Melega Kálmán

Rendszergazdai ismeretek

Linux szolgáltatások

2005

Tartalomjegyzék

Bevezetés

Napjainkban dinamikusan fejlődik a Linux operációs rendszer, és egyaránt sikeresen alkalmazható szerver vagy munkaállomás operációs rendszereként.

Előnyei közé tartozik ingyenes használhatósága mellett az is, hogy a vírusok kevésbé veszélyeztetik működését. Egy rendszergazda számára ismerete napjainkban már nélkülözhetetlen. Parancsai általában megegyeznek a Unix rendszerek (például FreeBSD) parancsaival.

Kiknek szól a könyv?

A könyv azoknak íródott, akik már Linux felhasználói gyakorlattal rendelkeznek. Ismerik a parancsokat, és képesek egyszerűbb héjprogramok elkészítésére is. Aki tehát most ismerkedik ezzel az operációs rendszerrel, annak nem ezzel a könyvvel kell kezdenie tanulmányait.

Jelölések

A szövegben lényegesnek tartott szavak, szövegrészek **félkövér** stílusban jelennek meg.

Ha a konzolon (shell, parancsértelmező, terminál) adunk ki egy parancsot, akkor ezt a készenléti jel (prompt) után tehetjük meg. A root prompt **#**, a felhasználói prompt pedig **$** karakterre végződik. A prompt általános alakja root-ként, illetve felhasználóként :

```
root@hostnév:elérési-út#
user@hostnév:elérési-út$
```

A gépnév csak akkor jelenik meg a prompt-ban, ha az alapértelmezett **localhost** helyett egyedi gépnevet adunk meg.

UHU-Linux

A könyv az UHU-Linux operációs rendszer használatán keresztül mutatja be a konkrét lehetőségeket. Ez természetesen nem jelenti azt, hogy más Linux disztribúció alkalmazása esetén használhatatlan lenne.
Az UHU-Linux egy hazai csapat fejlesztésének eredménye, alapváltozata ingyenesen használhatóak bárki számára. A telepítő CD írásához szükséges képfájlok (.iso) az **ftp://www.uhulinux.hu** helyről tölthetők le.

Létezik az UHU-Linux közvetlen, telepítés nélküli használatra alkalmas változata, az **UHU-Live CD,** ami a **http://www.linuxforum.hu** helyen érhető el.

Ajánlott az UHU-Linux levelezési listákra feliratkozni: **http://lists.uhulinux.hu,** itt az operációs rendszerrel kapcsolatos kérdésekre többek között a fejlesztők válaszolnak.

Az UHU-Linux telepítésének hardver igénye

A könyvben lévő tananyag elsajátításához szükségünk lesz egy működő UHU-Linux operációs rendszerre, amelyhez ajánlott legalább 256 MB RAM-ot és 800 MHz-es processzort tartalmazó számítógép, legalább 6 GB méretű Linux partícióval. Ez a konfiguráció lehetővé teszi, hogy virtuális gépen még egy Linux fusson, amit szerverként konfigurálunk.

Mi a rendszergazda feladata?

A rendszergazda feladatainak felsorolása nem egyszerű, mert szinte minden ide tartozik, ami egy szervezeti egység informatikai rendszerének működtetéséhez szükséges. Például operációs rendszerek telepítése, beállítása, jogosultságok kiosztása, hálózat kialakítása, harc a vírusok és a behatolók ellen, elektronikus levelezés működtetése, csoportmunka lehetőségének biztosítása, biztonsági mentések, és még hosszasan sorolhatnák. Aki tehát erre a pályára lép, az valószínűleg élete végéig tanulni fog.

A szerver-kliens modell

A gyakorlatban alkalmazott vállalati, intézményi számítógépes hálózatok szerverekből és munkaállomásokból (kliensek) állnak. A szervereken tároljuk az adatokat, és ezeken futnak a különböző szolgáltatások. A kliensek a felhasználók gépei, amik igénybe veszik a szerverek különböző szolgáltatásait.

Gyakran használt szolgáltatások és protokollok

A szervereken különböző szolgáltatások (services) futnak, amelyek úgynevezett kapukon (port) keresztül kommunikálnak a megfelelő szabályzat (protokoll) szerint egymással, illetve a kliensekkel. Néhány gyakran használt szolgáltatás, a hozzátartozó protokollokkal:

Szolgáltatás	Protokoll	Leírás
WEB	http, https	Weboldalak, portálok böngészése
FTP	ftp	Fájlok másolása hálózaton keresztül
DNS	tcp, udp	Domain név szolgáltatás
E-MAIL	pop, smtp, imap	Elektronikus levelezés
MEGOSZTÁS	smb	Könyvtárak és fájlok megosztása a hálózaton
SSH	ssh2	Távoli gépek adminisztrációja
CÍMTÁR	ldap	Felhasználók központi nyilvántartása

A szervereken futó operációs rendszerek

A szervereken futhatnak speciális, csak erre a célra alkalmas (Novell, Windows szerver) vagy általános (Linux, ….BSD) operációs rendszerek. A szervereken leggyakrabban futtatott operációs rendszerek:

Megnevezés	Leírás
Linux	Szervernek és kliensnek is alkalmas, ingyenes Unix típusú
BSD	FreeBSD, NetBSD, OpenBSD, ingyenes Unix változatok
Novell	Novell, a jövőben várhatóan Linux kernelen is fut, fizetős
Windows	Szerver változat, grafikus felülettel, fizetős

A szervereken futó operációs rendszerek kiválasztásánál figyelembe kell venni a várható terhelést, a kívánt szolgáltatásokat, a biztonsági követelményeket, a vírusok és férgek elleni védettséget. Természetesen az ár sem közömbös.

Szerver-kliens környezet kialakítása

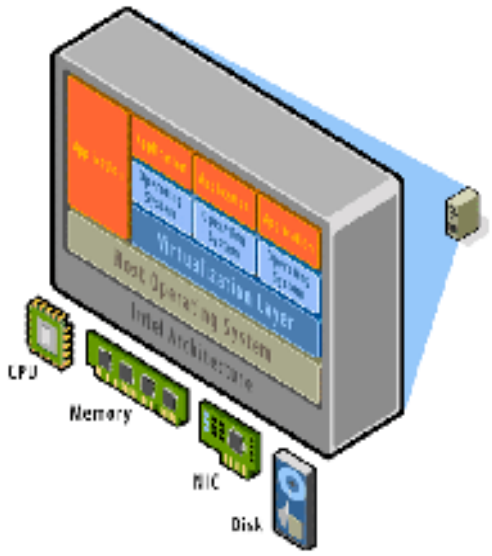
A rendszergazdai ismereteket természetesen nem lehet csak könyvekből, leírásokból megszerezni: a puding próbája az evés. Szükség van tehát olyan környezetre, ahol a szerveren beüzemelt szolgáltatásokat kliens oldalról is ki lehet próbálni.

Ha két gép áll rendelkezésre, akkor teljesen életszerű környezet alakítható ki. Egyik lesz a szerver, másik a kliens. A gépeket hálózatba kell kapcsolni, mindkettőre Linuxot telepíteni, és már kezdődhet is a kísérletezés.

Ha csak egy gép van, akkor a legolcsóbb megoldás kihasználni azt, hogy a Linux multiuser típusú operációs rendszer. Egyik konzolon adminisztráljuk a szervert, egy másik konzolon pedig felhasználóként kipróbáljuk a szolgáltatásokat. Ez a megoldás jó, de nem elégé életszerű.

Talán a legjobb egygépes megoldás egy olyan program alkalmazása, amely lehetővé teszi egyidejűleg két, vagy több operációs rendszer futtatását. Ilyen program például a fizetős vmware vagy az ingyenes bochs, amely virtuális számítógép(ek) emulálására alkalmas program.

A **vmware** vagy **bochs** programok segítségével tehát egy számítógépből csinálhatunk kettőt, vagy többet.



Ábra 1

Miután egy gépen több operációs rendszer fog futni, ezért célszerű ezeket jól láthatóan megkülönböztetni egymástól, például a gépnevek megfelelő megválasztásával.

Általában a virtuális gépen futó operációs rendszer lesz a szerver, ezért adjuk ennek a gépnek a **szerver** nevet.

A host operációs rendszerünk (amit magára a számítógépre telepítettünk) neve már közömbös, a könyvben azonban a **kliens** névre fog hallgatni.

Ha több operációs rendszert futtatunk egy hardveren, akkor ezek természetesen megosztoznak a hardver erőforrásokon. Időosztásos rendszerben használják a processzort, elosztják egymás között a fizikai memóriát, közösen kezelik a perifériákat.

A minimálisan szükséges számítógép konfiguráció, amely szükséges egyszerre két Linux operációs rendszer elfogadható sebességű működéséhez egy számítógépen:

Processzor órajel minimum 800 MHz (ajánlott legalább 2 Ghz)
Memória minimum 256 MB RAM (ajánlott 512 MB vagy több)
Merevlemez partíció minimum 6 GB (ajánlott legalább 10 GB)

Ennél szerényebb „vas” esetén is működik a rendszer, de a sebességgel problémák adódhatnak. A kliensen, illetve a szerveren is ajánlott a kis erőforrást igénylő ablakkezelők használata (pl. Blackbox, IceWM, Xfce), esetleg a grafikus felületet kikapcsolása a szerveren.

A virtuális gép képes teljes értékű szereplőjévé válni a hálózatunknak, emulált hálózati kártyáján kommunikál az alaprendszerrel, és a többi hálózati szereplővel. A vmware telepítésével és beállításával kapcsolatos tudnivalókat a következő fejezetben tárgyaljuk.

A vmware telepítése és konfigurálása

Telepítés

A **vm**ware beszerzése UHU-Linux használata esetén nagyon egyszerű:

- a második telepítő CD-n van hozzá UHU csomag,
- az ftp://ftp.uhulinux.hu helyről apt-get program segítségével telepíthető.

A vmware telepítésének menete Internetről:

```
root@kliens:~# apt-get update
root@kliens:~# apt-get install vmware
```

Vmware telepítése, ha a csomagot előbb letöltjük, vagy CD-n áll rendelkezésre:

```
root@kliens:/ahol_van_a_csomag# dpkg -i csomagnév
```

Vmware konfigurálás

Telepítése után konfigurálnunk kell a programot, a ***vmware-config.pl*** szkript rendszeradminisztrátorként (root) való futtatásával:

1. Indítsuk el a konfigurálásra szolgáló szkriptet root-ként, és válaszoljuk a feltett kérdésekre az utasításoknak megfelelően. A kérdésekre adott válaszok után ENTER billentyű leütésével léphetünk tovább.

```
root@kliens:~# vmware-config.pl
```

2. Fogadjuk el a felhasználói szerződést. A **vm**ware nem ingyenes szoftver, de 30 napig térítés mentesen kipróbálható. Weboldalán kérhetünk magunknak hozzá ideiglenes kódot. A licence szerződésből kilépni a **Q** billentyű leütésével lehet.

3. Fogadjuk el a felhasználói szerződést **y** begépelésével.

4. Megkérdezi a konfiguráló program, hogy legyen-e hálózat? Fogadjuk el az alapértelmezett értéket [yes]. Engedélyezzük a NAT alkalmazását? Igen [yes], így a virtuális gép ki fog látni arra a hálózatra. Kipróbáljuk a hálózatot? Igen [yes].

