

## Dischetti di emergenza per sistemi Linux Creazione ed utilizzo di dischetti di emergenza

Questo approfondimento vuole introdurre gli strumenti essenziali per il recupero di un sistema *Linux* dopo un eventuale *crash*, cioè i dischetti di avvio del sistema.

È sempre meglio cautelarsi preventivamente circa l'eventuale malfunzionamento del sistema attraverso la predisposizione di appositi dischetti di emergenza. Infatti, la realizzazione o la personalizzazione dei dischetti di emergenza in ambiente *Linux*, è sintomo di una seria e cauta amministrazione di sistema e questi, in alcuni casi, possono essere l'unico mezzo per potere intervenire sul sistema dopo un eventuale malfunzionamento.

### Tipi di dischetti di emergenza

I dischetti di emergenza vengono utilizzati nel caso in cui il sistema operativo non effettui il *boot* in maniera corretta e possono essere di diversi tipi:

- **boot disk**: è un dischetto che contiene il **kernel** di *Linux* per consentire la partenza del sistema. Su tale disco però manca un *root file system* che verrà montato da un altro *device* che può essere, a sua volta, un dischetto o un disco rigido;
- **root disk**: è un dischetto contenente un *root file system* (un *root file system* è un *file system* che viene montato direttamente sotto la *root directory /*) con tutti i *file* principali che servono a *Linux*;
- **boot/root disk**: è un dischetto che contiene sia il **kernel** che il *root file system*, risultando quindi autonomo per l'esecuzione di *Linux*;
- **utility disk**: è un dischetto con programmi di utilità aggiuntivi rispetto al *root disk*, che possono essere di ausilio per il *recovery* del sistema.

Nel gergo comune spesso si intende con il termine generico di *boot disks* sia il *boot disk* vero e proprio che il *root disk*.

### Utility disk

In un *utility disk* normalmente vengono posti programmi per manipolare ed esaminare dischi e *file system* (come *format*, *fdisk*, *mke2fs*, *fsck*, *debugfs*, ...), *editor* di testo (come *vi*, *elvis*, *pico*, ...), utilità di archiviazione e compressione (*gzip*, *tar*, ...) ed altre utilità di comunicazione o per l'utilizzo di particolari *device*, necessarie al *recovery* del sistema danneggiato.

### Boot disk

Creare un *boot disk* può servire non solo in caso di *recovery* del sistema, ma anche, per esempio, per testare una nuova versione di **kernel**, montando come *root file system* quello presente nel disco rigido del nostro sistema.

È possibile creare anche una versione con settore di avvio, ma senza la presenza del **kernel**. Il calcolatore si avvia e il programma contenuto nel primo settore ricerca il **kernel** e gli altri *file* necessari per il corretto caricamento del sistema, nel disco fisso (esempio: */boot/...*). Se il **kernel** e gli altri *file* di sistema sono stati spostati o sono corrotti, tale dischetto non servirà a portare a buon fine la procedura di emergenza. In effetti tale procedura ci permette di verificare una configurazione di *lilo* senza interferire con l'**MBR** del disco fisso configurando ad esempio il *file* */etc/lilo.conf* come segue:

```
boot=/dev/fd0  
prompt
```

```
timeout=60
image=/boot/vmlinuz
label=linux
root=/dev/hda1
read-only
```

Questo tipo di dischetto può essere utile in caso solo di corruzione del **MBR** del disco fisso (e non di altre parti fondamentali del *file system*) o, come appena detto, per testare in modo sicuro diverse configurazioni di `lilo`.

### Boot/root disk

Diversa è la situazione di un **boot/root disk** in cui il dischetto presenta sia il settore di avvio che il **kernel**. Tale *floppy* conterrà un *file system* nel quale sarà stata copiata sia la *directory* `/boot/`, con il suo intero contenuto, che la *directory* `/etc/` con il *file* `lilo.conf`, che la *directory* `/dev/` con il dispositivo `fd0`, più altri *file* di dispositivi che servono per la corretta individuazione dei dischi e delle partizioni.

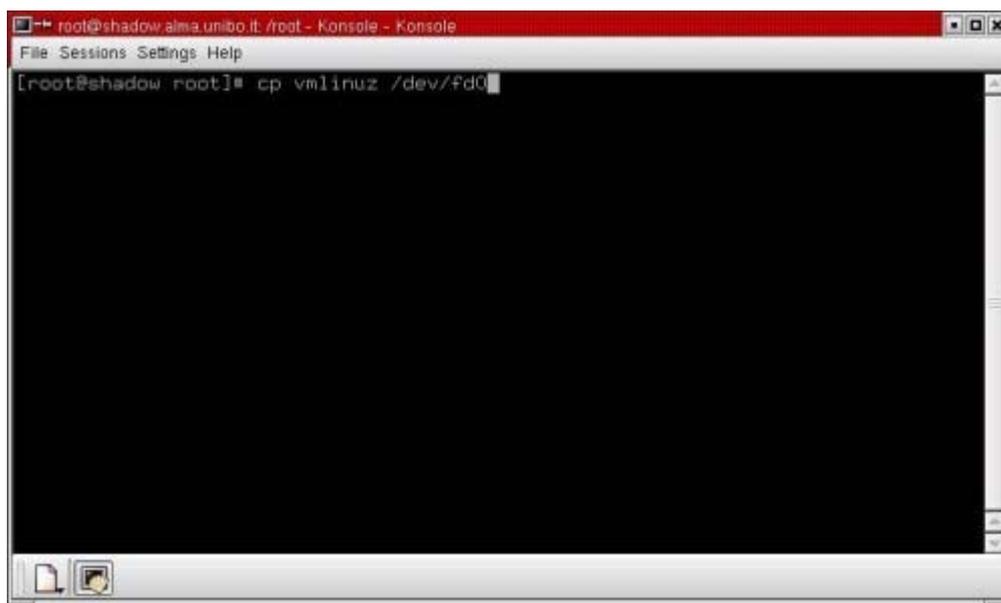
Questo sistema di avvio consente di specificare, sia attraverso la configurazione del *file* `/etc/lilo.conf`, che per mezzo del *prompt* di *boot*, alcuni parametri di avvio particolari che potrebbero risultare utili al proprio sistema, ed è anche per questo che viene considerato uno strumento di avvio completo.

