Hálózatok építése és üzemeltetése

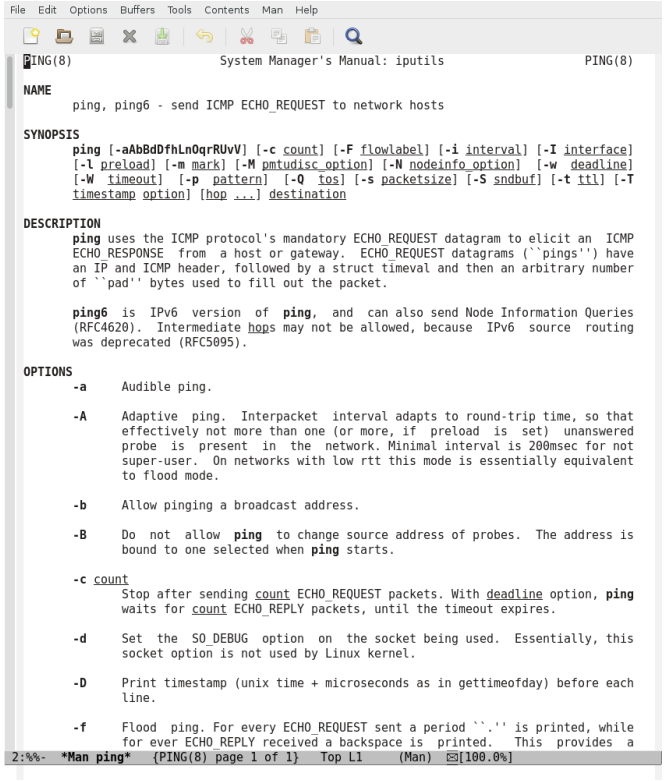
Szoftver szerszámok

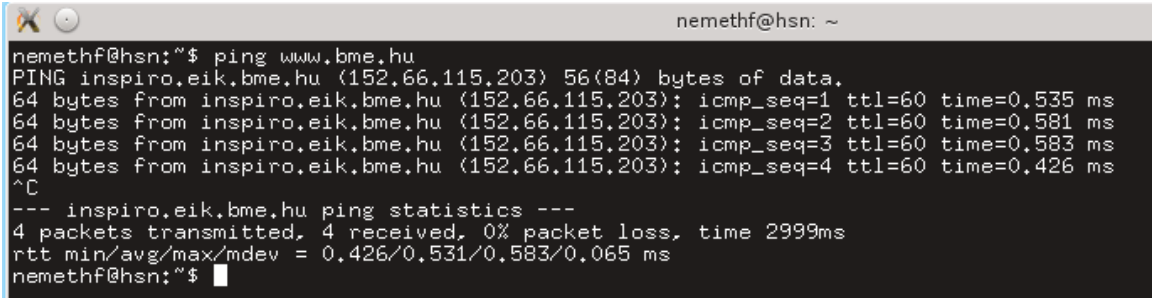
Hálózat megfigyelésére használható, egyszerű diagnosztikai eszközök

ping, traceroute, netstat,

tcpdump, wireshark

Ping





- ICMP echo request csomagokat küld a hálózatba
- körülfordulási idő mérésére, kacsolat tesztelésére használják
- Google helyett a program használatát a ‘man page’-ben érdemes megnézni

