

Connettività con sistemi Dos



Dos IPv4	803
Driver di pacchetto	805
Libreria WATTCP	807
Applicazioni standard per WATTCP	809
ABC-nslookup	810
MiniTelnet	812
SSHDOS	813
Bobcat (Lynx)	814
PPRD: servente di rete per la stampa	815
Trout	818
Talk	818
DosLynx	819
NCSA Telnet	821
POPMail	826
PCroute	830
Riferimenti	834
Dos PPP	837
Composizione	837
Configurazione e script	838
Connessione in pratica	841
Introduzione a NOS-KA9Q -- IPv4 per Dos	843

Preparazione	845
Interfacce, instradamento e nomi	851
Gestione delle sessioni	857
Attività nel sistema locale	859
Gestione della rete e delle connessioni	860
NOS come cliente	862
NOS come servente	864
NOS come router IPv4	867
nanoDos	869
Organizzazione della distribuzione	869
Installazione	870
Configurazione	871
Utilizzo	874
Conclusione	875
Riferimenti	876

Dos IPv4



Driver di pacchetto	805
Libreria WATTCP	807
Applicazioni standard per WATTCP	809
ABC-nslookup	810
MiniTelnet	812
SSHDOS	813
Bobcat (Lynx)	814
PPRD: servente di rete per la stampa	815
Trout	818
Talk	818
DosLynx	819
NCSA Telnet	821
File «CONFIG.TEL»	822
File «KEYFILE.TEL»	823
File «PASSWORD.TEL»	824
File «SERVICES.TEL»	824
File «TELBIN.EXE»	825
File «FTPBIN.EXE»	825
File «FINGER.EXE»	825
POPMail	826
Menù «Setup»	827

Menù «Window»	828
Menù «=»	828
PCroute	830
Configurazione dei driver di pacchetto	830
Configurazione di PCroute	831
Conclusione	833
Riferimenti	834

Come sistemi operativi liberi, i sistemi GNU costituiscono la scelta ottimale, se non altro dal punto di vista economico, per la realizzazione di reti locali. Ma anche il recupero di vecchia tecnologia può essere di grande aiuto: il vecchio hardware basato su i286 può essere introdotto in una rete TCP/IP per servizi classici quali TELNET, FTP e altro.

Negli esempi che appaiono nelle sezioni che seguono si immagina di avere una piccola rete locale con due elaboratori.

1. 192.168.1.1 *dinkel.brot.dg*, sistema GNU, con funzionalità di router e di servente DNS;
2. 192.168.1.15 *dos.brot.dg* con il sistema Dos.

Il secondo elaboratore è quello che si vuole utilizzare con i programmi Dos descritti in questo capitolo. Per quanto riguarda il caso particolare del programma PCroute, vengono mostrati esempi con dati differenti.

Prima di proseguire, è importante evitare di farsi illusioni: si tratta di programmi molto deboli, utili solo per IPv4 e a volte incapaci di attraversare i router.

Driver di pacchetto

Per poter comunicare attraverso un elaboratore con sistema operativo Dos in una rete TCP/IP occorre un cosiddetto *driver di pacchetto* (*packet driver*), ovvero un driver software in grado di fornire un minimo servizio basato sul protocollo IP. La raccolta di driver di pacchetto più comune è quella della Crynwr ¹ (<http://www.crynwr.com>).

I programmi che si intendono utilizzare devono essere predisposti per il tipo di driver di pacchetto a disposizione.

Una raccolta di driver di pacchetto organizzata da Crynwr, può essere ottenuta presso Simtel.Net (la nota distribuzione di software *shareware* e *freeware* per Dos e MS-Windows) all'indirizzo <ftp://ftp.simtel.net/pub/simtelnet/msdos/pktdrvr/pktd11.zip>.

All'interno della raccolta si può trovare un lungo elenco di driver per vari modelli di schede di rete Ethernet. In particolare, vale la pena di soffermarsi sui driver per le schede Ethernet NE2000 e per la connessione PLIP attraverso la porta parallela.

- **'NE2000.COM'**

Si tratta del driver adatto per la connessione attraverso schede Ethernet compatibili NE2000. Questo programma deve essere avviato con i parametri necessari per poter comunicare con la scheda di rete e con le applicazioni.

```
NE2000.COM [opzioni] [irq_software] [irq_della_scheda] [  
i/o_della_scheda]
```

Le schede NE2000 vengono configurate, attraverso ponticelli o software di configurazione, predisponendo un IRQ e un indirizzo di I/O. Oltre a queste indicazioni, è necessario specificare un indirizzo IRQ aggiuntivo che viene utilizzato per la comunicazione tra i programmi e il driver stesso. La scelta di questo IRQ software è la parte più delicata. L'indirizzo $7E_{16}$ dovrebbe andare bene. Supponendo di avere installato una scheda configurata con IRQ 11 ($0B_{16}$) e indirizzo di I/O 300_{16} , si deve avviare il driver nel modo seguente:

```
NE2000.COM 0x7e 0x0b 0x300
```

È opportuno aggiungere questa riga all'interno del file 'AUTOEXEC.BAT'.

- **'PLIP.COM'**

Si tratta del driver adatto per la connessione attraverso la porta parallela con un cavo PLIP. Questo programma deve essere avviato con i parametri necessari per poter comunicare con la scheda di rete e con le applicazioni.

```
PLIP.COM [opzioni] [irq_software] [irq_della_porta] [i/o_della_porta]
```

Come nel caso delle schede NE2000, è necessario specificare un indirizzo IRQ aggiuntivo che viene utilizzato per la comunicazione tra i programmi e il driver stesso. L'indirizzo $7E_{16}$ dovrebbe andare bene. Supponendo di avere a disposizione una porta pa-

rallela che utilizza IRQ 7 (07_{16}) e indirizzo di I/O 378_{16} , si deve avviare il driver nel modo seguente:

```
PLIP.COM 0x7e 0x07 0x378
```

È opportuno aggiungere questa riga all'interno del file 'AUTOEXEC.BAT'.

I driver di pacchetto Crynwr e altri simili, possono essere rimossi dalla memoria residente attraverso un programma speciale che accompagna la raccolta stessa. Si tratta di 'TERMIN.COM' che richiede soltanto l'indicazione dell'indirizzo IRQ software con il quale è stato installato il driver:

```
TERMIN.COM irq_software
```

Per esempio, per eliminare il driver installato utilizzando l'indirizzo $7E_{16}$, viene eliminato dalla memoria residente con il comando seguente:

```
TERMIN.COM 0x7e
```

Libreria WATTCP



software non libero: non è consentita la distribuzione di versioni modificate e non è consentita la commercializzazione

WATTCP ² (*University of Waterloo TCP*) è una libreria utilizzata da alcuni programmi per accedere alle funzionalità TCP/IP. Di conseguenza, questi programmi hanno in comune lo stesso tipo di file di configurazione, che normalmente è denominato 'WATTCP.CFG'.

Generalmente, questi programmi incorporano completamente il codice della libreria WATTCP, pertanto, i programmi sono autonomi, ma possono usare in comune lo stesso file di configurazione.

Questo file si compone di direttive molto semplici, in cui si assegna idealmente un valore a una variabile:

```
variabile_di_configurazione = valore
```

In generale, viene definito l'indirizzo IP, il nome corrispondente e la maschera di rete (o della sottorete), come si vede nell'esempio seguente:

```
HOSTNAME = dos  
MY_IP = 192.168.1.15  
NETMASK = 255.255.255.0
```

Inoltre, di solito si indicano anche i serveri DNS,³ il nome del proprio dominio e il router (gateway):

```
GATEWAY = 192.168.1.1  
NAMESERVER = 192.168.1.1  
DOMAINSLIST = brot.dg
```

Può essere interessante anche la definizione della dimensione massima dei pacchetti (MSS, *Max segment size*), se per qualche motivo il driver di pacchetto dovesse avere delle limitazioni:

```
MSS = 512
```

In breve, un esempio completo, senza l'indicazione della dimensione massima dei pacchetti:

```
MY_IP = 192.168.1.15
NETMASK = 255.255.255.0
GATEWAY = 192.168.1.1
NAMESERVER = 192.168.1.1
HOSTNAME = dos
DOMAINSLIST = brot.dg
```

Alcuni programmi potrebbero richiedere la presenza di una variabile di ambiente che permetta loro di individuare facilmente la collocazione e il nome del file di configurazione. Per esempio, se si tratta del file 'C:\TCP\WATTCP.CFG', potrebbe essere richiesto di includere nel file 'AUTOEXEC.BAT' la riga seguente:

```
SET WATTCP.CFG=\TCP
```

I programmi che utilizzano questa libreria, fanno spesso riferimento a file standard dei sistemi Unix, che devono trovare nella directory corrente:

'HOSTS'	contiene un elenco di indirizzi IP associati al nome a dominio corrispondente, equivalente al noto '/etc/hosts' dei sistemi Unix
'PROTOCOL'	contiene l'elenco dei protocolli con i nomi associati, equivalente al noto '/etc/protocols' dei sistemi Unix
'SERVICES'	contiene l'elenco dei servizi di rete, equivalente al noto '/etc/services' dei sistemi Unix.

Applicazioni standard per WATTCP

Assieme alla libreria WATTCP, si trovano i sorgenti di alcuni programmi comuni, ⁴ diffusi in forma binaria in un archivio compresso denominato 'apps.zip'. La tabella seguente ne elenca alcuni:

<code>tcpinfo</code>	mostra le informazioni tratte dal file di configurazione 'wattcp.cfg', consentendone il controllo;
<code>ping <i>nodo</i></code>	esegue una richiesta di eco ICMP verso il nodo indicato;
<code>finger [<i>utente</i>@] <i>nodo</i></code>	richiede le informazioni disponibili sugli utenti del nodo indicato;
<code>rexec <i>nodo</i> [<i>utente</i> [<i>parola_d'ordine</i>]] <i>comando</i></code>	esegue un comando remoto attraverso il protocollo RSH;
<code>lpr [<i>coda</i>] <i>nodo file</i></code>	invia un file alla stampante remota;
<code>lpq -P<i>coda</i> -S<i>nodo</i></code>	interroga la coda di una stampante remota;
<code>ftp [<i>utente</i>@] <i>nodo</i></code>	richiede l'instaurazione di un collegamento FTP con il nodo indicato;

ABC-nslookup

«

ABC-nslookup ⁵ consente l'interrogazione di un servizio DNS. Si tratta di un programma molto semplice che utilizza la libreria WATT-CP e anche la libreria ABC, ma la seconda richiede la preparazione di altri file; in particolare, per ciò che riguarda ABC-nslookup, è necessario predisporre un file contenente l'elenco dei server DNS a disposizione.

Il pacchetto originale del programma, corrispondente al file ‘nslb01a.zip’, include una sottodirectory che dovrebbe essere riprodotta tale e quale nella radice del disco: ‘\ETC\’. Questa directory contiene in particolare il file ‘RESOLV.CNF’, che corrisponde in pratica al file ‘/etc/resolv.conf’ dei sistemi Unix. Eventualmente, se si desidera collocare questi file in una posizione diversa, basta definire la variabile di ambiente ‘**ABCETCDIR**’; per esempio, si può scrivere nel file ‘AUTOEXEC.BAT’:

```
SET ABCETCDIR=\TCPIP
```

In questo caso, si intende dire che i file di tale directory si trovano invece in ‘\TCPIP\’.

È la libreria ABC che richiede la presenza di alcuni file nella directory ‘\ETC\’, pertanto è questo il motivo del nome della variabile di ambiente.

Il pacchetto si compone di due eseguibili, che cercano il file ‘WATTCP.CFG’ nella directory corrente, ignorando la variabile di ambiente ‘**WATTCP.CFG**’:

```
NSLOOKUP [nodo_da_trovare [servente_dns ] ]
```

```
NSQUERY [-d] nodo_da_trovare [servente_dns ]
```

I due comandi consentono di interrogare un server DNS per risolvere un nome in numero e viceversa, oppure per avere maggiori dettagli sulle registrazioni del DNS che riguardano il nodo cercato.

Se non viene indicato il server DNS nella riga di comando, si fa riferimento a quanto indicato nella configurazione, precisamente nel file 'RESOLV.CNF', che a questo proposito si compila come quello dei sistemi Unix.

MiniTelnet

«

MiniTelnet ⁶ è un piccolo programma cliente per il protocollo TELNET, basato sulla libreria WATTCP. Il programma eseguibile corrispondente è '**MT.EXE**', che si usa così:

```
MT nodo [-Pporta] [-Eemulazione] [-Ktastiera]
```

L'emulazione del terminale viene definita con l'opzione '**-E**', secondo la tabella:

Opzione	Descrizione
-EVT52	emula un terminale di tipo VT52;
-EHeath19	emula una variante del terminale VT52;
-EVT102	emula un terminale VT102;
-EVT200	emula un terminale VT200;
-EANSI	emula un terminale ANSI;

Per quanto riguarda l'emulazione della tastiera, il pacchetto di MiniTelnet include alcuni file di esempio, con estensione '.KBD'. Per selezionare uno di questi file, si utilizza l'opzione '**-K**', seguita dalla radice del nome di questi file. Per esempio, per utilizzare il file 'VT-

AT.KBD', si deve usare l'opzione '**-KVT-AT**'.

