

ELKS – Embeddable Linux Kernel Subset

ELKS, acronimo di Embeddable Linux Kernel Subset, è una versione di Linux a 16 bit in grado di funzionare su un processore 8086/88 o superiore.

Caratteristiche di base

ELKS è una versione compatta di Linux adatta a sistemi embedded che supporta buona parte delle funzionalità del kernel di Unix V7.

La sua homepage è <http://elks.sourceforge.net> ed è rilasciato sotto la GNU General Public License. Al momento in cui scrivo, l'ultima versione disponibile è la 0.1.3-pre1 del 13 Luglio 2003.

Per funzionare, come detto prima, esso richiede un processore di classe 8086/88 o superiore ed un quantitativo di RAM pari a 640 KB (applicando particolari modifiche anche meno di 400 KB potrebbero essere sufficienti). ELKS supporta tutti i processori successivi all'8086 di Intel, compresi i NEC V20 e V30 (che differiscono da quelli x86 standard in alcuni punti), ed è possibile abilitare anche la modalità protetta per gli 80286 (il cui supporto è tuttora sperimentale).

Gli applicativi per ELKS sono in realtà ben pochi, ma non per questo non è un ambiente valido: esso costituisce un ottimo strumento didattico per studiare il funzionamento di un sistema operativo e per recuperare computer vecchi ormai abbandonati.

Download di ELKS

Per scaricare ELKS è sufficiente collegarsi al sito <http://elks.sourceforge.net> ed ottenere l'ultima versione del pacchetto delle immagini (al momento in cui scrivo il file è **images-0.1.0.zip**) disponibile.

È possibile scaricare anche EDE - ELKS Distribution Edition, una mini-distribuzione che facilita le operazioni di installazione del sistema.

Avvio del sistema

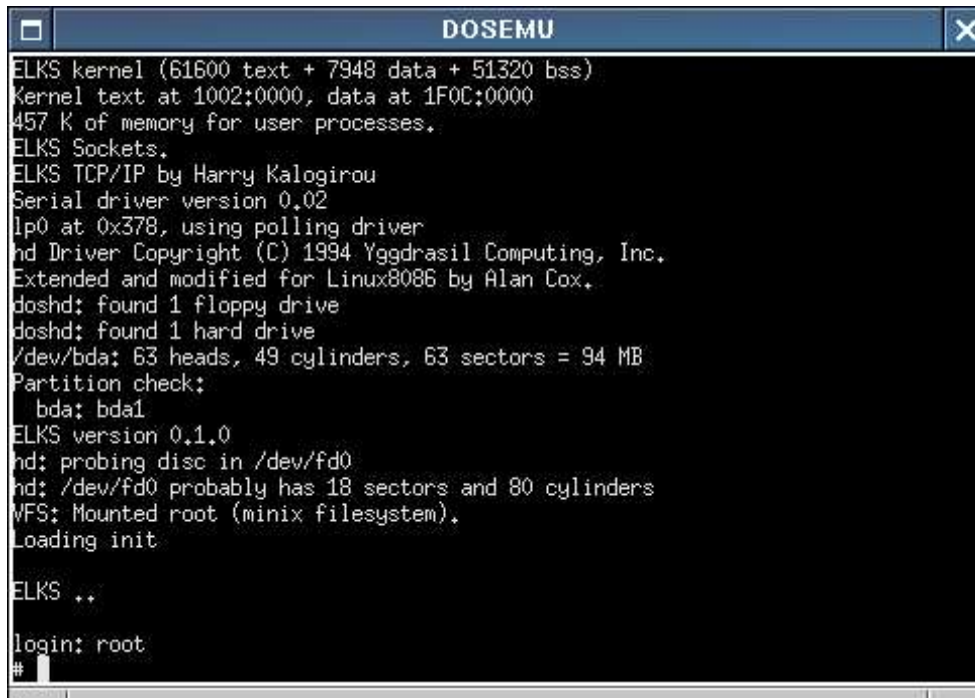
Per avviare ELKS è sufficiente decomprimere il pacchetto images.zip in una cartella temporanea (es. /tmp/elks) e riversare quindi il contenuto del file **comb** (comprendente il kernel ed il filesystem di root) su un floppy disk standard da 1,44 MB con:

```
$ dd if=comb of=/dev/fd0
```

Per utilizzare ELKS ci sono due alternative: servirsi di una macchina fisica oppure utilizzare l'emulatore DOSEMU (vedi il box relativo ad esso per maggiori informazioni su come installarlo ed utilizzarlo).

