

[\[successivo\]](#) [\[precedente\]](#) [\[inizio\]](#) [\[fine\]](#) [\[indice generale\]](#) [\[violazione licenza\]](#) [\[translators\]](#) [\[docinfo\]](#) [\[indice analitico\]](#) [\[volume\]](#) [\[parte\]](#)

Capitolo 196. Trasferimento e sincronizzazione di dati attraverso la rete

A fianco del problema della realizzazione di una riproduzione speculare di informazioni pubblicate sulla rete, c'è anche quello di gestire un sistema di copia remota tra elaboratori, per dati che non sono messi a disposizione del pubblico, soprattutto allo scopo di mantenerli allineati.

Per questo tipo di problema, non avrebbe senso utilizzare il protocollo FTP, come sarebbe necessario per un sito speculare standard. Piuttosto, si fa uso di script o programmi che si basano sui servizi di una shell per l'accesso remoto, come 'rsh' o 'ssh' (capitoli [164](#) e [228](#)), oppure su protocolli appositi.

196.1 Rsync

Rsync ([1](#)) è un sistema di copia tra elaboratori (o anche all'interno del file system dello stesso sistema locale), in grado di individuare e trasferire il minimo indispensabile di dati, allo scopo di allineare la destinazione con l'origine. L'uso di questo programma è molto semplice ed è simile a quello di 'rcp' (*Remote shell copy*) o anche di 'scp' (*Secure shell copy*).

L'aggiornamento dei dati, in funzione delle opzioni utilizzate, può basarsi sul confronto delle date di modifica, delle dimensioni dei file e anche sul calcolo di un codice di controllo (*checksum*). In linea di principio, a meno di utilizzare opzioni che specificano qualcosa di diverso, non conta il fatto che i dati siano più recenti o meno, basta che questi siano diversi per ottenerne il trasferimento.

196.1.1 Tipi di utilizzo

Rsync può utilizzare diverse modalità di trasferimento dei file, a seconda delle circostanze e delle preferenze. Per la precisione si distinguono tre possibilità fondamentali.

- Copia locale.

In tal caso, la copia, o l'allineamento, avviene all'interno dello stesso sistema, dove l'origine e la destinazione sono riferite semplicemente a posizioni differenti nel file system. In questa circostanza, Rsync viene utilizzato come metodo evoluto di copia.

- Copia tra elaboratori attraverso 'rsh' o simili.

Si tratta di un'operazione che coinvolge due elaboratori differenti, anche se uno dei due deve essere necessariamente quello locale, in cui il trasferimento dei dati avviene attraverso 'rsh' o un suo equivalente (come 'ssh'), utilizzando una copia del programma 'rsync' anche nell'elaboratore remoto.

- Copia tra elaboratori attraverso un protocollo specifico di Rsync.

Si tratta di un sistema di copia tra elaboratori, dove in quello remoto si trova in funzione una copia del programma 'rsync', avviata in modo che resti in ascolto della porta TCP 873. In questo caso, la connessione tra elaboratore locale ed elaboratore remoto avviene direttamente senza l'utilizzo di una shell per l'accesso remoto.

196.1.2 Origine, destinazione e percorsi

La forma utilizzata per esprimere l'origine e la destinazione permette di distinguere anche la modalità con cui si vuole che la copia (l'allineamento) sia eseguita.

Sintassi	Descrizione
<i>percorso</i>	L'assenza del simbolo di due punti (' : '), indica che si tratta di un percorso riferito al file system locale.
[utente@]nodo:percorso	La presenza di un simbolo di due punti singolo (' : '), indica che si tratta di un percorso riferito a un nodo remoto e che per la connessione si vuole usare una shell per l'accesso remoto.
[utente@]nodo::percorso	La presenza di un simbolo di due punti doppio (' : : '), o l'indicazione esplicita

<code>[utente@]nodo . percorso</code>	del protocollo Rsync ('rsync://'), indica che si tratta di un percorso riferito a un nodo remoto e che per la connessione si vuole usare il protocollo specifico di Rsync.
<code>rsync://[utente@]nodo/percorso</code>	

L'indicazione dei percorsi merita attenzione. Per prima cosa si può dire che valgono regole simili a quelle della copia normale; per cui, si può copiare un file singolo, anche indicando espressamente il nome che si vuole nella destinazione (che potrebbe essere diverso da quello di origine); inoltre si possono copiare uno o più file e directory in una destinazione che sia una directory.

- Quando l'origine è locale, si possono indicare diversi percorsi, anche con l'aiuto di caratteri jolly che poi vengono interpretati opportunamente ed espansi dalla shell locale. L'esempio seguente, mostra il comando necessario a copiare o ad allineare i file che terminano per '.sgml', della directory corrente, con quanto contenuto nella directory '/tmp/prove/' del nodo roggen.brot.dg.

```
$ rsync *.sgml roggen.brot.dg:/tmp/prove[Invio]
```

- Quando l'origine è remota, si possono indicare diversi percorsi, anche con l'aiuto di caratteri jolly, che poi vengono interpretati opportunamente ed espansi dalla shell utilizzata nell'utenza remota. La differenza sta nel fatto che i caratteri jolly utilizzati non devono essere interpretati dalla shell locale, per cui è bene usare delle tecniche di protezione adatte. Probabilmente, ciò non è indispensabile, perché alcune shell come Bash ignorano l'espansione dei nomi se questi non possono avere una corrispondenza nel file system locale.

L'esempio seguente, mostra il comando necessario a copiare o ad allineare i file che terminano per '.sgml', della directory '/tmp/prove/' del nodo roggen.brot.dg, con quanto contenuto nella directory corrente dell'elaboratore locale.

```
$ rsync 'roggen.brot.dg:/tmp/prove/*.sgml' .[Invio]
```

- Quando l'origine fa riferimento a una directory, ma **non** si utilizza la barra obliqua finale, si intende individuare la directory, come se fosse un file normale. La directory di origine viene copiata nella directory di destinazione, aggiungendola a questa. Per cui, l'esempio seguente serve a copiare la directory locale '/tmp/ciao/' nella directory remota '/tmp/prove/', generando '/tmp/prove/ciao/' e copiando al suo interno i file e le sottodirectory che fossero eventualmente contenuti nel percorso di origine.

```
$ rsync -r /tmp/ciao roggen.brot.dg:/tmp/prove[Invio]
```

- Quando l'origine fa riferimento a una directory e si utilizza la barra finale, si intende individuare tutto il **contenuto** della directory, escludendo la directory stessa. Per cui, l'esempio seguente serve a copiare il contenuto della directory locale '/tmp/ciao/' nella directory remota '/tmp/prova/', generando eventuali file e sottodirectory contenuti nella directory di origine.

```
$ rsync -r /tmp/ciao/ roggen.brot.dg:/tmp/prove[Invio]
```

È diverso copiare il contenuto di una directory dal copiare una directory intera (assieme al suo contenuto); nel primo caso, si rischia di perdere la copia dei file «nascosti», ovvero quelli che iniziano con un punto.

196.1.3 Proprietà dei file

Come è possibile vedere in seguito, quando si utilizzano le opzioni '-o' ('--owner') e '-g' ('--group'), si intende fare in modo che nella destinazione sia mantenuta la stessa proprietà dei file (dell'utente o del gruppo) che questi hanno nell'origine.

Per ottenere questo risultato, si confrontano generalmente i nomi degli utenti e dei gruppi, assegnando i numeri UID e GID necessari. Quando questa corrispondenza dovesse mancare, viene utilizzato semplicemente lo stesso numero ID. In alternativa, con l'uso dell'opzione '--numeric-ids', si può richiedere espressamente l'uguaglianza numerica di UID o GID, indipendentemente dai nomi utilizzati effettivamente.

