

## Stampare

27.1	Introduzione alla stampa	1098
27.1.1	Dispositivi e stampa brutale	1098
27.1.2	Visione generale e astratta	1099
27.1.3	Problemi collegati con la stampa remota	1100
27.1.4	Filtri di stampa	1100
27.1.5	Stampanti virtuali multiple	1101
27.1.6	Quando il meccanismo dei filtri non funziona bene	1101
27.2	Sistema di stampa BSD o compatibile	1101
27.2.1	Configurazione con «/etc/printcap»	1102
27.2.2	Servizio di stampa	1105
27.2.3	Stampante predefinita	1106
27.2.4	Clienti di stampa	1106
27.2.5	Esame delle code di stampa	1107
27.2.6	Rimozione dei processi di stampa dalle code	1108
27.2.7	Controllo del sistema di stampa	1108
27.2.8	Particolarità del sistema BSD vero e proprio	1109
27.2.9	Particolarità di LPRng	1110
27.2.10	Stampare attraverso X con un sistema di stampa BSD	1111
27.2.11	Librerie GTK 2	1112
27.2.12	Sviluppo successivo	1112
27.3	Stampa remota attraverso Rlpr	1113
27.4	Stampa automatica da un file	1113
27.5	File e filtri per la stampa	1114
27.5.1	File per la stampa	1114
27.5.2	Filtri di stampa	1115
27.5.3	Magicfilter	1118
27.5.4	PostScript printer description	1122
27.5.5	Foomatic-RIP	1124
27.5.6	Controllo dell'impostazioni della carta	1124
27.6	Uniformità del sistema di stampa: da testo a PostScript	1125
27.6.1	A2ps	1125
27.6.2	Enscript	1129
27.7	Filtri particolari con LPRng	1134
27.7.1	Schema di funzionamento	1134
27.7.2	Script «lpr» e «lp»	1135
27.7.3	Funzioni usate da «client-side-print-filter»	1136
27.7.4	Lo script «client-side-print-filter»	1139
27.8	Introduzione a Cups	1140
27.8.1	Visione generale	1141
27.8.2	Stampanti logiche, classi e istanze	1141
27.8.3	Configurazione di una stampante locale	1141
27.8.4	Configurazione di una stampante remota	1143
27.9	Cups: programmi di servizio e configurazione	1144
27.9.1	Individuazione automatica delle stampanti	1144
27.9.2	Amministrazione	1144
27.9.3	Configurazione principale del servizio di stampa	1146
27.9.4	Configurazione dal lato dei clienti	1148
27.9.5	Utilizzo comune attraverso la riga di comando	1149
27.9.6	Personalizzazione dal lato cliente	1151
27.9.7	Accesso remoto	1152
27.9.8	Compatibilità con il sistema di stampa BSD	1155
27.9.9	Considerazioni finali	1155

27.10 Riferimenti ..... 1155

.cupsrc 1148 /etc/magicfilter/ 1121 checkpc 1110  
 client.conf 1148 cupsaccept 1144 cupsd.conf 1146  
 1152 cupsdisable 1144 cupsenable 1144 cupsreject  
 1144 gtkrc 1112 hosts.equiv 1109 hosts.lpd 1109 lp  
 1106 lp0 1098 lpadmin 1144 lpc 1108 lpd 1101 1105  
 lpd.conf 1110 lpd.perms 1110 lpinfo 1144 lpq 1107  
 lpr 1101 1106 lprm 1108 passwd.md5 1154 printcap 1102  
 \$PRINTER 1106

27.1 Introduzione alla stampa

Tradizionalmente, il dispositivo di stampa permette solo la scrittura, cioè si comporta come un file al quale si possono solo aggiungere dati. In questa situazione, la stampa si ottiene semplicemente trasferendo (copiando) un file alla stampante. Naturalmente, il file deve essere stato predisposto in modo da poter essere interpretato correttamente dalla stampante che si utilizza.

Quando si ha la necessità di applicare una trasformazione al file da stampare, prima che questo raggiunga la stampante, si utilizza normalmente un filtro di stampa, cioè un programma o uno script che può essere inserito in un condotto. I filtri di stampa vengono quindi utilizzati sia per adattare i file da stampare alle caratteristiche particolari della stampante che si ha a disposizione, sia per ottenere degli effetti, come l'aggiunta di intestazioni.

Esistono nel mercato delle stampanti che non si accontentano di ricevere un file per iniziare a stampare, ma richiedono l'utilizzo di un protocollo di comunicazione, che spesso è mantenuto segreto. Queste stampanti, per funzionare, hanno bisogno della presenza di un programma speciale, predisposto dalla casa produttrice; pertanto non sono compatibili in alcun modo con i sistemi GNU. Si tratta in particolare delle stampanti che utilizzano il cosiddetto *Windows printing system*. Si deve fare attenzione quindi, prima di acquistare una stampante da usare con un sistema GNU.

Questa parte del documento, dedicata alla stampa, fa riferimento a concetti che possono essere chiariti solo più avanti, come la stampa remota e l'utilizzo di strumenti grafici. Sotto questo aspetto, l'argomento dovrebbe essere trattato più tardi; tuttavia, dal momento che l'esigenza di stampare si avverte molto presto, l'argomento viene anticipato. Pertanto, chi ha l'esigenza di realizzare un server di stampa in grado di ricevere richieste da una rete, se non è già informato su queste cose, deve attendere e leggere una serie di capitoli sul TCP/IP a partire da 32.

Per poter utilizzare la stampante con un sistema GNU/Linux, occorre avere compilato il kernel inserendo la gestione delle porte di comunicazione necessarie e della stampa stessa.

27.1.1 Dispositivi e stampa brutale

I file di dispositivo associati alla stampa, in un sistema GNU/Linux, dipendono dal tipo di porta di comunicazione utilizzato. A titolo di esempio, la prima stampante collegata su porta parallela dovrebbe essere accessibile attraverso il file di dispositivo `/dev/lp0`, mentre alla prima stampante collegata a un bus USB si associa di solito il file `/dev/usb/lp0`.

Eventualmente, si può controllare con il comando seguente:

```
$ dmesg | less [Invio]
```

Dal risultato che si ottiene, si deve cercare una riga simile a quella seguente:

```
lp0: using parport0 (polling)
```

L'utente `'root'` può utilizzare direttamente il dispositivo di stampa copiando su di esso il file che vuole stampare:

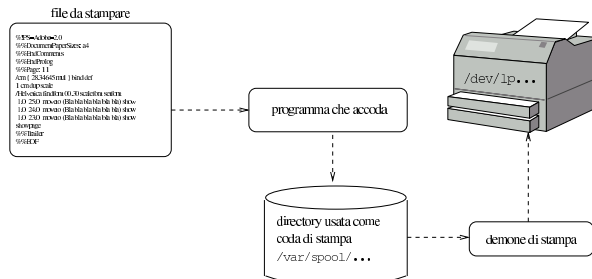
```
# cp stampa.prn /dev/lp0 [Invio]
```

Si tratta comunque di un modo di utilizzo della stampante decisamente sconsigliabile o quantomeno da riservare a circostanze particolari.

27.1.2 Visione generale e astratta

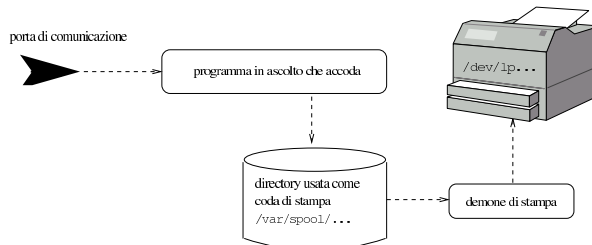
Il sistema di stampa tipico si avvale di una coda, ovvero di un deposito in cui accodare i file da inviare alla stampante. Di solito, il programma che si occupa di inserire il file da stampare nella coda non fa altro: per inviare questi file alla stampante c'è un demone apposito che attende di vedere qualcosa nella coda (figura 27.2).

Figura 27.2. Coda di stampa in un sistema locale chiuso.



Quando il sistema di stampa gestisce anche le stampe remote, si introduce un protocollo di comunicazione e, assieme a questo, anche qualche problema in più. Dal lato dell'elaboratore che offre il servizio ci deve essere un demone in grado di ricevere queste richieste di stampa, con il compito di accodare tali stampe nel proprio sistema (figura 27.3).

Figura 27.3. Coda di stampa in un sistema che riceve richieste dall'esterno, attraverso la rete.



Per inviare una richiesta di stampa a un elaboratore remoto, ci possono essere due tipi di approcci. Nella situazione più semplice, un programma potrebbe provvedere da solo a ricevere il file da stampare e a instaurare la connessione con il nodo di rete remoto al quale questo deve essere rinviato per la stampa; in alternativa, potrebbe essere la stessa coda di stampa locale che si occupa di rinviare la stampa a un nodo di rete remoto (figure 27.4 e 27.5).

Figura 27.4. Trasmissione delle stampe a un nodo di rete remoto, senza la mediazione della coda di stampa locale.

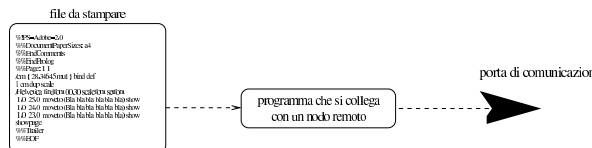
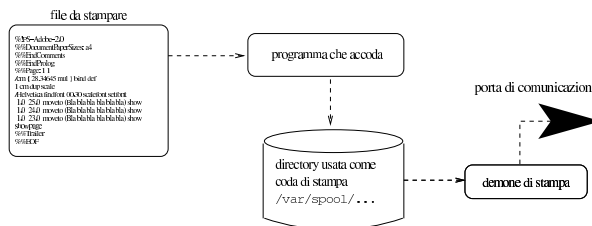


Figura 27.5. Trasmissione delle stampe a un nodo di rete remoto, ma attraverso la coda di stampa locale.



Spesso, molte funzionalità sono raggruppate assieme in uno stesso programma, o in uno stesso demone. Per esempio, è normale che il demone che si occupa di provvedere alla stampa di ciò che trova nella coda, sia anche in grado di ricevere una richiesta di stampa dall'esterno, provvedendo da solo ad accodarla, ed è normale che lo stesso demone sia in grado di instaurare una connessione con un altro servizio di stampa remoto quando deve demandare la stampa a quel sistema.

### 27.1.3 Problemi collegati con la stampa remota

La stampa remota introduce tanti piccoli problemi e spesso si deve pensare un po' prima di arrivare al risultato. Per prima cosa è necessario che ci sia accordo tra il programma che invia una richiesta di stampa e quello che deve riceverla, cosa questa che riguarda la coerenza con i protocolli relativi. Tuttavia, il protocollo standard che esiste attualmente è insufficiente per le esigenze reali (RFC 1179) e ogni sistema di stampa introduce le sue estensioni più o meno incompatibili con gli altri.

Quando un sistema offre un servizio attraverso la rete, come nel caso di un server di stampa, si pone il problema di non accettare tutte le richieste incondizionatamente e di stabilire chi sia abilitato ad accedere. In pratica, occorre autenticare gli accessi. Questo problema non è previsto dal protocollo citato e il sistema di stampa che vuole essere compatibile con tutto, può solo limitarsi a selezionare gli accessi in base alla loro origine.

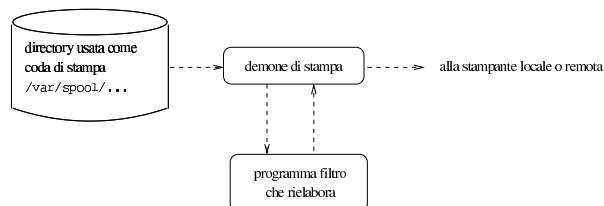
L'ultimo problema da considerare è legato al fatto che con la stampa remota si fanno transitare le informazioni relative attraverso la rete, rischiando l'intercettazione di informazioni che potrebbero essere delicate. Un sistema di stampa evoluto potrebbe prevedere la cifratura di queste comunicazioni, introducendo una propria estensione al protocollo standard.

### 27.1.4 Filtri di stampa

Nel momento in cui si considera che per stampare si prepara un file e lo si invia alla stampante, per gestire stampanti di tipo diverso in modo trasparente, basta realizzare dei programmi filtro appositi con lo scopo di rielaborare i dati nel modo più opportuno prima di passarli effettivamente alla stampante.

Questi programmi filtro potrebbero essere inseriti in diversi punti della catena di un sistema di stampa; in particolare si potrebbe scegliere se questa elaborazione deve avvenire prima dell'inserimento nella coda di stampa, o se questo debba avvenire dopo. Di solito, i file vengono messi nella coda così come sono, ed è il demone di stampa che si occupa di farli rielaborare da un programma filtro adatto (figura 27.6).

Figura 27.6. Introduzione di un filtro in un sistema di stampa tipico.



Tuttavia, occorre considerare che si possono fare delle acrobazie imprevedibili per un principiante, per cui la distinzione non diventa più tanto netta. Infatti, capita a volte che il programma filtro prenda i dati e non restituisca nulla, o meglio, invece di restituire qualcosa nel modo previsto, potrebbe farci qualcosa di diverso (si vede la sezione 27.1.6 e la figura 27.7).

### 27.1.5 Stampanti virtuali multiple

Un sistema di stampa come descritto astrattamente in queste sezioni, potrebbe essere in grado di gestire code differenti, senza che questo implichi la disponibilità effettiva di più stampanti collegate allo stesso elaboratore. Dal punto di vista del sistema di stampa, queste code sono delle stampanti «virtuali» collegate in qualche modo a delle stampanti reali.

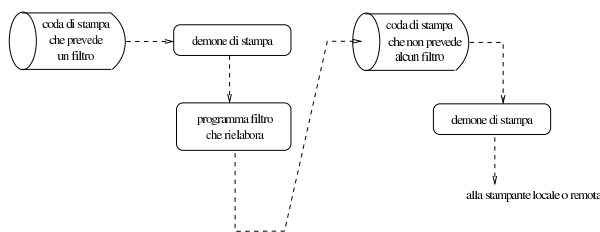
Per fare un esempio pratico, un sistema di stampa potrebbe essere stato configurato in modo da avere due code di stampa: per una stampante locale per una stampante remota. In questo modo, quando si richiede di stampare utilizzando una coda, si ottiene alla fine la stampa attraverso la propria stampante locale, mentre utilizzando l'altra, si ottiene l'invio di una richiesta di stampa a un sistema remoto.

Le possibilità non si limitano a questo; per esempio le code potrebbero essere state distinte perché a ognuna di queste viene attribuito un filtro di stampa diverso, di solito per permettere di utilizzare una stampante differente da quella solita. Per esempio, si potrebbe avere la coda denominata 'lp' per la stampa diretta senza filtri; la coda 'lp-1j' da utilizzare quando si collega una stampante HP Laserjet o compatibile; la coda 'lp-ps' da utilizzare quando si collega una stampante PostScript.

### 27.1.6 Quando il meccanismo dei filtri non funziona bene

In precedenza è stato mostrato lo schema di un sistema di stampa che permette l'inserimento di un filtro prima di arrivare alla stampante. Si è accennato anche al fatto che il demone che legge la coda, mandando i dati al filtro, potrebbe essere difettoso e non essere in grado di rileggere ciò che restituisce il filtro stesso. In questi casi, che sono capitati effettivamente, si può attuare un rimedio, apparentemente un po' strano: il programma filtro, invece di restituire il risultato della sua elaborazione attraverso lo standard output, lo invia in un'altra coda di stampa, per la quale non è previsto alcun filtro (figura 27.7).

Figura 27.7. Quando il meccanismo del filtro è zoppicante.

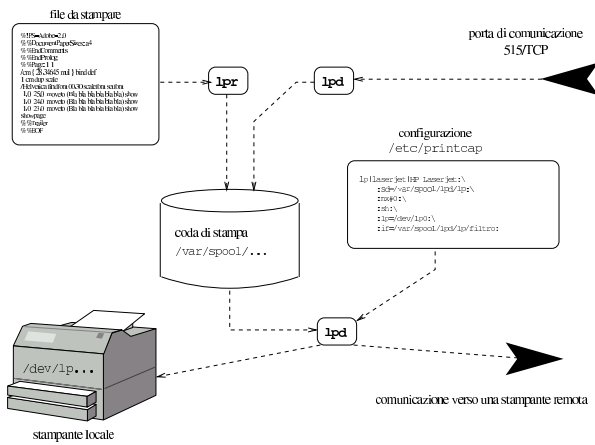


Riprendendo l'esempio già descritto in precedenza, in cui la coda denominata 'lp' è prevista per la stampa diretta senza filtri e la 'lp-1j' è fatta per stampare attraverso una stampante HP Laserjet o compatibile, il filtro abbinato a questa seconda coda, potrebbe semplicemente inviare il risultato della sua elaborazione nella coda di stampa normale, 'lp', che non avendo filtri non ha alcun problema.

## 27.2 Sistema di stampa BSD o compatibile

Nei sistemi Unix non esiste un sistema di stampa «standard» per vari motivi, a cominciare dal fatto che i pochi riferimenti disponibili hanno caratteristiche insufficienti rispetto alle esigenze attuali. Bene o male, i comandi per la stampa tendono a imitare il comportamento del sistema di stampa BSD, ovvero il lavoro di Berkeley.<sup>1</sup> Alcune distribuzioni GNU/Linux utilizzano proprio il sistema BSD, altre preferiscono qualcosa di più potente che gli assomiglia vagamente. In generale, non è il caso di approfondire questo o quel sistema di stampa, proprio perché si tratta di una materia in evoluzione, a meno che ci siano delle esigenze particolari, nel qual caso si possono studiare le pagine di manuale o la documentazione che accompagna il sistema di stampa che offre la propria distribuzione GNU.

Figura 27.8. Coda di stampa in stile BSD.



Come si può vedere nella figura 27.8, il sistema di stampa in stile BSD si avvale del programma `lpr` per accodare le stampe e del demone `lpd` per gestire la stampa di ciò che è stato accodato, oltre che per ricevere le richieste attraverso la rete. Fa parte della tradizione anche il file di configurazione `/etc/printcap`, nel quale vengono definite le varie code di stampa, a cui si possono abbinare o meno dei filtri opportuni.

### 27.2.1 Configurazione con `/etc/printcap`

La configurazione di un sistema di stampa in stile BSD avviene principalmente attraverso il file `/etc/printcap`, con il quale si definiscono le code di stampa e il loro comportamento. Il suo contenuto è organizzato in record, dove ognuno di questi contiene le informazioni relative a una coda. I campi di questi record sono separati da due punti verticali (alle volte doppi e altre singoli) e possono essere spezzati su più righe, utilizzando la barra obliqua inversa (`\`) seguita immediatamente dal codice di interruzione di riga. Si osservi il fatto che l'ultimo campo è concluso da due punti.

```
campo_1 : campo_2 : ... : campo_n :
```

```
campo_1 : \
: campo_2 : \
: campo_3 : \
...
: campo_n_1 : \
: campo_n :
```

Il sistema di stampa BSD originale richiede il simbolo di continuazione delle righe nel caso i record vengano spezzati, mentre altri sistemi compatibili, potrebbero farne a meno. In generale, è sempre meglio indicare la continuazione, anche se ciò non dovesse servire.

All'interno di questo file si possono trovare le indicazioni di code differenti che si riferiscono a un'unica stampante reale, per esempio quando si utilizzano configurazioni multiple per la stessa unità fisica. Si osservi che il simbolo `#` rappresenta l'inizio di un commento, con il quale, il testo che segue fino alla fine della riga non viene tenuto in considerazione; nello stesso modo vengono ignorate le righe vuote e quelle bianche.

Viene mostrato subito un esempio, il cui contenuto viene poi descritto gradualmente in questa sezione:

```
# Stampante predefinita
lp|laserjet|HP Laserjet:\
:sd=/var/spool/lpd/lp:\
:mx#0:\
:sh:\
```

```
:lp=/dev/lp0:\
:if=/var/spool/lpd/lp/filtro:

# Stampa di testo
ascii:\
:sd=/var/spool/lpd/tx:\
:mx#0:\
:sh:\
:lp=/dev/lp0:\
:if=/var/spool/lpd/ascii/filtro:

# Stampa diretta senza filtri
bare:\
:sd=/var/spool/lpd/bare:\
:mx#0:\
:sh:\
:lp=/dev/lp0:

# Stampante condivisa dell'elaboratore weizen.mehl.dg
net:\
:sd=/var/spool/lpd/net:\
:mx#0:\
:sh:\
:rm=weizen.mehl.dg:\
:rp=lp:\
:if=/var/spool/lpd/net/filtro:
```

Il primo campo di ogni record identifica tutti gli pseudonimi possibili di una certa coda di stampa, solitamente per identificare anche la stampante a cui la coda è abbinata. Questi sono separati da una barra verticale. Gli altri campi contengono tutti una sigla identificativa composta da due caratteri, seguita eventualmente da un valore che gli viene attribuito.

```
nome_coda [ |nome_stampante ] ... { sigla_campo [ assegnamento ] : } ...
```

La presenza di queste sigle permette in pratica di avere un numero variabile di campi, con un ordine variabile, dove solo il primo ha un ruolo prestabilito: quello di identificare la coda.

Nel preparare questo file è molto importante fare bene attenzione a non lasciare spazi di qualunque tipo dopo i simboli di continuazione (`\`), altrimenti questi simboli verrebbero intesi solo come una sequenza di escape che conferma il valore letterale del carattere che segue e non ci sarebbe alcuna continuazione. Questa considerazione è importante, perché poi è difficile scoprire errori del genere.

Il sistema di stampa BSD tradizionale prevede una quantità molto grande di campi nei record di `/etc/printcap`. Le esigenze attuali sono tali per cui i campi che si utilizzano in pratica sono molto pochi e non vale la pena di approfondire tecniche ormai superate, riferite a campi che in alcuni sistemi derivati potrebbero anche non essere più disponibili. La tabella 27.10 riepiloga i campi più comuni.

Tabella 27.10. I campi più importanti dei record che compongono il file `/etc/printcap`.

Campo	Significato	Descrizione
if	<i>Input filter</i>	Filtro di ingresso.
lf	<i>Log file</i>	File per la registrazione degli errori.
af	<i>Account file</i>	File per la registrazione della contabilità.
lp	<i>Line printer</i>	File di dispositivo di stampa.
mx	<i>Max</i>	Dimensione massima di una stampa.
pl	<i>Page length</i>	Altezza della pagina in righe.
pw	<i>Page width</i>	Larghezza della pagina in caratteri.
pc	<i>Price</i>	Prezzo per pagina.
rm	<i>Remote machine</i>	Nodo di rete remoto per la stampa.
rp	<i>Remote printer</i>	Coda di stampa remota.

Campo	Significato	Descrizione
sd	<i>Spool directory</i>	Directory usata per la coda.
sf	<i>Suppress feed</i>	Suppressione dell'avanzamento di separazione.
sh	<i>Suppress header</i>	Suppressione dell'intestazione.