In entrambi i casi sarà necessario inserire il floppy nel lettore ed avviare la macchina (fisica o virtuale) da esso. A questo punto vi appariranno le schermate classiche di avvio del sistema: le periferiche standard (quali lettori floppy, dischi rigidi e così via) verranno riconosciute agevolmente.

Dopo un po' di tempo il caricamento del sistema verrà completato.
Fate il login come utente root (privo di password).



```
DOSEMU
ELKS kernel (61600 text + 7948 data + 51320 bss)
Kernel text at 1002:0000, data at 1F0C:0000
457 K of memory for user processes.
ELKS Sockets.
ELKS TCP/IP by Harry Kalogirou
Serial driver version 0.02
lp0 at 0x378, using polling driver
hd Driver Copyright (C) 1994 Yggdrasil Computing, Inc.
Extended and modified for Linux8086 by Alan Cox.
doshd: found 1 floppy drive
doshd: found 1 hard drive
/dev/bda: 63 heads, 49 cylinders, 63 sectors = 94 MB
Partition check:
  bda: bda1
ELKS version 0.1.0
hd: probing disc in /dev/fd0
hd: /dev/fd0 probably has 18 sectors and 80 cylinders
MFS: Mounted root (minix filesystem).
Loading init
ELKS ..
login: root
#
```

Figura 1: La schermata di login di ELKS

La shell con cui interagirete è una versione di **ash** opportunamente modificata per poter funzionare in condizioni così estreme. I comandi disponibili sono quelli presenti in tutti i sistemi Unix: si va da quelli per la gestione dei file e delle cartelle come **ls** e **cp** fino ad altri come **mknod** oppure **vi**.

Per uscire dal sistema digitate:

```
# sync
# umount /
```

Quindi premete il bottone di spegnimento della vostra macchina o, nel caso in cui abbiate scelto di usare DOSEMU, digitate:

```
# exitemu
```

Networking – Lato Kernel

Questo sistema operativo supporta in maniera egregia il protocollo TCP/IP, mentre i socket di tipo NANO o UNIX non sono ancora stati implementati completamente. Le uniche connessioni supportate sono quelle di tipo SLIP (eventualmente con compressione CSLIP dei pacchetti TCP/IP), e per questo non sono supportate né quelle PLIP né tantomeno quelle Ethernet.

Per collegare la vostra macchina ad un altro pc dovrete acquistare oppure costruirvi (vedi il box relativo) un **cavo seriale null-modem**.

Esso vi consentirà di instaurare una connessione SLIP con un'altra macchina e di scambiare dati tra esse. La velocità non è elevatissima, ma può essere più che soddisfacente per macchine così vecchie.

Networking - Realizzare una connessione SLIP

Per poter configurare il networking bisogna eseguire alcune semplici operazioni. Posizionatevi nella cartella contenente i file immagine presenti nell'archivio images.zip, create una copia del file **full3** chiamata "my_elks" ed una cartella temporanea con:

```
$ cp full3 my_elks
$ mkdir tmp
```

A questo punto occorrerà lavorare con i privilegi di root:

```
$ su
Password: (Inserite la password di root)
```

Montate il file immagine **my_elks** nella cartella temporanea:

```
# mount -o loop my_elks tmp
```

Posizionatevi nella cartella contenente i file di configurazione della rete:

```
# cd tmp/etc/rc.d/
```

Aperte il file **rc.sysinit** con un editor di testo e modificate i parametri relativi alla rete (in particolare l'indirizzo IP, la porta e la velocità di trasferimento dei dati). Ecco un esempio di configurazione:

```
# Sezione dedicata alla configurazione della rete locale
localip=192.168.0.2
sliptty=/dev/ttyS0
ttybaud=9600

if test -f /bin/ktcp
then
    echo -n "Starting kernel network facilities: TCP/IP"
    stty $ttybaud < $sliptty
    ktcp $localip $sliptty &

    echo .
```

In questo caso tale macchina avrà l'indirizzo IP 192.168.0.2 ed instaurerà una connessione a 9600 bps sulla porta seriale /dev/ttyS0.

