



Johanyák Zsolt Csaba  
Kovács Péter  
Göcs László

# Linux hálózati adminisztráció a gyakorlatban

1.3

*Készült az FSF.HU Alapítvány támogatásával*

2013

**Johanyák Zsolt Csaba, Kovács Péter,  
Göcs László**

**Linux hálózati adminisztráció a  
gyakorlatban**

2013

© 2012-2013, Johanyák Zsolt Csaba, Kovács Péter, Göcs László

### 1.3. Kiadás

Szakmai lektor: Bodor Zoltán

Nyelvi lektor: Bodorné Takács Éva

A könyv az FSF.hu (Alapítvány a szabad szoftverek magyarországi népszerűsítéséért és honosításáért) támogatásával készült.

A szerzők a könyv írása során törekedtek arra, hogy a leírt tartalom a lehető legpontosabb és naprakész legyen. Ennek ellenére előfordulhatnak hibák, vagy bizonyos információk elavulttá válhattak.

A könyvben leírt példákat mindenki saját felelősségére alkalmazhatja. Javasoljuk, hogy ezeket ne éles környezetben próbálják ki. A felhasználásból fakadó esetleges károkért sem a szerzők, sem az FSF.hu Alapítvány nem vonható felelősségre.

Az oldalakon előforduló márka- valamint kereskedelmi védjegyek bejegyzőjük tulajdonában állnak.

A könyv elektronikus változata elérhető a <http://gamfinfo.hu/> oldalról.

## Tartalomjegyzék

Tartalomjegyzék.....	3
Bevezetés.....	5
1. Virtualizáció és telepítés virtuális gépre (Kovács Péter) .....	7
1.1. Virtualizáció, VirtualBox .....	7
1.2. Ubuntu 12.04 LTS desktop és server telepítése VirtualBox -ra.....	8
1.3. A létrehozott és előkonfigurált „üres” virtuális gépre immáron telepíthető a rendszer	11
1.4. A szerver változat telepítése.....	16
1.5. Belső hálózat kialakítása virtuális gépek között .....	22
1.6. Vonatkozó irodalomjegyzék .....	23
2. Kezelési alapismeretek (Kovács Péter).....	24
2.1. Néhány fontosabb parancssori utasítás használata az asztali (Desktop) Ubuntu változat	24
esetén.....	24
Az elérési utak.....	24
Munka állományokkal és könyvtárakkal.....	25
Az ls parancs .....	26
Állományok és könyvtárnevek megadása maszk alkalmazásával .....	26
Átírányítás és csövezeték .....	27
A chmod parancs .....	28
Szűrők használata.....	29
2.2. A vi használata .....	30
2.3. A nano szerkesztő használata.....	32
2.3. A Midnight Commander használata.....	34
2.2.1. FTP kapcsolat létrehozása távoli számítógéppel.....	34
2.2.2. Távoli gép könyvtárának csatolása .....	35
2.2.3. Állományok és könyvtárak védelmi kódsorának beállítása .....	36
2.2.4. Szimbolikus keresztkapcsolat létrehozása .....	36
2.2.5. Tulajdonos és csoport.....	37
2.4 Vonatkozó irodalomjegyzék .....	37
3. Hálózati beállítások lekérdezése és módosítása (Johanyák Zsolt Csaba).....	39
3.1. Előkészítés.....	39
3.2. Beállítás karakteres felületen (szerver) .....	41
3.2.1. Beállítás konfigurációs állományok nélkül .....	42
3.2.2. Beállítás konfigurációs állományokkal (tartós beállítás) .....	42
3.3. Beállítás grafikus felületen (asztali gép) .....	44
3.4. Gépnév beállítása .....	49
3.5. Vonatkozó irodalomjegyzék .....	50
4. DNS szerver telepítése és beállítása (Johanyák Zsolt Csaba).....	51
4.1. Előkészítés.....	51
4.2. Telepítés és konfigurálás .....	52
4.3. Tesztelés .....	56
4.4. Vonatkozó irodalomjegyzék .....	57
5. DHCP szerver telepítése és konfigurálása (Johanyák Zsolt Csaba) .....	59
5.1. Előkészítés.....	59
5.2. Telepítés és konfigurálás .....	60
5.3. Vonatkozó irodalomjegyzék .....	62
6. Megosztás NFS segítségével (Johanyák Zsolt Csaba).....	63
6.1. Előkészítés.....	63

---

6.2. NFS szerver telepítése és beállítása .....	64
6.3. NFS kliens telepítése és beállítása .....	65
6.4. Tesztelés .....	66
6.5. Vonatkozó irodalomjegyzék .....	66
7. Megosztás Samba segítségével ( <i>Johanyák Zsolt Csaba</i> ) .....	69
7.1. Előkészítés.....	69
7.2. Samba kiszolgáló telepítése .....	70
7.3. Samba kliens telepítése és konfigurálása .....	72
7.4. Tesztelés .....	73
7.5. Vonatkozó irodalomjegyzék .....	73
8. WebDAV kiszolgáló konfigurálása ( <i>Kovács Péter</i> ) .....	75
8.1. Előkészítés.....	75
8.2. Telepítés és konfigurálás .....	76
8.3. Tesztelés .....	79
8.4. Vonatkozó irodalomjegyzék .....	80
9. SQUID proxy szerver konfigurálása ( <i>Kovács Péter</i> ).....	81
9.1. Előkészítés.....	81
9.1.1. Telepítés Ubuntu serverre .....	83
9.2. Telepítés és konfigurálás .....	84
9.3. Vonatkozó irodalomjegyzék .....	85
10. A hálózati címfordítás (NAT) megvalósítása ( <i>Göcs László</i> ) .....	87
10.1. A címfordítás típusai .....	88
10.1.1. Egy az egyben címfordítás (Full cone NAT) .....	88
10.1.2. Címhez kötött címfordítás (Address Restricted cone NAT) .....	88
10.1.3. Porthoz és címhez kötött címfordítás (Port-Restricted cone NAT) .....	89
10.1.4. Szimmetrikus címfordítás (Symmetric NAT).....	89
10.2. Előkészítés.....	90
10.2. Tűzfal beállítása .....	91
10.3. A NAT telepítése és konfigurálása.....	93
10.4. Tesztelés .....	94
10.5. Vonatkozó irodalomjegyzék .....	95
11. LDAP kiszolgáló telepítése ( <i>Göcs László</i> ).....	97
11.1. Mi az LDAP? .....	97
11.2. Előkészítés.....	98
11.3. Telepítés és konfigurálás .....	100
11.3.1. Az LDAP alapú hitelesítés konfigurálása .....	104
11.3.2. Felhasználók és csoportok létrehozása.....	108
11.4. Vonatkozó irodalomjegyzék .....	110
12. A ClamAV víruskereső telepítése ( <i>Johanyák Zsolt Csaba</i> ).....	111
12.1. Telepítés és beállítás.....	111
12.2. Vonatkozó irodalomjegyzék .....	113

## Bevezetés

A szabad szoftverek térhódításával és az asztali Linux operációs rendszerek egyre magasabb szintű szolgáltatásainak eredményeképpen folyamatosan nő az érdeklődés a Linux alapú megoldások iránt. A Kecskeméti Főiskola GAMF Karának<sup>1</sup> illetve jogelőd intézményeinek oktatási palettáján már lassan húsz éve megjelentek úgy a felhasználói szintű, mint a rendszermenedzsment szintű Unix (a kezdeti években) illetve Linux (jelenleg is) ismeretek. Az elmúlt években számos disztribúciót kipróbáltunk (Unix vonalon: Sun Solaris<sup>2</sup>, Silicon Graphics Irix<sup>3</sup>, Linux vonalon: RedHat<sup>4</sup>, SuSE<sup>5</sup>, Debian<sup>6</sup>, UHU<sup>7</sup>), míg végül az Ubuntu<sup>8</sup> mellett döntöttünk.

A hardver adottságokból következően kezdetben dual-boot<sup>9</sup>-os megoldásokat alkalmaztunk a Linux oktatásra használt géptermekekben, és komoly kihívásokkal szembesültünk, amikor megkíséreltük összeegyeztetni a hálózatmenedzsment oktatás igényeit az intézményi informatikai biztonsági előírásokkal. Időnként az egyetlen megoldás a labor hálózatának teljes elszigetelése volt.

Az infrastruktúra fejlődése a virtualizáció megjelenését és előretörését eredményezte, és ennek köszönhetően ma már minden hallgató egy saját kis virtuális hálózatot hozhat létre a gépén akár négy virtuális gép jó minőségű egyidejű futtatásával. Virtualizációs szoftverként a VirtualBox<sup>10</sup>-ot használjuk az oktatási célból ingyenes hozzáférés, egyszerű kezelhetőség és a platformfüggetlenség miatt. Ez utóbbi különösen fontos volt, mivel az utóbbi években kétszer is gazda operációs rendszert kellett váltanunk.

Könyvünk elsődleges célja az volt, hogy könnyen feldolgozható és azonnal hasznosítható segédanyagot nyújtson a mérnökinformatikus képzés Linux hálózati adminisztráció című tantárgya gyakorlatainak oktatásához. A könyv első tizenegy fejezete lefedi a tárgy gyakorlatait, minden fejezet egy-egy 2x45 perces gyakorlat anyagát tartalmazza lépésről-lépésre, ismertetve a gyakorlat célját, a szükséges előkészítést, az elvégzendő műveleteket és kiadandó utasításokat képernyőképekkel kiegészítve, ahol az szükséges. A további fejezetekben olyan kapcsolódó konfigurációs témakörök szerepelnek, amelyeknek ismeretét és begyakorlását hasznosnak és szükségesnek tekintettük, de az óraszám korlátok miatt már nem kerülhettek be a géptermi gyakorlatra.

A könyvet bátran ajánljuk tágabb körben is a téma iránt érdeklődő olvasók számára. Az összes gyakorlat (fejezet) önállóan, tanári segítség nélkül is elvégezhető, és a benne foglalt tananyag elsajátítható. Mivel a feladatok végrehajtásánál egyszerre legfeljebb két futó virtuális gép szükséges, ezért egy optimálisnak tekinthető hardverigény (3 GB memória, 2 magos processzor, 20-30 GB szabad merevlemez terület) teljesítése mellett elfogadható sebességgel otthon is minden kipróbálható. A könyv megírásakor feltételeztük, hogy az

---

<sup>1</sup> <http://www.gamf.hu/>

<sup>2</sup> [http://en.wikipedia.org/wiki/Solaris\\_%28operating\\_system%29](http://en.wikipedia.org/wiki/Solaris_%28operating_system%29)

<sup>3</sup> <http://www.sgi.com/products/software/irix/>

<sup>4</sup> <http://www.redhat.com/>

<sup>5</sup> <https://www.suse.com/>

<sup>6</sup> <http://www.debian.org/>

<sup>7</sup> <http://uhulinux.hu/>

<sup>8</sup> <http://www.ubuntu.com/>

<sup>9</sup> [http://en.wikipedia.org/wiki/Multi\\_boot](http://en.wikipedia.org/wiki/Multi_boot)

<sup>10</sup> <https://www.virtualbox.org/>

olvasó rendelkezik számítógépes hálózati alapismeretekkel így ezekre a témakörökre nem tértünk ki részletesen. A gyakorlatok kidolgozása során az Ubuntu Linux 12.04 LTS Server és Desktop változatait használtuk.

Végül szeretnénk köszönetet mondani az FSF.hu alapítványnak a könyv megírásához nyújtott anyagi támogatásért.

Kecskemét, 2012. augusztus 31.

A szerzők

## 1. Virtualizáció és telepítés virtuális gépre (Kovács Péter)

Könyvünk első fejezetében röviden elmagyarázunk néhány alapfogalmat, megismerkedünk a VirtualBox-ban történő virtuális gép létrehozás lépéseivel, majd áttekintjük a továbbiakban használni kívánt asztali (Desktop) és kiszolgáló (Server) Ubuntu 12.04 LTS változatok telepítésének lépéseit.

### 1.1. Virtualizáció, VirtualBox

A virtualizáció egy igen hasznos eszköz számítógépünk fizikai erőforrásainak megosztására, és szimulációs környezet kialakítására, egy fizikai gépen több virtuális rendszer létrehozására. A virtualizációban használt két fontos fogalom a gazda (host) és a vendég (guest). A gazda gép vagy rendszer jelenti azt a fizikai gépet, mely a virtuális környezetet futtatja, hallgatói laborok esetén ez gyakorlatilag az asztalon megtalálható PC. A vendég pedig az a rendszer, amely a virtuális környezeten belül fut.

A vendég rendszert tekinthetjük egy elképzelt PC-nek, amelyre tetszőleges operációs rendszert telepíthetünk és konfigurálhatunk, ezen rendszer a virtualizációs környezeten (pl.: VirtualBox) keresztül tart kapcsolatot a fizikai gépünkkel (gazda) és általa a külvilággal. A virtuális környezetben testre szabható, hogy a fizikai gép mely erőforrásait, és milyen mértékben szeretnénk láttatni a vendég rendszerben. Virtuális keretrendszerben van lehetőségünk konfigurálni a tárolókat és a hálózati csatolókat is, amelyre minden bizonnyal szükségünk lesz.

Az Oracle VirtualBox<sup>11</sup> egy ingyenesen használható virtualizációs platform, mely telepíthető Windows, Linux, OS X és Solaris rendszerekre is. A program támogatja az IDE és SATA csatolók emulációját, USB portok és eszközök megosztását, valamint hat darab hálózati csatoló módot. A hálózati csatolók kiemelt fontossága miatt, az alábbiakban ismertetjük ezeket a módokat.

VirtualBox hálózati emulációs módok:

- *Nincs csatlakoztatva*: ez a mód sok magyarázatot nem igényel, ezen mód mellett nincs hálózati kapcsolat a vendég rendszeren.
- *NAT*: hálózati címfordítási mód, ahol a VirtualBox a fizikai gépünk elsődleges hálózati csatolója és a vendég rendszer adott csatolója között címfordítást végez. Ez az alapértelmezett mód, hiszen az esetek nagy részben szükséges Internet kapcsolat a vendég rendszerre, amit a fizikai gépünk Internet kapcsolatának megosztásával érünk el.
- *Bridge-elt kártya*: ezen módban a vendég operációs rendszer és a fizikai gépünk egy kijelölt hálózati kártyája között képezhetünk kapcsolatot, mely megkerüli a gazda

---

<sup>11</sup> <https://www.virtualbox.org>



operációs rendszer hálózati vermet is. Gyakorlatilag olyan hatást érünk el, mintha a vendég gépbe építettük volna be a fizikai csatolót.

- *Belső csatoló*: virtuális (belső) vendég-vendég hálózatok létesítésére használható. Ezen módon lehetőségünk van több vendég rendszer összekapcsolására is. A virtuális hálózatok összekapcsolásának alapja a hálózat neve, azaz két vendég akkor van azonos virtuális hálózatban, ha van olyan belső csatolós adapterük, melynek hálózati neve azonos.
- *Host-only kártya*: logikai hálózatot képezhetünk vele a gazda és a vendég (vagy akár több vendég) között, anélkül, hogy a gazda hálózati kártyájára szükség lenne.
- *Általános driver*: ritkán alkalmazott mód, azonos hálózati csatolók használatát teszi lehetővé különböző vendég rendszereknek, saját meghajtó programjaik (drivereik) segítségével.

## 1.2. Ubuntu 12.04 LTS desktop és server telepítése VirtualBox -ra

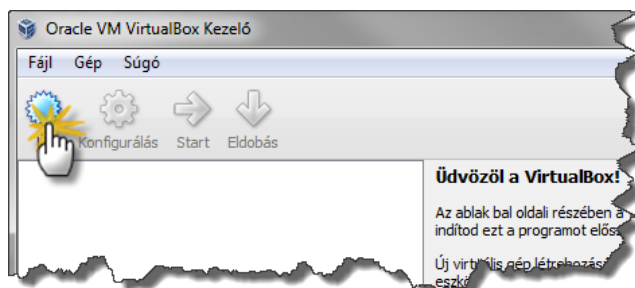
Első lépésként a fizikai gépre telepítjük az aktuális, rendszer specifikus VirtualBox kiadást és a hozzá tartozó kiegészítő csomagot (<http://www.virtualbox.org>), valamint letöltjük a megfelelő Ubuntu telepítő lemezképeket (<http://ubuntu.hu/letoltes/ubuntu>).

Ezen lépések után az alább leírt lépésekben végezhető a rendszer telepítése (a könyv írásakor a 4.1.18 -as volt az aktuális VirtualBox verzió, későbbi kiadásokban előfordulhatnak a leírtól eltérő opciók, lépések).

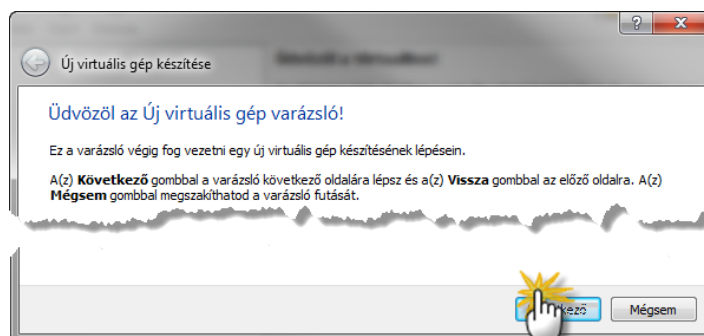
VirtualBox telepítésének négy fő feltétele van:

- kompatibilis Operációs rendszer (Windows, Linux, Solaris, OS X);
- kompatibilis processzor (virtualizációs támogatással);
- megfelelő méretű memória (megfelelő méret nagyságrendi kalkulációja: fizikai operációs rendszerhez ajánlott memória mennyiség + a kívánt virtuális rendszerhez ajánlott mennyiség + 10% - 15%), itt érdemes megjegyezni a lényeges memóriakezelés belső különbséget a 32 és 64 bites rendszerek között, valamint a számítógépünk kialakításából és a BIOS/chipset sajátosságaiból eredő korlátozásokat;
- elegendő merevelemez terület (megfelelő méret nagyságrendi kalkulációja: fizikai operációs rendszerhez ajánlott lemezterület + a kívánt virtuális rendszerhez ajánlott lemezterület + 10% - 15%).

Virtuális gép létrehozásának lépéseit az 1. - 7. ábrásor segítségével és rövid szöveges magyarázatokkal mutatjuk be.

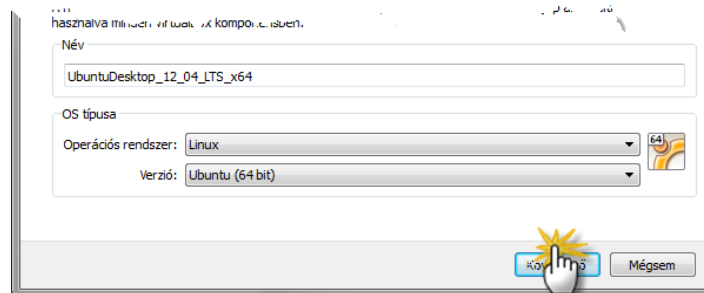


1. ábra. Kattintás az új ikonra



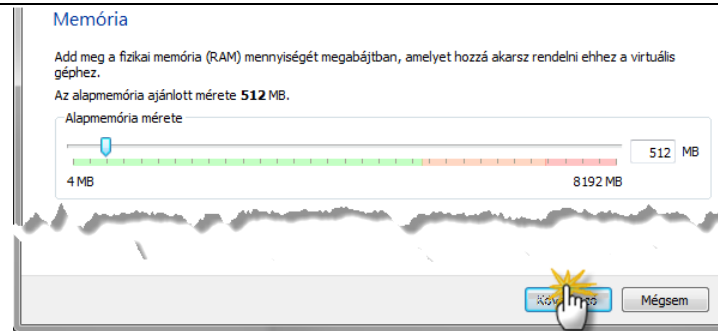
2. ábra. A varázsló továbbléptetése

Lehetőségünk van többféle operációs rendszer virtualizálására is, a különféle rendszerek optimális futtatásához a varázslóban (3. ábra. A rendszer típusának megadása) be kell állítanunk a kívánt operációs rendszer típusát. A beállítás közben a VirtualBox figyeli a „név” mező tartalmát, és ez alapján automatikusan is állítja az „OS típusa” szekció értékeit.



3. ábra. A rendszer típusának megadása

Amennyiben ismert rendszert választunk, a varázsló a következő lépésben felajánlja alapbeállításként a rendszerhez ajánlott memória mennyiséget. Emellett a méret beállítás skála alatti színes sáv vizuálisan is segít eldönteni, mi az mennyiség, amelyet biztonsággal adhatunk a virtuális gépnek (zöld tartomány) - 4. ábra. A memória beállítása



4. ábra. A memória beállítása

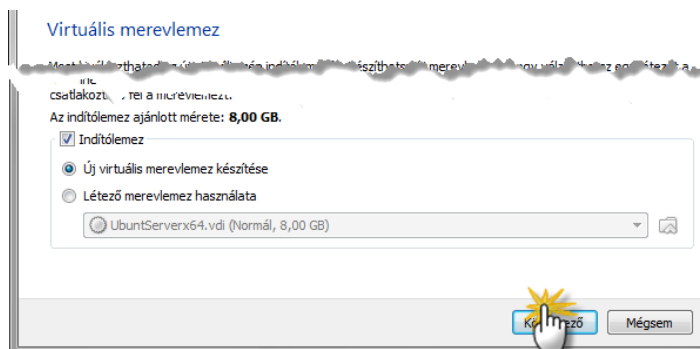
A vendég operációs rendszer számára biztosított memória beállítását követően a virtuális merevlemez paramétereit kell beállítanunk. A virtuális merevlemez tulajdonképpen egy lemezkép fájl lesz a fizikai gépünk egy adott mappájában.

A beállítás négy fő lépéből áll:

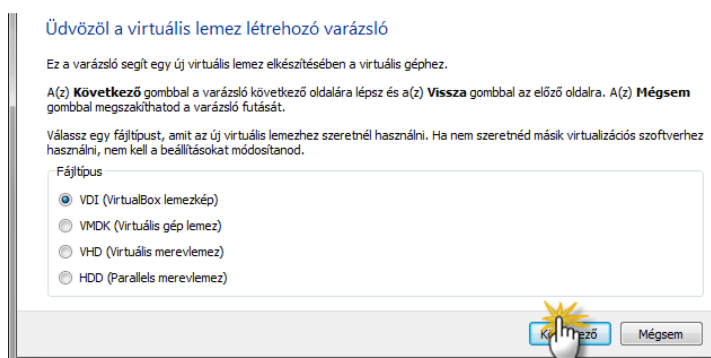
- eldönteni, hogy új, vagy létező lemezt szeretnénk-e használni;
- kiválasztani a virtuális lemezkép formátumát (lehetőség van arra, hogy kompatibilitást biztosítsunk más virtualizációs eszközökkel);
- a tároló részleteinek beállítása, ahol lehetőség van eldönteni, hogy a későbbi lépésben beállított lemezméret ténylegesen lefoglalásra kerüljön-e a fizikai gép merevlemezén (fix méretű), vagy növekményes fájlként mindig csak annyi helyet foglaljon, amennyit a vendég adott állapotában megkíván, de maximum az általunk beállítottat (dinamikusan növekvő);
- megadni a virtuális lemez helyét, fájlnevét és méretét.

A virtuális merevlemez paraméterezése (5. ábra. Virtuális merevlemez paraméterezése – 1) szintén több részből áll, melyben el kell döntenünk:

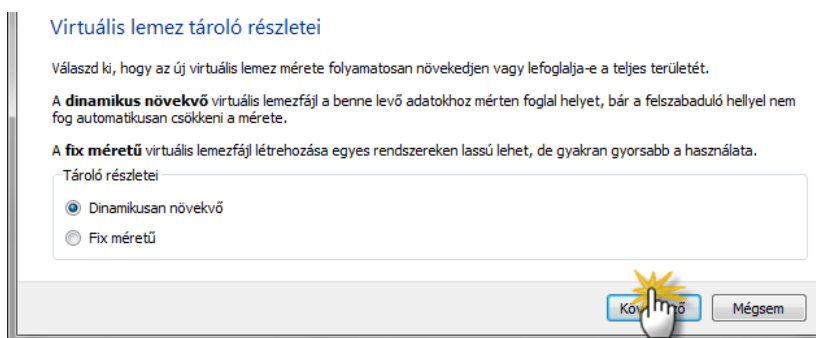
- a virtuális méretet (ezt fogja látni a vendég rendszer)
- a lemez típusát (lehetőség van kompatibilitást tartani más virtualizációs szoftverekkel)
- a tároló helyfoglalásnak alakulását a gazda rendszer szempontjából (azonnal foglaljon fizikai helyet, vagy dinamikusan a virtuális adatmennyiség függvényében).



5. ábra. Virtuális merevlemez paraméterezése – 1



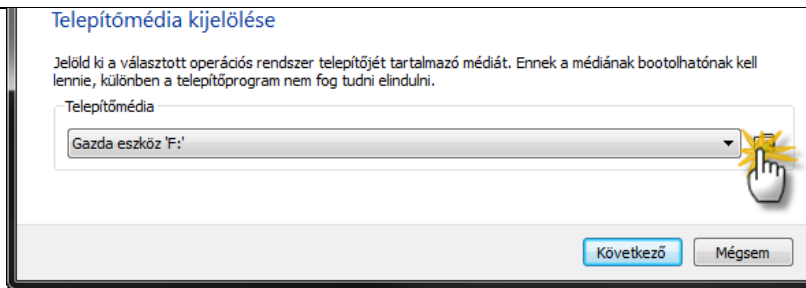
6. ábra. Virtuális merevlemez paraméterezése – 2



7. ábra. Merevlemez helyfoglalásának szabályozása

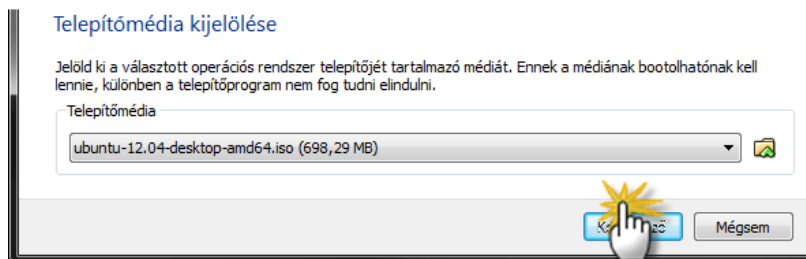
### 1.3. A létrehozott és előkonfigurált „üres” virtuális gépre immáron telepíthető a rendszer

A virtuális gép indításakor a VirtualBox érzékeli, hogy azon nincs telepített operációs rendszer, így elindít egy varázslót, amely segít az első telepítésben (8. ábra. Telepítés varázsló, telepítő média).



8. ábra. Telepítés varázsló, telepítő média

Ezen varázslóban megadhatjuk, hogy hol található a telepítendő operációs rendszer telepítő közege. Esetünkben ez a korábban a <http://www.ubuntu.hu> helyről letöltött iso lemezképeket jelentik, melyet a varázslóban betallózva (9. ábra. lemezkép választás), a VirtualBox automatikusan a vendég rendszer optikai egységébe emulálja, így érve el azt a hatást, mintha valójában betettük volna a telepítő korongot a gépbe.



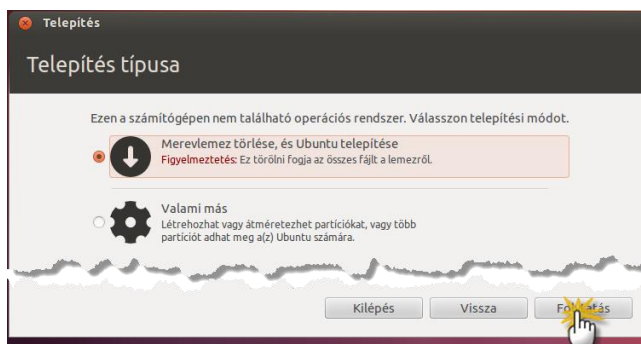
9. ábra. lemezkép választás

Ezt követően indul a telepítési folyamat, melynél ugyanúgy kell eljárunk, mintha egy fizikai gépre telepítenénk a rendszert.

Alább látható a telepítés menete képekben, rövid megjegyzésekkel:



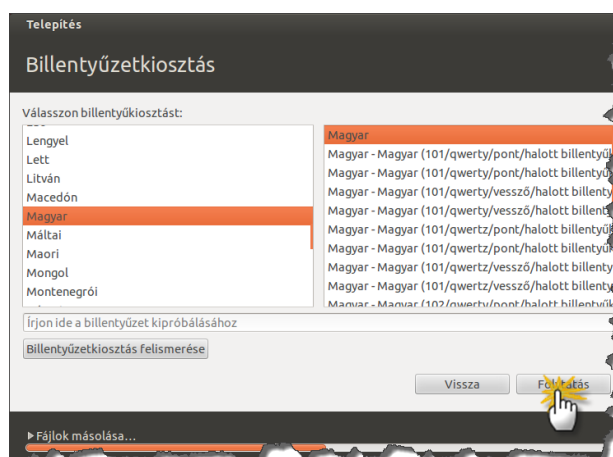
10. ábra. Nyelv választás



11. ábra. Telepítési típus meghatározása



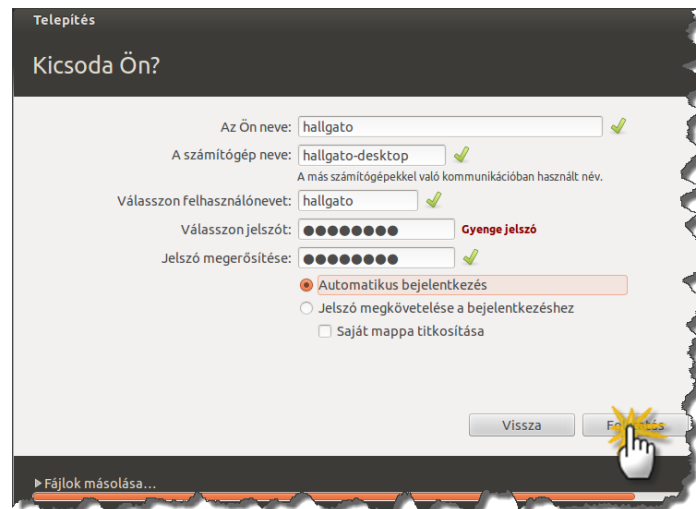
12. ábra. Időzóna meghatározás



13. ábra. Billentyűzetkiosztás választás

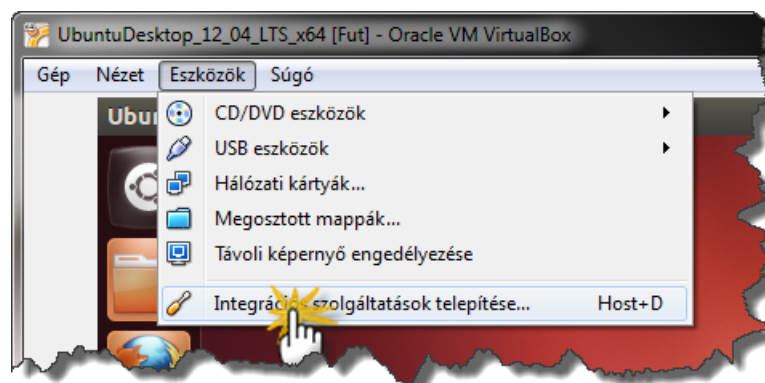
A felhasználó beállításakor (14. ábra. Gép és felhasználónév és jelszó) ne feledjük, hogy Ubuntu rendszereken másként működik a *root* felhasználó mint a legtöbb Linuxnál, és a telepítéskor létrehozott felhasználó jelszavával tudunk majd később rendszergazdai

feladatokat ellátni. A laborgyakorlatokhoz készített virtuális gépeknél az egyszerűség kedvéért a *hallgato* felhasználói nevet és *hallgato* jelszót fogjuk használni.