### Altro sistema di avvio

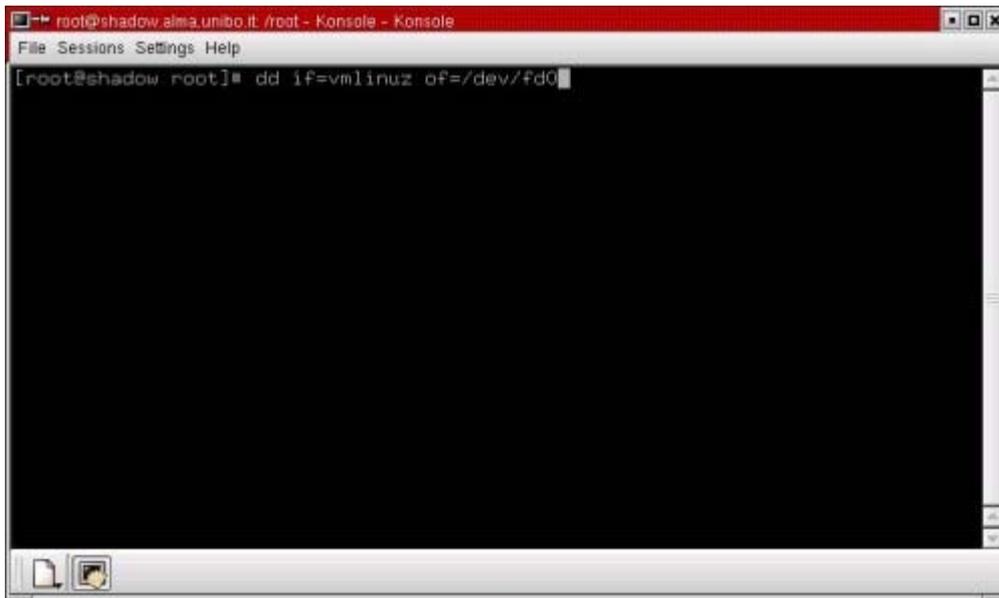
Un altro sistema di avvio è quello tramite un dischetto contenente l'intera immagine del **kernel** senza fare uso di alcun *file system* (**boot disk**).

È il metodo più sicuro e per effettuare la copia del *file kernel* sul *floppy* si può eseguire uno tra i seguenti comandi:

```
#cp vmlinuz /dev/fd0
```



```
# dd if=vmlinuz of=/dev/fd0
```



Il programma 'rdev'

Il **kernel** riesce ad avviare se stesso, ma non è detto che sappia quale dispositivo contenga il *file system* principale da montare all'avvio. Per inserire queste, ed altre, informazioni, si utilizza il programma `rdev`.

`rdev` è un programma che permette di modificare, all'interno di un'immagine del **kernel** di *Linux*, le informazioni relative alla periferica contenente il *file system root*, la modalità del video e la dimensione del RAM *disk*.

Queste informazioni sono rappresentate da coppie di *bytes* che partono dall'**offset** decimale 504 dell'immagine del **kernel**, secondo quanto riportato nella seguente tabella:

<b>Offset (decimale)</b>	<b>Descrizione</b>
498	<i>Root flags</i>
500	<i>Reserved</i>
502	<i>Reserved</i>
504	<i>RAM Disk Size</i>
506	<i>VGA Mode</i>
508	<i>Root Device</i>
510	<i>Boot Signature</i>

Sintassi del comando 'rdev'

La sintassi del comando `rdev` è la seguente:

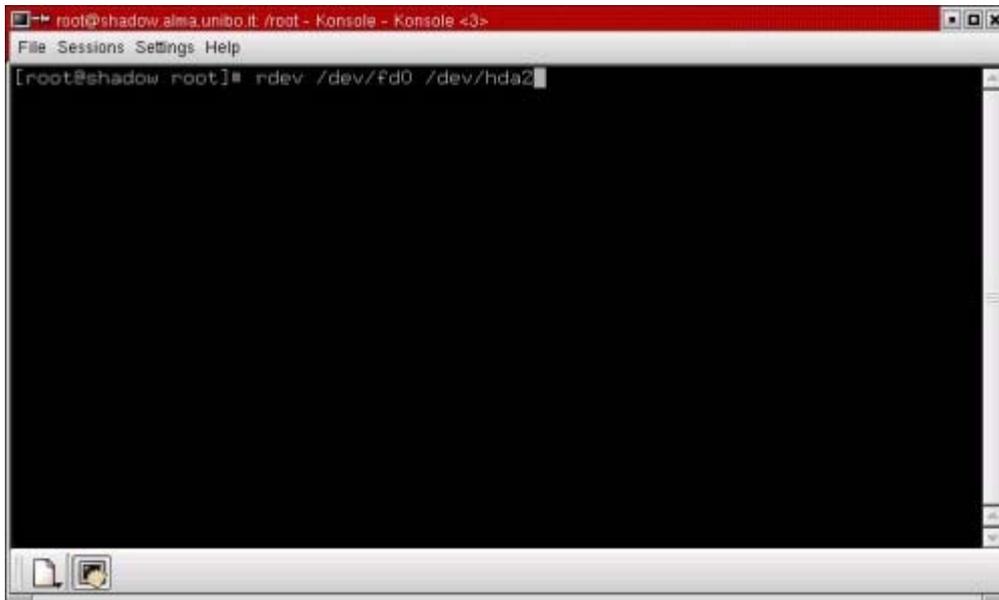
```
rdev [ -rRvh ] [ -o offset ] [ image [ value [ offset ] ] ]
```