196.1.4 Avvio del programma

Il programma eseguibile 'rsync' è quello che svolge tutte le funzioni necessarie ad allineare una destinazione, in base al contenuto di un'origine. Per questo si può avvalere di 'rsh', di un'altra shell per l'accesso remoto o di un server Rsync remoto.

```
rsync [opzioni] origine destinazione
```

L'origine e la destinazione possono essere riferite indifferentemente al nodo locale o a un nodo remoto. Quello che conta è che almeno una delle due sia riferita al nodo locale.

Tabella 196.2. Alcune opzioni.

Opzione	Descrizione
-v --verbose	Permette di ottenere più informazioni sullo svolgimento delle operazioni. Questa opzione può essere usata più volte, incrementando il livello di dettaglio di tali notizie.
-q --quiet	Permette di ridurre la quantità di informazioni che vengono emesse. Di solito può essere utile quando si usa il programma attraverso Cron.
-z --compress	Prima di trasmettere i dati, li comprime. Ciò permette di risparmiare risorse di rete durante il trasferimento dei dati.
-I --ignore-times	Normalmente, si considera che i file che hanno la stessa dimensione e la stessa data di modifica, siano identici. Con questa opzione, si fa in modo che tale presunzione non sia valida.
-c --checksum	Fa in modo che vengano confrontati tutti i file attraverso un codice di controllo prima di decidere se devono essere trasferiti o meno. L'uso di questa opzione implica un tempo di elaborazione più lungo, anche se garantisce una sicurezza maggiore nella determinazione delle differenze esistenti tra l'origine e la destinazione.
-a --archive	Questa opzione rappresenta in pratica l'equivalente di '-r-lptgoD' (ovvero la sequenza delle opzioni '--recursive', '--links', '--perms', '--time', '--group', '--owner', '--devices'), allo scopo di duplicare fedelmente tutte le caratteristiche originali, discendendo ricorsivamente le directory di origine.
-r --recursive	Richiede la copia ricorsiva delle directory, cioè di tutte le sottodirectory.
-R --relative	Fa in modo di replicare nella destinazione, aggiungendolo a questa, il percorso indicato nell'origine, che comunque deve essere relativo.
-b --backup	Fa in modo di salvare temporaneamente i file che verrebbero sovrascritti da un aggiornamento. Questi vengono rinominati, aggiungendo un'estensione che generalmente è rappresentata dalla tilde ('~'). Questa estensione può essere modificata attraverso l'opzione '--suffix'.
--suffix= <i>suffisso</i>	Permette di definire il suffisso da usare per le copie di sicurezza dei file che vengono sovrascritti.
-u --update	Con questa opzione, si evita l'aggiornamento di file che nella destinazione risultano avere una data di modifica più recente di quella dei file di origine corrispondenti.
-l --links	Fa in modo che i collegamenti simbolici vengano ricreati fedelmente, come nell'origine.
-L --copy-links	Fa in modo che i collegamenti simbolici nell'origine, si traducano nella destinazione nei file a cui questi puntano.

-H --hard-links	Richiede la riproduzione fedele dei collegamenti fisici. Perché ciò possa avvenire, occorre che questi collegamenti si riferiscano allo stesso gruppo di file di origine che viene indicato nella riga di comando.
-W --whole-file	Rsync utilizza normalmente un metodo che gli permette di trasferire solo il necessario per aggiornare ogni file. Con questa opzione, si richiede espressamente che ogni file da aggiornare sia inviato per intero. Questo può essere utile quando si allineano dati contenuti nella stessa macchina e qualunque elaborazione aggiuntiva servirebbe solo a rallentare l'operazione.
-p --perms	Riproduce fedelmente i permessi.
-o --owner	Quando Rsync viene utilizzato con i privilegi dell'utente 'root', permette di assegnare a ciò che viene copiato lo stesso utente proprietario che risulta nell'origine.
-g --group	Quando Rsync viene utilizzato con i privilegi dell'utente 'root', permette di assegnare a ciò che viene copiato lo stesso gruppo proprietario che risulta nell'origine.
--numeric-ids	Fa in modo di mantenere gli stessi numeri ID, quando le altre opzioni richiedono la riproduzione della proprietà dell'utente ('-o') o del gruppo ('-g').
-D --devices	Quando Rsync viene utilizzato con i privilegi dell'utente 'root', permette di copiare file di dispositivo.
-t --times	Fa in modo che venga riprodotta fedelmente la data di modifica dei file.
--partial	Durante il trasferimento dei file, nella destinazione Rsync scarica i dati in un file «nascosto» (in quanto inizia con un punto). Quando un trasferimento viene interrotto, l'ultimo file il cui trasferimento non è stato completato, viene cancellato. Con questa opzione, si fa in modo di non perdere i trasferimenti parziali, recuperandoli la volta successiva.
--progress	Fa in modo di mostrare l'avanzamento del trasferimento dei singoli file, in modo da poter conoscere la situazione anche in presenza di file di grandi dimensioni.
-P	È l'equivalente della somma di '--partial' e di '--progress'.
-n --dry-run	Si limita a simulare l'operazione, senza eseguire alcuna copia. È utile per verificare l'effetto di un comando prima di eseguirlo veramente.
-x --one-file-system	Permette di non superare il file system di partenza, nell'origine.
--delete	Fa sì che vengano cancellati i file nella destinazione che non si trovano nell'origine. Come si può intuire, si tratta di un'opzione molto delicata, in quanto un piccolo errore nell'indicazione dei percorsi si può tradurre nella perdita involontaria di dati. È questa la situazione più indicata per utilizzare l'opzione '-n' in modo da verificare in anticipo l'effetto del comando. In alcune circostanze può essere utile anche l'opzione '--force'.

	Se Rsync incontra dei problemi di lettura, la funzione di cancellazione viene inibita, salvo mantenerla attiva con l'opzione ' <code>--ignore-errors</code> '.
<code>--delete-after</code>	Con questa opzione, assieme a ' <code>--delete</code> ', si fa in modo che la cancellazione dei file che non sono più nell'origine, avvenga alla fine delle altre operazioni; diversamente, ciò avviene all'inizio.
<code>--force</code>	Con questa opzione si consente la cancellazione di directory che non sono vuote quando devono essere rimpiazzate da file normali o comunque da file che non sono directory. Perché questa opzione venga presa in considerazione è necessario usare anche l'opzione ' <code>-r</code> ' (' <code>--recursive</code> ').
<code>--ignore-errors</code>	Con questa opzione, assieme a ' <code>--delete</code> ', si conferma la richiesta di cancellazione anche in presenza di errori di lettura e scrittura dei dati.
<code>--exclude=modello</code>	Permette di indicare un nome di file (o directory), o un modello contenente caratteri jolly, riferito a nomi da escludere dalla copia. Il nome o il modello indicato, non deve contenere riferimenti a percorsi; inoltre è bene che sia protetto in modo che non venga espanso dalla shell usata per avviare il comando. È il caso di sottolineare che, se viene escluso il nome di una directory si impedisce un eventuale attraversamento ricorsivo del suo contenuto.
<code>--exclude-from=file</code>	Si comporta come l'opzione ' <code>--exclude</code> ', con la differenza che il suo argomento è il nome di un file locale contenente un elenco di esclusioni.
<code>--password-file=file</code>	Quando è richiesta una forma di autenticazione fornendo una parola d'ordine, si può usare questa opzione per indicare il nome di un file di testo che la contenga. In alternativa, si può inserire questa informazione nella variabile di ambiente ' <code>RSYNC_PASSWORD</code> '.
<code>--password-file=file</code>	Quando è richiesta una forma di autenticazione fornendo una parola d'ordine, si può usare questa opzione per indicare il nome di un file di testo che la contenga; il file non deve consentire l'accesso a utenti diversi dal proprietario. In alternativa, si può inserire questa informazione nella variabile di ambiente ' <code>RSYNC_PASSWORD</code> '.
<code>-e=comando</code> <code>--rsh=comando</code>	Permette di specificare il comando (il programma) da utilizzare come shell per l'accesso remoto. Normalmente viene usata ' <code>rsh</code> ', ma in alternativa si potrebbe utilizzare ' <code>ssh</code> ', o altro se disponibile. L'uso di una shell alternativa per l'accesso remoto, può essere configurato utilizzando la variabile di ambiente ' <code>RSYNC_RSH</code> '.
<code>--rsync-path=percorso</code>	Permette di specificare il percorso assoluto necessario ad avviare ' <code>rsync</code> ' nell'elaboratore remoto. Ciò è utile quando il programma non è nel percorso degli eseguibili nell'utenza remota.
<code>-C</code> <code>--cvs-exclude</code>	Questa opzione permette di escludere una serie di file, usati tipicamente da CVS, RCS e anche in altre situazioni, che generalmente non conviene trasferire. Si tratta dei nomi e dei modelli seguenti: ' <code>RCS</code> ', ' <code>SCCS</code> ', ' <code>CVS</code> ', ' <code>CVS.adm</code> ', ' <code>RCSLOG</code> ', ' <code>cvslog.*</code> ', ' <code>tags</code> ', ' <code>TAGS</code> ', ' <code>.make.state</code> ', ' <code>.nse_depinfo</code> ', ' <code>*~</code> ', ' <code>.#*</code> ', ' <code>.*</code> ', ' <code>*.old</code> ', ' <code>*.bak</code> ', ' <code>*.BAK</code> ', ' <code>*.orig</code> ', ' <code>*.rej</code> ', ' <code>.del-*</code> ', ' <code>*.a</code> ', ' <code>*.o</code> ', ' <code>*.obj</code> ', ' <code>*.so</code> ', ' <code>*.Z</code> ', ' <code>*.elc</code> ', ' <code>*.ln</code> ', ' <code>core</code> '. Inoltre, vengono esclusi anche i file elencati all'interno di ' <code>~/cvsignore</code> ', della variabile di ambiente ' <code>CVSIGNORE</code> ' e all'interno di ogni file ' <code>.cvsignore</code> ', ma in questo ultimo caso, solo in riferimento al contenuto della directory in cui si trovano.