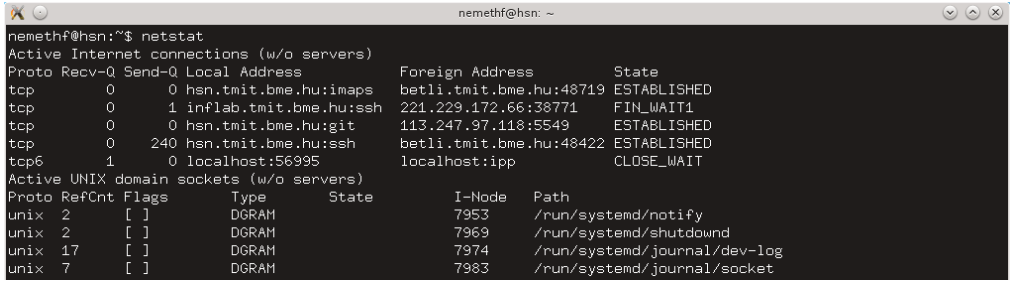
Traceroute – csomagútvonal felderítése

- Egyre nagyobb Time-To-Live értékkel küld IP csomagokat a célcímre.
- A routerek csomagtovábbításkor csökkentik a TTL értéket.
- Ha lejár a TTL, a router *általában* ICMP TIME_EXCEEDED csomagot küld a feladónak.

```
nemethf@hsn:~$ traceroute www.google.com
traceroute to www.google.com (216.58.214.36), 30 hops max, 60 byte packets
 1  244gw.tmit.bme.hu (152.66.244.254)  0.289 ms  0.308 ms  0.302 ms
 2  vl121.ixion.net.bme.hu (152.66.245.254)  0.992 ms  1.104 ms  1.272 ms
 3  xge4-2.styx.net.bme.hu (152.66.0.72)  0.879 ms  1.086 ms  1.254 ms
 4  xge2-2.taz.net.bme.hu (152.66.0.78)  0.943 ms  1.056 ms  1.058 ms
 5  tg0-1-0-1.rtr.bme.hbone.hu (152.66.0.126)  2.093 ms  2.091 ms  2.085 ms
 6  tg0-0-0-6.rtr1.vh.hbone.hu (195.111.100.43)  2.598 ms  2.097 ms  1.296 ms
 7  hungarnet-ias-geant-gw.bud.hu.geant.net (83.97.88.81)  1.010 ms  1.010 ms  1.002 ms
 8  google.mx1.fra.de.geant.net (62.40.125.201)  14.379 ms  14.039 ms  14.030 ms
 9  google-gw.mx1.fra.de.geant.net (62.40.125.202)  14.014 ms  14.013 ms  14.008 ms
10  209.85.243.17 (209.85.243.17)  14.942 ms  14.756 ms 209.85.244.5 (209.85.244.5)  16.202 ms
11  209.85.240.185 (209.85.240.185)  14.532 ms 209.85.240.187 (209.85.240.187)  15.774 ms  15.768 ms
12  fra15s09-in-f4.1e100.net (216.58.214.36)  15.729 ms  15.726 ms  15.721 ms
nemethf@hsn:~$
```

NETSTAT – hálózati kapcsolatok listázása

- Továbbá:
  - routing táblák,
  - interfész statisztikák,
  - multicastcsoport-tagság



