal kernel (sezione 8.3.1).

16.10.1 Formato dei file

Come accennato, una delle caratteristiche importanti di questi file è il fatto di non essere file di testo normali. Il formato del loro contenuto varia da sistema a sistema e anche da una versione all'altra dello stesso sistema operativo. Pertanto, può succedere alle volte che qualcosa non funzioni, nel senso che i programmi che vi accedono non riescono a interpretare i dati in modo corretto, o peggio eseguono delle registrazioni errate.

Questa annotazione serve per tenere in considerazione il problema, ma tutto quello che si può fare, quando si notano delle anomalie legate a queste componenti del sistema, è l'aggiornamento del software.

16.10.2 Contabilità basata sul file «/var/log/wtmp»

Il file ‘/var/log/wtmp’ è il registro storico degli accessi al sistema. Al suo interno vengono annotate le informazioni della data e dell’ora di accesso di ogni utente, assieme alla provenienza. I dati contenuti in questo file hanno valore solo se sono completi, nel senso che per ogni accesso si deve trovare anche la registrazione della conclusione della sessione di lavoro, altrimenti non possono essere calcolati i tempi di utilizzo.

Purtroppo, questo file non offre le garanzie di una base di dati vera e propria, così le registrazioni che vengono fatte al suo interno non sono mai sicure. Pertanto, i dati che si riescono a estrapolare sono da considerare approssimativi in generale.

Questo file tende a ingrandirsi rapidamente, tanto che periodicamente conviene fare pulizia. Di solito, le distribuzioni GNU provvedono a fornire degli script necessari per gestire in modo elegante, attraverso il sistema Cron, l’archiviazione e la rotazione dei file delle registrazioni, compreso ‘/var/log/wtmp’.

16.10.2.1 Utilizzo di «last»

Il programma ‘**last**’ visualizza il contenuto del file delle registrazioni degli accessi (*login*) e disconnessioni (*logout*) per le informazioni riguardanti gli utenti e i terminali. Il file dal quale queste informazioni vengono attinte è ‘/var/log/wtmp’.[43]

```
last [opzioni] [nome…]
```

L’esempio seguente mostra una parte dell’output che potrebbe essere generato da questo programma.

```
daniele tty5                    Tue Mar 30 16:18   still logged in
daniele tty5                    Tue Mar 30 16:17 - 16:18  (00:01)
tizio   ttyp1   roggen.brot.dg Tue Mar 30 14:33   still logged in
reboot  system boot             Tue Mar 30 14:30
root    tty3                    Mon Mar 29 22:18 - down   (01:29)
daniele tty2                    Mon Mar 29 21:29 - 23:47  (02:18)
caio    ttyp1   roggen.brot.dg Mon Mar 29 21:14 - 23:47  (02:33)
reboot  system boot             Mon Mar 29 21:10
```

Si osserva in particolare che la prima voce rappresenta l'accesso più recente, quello dell'utente '**daniele**' dalla quinta console virtuale, dove risulta essere ancora collegato. Si vede anche che lo stesso vale per l'utente '**tizio**' che sta utilizzando il sistema attraverso un accesso remoto proveniente dall'elaboratore *roggen.brot.dg*. Si notano anche gli accessi conclusi regolarmente (quelli che hanno un orario di inizio e un orario di fine, oltre che l'indicazione della durata dell'accesso tra parentesi) e quindi si distinguono gli accessi sicuramente conclusi, di cui non è stata annotata la fine. Infatti, il giorno 30 marzo alle ore 14:30 il sistema è stato riavviato e, di conseguenza, gli accessi in essere in precedenza sono da considerare conclusi: l'accesso dell'utente '**root**' del 29 marzo alle ore 22:18 non è stato concluso in modo normale, probabilmente perché ha avviato il programma '**shutdown**' e non ha fatto in tempo a concludere la sessione di lavoro.

Tabella 16.83. Alcune opzioni.

Opzione	Descrizione
`-`***numero*** `-n` ***numero*** `--lines` ***numero***	Limita il numero di elementi visualizzati allo specifico valore numerico indicato.
`-f` ***file*** `--file` ***file***	Analizza il file specificato invece di utilizzare quello predefinito, cioè '`/var/log/wtmp`'.
`-x` `--more-records`	Permette di conoscere anche le informazioni sull'arresto del sistema e in generale sui cambiamenti del livello di esecuzione (*runlevel*).

Segue la descrizione di alcuni esempi.

- `$` **`last`** [*Invio*]

  Visualizza gli ultimi eventi del registro degli accessi.

- `$` **`last tizio root`** [*Invio*]

  Visualizza gli accessi e le disconnessioni da parte degli utenti '**`tizio`**' e '**`root`**'.

16.10.2.2 Utilizzo di «ac»

«

Il programma '**`ac`**'[44] si basa sul contenuto del file '`/var/log/wtmp`' per determinare i tempi di accesso complessivi del periodo a cui si riferisce il file stesso.

```
ac [opzioni] [utente...]
```

Se viene utilizzato senza argomenti, si limita a emettere il tempo complessivo di tutti gli accessi, pertanto è utile in pratica solo quando si indicano delle opzioni. Se viene indicato il nome di uno o più utenti, si ottengono soltanto i dati relativi a questi.

L'accuratezza delle informazioni ottenute con '**ac**' dipende naturalmente dall'integrità del file che viene analizzato.

Tabella 16.84. Alcune opzioni.

Opzione	Descrizione
`-d` `--daily-totals`	Mostra l'elenco dei tempi di accesso giornalieri.
`-p` `--individual-totals`	Mostra l'elenco dei tempi di accesso suddivisi per utente.
`-f` ***file*** `--file` ***file***	Analizza il file specificato invece di utilizzare quello predefinito, cioè '`/var/log/wtmp`'.

Segue la descrizione di alcuni esempi.

- `$` **ac** [*Invio*]

  Mostra il totale degli accessi, per esempio ciò che appare di seguito, tenendo conto che il valore fa riferimento alle ore. Per la precisione si tratta di 4198 ore e 51 minuti.

```
total     4198.85
```

- `$` **`ac -d`**[*Invio*]

  Mostra l'elenco dei tempi di accesso giornalieri, per esempio il listato seguente che viene mostrato solo nella sua parte finale:

```
…
Mar 24  total       35.21
Mar 25  total       26.95
Mar 26  total        2.67
Mar 28  total       61.54
Mar 29  total       35.55
Today   total       45.64
```

- `$` **`ac -p`**[*Invio*]

  Mostra l'elenco dei tempi di accesso suddivisi per utente:

```
        pippo                                1.84
        ftp                                  0.99
        tizio                                2.93
        daniele                           3100.52
        root                              1083.21
        sempronio                            6.41
        caio                                 3.41
        total     4199.32
```

- `$` **`ac -p tizio caio`** [*Invio*]

  Come nell'esempio precedente, ma limitatamente agli utenti '**`tizio`**' e '**`caio`**':

```
        tizio                                2.93
        caio                                 3.41
        total        6.34
```

- `$` **`ac -p tizio caio -f /var/log/wtmp.1`** [*Invio*]

Come nell'esempio precedente, ma analizzando il file '`/var/log/wtmp.1`' che presumibilmente è il file delle registrazioni precedente.

16.10.3 Contabilità dei processi

Come già accennato all'inizio del capitolo, la contabilità riferita ai processi è gestita direttamente dal kernel. Questa viene attivata attraverso una chiamata di sistema, ***acct()***, per cui si usa un programma apposito: '**accton**'.[45]

```
accton [file_delle_registrazioni]
```

Per la precisione, se '**accton**' viene usato senza argomenti, la contabilizzazione da parte del kernel viene disattivata; al contrario, se si indica il file da utilizzare, la contabilizzazione viene attivata e diretta verso quel file.

Il file in questione può essere '`/var/log/pacct`', o anche '`/var/account/pacct`'. Nel secondo caso, si attiva la registrazione contabile dei processi con il comando seguente (naturalmente è necessario che il file esista già).

```
# accton /var/account/pacct [Invio]
```

Il problema della contabilità dei processi sta nel fatto che viene considerato un accessorio di importanza minore, pertanto può capitare che i programmi di cui si dispone non siano perfettamente conformi al formato del file generato dal kernel, in quanto non aggiornati.

Al contrario della contabilità legata al file ‘`/var/log/wtmp`’, le informazioni riferite ai processi vengono considerate delle informazioni riservate, pertanto i permessi del file ‘`/var/account/pacct`’ dovrebbero impedire anche la lettura da parte degli utenti comuni.