I campi possono servire a indicare informazioni di tipo diverso e a seconda di questo cambia il modo con cui i dati relativi sono indicati:

- campi alfanumerici -- dopo il nome del campo viene aggiunto il simbolo '=' seguito dalla stringa racchiusa eventualmente tra apici doppi;
- campi numerici -- dopo il nome del campo viene aggiunto il simbolo '#' seguito dal numero;
- campi booleani -- hanno il valore *Vero* se appaiono nel record.

Campo	Descrizione
<i>if=filtro_di_ingresso</i>	Indica il percorso assoluto del programma, o dello script, che serve come filtro dei dati in ingresso. Il programma o lo script in questione deve essere in grado di ricevere i dati dallo standard input e di emettere il risultato attraverso lo standard output.
<i>lf=file_delle_registrazioni</i>	Permette di specificare il percorso assoluto di un file da utilizzare come registro degli eventi importanti che riguardano la stampa. In generale serve per annotare gli errori.
<i>lp=file_di_dispositivo</i>	È indispensabile e serve a indicare il file di dispositivo corrispondente alla porta presso cui è connessa la stampante.
<i>mx#n</i>	Indica la dimensione massima (in multipli di 1024 byte) di un file di stampa. Di solito, questo campo viene indicato con il valore zero (' <i>mx#0</i> '), per non porre alcun limite di dimensione.
<i>rm=nodo</i>	Indica un nodo di rete a cui ci si deve connettere per richiedere la stampa.
<i>rp=nome_coda_remota</i>	Si utilizza in abbinamento con il campo ' <i>rm</i> ', allo scopo di indicare esplicitamente il nome della coda di stampa remota.
<i>sd=directory_contenente_la_coda</i>	Indica il percorso della directory da utilizzare per la gestione di questa coda. In linea di principio, dovrebbe essere possibile usare una sola directory per più code di stampa, al contrario di quanto si vede nell'esempio introduttivo.
<i>sf</i>	È un campo booleano che, se presente, elimina l'avanzamento della carta alla fine di ogni processo di stampa.
<i>sh</i>	È un campo booleano che, se presente, elimina l'emissione dell'informazione sul programma cliente che ha richiesto la stampa, nella pagina di intestazione (ovvero di separazione) tra un processo di stampa e il successivo. Se viene usato in combinazione con ' <i>sf</i> ', viene eliminata completamente la pagina di separazione.

Di seguito vengono descritte alcune porzioni dell'esempio introdut-

tivo.

```
bare:\
    :sd=/var/spool/lpd/bare:\
    :mx#0:\
    :sh:\
    :lp=/dev/lp0:
```

La voce '**bare**' indica semplicemente le informazioni seguenti:

- la directory usata per la coda di stampa è '*/var/spool/lpd/bare/*', tenendo conto che la scelta del nome finale '*.../bare/*' non è obbligatoria, ma è solo conveniente all'amministratore del sistema;
- l'indicazione della dimensione massima del file è azzerata, stando a significare che non vengono posti dei limiti;
- la pagina di intestazione non viene generata;
- il file di dispositivo corrispondente alla porta della stampante è '*/dev/lp0*'.

La cosa importante da notare in questo tipo di definizione è che non è stato indicato un filtro per i dati. Ciò significa che i dati da inviare alla stampante non subiscono trasformazioni; infatti, il nome '**bare**' è stato scelto opportunamente.

```
lp|laserjet|HP Laserjet:\
    :sd=/var/spool/lpd/lp:\
    :mx#0:\
    :sh:\
    :lp=/dev/lp0:\
    :if=/var/spool/lpd/lp/filtro:
```

Questo record del file '*/etc/printcap*' è più complesso. Per prima cosa si nota che è possibile fare riferimento a questo utilizzando tre nomi possibili: '**lp**', '**laserjet**' o '**HP Laserjet**'. A parte questo, si nota l'inserimento di un filtro di ingresso. Il file '*/var/spool/lpd/lp/filtro*' potrebbe essere un programma o uno script che esegue un qualche tipo di trasformazione sui dati ricevuti.

```
net:\
    :sd=/var/spool/lpd/net:\
    :mx#0:\
    :sh:\
    :rm=weizen.mehl.dg:\
    :rp=lp:\
    :if=/var/spool/lpd/net/filtro:
```

Questo esempio rappresenta un record del file '*/etc/printcap*' che dichiara l'utilizzo di una stampante remota. La differenza sta quindi nel fatto che il campo '**lp**' è assente e al suo posto si utilizzano '**rm**' e '**rp**' per indicare rispettivamente il nome dell'elaboratore remoto (*weizen.mehl.dg*) e il nome della coda presso l'elaboratore remoto.

Quando si utilizza una stampante remota, nel caso in cui i dati da stampare richiedano un'elaborazione attraverso un filtro, occorre decidere se tale elaborazione debba avvenire prima dell'invio, o alla destinazione. In questo caso, viene indicato un filtro attraverso il campo '**if**': probabilmente, la coda corrispondente al nome '**lp**' dell'elaboratore remoto non ha un filtro adatto.

### 27.2.2 Servizio di stampa

Il servizio di stampa nel sistema derivato da BSD è gestito dal demone '**lpd**'. Questo si occupa principalmente di scandire le code e di mettere in stampa ciò che vi dovesse trovare. È anche in grado di ricevere richieste di stampa attraverso la rete, occupandosi di metterle in coda; infine, è anche capace di inviare una richiesta di stampa a un nodo di rete remoto.

In condizioni normali, '**lpd**' non richiede argomenti nella riga di comando; comunque, la sintassi degli argomenti di '**lpd**' è molto diversa da un sistema all'altro.

Ogni sistema di stampa utilizza le proprie tecniche di autorizzazione per concedere l'accesso al servizio di stampa. In generale, un sistema di stampa installato attraverso i pacchetti della propria distribuzione GNU/Linux dovrebbe consentire la stampa quando questa è richiesta a partire dallo stesso elaboratore locale; mentre per consentire l'accesso dall'esterno, occorre predisporre altri file di configurazione che non sono standard.

Di solito, il servizio di stampa viene avviato e arrestato attraverso uno script della procedura di inizializzazione del sistema, che potrebbe assomigliare all'esempio seguente:

```
#!/bin/sh

test -f /usr/sbin/lpd || exit 0

case "$1" in
  start)
    printf "Avvio del servizio di stampa: "
    /usr/sbin/lpd
    echo
    ;;
  stop)
    printf "Disattivazione del servizio di stampa: "
    killall lpd
    echo
    ;;
  *)
    echo "Utilizzo: lpd {start|stop}"
    exit 1
esac
```

Dal momento che la stampa è controllata da un demone, quando si modifica il file di configurazione `/etc/printcap`, bisogna fare in modo che `lpd` lo rileggi. Questo lo si può ottenere arrestando e riavviando il servizio, oppure inviando al processo del demone un segnale di aggancio (`SIGHUP`):

```
kill -HUP pid_di_lpd
```

### 27.2.3 Stampante predefinita

Il file `/etc/printcap` permette di definire le code di stampa, comprese quelle che fanno riferimento a servizi remoti. Tra queste code, è necessario stabilire quale sia quella predefinita, ovvero quella che deve essere presa in considerazione quando non vi si fa riferimento in modo esplicito.

La coda predefinita (ovvero la stampante predefinita) corrisponde per tradizione al nome `lp`, ma questa definizione può essere alterata utilizzando la variabile di ambiente `PRINTER`. Se esiste, definisce il nome della stampante predefinita, altrimenti resta `lp`.

È importante tenere presente che la politica del proprio sistema di stampa potrebbe essere anche differente; per esempio, in mancanza di indicazioni la coda predefinita potrebbe essere quella corrispondente alla prima dichiarazione del genere nel file `/etc/printcap`. A questo proposito, è bene che la definizione della coda tradizionale `lp`, sia sempre la prima.

### 27.2.4 Clienti di stampa

Il cliente del sistema di stampa è un programma in grado di accodare una stampa. In generale, nei sistemi di stampa derivati da quello di BSD si utilizza il programma `lpr`, ma nello standard POSIX è previsto invece il programma `lp`:

```
lpr [opzioni] [file...]
```

```
lp [opzioni] [file...]
```

In condizioni normali, questi programmi sono in grado di mettere in stampa i file indicati alla fine della riga di comando, oppure, in loro mancanza, utilizzano per questo lo standard input. Sono molto poche le opzioni standard di questi programmi e, in generale, la cosa più importante è la definizione della coda di stampa a cui si vuole inviare il file:

```
lpr -P coda [file...]
```

```
lp -d coda [file...]
```

Lo schema sintattico semplificato mostra esattamente questa possibilità, sia per `lpr` che per `lp`. Si osservi in particolare che nel caso di `lpr`, la tradizione prevede anche che il nome della coda possa essere attaccato alla lettera dell'opzione.

Tabella 27.16. Alcune opzioni di `lpr`.

Opzione	Descrizione
<code>-P <i>coda</i></code>	Permette di specificare una coda di stampa particolare, tra quelle previste all'interno di <code>/etc/printcap</code> . Se non viene utilizzata questa opzione, si fa riferimento alla stampante predefinita (che di solito è <code>lp</code> ).
<code>-m</code>	Al termine della stampa, invia un messaggio attraverso <code>mail</code> all'utente che ha avviato il programma.
<code>-#<i>n_copie</i></code>	Permette di specificare il numero di copie che si vuole siano stampate. Il numero di copie è indicato da un numero che segue il simbolo <code>#</code> .

Tabella 27.17. Alcune opzioni di `lp`.

Opzione	Descrizione
<code>-d <i>coda</i></code>	Permette di specificare una coda di stampa particolare, tra quelle previste all'interno di <code>/etc/printcap</code> . Se non viene utilizzata questa opzione, si fa riferimento alla stampante predefinita (che di solito è <code>lp</code> ).
<code>-m</code>	Al termine della stampa, invia un messaggio attraverso <code>mail</code> all'utente che ha avviato il programma.
<code>-n <i>n_copie</i></code>	Permette di specificare il numero di copie che si vuole siano stampate.

Segue la descrizione di alcuni esempi.

```
• $ lpr lettera [Invio]
```

oppure

```
$ lp lettera [Invio]
```

Accoda la stampa del file `lettera` utilizzando la coda predefinita.

```
• $ lpr -P laser lettera [Invio]
```

oppure

```
$ lp -d laser lettera [Invio]
```

Accoda la stampa del file `lettera` utilizzando la coda identificata con il nome `laser` all'interno del file `/etc/printcap`.

```
• $ lpr -Plaser lettera [Invio]
```

Esattamente come nell'esempio precedente.

```
• $ ls -l | lpr [Invio]
```

oppure

```
$ ls -l | lp [Invio]
```

Accoda la stampa dell'elenco della directory corrente. In pratica, viene accodato quanto proveniente dallo standard input che proviene dal comando `ls -l`.

### 27.2.5 Esame delle code di stampa

Per conoscere la situazione delle code di stampa si utilizza il comando `lpq`:

```
lpq [opzioni] [numero_processo_di_stampa...] [utente...]
```

Il programma `lpq` esamina le code di stampa e restituisce lo stato di una o di tutte le stampe accodate dall'utente specificato. Se `lpq` viene eseguito senza alcun argomento, restituisce lo stato di tutte le stampe accodate.

Tabella 27.18. Alcune opzioni.

Opzione	Descrizione
<code>-P coda</code>	Permette di specificare una coda particolare. Se non viene specificato, si fa riferimento a quella predefinita.
<code>-l</code>	Restituisce maggiori informazioni su ogni processo di stampa.

### 27.2.6 Rimozione dei processi di stampa dalle code

I processi di stampa che risultano ancora visibili nelle code, possono essere rimossi dall'utente che li ha generati, o dall'utente `root`.

```
lprm [opzioni] [utente...]
```

Permette di rimuovere uno o più processi di stampa accodati precedentemente. Il nome dell'utente può essere specificato solo se il comando viene utilizzato dall'utente `root`, nel senso che solo lui può interrompere la stampa di altri utenti. Se non viene specificato il nome dell'utente, si intende che si tratti dello stesso che ha eseguito `lprm`. Se non vengono specificati argomenti, l'esecuzione del comando `lprm` implica l'eliminazione della stampa in corso per l'utente che lo ha richiesto. Naturalmente, ciò vale solo se l'utente in questione ha, in quel momento, una stampa in esecuzione.

Se l'utente `root` utilizza `lprm` senza specificare un utente a cui fare riferimento, ottiene l'eliminazione di tutti i processi di stampa nelle code, attivi o meno che siano.

Tabella 27.19. Alcune opzioni.

Opzione	Descrizione
<code>-P coda</code>	Permette di specificare una coda particolare. Se non viene specificata, si fa riferimento alla coda predefinita.
<code>numero_processo_di_stampa...</code>	Se tra gli argomenti vengono indicati uno o più numeri, questi si intendono riferiti ai processi di stampa che si vogliono eliminare.

### 27.2.7 Controllo del sistema di stampa

L'utente `root` controlla il sistema di stampa, ovvero il funzionamento dei vari demoni `lpd`, attraverso il programma `lpc`:

```
lpc [comando [argomento...]]
```

Le possibilità effettive di `lpc` dipendono dalle caratteristiche del sistema di stampa. In generale, per ogni coda di stampa configurata all'interno di `/etc/printcap`, `lpc` può eseguire le azioni seguenti:

- disabilitare o abilitare una stampante;
- disabilitare o abilitare una coda di stampa;
- modificare l'ordine dei processi di stampa in coda;
- visualizzare lo stato delle stampanti, delle code relative e dei demoni che se ne occupano.

Se `lpc` viene avviato senza argomenti, si attiva la modalità di comando evidenziata dalla presenza dell'invito `lpc>`. Se invece vengono forniti degli argomenti, il primo di questi viene interpretato

come un comando, mentre i restanti come parametri del comando. È possibile inviare a `lpc`, attraverso lo standard input, un file contenente una serie di comandi.

Il programma `lpc` può essere eseguito anche da un utente comune, ma in tal caso sono disponibili solo alcune funzioni.

Tabella 27.20. Comandi a disposizione di tutti gli utenti.

Comando	Descrizione
<code>? [comando...]</code>	Visualizza una descrizione sintetica dei comandi elencati, oppure, se non ne viene indicato alcuno, l'elenco di tutti i comandi a disposizione.
<code>help [comando...]</code>	
<code>exit</code> <code>quit</code>	Termina l'esecuzione di <code>lpc</code> .
<code>status {all   coda}</code>	Visualizza lo stato della coda di stampa locale indicata, oppure di tutte, se si utilizza la parola chiave <code>all</code> .

Tabella 27.21. Comandi a disposizione dell'utente `root`.

Comando	Descrizione
<code>abort {all   coda}</code>	Termina l'esecuzione del demone attivo che si occupa della stampa nell'elaboratore locale, quindi disabilita la stampa, prevenendo l'avvio di altri demoni da parte di <code>lpr</code> . Quando viene riavviata la stampa, riprende il processo di stampa attivo nel momento dell'interruzione.
<code>enable {all   coda}</code> <code>disable {all   coda}</code>	Abilita o disabilita l'uso della coda specificata, consentendo o impedendo la generazione di nuovi processi di stampa.
<code>up {all   coda}</code> <code>down {all   coda} messaggio</code>	Abilita o disabilita l'uso della coda indicata, consentendo o bloccando la stampa dei processi di stampa esistenti.
<code>start {all   coda}</code> <code>stop {all   coda}</code>	Avvia o arresta il demone che gestisce la coda. Nel caso di arresto, questo avviene al termine del processo di stampa eventualmente in esecuzione. Tuttavia, gli utenti possono continuare ad accodare stampe. Il comando <code>start</code> può essere utile anche per riavviare un demone che per qualche ragione ha cessato di funzionare in modo inatteso.
<code>topq coda ↵</code> <code>↵ [numero_processo_di_stampa...]</code>	Cambia l'ordine di esecuzione dei processi di stampa ponendo quelli indicati in precedenza rispetto agli altri.

### 27.2.8 Particolarità del sistema BSD vero e proprio

Il sistema di stampa BSD prevede l'uso dei file `/etc/hosts.equiv` e `/etc/hosts.lpd`. Questi servono a elencare i nomi degli elaboratori remoti cui è consentito collegarsi per ottenere l'accesso al sistema di stampa locale. Per la precisione, è il file `/etc/hosts.lpd` che dovrebbe essere utilizzato per questo tipo di autorizzazione; tuttavia, dal momento che l'elenco contenuto in `/etc/hosts.equiv` serve già per consentire l'accesso attraverso programmi come Rsh (36.7.2), è ragionevole che anche a questi sia concesso di accedere al servizio di stampa.

È importante ribadire che con questo sistema di stampa, se non si predispongono correttamente il file `/etc/hosts.lpd`, oppure il file `/etc/hosts.equiv`, o entrambi, non si ottiene l'accesso da clienti remoti.

Con il sistema di stampa BSD non è possibile accedere a stampanti remote se non è stata prevista una coda locale corrispondente nel

file di configurazione `/etc/printcap` (con l'uso dei campi `'rm'` e `'rp'`). Per questo esistono anche dei programmi di servizio specifici che instaurano una connessione con il sistema remoto di stampa in modo autonomo. Si tratta di `'r1pr'` e `'r1pq'`, che vengono descritti più avanti.

### 27.2.9 Particolarità di LPRng

« Il sistema di stampa LPRng<sup>2</sup> è molto più evoluto rispetto a quello della tradizione BSD, anche se di solito viene utilizzato in modo abbastanza conforme a quello; tuttavia consentirebbe di accedere a delle estensioni sofisticate, soprattutto per ciò che riguarda la stampa remota.

LPRng fa uso di `/etc/printcap` e di altri file di configurazione; precisamente si tratta di `/etc/lprng/lpd.conf` e di `/etc/lprng/lpd.perms`. Per quanto riguarda `/etc/printcap`, c'è da osservare che i record di definizione delle code, possono essere continuati su più righe, anche senza utilizzare il simbolo di continuazione (`'\'`).

Se il pacchetto utilizzato per installare LPRng è stato predisposto correttamente, non dovrebbe essere necessario indicare alcunché nel file di configurazione `/etc/lprng/lpd.conf`, che di solito viene fornito commentato completamente, con gli esempi delle varie direttive che vi potrebbero apparire. Eventualmente, può essere interessante abilitare la stampa di più copie attraverso la direttiva seguente nel file `/etc/lprng/lpd.conf`:

```
# Max number of copies.
mc#10000000
```

Il file `/etc/lprng/lpd.perms` serve a definire i permessi di accesso al servizio. Di solito, questo file viene fornito già predisposto per l'utilizzo locale normale; se si vuole concedere l'accesso da parte di clienti remoti è indispensabile modificare questo file, allo scopo di attivare i permessi necessari. Con ciò, si può intendere che LPRng non considera i file `/etc/hosts.equiv` e `/etc/hosts.lpd`.

```
# concede all'utente root sul servente di controllare i
# processi di stampa
ACCEPT SERVICE=C SERVER REMOTEUSER=root

# concede a chiunque di ottenere lo stato dei processi di
# stampa
ACCEPT SERVICE=S

# rifiuta le richieste di stampa dai nodi remoti
REJECT SERVICE=XRPQ NOT SERVER

# rifiuta tutto il resto
REJECT SERVICE=CSU

# concede agli utenti che accedono dai nodi originari,
# di eliminare i propri processi di stampa
ACCEPT SERVICE=M SAMEHOST SAMEUSER

# concede all'utente root sul servente di eliminare i
# processi di stampa
ACCEPT SERVICE=M SERVER REMOTEUSER=root
REJECT SERVICE=M

# tutte le altre operazioni sono concesse
DEFAULT ACCEPT
```

L'esempio appena mostrato rappresenta un file `/etc/lprng/lpd.perms` tipico, dove in particolare sono esclusi gli accessi da parte di clienti remoti. Per fare in modo di consentire l'accesso sommario da parte di una sottorete, si può modificare la direttiva seguente:

```
REJECT SERVICE=XRPQ NOT SERVER
```

Questa deve essere trasformata così:

```
REJECT SERVICE=XRPQ NOT SERVER NOT REMOTEIP=192.168.0.0/255.255.0.0
```

In tal modo, secondo l'esempio, si concede a tutta la sottorete `192.168.*.*` di accedere.

Naturalmente, si può concedere l'accesso a chiunque, senza limi-

tazioni, mettendo nel file `/etc/lprng/lpd.perms` soltanto la direttiva seguente:

```
DEFAULT ACCEPT
```

Un vantaggio importante nell'uso di LPRng sta nella possibilità di accedere direttamente a servizi di stampa remoti, senza dover passare per una coda locale configurata nel file `/etc/printcap`. Tutto è molto semplice: nelle situazioni in cui è consentito indicare il nome di una coda di stampa, si può usare la notazione seguente per accedere direttamente al servizio remoto corrispondente:

```
coda@nodo
```

L'esempio seguente invia alla stampa, presso la coda `'lp'` del nodo di rete `roggen.brot.dg`, il file `'lettera'`.

```
$ lpr -P lp@roggen.brot.dg lettera[Invio]
```

Infine, è bene tenere presente che è possibile verificare la correttezza della configurazione attraverso il programma di servizio `'checkpc'` (*Check printcap*):

```
checkpc [opzioni] [file_printcap]
```

Di solito si utilizza `'checkpc'` senza argomenti di alcun tipo, allo scopo di controllare il file `/etc/printcap` (ovvero quello predefinito), gli altri file di configurazione e le directory delle code. Il controllo riguarda sia la configurazione, sia i permessi dei file. È molto importante l'opzione `'-f'`, con la quale si richiede a `'checkpc'` di provvedere da solo a sistemare ciò che è possibile. Naturalmente, l'uso di `'checkpc'` con l'opzione `'-f'` è riservato all'utente `'root'`.