Netstat –aputne

- kapcsolódó program beazonosítható PID alapján

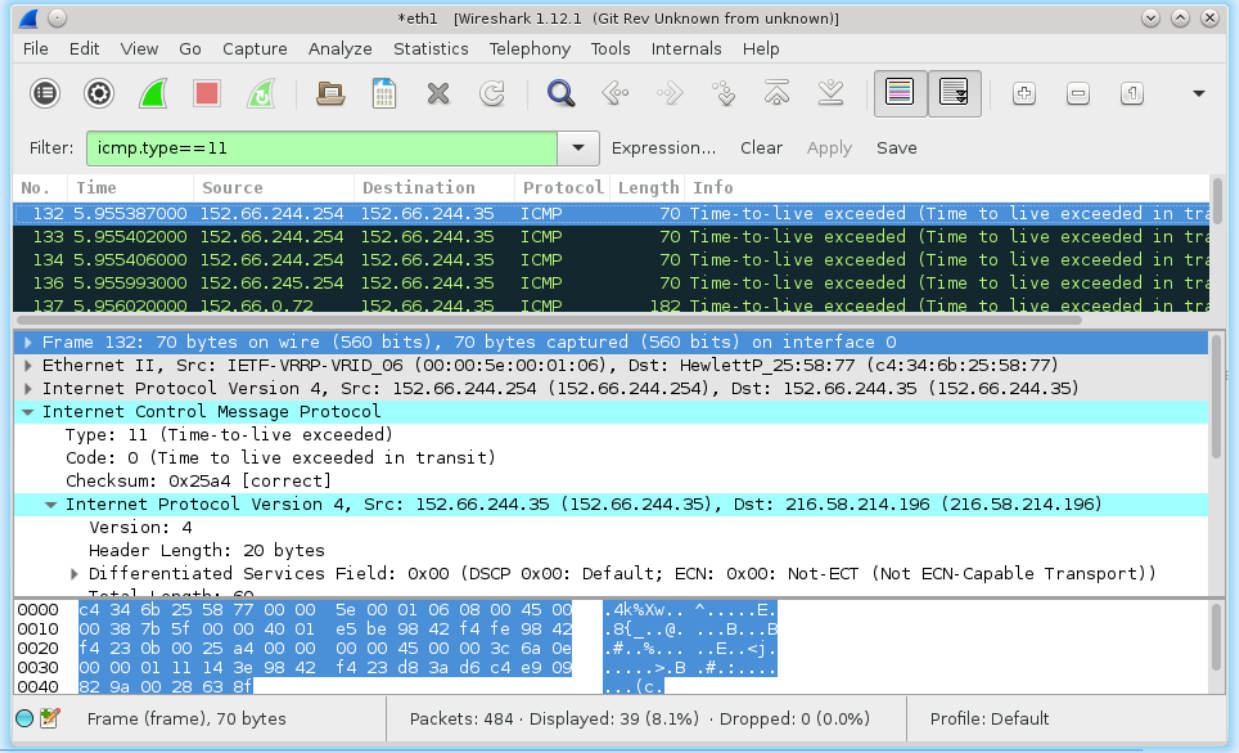
TCPDUMP – hálózati forgalom megfigyelése

- promiscuous mode: nem csak a gépnek küldött forgalom figyelése
- -n címfeloldás kihagyása
- -i *ifname*: a lehallgatott az intetfész megadása (-i eth0)
- -w *file*: az elfogott csomagok fájlba mentése *pcap* formátumban
- -s *snaplen:* csak a csomag *snaplen* bájtját olvassa be (def: 65535)

```
root@betli:~# tcpdump -c 5 -p icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
07:28:14.010286 IP betli.tmit.bme.hu > site.tmit.bme.hu: ICMP echo request, id 6066, seq 23, length 64
07:28:14.010522 IP site.tmit.bme.hu > betli.tmit.bme.hu: ICMP echo reply, id 6066, seq 23, length 64
07:28:15.010330 IP betli.tmit.bme.hu > site.tmit.bme.hu: ICMP echo request, id 6066, seq 24, length 64
07:28:15.010633 IP site.tmit.bme.hu > betli.tmit.bme.hu: ICMP echo reply, id 6066, seq 24, length 64
07:28:16.010325 IP betli.tmit.bme.hu > site.tmit.bme.hu: ICMP echo request, id 6066, seq 25, length 64
5 packets captured
6 packets received by filter
0 packets dropped by kernel
root@betli:~# tcpdump -c 5 -p icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
07:28:39.706429 IP 244gw.tmit.bme.hu > betli.tmit.bme.hu: ICMP time exceeded in-transit, length 36
07:28:39.706454 IP 244gw.tmit.bme.hu > betli.tmit.bme.hu: ICMP time exceeded in-transit, length 36
07:28:39.706470 IP 244gw.tmit.bme.hu > betli.tmit.bme.hu: ICMP time exceeded in-transit, length 36
07:28:39.706998 IP vl121.ixion.net.bme.hu > betli.tmit.bme.hu: ICMP time exceeded in-transit, length 36
07:28:39.707097 IP vl121.ixion.net.bme.hu > betli.tmit.bme.hu: ICMP time exceeded in-transit, length 36
5 packets captured
50 packets received by filter
0 packets dropped by kernel
root@betli:~#
```