Il parametro `-R` serve per modificare i *Root flags* che contengono informazioni aggiuntive da usare nel momento in cui viene montato il *root file system*. Attualmente l'unico significato dei *Root flags* è quello di avere l'effetto di montare il *root file system* in modalità di sola lettura quando questi abbiano un valore diverso da 0.

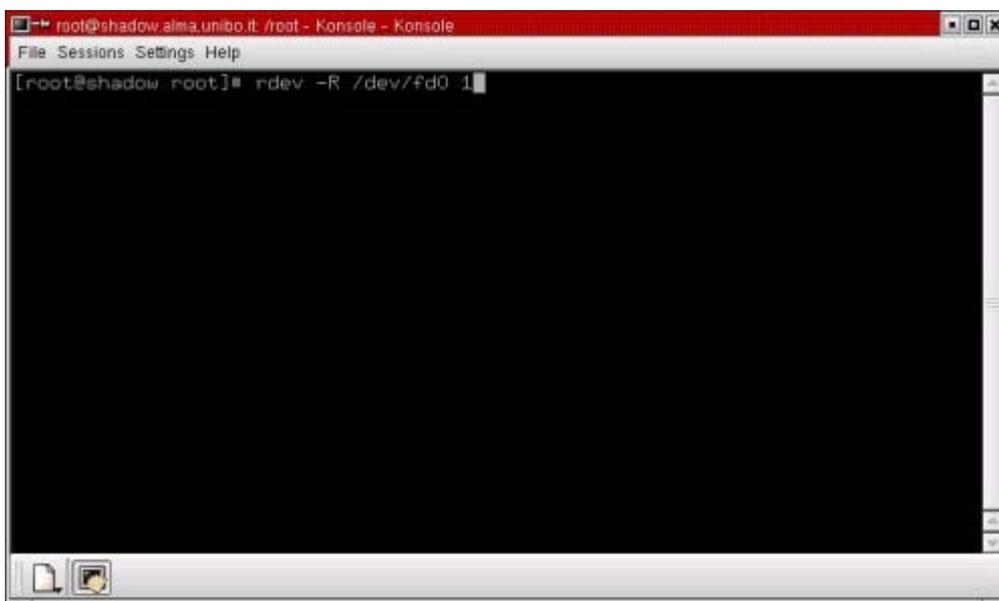
In base a quanto appena detto, ad esempio, per modificare l'immagine del **kernel** prima copiata nel

*floppy* /dev/fd0 per fornirgli l'informazione che il *file system root* è nella partizione /dev/hda2 del disco fisso hda, e per montare quest'ultima inizialmente solo in lettura, si possono eseguire i seguenti comandi:

```
# rdev /dev/fd0 /dev/hda2
```



```
# rdev -R /dev/fd0 1
```



Altri dischetti di emergenza

In caso di malfunzionamento del calcolatore, ci si potrebbe avvalere anche dei dischetti di avvio già pronti delle varie distribuzioni *GNU/Linux*. Anche tali *floppy* rientrano nella categoria di emergenza e possono consentire un accesso al disco fisso per effettuare copie di *backup* o piccole riparazioni. Tra quelli più semplici ed efficaci consideriamo quelli della distribuzione *Slackware*. Questi sono reperibili presso i seguenti indirizzi Internet:

<http://distro.ibiblio.org/pub/Linux/distributions/slackware/slackware-current/rootdisks/>

<http://distro.ibiblio.org/pub/Linux/distributions/slackware/slackware-current/bootdisks/>

I dischi di avvio di altre distribuzioni possono essere recuperati alle URL corrispondenti.

Ne riportiamo di seguito alcune :

Distribuzione	URL
<i>Red Hat</i>	<a href="ftp://ftp.redhat.com/pub/redhat/linux/current/en/os/i386/images/">ftp://ftp.redhat.com/pub/redhat/linux/current/en/os/i386/images/</a>
<i>Debian</i>	<a href="ftp://ftp.debian.org/debian/dists/stable/main/disks-i386/current/">ftp://ftp.debian.org/debian/dists/stable/main/disks-i386/current/</a>
<i>Mandrake</i>	<a href="http://www.linux-mandrake.com/en/ftp.php3">http://www.linux-mandrake.com/en/ftp.php3</a>

I dischi di emergenza della distribuzione *Slackware*, per essere avviati, richiedono un dischetto di *boot* contenente il *kernel*. Tale dischetto può essere predisposto localmente, in modo da avere a disposizione la configurazione più adatta al proprio sistema.

Il dischetto più efficace per la risoluzione dei problemi è quello che usa l'immagine compressa *root/rescue.gz* che contiene un *file system* della dimensione di alcuni *Mbyte* in formato *Second-extended (Ext2)*. Per poterlo usare è necessario disporre di una certa quantità di memoria RAM da riservargli.