Segue la descrizione di alcuni esempi.

- `$ rsync -r /tmp/prove roggen.brot.dg:/tmp/prove`[Invio]

Copia la directory '`/tmp/prove/`' del nodo locale, assieme a tutto il suo contenuto, nel nodo `roggen.brot.dg`, generando lì, la directory '`/tmp/prove/prove/`' contenente tutto ciò che discende dall'origine.

Si osservi che questa copia non riproduce le informazioni data-orario dei file e delle directory (opzione '`-t`'), pertanto, se dovesse essere ripetuto il comando, si otterrebbe nuovamente il trasferimento di tutti i file.

- `# rsync -a /tmp/prove roggen.brot.dg:/tmp/prove`[Invio]

Copia la directory '`/tmp/prove/`' del nodo locale, assieme a tutto il suo contenuto, nel nodo `roggen.brot.dg`, generando lì, la directory '`/tmp/prove/prove/`' contenente tutto ciò che discende

dall'origine. La copia viene fatta riproducendo il più possibile le caratteristiche originali, comprese informazioni data-orario dei file e delle directory, così che un utilizzo successivo dello stesso comando trasferirebbe solo quanto necessario ad aggiornare la copia.

- **# rsync -a /tmp/prove/ roggen.brot.dg:/tmp/prove**[Invio]

Copia il contenuto della directory '/tmp/prove/' del nodo locale nel nodo roggen.brot.dg, nella directory '/tmp/prove/'. La copia viene fatta riproducendo il più possibile le caratteristiche originali e la ripetizione del comando in momenti successivi trasferisce solo il necessario.

- **\$ rsync -R prove/mie/*.txt roggen.brot.dg:/home/tizio**[Invio]

Copia i file che terminano per '.txt' della directory 'prove/mie/', discendente da quella attuale, nella directory '/home/tizio/prove/mie/' del nodo dinkel.brot.dg.

Si osservi che questa copia non riproduce le informazioni data-orario dei file e delle directory (opzione '-t'), pertanto, se dovesse essere ripetuto il comando, si otterrebbe nuovamente il trasferimento di tutti i file.

- **# rsync -a -z -v /tmp/prove/ roggen.brot.dg:/tmp/prove**[Invio]

Copia il contenuto della directory '/tmp/prove/' del nodo locale nel nodo roggen.brot.dg, nella directory '/tmp/prove/'. La copia viene fatta riproducendo il più possibile le caratteristiche originali, trasferendo dati compressi e visualizzando le operazioni compiute.

- **# rsync -azv -e ssh /tmp/prove/ roggen.brot.dg:/tmp/prove**[Invio]

Come nell'esempio precedente, ma utilizza 'ssh' come shell per l'accesso remoto.

- **# rsync -rlptD -zv /tmp/prove/ tizio@roggen.brot.dg:/tmp/prove**[Invio]

Come nell'esempio precedente, ma utilizza la shell predefinita per l'accesso remoto e accede come utente 'tizio'. Per questo, non tenta di riprodurre la proprietà dei file (utente e gruppo proprietario).

- **# rsync -rlptD -zv --progress **
\tmp/prove/ tizio@roggen.brot.dg:/tmp/prove[Invio]

Come nell'esempio precedente, aggiungendo informazioni sul trasferimento dei singoli file.

- **# rsync -rlptD -zv --progress **
\tmp/prove/ tizio@roggen.brot.dg::prove[Invio]

Questo esempio è simile a quello precedente, con la differenza che nella destinazione si fa riferimento al modulo 'prove'. I moduli di Rsync vengono descritti nelle sezioni seguenti, in occasione della presentazione delle funzionalità di server di Rsync.

- **# rsync -rlptD -zv --progress **
\tmp/prove/ rsync://tizio@roggen.brot.dg/prove[Invio]

Esattamente come nell'esempio precedente, usando una notazione differente per la destinazione.

- **# rsync -rlptD -zv --progress **
\tmp/prove/varie/ rsync://tizio@roggen.brot.dg/prove/varie[Invio]

Come nell'esempio precedente, con la differenza che si intende allineare solo una sottodirectory, precisamente '/tmp/prove/varie/', con la sottodirectory corrispondente nel modulo 'prove'.

- **\$ rsync --recursive**
\ --compress
**\ --links **
\ --perms

```

\ --times\
\ --partial\
\ --checksum\
\ --verbose\
\ --progress\
\ rsync://roggen.brot.dg/prove/varie/ /home/prove/varie[Invio]

```

In questo caso si vuole aggiornare il contenuto della directory locale '/home/prove/varie/' con il contenuto della directory 'varie/' del modulo 'prove' presso l'elaboratore roggen.brot.dg che offre un accesso Rsync anonimo.

Come si può osservare dalle opzioni, si fa in modo di avere informazioni abbastanza dettagliate sullo svolgimento dell'operazione, per la presenza di '--verbose' e di '--progress'; inoltre, viene richiesto espressamente di verificare sempre i file da trasferire con un codice di controllo (opzione '--checksum') e di conservare i trasferimenti parziali (in modo da ridurre il lavoro di un aggiornamento successivo, in caso di interruzione della comunicazione).

Si osservi che la presenza dell'opzione '--checksum' richiede un impiego maggiore di risorse da parte di entrambi gli elaboratori coinvolti nel trasferimento, che si traduce in tempi di attesa più lunghi.

- \$ **rsync rsync://roggen.brot.dg**[Invio]

Con questo comando ci si limita a interrogare il server Rsync remoto sulla sua disponibilità di moduli. Si osservi però che alcuni o anche tutti i moduli possono risultare nascosti, cioè non visibili in questo elenco, in base alla configurazione del server stesso.

196.1.5 Accesso attraverso autenticazione

Quando è richiesta l'autenticazione attraverso una parola d'ordine l'uso della variabile di ambiente 'RSYNC_PASSWORD' può essere molto utile per automatizzare le operazioni di sincronizzazione dati attraverso Rsync.