Per accedere a un sistema GNU con MiniTelnet, attraverso un vecchio elaboratore con tastiera standard, può essere conveniente l'uso della sintassi seguente:

```
MT nodo -EANSI -KVT-AT
```

In questo modo si seleziona l'emulazione ANSI e il file 'VT-AT.KBD' per la tastiera. Tuttavia, il server TELNET potrebbe non essere in grado di passare l'informazione sul tipo di terminale utilizzato alla shell, pertanto conviene impostare manualmente la variabile '**TERM**' una volta iniziato il collegamento:

```
$ export TERM=linux [Invio]
```

L'esempio riguarda il caso di un sistema GNU/Linux, dove la voce '**linux**' per identificare il tipo di terminale sembra essere la più vicina al funzionamento ottimale.

SSHDOS

SSHDOS ⁷ è un cliente per il protocollo SSH nelle versioni 1.*. Si tratta di un programma che usa la libreria WATTCP, pertanto non crea problemi di configurazione. Il vero problema, semmai, riguarda la versione del protocollo, dal momento che un server che offre solo il protocollo 2, non può comunicare con SSHDOS.

SSHDOS viene distribuito in due eseguibili differenti, a seconda della disponibilità o meno di una CPU i386. In generale, funziona sempre l'eseguibile '**SSHDOS.EXE**', anche se è molto lento:

```
SSHDOS [opzioni] utente nodo [comando]
```

In condizioni normali, non si usano le opzioni e nemmeno il comando da eseguire nell'elaboratore remoto; in questo modo viene chiesto di inserire la parola d'ordine, dopo la quale si ottiene di interagire con la shell. Eventualmente, per ottenere l'elenco delle opzioni disponibili, è sufficiente avviare l'eseguibile senza argomenti.

Se non si specificano opzioni particolari al riguardo, SSHDOS funziona emulando il comportamento di un terminale di tipo 'xterm'; generalmente non è necessario cambiare questa impostazione con le opzioni.

Il pacchetto con cui è distribuito SSHDOS contiene anche un programma per l'uso del protocollo TELNET:

```
TELNET [opzioni] nodo [porta]
```

Dovrebbe essere disponibile anche un pacchetto separato per il protocollo SSH 2, ma potrebbe non essere completamente efficiente nella versione per CPU x86-16.

Bobcat (Lynx)

«

Bobcat è una raccolta di applicativi, organizzata attorno a una versione di Lynx ⁸ per Dos. Le licenze dei vari applicativi inseriti sono varie; tuttavia, il pacchetto più importante è proprio Lynx, che può funzionare anche da solo, per accedere ai servizi HTTP comuni.

Bobcat è ottenibile dall'indirizzo <http://www.fdisk.com/doslynx/bobcat.htm>. Eventualmente si può fare una ricerca con <http://www>.

google.com/search?q=bcat-e07.exe .

Di tutti i file che compongono il pacchetto, sono sufficienti il programma '**LYNX.EXE**' e il file '**LYNX.CFG**'. Il secondo è il file di configurazione, in cui è bene definire la collocazione di alcuni file HTML (con estensione '**.HTM**'), che vengono usati quando si richiede la guida, altre informazioni e per accumulare eventualmente lo storico degli indirizzi richiesti. Inoltre, questa edizione di Lynx utilizza la libreria WATTCP, pertanto il programma si aspetta di trovare il file di configurazione '**WATTCP.CFG**' nella directory corrente.

PPRD: servente di rete per la stampa



software non libero: licenza Artistic

PPRD ⁹ è una piccola raccolta di programmi per la gestione di un servente di stampa secondo lo stile del demone '**lpd**'. Nelle situazioni in cui il sistema riesce a funzionare, permette di riutilizzare un vecchio PC, anche un XT, per questo scopo. Può essere ottenuto presso Simtel.Net all'indirizzo <ftp://ftp.simtel.net/pub/simtelnet/msdos/lan/pprd200.zip> .

PPRD si avvale della libreria WATTCP, pertanto può condividere la configurazione con altri programmi simili. In particolare, nel file di configurazione '**WATTCP.CFG**' si deve specificare la dimensione della memoria tampone (*buffer*) di trasmissione e ricezione.

```
TXBUFSIZE=8192
RXBUFSIZE=8192
```

Il programma eseguibile che svolge il lavoro è '**PPRD.EXE**', che viene avviato normalmente senza l'indicazione di alcuna opzione,

purché il file 'WATTCP.CFG' sia stato predisposto correttamente e collocato nella directory corrente:

```
PPRD [opzioni]
```

Salvo una diversa configurazione, il programma offre la stampante (o le stampanti) con un nome corrispondente a quello usato dal Dos per identificare il dispositivo: 'lpt1', 'lpt2', ...

Nell'elaboratore con un sistema GNU dal quale si vogliono inviare le stampe occorre sistemare il file '/etc/printcap' in modo adeguato. Quello che segue è un esempio in cui:

1. la stampante predefinita punta direttamente alla stampante remota, il cui nome è 'lpt1';
2. la stampante 'ps' utilizza un filtro che trasforma un documento PostScript in un file adatto alla stampante remota e poi lo ridirige alla stampante predefinita;
3. la stampante 'tx' utilizza un filtro che trasforma un file di testo in stile Unix in un file di testo in stile Dos e poi lo ridirige alla stampante predefinita.

```
#  
# /etc/printcap  
#  
lp:\  
    :lp=:\  
    :sd=/var/spool/lpd/lp:\  
    :rm=192.168.1.15:\  
    :rp=lpt1:\  
    :mx#0:\  
    :sf:\  
    :sh:  
#
```

```

ps:\
    :lp=/dev/null:\
    :sd=/var/spool/lpd/postscript:\
    :lf=/var/spool/lpd/postscript/log:\
    :if=/var/spool/lpd/postscript/input-filter:\
    :sh:\
    :sf:\
    :mx#0:
#
tx:\
    :lp=/dev/null:\
    :sd=/var/spool/lpd/text:\
    :lf=/var/spool/lpd/text/log:\
    :if=/var/spool/lpd/text/input-filter:\
    :sh:\
    :sf:\
    :mx#0:

```

Segue lo script usato come filtro di input per la stampa in PostScript. Lo script riceve i dati dallo standard input e attraverso Ghostscript lo trasforma in un file adatto per la stampa su una stampante a nove aghi tipo IBM-EPSON e dirige l'output verso la stampante predefinita, cioè quella remota.¹⁰

```

#!/bin/sh
/bin/grep -v '(%%' | /usr/bin/gs -q -dNOPAUSE -sPAPERSIZE=letter \
-sDEVICE=eps9high -sOutputFile=- - | /usr/bin/lpr

```

Segue lo script usato come filtro di input per la stampa dei file di testo. Lo script riceve i dati dallo standard input e attraverso il programma '**unix2dos**' lo trasforma in un file di testo in cui ogni riga è terminata dalla sequenza $\langle CR \rangle \langle LF \rangle$ come richiesto dalle stampanti normali. L'output viene quindi diretto verso la stampante predefinita, cioè quella remota.

```

#!/bin/sh
/bin/cat | /usr/bin/unix2dos | /usr/bin/lpr

```

Se poi **'unix2dos'** non si comporta come previsto, si può realizzare un programma Perl:

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

Trout

«

Trout ¹¹ è una versione Dos del noto Traceroute per sistemi Unix, che utilizza la libreria WATTCP; l'eseguibile che svolge il lavoro è **'TROUT.EXE'**.

Per ottenere il pacchetto, si può fare una ricerca per i file `'trtb01b.zip'` o `'trt-e01.exe'`.

Talk

«

Talk ¹² è una versione Dos del noto programma con lo stesso nome per i sistemi Unix. Questa versione per Dos utilizza la libreria WATTCP e l'eseguibile che svolge il lavoro è **'TALK.EXE'**. Il pacchetto può essere recuperato presso l'indirizzo <http://http://users.libero.it/kht/mirror/half-mirror/dos/network/tcp.ip/wattcp/talk-13.zip>.

L'eseguibile '**TALK.EXE**' si comporta simultaneamente da servente e da cliente:

```
TALK [opzioni] [utente@nodo [terminale]]
```

Se non si indica l'utente e il nodo da contattare, si avvia il programma in attesa di chiamate, a cui poi viene data risposta, qualunque sia il nominativo utente che viene richiesto. Le opzioni disponibili sono poche; in particolare, '-1' consente di avere una registrazione della comunicazione in un file ('TALK.LOG'), mentre '-o' consente di richiedere l'uso di un protocollo più vecchio.

Durante il funzionamento, è possibile usare il tasto [F1] per ottenere una guida rapida all'uso dei comandi da tastiera; in particolare, la combinazione [Alt s], consente di cambiare la modalità visiva della comunicazione (a schermo unico o a schermo diviso).

La configurazione con il file 'WATTCP.CFG' prevede l'aggiunta di direttive specifiche, che comunque non sono indispensabili.

DosLynx



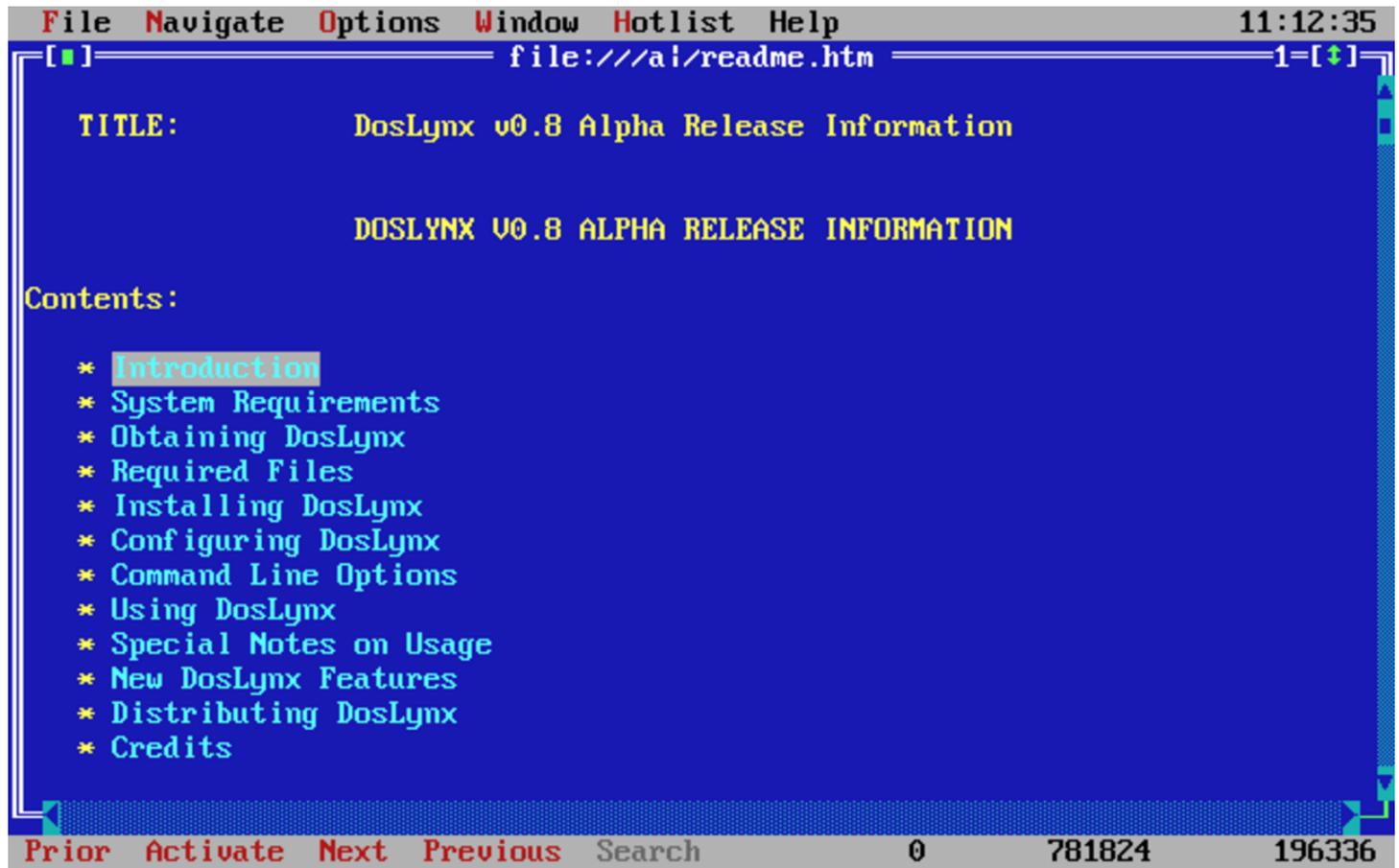
software didattico, non libero



DosLynx ¹³ è un programma di navigazione a caratteri, ma relativamente completo, da utilizzare insieme a un driver di pacchetto per il TCP/IP (non fa uso della libreria WATTCP, pertanto ha una configurazione indipendente). Può essere ottenuto presso <http://www.fdisk.com/doslynx/doslynx.htm>. Per quanto possibile, questo applicativo è molto accurato, per esempio permette l'uso del mouse.