#### Dischetti personalizzati

Si può rendere necessario anche l'utilizzo di dischetti realizzati appositamente, qualora le situazioni previste dai classici dischetti di avvio o da quelli di recupero forniti da una distribuzione non riescano ad individuare le caratteristiche e le esigenze particolari del sistema (esempio: *controller* dei dischi particolari).

Una prima personalizzazione effettuabile è quella sul *kernel* attraverso l'attivazione della gestione diretta delle immagini tramite *loopback device support*. Un *loopback device* di *Linux* è una periferica virtuale (ha un nome del tipo */dev/loop0*, */dev/loop1*, ...) a cui è possibile associare un *file* presente in un altro *file system* per poterlo trattare come un qualunque altro *device* tipo */dev/hda*, */dev/fd0* e quindi, eventualmente, potergli associare un *file system* formattandolo e montandolo su di una *directory* del proprio sistema.

Questa consente di montare un'immagine non compressa di un *floppy* con un comando del tipo:

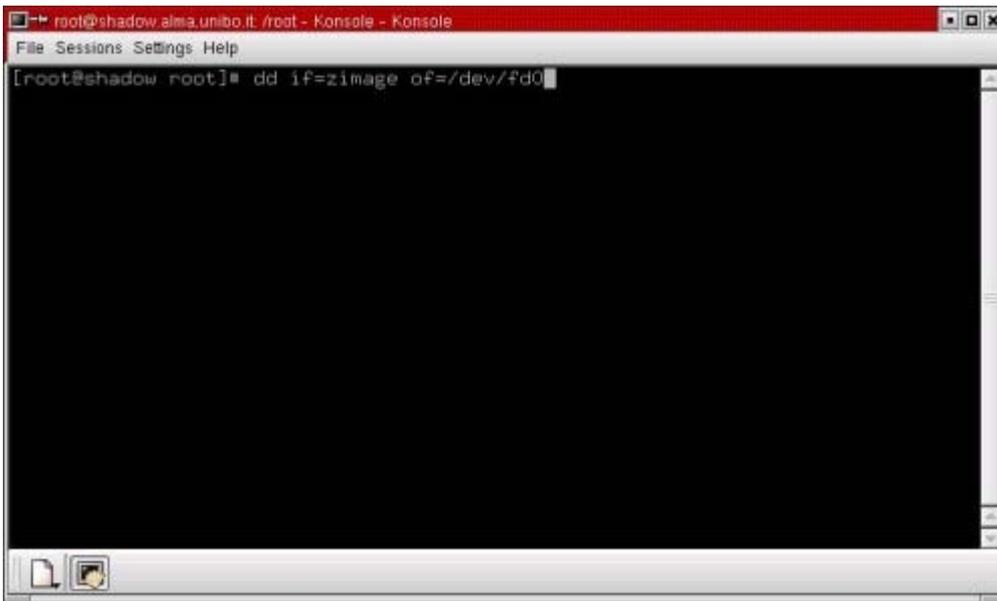
```
# mount -o loop -t <file-system> <immagine> <punto di innesto>
```

Una volta montato in questo modo il *file system*, presente nel dischetto di avvio della distribuzione, è possibile personalizzarne il contenuto potendovi accedere come un qualunque altro *file system* montato nel nostro sistema.

#### Dischetti personalizzati e dischetti di distribuzione Slackware

È altresì possibile combinare l'uso di dischetti personalizzati con quelli forniti dalla distribuzione *Slackware*; ad esempio si potrebbe desiderare di abbinare un *kernel* personalizzato ad un dischetto contenente il sistema minimo della distribuzione sopra citata. Il modo più semplice per realizzare tale combinazione, per esempio, è quello di copiare l'immagine personalizzata del *kernel* (*zimage*) su di un dischetto con il comando *dd*, come avevamo visto in precedenza, e poi intervenire con il programma *rdev*, ovvero

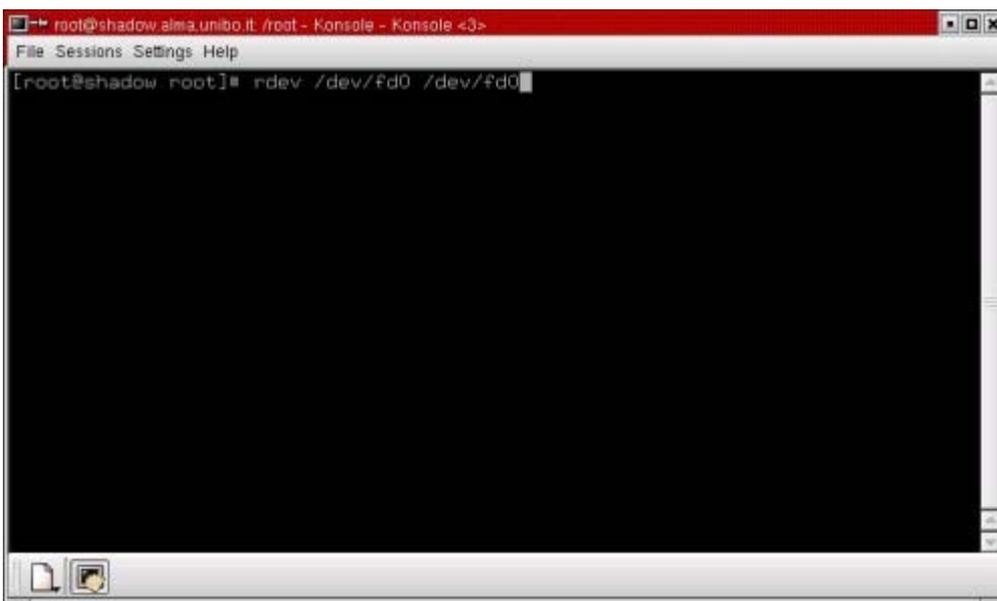
```
# dd if=zimage of=/dev/fd0
```



```
root@shadow alma.unibo.it /root - Konsole - Konsole
File Sessions Settings Help
[root@shadow root]# dd if=zimage of=/dev/fd0
```