5. A hálózatot sikeresen kipróbálása után további információkat kapunk a DHCP szerverről.

6. Megkérdezi a konfiguráló szkript, hogy olyan hálózatot akarunk-e, ahol csak az alaprendszerünkkel (kliens) tudunk kommunikálni? Nem [no].
   El akarjuk-e érni a kliens fájlrendszerét? Nem, begépeljük: no. Később ezt a problémát majd megoldjuk.

7. A konfiguráló elvégzi a beállításokat, majd jó szórakozást kíván.

Ezzel a vmware konfigurálását befejeztük. Jelen beállítás a virtuális gépre telepítendő operációs rendszereknek lehetővé teszi, hogy a hálózat teljes értékű szereplőjévé váljanak. Ez azt jelenti, hogy például a valódi munkaállomásokhoz hasonlóan elérik az Internetet, ha megadjuk számukra az adott szegmensen érvényes alapértelmezett átjáró IP számát.

A vmware virtuális gép indítása

A **vm**ware telepítése után az **Alkalmazások** > **Eszközök** > **Emulátorok** paranccsal felhasználóként indítandó, GNOME, vagy KDE grafikus munkakörnyezetet esetén.

Más ablakkezelők alkalmazásakor kattintás jobb-gombbal az asztalon, majd a helyi menüben **Eszközök** > **Emulátorok > vmware.**

A virtuális gép nem más, mint egy ablakban futó program, ami képes egy számítógép emulálására. Még virtuális BIOS, valamint virtuális video-vezérlő, hálózati kártya, stb. is rendelkezésünkre áll.

Új virtuális gép létrehozása

Mielőtt egy virtuális operációs rendszert telepítenénk, létre kell hoznunk számára egy olyan virtuális gépet, amely megfelelő hardver elemekkel rendelkezik. Ne feledjük, hogy a virtuális gépre telepített, illetve a virtuális gépen futó operációs rendszerünk megosztozik majd a valódi hardver összetevőkön.

Ha például a gépünkben 512 MB RAM van, akkor a virtuális gépnek adhatunk belőle 256 MB RAM-ot, így az alaprendszerünk memóriája is 256 MB-ra csökken.

Annak sincs akadálya, hogy kettőnél több operációs rendszert futtassunk a számítógépünkön, ha ezt a rendelkezésre álló hardver elemek lehetővé teszik.

Az új virtuális gép létrehozásának lépései:

1. Új virtuális gép létrehozását **Ctrl+N** billentyű-kombinációval indítjuk, a megjelenő ablakban a **Custom** lehetőséget választjuk, majd **Next >**

2. A következő lépésben vmware 4.5, illetve UHU-Linux 1.2 verziókat feltételezve a Guest operating system Lin**Linux**, a **Version** listában pedig **Other Linux 2.6.x kernel**, majd **Next >**

3. A virtuális gép neve legyen **UHU-Linux-szerver**, hagyjuk jóvá a telepítő könyvtár helyét és nevét, **Next>** A virtuális operációs rendszerünket tartalmazó fájlok saját könyvtárunkon belül a program által létrehozott könyvtárban lesznek elhelyezve.

4. Állítsuk be a virtuális gép által használható memóriát, legyen legalább 128 MB. Ha több operációs rendszert futtatunk egy számítógépen, akkor a virtuális gép gondoskodik a fizikai memória megosztásáról. Az host operációs rendszernek legalább akkora memóriát hagyjunk, hogy futása elfogadható sebességű legyen, ne kelljen sokat használni a swap partíciót. Két grafikus operációs rendszer esetén 256 MB fizikai RAM alatt nem fogunk elfogadható teljesítményt elérni, **Next>**

5. Virtuális gépünk saját hálózati kapcsolatokat állít be magának, így tud kommunikálni a host operációs rendszerrel, és ezen keresztül annak hálózati kapcsolataival. Válasszuk a **Use bridge networking** lehetőséget (ez az alapértelmezett), **Next >.** Virtuális gépünk ebben az esetben nemcsak a host operációs rendszerrel, hanem az ahhoz kapcsolódó hálózattal is tud majd kommunikálni.

6. Az adaptertípusoknál általában nem módosítunk, **Next >.**

7. A következő lépésben válasszuk a **Create a new virtual disk** lehetőséget, majd **Next >**. Ezzel néhány fájlt hozunk létre, amely tartalmazni fogja a virtuális gépre telepített operációs rendszerünk teljes fájlrendszerét. A második lehetőség **(*Use an exiting virtual disk*)** választásával egy korábbi virtuális operációs rendszer vehetünk használatba, a harmadik változat **(*Use a physical disk*)** pedig egy merevlemezre telepített operációs rendszer futtatását teszi lehetővé virtuális gép segítségével.

8. Fogadjuk el a rendszer által ajánlott virtuális disk típusát, **Next >.**

9. Adjuk meg a virtuális fájlrendszerünk maximális méretét, az alapbeállítás 4 GB. Ez persze nem azt jelenti, hogy ezek a fájlok mindig ilyen nagyhttp://www.webtar.hu/ méretűek. Tényleges méretük attól függ, hogy mennyi adat van a virtuális fájlrendszeren. Nem érdemes tehát az alapértelmezett értéket csökkentenünk, akár növelhetjük is, **Next >**

10. Végül megkérdezi a varázsló, hogy mi legyen a virtuális fájlrendszert tartalmazó fájlok neve, a felajánlottat hagyjuk jóvá, **Finish**.

Létrehoztuk azt a virtuális gépet, amelyre leendő szerverünket, az UHU-Linuxot telepíteni fogjuk.

Az UHU-Linux telepítése virtuális gépre

A **vm**ware programmal létrehozott virtuális gépekre az operációs rendszerek telepítését telepítő CD-ről, vagy **.iso** fájlból egyaránt végezhetjük.

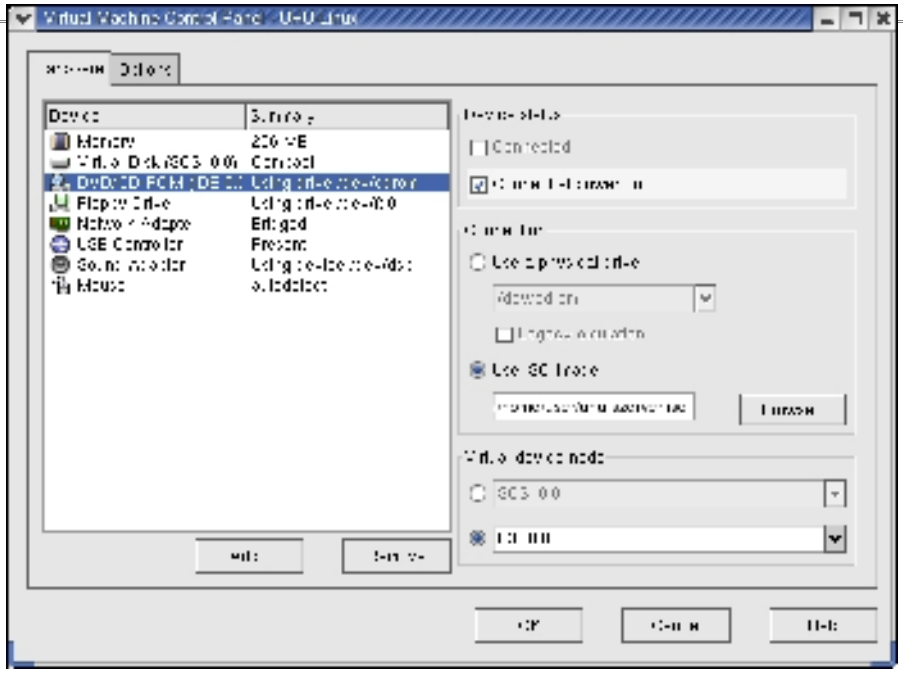
Az UHU-Linux **.iso** fájlokat ingyenesen letölthetjük az ftp://ftp.uhulinux.hu helyről, de saját magunk is előállíthatjuk a **dd** parancs segítségével, ha telepítő CD-vel már rendelkezünk. Akinek módjában áll, az természetesen megvásárolhatja az UHU-Linux operációs rendszert itt: https://bolt.uhulinux.hu/

A **dd** parancs bemenete (input file) a cdrom, kimenete (output file) padig az aktuális könyvtárban létrehozandó **UHU-Linux-1.2-cd1.iso** lesz. Álljunk a saját könyvtárunkba, helyezzük be az UHU-Linux CD1 telepítő CD-t a meghajtóba, és adjuk ki a következő parancsot felhasználóként:

```
user@kliens:~$ dd if=/dev/cdrom of=UHU-Linux-1.2-cd1.iso
```

A virtuális gépen ezután be kell állítanunk, hogy a bootolás az **.iso** fájlról történjen:

VM > Settings > DVD/CD-ROM > Use ISO image > Browse > .iso keresése > OK



Ábra 2

A virtuális operációs rendszer telepítése

1. Indítsuk el a telepítést a **Start this virtual machine** gombra kattintva. Lezajlik a boot folyamat, a nyitókép (bagolypöttyözés) után válasszuk a **Telepítés** lehetőséget.
Térjünk át teljes képernyős módba az **Alt+Ctrl-F8** billentyűkombinációval, így látni fogjuk a továbblépéshez szükséges gombokat. A telepítés részletes leírását az **UHU-Linux Felhasználói kézikönyve** tartalmazza, letölthető például innen: http://linux.tux.hu/

2. A **MEHET** gombra kattintással fogadjuk el a licenc szerződést.

3. Virtuális gépre telepítéskor a particionálásnak nincs értelme, válasszuk **A teljes merevlemez felhasználása** lehetőséget. A telepítő megkérdezi a virtuális memória (swap partíció) méretét, hagyjuk jóvá a 256 MB alapértelmezett értéket. Azért nincs értelme az egyéni particionálásnak, mert a virtuális gép az egész fájlrendszerét néhány fájlban tárolja, melynek mérete a ténylegesen tárolt állományok nagyságától függ.

11. Megkérdezi, hogy akarjuk-e ellenőrizni a telepítő CD-t : **Nem.**

12. Nem fogjuk telepíteni a **KDE** csomagjait, a **Gnome** csomagokat (kivéve gdm), az **Összes csomag > Alkalmazások > Iroda** szekcióból pedig kiszedünk minden számunkra felesleges összetevőt.
OK, elindul a telepítés.

13. A **GRUB** telepítése az első merevlemez Master Boot Record területére.

14. Adjuk meg a rendszergazda (root) jelszavát, majd ismételjük meg. Az egyszerűség kedvéért az összes jelszavunk például **kamion** lehet. A jelszónak legalább hat karakterből kell állnia.

15. A következő képernyőn hozzunk létre a **Hozzáadás** gombbal felhasználót saját tetszőleges bejelentkező (login) nevünkkel, ennek jelszava is legyen **kamion**.

16. A grafikus felület felbontását állítsuk be 800x600-ra, vagy 1024x768-re. Teszteljük a beállítást. Ha a teszt nem sikerül a felkínált video-vezérlővel, akkor válasszunk a hardvernek megfelelőt. Ha ez sem vezetne eredményre, akkor a VESA mód beállítása rendszerint segít.

17. A **Juhhu!** gombra kattintva a telepítés véget ér, a virtuális gépünk újra indul. Megjelenik a grafikus felület. Állítsuk át a grafikus munkakörnyezetet, legyen Blackbox. Jelentkezzünk be felhasználóként saját azonosítónkkal. Ezzel a telepítést befejeztük, a következő fejezetben a konfigurálásról lesz szó.

FIGYELEM!