Prima di utilizzare `'checkpc'` è opportuno concludere il funzionamento di tutti i demoni `'lpd'` che fossero eventualmente in funzione.

```
# checkpc -f[Invio]
```

A titolo di esempio viene mostrato quello che potrebbe essere generato da questo comando:

```
Checking permission file '/etc/lpd.perms:/usr/etc/lpd.perms'
Freeing Perms
Done Perms
LPD lockfile '/var/spool/lpd/lpd.lock.tizio.printer'
  Checking directory: '/var/spool/lpd'
    checking file '/var/spool/lpd/lpd.lock.tizio.printer'
Truncating LPD log file '/var/spool/lpd/lpd.log.tizio'
Checking /var/spool/lpd/lpd.log.tizio file ↵
↳ /var/spool/lpd/lpd.log.tizio'
checkpc: Warning - cannot open '/var/spool/lpd/lpd.log.tizio'
lp: Checking printer 'lp'
lp: Checking directory: '/var/spool/lpd/lp'
lp: checking file '/var/spool/lpd/lp/control.lp'
lp: checking file '/var/spool/lpd/lp/status.lp'
lp: checking file '/var/spool/lpd/lp/status'
lp: checking file '/var/log/lp-errs'
lp: checking file '/var/log/lp-acct'
lp: Checking log file '/var/log/lp-errs'
lp: 'log' file 0 bytes long: no truncation
lp: Checking accounting file '/var/log/lp-acct'
lp: 'accounting' file 2316 bytes long: no truncation
lp: Checking filter status file '/var/spool/lpd/lp/status'
lp: 'filter status' file 0 bytes long: no truncation
```

### 27.2.10 Stampare attraverso X con un sistema di stampa BSD

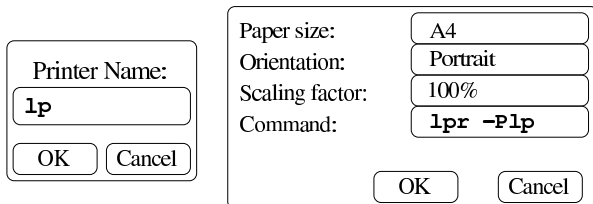
« Quando si inizia a utilizzare il sistema grafico X con la gestione della stampa basata sul modello tradizionale di BSD, ci si deve accontentare di stampare senza pretendere di interagire con la stampante. Tuttavia può essere necessario conoscere le code di stampa disponibili, secondo la configurazione del file `/etc/printcap`, e di conseguenza può essere richiesto di verificare il comando usato effettivamente dai programmi per la stampa.



Generalmente, i programmi che hanno la necessità di stampare propongono una riga di comando per la stampa, per cui sarebbe anche possibile utilizzare un sistema di stampa che dispone di un cliente diverso dal solito programma `'lpr'`.

Alcuni programmi più vecchi possono stampare soltanto con l'aiuto di un programma che si chiami `'lpr'` e che si comporti come quello tradizionale di BSD, richiedono solo l'indicazione della voce del file `'/etc/printcap'` e quindi pretendendo di utilizzare l'opzione `'-P'`.

Figura 27.28. A sinistra si vede un esempio di programma che è in grado di stampare solo attraverso `'lpr'`. Se non viene indicato il nome di una coda di stampa, si fa riferimento a `'lp'`, o comunque a quella predefinita. A destra si vede un programma normale che permette l'indicazione di una riga di comando completa (o quasi). In ogni caso non deve essere inserito il nome del file da stampare che di norma viene fornito attraverso lo standard input, oppure come ultimo argomento.



### 27.2.11 Librerie GTK 2

Alcuni programmi che richiedono la grafica, basano l'accesso alla stampa sulle funzionalità della libreria GTK. In tal caso, per poter utilizzare un servizio di stampa in stile BSD, occorre predisporre la configurazione della libreria per questo scopo, con il file `'/etc/gtk-2.0/gtkrc'`. In questo file serve la riga seguente:

```
gtk-print-backends = "file,lpr,cups"
```

Utilizzando la direttiva mostrata nell'esempio, così come si vede, si abilita la possibilità di stampare su file, su un sistema di stampa BSD ed eventualmente anche su CUPS.

### 27.2.12 Sviluppo successivo

Il sistema di stampa BSD tradizionale consente soltanto di inviare un file alla stampante. Sta all'amministratore del sistema configurare le code di stampa in modo da predisporre dei filtri appropriati.

Così facendo, però, diventa difficile richiedere alla stampante delle opzioni di funzionamento particolari; per esempio non si può selezionare il formato della carta (supponendo di disporre di cassette con formati alternativi), così come non è possibile abilitare o disabilitare la stampa fronte-retro (*duplex*) automatica, a meno di agire fisicamente sulla stampante.

Per risolvere questo problema, esiste la possibilità di definire dei file di configurazione particolari, associati ognuno a un tipo specifico di stampante: i file PPD (*PostScript printer description*). Benché il nome suggerisca la relazione stretta con stampanti PostScript, in pratica si possono realizzare file PPD anche per altri tipi.

Per approfondire l'argomento si può leggere il documento *Tutorial on CUPS and Foomatic*, in particolare la parte terza: *Some theoretical background: CUPS, PPDs, PostScript and GhostScript*, di Till Kampeter e Kurt Pfeifle <http://www.openprinting.org/download/kpfeifle/LinuxKongress2002/Tutorial/III.PostScript-and-PPDs/III.PostScript-and-PPDs.html>. Il sistema di stampa CUPS viene introdotto nella sezione 27.8, ma senza approfondire il problema dei file PPD.

## 27.3 Stampa remota attraverso Rlpr

Il pacchetto di programmi Rlpr,<sup>3</sup> estraneo al sistema di stampa BSD tradizionale, può essere molto utile per stampare utilizzando servizi remoti, senza passare per la configurazione del file `'/etc/printcap'` locale. Tuttavia, è il caso di ricordare che non c'è bisogno di tali programmi nel caso si disponga già di un sistema di stampa LPRng, in cui i programmi clienti normali sono in grado di fare questo da soli.

Rlpr sostituisce in pratica alcuni programmi di servizio, tipici del sistema di stampa BSD, con altri, caratterizzati da una lettera «r» iniziale, che sta per «remoto»:

```
rlpr [opzioni] [file_da_stampare...]
```

```
rlpq [opzioni] [numero_processo_di_stampare...] [utente...]
```

```
rlprm [opzioni] [utente...]
```

Come si può intuire, questi programmi sostituiscono, nell'ordine, `'lpr'`, `'lpq'` e `'lprm'`.

Questi programmi, per compiere il loro lavoro correttamente, richiedono i privilegi dell'utente `'root'`, pertanto hanno solitamente il bit SUID attivo, appartenendo all'utente `'root'` (SUID-root), come documentato chiaramente nelle pagine di manuale *rlpr(1)*, *rlpq(1)* e *rlprm(1)*.

Molte delle opzioni dei programmi originali, sono preservate; con qualche aggiunta e qualche modifica. In particolare:

Opzione	Descrizione
<code>-H nodo</code> <code>--printhead=nodo</code>	Definisce l'elaboratore remoto al quale ci si vuole rivolgere per ottenere la stampa. Al posto di utilizzare questa opzione si può sfruttare il nome della coda di stampa per includervi anche l'indicazione del nodo di rete remoto. Si osservi a tal fine la descrizione dell'opzione <code>'-P'</code> .
<code>-Pcoda [@nodo]</code> <code>--printer=coda [@nodo]</code> <code>--queue=coda [@nodo]</code>	Seleziona la coda di stampa remota. Se non si utilizza l'opzione <code>'-H'</code> , si può usare la notazione <code>coda@nodo</code> .

A titolo di esempio vengono mostrati due comandi equivalenti, per ottenere la stampa del contenuto della directory corrente, attraverso la coda `'lp'` presso l'indirizzo 192.168.1.1:

```
$ ls -l | rlpr --printhead=192.168.1.1 -Plp [Invio]
```

```
$ ls -l | rlpr -Plp@192.168.1.1 [Invio]
```

## 27.4 Stampa automatica da un file

Può capitare di utilizzare programmi che sono in grado esclusivamente di stampare utilizzando un file su disco. Per fare in modo che questo file venga trasmesso automaticamente alla stampante, si può realizzare uno script come quello seguente:

```
#!/bin/sh
PRINT_FILE=$1
PREVIOUS_SIZE="0"
while true
do
    sleep 2
    SIZE=`ls -s --block-size=1 $PRINT_FILE \
        | sed "s/^\ *([0-9]*)\ *.*$/\1/"`
    if [ "$PREVIOUS_SIZE" = "$SIZE" ] && [ "$SIZE" != "0" ]
    then
        lpr $PRINT_FILE
        cat /dev/null > $PRINT_FILE
        PREVIOUS_SIZE="0"
    else

```

```

PREVIOUS_SIZE="$SIZE"
fi
done

```

In pratica, dopo un'attesa di due secondi, viene verificata la dimensione del file, confrontandola con la dimensione precedente (inizialmente è zero). Se la dimensione è cambiata, il file viene inviato alla stampa e viene annullato il suo contenuto.

Per interpretare cosa accade nel comando che serve a estrarre la dimensione del file, si può immaginare di averne uno di 1 Mibyte:

```
$ ls -s --block-size=1 mio_file.ps [Invio]
```

```
1048576 mio_file.ps
```

Il comando `'sed'` riceve questo risultato attraverso lo standard input ed estrae solo il valore numerico:

```
$ echo 1048576 mio_file.ps ↵
↵| sed "s/^ *\[0-9\]*.*$/\1/" [Invio]
```

```
1048576
```

Questo meccanismo di stampa non può essere perfetto, però può essere una soluzione accettabile in mancanza di altro. Supponendo che lo script si chiami `'STAMPA'`, supponendo che il file da controllare sia `'/tmp/mio_file.ps'`, si potrebbe usare così:

```
$ STAMPA /tmp/mio_file.ps & [Invio]
```

## 27.5 File e filtri per la stampa

«

Il sistema che gestisce le code dei processi di stampa e la stampa remota, rappresenta solo una parte della soluzione del problema della stampa. È importante chiarire in che modo devono essere composti i file per la stampa e come questi vanno gestiti dai filtri di stampa.

### 27.5.1 File per la stampa

«

Negli ambienti Unix si utilizzano normalmente due tipi fondamentali di file per la stampa:

- file di testo, o ASCII;
- file PostScript.

Teoricamente, i file di testo sono stampabili con qualunque tipo di stampante, mentre i file PostScript richiedono una stampante PostScript. In pratica, quasi sempre non è possibile stampare un file di testo così com'è e raramente si dispone di una stampante PostScript.

Negli ambienti Unix i file di testo (o file ASCII) seguono la convenzione dell'interruzione di riga attraverso il codice ASCII `<LF>`. Con il sistema operativo Dos è stato introdotto un codice differente, corrispondente a `<CR><LF>`. La maggior parte delle stampanti in circolazione è adatta a questo ultimo tipo di interruzione di riga, per cui, il solo carattere `<LF>` produce un avanzamento alla riga successiva, senza il ritorno alla prima colonna. Quando si invia un file di testo in stile Unix a una stampante che richiede l'interruzione di riga in stile Dos, si ottiene il noto *effetto scalettatura*. Per esempio, si osservi il testo seguente:

```

Uno
Due
Tre
Quattro
Cinque
Sei
Sette

```

La stampa potrebbe produrre questo effetto:

```

Uno
  Due
    Tre
      Quattro
        Cinque
          Sei
            Sette

```

Per ovviare a tale inconveniente, prima di inviare un file di testo Unix a una stampante normale, occorre trasformare i codici di interruzione di riga in modo che comprendano sia `<CR>`, sia `<LF>`.

Il programma che tipicamente è in grado di eseguire questa conversione è `'unix2dos'`. Di questo ne esistono diverse edizioni incompatibili tra loro, accomunate solo dallo scopo. Qui si fa riferimento a un programma filtro, ovvero a uno `'unix2dos'` che riceve il file da convertire dallo standard input e restituisce il risultato attraverso lo standard output (è importante sottolineare questo fatto perché esistono delle versioni che non si comportano così). I filtri di stampa sono descritti più avanti in questo capitolo, per il momento dovrebbe bastare sapere che si può utilizzare il programma `'unix2dos'` (o un altro programma simile) prima di inviare il file al programma `'lpr'`, come si vede nell'esempio seguente:

```
$ cat esempio.txt | unix2dos | lpr [Invio]
```

Come accennato, il programma `'unix2dos'` non è standard e a volte si può incontrare una versione che non funziona esattamente come negli esempi indicati qui. Eventualmente, se la propria distribuzione GNU/Linux dispone di questo programma di servizio, conviene consultare la sua documentazione: *unix2dos(1)*.

In alternativa al programma `'unix2dos'` si può scrivere uno script Perl molto semplice e intuitivo, anche per chi non conosce tale linguaggio (che viene descritto nel capitolo 24).

```

#!/usr/bin/perl
# filtro-crlf.pl < FILE_INPUT > FILE_OUTPUT
$riga = "";
while ($riga = <STDIN>)
{
  # Elimina il codice di interruzione di riga finale.
  chomp $riga;
  # Emette la riga con l'aggiunta di <CR> e <LF>.
  print "$riga\r\n";
};

```

Il sistema PostScript ha introdotto una sorta di rivoluzione nel modo di stampare: attraverso un linguaggio standardizzato ha reso la stampa indipendente dal tipo particolare di stampante utilizzato. L'unico inconveniente delle stampanti PostScript è il prezzo.

Fortunatamente, negli ambienti Unix è disponibile il programma Ghostscript in grado di trasformare un file PostScript in diversi formati, ognuno compatibile con un tipo diverso di stampante.

Nella maggior parte dei casi, quando cioè non si dispone di una stampante PostScript, si devono convertire i file PostScript in un formato accettabile dalla propria stampante. L'uso dei filtri di stampa permette di automatizzare questa operazione. Nella sezione 26.1 viene descritto con maggiore dettaglio in che modo questi file PostScript possono essere gestiti.

### 27.5.2 Filtri di stampa

Attraverso il file `'/etc/printcap'`, per ogni singolo record di descrizione di una coda, è possibile definire un gran numero di filtri di stampa, ognuno con uno scopo particolare. Di fatto, è preferibile limitarsi a utilizzarne uno solo, precisamente quello del campo `'if'`, o *Input filter*. Il programma o lo script indicato nel campo `'if'` riceve alcuni argomenti:

```
filtro_if [opzioni]
```

In particolare:

Argomento	Descrizione
-c	viene passato solo quando il programma <code>'lpr'</code> riceve l'opzione <code>'-1'</code> e dovrebbe servire al programma filtro per sapere che non deve applicare alcuna trasformazione;

«

Argomento	Descrizione
-wlarghezza	rappresenta la larghezza del foglio in caratteri, ammesso che ciò possa avere significato;
-llunghezza	rappresenta la lunghezza del foglio in righe, ammesso che ciò possa avere significato;
-irientro	rappresenta il rientro sinistro in caratteri, ammesso che ciò possa avere significato;
-n utente	rappresenta il nome dell'utente che ha avviato la stampa;
-h nodo	rappresenta l'elaboratore dal quale parte la richiesta di stampa.

A meno di voler studiare in modo approfondito l'uso del sistema di stampa di cui si dispone, la maggior parte di questi argomenti sono inutilizzabili. È molto più facile costruire un file di configurazione aggiuntivo, da fare leggere al filtro ogni volta che viene avviato, piuttosto che pretendere di fare tutto attraverso l'interpretazione degli argomenti ottenuti automaticamente. In ogni caso, si può contare su due argomenti, eventualmente utilizzabili per produrre intestazioni, o per produrre un registro (un *log*): il nome dell'utente e il nome dell'elaboratore.

### 27.5.2.1 Filtro diagnostico

Gli argomenti forniti al filtro di stampa potrebbero essere diversi da quanto dichiarato dalla documentazione; comunque, vale la pena di verificare cosa succede costruendo la prima volta un filtro diagnostico simile allo script seguente:

```
#!/bin/sh
pwd > /tmp/test-stampa
echo $1 >> /tmp/test-stampa
echo $2 >> /tmp/test-stampa
echo $3 >> /tmp/test-stampa
echo $4 >> /tmp/test-stampa
echo $5 >> /tmp/test-stampa
echo $6 >> /tmp/test-stampa
echo $7 >> /tmp/test-stampa
echo $8 >> /tmp/test-stampa
echo $9 >> /tmp/test-stampa
echo ${10} >> /tmp/test-stampa
echo ${11} >> /tmp/test-stampa
echo ${12} >> /tmp/test-stampa
echo ${13} >> /tmp/test-stampa
echo ${14} >> /tmp/test-stampa
echo ${15} >> /tmp/test-stampa
echo ${16} >> /tmp/test-stampa
echo ${17} >> /tmp/test-stampa
echo ${18} >> /tmp/test-stampa
echo ${19} >> /tmp/test-stampa
echo ${20} >> /tmp/test-stampa
```

Come si può vedere, viene creato il file `/tmp/test-stampa` con l'indicazione della directory corrente (`pwd`) e quindi l'elenco dei contenuti dei vari parametri, ovvero l'elenco degli argomenti ricevuti. La voce (il record) di `/etc/printcap` che utilizza questo filtro potrebbe essere composta nel modo seguente (`/var/spool/lpd/prova/filtro-prova` è il nome dello script visto sopra).

```
prova:\
:sd=/var/spool/lpd/prova:\
:mx#0:\
:sh:\
:lp=/dev/lp0:\
:if=/var/spool/lpd/prova/filtro-prova:
```

Quando si stampa utilizzando la voce `prova` non si ottiene alcuna stampa: viene creato il file `/tmp/test-stampa`.<sup>4</sup>

```
tizio@dinkel.brot.dg$ lpr -Pprova lettera [Invio]
```

Il comando precedente, avviato dall'utente `tizio` a partire dall'elaboratore `dinkel.brot.dg`, dovrebbe generare il file `/tmp/test-stampa` con il contenuto seguente:

```
/var/spool/lpd/prova
-hdinkel.brot.dg
-l66
-ntizio
-w80
...
```

Al contrario, un sistema imperfetto potrebbe non corrispondere alle aspettative. Si osservi a questo proposito l'esito seguente che è stato ottenuto in passato da un sistema di stampa BSD che non si comportava nel modo previsto:

```
/var/spool/lpd/prova
-w132
-l66
-i0
tizio
-h
dinkel.brot.dg
```

Qui si può notare che l'opzione `-n` non esiste e al suo posto viene fornito il nome dell'utente senza il prefisso di alcuna opzione.

Una cosa utile da osservare è la directory corrente: corrisponde sempre alla directory della coda di stampa.

### 27.5.2.2 Filtri elementari

Quando si realizza un filtro di stampa personalizzato, raramente si vanno a cercare sottigliezze che sono comunque già disponibili all'interno di pacchetti di filtri già fatti da altri. Di solito ci si accontenta di trasformare lo standard input e di restituire uno standard output adatto alle proprie esigenze, ignorando completamente gli argomenti che il filtro riceve.

L'esempio tipico è il filtro che permette di stampare un file di testo in stile Unix su una stampante che richiede la conclusione della riga attraverso `<CR><LF>`. Come già accennato all'inizio del capitolo, basta utilizzare il programma `unix2dos` (purché ci sia e funzioni come filtro), oppure lo script che è stato mostrato.

Bisogna fare attenzione: il filtro di stampa riceve degli argomenti, anche se questi non servono. Se si tenta di utilizzare `unix2dos`, o qualunque altro programma direttamente come filtro, si rischia di ottenere solo una segnalazione di errore in quanto potrebbe non essere in grado di comprendere gli argomenti ricevuti. Per risolvere il problema, occorre realizzare uno script, in modo da poter eliminare gli argomenti inutilizzati.

Segue l'esempio di una voce del file `/etc/printcap`:

```
testo:\
:sd=/var/spool/lpd/testo:\
:mx#0:\
:sh:\
:lp=/dev/lp1:\
:if=/var/spool/lpd/testo/filtro:
```

Segue l'esempio dello script utilizzato come filtro:

```
#!/bin/sh
/usr/bin/unix2dos
```

È necessario osservare un paio di particolari:

- è importante indicare il filtro con il suo percorso assoluto;
- i programmi utilizzati all'interno di uno script che funge da filtro di stampa devono essere indicati con il loro percorso assoluto.

Infatti, non si può contare sulla conoscenza della directory corrente nel momento in cui questi vengono messi in esecuzione e nemmeno sulla disponibilità di una variabile di ambiente `PATH` appropriata.

### 27.5.2.3 Filtri PostScript

Tutti i filtri di stampa in grado di convertire file PostScript in qualcosa di stampabile senza una stampante PostScript, si avvalgono del programma Ghostscript (`gs`). L'esempio seguente mostra uno script che riceve dallo standard input un file PostScript e restituisce

attraverso lo standard output un file stampabile con una HP Laserjet o compatibile.

```
#!/bin/sh
/usr/bin/gs -q -dNOPAUSE -sPAPERSIZE=a4 -sDEVICE=laserjet \
-sOutputFile=- -
```

Dal momento che diventa complicato per Ghostscript gestire una quantità smisurata di stampanti differenti, esiste anche la possibilità di mettere in comunicazione questo con un altro programma, attraverso un protocollo noto come IJS. Il protocollo IJS consente di realizzare un programma specifico per la gestione della stampante, controllabile però da Ghostscript. Il programma più importante (e probabilmente l'unico) in grado di comunicare con Ghostscript attraverso il protocollo IJS è HPIJS (sviluppato originariamente da Hewlett-Packard per le proprie stampanti, ma utilizzabile attualmente anche per quelle di altre case produttrici),<sup>5</sup> costituito dal file eseguibile `'hpijs'`. Utilizzando HPIJS, il programma `'hpijs'` non viene avviato da solo, ma attraverso `'gs'`. Per esempio, volendo gestire una stampante HP Laserjet, come nell'esempio già visto, ma attraverso HPIJS, si potrebbe modificare lo script già presentato nel modo seguente:

```
#!/bin/sh
/usr/bin/gs -q -dNOPAUSE \
-sDEVICE=ijs -sIjsServer=hpijs -dIjsUseOutputFD \
-sDeviceManufacturer="HEWLETT-PACKARD" \
-sDeviceModel="HP LaserJet" \
-sOutputFile=- -
```

#### 27.5.2.4 Problemi

In passato è capitato che una versione particolare del sistema di stampa BSD per GNU/Linux avesse un difetto che non le permetteva di utilizzare il flusso di dati proveniente dal filtro di stampa. Nel caso dovesse verificarsi nuovamente questa situazione, si può utilizzare un trucco: il filtro di stampa riceve i dati dallo standard input nel modo solito e li trasforma. Quindi, invece di emettere il risultato della sua elaborazione attraverso lo standard output, lo invia a un'altra coda di stampa.

In pratica, si può supporre che il file `'/etc/printcap'` sia composto come segue:

```
lp:\
:sd=/var/spool/lpd/lp:\
:mx#0:\
:sh:\
:lp=/dev/lp0:

testo:\
:sd=/var/spool/lpd/testo:\
:mx#0:\
:sh:\
:lp=/dev/lp0:\
:if=/var/spool/lpd/testo/filtro:
```

Supponendo che la trasformazione del testo avvenga tramite il programma `'unix2dos'`, il filtro `'/var/spool/lpd/testo/filtro'` potrebbe essere realizzato nel modo seguente:

```
#!/bin/sh
/usr/bin/unix2dos | lpr -Plp
```

#### 27.5.2.5 Filtri sofisticati

Non è necessario complicarsi troppo la vita. Spesso la distribuzione GNU/Linux che si ha a disposizione è già predisposta in modo da facilitare la creazione di filtri di stampa.

Anche quando non si è così «fortunati», esiste sempre un'alternativa migliore allo scrivere il proprio filtro (salvo casi particolari). Un esempio è Apsfilter<sup>6</sup> che senza molta fatica genera da solo il file `'/etc/printcap'`, le directory per le code di stampa e i filtri necessari; un altro è Magicfilter, più semplice, ma efficace.

Infine, è il caso di ricordare il pacchetto PSUtils<sup>7</sup> che è composto da una serie di programmi di servizio in grado di rielaborare file PostScript, cosa utile per esempio quando su un solo foglio si vogliono stampare più pagine ridotte.