L'eseguibile è precisamente 'DOSLYNX.EXE' che si avvia senza l'indicazione di argomenti particolari; ma prima di poter essere utilizzato occorre predisporre il file 'DOSLYNX.CFG'.

Figura u59.21. La guida interna di DosLynx.



DosLynx permette anche l'invio di messaggi di posta elettronica, ma non la loro ricezione o lettura.

La configurazione di DosLynx avviene attraverso il file 'DOSLYNX.CFG', collocato nella directory corrente nel momento in cui si avvia il programma. In questo file, per prima cosa deve essere definito l'indirizzo IP e la maschera di rete (*netmask*).

```
my_ip=192.168.1.15
netmask=255.255.255.0
```

Quindi occorre indicare l'indirizzo del router (gateway) e del ser-

vente DNS anche se in realtà possono non esistere nella rete locale che si utilizza.

```
gateway=192.168.1.1
nameserver=192.168.1.1
```

Viene specificato quindi il dominio e il nome dell'elaboratore locale.

```
domainslist="brot.dg"
hostname=dos
```

Per il resto, questo file di configurazione viene già fornito con un esempio molto ben commentato. Vale comunque la pena di indicare:

- l'attivazione del collegamento con l'esterno;
- il proprio indirizzo di posta elettronica, che viene utilizzato come mittente per i messaggi inviati;
- l'indicazione dell'elaboratore a cui fare riferimento per l'inoltro dei messaggi di posta elettronica inviati, attraverso il protocollo SMTP;
- l'indicazione dell'elaboratore a cui fare riferimento per l'accesso a NNTP (news).

```
networked=YES
mailaddr=daniele@dinkel.brot.dg
smtpghost=192.168.1.1
nntpghost=192.168.1.1
```

NCSA Telnet

NCSA Telnet ¹⁴ è una piccola raccolta di programmi da utilizzare insieme a un driver di pacchetto per il TCP/IP (si tratta di programmi autonomi dalla libreria WATTCP). Può essere ottenuta presso Simtel.Net all'indirizzo <ftp://ftp.simtel.net/pub/simtelnet/msdos/ncsatln/>.

Nelle sezioni seguenti vengono descritti solo alcuni dei programmi di questa raccolta. Prima di poterli utilizzare occorre predisporre il file 'CONFIG.TEL'.

L'ultima versione di questa raccolta dovrebbe essere la 2.3.08 che però sembra avere qualche problema, in particolare non può essere utilizzata quando si abilita il *Path MTU discovery* durante la compilazione del kernel Linux. La versione 2.3.07.4 (precedente) dovrebbe essere esente da questo difetto. Inoltre, alcune versioni precedenti alla 2.3.08, compresa la 2.3.07.4, contengono più programmi di servizio accessori, come un programma per il ping e uno per il tracciamento dell'instradamento. Per trovare la versione 2.3.07.4 si può provare con: <http://www.google.com/search?q=tel23074.zip>.

File «CONFIG.TEL»

«CONFIG.TEL» è un file di testo contenente l'indicazione della configurazione del gruppetto di programmi che compongono il pacchetto Telnet NCSA. Per prima cosa deve essere definito l'indirizzo IP e la maschera di rete (*netmask*).

```
myip=192.168.1.15
netmask=255.255.255.0
```

Quindi occorre definire le caratteristiche del driver di pacchetto utilizzato, che a loro volta dipendono dal tipo di scheda di rete. L'esempio seguente riguarda il caso del driver di pacchetto Crynwr per la scheda NE2000 ('**NE2000.COM 0x7e 0x0b 0x300**'). Occorre fare attenzione alla voce '**ioaddr=**' che non si riferisce a un indirizzo di I/O, ma all'IRQ software del driver di pacchetto.

```
hardware=packet
interrupt=11
ioaddr=0x7e
```

Segue una serie di altre informazioni, in particolare sono interessanti le seguenti.

```
myname=dos.brot.dg
termtype="vt100"
keyfile=".\\keymap.tel"
services=".\\services.tel"
ftp=yes
ftpwrt=yes
passfile=".\\password.tel"
```

Quindi vengono richieste le informazioni sui nodi che possono essere contattati. Vengono inizialmente indicate delle caratteristiche generali predefinite, quindi i dati particolari di nodi determinati.

```
name=default
# Segue una serie di caratteristiche predefinite
# ...
# Inizia la definizione dell'elaboratore «dinkel.brot.dg»
name=dinkel.brot.dg
hostip=192.168.1.1
gateway=0          # Non è un router
nameserver=1      # È un Name Server
# Inizia la definizione dell'alias «dinkel»
name=dinkel
copyfrom=dinkel.brot.dg
```

File «KEYFILE.TEL»

All'interno del file 'CONFIG.TEL', con la voce '**keyfile=**', viene dichiarato il nome e la collocazione di un file di configurazione della tastiera. Lo scopo di questo file è definire una corrispondenza tra tasti premuti e segnali inviati. Telnet NCSA fornisce già questo file e ha il nome 'KEYFILE.TEL'.

Per poter conoscere i codici a cui corrispondono i tasti della propria tastiera, si può utilizzare il programma '**SCANCHEK.EXE**'.

In generale, non conviene modificare il file originale, piuttosto, è meglio tentare diversi tipi di configurazione di terminale assegnando un valore opportuno alla variabile di ambiente '**TERM**' di GNU/Linux o alla voce '**termttype=**' del file '**CONFIG.TEL**'. si possono provare, in particolare, i valori '**vt100**' e '**vt220**'.

File «**PASSWORD.TEL**»

«

Il programma '**TELBIN.EXE**' può funzionare anche come un semplice servente FTP per accessi singoli.¹⁵

Per questo, è necessario definire un file contenente informazioni sugli utenti e sui loro permessi di accesso. Il nome e la posizione di questo file viene definito all'interno di '**CONFIG.TEL**', con la voce '**passfile=**' e di solito si tratta di '**PASSWORD.TEL**'.

Per crearlo o modificarlo, conviene utilizzare il programma '**TELPASS.EXE**', per esempio nel modo seguente. Il programma stesso suggerisce le operazioni da compiere.

```
C:\NCSATELN> telpass password.tel [Invio]
```

File «**SERVICES.TEL**»

«

'**SERVICES.TEL**' è il file dei servizi di rete ed è analogo al file '**/etc/services**' (32.8.2). Viene già fornito configurato correttamente.

File «TELBIN.EXE»

```
TELBIN [nodo]
```

Il programma '**TELBIN.EXE**' è il più importante di questo gruppo, essendo quello che permette di attivare una connessione TELNET con un elaboratore GNU/Linux o un altro Unix. Se il programma non riesce a connettersi con l'elaboratore indicato come argomento, o se questo non viene indicato, si avvia come servente FTP e accetta una sola connessione alla volta.

Quando si vuole utilizzare '**TELBIN.EXE**' come servente FTP occorre predisporre il file degli utenti, che solitamente è 'PASSWORD.TEL'.

File «FTPBIN.EXE»

```
FTPBIN [nodo]
```

'**FTPBIN.EXE**' è il secondo programma come importanza. Si tratta di un semplice cliente FTP abbastanza funzionante. I comandi che mette a disposizione sono i soliti per questo tipo di programma; per ottenere aiuto si può utilizzare il punto interrogativo ('?').

File «FINGER.EXE»

```
FINGER [utente]@nodo
```

Il programma '**FINGER.EXE**' permette di ottenere informazioni sugli utenti connessi in un elaboratore determinato. Il risultato di

questa interrogazione è analogo a quello del suo omonimo negli ambienti Unix.

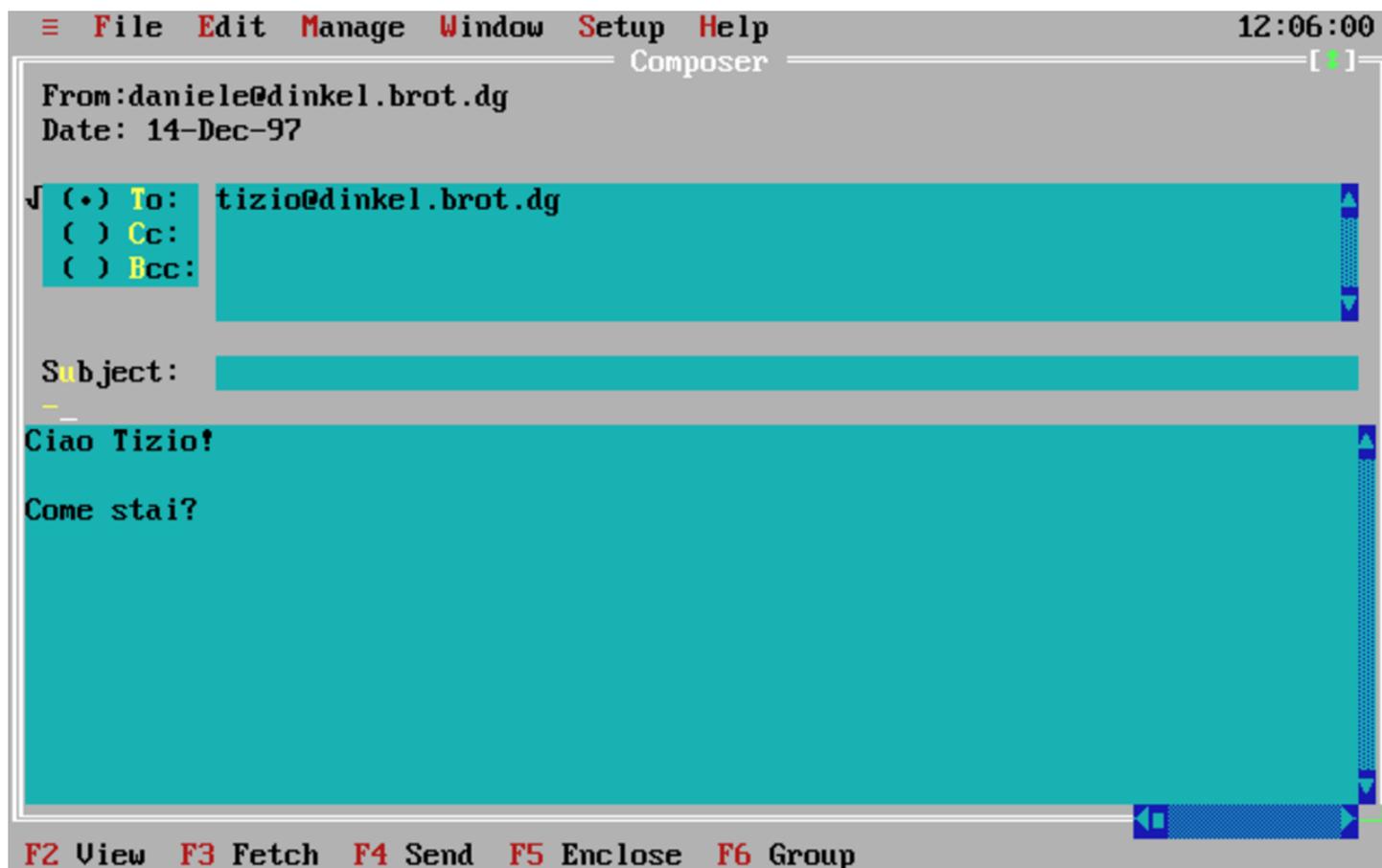
POPMail

«

POPMail ¹⁶ è un ottimo programma per la gestione della posta elettronica attraverso la connessione con un servizio POP2 o POP3. Può essere ottenuto presso Simtel.Net all'indirizzo <ftp://ftp.simtel.net/pub/simtelnet/msdos/pktdrvr/popml322.zip>.

La configurazione viene fatta attraverso il programma stesso e non richiede la preparazione di alcun file.

Figura u59.30. La composizione di un messaggio di posta elettronica attraverso POPMail.



POPMail si compone di un solo eseguibile monolitico:

'POPMAIL.EXE'. Tutte le sue funzionalità sono incorporate in questo, compresa la configurazione. Appena si avvia il programma si ottiene un'interfaccia amichevole che permette l'uso del mouse.