Una gestione seria di questo sistema contabile richiede la sua attivazione e disattivazione attraverso la stessa procedura di inizializzazione del sistema. Semplificando molto le cose, lo script che attiva e disattiva la contabilità potrebbe essere fatto nel modo seguente:

```
#! /bin/sh
test -x /usr/sbin/accton || exit 0
case "$1" in
    start)
        echo "Avvio della contabilità dei processi."
        /usr/sbin/accton /var/account/pacct 2>/dev/null
        ;;
    stop)
        echo "Arresto della contabilità dei processi."
        /usr/sbin/accton 2>/dev/null
        ;;
    *)
        echo "Utilizzo: acct {start|stop}"
        exit 1
esac
exit 0
```

16.10.3.1 Utilizzo di «lastcomm»

«

Il programma ‘**`lastcomm`**’[46] è fondamentale per la lettura del file della contabilità dei processi. Di per sé, per funzionare, non richiede i privilegi dell’utente ‘**`root`**’, però il file utilizzato per questa contabi-

lità, ‘`/var/log/pacct`’, è normalmente protetto contro qualunque accesso non privilegiato.

`lastcomm` [***comando***…] [***utente***…] [***terminale***…] [***opzioni***]

Il programma ‘**`lastcomm`**’ può essere utilizzato senza argomenti, per ottenere tutte le informazioni contenute all’interno del file ‘`/var/log/pacct`’, oppure può essere avviato con l’indicazione di comandi, utenti e terminali, in modo da limitare le informazioni che si vogliono estrarre da quel file.

Il listato tipico che si dovrebbe ottenere da questo programma è simile all’esempio seguente:

```
...
cat              tizio    tty1         0.03 secs Tue Mar 30 07:38
ls               tizio    tty1         0.04 secs Tue Mar 30 07:38
clear            tizio    tty1         0.01 secs Tue Mar 30 07:38
```

Tabella 16.91. Alcune opzioni.

Opzione	Descrizione
`--user` ***nome_utente***	Se l’indicazione del nome di un utente può essere ambigua, nel senso che potrebbe essere confuso con un comando, si può utilizzare questa opzione.
`--command` ***comando***	Questa opzione permette di indicare un comando in modo da evitare ambiguità con i nomi degli utenti e dei terminali.

Opzione	Descrizione
`--tty` ***terminale***	Questa opzione permette di indicare un terminale (il nome del dispositivo senza il prefisso '`/dev/`') in modo da evitare ambiguità con i nomi degli utenti e dei comandi.
`-f` ***file_della_contabilità*** `--file` ***file_della_contabilità***	Se si desidera consultare un file diverso da quello predefinito, si può utilizzare questa opzione per specificarlo.

Segue la descrizione di alcuni esempi.

- `#` **`lastcomm tizio`** [*Invio*]

  Mostra la contabilità dei processi riferita all'utente '**`tizio`**'.

- `#` **`lastcomm --user tizio`** [*Invio*]

  Esattamente come nell'esempio precedente, ma con l'indicazione esplicita che '**`tizio`**' è inteso essere un utente.

16.10.3.2 Utilizzo di «sa»

Il programma '**`sa`**' [47] genera delle statistiche dai dati contenuti nel file '`/var/account/pacct`', o in un altro che venga indicato come ultimo argomento della riga di comando. Oltre a questo, '**`sa`**' utilizza altri due file: '`/var/account/savacct`' e '`/var/account/usracct`'. Questi gli permettono di annotare le informazioni generate: nel primo caso riferite alla situazione complessiva; nel secondo distinte in base all'utente.

`sa` [***opzioni***] [***file_della_contabilità***]

A seconda di come è stato compilato il sorgente del programma, alcune opzioni possono essere disponibili o meno; inoltre, non è stabilito in modo univoco quale sia la collocazione esatta dei file utilizzati per questa contabilità. Per conoscere queste cose, basta avviare '**sa**' con l'opzione '**-h**'. In particolare, si potrebbe vedere il risultato seguente:

```
The system's default process accounting files are:

  raw process accounting data: /var/account/pacct
      summary by command name: /var/account/savacct
          summary by username: /var/account/usracct
```

In condizioni normali, quando '**sa**' viene avviato senza opzioni (o al massimo con l'indicazione del file contenente la contabilità), si ottiene un listato simile a quello seguente:

```
     246     112.57re       1.38cp
      24       8.60re       0.95cp   ***other*
       2       1.03re       0.19cp   dpkg
       5       5.08re       0.05cp   troff
      48       8.08re       0.03cp   sh
       2       0.43re       0.02cp   rm
      12       8.42re       0.02cp   man
      36       0.13re       0.02cp   sa
...
```

La prima colonna rappresenta l'utilizzo in termini di chiamate di sistema, dove per esempio '**rm**' è stato avviato solo due volte; la seconda colonna, dove i valori sono seguiti dalla sigla '**re**', indica il tempo reale di CPU; la terza colonna riporta la somma tra il tempo di sistema e quello utente dell'utilizzo della CPU; l'ultima colonna indica il nome del processo relativo.

Nel seguito vengono descritte solo alcune delle opzioni, dove in particolare sono state saltate quelle che possono aiutare a riordinare in modo differente i dati. Eventualmente, si può consultare la pagina di manuale *sa(8)*.

Tabella 16.94. Alcune opzioni.

Opzione	Descrizione
`-c` `--percentages`	Per ogni colonna di valori, ne aggiunge un'altra con le percentuali relative.
`-m` `--user-summary`	Invece di generare un listato normale organizzato secondo i processi, genera un riassunto dell'utilizzo in base agli utenti proprietari dei processi.
`-u` `--print-users`	Genera un elenco differente, composto dagli utenti, il tempo di CPU e il nome dei processi utilizzati dagli utenti stessi. Il risultato è un elenco molto più lungo del solito.

16.11 Configurazione e personalizzazione

«

Durante la fase di installazione di un sistema GNU, è normale per le varie distribuzioni di prendersi cura di un minimo di configurazione del sistema, soprattutto per ciò che riguarda le convenzioni nazionali. A questo proposito è bene conoscere l'uso di due termini comuni:

- ***internazionalizzazione***, abbreviato con la sigla ***i18n***, riferito alla creazione o alla modifica di un programma in modo che

sia in grado di tenere conto delle preferenze dell'utente (basate generalmente sulle convenzioni nazionali);

- ***localizzazione***, abbreviato con la sigla ***l10n***, riferito all'azione di informare un programma sulla scelta di un insieme particolare di preferenze.

Ci sono aspetti della configurazione che riguardano il sistema nel suo complesso, come la definizione della mappa della tastiera, oppure solo una sessione di lavoro particolare. Questo significa che parte della configurazione è riservata all'amministratore, mentre il resto può essere modificato dal singolo utente, senza interferire sull'attività degli altri.

In questo capitolo si fa riferimento a concetti che possono essere chiariti solo in capitoli successivi, in particolare ciò che riguarda la shell e con essa la definizione delle variabili di ambiente. In particolare, gli esempi mostrati fanno riferimento alla shell standard (compatibili con quella di Bourne).

Nelle sezioni 14.3 e 14.4 viene descritta la configurazione della tastiera per l'uso con la console di un sistema GNU/Linux.

Nella sezione 28.6 viene descritta la configurazione della tastiera con il sistema grafico X.

16.11.1 Frammentazione del sistema di configurazione

Lo sconforto maggiore per chi si avvicina a un sistema operativo Unix (quali i sistemi GNU) per la prima volta, è dato dalla complessità del sistema di configurazione. Il problema è che non esiste una

«autorità» unica di configurazione, perché le esigenze di questo tipo sono dinamiche, in funzione delle caratteristiche particolari dei programmi utilizzati.

A ben guardare, questo problema riguarda qualunque sistema operativo che abbia un minimo di complessità.

16.11.1.1 Collocazione

«

In linea di massima si distinguono due livelli: la configurazione globale del sistema, definita nei file contenuti nella directory '`/etc/`' che sono di competenza dell'amministratore del sistema; la configurazione particolare di ogni utente, definita da una serie di file, contenuti nelle rispettive directory personali (*home*), che si distinguono perché generalmente iniziano con un punto singolo.

La configurazione globale dovrebbe essere predisposta in modo da garantire i servizi previsti e la sicurezza richiesta dalle caratteristiche del sistema. Oltre a questo, dovrebbe offrire un'impostazione standard per gli utenti che poi potrebbero limitarsi a modificare il minimo indispensabile.

16.11.1.2 Sequenza

«

Si possono distinguere tre fasi nella definizione della configurazione del sistema:

1. la procedura di inizializzazione del sistema (Init);
2. lo script di configurazione globale della shell (nel caso di quelle standard, derivate dalla shell di Bourne, si tratta di '`/etc/profile`');

3. lo script di configurazione personale della shell (per esempio '`~/.profile`', o qualcosa di simile);
4. i programmi avviati successivamente utilizzano i loro metodi di configurazione, basati eventualmente su file di configurazione globale collocati nella directory '`/etc/`', su file di configurazione personalizzata collocati nelle directory personali degli utenti che li utilizzano, sulla presenza e sul contenuto di variabili di ambiente determinate.