### 27.5.3 Magicfilter

Magicfilter<sup>8</sup> è un sistema di filtri per la stampa organizzato in modo semplice ed efficace. Si tratta di un programma, precisamente l'eseguibile `'magicfilter'`, in grado di individuare il tipo di file che gli viene fornito attraverso lo standard input e di conseguenza di elaborarlo nel modo migliore ai fini della stampa. Per ottenere questo risultato, è necessaria la preparazione di un file di configurazione, con il quale si indicano le impronte di riconoscimento dei file, ovvero il magic number, e le azioni da compiere a seconda del tipo di file individuato. Questo comportamento spiega la ragione del nome: un filtro di stampa abbinato all'individuazione del magic number.

L'idea più importante di Magicfilter sta nel fatto che i suoi file di configurazione, distinti in base al tipo di stampante per i quali devono essere utilizzati, sono degli script per Magicfilter. Questo fatto semplifica tante cose, soprattutto nella configurazione del file `'/etc/printcap'`.

#### 27.5.3.1 Configurazione di Magicfilter

Il file di configurazione tipico di Magicfilter inizia generalmente con la dichiarazione del suo interprete, essendo in pratica uno script dell'eseguibile `'magicfilter'`:

```
#!/usr/sbin/magicfilter
...
```

In questo file, il simbolo `'#'` serve a indicare l'inizio di un commento, fino alla fine della riga; le righe bianche e quelle vuote vengono ignorate. Le altre righe, sono direttive, secondo la sintassi seguente:

```
scostamento stringa_di_riconoscimento operazione_da_compiere
```

In pratica, si tratta di campi separati da uno o più spazi: il primo è un numero che esprime lo scostamento in byte dall'inizio del file, per individuare il punto a partire dal quale si deve iniziare il confronto con la stringa indicata nel secondo campo (quello che sarebbe il magic number); il terzo campo è la descrizione delle azioni da compiere nel caso in cui il file corrisponda alla stringa di riconoscimento.

Il numero che indica lo scostamento è espresso normalmente in base decimale; può essere usata una notazione ottale, se la prima cifra è uno zero; si può utilizzare anche una notazione esadecimale che deve essere preceduta dal prefisso `0x...` Il valore zero corrisponde all'inizio del file, qualunque altro valore (positivo) rappresenta un numero equivalente di byte da saltare prima di iniziare il confronto con la stringa di riconoscimento.

La stringa di riconoscimento è una stringa normale, che può contenere delle sequenze di escape secondo la convenzione del linguaggio C, oltre a due aggiunte: `'\?'` che rappresenta un carattere qualunque e `'\''` seguita da uno spazio che rappresenta uno spazio letterale, allo scopo di non interrompere il campo. Eventualmente, questa stringa può anche essere racchiusa tra apici doppi; in tal caso, non c'è bisogno di proteggere lo spazio con la barra obliqua inversa.

Una direttiva di questo file di configurazione può essere continuata sulla riga successiva, ponendo alla fine della riga da continuare il simbolo `'\''`.

L'ultima parte di queste direttive è più complessa da descrivere, in quanto si compone di una parola chiave iniziale, a cui possono seguire altre indicazioni che variano in base alla parola chiave stessa.

Questo file viene scandito dal suo interprete, `'magicfilter'`, dall'inizio alla fine; la scansione termina nel momento in cui una direttiva corrisponde al file, ovvero, quando i primi due campi sono tali da determinare la corrispondenza. In questo senso, le combinazioni più dettagliate devono avere la precedenza rispetto a quelle più generiche. Inoltre, esiste una variante alla sintassi di queste direttive, costituita dalla forma seguente:

```
default operazione_da_compiere
```

Questa direttiva va posta alla fine del file di configurazione, per indicare cosa fare con i file che non sono stati riconosciuti diversamente e di solito viene usata proprio per gestire i file di testo.

Quando l'operazione da compiere prevede l'avvio di un programma o di uno script, vengono rese disponibili alcune variabili di ambiente, che possono essere indicate anche nell'ambito degli argomenti di questo comando. Le variabili disponibili effettivamente dipendono dalla quantità di informazioni a cui Magicfilter può accedere, cosa che dipende a sua volta dalle caratteristiche del demone di stampa; tuttavia, sono disponibili sempre la variabile *LPUSER*, che contiene il nome dell'utente proprietario del processo di stampa, e *LPHOST*, con il nome dell'elaboratore da cui ha avuto origine la richiesta di stampa.

Tabella 27.49. Alcune operazioni.

Opzione	Descrizione
cat [ <i>prefisso</i> [ <i>suffisso</i> ]]	Questa definizione serve a ottenere la copia dei dati senza conversioni, aggiungendo eventualmente una stringa come prefisso e una come suffisso, prima e dopo il file. Le stringhe in questione si possono realizzare nello stesso modo in cui si realizza la stringa di riconoscimento, con l'eccezione della sequenza '\?', che in questo contesto non ha significato.
ignore reject <i>messaggio</i>	Queste due definizioni alternative, fanno in modo di non restituire dati in uscita; in particolare, 'reject' tenta di inviare un messaggio all'utente a cui appartiene il processo di stampa, attraverso la posta elettronica.
filter <i>comando</i>	Il comando indicato viene eseguito passando il flusso di dati che interessa il filtro e ci si attende che restituisca i dati trasformati nel modo adatto per la stampa. Questo comando può accedere alle variabili di ambiente create da Magicfilter e queste gli possono essere fornite anche tra le opzioni, usando la notazione '\$variabile'.
pipe <i>comando</i>	Con la parola chiave 'pipe' si ottiene una cosa simile a quella che si ha con 'filter', con la differenza che il risultato emesso dal comando, viene rielaborato dal filtro, come passaggio ulteriore. Questo meccanismo, se non viene usato nel modo corretto, potrebbe creare un ciclo infinito. Bisogna essere certi che prima o poi, i dati reimmessi nel filtro, trovino a un certo punto l'uscita corretta.
fpipe <i>comando</i> ffilter <i>comando</i>	Si tratta dell'unione di 'filter' e 'pipe', con la differenza che il file proveniente dallo standard input viene prima salvato in un file temporaneo, il cui nome è disponibile attraverso la variabile di ambiente <i>FILE</i> . Questo permette di utilizzare un comando che accede a un file su disco, presumibilmente perché ha bisogno di accedervi in modo non sequenziale. L'elaborazione che si ottiene viene reimpressa nel filtro per un passaggio successivo.

Segue la descrizione di alcuni esempi.

```
0 %! filter \
  /usr/bin/gs -q -dSAFER -dNOPAUSE -r300 \
  -sDEVICE=ljet4 -sOutputFile=- -
```

Questa direttiva serve a individuare i file PostScript, in quanto questi dovrebbero iniziare con la stringa '%!'. A questo tipo di file viene abbinata l'elaborazione da parte di 'gs' (Ghostscript), con le opzioni opportune per ottenere un risultato adatto a

una stampante di tipo HP Laserjet 4, emesso attraverso lo standard output.

```
0 \037\235 pipe /bin/gzip -cdq
0 MM\0\x2a fpipe /usr/bin/tiff2ps $FILE
0 II\x2a\0 fpipe /usr/bin/tiff2ps $FILE
```

In questo caso, sembra trattarsi di un file compresso con 'gzip', pertanto viene estratto e rinviato al filtro, in modo da rianalizzare il contenuto prima di inviarlo alla stampa.

Nel caso il file sembri un formato TIFF, viene utilizzato il programma 'tiff2ps' per convertire l'immagine in un file PostScript, che poi viene reimpresso nel filtro, in modo che questo file possa raggiungere la direttiva corretta per la sua utilizzazione finale. Si osservi l'uso della variabile di ambiente *FILE*, per fornire al programma 'tiff2ps' il nome del file temporaneo che viene generato in questo caso.

```
0 \177ELF reject \
  Tentativo di stampare un binario ELF.
```

Sembra trattarsi di un binario ELF che ovviamente non può essere stampato.

```
default cat
```

Questo è l'esempio più semplice di una direttiva finale che serve a definire cosa fare con i file di testo. In questo caso, i dati vengono lasciati tali e quali, mentre in un'altra situazione ci si potrebbe accettare di convertire il codice di interruzione di riga in modo che corrisponda alla sequenza <CR><LF>.

Magicfilter non prevede degli script pronti per l'uso di HPIJS, ma conoscendo le opzioni necessarie, è possibile modificarne uno di quelli già esistenti. L'esempio seguente mostra una porzione di una copia dello script 'laserjet-filter', modificata in modo da usare HPIJS. Si osservi che le righe sono suddivise in più punti, per motivi tipografici:

```
...
# PostScript
0 %! filter /usr/bin/gs -q -dSAFER -dNOPAUSE -r300 ↵
↵ -sDEVICE=ijs -sIjsServer=hpijs -dIjsUseOutputFD ↵
↵ -sDeviceManufacturer="HEWLETT-PACKARD" ↵
↵ -sDeviceModel="HP LaserJet" ↵
↵ -sOutputFile=/usr/share/magicfilter/quiet.ps -c quit
0 \004! filter /usr/bin/gs -q -dSAFER -dNOPAUSE -r300 ↵
↵ -sDEVICE=ijs -sIjsServer=hpijs -dIjsUseOutputFD ↵
↵ -sDeviceManufacturer="HEWLETT-PACKARD" ↵
↵ -sDeviceModel="HP LaserJet" ↵
↵ -sOutputFile=/usr/share/magicfilter/quiet.ps -c quit
...

# Wind/U postscript driver. This one creates broken postscript, so we have to
# put in a special case to convert it to PCL first
65 %%Creator: Wind/U filter /usr/bin/gs -q -dSAFER -dNOPAUSE -r300 ↵
↵ -sDEVICE=ijs -sIjsServer=hpijs -dIjsUseOutputFD ↵
↵ -sDeviceManufacturer="HEWLETT-PACKARD" ↵
↵ -sDeviceModel="HP LaserJet" ↵
↵ -sOutputFile=/usr/share/magicfilter/quiet.ps -c quit
...
```

### 27.5.3.2 Funzionamento e utilizzazione pratica di Magicfilter

L'eseguibile 'magicfilter' legge il file di configurazione che gli viene fornito come primo argomento nella riga di comando, comportandosi di conseguenza:

```
magicfilter file_di_configurazione [opzioni]
```

È necessario ricordare che 'magicfilter' non viene avviato dall'utente, ma dal demone di stampa; pertanto, le opzioni sono quelle che passa lo stesso demone e 'magicfilter' deve essere in grado di interpretarle. Dal momento che il demone di stampa non fornisce alcuna informazione sul file di configurazione, per fare in modo che questo sia indicato, si trasformano i file di configurazione in script, come è già stato mostrato, utilizzando tali script come se fossero i veri filtri di stampa. In effetti, in questo modo, si ottiene proprio di avviare 'magicfilter' con il nome dello script come primo argomento e le altre opzioni subito dopo, esattamente come si vede nello schema sintattico.

Magicfilter è stato realizzato allo scopo di essere utilizzato come filtro di ingresso ('if') ed è in grado di interpretare solo le opzioni

che vengono passate in questa situazione dal demone di stampa. A titolo informativo, la tabella 27.56 elenca le opzioni principali che l'eseguibile **magicfilter** è in grado di interpretare.

Tabella 27.56. Opzioni standard che vengono interpretate da Magicfilter, in quanto filtro di ingresso di un sistema di stampa.

Opzione	Descrizione
-c	Lascia i dati inalterati.
-nutente	Nominativo utente, disponibile attraverso la variabile <b>LPUSER</b> .
-hnode	Elaboratore di origine, disponibile attraverso la variabile <b>LPHOST</b> .
-irientro	Rientro, disponibile attraverso la variabile <b>LPINDENT</b> .

Magicfilter si compone già di un buon numero di file di configurazione, ovvero di script, realizzati per altrettanti tipi di stampanti differenti. Di solito è sufficiente scegliere quello adatto, salvo la possibilità di provare tutti quelli simili in modo da poter scegliere il migliore in base al risultato preferito. Dal momento che, bene o male, si tratta di file di configurazione, questi script dovrebbero essere collocati nella directory `/etc/magicfilter/`. Quello che segue è l'esempio di un file `/etc/printcap` predisposto per gestire una stampante compatibile con il tipo HP Laserjet normale. Per la precisione, il file `/etc/magicfilter/laserjet-filter` è uno di questi script di configurazione.

```
lp|Stampante predefinita
:lp=/dev/lp0:\
:sd=/var/spool/lpd/lp:\
:if=/etc/magicfilter/laserjet-filter:\
:mx#0:\
:sh:
```

In condizioni normali, una coda di stampa organizzata in questo modo va bene per qualunque file da stampare. Eventualmente, in caso di bisogno, si può modificare leggermente qualche direttiva del file di configurazione scelto, magari dopo averne fatta una copia.

#### 27.5.4 PostScript printer description

Le stampanti PostScript consentono ai programmi di produrre un unico formato per la stampa, indipendentemente dalle caratteristiche fisiche della stampante; tuttavia, rimangono delle caratteristiche spicchiole da considerare, come il formato della carta disponibile, la possibilità di selezionare nella stampante un vassoio di carta invece di un altro, assieme ad altri dettagli. Dal momento che le stampanti hanno quindi delle caratteristiche che vanno al di là del linguaggio di stampa, il software che si occupa di gestire la stampante, ha comunque la necessità di avere delle notizie sulla stessa, anche per consentire all'utente di selezionare le funzioni che gli interessano.

A fianco del linguaggio PostScript è stato sviluppato quindi un file di configurazione definito PPD, ovvero *PostScript printer description*, con il quale i produttori di stampanti PostScript possono dichiarare le caratteristiche particolari della propria stampante, in modo che il software di gestione della stampa le acquisisca senza complicazioni.

Nei sistemi Unix, il sistema dei file PPD per la definizione delle caratteristiche delle stampanti, viene esteso anche a quelle che non utilizzano il linguaggio PostScript direttamente, attraverso delle estensioni molto semplici. Segue un estratto del contenuto di un file PPD, prodotto dalla stessa casa costruttrice di una stampante PostScript, preparato per un sistema MS-Windows:

```
*PPD-Adobe: "4.3"
*% =====
*% Printer Description File
*% Copyright 1992-97 Hewlett-Packard Company
*%   October 7, 1997
*% =====
*% PPD for HP LaserJet 4000 Series
*% For Windows NT 3.51
*% =====
*% === PPD File Version Information ===
*FileVersion:   "1.1"
```

```
*FormatVersion:   "4.3"
*LanguageEncoding: ISOLatin1
*LanguageVersion:  English
*PCFileName:      "HP4000_5.PPD"

*% === Product Version Information ===
*ModelName:       "HP LaserJet 4000 Series"
*ShortNickName:   "HP LaserJet 4000 Series PS"
*NickName:        "HP LaserJet 4000 Series PS"
*Product:         "(HP LaserJet 4000 Series)"
*Manufacturer:    "HP"
*PSVersion:       "(2014.108) 1"

*%=== Device Capabilities =====
*ColorDevice:     False
*DefaultColorSpace: Gray
*FileSystem:      True
*?FileSystem:     "
  save
  false
  (%disk?)
  { currentdevparams dup /Writeable known
    { /Writeable get {pop true} if } { pop } ifelse
  } 100 string /IODevice resourceforall
  {(True)}{(False)} ifelse = flush
  restore
"
*End

*LanguageLevel:  "2"
*Throughput:     "17"
*TRasterizer:    Type42
*?TRasterizer:   "
  save
  42 /FontType resourcestatus
  { pop pop (Type42)} {pop pop (None)} ifelse = flush
  restore
"
*End
...
...
*Reset: "
  count 0 eq { false } { true exch startjob } ifelse
  not {
    (WARNING: Cannot reset printer.) =
    (Missing or invalid password.) =
    (Please contact the author of this software.) = flush quit
  } if
  systemdict /quit get exec
  (WARNING : Printer Reset Failed.) = flush
"
*End

*% =====
*% For "HP LaserJet 4000 Series" version 2014.108
*% =====
```

Come accennato, nei sistemi Unix i file PPD vengono usati utilmente anche per stampanti che non riconoscono il linguaggio PostScript direttamente, aggiungendo delle estensioni con cui si specifica in che modo usare Ghostscript. Segue un estratto di un file PPD utilizzato da Foomatic-RIP, dove viene evidenziato proprio il riferimento a Ghostscript:

```
*PPD-Adobe: "4.3"
*%
*% For information on using this, and to obtain the required
*% backend script, consult http://www.linuxprinting.org/
*%
*FormatVersion:  "4.3"
*FileVersion:    "1.1"
*LanguageVersion: English
*LanguageEncoding: ISOLatin1
*PCFileName:     "LASERJET.PPD"
*Manufacturer:   "HP"
*Product:        "(LaserJet)"
*cupsVersion:    1.0
*cupsManualCopies: True
*cupsModelNumber: 2
*cupsFilter:     "application/vnd.cups-postscript 0 ↵
↳foomatic-rip"
*%pprRIP:       foomatic-rip other
```

```

*ModelName: "HP LaserJet"
*ShortNickName: "HP LaserJet laserjet"
*NickName: "HP LaserJet Foomatic/laserjet ←
↳(recommended)"
*PSVersion: "(3010.000) 550"
*PSVersion: "(3010.000) 651"
*PSVersion: "(3010.000) 652"
*PSVersion: "(3010.000) 653"
*PSVersion: "(3010.000) 704"
*PSVersion: "(3010.000) 705"
*PSVersion: "(3010.000) 800"
*LanguageLevel: "3"
*ColorDevice: False
*DefaultColorSpace: Gray
*FileSystem: False
*Throughput: "1"
*LandscapeOrientation: Plus90
*TRasterizer: Type42

*VariablePaperSize: False

*FoomaticIDs: HP-LaserJet laserjet
*FoomaticRIPCommandLine: "gs -q -dPATCH -dPARANOIDSAFER ←
↳-dQUIET -dNOPAUSE -sDEVICE=laserjet%A%Z -sOutputFile=- - ←
↳| perl -p -0033 ←
↳-e &quot; s/^&amp;l\d+[aA]/$&E/; &quot; "
*End

*OpenGroup: General/General
...

```

### 27.5.5 Foomatic-RIP

Foomatic-RIP<sup>910</sup> è una parte del sistema Foomatic per la gestione della stampa. Foomatic-RIP è un filtro universale, in quanto adattabile a qualsiasi sistema di stampa, che si avvale di file PPD per acquisire le caratteristiche delle stampanti.

Foomatic-RIP viene usato generalmente all'interno del sistema di stampa, attraverso la configurazione del file `/etc/printcap`; tuttavia, la sintassi di utilizzo può variare leggermente da un sistema all'altro. Segue un esempio di configurazione abbastanza generalizzato di tale file:

```

lp:\
:lp=/dev/lp0:\
:sd=/var/spool/lpd/lp:\
:af=/var/log/lp-acct:\
:lf=/var/log/lp-errs:\
:if=/usr/bin/foomatic-rip:\
:ppd=/usr/share/ppd/hp-laserjet.ppd:\
:pl#66:\
:pw#80:\
:pc#150:\
:mx#0:\
:mc#999:\
:sh:

```

Come si può intuire leggendo il contenuto, il tipo di stampante viene indicato facendo riferimento a un file PPD, `/usr/share/ppd/hp-laserjet.ppd`, che comunque deve essere stato realizzato specificatamente per Foomatic-RIP, con le estensioni a lui necessarie.

Attraverso il file PPD, può essere specificato anche se per la stampa ci si deve avvalere di programmi come HPIJS, con tutte le opzioni che eventualmente fossero necessarie.

### 27.5.6 Controllo dell'impostazioni della carta

Nel momento in cui la stampa è gestita attraverso un sistema di filtri, per come è stato mostrato in questo capitolo, i programmi non hanno la possibilità di definire il formato della carta. Infatti, non hanno alcun modo di colloquiare con il sistema di stampa sottostante; al massimo possono scegliere la coda di stampa.

In questo senso, se si dispone di una stampante con la quale possono essere utilizzati diversi formati di carta, occorre definire altrettante

code di stampa differenti: ognuna predisposta per un formato diverso. In questo modo, il programma che ha bisogno di un certo formato, deve inviare la richiesta di stampa utilizzando la coda adatta per questo.

Nella sezione 26.2 viene affrontato anche il problema dell'allineamento delle stampanti, dal momento che alle volte queste introducono dei margini che sfasano la stampa, cosa che crea problemi specialmente nel momento della rilegatura di un testo.

## 27.6 Uniformità del sistema di stampa: da testo a PostScript

Un sistema di filtri di stampa ben organizzato deve passare per la generazione di un formato intermedio (prima di quello finale adatto alla stampante) per poter gestire l'impostazione della stampa in modo completamente trasparente. La figura 27.61 mostra questa idea.

Figura 27.61. Stampa attraverso un formato intermedio uniforme.



L'esistenza di Ghostscript, descritto meglio nella sezione 26.1, giustifica la scelta del formato PostScript come standard per il formato intermedio di stampa, benché questo formato sia proprietario. L'utilità di questo passaggio obbligato sta anche nel fatto che prima della conversione finale, il formato PostScript può essere rimaneggiato, per adattarlo a esigenze particolari, compresa la riduzione o l'ingrandimento. Tuttavia, in questa situazione, anche i file di testo vanno rielaborati in modo da generare prima un file PostScript. A questo scopo sono stati realizzati programmi come A2ps.

### 27.6.1 A2ps

A2ps<sup>11</sup> è un programma per generare file PostScript a partire, prevalentemente, da file di testo. Gli obiettivi di chi sviluppa A2ps vanno oltre tale livello; tuttavia, questa è la sua funzionalità più importante.

A2ps è controllato da un file di configurazione generale, `/etc/a2ps.cfg`, al quale può essere affiancato un file personale, `~/a2ps/a2psrc`, e anche uno locale (nella directory corrente), `.a2psrc`. Pur senza approfondire la configurazione di A2ps, vale la pena di descrivere brevemente come si compongono questi file. Il simbolo `#` rappresenta l'inizio di un commento che termina alla fine della riga; le righe bianche e quelle vuote vengono ignorate; le altre righe sono direttive nella forma:

```
tipo_dichiarazione : dichiarazione
```

Il file di configurazione generale che di solito viene fornito assieme al programma (`/etc/a2ps.cfg`) dovrebbe essere adatto alla maggior parte delle situazioni e in generale non serve altro per utilizzare A2ps. In ogni caso, questo file è commentato molto bene e la documentazione che fornisce A2ps è molto dettagliata (*info a2ps*).