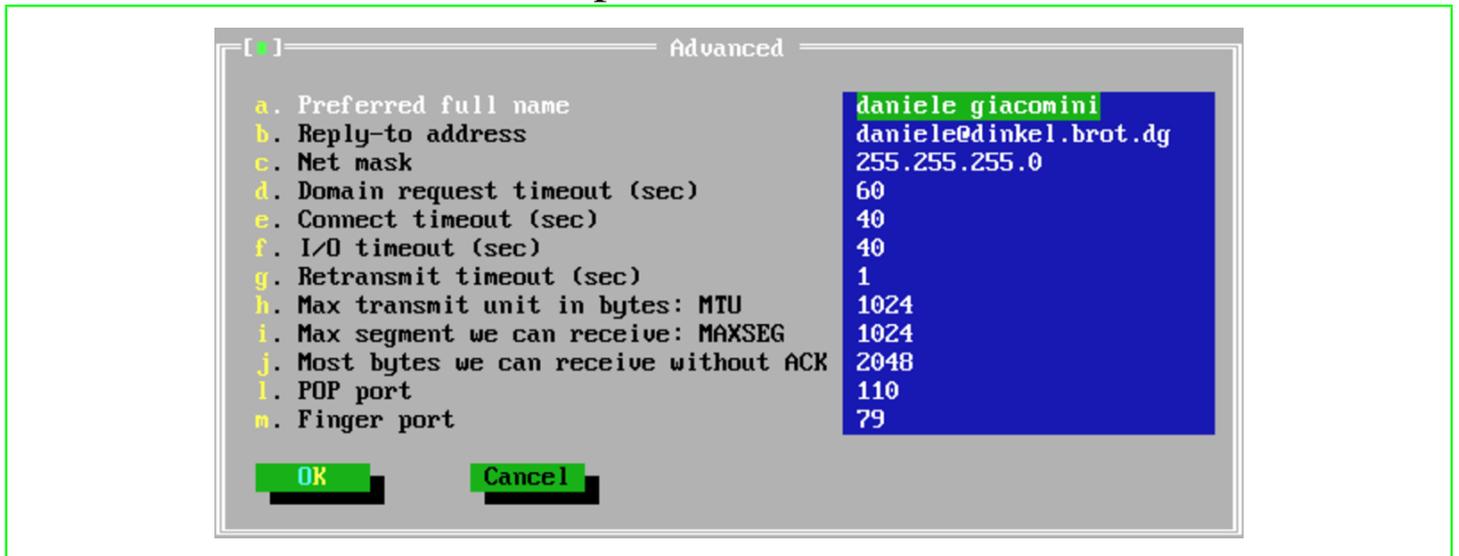
Menù «Setup»

La configurazione del programma si definisce attraverso le funzioni del menù *Setup*. In particolare è importante la voce *Network*, attraverso cui si accede a una maschera per la definizione degli indirizzi e dei nomi utilizzati. In questa fase, è importante stabilire il tipo di protocollo che si intende utilizzare. Questo lo si fa attraverso l'indicazione della porta di comunicazione. Quella predefinita è 109 corrispondente a POP2, altrimenti si può utilizzare la porta 110 in modo da collegarsi a un servizio POP3.

Figura u59.31. La finestra principale della configurazione di POPMail. Si può osservare che il nodo da specificare alla voce *Host Computer* è quello che fornisce il servizio SMTP, mentre subito sotto è richiesto l'indirizzo dell'elaboratore locale.



Figura u59.32. Selezionando il pulsante `ADVANCED` dalla finestra principale di configurazione, si ottiene questa finestra di informazioni aggiuntive. La selezione del tipo di protocollo dipende dal numero di porta selezionato. In questo caso, essendo il numero 110, si utilizza il protocollo POP3.



Menù «Window»

«

Una volta definita la configurazione, si può iniziare a utilizzare il programma per ricevere e spedire posta. Esistono tre finestre: una per la composizione dei messaggi, un'altra per la loro lettura e l'ultima per le operazioni di «taglia-copia-incolla». Per passare da una finestra all'altra, occorre richiamare questo menù.

Menù «=»

«

Il menù dell'applicazione, quello precedente a *File*, permette di accedere a funzionalità aggiuntive e molto utili. Si può utilizzare una sessione TELNET in una finestra, si può ottenere la risoluzione di indirizzi IP e si può eseguire il ping.

Figura u59.33. Una sessione TELNET attraverso POPMail.

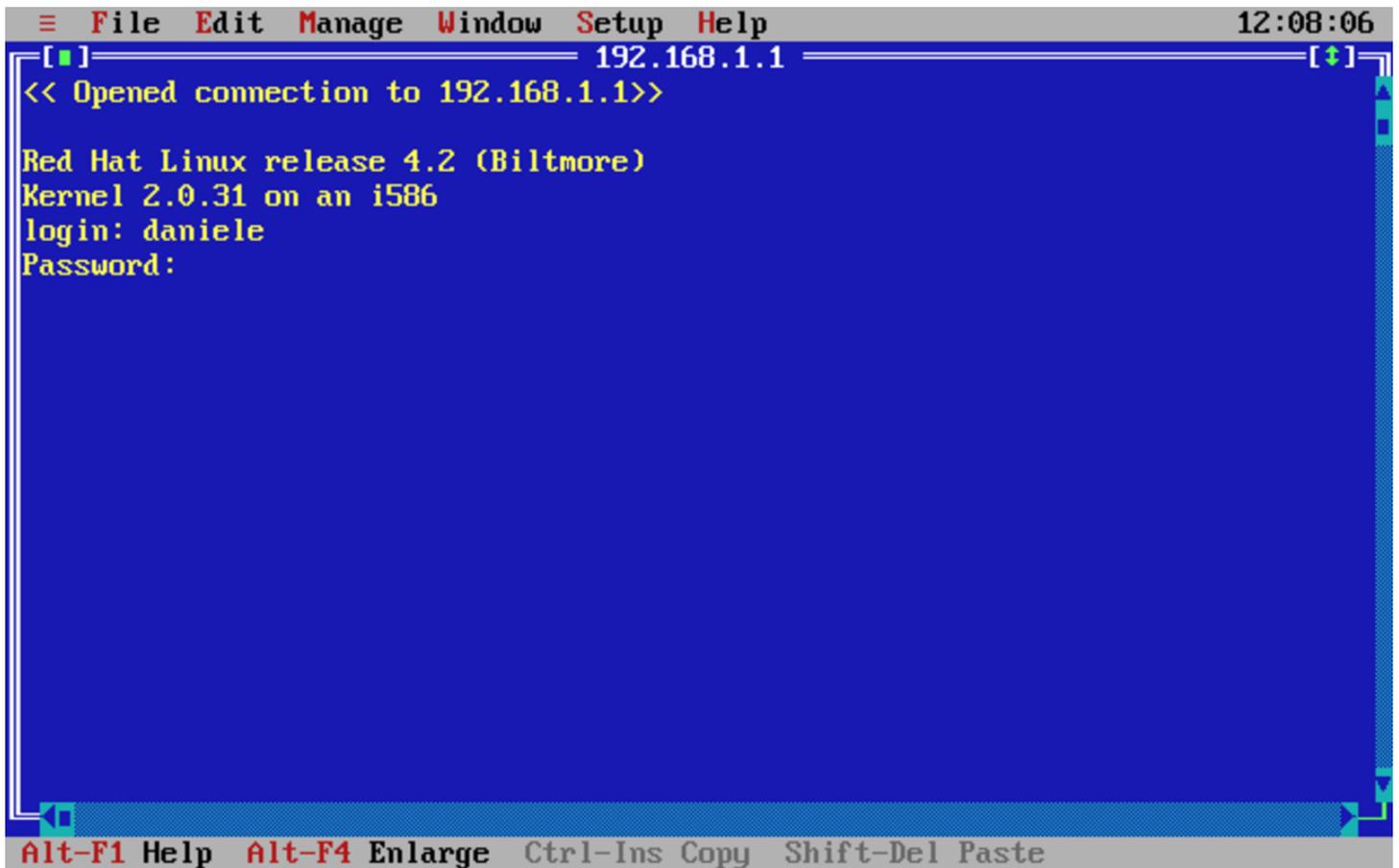


Figura u59.34. La presenza di una funzione di ping completa l'applicativo POPMail.



PCroute

«

PCroute permette di trasformare un vecchio PC (i286 o inferiore) in un router IPv4. Può essere ottenuto presso Simtel.Net all'indirizzo <ftp://ftp.simtel.net/pub/simtelnet/msdos/network/pcrte224.zip>.

Nell'archivio che viene distribuito, è presente il sorgente e diverse versioni compilate, per l'uso delle schede di rete più comuni nel passato. Tra queste versioni già pronte ne esiste una in grado di utilizzare i driver di pacchetto descritti all'inizio di questo capitolo. Gli esempi che vengono mostrati qui si riferiscono all'utilizzo dei driver di pacchetto.

Configurazione dei driver di pacchetto

«

Se si decide di utilizzare la versione già compilata per i driver di pacchetto, cioè 'PKTPKT.EXE', è necessario prima configurare i driver di pacchetto, poi si può pensare alla configurazione di PCroute.

La versione precompilata, 'PKTPKT.EXE', prevede l'utilizzo di due indirizzi di interruzione (*interrupt*) software per comunicare con i driver di pacchetto, 60_{16} e 61_{16} , dove il primo si riferisce alla prima interfaccia e l'altro alla seconda.

Supponendo di disporre di schede di rete compatibili NE2000, che utilizzino rispettivamente le risorse IRQ 10 e I/O 280_{16} , IRQ 11 e I/O 300_{16} , la configurazione dei driver di pacchetto dovrebbe essere la seguente:

```
ne2000 0x60 0x0a 0x280
ne2000 0x61 0x0b 0x300
```

Configurazione di PCroute

Per fare funzionare PCroute è necessario l'eseguibile 'PCROUTE.EXE'; nel caso di utilizzo dei driver di pacchetto, si tratta di 'PKTPKT.EXE'. Inoltre serve anche 'CONFIG.EXE', per generare il file di configurazione di PCroute.

Si suppone che la prima scheda sia inserita nella rete 192.168.1.0 e che abbia l'indirizzo 192.168.1.254; inoltre si suppone che la seconda sia nella rete 192.168.2.0 con l'indirizzo 192.168.2.254. Non si prevede la necessità di indicare altri instradamenti per mezzo di altri router.

```
C:\PCROUTE> CONFIG [Invio]
```

```
This program creates/edits the pcroute.cfg file
```

Inizia una configurazione interattiva, a cominciare dalle indicazioni riferite alla prima interfaccia, cioè quella collegata al driver di pacchetto attraverso l'indirizzo IRQ 60₁₆.

```
Configuring an interface
```

```
Address for the interface [0.0.0.0] ? 192.168.1.254 [Invio]
```

```
Subnet mask for the interface [255.255.255.0] ? 255.255.255.0 [Invio]
```

```
Flag Meanings (if set)
```

```
Bit 0 (1h) - Don't send routing updates out this interface  
Bit 1 (2h) - Don't listen to routing updates from this interface  
Bit 2 (4h) - Proxy Arp for all subnets  
Bit 3 (8h) - Turn off directed broadcasts  
Bit 4 (10h) - Turn off the issuing of ICMP redirects  
Bit 5 (20h) - Broadcast using old (0's) format
```

```
Flags (HEX) for the interface [0H] ? 0H [Invio]
```

Routing Metric (HEX) for the interface [1H] ? **1H**[Invio]

A questo punto si passa alla configurazione della seconda interfaccia, cioè quella collegata al driver di pacchetto attraverso l'indirizzo $IRQ\ 61_{16}$.

Configuring an interface

Address for the interface [0.0.0.0] ? **192.168.2.254** [Invio]

Subnet mask for the interface [255.255.255.0] ? **255.255.255.0** [Invio]

Flag Meanings (if set)

- Bit 0 (1h) - Don't send routing updates out this interface
- Bit 1 (2h) - Don't listen to routing updates from this interface
- Bit 2 (4h) - Proxy Arp for all subnets
- Bit 3 (8h) - Turn off directed broadcasts
- Bit 4 (10h) - Turn off the issuing of ICMP redirects
- Bit 5 (20h) - Broadcast using old (0's) format

Flags (HEX) for the interface [0H] ? **0H** [Invio]

Routing Metric (HEX) for the interface [1H] ? **1H** [Invio]

Gli instradamenti sulle reti cui sono connesse le interfacce vengono definiti in modo automatico. Si decide di non indicare altri instradamenti particolari.

If you wish to configure static routes do so here. To stop type a '.'

Flag Meanings (if set)

- Bit 0 (1h) - Local route, do not propagate it
- Bit 1 (2h) - Transient route, subject to RIP protocol

Network [0.0.0.0] ? **.** [Invio]

Da questo punto non si seleziona alcuna opzione particolare.

If you wish to forward bootp packets please enter the address of the address to forward it to. This address can be a directed broadcast. 0.0.0.0 means don't forward

Address to forward bootp packets [0.0.0.0] ? **0.0.0.0**[Invio]

Once PCroute boots up, it sends all log messages to a network host running a BSD UNIX syslogd daemon. To disable logging enter 0.0.0.0

Host to send logging info to [0.0.0.0] ? **0.0.0.0**[Invio]

Mask Meanings (0 = Log, 1 = Don't log)

Bit 0 (1h) - System
Bit 1 (2h) - Routing
Bit 2 (4h) - Monitor
Bit 3 (8h) - Localtalk

Logging mask for this router [0H] ? **0H**[Invio]

There are 8 routing 'levels' supported

0 - Emergency 1 - Alert 2 - Critical 3 - Error
4 - Warning 5 - Notice 6 - info 7 - Debug

Only messages with a level less than the logging level are sent

Logging level [0H] ? **0H**[Invio]

A questo punto la configurazione termina e ne viene generato il file 'PCROUTE.CFG'.