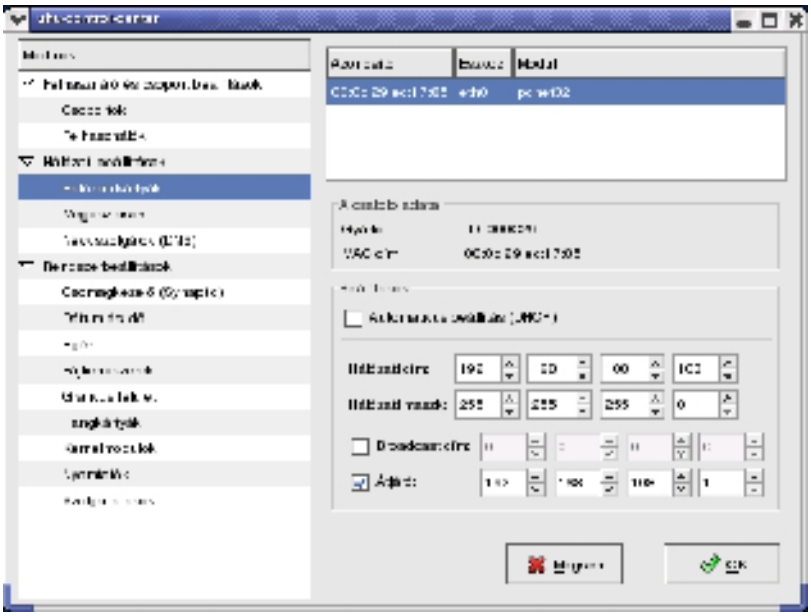
Virtuális gépünkből az **Alt+Ctrl** billentyűk együttes lenyomásával tudunk kilépni. Visszalépni a virtuális gépbe az ablakára való rákattintással lehet.

Az UHU-Linux szerver konfigurálása

Továbbiakban a virtuális gépre telepített UHU-Linux operációs rendszert szervernek fogjuk nevezni. Az alapvető beállítások elvégzéséhez először az UHU-vezérlőpultot indítjuk el. Blackbox grafikus munkakörnyezet használatát feltételezve:

Jobb gombbal kattintás az asztalon > Beállítások > UHU Vezérlőpult

A hálózati kártya beállítása



Ábra 3

A virtuális gép saját hálózati kártyát emulál, ezzel ugyanúgy kell bánnunk, mintha igazi hálózati kártya lenne. Lényeges, hogy a virtuális gépen futó szerver ugyanazon az alhálózaton legyen, mint a host szerepét betöltő kliens. A virtuális gépen futó **szerver** IP címe legyen **192.168.1.200.** Az alapértelmezett átjáró **192.168.1.1** ugyanaz, mint a kliensé. Ezen keresztül a szerver is eléri az Internetet. Ha nincs Internet csatlakozásunk, akkor alapértelmezett átjárót nem kell beállítanunk. A broadcast cím megadása nem kötelező, alapértelmezett értéke ebben a beállításban 192.168.1.255 lesz, ami a legnagyobb kioszthatóIP cím az adott tartományban.

FIGYELEM! Az **OK** gomb megnyomása kötelező a beállítások véglegesítéséhez!

Beállítási módok

Linuxban általában háromféle módon lehet valamilyen jellemzőt beállítani:

- beállítóprogram vagy varázsló segítségével: ez sokszor grafikus felületen fut, ilyen az UHU Vezérlőpult,
- parancsok segítségével: az IP címet és átjárót az **ifconfig**, valamint a **route** parancsokkal lehet beállítani,
- konfigurációs állomány szerkesztésével: az /etc/sysconfig/network/ethernet könyvtárban vannak a hálózati kártyák MAC címének megfelelő alkönyvtárak, az ipv4 alkönyvtárban vannak a szerkesztendő állományok.

FEALDATOK

1. Állítsuk be parancsok segítségével a **szerver** IP címét! Legyen az IP cím 192.168.1.200, az alapértelmezett átjáró pedig 192.168.1.1.

```
root:~# ifconfig eth0 192.168.1.200
root:~# route add default gw 192.168.1.1
```

2. Állítsuk be a szerver IP címét a konfigurációs állományok szerkesztésével és a hálózati szolgáltatás újraindításával! A megadott értékek alkalmazkodjanak az éppen aktuális hálózathoz! (A … helyén a hálózati kártya MAC címének megfelelő könyvtárat kell be–helyettesíteni)

```
root:~#            cat            /etc/sysconfig/network/ethernet/.../ipv4/address
192.168.1.200
CTRL+d
root:~#            cat            /etc/sysconfig/network/ethernet/.../ipv4/netmask
255.255.255.0
CTRL+d
root:~#            cat            /etc/sysconfig/network/ethernet/.../ipv4/gateway
192.168.1.1
CTRL+d

root:~# service network restart
Hálózati eszköz leállítása: eth0...                        [ OK ]
Hálózati eszköz leállítása: lo...                          [ OK ]
Hálózati visszacsatolás aktiválása...                      [ OK ]
Ethernet hálózati csatoló aktiválása: eth0...              [ OK ]
```

3. Ellenőrizzük a kliens és a szerver közötti hálózati kapcsolatot!

A kliens operációs rendszerben adjuk ki a ping parancsot (csak egy csomag legyen):

```
user@kliens:~$ ping -c 1 192.168.1.200
PING 192.168.1.200 (192.168.1.200) 56(84) bytes of data.
64 bytes from 192.168.108.1: icmp_seq=0 ttl=64 time=0.424 ms
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

A névfeloldás biztosítása

A szerverünkön fix IP címet állítottunk be, tehát nem DHCP szolgáltatáson keresztül kap IP címet. Ebben az esetben a névfeloldás biztosítása érdekében egy, vagy több névszervert is meg kell adnunk. Elvileg bármilyen elérhető névszerver megadható. Lekérdezhetjük a szolgáltatónk névszervereit:

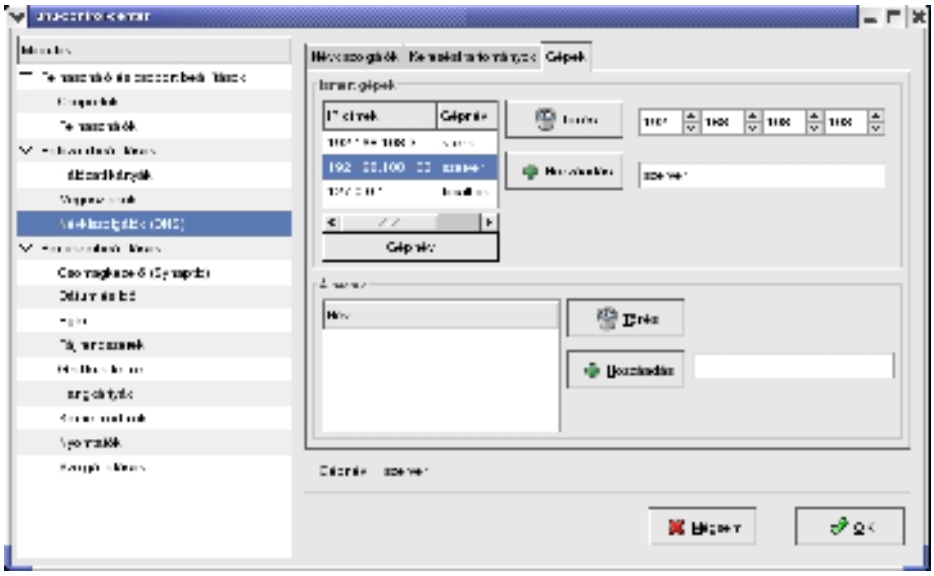
```
melegak:~$ host -t ns chello.hu
chello.hu name server ns02.chello.com.
chello.hu name server ns2.chello.at.

melegak:~$ nslookup ns02.chello.hu
Non-authoritative answer:
Name:   ns02.chello.hu
Address: 195.34.131.180
```

Az UHU Vezérlőközpontban, a **Névkiszolgálók (DNS)** hálózati beállítás kiválasztása után a **Névkiszolgálók** panelen az IP cím be–gépelése után kattintsunk a **Hozzáadás** gombra.

A névszerverek bekerülnek az **/etc/resolv.conf** állományba, amit utána „kézzel” is szerkeszthetünk.

A **Gépek** fülre kattintva adjuk meg a szerver IP címét: **192.168.1.200** és gépnevét: **szerver**, majd kattintsunk a **Gépnév** gombra. A gépnév megváltoztatása után indítsuk újra a szervert!



Ábra 4

A szerver újraindítása után bejelentkezve egy konzolra így alakul a prompt:

```
root@szerver:~#
```

A szerver és a kliens ajánlott beállításai a tesztkörnyezet probléma mentes működéséhez :

Az alhálózat adatai:

Az alhálózat IP címe: 192.168.1.0
Alhálózati maszk: 255.255.255.0
Alapértelmezett átjáró IP címe: 192.168.1.1
Névszerver: 195.34.131.180

A kliens adatai:

Gépnév: kliens
IP cím: 192.168.1.2
Alhálózati maszk: 255.255.255.0
Alapértelmezett átjáró és névszerver IP címe: 192.168.1.1
Névszerver: 195.34.131.180

A szerver adatai:

Gépnév: szerver
IP cím: 192.168.1.200
Alhálózati maszk: 255.255.255.0
Alapértelmezett átjáró és névszerver IP címe: 192.168.1.1195.34.131.180
Névszerver: 195.34.131.180

FELADAT

1. Listázzuk ki a névszervereket tartalmazó állományt a szerveren!