Wireshark – grafikus tcpdump

- Tshark: (konzolban futtatható) szöveges wireshark
- Szintén pcap formátumot használ
- Moduláris felépítésű
  - *dissector* írja le egy protokoll működését
  - Python, Lua nyelven is írható hozzá dissector



Hálózat konfigurálása linux hoszton

Hálózat beállítási lehetőségei

- Disztribuciónként kicsit eltérő automatizmusok léteznek
  - De a “network manager”-t kell kikapcsolni, ha át akarjuk venni az irányítást
  - /etc/network könyvtár tartalmazza a konfig. fájlokat
- Félautomatikus megoldás: DHCP
  - Dynamic Host Configuration Protocol
  - # dhclient eth0
  - beállítható: IP address/netmask, default gateway, DNS, …
- Manuális beállítás
  - Ifconfig, ip (iproute2 csomag), route, iptables, …

ifconfig

- Könnyen áttekinthető a használata, mégis az 'ip'-t preferáljuk

```
root@betli:~# /etc/init.d/network-manager stop
Stopping network-manager (via systemctl): network-manager.service.
root@betli:~# ifconfig eth1 down
root@betli:~# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:96664 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96664 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:15753772 (15.0 MiB)  TX bytes:15753772 (15.0 MiB)
root@betli:~# ifconfig eth1 up
root@betli:~# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr c4:34:6b:25:58:77
          inet addr:152.66.244.35  Bcast:152.66.244.255  Mask:255.255.255.0
          inet6 addr: fe80::c634:6bff:fe25:5877/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16969186 errors:0 dropped:56 overruns:0 frame:0
          TX packets:14331562 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
          RX bytes:18852062080 (17.5 GiB)  TX bytes:12559976993 (11.6 GiB)
          Interrupt:20 Memory:d0700000-d0720000
```

```
root@betli:~# ifconfig eth1 152.66.244.35/24
root@betli:~# ifconfig
eth1      Link encap:Ethernet  HWaddr c4:34:6b:25:58:77
          inet addr:152.66.244.35  Bcast:152.66.244.255  Mask:255.255.255.0
          inet6 addr: fe80::c634:6bff:fe25:5877/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16969664 errors:0 dropped:56 overruns:0 frame:0
          TX packets:14331565 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18852095741 (17.5 GiB)  TX bytes:12559977231 (11.6 GiB)
          Interrupt:20 Memory:d0700000-d0720000
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:96696 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96696 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
          RX bytes:15756140 (15.0 MiB)  TX bytes:15756140 (15.0 MiB)
root@betli:~# ping -c 1 152.66.244.189
64 bytes from 152.66.244.189: icmp_seq=1 ttl=64 time=0.419 ms
root@betli:~# ping -c 1 hsn.tmit.bme.hu
ping: unknown host hsn.tmit.bme.hu
root@betli:~#  `
```

route

```
root@betli:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         152.66.159.254  0.0.0.0         UG    1024   0        0 wlan1
152.66.156.0    0.0.0.0         255.255.252.0   U     0      0        0 wlan1
152.66.244.0    0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth1
root@betli:~#
```

- `# route add default gw 152.66.159.254 eth1`
- `# ip route add 152.66.244.0/24 dev eth1`
- `# ip route add 152.66.244.0/24 via 152.66.159.254`