A2ps si utilizza in pratica attraverso l'eseguibile `a2ps`, il cui scopo è quello di ricevere uno o più file in modo da poter generare una trasformazione adeguata in formato PostScript:

```
a2ps [opzioni] [file...]
```

I file possono essere indicati attraverso la riga di comando e in mancanza della loro indicazione viene usato lo standard input. Lo scopo di A2ps è quello di generare un file PostScript, secondo quanto

definito nella configurazione, oppure dalle opzioni della riga di comando. In generale, il risultato viene inviato alla stampa attraverso il comando `lpr`, come si può vedere dalla direttiva seguente, che appare generalmente nel file di configurazione globale, con la quale si dichiara l'invio dello standard output verso tale comando di stampa:

```
# First, the default printer (option -d, no argument)
DefaultPrinter: | lpr
```

Di solito non c'è ragione di cambiare questo comportamento di A2ps, ma è importante sapere che non è sempre necessariamente così.

Il fatto che A2ps sia configurato in questo modo, lo rende simile a un comando di stampa alternativo a quello normale, per cui, il comando

```
$ lpr pippo [Invio]
```

viene sostituito direttamente dal comando

```
$ a2ps pippo [Invio]
```

che in più si occupa di impaginare meglio il testo.

Ovviamente, tutto questo presuppone che la coda di stampa predefinita, ovvero quella che viene utilizzata quando si usa il comando `lpr` senza specificare altro, sia in grado di gestire file PostScript.

A2ps offre molte possibilità nel modo di impaginare il testo e non si limita semplicemente a consentire la stampa ridotta di più pagine virtuali su una facciata singola. È molto importante anche la sua capacità di evidenziare il testo in funzione del suo contenuto, cosa che diventa molto utile per la lettura dei sorgenti di un programma. La tabella 27.64 riassume brevemente alcune opzioni più importanti che possono essere usate nella riga di comando dell'eseguibile `a2ps`, opzioni che possono essere anche incluse nella configurazione attraverso direttive nella forma:

```
Options: opzione_della_riga_di_comando
```

Per esempio, per selezionare il formato di carta A4 si può utilizzare l'opzione `--medium=A4` nella riga di comando, oppure la direttiva seguente nel file di configurazione:

```
Options: --medium=A4
```

Tabella 27.64. Alcune opzioni di A2ps.

Opzione	Alternativa	Descrizione
<code>-M carta</code>	<code>--medium=carta</code>	Dimensione della carta.
<code>-r</code>	<code>--landscape</code>	Orientamento orizzontale.
<code>-R</code>	<code>--portrait</code>	Orientamento verticale.
<code>-n</code>		<i>n</i> pagine virtuali per ogni pagina reale.
<code>-j</code>	<code>--borders={yes no}</code>	Bordi attorno alle pagine virtuali.
<code>-A</code>	<code>--compact={yes no}</code>	Unione di più file in un solo foglio.
<code>-f n</code>	<code>--font-size=n</code>	<i>n</i> punti per la dimensione dei caratteri.
<code>-l n</code>	<code>--chars-per-line=n</code>	<i>n</i> caratteri per riga.
<code>-L n</code>	<code>--lines-per-page=n</code>	<i>n</i> righe per pagina virtuale.
<code>-m</code>	<code>--catman</code>	Pagina di manuale; come <code>'-L66'</code> .
<code>-T n</code>	<code>--tabsize=n</code>	Dimensione degli stop di tabulazione.
<code>-B</code>	<code>--no-header</code>	Nessuna intestazione.
<code>-a m [-n]</code>	<code>--pages=m [-n]</code>	Seleziona un intervallo di pagine.
<code>-o file</code>	<code>--output=file</code>	Crea un file e non invia alla stampa.
<code>-P coda</code>	<code>--printer=coda</code>	Coda di stampa a cui inviare il risultato.

Opzione	Alternativa	Descrizione
<code>-d</code>		Invia il risultato alla coda di stampa predefinita.
<code>-Etipo</code>	<code>--pretty-print=tipo</code>	Definizione dello stile di evidenziazione del testo (tabella 27.65).
<code>-X codifica</code>	<code>--encoding=codifica</code>	Definizione della codifica in cui è scritto il testo (tabella 27.66).

La tabella 27.65 riporta invece l'elenco di alcuni nomi di stili di evidenziazione della stampa, in funzione del contenuto del file di testo che si intende stampare. Questi nomi si usano con l'opzione `'-E'`.

Tabella 27.65. Alcuni nomi che definiscono uno stile nel modo di evidenziare il testo.

Tipo	Descrizione
sh	Script di una shell Bourne.
html	Sorgente HTML.
mail	Messaggio di posta elettronica.
udiff	File di differenze in formato unificato.
make	File-make.
ada	Sorgente in linguaggio ADA.
c	Sorgente in linguaggio C.
cpp	Sorgente in linguaggio C++.
gnuc	Sorgente in linguaggio GNU C.
clisp	Sorgente in linguaggio Common Lisp.
eiffel	Sorgente in linguaggio Eiffel.
elisp	Sorgente in linguaggio Emacs Lisp.
fortran	Sorgente in linguaggio Fortran.
java	Sorgente in linguaggio Java.
pascal	Sorgente in linguaggio Pascal.
python	Sorgente in linguaggio Python.
scheme	Sorgente in linguaggio Scheme.
sql92	Sorgente in linguaggio SQL92.

La tabella 27.66 riporta l'elenco delle sigle che si possono utilizzare con l'opzione `'-X'` per definire la codifica con cui è scritto il testo da convertire. Si osservi che è probabile non sia disponibile una codifica UTF-8.

Tabella 27.66. Alcuni nomi che identificano la codifica del testo.

Nome	Descrizione
ASCII	ASCII normale.
IBM-CP437	IBM CP437.
IBM-CP850	IBM CP850.
ISO-8859-1	ISO 8859-1 (Latin 1).
ISO-8859-2	ISO 8859-2 (Latin 2).
ISO-8859-10	ISO 8859-10.
ISO-8859-15	ISO 8859-15 (Latin 9).
ISO-8859-3	ISO 8859-3.



Nome	Descrizione
ISO-8859-4	ISO 8859-4.
ISO-8859-5	ISO 8859-5.
ISO-8859-7	ISO 8859-7.
ISO-8859-9	ISO 8859-9.
MS-CP1250	Microsoft CP1250.

Prima di passare all'elenco di esempi comuni, vale la pena di proporre il comando che potrebbe essere usato in un filtro di stampa per gestire i file di testo, senza lasciare che questi vengano inviati direttamente alla stampante:

```
a2ps -l -M A4 -f l1 --borders=no -B -o -
```

In questo modo si specifica che: si vuole ottenere una pagina virtuale per ogni pagina reale; il formato della carta è A4; il testo deve utilizzare un carattere da 11 punti (è la dimensione ottimale per stampare 80 colonne); non si vogliono bordi attorno alla pagina virtuale; non si vuole alcuna intestazione; il risultato in PostScript deve essere emesso attraverso lo standard output.

Volendo intervenire nella configurazione di Magicfilter, si potrebbe sostituire la solita direttiva:

```
default cat
```

La nuova direttiva potrebbe essere come quella seguente:

```
default pipe /usr/bin/a2ps -l -M A4 -f l1 --borders=no -B -o - 2> /dev/null
```

Segue la descrizione di alcuni esempi. Questi esempi fanno riferimento alla configurazione tipica di A2ps.

- `$ a2ps pippo [Invio]`

Mette in stampa il file 'pippo', dopo averlo convertito in PostScript. In condizioni normali, si ottiene la stampa di due pagine virtuali per foglio reale, con un orientamento orizzontale.

- `$ a2ps -2 pippo [Invio]`

Probabilmente, si tratta della stessa cosa dell'esempio precedente, con l'indicazione esplicita della richiesta di stampare due pagine virtuali per foglio.

- `$ a2ps -2 -P laser pippo [Invio]`

Come nell'esempio precedente, indicando espressamente la scelta della coda di stampa denominata 'laser'.

- `$ a2ps -2 -o pippo.ps pippo [Invio]`

Come nell'esempio precedente, ma senza stampare: viene generato il file 'pippo.ps'.

- `$ a2ps -2 -R pippo [Invio]`

Invia alla stampa il solito file, dopo averlo convertito in PostScript, in modo tale da ottenere due pagine virtuali, ma con un orientamento verticale. In questo modo, il testo è più piccolo e ogni pagina virtuale contiene un numero molto elevato di righe.

- `$ man 1 a2ps | a2ps -2 -m [Invio]`

Stampa la pagina di manuale *a2ps(1)*, impaginandola nel modo migliore per questo tipo di informazioni.

- `$ a2ps -2 -Esh /etc/profile [Invio]`

Stampa il file '/etc/profile', utilizzando un sistema di evidenziazione ottimale per gli script di shell Bourne o derivate.

```
#!/bin/sh
a2ps --portrait --columns=1 --margin=30 --no-header --borders=no \
--lines-per-page=66 --chars-per-line=80 --font-size=10 \
--interpret=yes --output=-
```

Lo script permette di generare un file PostScript attraverso lo standard output a partire da un file di testo normale, fornito attraverso lo standard input. Per la precisione, si vuole ottenere una

sola colonna verticale, con un margine simmetrico (nel caso di carta A4), senza scritte o bordi aggiuntivi, garantendo 80 colonne e 66 righe per pagina.

```
#!/bin/sh
a2ps --portrait --columns=1 --margin=30 --no-header --borders=no \
--chars-per-line=80 --font-size=10 \
--interpret=yes --output=-
```

Come nell'esempio precedente, ma senza porre un limite alla quantità di righe per pagina, che così possono essere anche di più delle 66 standard.

```
#!/usr/sbin/magicfilter
...
...
# Default entry -- for normal (text) files. MUST BE LAST.
#default cat \eE\ek2G\e(ON \eE
default pipe a2ps --portrait --columns=1 --margin=30 \
--no-header --borders=no \
--chars-per-line=80 --font-size=10 \
--interpret=yes --output=- \
2> /dev/null
```

L'esempio mostra l'ultima istruzione di uno script di Magicfilter, allo scopo di stampare automaticamente del testo normale aggiungendo un po' di margini, come già mostrato nell'esempio precedente.

```
#!/bin/sh
/usr/bin/a2ps --medium=A4 -l --lines-per-page=66 \
--chars-per-line=136 --tabsize=8 \
--encoding=ISO-8859-1 \
--non-printable-format=space \
--no-header --landscape --debug
```

Si tratta di un altro script, realizzato in modo da gestire file di testo fino a 136 colonne, in orizzontale. Non tutte le opzioni sono indispensabili.

## 27.6.2 Enscript

Enscript<sup>12</sup> è un programma per generare file PostScript e altri formati, a partire da file di testo.

Enscript prevede una serie di file di configurazione, '~/.enscriptrc', '/etc/enscriptsitesite.cfg' e '/etc/enscript.cfg', dove il primo è quello personale di ogni utente, il secondo è un primo file di configurazione generale, mentre il terzo è un file di configurazione di sistema. L'ordine in cui sono indicati questi file, rappresenta anche l'ordine di precedenza che hanno le direttive al loro interno.

Senza approfondire la configurazione di Enscript, vale la pena di descrivere brevemente come si compongono questi file. Il simbolo '#' rappresenta l'inizio di un commento che termina alla fine della riga; le righe bianche e quelle vuote vengono ignorate; le altre righe sono direttive nella forma:

```
tipo_dichiarazione : dichiarazione
```

Il file di configurazione generale che di solito viene fornito assieme al programma ('/etc/enscript.cfg') dovrebbe essere adatto alla maggior parte delle situazioni. In ogni caso, questo file è commentato molto bene e la documentazione disponibile è molto dettagliata (*enscript(1)*).

Enscript si utilizza in pratica attraverso l'eseguibile 'enscript', il cui scopo normale è quello di comporre in formato PostScript uno o più file indicati come argomento:

```
enscript [opzioni] [file...]
```

I file possono essere indicati attraverso la riga di comando, ma in mancanza di questa indicazione viene usato lo standard input. Lo scopo di Enscript è quello di generare un file PostScript, o un altro tipo di formato finale, ma il modo in cui questo file viene restituito dipende dalla configurazione, oppure dalle opzioni della riga di co-

mando. In generale, il risultato viene inviato alla stampa attraverso il comando `lpr`, in base alla direttiva seguente:

```
# Where output goes as a default: 'printer' or 'stdout'
DefaultOutputMethod: printer
```

Di solito non c'è motivo di cambiare questo comportamento di Enscript, ma è importante sapere che non è sempre necessariamente così. In tal modo, Enscript può essere usato come se fosse un comando di stampa, dove uno qualunque dei due comandi seguenti, si traduce direttamente nella stampa del file `pippo`:

```
$ enscript pippo [Invio]
```

```
$ cat pippo | enscript [Invio]
```

La configurazione di Enscript prevede anche l'indicazione esplicita del comando di stampa e della coda di stampa a cui inviare il file generato dalla composizione. Generalmente non è necessario preoccuparsi di questo fatto, dal momento che un sistema di stampa configurato correttamente è in grado di gestire tutto attraverso la coda predefinita. Tuttavia è bene tenere in considerazione le direttive necessarie a tale definizione:

```
# Printer name to spool to. As a default we want to use
# system's default printer.
# Printer: ps

# The spooler command switch to select the printer queue.
# This option can also be used to pass other flags to the
# spooler command but they must be given before the queue
# switch.
QueueParam: -P

# Printer spooler command name.
Spooler: lpr
```

Enscript offre molte possibilità nel modo di impaginare il testo, senza limitarsi a consentire la stampa ridotta di più pagine virtuali su una facciata singola. È molto importante anche la sua capacità di evidenziare il testo in funzione del suo contenuto, per facilitare la lettura dei sorgenti di un programma. La tabella 27.76 riepiloga brevemente alcune opzioni più importanti che possono essere usate nella riga di comando dell'eseguibile `enscript`, opzioni che possono essere anche incluse automaticamente attraverso la variabile di ambiente `ENSCRIPT`, nello stesso modo in cui apparirebbero nella riga di comando.

Tabella 27.76. Alcune opzioni di Enscript.

Opzione	Alternativa	Descrizione
<code>-M carta</code>	<code>--media=carta</code>	Dimensione della carta.
	<code>--list-media</code>	Elenca i formati disponibili della carta.
<code>-r</code>	<code>--landscape</code>	Orientamento orizzontale.
<code>-R</code>	<code>--portrait</code>	Orientamento verticale.
<code>-n</code>	<code>--columns=n</code>	Testo su <i>n</i> colonne.
<code>-U n</code>	<code>--nup=n</code>	<i>n</i> pagine virtuali.
<code>-j</code>	<code>--borders</code>	Bordi attorno alle pagine virtuali.
<code>-f nome@n</code>	<code>--font=nome@n</code>	Nome e dimensione del carattere da stampa.
<code>-f nome@m/n</code>	<code>--font=nome@m/n</code>	Nome, larghezza e altezza del carattere.
	<code>--margins=sx:dx:sopra:sotto</code>	Margini in punti.
<code>-l</code>	<code>--lineprinter</code>	66 righe per pagina senza intestazione.
<code>-L n</code>	<code>--lines-per-page=n</code>	<i>n</i> righe per pagina virtuale.
<code>-T n</code>	<code>--tabsize=n</code>	Dimensione degli stop di tabulazione.

Opzione	Alternativa	Descrizione
<code>-B</code>	<code>--no-header</code>	Nessuna intestazione.
<code>-a m[-n]</code>	<code>--pages=m[-n]</code>	Seleziona un intervallo di pagine.
<code>-a odd even</code>	<code>--pages=odd even</code>	Seleziona pagine dispari o pagine pari.
<code>-o file</code> <code>-p file,</code>	<code>--output=file</code>	Crea un file e non invia alla stampa.
<code>-P coda</code> <code>-d coda</code>	<code>--printer=coda</code>	Coda di stampa a cui inviare il risultato.
<code>-q</code>	<code>--quiet</code> <code>--silent</code>	Funzionamento silenzioso.
<code>-Etipo</code>	<code>--pretty-print=tipo</code>	Stile di evidenziazione del testo (tabella 27.77).
	<code>--help-pretty-print</code>	Elenca gli stili disponibili.
<code>-W PostScript</code>	<code>--language=PostScript</code>	Composizione in PostScript.
<code>-W html</code>	<code>--language=html</code>	Composizione in HTML.
<code>-W overstrike</code>	<code>--language=overstrike</code>	Testo con comandi di sovrascrittura.
<code>-W rtf</code>	<code>--language=rtf</code>	Composizione in RTF.
<code>-X codifica</code>	<code>--encoding=codifica</code>	Codifica in cui è scritto il testo (tabella 27.78).
<code>-Z</code>	<code>--pass-through</code>	Non modifica i file PostScript e PCL.

La tabella 27.77 riporta invece l'elenco di alcuni nomi di stili di evidenziazione della stampa, in funzione del contenuto del file di testo che si intende stampare. Questi nomi si usano con l'opzione `-E` e possono essere elencati completamente con l'opzione `--help-pretty-print`.

Tabella 27.77. Alcuni nomi che definiscono uno stile nel modo di evidenziare il testo.

Tipo	Descrizione
sh	Script di una shell Bourne.
html	Sorgente HTML.
mail	Messaggio di posta elettronica.
diff	File di differenze standard.
diffu	File di differenze in formato unificato.
makefile	File-make.
ada	Sorgente in linguaggio ADA.
c	Sorgente in linguaggio C.
cpp	Sorgente in linguaggio C++.
elisp	Sorgente in linguaggio Emacs Lisp.
fortran	Sorgente in linguaggio Fortran.
java	Sorgente in linguaggio Java.
pascal	Sorgente in linguaggio Pascal.
perl	Sorgente in linguaggio Perl.
python	Sorgente in linguaggio Python.

Tipo	Descrizione
scheme	Sorgente in linguaggio Scheme.
sql	Sorgente in linguaggio SQL.

La tabella 27.78 riporta l'elenco delle sigle che si possono utilizzare con l'opzione '-x' per definire la codifica con cui è scritto il testo da convertire. Si osservi che potrebbe non essere disponibile la codifica UTF-8.

Tabella 27.78. Alcuni nomi che identificano la codifica del testo.

Nome	Descrizione
88591 latin1	ISO 8859-1
88592 latin2	ISO 8859-2
88593 latin3	ISO 8859-3
88594 latin4	ISO 8859-4
88595 cyrillic	ISO 8859-5
88597 greek	ISO 8859-7
88599 latin5	ISO 8859-9
885910 latin6	ISO 8859-10
ascii	ASCII normale
asciifise asciifi asciise	ASCII finlandese e svedese
asciidkno asciidk asciino	ASCII danese e norvegese
ibmpc pc dos	IBM CP437.
mac	Mac
vms	VMS
hp8	HP
koi8	
ps PS	PostScript
pslatin1 ISOLatin1Encoding	PostScript ISO 8859-1

Prima di passare all'elenco di esempi comuni, vale la pena di proporre il comando che potrebbe essere usato in un filtro di stampa per gestire i file di testo, senza lasciare che questi vengano inviati direttamente alla stampante:

```
enscript -l -M a4 -f Courier@9.1/9.5 -B --margin=72:72:72:72 -o -
```

In questo modo si specifica che: si vuole ottenere una pagina virtuale per foglio; il formato della carta è A4 (si osservi l'uso del minuscolo); il testo deve utilizzare un carattere largo 9,1 punti e alto 9,5 punti (è la dimensione ottimale per stampare 80 colonne e 66 righe nello spazio disponibile, all'interno dei margini indicati); non si vogliono bordi attorno alla pagina virtuale, ma si lascia un margine di 72 punti; non si vuole alcuna intestazione; il risultato in PostScript deve essere emesso attraverso lo standard output.

Volendo intervenire nella configurazione di Magicfilter, si potrebbe sostituire la solita direttiva:

```
default cat
```

La nuova direttiva potrebbe essere come quella seguente:

```
default pipe /usr/bin/enscript -l -M a4 -f ↵
↵Courier@9.1/9.5 -B --margin=72:72:72:72 -o - 2> /dev/null
```

Gli esempi che vengono mostrati fanno riferimento alla configurazione tipica di Enscript.

- \$ **enscript pippo** [Invio]

Mette in stampa il file 'pippo', dopo averlo convertito in PostScript.

- \$ **enscript -2 pippo** [Invio]

Come nell'esempio precedente, su due colonne, ma senza cambiare la dimensione del carattere.

- \$ **enscript -2 -f Courier@5 pippo** [Invio]

Come nell'esempio precedente, su due colonne, con un carattere molto piccolo.

- \$ **enscript -2 -r -P laser pippo** [Invio]

Mette in stampa il file 'pippo', su due colonne orientando la carta in modo orizzontale. Richiede specificatamente l'uso della coda di stampa 'laser'.

- \$ **enscript -2 -r -o pippo.ps pippo** [Invio]

Come nell'esempio precedente, ma senza stampare: viene generato il file 'pippo.ps'.

- \$ **man 1 enscript | enscript -2 -r -l** [Invio]

Stampa la pagina di manuale *enscript(1)*, in orizzontale, su due colonne, impaginandola nel modo migliore per questo tipo di informazioni.

- \$ **enscript -2 -r -Esh /etc/profile** [Invio]

Stampa il file '/etc/profile', in orizzontale, su due colonne, utilizzando un sistema di evidenziamento ottimale per gli script di shell Bourne o derivate.

```
#!/bin/sh
enscript --portrait --columns=1 --margins=72:72:72:72 \
--no-header --lines-per-page=66 \
--font=Courier@9.5 --output=-
```

Lo script permette di generare un file PostScript attraverso lo standard output a partire da un file di testo normale, fornito attraverso lo standard input. Per la precisione, si vuole ottenere una sola colonna verticale, con un margine simmetrico di 72 punti (circa un pollice), senza scritte o bordi aggiuntivi, con un carattere di 9,5 punti (ciò dovrebbe garantire la stampa di 80 colonne su un foglio A4 normale).

```
#!/bin/sh
enscript --portrait --columns=1 --margins=72:72:72:72 \
--no-header --font=Courier@9.5 --output=-
```

Come nell'esempio precedente, ma senza porre un limite alla quantità di righe per pagina, che così possono essere anche di più delle 66 standard.

```
#!/usr/sbin/magicfilter
...
# Default entry -- for normal (text) files. MUST BE LAST.
#default cat \eE\ek2G\e(0N \eE
• default pipe /usr/bin/enscript -l -M a4 \
-f Courier@9.1/9.5 -B \
--margin=72:72:72:72 \
-o - \
2> /dev/null
```

L'esempio mostra l'ultima istruzione di uno script di Magicfilter, allo scopo di stampare automaticamente del testo normale aggiungendo un po' di margini, come già mostrato nell'esempio precedente.

```
#!/bin/sh
enscript --landscape --columns=1 --media=a4 \
--margins=72:72:72:72 --no-header --tabsize=8 \
• --encoding=88591 --non-printable-format=space \
--font=Courier@8.2/5.7 --output=-
```

Si tratta di un altro script, realizzato in modo da gestire file di testo fino a 136 colonne, in orizzontale, con margini da 72 punti. In questo senso, il carattere è stato scelto per riempire in modo ottimale lo spazio.