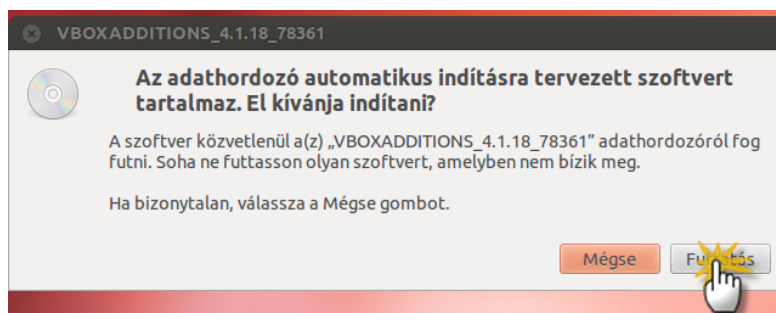
14. ábra. Gép és felhasználónév és jelszó

A telepítés befejezése után, amennyiben fizikai gépről lenne szó az eszközmeghajtók (drivereket) telepítése következne. VirtualBox esetén ezt a funkciót az „Integrációs szolgáltatások” (15. ábra. Integrációs szolgáltatások telepítése) látják el, ezért a frissen telepített rendszer indulása után ennek telepítésével kell folytatnunk.



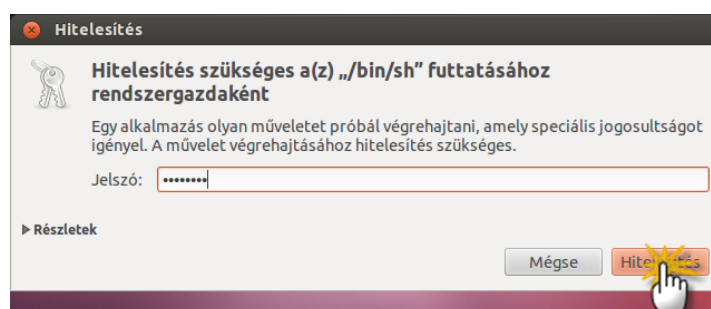
15. ábra. Integrációs szolgáltatások telepítése

Technikailag ez úgy zajlik, hogy a VirtualBox mappájában található telepítőlemez lemezképét a vendég rendszerünkbe emulálja, így a vendég rendszeren egy új CD/DVD lemez behelyezését fogja érzékelni a rendszer, amelyről telepíthető (16. ábra. Automatikus lejátszás) a szükséges komponens.



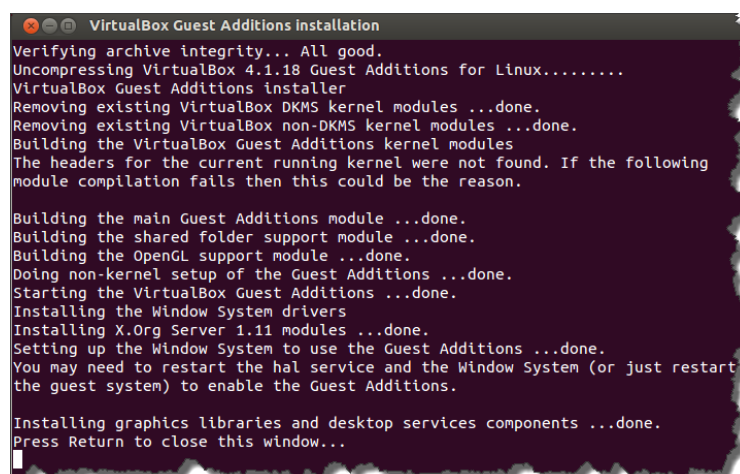
16. ábra. Automatikus lejátszás

A futtatáshoz és később még sok rendszergazdai szintű művelethez az Ubuntu hitelesítő ablaka tárul elő, melyben meg kell adnunk (17. ábra. Hitelesítés) a rendszergazdai jelszót (ami nem más, mint a telepítéskor létrehozott felhasználó jelszava)



17. ábra. Hitelesítés

A telepítési folyamatot (18. ábra. Telepítési részletek) egy terminál ablakban követhetjük.



18. ábra. Telepítési részletek



A telepítés végén a „telepítő lemezt” már kiadhatjuk (19. ábra. Lemez kiadása) a vendég gépből, ami történhet a VirtualBox eszközök menüjének segítségével, vagy az Ubuntu beépített eszközeivel is.



19. ábra. Lemez kiadása

A telepítés után egy újraindítás ajánlott.

Ezután már szinte indulásra készen állunk, azonban ne feledkezzünk el a frissen telepített rendszer frissítéséről sem, melyet a frissítés kezelő (20. ábra. Frissítés kezelő) segítségével könnyen elvégezhetünk (Internet kapcsolat szükséges).



20. ábra. Frissítés kezelő

Ezen folyamat során ismét találkozhatunk a hitelesítő ablakkal, illetve több újraindítás is szükséges lehet.

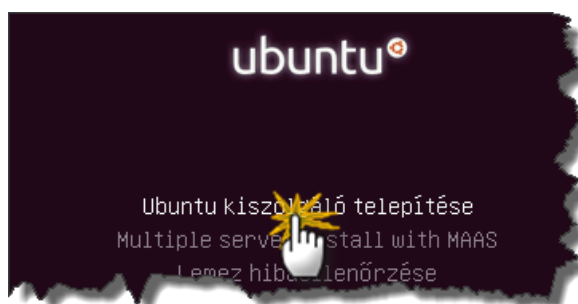
#### 1.4. A szerver változat telepítése

A szerver változathoz készítenünk kell egy új virtuális gépet hasonló módon, de immáron a szerver telepítő média kiválasztásával.

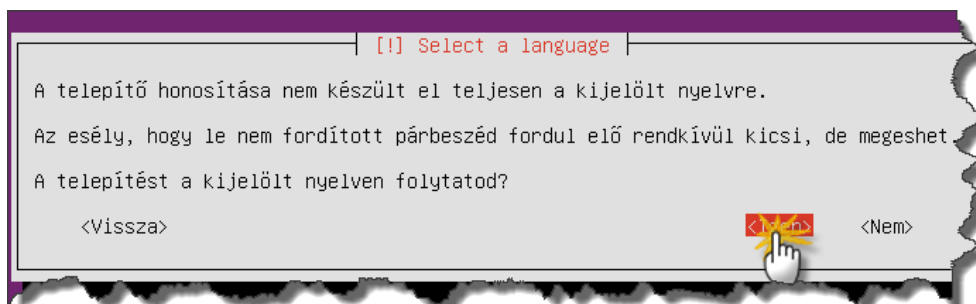
A szerver telepítése képekben, esetenként szükséges megjegyzésekkel kiegészítve:



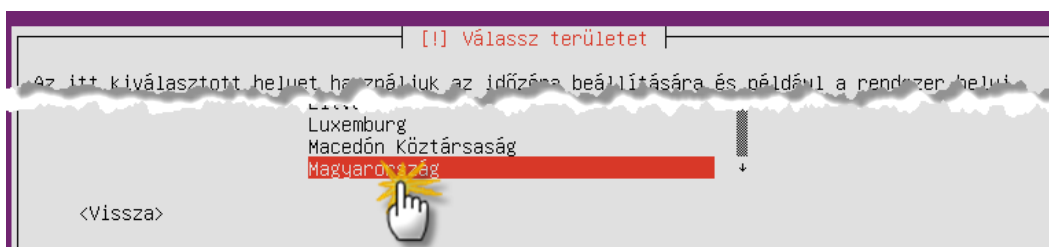
21. ábra. Nyelvválasztás



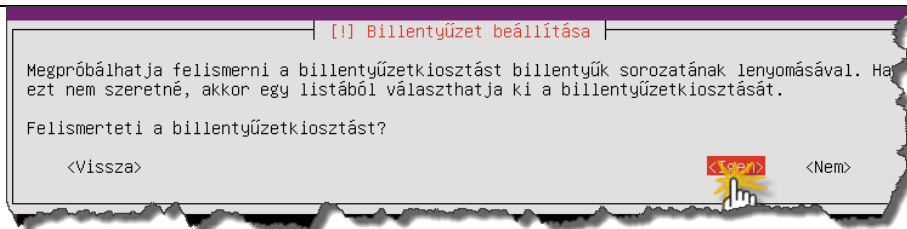
22. ábra. Telepítés indítása



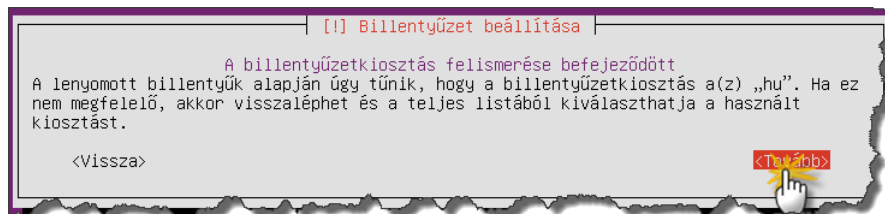
23. ábra. Figyelmeztetés hiányos lokalizációra



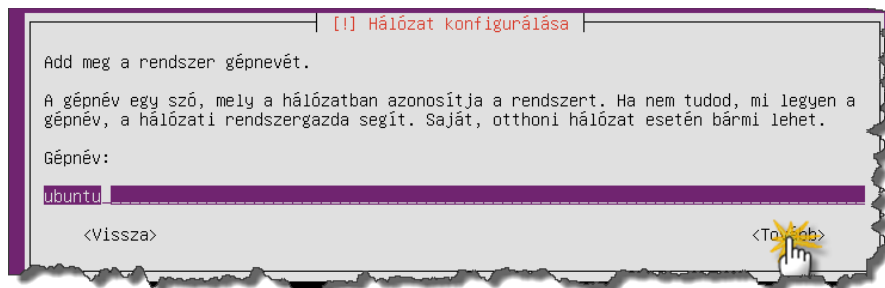
24. ábra. Régió megadása



25. ábra. Billentyűzetkiosztás beállítása

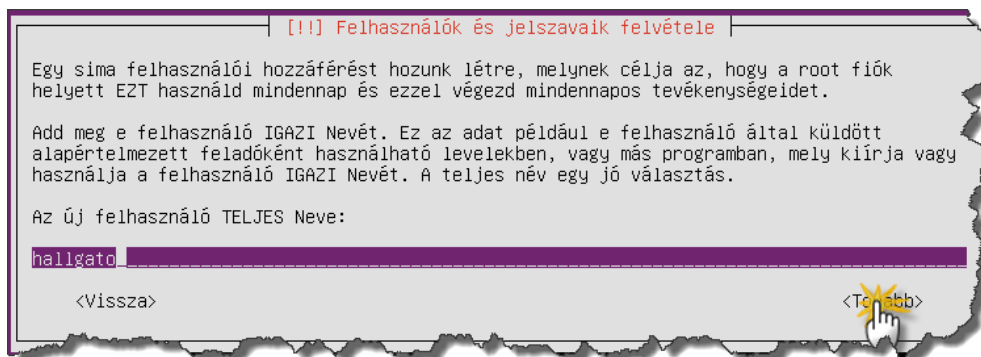


26. ábra. Billentyűzet detektálás eredménye

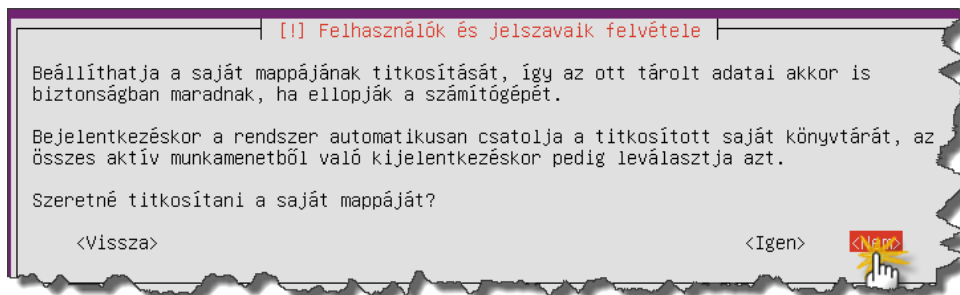


27. ábra. Gépnév megadása

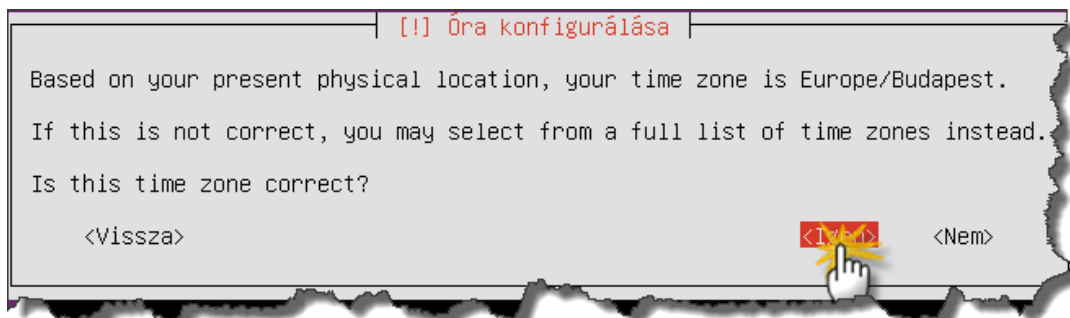
A felhasználónév (28. ábra. Felhasználónév megadása) és jelszó vonatkozásában az Ubuntu szerver rendszerre is igaz, amit desktop esetén írtunk.



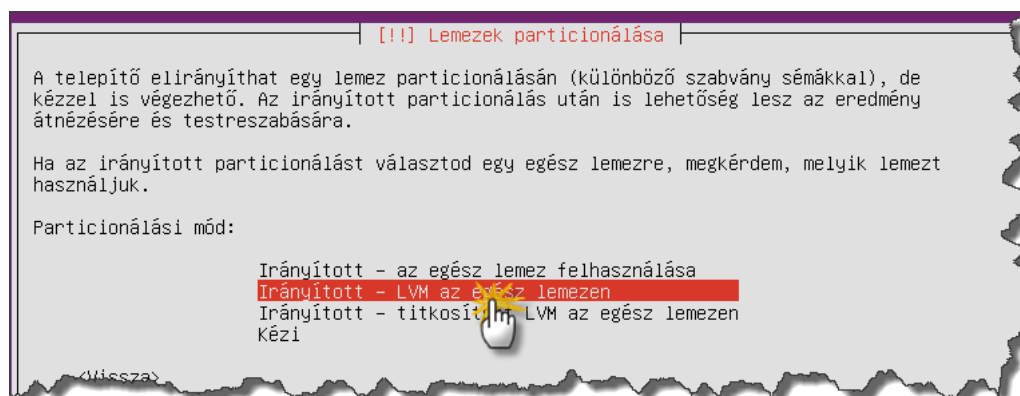
28. ábra. Felhasználónév megadása



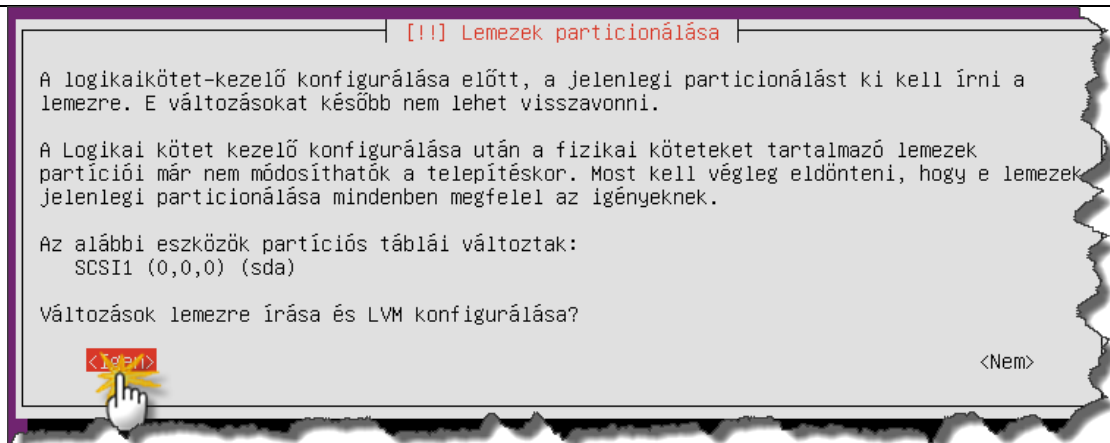
29. ábra. Titkosítás beállítása



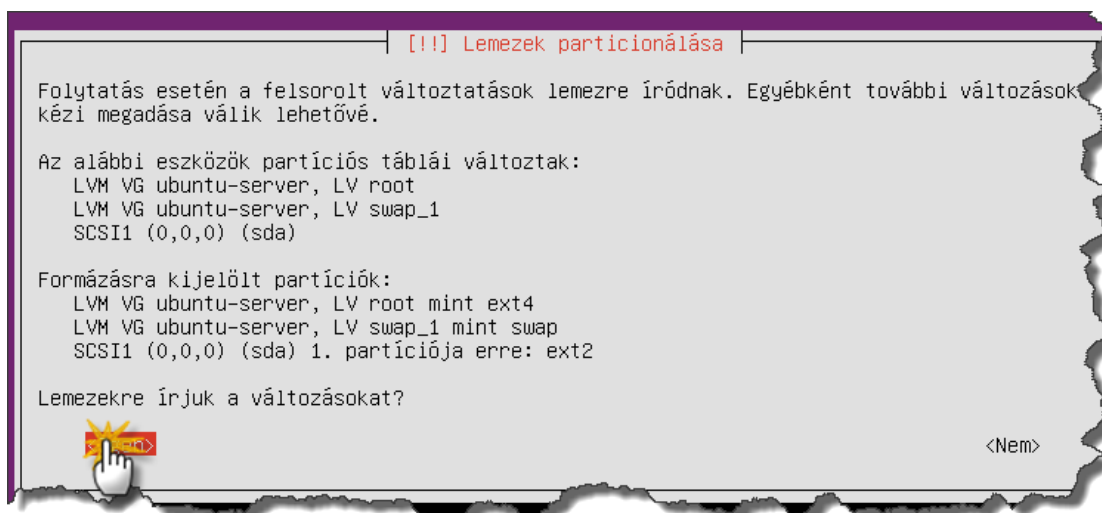
30. ábra. Időzóna beállítás



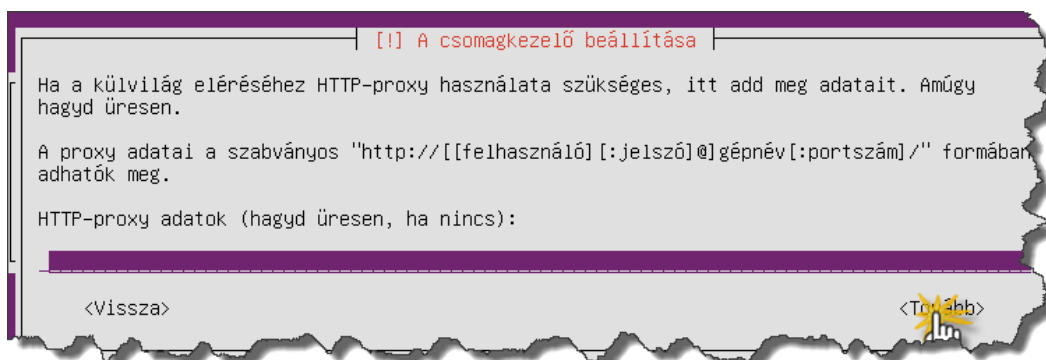
31. ábra. Lemez és partíció kezelés



32. ábra. Particionálás összegző



33. ábra. Lemez és particionálás összegzése

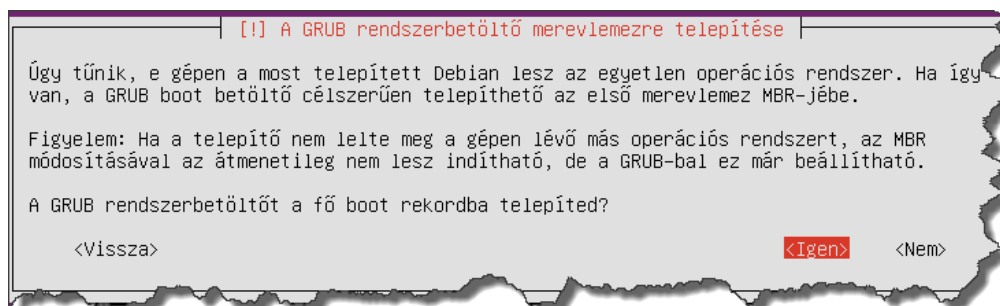


34. ábra. Proxy beállítás

Tekintettel arra, hogy ez egy gyakorló rendszer lesz, amelyen majd mi magunk telepítjük és konfiguráljuk a különféle szolgáltatásokat (35. ábra. Szolgáltatás választó), a telepítés során egyetlen szolgáltatást sem telepítetünk a telepítővel.

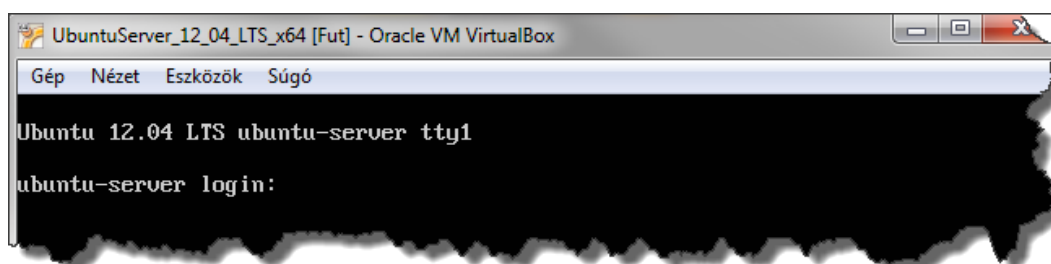


35. ábra. Szolgáltatás választó



36. ábra. GRUB helyének beállítása

A telepítés végső fázisaként (37. ábra. Telepítés eredménye) megkapjuk a konzolos felületet.



37. ábra. Telepítés eredménye

A Desktop rendszerhez hasonlóan, itt is célszerű frissítéseket végeznünk a telepítés után.

Az alábbi utasítás-példákban a szakirodalomban használatos konvenciónak megfelelően a „\$” jel a készenléti jel utolsó karakterét jelzi, ezt a jelet nem kell begépelni a kipróbálás során.

A frissítések telepítéséhez az alábbi lépésekre van szükség:

1. bejelentkezünk a rendszerbe,

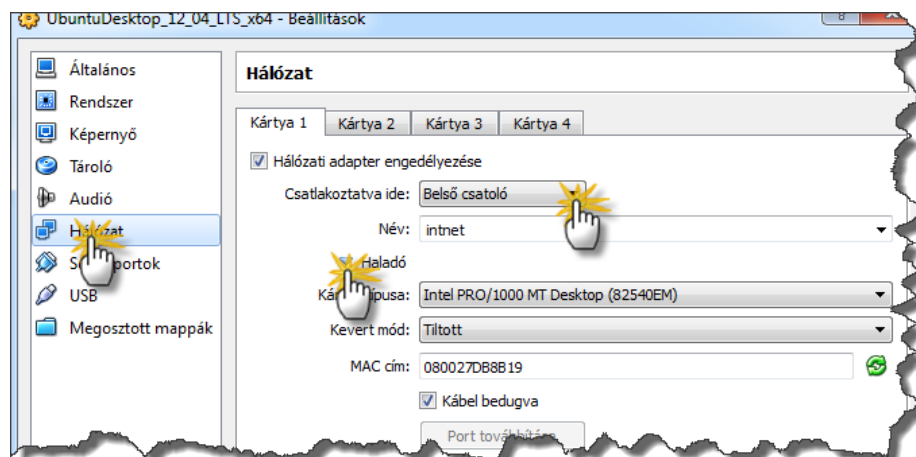
2. rendszergazda módba lépünk ( ehhez a javasolt eljárás a `$ sudo -s` parancs használata, mely után ismét meg kell adni jelszavunkat ),
3. `apt` csomagtelepítő segítségével lekérjük a frissítéseket ( `$ apt-get update` parancs ),
4. `apt` segítségével telepítjük a frissítéseket ( `$ apt-get upgrade -y` parancs segítségével),
5. frissítések után a gép leállítását ( `$ halt` parancs ) vagy újraindítását ( `$ reboot` parancs ) is kezdeményezhetjük.

A két frissen telepített rendszer jelen állapotáról érdemes egy pillanatképet készíteni a VirtualBox Gép -> Pillanatképfelvétel készítése funkciója segítségével a gép kikapcsolt állapota mellett. Az így készített pillanatképekre vissza tudunk állni később, így egy-egy új feladat megoldása során minden alkalommal friss rendszerrel dolgozhatunk.

## 1.5. Belső hálózat kialakítása virtuális gépek között

Hálózati szolgáltatások teszteléséhez, gyakran szükségünk van egy hálózatra, ami a futó virtuális gépeket köti össze. VirtualBox-ban ez a feladat néhány egyszerű lépéssel elvégezhető, az alábbiak szerint.

Állítsuk le a virtuális gépeket, és a VirtualBox kezelőben a gép kiválasztása után válasszuk a konfigurálás lehetőséget. A hálózat szekcióban az első kártya adatait szerkesszük az alábbi (38. ábra. Hálózati csatoló konfigurációja) kép szerint.



38. ábra. Hálózati csatoló konfigurációja

A bevezetőben írt módok közül jelen esetben a belső csatolós üzemmód lesz a segítségünkre. Ezután lehetőség van a hálózatunk elnevezésére. Ez az elnevezés igen fontos, hiszen ez határozza meg, hogy az esetleg létező belső hálózatok közül melyek vannak összekapcsolva.

Tehát mindkét gép esetén ezeket a beállításokat kell elvégeznünk, és ügyelnünk kell az azonos elnevezésre is.

Az Ubuntu szerver változata esetén hatványozottan érdekes a haladó fül alatt található MAC azonosító is, ezt érdemes feljegyeznünk valahová, mert fontos szerepet játszik később a hálózat konfigurációjában.

Miután ezt elvégeztük, indíthatjuk a virtuális gépeket. Belső csatoló esetén a VirtualBox nem lát el többé DHCP kiszolgálói funkciót, mint NAT esetben, így nekünk kell gondoskodnunk arról, hogy a gépek a megfelelő IP beállításokat kapják. Ennek végrehajtásával a 3. fejezetben ismerkedünk meg részletesen.

## **1.6. Vonatkozó irodalomjegyzék**

1. VirtualBox  
<https://www.virtualbox.org/>
2. Ubuntu Magyarország  
<http://ubuntu.hu/>
3. Ubuntu felhasználói kézikönyv  
<https://help.ubuntu.com/12.04/ubuntu-help/index.html>
4. Ubuntu.com – Installing Ubuntu 12.04  
<https://help.ubuntu.com/12.04/installation-guide/index.html>



## 2. Kezelési alapismeretek (Kovács Péter)

Könyvünk második fejezetében a fontosabb parancssori utasításokkal és eszközökkel ismerkedünk.

### 2.1. Néhány fontosabb parancssori utasítás használata az asztali (Desktop) Ubuntu változat esetén

Nyissunk egy karakteres terminált a grafikus felületen:



39. ábra. Terminál

Az alábbi utasítás-példákban a szakirodalomban használatos konvenciónak megfelelően a „\$” jel a készenléti jel utolsó karakterét jelzi, ezt a jelet nem kell begépelni a kipróbálás során.

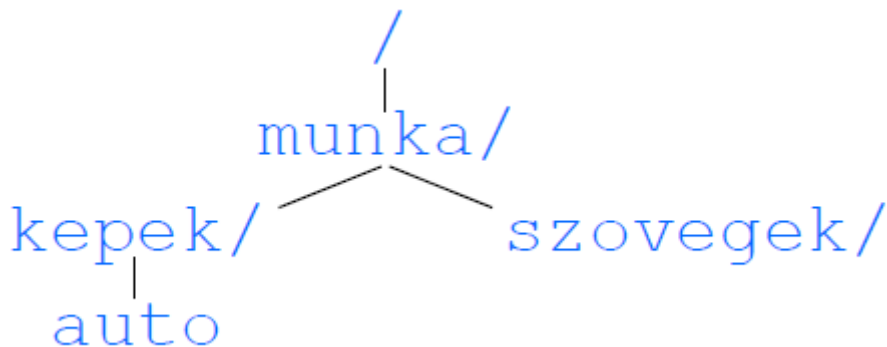
#### Az elérési utak

A terminálban történő munkavégzés során kulcsfontosságú tudni, hogy a mappák/fájlok, amikkel dolgozunk, hol találhatóak. Ez a szövegesen meghatározott útvonal a fájl/mappa elérési útja. Amikor terminálban dolgozunk a TAB billentyűvel elérhető automatikus kiegészítés segít nekünk beírni ezeket, amennyiben helyesen kezdtük el. Elérési utak esetében két típust különböztetünk meg az abszolút (a gyöklér könyvtárból indulunk ki) és a relatív (a jelenlegi könyvtárból indulunk ki) utat. A navigációban használatos jelek:

- / - a fájlrendszer gyökere
- . - az aktuális könyvtár
- .. - egy könyvtárral feljebb

Példa az elérési utakról:

Tegyük fel, hogy a gyökérkönyvtárból nyílik egy „munka” nevű alkönyvtár, amely tartalmazza a „kepek” és „szovegek” alkönyvtárakat. Továbbá a „kepek” könyvtárban legyen egy „auto” nevű állomány. Szemléletesebben az alábbi ábrán



40. ábra. Könyvtár felépítés

Legyen az aktuális könyvtár a „kepek”. Ekkor:

- a „kepek” könyvtár relatív elérési útja: `.` vagy `./`
- az „auto” állomány abszolút elérési útja: `/munka/kepek/auto`
- az „auto” állomány relatív elérési útja: `auto` vagy `./auto`
- a „munka” könyvtár abszolút elérési útja: `/munka` vagy `/munka/`
- a „munka” könyvtár relatív elérési útja: `..` vagy `../`
- a „szovegek” könyvtár relatív elérési útja: `../szovegek` vagy `../szovegek/`

## Munka állományokkal és könyvtárakkal

A `cd` parancs használatának személtetése:

- `cd, cd ~`: belépés az aktuális felhasználó saját könyvtárába
- `cd ~root`: belépés a root felhasználó saját könyvtárába

Legyenek „level” és „masolat” állományok, „szovegek” és „dokumentumok” pedig könyvtárak. A `cp` parancs használatának személtetése:

- `cp level masolat`: a „level” állomány (tartalmának) átmásolása a „masolat” állományba
- `cp level szovegek`: a „level” állomány bemásolása a „szovegek” könyvtárba változatlan néven
- `cp -r szovegek dokumentumok`: a „szovegek” könyvtár teljes tartalmának átmásolása a „dokumentumok” könyvtárba

Legyenek „szoveg” és „level” állományok, „dokumentumok” pedig egy könyvtár.

Az `mv` parancs használatának személtetése:

- `mv szoveg level`: a „szoveg” állomány átnevezése „level”-re

- `mv szoveg dokumentumok`: a „szoveg” állomány átmozgatása (áthelyezése) a „dokumentumok” könyvtárba

Legyen „level” egy állomány, „szovegek” pedig egy könyvtár. Az `rm` parancs használatának személtetése:

- `rm level`: a „level” állomány kitörlése
- `rm -rf szovegek`: a „szovegek” könyvtár törlése annak teljes tartalmával együtt
- A `basename` és `dirname` parancsok használatának személtetése:
  - `basename /munka/kepek/auto`: az eredmény az `auto` útvonal
  - `dirname /munka/kepek/auto`: az eredmény a `/munka/kepek` útvonal

## Az `ls` parancs

Legyen „level” egy állomány, „szovegek” pedig egy könyvtár. Az `ls` parancs használatának személtetése:

- `ls -l level` („`ell`” opció): a „level” állomány adatainak kiírása (bővített listázás)
- `ls -d szovegek`: a „szovegek” könyvtár mint speciális állomány adatainak kiírása
- `ls szovegek`: a „szovegek” könyvtár tartalmának kiírása
- `ls -l szovegek` („`egy`” opció): mint előbb, de az egyoszlopos módot használva
- `ls -R szovegek`: a „szovegek” könyvtár teljes tartalmának kiírása, az alkönyvtárakat is beleértve
- A „kepek” és a „szovegek” könyvtárak tartalmának ki listázása.  
`ls kepek szovegek`

## Állományok és könyvtárnevek megadása maszk alkalmazásával

- `kep?`: olyan négybetűs nevek, amelyek a „kep” szóval kezdődnek, és az utolsó karakterük tetszőleges
- `kep*`: olyan nevek, amelyek a „kep” szóval kezdődnek, amit bármi egyéb követhet (akár az üres szó is)
- `*kep`: olyan nevek, amelyek a „kep” szóra végződnek, amit bármi egyéb megelőzhet (akár az üres szó is)
- `kep[A12]`: olyan négybetűs nevek, amelyek a „kep” szóval kezdődnek, és az utolsó karakterük „A”, „1” vagy „2”
- `kep[A-Z0-9]`: mint előbb, de az utolsó karakterük nagybetű vagy számjegy
- `kep[^A-Z]`: mint előbb, de az utolsó karakterük nem nagybetű
- `kep[^A-Z]*[23]`: olyan nevek, amelyek a „kep” szóval kezdődnek, amit egy nagybetűtől eltérő karakter követ, utána bármi állhat, az utolsó karakterük pedig „2” vagy „3”

## Átírányítás és csővezeték

A könyvtárlista elmentése a „lista” állományba:

```
ls > lista
```

Mint előbb, de ha már létezik az állomány, akkor hozzáfűzést alkalmazunk.

```
ls >> lista
```

A „level” állomány tartalmának kiírása. Az esetlegesen megjelenő hibaüzeneteket (pl. ha „level” egy könyvtár lenne, vagy ha nem lenne rá olvasási jogunk) a „hiba” állományba irányítjuk.

```
cat level 2> hiba
```

A „teszt” nevű alkönyvtár létrehozása úgy, hogy ha már létezett ilyen nevű könyvtár, akkor ne jelenjen meg hibaüzenet a képernyőn.

```
mkdir teszt 2> /dev/null
```

Kihasználtuk, hogy a /dev/null speciális állomány minden bele írt adatot elnyel.