Iproute2 – ip:
egységes interface, moduláris felépítés

- `# ip link set dev eth1 up`
- `# ip addr ls`
- `# ip addr del 152.66.244.35/24 dev eth1`
- `# ip addr add 152.66.244.35/32 dev eth1`
- `# ip link set dev eth1 mtu 1412`
- `# ip link set dev eth0 address 22:ce:e0:99:63:6f`
- `# ip link set eth0 arp off`
- `# ip route add default dev eth1`
- `# ip route add 152.66.244.0/24 dev eth1 metric 100`
- `# ip route add 152.66.244.0/24 dev eth2 metric 200`
- `# ip route add 152.66.244.0/24 via 152.66.159.254`
- Manages: address, route, link, link group, tun/tap devices, ARP/NDP tables, tunnels, police routing, …

DNS: hostname → IP address (vagy fordítva)

- /etc/resolv.conf
- Felhasználói programok nem direktben használják a DNS protokollt
- Libnss:
  GNU Name Service Switch
  - Moduláris
  - Avahi module (zeroconf)
    lan multicast query, response

```
root@betli:~# nslookup www.bme.hu
Server:         152.66.115.1
Address:        152.66.115.1#53

www.bme.hu      canonical name = inspiro.eik.bme.hu.
Name:   inspiro.eik.bme.hu
Address: 152.66.115.203

root@betli:~# host www.bme.hu
www.bme.hu is an alias for inspiro.eik.bme.hu.
inspiro.eik.bme.hu has address 152.66.115.203
inspiro.eik.bme.hu has IPv6 address 2001:738:2001:2001::f0c1
root@betli:~# dig www.bme.hu

; <<>> DiG 9.9.5-9+deb8u6-Debian <<>> www.bme.hu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9710
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 6

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.bme.hu.                    IN      A

;; ANSWER SECTION:
www.bme.hu.             14400   IN      CNAME   inspiro.eik.bme.hu.
inspiro.eik.bme.hu.     14400   IN      A       152.66.115.203

;; AUTHORITY SECTION:
bme.hu.                 14400   IN      NS      ns2.pantel.net.
bme.hu.                 14400   IN      NS      ns.bme.hu.
bme.hu.                 14400   IN      NS      nic.bme.hu.

;; ADDITIONAL SECTION:
ns.bme.hu.              14400   IN      A       152.66.116.1
nic.bme.hu.             14400   IN      A       152.66.115.1
ns2.pantel.net.         20516   IN      A       212.24.160.1
ns.bme.hu.              14400   IN      AAAA    2001:738:2001:8001::2
nic.bme.hu.             14400   IN      AAAA    2001:738:2001:2001::2

;; Query time: 1 msec
;; SERVER: 152.66.115.1#53(152.66.115.1)
;; WHEN: Mon Sep 19 10:45:59 CEST 2016
;; MSG SIZE  rcvd: 248
```

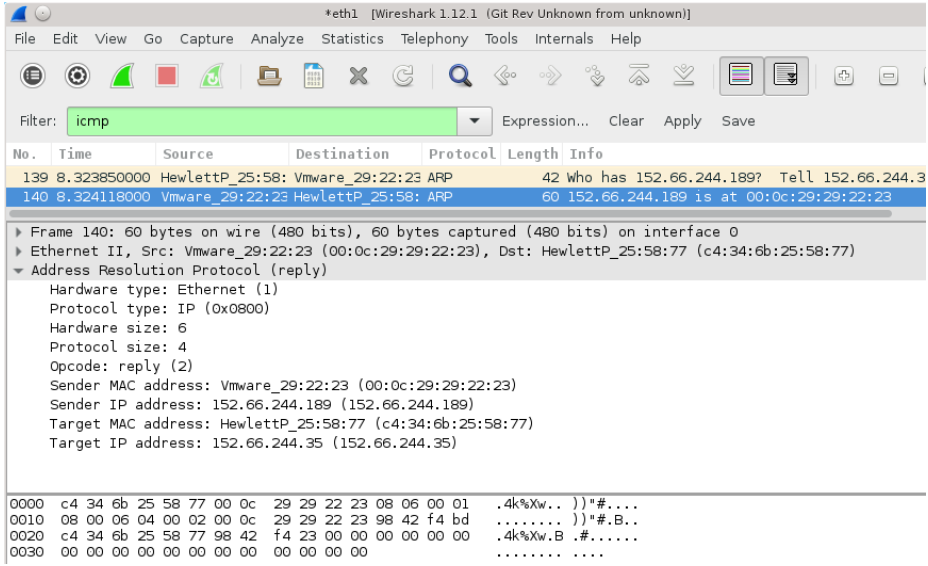
ARP: IP address → MAC address

- ARP requestre, ARP response a válasz,
- A kernel automatikusan karban tartja az ARP-táblát