```
user@szerver:~$ cat /etc/resolv.conf

# /etc/resolv.conf
# Ez egy generalt file, a dns-config felulirhatja.

Nameserver 195.34.131.180
```

Váltás a operációs rendszerek között

A továbbiakban azt az esetet tárgyaljuk, amikor két operációs rendszerrel dolgozni, de ezek egy számítógépen futnak. Host gépünkön az alaprendszerünk fut, ez lesz a kliens, virtuális gépen futó operációs rendszerünk pedig a **szerver**.

Fontos mindig tudnunk, hogy éppen melyik gépen is dolgozunk. Például ha a kliensen Xfce, a szerveren Blackbox ablakkezelő fut, akkor nem lesz problémánk. Ha konzolon dolgozunk, akkor a prompt-ban megjelenő gépnév alapján tudunk tájékozódni.

Alt+Ctrl+F1 … F6

Így érjük el a **kliens** karakteres konzoljait. A virtuális gépen futó szerver karakteres konzoljait természetesen ezekkel a billentyűkombinációkkal nem érhetjük el.

Alt+Ctrl+F7

A számítógép bootolása után ezen a konzolon érhető el **kliens** a kliens grafikus felületén a bejelentkezésre szolgáló program, a **gdm**. Itt indítjuk el a **vm**ware virtuális gépünket, és azon belül a virtuális szerverünket.

Alt+Ctrl+F8

Itt érhetjük el legegyszerűbben a virtuális gépen futó **szervert**, teljes képernyős módban. Javasolt a szerveren Blackbox grafikus környezet futtatása, kis erőforrásigénye miatt.

Tanácsok szerver-kliens környezet használatához

1. Gyengébb hardver esetén kliens rendszerünkön is futtassunk kis erőforrás igényű ablakkezelőt (pl. Xfce).

2. Ha mégis két számítógépünk lenne, akkor az egyikre telepíthetjük a szervert, kliensként pedig használhatjuk például az UHU-Linux Live-CD változatát is. Ahogyan nevéből is kiderül, ez egy CD-ROM -ra telepített rendszer, bootolás után azonnal használható.

AzUHU-Linux Live-CD letöltési helye .iso formában: **http://www.linuxforum.hu**

3. A virtuális gépre feltelepített szerverünkről készítsünk biztonsági másolatot. Lépjünk be abba a könyvtárba, amit telepítéskor megadtunk, és másoljuk le az ott lévő fájlokat biztonságos helyre. Ha bármilyen probléma adódik, ezek visszamásolásával ismét kifogástalan virtuális szerverünk lesz.

4. Ha a virtuális gépre telepített szerveren már fut az ssh szolgáltatás, akkor lehet ezen keresztül is adminisztrálni. Engedélyezhetjük ssh fölött az X továbbítást is, ilyenkor még a szerveren futó grafikus programok is megjeleníthetők a kliensen.

Összegzés, kérdések, feladatok

Az előző részben megismertük a virtuális gépet, foglalkoztunk annak telepítésével és konfigurálásával. Telepítettük virtuális gépre az UHU-Linux operációs rendszert, lehetővé téve, hogy egy gépen is vizsgálni lehessen az általában két számítógépes környezetet igénylő szerver-kliens modellt.

KÉRDÉSEK

1. Mit nevezünk virtuális gépnek?
2. Ábrázolja a virtuális gép és az alap operációs rendszer kapcsolatát!
3. Mit tud a virtuális gép beállításairól?
4. Ismertesse az Ön által használt számítógépes szolgáltatásokat!
5. Mit nevezünk a számítástechnikában protokollnak? Soroljon fel néhányat!
6. Mutassa be a szerver-kliens modellt!
7. Milyen lépésekből áll az UHU-Linux telepítése?
8. Ismertesse az UHU Vezérlőpult lehetőségeit!
9. Milyen parancsokat használna a hálózat ellenőrzésére?
10. Ismertesse az **ifconfig** és a **route** parancsok használatát!

FELADATOK

1. Keressen a **vm**ware programhoz hasonló virtuális gép programokat, különböző operációs rendszerekhez, és adjon meg a webol–dalukra mutató linkeket!
2. Mentse el a virtuális gépre telepített UHU-Linux-hoz tartozó fájlokat! Mekkora helyet foglalnak el összesen ezek a fájlok?
3. Változtassa meg a szerver IP számát, és az alapértelmezett átjárót parancssorból! Ellenőrizze a hálózat működőképességét!
4. Váltson munkafelületet a megfelelő billentyű kombinációkkal a következő sorrendben: **kliens** konzol, **szerver** teljes képernyős grafikus felület, **kliens** teljes képernyős grafikus felület!

Könyvtár és fájlmegosztás: Samba

Egy fájlszerverrel kapcsolatban követelmény, hogy a felhasználók az ott elhelyezkedő könyvtáraikat valamilyen módon elérjék. Ennek egyik módja a Samba használata. Előnye, hogy segítségével más operációs rendszerekkel (pl. Windows) is kapcsolatot teremthetünk.

Mi a Samba?

Samba olyan program, amely elérhetővé teszi a kliens gépek számára a kiszolgálón megosztott mappákat, fájlokat és nyomtatókat, mégpedig az **SMB** (Session Message Block) protokoll segítségével.

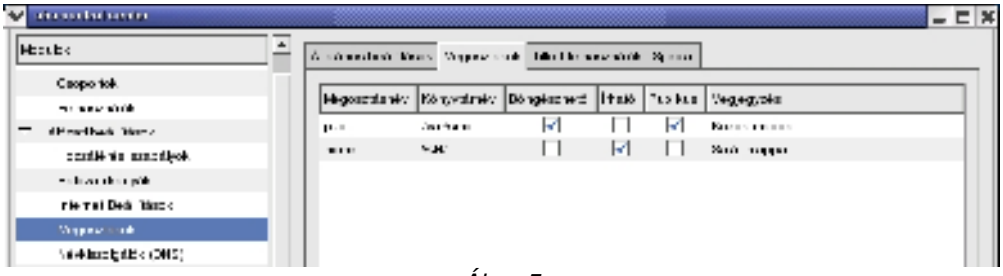
A Samba működéséhez szükséges a **smbd** (SMB démon vagy SMB-szerver) és az **nmbd** (NetBIOS névkiszolgáló) futtatása. A szerveren megosztunk valamilyen néven egy könyvtárat (vagy nyomtatót), amit elérhetővé teszünk a hálózaton a kliens gépek számára. A gépek közötti kapcsolat adminisztrálása gépnevek alapján történik, ezt a NetBIOS névkiszolgáló felügyeli. Egy munkacsoportban egy nevet csak egy gépnek lehet kiosztani.

A Samba konfigurálását az /etc/samba/**smb.conf** fájl szerkesztésével végezhetjük el, de bizonyos beállításokat az **UHU Vezérlőpult > Megosztások** parancsával is elvégezhetünk. UHU-Linux operációs rendszeren a Samba az alaprendszerrel együtt települ.

Kliens oldalon sokféle grafikus megoldás mellett Linux operációs rendszerben rendelkezésünkre áll a parancssori **smbclient** program, illetve a megosztott könyvtárakat fájlrendszerünkhöz csatolhatjuk a **mount** parancs segítségével.

A megosztások beállítása

A szerveren indítsuk el az UHU Vezérlőpultot, és válasszuk a **Megosztások** parancsot. Kattintsunk a **Megosztások** fülre, majd a **Módosítás** gombra, és végezzük el a 22. ábrán látható változtatásokat:



Ábra 5

A beállítás következtében lesz egy **pub** nevű megosztás, amely mindenki számára olvashatóvá teszi a **/var/smb** könyvtárat. Hozzunk létre ebben a könyvtárban egy szövegfájlt **samba.txt** néven, és írjuk bele: „Ez a pub megosztás”.

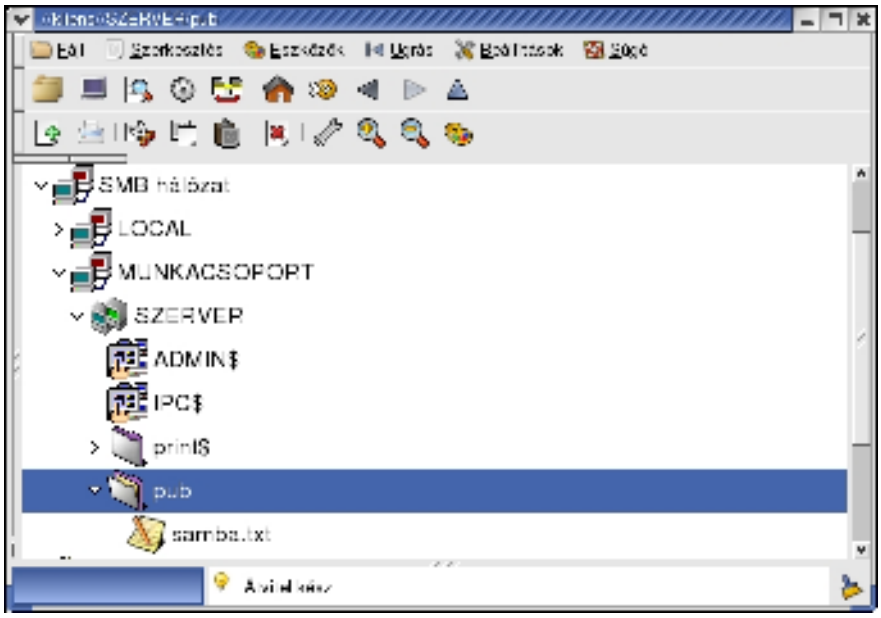
A beállítások után az UHU-vezérlőpult segítségével indítsuk el az **smbd** és az **nmbd** szolgáltatásokat. Ellenőrizzük egy konzolon a Samba helyes működését a következő paranccsal:

```
user@kliens:~$ smbclient -L szerver
Password:
Domain=[MUNKACSOPORT] OS=[Unix] Server=[Samba 3.0.2a]

        Sharename      Type      Comment
        ---------      ----      -------
        print$         Disk
        pub            Disk      Közös mappa
        IPC$           IPC       IPC Service (szerver)
        ADMIN$         IPC       IPC Service (szerver)
```

Megosztás elérése a kliens grafikus felületéről

Váltsunk a **kliens** grafikus felületére, majd a menüben (jobb gomb) az **Eszközök > Fájlkezelők > xffm** parancs kiadásával keressük meg a publikus megosztást:



Ábra 6

Megosztás elérése a parancssorból

Sokszor lehet szükségünk arra, hogy konzolból is be tudjunk jelentkezni a samba szerverre.

A **kliens** operációs rendszerből így léphetünk be a samba szerverre a **smbclient** parancs segítségével:

```
user@kliens:~$ smbclient //IPcím_vagy_netBIOSnév/megosztásnév
```

Esetünkben a bejelentkezés parancssorból a következő módon történik:

```
user@kliens:~$ smbclient //szerver/pub
```

Mivel publikus megosztásról van szó, ezért a jelszót nem kell begépelni, ENTER elég. Megjelenik a **smb: \>** alakú prompt, itt a **help** paranccsal megnézhetjük, hogy milyen további parancsok kiadására van lehetőségünk. A **dir** parancsot kiadva látjuk, hogy a samba.txt fájl is megvan. A samba konzolból kijelentkezni az **exit**, vagy **quit** parancs segítségével tudunk.

Megosztott könyvtár csatolása a fájlrendszerhez

Ha a szerveren megosztott könyvtárhoz csak a smbclient programmal férnénk hozzá, az elég kényelmetlen lenne. Előnyösebb, ha a **mount** parancs segítségével hozzácsatoljuk azt a saját fájlrendszerünkhöz. Ezt a műveletet rootként végezhetjük, formátuma a következő:

```
root@kliens:~# mount -t típus //szerver/megosztás /célkönyvtár
```

vagy

```
root@kliens:~# smbmount //szerver/megosztás /célkönyvtár
```

A kliensen előbb root-ként készítsünk egy ***/mnt/pub*** mappát, majd csatoljuk fel a **pub** megosztást:

```
root@kliens:~# mount -t smbfs //szerver/pub /mnt/pub
```

vagy

```
root@kliens:~# smbmount //szerver/pub /mnt/pub
```

Jelszót nem kell megadnunk, ENTER leütése elég. A végén **ls** paranccsal listázzuk a ***/mnt/pub*** könyvtár tartalmát, ellenőrizve a **pub** megosztás felcsatolódásának tényét.

A felcsatolt megosztás leválasztása a fájlrendszerünkről az alábbi módon történik:

```
root@kliens:~# umount /mnt/pub
```

A Samba szerver finomhangolása

Bármilyen kényelmes is az UHU Vezérlőpult használata, a samba finomhangolását a konfigurációs állomány szerkesztésével lehet a legjobban megoldani. Használhatjuk ehhez például a **vi**, vagy az **mcedit** szövegszerkesztőt. Adjuk ki a szerveren rootként a következő parancsot:

```
root@szerver:~# vi /etc/samba/smb.conf
```

A smb.conf állomány

Ez a fájl különböző szekciókból áll. A [global] szekcióban az egész Samba szolgáltatásra vonatkozó beállításokat adunk meg. A [home] szekció a felhasználók szerveren létrehozott, saját, megosztott könyvtárára vonatkozó szabályokat tartalmazza. A [netlogon] szekció csak Windowsos kliensek esetén szükséges. Új megosztás létrehozásakor külön szekcióba kerülnek annak szabályai.

A **smb.conf** állományra egy gyakorlatban is használható példa:

```
;[GLOBAL]--------------------------------------------------------
```

```
; Az /etc/samba/smb.conf állományban szekciók vannak,
; a [global] szekció beállításai a teljes kiszolgálóra érvényesek,
; de egy részük később felülbírálható