Conclusione

PCroute, per funzionare richiede solo l'avvio dell'eseguibile ('**PCROUTE.EXE**'), che ha la necessità di trovare il file 'PCROUTE.CFG' nella directory corrente. Dopo l'avvio, l'elaboratore risulta bloccato, essendo destinato esclusivamente alla funzione di instradamento.

La documentazione di PCroute spiega meglio come gestire le varie opzioni, che nell'esempio sono state evitate semplicemente, descrivendo anche come sfruttare la possibilità di tenere sotto controllo il funzionamento di PCroute attraverso il registro di sistema di un elaboratore come GNU/Linux.

Riferimenti



- Erick Engelke, *WATTCP*
<http://www.wattcp.com/>
- Smash-Co Communications, *TCP/IP for MS-DOS*
<http://www.smashco.com/wattcp.asp>
- *WATTCP*
<http://www.wattcp.com/>
<http://www.wattcp.com/wat1104.zip>
- *The U-M Software Archive*
<http://www.umich.edu/~archive/msdos/communications/wattcp/>
<http://www.umich.edu/~archive/msdos/communications/packet/>
- *Arachne labs*
<http://www.arachne.cz/>
- *Arachne GPL*
<http://home.hetnet.nl/~ba8tian/arachne/arachne.htm>
<http://home.hetnet.nl/~ba8tian/arachne/175-gpl/ar175.htm>
<http://home.hetnet.nl/~ba8tian/arachne/175-gpl/a175gp75lf.zip>
- *Using the Internet (or an IP Network) from a MSDOS Machine*
<http://www.geocities.com/SiliconValley/Park/2884/dosint.htm>

- ¹ **Crynwr packet driver collection** GNU GPL
- ² **WATTCP** software non libero: non è consentita la distribuzione di versioni modificate e non è consentita la commercializzazione
- ³ Per indicare più server DNS, è sufficiente usare la direttiva **'NAMESERVER'** ripetutamente.
- ⁴ **WATTCP apps** software non libero: non può essere commercializzato
- ⁵ **ABC-nslookup** UCB BSD
- ⁶ **MiniTelnet** software libero con licenza speciale
- ⁷ **SSHDOS** GNU GPL
- ⁸ **Lynx** GNU GPL
- ⁹ **PPRD** software non libero: licenza Artistic
- ¹⁰ Si suppone di usare carta a modulo continuo, pertanto viene indicato il formato lettera (11 in).
- ¹¹ **Trout** software gratuito senza sorgenti
- ¹² **Talk** GNU GPL
- ¹³ **DosLynx** software didattico, non libero
- ¹⁴ **NCSA Telnet** dominio pubblico
- ¹⁵ Non è il caso di fare affidamento su questa funzionalità di **'TELBIN.EXE'** perché non è perfettamente funzionante.
- ¹⁶ **POPMail** software gratuito non modificabile e senza sorgenti

Dos PPP

Composizione	837
Configurazione e script	838
Opzioni particolari per il PPP	840
Connessione in pratica	841

Per realizzare una connessione PPP con un sistema Dos, è necessario un driver di pacchetto speciale, più o meno derivato dal demone ‘**pppd**’ tradizionale dei sistemi Unix. Dal momento che di solito si usa una connessione PPP attraverso un modem e una linea commutata, è necessario anche un programma analogo a ‘**chat**’ per attivare il modem e per superare la procedura iniziale.

Esistono diversi programmi per Dos in grado di gestire in qualche modo il protocollo PPP, ma sembra essere solo la realizzazione DOS PPP ¹ ad avere il pregio di essere semplice e compatibile con i driver di pacchetto Ethernet.

L’archivio contenente DOS PPP dovrebbe essere accessibile dall’indirizzo <ftp://ftp.simtel.net/pub/simtelnet/msdos/pktdrvr/dosppp05.zip>.

Composizione

Il pacchetto di distribuzione di DOS PPP si compone di alcuni programmi, dove i più importanti sono:

- ‘**EPPPD.EXE**’

il programma residente in memoria che svolge il ruolo del demone PPP tradizionale, emulando una scheda Ethernet;

- **'CHAT . EXE'**

il programma utilizzato attraverso **'EPPPD . EXE'** per comandare il modem.

Questi due programmi emulano il più possibile i loro progenitori per Unix: **'pppd'** e **'chat'**, tenendo conto di alcuni aggiustamenti dovuti alle carenze del Dos.

Configurazione e script

«

La configurazione di DOS PPP segue idealmente quella del demone Unix, con la differenza che i file hanno nomi e collocazioni differenti. Considerando che si tratta di sistemi Dos, si possono anche semplificare un po' le cose, come descritto nel seguito.

- Il file **'PPPD . CFG'**, collocato nella stessa directory in cui si trova l'eseguibile **'EPPPD . EXE'**, oppure nella directory corrente, rappresenta in pratica quello che tradizionalmente è il file **'/etc/ppp/options'**.

Naturalmente, si possono usare anche opzioni della riga di comando, le quali prevalgono sulle opzioni fissate con il file di configurazione.

- I file **'PPPDCOM n . CFG'**, collocati nella directory corrente, permettono di indicare opzioni specifiche per ogni porta seriale: **'PPPDCOM1 . CFG'** per **'COM1:'**, **'PPPDCOM2 . CFG'** per **'COM2:'**, ecc. In questo modo si emulano i file di configurazione **'/etc/ppp/options/options.tty*'** tradizionali nei sistemi Unix.

Dai file di configurazione è esclusa la presenza di qualcosa che serva per contenere i segreti PAP e CHAP. Per queste informazioni sono state aggiunte delle opzioni da inserire nei file di configurazione normali.

Dal momento che non c'è un modo migliore per fare sapere quali sono le caratteristiche IP della connessione che si instaura, viene generato automaticamente lo script '**IP-UP.BAT**', il cui unico scopo è quello di inizializzare alcune variabili di ambiente:

```
SET MYIP=indirizzo_ip_locale
SET REMIP=indirizzo_ip_della_controparte
SET NETMASK=maschera_di_rete
SET PEERMRU=valore_MRU_della_controparte
```

Questo dovrebbe facilitare la realizzazione di un altro script che generi al volo i file di configurazione degli applicativi che si intendono usare. Per esempio, volendo realizzare il file di configurazione '**WATTCP.CFG**' per i programmi che incorporano la libreria WATTCP, si potrebbe procedere come si vede nell'esempio seguente:

```
CALL IP-UP.bat
ECHO MY_IP=%MYIP% > WATTCP.CFG
ECHO GATEWAY=%REMIP% >> WATTCP.CFG
ECHO NETMASK=%NETMASK% >> WATTCP.CFG
ECHO NAMESERVER=195.210.91.1 >> WATTCP.CFG
ECHO MSS=512 >> WATTCP.CFG
```

Si può osservare che lo script, oltre a tradurre le variabili di ambiente in direttive del file '**WATTCP.CFG**', aggiunge anche le direttive necessarie per definire il server e per definire la dimensione massima dei segmenti di pacchetto.

Su **'CHAT.EXE'** non c'è nulla di speciale, tranne il fatto che questo programma non può funzionare da solo, ma deve trovarsi sotto il controllo di **'EPPPD.EXE'**. Le opzioni sono molto simili alla versione originale per i sistemi Unix. In generale vale la pena di utilizzare l'opzione **'-v'** per vedere cosa succede durante l'avvio della connessione.

Opzioni particolari per il PPP

«

Il programma **'EPPPD.EXE'** accetta la maggior parte delle opzioni delle vecchie edizioni di **'pppd'** per i sistemi Unix. Per verificare quali sono le opzioni disponibili basta leggere la documentazione allegata, che riproduce la pagina di manuale relativa.

- `user utente`

`passwd parola_d'ordine`

DOS PPP è in grado di gestire esclusivamente l'autenticazione PAP, ma senza l'ausilio di un file dei segreti. In pratica, si fa uso delle opzioni **'user'** e **'passwd'**, con le quali si fornisce il nominativo utente e la parola d'ordine, senza altre specifiche.

- `pktvec irq`

Dal momento che si tratta di driver di pacchetto, DOS PPP si avvale di un IRQ software che può essere scelto esplicitamente, oppure può essere definito automaticamente dal programma. L'opzione **'pktvec'** permette di fissare il valore di tale

IRQ, assegnando valori esadecimali nella forma ‘**0xnn**’. Il valore predefinito usuale è 60_{16} .

Connessione in pratica

Si suppone di avere la possibilità di collegarsi a un servizio di accesso a Internet che ha le caratteristiche seguenti:

- telefono 0987 6543210;
- utenza ‘**tizio**’;
- parola d’ordine ‘**asdfghjk**’;
- indirizzo IP del DNS primario 123.123.123.1.

Inoltre, si utilizza la prima porta seriale, ovvero ‘COM1:’, che viene configurata per una velocità di 57600 bit/s. Si realizza il file ‘PPPD.CFG’ con il contenuto seguente:

```
com1
57600
user tizio
passwd asdfghjk
connect "chat -v '' ATZ OK ATX3 OK ATDT9876543210 CONNECT '' "
```

In questo modo, quando si avvia ‘**EPPPD.EXE**’, questo avvia prima ‘**CHAT.EXE**’ in modo da inizializzare il modem, comporre il numero telefonico e attendere la connessione; successivamente, l’autenticazione avviene attraverso il protocollo PAP.

Si può osservare l’opzione ‘**-v**’ di ‘**CHAT.EXE**’, che serve per vedere i messaggi scambiati tra questo programma e il modem, durante le operazioni. La conoscenza di questi dettagli serve per correggere eventualmente la stringa, in base al comportamento effettivo del modem.²

Una volta instaurata la connessione, **'EPPPD.EXE'** crea il file **'IP-UP.BAT'**, che può essere sfruttato come è già stato visto in precedenza da un altro script che generi i file di configurazione necessari agli altri applicativi, specificando così anche il DNS primario e la dimensione massima del segmento (MSS). L'esempio seguente mostra uno script necessario a generare un file di configurazione per gli applicativi che usano la libreria WATTCP:

```
CALL IP-UP.bat
ECHO MY_IP=%MYIP% > WATTCP.CFG
ECHO GATEWAY=%REMIP% >> WATTCP.CFG
ECHO NETMASK=%NETMASK% >> WATTCP.CFG
ECHO NAMESERVER=123.123.123.1 >> WATTCP.CFG
ECHO MSS=512 >> WATTCP.CFG
```

Per concludere la connessione, si usa il programma **'TERMIN.COM'**, che viene distribuito anche assieme a DOS PPP, ma per questo occorre conoscere l'indirizzo IRQ software utilizzato da **'EPPPD.EXE'**. Per esempio, se si tratta dell'indirizzo IRQ 60₁₆ (quello predefinito), basta procedere come segue:

```
C :>TERMIN 0x60 [Invio]
```

¹ **DOS PPP** software non libero: non è consentita la modifica e nemmeno la commercializzazione

² Naturalmente, nello stesso modo si potrebbe realizzare un accesso di tipo tradizionale, in cui sia **'CHAT.EXE'** a inviare il nominativo utente e la parola d'ordine. Tuttavia, è sempre meno probabile che un fornitore di accesso a Internet utilizzi ancora tale vecchia procedura.

Introduzione a NOS-KA9Q -- IPv4 per Dos



Preparazione	845
Configurazione con il file «AUTOEXEC.NET»	846
Driver di pacchetto	848
Avvio del sistema NOS (NET.EXE)	848
Conclusione del funzionamento del sistema NOS	850
Guida interna	851
Interfacce, instradamento e nomi	851
Comandi «attach» e «detach»	852
Comando «ifconfig»	854
Comando «route»	855
Comando «domain»	856
Gestione delle sessioni	857
Attività nel sistema locale	859
Gestione della rete e delle connessioni	860
Indirizzi IP e indirizzi MAC	860
Ping e instradamento	861
Varie	862
NOS come cliente	862
Cliente TELNET	863
Cliente FTP	863
NOS come servente	864

Registrazione degli eventi	865
Servente FTP	865
NOS come router IPv4	867

NOS ¹ è una sorta di sistema operativo per le reti IPv4 nato per soddisfare le esigenze dei radioamatori. Se si considerano l'età e il fatto che funziona perfettamente su un sistema operativo Dos, si tratta di un applicativo eccezionale quando si dispone di hardware molto vecchio. La sigla KA9Q è il nominativo da radioamatore dell'autore originale di questo programma, Phil Karn, ma spesso si fa riferimento a questo software indifferentemente con le sigle KA9Q, NOS o qualcosa che termina per *NOS.²

Esistono diverse interpretazioni del sistema NOS-KA9Q; probabilmente il riferimento migliore per ottenere il materiale necessario è il deposito Simtel.Net, che ospita una directory apposita per questo: <ftp://ftp.simtel.net/pub/simtelnet/msdos/tcpip/>. In particolare è necessario prelevare il file contenente l'eseguibile '**NET.EXE**', che potrebbe avere un nome simile a 'e920603.zip' (dove il numero corrisponde alla data ed eventualmente potrebbe essere sostituito da una versione più recente) e poi conviene prelevare altri file per ottenere della documentazione: 'intronos.zip', 'ka9qbgn.zip' e 'nos_slfp.zip' (questo ultimo file può essere utile soprattutto per vedere come potrebbe essere effettuata una connessione PPP attraverso la porta seriale e il modem).