A „kepek” és „szovegek” könyvtárak listájának eltárolása a „lista” állományba. Az esetlegesen keletkező hibaüzeneteket a „hiba” állományba irányítjuk. Gyakorlatilag tehát semmilyen látható kimenetet nem produkálunk.

```
ls kepek szovegek 2> hiba > lista
```

Mint előbb, de most a „lista” állományba irányítjuk a hibaüzeneteket is.

```
ls kepek szovegek &> lista
```

A könyvtárlistát elmentjük a „lista” állományba, de a képernyőn is szeretnénk látni az eredményt.

```
ls | tee lista
```

Mint előbb, de a listát továbbra is többoszloposként szeretnénk látni.

```
ls -C | tee lista
```

A „kepek” és „szovegek” könyvtárak listájának eltárolása a „lista” állományba úgy, hogy az eredmény a képernyőn is megjelenik. Az esetlegesen keletkező hibaüzeneteket ugyanígy kezeljük, azaz az állományba beírjuk és a képernyőn is megjelenítjük.

```
ls kepek szovegek 2>&1 | tee lista
```

Az „uzenet” állomány tartalmának kiírása a szabványos hibakimenetre.

```
cat uzenet 1>&2
```

Mint előbb, de a hibaüzenetet a „hiba” állományba irányítjuk.

```
cat uzenet 2> hiba 1>&2
```

Az átirányítások sorrendje most fontos! Ha felcserélnénk őket, akkor a szabványos kimenet tartalma oda menne, ahová a hibakimenet akkor éppen irányítva van, azaz a képernyőre!

## A chmod parancs

Tegyük fel, hogy az aktuális mappában létezik egy *pelda* nevű állomány és egy *szovegek* könyvtár.

A „pelda” állomány futtathatóvá tétele a tulajdonos számára (a többi jog nem módosul):

```
chmod u+x pelda
```

A „pelda” állomány olvasási és írási jogainak tiltása az állomány csoportja és az egyéb felhasználók számára (a többi jog nem módosul):

```
chmod go-rw pelda
```

A „szovegek” könyvtár, valamint az abban levő állományok és az alkönyvtárak teljes tartalmának futtathatóvá tétele mindenki számára (a többi jog nem módosul). A futtatási jogot csak könyvtáraknak és az eleve futtatható állományoknak adjuk meg:

```
chmod -R a+X szovegek
```

A „pelda” állomány összes jogának megvonása az egyéb felhasználók számára (a többi jog nem módosul):

```
chmod o= pelda
```

A „pelda” állományt mindenki számára olvashatóvá tesszük, a többi jogot pedig letiltjuk:

```
chmod a=r pelda
```

Egy ekvivalens megoldás a jogok numerikus alakjának használatával. A numerikus érték (UMASK) meghatározásának módja igen egyszerű, az első számjegy a tulajdonos jogosultsága, a második a csoporté, a harmadik pedig az egyéb felhasználóké. A számjegyek értéke az adott pozíción a jogosultságo(ka)t mondja meg, ahol 1 a futtatás, 2 az írás és 4 az

olvasás. ezen értékek összeadásával adhatunk több jogosultságot az adott pozíción az adott felhasználónak/csoportnak. A 0 érték képviseli a jogosultságok megvonását.

```
chmod 444 pelda
```

A „pelda” állomány a következő jogokkal fog rendelkezni: a tulajdonos olvasni, írni és futtatni (az első pozíció:  $4+2+1 = 7$ ) is tudja, a csoport pedig képes olvasni és futtatni (a második pozíció:  $4+1 = 5$ ), az egyéb felhasználóknak pedig semmilyen joguk sincs (harmadik pozíció: 0):

```
chmod 750 pelda
```

A szimbolikus jogok alkalmazásával kicsit hosszabb megoldást kapunk. A szimbolikus jogok esetén a fenti numerikus meghatározás helyett a jogosultságok angol elnevezésének rövidítését használjuk (r = read/olvasás, w = write/írás, x = execute/futtatás) az üres érték itt is a jogosultságok megvonását jelenti, illetve a helyiérték pozíciója helyett szintén rövidítéssel adjuk meg, hogy a jogosultság kire vonatkozzon (u = user/tulajdonos, g = group/csoport, o = other/egyéb, a = all/mindenki).

```
chmod u=rwx pelda
```

```
chmod g=rwx pelda
```

```
chmod o= pelda
```

## Szűrők használata

A „Kovács Jancsi” nevű felhasználóról nyilvántartott kritikus információkat tartalmazó sor megjelenítése az /etc/passwd állományból.

```
cat /etc/passwd | grep ':Kovács Jancsi:'
```

Az összes bejelentkezések számának kiírása. A többször bejelentkezett felhasználókat többször számoljuk.

```
who | wc -l
```

A bejelentkezések felhasználói azonosító szerint rendezett listája.

```
who | sort
```

A „h123456” azonosítójú felhasználó bejelentkezéseinek kiírása. Annyi sor fog megjelenni, ahányszor be van jelentkezve.

```
who | grep 'h123456'
```

Kiírja, hogy hányszor van bejelentkezve a „h123456” azonosítójú felhasználó.

```
who | grep 'h123456' | wc -l
```

A „szoveg” állomány sorai számának kiírása.

```
cat szoveg | wc -l
```

Fontos, hogy a következő megoldás már nemcsak a sorok számát írja ki, hanem az állomány nevét is! Ezért inkább az előzőt célszerű használni.

```
wc -l szoveg
```

Vegyük észre, hogy mind a két esetben a szám előtt szóközök is kiíródnak! Ezek eltüntetésére a shellnél látunk majd módszert.

Tegyük fel, hogy a „lista1” és „lista2” állományok minden sora egy-egy szót tartalmaz. Készítünk egy rendezett listát, amely a két állományban előforduló szavakat tartalmazza fordított (csökkenő) sorrendben. Minden sor pontosan egy szót tartalmaz, a többször előforduló szavaknak csak egyetlen példányát hagyjuk meg, továbbá a kisbetűket és a nagybetűket nem különböztetjük meg!

```
sort -fru lista1 lista2
```

A „szoveg” állomány harmadik sorának megjelenítése.

```
head -n 3 szoveg | tail -n 1
```

Egy ekvivalens megoldás:

```
head -n 3 szoveg | tail -n +3
```

## 2.2. A vi használata

Készítsünk egy új szöveges állományt három sorral a vi program segítségével:

```
$ vi valami
```

A vi editor alkalmas új állományok létrehozására, meglévők módosítására bármilyen terminálon. Ennek akkor van nagy jelentősége, ha nincs lehetőség grafikus szerkesztő használatára, illetve egy grafikus kapcsolat létrehozásánál egyszerűbb vi editort használni, például, ha csak kis mértékben szeretnénk megváltoztatni az állomány tartalmát. A program indítása

```
$ vi állománynév
```

utasítással történik, ahol az állománynév a létrehozandó vagy módosításra szánt állomány neve. A szövegírónak három üzemmódban dolgozhat, ezek a *parancs*, az *utolsó sori* és a *beíró üzemmód*. A szoftver indítása után parancs üzemmódba kerülünk. Ekkor az alábbi utasításokat használhatjuk:

- x** a kurzor melletti karakter törlése;
- dw** kurzortól kezdődő szó törlése;
- dd** az aktuális sort törlése;
- u** az utolsó változtatás visszavonása;
- U** aktuális sor minden változtatásának visszavonása;
- .** az utolsó szöveglétrehozó, módosító vagy törlő parancs megismétlése;
- J** az aktuális sor összevonása a következővel.

Szöveg begépeléséhez először át kell térni beíró üzemmódba, ami az <a>, <A>, <i>, <I>, <o>, <O> billentyűk egyikének lenyomásával lehetséges. Hatásuk:

- a** a kurzor mögé írhatunk,
- A** az aktuális sor végére írhatunk,
- i** a kurzor elé írhatunk,
- I** az aktuális sor elejére írhatunk,
- o** új sort kezdhetünk az aktuális sor után,
- O** új sort kezdhetünk az aktuális sor előtt.

Ha begéveltük a szöveget, akkor az <ESC> billentyűvel térhetünk vissza parancs üzemmódba. A beírt szöveg elmentésére az utolsó sori üzemmódban nyílik lehetőség. Parancs üzemmódból utolsó sori üzemmódba kettőspont megnyomásával léphetünk át.

Ezután az alábbi lehetőségek állnak rendelkezésünkre:

- :w** az állomány elmentése;
- :w állománynév**  
a dokumentum elmentése a megadott néven;
- :wq** a dokumentum elmentése, és kilépés a vi editorból;
- :q!** figyelmeztetés és mentés nélküli kilépés a szövegíróból.

Ha olyan parancsot adtunk ki, amelyik nem lép ki a szerkesztőből, akkor az <ENTER> billentyű lenyomása után visszakerülünk parancs üzemmódba.

Készítsük el a valami állományban a következő sorokat. Milyen alapértelmezésbeli védelmi kódokat adott a rendszer az új állománynak?

```
baba
szep
piroska
farkas
mese
mese
```



matka  
vadaszroka

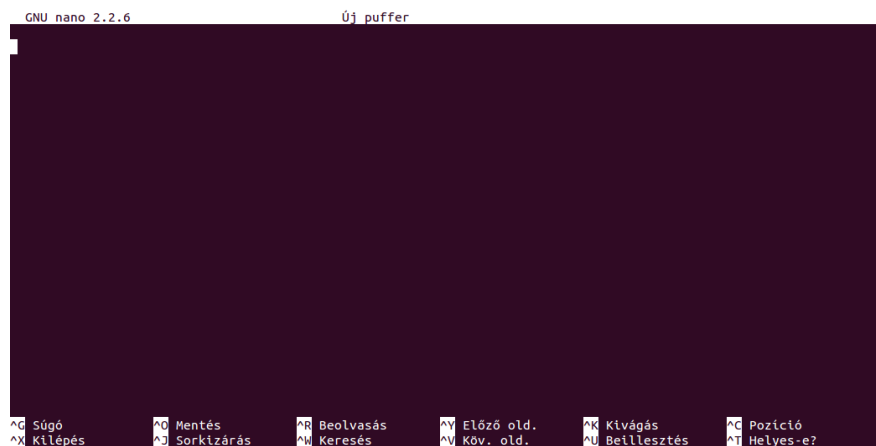
Töröljük a valami állományt.

```
$ rm valami
```

## 2.3. A nano szerkesztő használata

A nano szerkesztő egy egyszerűen kezelhető, az Ubuntu rendszerekben alapértelmezésként telepített szövegszerkesztő alkalmazás. Használata jóval egyszerűbb, mint a korábban említett vi szerkesztőé, így kezdőknek különösen ajánlott, a későbbi példákban könyvünk is ezt használja. Megjelenése és funkcionalitása talán az MS-DOS rendszerek EDIT programjához hasonlítható leginkább.

A szerkesztő indítása a `$ nano` paranccsal történik, amennyiben egy fájlnevet is megadunk szóközzel elválasztva a szerkesztő azzal a fájlal dolgozik. Természetesen lehetőség van a fájlnev előtt parancssori opciók megadására is. Az opciók közül kiemelnénk a `-w` kapcsolót, melynek hatására megváltozik a sortörések kezelése, ez hasznos lehet olyan konfigurációs állományok szerkesztésekor, ahol lényeges a sortörések helye és száma. Ha a megadott fájl létezik olvasásra/szerkesztésre jeleníti meg a tartalmát, amennyiben nem, megkísérli létrehozni azt.



41. ábra. A nano szerkesztő

Az alsó két sorban láthatók az elérhető parancsok, és a hozzájuk kapcsolódó, fehér háttérrel kiemelt billentyűkódok, amelyekben a ^ jel a CTRL billentyű lenyomását jelenti (figyelem, mivel a VirtualBox alapértelmezésként a jobboldali CTR billentyűt használja, így amennyiben a vendég rendszernek szeretnénk CTRL leütést küldeni, a baloldali CTRL billentyűt kell használnunk). Amennyiben az adott menüpontnak almenüi is vannak, hasonló szisztéma szerint az alsó két sorban kerülnek megnyitásra, illetve a menü feletti sorban van lehetőségünk a szerkesztő kérdéseire válaszolni (pl.: fájlnev megadása, igen-nem jellegű opciók).

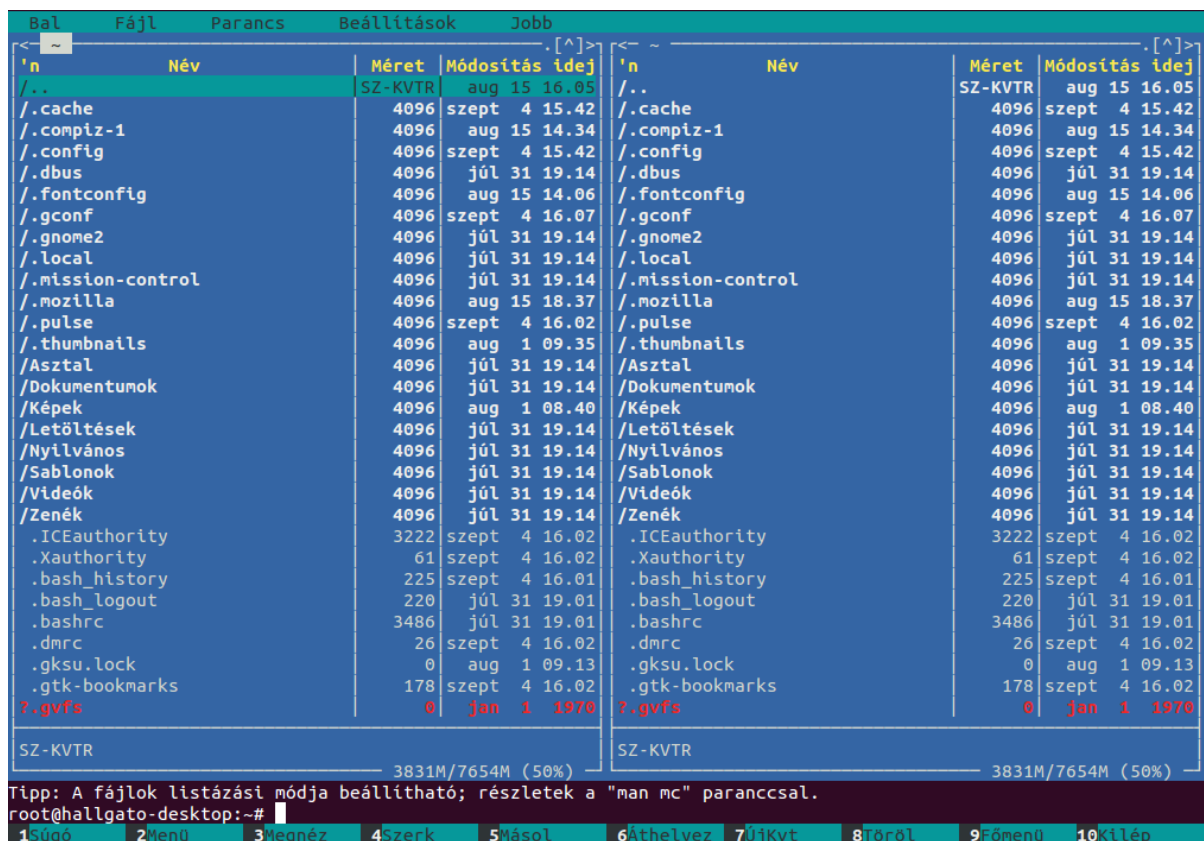
A menüpontok ismertetésére a lokalizált verzió miatt nem szentelnénk most időt, a parancsok és neveik magától értetődővé teszik a program használatát, aki mégis bővebben utána kíván járni, a fejezet irodalomjegyzékében megtalálja a program weboldalának hivatkozását.

## 2.3. A Midnight Commander használata

Az állományokkal kapcsolatos műveletek során igen hasznos segédeszköznek bizonyulhat a Midnight Commander. Karakteres képernyőkezeléssel dolgozik, így nem támaszt különösebb igényeket a terminállal szemben. Elindítása az mc paranccsal lehetséges. Megjelenése (42. ábra) hasonlít a jól ismert Total Commanderhez, azaz az alsó sorban az egyes funkcióbillentyűkhöz rendelt feladatok láthatók, két ablakban párhuzamosan két könyvtár tartalomjegyzékét kísérhetjük figyelemmel. Az <F9>-es billentyű lenyomásával juthatunk a felső menüsorba.

A program segítségével állományokat másolhatunk, mozgathatunk, törölhetünk, szerkeszthetünk és hozhatunk létre, valamint beállíthatjuk a védelemhez kapcsolódó információkat, természetesen amennyiben jogosultak vagyunk ezek megtételére.

A Total Commanderrel való nagymértékű hasonlóság miatt a továbbiakban csak az attól eltérő jellemzőkre térünk ki.

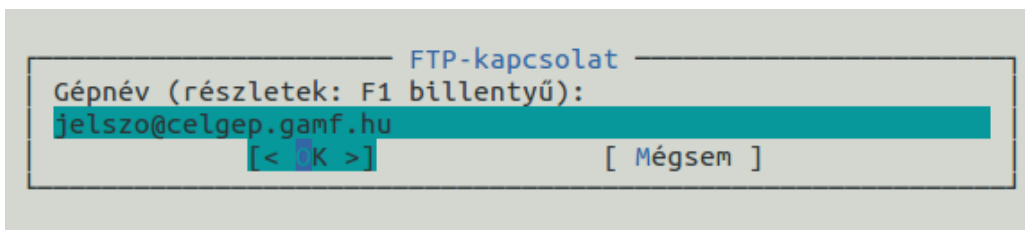


42. ábra. Midnight Commander

### 2.2.1. FTP kapcsolat létrehozása távoli számítógéppel

A két ablak közül bármelyikbe behozhatjuk egy távoli számítógép valamely FTP segítségével elérhetővé tett könyvtárának tartalomjegyzékét, majd letölthetünk, illetve felmásolhatunk állományokat, ugyanúgy mintha az eredeti gép egyik könyvtárából a másikba másolnánk. Lehetőség van könyvtár váltásra és törlésre is. A kapcsolat létrehozásához a *FT-kapcsolat...*

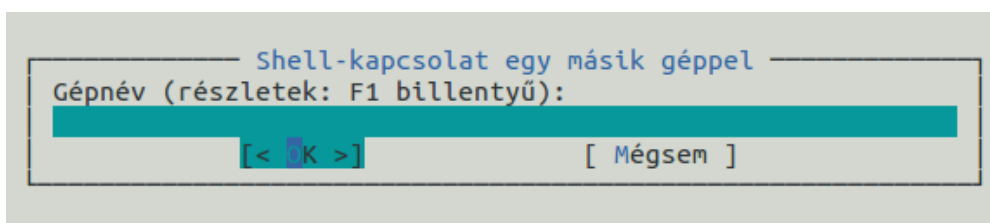
menüpontot kell kiválasztanunk a *Bal* vagy *Jobb* legördülő menüben. Ezután a 43. ábrán látható párbeszédablakban kell megadni a távoli gépen érvényes felhasználói azonosítónkat, jelszavunkat, a gép Internet címét (IP vagy FQDN), és a könyvtár elérési útvonalát.



43. ábra. FTP kapcsolat létrehozása távoli számítógéppel

### 2.2.2. Távoli gép könyvtárának csatolása

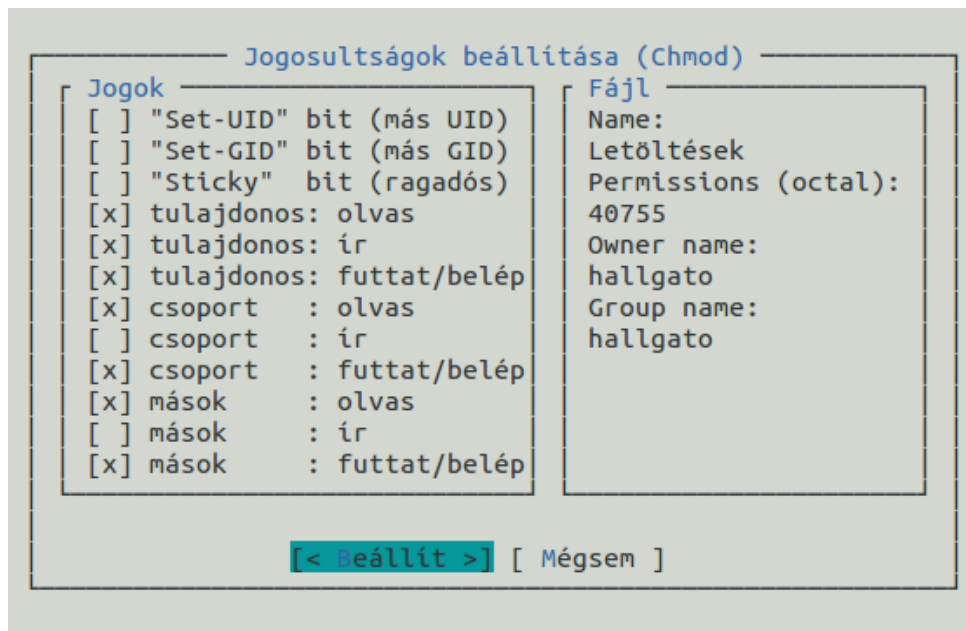
Kényelmesebb munkát biztosíthat számunkra a távoli gép könyvtárának becsatolása a Midnight Commander valamely ablakába. A kapcsolatot a *Bal* vagy *Jobb* legördülő menü *Shell-kapcsolat...* pontjának kiválasztásával, és a 44. ábrán látható párbeszédablak kitöltésével hozhatjuk létre. A csatolás létrehozásának előfeltétele az, hogy a távoli gépen fusson az *mcsvr* nevű program. Természetesen csak olyan könyvtárat tudunk csatolni, amelyhez hozzáférési jogosultságokkal rendelkezünk.



44. ábra. Távoli könyvtár csatolása

### 2.2.3. Állományok és könyvtárak védelmi kódsorának beállítása

A jogosultságok beállításának első lépéseként a sorkurzorral kijelölünk egy állományt vagy könyvtárat, majd a *Fájl* legördülő menüből kiválasztjuk a *Chmod* pontot. A megjelenő párbeszédablakban (45. ábra) a kurzor sort a nyíl billentyűkkel mozgathatjuk, és a jogosultságokat a szóköz billentyűvel állíthatjuk be.

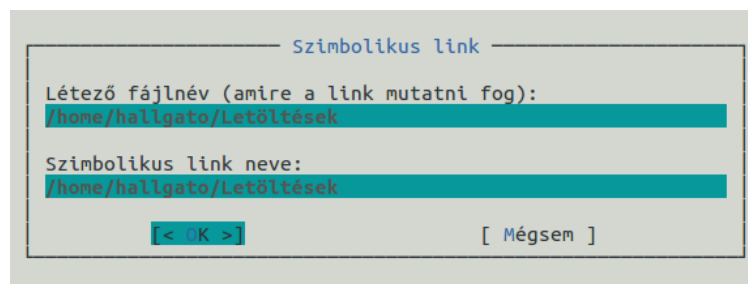


45. ábra. Állományok és könyvtárak védelmi kódsorának beállítása

A párbeszédablak jobb oldali téglalapjában az állomány vagy a könyvtár neve, a teljes védelmi kódsor nyolcas számrendszerben, a tulajdonos neve és a csoportnév szerepelnek.

### 2.2.4. Szimbolikus keresztkapcsolat létrehozása

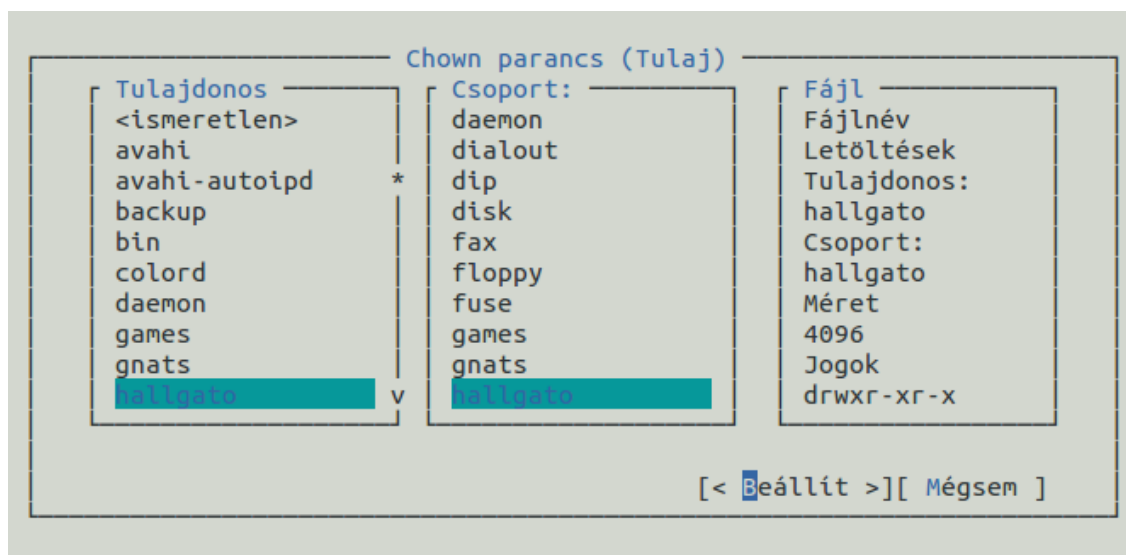
Szimbolikus keresztkapcsolat létrehozásához először ki kell jelölni azt az állományt, amire egy új helyen és/vagy új néven hivatkozni kívánunk. Ezután kiválasztjuk a *Fájl* legördülő menüből a *Szimb. link* pontot, és megjelenik a 46. ábrán látható párbeszédablak. Első sora tartalmazza az eredeti állomány elérési útvonalát és nevét, második sorában adhatjuk meg, hogy hol és milyen néven hozunk létre egy rá vonatkozó hivatkozást.



46. ábra. Szimbolikus keresztkapcsolat létrehozása

## 2.2.5. Tulajdonos és csoport

A tulajdonosra és a csoportra vonatkozó információkat állíthatjuk be a `chown` parancshoz hasonlóan a *Fájl* menü *Chown* pontja segítségével. A párbeszédpanel (47. ábra) első ablakában a kurzorsorral jelölhetjük ki az új tulajdonos személyét, az adott gépen létező felhasználók listájáról. A csoport meghatározása az új tulajdonos beállításával azonos módon történik. Az ablakok között a nyíl billentyűvel mozoghatunk. A *Fájl* címkéjű ablakban látható az éppen érvényes beállítás, az állomány mérete és a védelmi kódsor szimbolikusan megadva.



47. ábra. Tulajdonos és csoport beállítása

## 2.4 Vonatkozó irodalomjegyzék

1. Ubuntu documentation  
<https://help.ubuntu.com/community/UsingTheTerminal>
2. UNIX Manuals  
<http://www.unix-manuals.com/refs/vi-ref/vi-ref.htm>
3. Midnight Commander Development Center  
<https://www.midnight-commander.org/>
4. Nano editor weboldal  
<http://www.nano-editor.org/>

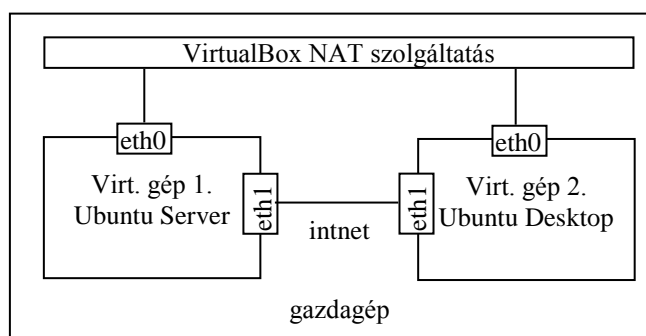


### 3. Hálózati beállítások lekérdezése és módosítása (Johanyák Zsolt Csaba)

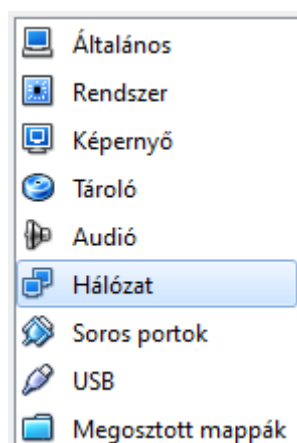
Ebben a fejezetben megismerkedünk a TCP/IP konfiguráció beállítási lehetőségeivel a szerver és a munkaállomás esetében. A szervernél a beállítások karakteres felületen történnek a megfelelő konfigurációs állományok illetve parancssori utasítások segítségével. A munkaállomásnál a beállításokat grafikus felületen fogjuk elvégezni az `nm_applet` segítségével.

#### 3.1. Előkészítés

Mivel a szerveren csak az 10. fejezetben hozzuk létre a NAT kiszolgálót, ezért most az Internet elérése érdekében mindkét virtuális gépet úgy konfiguráljuk, hogy két hálózati interfésszel rendelkezzen. Az első (`eth0`) a VirtualBox által nyújtott NAT szolgáltatáson keresztül kapcsolódjon a külvilág felé, míg a második (`eth1`) az `intnet` nevű belső hálózatra csatlakozzon (48. ábra). Az `intnet` belső hálózat célja az lesz, hogy a két virtuális gép közvetlenül kapcsolódhasson egymáshoz.