[global]

; A csoport vagy tartomány neve, ahol a kiszolgáló üzemel
workgroup = MUNKACSOPORT

; A Samba kiszolgáló neve, pillanatnyilag a hosztnév
server string = %h

; Hálózati interfészek, amire a Samba válaszol
interfaces = lo, eth*

; Bejelentkezési mód, lehetőségek: share | domain | server
; User megadása esetén a felhasználók azonosítása smbpasswd  alapján történik
security = user

; Jelszótitkosítás engedélyezése, hiányában Windows rendszert futtató gépekről
; nem lehet bejelentkezni. A kiszolgálón a smbpasswd nevű állományban szerepelnek
; a felhasználók jelszavai, amiket a smbpasswd paranccsal lehet létrehozni
encrypt passwords = yes

; Akiknek nem engedélyezett a megosztások elérése
invalid users = root

; A megosztások elérése csak az itt felsorolt gépeknek engedélyezett (alhálózat)
hosts allow = 192.168.1. localhost

; Amelyik IP címről nem érhetőek el a megosztások.
hosts deny = 192.168.1.1

; Legyen a szerver tartomány-főtallózó? Lehetőségek: auto | yes | no
domain master = auto

; A hálózatra kapcsolódásakor versenyezzen a szerver, hogy főtallózóvá váljon?
preferred master = yes

; A kiszolgáló nyerési esélyei a főtallózó címért folytatott versenyben
os level = 255
```

```
; A gépek NetBIOS nevét a DNS-ben is megpróbáljuk feloldani?
dns proxy = no

; A gépnevek visszafejtési sorrendje
name resolve order = lmhosts host wins bcast

; Naplózási szint (3-as vagy nagyobb érték lassíthatja a rendszert)
log level = 2

; Felhasználókénti logolás, használhatók a %-os változók
log file = /var/log/samba/log.%m

; A log-fájl mérete [kb],az érték elérésekor új naplófájl kezdődik. Az aktuális
; naplófájl .old kiterjesztésel felülírja a korábbi, ilyen nevű fájlt.
max log size = 10000

; Írjon a Samba a rendszernaplóba? Igen=1, nem=0
syslog = 0

; Nyomtatási típusok: bsd, cups, sysv, hpux, aix, qnx, plp, lprng
printing = cups

; Kis méretű csomagok küldésének beállítása
socket options = TCP_NODELAY

; Egyéb beállítások
map to guest = Bad Password
winbind separator = /

;[HOME]------------------------------------------------------------

; A [home] szekcióban bejegyzései a felhasználók saját könyvtárára vonatkoznak.
[home]
comment = Home mappa megosztas

; A home könyvtár elérési útja
path = %H/

; Hozzáférés: nem publikus, mások nem böngészhetik,tulajdonos részére írható
public = no
browseable =no
writeable = yes

; Új fájlokhoz, illetve könyvtárakhoz  maximálisan megadható engedélyek
create mask = 0600
directory mask = 0700

;[PUB]-------------------------------------------------------------

; Mindenki számára elérhető, írható megosztás
[pub]
comment = Kozos mappa

; A megosztott közös mappa elérési útja
path = /var/smb

; Hozzáférési jogok
public = yes
browseable =yes
writeable = no
```

```
;[NYOMTATÓK]--------------------------------------------------------

; A megosztott nyomtatók beállításai
[printers]
path = /tmp
printable = yes
public = no
browseable = no
writeable = no
create mode = 0700

[print$]
path = /etc/samba/drivers
write list = root

;[WIN KLIENSEK ESETÉN]-----------------------------------------

; A [netlogon] szekció csak Windows kliensek esetén szükséges
; A /tmp/netlogon könyvtárba tehetjük azokat a fájlokat (pl. Házirend),
; amelyeket a Windows kliensnek a bejelentkezés után le kell futtatni

[netlogon]
comment = Csak Windos kliensek eseten szukseges
path = /home/netlogon
public = no
browseable = no
writeable = no

; A felhasználók listája, akik bejelentkezhetnek egy megosztásra
; valid users = user1 user2

;[VÉGE]--------------------------------------------------------------
```

A smb.conf állományt a szerkesztés után mentsük el, és indítsuk újra a démonokat parancssorból:

```
root@szerver:~# service smbd restart
root@szerver:~# service nmbd restart
```

A démonokat újraindítása természetesen az **UHU Vezérlőpult** segítségével is megoldható.

FELADATOK

1. Csatoljuk fel a „pub” megosztást a **/mnt/pub** könyvtárba!
2. Listázzuk ki a **/mnt/pub** könyvtárat, majd csatoljuk le a fájlrendszerünkről a pub megosztást!

Felhasználók saját könyvtárának elérése a szerveren

A samba általában akkor használjuk, amikor egy linuxos szerverhez csatlakozva csatoljuk fel saját könyvtárukat a munkaállomásuk fájlrendszeréhez. Ehhez rendelkezniük kell a szerveren egy ugyanolyan felhasználói fiókkal, mint amilyennel a kliens gépen dolgozunk. A **smb.conf** állományban látható, hogy a root ki van tiltva a távoli bejelentkezésre hivatott felhasználók sorából.

Felhasználó létrehozása

A szerveren hozzunk létre egy felhasználót ugyanolyan bejelentkezési névvel, mint amilyennel a kliens gépre bejelentkeztünk. Ezt megtehetjük az UHU Vezérlőpult segítségével, vagy parancssorból is az alábbi módon:

```
root@szerver:~# useradd -m felhasználónév
```

Ezután megadjuk a felhasználó jelszavát (célszerű ugyanazt a jelszót megadni, mint amivel a kliens gépen jelentkeztünk be):

```
root@szerver:~# passwd felhasználónév
```

Mivel titkosított jelszót állítottunk be a smb.conf állományban, ezért a Samba részére is titkosítva kell megadnunk a jelszót a következő módon:

```
root@szerver:~# smbpasswd -a felhasználónév
```

Indítsuk újra az **smbd** és **nmbd** szolgáltatásokat, most parancssorból:

```
root@szerver:~# service smbd restart
root@szerver:~# service nmbd restart
```

A saját könyvtár elérése parancssorból

```
user@kliens:~$ smbclient //szerver/home -U felhasználónév
```

Saját könyvtár csatolása a fájlrendszerhez

A szerveren lévő saját könyvtárunkat többféle módon csatolhatjuk a kliens fájlrendszerhez. Szerkeszthetjük az /etc/fstab állományt, használhatjuk root jogokkal a smbmount, vagy a mount parancsokat, vagy a felhasználók számára is lehetővé tesszük fájlrendszerek felcsatolását például a sudo segítségével.

Az /etc/fstab szerkesztése

Ha az adott munkaállomást csak egy felhasználó dolgozik, akkor írjuk bele a következőhöz hasonló sort az /etc/fstab állományba a kliensen (a /mnt/samba könyvtárat természetesen létre kell előbb hozni):

```
//szerver/megosztas  /mnt/samba  smbfs  username=user,password=123456
```

Szükség esetén a még a következő opciókat is megadhatjuk a jelszó után:

```
uid=user,gid=users,ip=192.168.1.200,iocharset=iso8859-2,codepage=cp852
```

Saját könyvtár csatolása

A megosztások csatolását alapbeállításban csak a root teheti meg, például a felhasználó saját könyvtára esetében a következő módon:

```
root@kliens:~# smbmount //szerver/home /home/user/smb/ -o uid=user,gid=users,username=user
```

Az **uid** illetve **gid** opciók azt állítják be, hogy a csatolás milyen felhasználói, illetve csoportjogokkal valósuljon meg. Ha ezt nem adjuk meg, akkor könnyen lehet, hogy nem tudunk írni a felcsatolt könyvtárunkba.

Krusader

Használhatjuk a szerveren megosztott saját könyvtárunkat a ***Krusader*** fájlkezelő segítségével, szintén a root jelszó ismerete nélkül. Indítsuk el a programot a kliensen, majd válasszuk a **Parancsok > Új kapcsolat** menüpontot. A megjelenő ablakban állítsuk át a protokollt smb:// -re, és gépeljük be a megosztást: //szerver/home. A ***Kapcsolódás*** gombot megnyomva újabb ablak bukkan fel, ebben adjuk meg a felhasználói nevünket és a jelszót. A Krusader egyik ablakában a szerveren elhelyezkedő saját könyvtárunk jelenik meg.

Samba adminisztráció webes felületen

A samba webes adminisztrációját a ***swat*** program teszi lehetővé. Amennyiben rendelkezünk Internet hozzáféréssel, akkor így telepíthetjük fel a virtuális szerverünkre:

```
root@szerver:~# apt-get install swat
```

Távoli gépről (kliens) is akarunk adminisztrálni, ezért egészítsük ki a szerveren az ***/etc/xinetd.d/swat*** fájlban a „bind” sort a szerverünk IP számával (root-ként):

```
bind = 127.0.0.1 192.168.1.200
```

Mielőtt a swat elindulna, készítsünk másolatot az /etc/samba/smb.conf állományról, mert a swat ez a fájlt felül fogja írni.

A swat futtatásunkhoz szükséges még a xinetd szolgáltatás, indítsuk el a szerveren:

```
root@szerver:~# service xined start
```

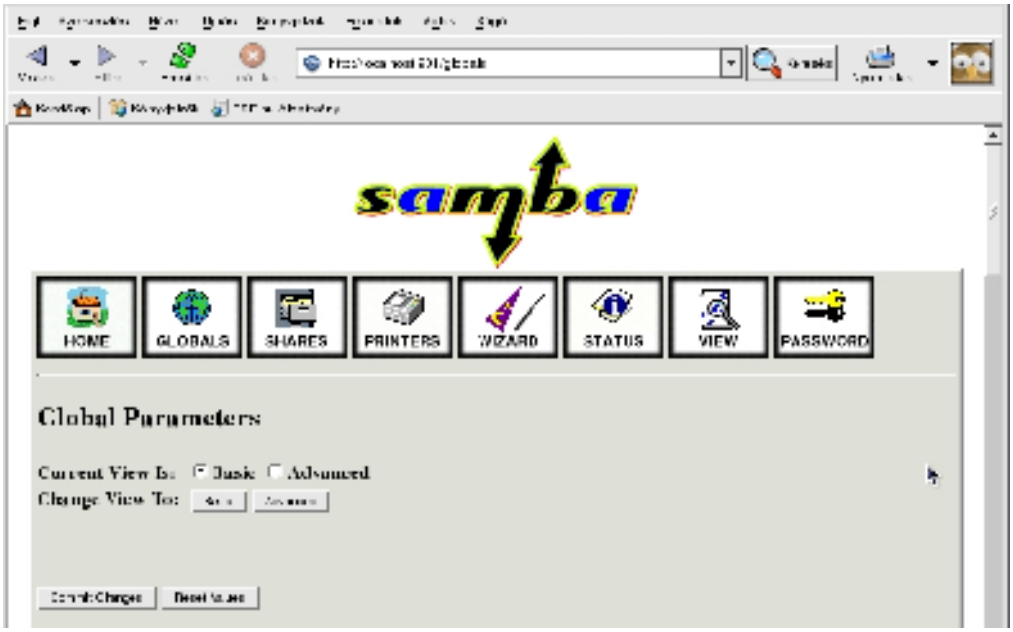
Ha a szerveren adminisztrálunk, akkor indítsunk el egy böngészőt, és gépeljük be:

```
http://localhost:901
```

Ha kliensen szeretnénk a sambát webesen adminisztrálni, akkor azt így lehet megoldani:

```
http://szerver_ip_száma:901
```

Felhasználó a root, adjuk meg a jelszavát és már adminisztrálhatunk is:



Ábra 7

KÉRDÉSEK

1. Milyen szolgáltatásokat nyújt a Samba?
2. Melyik démonok indítása szükséges a Samba működéséhez,
3. Hogyan kell a Samba működéséhez szükséges démonokat parancssorból elindítani?
4. Konzolon milyen paranccsal ellenőrizhetők az elérhető Samba megosztások?
5. Ismertesse a *smbclient* program használatát!
6. Hogyan lehet a megosztott könyvtárat fájlrendszerünkhöz csatolni?
7. Milyen paranccsal választható le a fájlrendszerünkről egy megosztott könyvtár?
8. Melyik állományt kell szerkesztenünk a Samba szerver beállításához?
9. Soroljon fel szekciókat a Samba konfigurációs állományából!
10. Mit jelentenek ezek a fogalmak: encrypt password = Yes, security = user?
11. Hogyan érdemes beállítani a *[home]* szekciót?
12. Milyen feltételei vannak a szerveren lévő saját könyvtárunk elérésének?
13. Mire használható a *smbpasswd* parancs?
14. Hogyan érhetjük grafikus felületről a szerveren megosztott saját könyvtárunkat?
15. Mi a *swat,* és hogyan kell üzembe helyezni?

Jogosultság átruházása

A samba szolgáltatásnál merül fel az a probléma, hogy fájlrendszert felcsatolni csak root jogokkal lehet. Az UHU-Linux operációs rendszerben persze egyes fájlrendszerek (CD-ROM, pendrive) automatikusan felcsatolódnak, de az smbfs nem. A smbmount parancsot tehát root jogokkal kellene futtatni a felhasználóknak.

A set-UID bit átállítása

Nem biztonságos, de használható megoldás, ha a **kliensen** a smbmnt parancs set-UID bitjét 1-re állítjuk. A smbmount - mivel a smbmnt-ra épül – már felhasználóként futtatva fel tudja majd csatolni a samba megosztásokat. Nézzük meg először, hogy hol található a smbmnt parancs:

```
root@kliens:~# which smbmnt
/usr/bin/smbmnt
```

Kérdezzük le a pillanatnyi hozzáférési jogokat (természetesen a kliensen):

```
root@kliens:~# ls -l /usr/bin/smbmnt
-rwxr-xr-x  1 root root 8020 2004-11-19 18:21 /usr/bin/smbmnt
```

Állítsuk át a set-UID bitet, és ellenőrizzük ismét a jogokat:

```
root@kliens:~# chmod 4755 /usr/bin/smbmnt

root@kliens:~# ls -l /usr/bin/smbmnt
-rwsr-xr-x  1 root root 8020 2004-11-19 18:21 /usr/bin/smbmnt
```

A set-UID bit módosítása után már felhasználóként is fel tudjuk csatolni a megosztást :

```
user@kliens:~$ mkdir ~/HOME
user@kliens:~$ smbmount //szerver/home ~/HOME -o username=[user],password=[titok]
```

Végül állítsuk vissza az eredeti jogokat a kliensen a smbmnt parancsra, és csatoljuk le a megosztást:

```
root@kliens:~# chmod 0755 /usr/bin/smbmnt
root@kliens:~# umount /home/user/HOME
```

A sudo parancs

A sudo parancs alkalmas root jogok részleges átruházására a felhasználók részére. Ez is egy set-UID jellegű program, amely egyes parancsokat - root jogokkal – egy felhasználó, vagy egy csoport nevében hajt végre.

A jogok átruházásának beállításához az **/etc/sudoers** állományt kell root-ként szerkeszteni. Használjuk erre a **visudo** programot, amely a vi editorba tölti be az /etc/sudoers állományt. További lényeges információkhoz juthatunk a **man sudoers** parancs kiadása után.

A sudoers állományban lévő bejegyzések (sorok) formátuma a következő:

```
felhasználó gép=(felhasználóként) parancs1,parancs2,...
```

Az parancs első oszlopában az a **felhasználó** vagy csoport szerepel, aki számára jogokat akarunk átruházni. A második oszlop **gép** bejegyzésében azokat a gépeket kell felsorolni, amelyeken érvényes a jogok átruházása. A következő oszlopban meg lehet adni, hogy melyik felhasználó jogit akarjuk átruházni, ez nem csak a root lehet. Ezután következhetnek egyéb opciók, például NOPASSWD:, mely–nek hatására nem kell megadni a jelszót. Végül felsoroljuk a parancsokat (általában elérési úttal együtt), amelyekre az átruházott jogok érvényesek.
Lehetőség van a sudo -alias változók használatára (példák):

```
User_Alias  ADMINS=melegak,kovacsb
Run_Alias   DAEMONS=postfix,nobody
Host_Alias  HOSTS=mk4.melega,bubo.pataky.hu
Cmnd_Alias  PROCESS=/usr/bin/smbmount,/sbin/shutdown
```

PÉLDÁK

A rendszergazda és az user nevű felhasználó mindent futtathat root jogokkal:

```
root ALL=(ALL) ALL
user ALL=(ALL) ALL
```

A users csoport tagjai felcsatolhatnak samba megosztásokat a helyi gépen, jelszó megadása nélkül:

```
%users localhost=(root) NOPASSWD: /usr/bin/smbmount
```

A user nevű felhasználó minden gépen root adminisztrátorként, a jelszó megadása nélkül szerkesztheti **vi** editorral az **/etc/resolv.conf** állományt, azaz beállíthatja a a használandó névszervert:

```
user ALL=(root) NOPASSWD: /bin/vi /etc/resolv.conf
```

FELADATOK

1. Engedélyezzük a felhasználóknak, hogy a kliensen fel- és lecsatolhassák a megosztott könyvtárakat, és beállíthassák a névszervert!
2. Csatoljunk fel felhasználóként egy saját könyvtárat a kliensen úgy, hogy a jelszót ne láthassák mások!

MEGOLDÁS

1. A visudo parancsot kiadjuk rendszeradminisztrátorként, és segítségével hozzáadjuk az engedélyeket tartalmazó sorokat az /etc/sudoers állományhoz:

```
root@szerver:~# visudo

# Engedélyek
%users  localhost=(root) NOPASSWD: /usr/bin/smbmount,/usr/bin/umount
%users  localhost=(root) NOPASSWD: /bin/vi /etc/resolv.conf
```

2. A felcsatoláshoz a sudo parancsot használjuk (feltételezve, hogy root-ként megadtuk az engedélyt). A jelszót csak akkor gépeljük be, amikor a smbmount parancs bekéri:

```
melegak:~$ sudo smbmount //szerver/home /home/melegak/HOME -o username=melegak
added interface ip=127.0.0.1 bcast=127.255.255.255 nmask=255.0.0.0
added interface ip=192.168.108.4 bcast=192.168.108.255 nmask=255.255.255.0
Password:
```

Távoli adminisztráció: SSH

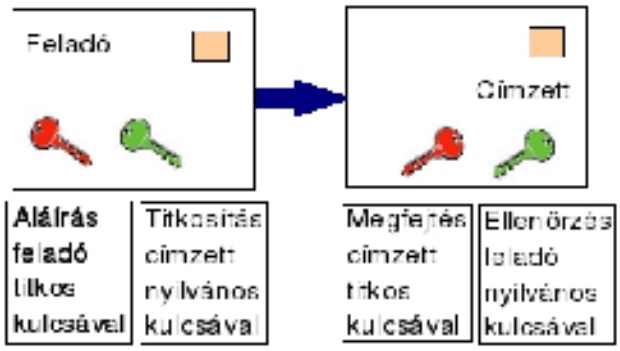
Régebben ***telnet*** segítségével adminisztrálták távolról a szervereket, de ez nem volt biztonságos. A telnet helyett fejlesztették ki az ***ssh*** protokollt, amely kétkulcsos titkosítással biztosítja a távoli számítógépre való bejutást, és ott parancsok kiadását.

Az ssh eljárások gyűjteménye, amely a távoli adminisztráció mellett lehetővé teszi például fájlok biztonságos (titkosított) másolását is (scp, sftp).

A nyilvános kulcsú titkosítás elve

A kulcspárok segítségével aláírás, kódolás, visszafejtés , ellenőrzés valósítható meg, az alábbi módon:

1. A feladó aláírja az üzenetet a saját titkos kulcsával, majd titkosítja a címzett nyilvános (publikus) kulcsával.
2. A címzett visszafejti az üzenetet saját titkos kulcsával, ellenőrzi a feladó aláírását a feladó publikus kulcsával.



Ábra 8 Nyilvános kulcsú titkosítás

Az SSH működése

Az SSH hasonlít az OpenSSL (Secure Socket Layer) működéséhez. Ez azon alapszik, hogy minden hálózati szereplő rendelkezik egy titkos és egy nyilvános kulccsal.

A kapcsolat felépítésekor az ügyfél és a kiszolgáló publikus kulcsot cserél, ezután megegyeznek egy kapcsolati kulcsban (session key), amellyel a továbbiakban titkosítani fognak. A felhasználónév-jelszó szerinti belépés, természetesen így már titkosítva történik.

Az *ssh* működésének alapvető követelménye, hogy a távoli gépen (ami például a virtuális gépen futó szerver) futnia kell a megfelelő démonnak (*sshd*). Emlékezzünk vissza, hogy a szolgáltatások (démonok) indítása az *UHU-Vezérlőpult*, vagy a *service* parancs segítségével oldható meg UHU-Linux operációs rendszerben. Hogy egy gépen fut-e már az sshd, azt például így nézhetjük meg:

```
user@szerver:~$ pidof sshd
```

Ha a parancs kiadása után kapunk egy process-azonosító számot (pid), akkor az *sshd* már fut. Ellenkező esetben az sshd indítása parancssorból az alábbi módon lehetséges:

```
root@szerver:~# service sshd start
```

Ha az sshd indítása parancssorból történik, akkor a számítógép újraindítása után nem fog ismét elindulni a démon. Két lehetőség van, ami ezt a problémát megoldja. Egyszerűbb lehetőség az UHU vezérlőpult használata a démon indítására. Beállítható az is, hogy milyen futási szinteken induljon el az sshd automatikusan.

Karakteres megoldás, ha az **/etc/runlevel.d/custom/sshd.service** állományba begépeljük az alábbi sort:

```
Runlevels=2345
```

Az ssh kapcsolat beállítása

Az ssh beállítására alkalmas konfigurációs állományok közül az egyik az ***/etc/ssh/sshd_config*** a szerveren. A másik a állomány a kliens gépen elhelyezkedő ***/etc/ssh/ssh_config***, amit felülbírálhat a felhasználó saját könyvtárában lévő ***~/.ssh/config*** állomány.

Az ssh_config paraméterei

A kliensen kell szerkeszteni az állományt. A táblázat a leggyakrabban használt paramétereket tartalmazza, az alapértelmezett értéket kiemelve. További információk: **man ssh_config**

Paraméter	Érték	Leírás
CheckHostIP	**Yes**, No	Figyelmeztet, ha eltér a kulcsban rögzítettől az IP cím
Compression	Yes, **No**	Tömörítve küldje az adatokat
ForwardX11	Yes, **No**	X kapcsolat küldés titkosított csatornán
PasswordAuthentication	**Yes**, No	Jelszó hitelesítés a kulcsok mellett vagy helyett

Az sshd_config paraméterei

A szerveren kell szerkeszteni az állományt. Az alapértelmezett értékek kiemelten szerepelnek a táblázatban. További információk: **man sshd_config**

Paraméter	Érték	Leírás
Port	1-65535, **22**	Az a TCP port, amin az sshd figyel.
PermitRootLogin	Yes, No	Root bejelentkezésének elfogadása, ajánlott beállítás: **No**
ListenAddress	**0.0.0.0**	Bejelentkezés engedélyezése IP címekről
PermitEmptyPassword	Yes, **No**	Üres jelszavas bejelentkezés engedélyezés
X11Forwarding	Yes, **No**	X kapcsolat küldés titkosított csatornán
PasswordAuthentication	**Yes**, No	Felhasználónév-jelszó azonosítás
Compression	Yes, **No**	Tömörítve küldje az adatokat

Bejelentkezés a távoli gépre

Távoli gépre ssh segítségével általánosan így tudunk bejelentkezni:

```
user@kliens:~$ ssh felhasználónév@távoligépdomainnév_vagy_IPcím
```

Esetünkben bejelentkezés a szerverre a kliensről:

```
user@kliens:~$ ssh user@192.168.1.200
RSA key fingerprint is 88:ca:70:91:80:4d:e9:44:92:2f:9f:d6:2c:ad:c5:1a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.200' (RSA) to the list of known hosts.
Jelszó:
```

Ha első alkalommal jelentkezünk be egy szerverre, akkor megkérdezi, hogy elfogadjuk-e a szerver publikus kulcsát, y*es* válaszra gépünkre másolja a publikus kulcsot, mégpedig a saját könyvtárunkból nyíló *.ssh* alkönyvtárban elhelyezkedő *known_hosts* nevű állományba. Végül kéri a jelszót, és megjelenik a szerveren futó shell promptja. Itt ugyanúgy adhatunk ki parancsokat, mintha a szerver konzolja előtt ülnénk.

Ha bármilyen problémánk adódna a bejelentkezéssel kapcsolatban, akkor próbáljuk meg törölni a távoli szerver publikus kulcsát a *known_hosts* fájlból, és kíséreljük meg újra a bejelentkezést. A szerverről kijelentkezni az ***exit***, vagy a ***logout*** paranccsal lehet.

Fájlok biztonságos másolása

Több lehetőségünk is van a fájlmásolás biztonságos megoldására a távoli gép és saját számítógépünk között. Használhatjuk az ***scp*** vagy az ***sftp*** parancsot, de például a Midnight Commander segítségével is megoldhatjuk a feladatot.

Biztonságos fájlmásolás scp paranccsal

Az scp parancs működésének feltétele, hogy a célgépen fusson az sshd. A parancs általános alakja:

```
user@kliens:~$ scp felhasználó@honnan/mit /hová/milyen_néven
```

A távoli gép a honnan, vagy a hová pozícióban is lehet, szintaktikailag pedig így néz ki:

```
gépdomainnév_vagy_IPszám:/elrési_út/fájlnév
```

Komplett könyvtárakat scp segítségével történő lemásolásához a ***-r*** (rekurzív)kapcsolót használjuk, például:

```
user@kliens:~$ scp -r felhasználó@gépnév:/könyvtár /könyvtár
```

Ezután egy konkrét példa: a szerverről akarjuk másolni az /etc/hosts fájlt saját könyvtárunkba:

```
user@kliens:~$ scp user@192.168.108.200:/etc/hosts ~
Jelszó:
hosts                         100%   68     0.1KB/s   00:00
```

A jelszó megadása után látjuk a másolás folyamatát, végül visszakapjuk az eredeti promptot. A home könyvtár helyett ~ jelet írtunk, és mivel nem adtunk meg a másolatnak új nevet, ezért az eredeti nevén másolódott le saját, kliensen lévő könyvtárunkba.

FELADAT

Másoljuk le a szerveren lévő ***/etc/apt*** könyvtár tartalmát a kliensen lévő saját könyvtárunkba! Listázzuk ki a lemásolt **~/apt** könyvtár tartalmát, majd töröljük le a könyvtárral együtt!

MEGOLDÁS

```
user@kliens:~$ scp -r user@192.168.108.200:/etc/apt ~
Jelszó:
preferences               100%   44     0.0KB/s   00:00
sources.list              100%  675     0.7KB/s   00:00
apt.conf                  100%  116     0.1KB/s   00:00
```

```
user@kliens:~$ ls -l ~/apt
összesen 12
-rw-r--r--  1 user users  116 2004-10-13 10:21 apt.conf
-rw-r--r--  1 user users   44 2004-10-13 10:21 preferences
-rw-r--r--  1 user users  675 2004-10-13 10:21 sources.list
```

```
user@kliens:~$ rm -r ~/apt
```

Biztonságos fájlmásolás sftp paranccsal

Ha valaki már hozzászokott a parancssori ftp programokhoz, akkor az sftp segítségével biztonságosan használhatja azokat. Bejelentkezés a szerverre:

```
user@kliens:~$ sftp user@192.168.108.200
Connecting to 192.168.108.200...
Jelszó:
sftp>
```

Az sftp prompt megjelenése után adjuk ki a *help* parancsot, ami megjeleníti a használható parancsok listáját és azok rövid leírását. Ha a parancsok elé felkiáltójelet teszünk, akkor azok a helyi gépen hajtódnak végre.

FELADAT

1. Lépjünk be sftp segítségével a szerverre! Nézzük meg, hogy a kliens gépen melyik könyvtárban vagyunk! Lépjünk át a saját könyvtárunkba, és másoljuk oda a szerverről az */etc/bind* könyvtárat, tartalmával együtt!

MEGOLDÁS

```
user@kliens:~$ sftp user@192.168.108.200
Connecting to 192.168.108.200...
Jelszó:
```

```
sftp> !pwd
Local working directory: /home/user
```

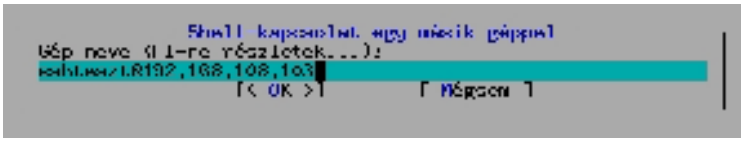
```
sftp> get /etc/bind/*
```

Biztonságos fájlmásolás Midnight Commander segítségével

A kétpaneles fájlkezelők egy része (Midnight Commander, Krusader) fel van készítve a biztonságos fájlmásolás végrehajtására. Indítsuk el az **mc** programot, és válasszuk például a *Bal* menü *Shell-kapcsolat* parancsát! Ezután kell begépelni

```
felhasználónév@gépdomainnév_vagy_IPszám
```

alakban a bejelentkezéshez szükséges adatokat. A következő ábrán egy konkrét esetre láthatunk példát:



Ábra 9

[< OK >] után adjuk meg a jelszót, és a bal panelen megjelenik a távoli fájlrendszer. A távoli fájlrendszerbe jogosultságunknak megfelelően másolhatunk, mozgathatunk stb. fájlokat vagy könyvtárakat.

Ha első alkalommal Midnight Commander segítségével jelentkezünk be egy távoli gépre, akkor a publikus kulcs letöltésével kapcsolatos kérdés miatt az *mc* képe széteshet. Ebben az esetben egymás után kétszer Ctrl+O leütése segít az eredet állapot visszaállításában.

FELADATOK

1. Jelentkezzünk be *mc* segítségével a szerverünkre, és saját könyvtárunkból töröljük le az előbbi feladatokban átmásolt könyvtárakat és fájlokat!
2. Mit eredményeznek a Midnight Commander alábbi billentyűkombinációi?

Alt+C, Alt+H, Ctrl+S, Alt+O, Alt+T, Alt+ENTER, Alt+\ , Ctrl+O

MEGOLDÁS

Alt+C	Gyors könyvtárváltás
Alt+H	Előzmények (history) megjelenítése
Ctrl+S	Fájlkeresés az aktuális panelen
Alt+O	Másik panel tartalmának beállítása
Alt+T	Listázási mód megváltoztatása
Alt+ENTER	Kiválasztott fájlt vagy könyvtárat a parancssorba másolja
Alt+\	Könyvtár gyorslista megjelenítése
Ctrl+O	Parancsok kimenetének megtekintése

Kulcsfájlok használata gyors ssh kapcsolathoz

Gyakran használt ssh kapcsolatnál unalmassá válik, ha állandóan felhasználónevet, gépnevet vagy IP számot illetve jelszót kell begépelni. Sokkal jobb lenne ezt csak a távoli gép nevének begépelésével megoldani. A következő lépések végrehajtásával elérhetjük ezt a célt:

1. A kliensen felhasználóként hozzunk létre egy kulcspárt, ami nyilvános és titkos kulcsokból áll:

```
user@kliens:~$ ssh-keygen -t rsa
```

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
2b:49:3b:fc:5e:e3:16:98:de:b6:59:57:d1:e1:52:77
```

Többszöri ENTER leütés után megkapjuk a kliensen az *~/.ssh* könyvtárban a titkos (id_rsa) és a publikus (id_rsa.pub) kulcsokat.

2. A szerveren annak a felhasználónak saját könyvtárában, aki a kliensről be fog jelentkezni hozzunk létre egy .ssh nevű alkönyvtárat úgy, hogy ehhez sem a csoportjának, sem másoknak még olvasási joga sem legyen.

```
user@szerver:~$ mkdir .ssh
user@szerver:~$ chmod 0700 .ssh
```

3. A nyilvános kulcsot (id_rsa.pub) másoljuk át ***authorized_keys2*** néven a szerverre az imént létrehozott .ssh könyvtárába. A másolás például scp paranccsal végezhető el (a parancsot folyamatosan kell begépelni, a \ jel erre utal:

```
user@kliens:~$ scp /home/user/.ssh/id_rsa.pub user@192.168.1.200:~/.ssh/authorized_keys2
Jelszó:
id_rsa.pub                   100%  221     0.2KB/s   00:00
```

4. Az ***alias*** parancs segítségével oldjuk meg, hogy csak a távoli gép nevét (most szerver) kelljen begépelni és máris be legyünk jelentkezve:

```
user@kliens:~$ alias szerver="ssh user@192.168.1.200"
```

Ha ezt az aliast véglegesíteni akarjuk, akkor rootként gépeljük be a */etc/bashrc* nevű állományba.

5. Végül jelentkezzünk be:

```
user@kliens:~$ szerver
Utolsó belépés: 2004. Oct. 14. (Thu) 20.51.15; 192.168.1.2 gépről; ssh terminálról
user@szerver:~$
```

Grafikus programok futtatása ssh fölött

Néha hasznos lehet, ha a kliensen látjuk a szerverünkön futó grafikus program kimenetét (ablakát). Erre Linuxban több lehetőségünk is van, de a biztonság fokozása érdekében célszerű , ha ezt ssh fölött tesszük. A megvalósításhoz a következő lépésekre van szükség:

1. A szerveren az ***/etc/ssh/sshd_config*** fájlban szerepelnie kell az alábbi bejegyzésnek:

```
X11Forvarding  yes
```

Az ***sshd_config*** szerkesztése után rootként olvastassuk újra az **sshd** démonnal a konfigurációs állományt:

```
root@szerver:~# service sshd reload
```

2. A kliensen vagy az ***~/.ssh/config*** fájlt hozzuk létre a **ForwardX11 yes** bejegyzéssel, vagy az /etc/ssh/ssh_config állományba írjuk bele ugyanezt root-ként.

```
User@kliens:~$  cat > ~/.ssh/config
ForwardX11 yes
```

3. A kliens gépen grafikus konzolban adjuk ki az alábbi parancsot:

```
user@gépnév:~$ szerver xterm
```

A szerveren indított terminál a kliens gép képernyőjén jelenik meg. Ennél összetettebb grafikus átvitelre is van lehetőség. Próbáljuk ki a következő parancsokat:

```
user@gépnév:~$ szerver xfterm4 -fn 9x15 -e mc
user@gépnév:~$ szerver xffm
user@gépnév:~$ szerver mozilla
```

KÉRDÉSEK

1. Mit jelent az ssh és milyen feladatokra megoldására használható?
2. Ismertesse a jelszó nélküli bejelentkezés kialakításának lépéseit!
3. Hogyan lehet ssh fölött grafikus programokat futtatni?
4. Hogyan használható az ***scp*** parancs?
5. Ismertesse az ***sftp*** parancs lehetőségeit!
6. Jelenítse meg kliens gépén a szerveren futó Mozilla böngészőt!