## 27.7 Filtri particolari con LPRng

LPRng è un sistema di stampa più evoluto di quello BSD tradizionale, anche se l'utilizzo è conforme a tale tradizione. LPRng ha la particolarità di consentire all'utente l'uso di un filtro di stampa aggiuntivo, fornito nella riga di comando dei programmi 'lpr' e 'lp', con l'opzione '-x'. Attraverso questa facoltà è possibile realizzare degli script che con gli stessi nomi 'lpr' e 'lp', da collocare in una directory che venga presa in considerazione prima di altre nell'elenco dei percorsi degli eseguibili. Così facendo si potrebbe controllare la stampa prima della sua esecuzione effettiva, senza dover intervenire in punti delicati della configurazione di LPRng stesso.

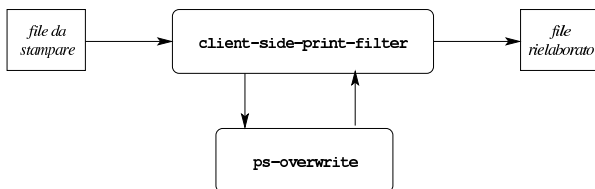
Questo capitolo mostra un esempio abbastanza completo in cui si sfrutta la possibilità offerta da LPRng di gestire dei filtri esterni, allo scopo di impedire la stampa di documenti troppo voluminosi e di sovrapporre i dati identificativi di chi ha stampato.

### 27.7.1 Schema di funzionamento

Il programma specificato con l'opzione '-x' di 'lpr' o di 'lp', nella versione di LPRng, viene eseguito inviandogli allo standard input il file da stampare. Questo programma, che qui viene chiamato 'client-side-print-filter', fa qualcosa, anche con l'ausilio di un secondo programma, quindi emette attraverso lo standard output il risultato da mandare finalmente al sistema di stampa (dove probabilmente viene elaborato ulteriormente prima di arrivare effettivamente alla stampa).

Figura 27.86. Schema di funzionamento del filtro proposto in questo capitolo.

```
lpr -x client-side-print-filter ...
```



Il programma 'ps-overwrite' che si vede nella figura, viene usato secondo la sintassi seguente. In pratica, 'ps-overwrite' riceve come argomento il nome di un file contenente una porzione di codice PostScript, quindi, riceve dallo standard input un file PostScript completo e dallo standard output emette un altro file PostScript risultante dalla fusione di quello ricevuto in ingresso e del codice aggiuntivo fornito dalla riga di comando:

```
ps-overwrite codice_ps < input_ps > output_ps
```

La funzione che esegue 'ps-overwrite' serve a sovrapporre, su ogni pagina del file ricevuto in ingresso, una scritta o dei disegni. La spiegazione dettagliata del funzionamento di 'ps-overwrite' appare in altra sezione (48.17), dove viene chiamato 'ps-overwrite.pl', ma qui, per completezza, viene riproposto il codice con cui lo si realizza:

```
#!/usr/bin/perl
local ($external_postscript_file) = $ARGV[0];
local ($external_postscript_code) = "";
local ($line) = "";

# Carica dal file esterno il codice da aggiungere.
open (PSCODE, "< $external_postscript_file");
while ($line = <PSCODE>)
{
    $external_postscript_code = $external_postscript_code
        . $line;
}
close (PSCODE);

# Legge lo standard input e cerca l'istruzione "showpage"
# per inserire il codice aggiunto.
while ($line = <STDIN>)
{
    if ($line =~ m/^\%/)
    {
        # Si tratta di un commento o di un'istruzione
        # speciale.
        print STDOUT ($line);
    }
    elsif ($line =~ m/^(.*)\bshowpage\b(.*?)$/)
    {
        # Si aggiunge il codice.
        print STDOUT ($1);
        print STDOUT ("\n");
        print STDOUT ("save");
        print STDOUT ("\n");
        print STDOUT ("{$external_postscript_code}");
        print STDOUT ("\n");
        print STDOUT ("restore");
        print STDOUT ("\n");
        print STDOUT ("showpage");
        print STDOUT ("\n");
        print STDOUT ($2);
        print STDOUT ("\n");
    }
    else
    {
        print STDOUT ($line);
    }
}
}
```

### 27.7.2 Script «lpr» e «lp»

Gli script 'lpr' e 'lp' vanno collocati in una directory che, nel percorso degli eseguibili (la variabile di ambiente PATH) appare prima di quella che contiene i programmi 'lpr' e 'lp' veri e propri di LPRng. Il contenuto di tali script potrebbe essere quello successivo:

```
#!/bin/sh
if [ -x /etc/script/client-side-print-filter ]
then
    /usr/bin/lpr -X /etc/script/client-side-print-filter "$@"
else
    /usr/bin/lpr "$@"
fi
```

```
#!/bin/sh
if [ -x /etc/script/client-side-print-filter ]
then
    /usr/bin/lp -X /etc/script/client-side-print-filter "$@"
else
    /usr/bin/lp "$@"
fi
```

Come si vede, si presume che i programmi 'lpr' e 'lp' originali si trovino effettivamente nella directory '/usr/bin/' e che lo

script `'client-side-print-filter'` sia collocato nella directory `'/etc/script/'`.

### 27.7.3 Funzioni usate da «client-side-print-filter»

Per semplificare la struttura dello script `'client-side-print-filter'`, alcune funzioni vengono realizzate in file esterni, da incorporare al momento dell'esecuzione. Segue il contenuto e la descrizione di tali funzioni.

Listato 27.90. La funzione `'f_postscript_a4_extra_text'` richiede un testo attraverso gli argomenti della riga di comando (uno o più di uno, indifferentemente) e produce un file, attraverso lo standard output, contenente del codice PostScript, il quale va successivamente sovrainposto alle pagine del documento finale da stampare.

```
#!/bin/sh
f_postscript_a4_extra_text () {
    local TEXT="$*"
    # Remove text that cannot be inserted inside PostScript
    # as is.
    TEXT='echo $TEXT \
| sed --posix "s/[^0-9a-zA-Z\\|\\|\\|?|!|.|:|+|=|_| / "g'
    #
    echo "/mm { 2.8346456 mul } def"
    echo " 0.1 mm setlinewidth"
    echo " 10 mm 228.5 mm moveto"
    echo " 9.5 mm 228.5 mm lineto stroke"
    echo " 10 mm 148.5 mm moveto"
    echo " 9.5 mm 148.5 mm lineto stroke"
    echo " 10 mm 68.5 mm moveto"
    echo " 9.5 mm 68.5 mm lineto stroke"
    echo " 5 mm 287 mm moveto"
    echo " 10 mm 292 mm lineto"
    echo " 5 mm 292 mm lineto"
    echo " 5 mm 287 mm lineto stroke"
    echo " 10 mm 130.65 mm moveto"
    echo " 10 mm 131.15 mm lineto stroke"
    echo " 14 mm 30 mm moveto"
    echo " 90 rotate"
    echo " /Helvetica findfont 2 mm scalefont setfont"
    echo " ($TEXT)"
    echo " 0 setgray"
    echo " show"
    echo " stroke"
}
```

Listato 27.91. La funzione `'f_postscript_check_max_pages'` serve a controllare che il file PostScript indicato come primo argomento contenga al massimo la quantità di pagine specificata come secondo. La funzione serve solo in quanto restituisce *Vero* se la quantità di pagine è minore o uguale al limite e altrimenti restituisce *Falso*.

```
#!/bin/sh
f_postscript_check_max_pages () {
    local INPUT_FILE="$1"
    local MAX_PAGES="$2"
    local PAGES_LINE=""
    local PAGES=""
    #
    PAGES_LINE='grep "^%Pages: [0-9]" \
$INPUT_FILE 2> "/dev/null"'
    PAGES='echo $PAGES_LINE \
| sed "s/^%Pages: \([0-9][0-9]*\)\/1/" 2> "/dev/null"'
    #
    if [ $PAGES -gt $MAX_PAGES ]
    then
        false
    else
        true
    fi
}
```

Listato 27.92. La funzione `'f_file_to_postscript'` serve a convertire un file in un formato PostScript adatto alla sovrainposizione di altro codice. Il file da elaborare proviene dallo standard input, mentre il risultato dell'elaborazione viene emesso attraverso lo standard output. Se il file in ingresso non viene riconosciuto, non avviene alcuna trasformazione.

```
#!/bin/sh
f_file_to_postscript () {
    local FILE_TYPE=""
    local FILE_MAIN_TYPE=""
    local TEMPORARY='tempfile'
    touch $TEMPORARY
    # Save the input inside a temporary file.
    cat > $TEMPORARY
    # Get file type as MIME type.
    FILE_TYPE='file --brief --mime $TEMPORARY 2> "/dev/null"'
    FILE_MAIN_TYPE='echo "$FILE_TYPE" | sed "s/\/.*$//"'
    # Check file type.
    if [ "$FILE_TYPE" = "text/plain; charset=utf-8" ] \
|| [ "$FILE_TYPE" = "text/x-asm; charset=utf-8" ]
    then
        # Enscript is not able to work with UTF-8.
        recode -f UTF-8..Latin1 $TEMPORARY \
> "/dev/null" 2> "/dev/null"
        # Convert to PostScript.
        cat $TEMPORARY \
| /usr/bin/enscript -X 88591 -l -M a4 \
-f Courier@5.5/9.5 \
--no-header \
--margin=72:72:72:72 \
-o - 2> "/dev/null" \
> $TEMPORARY.output0
        ps2ps $TEMPORARY.output0 $TEMPORARY.output
    elif [ "$FILE_TYPE" = "text/plain; charset=iso-8859-1" ] \
|| [ "$FILE_TYPE" = "text/plain; charset=us-ascii" ] \
|| [ "$FILE_MAIN_TYPE" = "text" ]
    then
        # Convert to PostScript and suppose it is Latin1.
        cat $TEMPORARY \
| /usr/bin/enscript -X 88591 -l -M a4 \
-f Courier@5.5/9.5 \
--no-header \
--margin=72:72:72:72 \
-o - 2> "/dev/null" \
> $TEMPORARY.output0
        ps2ps $TEMPORARY.output0 $TEMPORARY.output
    elif [ "$FILE_TYPE" = "application/octet-stream" ]
    then
        # This is intended to be a text file with some bad
        # characters inside.
        cat $TEMPORARY \
| tr "\000" " " \
| /usr/bin/enscript -X 88591 -l -M a4 \
-f Courier@5.5/9.5 \
--no-header \
--margin=72:72:72:72 \
-o - 2> "/dev/null" \
> $TEMPORARY.output0
        ps2ps $TEMPORARY.output0 $TEMPORARY.output
    elif [ "$FILE_TYPE" = "application/pdf" ]
    then
        pdf2ps $TEMPORARY $TEMPORARY.output \
2> "/dev/null" 1> "/dev/null"
    elif [ "$FILE_TYPE" = "application/postscript" ]
    then
        ps2ps $TEMPORARY $TEMPORARY.output \
2> "/dev/null" 1> "/dev/null"
    elif [ "$FILE_MAIN_TYPE" = "image" ]
    then
        convert $TEMPORARY $TEMPORARY.pnm \
2> "/dev/null" 1> "/dev/null"
        pnmtops -imagewidth 7 -imageheight 10 $TEMPORARY.pnm \
> $TEMPORARY.output0 2> "/dev/null"
        ps2ps $TEMPORARY.output0 $TEMPORARY.output
    else
        # Cannot convert it.
        ln $TEMPORARY $TEMPORARY.output
    fi
    # Get output file type.
    FILE_TYPE='file --brief --mime $TEMPORARY.output \
2> "/dev/null"'
    if [ "$FILE_TYPE" = "application/postscript" ]
    then
```

```

cat $TEMPORARY.output
rm $TEMPORARY $TEMPORARY.* \
  1> "/dev/null" 2> "/dev/null"
true
return
else
cat $TEMPORARY.output
rm $TEMPORARY $TEMPORARY.* \
  1> "/dev/null" 2> "/dev/null"
false
return
fi
}

```

Listato 27.93. La funzione `'f_network_get_eth'` riceve come argomento due numeri interi, da zero in su. Il primo numero, ovvero  $n$ , rappresenta la  $n$ -esima interfaccia di rete di tipo Ethernet. La funzione restituisce il nome effettivo della  $n$ -esima interfaccia esistente, ammesso che esista. Lo zero rappresenta la prima di tali interfacce. Il secondo numero fornito come argomento rappresenta la quantità massima di interfacce reali da scandire; di solito è sufficiente il valore 16.

```

#!/bin/sh
f_network_get_eth () {
  local INTERFACE_N="$1"
  local INTERFACE_MAX="$2"
  local ETH_NAME=""
  local ETH_N=0
  local N=0
  while [ $N -lt $INTERFACE_MAX ]
  do
    if /sbin/ifconfig eth$N 1> "/dev/null" 2> "/dev/null"
    then
      ETH_NAME="eth$N"
      #
      if [ "$INTERFACE_N" == "$ETH_N" ]
      then
        echo $ETH_NAME
        true
        return
      else
        ETH_N=$((ETH_N+1))
        N=$((N+1))
      fi
    else
      N=$((N+1))
    fi
  done
  # If still here, there is a problem.
  echo ""
  false
  return
}

```

Listato 27.94. La funzione `'f_network_get_wlan'` riceve come argomento due numeri interi, da zero in su. Il primo numero, ovvero  $n$ , rappresenta la  $n$ -esima interfaccia di rete senza fili (WiFi). La funzione restituisce il nome effettivo della  $n$ -esima interfaccia esistente, ammesso che esista. Lo zero rappresenta la prima di tali interfacce. Il secondo numero fornito come argomento rappresenta la quantità massima di interfacce reali da scandire; di solito è sufficiente il valore 16.

```

#!/bin/sh
f_network_get_wlan () {
  local INTERFACE_N="$1"
  local INTERFACE_MAX="$2"
  local WLAN_NAME=""
  local WLAN_N=0
  local N=0
  while [ $N -lt $INTERFACE_MAX ]
  do
    if /sbin/iwconfig wlan$N 2> "/dev/null" \
    | grep IEEE 1> "/dev/null" 2> "/dev/null"
    then
      WLAN_NAME="wlan$N"
      if [ "$INTERFACE_N" == "$WLAN_N" ]
      then
        echo $WLAN_NAME
        true

```

```

return
else
  WLAN_N=$((WLAN_N+1))
  N=$((N+1))
fi
else
  N=$((N+1))
fi
done
# If still here, there is a problem.
echo ""
false
return
}

```

Listato 27.95. La funzione `'f_network_get_ipv4_from_ifname'` richiede come argomento l'indicazione di un nome di interfaccia reale e si occupa di emettere attraverso lo standard output l'indirizzo IPv4 associato a tale interfaccia.

```

#!/bin/sh
f_network_get_ipv4_from_ifname () {
  local IF_NAME="$1"
  local IPV4_LINE=""
  local IPV4=""
  IPV4_LINE='/sbin/ifconfig $IF_NAME 2> /dev/null \
  | grep "inet addr:"'
  IPV4='echo $IPV4_LINE \
  | sed "s/^.*inet addr:\([0-9.]*\).*$/\1/'
  echo $IPV4
}

```

#### 27.7.4 Lo script «client-side-print-filter»

Lo script `'client-side-print-filter'` utilizza le funzioni descritte nella sezione precedente. Si suppone che ogni funzione risieda in un proprio file con lo stesso nome, nella directory `'/etc/script.functions/'`.

Listato 27.96. Lo script `'client-side-print-filter'`.

```

#!/bin/sh
#
# lpr -X client-side-print-filter
#
# Please, change the value below, to the max page number a
# user can print at the time.
MAX_ALLOWED_PAGES="11"
# Include public functions.
. /etc/script.functions/f_postscript_a4_extra_text
. /etc/script.functions/f_postscript_check_max_pages
. /etc/script.functions/f_file_to_postscript
. /etc/script.functions/f_network_get_eth
. /etc/script.functions/f_network_get_wlan
. /etc/script.functions/f_network_get_ipv4_from_ifname
#
get_ipv4 () {
  #
  local IPV4=""
  local WLAN0='f_network_get_wlan 0 16'
  local WLAN1='f_network_get_wlan 1 16'
  local ETH0='f_network_get_eth 0 16'
  local ETH1='f_network_get_eth 1 16'
  if [ "$IPV4" = "" ] && [ ! "$WLAN0" = "" ]
  then
    IPV4='f_network_get_ipv4_from_ifname $WLAN0'
  fi
  if [ "$IPV4" = "" ] && [ ! "$WLAN1" = "" ]
  then
    IPV4='f_network_get_ipv4_from_ifname $WLAN1'
  fi
  if [ "$IPV4" = "" ] && [ ! "$ETH0" = "" ]
  then
    IPV4='f_network_get_ipv4_from_ifname $ETH0'
  fi
  if [ "$IPV4" = "" ] && [ ! "$ETH1" = "" ]
  then
    IPV4='f_network_get_ipv4_from_ifname $ETH1'
  fi
  if [ "$IPV4" = "" ]

```

```

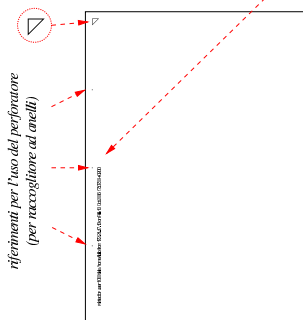
then
    IPV4='f_network_get_ipv4_from_ifname lo'
fi
echo "$IPV4"
}
TEMPORARY='tempfile'
touch $TEMPORARY
IPV4='get_ipv4'
DATE='date -R'
if f_file_to_postscript > $TEMPORARY.output
then
    # Check max pages.
    if f_postscript_check_max_pages "$TEMPORARY.output" \
        "$MAX_ALLOWED_PAGES"
    then
        logger "[${0}] \"\$USER\" prints $PAGES pages."
        if f_postscript_a4_extra_text \
            printed by user $UID $USER $HOME from $IPV4 \
            on $DATE > $TEMPORARY.fg.ps
        then
            cat $TEMPORARY.output \
                | /etc/script/ps-overwrite $TEMPORARY.fg.ps
        else
            logger "Error: function \
f_postscript_a4_extra_text failed."
            cat $TEMPORARY.output
        fi
    else
        logger "[${0}] \"\$USER\" would like to print \
$PAGES pages, but only $MAX_ALLOWED_PAGES is allowed."
        fi
    else
        logger "[${0}] Error: function f_file_to_postscript \
failed."
        cat $TEMPORARY.output
    fi
fi
rm $TEMPORARY $TEMPORARY.*

```

Questo script trasmette il proprio standard input alla funzione `'f_file_to_postscript'`, la quale produce un file temporaneo che dovrebbe risultare in un formato PostScript abbastanza semplificato. Se l'operazione ha successo, con l'ausilio della funzione `'f_postscript_check_max_pages'` si verifica che il numero massimo di pagine non sia stato superato (questo script pone come limite 11 pagine) e se tutto va bene, produce un altro file temporaneo contenente il codice PostScript da sovrapporre alle pagine originali. La fusione tra i codici PostScript avviene per opera del programma `'ps-overwrite'`, già descritto. Il risultato viene emesso attraverso lo standard output. Se invece ci sono dei problemi, a seconda dei casi non viene emesso alcunché, oppure viene riproposto lo stesso file originale.

Figura 27.97. Risultato della sovrapposizione delle stampe, attraverso il filtro proposto in questo capitolo.

printed by user 1000 tizio /home/tizio from 172.21.17.13 on Fri, 13 Oct 2006 17:07:09 +0200



## 27.8 Introduzione a Cups

Cups,<sup>13</sup> ovvero *Common unix printing system*, è un sistema di stampa nettamente differente rispetto al modello BSD e a quello di System V. Cups utilizza il protocollo IPP (*Internet printing protocol*) con cui è anche possibile una gestione remota dei server, attraverso un navigatore HTTP comune.

L'utilizzo di Cups richiede delle nozioni di reti TCP/IP (capitolo 32 e successivi); tuttavia si preferisce anticipare l'argomento a completamento della trattazione della gestione delle stampanti.

### 27.8.1 Visione generale

Cups incorpora le funzionalità di un server HTTP, attraverso il quale è possibile anche interagire per definire e modificare la configurazione delle stampanti. Pertanto, il programma frontale per intervenire in questo modo diventa un navigatore normale. In condizioni normali, la comunicazione con il protocollo HTTP avviene usando la porta 631, come prescrive il protocollo IPP.<sup>14</sup>

Questo server HTTP specifico deve essere configurato, in modo da stabilire chi può accedere e a cosa si può accedere. A tale proposito, la prima cosa che si dovrebbe fare normalmente è modificare il file di configurazione del demone `'cupsd'` (`'/etc/cups/cupsd.conf'`).

La configurazione delle stampanti viene fatta attraverso il protocollo HTTP, oppure attraverso programmi di servizio specifici, che però possono agire solo nell'ambito dell'elaboratore locale; pertanto non è necessario intervenire manualmente all'interno di file, che comunque esistono e sono gestiti autonomamente da Cups.

Cups deve poter riconoscere il tipo dei file che vengono inviati alla stampa ed essere a conoscenza delle caratteristiche delle stampanti utilizzabili. L'individuazione del tipo di file avviene attraverso delle definizioni di tipi MIME; la gestione delle stampanti si avvale di una serie di file di configurazione specifici per ogni modello utilizzabile (file PPD). Queste informazioni, naturalmente, sono già fornite in modo predefinito. Per la precisione, i file di definizione delle stampanti dovrebbero trovarsi al di sotto della directory `'/usr/share/ppd/'`.

Bisogna osservare però che Cups gestisce i filtri di stampa in proprio e potrebbe incorporare alcuni programmi appositi.

Infine, Cups è in grado di comandare delle stampanti locali, oppure di raggiungere delle stampanti remote, attraverso vari protocolli; in particolare: IPP, LPD (quello dei sistemi di stampa tradizionali) e SMB.

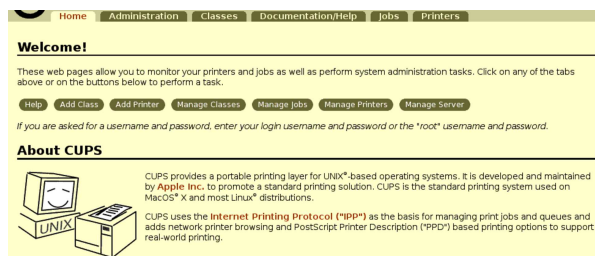
### 27.8.2 Stampanti logiche, classi e istanze

Per la gestione delle stampanti, Cups utilizza una terminologia particolare. Per come si intendono normalmente, le stampanti sono per Cups delle *stampanti logiche*. Quando per una stampante logica si vuole definire una configurazione alternativa, si dichiara una *istanza*. Un insieme di stampanti può essere raggruppato in una *classe*; quando si invia una stampa a una classe, invece che a una stampante logica precisa (o a una sua istanza particolare), si fa in modo che la stampa avvenga sulla prima stampante libera della classe indicata.