Viene così generato un dischetto di avviamento dove risiede un *kernel*, che deve però essere informato da dove e come fare il caricamento del sistema. Poiché il *file system* principale viene caricato da un dischetto, l'informazione al *kernel* viene data attraverso il seguente comando:

```
# rdev /dev/fd0 /dev/fd0
```



```
root@shadow alma.unibo.it /root - Konsole - Konsole <3>
File Sessions Settings Help
[root@shadow root]# rdev /dev/fd0 /dev/fd0
```

### File system in memoria RAM

Per poter utilizzare i dischetti di emergenza, è quasi obbligatorio montare il *file system* principale facendolo risiedere in memoria RAM. Questo procedimento è possibile sfruttando un altro tipo di periferica virtuale di *Linux* chiamata *RAM disk*. Un disco RAM è una porzione di memoria RAM dedicata ad essere montata come un *file system*. Viene riferita mediante nomi del tipo `/dev/ram0`, `/dev/ram1`,... e può essere trattata allo stesso modo delle *loopback device*, cioè formattata, montata, eccetera...

Usare un disco RAM è vantaggioso dal punto di vista delle prestazioni del sistema in quanto la memoria volatile è molto più veloce sia dei *floppy disk* che degli *hard disk* e consente di costruire un disco di avvio che contiene un'immagine compressa, in modo che sia possibile memorizzarvi una quantità di informazioni superiore a quella che consentirebbe la capacità di un dischetto nella forma

non compressa. Al momento del caricamento del sistema il **kernel** si preoccupa di espandere nel disco RAM l'immagine compressa del *floppy* e di montarla.

Avere trasferito in RAM il *file system* che risiedeva sul dischetto, consente di rimuovere il dischetto stesso subito dopo il caricamento, in modo da poter riutilizzare l'unità a dischetti per accedere ad altri programmi di utilità non presenti nel disco RAM.

Per gestire il disco RAM il **kernel** deve essere opportunamente configurato.

Nel caso in cui il **kernel** e l'immagine da caricare siano contenute in uno stesso dischetto, il **kernel** deve conoscere la posizione (**offset**) di inizio dell'immagine.

Queste informazioni vengono impostate usando il comando `rdev` con *switch* `-r` (`rdev -r ...`) per modificare i due *bytes* che costituiscono la *RAM Disk word* presente nell'immagine del **kernel**.

RAM Disk word

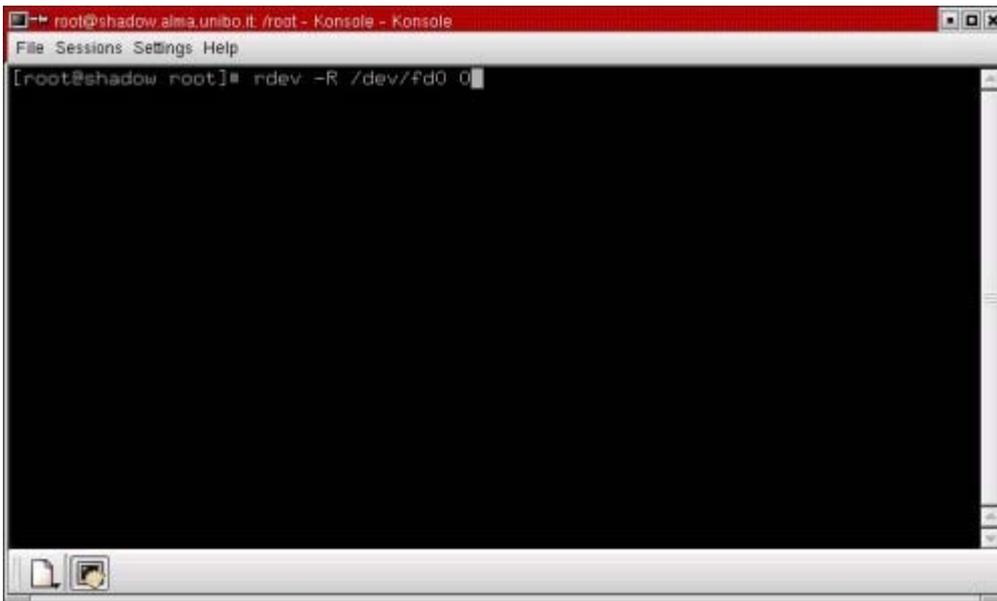
Il contenuto della *RAM Disk word* va interpretato secondo la seguente tabella:

<b>Bits</b>	<b>Descrizione</b>
0-10	<b>Offset</b> di inizio dell'immagine del <i>RAM Disk</i> in blocchi da 1024 <i>bytes</i>
11-13	Non usati
14	<i>Flag</i> indicante se il <i>RAM Disk</i> debba essere caricato
15	<i>Flag</i> che indica che deve essere richiesto l'inserimento di un secondo dischetto prima che venga montato il <i>root file system</i>

Nel caso si stia costruendo un *boot/root disk*, cioè un dischetto contenente sia il **kernel** che un'immagine compressa del *root file system*, avremo il bit 15 a 0, in quanto non occorre inserire un secondo dischetto, il bit 14 a 1, perché il *root file system* va decompresso e caricato nel *RAM Disk* ed i bit da 0 a 10 che ci indicheranno, in *kilobytes*, l'**offset** di inizio dell'immagine del *root file system* rispetto all'inizio del dischetto. Tale **offset** indicherà il primo blocco successivo a quelli occupati dal **kernel**, essendo l'immagine da caricare nel disco RAM posta contigualmente a quella del **kernel**, nel dischetto.