48. ábra. Virtuális gépek és hálózati interfészeik

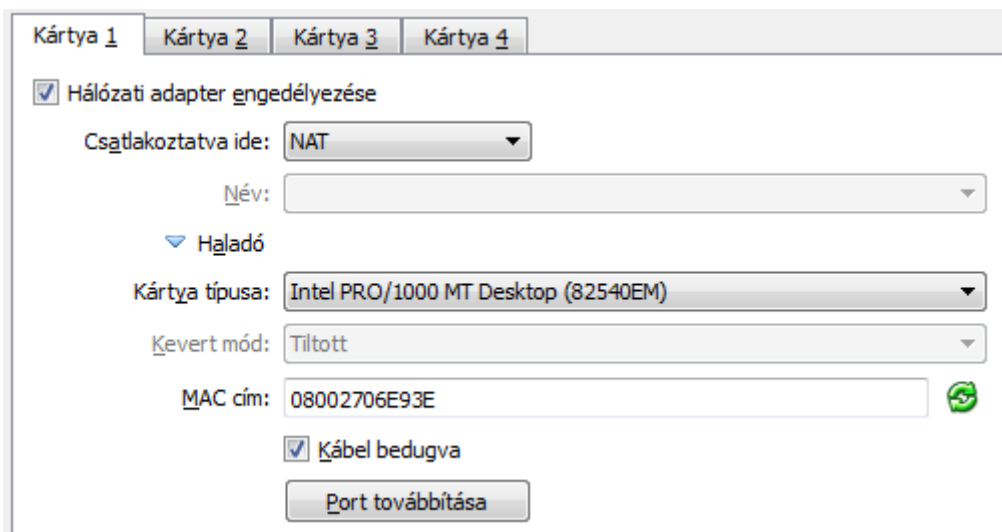


49. ábra. Virtuális gép beállítási kategóriák

A két hálózati interfész engedélyezését a virtuális gépek elindítása előtt kell megtennünk. Ehhez a virtuális gép kiválasztása után a Gép/Konfigurálás... menüponton kattintunk, majd a Beállítások ablak bal oldali listájában (49. ábra) a Hálózat elemet kiválasztjuk, majd a jobb

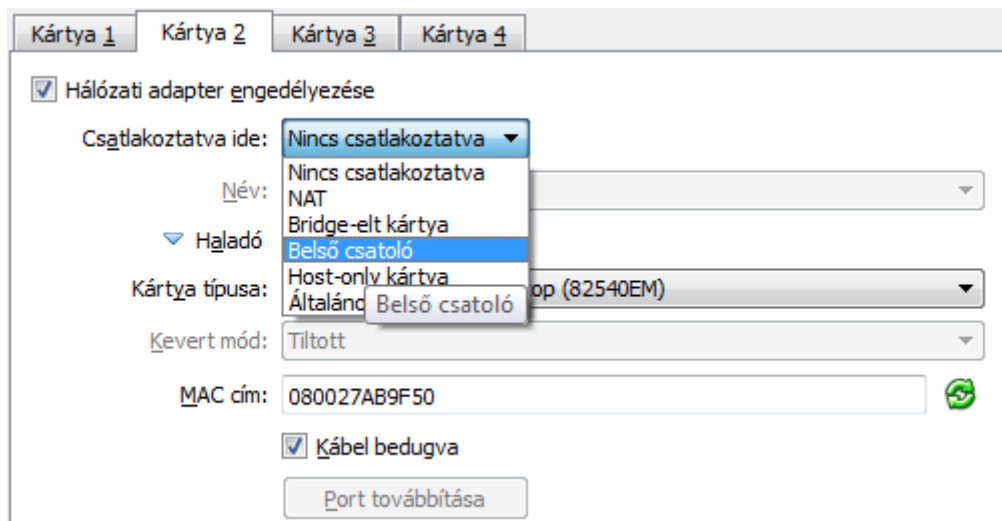


oldali részben az első fülnél engedélyezzük a hálózati adaptert a Kártya 1 és a Kártya 2 füleken. A Haladó címke előtti háromszögre kattintva (50. ábra) válik láthatóvá egyéb beállítások mellett a fizikai (MAC) cím, amire szükségünk lesz a későbbiekben a logikai interfész név beállításá/ellenőrzése során. Az első kártya alapból engedélyezett állapotú, és a VirtualBox NAT szolgáltatásához kapcsolódik. Az itt megjelenő beállításokat változtatlanul hagyjuk.



50. ábra. Virtuális gép első hálózati interfészének beállítása (NAT)

A Kártya 2 fülön a legördülő listából válasszuk ki a „Belső csatoló”-t (51. ábra). Ekkor a Név mezőben alapértelmezés szerint az *intnet* név jelenik meg. Mivel csak egyetlen belső hálózatunk lesz, ez a név tökéletesen megfelelő számunkra.



51. ábra. Második kártya beállítása (belső csatoló)

A hálózati interfészekre egy `ethx` alakú logikai név segítségével hivatkozhatunk, ahol az `x` helyén egy szám áll. Alap esetben a számozás 0-val kezdődik. A felismert és névvel ellátott interfészek listáját az

```
$ ifconfig -a
```

paranccsal kérdezhetjük le. A gyakorlat során az első interfészre eth0, míg a másodikra eth1-ként fogunk hivatkozni.

Időnként előfordul, hogy a hálózati interfészek neve nem az általunk elképzelt kiosztás szerint alakul, pl. ha hálózati interfészt cseréltünk a gépben vagy egy virtuális gép esetén megváltoztattuk a hálózati interfész típusát és MAC címét. A probléma megoldása érdekében a VirtualBox-ban nézzük meg, hogy az egyes interfészekhez milyen fizikai cím tartozik, majd a Linuxon belül nyissuk meg rendszergazdai jogosultsággal a /etc/udev/rules.d/70-persistent-net.rules állományt, és írjuk át az adott interfészhez tartozó elnevezést.

```
$ sudo nano /etc/udev/rules.d/70-persistent-net.rules
```

Az alábbi mintát követve ellenőrizzük a beállításokat, és szükség esetén módosítsuk azokat úgy, hogy az egyes fizikai címekhez (ATTR{address}) az elvárt azonosítók (NAME) legyenek hozzárendelve.

```
# PCI device 0x8086:0x100e (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="08:00:27:06:e9:3e", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
```

```
# PCI device 0x8086:0x100e (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="08:00:27:ab:9f:50", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
```

A beállítást követően indítsuk újra a virtuális gépet, majd a bejelentkezést követően ellenőrizzük le az interfészek elnevezését.

### 3.2. Beállítás karakteres felületen (szerver)

A karakteres felületen történő beállítás két módon történhet. Az első megoldás nem használ konfigurációs állományokat csak közvetlenül kiadott parancsokat, a beállítások azonban elvesznek egy újraindítást követően. A második tartós megoldást kínál a megfelelő konfigurációs állományok szerkesztésével. Az első akkor lehet előnyös, ha egy adott beállítást csupán tesztelni szeretnénk és nincs rá hosszú távon szükségünk. A továbbiakban mindkét módszert kipróbáljuk az eth1 interfésznél az alábbi konfiguráció megvalósítása érdekében:

IPv4 cím:	192.168.1.254
Hálózati maszk:	255.255.255.0
Alapértelmezett átjáró:	192.168.1.253
DNS kiszolgáló:	10.1.51.23
Névkeresési tartomány:	gamf.hu

Az alapértelmezett átjáró címe ebben az esetben nem lesz valós, mivel 253-as gép nem lesz az *intnet* hálózaton, itt csak a gyakorlás érdekében állítjuk be ezt az értéket.

---

### 3.2.1. Beállítás konfigurációs állományok nélkül

Az IPv4 cím ideiglenes beállítását az `ifconfig` utasítással végezhetjük el:

```
$ sudo ifconfig eth1 192.168.1.254 netmask 255.255.255.0
```

Mivel alapértelmezett átjáróból csak egy lehet, és a DHCP kiszolgáló már beállított egyet az `eth0` interfész konfigurálásakor, ezért a parancssori beállítás kipróbálásának idejére leállítjuk az `eth0` interfészt:

```
$ sudo ifdown eth0
```

Az alapértelmezett átjárót a `route` paranccsal állíthatjuk be a kernel routing táblájában:

```
$ sudo route add default gw 192.168.1.253 eth1
```

Az `add` helyett `del`-t alkalmazva törölhetünk egy korábbi bejegyzést. A teljes táblát írassuk ki a

```
$ route -n
```

paranccsal. A DNS kiszolgáló és a névkeresési tartomány beállítása csak konfigurációs állományon keresztül lehetséges a szerveren, ezért ezeket a következő szakaszban tárgyaljuk. A fentiekben megadott ideiglenes beállításokat nemcsak a virtuális gép újraindításával törölhetjük, hanem a

```
$ sudo ip addr flush eth1
```

parancs segítségével is. A korábban leállított `eth0` interfészt a

```
$ sudo ifup eth0
```

paranccsal indítjuk újra.

### 3.2.2. Beállítás konfigurációs állományokkal (tartós beállítás)

A 3.2. fejezet elején megadott beállítások megvalósításához két konfigurációs állomány módosítása szükséges. Az IPv4 cím és az alapértelmezett átjáró megadásához nyissuk meg az `/etc/network/interfaces` konfigurációs állományt:

```
$ sudo nano /etc/network/interfaces
```

Alapból a visszacsatolási (loopback) és a VirtualBox-hoz kapcsolódó (`eth0`) interfész beállításai jelennek meg:

```
auto lo
iface lo inet loopback
```

```
auto eth0
```

```
iface eth0 inet dhcp
```

Mivel az `eth0` a VirtualBox-ba implementált DHCP kiszolgálótól kapja a beállításokat, ezért a rá vonatkozó automatikusan generált két sort nem módosítjuk. A visszacsatolási interfész (`lo`) alapbeállításai szintén megfelelőek. Az `eth1` vonatkozásában nem jelent meg semmi a konfigurációs állományban, ezért itt kézzel kell beírunk a vonatkozó statikus beállításokat.

```
auto eth1
iface eth1 inet static
address 192.168.1.254
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.253
```

Mentsük el a konfigurációs állományt (`Ctrl+O`), és zárjuk be a szerkesztőt (`Ctrl+X`). A beállítások nem érvényesülnek automatikusan, ezért vagy az érintett interfész leállítása/újraengedélyezése (`ifdown/ifup`) vagy a hálózati alrendszer újraindítása szükséges. A dinamikusan (DHCP) és statikusan megadott alapértelmezett átjáróbeállítások közötti konfliktus elkerülése érdekében a fenti beállítások érvényesítése előtt először állítsuk le az `eth0` interfészt.

```
$ sudo ifdown eth0
```

Ennek elmaradása esetén az `eth1` újraengedélyezését követően a konfliktust „RTNETLINK answers: File exist” hibaüzenetet jelzi. Mivel most csak a különböző beállítási módokat szeretnénk kipróbálni, ezért az `eth0` ideiglenes kikapcsolásával nem veszünk semmit. A beállítások érvényesítése érdekében az `eth1` interfészt ki és bekapcsoljuk:

```
$ sudo ifdown eth1 && sudo ifup eth1
```

Ellenőrizzük le a beállítás sikerességét `ifconfig`-gal. Egy éles beállítás elkészítésekor a többszörös alapértelmezett átjáró megadás okozta konfliktus elkerülése érdekében amennyiben ragaszkodunk a statikusan megadott alapértelmezett átjáróhoz, akkor be kell állítanunk, hogy a DHCP kiszolgáló ne adjon alapértelmezett átjáró címet.

A DNS szerver és a keresési tartomány aktuális beállítását az `/etc/resolv.conf` konfigurációs állományban tekinthetjük meg.

```
$ nano /etc/resolv.conf
```

Most a DHCP kiszolgálótól kapott adatok jelennek itt meg.

```
nameserver 10.1.51.23
nameserver 10.1.51.25
domain gamf.hu
search gamf.hu
```

---

A nameserver kulcsszóval kezdődő sorból több is lehet, ezek mindegyike egy DNS névkiszolgáló IP címét tartalmazza. A harmadik sor megadja a tartományt (domént) ahova gépünk tartozik, a negyedik sorban a `search` kulcsszó után egy vagy több keresési tartománynév jelenik meg egymástól szóközzel elválasztva. Ennek az a szerepe, hogy amikor a felhasználó nem a teljes FQDN (Fully Qualified Domain Name) nevet adja meg, hanem annak csak az első tagját (pl. `www.gamf.hu` helyett csak `www-t`) a gépünkön futó resolver a névhez hozzáilleszti a `gamf.hu` utótagot, és az így kapott névre próbálja meg végrehajtani a névfeloldást.

A fentiekben szereplő beállításokat úgy szeretnénk módosítani, hogy a `kefo.hu` is bekerüljön a keresési tartományok közé. Ezt megtehetjük a `resolv.conf` állomány átírásával is, azonban a hatás ideiglenes lesz csupán. A rendszer újraindítását vagy akár egy DHCP kliens futtatást követően az állomány tartalma újra a régi lesz. Tartós beállítást az előzőekben megismert `interfaces` állomány használata biztosít. Ebben az `eth1` interfészre vonatkozó részt kiegészítjük a következő két sorral

```
dns-search gamf.hu kefo.hu
dns-nameservers 10.1.51.23 10.1.51.25
```

majd elmentjük a konfigurációs állományt. A beállítások érvényesítése érdekében kapcsoljuk ki és be az `eth1` interfészt:

```
$ sudo ifdown eth1 && sudo ifup eth1
```

A beállítások ellenőrzéseként megpingeljük az egyik névszerverünket, majd kezdeményezzük a névfeloldást a `kefodok` és az `ubuntu.com` gépek esetén:

```
$ ping 10.1.51.23
$ host kefodok
$ nslookup ubuntu.com
```

A `kefodok` esetében a resolver először a `kefodok.gamf.hu`-val próbálkozik, majd annak sikertelenségét követően a `kefodok.kefo.hu`-val. Bár a jelen esetben nem okozott problémát, de meg kell említenünk, hogy a `resolv.conf`-ot előállító `resolvconf` valójában összefésülte a DHCP-vel kapott és a statikusan beállított keresési tartomány és névkiszolgáló adatokat. Amennyiben el szeretnénk kerülni a DHCP kiszolgáló által nyújtott névszerver adat felhasználását, akkor a DHCP kliensünk konfigurációs állományában (`/etc/dhcp/dhclient.conf`) el kell helyeznünk a

```
supersede domain-name-servers 127.0.0.1
```

sort, ami azt jelenti, hogy a dinamikusan kapott kiszolgáló helyett a megadott IP című gépet használjuk névszerverként.

### 3.3. Beállítás grafikus felületen (asztali gép)


Grafikus felületen a Network Manager (nm-applet) kezeli a hálózati beállításokat. Feladatunk az alábbi beállítások megadása:

eth0 interfész

Konfiguráció fogadása DHCP kiszolgálótól

eth1 interfész

IPv4 cím:	192.168.1.2
Hálózati maszk:	255.255.255.0
Alapértelmezett átjáró:	192.168.1.253
DNS kiszolgáló:	10.1.51.23, 10.1.51.25
Névkeresési tartomány:	gamf.hu, kefo.hu

A konfiguráláshoz a felső sávon a  ikonon, majd válasszuk ki a *Kapcsolatok szerkesztése ...* menüpontot. Alternatív lehetőségként a Dash kezdőoldal keresőmezőjében a hálózat kulcsszót megadva a megjelenő listából kiválasztjuk a Hálózati kapcsolatok programot (52. ábra).

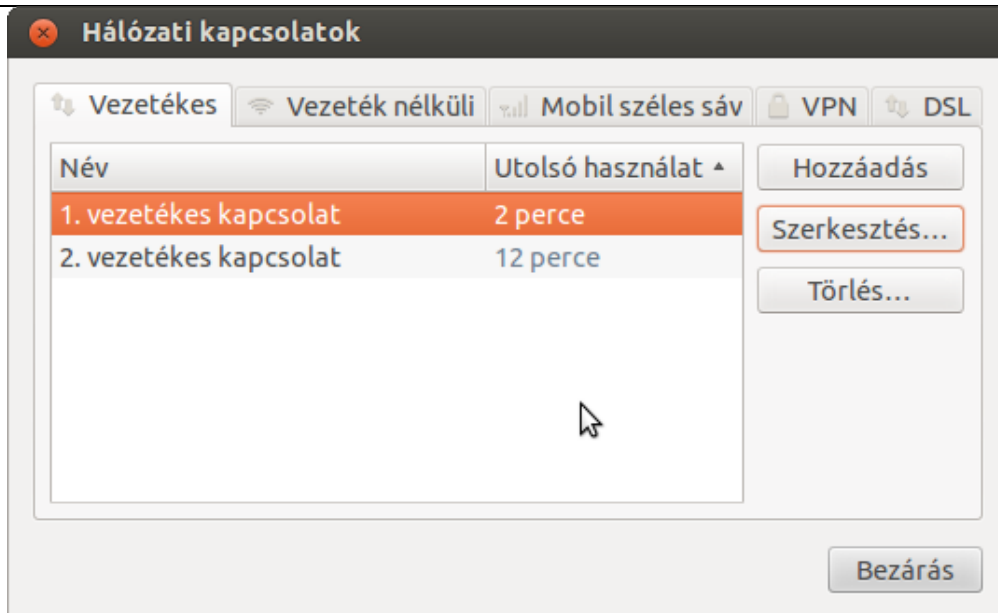


52. ábra. Hálózati kapcsolatok segédprogram indítása Dash kezdőoldalról

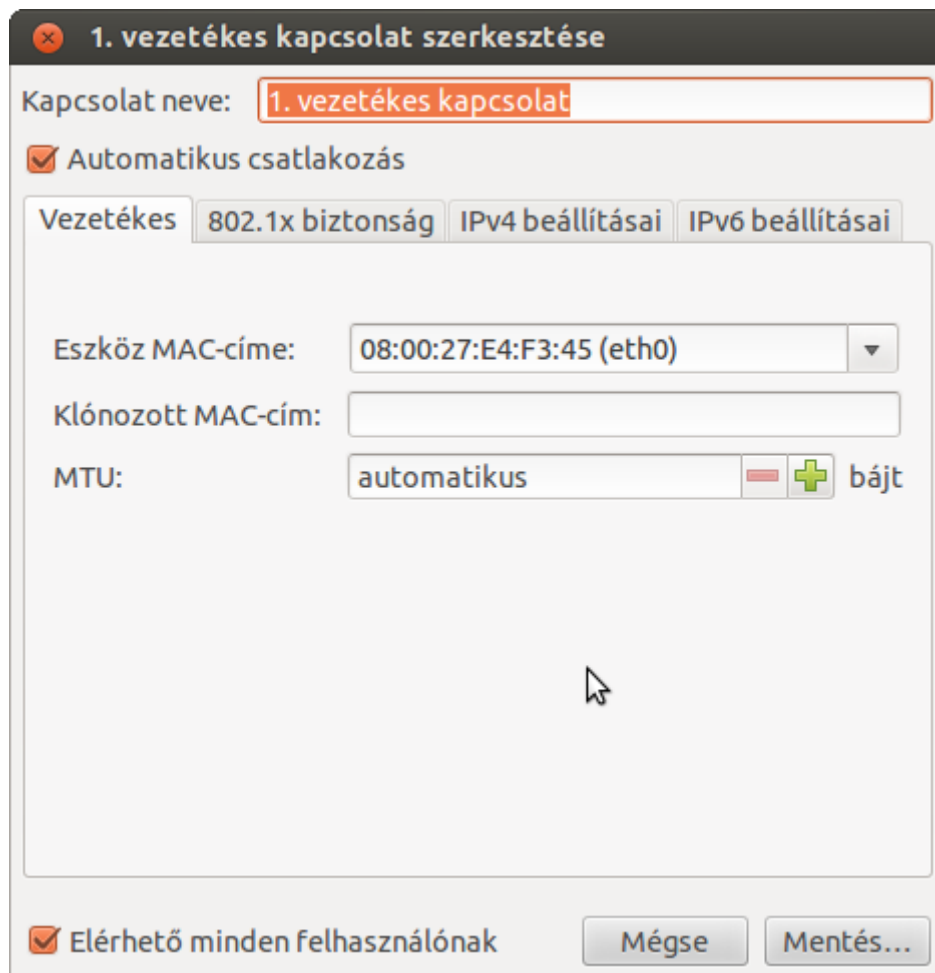
A beállító program grafikus konzolról is indítható a

```
$ nm-connection-editor
```

paranccsal. Elsőként kiválasztjuk a *Vezetékes* fület (53. ábra). Itt kiválasztjuk az 1. vezetékes kapcsolat sort (ez a NAT-os interfészt azonosítja), majd kattintunk a *Szerkesztés ...* gombon. Egy többfüles párbeszédpanel jelenik meg, aminek az első füle (*Vezetékes*, 54. ábra) az interfész fizikai címének megtekintését illetve beállítását teszi lehetővé. Az IPv4 szerinti TCP/IP konfigurációt a harmadik fülön állíthatjuk be (55. ábra). Itt alából a *Módszer* részben *Automatikus* (DHCP) beállításkérést találunk, ami megfelel céljainknak. A *Mentés* gombon kattintva írhatjuk elő a beállítások eltárolását.

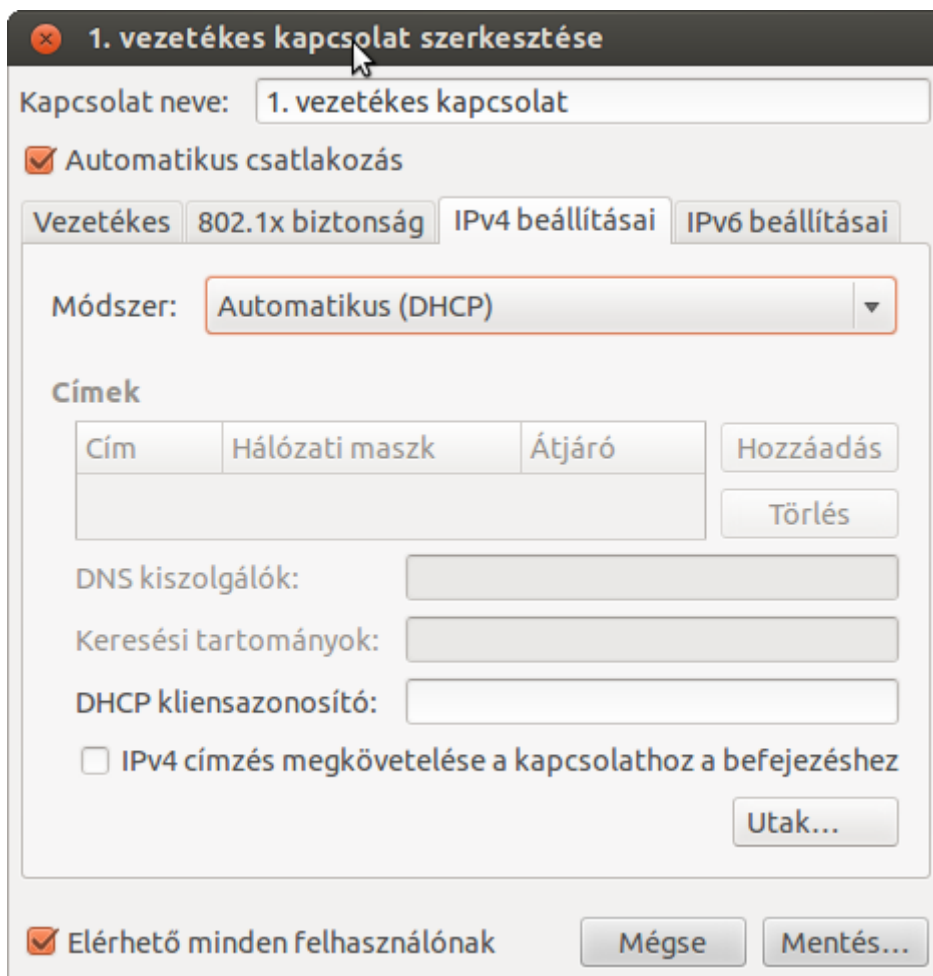


53. ábra. Vezetékes kapcsolatok fül



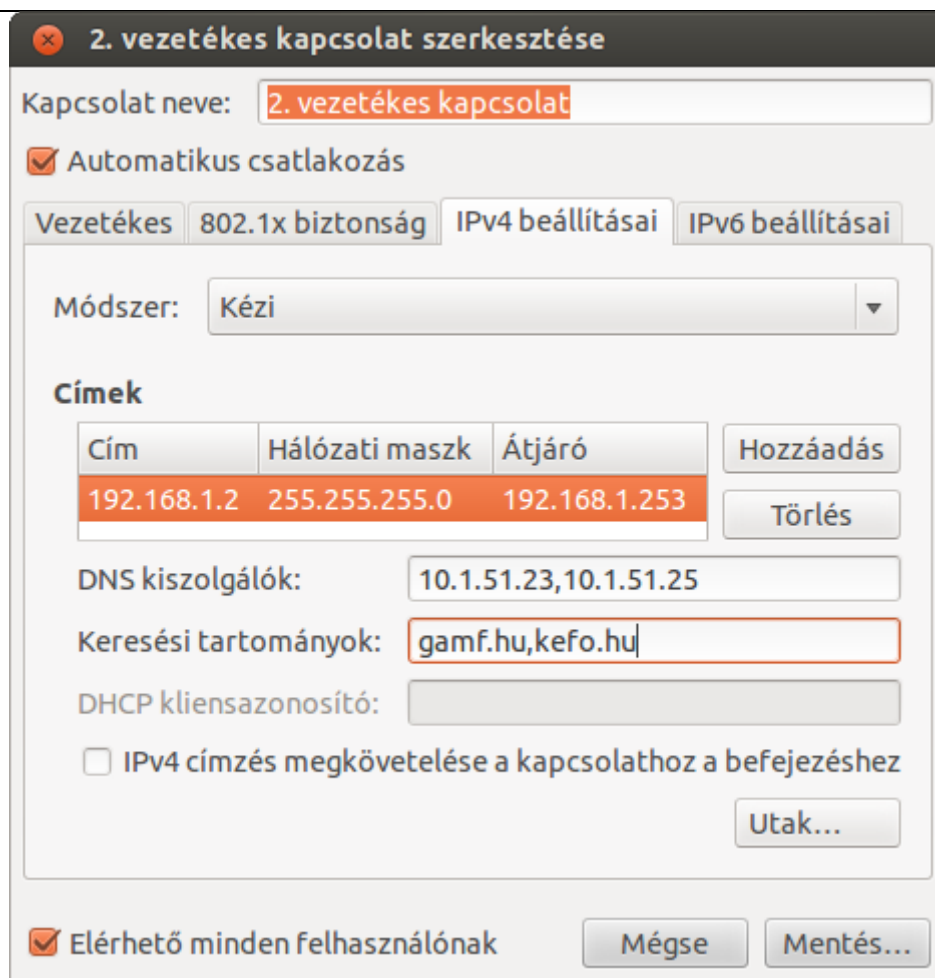
54. ábra. Interfész fizikai címének megtekintése/beállítása

A fentiekhez hasonló módon járunk el az `eth1` interfész esetén is, azzal az eltéréssel, hogy itt a 2. vezetékes kapcsolatot választjuk (53. ábra), majd az IPv4 beállításai fülön a Módszer rész legördülő listájából a *Kézi* listaelemet választjuk. Ezt követően a Címek csoportban kattintunk a Hozzáadás gombon, és a megnyíló mezőkbe begépeljük a fentiekben felsorolt konfigurációs adatokat (56. ábra), majd kattintunk a Mentés és a bezárás gombokon.




55. ábra. IPv4 beállításai (`eth0`)





56. ábra. IPv4 beállításai (eth1)

A beállítások tényleges érvényesítéséhez kattintsunk bal egérgombbal a felső tálcán a  ikonon. A legördülő menüben válasszuk ki a *2. vezetékes kapcsolat* menüpontot. A beállított konfigurációt a `nmcli` program az `/etc/NetworkManager/system-connections/2. vezetékes kapcsolat` állományban tárolja.

Az Ubuntu 12.04 Desktop változatában a használható névszerverek listáját tartalmazó `/etc/resolv.conf` állományban mindig megjelenik egy

```
nameserver 127.0.0.1
```

bejegyzés, aminek köszönhetően a grafikus felületen megadottaktól függetlenül a resolver a `dnsmasq` segítségével próbálja megoldani a névfeloldást. Ennek elkerülése érdekében az `/etc/NetworkManager/NetworkManager.conf` állományban tegyük megjegyzésbe (#) a

```
dns=dnsmasq
```

sort, majd indítsuk újra a NetworkManager szolgáltatást parancssorból.

```
$ sudo service network-manager restart
```

A beállítások ellenőrzéséhez

```
$ ifconfig
```

paranccsal kérdezzük le az aktuális beállítást, majd a `ping` parancs segítségével próbáljuk ki mindkét kapcsolatot. Elsőként a szerver virtuális gépet kívánjuk elérni a belső hálózaton keresztül:

```
$ ping 192.168.1.254
```

majd az egyik névszerverünkkel próbálkozunk a VirtualBox-on keresztül:

```
$ ping 10.1.51.25
```

A névfeloldás működését a `www` (`www.gamf.hu`) és a `kefoposta` (`kefoposta.kefo.hu`) gépek IPv4 címének lekérdezésével ellenőrizzük:

```
$ host www
$ nslookup kefoposta
```

### 3.4. Gépnév beállítása

A gépnév beállítását elsőként a szerver esetében próbáljuk ki. A gép nevének (9-szerver) beállítása érdekében először nyissuk meg szerkesztésre az `/etc/hostname` állományt, és írjuk be az új nevet:

```
$ sudo nano /etc/hostname
```

```
9-szerver
```

Ezt követően nyissuk meg hasonlóképpen az `/etc/hosts` állományt, és állítsuk be az új nevet:

```
$ sudo nano /etc/hosts
```

```
127.0.1.1 9-szerver
```

Győződjünk meg róla, hogy az alábbi beállítás is létezzen, majd mentjük el az állományt.

```
127.0.0.1 localhost
```

Indítsuk újra a gépet.

```
$ sudo reboot
```

Gyakorlásként a fentiekhez hasonlóan állítsuk be az asztali gép nevét `Belzebub-ra`.

### 3.5. Vonatkozó irodalomjegyzék

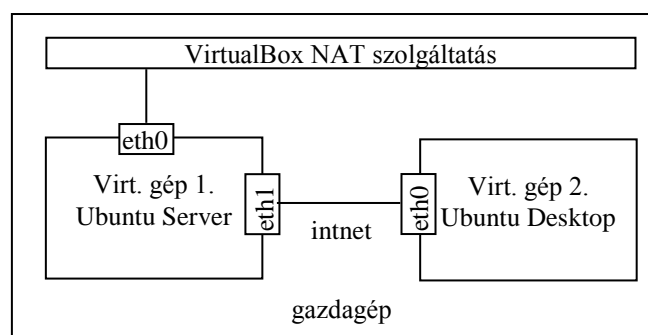
1. Ubuntu Server Guide – Network Configuration  
<https://help.ubuntu.com/12.04/serverguide/network-configuration.html>
2. Ubuntu Server Guide – TCP/IP  
<https://help.ubuntu.com/12.04/serverguide/tcpip.html>
3. interfaces man page  
<http://manpages.ubuntu.com/manpages/precise/en/man5/interfaces.5.html>
4. ifconfig man page  
<http://manpages.ubuntu.com/manpages/precise/en/man8/ifconfig.8.html>
5. ip man page  
<http://manpages.ubuntu.com/manpages/precise/en/man8/ip.8.html>

## 4. DNS szerver telepítése és beállítása (Johanyák Zsolt Csaba)

A fejezet célja az, hogy megismerjük és kipróbáljuk egy elsődleges mester névkiszolgáló konfigurálását. Ennek érdekében a *gamf.hu* domén alatt egy gyakorlat nevű aldomént (*gyakorlat.gamf.hu*) hozunk létre a szerver virtuális gépen (*9-szerver*), majd a szerverről és a munkaállomásról igénybe vesszük a frissen konfigurált névfeloldási szolgáltatást. A konfigurálás nem lesz teljes, ugyanis a felettes - a *gamf.hu* doménért felelős - névkiszolgálóban hozzáférés hiányában nem delegálhatjuk a gyakorlat zónát szerverünkhöz, így csak a belső hálózatunkra csatlakozó két gépről lesz elérhető ez a szolgáltatás.

### 4.1. Előkészítés

A gyakorlat során elsősorban a szerver virtuális gépet használjuk, a munkaállomásra csak az ellenőrzésnél lesz szükség. A szerver virtuális gépet úgy konfiguráljuk, hogy két hálózati interfésszel rendelkezzen. Az első (*eth0*) a VirtualBox által nyújtott NAT szolgáltatáson keresztül kapcsolódjon a külvilág felé, míg a második (*eth1*) az *intnet* nevű belső hálózatra csatlakozzon (57. ábra). Az *intnet* belső hálózat célja az lesz, hogy a két virtuális gép közvetlenül kapcsolódhasson egymáshoz. A munkaállomás (Ubuntu Desktop) számára elegendő egy hálózati kártya (*eth0*), ami a belső hálózati gépre csatlakozik.