Quello che si vede sotto, potrebbe essere uno script personale di un utente che deve aggiornare frequentemente il modulo 'prove' nel nodo roggen.brot.dg (identificandosi come 'tizio'). Quando il server remoto richiede la parola d'ordine, il cliente locale 'rsync' la legge direttamente dalla variabile 'RSYNC_PASSWORD':

```

#!/bin/sh
RSYNC_PASSWORD=1234ciao
export RSYNC_PASSWORD
rsync -rlptD -zv /tmp/prove/ rsync://tizio@roggen.brot.dg/prove

```

In alternativa alla variabile di ambiente 'RSYNC_PASSWORD', si può usare un file esterno, con permessi di accesso limitati, specificando l'opzione '--password-file', come nell'esempio seguente:

```

#!/bin/sh
touch ~/.rsync-password
chmod go-rwx ~/.rsync-password
echo "1234ciao" > ~/.rsync-password
rsync -rlptD -zv --password-file=~/.rsync-password \
/tmp/prove/ rsync://tizio@roggen.brot.dg/prove
rm -f ~/.rsync-password

```

Naturalmente, se Rsync non ottiene la parola d'ordine in uno di questi modi, la chiede in modo interattivo all'utente.

196.1.6 Server Rsync

Se si vuole utilizzare Rsync per trasferire dati tra elaboratori differenti, senza usare una shell remota, occorre attivare nell'elaboratore remoto un server Rsync. Si tratta in pratica dello stesso programma 'rsync', ma avviato con l'opzione '--daemon'.

Il serverente Rsync può essere avviato in modo indipendente, in ascolto da solo sulla porta TCP 873, oppure sotto il controllo del supervisore dei servizi di rete. In questa modalità di funzionamento è necessario predisporre un file di configurazione: '/etc/rsyncd.conf'.

Nel caso si voglia avviare il serverente Rsync in modo autonomo dal supervisore dei servizi di rete, basta un comando come quello seguente:

```
# rsync --daemon[Invio]
```

Se si vuole inserire Rsync nel controllo del supervisore dei servizi di rete (cosa di sicuro consigliabile), occorre intervenire nel file '/etc/services' per definire il nome del servizio:

```
rsync      873/tcp
```

Inoltre occorre agire nel file '/etc/inetd.conf' (nel caso specifico di Inetd) per annunciarlo al supervisore dei servizi di rete:

```
rsync  stream tcp  nowait  root  /usr/bin/rsync  rsyncd --daemon
```

Durante il funzionamento del serverente Rsync si può avvertire un carico eccessivo del sistema operativo, causato dalla procedura di scansione dei file per il confronto con la copia remota da sincronizzare. Quando questo fatto diventa un problema, si può cercare di ovviare all'inconveniente avviando il programma 'rsync' controllandone la priorità di esecuzione. Per fare questo conviene realizzare uno script, simile a quello seguente:

```
#!/bin/sh
nice -n 19 /usr/bin/rsync $@
```

Supponendo che questo script corrisponda al file '/etc/script/rsync-slow', si può modificare così la riga che riguarda tale servizio nel file '/etc/inetd.conf':

```
rsync  stream tcp  nowait  root  /etc/script/rsync-slow  rsyncd --daemon
```

Naturalmente, questo script va bene anche se si intende gestire il serverente Rsync al di fuori del controllo del supervisore dei servizi di rete.

Rsync utilizzato come serverente si avvale del file di configurazione '/etc/rsyncd.conf' per definire una o più directory che si vogliono rendere accessibili attraverso il protocollo di Rsync, come una sorta di servizio FTP. Come nel caso dell'FTP, è possibile offrire l'accesso a chiunque, in modo anonimo, oppure si può distinguere tra utenti definiti all'interno della gestione di Rsync. Questi utenti sono potenzialmente estranei all'amministrazione del sistema operativo in cui Rsync si trova a funzionare, per cui occorre aggiungere un file di utenti e parole d'ordine specifico.

Rsync definisce **moduli** le aree che mette a disposizione (in lettura o anche in scrittura a seconda della configurazione). Quando si vuole accedere a un modulo di Rsync si utilizza una delle due notazioni seguenti:

```
[utente_rsync@]nodo::modulo[/percorso_successivo]
```

```
rsync://[utente_rsync@]nodo/modulo[/percorso_successivo]
```

Quando si accede a un modulo, il serverente Rsync può eseguire un 'chroot()' nella directory a cui questo fa riferimento, più o meno come accade con l'FTP anonimo. Per fare un esempio concreto, se il modulo 'prova' fa riferimento alla directory '/home/dati/ciao/' nel nodo dinkel.brot.dg, l'indirizzo 'dinkel.brot.dg::prova/uno/mio', oppure 'rsync://dinkel.brot.dg/prova/uno/mio', fa riferimento al percorso '/home/dati/ciao/uno/mio' in quell'elaboratore.

196.1.7 Configurazione di Rsync come servente

Il file di configurazione di Rsync, `/etc/rsyncd.conf`, serve solo nel caso lo si voglia utilizzare come servente. Se si intende fare affidamento sul servizio di `rsh` o di `ssh`, non c'è alcun bisogno di preoccuparsene.

Il contenuto del file di configurazione è organizzato in moduli, ognuno dei quali descrive le impostazioni riferite a una directory del file system sottostante.

Ogni riga descrive un tipo di informazione. Le righe vuote, quelle bianche e ciò che è preceduto dal simbolo `#` viene ignorato. È ammessa la continuazione nella riga successiva utilizzando la barra obliqua inversa (`\`) alla fine della riga.

I moduli vengono identificati da un nome racchiuso tra parentesi quadre e la loro indicazione occupa tutta una riga; le informazioni riferite a un modulo sono costituite da tutte le direttive che appaiono nelle righe seguenti, fino all'indicazione di un altro modulo. Le direttive che descrivono i moduli sono delle opzioni che definiscono dei parametri e sono in pratica degli assegnamenti di valori a questi parametri. Alcuni tipi di parametri possono essere collocati prima di qualunque dichiarazione di modulo e si tratta in questo caso di opzioni globali che riguardano tutti i moduli (alcuni parametri possono apparire solo all'inizio e non all'interno della dichiarazione dei moduli).

Le opzioni globali sono quelle direttive (o parametri) che si collocano prima della dichiarazione dei moduli. Alcuni parametri possono essere collocati solo in questa posizione, mentre gli altri, le opzioni dei moduli, pur essendo stati preparati per la descrizione dei singoli moduli, possono essere usati all'inizio per definire un'impostazione generale. L'elenco seguente mostra solo l'uso di alcuni parametri delle opzioni globali.

Tabella 196.3. Alcune direttive globali.

Direttiva	Descrizione
<code>motd file = file</code>	Se presente, indica un file all'interno del quale viene prelevato il testo da mostrare agli utenti quando si connettono (il «messaggio del giorno»).
<code>max connections = n_massimo_conessioni_simultanee</code>	Come avviene nel protocollo FTP, anche con Rsync può essere importante porre un limite alle connessioni simultanee. Se non viene specificata questa opzione, oppure se si usa il valore zero, non si intende porre alcuna restrizione.
<code>log file = file</code>	In generale, i messaggi generati da Rsync durante il funzionamento come demone, sono diretti al registro di sistema, ma con l'uso di questa direttiva si può generare un file autonomo.
<code>pid file = file</code>	Questa direttiva fa sì che Rsync scriva il numero del proprio processo elaborativo (PID) nel file indicato.

Tabella 196.4. Alcune direttive dei moduli.