### 27.8.3 Configurazione di una stampante locale

L'utente poco esperto che intende utilizzare Cups, può limitarsi a configurare le proprie stampanti utilizzando semplicemente l'interfaccia a cui si accede tramite il protocollo IPP. In pratica, serve un navigatore ipertestuale, con il quale ci si connette all'indirizzo locale <http://localhost:631>. Inizialmente, la configurazione standard di Cups dovrebbe consentire l'amministrazione da parte dell'utente `'root'`, il quale si identifica con la stessa parola d'ordine necessaria per accedere al sistema operativo.

Figura 27.98. La pagina iniziale di Cups.



Per aggiungere una stampante logica, si seleziona normalmente la scheda di amministrazione e da lì si sceglie la voce di aggiunta di una stampante. È bene che la prima stampante logica, o comunque quella che si intende usare come stampante predefinita, abbia il nome 'lp', per rispettare la tradizione.

Figura 27.99. La scheda di amministrazione di Cups.

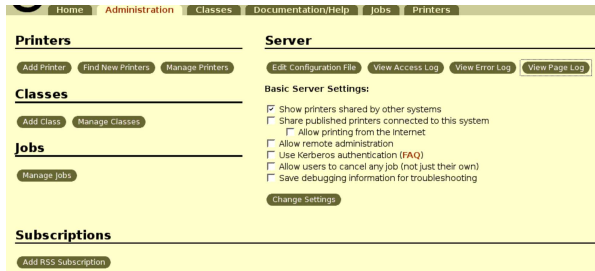
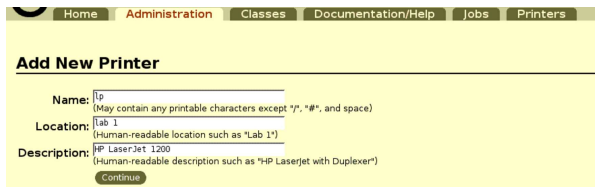
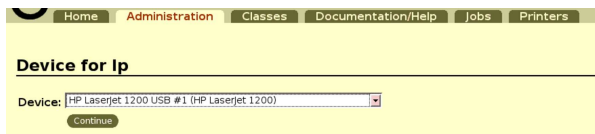


Figura 27.100. Aggiunta di una stampante: viene richiesta l'indicazione del nome della stampante logica e altre informazioni descrittive.



Una volta dichiarato il nome della stampante logica che si intende aggiungere, viene richiesto di specificare dove si trova tale stampante. Se si tratta di una stampante locale, dovrebbe risultare individuata automaticamente, come si vede nella figura successiva.

Figura 27.101. Selezione della stampante locale, corrispondente a una HP LaserJet 1200, collegata alla porta USB.



Viene quindi chiesto di specificare il file PPD di definizione delle caratteristiche della stampante. In questo caso, essendo stato individuato automaticamente il modello della stampante, viene anche suggerito il file PPD più adatto, per cui basta confermare.

Figura 27.102. Selezione del file PPD.

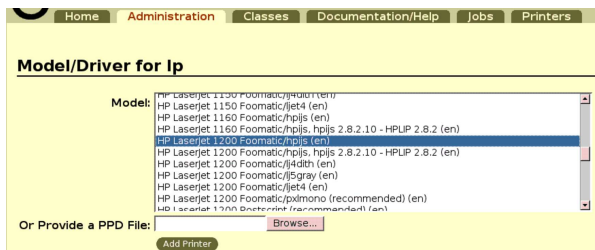
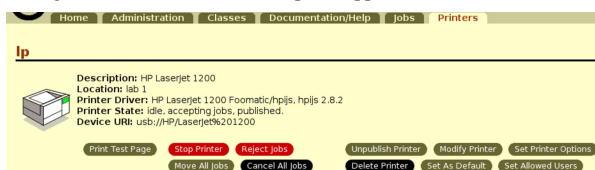


Figura 27.103. Stato della stampante appena definita.



A questo punto, la stampante logica è stata creata, ma rimane ancora da indicare tale stampante, essendo l'unica, come predefinita: se non si specifica qual è la stampante predefinita, ogni volta che si stampa occorre indicarlo espressamente.

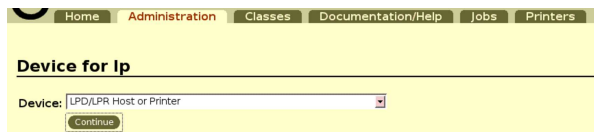
Figura 27.104. Stampante predefinita.



## 27.8.4 Configurazione di una stampante remota

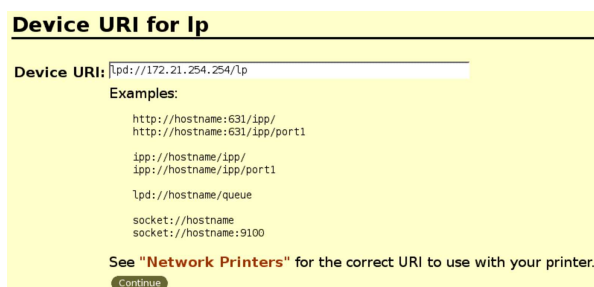
Se si vuole stampare attraverso una stampante accessibile dalla rete locale, Cups va configurato di conseguenza. Quando viene richiesto di specificare dove si trova la stampante, occorre scegliere un protocollo di rete, come si vede nella figura successiva, dove si presume di disporre di un server LPD (lo standard del sistema di stampa BSD).

Figura 27.105. Selezione della stampante remota, utilizzando il protocollo tradizionale LPD.



Quindi occorre specificare in modo preciso un indirizzo URI per raggiungere tale stampante. Nella figura successiva si suppone che si tratti dell'indirizzo IPv4 172.21.254.254.

Figura 27.106. Indicazione dell'indirizzo URI della stampante remota. La coda di stampa remota corrisponde al nome 'lp'.



In tal caso, non è possibile individuare automaticamente il tipo di stampante, pertanto occorre specificare manualmente la casa e il modello.

Figura 27.107. Indicazione della stampante: casa.

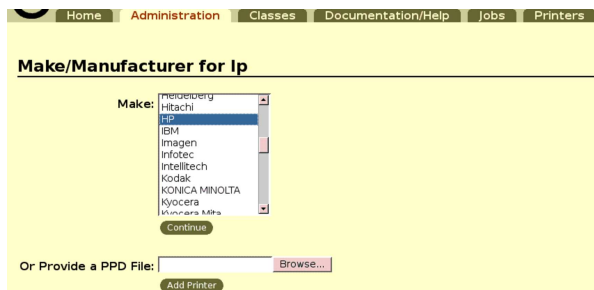
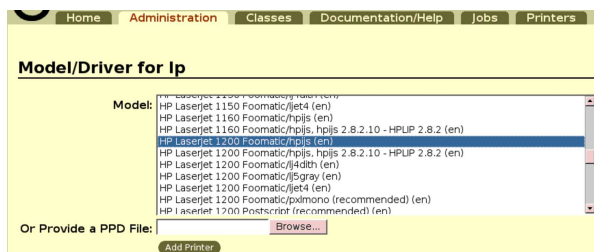


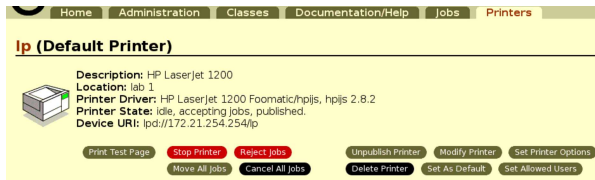
Figura 27.108. Indicazione della stampante: file PPD.





Al termine, occorre avere la stessa accortezza di impostare la stampante predefinita.

Figura 27.109. Stampante predefinita.



## 27.9 Cups: programmi di servizio e configurazione

Cups potrebbe essere configurato manualmente e amministrato attraverso programmi di servizio, benché questo tipo di approccio sia sconsigliabile data la natura di Cups stesso. Tuttavia, qualche annotazione al riguardo potrebbe essere utile in certe situazioni.

### 27.9.1 Individuazione automatica delle stampanti

Cups è in grado di individuare le stampanti locali e quelle di rete, in modo autonomo; attraverso il comando `lpinfo` si può scorrere in particolare l'elenco delle disponibilità locali:

```
# lpinfo -v [Invio]

network socket
direct usb://HP/LaserJet%201200
direct hp:/usb/HP_LaserJet_1200?serial=00CNC2180308
network http
network ipp
network lpd
direct parallel:/dev/lp0
direct scsi
network smb
```

Intuitivamente si comprende il significato delle varie notazioni, che si rifanno allo schema di un URI. Per esempio, la porta parallela corrispondente al file di dispositivo `/dev/lp0`, si indica come `parallel:/dev/lp0`; eventualmente, volendo definire una stampante logica diretta in pratica a un file, basta usare un URI del tipo `file:/percorso_assoluto`.

Quando un server mette a disposizione il proprio servizio di stampa attraverso il protocollo IPP, si può fare riferimento a questo «dispositivo» remoto attraverso una notazione del tipo `ipp://nodo/printers/stampante_logica`. Nello stesso modo, se il protocollo disponibile è quello tradizionale della stampa BSD, la notazione è del tipo: `lpd://nodo/coda`.

### 27.9.2 Amministrazione

Il sistema di stampa Cups può essere impostato localmente attraverso il programma `lpadmin`. Per definire una *stampante logica*, si procede con un comando simile all'esempio seguente:

```
# lpadmin -p lp -E ↵
↳ -v usb://HP/LaserJet%201200 ↵
↳ -P /usr/share/ppd/cups-included/HP/laserjet.ppd [Invio]
```

In questo modo, è stata definita la stampante `lp`, abbinata a una porta USB e individuata esattamente come HP LaserJet 1200. La stampa viene gestita attraverso la definizione contenuta nel file `/usr/share/ppd/cups-included/HP/laserjet.ppd` che intuitivamente rappresenta una stampante HP LaserJet generica.

In modo analogo, si può definire una stampante di rete. L'esempio seguente definisce la stessa stampante logica `lp`, ma offerta dal nodo di rete 172.21.254.254, attraverso il protocollo LPD (quello tradizionale del sistema di stampa BSD). Presso il server remoto si fa riferimento a una coda di stampa con lo stesso nome `lp`:

```
# lpadmin -p lp -E -v lpd:172.21.254.254/lp [Invio]
```

Volendo adattare il file trasmesso alla stampante remota, prima dell'invio, supponendo che si tratti sempre di una stampante HP LaserJet 1200, si potrebbe indicare il file PPD come già fatto nel caso della stampante locale:

```
# lpadmin -p lp -E ↵
↳ -v lpd:172.21.254.254/lp ↵
↳ -P /usr/share/ppd/cups-included/HP/laserjet.ppd [Invio]
```

Per far sì che la stampante logica `lp` locale sia anche quella predefinita, occorre il comando seguente:

```
# lpadmin -d lp [Invio]
```

Eventualmente, per eliminare la stampante logica `lp`, occorre il comando seguente:

```
# lpadmin -x lp [Invio]
```

La sintassi di `lpadmin` si può riassumere nei modelli seguenti:

```
lpadmin -p stampante_logica altre_opzioni
```

```
lpadmin -d stampante_logica
```

```
lpadmin -x stampante_logica
```

La tabella successiva descrive brevemente alcune opzioni di `lpadmin`.

Tabella 27.111. Alcune opzioni di `lpadmin`.

Opzione	Descrizione
<code>-p stampante_logica</code>	Definisce la stampante logica su cui intervenire.
<code>-x stampante_logica</code>	Elimina la stampante logica o la classe indicata. È incompatibile con l'opzione <code>-p</code> .
<code>-x classe</code>	Elimina la stampante logica o la classe indicata. È incompatibile con l'opzione <code>-p</code> .
<code>-d stampante_logica</code>	Stabilisce che la stampante logica o la classe indicata è la destinazione predefinita.
<code>-d classe</code>	Stabilisce che la stampante logica o la classe indicata è la destinazione predefinita.
<code>-c classe</code>	Abbinata la stampante logica (specificata con l'opzione <code>-p</code> ) a una classe.
<code>-r classe</code>	Toglie la stampante logica (specificata con l'opzione <code>-p</code> ) dalla classe specificata.
<code>-v uri</code>	Abbinata la stampante a un dispositivo di stampa, oppure a un servizio remoto.
<code>-P file_ppd</code>	Abbinata un file PPD, contenente le informazioni necessarie a usare la stampante.
<code>-E</code>	Abilita la stampante logica e consente di inviarle dei processi di stampa.

Una volta definita una stampante logica, questa può essere abilitata e disabilitata con i comandi `cupsenable` e `cupsdisable` rispettivamente (l'opzione `-E` di `lpadmin` abilita contestualmente la stampante logica); inoltre, i processi di stampa possono essere accettati o rifiutati. La distinzione tra i due tipi di controlli sta nel fatto che la disabilitazione ferma la stampa, ma non impedisce l'accodarsi di altri processi di stampa; nel secondo caso si impedisce proprio l'aggiunta di altri processi. I comandi di disabilitazione e di rifiuto, hanno in comune l'opzione `-r`, con cui è possibile descriverne il motivo. La tabella successiva riepiloga l'uso di questi comandi di controllo.

Tabella 27.112. Comandi di contorno a `lpadmin`.

Comando	Descrizione
<code>cupsenable [stampante_logica   classe]</code>	Abilita il funzionamento della stampante logica o della classe indicata.
<code>cupsdisable [-r motivazione] ↵</code> <code>↳ [stampante_logica   classe]</code>	Disabilita il funzionamento della stampante logica o della classe indicata, specificando eventualmente la motivazione.
<code>cupsaccept [stampante_logica   classe]</code>	Fa sì che la stampante logica o la classe indicata accetti dei nuovi processi di stampa.

Comando	Descrizione
<code>cupsreject [-r <i>motivazione</i>] ← ↔[<i>stampante_logica</i>   <i>classe</i>]</code>	Impedisce l'invio alla stampante logica o alla classe di nuovi processi di stampa.

Le **classi** sono dei raggruppamenti di stampanti logiche, a cui si può fare riferimento. In pratica, si può inviare un processo di stampa a una classe invece che a una stampante logica ben precisa; in tal modo è il sistema di stampa a decidere quale deve essere la stampante vera e propria che deve eseguire il lavoro, nell'ambito della classe stessa.

I comandi di controllo già descritti funzionano quindi anche in riferimento alle classi; tuttavia, occorre considerare che le stampanti logiche non ereditano i comandi dati alle classi. In questo modo, si può disabilitare una classe, ma ciò non disabilita implicitamente le stampanti che ne fanno parte.

Le informazioni sulle stampanti logiche e sulle classi vengono memorizzate all'interno di due file: `/etc/cups/printers.conf` e `/etc/cups/classes.conf`. Si tratta di file di testo che, teoricamente, possono essere modificati a mano, ma in generale conviene lasciare fare a `lpadmin` e agli altri comandi di controllo.

### 27.9.3 Configurazione principale del servizio di stampa

Cups si configura principalmente attraverso il file `/etc/cups/cupsd.conf`. La configurazione standard prevede che sia consentito l'accesso al servizio dalla rete, attraverso la porta 631, ma l'amministrazione remota e l'accesso al servizio di stampa potrebbero risultare esclusi.

Il file di configurazione è un file di testo normale, in cui le righe bianche e quelle vuote vengono ignorate; inoltre i commenti sono introdotti dal simbolo `#` e conclusi dalla fine della riga. La prima parte del file di configurazione è utilizzato normalmente per una serie di direttive globali, che hanno la forma di assegnamenti, del tipo:

*opzione valore\_associato*

Alcune di queste direttive sono descritte nella tabella successiva.

Tabella 27.113. Alcune direttive del file di configurazione `/etc/cups/cupsd.conf`.

Direttiva	Descrizione
<code>AccessLog {<i>file</i>   <i>syslog</i>}</code> <code>PageLog {<i>file</i>   <i>syslog</i>}</code> <code>ErrorLog {<i>file</i>   <i>syslog</i>}</code>	Le direttive specificano rispettivamente il file delle registrazioni da usare per annotare gli accessi, le pagine stampate (ogni pagina stampata ha una propria registrazione) e gli errori. Se si usa la parola chiave <code>'syslog'</code> , questi messaggi vengono inviati al registro di sistema.
<code>LogLevel {<i>debug2</i>   <i>debug</i> ← ↔   <i>info</i>   <i>warn</i>   <i>error</i>   <i>none</i>}</code>	Specifica il livello di dettaglio delle annotazioni che riguardano gli errori. La parola chiave <code>'debug2'</code> richiede una registrazione completa, mentre <code>'none'</code> annulla qualunque registrazione. Le altre parole chiave rappresentano ordinatamente dei valori intermedi.
<code>DataDir <i>directory</i></code>	Specifica la directory a partire dalla quale sono collocati vari file di dati di Cups. Generalmente si tratta di <code>/usr/share/cups/</code> .
<code>DocumentRoot <i>directory</i></code>	Specifica la directory a partire dalla quale sono collocati i file usati dal servizio HTTP incorporato in Cups. In altri termini, per ciò che riguarda la configurazione comune, corrisponde a <code>http://localhost:631/</code> .

Direttiva	Descrizione
<code>ServerRoot <i>directory</i></code>	Specifica la directory a partire dalla quale si articolano i file di configurazione. Di solito si tratta di <code>/etc/cups/</code> .
<code>Printcap <i>file</i></code>	Specifica il file da usare al posto del tradizionale <code>/etc/printcap</code> , allo scopo di garantire la compatibilità con i programmi che vanno a leggerlo per conoscere le code di stampa disponibili.
<code>User <i>utente</i></code> <code>Group <i>gruppo</i></code>	Specifica l'utente e il gruppo da usare quando Cups avvia dei sottoprocessi. Si tratta normalmente dell'utente <code>'lp'</code> e del gruppo <code>'sys'</code> .
<code>TempDir <i>directory</i></code>	Specifica la directory temporanea da usare per la creazione di file temporanei. Questa directory, che normalmente corrisponde a <code>/var/spool/cups/tmp/</code> , deve essere accessibile in scrittura dall'utente e dal gruppo specificati con le direttive <code>'User'</code> e <code>'Group'</code> .
<code>Port <i>n_porta</i></code> <code>Listen <i>nodo</i> [<i>:n_porta</i>]</code>	La direttiva <code>'Port'</code> specifica una porta alla quale Cups deve mettersi in ascolto. La porta predefinita è 631, che notoriamente è riservata al protocollo IPP. La direttiva <code>'Listen'</code> consente di definire un nodo di rete a cui concedere accesso, che deve rivolgersi eventualmente alla porta indicata dopo i due punti. Una o più direttive <code>'Listen'</code> , usate da sole, senza alcuna direttiva <code>'Port'</code> , consentono in pratica di limitare l'accesso al servizio di stampa, escludendo tutti gli altri.
<code>Browsing {<i>On</i>   <i>Off</i>}</code>	Abilita o disabilita la risposta a interrogazioni circolari e l'utilizzo dello stesso sistema per interrogare la disponibilità di altre stampanti dalla rete.
<code>BrowseAddress <i>indirizzo_broadcast</i></code>	Permette di stabilire l'indirizzo broadcast da usare per le interrogazioni circolari. Di solito si usa quello corrispondente alla propria rete locale.
<code>BrowseAllow <i>modello_nodi</i></code> <code>BrowseDeny <i>modello_nodi</i></code>	La direttiva <code>'BrowseAllow'</code> consente, mentre la direttiva <code>'BrowseDeny'</code> vieta, l'interrogazione circolare del servizio di stampa ai nodi di rete specificati dal modello.
<code>BrowseOrder {<i>allow</i>, <i>deny</i> ← ↔   <i>deny</i>, <i>allow</i>}</code>	Definisce l'ordine di analisi delle direttive <code>'BrowseAllow'</code> e <code>'BrowseDeny'</code> . La sequenza <code>'allow, deny'</code> fa sì che i pacchetti di interrogazione siano accettati a meno che espressamente rifiutati; la sequenza <code>'deny, allow'</code> fa sì che i pacchetti di interrogazione siano rifiutati a meno che espressamente accettati.
<code>BrowsePoll <i>nodo</i> :<i>porta</i></code>	Permette di richiedere espressamente l'interrogazione del nodo di rete indicato, alla porta specificata.

Direttiva	Descrizione
<code>ImplicitClass {On Off}</code>	Abilita o disabilita la definizione di classi implicite. Si può avere una classe implicita quando in una rete più elaboratori offrono l'accesso a una stampante denominata nello stesso modo.
<code>SystemGroup gruppo</code>	Stabilisce il gruppo a cui deve appartenere un utente per essere classificato come <code>'system'</code> in fase di autenticazione. Di solito, l'utente viene aggregato al gruppo stabilito in questo modo nel file <code>'/etc/group'</code> .

Generalmente, la parte finale del file di configurazione è occupata da direttive racchiuse in contesti particolari, definiti come *collocazioni*. Questo tipo di informazioni viene preso in considerazione in seguito, in occasione della descrizione dell'amministrazione remota attraverso il protocollo HTTP. Tuttavia, si tenga presente che per consentire l'accesso di richieste di stampa, occorre intervenire proprio all'interno di queste collocazioni.

Per abilitare l'accesso al servizio di stampa da parte di altri elaboratori, oltre che configurare la collocazione corrispondente alle stampanti (viene mostrato in seguito), occorre che il demone `'cupsd'` sia in ascolto della porta prevista e consenta l'accesso a tali elaboratori. Si può usare la direttiva `'Port'` per indicare una o più porte differenti rispetto allo standard, costituito da 631; inoltre, la direttiva `'Listen'` consente di specificare un nodo di rete particolare, nel caso si voglia limitare questo accesso. Infatti, se si usano solo direttive `'Listen'`, senza alcuna direttiva `'Port'`, si esclude di fatto l'accesso da parte di tutti gli altri nodi di rete.

Eventualmente, può essere consentita l'interrogazione (ovvero la scansione) del servizio, attraverso richieste circolari. Ciò permette ad altri nodi di rete di venire a conoscenza dell'esistenza di un servizio di stampa nel proprio elaboratore. Nello stesso modo, si può consentire al demone `'cupsd'` di fare altrettanto alla ricerca di altri servizi di stampa alla propria portata. Per questo, si usano le direttive del tipo `'Browse*'`.

Cups non ha bisogno del file `'/etc/printcap'`, né di un altro equivalente. Tuttavia, altri programmi potrebbero richiedere la sua presenza per conoscere la disponibilità di code di stampa. Cups può quindi tenere aggiornato questo file, in base alla disponibilità di stampanti logiche e di classi. Per esempio, il contenuto di questo file potrebbe apparire così:

```
laser1:
laser2:
```

Come si intende, tutti gli altri campi dei record tipici di questo file, sono assenti. La direttiva `'Printcap'` può servire a specificare un file diverso dal solito `'/etc/printcap'`, nel caso si desideri conservare un vecchio file del genere, adatto a un sistema di stampa BSD.