- Egyéb eszközök:
  - arpwatch/arpmonitor, arpoison, arpspoof, arping, arp-sk

```
root@betli:~# arp -i eth1 -s 10.0.0.10 aa:bb:cc:dd:ee:ff
root@betli:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
152.66.159.160           ether   a4:5e:60:c1:12:75   C                     wlan1
152.66.156.39            ether   b8:e8:56:0f:03:6c   C                     wlan1
152.66.244.174           ether   00:0c:29:29:22:23   C                     eth1
152.66.158.251           ether   0c:8b:fd:63:be:33   C                     wlan1
152.66.156.101                   (incomplete)                              wlan1
152.66.159.113           ether   00:24:d7:2a:81:10   C                     wlan1
10.0.0.10                ether   aa:bb:cc:dd:ee:ff   CM                    eth1
152.66.158.252           ether   28:cf:e9:4d:8e:83   C                     wlan1
152.66.159.231                   (incomplete)                              wlan1
root@betli:~#
```

```
root@betli:~# arping -c 3 -I eth1 152.66.244.189
ARPING 152.66.244.189 from 152.66.244.35 eth1
Unicast reply from 152.66.244.189 [00:0C:29:29:22:23]  0.738ms
Unicast reply from 152.66.244.189 [00:0C:29:29:22:23]  0.854ms
Unicast reply from 152.66.244.189 [00:0C:29:29:22:23]  0.720ms
Sent 3 probes (1 broadcast(s))
Received 3 response(s)
root@betli:~#
```



iptables: tűzfal és NAT beállítása

- (későbbi órán tárgyaljuk)

MPLS

- Enable mpls support
  - `sysctl -w net.mpls.conf.eth0.input=1`
  - `sysctl -w net.mpls.conf.eth1.input=1`
  - `sysctl -w net.mpls.platform_labels=1000`
- Routing 10.10.10.10/32 to 192.168.1.2 with label 100:
  - `ip route add 10.10.10.10/32 encap mpls 100 via inet 192.168.1.2`
- Label swapping 100 for 200 and sent to 192.168.2.2:
  - `ip -f mpls route add 100 as 200 via inet 192.168.2.2`
- Decapsulating label 300 and delivering locally:
  - `ip -f mpls route add 300 dev lo`
- To show MPLS routes you can do:
  - `ip -f mpls route show`

Több IP cím egy interfészen

- `# ifconfig eth0:0 192.168.1.6 up`

Egyéb hasznos eszközök

netcat

Xterm-1

- `$ nc -l 2389`
- `HI, server`

Xterm-2

- `$ nc localhost 2389`
- `HI, server`

telnet

- Távoli terminál elérése **titkosítatlan** adatforgalommal
- Használjuk helyette az SSH-t
- Debuggolásra, fejlesztérse azért nagyszerű:
- Portkiosztás /etc/services

```
root@betli:~# telnet www.bme.hu 80
Trying 2001:738:2001:2001::f0c1...
Connected to inspiro.eik.bme.hu.
Escape character is '^]'.
GET / HTTP/1.0
host: www.bme.hu

HTTP/1.1 200 OK
Date: Mon, 19 Sep 2016 09:20:40 GMT
Server: Apache
X-Powered-By: PHP/5.4.45-0+deb7u5
X-Drupal-Cache: HIT
Etag: "1474276784-0"
Content-Language: hu
X-Frame-Options: SAMEORIGIN
X-UA-Compatible: IE=edge
X-Generator: Drupal 7 (http://drupal.org)
```

Bash szkriptek

A főcél: AUTOMATIZÁLÁS!

