

Lucrarea 9

INSTALAREA UNEI INTERFEȚE ETHERNET PE UN SISTEM DE OPERARE LINUX

Sistemele de operare din familia **UNIX** permit construirea **nucleului** propriu sub formă de **module încărcabile**, specifice componentelor *hardware* utilizate. În funcție de **distribuție**, unele module pot fi **incluse** în pachet și se pot **încărca automat** la **instalare**.

Există situații în care **anumite componente hardware** **nu** sunt **recunoscute**, fiind rar utilizate sau mai noi decât **versiunea nucleului** sistemului de operare (*Linux kernel*) [15].

În aceste situații, codurile sursă ale modulelor, necesare instalării componentei respective, **trebuie** căutate și **descărcate** din paginile de web în care producătorul oferă resurse *software* și suport tehnic. Să presupunem că avem o interfață de rețea din familia **Intel Pro/1000 Ethernet Gigabit**. Pentru aceasta se apelează la <http://downloadcenter.intel.com/> unde se va căuta un „*driver linux*” pentru interfața respectivă și pentru versiunea nucleului sistemului de operare folosit. După ce se descarcă arhiva `e1000-x.x.x.tar.gz` trebuie parcurse **următoarele 5 etape**:

1. Despachetarea arhivei, utilizând comanda `tar` cu opțiunile necesare, așa cum se poate vedea în rândurile de mai jos:

```
root@masserv:~# tar xzf e1000-x.x.x.tar.gz
root@masserv:~#
```

Aceasta va **despacheta** arhiva în **directorul curent**, creând un subdirector cu **numele fișierului**, mai puțin extensia `.tar.gz`, în care se găsesc fișierele, în formă de **cod sursă**, scrise în limbajul de programare **C++**, împreună cu fișierele *script* necesare **compilării** acestora.

Înainte de parcurgerea următoarelor etape, se recomandă copierea acestui director în același loc unde se află celelalte fișiere sursă, ale nucleului distribuției inițiale, `/usr/src/linux-<Versiune Kernel>/drivers/net`.

Copierea se face utilizând comanda `cp` cu opțiunea `-R`, pentru copierea recursivă a mai multor directoare și a fișierelor conținute de acestea.

```
root@masserv:~# cp -R e1000-x.x.x /usr/src/linux-<Versiune Kernel>/drivers/net/
```

Apoi se va schimba directorul curent, cu cel care conține acele fișiere, utilizând comanda:

```
root@masserv:~# cd /usr/src/linux-<Versiune Kernel> /drivers/net/
root@masserv:~/usr/src/linux-<Versiune Kernel> /drivers/net/e1000-x.x.x/src#
```

2. Compilarea modulului se face cu ajutorul comenzii `make install` care interpretează conținutul fișierului `Makefile` și vor urma instrucțiunile conținute de acesta.

```
root@masserv:~.../e1000-x.x.x/src# make install
....
```

După **compilarea** codurilor sursă, fișierele binare obținute vor fi **copiate automat** în `/lib/modules/<Versiune Kernel> /kernel/drivers/ net/e1000/e1000.[k]o`.

3. Încărcarea modulului în nucleul propriu-zis se face doar după **instalarea** acestuia cu una dintre comenzile `insmod` sau `modprobe`. Pentru versiunile de nucleu `2.6.x`, comenziile `insmod` trebuie să i se specifice toată calea spre fișierul binar (compilat) al modulului. Spre deosebire de aceasta, comanda `modprobe` poate fi apelată având argument doar numele modulului, însoțit de parametrii de configurare.

Înainte de instalării modulului curent, se va **elimina** din **nucleu** o eventuală **variantă anterioară** utilizând comanda `rmmod`, urmată de numele modulului.

```
root@masserv:~# rmmod e1000
root@masserv:~# insmod /lib/modules/<Versiune Kernel>
/kernel/drivers/net/e1000/e1000.ko <parametri>=<valoare>
```

sau:

```
root@masserv:~# modprobe e1000 <parametri>=<valoare>
root@masserv:~#
```

În cazul **instalării fără erori**, se poate trece la faza următoare de **introducere** a comenzii `modprobe` în scriptul de **încărcare automată** la pornirea sistemului.