Quando si modifica la configurazione del servizio, intervenendo nel file `'/etc/cups/cupsd.conf'`, è necessario fare in modo che il demone `'cupsd'` la rilegga. Per farlo deve ricevere un segnale di aggancio:

```
# killall -HUP cupsd [Invio]
```

#### 27.9.4 Configurazione dal lato dei clienti

« Esiste anche una configurazione che riguarda le richieste di stampa da parte dei clienti, intesi come gli elaboratori o semplicemente i programmi che accodano dei processi di stampa. Il file di configurazione principale è `'/etc/cups/client.conf'`, a cui si possono affiancare dei file personali, di ogni utente, corrispondenti a `'~/cupsrc'`. Generalmente, questi file di configurazione non contengono direttive; tuttavia potrebbero servire per indirizzare la stampa ver-

so un servizio particolare, diverso da uno residente presso lo stesso elaboratore locale.

Per esempio, l'utente che vuole stampare in modo predefinito usando il servizio offerto dall'elaboratore `dinkel.brot.dg`, deve indicare nel proprio file `'~/cupsrc'` una direttiva come quella seguente:<sup>15</sup>

```
ServerName dinkel.brot.dg
```

Questi file di configurazione, utili dal lato cliente, consentono anche di specificare il tipo di comunicazione che ci deve essere: in chiaro o cifrato. Tuttavia, la disponibilità del sistema cifrato dipende dal modo in cui viene compilato Cups, che così potrebbe includere o meno le librerie necessarie.

#### 27.9.5 Utilizzo comune attraverso la riga di comando

« Il servizio di stampa di Cups può essere usato normalmente con i comandi standard di un sistema BSD o System V. La tabella 27.116 riassume le situazioni più comuni a questo riguardo. Tuttavia, si deve tenere presente, in particolare, che una coda di stampa può essere specificata indicando un nome locale, oppure un nome remoto, nella forma:<sup>16</sup>

```
nome@nodo
```

Tabella 27.116. I comandi più comuni per la stampa, come da tradizione.

Comando	Descrizione
<code>lp file</code> <code>lpr file</code>	Invia alla coda di stampa predefinita il file indicato.
<code>lpstat -p</code> <code>lpstat -p -d</code>	Ottiene un elenco delle stampanti logiche disponibili; nel secondo caso si viene a conoscere anche quale risulta essere la stampante logica o la classe predefinita.
<code>lp -d coda file</code> <code>lpr -p coda file</code>	Invia il file alla stampante logica o alla classe indicata.
<code>lp [-d coda] -n n_copia file</code> <code>lpr [-p coda] -#n_copia file</code>	Invia il file alla stampante logica o alla classe indicata, o a quella predefinita, per ottenerne la quantità di copie specificate.
<code>lpstat</code> <code>lpstat -o</code>	Ottiene un elenco dei processi di stampa accodati dall'utente stesso.
<code>lpstat -p</code>	Ottiene un elenco dei processi di stampa attivi appartenenti all'utente.
<code>lpstat -o -p</code>	Ottiene un elenco dei processi di stampa accodati dall'utente e di quelli attivi.
<code>lpr [-E coda]</code>	Ottiene un elenco dei processi di stampa di una coda particolare, oppure di tutte le code.
<code>cancel processo_di_stampa</code> <code>lprm [processo_di_stampa]</code>	Elimina il processo di stampa specificato; nel caso di <code>'lprm'</code> si può omettere, a indicare di intervenire sul primo processo disponibile (quello attivo, oppure il primo di quelli in coda). Il processo di stampa è identificato da una stringa di riconoscimento che può essere consultata con <code>'lpstat'</code> .

Per quanto riguarda `'lpr'` e `'lp'`, occorre considerare che rimane la possibilità di inviare un file alla stampa attraverso lo standard input, anche se dalla tabella ciò non appare evidente.

Ciò che rende speciale Cups è la possibilità di usare i comandi `'lp'` o `'lpr'` con l'opzione `'-o'`, che può anche apparire più volte, attraverso la quale si possono indicare molte altre richieste particolari.

La tabella successiva riepiloga brevemente queste possibilità.

Tabella 27.117. L'opzione '-o' dei comandi 'lpr' e 'lp'.

Comando	Descrizione
-o raw	Disabilita l'intermediazione di qualsiasi filtro, mandando il file direttamente alla stampante.
-o landscape	Stampa in orizzontale, mentre in condizioni normali la stampa avviene in verticale.
-o media={Letter Legal A4↵ ↵ altri_formati}	Stampa utilizzando carta del formato indicato.
-o sides=two-sides-long-edge -o sides=two-sides-short-edge	Stampa fronte e retro con una stampante duplex. La differenza nelle due opzioni sta nel modo in cui viene ruotata la pagina posteriore.
-o sides=one-side	Stampa su un solo lato.
-o job-sheets=↵ ↵{none standard altri_tipi}↵ ↵[,{none standard altri_tipi}]	Richiede di stampare una pagina di separazione. La parola chiave 'none' annulla tale richiesta, mentre altre parole chiave descrivono un tipo particolare di pagina di separazione. Se si indica una parola chiave dopo la virgola, si richiede una pagina di separazione anche alla fine della stampa.
-o page-ranges=↵ ↵n_pag_iniziale[-n_pag_finale]↵ ↵[,intervallo_pagine]...	Seleziona uno o più intervalli di pagine.
-o page-set={odd even}	Seleziona le pagine dispari, oppure quelle pari.
-o number-up=n	Stampa n pagine logiche in una sola pagina reale. La scelta di valori è limitata.
-o brightness=n	Stampa con una luminosità del n %, dove il valore 100 rappresenta la tonalità normale e valori superiori schiariscono in proporzione il risultato, mentre valori inferiori lo anneriscono.
-o gamma=n	Stampa con un contrasto di n/1000, dove il valore 1000 rappresenta la gamma e valori superiori aumentano, mentre valori inferiori diminuiscono in proporzione la gamma.
-o cpi=n	Nel caso di stampa di testo puro, definisce la densità orizzontale di caratteri per pollice.
-o lpi=n	Nel caso di stampa di testo puro, definisce la densità verticale di righe per pollice.
-o columns=n	Nel caso di stampa di testo puro, definisce la quantità di colonne in cui dividere lo spazio della pagina.
-o page-left=n -o page-right=n -o page-top=n -o page-bottom=n	Nel caso di stampa di testo, definisce dei margini alla pagina, rispettivamente sinistro, destro, superiore e inferiore. Il valore che si attribuisce rappresenta una quantità di punti PostScript, dove un punto equivale a 1/72 di pollice.
-o prettyprint	Nel caso di stampa di testo, abbellisce la stampa con un'intestazione; inoltre, se il contenuto viene riconosciuto da Cups, questo viene evidenziato secondo le sue caratteristiche.

Comando	Descrizione
-o position=center -o position=left -o position=top-left -o position=top -o position=top-right -o position=right -o position=bottom-right -o position=bottom -o position=bottom-left	Nel caso di un'immagine, la colloca nella posizione specificata dalla parola chiave che si assegna all'argomento 'position'.
-o scaling=n	Definisce la dimensione di un'immagine, in relazione allo spazio nella pagina. Un valore pari a 100, rappresenta lo spazio massimo a disposizione; valori inferiori producono un'immagine più piccola; valori superiori producono immagini più grandi, in relazione, con la stampa su più pagine.
-o ppi=n	Definisce la densità di punti per pollice della stampa. Maggiore è questo valore, minore diviene di conseguenza quella dell'immagine stampata.
-o natural-scaling=n	Definisce la scala dell'immagine in percentuale.
-o hue=n	Definisce il valore della tinta. Zero corrisponde alla situazione di partenza e si possono attribuire valori in un intervallo che va da -360 a 360.
-o saturation=n	Definisce il valore della saturazione del colore. Il valore esprime una percentuale, dove 100 è la posizione normale, mentre zero corrisponde al solo bianco e nero.

### 27.9.6 Personalizzazione dal lato cliente

Ogni utente può configurare per conto proprio la stampa. Per questo si utilizza il comando 'lpoptions', che si usa fundamentalmente come riepilogato nella tabella 27.118.

Tabella 27.118. Utilizzo del comando 'lpoptions'.

Comando	Descrizione
lpoptions ↵ ↵[-p stampante_logica [/istanza]] ↵ ↵-o opzioni_comuni [-o opzioni_comuni]...	Configura la stampante logica indicata, oppure quella predefinita in sua mancanza, con le opzioni indicate.
lpoptions -d coda	Definisce la coda di stampa predefinita, che può essere una stampante logica (locale o remota), oppure una classe. Nel contempo, mostra le differenze di configurazione rispetto a quanto predefinito.
lpoptions -x coda	Cancella la configurazione associata alla coda indicata.

In particolare, l'opzione '-o' si usa nello stesso modo dei comandi 'lp' e 'lpr', cioè come indicato nelle tabelle ??a2 t cups opzioni o 1?? e ??a2 t cups opzioni o 2??.

### 27.9.6.1 Istanze

Fino a questo punto si è fatto riferimento alle code come a un concetto che include simultaneamente le stampanti logiche e le classi; in pratica, per CUPS le code includono anche le istanze di una stampante logica. Nell'ambito della configurazione delle stampanti, si possono definire delle *istanze*, che rappresentano un modo di distinguere configurazioni differenti per una stessa stampante logica. Si osservi l'esempio seguente:

```
$ lpoptions -p laser/ridotta -o number-up=4 [Invio]
```

In questo modo si vuole definire l'istanza **'ridotta'** che si riferisce alla stampante logica **'laser'**, dove si ottengono quattro pagine logiche per ogni pagina reale, senza modificare la configurazione normale della stampante logica. Per utilizzare questa configurazione, i comandi **'lp'** e **'lpr'** possono fare riferimento all'istanza, completa del nome della stampante:

```
$ lp -d laser/ridotta prova.ps [Invio]
```

Oppure:

```
$ lpr -P laser/ridotta prova.ps [Invio]
```

In questi esempi, si fa riferimento evidentemente alla stampa del file **'prova.ps'**.

### 27.9.7 Accesso remoto

Il protocollo IPP è stato ideato per arrivare fino all'amministrazione remota di una stampante attraverso il protocollo HTTP, dove il servizio di stampa risponde come se fosse un server HTTP, interrogato da un navigatore comune sulla porta TCP 631. CUPS applica in pieno questa filosofia e consente l'amministrazione remota, con le limitazioni definite nel file **'/etc/cups/cupsd.conf'**.

#### 27.9.7.1 Configurazione per l'accesso remoto al server IPP CUPS

Nel file di configurazione **'/etc/cups/cupsd.conf'**, alcune direttive consentono di delimitare l'accesso remoto al servizio IPP di CUPS. Si tratta precisamente di direttive nella forma:

```
<Location percorso>
  direttiva
  ...
</Location>
```

Il percorso rappresenta realmente, o idealmente, una directory che parte dalla posizione stabilita con la direttiva **'DocumentRoot'**, che in condizioni normali corrisponde a **'/usr/share/doc/cups/'**, o simile. Per esempio, una direttiva del tipo **'<Location /prova>'** dovrebbe riferirsi in pratica alla directory o al file **'/usr/share/doc/cups/prova'**.

Molte delle direttive **'Location'** fanno riferimento a directory che in realtà non esistono e hanno un significato particolare. La tabella successiva descrive brevemente il loro scopo. Il controllo dell'accesso a queste directory speciali si traduce in un controllo corrispondente all'accesso delle funzionalità di amministrazione remota.

Tabella 27.119. Directory virtuali controllabili con le direttive **'Location'**.

Directory	Descrizione
<b>'/admin'</b>	Il percorso per le operazioni amministrative.
<b>'/classes'</b>	Il percorso per il controllo delle classi.
<b>'/classes/classe'</b>	Il percorso per il controllo di una classe particolare.
<b>'/jobs'</b>	Il percorso per il controllo dei processi di stampa.
<b>'/printers'</b>	Il percorso per il controllo delle stampanti logiche.
<b>'/printers/stampante_logica'</b>	Il percorso per il controllo di una stampante logica particolare.

Prima di queste direttive, ne appare normalmente una particolare, con cui si definisce il gruppo degli utenti amministratori:

```
SystemGroup lpadmin
```

In questo caso si dichiara che **'lpadmin'** è il gruppo degli amministratori riconosciuti. In pratica, gli utenti che devono amministrare CUPS vengono aggregati al gruppo indicato nel file **'/etc/group'**; tuttavia, non dovrebbe essere necessario fare questo per l'utente **'root'**, che è già parte di tutti i gruppi, per ciò che riguarda CUPS.

Le direttive più comuni che possono apparire nell'ambito della delimitazione di una direttiva **'Location'**, sono elencate brevemente nella tabella successiva.

Tabella 27.120. Direttive utilizzabili nell'ambito di un gruppo **'Location'**, nel file di configurazione **'/etc/cups/cupsd.conf'**.

Direttiva	Descrizione
AuthType {None Basic Digest}	Definisce il modo in cui deve essere identificato l'accesso. La parola chiave <b>'None'</b> indica che non si richiede alcuna autenticazione; <b>'Basic'</b> utilizza un nominativo utente e una parola d'ordine che deve corrispondere a un'utenza del sistema operativo; <b>'Digest'</b> utilizza un nominativo utente e una parola d'ordine che deve corrispondere a un'utenza specifica di CUPS.
AuthClass {Anonymous User System Group}	Limita l'accesso in base all'appartenenza a una categoria: la parola chiave <b>'Anonymous'</b> annulla qualunque richiesta di autenticazione; <b>'User'</b> consente l'accesso a tutti gli utenti che riescono a identificarsi come tali; <b>'System'</b> consente l'accesso agli utenti che appartengono al gruppo definito con la direttiva <b>'SystemGroup'</b> ; <b>'Group'</b> consente l'accesso agli utenti che appartengono al gruppo definito dalla direttiva <b>'AuthGroupName'</b> .
AuthGroupName gruppo	Definisce il gruppo a cui devono appartenere gli utenti che vogliono accedere quando la direttiva <b>'AuthGroup'</b> indica la parola chiave <b>'Group'</b> .
Order {allow,deny deny,allow}	Definisce l'ordine di analisi delle direttive <b>'Allow'</b> e <b>'Deny'</b> . La sequenza <b>'allow,deny'</b> fa sì che l'accesso sia accettato a meno che espressamente rifiutato; la sequenza <b>'deny,allow'</b> fa sì che l'accesso sia rifiutato a meno che espressamente accettato.
Allow modello_nodi	Consente o vieta l'accesso al gruppo di nodi di rete specificato dal modello.
Deny modello_nodi	

L'esempio seguente riguarda un piccolo estratto del file **'/etc/cups/cupsd.conf'**:

#### Listato 27.121.

```
SystemGroup lpadmin

<Location />
Order Deny,Allow
Allow From 127.0.0.1
Allow From 192.168.1.*
</Location>

<Location /admin>
AuthType Basic
AuthClass System
Order Deny,Allow
Allow From 127.0.0.1
Allow From 192.168.1.*
</Location>
```

In questo modo, se non vengono dichiarate altre posizioni, tutto risulta accessibile dall'elaboratore locale (127.0.0.1) e dalla rete 192.168.1.\*; tuttavia, nel caso particolare della posizione '/admin', è richiesta un'autenticazione basata sul riconoscimento dell'utente nel sistema e la sua appartenenza al gruppo di sistema (come stabilito nella direttiva '**SystemGroup**').

### 27.9.7.2 Accesso a una coda remota

« Come già accennato in precedenza, si può fare riferimento a una coda di stampa, con i comandi '**lp**' e '**lpr**', anche nella forma:

```
nome@nodo
```

Tuttavia, l'elaboratore che si intende raggiungere, deve consentire l'accesso. Per questo si deve consentire l'accesso alla posizione '/printers', oppure, in modo più dettagliato, alla posizione '/printers/nome'. Nell'esempio del listato 27.121, la posizione '/printers' eredita la configurazione da quella della radice:

```
<Location />
Order Deny,Allow
Allow From 127.0.0.1
Allow From 192.168.1.*
</Location>
```

In questo modo, si consente l'accesso dalla rete 192.168.1.\*, senza richiedere alcuna autenticazione.

Quando una classe o una stampante logica di un elaboratore remoto è accessibile a quello locale, '**lpstat**' segnala la disponibilità di queste code, come se si trattasse di stampanti locali. Tuttavia, perché ciò accada è necessario che sia stata concessa l'interrogazione con le direttive '**Brows\***', già descritte in precedenza.

Inoltre, se più elaboratori mettono a disposizione una stampante logica con uno stesso nome, si può creare una classe implicita, con quello stesso nome, a meno che ciò sia stato disabilitato espressamente nel file di configurazione.

### 27.9.7.3 Autenticazione digest

« Nel file di configurazione '/etc/cups/cupsd.conf', all'interno delle direttive che delimitano l'accesso a una posizione particolare, la direttiva '**AuthType**' consente di richiedere e specificare il tipo di autenticazione. Il tipo '**Basic**' richiede che l'utente si identifichi attraverso il proprio nominativo e la parola d'ordine usati normalmente per accedere al sistema. Al contrario, il tipo '**Digest**' richiede un nominativo e una parola d'ordine annotati nel file '/etc/cups/passwd.md5'.

Il principio di funzionamento di questo file è simile a quello usato dai sistemi Unix nel file '/etc/passwd', con la differenza che la parola d'ordine viene cifrata con un algoritmo differente (MD5, come suggerisce l'estensione del file). Per aggiungere utenti di tipo '**Digest**' e per rimuoverli, si usa il comando '**lppasswd**'.

Tabella 27.123. Il comando '**lppasswd**'.

Comando	Descrizione
<code>lppasswd -a <i>nominativo</i> [-g <i>gruppo</i>]</code>	Crea un nominativo e lo associa eventualmente a un gruppo.
<code>lppasswd -x <i>nominativo</i></code>	Elimina un nominativo.

Si crea un nominativo utente con l'opzione '-a', in un modo simile a quello seguente:

```
# lppasswd -a pippo [Invio]
```

```
Enter password ***** [Invio]
```

```
Enter password again ***** [Invio]
```

Se la parola d'ordine viene ripetuta correttamente e soddisfa i requisiti minimi imposti da '**lppasswd**' (non deve assomigliare al nominativo e deve essere sufficientemente varia), si ottiene l'ag-

giunta di una riga nel file '/etc/cups/passwd.md5', che potrebbe assomigliare a quella seguente:

```
pippo:sys:d091266158b6c7620c306db2ece25e42
```

Se si usa anche l'opzione '**-g**', come mostrato nella tabella 27.123, si può intervenire anche nel secondo campo, che descrive il nome del gruppo.

Se la richiesta di autenticazione '**Digest**' si integra con la direttiva '**AuthClass**' e viene richiesta l'appartenenza a un gruppo particolare (che può essere quello stabilito dalla direttiva '**AuthGroupName**', oppure '**SystemGroup**', il gruppo che si associa all'utente deve esistere realmente nel sistema Unix e corrispondere a quello richiesto; diversamente l'autenticazione fallirebbe.

### 27.9.7.4 Problemi di accesso attraverso un proxy

« Se ci si avvale di un proxy HTTP, può diventare impossibile l'accesso al servizio di Cups. Per prima cosa, l'indirizzo corrispondente al proxy deve essere ammesso ad accedere; inoltre, ci possono essere dei problemi con la porta TCP: se si consente a Cups di usare la porta 80, può darsi che il proxy riesca ad accedere, ma non è garantito che poi si possano raggiungere le posizioni che richiedono una forma di autenticazione.

### 27.9.8 Compatibilità con il sistema di stampa BSD

« La compatibilità con il sistema di stampa BSD, si attua in due modi. Da un lato, viene mantenuto il file '/etc/printcap' (salvo che gli si voglia cambiare nome per qualche motivo); dall'altro è possibile attivare un demone, '**cups-lpd**', che è in grado di accettare richieste di stampa secondo il vecchio protocollo. Per attivare il demone, occorre intervenire nel file di configurazione '/etc/inetd.conf', aggiungendo un record simile a quello seguente:

```
printer stream tcp nowait lp /usr/lib/cups/daemon/cups-lpd cups-lpd
```

Naturalmente, ci si deve poi prendere cura di informare il supervisore dei servizi di rete (di solito l'eseguibile '**inetd**'), inviando un segnale di aggancio:

```
# killall -HUP inetd [Invio]
```

Inoltre, per poter utilizzare un vecchio servente conforme al protocollo usato dal sistema di stampa BSD, si può farvi riferimento con URI del tipo:

```
lpd://nodo/coda_remota
```

### 27.9.9 Considerazioni finali

« Cups è un sistema di stampa molto complesso e anche molto comodo; tuttavia, il carico elaborativo che si crea nel momento in cui viene accodato un processo di stampa, è molto elevato. In pratica, se l'elaboratore in cui si attiva un servizio di stampa con Cups non ha la potenza sufficiente, si possono generare dei ritardi che inizialmente sembrano inspiegabili; inoltre, se l'elaboratore è particolarmente carente di risorse, si rischia lo stallo del sistema operativo.

### 27.10 Riferimenti

- L. McLaughlin III, *RFC 1179: Line Printer Daemon Protocol*, 1990, <http://www.ietf.org/rfc/rfc1179.txt>
- Till Kamppeter, Kurt Pfeifle, *Some theoretical background: CUPS, PPDs, PostScript and GhostScript*, <http://www.openprinting.org/download/kpfeifle/LinuxKongress2002/Tutorial/III.PostScript-and-PPDs.html>
- OpenPrinting, <http://www.linuxfoundation.org/collaborate/workgroups/openprinting>
- Easy Software Products, *Common UNIX printing system*, <http://www.cups.org/>
- PWG, *Internet Printing Protocol*, <http://www.pwg.org/ipp/>

- S. Zilles, *RFC 2568: Rationale for the Structure of the Model and Protocol for the Internet Printing Protocol*, 1999, <http://www.ietf.org/rfc/rfc2568.txt>

<sup>1</sup> **BSD line printing spool system** UCB BSD

<sup>2</sup> **LPRng** GNU GPL o Artistic

<sup>3</sup> **Rlpr** GNU GPL

<sup>4</sup> Se si fanno modifiche al file `/etc/printcap` bisogna ricordare di inviare un segnale di aggancio al demone `lpd` per fare in modo che venga riletto questo file: `kill -s SIGHUP pid_di_lpd`.

<sup>5</sup> **HPLJS** GNU GPL, BSD e MIT

<sup>6</sup> **Apsfilter** GNU GPL

<sup>7</sup> **PSUtils** licenza speciale formulata in modo poco preciso

<sup>8</sup> **Magicfilter** GNU GPL

<sup>9</sup> **Foomatic** GNU GPL

<sup>10</sup> La sigla «RIP» sta per *Raster image processor*.

<sup>11</sup> **A2ps** GNU GPL

<sup>12</sup> **Enscript** GNU GPL

<sup>13</sup> **Cups** GNU GPL

<sup>14</sup> Il protocollo IPP utilizza a sua volta il protocollo HTTP.

<sup>15</sup> Si rammenta che per il momento non è ancora stato spiegato come fare per abilitare l'accesso alle stampanti locali da parte di nodi di rete remoti.

<sup>16</sup> Come già chiarito in precedenza, per coda di stampa si fa riferimento indifferentemente a una classe, una stampante logica o a un'istanza di una stampante logica.