La versione 920603, corrispondente al file 'e920603.zip', è adatta ad architetture x86-16. Probabilmente ciò vale anche per qualche versione più recente, ma si deve fare attenzione: la versione 951123 è fatta per x86-32. Se non si riesce a trovare una versione del programma '**NET.EXE**' abbastanza vecchia, si può provare a usare quella contenuta nel pacchetto 'nos_slfp.zip'. In questo capitolo si fa riferimento a una versione di NOS per architetture modeste (i286 o inferiori).

NOS, una volta avviato, prende il controllo del sistema e i comandi che si impartiscono sono interpretati da questo, senza passare per il Dos sottostante. Anche per questa ragione si introduce l'uso di NOS in un capitolo separato, rispetto a quello già dedicato agli applicativi Dos (capitolo [u59](#)).

Qui si mostrano le caratteristiche «normali» di NOS, nel senso che di questo sistema di rete sono state realizzate un'infinità di varianti. Evidentemente, il NOS che si può trovare può corrispondere o meno alle caratteristiche che vengono descritte qui. Se il pacchetto NOS che si trova contiene qualche file di documentazione, conviene leggerlo per verificare che tutto corrisponda a quanto previsto.

Preparazione

Anche se non si intendono sfruttare a fondo tutte le possibilità di NOS, conviene creare tutte le directory previste da questo mini sistema di rete. Se non si vuole fare fatica nella configurazione, conviene

predisporre quelle seguenti, che riguardano le versioni «normali» di NOS:

```
C:\SPOOL
C:\SPOOL\HELP
C:\SPOOL\MAIL
C:\SPOOL\MQUEUE
C:\SPOOL\RQUEUE
C:\SPOOL\NEWS
```

Come si può intuire, si tratta di spazi predisposti per la gestione della posta elettronica; cosa che comunque non viene mostrata in questo capitolo.

Volendo utilizzare una posizione diversa, nello stesso disco o in un altro, occorre almeno mantenere la stessa struttura; per esempio come nel modo seguente, tenendo conto che occorre avviare il programma '**NET.EXE**' specificando questa variante nelle opzioni.

```
D:\NOS\SPOOL
D:\NOS\SPOOL\HELP
D:\NOS\SPOOL\MAIL
D:\NOS\SPOOL\MQUEUE
D:\NOS\SPOOL\RQUEUE
D:\NOS\SPOOL\NEWS
```

Configurazione con il file «AUTOEXEC.NET»

«

Nella directory utilizzata come punto di inizio della gerarchia del sistema NOS, va collocato il file di configurazione '**AUTOEXEC.NET**'. Questo rappresenta semplicemente una sequenza di comandi NOS da eseguire prima di mostrare l'invito all'utente. È abbastanza importante predisporre questo file, per non dover ogni volta ridefinire la configurazione delle interfacce e gli instradamenti relativi.

Ovviamente, per sapere come predisporre questo file occorre conoscere i comandi del sistema NOS. Per cominciare si tenga presente

che sono ammessi i commenti prefissati dal simbolo ‘#’ e terminati dalla fine della riga in cui appaiono; inoltre, se si utilizza un elaboratore appartenente alla famiglia «AT», cioè quelli che hanno un’architettura i286 o superiore, può essere utile indicare il comando ‘**isaton**’. Per il momento si osservi l’esempio seguente, che si riferisce all’uso di una scheda di rete gestita attraverso un driver di pacchetto di quelli descritti nel capitolo precedente ([u59](#)).

```
# Se non si tratta di un elaboratore compatibile IBM AT (o superiore),
# la riga seguente deve essere commentata o eliminata.
#isat on

# Configurazione dell’interfaccia di rete utilizzando il
# riferimento al packet driver (il nome ethernet0 viene stabilito qui,
# e non si tratta di una convenzione di NOS).
attach packet 0x7e ethernet0 8 1500

# Definizione dell’indirizzo IP dell’interfaccia di rete.
ifconfig ethernet0 ipaddress 192.168.1.10
ifconfig ethernet0 netmask 255.255.255.0

# Instradamento.
route add 192.168.1.0/24 ethernet0
route add default ethernet0 192.168.1.254

# DNS
domain addserver 192.168.1.1
domain suffix brot.dg.

# Servizi abilitati.
start discard
start echo
start finger
start ftp
```

Driver di pacchetto

«

Il sistema NOS richiede per funzionare che le interfacce di rete da utilizzare siano controllate da un driver di pacchetto, tranne nei casi in cui è in grado di gestirselo da solo. NOS può utilizzare i driver di pacchetto già mostrati nel capitolo [u59](#) e altri specifici, come nel caso del file 'nos_slfp.zip' che contiene il necessario per gestire una connessione PPP partendo dal controllo della porta seriale e del modem.

Per non appesantire troppo la presentazione del sistema NOS, vengono mostrati solo esempi che fanno riferimento a una scheda di rete gestita attraverso un driver di pacchetto configurato in modo da utilizzare l'indirizzo IRQ $7E_{16}$ per comunicare con le applicazioni. Volendo fare il solito esempio della scheda NE2000 configurata per usare l'indirizzo IRQ 11 e la porta di I/O 300_{16} , si tratta di usare il comando seguente:

```
NE2000.COM 0x7e 0x0b 0x300
```

Avvio del sistema NOS (NET.EXE)

«

Tutto il sistema NOS è inserito in un solo eseguibile Dos: '**NET.EXE**'. All'avvio del programma può essere conveniente utilizzare qualche opzione.

```
NET [-b] [-s n_porte] [-d directory_nos] [file_configurazione]
```

Dopo l'avvio, il sistema NOS mostra alcune informazioni riferite alla versione e quindi l'invito a inserire dei comandi:

```
KA9Q NOS version 910618 -> 911007 (ghm/was)
Copyright 1990 by Phil Karn, KA9Q
net> _
```

Qui viene mostrata una versione particolarmente vecchia del programma; se si trattasse di un'edizione specifica per microprocessori i386 o superiori, tale informazione apparirebbe tra quelle che precedono l'invito.

Tutti i comandi che vengono descritti nelle sezioni successive devono essere impartiti al sistema NOS attraverso l'invito '**net>**', oppure possono essere collocati nel file di configurazione (di solito '**AUTOEXEC.NET**').

Opzione o argomento	Descrizione
-b	Con questa opzione si costringe NOS ad aggiornare lo schermo attraverso le funzioni del BIOS, anziché accedendo direttamente alla memoria video. Ciò rallenta le operazioni, ma può essere necessario in alcune circostanze, quando si vede che la visualizzazione sullo schermo non funziona come ci si aspetterebbe.
-s <i>n_porte</i>	In condizioni normali, NOS gestisce un massimo di 40 connessioni (un massimo di 40 porte). Se si vuole modificare questo valore si può intervenire con l'opzione ' -s '.
-d <i>directory_nos</i>	Se la struttura di directory che richiede NOS si trova a partire da una posizione differente dalla directory radice del disco 'C:', l'opzione ' -d ' permette di indicarlo esplicitamente.

Opzione o argomento	Descrizione
<i>file_configurazione</i>	Se si vuole indicare esplicitamente il file di configurazione (che di solito dovrebbe essere ' <i>directory_nos</i> \AUTOEXEC.NET'), questo può essere inserito come ultimo argomento della riga di comando, senza l'indicazione di un'opzione apposita.

Segue la descrizione di alcuni esempi.

- `C:\> NET [Invio]`

Avvia il sistema NOS utilizzando la gerarchia che si articola a partire dalla directory radice del disco 'C:' ('C:\SPOOL*') e il file di configurazione 'C:\AUTOEXEC.NET'.

- `C:\> NET -d C:\NOS [Invio]`

Avvia il sistema NOS utilizzando la gerarchia che si articola a partire dalla directory 'C:\NET\' ('C:\NET\SPOOL*') e il file di configurazione 'C:\NET\AUTOEXEC.NET'.

- `C:\> NET -d C:\NOS C:\NOS.RC [Invio]`

Avvia il sistema NOS utilizzando la gerarchia che si articola a partire dalla directory 'C:\NET\' ('C:\NET\SPOOL*') e il file di configurazione 'C:\NOS.RC'.

Conclusione del funzionamento del sistema NOS

«

Dal momento che il sistema NOS si comporta come una shell, si può intuire il modo attraverso il quale si conclude il suo funzionamento: con il comando '**exit**':

```
net> exit [Invio]
```

Guida interna

Come suggerisce lo stesso NOS quando si inserisce un comando errato, è disponibile una mini guida interna costituita dall'elenco dei comandi. Si ottiene con '**help**', oppure semplicemente con '?'. Non è molto, dal momento che non viene mostrata la sintassi rispettiva, comunque è sempre meglio di nulla.

```
net> help [Invio]
```

Interfacce, instradamento e nomi

Le cose più importanti da fare per poter utilizzare il sistema NOS, sono la definizione delle interfacce, l'instradamento e la risoluzione dei nomi. Le interfacce vengono «attaccate» attraverso il comando '**attach**', quindi vengono configurate attraverso '**ifconfig**', alla fine l'instradamento viene definito attraverso il comando '**route**'. NOS non ha funzionalità di DNS, a parte la possibilità di risolvere alcuni nomi a dominio per conto proprio, ma si può avvalere di un DNS esterno attraverso il comando '**domain**'.

I comandi che vengono descritti in queste sezioni sono usati generalmente per la configurazione attraverso il file '**AUTOEXEC.NET**'. Ciò dovrebbe essere intuitivo dato il tipo di operazioni che si svolgono con questi.

Comandi «attach» e «detach»

«

Attraverso il comando ‘**attach**’ si possono definire le interfacce utilizzate. Di solito l’eseguibile ‘**NET.EXE**’ è predisposto per la gestione delle porte seriali (‘**asy**’) e per l’uso di un driver di pacchetto esterno.

```
attach asy i/o irq { ppp | slip } nome_interfaccia dim_buffer mtu bps [c] [r] [v]
```

```
attach packet irq nome_interfaccia coda_trasmissione mtu
```

Quello che si vede rappresenta la sintassi per la definizione di un’interfaccia seriale (PPP, SLIP, o altre che non sono state indicate) e per un’interfaccia comandata da un driver di pacchetto esterno.

Nel caso del tipo ‘**asy**’, cioè della connessione seriale, il numero di IRQ e l’indirizzo di I/O si riferiscono a quelli della porta seriale stessa; inoltre, gli ultimi argomenti sono la velocità espressa in bit/s (bps) e una stringa facoltativa dove possono apparire le lettere ‘**c**’, ‘**r**’ e ‘**v**’. Queste lettere rappresentano tre modalità: se appare la ‘**c**’ si utilizza il protocollo RTS/CTS; se appare la ‘**r**’ si abilita la sensibilità al segnale CD (*Carrier detect*); se appare la ‘**v**’ si abilita la compressione Van Jacobson delle intestazioni TCP/IP, ma solo per le connessioni SLIP.

Con le interfacce gestite da un driver di pacchetto esterno diventa tutto più facile, dal momento che la cosa più importante è solo l’indicazione dell’indirizzo IRQ software (quello che serve a individuare il driver).

Per eliminare un'interfaccia si utilizza invece il comando '**detach**' secondo la sintassi seguente:

```
detach interfaccia
```

Segue la descrizione di alcuni esempi.

- ```
attach packet 0x7e ethernet0 8 1500
```

Utilizza un driver di pacchetto per gestire una scheda Ethernet. L'indirizzo IRQ per comunicare con il driver è  $7E_{16}$ ; viene definito il nome '**ethernet0**' per fare riferimento a questa scheda; si pone il limite di otto pacchetti per la coda di trasmissione; si stabilisce l'unità massima di trasmissione in 1500 byte.

- ```
attach asy 0x3f8 4 slip sl0 1024 256 9600
```

Questo esempio è tratto dalla documentazione di NOS e si riferisce a una connessione SLIP attraverso la porta seriale individuata dall'indirizzo di I/O $3F8_{16}$ e dall'indirizzo IRQ 4. Il nome che viene attribuito è '**s10**'; viene definito un *buffer* di ricezione di 1024 byte; la dimensione massima dei pacchetti trasmessi è di 256 byte; la velocità della porta seriale è di 9600 bit/s.

- ```
attach asy 0x3f8 4 ppp pp0 4096 1500 9600 r
```

Anche questo esempio è tratto dalla documentazione di NOS e si riferisce a una connessione PPP attraverso la porta seriale individuata dall'indirizzo di I/O  $3F8_{16}$  e dall'indirizzo IRQ 4. Il nome che viene attribuito è '**pp0**'; viene definito un *buffer* di ricezione di 4096 byte; la dimensione massima dei pacchetti trasmessi è di 1500 byte; la velocità della porta seriale è di 9600 bit/s; viene abilitato il controllo della linea CD del modem.

- `detach ethernet0`

Elimina l'interfaccia '**ethernet0**'.