Acest lucru se face cu ajutorul unor fișiere, care depind de distribuția folosită, de exemplu `/etc/modules.conf` sau `/etc/rc.d/rc.modules`, în care trebuie să se afle o linie care se referă la acest modul.

Conținutul unui astfel de fișier poate să arate ca în exemplul de mai jos:

```
#!/bin/sh
# rc.modules 3.1 Thu Aug 28 14:01:35 PDT 2003 pjv
<volkerdi@slackware.com>
#
....
# Intel Pro/1000 Ethernet Gigabit Family support:
/sbin/modprobe e1000 <parametri>=<valoare>
#
```

Pentru **verificarea instalării corecte a modulului** se poate cere listarea conținutului fișierului `/proc/net/dev` unde trebuie să existe o linie care se referă la această interfață, utilizând comanda:

```
root@masserv:~# more /proc/net/dev
Inter-| Receive                               | Transmit
face |bytes packets errs drop fifo frame compressed multicast|bytes  packets errs drop fifo
colls carrier compressed
...
lo:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
eth0:875 289  0  0  0  0  0  0  27 22 0  0  0  0  0  0
eth1:162 265  0  0  0  1  0  0  28 27 0  0  0  0  0  0
....
root@masserv:~#
```

4. Configurarea interfeței de rețea se face după instalarea modulului, utilizând comanda : `ifconfig` urmată de **parametri specifici** tehnologiei **Ethernet** și **protocoalelor**, de nivel **OSI 2** și **3**, folosite.

```
root@masserv:~# ifconfig ethx Adresa_IP netmask Masca_De_Rețea broadcast
Adresa_de_Broadcast
```

unde `x` este numărul de identificare al interfeței, iar `Adresa_IP` `Masca_De_Rețea` și `Adresa_de_Broadcast`, – adresa, masca respectiv adresa de broadcast, corespunzătoare nivelului **OSI 3-rețea**.

Dacă se **optează** pentru transmisii de date cu rată de transfer de **1Gbps** și **cadre jumbo** (cu dimensiunea câmpului de date 9.000 octeți), în linia de mai sus **trebuie** introdusă **opțiunea** `mtu 9000` (*MTU – Maximum Transmit Unit*), sau adăugată o comandă suplimentară, astfel:

```
root@masserv:~# ifconfig ethx mtu 9000 up
root@masserv:~#
```

5. Verificarea configurației și a funcționării corecte a interfeței, în acest caz eth0, se poate face folosind aceeași comandă ifconfig, astfel:

```
root@masserv:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:A0:D2:A4:91:73
          inet addr:192.168.120.1  Bcast:192.168.120.63
Mask:255.255.255.192
          UP BROADCAST RUNNING MULTICAST  MTU:1500
Metric:1
          RX packets:29012900 errors:0 dropped:0
overruns:0 frame:0
          TX packets:22818720 errors:0 dropped:0
overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:945052525 (901.2 Mb)  TX
bytes:2823986908 (2693.1 Mb)
          Interrupt:11 Base address:0xbf00
```

Pentru **verificarea** funcționării componentei de nivel **OSI 3**, asociată acestei interfețe (protocolul de rețea **IP**), se poate folosi comanda **ping**, așa cum se poate vedea în rândurile următoare:

```
root@masserv:~# ping 192.168.120.1
PING 192.168.120.1 (192.168.120.1) 56(84) bytes of data
64 bytes from 192.168.120.1: icmp_seq=1 ttl=64 time=0.442 ms
64 bytes from 192.168.120.1: icmp_seq=2 ttl=64 time=0.326 ms
....
--- 192.168.120.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4995ms
rtt min/avg/max/mdev = 0.266/0.315/0.442/0.065 ms
root@masserv:~#
```

Notă: Răspunsul obținut, în urma interogării nucleului, depinde de configurația și parametri sistemului, exemplul de mai sus fiind doar orientativ.

Pentru utilizarea altor opțiuni de configurare a interfeței, se va consulta documentația care însoțește modulul și paginile de manual ale comenzilor utilizate. Dacă toate aceste etape au fost parcurse fără erori, interfața este configurată și aptă pentru exploatare. Pentru opțiunile de configurare specifice nivelelor **OSI** superioare, se vor consulta cursurile următoare.