La prima fase viene eseguita una volta sola all'atto dell'avvio del sistema. Serve per attivare i servizi previsti, generalmente in forma di programmi demone, oltre che per fissare alcuni elementi di configurazione che non possono essere demandati in alcun caso alla gestione da parte degli utenti comuni.

In questa fase, tra le altre cose, viene impostata la mappa della tastiera, si definiscono le interfacce di rete e gli instradamenti.

Tutto questo, naturalmente, può essere modificato dall'amministratore durante il funzionamento del sistema, attraverso comandi opportuni, ma è bene che il meccanismo funzioni correttamente all'avvio, in modo da ridurre i problemi.

La maggior parte delle distribuzioni GNU è organizzata in modo che uno script di questa procedura di avvio del sistema sia destinato a essere eseguito per ultimo. Il nome è solitamente '`rc.local`' e potrebbe trovarsi nella directory '`/etc/rc.d/`', '`/etc/init.d/`', o più semplicemente in '`/etc/`'. Questo script è il luogo conveniente per aggiungere l'avvio di alcuni servizi eccezionali o per definire parte della configurazione di rete, quando non si riesce a intervenire in modo più elegante.

Superata la fase di avvio sotto il controllo della procedura di inizializzazione del sistema, Init mette in funzione i programmi Getty che si occupano di attivare la procedura di accesso attraverso i terminali previsti (console inclusa). L'accesso attraverso uno di questi terminali fa sì che venga avviata la shell definita per quell'utente particolare.

Le shell usuali utilizzano uno script di configurazione globale, collocato nella directory '`/etc/`' e almeno uno personalizzato nella directory personale dell'utente: prima viene eseguito quello globale, quindi quello personalizzato.

Gli script di configurazione delle shell sono utilizzati prevalentemente per definire alcune variabili di ambiente utili per controllare il comportamento della shell stessa e di tutti i programmi che ne possono avere bisogno.

16.11.1.3 Effetto

«

È importante rendersi conto che le variabili di ambiente sono delle entità definite all'interno di un processo e si trasmettono ai processi discendenti con gli stessi valori, fino a quando non vengono modificate in qualche modo.

Questo significa anche che processi paralleli, avviati dallo stesso utente, possono avere configurazioni differenti per ciò che riguarda le variabili di ambiente, proprio perché questo «ambiente» viene modificato.

I programmi consentono spesso l'utilizzo di una configurazione basata sulla combinazione dell'uso di file e di variabili di ambiente, dove queste ultime prevalgono.

16.11.2 Configurazione in base alla nazionalità: localizzazione

La configurazione più importante a cui dovrebbe provvedere ogni singolo utente, è la definizione della localizzazione. Attraverso questa, con i programmi che sono in grado di riconoscerla e di adattarsi di conseguenza, si può specificare il linguaggio, l'insieme di caratteri e altre opzioni che dipendono tipicamente dalle convenzioni nazionali e locali.

Questo tipo di configurazione avviene attraverso la definizione di variabili di ambiente opportune.

La sigla «i18n» rappresenta scherzosamente il termine *internationalization*, in quanto la prima e l'ultima lettera, «i» e «n», sono separate da 18 caratteri. Nello stesso modo e con lo stesso ragionamento, la sigla «l10n» rappresenta il termine *localization*.

16.11.2.1 Disponibilità della localizzazione

Prima di configurare determinate variabili per attivare la localizzazione nei programmi che ne sono predisposti, occorre verificare che il sistema sia in grado di fornire le informazioni necessarie ai programmi. Infatti, a parte l'uso di variabili di ambiente, cosa che rappresenta solo l'aspetto più esterno del problema, occorre che siano stati definiti i file di conversione per il tipo di localizzazione che si intende ottenere.

Si ottiene un elenco dei nomi utilizzabili per definire la localizzazione con il comando seguente:

```
$ locale -a[Invio]
```

Il vero problema nella localizzazione sta nel fatto che i nomi utilizzabili per definirla non sono standard e occorre almeno fare una piccola verifica in questo modo, una volta stabilito come si vuole agire.

I file di conversione utilizzati dal sistema per sostenere la localizzazione dovrebbero trovarsi a partire dalla directory '`/usr/share/locale/`', dalla quale si diramano tante directory quanti sono effettivamente i tipi di localizzazioni gestibili.

Se nell'elenco ottenuto non c'è ciò che serve alla propria lingua, è molto probabile che non siano state compilate le informazioni necessarie a partire dai sorgenti di queste. A tale proposito si può consultare il capitolo 13.

16.11.2.2 Scelta della definizione

«

La localizzazione, così come risulta organizzata nei sistemi Unix, può essere definita solo in base all'appartenenza a un certo paese, o al massimo, in alcuni casi, a una certa regione. Per la precisione, questa regionalizzazione si basa sulla scelta di una lingua e di una nazione (si pensi al caso della Svizzera che ha tre lingue nazionali). Eventualmente è consentito scegliere l'insieme di caratteri.

La tabella 16.95 mostra l'elenco di alcuni codici tipici per la definizione della localizzazione.

Tabella 16.95. Alcuni codici per la definizione della localizzazione.

Nome	Descrizione
`it_IT`	Lingua italiana, nazionalità italiana, codifica predefinita.
`it_IT.ISO-8859-1`	Lingua italiana, nazionalità italiana, codifica ISO 8859-1.
`it_IT.UTF-8`	Lingua italiana, nazionalità italiana, codifica UTF-8.
`de_DE`	Lingua tedesca, nazionalità tedesca, codifica predefinita.
`de_DE.ISO-8859-1`	Lingua tedesca, nazionalità tedesca, codifica ISO 8859-1.
`de_DE.UTF-8`	Lingua tedesca, nazionalità tedesca, codifica UTF-8.
`fr_FR`	Lingua francese, nazionalità francese, codifica predefinita.
`fr_FR.ISO-8859-1`	Lingua francese, nazionalità francese, codifica ISO 8859-1.
`fr_FR.UTF-8`	Lingua francese, nazionalità francese, codifica UTF-8.
`it_CH`	Lingua italiana, nazionalità svizzera, codifica predefinita.
`it_CH.ISO-8859-1`	Lingua italiana, nazionalità svizzera, codifica ISO 8859-1.
`it_CH.UTF-8`	Lingua italiana, nazionalità svizzera, codifica UTF-8.
`de_CH`	Lingua tedesca, nazionalità svizzera, codifica predefinita.
`de_CH.ISO-8859-1`	Lingua tedesca, nazionalità svizzera, codifica ISO 8859-1.
`de_CH.UTF-8`	Lingua tedesca, nazionalità svizzera, codifica UTF-8.

Nome	Descrizione
`fr_CH`	Lingua francese, nazionalità svizzera, codifica predefinita.
`fr_CH.ISO-8859-1`	Lingua francese, nazionalità svizzera, codifica ISO 8859-1.
`fr_CH.UTF-8`	Lingua francese, nazionalità svizzera, codifica UTF-8.
`de_AT`	Lingua tedesca, nazionalità austriaca, codifica predefinita.
`de_AT.ISO-8859-1`	Lingua tedesca, nazionalità austriaca, codifica ISO 8859-1.
`de_AT.UTF-8`	Lingua tedesca, nazionalità austriaca, codifica UTF-8.

Per l'Italia, la definizione corretta, completa, dovrebbe essere '**`it_IT.UTF-8`**', oppure '**`it_IT.ISO-8859-1`**' se si preferisce usare una codifica tradizionale.

Prima di proseguire, è il caso di insistere sul fatto che tra un sistema Unix e l'altro, le definizioni usate per distinguere i vari tipi di localizzazione potrebbero essere anche molto diverse. Seguono gli esempi di alcuni modi possibili, ma non sempre validi, per rappresentare la localizzazione italiana, specificando eventualmente la codifica UTF-8:

- **`it`**
- **`italian`**
- **`it_IT`**
- **`it_IT.UTF-8`**
- **`italian.UTF8`**

- **it_IT.utf-8**
- **it_IT.utf8**

16.11.2.3 Variabili per la localizzazione

Una volta stabilita la definizione da adottare per l'impostazione corretta della localizzazione, si deve passare alla «attivazione» delle variabili di ambiente desiderate, assegnando loro le scelte rispettive. Per controllare l'effetto di una configurazione particolare, basta usare '**locale**' senza argomenti.

LC_ALL

Questa variabile serve a definire in un colpo solo tutta la localizzazione, sovrapponendosi a tutte le altre variabili di ambiente destinate a questo scopo, qualunque sia il loro contenuto effettivo. Per questo motivo è decisamente **sconsigliabile** il suo utilizzo, almeno in una configurazione accurata.