Opzione	Descrizione
<code>comment = stringa_di_descrizione_del_modulo</code>	Questa opzione permette di fornire una descrizione che può essere letta dagli utenti che accedono. Il suo scopo è chiarire il contenuto o il senso di un modulo il cui nome potrebbe non essere sufficiente per questo. Non è necessario racchiudere tra apici doppi il testo della stringa.
<code>list = yes no</code> <code>list = true false</code>	Con questa direttiva si controlla la possibilità di mostrare l'esistenza del modulo quando viene interrogato l'elenco di quelli esistenti. In condizioni normali, questa funzionalità è attiva e si può impedire la lettura assegnando il valore <code>'no'</code> o <code>'false'</code> .
<code>path = percorso_della_directory</code>	Questo parametro è obbligatorio per ogni modulo. La direttiva serve a definire la directory, nel file system dell'elaboratore presso cui è in funzione il servente Rsync, a cui il modulo fa riferimento. Normalmente, attraverso la direttiva <code>'use chroot'</code> si fa in modo che, quando si accede al modulo, Rsync esegua la funzione <code>'chroot()'</code>

	in modo che la directory corrispondente appaia come la radice del modulo stesso.
<pre>use chroot = yes no use chroot = true false</pre>	Questo parametro è molto importante e consente, se si attribuisce un valore affermativo, di accedere alla directory del modulo attraverso la funzione 'chroot()'. Tuttavia, questa funzionalità può essere attivata solo se Rsync viene avviato con i privilegi dell'utente 'root'.
<pre>read only = true false read only = yes no</pre>	Questa opzione permette di definire se il modulo debba essere accessibile solo in lettura oppure anche in scrittura. Se l'opzione non viene specificata, si intende che l'accesso debba essere consentito in sola lettura. Assegnando il valore booleano 'false' (oppure 'no') si ottiene di consentire anche la scrittura.
<pre>uid = nome_utente id_utente gid = nome_gruppo id_gruppo</pre>	Queste due opzioni permettono di definire l'utente e il gruppo per conto dei quali devono essere svolte le operazioni all'interno del modulo. In pratica, Rsync utilizza quella identità per leggere o scrivere all'interno del modulo; questo può essere un mezzo attraverso il quale controllare gli accessi all'interno della directory corrispondente.
<pre>auth users = utente_rsync[, utente_rsync]...</pre>	Questa opzione permette di indicare un elenco di nomi di utenti di Rsync a cui è consentito di accedere al modulo. Senza questa opzione, si concede l'accesso a chiunque, mentre in tal modo si impone il riconoscimento in base a un file di utenti definito attraverso il parametro 'secrets file'.
<pre>secrets file = file_di_utenti_e_parole_d'ordine</pre>	Questa opzione è obbligatoria se viene usato il parametro 'auth users'. Serve a indicare il file all'interno del quale Rsync può trovare l'elenco degli utenti e delle parole d'ordine (in chiaro).
<pre>strict modes = true false strict modes = yes no</pre>	Questa opzione permette di stabilire se il file indicato con la direttiva 'secrets file' debba essere accessibile esclusivamente all'utente associato al processo elaborativo di Rsync (di solito corrisponde a 'root'), oppure se può mancare questa accortezza. In generale, questa opzione è attiva, a indicare che il file deve essere protetto.
<pre>ignore nonreadable = true false strict modes = yes no</pre>	Questa opzione permette di accettare la presenza di file che non risultano leggibili al server Rsync. In pratica, con questa opzione attiva, si fa in modo che i file non leggibili siano trattati come se non esistessero del tutto.
<pre>transfer logging = true false strict modes = yes no</pre>	Questa opzione, che normalmente non risulta attiva, se viene abilitata consente di far annotare nel registro i file trasferiti.
<pre>timeout = n_secondi</pre>	Questa opzione consente di specificare una scadenza alle connessioni, indicando un numero che esprime una quantità di secondi. Normalmente non c'è alcuna scadenza, ma in questo modo un errore da parte di un programma cliente potrebbe lasciare aperta una connessione inesistente all'infinito. In generale, se non ci sono altri problemi, conviene lasciare un tempo ragionevolmente grande, di una o più ore.
<pre>max connections = n</pre>	Questa opzione consente di limitare la quantità massima di connessioni simultanee. In mancanza di questa direttiva, nessun limite viene posto.

Segue la descrizione di alcuni esempi.

•

```
uid = nobody
gid = nobody
[ftp]
    path = /home/ftp
    comment = Esportazione dell'area FTP attraverso Rsync
```

Questo esempio, preso da *rsyncd.conf(5)*, rappresenta una configurazione minima allo scopo di definire il modulo 'ftp' che consenta l'accesso in sola lettura alla directory '/home/ftp' per qualunque utente. In pratica, si vuole permettere l'accesso all'area FTP anche attraverso Rsync. Si osservi in particolare l'uso dei parametri 'uid' e 'gid', all'inizio del file, in modo che Rsync utilizzi i privilegi dell'utente e del gruppo 'nobody' per la lettura dei file.

```
[ftp]
    path = /home/ftp
    comment = Esportazione dell'area FTP attraverso Rsync
    uid = nobody
    gid = nobody
```

Si tratta di una variante dell'esempio precedente, in cui i parametri 'uid' e 'gid' sono stati collocati all'interno del modulo. In questo caso, dal momento che non ci sono altri moduli, l'effetto è lo stesso.

```
[pippo]
    comment = Applicativo PIPPO
    path = /opt/pippo
    read only = false
    uid = tizio
    gid = tizio
    auth users = caio, sempronio
    secrets file = /etc/rsyncd.secrets
```

L'esempio mostra la descrizione del modulo 'pippo' all'interno di un file di configurazione che potrebbe contenerne anche altri. In pratica, gli utenti che Rsync identifica come 'caio' e 'sempronio', possono scrivere all'interno della directory '/opt/pippo/', generando eventualmente anche delle sottodirectory, utilizzando i privilegi dell'utente e del gruppo 'tizio' (secondo quanto definito dal sistema operativo di quell'elaboratore). Il file delle parole d'ordine necessario a identificare gli utenti 'caio' e 'sempronio' è '/etc/rsyncd.secrets'.

```
pid file=/var/run/rsyncd.pid
use chroot = yes
read only = yes
list = yes
uid = rsync
gid = rsync
secrets file = /etc/rsyncd.secrets
strict modes = yes
ignore nonreadable = yes
transfer logging = yes
timeout = 10800

[a2dist]
    comment = a2 distribution
    max connections = 7
    path = /home/a2dist/distribution
    auth users = tizio, caio, sempronio
```

Questo è un esempio abbastanza completo. Nella parte iniziale, le direttive globali servono a: specificare il file da usare per annotare il numero del processo elaborativo (PID); richiedere che venga utilizzata la funzione

'chroot()' all'inizio di ogni modulo; consentire un accesso in sola lettura; consentire la visualizzazione dell'elenco dei moduli disponibili; a far funzionare il programma servente con i privilegi dell'utente e del gruppo 'rsync' (ma all'avvio il programma deve avere i privilegi dell'utente 'root' e con questi privilegi va poi a leggere il file contenenti le parole d'ordine); specificare quale sia il file contenente le parole d'ordine e che questo non deve risultare accessibile ad altri utenti; ignorare i file che non risultano leggibili, come se non ci fossero; annotare il trasferimento di tutti i file nel registro; infine l'ultima direttiva pone una scadenza alle connessioni sospese di tre ore al massimo.

Dopo le direttive globali appare un solo modulo, denominato 'a2dist', nel quale si indica: una descrizione del modulo; il limite massimo di connessioni (sette); il percorso del modulo (la directory '/home/a2dist/distribution/'); gli utenti autorizzati ad accedere al modulo.

196.1.8 File degli utenti e delle parole d'ordine secondo Rsync

Quando si utilizza Rsync come servente e si richiede una forma di autenticazione agli utenti che accedono, è necessario predisporre un file di testo contenente dei record secondo la sintassi seguente:

```
nome_utente:parola_d'ordine_in_chiaro
```

Dal momento che normalmente il file viene letto da Rsync con i privilegi dell'utente 'root', è sufficiente che questo file abbia il permesso di lettura per l'amministratore del sistema.