Nota: è caldamente consigliato usare solo la prima porta seriale con ELKS.

```
localip=192.168.0.1
sliptty=/dev/ttyS0
ttybaud=4800

if test -f /bin/ktcp
then
    echo -n "Starting kernel network facilities: TCP/IP"
    stty $ttybaud < $sliptty
    ktcp $localip $sliptty &

    echo .

    #echo -n "Starting network daemons: "
    #for daemon in httpd
    #do
    #    if test -f /bin/$daemon
    #    then
    #        echo -n " $daemon"
    #        $daemon &
    #    fi
    #done
    #echo .

fi
```

Figura 2: Configurazione dei parametri di rete di ELKS

Vi consiglio, inoltre, di commentare (inserendo un "#" all'inizio di ogni riga) tutta la sezione dedicata ai demoni di rete come **httpd**.

Salvate il file, uscite dall'editor e digitate:

```
# cd ../../..
# umount tmp
```

Quindi riversate l'immagine appena creata su un nuovo floppy con:

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

Avviate il sistema con il disco appena creato e fate il login come root.

Passiamo ora alla configurazione dell'altra macchina. Su un sistema Linux le operazioni da svolgere sono semplicissime. I comandi da digitare sono:

```
# slattach -p slip -L -s 9600 /dev/ttyS0 &  
# ifconfig sl0 192.168.0.3 pointopoint 192.168.0.2
```

In questo caso tale macchina avrà l'indirizzo IP 192.168.0.3 ed instaurerà una connessione a 9600 bps sulla porta seriale /dev/ttyS0.

Per verificare il funzionamento corretto della rete digitate dalla vostra macchina Linux:

```
# ping 192.168.0.2
```

Il tempo di risposta è variabile e dovrebbe attestarsi su quota 200 ms.

Networking – Lato Utente

Le applicazioni di rete disponibili sono ben poche. In particolare, abbiamo a disposizione i comandi telnet, netstat, urlget, ktcp, httpd.

In questo esempio analizzeremo la configurazione del server di rete httpd. Per prima cosa, create la cartella /usr/lib/httpd sulla vostra macchina ELKS e create un file HTML di prova con:

```
# cd /usr/lib  
# mkdir httpd  
# cd httpd  
# echo "Ciao" > index.htm
```

Per avviare il webserver digitate:

```
# httpd
```

A questo punto sarà possibile visualizzare il file index.htm attraverso la vostra Linux box aprendo con un browser la pagina:

<http://192.168.0.2/index.htm>

Sviluppo

Per i programmatori C è disponibile l'ambiente di sviluppo dev86, che si serve del compilatore bcc (Bruce's C Compiler). Esso non gira direttamente su piattaforma ELKS, ma necessita di un sistema Linux per lavorare.

Per prima cosa scaricate il pacchetto **dev86** dal sito <http://www.cix.co.uk/~mayday> in una cartella temporanea, quindi decomprimetelo con:

```
# tar xzvf Dev86src-0.16.14.tar.gz
```

Ed avviate la compilazione:

```
# cd dev86-0.16.14  
# make
```

Nella maggior parte dei casi la configurazione di default è più che sufficiente: digitate “**quit**” accompagnato da Invio per avviare la compilazione del programma con le opzioni di default.

Durante la compilazione del pacchetto vi si potrebbero presentare diversi errori: in ogni caso potete usare la versione precompilata del pacchetto dev86 (disponibile allo stesso indirizzo indicato in precedenza) che va decompressa nella directory "/" con il comando tar.

Per avviare l'installazione di dev86 digitate:

```
# make install
```

A questo punto l'ambiente di sviluppo per ELKS in C è pronto per l'uso.