Un buon motivo per evitare di utilizzare questa variabile è quello per cui alcuni applicativi, come Perl, non accettano l'incoerenza tra questa variabile e altre del gruppo ***LC_**∗*, rendendo inutile l'uso di una variabile che si impone sulle altre.

LANG

LANG permette di definire la localizzazione predefinita per le variabili del gruppo ***LC_**∗* che non siano state definite. Per questo, è molto importante definire e assegnare un valore alla variabile ***LANG***, in modo da garantire che siano considerati tutti i vari aspetti della localizzazione, anche se non specificati esplicitamente. Segue un esempio di script per configurare la variabile ***LANG*** secondo la localizzazione italiana predefinita:

```
#!/bin/sh
…
LANG=it_IT.UTF-8
export LANG
```

LC_COLLATE

Questa variabile permette di definire l'ordine dei caratteri, influenzando le operazioni di ordinamento (vero e proprio) e in generale quelle di confronto. Segue un esempio:

```
#!/bin/sh
…
LC_COLLATE=it_IT.UTF-8
export LC_COLLATE
```

LC_CTYPE

Questa variabile permette di definire l'insieme di caratteri. Ciò può avere effetto sulla loro rappresentazione, sull'abbinamento tra minuscole e maiuscole, sulla classificazione dei caratteri; per esempio: numerici, alfabetici, di punteggiatura e diversi. Segue un esempio:

```
#!/bin/sh
…
LC_CTYPE=it_IT.UTF-8
export LC_CTYPE
```

LC_NUMERIC

Questa variabile permette di definire il modo di rappresentazione dei numeri. A livello pratico, quello che si può ottenere è lo scambio tra il punto e la virgola per la rappresentazione della parte numerica decimale e per la separazione delle migliaia. Segue un esempio:

```
#!/bin/sh
…
LC_NUMERIC=it_IT.UTF-8
export LC_NUMERIC
```

LC_MONETARY

Questa variabile permette di definire il modo di rappresentazione delle valute: il simbolo di valuta, il numero di decimali da adottare e altre caratteristiche eventuali.

LC_TIME

Questa variabile permette di definire la rappresentazione delle informazioni data-orario. Si tratta di un'impostazione importante, perché, tra le altre cose, fa sì che i comandi di sistema restituiscano i nomi dei mesi e dei giorni della settimana in italiano. Segue un esempio:

```
#!/bin/sh
…
LC_TIME=it_IT.UTF-8
export LC_TIME
```

L'esempio successivo mostra come potrebbe essere visualizzata la data dal comando '**date**', quando la variabile ***LC_TIME*** è configurata per la localizzazione italiana.

```
$ date[Invio]

  dom ago  2 15:35:48 CEST 1998
```

16.11.2.4 Definizioni standard di localizzazione

Esistono due definizioni locali standard che è bene conoscere: '**C**' e '**POSIX**'. Entrambe rappresentano la stessa impostazione predefinita, in mancanza di altre definizioni. La distinzione tra i due nomi deriva dall'origine, rispettivamente lo standard del linguaggio C e lo standard dei sistemi POSIX. La definizione locale '**C**' riguarda la programmazione in linguaggio C, indipendente dal sistema operativo, mentre la definizione '**POSIX**' riguarda la programmazione (in C o in altri linguaggi), nell'ambito più ristretto dei sistemi operativi che aderiscono allo standard POSIX.

16.11.2.5 Utilizzo di «locale»

Il programma '**locale**'[48] permette di conoscere l'impostazione del proprio sistema di localizzazione ed è utile per verificare la configurazione delle variabili di ambiente relative.

```
locale [opzioni]
```

Tabella 16.102. Alcune opzioni.

Opzione	Descrizione
`-a` `--all-locale`	Emette l'elenco di tutti i nomi utilizzabili nelle definizioni di localizzazione.
`-m` `--charmaps`	Emette l'elenco di tutti i nomi riferiti a definizioni di mappe di caratteri.

Utilizzando ‘**locale**’ senza argomenti, si ottiene la situazione corrente dell’impostazione della localizzazione. Si supponga di ottenere quanto segue:

```
$ locale[Invio]

  LANG=POSIX
  LC_CTYPE=it_IT
  LC_NUMERIC="POSIX"
  LC_TIME=it_IT.UTF-8
  LC_COLLATE="POSIX"
  LC_MONETARY="POSIX"
  LC_MESSAGES="POSIX"
  LC_ALL=
```

Quanto ottenuto in questo esempio rappresenta l’impostazione delle sole variabili ***LC_CTYPE*** e ***LC_TIME***, con valori simili e di fatto equivalenti. Tutte le altre variabili, non essendo state definite, sono impostate secondo la localizzazione ‘**POSIX**’, ovvero quella predefinita in un sistema Unix aderente allo standard.

16.11.3 Insieme di caratteri

In linea di massima, la localizzazione definita attraverso le variabili di ambiente ***LC_**∗***, descritte nelle sezioni precedenti, dovrebbe essere sufficiente per stabilire implicitamente anche le esigenze relative all’insieme dei caratteri utilizzato per la visualizzazione dei dati. In pratica, la localizzazione ‘**it_IT.UTF-8**’ dovrebbe stabilire che l’insieme dei caratteri è quello universale (ISO 10646), secondo la codifica UTF-8. In pratica, alcuni programmi ignorano la localizzazione, oppure sono configurati in modo predefinito in senso contrario.

16.11.3.1 Variabile di ambiente «LESSCHARSET»

«

Il programma '**less**', utilizzato generalmente per lo scorrimento a video del testo delle pagine di manuale, è sensibile al contenuto della variabile ***LESSCHARSET***. In situazioni particolari, per visualizzare correttamente del testo che contenga lettere accentate e altri simboli utilizzati nella codifica UTF-8 (insieme di caratteri universale ISO 10646), potrebbe essere necessario che contenga la stringa '**utf-8**'.

```
#!/bin/sh
…
LESSCHARSET=utf-8
export LESSCHARSET
```

L'esempio mostra un pezzo di uno script attraverso cui viene definita la variabile ***LESSCHARSET*** nel modo descritto.

16.11.3.2 File «/etc/man.config» o «/etc/manpath.config»

«

Il programma '**man**' potrebbe essere configurato attraverso il file '`/etc/man.config`' o '`/etc/manpath.config`'. Questo file serve a definire i comportamenti di '**man**' e in particolare gli argomenti da utilizzare per i programmi usati per la formattazione del testo della documentazione tradizionale.

I programmi '**groff**' e '**geqn**', quando vengono usati per generare il testo da visualizzare a video (testo che poi viene gestito attraverso '**more**' o '**less**'), potrebbero richiedere l'uso dell'opzione '**-T**' con l'argomento '**utf8**' (quando si intende utilizzare tale codifica), in modo da consentire l'emissione di caratteri secondo la codifica UTF-8. Nel caso si dovessero riscontrare problemi a visualizzare le

lettere accentate, la configurazione con il file '`/etc/man.config`' potrebbe essere cambiata in modo simile a quella seguente:

```
TROFF           /usr/bin/groff -Tps -mandoc
NROFF           /usr/bin/groff -Tutf8 -mandoc
EQN             /usr/bin/geqn -Tps
NEQN            /usr/bin/geqn -Tutf8
TBL             /usr/bin/gtbl
# COL           /usr/bin/col
REFER           /usr/bin/grefer
PIC             /usr/bin/gpic
VGRIND
GRAP
PAGER           /usr/bin/less -is
CAT             /bin/cat
```

Nell'esempio appena mostrato si vede in particolare l'uso dell'opzione '**`-Tutf8`**' per '**`groff`**' e '**`geqn`**', quando questi programmi servono per generare testo da visualizzare attraverso lo schermo a caratteri. Nell'esempio successivo si vede come potrebbe essere necessario modificare il file '`/etc/manpath.config`':

```
DEFINE  pager   exec /usr/bin/pager -s
DEFINE  cat     /bin/cat
DEFINE  tr      /usr/bin/tr '\255\267\264\327' ↩
↪'\055\157\047\170'
DEFINE          grep    /bin/grep
DEFINE  troff   /usr/bin/groff -mandoc
DEFINE  nroff   /usr/bin/nroff -Tutf8 -mandoc
DEFINE  eqn     /usr/bin/eqn
DEFINE  neqn    /usr/bin/neqn -Tutf8
DEFINE  tbl     /usr/bin/tbl
DEFINE  col     /usr/bin/col
DEFINE  vgrind  /usr/bin/vgrind
DEFINE  refer   /usr/bin/refer
DEFINE  grap    /usr/bin/grap
```

```
DEFINE  pic     /usr/bin/pic -S

DEFINE          decompressor    /bin/gzip -dc
DEFINE          compressor      /bin/gzip -c7
```

Si osservi, comunque, che nelle situazioni comuni, l'uso dell'opzione '**-T**' non è necessario, perché i programmi rispondono correttamente alla configurazione stabilita con le variabili di ambiente ***LC_**** e ***LANG***.