57. ábra. Virtuális gépek és hálózati interfészeik

A két gép TCP/IP konfigurációját úgy alakítjuk ki, hogy mindegyik interfész esetében statikusan állítjuk be az adatokat. A szerver esetében az *eth0* interfész:

IPv4 cím:	10.0.2.15
Hálózati maszk:	255.255.255.0
Alapértelmezett átjáró:	10.0.2.2
Hálózat:	10.0.2.0
Üzenetszórési cím:	10.0.2.255
DNS kiszolgáló:	10.1.51.23
Névkeresési tartományok:	gyakorlat.gamf.hu gamf.hu kefo.hu

A DNS kiszolgáló telepítését és konfigurálását követően itt a helyi gépet (127.0.0.1) fogjuk beállítani DNS szerverként. A szerver *eth1* interfészének beállítása az alábbi lesz:

IPv4 cím:	192.168.1.254
-----------	---------------

---

Hálózati maszk:	255.255.255.0
Hálózat:	192.168.1.0
Üzenetszórási cím:	192.168.1.255

A munkaállomás `eth0` interfészének konfigurációja az alábbi lesz:

IPv4 cím:	192.168.1.2
Hálózati maszk:	255.255.255.0
Hálózat:	192.168.1.0
Üzenetszórási cím:	192.168.1.255
DNS kiszolgáló:	192.168.1.254
Névkeresési tartományok:	gyakorlat.gamf.hu gamf.hu kefo.hu

## 4.2. Telepítés és konfigurálás

A DNS kiszolgálóként a BIND9 (Berkeley Internet Name Domain) szoftvert használjuk. Ehhez a `bind9` és a `dnsutils` csomagokat kell telepítenünk. Elsőként frissítjük gépünkön a csomag adatbázist, majd telepítjük a csomagokat.

```
$ sudo apt-get update
$ sudo apt-get install bind9 dnsutils
```

Telepítés után automatikusan elindul a szerver. Az alap konfiguráció egy csak gyorsítótárazó szervert ad egy egyszerű konfigurálást követően, a jelen gyakorlatban azonban egy elsődleges mestert kell létrehoznunk, ezért leállítjuk a kiszolgálót

```
$ sudo service bind9 stop
```

majd elkezdhetjük a konfigurálást. A konfigurációs állományok az `/etc/bind` könyvtárban vannak. A fontosabb állományok az alábbiak:

<code>named.conf</code>	– itt szerepelnek a további konfigurációs állományok befűzésére vonatkozó utasítások, tartalmát nem változtatjuk.
<code>named.conf.options</code>	– itt állítjuk be a lekérdezés továbbítást, rekurzív lekérdezést, stb.
<code>named.conf.local</code>	– itt definiáljuk a zónákat.
<code>db.root</code>	– gyökérszintű névkiszolgálók listája

Szerverünkkel a következő szolgáltatásokat szeretnénk nyújtani:

- Autoritativ (mérvadó) névfeloldás a saját zóna számára (mester).
- Gyorstárazó (caching) névszolgáltatás a többi domén gépeiről. A gyorsítótárazó szerver más kiszolgálóktól szerzi be az információt, és lokálisan (gyorstárban) tárolja azt.
- Rekurzív lekérdezés a saját alhálózat rezolverei által kezdeményezett névfeloldási kéréseknél. Rekurzív névfeloldásnál a DNS kiszolgáló teljes mértékben megválaszolja keresési kérdést vagy hibaüzenetet ad. Ennek alternatívája az iteratív névfeloldás, amikor a kiszolgáló részleges választ ad (az adott zónához közelebbi felelős DNS kiszolgáló címe). Pl. ha a `www.iit.uni-miskolc.hu` IPv4 címét keressük egy külső hálózat névszervere segítségével, akkor, ha az adott kiszolgáló rekurzívra van konfigurálva, akkor visszaküldi rezolverünknek a célgép címét, egyébként pedig rezolverünk visszakapja az uni-

miskolc.hu névkiszolgálójának címét. Ez utóbbi esetben gépünk rezolvere fog további kérést küldeni az uni-miskolc.hu névkiszolgálójához.

Első lépésként a `/etc/bind/named.conf.options` állományban szükséges módosításokat, beállításokat készítjük el. Ehhez nyissuk meg szerkesztésre az állományt.

```
$ sudo nano /etc/bind/named.conf.options
```

A kéréstovábbítás beállításához vegyük ki megjegyzésből a `forwarders` blokkot, és adjuk meg a névszervereink azonosítóit.

```
forwarders{
    10.1.51.23;
    10.1.51.25;
};
```

Itt megadtuk, hogy hova továbbítsa a DNS szerver azokat a névfeloldási kéréseket, amelyeket nem tudott kiszolgálni a saját adatbázisa alapján. Mentsük el az állományt. Amennyiben nem a főiskolai hálózatban hajtjuk végre a jelen gyakorlatot, akkor ez a lépés elhagyható. Ilyenkor az általunk konfigurált névszerver a rekurzív névfeloldási folyamat során közvetlenül a gyökérszintű névkiszolgálóknál kezdi a keresést.

A rekurzív névfeloldás engedélyezése érdekében a `forwarders` blokkot követően helyezzük el a

```
recursion yes;
```

bejegyzést, majd szabályozzuk, hogy mely gépek vehetik igénybe a névszolgáltatást és a rekurzív névfeloldást az alábbi sorokkal

```
allow-query { belso; };
allow-recursion { belso; };
```

A `belso` egy címke, aminek definiálásához az `options` blokkot követően egy külön blokkot hozunk létre, amiben a helyi gépet és a saját alhálózatot nevezzük meg.

```
acl belso {
    127.0.0.1;
    192.168.1.0/24;
};
```

Mivel IPv6 hálózatunk nincs ezért a

```
listen-on-v6 { any; };
```

sort tegyük megjegyzésbe (`//`).

A `gamf.hu` alatt hozzuk létre saját zónánkat `gyakorlat.gamf.hu` néven. A névfeloldási zóna mellett címfeloldási zónát (`1.168.192.in-addr.arpa`) is készítünk.

Mindkettő mester (master) zóna lesz. A zónák deklarálásához nyissuk meg szerkesztésre a `/etc/bind/named.conf.local` állományt.

```
$sudo nano /etc/bind/named.conf.local
```

Az állomány végére begépeljük a két zóna definícióját:

```
zone "gyakorlat.gamf.hu" {
    type master;
    file "/etc/bind/gyakorlat.gamf.hu";
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/1.168.192";
};
```

Elmentjük az állományt, majd létrehozuk először a névfeloldáshoz szükséges zónafájt. Kiindulásként (mintaként) használhatjuk a `db.local` állományt vagy létrehozhatunk egy teljesen üres állományt is a

```
$ sudo nano /etc/bind/gyakorlat.gamf.hu
```

paranccsal. Begépeljük az alábbiakat:

```
$TTL 604800
@      IN      SOA    9-szerver.gyakorlat.gamf.hu.
hallgato.9-szerver.gyakorlat.gamf.hu. (
                                1      ; Sorszám
                                604800 ; Frissítés
                                86400  ; Újrapróbálkozás
                                3600000 ; Lejárát
                                2419200 ) ; Negatív gyorstárazási idő
;
@      IN      NS     9-szerver.gyakorlat.gamf.hu.
9-szerver      IN      A          192.168.1.254
belzebub       IN      A          192.168.1.2
posta          IN      CNAME     9-szerver
```

A fentiekben szereplő második és harmadik sor az állományban egyetlen sorként kell szerepeljen, és a `hallgato` előtt szóköz vagy tabulátor jel kell álljon! Az utolsó sor után nyomjuk meg az Enter billentyűt, ugyanis az állomány végén újsor jel kell álljon.

Magyarázat:

- **TTL** (sec): a zóna rekordjaira érvényes alapértelmezett elavulási idő (Time To Live)
- **SOA** (Start Of Authority) rekord: tartalmazza a névkiszolgáló FQDN-jét (`9-szerver.gyakorlat.gamf.hu.`) ponttal lezárva, a levelezésért felelős felhasználó postafiók címét úgy, hogy a `@` jelet ponttal helyettesítjük (`hallgato.9-szerver.gyakorlat.gamf.hu.`) és az alábbi érvényességi időadatokat.

- **Sorszám:** hányadik változata ez a zónaállománynak, ennek alapján tudják beazonosítani a slave szerverek, hogy történt-e módosítás a zónaállományban.
- **Frissítés (sec):** mennyi időnként kell a slave szervereknek a master-től megkérdezni, hogy a zóna sorszáma mennyi.
- **Újrapróbálkozás (sec):** ha a frissítés nem sikerült, akkor a slave ennyi időt vár, mielőtt újra próbálkozik.
- **Lejárat (sec):** ha nem sikerül a master-rel kommunikálniuk, ennyi ideig szolgáltatják a zónát a világ számára
- Használható a 1W2D3H alak is

Úgy a szerver (9-szerver), mint az asztali (belzebub) operációs rendszerrel futó virtuális gépet felvettük, valamint készítettünk egy álnevet (posta) is a szervernek. Mentsük el a gyakorlat.gamf.hu állományt, majd hozzuk létre az inverz feloldáshoz (címfeloldáshoz) szükséges /etc/bind/1.168.192 zónafájlt. Itt sablonként használhatjuk a db.127 állományt. A jelen gyakorlat során az üres állomány létrehozását választjuk

```
$ sudo nano /etc/bind/1.168.192
```

majd begépeljük az alábbi konfigurációt

```
$TTL 604800
@      IN      SOA  9-szerver.gyakorlat.gamf.hu.
hallgato.9-szerver.gyakorlat.gamf.hu. (
        1      ; Sorszám
        604800 ; Frissítés
        86400  ; Újrapróbálkozás
        2419200 ; Lejárat
        604800 ) ; Negatív gyorstárazási idő
;
@      IN      NS   9-szerver.gyakorlat.gamf.hu.
254    IN      PTR  9-szerver
2      IN      PTR  belzebub
```

A fentiekben szereplő második és harmadik sor az állományban egyetlen sorként kell szerepeljen, és a hallgato előtt szóköz vagy tabulátor jel kell álljon! Az utolsó sor után nyomjuk meg az Enter billentyűt, ugyanis az állomány végén újsor jel kell álljon. Mentsük el a 1.168.192 állományt.

A zónafájlok létrehozását követően a helyi gépet (127.0.0.1) kell beállítanunk DNS szerverként. Ennek érdekében az /etc/network/interfaces állományban a

```
dns-nameservers 10.1.51.23
```

sort lecseréljük az alábbi sorra

```
dns-nameservers 127.0.0.1
```

majd újraindítjuk a hálózati alrendszert



```
$ sudo /etc/init.d/networking restart
```

A DNS kiszolgáló alpból IPv6-on próbálkozik a névfeloldással. Mivel csak IPv4 hálózatunk van a szerver indítási opciói között ezt jeleznünk kell az `/etc/defaults/bind9` állományban. Nyissuk meg szerkesztésre az állományt:

```
$ sudo /etc/defaults/bind9
```

és egészítsük ki az `OPTIONS` sort a `-4` kapcsolóval az alábbiakban megfelelően:

```
OPTIONS="-u bind -4"
```

### 4.3. Tesztelés

A névkiszolgálót az előtérben futtatva teszteljük:

```
$ sudo named -g -4
```

Amennyiben a megjelenő egyoldalas outputban `rndc.key: permission denied` hibaüzenet jelenik meg, akkor gondoskodjunk arról, hogy az `rndc.key` állomány tulajdonosa legyen a `root` felhasználó és csoportja legyen a `bind` csoport, valamint a hozzáférési engedélyek legyenek a következők szerint beállítva: `- rw- -r- ---`. Amikor a kiszolgáló hibamentesen elindul, akkor váltsunk át egy új terminálra, jelentkezünk ott be, majd teszteljük a szerver működését az név- és címfeloldási lekérdezésekkel úgy a helyi zóna gépei, mint távoli gépek esetén.

```
$ host 9-szerver.gyakorlat.gamf.hu
$ host belzebug
$ host posta
$ host 192.168.1.2
$ host 192.168.1.254
$ nslookup ubuntu.hu
$ nslookup kefodok
```

Teszteljük a szerver működését a munkaállomás gépről is a fenti vagy hasonló lekérdezésekkel. Amennyiben kiszolgálónk hibamentesen működik, akkor váltsunk arra a terminálra, ahol elindítottuk a névszervert, majd állítsuk le azt `Ctrl+C`-vel. Végül indítsuk a szolgáltatást tartós használatra a

```
$ sudo service bind9 start
```

paranccsal. Indítsuk újra a virtuális gépet, majd a bejelentkezést követően ellenőrizzük a kiszolgáló működési állapotát pl. a

```
$ service bind9 status
```

paranccsal.

#### **4.4. Vonatkozó irodalomjegyzék**

1. Ubuntu Server Guide – Domain Name Service  
<https://help.ubuntu.com/12.04/serverguide/dns.html>
2. Domain Name System – Wikipédia  
[http://hu.wikipedia.org/wiki/Domain\\_Name\\_System](http://hu.wikipedia.org/wiki/Domain_Name_System)

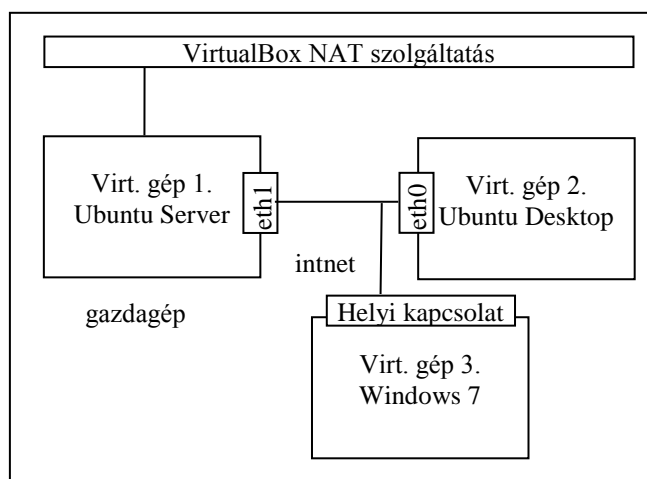


## 5. DHCP szervertelapítése és konfigurálása (Johanyák Zsolt Csaba)

A fejezet célja az, hogy megismerjük és kipróbáljuk a DHCP kiszolgáló konfigurálását és használatát egy lokális hálózatban. A DHCP szervert (Ubuntu Server) az *intnet* belső hálózaton fog a kliens gépek (Ubuntu Desktop és Windows 7) számára konfigurációs adatokat szolgáltatni.

### 5.1. Előkészítés

A szervert virtuális gépet úgy konfiguráljuk, hogy két hálózati interfésszel rendelkezzen. Az első (*eth0*) a VirtualBox által nyújtott NAT szolgáltatáson keresztül kapcsolódjon a külvilág felé, míg a második (*eth1*) az *intnet* nevű belső hálózatra csatlakozzon (58. ábra). Az *intnet* belső hálózat célja az lesz, hogy a két virtuális gép közvetlenül kapcsolódhasson egymáshoz. A munkaállomások (Ubuntu Desktop és Windows 7) számára elegendő egy hálózati kártya (*eth0* ill. Helyi kapcsolat), ami a belső hálózati gépre csatlakozik.



58. ábra. Virtuális gépek és hálózati interfészeik

A szervert gép (9-szervert) TCP/IP konfigurációját úgy alakítjuk ki, hogy mindegyik interfész esetében statikusan állítjuk be az adatokat. Ez az *eth0* interfész esetében az alábbi:

IPv4 cím:	10.0.2.15
Hálózati maszk:	255.255.255.0
Alapértelmezett átjáró:	10.0.2.2
Hálózat:	10.0.2.0
Üzenetszórás cím:	10.0.2.255
DNS kiszolgáló:	10.1.51.23, 10.1.51.25
Névkeresési tartományok:	gamf.hu kefo.hu

A szervert *eth1* interfészének beállítása az alábbi lesz:

IPv4 cím:	192.168.1.254
-----------	---------------

---

Hálózati maszk:	255.255.255.0
Hálózat:	192.168.1.0
Üzenetszórási cím:	192.168.1.255

Az Ubuntu Desktop munkaállomás (neve legyen belzebub) eth0 interfészét úgy állítjuk be hogy DHCP-vel fogadja a beállításokat. A Windows 7 munkaállomás Helyi kapcsolat interfészét úgy állítjuk be hogy DHCP-vel fogadja a beállításokat.

## 5.2. Telepítés és konfigurálás

Az alábbi beállításokat a szerver virtuális gépen kell végrehajtani. Elsőként frissítjük gépünkön a csomag adatbázist, majd telepítjük a kiszolgálót.

```
$ sudo apt-get update
$ sudo apt-get install isc-dhcp-server
```

A telepítő megkísérli elindítani a kiszolgálót, de az a megfelelő konfigurálás hiányában nem fog működni. Kiszolgálónktól az alábbi konfigurációs adatok nyújtását várjuk el (zárójelben a konfigurációnál használt kulcsszó szerepel):

- Kiszolgálónk felelős a szolgáltatással megcélzott alhálózatért (authoritative).
- Nem támogatott a dinamikus DNS frissítés (ddns-update-style).
- A DNS tartomány: gamf.hu (option domain-name).
- Két DNS kiszolgálónk címe: 10.1.51.23, 10.1.51.25 (option domain-name-servers).
- Az üzenetszórási cím a kiszolgált alhálózatban: 192.168.1.255 (option broadcast-address).
- Az alapértelmezett átjáró: 192.168.1.254 (option routers).
- Az alhálózati maszk: 255.255.255.0 (option subnet-mask).
- Az alapértelmezett bérleti idő: 10 perc (default-lease-time).
- Maximális idő, amíg használható a konfiguráció: 2 óra (max-lease-time).
- Rögzített 192.168.1.2 IPv4 cím kiosztása a belzebub gép számára (host).
- Dinamikus IPv4 címkiosztás a 192.168.1.10-250 tartományból a többi ügyfél (jelen esetben a Windows 7) számára (subnet).
- DHCP szolgáltatás nyújtása az eth1 interfészen.

A DHCP kiszolgáló konfigurálásához nyissuk meg szerkesztésre a /etc/dhcp/dhcpd.conf állományt:

```
$ sudo nano /etc/dhcp/dhcpd.conf
```

Az állomány tartalmát töröljük, majd írjuk be az alábbiakat:

```
authoritative;
ddns-update-style none;

option domain-name "gamf.hu";
option domain-name-servers 10.1.51.23, 10.1.51.25;
option broadcast-address 192.168.1.255;
```

```
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;
```

```
default-lease-time 600; # 10 perc
max-lease-time 7200; # 2 óra
```

A rögzített IPv4 cím beállításához nézzük meg az asztali Ubuntu operációs rendszert futtató virtuális gép belső hálózatra kapcsolódó interfészének fizikai címét VirtualBoxban. Az alábbi példában a 08:00:27:e4:f3:45 fizikai címet feltételezzük.

```
host belzebub
{ hardware ethernet 08:00:27:e4:f3:45;
  fixed-address 192.168.1.5;
  option host-name belzebub;
}
```

A dinamikus IPv4 cím kiosztáshoz egy címtartományt (192.168.1.10 192.168.1.250) definiálunk.

```
subnet 192.168.1.0 netmask 255.255.255.0{
  range 192.168.1.10 192.168.1.250;
}
```

Következő lépésként be szeretnénk állítani, hogy melyik interfészen nyújtson DHCP szolgáltatást a szerver. Ehhez nyissuk meg szerkesztésre a /etc/default/isc-dhcp-server hcp-server állományt, és módosítsuk tartalmát az alábbiak szerint:

```
$ sudo nano /etc/default/isc-dhcp-server
```

```
INTERFACES="eth1"
```

Mentsük el az állományt, majd indítsuk el a szolgáltatást.

```
$ sudo service isc-dhcp-server start
```

A szervert konfiguráció próbájaként először az asztali gépen (belzebub) állítsuk be, hogy fogadja DHCP-vel a IPv4 konfigurációt, majd parancssorból ellenőrizzük a beállítások meglétét. Amennyiben nem jelenik meg a kívánt cím azonnal, akkor futtassuk a DHCP kliens programot:

```
$ sudo dhclient
```

A próba második lépéseként állítsuk be VirtualBox-ban, hogy a Windows 7-es gép hálózati kártyája a belső hálózatra (*intnet*) csatlakozzon, majd indítsuk el a Windows 7-es gépet. Ellenőrizzük le, hogy megkapja-e a beállításokat a Linuxos DHCP kiszolgálótól. A szerver a /var/lib/dhcp/dhcpd.leases állományban tartja nyilván a „bérletbe” kiadott konfigurációs adatokat. Tekintsük meg az állomány tartalmát

```
$ more /var/lib/dhcp/dhcpd.leases
```

A helyes konfigurálást követően a szerver indításakor a DHCP szolgáltatás automatikusan el kell induljon. Ennek ellenőrzése érdekében indítsuk újra a szerver virtuális gépet, majd ellenőrizzük le a klienseken, hogy megkapják-e a konfigurációt.

### 5.3. Vonatkozó irodalomjegyzék

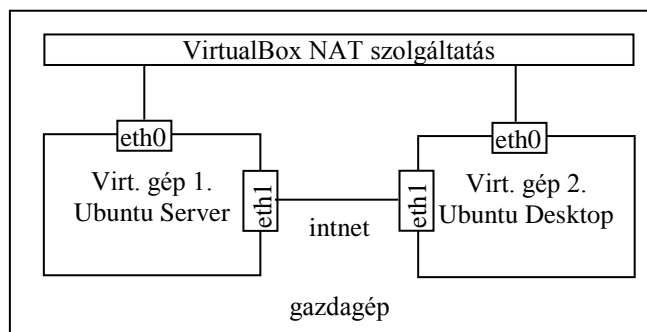
1. Ubuntu Server Guide – Dynamic Host Configuration Protocol  
<https://help.ubuntu.com/12.04/serverguide/dhcp.html>
2. How to Install the DHCP Server on Ubuntu 12.04LTS  
<http://rbgeek.wordpress.com/2012/04/29/how-to-install-the-dhcp-server-on-ubuntu-12-04lts/>
3. dhcpd.conf man page  
<http://manpages.ubuntu.com/manpages/precise/en/man5/dhcpd.conf.5.html>
4. ISC DHCP  
<http://www.isc.org/software/dhcp>
5. What exact purpose have transitional packages?  
<http://askubuntu.com/questions/20377/what-exact-purpose-have-transitional-packages>

## 6. Megosztás NFS segítségével (Johanyák Zsolt Csaba)

Az NFS (Network File System) segítségével könyvtárakat oszthatunk meg Linux/Unix operációs rendszert futtató gépek között. A megosztás kliens-szerver modellt követi, ahol az erőforrást megosztó gép a szerver, az erőforrást igénybe vevő gép a kliens. Egy gép lehet egyszerre szerver és kliens is, azaz megoszthatja saját könyvtárait és felcsatolhat más gép által megosztott könyvtárakat. A fejezetben áttekintjük úgy a szerver, mint a kliens oldali beállításokat és az ellenőrzés lehetőségét.

### 6.1. Előkészítés

Mivel a szerveren csak az 10. fejezetben hozzuk létre a NAT kiszolgálót, ezért most az Internet elérése érdekében mindkét virtuális gépet úgy konfiguráljuk, hogy két hálózati interfésszel rendelkezzen. Az első (`eth0`) a VirtualBox által nyújtott NAT szolgáltatáson keresztül kapcsolódjon a külvilág felé, míg a második (`eth1`) az *intnet* nevű belső hálózatra csatlakozzon (59. ábra). Az *intnet* belső hálózat célja az lesz, hogy a két virtuális gép közvetlenül kapcsolódhasson egymáshoz.



59. ábra. Virtuális gépek és hálózati interfészeik

A két gép TCP/IP konfigurációját úgy alakítjuk ki, hogy mindegyik interfész esetében statikusan állítjuk be az adatokat. A szerver esetében az `eth0` interfész:

IPv4 cím:	10.0.2.15
Hálózati maszk:	255.255.255.0
Alapértelmezett átjáró:	10.0.2.2
Hálózat:	10.0.2.0
Üzenetszórési cím:	10.0.2.255
DNS kiszolgáló:	10.1.51.23
Névkeresési tartományok:	gyakorlat.gamf.hu gamf.hu kefo.hu

A szerver `eth1` interfészének beállítása az alábbi lesz:

IPv4 cím:	192.168.1.254
Hálózati maszk:	255.255.255.0
Hálózat:	192.168.1.0
Üzenetszórési cím:	192.168.1.255



A munkaállomás `eth0` interfészének konfigurációja megegyezik a szerver `eth0` interfészének konfigurációjával. A munkaállomás `eth1` interfészének konfigurációja az alábbi lesz:

```
IPv4 cím:           192.168.1.2
Hálózati maszk:    255.255.255.0
Hálózat:           192.168.1.0
Üzenetszórási cím: 192.168.1.255
DNS kiszolgáló:    192.168.1.254
Névkeresési tartományok: gyakorlat.gamf.hu gamf.hu kefo.hu
```

## 6.2. NFS szerver telepítése és beállítása

Az alábbi beállításokat a szerver virtuális gépen kell végrehajtani. Frissítsük a csomag adatbázist, majd telepítsük az NFS szolgáltatáshoz szükséges programcsomagokat:

```
$ sudo apt-get update
$ sudo apt-get install nfs-kernel-server nfs-common portmap -y
```

Tegyük fel, hogy a `/home/megosztas` könyvtárat szeretnénk megosztani. Először hozzuk létre a könyvtárat, majd állítsuk be, hogy bárki olvashassa, írhatta vagy futtathasson benne.

```
$ sudo mkdir /home/megosztas
$ sudo chmod 777 /home/megosztas
```

A megosztott könyvtárakat az `/etc/exports` konfigurációs állományban kell felsorolnunk. Nyissuk meg szerkesztésre az állományt.

```
$ sudo nano /etc/exports
```

Az állományban minden megosztáshoz egy sor tartozik. A sor a megosztani kívánt könyvtár teljes elérési útvonalával kezdődik, ez a mi esetünkben `/home/megosztas`. A könyvtárat teljes hozzáféréssel (írható/olvasható) szeretnénk megosztani a `192.168.1.0` alhálózat összes gépe számára. A megosztást leíró sor a következő:

```
/home/megosztas
192.168.1.0/24(rw, sync, root_squash, no_subtree_check)
```

A fenti konfiguráció valójában egy sor, csak az oldalszélesség korlát miatt jelenik meg fentebb két sorban. A hálózati IPv4 cím helyett egy csillagot megadva az összes számítógép számára megoszthatjuk könyvtárunkat. A megosztás kedvezményezettje lehet egyetlen gép is, ilyenkor az adott gép IP címét vagy nevét kell itt megadnunk. Figyeljünk oda arra, hogy a nyitó zárójel előtt nem állhat szóköz és minden könyvtármegosztás külön sorban kell álljon. Mintaként nézzünk meg néhány példát:

```
/home 192.168.1.1/255.255.255.0(rw)
/segedlet belzebub(rw) pandora(ro)
```

```
/ubuntu *(ro, sync, no_root_squash)
```

A zárójelben megadott jellemzőkkel szabályozhatjuk a könyvtárhoz történő hozzáférést. Jelentésük a következő.

ro	csak olvasható
rw	olvasható és írható
root_squash	a kliens root felhasználója semmiképp nem kaphat root jogokat erre a fájlrendszerre
sync	a szerver szinkron módon hajtja végre a változtatásokat (csak a végrehajtás után jelez vissza)
link_absolute	a szimbolikus hivatkozások változatlanok maradnak
subtree_check	a kérés beérkezése után a szerver leellenőrzi, hogy a cél a fájlrendszeren belül van-e illetve az exportált könyvtárstruktúrában található-e – biztonsági probléma: a kliens kap egy leíró és infót a fájlrendszerről, ezért csak ro könyvtárakra használjuk!
no_subtree_check	rw könyvtárakra

Mentsük el a konfigurációs állományt, és lépünk ki a szövegszerkesztő programból. Indítsuk újra az NFS kiszolgáló programot.

```
$ sudo /etc/init.d/nfs-kernel-server restart
$ sudo exportfs -a -v
```

Az `exportfs` parancs segítségével karbantartható a közzétett (exportált) könyvtárak táblázata. A `-a` parancs hatására a konfigurációs állományban megadott összes állományt exportáljuk, míg a `-v` kapcsoló hatására részletes információt kapunk a parancs eredményéről

### 6.3. NFS kliens telepítése és beállítása

Az alábbi beállításokat a kliens virtuális gépen kell végrehajtani. Frissítsük a csomag adatbázist, majd telepítsük fel az NFS megosztás igénybe vételéhez szükséges programcsomagokat:

```
$ sudo apt-get update
$ sudo apt-get install nfs-common portmap
```

A megosztás igénybevétele, azaz a megosztott könyvtár használata úgy lehetséges, hogy a kliens gép könyvtárrendszerében egy könyvtárhoz felcsatoljuk a szerver által megosztott könyvtárat. Ezt követően a felhasználó számára a távoli könyvtár ugyanúgy jelenik meg és ugyanúgy használható, mint egy helyi könyvtár.

Hozzuk létre a könyvtárfában azt a mappát, ahova fel kívánjuk csatolni a kiszolgáló által megosztott könyvtárat.

```
$mkdir /home/hallgato/megosztas
```

Következő lépésként felcsatoljuk (importáljuk) a kiszolgáló által megosztott könyvtárat:

```
$ sudo mount -t nfs 192.168.1.254:/home/megosztas  
/home/hallgato/megosztas
```

A fenti konfiguráció valójában egy sor, csak az oldalszélesség korlát miatt jelenik meg fentebb két sorban.

A kliens gép leállításakor a felcsatolás megszűnik. Amennyiben azt szeretnénk, hogy minden indításkor automatikusan csatolódjon fel a könyvtár, akkor az `/etc/fstab` állományba egy új sort kell írunk. Ehhez nyissuk meg az állományt.

```
$ sudo nano /etc/fstab
```

Helyezzük el a következő sort (egyetlen sorba írva, és a sor végén az Enter-t lenyomva):

```
192.168.1.254:/home/megosztas /home/hallgato/megosztas nfs  
rw,hard,intr 0 0
```

Mentsük el az állományt, majd próbáljuk ki a beállítást.

```
$ sudo mount /home/hallgato/megosztas
```

## 6.4. Tesztelés

Hozzunk létre a kliens gépen egy állományt a `megosztas` könyvtárban, írjunk bele valamilyen szöveget. Ellenőrizzük le a kiszolgálón az állomány meglétét.

Lépünk ki a könyvtárból, majd csatoljuk azt le.

```
$ sudo umount /home/hallgato/megosztas
```

Lépünk be a felcsatolt könyvtárba, ellenőrizzük a csatolást, ellenőrizzük, hogy megtalálható-e az előzőleg létrehozott állomány. Az állomány természetesen nincs ott, hiszen az a szerveren található.

Indítsuk újra a virtuális gépet az automatikus felcsatolás ellenőrzése érdekében.

## 6.5. Vonatkozó irodalomjegyzék

1. Ubuntu Server Guide – Network File System (NFS)  
<https://help.ubuntu.com/12.04/serverguide/network-file-system.html>
2. Network File System – Wikipedia  
[http://hu.wikipedia.org/wiki/Network\\_File\\_System](http://hu.wikipedia.org/wiki/Network_File_System)
3. Network File System – Wikipedia  
[http://en.wikipedia.org/wiki/Network\\_File\\_System](http://en.wikipedia.org/wiki/Network_File_System)
4. Network File System (NFS) version 4 Protocol (RFC 3530)  
<http://tools.ietf.org/html/rfc3530>
5. nfsv4 - NFS Version 4 Protocol – Ubuntu manpage  
<http://manpages.ubuntu.com/manpages/precise/man4/nfsv4.4freebsd.html>

6. exportfs - maintain table of exported NFS file systems – Ubuntu manpage  
<http://manpages.ubuntu.com/manpages/precise/en/man8/exportfs.8.html>

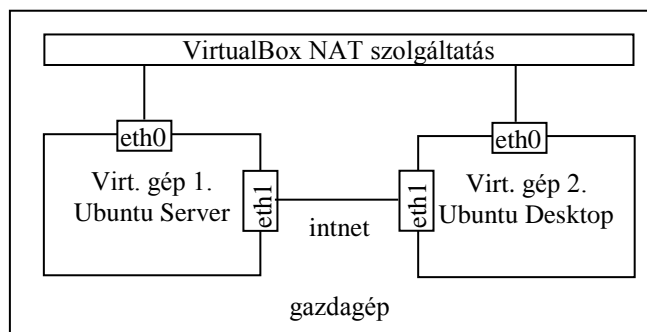


## 7. Megosztás Samba segítségével (Johanyák Zsolt Csaba)

A Samba segítségével könyvtárakat, nyomtatókat oszthatunk meg Linux és Windows operációs rendszert futtató gépek között. A megosztás kliens-szerver modellt követi, ahol az erőforrást megosztó gép a szerver, az erőforrást igénybe vevő gép a kliens. Egy gép lehet egyszerre szerver és kliens is, azaz megoszthatja saját könyvtárait és felcsatolhat más gép által megosztott könyvtárakat. A fejezetben áttekintjük úgy a szerver, mint a kliens oldali beállításokat és az ellenőrzés lehetőségét a könyvtármegosztás feladatán keresztül.

### 7.1. Előkészítés

Mivel a szerveren csak az 10. fejezetben hozzuk létre a NAT kiszolgálót, ezért most az Internet elérése érdekében mindkét virtuális gépet úgy konfiguráljuk, hogy két hálózati interfésszel rendelkezzen. Az első (`eth0`) a VirtualBox által nyújtott NAT szolgáltatáson keresztül kapcsolódjon a külvilág felé, míg a második (`eth1`) az `intnet` nevű belső hálózatra csatlakozzon (60. ábra). Az `intnet` belső hálózat célja az lesz, hogy a két virtuális gép közvetlenül kapcsolódhasson egymáshoz.