Nel caso della distribuzione *Slackware* i dischetti contenenti il sistema minimo (*boot disk*) non prevedono il controllo del *file system* e il successivo montaggio in lettura e scrittura. Per montare il *file system* principale in lettura e scrittura si deve fare uso, come visto in precedenza, del comando `rdev`:

```
# rdev -R /dev/fd0 0
```



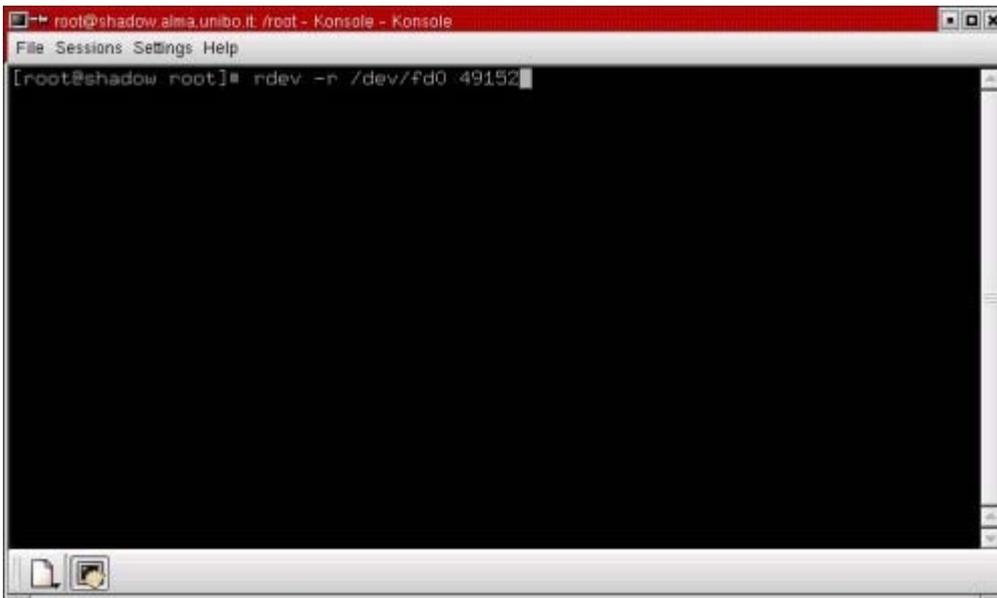
Oltre a questa informazione è necessario specificare se l'immagine contenente il sistema si trova in un dischetto separato e se si vuole quindi avere una segnalazione dal sistema quando è stato caricato il dischetto con il **kernel** e deve essere inserito il dischetto con il *root file system*. Infine va anche specificato se si vuole caricare l'immagine in un disco RAM.

Per fare questo occorre usare la serie di bit specificati nella tabella precedente relativa alla RAM *Disk word*, che sono da configurare opportunamente utilizzando lo *switch* `-r` del comando `rdev`. Come si vede dalla tabella i primi 11 bit della *word* permettono di definire lo scostamento in blocchi da 1024 *byte* dell'immagine del sistema, il bit 14 se impostato a 1 impone al sistema di caricare un disco RAM e infine il bit 15 impone che il sistema segnali quando si deve eseguire lo scambio di *floppy*.

Se, per esempio, si vuole usare un disco RAM con scambio di dischetti, la RAM *Disk word* dovrà valere  $2^{15}+2^{14}+0=49152$ . I bit relativi all'**offset** dell'immagine del *root file system* sono a 0 perché risiedendo tale immagine in un secondo dischetto, diverso da quello ove risiede il **kernel**, potrà iniziare dal primo blocco disponibile, cioè quello con **offset** 0.

Per impostare questo valore si farà uso del comando `rdev` come di seguito riportato:

```
#rdev -r /dev/fd0 49152
```



```

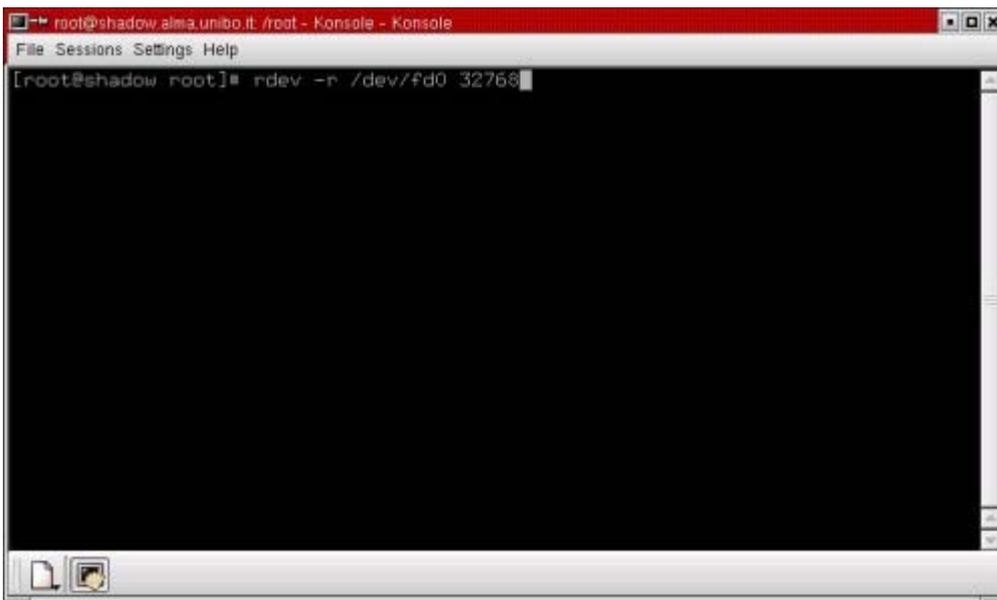
root@shadow alma.unibo.it /root - Konsole - Konsole
File Sessions Settings Help
[root@shadow root]# rdev -r /dev/fd0 49152