16.11.4 Configurazioni comuni varie

«

Alcuni tipi di configurazione comune, sono di minore importanza e in parte già descritti altrove in questo documento, ma può essere utile raccoglierli come riferimento.

16.11.4.1 Invito della shell

«

Per quanto banale, la configurazione dell'invito della shell può essere molto importante. Il suo aspetto e la sua configurazione dipendono dalla shell stessa.

Chi utilizza una shell standard, o comunque derivata da quella di Bourne deve impostare la variabile di ambiente ***PS1***. Nel caso di Bash si può utilizzare eventualmente la definizione seguente, nel file '`/etc/profile`', se deve riguardare la configurazione standard per tutti gli utenti, oppure nel file '`~/.bash_profile`' se si tratta della configurazione personale (o in mancanza il file '`~/.profile`').

```
PS1='\u@\h:\w\$ '
export PS1
```

Sempre nell'ipotesi di una shell Bash, potrebbe essere piacevole avere un modo per visualizzare il successo o meno dell'esecuzione dell'ultimo programma; in pratica, si tratta di un modo per controllare il contenuto del parametro '**$?**':

```
dynamic_prompt () {
    if [ $? = 0 ]
    then
        echo ":)"
    else
        echo ":("
    fi
}
export -f dynamic_prompt
PS1='\u@\h:\w\$ '
if [ "$BASH" != "" ]
then
    # This is BASH.
    PS1="\$(dynamic_prompt) $PS1"
fi
export PS1
```

Come si vede, si tratta di una funzione, denominata '**dynamic_prompt**', che viene utilizzata nella stringa della variabile ***PS1*** solo se ci si accerta che si tratta proprio della shell Bash.

16.11.4.2 Prevenzione dalla cancellazione involontaria

Più volte, in questo documento, è ripetuto quanto sia facile eliminare inavvertitamente dei file, per un utilizzo improprio del comando di cancellazione, '**rm**', oppure per una sovrascrittura involontaria attraverso la copia o lo spostamento dei file.

La shell Bash permette di creare degli alias a comandi normali, definendo l'utilizzo sistematico di opzioni determinate. I comandi seguenti definiscono tre alias ai comandi '**rm**', '**cp**' e '**mv**', in modo che venga usata sempre l'opzione '**-i**', con la quale si ottiene una richiesta di conferma nel momento in cui si richiede la cancellazione di un file per qualunque motivo.

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```

Successivamente, per evitare la seccatura di dover confermare la cancellazione o la sovrascrittura di file, è sufficiente utilizzare l'opzione '**-f**' (*force*).

16.11.4.3 Libreria Readline

«

Molti programmi che funzionano in modo interattivo mostrando un invito all'inserimento dei comandi (un *prompt*) e offrendo una riga di comando, sfruttano una libreria molto sofisticata per farlo: si tratta generalmente della libreria Readline. La shell Bash è l'applicativo più comune che utilizza questa libreria.

Può essere utile definire la configurazione di questa libreria attraverso il file '`~/.inputrc`' (il file di configurazione generale, '`/etc/inputrc`', potrebbe essere ignorato), in modo da facilitare l'uso della tastiera e l'inserimento di caratteri che utilizzano anche l'ottavo bit. L'esempio seguente si riferisce alla configurazione necessaria per l'uso ottimale di una console virtuale su un elaboratore con architettura x86.

```
# Abilita l'inserimento di caratteri a 8 bit.
set meta-flag            on
# Disabilita la conversione dei caratteri con l'ottavo bit
```

```
# attivo in sequenze di escape.
set convert-meta        off
# Abilita la visualizzazione di caratteri a 8 bit.
set output-meta         on
# Modifica l'abbinamento con i tasti rispetto a determinati
# comportamenti.
"\e[1~": beginning-of-line      # [home]            era C-a
"\e[4~": end-of-line            # [fine]            era C-e
"\e[3~": delete-char            # [canc]            era C-d
"\e[5~": backward-word          # [pagina su]       era M-b
"\e[6~": forward-word           # [pagina giù]      era M-f
```

16.11.5 Fuso orario

Nei sistemi Unix in generale, l'orologio «fisico» dell'elaboratore viene regolato sul tempo universale (UT, in passato noto come GMT), in modo tale che il sistema operativo possa fornire l'ora locale in base alla configurazione, la quale potrebbe variare anche a livello di ogni utente.

Lo standard comune prevede la presenza di un file di configurazione costituito da '`/usr/share/zoneinfo/localtime`'; tuttavia, se esiste la variabile di ambiente ***TZ*** (*Time zone*), il suo contenuto prende il sopravvento.

Il file '`/usr/share/zoneinfo/localtime`' deve essere realizzato secondo un formato particolare, pertanto sono spesso presenti file già pronti per i vari fusi orari utilizzati, così che '`/usr/share/zoneinfo/localtime`' può essere semplicemente un collegamento simbolico al file effettivo. Per maggiore semplicità, succede normalmente che '`/usr/share/zoneinfo/localtime`' sia un colle-

gamento simbolico a ‘`/etc/localtime`’, che a sua volta è un altro collegamento simbolico al file che contiene l’informazione.

A titolo di esempio, per fare riferimento al fuso orario che riguarda le convenzioni italiane, si prende in considerazione il file ‘`/usr/share/zoneinfo/Europ/Rome`’ e si puntano su questo file, direttamente o indirettamente, i collegamenti simbolici ‘`/usr/share/zoneinfo/localtime`’ e ‘`/etc/localtime`’.

La configurazione, eventuale, della variabile di ambiente ***TZ***, prevede l’indicazione delle informazioni in diversi modi alternativi; per maggiori dettagli, si veda la pagina di manuale *tzset(3)*.

16.12 Limiti alle utenze

«

Nella gestione di sistemi con molte utenze, diventa presto importante trovare un modo semplice per limitare l’accesso o le risorse loro concesse. In questo capitolo si annotano alcune soluzione «semplici» dal punto di vista realizzativo.

16.12.1 Una shell per impedire l’accesso

«

Un metodo molto semplice per impedire l’accesso a un utente, tanto più se si tratta di un utente fittizio, il quale non può e non deve avere materialmente accesso, consiste nell’attribuire come shell un programma che si comporti in modo diverso dal previsto. Di solito, per queste cose si usa il programma ‘**`false`**’ che, tradizionalmente, si limita a restituire un valore che rappresenta un errore, cosa che generalmente si considera equivalere a *Falso*. Ecco un esempio estratto da un file ‘`/etc/passwd`’:

```
…
daemon:x:1:1:daemon:/usr/sbin:/bin/false
bin:x:2:2:bin:/bin:/bin/false
sys:x:3:3:sys:/dev:/bin/false
…
```

È evidente che tali utenti non hanno nemmeno una parola d'ordine valida, ma questa prudenza ulteriore non può essere dannosa.

Quando si vuole usare un metodo del genere per delle utenze reali, allo scopo di impedire temporaneamente l'accesso per qualche ragione, può essere più conveniente l'uso di un programma specifico, per dare delle informazioni all'utente che viene allontanato dal sistema. A questo proposito si può usare, per esempio, il programma '**`falselogin`**',[49] come nel modo seguente nel file '`/etc/passwd`':

```
…
tizio:x:499:499:Tizio,,,:/home/tizio:/usr/bin/falselogin
caio:x:498:498:Caio,,,:/home/caio:/usr/bin/falselogin
sempronio:x:497:497:Sempronio,,,:/home/sempronio:↩
↪/usr/bin/falselogin
mevio:x:496:496:Mevio,,,:/home/mevio:/usr/bin/falselogin
filano:x:495:495:Filano,,,:/home/filano:/usr/bin/falselogin
martino:x:494:494:Martino,,,:/home/martino:↩
↪/usr/bin/falselogin
calpurnio:x:493:493:calpurnio,,,:/home/calpurnio:↩
↪/usr/bin/falselogin
…
```

Lo scopo di '**`falselogin`**' è quello di mostrare all'utente un messaggio, che viene configurato con il file '`/etc/falselogin.conf`'. Quello che segue è l'esempio del file di configurazione predefinito nella distribuzione GNU/Linux Debian:

```
Welcome at %host% (Debian %debian_version% %sysname% %release%)!

%mail%

Sorry %user% but our server does not accept shell logins.
So long and thanks for all the fish.
```

Come si può intuire, sono disponibili delle metavariabili indicate secondo la forma '**%*nome*%**'. L'elenco completo di queste è disponibile nella pagina di manuale *falselogin(1)*.