60. ábra. Virtuális gépek és hálózati interfészeik

A két gép TCP/IP konfigurációját úgy alakítjuk ki, hogy mindegyik interfész esetében statikusan állítjuk be az adatokat. A szerver esetében az `eth0` interfész:

IPv4 cím:	10.0.2.15
Hálózati maszk:	255.255.255.0
Alapértelmezett átjáró:	10.0.2.2
Hálózat:	10.0.2.0
Üzenetszórési cím:	10.0.2.255
DNS kiszolgáló:	10.1.51.23
Névkeresési tartományok:	gyakorlat.gamf.hu gamf.hu kefo.hu

A szerver `eth1` interfészének beállítása az alábbi lesz:

IPv4 cím:	192.168.1.254
Hálózati maszk:	255.255.255.0
Hálózat:	192.168.1.0
Üzenetszórési cím:	192.168.1.255

A munkaállomás `eth0` interfészének konfigurációja megegyezik a szerver `eth0` interfészének konfigurációjával. A munkaállomás `eth1` interfészének konfigurációja az alábbi lesz:

```
IPv4 cím:           192.168.1.2
Hálózati maszk:    255.255.255.0
Hálózat:           192.168.1.0
Üzenetszórási cím: 192.168.1.255
DNS kiszolgáló:    192.168.1.254
Névkeresési tartományok: gyakorlat.gamf.hu gamf.hu kefo.hu
```

## 7.2. Samba kiszolgáló telepítése

Hozzuk létre a megosztani kívánt könyvtárat, és állítsuk be, hogy a `hallgato` felhasználó legyen a tulajdonos és a csoporttulajdonos legyen a `hallgato` csoport.

```
$ sudo mkdir /home/smbmegosztas
$ sudo chown hallgato /home/smbmegosztas
$ sudo chgrp hallgato /home/smbmegosztas
```

Megj.: A Filesystem Hierarchy Standard ajánlása szerint a megosztott könyvtárakat a `/srv/samba` könyvtár alá ajánlott elhelyezni.

A szerver virtuális gépen frissítsük a csomag adatbázist, majd telepítsük a Samba kiszolgálóhoz szükséges csomagokat.

```
$ sudo apt-get update
$ sudo apt-get install samba smbfs
```

Nyissuk meg a konfigurációs állományt.

```
$ sudo nano /etc/samba/smb.conf
```

Az általános, minden megosztás esetén érvényes beállításokkal kezdjük. Ezek a `[global]` szakaszban találhatóak. Elsőként állítsuk be a munkacsoport nevét. Ehhez megkeressük a megfelelő sort (`workgroup`), majd az egyenlőség jel utáni részt megváltoztatjuk.

```
workgroup=GYAKORLAT
```

Az alábbiakban a legtöbb beállítás esetében hasonlóképpen fogunk eljárni. Amennyiben valamelyik kulcs (pl. `netbios name`) nem szerepel a kezdeti konfigurációs állományban, akkor a teljes sort begépeljük, egyébként csak az egyenlőség jel utáni részt módosítjuk. Néhány esetben a kulcs (pl. `security`) kezdetben megjegyzésben áll, ilyenkor el kell távolítani a sor elején álló pontosvesszőt.

Második beállításunk a gép NetBIOS neve lesz (ezt be kell gépelni):

```
netbios name=ubuntu-server
```

Ezt követően a megosztások biztonsági modelljét (szintjét) felhasználóira állítjuk. Ez azt jelenti, hogy egyedi felhasználói névvel és jelszóval férhetnek hozzá a megosztott erőforrásokhoz, illetve felhasználónként szabályozhatjuk a jogosultságokat.

```
security=user
```

Bekapcsoljuk a Samba szerver WINS szolgáltatását, ami azt jelenti, hogy szerverünk egy Windows Internet Name Service kiszolgáló feladatait is el fogja látni. A kulcs kezdetben megjegyzésben van.

```
wins support=yes
```

Engedélyezzük, hogy a Samba szerver megkísérelje a megváltoztatott Samba jelszavakat a Linux jelszó adatbázisban is érvényesíteni.

```
unix password sync=yes
```

Az általános beállítások után létrehozunk egy megosztást. A könyvtárat szeretnénk megosztani a hallgato nevű felhasználó számára. A konfigurációs állomány végén létrehozunk egy új szakaszt az új megosztás számára.

```
[smbmegosztas]
```

Adunk hozzá egy rövid magyarázó szöveget.

```
comment=Ez egy megosztás
```

Beállítjuk az elérési útvonalát és a hozzáférés szabályozást. Legyen a könyvtár írható.

```
writeable=yes  
path=/home/smbmegosztas
```

Jelszó nélkül ne lehessen hozzáférni

```
public=no
```

Ne legyen rejtett, jelenjen meg a megosztások listájában

```
browseable=yes
```

A hallgato felhasználó írhatja és olvashatja.

```
read list=hallgato  
write list=hallgato
```

Mentsük el a konfigurációs állományt, majd teszteljük azt.

```
$ sudo testparm
```



A rendszer adatbázisai a `/var/lib/samba` könyvtárban találhatóak  
Vegyük fel a hallgató felhasználót a Samba adatbázisba.

```
$ sudo smbpasswd -a hallgato
```

A jelszó legyen „hallgato”.  
Indítsuk el/újra a szervert.

```
$ sudo service smb restart
```

### 7.3. Samba kliens telepítése és konfigurálása

Hozzunk létre egy `smbkliens` nevű könyvtárat, ide fogjuk felcsatolni a szerver által megosztott mappát.

```
$ mkdir /home/hallgato/sbmeosztas
```

Az asztali Ubuntu operációs rendszert futtató virtuális gépen telepítsük fel a megosztás eléréséhez szükséges csomagokat.

```
$ sudo apt-get update  
$ sudo apt-get install smbfs smbclient
```

Kérdezzük le a kiszolgáló által megosztott könyvtárakat.

```
$ smbclient -L ubuntu-server -N
```

A `Sharename` oszlopban meg kell jelenjen az `sbmeosztas` sor.  
Hajtsuk végre a felcsatolást.

```
$ sudo smbmount //192.168.1.254/sbmeosztas  
/home/hallgato/sbmeosztas -o username=hallgato
```

Lépünk be a könyvtárba, és hozzunk létre ott egy új szöveges állományt. Lépünk ki a könyvtárból, majd csatoljuk azt le.

```
$ sudo smbmount /home/hallgato/sbmeosztas/
```

Ha azt szeretnénk, hogy minden indításkor automatikusan csatolódjon fel a könyvtár, akkor az `/etc/fstab` állományba egy új sort kell írunk. Ehhez nyissuk meg az állományt.

```
$ sudo nano /etc/fstab
```

Helyezzük el a következő sort (egyetlen sorba írva, és a sor végén az Enter-t lenyomva):

```
//192.168.1.254/sbmeosztas /home/hallgato/sbmeosztas  
smbfs username=hallgato,password=hallgato,umask=000 0 0
```

Mentsük el az állományt, majd próbáljuk ki a beállítást.

```
$sudo mount /home/hallgato/sbmeosztas
```

## 7.4. Tesztelés

Lépünk be a felcsatolt könyvtárba, ellenőrizzük a csatolást. Indítsuk újra a virtuális gépet az automatikus felcsatolás ellenőrzése érdekében. Ezután indítsunk el egy virtuális gépet Windows 7 operációs rendszerrel, és csatoljuk fel egy meghajtóként az ubuntu-server által megosztott könyvtárat.

## 7.5. Vonatkozó irodalomjegyzék

1. Ubuntu Server Guide – Samba File Server  
<https://help.ubuntu.com/12.04/serverguide/samba-fileserver.html>
2. Samba  
<http://www.samba.org/>
3. Samba – Wikipédia  
<http://hu.wikipedia.org/wiki/Samba>
4. Filesystem Hierarchy Standard  
<http://www.pathname.com/fhs/pub/fhs-2.3.html>



## 8. WebDAV kiszolgáló konfigurálása (Kovács Péter)

A WebDAV (Web Distributed Authoring and Versioning) a HTTP protokoll kiterjesztéseként lehetővé teszi a fájl és dokumentum szintű együttműködést a felhasználók között egy webkiszolgálón keresztül. Ezt oly módon éri el, hogy a HTTP hét műveletéhez továbbiakat definiál. A HTTP csak megnevezett erőforrások lekérdezését és egy-az-egyben feltöltését teszi lehetővé. A WebDAV a következő témakörökben újít:

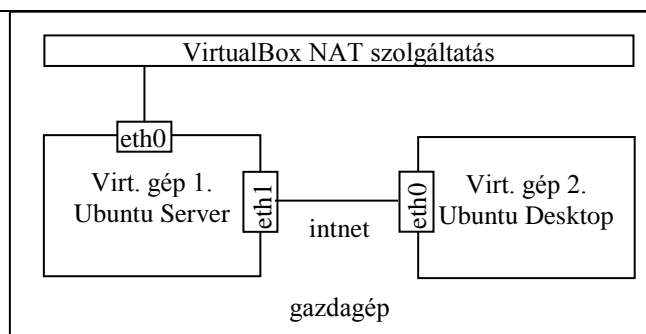
- Szöveges metainformáció (properties) hozzáadása/lekérdezése: PROPFIND, PROPPATCH műveletek, illetve néhány XML-elem a név-érték párok számára.
- Erőforrások gyűjteményekbe (collections) szervezése: gyakorlatilag könyvtárakba szervezett fájlokról van szó. A MKCOL művelet új könyvtárat hoz létre, a COPY és MOVE műveletek másolatot készítenek, illetve átmozgatják a megnevezett erőforrást. Ha meggondoljuk, hogy a HTTP-ben már van GET, PUT és DELETE művelet, akkor FTP-szerverünket máris helyettesíthetjük egy HTTP+WebDAV összeállítással.
- Zárolás (locking): LOCK és UNLOCK műveletek, DAV fejléc, erre a konkurenciakazelés miatt van szükség

Webszervereken lévő mappákat a WebDAV protokoll segítségével megoszthatunk a kliens oldali alkalmazások számára. A megosztást követően ugyanúgy elérhetők lesznek, mintha a helyi gépen lennének: fájlokat lehet feltölteni, letölteni egyetlen egérmozdulattal. Mindezt a HTTP protokollal a 80-as porton keresztül, tetszőlegesen helyileg vagy az Interneten keresztül történhet, ezért gyakorlatilag az összes forgalomszabályozó hálózati eszközön (proxy, tűzfal, útválasztók, stb.) minden különösebb konfiguráció nélkül használható.

A fejezetben megismerkedünk a szerver telepítés, konfigurálás és tesztelés alapjaival, valamint teszteljük a kiszolgáló működését a kliens gépről.

### 8.1. Előkészítés

A gyakorlat során elsősorban a szerver virtuális gépet használjuk, a munkaállomásra csak az ellenőrzésnél lesz szükség. A szerver virtuális gépet úgy konfiguráljuk, hogy két hálózati interfésszel rendelkezzen. Az első (eth0) a VirtualBox által nyújtott NAT szolgáltatáson keresztül kapcsolódjon a külvilág felé, míg a második (eth1) az *intnet* nevű belső hálózatra csatlakozzon (**Hiba! A hivatkozási forrás nem található.** ábra). Az *intnet* belső hálózat célja az lesz, hogy a két virtuális gép közvetlenül kapcsolódhasson egymáshoz. A munkaállomás (Ubuntu Desktop) számára elegendő egy hálózati kártya (eth0), ami a belső hálózati gépre csatlakozik.



62. ábra. Virtuális gépek és hálózati interfészeik

A szerver gép TCP/IP konfigurációját úgy alakítjuk ki, hogy mindegyik interfész esetében statikusan állítjuk be az adatokat. Ez az `eth0` interfész esetében az alábbi:

IPv4 cím:	10.0.2.15
Hálózati maszk:	255.255.255.0
Alapértelmezett átjáró:	10.0.2.2
Hálózat:	10.0.2.0
Üzenetszórési cím:	10.0.2.255
DNS kiszolgáló:	10.1.51.23, 10.1.51.25
Névkeresési tartományok:	gamf.hu kefo.hu

A szerver `eth1` interfészének beállítása az alábbi lesz:

IPv4 cím:	192.168.1.254
Hálózati maszk:	255.255.255.0
Hálózat:	192.168.1.0
Üzenetszórési cím:	192.168.1.255

A munkaállomás `eth0` interfészének konfigurációja az alábbi lesz:

IPv4 cím:	192.168.1.2
Hálózati maszk:	255.255.255.0
Hálózat:	192.168.1.0
Üzenetszórési cím:	192.168.1.255
DNS kiszolgáló:	192.168.1.254
Névkeresési tartományok:	gamf.hu kefo.hu

## 8.2. Telepítés és konfigurálás

Az alábbi műveleteket, mindig rendszergazda parancssorban ( `sudo -s` ) végezzük.

Tiltsuk le a belső csatlós kártyát

```
$ ifconfig eth0 down
```

Frissítsük az elérhető csomagok listáját

```
$ apt-get update
```

Telepítsük az Apache webkiszolgálót

```
$ apt-get install apache2 -y
```

Engedélyezzük a WebDAV modulokat

```
$ a2enmod dav_fs
```

Indítsuk újra az Apache kiszolgálót

```
$ /etc/init.d/apache2 restart
```

Készítsünk egy könyvtárat, melyet majd tárterületként használhatunk

```
$ mkdir -p /var/www/webdav
```

Állítsuk be, hogy a mappa tulajdonosa az Apache felhasználó (www-data) legyen

```
$ chown www-data /var/www/webdav
```

Készítsünk egy biztonsági másolatot az Apache alapértelmezett virtuális host konfigurációs állományáról

```
$ mv /etc/apache2/sites-available/default /etc/apache2/sites-available/default_orig
```

majd készítsük el a saját konfigurációs állományunkat

```
$ nano /etc/apache2/sites-available/default
```

az alábbi tartalommal

```
NameVirtualHost *
<VirtualHost *>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/webdav/
    <Directory /var/www/webdav/>
        Options Indexes MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
```

```
</Directory>
```

```
</VirtualHost>
```

Töltsük újra az Apache beállításokat

```
$ /etc/init.d/apache2 reload
```

### **Állítsuk be a virtual hostunkat a WebDAV –hoz**

Készítsünk egy jelszó fájlt, és adjuk hozzá a `test` felhasználót

```
$ htpasswd -c /var/www/webdav/passwd.dav test
```

A kérdésre kétszer begépeljük ugyanazt a jelszót, ami az egyszerűség kedvéért legyen `test`.  
Módosítsuk a jelszófájl jogait, hogy csak a rendszergazda, és a `www-data` csoport férhessen hozzá

```
$ chown root:www-data /var/www/webdav/passwd.dav  
$ chmod 640 /var/www/webdav/passwd.dav
```

Állítsuk be, hogy a `webdav` mappa kérjen hitelesítést

```
$ nano /etc/apache2/sites-available/default
```

Adjuk a fájl végéhez a `</VirtualHost>` elé az alábbiakat:

```
Alias /webdav /var/www/web1/web  
<Location /webdav>  
    DAV On  
    AuthType Basic  
    AuthName "webdav"  
    AuthUserFile /var/www/web1/passwd.dav  
    Require valid-user  
</Location>
```

Töltsük újra az Apache beállításokat

```
$ /etc/init.d/apache2 reload
```

### **Állítsuk be a virtual hostunkat a WebDAV –hoz**

Készítsünk egy jelszó fájlt, és adjuk hozzá a `test` felhasználót

```
$ htpasswd -c /var/www/webdav/passwd.dav test
```

A kérdésre kétszer begépeljük ugyanazt a jelszót, ami az egyszerűség kedvéért legyen `test`.  
Módosítsuk a jelszófájl jogait, hogy csak a rendszergazda, és a `www-data` csoport férhessen hozzá

```
$ chown root:www-data /var/www/webdav/passwd.dav
$ chmod 640 /var/www/webdav/passwd.dav
```

Állítsuk be, hogy a `webdav` mappa kérjen hitelesítést

```
$ nano /etc/apache2/sites-available/default
Adjuk a fájl végéhez a </VirtualHost> elé az alábbiakat:
```

```
Alias /webdav /var/www/web1/web
<Location /webdav>
    DAV On
    AuthType Basic
    AuthName "webdav"
    AuthUserFile /var/www/web1/passwd.dav
    Require valid-user
</Location>
```

Töltsük újra az Apache beállításokat

```
$ /etc/init.d/apache2 reload
```

### 8.3. Tesztelés

Teszteljük a konfigurációt egy parancssoros WebDAV klienssel a szerver gépen. Ehhez telepítsük fel a `cadaver-t`:

```
$ apt-get install cadaver -y
```

Próbáljuk elérni a helyi WebDAV-ot

```
$ cadaver http://localhost/webdav/
```

Ha minden jól működik, akkor itt hitelesítést kér tőlünk, ahol a **test** – **test** adatokkal juthatunk tovább, és ha sikerült a készenléti jel átvált `$ dav:/webdav/ -re`



Teszteljük a konfigurációt a kliens gépről is. Ezt megtehetnénk a szerver esetében alkalmazott parancssoros módszerrel is, de gyorsabb megoldás, ha az előtelepített Firefox böngészővel teszteljük. Írjuk be a **http://192.168.1.254/webdav/** címet a címsorba, majd üssünk entert. A kért hitelesítő ablakba ismét írjuk a **test – test** név – jelszó párost, és ha sikerrel járunk, láthatjuk a mappa tartalmát a böngészőben.

## 8.4. Vonatkozó irodalomjegyzék

1. Ubuntu documentation  
<https://help.ubuntu.com/community/UsingTheTerminal>
2. HowtoForge – linux dokumentum gyűjtemény  
<http://www.howtoforge.com/>
3. Apache http server honlapja  
<http://httpd.apache.org/>
4. WebDAV Resources  
<http://www.webdav.org/>

## 9. SQUID proxy szerver konfigurálása (Kovács Péter)

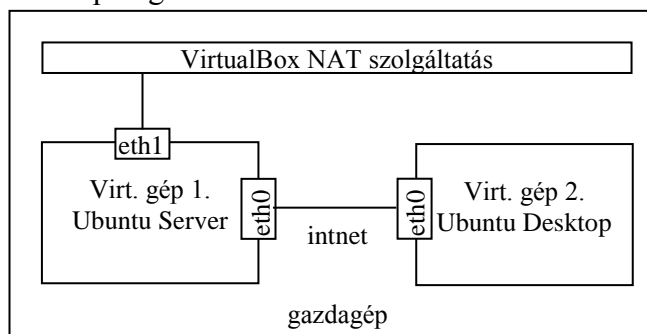
A Squid egy teljes körű szolgáltatásokat nyújtó webes proxy gyorsítótár-kiszolgáló, amely proxy<sup>12</sup> és gyorsítótár-szolgáltatásokat biztosít a HTTP, FTP és más népszerű hálózati protokollokhoz. A Squid képes SSL kérések gyorsítótárazására és proxyzására, DNS-kikeresések gyorsítótárazására és transzparens gyorsítótárazásra. A Squid gyorsítótárazási protokollok széles körét támogatja, például az ICP, HTCP, CARP és WCCP protokollokat.

A Squid proxy gyorsítótár-kiszolgáló kitűnő megoldás rengeteg proxyzási és gyorsítótárazási kiszolgálói igényre, és a telephelyi irodától a vállalati hálózatokig remekül skálázódik, miközben átfogó és részletes hozzáférés-felügyeleti mechanizmusokat, valamint a kritikus paraméterek SNMP feletti figyelését is biztosítja. A dedikált Squid proxy vagy gyorsítótár-kiszolgálóként használandó számítógép kiválasztásakor gondoskodni kell arról, hogy a rendszer nagy mennyiségű fizikai memóriát tartalmazzon, mivel a Squid a jobb teljesítmény érdekében memóriabeli gyorsítótárat tart fenn.

A fejezetben telepítjük, konfiguráljuk, és teszteljük a SQUID szoftvert Ubuntu rendszerek segítségével.

### 9.1. Előkészítés

Ezen gyakorlathoz a két virtuális gépünket egy hálózatba kell kapcsolni, és IP címeiket konfigurálni. Maradjunk a korábbi példákban használtak mellett, és legyen a desktop gép címe 192.168.1.2 a szerveré pedig 192.168.1.254.



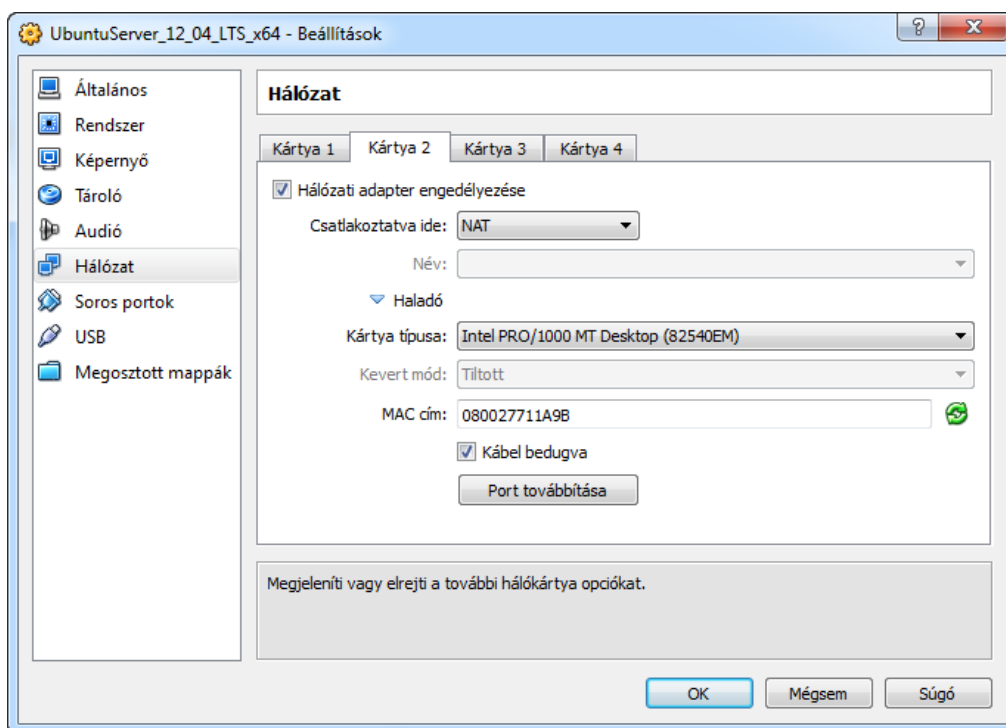
63. ábra. Virtuális gépek és hálózati interfészeik

<sup>12</sup> Számítógép-hálózatokban proxynak, helyesebben proxy szervernek (angol „helyettes”, „megbízott”, „közvetítő”) nevezzük az olyan szervert (számítógép vagy szerveralkalmazás), ami a kliensek kéréseit köztes elemként más szerverekhez továbbítja. A kliens csatlakozik a proxyhoz, valamilyen szolgáltatást (fájlt, csatlakozást, weboldalt vagy más erőforrást) igényel, ami egy másik szerveren található. A proxy szerver a kliens nevében eljárva csatlakozik a megadott szerverhez, és igényli az erőforrást a számára. A proxy esetlegesen megváltoztathatja a kliens kérését vagy a szerver választát, és alkalomadtán kiszolgálhatja a kérést a szerverhez való csatlakozás nélkül is. Az olyan proxy szervernek, ami változtatás nélkül továbbítja a kérélmeket és a válaszokat külön neve is van: ez a gateway, vagy néha tunneling proxy.

Mivel a telepítendő csomagok Interneten találhatóak, szükség van egy olyan hálózati csatlóóra is, amely kilát az Internetre. Ehhez az alábbi lépések szükségesek:

állítsuk le a szerver gépet ( halt )

a VirtualBox kezelőben a szerver gép hálózati konfigurációjánál aktiváljuk be a második hálózati kártyát NAT csatlóval, és olvassuk ki a MAC címét a későbbi konfiguráláshoz.



64. ábra

indítsuk el a szerver gépet, és jelentkezzünk be, majd lépünk rendszergazda parancssorba

```
$ sudo -s
```

az újonnan behelyezett hálózati kártyát konfiguráljuk eth1 névre

```
$ nano /etc/udev/rules.d/70-persistent-net.rules
```

állítsuk be, hogy dinamikusan kapja az IP címet

```
$ nano /etc/network/interfaces
```

az alábbiak szerint:

```
auto eth1
```

```
iface eth1 inet dhcp
```

indítsuk újra a hálózatkezelőt a változások érvényesítéshez

```
$ nano /etc/network/interfaces
```

ellenőrizzük a hálózati beállításokat az

```
$ ifconfig
```

paranccsal, ennek kimenetében három adaptert kell látnunk IP címükkel együtt eth0, eth1, lo

Ezen beállítások után már van két működő hálózati kártyánk `eth0` egy belső hálózatra kapcsolódik, `eth1` pedig a fizikai gépünkön keresztül az Internetre.

A könnyebb követhetőség kedvéért egyszerre csak egy kártyával foglalkozunk, közben a másikat letiltjuk. Telepítésékor az `eth0` kártyát tiltjuk majd le, teszteléskor, vagy mikor a klienssel kívánunk kommunikálni, akkor pedig az `eth0` kártyát.

A tiltást az `$ ifconfig ethX down` az engedélyezést pedig az `$ ifconfig ethX up` parancsokkal végezzük, ahol X a kérdéses kártya száma.

### 9.1.1. Telepítés Ubuntu serverre

Az alábbi műveleteket, mindig rendszergazda parancssorban (`$ sudo -s`) végezzük.

frissítsük a csomaglistát

```
$ apt-get update
```

csomagkezelővel telepítsük a squid csomagot

```
$ apt-get install squid -y
```

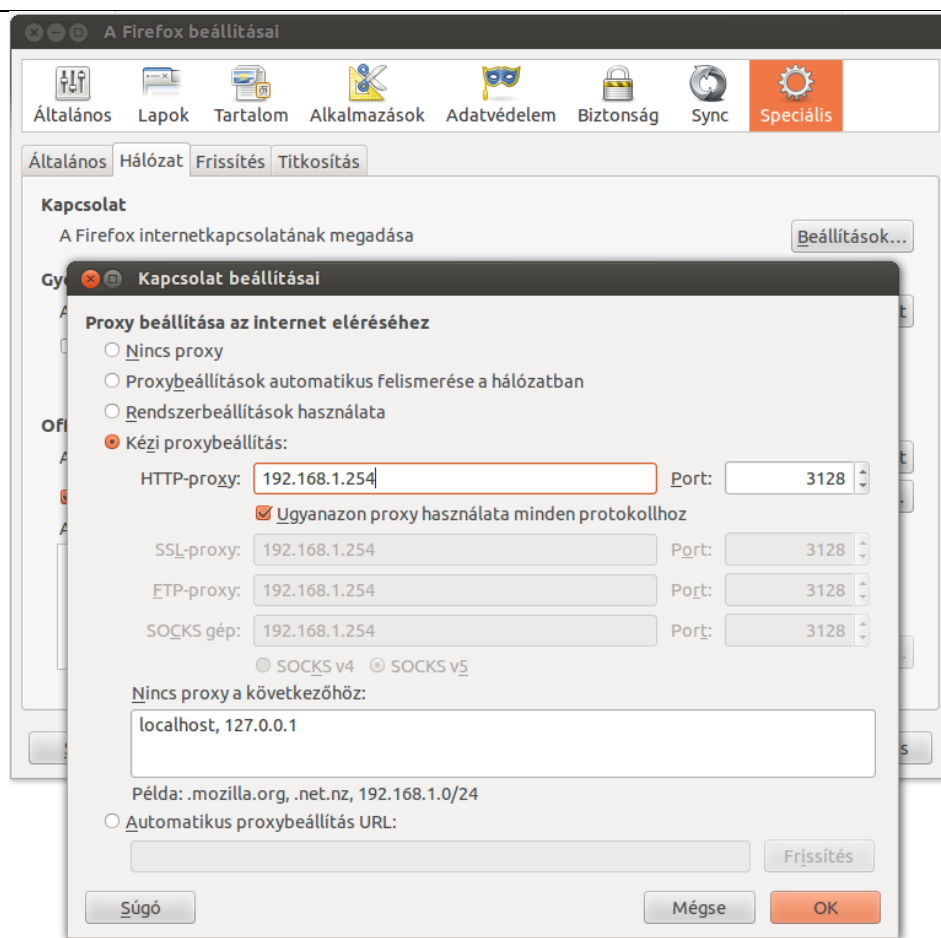
sikerés telepítés után a konfigurálás előtt készítsünk biztonsági másolatot az eredeti fájlról

```
$ cp /etc/squid3/squid.conf /etc/squid3/squid.conf.original
```

ezen a ponton már tesztelhető a beállítás az alábbi lépésekkel engedélyezzük a belső csatlós kártyát

```
$ ifconfig eth0 up
```

a desktop gépen indítsuk el a Firefox böngészőt, majd a Szerkesztés->Beállítások->Haladó ablak Hálózat fülén a Kapcsolat rész alatt kattintsunk a beállítások gombra, és állítsuk be proxykiszolgálónak a szerveret, majd zárjuk be a beállításokat és írjunk be a címsorba egy tetszőleges címet pl.: [www.google.com](http://www.google.com). Erre egy a proxy serverünk által generált megtagadó üzenetet kell kapnunk.



65. ábra.

## 9.2. Telepítés és konfigurálás

Engedélyezzük a hozzáférést a kliensünk számára, a `/etc/squid/squid.conf` fájl szerkesztésével

```
$ nano /etc/squid3/squid.conf
```

keressük meg a fájlban az `acl` kezdetű sorokat, és utánuk egy új sorba adjuk hozzá az alábbi bejegyzést

```
acl internal_network src 192.168.1.0/24
```

majd keressük meg a `http_access` kezdetű sorokat, és ezek elé egy új sorba vegyük fel az alábbi sort

```
http_access allow internal_network
```

Ezen a ponton ismét tesztelhetjük a beállításokat a kliens web böngészőjében, ha mindent jól csináltunk, akkor eltűnik a tiltó üzenet, ám a weboldal nem jelenik meg, helyette egy DNS szerver hibát kapunk.

Itt tulajdonképp véget ért a `squid` konfigurálása, de ahhoz hogy valós internetelérést is lássunk a kliens gépen, a szerveren be kell üzemelni egy címfordítást a szerver két kártyája és ezzel együtt a belső és a külső hálózat közt. Ennek beállításával a 10. fejezetben ismerkedünk meg.

### **9.3. Vonatkozó irodalomjegyzék**

1. Ubuntu documentation  
<https://help.ubuntu.com/>
2. HowtoForge – linux dokumentum gyűjtemény  
<http://www.howtoforge.com/>
3. SQUID Home Page  
<http://www.squid-cache.org/>



## 10. A hálózati címfordítás (NAT) megvalósítása (Göcs László)

A hálózati címfordítás (angolul Network Address Translation, röviden NAT) a csomagszűrő tűzfalak, illetve a címfordításra képes hálózati eszközök (pl. router) kiegészítő szolgáltatása, mely lehetővé teszi a belső hálózatra gépek közvetlen kommunikációját tetszőleges protokollokon keresztül külső gépekkel anélkül, hogy azoknak saját nyilvános IP-címmel kellene rendelkezniük. Címfordításra akár egyetlen számítógép is képes, így valósítható meg például az internet-kapcsolat megosztás is, amikor a megosztó gép a saját publikus címébe fordítja bele a megosztást kihasználó kliens gép forgalmát.

Az egész címfordítás témaköre abból az igényből nőtt ki magát, hogy az IPv4 tartománya viszonylag kevés,  $2^{32} = 256^4$  azaz 4 294 967 296 db egyedi IP címet tesz ki. Ebben persze benne van az összes üzenetszórás (broadcast) cím és a külső hálózatra nem továbbítható (route-olható) belső címtartomány is, tehát az interneten globálisan használható címek összessége így még kevesebb. A gépek hálózati interfészei egynél több címet is kaphatnak egyszerre, ha szükséges, illetve nemcsak a számítógépeknek, hanem szinte az összes fontosabb hálózati eszköznek is szüksége van legalább egy címre. Belátható, hogy így a soknak tűnő 4 milliárd cím világviszonylatban már sajnos kevés.