Potete eventualmente compilare alcuni componenti aggiuntivi come il supporto per i binari ELKS in Linux. Entrate nella cartella elksemu e digitate:

```
# make module
```

Sulla mia distribuzione (una Slackware 9.0) la compilazione non è andata a buon fine. Vi auguro maggiore fortuna di quanta ne ho avuta io!

Esiste anche un altro programma da compilare chiamato **doselks** che permette di eseguire binari per ELKS sotto DOS. Poichè la compilazione di questo pacchetto mi ha comportato nuovamente dei problemi, vi consiglio di scaricare direttamente la versione precompilata dall'indirizzo <http://ftp.linux.org.uk/pub/linux/8086/emu/doselks-0.0.0.tgz>

Posizionatevi in una cartella temporanea e decomprimete il pacchetto con:

```
# tar xzvf doselks-0.0.0.tgz
```

All'interno della cartella chiamata "doselks" troverete un programma di nome **doselks.com**: esso dovrà essere copiato su un computer con MS-DOS e vi permetterà di eseguire binari ELKS digitando:

```
C:\> doselks NOME_BINARIO
```

A questo punto possiamo testare il nostro ambiente di sviluppo. Per realizzare un classico programma "Hello World" salvate in un file chiamato hello_world.c queste righe di codice:

```
#include <stdio.h>  
main()
```

```
{
    printf("Hello World !\n");
}
```

E compilatelo con:

```
# bcc -o hello_world hello_world
```

Verrà generato un eseguibile di nome **hello_world** che potrà essere utilizzato in un sistema ELKS.

Per onor di cronaca, esiste anche un ambiente di sviluppo alternativo per ELKS basato sul compilatore Turbo C. Per maggiori informazioni consultate il box relativo a questa soluzione.

Applicativi extra

Non esiste una collezione ufficiale di applicativi extra per ELKS.

Tuttavia, non molto tempo fa ho pubblicato su Internet una piccola serie di sorgenti (ad esempio programmi per Minix) tratti da diversi siti e pronti per essere ricompilati ed utilizzati con ELKS. Potete trovarli all'indirizzo <http://members.xoom.virgilio.it/pctips2/elks/apps/>

Ad esempio, il programma **3d.c.Z** permette di far apparire delle scritte sullo schermo con caratteri particolari in stile ASCII-Art.

Dopo averlo scaricato, è possibile compilarlo con:

```
# zcat 3d.c.Z > 3d.c
# bcc -o 3d 3d.c
```

Verrà generato un eseguibile di nome **3d** che potrà essere usato con ELKS.

Il programma **xxencode-2.c**, invece, è un encoder/decoder compatibile con i classici uuencode/uudecode, a cui aggiunge il supporto ad altri standard.

Dopo averlo scaricato, compilatelo con:

```
# bcc -o xxencode xxencode-2.c
```

A questo punto avrete a disposizione uno strumento molto potente che vi permetterà di codificare e decodificare i vostri dati.

Installazione del sistema

Come si è detto in precedenza, per chi volesse installare ELKS sul proprio PC è disponibile il pacchetto EDE (ELKS Distribution Edition), ovvero una mini-distribuzione pronta per l'uso.

Bisogna sottolineare che essa è priva del networking poichè ritenuto dall'autore, Neil Holmes, ancora poco user-friendly.

Dopo aver decompresso l'archivio contenente EDE digitate:

```
$ dd if=Image of=/dev/fd0
```

Se avete la necessità di ripartizionare il sistema (ad esempio avete una partizione con DOS), inserite un nuovo dischetto nel lettore e digitate:

```
$ dd if=FREEEDOS.IMG of=/dev/fd0
```

Nel caso in cui abbiate creato anche il secondo dischetto, avviate la vostra macchina con tale floppy inserito e ripartizionate il disco con **fdisk**.



Figura 3: Pre-partizionamento del disco con FreeDOS

Dopo aver completato il partizionamento, riavviate il sistema con il floppy su cui avete riversato il contenuto del file **Image**.

Nel corso dell'installazione dovrete creare una partizione per ELKS: allocate non oltre 20 MB in una di tipo 80.

Rendetela avviabile, salvate la tavola delle partizioni ed uscite da fdisk.