Qualunque sia il programma che si intende indicare in funzione di shell (reale o fittizia che sia), è necessario che questo sia previsto nel file '`/etc/shells`':

```
…
/bin/false
/usr/bin/falselogin
…
```

16.12.2 Controllo dello spazio utilizzato, senza l'uso tradizionale delle quote

«

In certi casi, la gestione delle quote (sezione 19.6) può essere scomoda o creare una complicazione eccessiva. Se si limita agli utenti la disponibilità di poche shell compatibili nell'uso dei file di configurazione, si può intervenire proprio su questi per svolgere una serie di controlli prima di mettere l'utente in condizione di operare.

Se si può contare sull'uso del file '`/etc/profile`' per un gruppo limitato di shell più o meno compatibili con lo standard POSIX, si può aggiungere in coda a tale script il codice seguente:[50]

```
1  # Check "$HOME" usage
2  if [ "$USER" != "root" ]
3  then
4      HOME_DISK_SPACE_ALLOWED="10000000"
5      echo ""
6      echo "Please wait for disk space usage verification..."
7      echo ""
8      # Calculate user's disk space usage.
9      HOME_DISK_SPACE_USED=`du -bs $HOME 2> /dev/null | sed "s/\t.*$//"`
10     HOME_DISK_SPACE_USED_PERCENTAGE=
   ↪$(($HOME_DISK_SPACE_USED*100/$HOME_DISK_SPACE_ALLOWED))
11     echo "Your disk usage is $HOME_DISK_SPACE_USED bytes."
12     echo "You are allowed to use up to $HOME_DISK_SPACE_ALLOWED bytes."
13     echo "You are using $HOME_DISK_SPACE_USED_PERCENTAGE%
   ↪of the allowed disk space."
14     echo ""
15     if [ "$HOME_DISK_SPACE_USED_PERCENTAGE" -gt "100" ]
16     then
17         echo "YOU ARE REQUIRED TO REDUCE YOUR DISK USAGE TO THE
   ↪ALLOWED VALUE!"
18         echo "If you don't do it alone, your account might be removed
   ↪by the administrator."
19     fi
20     echo ""
21 fi
```

Lo scopo è, come si può intuire, quello di informare l'utente, contando sulla sua collaborazione. È evidente che lo script può essere reso più efficace, per esempio inviando un messaggio di posta elettronica all'amministratore quando un utente supera lo spazio consentito, arrivando anche a interdire l'utenza se si l'utente non provvede. Per esempio, si potrebbe intervenire così:

```
11      echo "Your disk usage is $HOME_DISK_SPACE_USED bytes."
12      echo "You are allowed to use up to $HOME_DISK_SPACE_ALLOWED bytes."
13      echo "You are using $HOME_DISK_SPACE_USED_PERCENTAGE% ↩
    ↪of the allowed disk space."
14      echo ""
15      if   [ "$HOME_DISK_SPACE_USED_PERCENTAGE" -gt "200" ]
16      then
17          echo "YOUR ACCOUNT IS LOCKED!"
18          echo "Please contact the administrator."
19          echo "account $USER locked" | mail root
20          exit
21      elif [ "$HOME_DISK_SPACE_USED_PERCENTAGE" -gt "100" ]
22      then
23          echo "YOU ARE REQUIRED TO REDUCE YOUR DISK USAGE TO THE ↩
    ↪ALLOWED VALUE!"
24          echo "If you don't do it alone, your account might be removed ↩
    ↪by the administrator."
25      fi
26      echo ""
27  fi
```

Vale la pena di descrivere alcuni comandi che possono risultare un po' complessi a prima vista. Nella riga numero 13 si vede l'uso del comando '**du**' per contare lo spazio utilizzato a partire dalla directory personale dell'utente:

```
du -bs $HOME
```

L'opzione '**-bs**' serve a richiedere un conteggio complessivo, espresso in byte. Il risultato viene filtrato da '**sed**' per conservare solo l'informazione numerica, infatti, ciò che emette '**du**' potrebbe essere un testo simile a quello seguente:

```
29546091        /home/tizio
```

Dal momento che tra il numero che esprime lo spazio utilizzato e la directory c'è esattamente un carattere di tabulazione (il carattere <*HT*>), '**sed**' va a cercare proprio quello ed elimina tutto il resto. Alla fine, il valore viene assegnato alla variabile di ambiente

HOME_DISK_SPACE_USED.

Nella riga numero 15 viene eseguito un calcolo, assegnando il risultato alla variabile ***HOME_DISK_SPACE_USED_PERCENTAGE***.

16.12.3 Accesso consentito soltanto ad alcuni utenti

Quando si condividono le stesse utenze in una rete locale (si veda la sezione sul NIS: 36.4), può capitare che si voglia evitare di consentire l'accesso agli utenti comuni presso un elaboratore particolare. Si può svolgere un controllo di questo tipo, rifiutando l'accesso a tutti gli utenti, tranne l'amministratore e altre utenze particolari, utilizzando la stessa tecnica già mostrata nella sezione precedente, intervenendo nel file '`/etc/profile`':[51]

```
# Refuse access to most users.
USER_ALLOWED="0"
for u in root tizio caio
do
    if [ "$u" = "$USER" ]
    then
        USER_ALLOWED="1"
    fi
done
if [ "$USER_ALLOWED" = "0" ]
then
    # The user is not allowed.
    echo "You are not allowed to use this computer."
    exit
fi
```

16.13 Samba e utenze Unix

Samba è un programma servente che offre dei servizi di rete, tali da consentire a elaboratori con sistemi MS-Windows di accedere a risorse condivise. Samba è in grado di gestire il protocollo SMB/CIFS (*Server message block*) e anche NetBIOS. In questo capitolo si vuole considerare la possibilità di condividere le stesse utenze Unix, in modo che da elaboratori con sistemi MS-Windows, ogni utente possa accedere alla propria directory personale presso un elaboratore funzionante con un sistema GNU/Linux. Tuttavia, i dettagli sulla configurazione della rete non vengono affrontati (riguardano i capitoli a partire da 32), inoltre la configurazione stessa della gestione delle utenze di Samba viene considerata nel modo più simile al modello Unix tradizionale.

16.13.1 Avvio del servizio di rete

Samba può essere avviato in due modi: come demone indipendente, oppure sotto il controllo del supervisore dei servizi di rete (sezione 36.1). La prima ipotesi è la migliore se si attende un uso frequente del servizio, mentre la seconda va bene per una rete molto contenuta. Generalmente è la stessa procedura di installazione a chiedere all'utente come va predisposto il servizio di Samba e diversamente occorre approfondire la questione nella documentazione originale. Eventualmente, va tenuto presente che per una gestione del servizio in modo indipendente dal supervisore dei servizi di rete, occorre provvedere ad avviare i demoni '**nmbd**' (per il protocollo NetBIOS) e '**smbd**' (per il protocollo SMB/CIFS); pertanto vanno lette le pagine di manuale *nmbd(8)* e *smbd(8)*:

```
# nmbd -D [Invio]
```

```
# smbd -D [Invio]
```

L'opzione '**-D**', evidenziata negli esempi, rappresenta la richiesta ai programmi di funzionare sullo sfondo, come demoni.

16.13.2 Configurazione essenziale

La configurazione di Samba avviene con il file '`/etc/samba/smb.conf`'. L'esempio seguente rappresenta un modello generalizzato per la gestione di un servizio di condivisione delle directory personali degli utenti, attraverso i protocolli di MS-Windows. Logicamente, la stringa «il mio servente Samba» e i due nomi «miosmb», vanno modificati secondo la propria preferenza:

```
[global]
    server string = il mio servente Samba
    workgroup = MIOSMB
    netbios name = miosmb
    hosts allow = 127.0.0.0/8 10.0.0.0/8 172.16.0.0/12 ↩
↪                 192.168.0.0/16
    local master = no
    domain master = no
    security = user
    hostname lookups = no
    dns proxy = no
    log file = /var/log/samba/log.%h
    max log size = 0
    log level = 1
    encrypt passwords = true
    smb passwd file = /etc/samba/smbpasswd
    passdb backend = smbpasswd:/etc/samba/smbpasswd
```

```
    invalid users = root shutdown reboot nobody daemon ↩
↪                   bin sys sync games man lp mail news ↩
↪                   uucp proxy dip postgres www-data ↩
↪                   backup msql operator list irc gnats ↩
↪                   ftp sshd clamav bind saned ↩
↪                   dansguardian partimag wims fetchmail ↩
↪                   messagebus Debian-console-log

[homes]
    comment = personal data
    browseable = no
    writable = yes
    create mask = 0755
    directory mask = 0755
```

La tabella successiva descrive alcune direttive relative alla sezione globale (‘**global**’) del file di configurazione. La sezione ‘**homes**’ riguarda specificatamente l’accessibilità delle directory personali degli utenti: il senso delle direttive dell’esempio dovrebbe essere intuitivo; in particolare si può osservare la maschera per la creazione dei file e delle directory che, nell’esempio, consente di ottenere file e directory accessibili e leggibili da chiunque.