## Comando «ifconfig»

«

Attraverso il comando '**ifconfig**' si possono configurare le interfacce definite in precedenza con il comando '**attach**'. Il comando può assumere diverse forme, ma in particolare sono importanti gli schemi seguenti:

```
ifconfig [nome_interfaccia]
```

```
ifconfig nome_interfaccia ipaddress indirizzo_ip
```

```
ifconfig nome_interfaccia netmask maschera_ip
```

Utilizzando il comando da solo, senza argomenti, si ottiene la visualizzazione dello stato di tutte le interfacce di rete, comprese quelle predefinite; se si specifica il nome di un'interfaccia, il risultato si limita allo stato di questa. La seconda e la terza modalità servono invece per abbinare un indirizzo IP e una maschera di rete all'interfaccia.

Segue la descrizione di alcuni esempi.

- `ifconfig ethernet0`

Mostra lo stato dell'interfaccia '**ethernet0**', che in precedenza è stata dichiarata con questo nome.

- `ifconfig ethernet0 ipaddress 192.168.1.10`

Abbina all'interfaccia l'indirizzo IP 192.168.1.10.

- `ifconfig ethernet0 netmask 255.255.255.0`

Abbina all'interfaccia la maschera IP 255.255.255.0.

## Comando «route»

Il comando '**route**' permette di definire gli instradamenti attraverso le interfacce di rete configurate precedentemente, specificando eventualmente anche i router necessari a raggiungere le reti esterne. «

```
route
```

```
route add indirizzo_ip/n_bit_maschera nome_interfaccia [router]
```

```
route add default nome_interfaccia [router]
```

La sintassi mostrata rappresenta una semplificazione del comando necessario a definire un instradamento. La coppia *indirizzo\_ip/n\_bit\_maschera* è un modo per rappresentare l'indirizzo di una rete in modo compatto: il numero di bit rappresenta quanti bit iniziali devono essere posti a uno nella maschera di rete.

Per eliminare un instradamento si utilizza la forma seguente:

```
route drop indirizzo_ip/n_bit_maschera
```

```
route drop default
```

Segue la descrizione di alcuni esempi.

- ```
route
```

Mostra l'instradamento delle interfacce.

```
route add 192.168.1.0/24 ethernet0
```

- Definisce l'instradamento per la rete identificata dagli indirizzi 192.168.1.* attraverso l'interfaccia '**ethernet0**'.

```
route add default ethernet0 192.168.1.254
```

Definisce l'instradamento predefinito attraverso il router 192.168.1.254.

Comando «domain»

«

Il comando '**domain**' permette di definire quali sono i servizi DNS a cui il sistema NOS può rivolgersi; permette anche di configurare il loro utilizzo e di definire eventualmente una risoluzione locale per alcuni indirizzi. Qui viene mostrato solo come dichiarare l'uso dei servizi DNS e il dominio predefinito.

```
domain addserver indirizzo_ip_DNS
```

```
domain suffix [suffisso_predefinito]
```

Segue la descrizione di alcuni esempi.

- ```
domain addserver 192.168.1.1
domain addserver 192.168.1.2
```

Dichiara l'uso del servizio DNS collocato presso i nodi 192.168.1.1 e 192.168.1.2.

- ```
domain suffix brot.dg.
```

- Dichiara che in caso di nomi a dominio incompleti viene aggiunto il suffisso *brot.dg*.

```
domain suffix
```

Mostra il suffisso predefinito per i nomi a dominio.

Gestione delle sessioni

Il sistema NOS può gestire diverse sessioni di lavoro, corrispondenti ad altrettante attività che implicano l'instaurarsi di una connessione. Per esempio si possono gestire diverse connessioni TELNET simultaneamente e lo stesso vale per l'utilizzo del protocollo FTP. Tutto questo funziona in modo paragonabile al sistema delle console virtuali di GNU/Linux: con il sistema NOS c'è una finestra per la modalità di comando, dove si trova l'invito, attraverso la quale si impartiscono i comandi, e le finestre delle sessioni che vengono aperte automaticamente in base al tipo di comando che viene dato.

Quando ci si trova a interagire con una sessione è possibile tornare alla finestra della modalità di comando attraverso il tasto [*F10*] (vale solo per il NOS che si basa sul Dos) e poi, da lì è possibile tornare a una sessione attraverso il comando '**session**'. Da questo si comprende che le sessioni sono numerate, cosa che avviene in modo automatico. Una di queste è anche la sessione attiva, ovvero quella a cui si potrebbe fare riferimento quando non se ne specifica il numero.

Comando	Descrizione
<code>session [n_sessione]</code>	Il comando ' session ' permette di tornare a una sessione ancora attiva, specificandone il numero. Se questo non viene indicato, si ottiene l'elenco delle sessioni esistenti.
<code>session</code>	Elenca le sessioni esistenti.
<code>session 1</code>	Torna alla prima sessione.
<code>close [n_sessione]</code>	Il comando ' close ' interrompere una connessione TCP attraverso l'invio di un pacchetto FIN (che serve a chiudere una connessione del genere). Il risultato che si ottiene di solito è che la sessione corrispondente termina.
<code>close</code>	Interrompe la connessione della sessione corrente.
<code>close 1</code>	Interrompe la connessione della prima sessione.
<code>reset [n_sessione]</code>	Il comando ' reset ' interrompere una connessione TCP attraverso l'invio di un pacchetto RST. Il risultato che si vuole ottenere è la conclusione della sessione corrispondente, ma dal momento che il metodo dell'invio di un pacchetto RST non garantisce l'ottenimento di ciò, sarebbe preferibile utilizzare il comando ' close ' al suo posto.
<code>abort [n_sessione]</code>	Il comando ' abort ' permette di interrompere un'operazione di carico o scarico dati attraverso una sessione FTP. La sessione in questione non viene chiusa.<
<code>abort</code>	Interrompe le operazioni di carico-scarico nella sessione corrente, purché di FTP.

Comando	Descrizione
<code>abort 1</code>	Interrompe le operazioni di carico-scarico nella prima sessione, purché sia di FTP.

Attività nel sistema locale

Dal momento che non è possibile intervenire direttamente sul sistema operativo sottostante senza interrompere le connessioni che eventualmente fossero state instaurate, NOS deve incorporare alcune funzionalità che non hanno attinenza con la rete, ma che sono indispensabili a livello pratico.

Per quanto riguarda i percorsi delle directory possono essere indicati utilizzando sia le barre oblique inverse (‘\’) che quelle normali (‘/’) per la separazione dei nomi che li compongono.

Comando	Descrizione
<code>cd [percorso]</code>	Permette di cambiare la directory corrente nel sistema sottostante. Se viene utilizzato senza l’indicazione del percorso, si ottiene la visualizzazione della directory corrente.
<code>pwd</code>	Mostra quale sia la directory corrente.
<code>dir [percorso]</code>	Elenca il contenuto della directory corrente.
<code>mkdir [percorso]</code>	Crea una directory nel sistema operativo sottostante.
<code>rmdir [percorso]</code>	Elimina una directory nel sistema operativo sottostante.
<code>more file...</code>	Scorre il testo di uno o più file del sistema operativo sottostante, facendo una pausa tra le schermate.

Comando	Descrizione
<code>rename <i>file_origine file_destinazione</i></code>	Rinomina o sposta un file del sistema operativo sottostante.
<code>delete <i>file...</i></code>	Elimina i file indicati negli argomenti dal sistema operativo sottostante.
<code>! shell</code>	Sospende il funzionamento di NOS per aprire una shell del sistema operativo sottostante ('COMMAND.COM').

Gestione della rete e delle connessioni

«

Oltre a quanto visto inizialmente per ciò che riguarda la definizione delle interfacce, la loro configurazione, l'instradamento e la risoluzione dei nomi, c'è una serie di comandi e di funzionalità per la gestione della rete.

Indirizzi IP e indirizzi MAC

«

```
arp
```

```
arp flush
```

Il comando **'arp'** permette di conoscere il contenuto della tabella di trasformazione degli indirizzi IP in indirizzi fisici e viceversa. Questa viene costruita automaticamente dal sistema, durante il suo funzionamento. Sono disponibili degli argomenti particolari per inserire a forza delle voci nella tabella, anche se questa operazione non dovrebbe essere necessaria. In particolare, il comando **'arp flush'**

svuota la tabella attuale, costringendo il sistema NOS a ricominciare a costruirselo.

Ping e instradamento

Attraverso il comando **'ping'** si può inviare una richiesta di eco utilizzando il protocollo ICMP. Questo è il modo consueto per verificare che sia presente un certo nodo nella rete. In generale conviene utilizzare soltanto la sintassi seguente, con la quale viene inviata un'unica richiesta. <<

```
ping nodo
```

Per verificare il percorso dei pacchetti lungo la rete si può utilizzare il comando **'hop'**. Il comando normale si articola nel modo seguente:

```
hop check nodo
```

Tuttavia, si può intervenire su alcuni parametri di funzionamento di questo comando: il TTL (*Time to live*),

```
hop maxttl max_salti
```

l'attesa massima,

```
hop maxwait n_secondi
```

e il numero di pacchetti di prova che vengono inviati a ogni nodo.

```
hop queries n_pacchetti
```

Infine, è possibile abilitare o meno la visualizzazione di informazioni aggiuntive:

```
hop trace [on|off]
```

Varie

«

```
hostname [nome]
```

Attraverso il comando ‘**hostname**’ è possibile definire o visualizzare il nome attribuito al nodo. Questo dovrebbe corrispondere alla parte finale del nome a dominio, ma in ogni caso serve solo nei messaggi di presentazione del sistema.

```
socket [n_porta]
```

Attraverso il comando ‘**socket**’ è possibile conoscere lo stato delle porte. Utilizzandolo senza argomenti si ottiene l’elenco delle porte utilizzate, generalmente quelle dei servizi in ascolto ed eventualmente anche quelle gestite dalle sessioni in cui si utilizzano dei clienti di qualche tipo, mentre specificando una porta precisa si ottengono le statistiche sul traffico intrattenuto.

NOS come cliente

«

L’uso più importante del sistema NOS è quello di cliente in grado di utilizzare i servizi fondamentali di una rete TCP/IP. Si tratta principalmente di TELNET e FTP.

Cliente TELNET

Il sistema NOS permette di attivare una sessione TELNET verso un altro sistema che offra la possibilità di accedere attraverso questo tipo di protocollo. Purtroppo, il tipo di terminale corrispondente alla sessione TELNET è molto modesto, tanto che nelle versioni più limitate di NOS non si possono usare nemmeno i tasti freccia.

```
telnet nodo
```

Quando si utilizza questo tipo di cliente TELNET per accedere a un nodo corrispondente a un elaboratore GNU/Linux, il tipo di terminale che si vede nella variabile **'TERM'** è **'network'**, che però non corrisponde ad alcuna voce nel sistema Terminfo o nel sistema Termcap. Eventualmente si può cambiare questo nome con **'ansi'**, o **'ansi-mono'** se si preferisce.

Da una sessione TELNET è possibile tornare alla modalità di comando premendo il tasto [*F10*]. Per ritornare alla sessione con TELNET, si può poi utilizzare il comando **'session'**.

Cliente FTP

Il sistema NOS permette di attivare una sessione FTP. Una volta avviata, si ha a disposizione un cliente FTP tradizionale, con comandi molto simili a quelli del programma **'ftp'** dei sistemi Unix.

```
ftp nodo
```

L'unico vero difetto sta nel sistema operativo sottostante: utilizzando il Dos i nomi dei file che vengono salvati sono ridotti al modello «8.3».

È importante ricordare di modificare sempre il tipo di trasferimento dati, in modo che sia di tipo binario (*image*): **'type i'**.

NOS come servente

«

I servizi offerti da NOS sono limitati e comunque dipendono dalla versione di questo sistema. Questi servizi devono essere abilitati attraverso il comando **'start'**. Dal momento che dipende dalla versione di NOS se un tipo di servizio è disponibile o meno, attraverso il comando **'start ?'** si ottiene l'elenco di questi.

In generale non conviene avere grandi pretese; probabilmente è il caso di attivare sempre i servizi **'discard'**, **'echo'**, **'ftp'** e **'finger'** (ammesso che questo ultimo possa avere senso).

```
start discard
start echo
start finger
start ftp
```

Per converso, volendo disattivare un servizio basta utilizzare il comando **'stop'** nello stesso modo.

Registrazione degli eventi

NOS permette di annotare gli accessi in un registro abbastanza semplificato. Si attiva questa funzionalità attraverso il comando **'log'**:

```
log [stop | file_delle_registrazioni]
```

Per esempio, per attivare la registrazione degli accessi nel file 'C:\ACCESSI.LOG', si può usare il comando seguente:

```
log c:\accessi.log
```

Come si può intuire, il comando **'log stop'** termina l'attività di registrazione degli accessi, senza interferire con gli accessi stessi. Infine, il comando **'log'** senza argomenti permette di sapere se questo sia attivo e in tal caso su quale file vengono fatte le annotazioni.