A hálózati címfordító a belső gépekről érkező csomagokat az internetre továbbítás előtt úgy módosítja, hogy azok feladójaként saját magát tünteti fel, így az azokra érkező válaszcsomagok is hozzá kerülnek majd továbbításra, amiket – a célállomás címének módosítása után – a belső hálózaton elhelyezkedő eredeti feladó részére ad át. Ebből kifolyólag ez minden esetben egy aktív hálózati eszközt igényel, amely folyamatosan figyeli az érkező csomagokat, majd azok feladói és címzettjei alapján elvégzi a szükséges módosításokat. Ez többnyire (de nem szükségszerűen) egy tűzfal, amely megfelelően szétválasztja a külső hálózatot a belsőtől. A belső hálózatnak interfészei olyan címtartományból kell kapjanak azonosítót, amelyet minden hálózati eszköz a nemzetközi szabványoknak megfelelően belsőnek ismer el, és így azokat nem irányítja közvetlenül a külső hálózat felé.

A címfordítás segítségével megoldható, hogy akár egy egész cég teljes belső hálózati forgalma egyetlen külső IP cím mögött legyen, azaz gyakorlatilag egyetlen külső címet használ el egy több száz gépes hálózat. A belső forgalomban természetesen szükség van az egyedi belső címekre, de erről csak a címfordítást végző hálózati eszközöknek kell tudnia, kifelé ennek részletei már nem látható információk. Így létrejöhet olyan gazdaságos konfiguráció is, hogy egy viszonylag nagy cég teljes külső címfoglalása 10-20 db. cím, míg a belső forgalmukban akár több ezer belső cím is lehet.

Nagy előnye ennek a technikának, hogy ugyanazt a belső tartományt nyugodtan használhatja bárki más is; amíg mindegyik egyedi külső cím mögé van fordítva, ez nem okoz zavart. Akár az összes NAT-ot használó cég belső hálózatában lehet minden gép a 10.0.0.0 tartományban, ha kifelé valóban egyedi címmel látszanak. Éppen a címfordítás technológiája miatt nem került gyorsabban bevezetésre az IPv6 szabvány, amely kifejlesztésének egyik oka az IPv4 fogyatkozó címtartományának kiváltása volt.



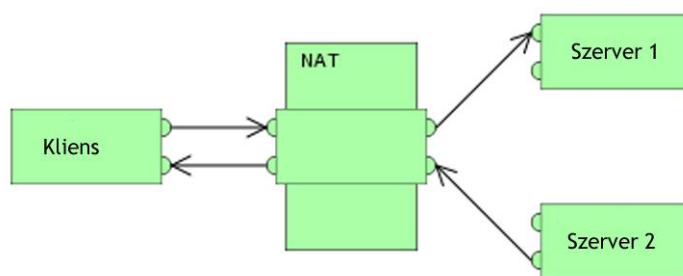
## 10.1. A címfordítás típusai

A címfordítás többféle séma szerint kerülhet megvalósításra, amelyekben a fordítás jellege, illetve a portok és protokollok kezelése tér el egymástól. A címek és a portok előtt álló vastagbetűs **b-** és **k-** előtagok a **belső** és **külső** fogalmakat rövidítik a magyarázatok egyszerűbb átláthatóságának érdekében.

### 10.1.1. Egy az egyben címfordítás (Full cone NAT)

Amikor egy belső cím (**b-Cím:b-Port**) egy külső címre fordul (**k-Cím:k-Port**), bármilyen csomag a belső címről (**b-Cím:b-Port**) a külső címen (**k-Cím:k-Port**) keresztül kerül kiküldésre.

Bármelyik külső gép tud csomagokat küldeni a belső gépnek úgy, hogy a csomagokat a külső címre küldi, amin keresztül fordítás után eljut a belső géphez.

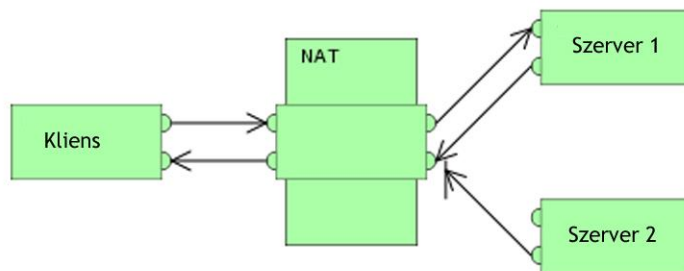


66. ábra. Egy az egyben címfordítás

### 10.1.2. Címhez kötött címfordítás (Address Restricted cone NAT)

Amikor egy belső cím (**b-Cím:b-Port**) egy külső címre fordul (**k-Cím:k-Port**), bármilyen csomag a belső címről (**b-Cím:b-Port**) a külső címen (**k-Cím:k-Port**) keresztül kerül kiküldésre.

Bármelyik külső gép tud csomagokat küldeni a belső gépnek úgy, hogy a csomagokat a külső címre küldi, amin keresztül fordítás után jut el a belső géphez, de csak akkor, ha előzőleg a belső gép küldött csomagot a külső gépnek. A Port-ra nézve itt nincs megkötés.



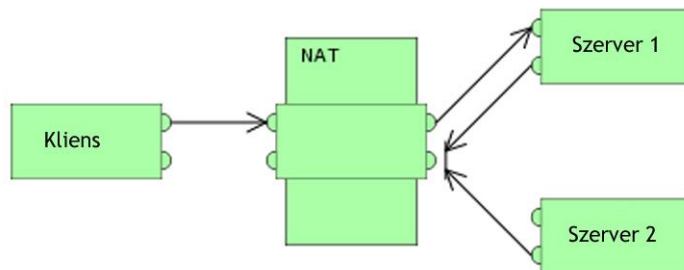
67. ábra. Címhez kötött címfordítás

### 10.1.3. Porthoz és címhez kötött címfordítás (Port-Restricted cone NAT)

Hasonló mint az előző (Address) Restricted cone NAT, de a megkötés a portszámra is vonatkozik.

Amikor egy belső cím (**b-Cím:b-Port**) egy külső címre fordul (**k-Cím:k-Port**), bármilyen csomag a belső címről (**b-Cím:b-Port**) a külső címen (**k-Cím:k-Port**) keresztül kerül kiküldésre.

Bármelyik külső gép tud csomagokat küldeni a belső gépnek úgy, hogy a csomagokat a külső címre küldi, amin keresztül fordítás után jut el a belső géphez, de csak akkor, ha előzőleg a belső gép küldött csomagot a külső gép előzőekben használt címére és portjára.



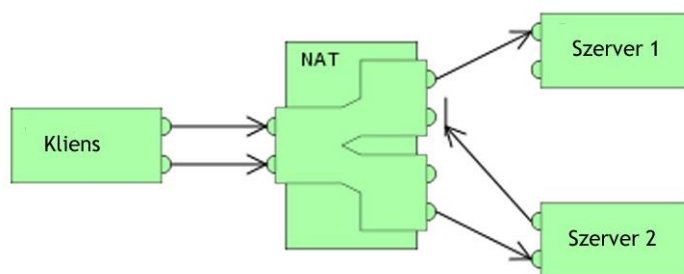
68. ábra. Porthoz és címhez kötött címfordítás

### 10.1.4. Szimmetrikus címfordítás (Symmetric NAT)

Bármilyen kérés egy adott belső (**b-Cím:b-Port**) gépről, amely egy külső (**k-Cím:k-Port**)-ra irányul egy egyedi külső címre és portra fordul, amit a külső gép forrásnak tekint (ahová majd válaszolnia kell, ha el akarja érni a belső gépet).

Ha ugyanaz a belső gép (akár ugyanazzal a belső címmel és porttal) egy másik külső gépnek küld csomagot, az már egy másik, szintén egyedi külső címet kap (ahová majd a másik megcímzett külső gép válaszol, ha el akarja érni a belső gépet)

Gyakorlatilag minden külső gép egy egyedi címen látja (akár ugyanazt) a belső gépet. Csak az a külső gép tud visszaküldeni választ, amelyik előzőleg kapott a belső géptől csomagot.



69. ábra. Szimmetrikus címfordítás

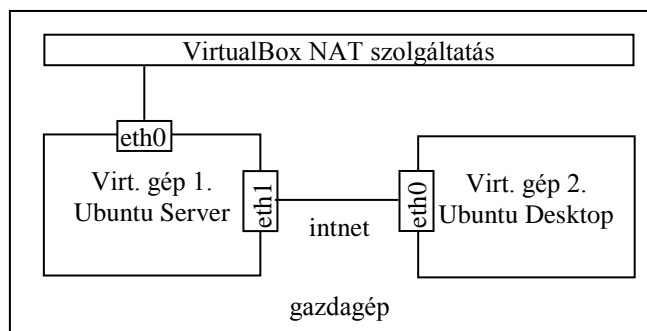
A legtöbb címfordítási megoldás kombinálja egymással az egyes típusokat, ezért jobb az adott esetben jellemző címfordítási viselkedésről, mint a konkrét típusról beszélni. Előfordulhat például olyan eset, amikor két belső gép is ugyanazzal a külső géppel akar kommunikálni ugyanazon a porton. Ilyenkor legtöbbször a második gép számára a külső port véletlenszerűen kerül kiválasztásra az ütközést elkerülendő, tehát ebben az esetben hol „címhez és porthoz kötött”, hol pedig „szimmetrikus fordítással” jut el a csomag egyik géptől a másikhoz, az igényeknek és a pillanatnyi lehetőségeknek megfelelően. Egyes protokollok ezt nehezen, vagy egyáltalán nem viselik el, így ennek kezelése megköveteli a megfelelő (a forgalmat értő) címfordító használatát.

Az általános, mindennapos internetböngészés (weboldal nézegetés) során a belső gép minden esetben a 80-as porton szólítja meg a külső gépet, de az egy teljesen véletlenszerűen választott porton küldi vissza a választ. Ezáltal megvalósítható, hogy egyetlen belső gép több külső gépről szolgáltatott weboldalhoz is egyszerre, egy időben hozzáférjen, mindegyiket a 80-as szabványos porton szólítva meg a kívánt webtartalom eléréséhez. Ha eközben egy másik belső gép valamelyik ugyanazon külső gép weboldalát kívánja szintén megnézni, az ő számára egy másik véletlenszerű porton érkezik majd a válasz, de ő is ugyanúgy a 80-as porton kezdeményezi ezt a kapcsolatot. Látható, hogy ebben az esetben a fordításhoz használt külső cím és port nem változott, de a visszirányító kapcsolat(ok)ban a port minden esetben más volt. Összességében elmondható tehát, hogy a választott címfordítási módszert mindig az elvégzendő feladat határozza meg.

## 10.2. Előkészítés

A gyakorlat során a szerver virtuális gépet használjuk, konfiguráljuk, a munkaállomásra csak az ellenőrzésnél lesz szükség. A szerver virtuális gépet úgy konfiguráljuk, hogy két hálózati

interfészsel rendelkezzen. Az első (`eth0`) a VirtualBox által nyújtott NAT szolgáltatáson keresztül kapcsolódjon a külvilág felé, míg a második (`eth1`) az *intnet* nevű belső hálózatra csatlakozzon. Az *intnet* belső hálózat célja az lesz, hogy a két virtuális gép közvetlenül kapcsolódhasson egymáshoz. A munkaállomás (Ubuntu Desktop) számára elegendő egy hálózati kártya (`eth0`), ami a belső hálózati gépre csatlakozik.



70. ábra. Virtuális gépek és hálózati interfészeik

A két gép TCP/IP konfigurációját úgy alakítjuk ki, hogy mindegyik interfész esetében statikusan állítjuk be az adatokat. A szerver esetében az `eth0` interfész:

IPv4 cím:	10.0.2.15
Hálózati maszk:	255.255.255.0
Alapértelmezett átjáró:	10.0.2.2
Hálózat:	10.0.2.0
Üzenetszórési cím:	10.0.2.255
DNS kiszolgáló:	10.1.51.23, 10.1.51.25
Névkeresési tartományok:	gamf.hu kefo.hu

A szerver `eth1` interfészének beállítása az alábbi lesz:

IPv4 cím:	192.168.1.254
Hálózati maszk:	255.255.255.0
Hálózat:	192.168.1.0
Üzenetszórési cím:	192.168.1.255

A munkaállomás `eth0` interfészének konfigurációja az alábbi lesz:

IPv4 cím:	192.168.1.2
Hálózati maszk:	255.255.255.0
Hálózat:	192.168.1.0
Üzenetszórési cím:	192.168.1.255

## 10.2. Tűzfal beállítása

A helyi hálózat és az internet felől érkező illetéktelen hozzáférés megakadályozása érdekében tűzfal telepítésére szükséges.

A Linux kernel tartalmazza a Netfilter alrendszert, amely a kiszolgálóra irányuló vagy azon átmenő hálózati forgalom sorsának befolyásolására vagy eldöntésére használható. Minden modern Linuxos tűzfalmegoldás ezt a rendszert használja csomagszűrésre.

A kernel csomagszűrő rendszere kevésbé lenne használható a kezelésére szolgáló, felhasználói térből használható felület nélkül. Amikor egy csomag eléri a kiszolgálóját, átkerül a Netfilter alrendszerhez elfogadásra, módosításra vagy elutasításra, a felhasználói térből az iptables segítségével megadott szabályok alapján. Így ha jól ismeri, akkor egyedül az iptablesre van szükség a tűzfal kezelésére, de számos előtétprogram érhető el a feladat egyszerűsítésére.

Az Ubuntu alapértelmezett tűzfalbeállító eszköze az ufw. Az iptables beállításának megkönnyítésére tervezett ufw felhasználóbarát módon teszi lehetővé IPv4 vagy IPv6 kiszolgáló alapú tűzfal létrehozását.

A tűzfal engedélyezése:

```
$ sudo ufw enable
```

A különböző portok megnyitásához, engedélyezéséhez az allow utasítás szükséges:

```
$ sudo ufw allow 80 - a http port engedélyezése
```

A deny paranccsal lehet a nyitott portot bezárni:

```
$ sudo ufw deny 80 - a http port tiltása
```

A tűzfal szabályokat számozott rendben is föl tudjuk venni:

```
$ sudo ufw insert 1 allow 631/tcp - nyomtató megosztás
```

A tűzfal szabály eltávolításához a delete parancs szükséges:

```
$ sudo ufw delete deny 80
```

Engedélyezni lehet egy adott hálózatról vagy kiszolgálóról a hozzáférést egy porthoz. Az alábbi példával megadjuk az SSH hozzáférést a 192.168.1.10 IP-című gép számára bármely IP-címhez ezen a kiszolgálón:

```
$ sudo ufw allow proto tcp from 192.168.1.10 to any port 22
```

Engedélyezhető az SSH hozzáférés egy teljes alhálózatból is:

```
$ sudo ufw allow proto tcp from 192.168.1.0/24 to any port 22
```

## 10.3. A NAT telepítése és konfigurálása

Az IP álcázás célja, hogy a privát, nem közvetíthető IP címekkel rendelkező gépek elérjék az internetet az álcázást végző gépen keresztül. A magánhálózatból az internetre irányuló forgalmat úgy kell módosítani, hogy visszairányítható legyen a kérést küldő gépre. Ehhez a kernelnek módosítani kell minden csomag forrás IP címét, hogy a válaszok hozzá legyenek visszairányítva, a kérést küldő gép IP címe helyett, különben a válaszok nem érkeznének meg. A Linux a kapcsolatkövetést (conntrack) használja a gépek és a hozzájuk tartozó kapcsolatok nyilvántartására, és a visszaküldött csomagok ennek megfelelő átirányítására. A hálózatát elhagyó forgalom így „álcázva” lesz, mintha az átjáró gépről indult volna.

Az IP álcázást `ufw` szabályok segítségével valósítjuk meg. Ezek aszerint vannak két állományra különválasztva, hogy a parancssori `ufw` szabályok előtt (`before.rules`) vagy után (`after.rules`) kerülnek-e végrehajtásra.

Első lépésként engedélyezzük a csomagtovábbítást. Ehhez nyissuk meg szerkesztésre a `/etc/default/ufw` állományt:

```
$ sudo nano /etc/default/ufw
```

A fájlban a

```
DEFAULT_FORWARD_POLICY="DROP"
```

beállítást cseréljük

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

-ra, és mentjük az állományt. Ezt követően nyissuk meg szerkesztésre a `/etc/ufw/sysctl.conf` állományt,

```
$ sudo mcedit /etc/ufw/sysctl.conf
```

vegyük ki a megjegyzésből a `net/ipv4/ip_forward=1` sort. Amennyiben az IPv6 csomagtovábbítást is engedélyezni kívánjuk, akkor a `net/ipv6/conf/default/forwarding=1` sort is vegyük ki megjegyzésből. Mentjük az állományt.

Második lépésként konfigurálnunk kell a `nat` táblát, mivel alapértelmezés szerint csak a `filter` tábla konfigurált. Ehhez nyissuk meg szerkesztésre a `/etc/ufw/before.rules` állományt

```
$ sudo mcedit /etc/ufw/before.rules
```

A bevezető megjegyzés (#) sorokat követően helyezzük el az alábbiakat

```
# nat tábla szabályai
*nat
```

```
:POSTROUTING ACCEPT [0:0]

# Továbbítsa az eth1-ről érkező forgalmat az eth0-n keresztül
-A POSTROUTING -s 192.168.1.0/24 -o eth0 -j MASQUERADE

COMMIT
```

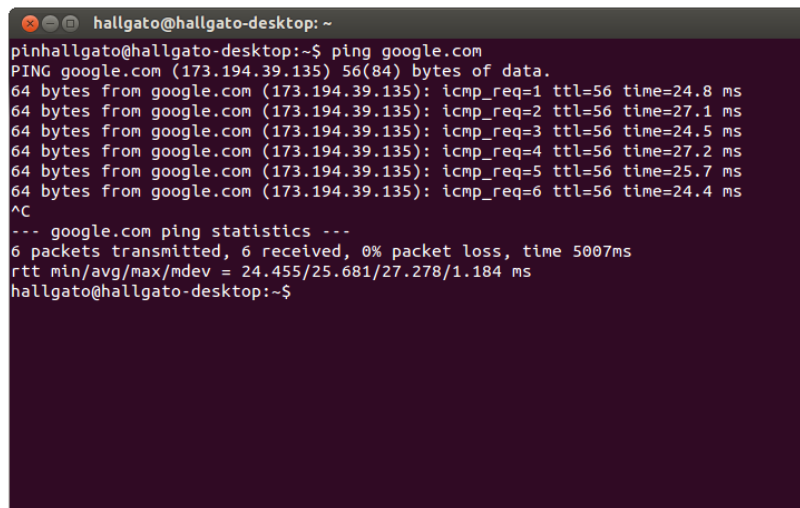
Utolsó lépésként letiltjuk, és újra engedélyezzük az ufw szoftver működését

```
$ sudo ufw disable
$ sudo ufw enable
```

## 10.4. Tesztelés

A kliens gépen ping teszt segítségével megkísérlünk elérni egy külső IP címet annak érdekében, hogy ellenőrizzük, hogy a kliens gép látja-e a külső hálózatot a NAT szerverünkön keresztül.

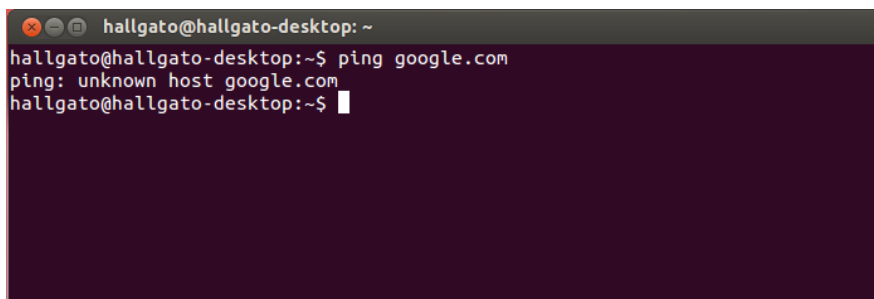
```
$ ping google.com
```



```
hallgato@hallgato-desktop: ~
pinhallgato@hallgato-desktop:~$ ping google.com
PING google.com (173.194.39.135) 56(84) bytes of data:
64 bytes from google.com (173.194.39.135): icmp_req=1 ttl=56 time=24.8 ms
64 bytes from google.com (173.194.39.135): icmp_req=2 ttl=56 time=27.1 ms
64 bytes from google.com (173.194.39.135): icmp_req=3 ttl=56 time=24.5 ms
64 bytes from google.com (173.194.39.135): icmp_req=4 ttl=56 time=27.2 ms
64 bytes from google.com (173.194.39.135): icmp_req=5 ttl=56 time=25.7 ms
64 bytes from google.com (173.194.39.135): icmp_req=6 ttl=56 time=24.4 ms
^C
--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 24.455/25.681/27.278/1.184 ms
hallgato@hallgato-desktop:~$
```

71. ábra. Sikeres teszt a kliens gépnek a külvilág felé

A tűzfalon a 80-as port (HTTP) tiltásával az internet böngészést tiltjuk le:



```
hallgato@hallgato-desktop: ~
hallgato@hallgato-desktop:~$ ping google.com
ping: unknown host google.com
hallgato@hallgato-desktop:~$
```

72. ábra. A kliens gép sikertelen ping tesztje

## **10.5. Vonatkozó irodalomjegyzék**

1. Hálózati címfordítás – Wikipédia  
[http://hu.wikipedia.org/wiki/Hálózati\\_Címfordítás](http://hu.wikipedia.org/wiki/Hálózati_Címfordítás)
2. Ubuntu Server Guide  
<http://sugo.ubuntu.hu/10.10/html/serverguide/hu/>





## 11. LDAP kiszolgáló telepítése (Göcs László)

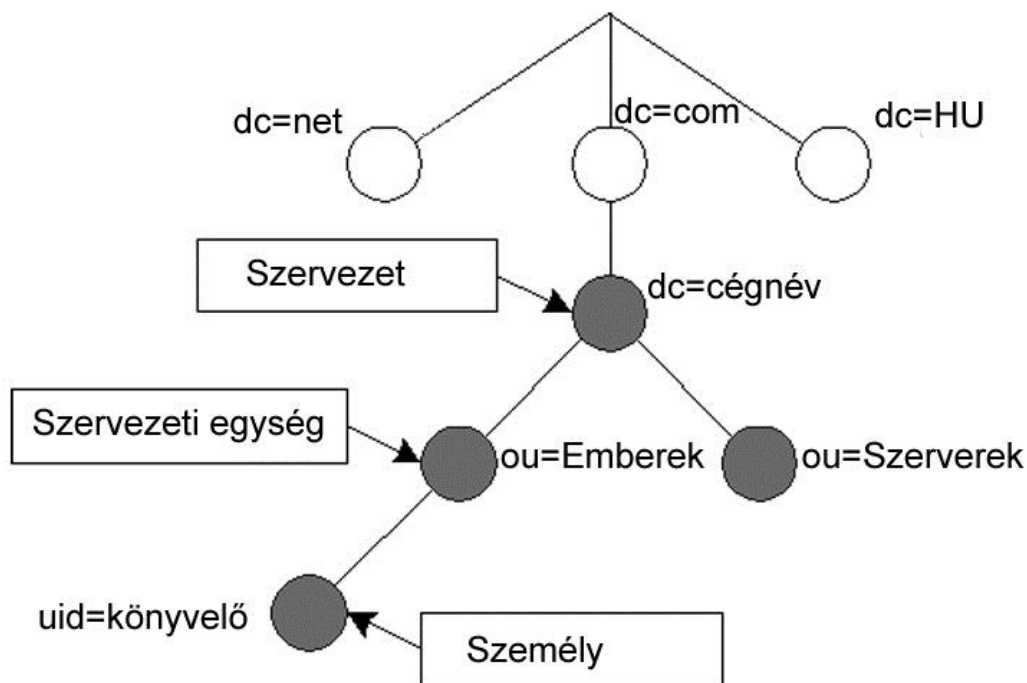
### 11.1. Mi az LDAP?

Az LDAP a Lightweight Directory Access Protocol rövidítése. Mint a neve mutatja, az LDAP nem több egy protokollnál, nem egy konkrét szoftverről van tehát szó. Ez a protokoll címtár szolgáltatások elérését szabályozza. A címtár olyan, mint egy adatbázis, de arra törekszik, hogy magába foglaljon egy részletesebb, tulajdonság alapú információkezelést. Általában az információt a címtárban gyakrabban olvassák, mint írják. Ennek következtében a címtárak általában nem alkalmaznak bonyolult tranzakció-kezelést vagy roll-back rendszereket, amiket általában az adatbázis kezelők használnak a nagyméretű, összetett frissítésekhez. A címtár aktualizálás tipikusan egyszerű, mindent vagy semmit jellegű változás.

Sok különböző lehetőség van címtár szolgáltatás nyújtására. Különböző eljárásokkal más és más információk tárolhatók a címtárban, különböző követelmények szerint lehet hivatkozni az információkra, lekérdezhetők és frissíthetők, védhetők a meg nem engedett hozzáféréstől, stb. Néhány helyi címtár szolgáltatás korlátozott környezetben nyújt szolgáltatásokat (pl a finger szolgáltatás önálló gépen). Más szolgáltatások globálisak, széles körben elérhetőek.

Az LDAP címtárszolgáltatás kliens-szerver modellen alapul. Egy vagy több LDAP szerveren tárolt adatból épül fel az LDAP fa vagy LDAP háttér adatbázis. Az LDAP kliens egy LDAP szerverhez csatlakozik, és felteszi a kérdéseit. A szerver a válasszal reagál, vagy egy mutatóval, hol talál több információt a kliens (tipikusan egy másik LDAP szerver). Mindegy, hogy a kliens melyik LDAP szerverhez csatlakozik, ugyanazt a címtárat látja, ugyanaz a név az egyik címtár szerveren ugyanazt az adatot jeleníti meg, mint a másikon. Ez fontos tulajdonsága az olyan globális címtárszolgáltatásoknak, mint az LDAP.

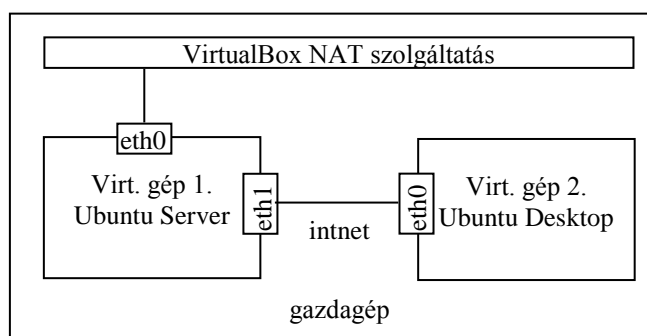
Az információ egy faszzerű szerkezetben tárolódik, amelynek minden csúcsában bejegyzések (entry) szerepelnek. Egy bejegyzésnek van típusa, amely meghatározza, hogy milyen attribútumai lehetnek. Minden egyes ilyen bejegyzésre egyértelműen hivatkozhatunk a bejegyzés DN-jével (Distinguished Name – megkülönböztető név), amely lényegében a fában a csúcshoz vezető utat írja le.



73. ábra. Fa szerkezet a címtárban

## 11.2. Előkészítés

A gyakorlat során a szerver virtuális gépet használjuk, konfiguráljuk, a munkaállomásra csak az ellenőrzésnél lesz szükség. A szerver virtuális gépet úgy konfiguráljuk, hogy két hálózati interfésszel rendelkezzen. Az első (`eth0`) a VirtualBox által nyújtott NAT szolgáltatáson keresztül kapcsolódjon a külvilág felé, míg a második (`eth1`) az `intnet` nevű belső hálózatra csatlakozzon. Az `intnet` belső hálózat célja az lesz, hogy a két virtuális gép közvetlenül kapcsolódhasson egymáshoz. A munkaállomás (Ubuntu Desktop) számára elegendő egy hálózati kártya (`eth0`), ami a belső hálózati gépre csatlakozik.