Tabella 16.121. Alcune direttive per la configurazione di Samba, nella sezione ‘**global**’.

Direttiva	Descrizione
`server string =` ***descrizione***	Descrizione del servente.
`workgroup =` ***nome***	Nome del gruppo di lavoro a cui appartiene l’elaboratore.
`netbios name =` ***nome***	Nome dell’elaboratore per il protocollo NetBIOS.
`hosts allow =` ***elenco***	Elenco di insiemi di elaboratori che possono accedere al servizio.

Direttiva	Descrizione
`security = user`	Seleziona il tipo di controllo di accesso. In questo caso il controllo è a livello di utente.
`log file =` ***modello_file***	Definisce il file o i file da usare per annotare ciò che succede. Nell'esempio si usa il modello '**`/var/log/samba/log.%h`**', con il quale si crea un solo file denominato '`log.`***hostname***' (ovvero «log.» seguito dal nome che restituisce il comando '**`hostname`**') nella directory '`/var/log/samba/`'.
`log level =` ***n***	Definisce il livello di dettaglio delle annotazioni fatte nel file delle registrazioni. Il livello uno, come appare nell'esempio, è sufficiente per ottenere un rapporto degli accessi remoti.
`encrypt passwords = true` `encrypt passwords = false`	Specifica se le parole d'ordine usate per l'autenticazione debbano viaggiare in forma cifrata o meno. Generalmente va attivata questa opzione ('**`true`**'), a meno di configurare diversamente i clienti MS-Windows.

Direttiva	Descrizione
`passdb backend = smbpasswd:`***file*** `smb passwd file =` ***file***	Queste due direttive, assieme, definiscono il modo in cui Samba conserva le informazioni sugli utenti e il file che deve essere usato in pratica. I modelli mostrati sono scelti appositamente per usare la forma di un file di testo normale, vagamente simile a '`/etc/passwd`'.
`invalid users =` ***utenti***	Consente di escludere l'accesso di alcuni utenti, tipicamente '**`root`**' e altri utenti speciali.

16.13.3 Elenco degli utenti

«

Secondo l'esempio di configurazione proposto nella sezione precedente, le utenze di Samba vengono annotate in un file di testo comune, strutturato concettualmente in modo simile a '`/etc/passwd`'. Per la precisione le righe di questo file hanno la struttura seguente:

utente **:** ***uid*** **:** ***pwd_1*** **:** ***pwd_2*** **:** ***opzioni*** **:** ***ultima_modifica***

Come si vede, appaiono due versioni della stessa parola d'ordine cifrata; la prima serve a sistemi MS-Windows 95/98, mentre la seconda serve a sistemi MS-Windows NT e conformi. L'aspetto di una di queste righe è simile all'esempio seguente:

```
…
tizio:1001:981BB8DA…D3745EDF4:3C9CFFE…C0FD6:[U          ]:LCT-4723A492:
…
```

Il campo delle opzioni contiene delle lettere tra parentesi quadre; nell'esempio, la lettera «U» indica che si tratta di un utente comune. L'ultimo campo che contiene la data di ultima modifica dell'utenza, inizia con la sigla «LCT» (*Local change time*) e prosegue con un trattino e poi un numero che rappresenta il tempo trascorso a partire dall'epoca di riferimento (l'ora zero del primo gennaio 1970).

16.13.4 Gestione delle utenze

Per la gestione delle utenze di Samba si usa il programma '**smbpasswd**'; tuttavia va osservato che le utenze che si creano per Samba devono essere già esistenti nel file '`/etc/passwd`' (Unix), mentre la loro eliminazione riguarda solo la rimozione dal file '`/etc/samba/smbpasswd`', o da qualunque altro sistema Samba utilizzi per tale funzione.

```
smbpasswd [opzioni] [utente]
```

Questo programma, se usato senza argomenti, si limita a cambiare la parola d'ordine dell'utente Unix attuale, usata però per accedere da un sistema MS-Windows. Attraverso l'uso di opzioni e con l'indicazione di un nominativo utente è possibile, tra le altre cose, eliminare un'utenza di Samba o crearne una nuova. I comandi significativi sono quelli seguenti:

- # **smbpasswd -s -a** ***utente*** [*Invio*]

  in questo modo si crea un'utenza nuova, la quale deve però avere già una corrispondenza con un'utenza Unix;

- # **smbpasswd -x** ***utente*** [*Invio*]

in questo modo si elimina un'utenza di Samba.

Il cambiamento della parola d'ordine necessaria per accedere all'utenza di Samba, può essere cambiata anche eliminando e ricreando l'utenza.

16.13.5 Allineamento delle utenze

Quando si utilizza Samba, secondo le modalità descritte in questo capitolo, per una gestione di utenze affiancata a quella in stile Unix, si pone evidentemente il problema di gestire uniformemente le due cose, soprattutto per ciò che riguarda la parola d'ordine necessaria per accedere. Viene qui proposto un metodo molto «semplice», almeno sul piano realizzativo, attraverso l'uso di script personali.

Il brano seguente, relativo a uno script per una shell POSIX, mostra in che modo potrebbe essere creata un'utenza, sia per la gestione in stile Unix, sia per Samba, ma tutto è molto semplificato e privo di controlli, pertanto va poi esteso e migliorato secondo le proprie abilità:

```
...
# $UTENTE è il nome dell'utente creato.
# $GECOS  è la descrizione dell'utente (priva di caratteri
# vietati).
# $PASSWD è la parola d'ordine stabilita per accedere.
...
# Aggiunge l'utente Unix.
adduser --disabled-password \
        --no-create-home \
        --home /home/$UTENTE \
        --gecos "$GECOS"\
        $UTENTE
# Crea la directory personale Unix.
```

```
cp -dpR /etc/skel /home/$UTENTE
chown -R $UTENTE: /home/$UTENTE
# Elimina l'utente per Samba, nel caso dovesse esistere già.
smbpasswd -x $UTENTE 2> "/dev/null"
# Attribuisce la parola d'ordine.
if ( sleep 1 ; echo $PASSWD ; sleep 1 ; echo $PASSWD ) \
   | passwd $UTENTE 2> /dev/null
then
    if ( sleep 1 ; echo $PASSWD ; sleep 1 ; echo $PASSWD ) \
       | smbpasswd -s -a $UTENTE 2> /dev/null
    then
        true
    else
        echo "Non posso creare l'utenza Samba!"
    fi
else
    echo "Non posso creare l'utenza Unix!"
fi
…
```

Il brano successivo mostra come si potrebbe procedere per cambiare la parola d'ordine a un'utenza già operativa, ma come si vede si tratta solo di una riduzione dell'esempio già fatto, in quanto per Samba si procede con la rimozione dell'utenza e la sua nuova creazione:

```
…
# $UTENTE è il nome dell'utente creato.
# $PASSWD è la nuova parola d'ordine stabilita per accedere.
…
# Elimina l'utente per Samba.
smbpasswd -x $UTENTE 2> "/dev/null"
# Attribuisce la parola d'ordine.
if ( sleep 1 ; echo $PASSWD ; sleep 1 ; echo $PASSWD ) \
   | passwd $UTENTE 2> /dev/null
```

```
then
    if ( sleep 1 ; echo $PASSWD ; sleep 1 ; echo $PASSWD ) \
       | smbpasswd -s -a $UTENTE 2> /dev/null
    then
        true
    else
        echo "Non posso creare l'utenza Samba!"
    fi
else
    echo "Non posso modificare la parola d'ordine "
    echo "dell'utenza Unix!"
fi
…
```

16.14 Sintesi dei comandi principali

«

Comando	Descrizione
`pwconv`	Genera o aggiorna il file '`/etc/shadow`' a partire dal file '`/etc/passwd`'.
`pwunconv`	Elimina il file '`/etc/shadow`' mettendo le parole d'ordine cifrate nel file '`/etc/passwd`'.
`useradd` ***utente***	Aggiunge l'utente specificato nel file '`/etc/passwd`' e se esiste anche da '`/etc/shadow`'; senza occuparsi di altre questioni.
`adduser` ***utente***	Aggiunge l'utente specificato nel file '`/etc/passwd`' e se esiste '`/etc/shadow`', chiedendo interattivamente tutte le altre informazioni che servono, creando anche la directory personale.