Servente FTP

Per abilitare il servizio FTP, oltre che usare il comando **'start ftp'**, occorre predisporre un file di autorizzazioni: 'ftpusers' collocato nella directory radice del servizio NOS. Il file deve contenere delle righe scomposte in quattro campi separati da uno o più spazi e si possono indicare anche dei commenti che si introducono con il simbolo '#'. <<

```
utente parola_d'ordine percorso permessi
```

I quattro campi sono obbligatori e il significato è intuitivo:

1. **utente** serve a specificare il nome dell'utente che può accedere;

2. *parola_d'ordine* rappresenta la parola d'ordine in chiaro necessaria per l'accesso -- se si utilizza un asterisco ('*'), viene accettata qualunque parola d'ordine;
3. *percorso* indica la directory a partire dalla quale si concede l'accesso all'utente;
4. *permessi* è un numero che esprime i permessi consentiti all'utente.

I permessi non sono indicati secondo la tradizione Unix, quindi occorre fare attenzione. I permessi sono espressi con un solo numero ottenuto sommandone altri, che comunque si riferiscono alla directory di partenza e a tutte le sottodirectory: uno rappresenta un permesso di lettura; due rappresenta un permesso di creazione (di aggiunta di file senza poter sovrascrivere o eliminare quelli esistenti); quattro rappresenta un permesso di scrittura (o di sovrascrittura). Si osservi l'esempio seguente:

```
tizio tazza \home\tizio 7
caio capperi \home\caio 7
sempronio sempre \progetto 3
ftp * \pub 1
anonymous * \pub 1
```

Gli utenti '**tizio**' e '**caio**' hanno una loro directory personale in cui possono fare quello che vogliono; l'utente '**sempronio**' partecipa a un lavoro che si trova nella directory '**\PROGETTO**' e lì ha la possibilità di immettere file, senza cancellare o sovrascrivere quelli presenti. Infine, gli utenti '**ftp**' e '**anonymous**' accedono con una parola d'ordine qualunque alla directory '**\PUB**', con il solo permesso di lettura.

NOS come router IPv4



NOS funziona perfettamente come router se l'elaboratore in cui si utilizza dispone di più interfacce di rete. A titolo di esempio viene mostrato in che modo potrebbero essere utilizzate due schede di rete compatibili NE2000. Supponendo che queste utilizzino rispettivamente le risorse IRQ 10, I/O 280₁₆, e IRQ 11, I/O 300₁₆, la configurazione del driver di pacchetto (si fa riferimento a quanto descritto nella sezione [u0.1](#)) potrebbe essere quella seguente:

```
NE2000 0x60 0x0a 0x280
NE2000 0x61 0x0b 0x300
```

Supponendo che queste due schede servano a connettere le reti 192.168.1.* e 192.168.2.*, supponendo anche che l'instradamento predefinito passi per il router 192.168.1.254, il file di configurazione di NOS potrebbe contenere in particolare le righe seguenti:

```
# Configurazione delle interfacce di rete.
attach packet 0x60 ethernet0 8 1500
attach packet 0x61 ethernet1 8 1500

# Definizione degli indirizzi IP.
ifconfig ethernet0 ipaddress 192.168.1.10
ifconfig ethernet0 netmask 255.255.255.0
ifconfig ethernet1 ipaddress 192.168.2.10
ifconfig ethernet1 netmask 255.255.255.0

# Instradamento.
route add 192.168.1.0/24 ethernet0
route add 192.168.2.0/24 ethernet1
route add default ethernet0 192.168.1.254
```

Non c'è bisogno di fare altro: l'attraversamento dei pacchetti da un'interfaccia all'altra avviene automaticamente (purché gli instradamenti siano corretti).

- ¹ **NOS** GNU GPL come descritto in <http://www.ka9q.net/code/>
- ² **NOS** è disponibile in varie versioni per diversi sistemi operativi: PMNOS per Presentation Manager (OS/2), AmigaNOS per Amiga e TNOS per GNU/Linux! L'attenzione di questo capitolo è comunque rivolta alle versioni di **NOS** per Dos.

nanoDos



Organizzazione della distribuzione	869
Installazione	870
Configurazione	871
Utilizzo	874
Conclusione	875
Riferimenti	876

nanoDos è una sorta di distribuzione FreeDOS (per architettura x86-16), finalizzata all'utilizzo di qualche servizio di rete essenziale, basata sull'uso delle librerie WATTCIP.

Il software contenuto nella distribuzione ha licenze di vario tipo; in ogni caso si intende che il software sia utilizzabile gratuitamente senza limitazioni. Il lavoro di realizzazione della raccolta nanoDos è semplicemente di pubblico dominio e offerto senza alcuna garanzia e senza sostegno di alcun tipo.

Organizzazione della distribuzione

La distribuzione è composta da un file-immagine, da usare per riprodurre il dischetto di avvio, e da una serie di file con estensione '.zip', da estrarre successivamente all'installazione del contenuto del dischetto di avvio.



Il dischetto di avvio che si ottiene dal file-immagine appena descritto, può essere usato così come si trova, senza installarlo, ma in tal caso è disponibile solo il sistema operativo e un programma per l'accesso a un server TELNET.

Per la precisione sono disponibili due file-immagine: uno adatto per dischetti da 1440 Kibyte (da 90 mm, ovvero da 3,5 in) e uno per i vecchissimi dischetti flessibili da 1200 Kibyte (da 133,35 mm, ovvero da 5,25 in). I nomi dei file sono rispettivamente 'boot.144' e 'boot.120'. Tanto per richiamare la memoria, in un sistema GNU/Linux è possibile riprodurre il dischetto con uno dei due comandi seguenti:

```
# cp boot.144 /dev/fd0 [Invio]
```

```
# cp boot.120 /dev/fd0 [Invio]
```

I file con estensione '.zip' contengono degli applicativi, suddivisi in modo da occupare ognuno una directory separata. Questi file possono essere usati dopo che il contenuto del dischetto di avvio è già stato installato nel disco fisso, per installare le applicazioni rimanenti.

Installazione

«

Una volta realizzato il dischetto di avvio adatto al proprio elaboratore, lo si può utilizzare per l'avvio del sistema operativo, con il quale si può procedere a predisporre la partizione del disco fisso che deve accoglierlo. Il dischetto **non** deve essere protetto contro la scrittura, perché durante il funzionamento vengono creati dei file temporanei.

Il sistema avviato da dischetto si presenta nello stesso modo in cui si presenterebbe una volta installato nel disco fisso, mostrando un menù di funzioni disponibili, che in realtà sono quasi tutte assenti.

Una volta avviato il dischetto, se necessario, si può usare il programma **'FDISK.EXE'** per creare le partizioni; successivamente occorre inizializzare la partizione che deve accogliere il sistema con il programma **'FORMAT.EXE'**:

```
A\> FORMAT C: /S [Invio]
```

Fatto questo, si passa alla copia del dischetto, tale e quale, nel disco 'C:':

```
A\> XCOPY A:\*.* C:\*.* /E /S /H /V [Invio]
```

Se tutto va bene, si può riavviare e vedere che il sistema parte regolarmente dal disco fisso. Se le cose stanno così, si possono copiare i file **'.zip'** in alcuni dischetti, per estrarli successivamente con un comando del genere:

```
C\> UNZIP A:\*.ZIP [Invio]
```

Per ogni file **'.zip'**, si deve ottenere una directory corrispondente che parte dalla radice del disco fisso, con lo stesso nome, ma senza l'estensione **'.zip'**.

Se tutto questo funziona con successo, il sistema nanoDos è pronto.

Configurazione

Per configurare nanoDos è necessario modificare il file **'\AUTOEXEC.BAT'**. <<

Inizialmente vengono definite alcune variabili di ambiente che contengono le informazioni necessarie alla gestione del TCP/IP e dell'interfaccia di rete:

```
...
set MY_IP=172.21.4.1
set NETMASK=255.255.0.0
set GATEWAY=172.21.254.254
set NAMESERVER=172.21.254.254
set HOSTNAME=nanodos

set NIC_DRIVER=\crynwr\ne2000.com
set NIC_IRQ_SOFT=0x7e
set NIC_IRQ=0x0b
set NIC_IO=0x300
...
```

Come si intende, le variabili di ambiente con il nome che corrisponde al modello '**NIC_***' servono alla configurazione relativa all'interfaccia di rete. In condizioni normali è previsto l'uso di una scheda NE2000, attraverso il programma '`\CRYNWR\NE2000.COM`', configurata in modo da usare l'indirizzo IRQ 11 ($0B_{16}$) e l'indirizzo I/O 300_{16} ; inoltre viene specificato l'uso dell'indirizzo IRQ $7E_{16}$ per la comunicazione con le applicazioni. Naturalmente bisogna conoscere come è configurata la propria scheda e se necessario si devono usare i programmi appropriati per configurarla come si desidera.

Le variabili di ambiente precedenti a queste dichiarano l'indirizzo IP dell'elaboratore, la maschera di rete, il router per le altre reti, il server per la risoluzione dei nomi e il nome dell'elaboratore stesso. Più avanti nel file '`\AUTOEXEC.BAT`' queste variabili vengono usate per costruire i vari file di configurazione necessari ai programmi che usano la rete.

Tabella u62.2. Variabili di ambiente usate per la configurazione della rete.

Variabile	Esempio	Descrizione
MY_IP	172.21.4.1	Indirizzo IPv4 dell'elaboratore.
NETMASK	255.255.0.0	Maschera di rete.
GATEWAY	172.21.254.254	Router per le altre reti.
NAMESERVER	172.21.254.254	Servente DNS.
HOSTNAME	nanodos	Nome dell'elaboratore.
NIC_DRIVER	'\CRYNWR\ NE2000.COM'	Programma per la gestione della scheda di rete.
NIC_IRQ_SOFT	7E ₁₆ , 126 ₁₀	Indirizzo IRQ usato per comunicare con le applicazioni (rimane fisso).
NIC_IRQ	0B ₁₆ , 11 ₁₀	Indirizzo IRQ dell'interfaccia di rete (dipende dalla configurazione dell'interfaccia).
NIC_IO	300 ₁₆ , 768 ₁₀	Indirizzo I/O dell'interfaccia di rete (dipende dalla configurazione dell'interfaccia).

Nel file '`\AUTOEXEC.BAT`' si può intervenire anche per modificare la mappa della tastiera, che secondo la configurazione predefinita è quella italiana; diversamente si può usare a mano il programma '`KEYB.EXE`':

```
...
\freedos\keyb it,850,\freedos\key\it.kl
...
```

Eventualmente, c'è da considerare la configurazione del mouse, che inizialmente è predisposta per un mouse PS/2. Per intervenire in questo occorre modificare il file '`\MOUSE.BAT`', eventualmente così per usare un mouse seriale collegato alla prima porta ('`COM1:`')

```
REM Serial mouse at COM1:
\driver\ctmouse.exe /S1
REM PS2 mouse
REM \driver\ctmouse.exe /P
```

Se poi si vuole utilizzare un elaboratore i386 o superiore, può essere conveniente modificare anche il file '`\CONFIG.SYS`', in modo da abilitare l'uso di '`HIMEM.SYS`':

```
files=100
buffers=25
dos=HIGH,UMB
REM country=039,,\freedos\country.sys
lastdrive=z
SHELL=COMMAND.COM /E:2048 /P
STACKS=9,256
device=\driver\himem.exe
REM device=\driver\emm386.exe
```

Si osservi che queste modifiche vanno apportate anche sul dischetto, se si intende usare il sistema minimo senza installarlo nel disco fisso.

Utilizzo



Una volta avviato il sistema operativo, dopo una breve schermata di avvertimento sulla mancanza assoluta di garanzie, appare un menù che riepiloga l'uso dei programmi principali.

Figura u62.6. Menù di nanoDos.

```
.-----  
|LYNX host                |Simple HTTP navigator.  
|ARACHNE http://host/...  |Graphic HTTP navigator for i386.  
|SSH user host           |SSH 1 client.  
|TELNET host             |TELNET client.  
|FTP host                |FTP client.  
|TALK user@host          |Call a user at a specified host.  
|TALK -a                 |Ready to receive a TALK call.  
|PING host               |Ping to host.  
|NSLOOKUP domain         |Domain resolution.  
|NSQUERY -d domain       |Domain resolution with more infos.  
|TROUTE host             |Trace the route to host.  
|FINGER [user]@host      |FINGER client.  
|LPR [queue] host file   |Print the local file to the remote printer.  
|LPQ -Pqueue -Shost      |Query remote printer queue.  
|TCPINFO                 |See network configuration.  
|-----  
|MENU                    |Recall this menu.  
|-----  
|[Ctrl]+[Alt]+[Del]     |Restart the computer.  
\-----
```

Si osservi che Arachne funziona soltanto in un sistema i386 o superiore, dove deve essere stato abilitato l'uso di 'HIMEM.SYS'; inoltre non è molto stabile.

Conclusione

nanoDos è un sistema abbastanza fragile, con il quale ogni tanto si può essere costretti a riavviare l'elaboratore, ma può consentire l'uso efficace di vecchie macchine, sia a scopo didattico, sia a scopo professionale, se ciò che serve è un terminale senza pretese.

Il senso di questo lavoro sta soprattutto nell'organizzazione della configurazione. Da quanto fatto, se si conosce il sistema operativo, è facile realizzare una riduzione che si adatti meglio alle proprie esigenze.

Riferimenti



- *nanoDos*
[*extra/nanoDos/*](#)