- Unix filozófia:
  - “write programs that do one thing, but do it well”
  - Egyszerű eszközök összefűzhetők:
  - `grep nemethf /etc/passwd | cut -f 5 -d:`
- Komplex parancssori megoldások újrahasznosíthatók:
  - “`. filename`”: parancsértelmező a fájl tartalmát végrehajtja mint utasítások sorozatát
  - “`filename`”: a fájlt végrehajtja az első néhány bájltja alapján
    - ld: “`file filename`”
- Shebang: szövegfájl elején megadja a parancsértelmezőt
  - Pl: `#!/bin/bash`

Példa: /etc/init.d/motd

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          motd
# Required-Start:    hostname $local_fs
# Required-Stop:
# Should-Start:
# Default-Start:     1 2 3 4 5
# Default-Stop:
# Short-Description: Create dynamic part of /etc/motd
# Description:       /etc/motd is user-editable and static.  This script
#                    creates the initial dynamic part, by default the
#                    output of uname, and stores it in /var/run/motd.dynamic.
#                    Both parts are output by pam_motd.
### END INIT INFO

PATH=/sbin:/usr/sbin:/bin:/usr/bin
. /lib/init/vars.sh

do_start () {
        # Update motd
        uname -snrvm > /var/run/motd.dynamic
}

do_status () {
        if [ -f /var/run/motd.dynamic ] ; then
                return 0
        else
                return 4
        fi
}
```

```
case "$1" in
  start|"")
        do_start
        ;;
  restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
        exit 3
        ;;
  stop)
        # No-op
        ;;
  status)
        do_status
        exit $?
        ;;
  *)
        echo "Usage: motd [start|stop|status]" >&2
        exit 3
        ;;
esac

:
```

Python

Python – általános célú programozási nyelv

- Beágyazott környezetben előszerettel használják
  - Idő, erőforráskritikus rész C-ben,
  - Konfigurációs, nem kritikus rész pythonban (luában, lispben)
- Szabad szoftver, közösségi fejlesztéssel
  - Benevolent Dictator For Live: Guido van Rossum
- python2, python3: enyhe inkompatibilitás
- Utasításblokkokat behúzás jelzi (nem zárójelek)

```
#!/usr/bin/env python

def fib(n):
    "Return the Fibonacci number of n"
    a, b = 0, 1
    while a < n:
        a, b = b, a+b
    return b

print fib(2)
```

Learn python in 5 minutes

https://www.stavros.io/tutorials/python/

```
>>> help(5)
Help on int object:
(etc etc)

>>> dir(5)
['__abs__', '__add__', ...]

>>> abs.__doc__
'abs(number) -> number

Return the absolute value of the argument.'
```

```
>>> myvar = 3
>>> myvar += 2
>>> myvar
5
>>> myvar -= 1
>>> myvar
4
```

```
>>> mystring = "Hello"
>>> mystring += " world."
>>> print mystring
Hello world.
# This swaps the variables in one line(!).
# It doesn't violate strong typing because values aren't
# actually being assigned, but new objects are bound to
# the old names.
>>> myvar, mystring = mystring, myvar
```

```
>>> sample = [1, ["another", "list"], ("a", "tuple")]
>>> mylist = ["List item 1", 2, 3.14]
>>> mylist[0] = "List item 1 again" # We're changing the item.
>>> mylist[-1] = 3.21 # Here, we refer to the last item.
>>> mydict = {"Key 1": "Value 1", 2: 3, "pi": 3.14}
>>> mydict["pi"] = 3.15 # This is how you change dictionary values.
>>> mytuple = (1, 2, 3)
>>> myfunction = len
>>> print myfunction(mylist)
3
```