Comando	Descrizione
`smbpasswd [-s] ↩ ↪ -a` ***utente***	Aggiunge l'utente specificato nella gestione di Samba (potrebbe essere il file '`/etc/samba/smbpasswd`').
`userdel` ***utente***	Elimina l'utente specificato dal file '`/etc/passwd`' e se esiste anche da '`/etc/shadow`'.
`deluser` ***utente***	Elimina l'utente specificato dal file '`/etc/passwd`' e se esiste anche da '`/etc/shadow`', occupandosi eventualmente anche dell'eliminazione della directory personale.
`smbpasswd -x` ***utente***	Elimina l'utente specificato nella gestione di Samba.
`passwd [`***utente***`]`	Cambia la parola d'ordine (Unix) utilizzata per accedere al sistema, aggiornando il file '`/etc/passwd`' o il file '`/etc/shadow`'.
`chage` ***utente***	Visualizza i tempi annotati nel file '`/etc/shadow`' riferiti all'utente indicato.
`pwck`	Verifica la coerenza del file '`/etc/passwd`' e se esiste anche di '`/etc/shadow`'.
`groupadd` ***gruppo*** `groupdel` ***gruppo***	Aggiunge o elimina un gruppo modificando il file '`/etc/group`' ed eventualmente anche il file '`/etc/gshadow`'.
`grpconv`	Genera o aggiorna il file '`/etc/gshadow`' a partire dal file '`/etc/group`'.
`grpunconv`	Elimina il file '`/etc/gshadow`' aggiornando per quanto possibile il file '`/etc/group`'.

Comando	Descrizione
`gpasswd` [***gruppo***]	Attribuisce o cambia la parola d'ordine associata al gruppo, aggiornando il file '`/etc/group`' o il file '`/etc/gshadow`'.
`grpck`	Verifica la coerenza del file '`/etc/group`' e di '`/etc/gshadow`' se esiste (utilizzando anche il file '`/etc/passwd`').
`su` [`-`] [***utente***]	Consente di cambiare utenza temporaneamente. Se si usa il segno '**–**' tra gli argomenti si ottiene l'ambiente normale che si otterrebbe attraverso un ingresso normale nel sistema (senza '**su**').
`newgrp` [`-`] [***gruppo***]	Consente di cambiare gruppo temporaneamente. Se si usa il segno '**–**' tra gli argomenti si ottiene l'ambiente normale che si otterrebbe attraverso un ingresso normale nel sistema.
`users`	Visualizza l'elenco degli utenti che stanno utilizzando il sistema.
`who`	Visualizza l'elenco degli utenti che stanno utilizzando il sistema, con l'indicazione del terminale da cui accedono.
`w`	Visualizza l'elenco degli utenti che stanno utilizzando il sistema, con più informazioni sulla loro attività.
`finger` [***utente***]	Fornisce informazioni su tutti gli utenti connessi all'elaboratore locale o soltanto sull'utente indicato (che può anche non essere connesso in quel momento).
`whoami`	Mostra il nominativo utente, associato al numero UID efficace, utilizzato dalla stessa persona che dà il comando.

Comando	Descrizione
`logname`	Mostra il nominativo utente utilizzato per accedere al sistema.
`chsh` [*utente*]	Cambia la shell utilizzata dall'utente.
`chfn` [*utente*]	Cambia le informazioni personali associate all'utente.
`groups`	Elenca i gruppi a cui è associato l'utente.
`id`	Elenca le informazioni sull'utenza (utente, gruppi, numeri UID e GID).
`last`	Visualizza gli ultimi accessi annotati nel file '`/var/log/wtmp`'.
`logger` ***messaggio***	Aggiunge un'annotazione nel registro del sistema.

16.15 Riferimenti

«

- Michael H. Jackson, *Linux Shadow Password HOWTO*, *http://tldp.org/HOWTO/Shadow-Password-HOWTO.html*
- Andrew G. Morgan, Thorsten Kukuk, *The Linux-PAM system administrators' guide*, *http://dainf.ct.utfpr.edu.br/~maziero/lib/exe/fetch.php/sas:linux-pam_system_administrators_guide.pdf*
- *Samba*, *http://www.samba.org*
- Fulvio Ferroni, *Samba e OpenLDAP*, *http://linuxdidattica.org/docs/altre_scuole/planck/samba/*

[1] **Linux' system and kernel logging daemons: syslogd** UCB BSD

[2] **BSD utils** UCB BSD

[3] **Linux' system and kernel logging daemons: klogd** GNU GPL

[4] **Linux' system and kernel logging daemons: syslogd** UCB BSD

[5] **Debianutils: savelog** GNU GPL

[6] **Logrotate** GNU GPL

[7] **Console-log** GNU GPL

[8] La parola *login* va pronunciata separando le due sillabe: «log-in». Lo stesso valga per la parola *logout* che va pronunciata: «log-out».

[9] **Shadow utilities** software libero con licenza speciale

[10] **Shadow utilities** software libero con licenza speciale

[11] **GNU core utilities** GNU GPL

[12] **Procps w** GNU GPL

[13] Si può verificare facilmente che la colonna '**`IDLE`**' del comando '**`w`**' riporta il tempo di funzionamento complessivo; per farlo basta avviare in un terminale un programma che utilizza intensamente la CPU, come '**`yes`**' quando è ridiretto verso '`/dev/null`', cronometrando il tempo e controllando ciò che riporta '**`w`**'.

[14] **GNU core utilities** GNU GPL

[15] **GNU core utilities** GNU GPL

[16] **GNU core utilities** GNU GPL

[17] **GNU core utilities** GNU GPL

[18] **GNU core utilities** GNU GPL

[19] **GNU core utilities** GNU GPL

[20] Naturalmente, questo vale finché nessuno riesce a trovare un algoritmo inverso che permetta di ricalcolare la parola d'ordine a partire

dalla stessa stringa cifrata.

[21] **Shadow utilities** software libero con licenza speciale

[22] In generale, i sistemi pongono anche un limite superiore alla lunghezza delle parole d'ordine. In tali casi, può capitare che la parte eccedente tale dimensione venga semplicemente ignorata, rendendo vano lo sforzo dell'utente.

[23] **Shadow utilities** software libero con licenza speciale

[24] **Shadow utilities** software libero con licenza speciale

[25] Questo metodo di comportamento è quello predefinito di alcune distribuzioni GNU.

[26] **Shadow utilities** software libero con licenza speciale

[27] **Shadow utilities** software libero con licenza speciale

[28] **Shadow utilities** software libero con licenza speciale

[29] **Shadow utilities** software libero con licenza speciale

[30] **Shadow utilities** software libero con licenza speciale

[31] **Shadow utilities** software libero con licenza speciale

[32] **Shadow utilities** software libero con licenza speciale

[33] **Shadow utilities** software libero con licenza speciale

[34] **Shadow utilities** software libero con licenza speciale

[35] **Shadow utilities** software libero con licenza speciale

[36] **Shadow utilities** software libero con licenza speciale

[37] **Debian adduser** GNU GPL

[38] **Shadow utilities** software libero con licenza speciale

[39] **Shadow utilities** software libero con licenza speciale

[40] **Linux-PAM** licenza in stile BSD che può trasformarsi in GNU GPL

[41] La gerarchia che parte dalla directory '`/usr/`' tipica, potrebbe essere contenuta in un disco diverso da quello che contiene quella principale; pertanto, se all'avvio ci sono delle difficoltà e non si può innestare la gerarchia '`/usr/`', si rischia di non poter usare il sistema perché mancano le librerie PAM.

[42] Nel capitolo non viene descritta la direttiva '**`@include`**', con cui si ottiene l'inclusione di un file in quel punto. Logicamente, se si vuole lasciare soltanto il file '`/etc/pam.d/other`' nella directory, se questo contiene direttive di inclusione di altri file, è necessario mantenere nella directory anche quelli.

[43] **System V Init** GNU GPL

[44] **GNU Accounting Utilities** GNU GPL

[45] **GNU Accounting Utilities** GNU GPL

[46] **GNU Accounting Utilities** GNU GPL

[47] **GNU Accounting Utilities** GNU GPL

[48] **GNU C Library** GNU GPL

[49] **Falselogin** GNU GPL

[50] Per evitare che lo script possa essere interrotto con un segnale generato attraverso una combinazione di tasti, come [*Ctrl c*], vanno intercettati alcuni segnali. Per farlo, con la shell standard, è sufficiente il comando '**`trap '' INT QUIT TSTP`**' all'inizio dello script stesso.

[51] Per evitare che lo script possa essere interrotto con un segnale generato attraverso una combinazione di tasti, come [*Ctrl c*], vanno intercettati alcuni segnali. Per farlo, con la shell standard, è sufficiente il comando '**`trap '' INT QUIT TSTP`**' all'inizio dello script stesso.