```

RAM Disk word senza uso di disco RAM

Nel caso di scambio di dischetti senza uso di disco RAM il valore della RAM *Disk word* sarà  $2^{15}+0+0=32768$  e quindi il comando `rdev` da usare risulterà:

```
#rdev -r /dev/fd0 32768
```



```

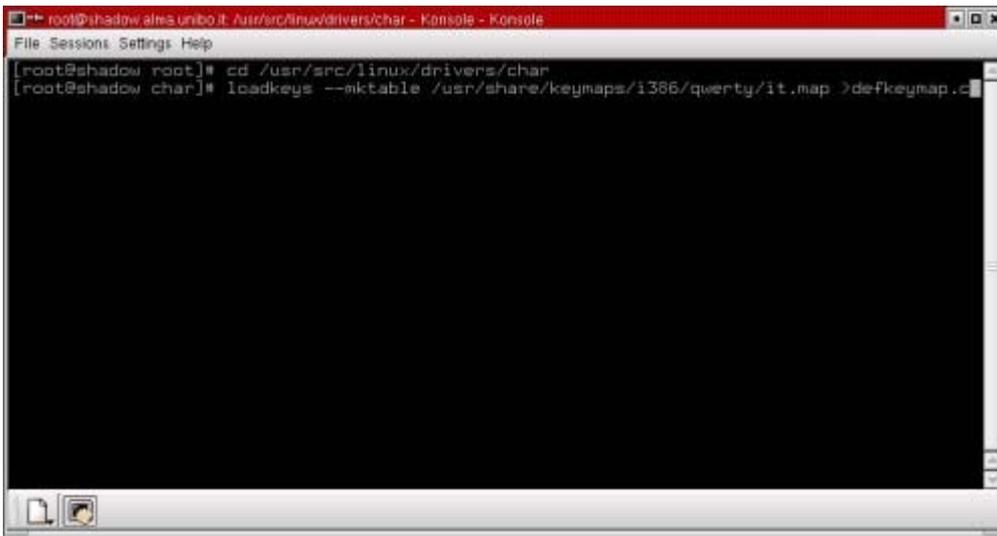
root@shadow alma.unibo.it /root - Konsole - Konsole
File Sessions Settings Help
[root@shadow root]# rdev -r /dev/fd0 32768

```

I dischetti di emergenza standard sono predisposti all'utilizzo di una tastiera USA, ma è possibile risolvere tale limitazione lavorando direttamente sul **kernel**. Una volta trovato, per esempio, il *file* `it.map` della mappa della tastiera italiana, si genera il sorgente `defkeymap.c` nel seguente modo:

```
# cd /usr/src/linux/drivers/char
# loadkeys --mktable /usr/share/keymaps/i386/qwerty/it.map > defkeymap.c
```

(esempio: valido nel caso di piattaforma i386)



```
root@shadow: /usr/src/linux/drivers/char - Konsole - Konsole
File Sessions Settings Help
[root@shadow root]# cd /usr/src/linux/drivers/char
[root@shadow char]# loadkeys --mktable /usr/share/keymaps/1386/qwerty/it.map >defkeymap.c
```

In questo modo, alla compilazione successiva del *kernel*, questi utilizzerà la mappa italiana come predefinita e non sarà più necessario utilizzare il programma `loadkeys`.

I *file* della mappa della tastiera italiana sono normalmente situati in una delle seguenti *directory*:

```
/usr/share/keymaps/(piattaforma)/qwerty/  
/usr/lib/kbd/keymaps/(piattaforma)/qwerty/
```

### Emergenza in presenza di una rete locale

Quando siamo in presenza di una rete locale, invece, i problemi di emergenza possono essere trattati in modo diverso, visto che se anche si è verificato un danno su di un elaboratore, normalmente questo non dovrebbe essersi manifestato anche in tutti gli altri presenti nella LAN.