Figura 4: Partizionamento del disco fisso durante l'installazione di EDE

A questo punto inizierà l'installazione vera e propria che non richiederà molto tempo.

Al riavvio vi si presenterà un prompt simile al seguente:

```
boot>
```

Digitate “**elks**” seguito da Invio per avviare il sistema.

Per maggiori informazioni consultate la documentazione allegata.

Ricompilazione del kernel

Avendo a disposizione una distribuzione Linux installata è possibile ricompilare il kernel di ELKS in pochissimo tempo. Dovete effettuare il download dell'ultima versione del kernel dal sito ufficiale.

Dopo aver ottenuto il pacchetto **elks-0.1.1.tar.gz**, posizionatevi in una cartella temporanea e decomprimate l'archivio con:

```
# tar xzvf elks-0.1.1.tar.gz
```

Ed entrate nella cartella contenete i sorgenti digitando:

```
# cd elks-0.1.1
```

È possibile configurare il kernel in due modi: utilizzando l'interfaccia testuale oppure quella semi-grafica basata sulle librerie ncurses.

Nel primo caso digitate:

```
# make config
```

Nel secondo, invece:

```
# make menuconfig
```

Nella maggior parte dei casi le opzioni predefinite possono andar bene. Tuttavia, ci sono diverse funzionalità (generalmente sperimentali) non abilitate di default che possono essere molto utili: abbiamo il supporto per i file EXE di DOS, la compressione delle connessioni SLIP, la modalità protetta per i processori 80286, la possibilità di compilare il kernel in modo ROM-bootable (caratteristica molto utile per macchine dedite a scopi particolari), l'opzione per ridurre i consumi di spazio a scapito della velocità e così via.

Dopo aver salvato la configurazione attuale, avviate il calcolo delle dipendenze digitando:

```
# make dep
```

A questo punto sarà possibile generare due differenti tipologie di immagini del kernel. Nel primo caso verrà generata un'immagine adatta al boot da dischetto o comunque da una memoria fisica. Per generare un binario di tale tipo occorre digitare:

```
# make Image
```

Per riversare il floppy di boot appena creato su floppy digitate:

```
# dd if=arch/i86/boot/Image of=/dev/fd0
```

Affinché sia possibile utilizzare il kernel appena creato, dovrete procurarvi o costruirvi una immagine di root. Per iniziare potete utilizzare il file di nome **root** presente nell'archivio images.zip; per riversarlo su floppy entrate nella cartella che lo contiene e digitate:

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

Quando avvierete il sistema con il floppy contenente il kernel appena compilato vi verrà chiesto di inserire un disco di root nel lettore. Inserendo il floppy di root e premendo Invio il sistema teoricamente dovrebbe avviarsi.

La pratica insegna che si possono verificare diversi problemi: purtroppo, una soluzione definitiva non è stata ancora trovata, e bisognerà attendere qualche tempo prima che ne venga individuata una.

In particolare, si ha un crash del sistema quando si cerca di avviare **init**.

Nel secondo caso, invece, verrà generata un'immagine adatta al boot da rete. Per generare un binario di questo tipo occorre digitare:

make nbImage

L'immagine di ELKS per il boot da network (avente di default il nome **nbImage**) sarà disponibile all'interno della cartella arch/i86/boot/ e potrà essere utilizzata secondo le vostre necessità.

Conclusioni

In questo articolo ho cercato di fornire ai lettori una panoramica completa su questo sistema operativo. Molti di voi potranno utilizzarlo per riciclare macchine vecchie ed abbandonate, mentre altri potranno adoperarlo come strumento didattico per studiare il funzionamento di un sistema operativo.

I più abili, invece, potranno migliorare il sistema mettendo mano al codice sorgente del programma: come si è detto prima, ELKS è software libero.

Nei vari box che trovate in queste pagine sono raccolte diverse informazioni utili, tra cui suggerimenti su come ricompilare integralmente l'ambiente di lavoro oppure su come generare eseguibili DOS con il compilatore TCC (Turbo C Compiler) per poi convertirli in un formato compatibile con ELKS.