Rsync non stabilisce quale sia la collocazione e il nome di questo file; è il parametro 'secrets file' del file di configurazione a definirlo volta per volta. In generale, nella documentazione originale si fa l'esempio del file '/etc/rsyncd.secrets'. L'esempio seguente mostra il caso degli utenti 'caio' e 'sempronio', a cui sono state abbinate rispettivamente le parole d'ordine 'tazza' e 'ciao'.

```
caio:tazza
sempronio:ciao
```

È bene ribadire che questo file non ha alcun nesso con il file '/etc/passwd' (né con '/etc/shadow'). Gli utenti di Rsync possono non essere stati registrati (nel modo consueto) nell'elaboratore presso cui accedono.

196.1.9 Tempi morti e scadenze

Rsync è un sistema molto sofisticato per la sincronizzazione dei dati, che consente anche l'esecuzione del lavoro a più riprese, persino su file singoli (opzione '--partial'), con il minimo traffico di rete possibile.

Questa parsimonia nella gestione delle risorse di rete ha però un effetto indesiderato, in quanto si possono creare dei tempi morti, anche lunghi, in cui la connessione TCP rimane aperta senza il passaggio di alcun pacchetto. Tale situazione si può verificare in modo particolare quando si trasmettono file di grandi dimensioni attraverso dei tentativi successivi, perché ogni volta i due elaboratori coinvolti devono ricalcolare i codici di controllo di questi, per stabilire se la porzione presente nella destinazione possa essere utilizzata o meno: durante questo calcolo il traffico della connessione rallenta fino a sospendersi.

Anche se la sospensione della comunicazione non dovrebbe portare conseguenze per la connessione, bisogna ricordare questo fatto quando si utilizza la direttiva 'timeout' (o l'opzione '--timeout'), in modo da lasciare un tempo sufficiente allo svolgimento delle operazioni necessarie. Inoltre, anche senza imporre alcun limite, ci potrebbero essere dei componenti tra i due elaboratori che non sono al corrente dell'esigenza di avere delle pause molto lunghe nelle connessioni. Potrebbe trattarsi di un router-NAT, che deve seguire tutte le comunicazioni che richiedono la trasformazione degli indirizzi e delle porte, introducendo anche per questo un problema di «scadenza» delle connessioni, che così si può manifestare con delle interruzioni inspiegabili della sincronizzazione dei dati attraverso Rsync.

Quando l'uso appropriato della direttiva 'timeout' o dell'opzione '--timeout' non porta a risolvere il problema, può essere necessario evitare l'uso dell'opzione '--partial'.

196.2 Rdist

Rdist ⁽²⁾ è un sistema di copia che permette di mantenere l'allineamento di uno o più elaboratori (*host*), mantenendo le informazioni relative alla proprietà, ai permessi e alla data di modifica dei file coinvolti.

L'aggiornamento dei dati si basa sul confronto delle date di modifica e delle dimensioni dei file. In linea di principio, non conta il fatto che i dati siano più recenti o meno, basta che siano diversi. Naturalmente, è possibile istruire Rdist in modo che aggiorni solo i file più recenti, così come si possono definire altri dettagli.

L'operazione di allineamento delle copie parte dall'elaboratore che contiene i dati originali, contattando i vari nodi presso cui si trovano le copie da allineare. In questo senso va inteso il nome di Rdist: *Remote distribution*, ovvero, distribuzione remota.

196.2.1 Principio di funzionamento

Rdist si avvale di due eseguibili per stabilire il collegamento necessario al trasferimento dei dati da allineare: 'rdist' dalla parte dell'elaboratore utilizzato come punto di partenza per la distribuzione dei file (l'origine) e 'rdistd' dalla parte degli elaboratori contenenti le copie da allineare (le destinazioni).

Tuttavia questo non basta. È necessario anche l'uso di 'rsh'; inoltre l'accesso remoto relativo deve essere configurato in modo che l'utente, generalmente 'root', possa accedere agli elaboratori da allineare senza l'inserimento di alcuna parola d'ordine.

In pratica, si utilizza 'rdist' per ordinare l'allineamento dei dati; questo utilizza 'rsh' per connettersi con uno degli elaboratori remoti coinvolti, allo scopo di avviare lì il programma 'rdistd'. Quindi, attraverso la connessione tra 'rdist' e 'rdistd', ottenuta per mezzo di 'rsh', avviene la verifica delle corrispondenze e il trasferimento dei dati necessari.

Dalla descrizione fatta, dovrebbe essere chiaro che Rdist non è un servizio di rete, nel senso che non esiste un demone in ascolto su una certa porta TCP/IP. Rdist si avvale del servizio reso da 'rsh', che da solo (probabilmente anche con 'rcp') non basterebbe a risolvere il problema dell'allineamento delle copie remote.

196.2.2 Origine e destinazione

Generalmente, quando si indicano i dati da allineare, si fa riferimento a un'origine, rappresentata da un percorso del file system locale, e a una destinazione composta semplicemente dal nome del nodo. In questa situazione, si intende che il percorso indicato come origine sia lo stesso nel file system del nodo di destinazione.

Per esempio, se si vuole allineare la directory '/tmp/prove/ciao/' del file system locale, con il nodo remoto linux.brot.dg, senza specificare una directory remota, si intende che debba trattarsi della stessa anche in quel file system.

Se invece si specifica anche la directory remota, la destinazione diventa esplicita, così che questa può essere anche una posizione diversa da quella del nodo di origine.

196.2.3 Modalità di distribuzione

Prima ancora di vedere come si utilizza e si configura Rdist, è utile analizzare alcune delle modalità di funzionamento. La loro conoscenza permette di comprendere le possibilità di Rdist e il senso di ciò che si fa.

Come è possibile vedere meglio più avanti, la modalità di funzionamento viene definita attraverso una o più parole chiave, fornite per mezzo dell'opzione '-o'. Segue l'elenco di alcune di queste parole chiave.

Modalità	Descrizione
'verify'	Non effettua alcun allineamento di dati, si verifica solo che i dati siano allineati. Se non lo sono, si ottiene l'elenco di ciò che andrebbe aggiornato.
'whole'	Se si specifica una directory remota, la modalità 'whole' fa sì che nella directory di destinazione venga riprodotto tutto il percorso originale. Per esempio, utilizzando la modalità 'whole', se si vuole allineare la directory '/tmp/prove/ciao/' del file system locale, con il nodo remoto linux.brot.dg nella directory '/tmp/', si ottiene la directory '/tmp/tmp/prove/ciao/'.
'younger'	Fa in modo che vengano aggiornati solo i dati che nell'origine risultano avere una data di modifica più recente.
'compare'	Fa in modo che vengano aggiornati solo i file che differiscono nel contenuto. Per determinarlo, i file vengono comparati in modo binario; se risultano diversi, avviene la modifica nelle copie da allineare.
'ignlnks'	Fa in modo che venga ignorato il problema dei collegamenti simbolici che non sono risolti. In pratica, con questa modalità i collegamenti vengono riprodotti tali e quali, anche se puntano a qualcosa che non c'è.
'chknfs'	Ignora la verifica e l'allineamento dei dati che nel nodo di destinazione risultano collocati in un file system di rete (NFS).
'chkreadonly'	Ignora la verifica e l'allineamento dei dati che nel nodo di destinazione risultano collocati in un file system montato in sola lettura.
'chksym'	Fa in modo di tollerare i collegamenti simbolici nella destinazione. In pratica, se nella destinazione i file o le directory sono spostati rispetto all'origine, ma per mantenere uniformità appaiono dei collegamenti, questi non vengono rimossi e si accetta la variazione locale.
'quiet'	Riduce la quantità di messaggi. Per la precisione non segnala i file che vengono aggiornati.
'remove'	Elimina i file estranei. Se viene aggiornata una directory intera, i file che si trovano già nella destinazione, ma non sono presenti nell'origine, vengono rimossi.
'nochkowner'	Non verifica l'utente proprietario per i file che esistono già nella destinazione. La proprietà dell'utente viene modificata solo se il file richiede un aggiornamento per altri motivi.
'nochkgroup'	Non verifica il gruppo proprietario per i file che esistono già nella destinazione. La proprietà del gruppo viene modificata solo se il file richiede un aggiornamento per altri motivi.
'nochkmode'	Non verifica i permessi per i file che esistono già nella destinazione. I permessi vengono modificati solo se il file richiede un aggiornamento per altri motivi.
'nodescend'	Fa in modo che le directory vengano trattate come file normali, senza allineare anche il loro contenuto e le sottodirectory eventuali.
'numchkowner'	L'allineamento delle copie è fatto in modo che i file nella destinazione abbiano lo stesso numero UID di quelli dell'origine. Quando non si utilizza questa modalità, Rdist fa in modo che i file abbiano lo stesso nominativo-utente, utilizzando il numero UID necessario in base alla configurazione del nodo di destinazione.
'numchkgroup'	L'allineamento delle copie è fatto in modo che i file nella destinazione abbiano lo stesso numero GID di quelli dell'origine. Quando non si utilizza questa modalità, Rdist fa in modo che i file abbiano lo stesso nominativo di gruppo, utilizzando il numero GID necessario in base alla configurazione del nodo di destinazione.
'savetargets'	Nel caso in cui dei file nella destinazione debbano essere sostituiti, questi vengono salvati con l'estensione '.OLD'. Se esistono già copie di sicurezza del genere, queste vengono sovrascritte senza essere rinominate ulteriormente.