```
>>> mylist = ["List item 1", 2, 3.14]
>>> print mylist[:]
['List item 1', 2, 3.1400000000000001]
>>> print mylist[0:2]
['List item 1', 2]
>>> print mylist[-3:-1]
['List item 1', 2]
>>> print mylist[1:]
[2, 3.14]
# Adding a third parameter, "step" will have Python step in
# N item increments, rather than 1.
# E.g., this will return the first item, then go to the third and
# return that (so, items 0 and 2 in 0-indexing).
>>> print mylist[::2]
['List item 1', 3.14]
```

Learn python in 5 minutes (min 3-4)

```
rangelist = range(10)
>>> print rangelist
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
for number in rangelist:
    # Check if number is one of
    # the numbers in the tuple.
    if number in (3, 4, 7, 9):
        # "Break" terminates a for without
        # executing the "else" clause.
        break
    else:
        # "Continue" starts the next iteration
        # of the loop. It's rather useless here,
        # as it's the last statement of the loop.
        continue
else:
    # The "else" clause is optional and is
    # executed only if the loop didn't "break".
    pass # Do nothing

if rangelist[1] == 2:
    print "The second item (lists are 0-based) is 2"
elif rangelist[1] == 3:
    print "The second item (lists are 0-based) is 3"
else:
    print "Dunno"

while rangelist[1] == 1:
    pass
```

```
# Same as def funcvar(x): return x + 1
funcvar = lambda x: x + 1
>>> print funcvar(1)
2

# an_int and a_string are optional, they have default values
# if one is not passed (2 and "A default string", respectively).
def passing_example(a_list, an_int=2, a_string="A default string"):
    a_list.append("A new item")
    an_int = 4
    return a_list, an_int, a_string

>>> my_list = [1, 2, 3]
>>> my_int = 10
>>> print passing_example(my_list, my_int)
([1, 2, 3, 'A new item'], 4, "A default string")
>>> my_list
[1, 2, 3, 'A new item']
>>> my_int
10
```

- Ami kimaradt:
  - Strings, classes, modules, file i/o, exceptions

Scapy – python-based packet manipulator

```
>>> send(IP(dst="1.2.3.4")/ICMP())
.
Sent 1 packets.
>>> sendp(Ether()/IP(dst="1.2.3.4",ttl=(1,4)), iface="eth1")
....
Sent 4 packets.
>>> sendp("I'm travelling on Ethernet", iface="eth1", loop=1, inter=0.2)
................^C
Sent 16 packets.
>>> sendp(rdpcap("/tmp/pcapfile")) # tcpreplay
..........
Sent 11 packets.
```

```
>>> p=sr1(IP(dst="www.slashdot.org")/ICMP()/"XXXXXXXXXXX")
Begin emission:
...Finished to send 1 packets.
.*
Received 5 packets, got 1 answers, remaining 0 packets
>>> p
<IP version=4L ihl=5L tos=0x0 len=39 id=15489 flags= frag=0L ttl=42 proto=ICMP
 chksum=0x51dd src=66.35.250.151 dst=192.168.5.21 options='' |<ICMP type=echo-reply
 code=0 chksum=0xee45 id=0x0 seq=0x0 |<Raw load='XXXXXXXXXXX'
 |<Padding load='\x00\x00\x00\x00' |>>>>
>>> p.show()
---[ IP ]---
version   = 4L
ihl       = 5L
tos       = 0x0
len       = 39
id        = 15489
flags     =
frag      = 0L
ttl       = 42
proto     = ICMP
chksum    = 0x51dd
src       = 66.35.250.151
dst       = 192.168.5.21
options   = ''
---[ ICMP ]---
   type      = echo-reply
   code      = 0
   chksum    = 0xee45
   id        = 0x0
   seq       = 0x0
---[ Raw ]---
      load      = 'XXXXXXXXXXX'
---[ Padding ]---
         load      = '\x00\x00\x00\x00'
```

- UTScapy: unit testing with scapy