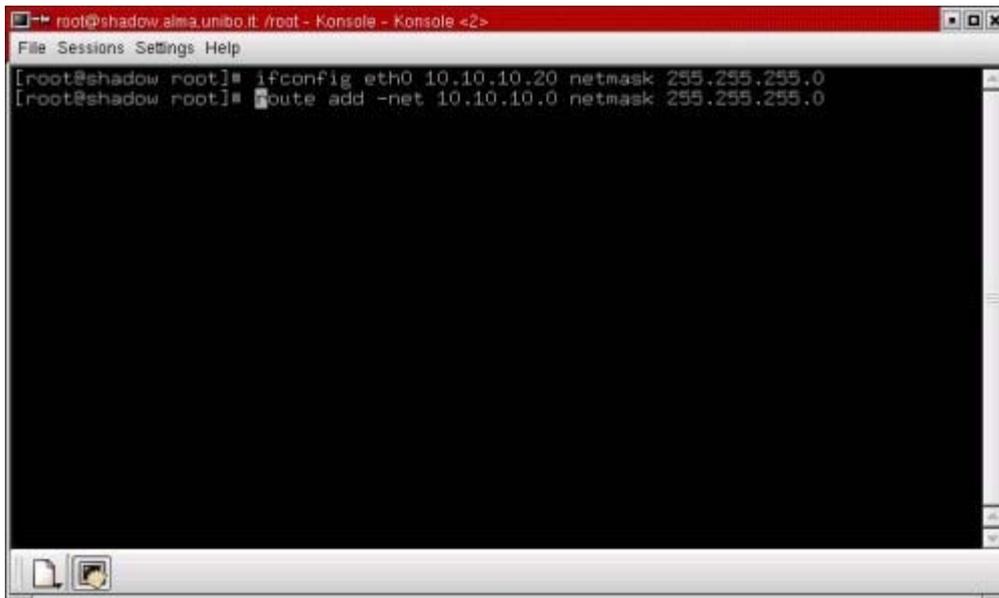
In uno scenario di questo tipo non è necessario configurare i dischetti di emergenza in modo da avere a disposizione in un mini *file system* tutte le utilità che possono servire per il *recovery* del sistema, ma è sufficiente che dispongano di un **kernel** con supporto della rete e del *file system* NFS, in modo da riuscire a raggiungere il *file system* di un calcolatore operante in rete, montandone un direttorio tramite il servizio NFS per copiare da e verso questo archivi e programmi.

Con una connessione **Ethernet**, di solito riconosciuta dal sistema come `eth0`, è possibile configurare e testare la scheda di rete in modo da raggiungere e montare il *file system* di un sistema remoto.

Supponiamo, per esempio, che il calcolatore in panne abbia indirizzo IP 10.10.10.20 e *netmask* 255.255.255.0 ed il *server* di cui voglio montare il direttorio abbia indirizzo IP 10.10.10.1 e *netmask* 255.255.255.0.

Di seguito vengono riportati i comandi necessari alla configurazione della scheda di rete sul *computer* avviato con i dischetti di emergenza.

```
# ifconfig eth0 10.10.10.20 netmask 255.255.255.0  
# route add -net 10.10.10.0 netmask 255.255.255.0
```

A terminal window titled 'root@shadow alma.unibo.it /root - Konsole - Konsole <2>' with a menu bar 'File Sessions Settings Help'. The terminal shows two commands: '[root@shadow root]# ifconfig eth0 10.10.10.20 netmask 255.255.255.0' and '[root@shadow root]# route add -net 10.10.10.0 netmask 255.255.255.0'. The window has a standard Linux desktop environment with a taskbar at the bottom.

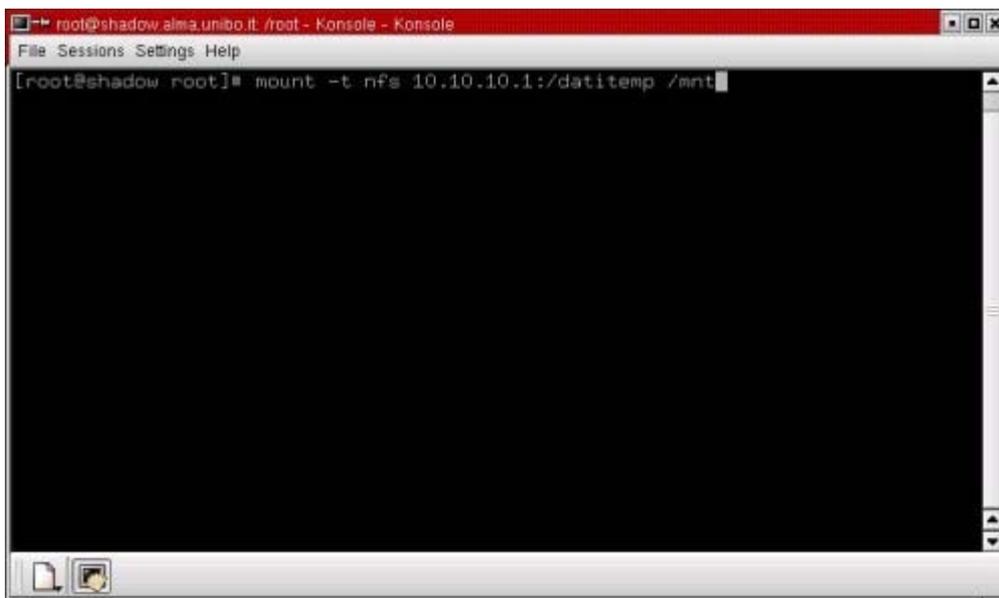
Una volta configurata la scheda di rete è possibile verificare la connettività verso il *server* NFS usando il comando *ping* (vedi **primo approfondimento**).

```
# ping 10.10.10.1
```

Una volta verificata positivamente la connettività, occorre montare il direttorio esportato dal *server* (esempio: /datitemp) che poi potremmo usare per salvare dati o lanciare i programmi di utilità che avremmo avuto cura di riporvi in precedenza, procedendo così nella fase di *recovery* del sistema.

Il comando per montare il direttorio che ci interessa è il seguente:

```
# mount -t nfs 10.10.10.1:/datitemp /mnt
```

A terminal window titled 'root@shadow alma.unibo.it /root - Konsole - Konsole' with a menu bar 'File Sessions Settings Help'. The terminal shows the command '[root@shadow root]# mount -t nfs 10.10.10.1:/datitemp /mnt'. The window has a standard Linux desktop environment with a taskbar at the bottom.

Se non si dispone di una scheda di rete è possibile simularla per mezzo delle porte parallele utilizzando il protocollo PLIP.

I **referimenti bibliografici** *on line* consentono di svolgere ulteriori attività di approfondimento.