196.2.4 Configurazione

Le operazioni da compiere con Rdist possono essere inserite in un file di configurazione. Se attraverso le opzioni non si fa riferimento a qualcosa di diverso, o comunque non si vuole ignorare questo file, il nome predefinito è 'distfile', oppure, in sua mancanza, 'Distfile', collocato nella directory corrente.

La struttura di questo file di configurazione è un po' strana. Come accade spesso, il simbolo '#' introduce un commento che termina alla fine della riga; inoltre, le righe vuote sono ignorate.

All'interno del file è possibile dichiarare delle variabili, attraverso le quali, il resto delle direttive può diventare più semplice da leggere. La loro dichiarazione avviene in direttive che utilizzano la sintassi seguente:

```
[ ]
```

```
nome_variabile = valore
```

La loro espansione avviene con la notazione seguente, dove le parentesi graffe (che in questo caso vanno intese letteralmente) sono necessarie per fare in modo che il nome della variabile venga preso in considerazione per intero (diversamente si utilizzerebbe solo il primo carattere).

```
${nome_variabile}
```

Nelle direttive che definiscono l'allineamento tra origine e destinazione, si fa spesso riferimento a elenchi di nomi. Questi possono essere indicati raggruppandoli attraverso l'uso di parentesi tonde, come mostrato nella sintassi seguente:

```
nome | ( [nome [nome]...] )
```

Come si vede, quando l'elenco è formato da un nome soltanto, non occorrono parentesi, anche se queste si possono usare comunque. L'elenco tra parentesi è spaziato semplicemente, senza bisogno di altri simboli di separazione; inoltre è possibile indicare l'elenco nullo.

Questi raggruppamenti possono essere assegnati a delle variabili che successivamente possono essere usate per rappresentarli. In questo senso, si possono eseguire delle operazioni elementari che si richiamano vagamente alla teoria degli insiemi.

```
elenco + elenco
```

```
elenco - elenco
```

```
elenco & elenco
```

La prima espressione restituisce un elenco che contiene tutti gli elementi dei due elenchi; in pratica, rappresenta l'unione dei due. La seconda restituisce un elenco contenente gli elementi presenti nel primo insieme, che non si trovano nel secondo. La terza restituisce l'intersezione dei due insiemi, cioè un elenco di elementi presenti in entrambi i raggruppamenti.

Gli elenchi di file possono essere definiti attraverso caratteri jolly, ma solo quando questi sono riferiti all'elaboratore locale (quello di origine). Sono ammissibili i caratteri jolly della shell C ('csh'); in pratica sono validi: l'asterisco ('*'), il punto interrogativo ('?'), le parentesi quadre e le parentesi graffe, con lo stesso significato che hanno anche con la shell Bash.

Il tipo di direttiva più importante è quello che definisce l'allineamento di un gruppo di file o directory nell'origine con un gruppo di nodi di destinazione. Anche se la sintassi seguente mostra una struttura scomposta su più righe, in realtà tutto potrebbe apparire su una riga sola, a discapito della leggibilità.

```
[etichetta:]
    origine -> destinazione
    [comando;]...
```

L'etichetta è un nome facoltativo, terminato da due punti (':'), a cui si può fare riferimento per selezionare un gruppo ristretto di azioni; l'origine è un elenco di file e directory; la destinazione è un elenco di nodi rappresentati per nome o attraverso l'indirizzo IP, con l'eventuale aggiunta di un prefisso costituito dal nominativo-utente da utilizzare per l'accesso remoto; infine, i comandi sono delle indicazioni aggiuntive che definiscono in particolare l'operazione da compiere e permettono di stabilire delle modalità dell'allineamento dei dati.

La destinazione può essere composta da un elenco di nomi che rispettano la sintassi seguente. L'utente, rappresenta il nome utilizzato per l'accesso attraverso 'rsh'.

```
[utente@]nodo
```

```

[...]
```

I comandi possono essere di tipo differente e così utilizzano sintassi differenti. Segue la descrizione di questi comandi.

Comando	Descrizione
<code>install [-oelenco_modalità] \</code> <code>\[destinazione]</code>	Copia dei file e delle directory che, in base alle modalità specificate (o predefinite), richiedono aggiornamento. Le modalità possono essere indicate come appare nella sintassi: precedute da '-o', proseguendo con un elenco dei nomi di modalità che si vogliono attivare. Questo elenco è staccato semplicemente utilizzando la virgola, senza spazi aggiuntivi. Se viene indicato un percorso finale, si intende specificare esplicitamente una destinazione nel file system dell'elaboratore a cui sono diretti i dati. Generalmente si tratta di una directory, a meno che l'origine sia composta semplicemente da un solo file.
<code>except elenco_da_escludere</code>	Esclude un gruppo di file e directory dalle operazioni che altrimenti avrebbero luogo in base agli altri comandi. In pratica permette di escludere la «installazione» di alcuni file e directory riferiti all'elaboratore di origine. È ammesso l'uso di caratteri jolly.
<code>except_pat \</code> <code>\elenco_modelli_regexp_da_escludere</code>	Esclude un gruppo di file e directory dalle operazioni che altrimenti avrebbero luogo in base agli altri comandi, indicandoli attraverso espressioni regolari.
<code>special [elenco] "comando_sh"</code>	Permette di eseguire un comando nell'elaboratore remoto, dopo l'aggiornamento di ogni file indicato nell'elenco. In pratica, il comando viene eseguito solo se il file è stato aggiornato. Se non viene indicato alcun file nell'elenco, il comando viene eseguito per ogni file aggiornato. Il comando viene eseguito nell'elaboratore remoto utilizzando la shell 'sh' e può anche essere composto da più comandi, separati con il punto e virgola. Questo comando eredita alcune variabili di ambiente: 'FILE', contenente il percorso assoluto, nell'origine, del file che è stato aggiornato; 'REMFIL', contenente il percorso assoluto, nella destinazione, del file che è stato aggiornato; 'BASEFILE', contenente il nome, senza percorso, del file che è stato aggiornato; si riferisce al nome nell'origine.
<code>cmdspecial elenco "comando_sh"</code>	Permette di eseguire un comando nell'elaboratore remoto, dopo l'aggiornamento di tutti i file. L'elenco fornito come primo argomento viene trasmesso attraverso la variabile di ambiente 'FILES', nella quale, i vari elementi appaiono separati da due punti (':').
<code>notify elenco_email</code>	Invia un messaggio contenente le operazioni compiute attraverso la posta elettronica. L'indirizzo può essere espresso con un nome, senza l'indicazione del nodo; in tal caso si riferisce a quello di destinazione.