Per commenti, critiche e suggerimenti potete contattarmi all'indirizzo e-mail pctips@hardwaretips.com

Daniele Cono D'Elia - pctips

Installazione di DOSEMU

DOSEMU è un emulatore DOS che permette di utilizzare anche ELKS. Può essere utile per effettuare dei test con tale sistema.

Dopo aver decompresso l'archivio che lo contiene, entrate nella cartella creata e digitate:

```
# ./default-configure
```

```
# make
```

```
# make install
```

Per avviare ELKS da floppy attraverso l'interfaccia grafica digitate da un terminale (es. xterm):

```
# dosemu -X-A
```

Il sistema si avvierà da floppy in una nuova finestra. Vi ricordo che non è possibile installare facilmente ELKS sul disco rigido emulato.

Costruire un cavo seriale null-modem

I lettori aventi a disposizione un semplice saldatore possono costruirsi agevolmente un cavo seriale null-modem a partire da due connettori RS232 (a 9 o 25 poli) femmina ed un cavo (se schermato garantisce un segnale migliore) a 8-10 poli. Ecco gli schemi:

<i>DB-25</i>	<i>DB-9</i>	<i>X</i>	<i>DB-9</i>	<i>DB-25</i>
1	-	X	-	1
2	3	X	2	3
3	2	X	3	2
4	7	X	8	5
5	8	X	7	4
7	5	X	5	7
6	6	X	4	20
8	1	X	4	20
20	4	X	1	8
20	4	X	6	6

I connettori DB-25 sono tipici dei computer più vecchi ed hanno una superficie maggiore, mentre quelli DB-9 sono meno ingombranti e più diffusi, quindi è preferibile usare questi ultimi.

Ricompilare integralmente il sistema

Oltre al kernel, è possibile ricompilare integralmente tutto il sistema operativo ELKS. Ciò vi permetterà di poter ottimizzare le applicazioni per la vostra macchina e di poter creare delle immagini di root personalizzate.

Per fare ciò avete bisogno dei pacchetti **elkscmd** e **elks-net** (oltre ovviamente ai sorgenti del kernel ed all'ambiente di sviluppo).

Tale operazione è piuttosto lunga da spiegare: vi invito a leggere la FAQ di ELKS disponibile nel sito stesso del sistema operativo.

Generare eseguibili per ELKS con TCC

È possibile generare degli eseguibili per ELKS servendosi del compilatore per DOS Turbo C Compiler, realizzato dalla Borland diversi anni fa ed

attualmente scaricabile gratuitamente.

Gli eseguibili DOS, infatti, differiscono da quelli ELKS solo in alcuni punti e possono essere adattati per funzionare su esso.

A questo proposito un nostro connazionale di nome Alfonso Martone ha realizzato un programma chiamato TCC4ELKS.

Per maggiori informazioni visitate questo sito:

<http://www.retepnet.it/alfmar/fhlnr.html>

Sviluppi futuri

Sono previsti moltissimi miglioramenti per prossime versioni di ELKS. Dal lato kernel, nella 0.2.0 verranno principalmente ottimizzati la dimensione del codice ed il supporto per le ROM.

Nella versione 0.3.0, invece, saranno presenti driver completi per porte parallele e seriali più il supporto definitivo per connessioni PLIP e processori 286. Nella versione 1.0.0, infine, avremo il supporto alle schede di rete e molti altri miglioramenti.

Per il lato utente e per approfondimenti vi invito a leggere il file TODO dei sorgenti del kernel e dei pacchetti elksnet ed elkscmd.

Sistemi operativi a 16 bit

Oltre ad ELKS, oggetto di questo articolo, esistono molti sistemi operativi tuttora sviluppati che sono in grado di funzionare egregiamente su architetture a 16 bit.

In particolare occorre citare Minix e FreeDOS.

Minix è da considerarsi un antenato di Linux ed offre all'utente un sistema Unix semplice ma completo.

La sua home page è <http://www.cs.vu.nl/~ast/minix.html>

FreeDOS invece è un sistema libero rilasciato sotto la GNU GPL e compatibile con MS-DOS, a cui aggiunge moltissime funzionalità.

La sua home page è <http://www.freedos.org>

Licenza

Copyright (c) 2003 pctips <pctips@hardwaretips.com>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".