70. ábra. Virtuális gépek és hálózati interfészeik

A két gép TCP/IP konfigurációját úgy alakítjuk ki, hogy mindegyik interfész esetében statikusan állítjuk be az adatokat. A szerver esetében az `eth0` interfész:

IPv4 cím: 10.0.2.15  
Hálózati maszk: 255.255.255.0  
Alapértelmezett átjáró: 10.0.2.2  
Hálózat: 10.0.2.0  
Üzenetszórési cím: 10.0.2.255  
DNS kiszolgáló: 10.1.51.23, 10.1.51.25  
Névkeresési tartományok: gamf.hu kefo.hu

A szerver `eth1` interfészének beállítása az alábbi lesz:

IPv4 cím: 192.168.1.254  
Hálózati maszk: 255.255.255.0  
Hálózat: 192.168.1.0  
Üzenetszórési cím: 192.168.1.255

A munkaállomás `eth0` interfészének konfigurációja az alábbi lesz:

IPv4 cím: 192.168.1.2  
Hálózati maszk: 255.255.255.0  
Hálózat: 192.168.1.0  
Üzenetszórési cím: 192.168.1.255

### 11.3. Telepítés és konfigurálás

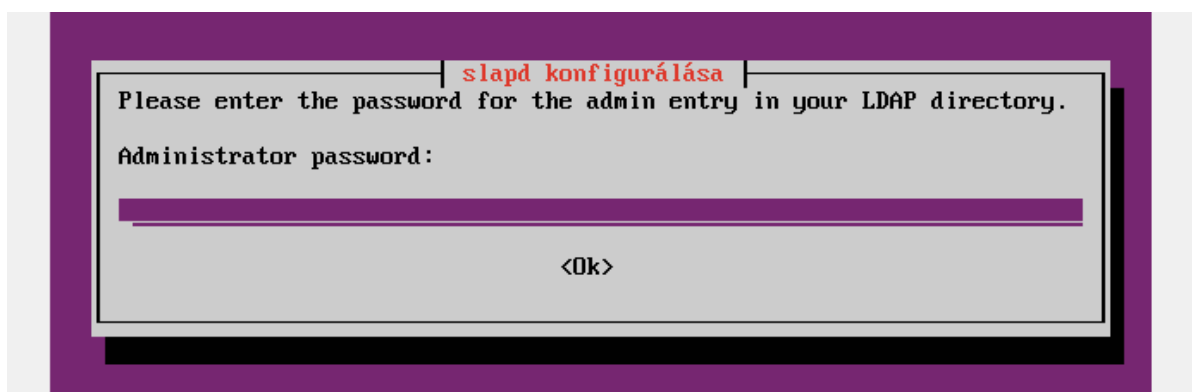
Frissítük a csomag adatbázist, majd telepítjük a címtárszolgáltatáshoz szükséges programcsomagokat:

```
$ sudo apt-get update
$ sudo apt-get install sldap
```

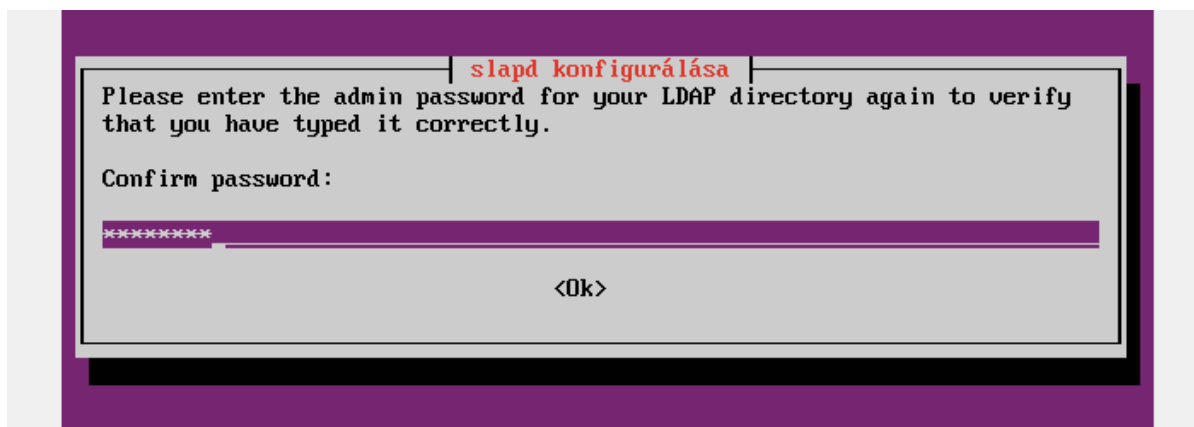
A telepítő varázsló egyes lépéseit az alábbi képernyő-képsoron követhetjük végig.

```
root@ubuntu-server:~# apt-get install slapd
Csomaglisták olvasása ■ Kész0%
Függőségi fa építése
Allapotinformációk olvasása ■ Kész
Az alábbi extra csomagok kerülnek telepítésre:
  libldap-2.4-2 libldap7 libodbc1 libperl5.14 libsldap1 libwrap0 tcpd
Javasolt csomagok:
  libmyodbc odbc-postgresql tdsodbc unixodbc-bin slapd openslap-doc ldap-utils
Az alábbi UJ csomagok lesznek telepítve:
  libldap7 libodbc1 libperl5.14 libsldap1 libwrap0 slapd tcpd
Az alábbi csomagok frissítve lesznek:
  libldap-2.4-2
1 frissített, 7 újonnan telepített, 0 eltávolítandó és 40 nem frissített.
Letöltendő adatmennyiség: 2.309 kB.
A művelet után 5.711 kB lemezterület kerül felhasználásra.
Folytatni akarja [I/n]? ^I
```

74. ábra. LDAP telepítése



75. ábra. Rendszergazdai jelszó megadása



76. ábra.

```

Kicsomagolás: libwrap0 innen: ../libwrap0_7.6.q-21_amd64.deb ...
Selecting previously unselected package libperl5.14.
Kicsomagolás: libperl5.14 innen: ../libperl5.14_5.14.2-6ubuntu2_amd64.deb ...
Selecting previously unselected package libslp1.
Kicsomagolás: libslp1 innen: ../libslp1_1.2.1-7.8ubuntu1_amd64.deb ...
Selecting previously unselected package slapd.
Kicsomagolás: slapd innen: ../slapd_2.4.28-1.1ubuntu4.1_amd64.deb ...
Selecting previously unselected package tcpd.
Kicsomagolás: tcpd innen: ../tcpd_7.6.q-21_amd64.deb ...
man-db triggereinek feldolgozása■
ureadahead triggereinek feldolgozása■
ureadahead will be reprofiled on next reboot
ufw triggereinek feldolgozása■
Beállítás: libldap-2.4-2 (2.4.28-1.1ubuntu4.1) ...
Beállítás: libldap2 (2.4.2-1ubuntu1) ...
Beállítás: libodbc1 (2.2.14p2-5ubuntu3) ...
Beállítás: libwrap0 (7.6.q-21) ...
Beállítás: libperl5.14 (5.14.2-6ubuntu2) ...
Beállítás: libslp1 (1.2.1-7.8ubuntu1) ...
Beállítás: slapd (2.4.28-1.1ubuntu4.1) ...
  Creating new user openldap... done.
  Creating initial configuration... done.
  Creating LDAP directory... done.
* Starting OpenLDAP slapd
Beállítás: tcpd (7.6.q-21) ...
libc-bin triggereinek feldolgozása■
ldconfig deferred processing now taking place
root@ubuntu-server:~#
    
```

77. ábra. LDAP telepítés befejezése

Az LDAP-hoz az `ldap-utils` csomag telepítése is szükséges:

```
$ sudo apt-get install ldap-utils
```

```
root@ubuntu-server:~# apt-get install ldap-utils
Csomaglisták olvasása █ Kész0%
Függőségi fa építése
Allapotinformációk olvasása █ Kész
Az alábbi ÚJ csomagok lesznek telepítve:
  ldap-utils
0 frissített, 1 újonnan telepített, 0 eltávolítandó és 40 nem frissített.
Letöltendő adatmennyiség: 282 kB.
A művelet után 693 kB lemezterület kerül felhasználásra.
FIGYELMEZTETÉS: Az alábbi csomagok nem hitelesíthetők!
  ldap-utils
Valóban ellenőrzés nélkül telepíti a csomagokat (i/N)?
```

78. ábra. LDAP kiegészítő csomag telepítése

A telepítést követően néhány sémafájlt kell betölteni a következő parancsok segítségével:

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f/etc/ldap/schema/cosine.ldif
```

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f/etc/ldap/schema/nis.ldif
```

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/inetorgperson.ldif
```

A címtár információk importálása és exportálása két directory alapú szerver között, és a címtárakban alkalmazott változások leírására az LDIF formátumként ismert (LDAP Data Interchange Format) file-okkal lehetséges. Az LDIF file a bejegyzéseket objektum orientált hierarchikus formában tárolja. Az LDAP csomag tartalmazza az LDIF file-ok LDBM formátumba konvertálásához szükséges eszközöket.

Mint látható, minden bejegyzésnek saját azonosítója van, a distinguished name (megkülönböztető név), vagy dn. A dn a bejegyzés nevéből áll, megtoldva a név elérési útjával vissza a címtár hierarchia csúcsáig.

Az LDAP objektumosztályok határozzák meg a bejegyzések azonosításához használható jellemzők csoportját.

Az LDAP standard az alábbi alapvető objektum osztályokat nyújtja:

- Group (csoport), független objektumok rendezetlen listája vagy objektumok csoportja.
- Location (elhelyezkedés), az országok nevei és leírásuk.
- Organization (szervezet).
- People (személy).

Egy bejegyzés több objektumosztályhoz tartozhat. Például a person bejegyzést definiálja a person objektumosztály, de szintén definiálja az inetOrgPerson, groupOfNames és organization objektumosztályok. Az ldap szerver objektumosztály struktúráját (sémáját) meghatározza az egyes bejegyzések számára szükséges és engedélyezett attribútumok egyéni listája.

Készítsük el az LDAP adatbázis konfigurációját, háttér definícióját (hdb – mint tranzakciós adatbázis).

Az alábbi LDIF állományt **backend.gyakorlat.hu.ldif** néven hozzuk létre:

```
# Dinamikus backend modulok betöltése
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: back_hdb

# Adatbázis-beállítások
dn: olcDatabase=hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcSuffix: dc=gyakorlat,dc=hu
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=admin,dc=gyakorlat,dc=hu
olcRootPW: jelszo
olcDbConfig: set_cachesize 0 2097152 0
olcDbConfig: set_lk_max_objects 1500
olcDbConfig: set_lk_max_locks 1500
olcDbConfig: set_lk_max_lockers 1500
olcDbIndex: objectClass eq
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcAccess: to attrs=userPassword by
dn="cn=hallgato,dc=gyakorlat,dc=hu" write by anonymous auth by
self write by * none
olcAccess: to attrs=shadowLastChange by self write by * read
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=hallgato,dc=gyakorlat,dc=hu" write
by * read
```

Az LDIF-fájl másolása a címtárhoz:

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f
backend.gyakorlat.hu.ldif
```

Az előtét címtár készen áll a feltöltésre. Hozzunk létre egy **frontend.gyakorlat.hu.ldif** nevű fájlt a következő tartalommal:

```
# A tartomány felső szintű objektumának létrehozása
```



```
dn: dc=gyakorlat,dc=hu
objectClass: top
objectClass: dcObject
objectclass: organization
o: Példaszervezet
dc: Example
description: LDAP példa

# Admin felhasználó.
dn: cn=hallgato,dc=hallgato,dc=hu
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: hallgato
description: LDAP adminisztrátor
userPassword: jelszo

dn: ou=people,dc=gyakorlat,dc=hu
objectClass: organizationalUnit
ou: people

dn: ou=groups,dc=gyakorlat,dc=hu
objectClass: organizationalUnit
ou: groups
```

Vegyük fel a bejegyzéseket az LDAP-címtárba:

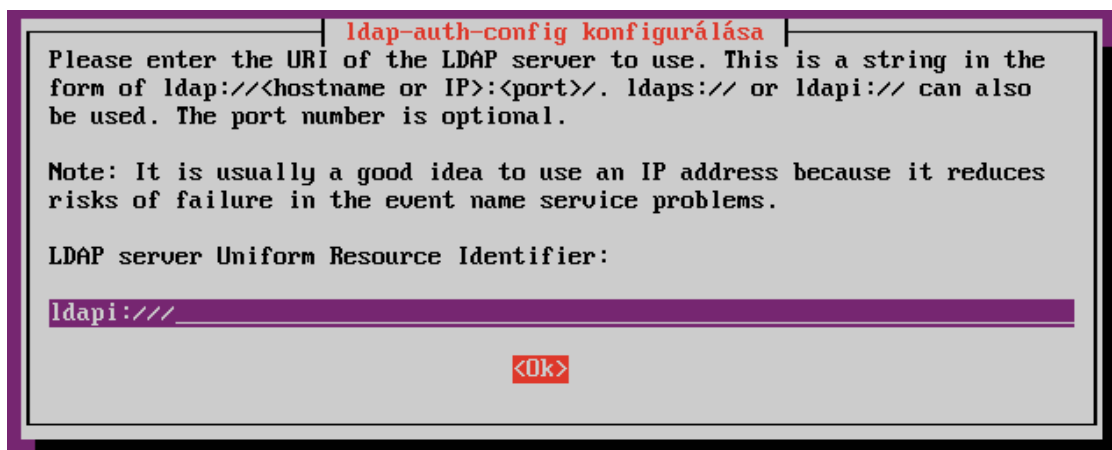
```
$ sudo ldapadd -x -D cn=hallgato,dc=gyakorlat,dc=hu -W -f
frontend.gyakorlat.hu.ldif
```

### 11.3.1. Az LDAP alapú hitelesítés konfigurálása

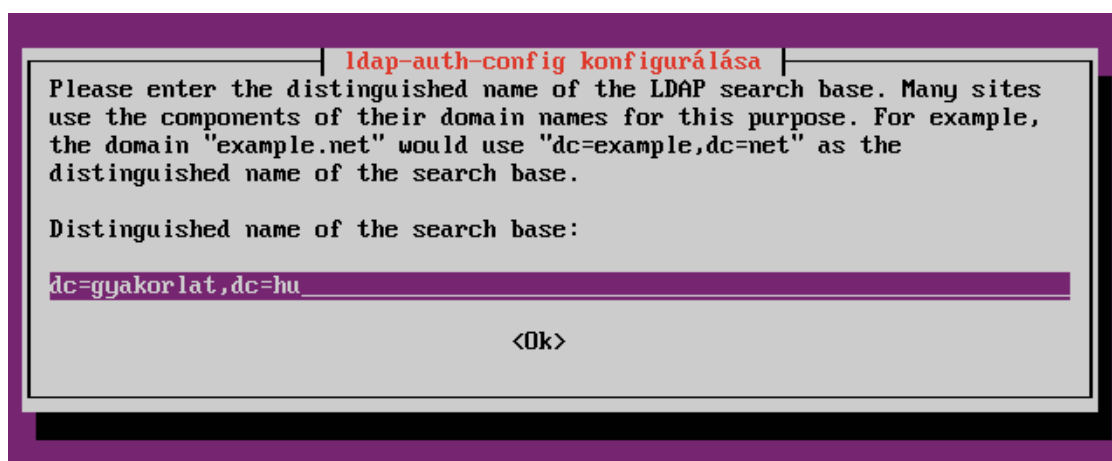
Az LDAP alapú hitelesítés megvalósításához telepítjük a `libnss-ldap` csomagot.

```
$ sudo apt-get install libnss-ldap
```

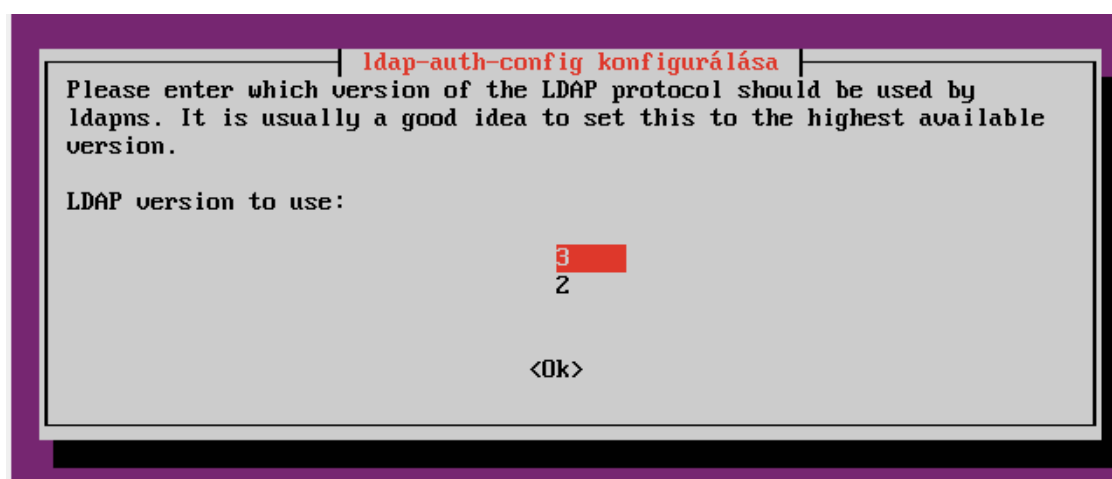
A telepítést követően egy varázsló indul el, aminek képernyőképeit az alábbi ábrák segítségével követhetjük végig.



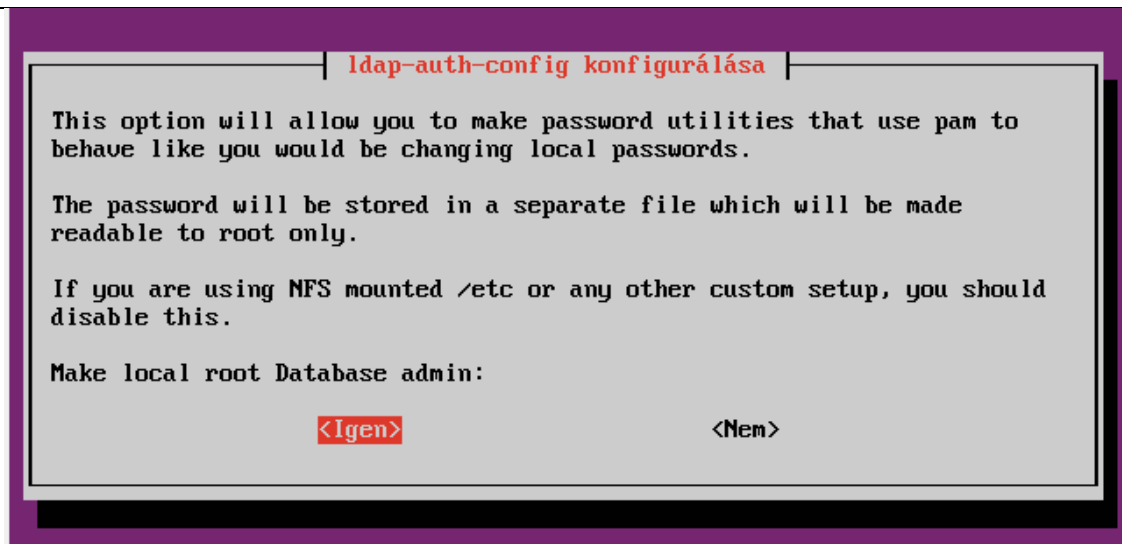
79. ábra. Az LDAP elérési útja



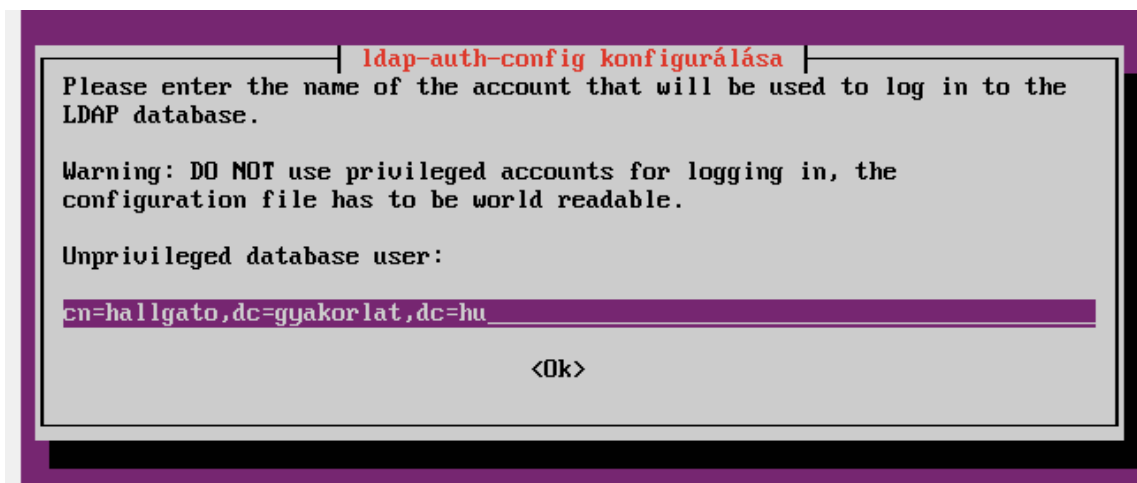
80. ábra. A szervezeti egység domain nevének megadása



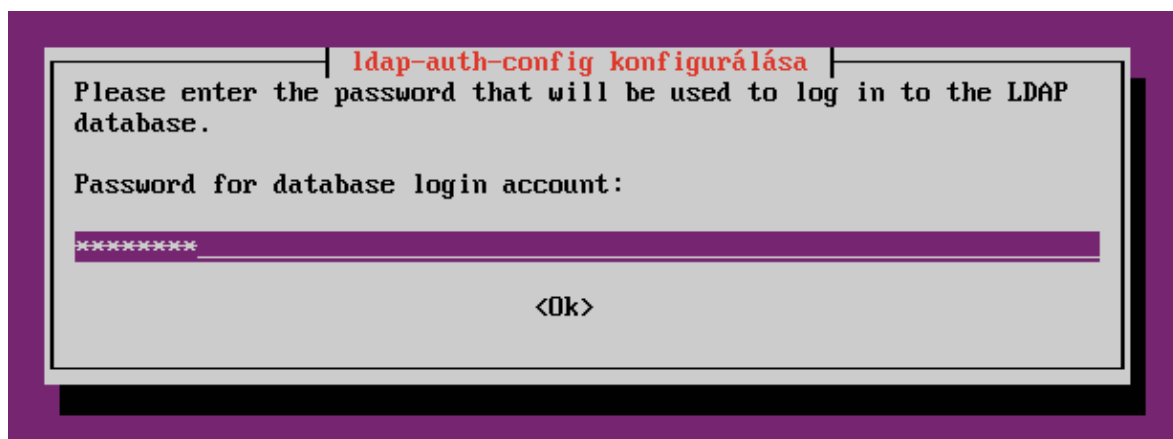
81. ábra. LDAP protokoll verziójának kiválasztása



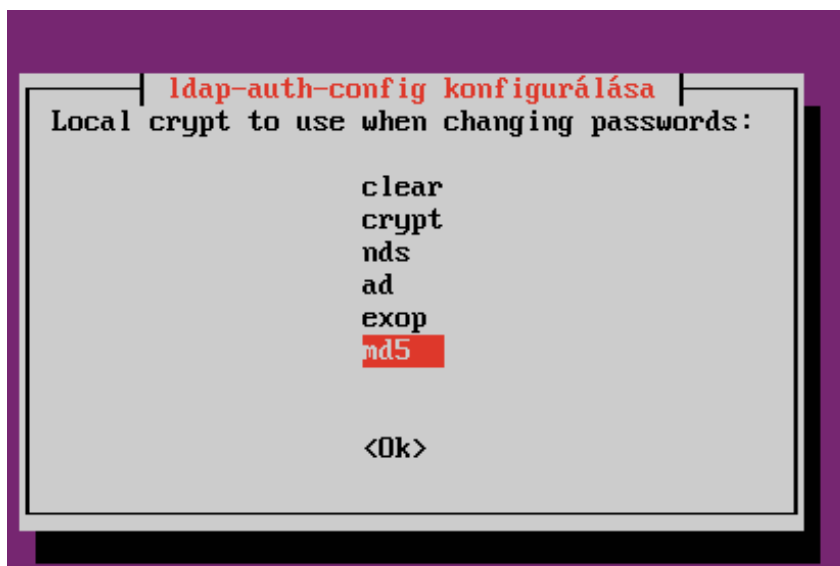
82. ábra. A helyi rendszergazda az adatbázis rendszergazdája



83. ábra. Az adatbázis eléréséhez szükséges felhasználó



84. ábra. Az adatbázis eléréséhez szükséges felhasználó jelszavának megadása



85. ábra. A jelszó titkosításának algoritmus

A párbeszédablakokban megadott adatok az `/etc/ldap.conf` fájlba kerülnek.

A konfiguráció hitelesítése beállítható a következő parancs kiadásával:

```
$ sudo auth-client-config -t nss -p lac_ldap
```

A parancs kapcsolóinak jelentése a következő:

-t: csak az `/etc/nsswitch.conf` fájlt módosítja.

-p: az engedélyezendő/letiltandó stb. profil neve.

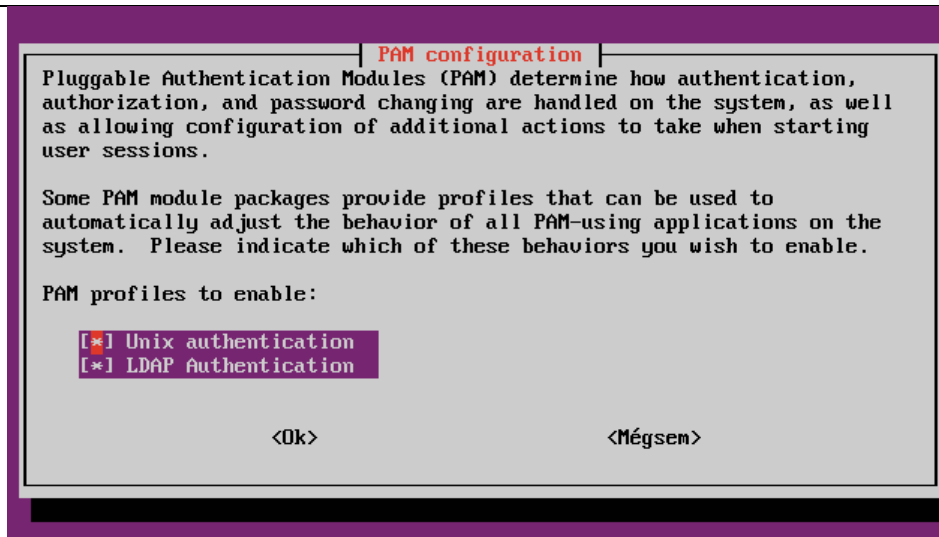
lac\_ldap: az auth-client-config profil, amely az ldap-auth-config csomag része.

Szükséges a felhasználó azonosítására szolgáló modul, a PAM (Pluggable Authentication Modules; Csatlakoztatható Azonosítási Modulok) telepítése. Amikor egy programnak szüksége van a felhasználó azonosítására, akkor a PAM programkönyvtárban tárolt függvényeket biztosít a megfelelő azonosítási módhoz. Mivel ez a programkönyvtár dinamikusan töltődik be, az azonosítási mód megváltoztatható a konfigurációs fájl szerkesztésével.

A pam-auth-update segédprogram segítségével állítsa be a rendszert az LDAP használatára hitelesítésre:

```
$ sudo pam-auth-update
```

A pam-auth-update menüből válassza ki az LDAP-t, és az egyéb szükséges hitelesítési mechanizmusokat.



86. ábra.

### 11.3.2. Felhasználók és csoportok létrehozása

Az `ldapscripts` csomag az LDAP felhasználók és csoportok egyszerű kezeléséhez tartalmaz konfigurálható parancsfájlokat.

```
$ sudo apt-get install ldapscripts
```

Ezután szerkesszük az `/etc/ldapscripts/ldapscripts.conf` konfigurációs fájlt, vegyük ki megjegyzésből és módosítsuk a következőket a környezetünknek megfelelően:

```
SERVER=localhost
BINDDN='cn=hallgato,dc=gyakorlat,dc=hu'
BINDPWDFILE="/etc/ldapscripts/ldapscripts.passwd"
SUFFIX='dc=gyakorlat,dc=hu'
GSUFFIX='ou=Groups'
USUFFIX='ou=Peoplek'
MSUFFIX='ou=Machines'
GIDSTART=10000
UIDSTART=10000
MIDSTART=10000
```

Ezután hozzuk létre az `ldapscripts.passwd` fájlt a címtár hitelesített elérésének lehetővé tételéhez:

```
$ sudo sh -c "echo -n 'jelszo'>/etc/ldapscripts/ldapscripts.
passwd"
```

A fájlt állítjuk be a tulajdonos számára olvasási joggal:

```
sudo chmod 400 /etc/ldapscripts/ldapscripts.passwd
```

### A felhasználó felvétele:

```
dn: cn=Kiss Istvan,ou=People,dc=gyakorlat,dc=hu
objectClass: posixAccount
cn: Kiss Istvan
gecos: Kiss Istvan
uid: kiss.istvan
uidNumber: 1100
gidNumber: 1100
homeDirectory: /home/kiss.istvan
loginShell: /bin/bash
userPassword: xX12345
```

```
$ sudo ldapmodify -a -x -D cn=hallgato,dc=gyakorlat,dc=hu -W
-f frontend.gyakorlat.hu.ldif
```

Fontos, hogy az uidNumber értéke ne legyen azonos lokális felhasználó értékével az /etc/passwd fájlban!

### Felhasználó jelszavának módosítása:

```
$ sudo ldapsetpasswd kiss.istvan
Changing passwordforuserid=kiss.istvan,ou=People,
dc=gyakorlat,dc=hu
New Password:
New Password (verify):
```

### Felhasználó törlése:

```
$ sudo ldapdeleteuser kiss.istvan
```

### Csoport hozzáadása:

```
#A ou alá létrehozza a felhasznalok csoportot:
dn: cn=felhasznalok,ou=groups,dc=gyakorlat,dc=hu
objectClass: posixGroup
cn: felhasznalok
gidNumber: 10000
```

```
$ sudo ldapmodify -a -x -D cn=hallgato,dc=gyakorlat,dc=hu -W
-f frontend.gyakorlat.hu.ldif
```

Fontos, hogy a gidNumber értéke ne legyen azonos a lokális csoport értékével az /etc/group fájlban!

### Parancssorból:

```
$ sudo ldapaddgroup munkasok
```

Csoport törlése:

```
$ sudo ldapdeletegroup munkasok
```

Felhasználó csoporthoz adása:

```
$ sudo ldapaddusertogroup kiss.istvan munkasok
```

Ezután a qa csoport memberUid attribútumának értéke kiss.istvan lesz.

Felhasználó eltávolítása csoportból:

```
$ sudo ldapdeleteuserfromgroup kiss.istvan munkasok
```

## 11.4. Vonatkozó irodalomjegyzék

1. Halász Gábor fordítása (University of Michigan LDAP információs oldala)  
<http://www.szabilinux.hu/ldap>
2. Ubuntu dokumentáció  
<http://sugo.ubuntu.hu/10.04/html/serverguide/hu/samba-ldap.html>
3. Papp Zoltán honlapja  
<http://padre.web.elte.hu/ldap.html>

## 12. A ClamAV víruskereső telepítése (Johanyák Zsolt Csaba)

### 12.1. Telepítés és beállítás

Bár eléggé elterjedt nézet, hogy Linuxon nincs szükség víruskereső/irtó programokra, azért nem szabad ennyire könnyen kezelni ezt a kérdést. Egyrészt ártó programokkal még Linux alatt is találkozhatunk (ld. [Hiba! A hivatkozási forrás nem található.]), másrészt Linuxos kiszolgálónk FTP vagy levelező szerver is lehet, ahol alapvető elvárás a feltöltött állományok illetve továbbítandó levelek ellenőrzése. A feladat megoldására viszonylag kevés termék közül választhatunk [Hiba! A hivatkozási forrás nem található.][Hiba! A hivatkozási forrás m található.].

Az alábbiakban egy szabad szoftveres víruskereső program, a ClamAV konfigurálásának néhány elemével ismerkedünk meg. A ClamAV a következő szolgáltatásokat nyújtja: vírusok keresése, a fertőzött állományok karanténba mozgatása vagy törlése. Vírust azonban nem képes eltávolítani egy fertőzött állományból.

Telepítése történhet közvetlenül a Main tárolóból. Itt a letesztelt kipróbált változat található, amennyiben azonban mindenképp ragaszkodunk a legfrissebb változathoz, akkor a ClamAV PPA-ból telepítsünk. A továbbiakban az első változatot mutatjuk be (Main tároló). Először frissítjük a csomagok listáját, majd telepítjük a clamav-daemon csomagot.

```
$ sudo apt-get update
$ sudo apt-get install clamav-daemon
```

A telepítés során létrejön egy clamav nevű felhasználói fiók. A telepítést követően egy figyelmeztető üzenet tájékoztat arról, hogy a vírus adatbázis hét napnál régebbi. Amennyiben hálózatunk egy proxy szerver mögött helyezkedik el akkor a frissítés elindítása előtt a szerver adatait meg kell adnunk a /etc/clamav/freshclam.conf állományban. Nyissuk meg az állományt,

```
$ sudo nano /etc/clamav/freshclam.conf
```

és írjuk be a végére a proxy szerver címét és portját megadó két sort az alábbi mintát követve.

```
HTTPProxyServer http://ns-proxy.xxx.hu
HTTPProxyPort 6543
```

Az URL és a négyjegyű szám helyére a saját hálózatunk adatait írjuk be. A konfigurációs állomány mentését követően a

```
$ freshclam
```

paranccsal frissítjük az adatbázist. A telepítést követően két démon indult el: maga a víruskereső szolgáltatást nyújtó démon (clamav-daemon) és a vírusadatbázis frissítést végző démon (clamav-freshclam). Működésükről a



```
$ ps ax | grep clam
```

paranccsal győződhetünk meg.  
A telepített démonokat a

```
$ sudo service clamav-daemon stop|start|restart  
$ sudo service clamav-freshclam stop|start|restart
```

parancsokkal állíthatjuk le, indíthatjuk el, illetve indíthatjuk újra. A ClamAV-ot két módon használhatjuk.

A kézi használat során a clamscan parancsot használjuk, pl. a

```
$ clamscan -r /munka
```

parancs megvizsgálja a munka és az alatta levő könyvtárakat és eltávolítja a felismert vírusokat. Az összes megvizsgált állomány nevét kiírja a standard kimenetre. a program működését számos kapcsolóval befolyásolhatjuk, pl. --quiet (a vizsgálat során csak a hibaüzeneteket írja ki), --no-summary (nem írja ki a végső összeggést), --log=fájlnév (a megadott fájlba naplóz), --move=könyvtárnév (a fertőzött állományokat a megadott könyvtárba mozgatja). Részletes tájékoztatást a

```
$ man clamscan
```

 paranccsal kaphatunk.

A ClamAV-ot démonként is futtathatjuk, ilyenkor egy kliensprogram segítségével vehetjük igénybe a szolgáltatásait. Számos program, pl. e-mail szerverek képesek erre, de saját szkriptből is meghívhatjuk pl. az alábbiak szerint. Hozzunk létre egy szkript állományt clamav.sh néven.

```
$ sudo nano clamav.sh
```

Írjuk bele az alábbi két sort.

```
#!/bin/bash  
/usr/bin/clamscan --remove "$1"
```

Az első sor jelzi, hogy egy ez Bash szkript. A második sorban szereplő --remove kapcsoló hatására a felismert vírus törlődik. A "\$1" azt eredményezi, hogy az első paraméterként átvett könyvtárban fog keresni, illetve ha egy állományról van szó, akkor azt fogja megvizsgálni. a második sorban további kapcsolókat is elhelyezhetünk, pl. --quiet (a vizsgálat során csak a hibaüzeneteket írja ki), --no-summary (nem írja ki a végső összeggést), --log=fájlnév (a megadott fájlba naplóz), --multiscan (több szálon fut), --move=könyvtárnév (a fertőzött állományokat a megadott könyvtárba mozgatja).

Az előzőekben létrehozott szkriptet kétféle képpen indíthatjuk el. Amennyiben nem kívánjuk futtatási (x) jogosultsággal ellátni, akkor a

```
$ /bin/bash clamav.sh
```

parancsot használjuk. Egyszerűbb azonban, ha először megadjuk az állományhoz a futtatási engedélyt a

```
$ chmod +x clamav.sh
```

paranccsal. Ezt követően a továbbiakban elegendő a szkript nevét megadnunk a futtatáshoz.

```
$ clamav.sh
```

## **12.2. Vonatkozó irodalomjegyzék**

1. Linux malware  
[http://en.wikipedia.org/wiki/Linux\\_malware#Anti-virus\\_applications](http://en.wikipedia.org/wiki/Linux_malware#Anti-virus_applications)
2. Antivirus  
<https://help.ubuntu.com/community/Antivirus>