Di seguito vengono mostrati e descritti alcuni esempi riferiti alla configurazione di Rdist attraverso il file 'distfile' oppure 'Distfile'.

```
GRUPPO_HOST = ( roggen.brot.dg root@linux.brot.dg )
```

Dichiara la variabile 'GRUPPO_HOST' a cui viene attribuito l'elenco di nodi composto da roggen.brot.dg e da linux.brot.dg specificando anche il nominativo-utente 'root', da utilizzare per accedere presso questo ultimo nodo.

```
GRUPPO_ORIGINE = ( /usr /opt )
```

Dichiara la variabile 'GRUPPO_ORIGINE' a cui viene attribuito l'elenco di file e directory da duplicare nella destinazione remota. Vengono specificate le directory '/usr/' e '/opt/'.

```
GRUPPO_ESCLUSO = ( /usr/src/linux* /opt/marameo )
```

Dichiara la variabile 'GRUPPO_ESCLUSO' a cui viene attribuito l'elenco di file e directory da escludere dalla duplicare nella destinazione remota. Vengono specificati tutti i file e le directory che corrispondono al modello '/usr/src/linux*' e la directory (o il file) '/opt/marameo'.

```

${GRUPPO_ORIGINE} -> ${GRUPPO_HOST}
    install -oremove,ignlnks ;
    except ${GRUPPO_ESCLUSO} ;
    except /usr/share/games ;
    special /opt/pippo/etc/configura "/opt/pippo/bin/rigenera" ;

```

Dichiara l'allineamento tra un gruppo di file e directory dell'elaboratore di origine, espresso dalla variabile 'GRUPPO_ORIGINE', con un gruppo di nodi di destinazione, espresso dalla variabile 'GRUPPO_HOST', utilizzando gli stessi percorsi nelle varie destinazioni, attivando le modalità 'remove' e 'ignlnks'. In particolare vengono esclusi dall'allineamento i file e le directory rappresentati dalla variabile 'GRUPPO_ESCLUSO' e anche quanto contenuto nella directory '/usr/share/games/' (si presume che sia una directory, ma questo non è indicato esplicitamente). Infine, quando viene aggiornato il file '/opt/pippo/etc/configura', viene avviato il programma '/opt/pippo/bin/rigenera' nella destinazione (probabilmente serve a ricostruire altri file in funzione del contenuto del file modificato).

```

kernel:
/usr/src/linux* -> root@linux.brot.dg
    install /usr/local/src ;
    notify ( tizio daniele@dinkel.brot.dg ) ;

```

Dichiara l'allineamento tra i file e le directory che corrispondono al modello '/usr/src/linux*' con il nodo linux.brot.dg (utente 'root'), nella directory '/usr/local/src/'. Al termine dell'allineamento, viene inviato un messaggio a tizio@linux.brot.dg e a daniele@dinkel.brot.dg.

196.2.5 Avvio del programma

L'eseguibile 'rdist' è quello attraverso il quale si ottiene l'allineamento tra un elaboratore di origine e uno o più elaboratori di destinazione. 'rdist' si avvale normalmente di un file di configurazione, indicato espressamente attraverso l'opzione '-f', oppure rappresentato dal file './distfile', o da './Distfile'. Se si vuole selezionare una o più etichette all'interno del file di configurazione, si possono indicare i nomi di queste alla fine della riga di comando.

```
rdist [opzioni] [etichetta...]
```

```
rdist [opzioni] -c origine... [utente_remoto@]nodo_destinazione[:directory_destinazione]
```

In alternativa all'uso del file di configurazione, si può indicare l'operazione di allineamento nella riga di comando, con l'opzione '-c'. In tal caso, sarebbe come se fosse stato usato il file di configurazione seguente:

```
( origine... ) -> [utente_remoto@]nodo_destinazione
    install [directory_destinazione] ;
```

Il programma 'rdist' si avvale di 'rsh' per avviare nell'elaboratore remoto il suo «collega», 'rdistd', nel modo seguente:

```
rsh -l utente_remoto rdistd -S
```

Tabella 196.7. Alcune opzioni.

Opzione	Descrizione
<code>-f file_di_configurazione</code>	Permette di definire il file di configurazione da utilizzare. Se al posto del nome viene indicato un trattino ('-'), questo viene atteso dallo standard input.
<code>-D</code>	Abilita l'emissione di messaggi diagnostici estremamente dettagliati.
<code>-m nodo</code>	Questa opzione può apparire più volte nella riga di comando e serve a specificare uno o più elaboratori da allineare, in modo da limitarne il numero, rispetto a quanto previsto nel file di configurazione.
<code>-n</code>	Simula l'allineamento, limitandosi a visualizzare ciò che farebbe (potrebbe non essere attendibile).
<code>-oelenco_modalità</code>	Permette di indicare una o più modalità di funzionamento. Le parole chiave delle modalità devono iniziare subito dopo la lettera «o» dell'opzione, senza spazi, e l'elenco di queste è separato esclusivamente con la virgola, senza inserire altri spazi.
<code>-p percorso_di_rdistd_remoto</code>	Permette di indicare il percorso assoluto di 'rdistd' da avviare nell'elaboratore remoto. Ciò può essere necessario se l'utente utilizzato per accedere attraverso 'rsh' non ha 'rdistd' in uno dei percorsi di avvio dei comandi (la variabile 'PATH').
<code>-P percorso_di_rsh_locale</code>	Permette di indicare il percorso assoluto necessario ad avviare 'rsh' nell'elaboratore locale. Ciò potrebbe servire anche per avviare un programma alternativo a 'rsh', purché accetti le stesse opzioni fondamentali e si comporti nello stesso modo (si può provare con 'ssh', ma non è detto che funzioni).

Segue la descrizione di alcuni esempi.

- **# rdist -f ~/distfile**[Invio]
Avvia 'rdist' per eseguire le direttive contenute nel file '~/distfile'.
- **# rdist -f ~/distfile prova**[Invio]
Avvia 'rdist', utilizzando il file di configurazione '~/distfile', specificando di voler eseguire esclusivamente l'etichetta 'prova'.
- **# rdist -f ~/distfile -p /usr/sbin/rdistd**[Invio]
Avvia 'rdist' per eseguire le direttive contenute nel file '~/distfile', indicando precisamente il percorso assoluto del programma 'rdistd' negli elaboratori remoti.
- **# rdist -n -f ~/distfile -p /usr/sbin/rdistd**[Invio]
Come nell'esempio precedente, ma limitandosi a simulare le operazioni.
- **# rdist -c /usr/lib root@roggen.brot.dg**[Invio]
Aggiorna il nodo roggen.brot.dg in modo che la directory '/usr/lib/' contenga le stesse cose di quello locale.
- **# rdist -c /usr/lib root@roggen.brot.dg:/usr/local/lib**[Invio]
Aggiorna il nodo roggen.brot.dg in modo che la directory remota '/usr/local/lib/' contenga le stesse cose della directory locale '/usr/lib/'.

2) **Rdist** software non libero: non è consentita la riproduzione fisica della documentazione e la vendita di assistenza; inoltre la modifica non è consentita esplicitamente

Dovrebbe essere possibile fare riferimento a questa pagina anche con il nome
[trasferimento e sincronizzazione di dati attraverso la rete.html](#)

[\[successivo\]](#) [\[precedente\]](#) [\[inizio\]](#) [\[fine\]](#) [\[indice generale\]](#) [\[violazione licenza\]](#) [\[translators\]](#) [\[docinfo\]](#) [\[indice analitico\]](#)