Csomagok telepítése, frissítése

Az UHU-Linux pillanatnyilag a dpkg csomagkezelőt használja, a csomagok kiterjesztése pedig **.uhu**. A csomagkezelők egyébként Linux disztribuciónkénk eltérőek, leggyakrabban .rpm vagy .deb csomagokat használnak.

UHU-Linux esetén a csomagok frissítése és telepítése elsősorban weben keresztül ajánlott, az ***apt-get*** program segítségével. A program képes a csomagokat adott forrásból letölteni, és a ***dpkg*** csomagkezelő segítségével feltelepíteni. Az apt-get általában *ftp* vagy *http* protokollal működik, de lehetőség van a hálózati forgalom jelentős csökkentésére is, az **rsync** protokoll felhasználásával.

Frissítés webről

Az *apt-get* az elérhető csomagokat a ***/var/lib/dpkg/available*** fájlban, ezek állapotát (telepítve van-e) pedig a ***/var/lib/dpkg/status*** fájlban nézi meg. UHU-Linux esetében a „gyári" csomagok telepítőforrása az ***ftp.uhulinu.hu,*** a baráti kör által készített „nem hivatalos" csomagok pedig az ***ubk.uhulinux.hu*** helyen érhetők el. A telepíthető csomagok természetesen lokális tükörszerveren is elhelyezhetők.

Minden telepítőforrás gyökérkönyvtárában lennie kell három fájlnak, amelyben fel vannak sorolva a rendelkezésre álló csomagok. Ezeknek az állományoknak a neve ***Packages, Packages.gz*** *és* ***Release***.

Az apt-get használata

1. Először az ***/etc/apt/sources.list*** fájlban szükség szerint módosítanunk kell a telepítőforrásokat.

FIGYELEM! Az éppen szükségtelen telepítőforrásokat tegyük megjegyzésbe (a sor elejére begépelt # jellel), hogy fölösleges csomagok véletlenül se kerülhessenek telepítésre!

Ha kilistázzuk az ***/etc/apt/sources.list*** fájl tartalmát például ***cat*** paranccsal, akkor ehhez hasonló bejegyzéseket látunk:

```
root@szerver:~# cat /etc/apt/sources.list

#/etc/apt/sources.list
#Hivatalos csomagok
#deb ftp://ftp.uhulinux.hu/pub/uhu/1.1 ./
#deb http://ftp.uhulinux.hu/pub/uhu/1.1 ./
deb rsync://rsync.uhulinux.hu/ftp/uhu/1.1 ./

#Tükörszerverek
#deb ftp://ftp.linuxforum.hu/uhulinux/pub/uhu/1.1 ./
#deb ftp://ftp.prew.hu/pub/Linux/UHULinux/pub/uhu/1.1 ./
```

Pillanatnyilag csak az *rsync* bejegyzés aktív, amivel gazdaságosabban frissíthetünk, mert csak a csomagok módosult részei kerülnek letöltésre.

2. A helyi csomagadatbázis frissítéséhez adjuk ki a következő parancsot:

```
root@szerver:~# apt-get update
Get:1 rsync://rsync.uhulinux.hu ./ Packages [142KB]
Get:2 rsync://rsync.uhulinux.hu ./ Release [53B]
Fetched 142KB in 7s (18,6KB/s)
Reading Package Lists... Done
```

3. A következő parancs hatására frissülnek azok a csomagok, amelyek már fel vannak telepítve a rendszerre:

```
root@szerver:~# apt-get upgrade
```

A frissebb csomagok először letöltődnek a ***/var/cache/apt/archives*** könyvtárba, onnan automatikusan települnek, és konfigurálódnak.

4. Az elavult csomagokat törölhetjük a ***/var/cache/apt/archives*** könyvtárból az alábbi parancs kiadásával:

```
root@szerver:~# apt-get autoclean
```

5. Ha csak egy csomagot akarunk telepíteni, akkor használjuk a következő formát:

```
root@szerver:~# apt-get install csomagneve
```

6. Eltávolít egy csomagot, de a konfigurációs állományok megmaradhatnak:

```
root@szerver:~# apt-get remove csomagneve
```

7. Ha a konfigurációs állományoktól is meg akarunk szabadulni a programmal együtt, akkor használjuk így a parancsot:

```
root@szerver:~# apt-get remove --purge csomagneve
```

Megjegyzés: A **purge** paraméter előtt nem véletlenül van 2 db mínusz jel.

A dpkg használata

Ezt a parancsot akkor szoktuk alkalmazni, ha egy csomag nem telepítőforrásból installálunk a rendszerünkre. A csomagot ilyenkor először lemásoljuk a gépünkre a rendelkezésünkre álló forrásból például a /tmp könyvtárba, majd a következő módon telepítjük:

```
root@szerver:~# dpkg -i csomagneve
```

Csomagok leszedéséhez használjuk a dpkg programot ***-r*** kapcsolóval, ***--purge*** a konfigurációs állományokat is törli, függőségproblé-ma esetén pedig a ***--- force-all*** opció alkalmazása ajánlott.

Saját telepítő forrás készítése a telepítő CD-k segítségével

Otthoni gépünkre kényelmesen – Internetről - az apt-get segítségével telepítünk csomagokat. Lokális hálózat esetén már meggondo-landó egy saját telepítő forrás (tükör, mirror) létrehozása.

Saját telepítőforrást a szerveren vagy a kliensen is készíthetünk, természetesen root-ként, a következő leírás szerint:

1. Másoljuk fel mindkét UHU-Linux telepítő CD **packages** könyvtárainak tartalmát a **/var/ftp/1.1/packages/release** könyvtárába rootként.

2. Telepítő forrásunk akkor lesz, ha a **/var/ftp/1.1** könyvtárban létrehozzuk a **Packages, Packages.gz** és **Release** fájlokat. Ehhez az **uhu-scanpkgs** parancsot fogjuk használni. Ha nincs ilyen, akkor az ezt tartalmazó csomagot fel kell telepíteni.

3. Keressük meg azt a csomagot, amely tartalmazza az uhu-scanpkgs programot:

```
root@szerver:~# dpkg -S uhu-scanpkgs
uhu-pkgutils: /usr/bin/uhu-scanpkgs
```

4. A keresés eredményeként látjuk, hogy a keresett program az uhu-pkgutils csomagban található. Lépjünk be a **/var/ftp/1.1/packa-ges/release** könyvtárba, és adjuk ki a ***dpkg*** parancsot a következő módon (egy sorban, \ jelet nek kell beírni, a csomag nevének ki-egészítéséhez használjuk a TAB billentyűt):

```
root@szerver:/var/ftp/packages# dpkg -i uhu-pkgutils_1.0.0-2.6_i386.uhu
```

5. A **/var/ftp/1.1** könyvtárba átlépve most már létrehozhatjuk a **Packages, Packages.gz** és **Release** fájlokat az **uhu-scanpkgs** pa-ranccsal:

```
root@szerver:/var/ftp/1.1# uhu-scanpkgs
```

6. Ellenőrizzük, hogy a fájlok létrejöttek a **/var/ftp/1.1** könyvtárban.

7. A következő lépésben indítsuk el az ftp démont, hogy a telepítőforrás ftp protokollal elérhető legyen:

```
root@szerver:~# service proftpd start
FTP szerver (proftpd) indítása                  [ OK ]
```

8. Ha a szervert is ebből a telepítőforrásból akarjuk csomagokkal ellátni, akkor gépeljük be az ***/etc/apt/sources.list*** nevű fájlba root-ként a következő sort:

```
deb file:/var/ftp/1.1 ./
```

9. Ha a weben keresztül szinkronizálni akarjuk telepítő forrásunkat, akkor írjuk meg az alábbi szkriptet rootként, a **/root** könyvtárban:

```
#!/bin/bash
rsync -avP --delete --delete-after rsync.uhulinux.hu::ftp/uhu/1.1 /var/ftp/1.1
```

Mentsük el a fájlt **uhu-mirror** néven a **/root** könyvtárba, és adjunk rá futási jogot:

```
root@szerver:~# chmod 0700 uhu-mirror
```

A szkriptet esetenként is futtathatjuk, de automatizálhatjuk a folyamatot a cron segítségével.

10. A kliensen módosítsuk az ***/etc/apt/sources.list*** nevű fájlt rootként:

```
deb ftp://szerver_neve_vagy_IP_száma/1.1  ./
```

11. Próbáljuk ki a kliensen, hogy működik-e a helyi telepítőforrás.

```
root@szerver:~# apt-get update
user@kliens:~# apt-get update
```

KÉRDÉSEK

1. Mi a különbség az ***apt-get***, illetve a ***dpkg*** program között?

2. Sorolja fel az ***apt-get*** lehetőségeit és az ezekhez használható opciókat!

3. Mit jelent a csomagkezelésben ez a fogalom: ***dependencies***?

4. Ismertesse az UHU-Linux telepítőforrás létrehozásának lépéseit!

FELADAT

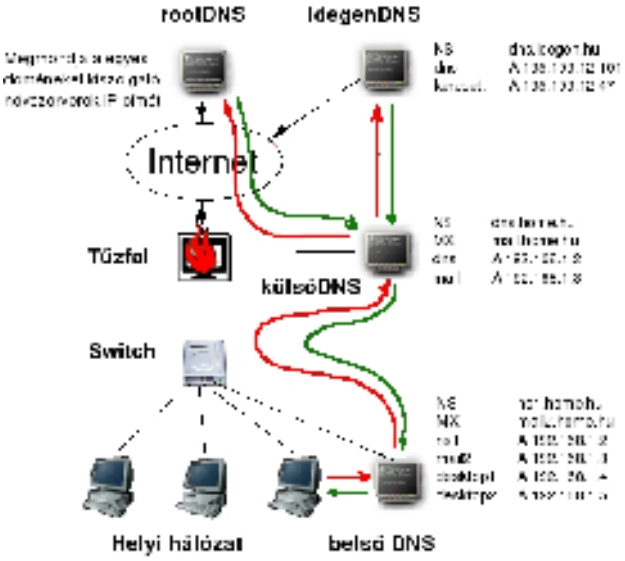
1. Nézzen utána az Interneten, hogy a felsorolt Linux disztribúciók milyen csomagtípusokat használnak?
Debian, SUSE, Fedora, Slackware, RedHat, Knoppix

Névszerver kialakítása

Kisebb helyi hálózatoknál általában nem szükséges névszerver használata, megteszi helyette az **/etc/hosts** fájl szerkesztése, majd másolása az összes gépre. A névszerver alkalmazásának előnye, hogy egyszerűbb adminisztrálni, változás esetén nem kell a hosts fájlokat módosítani. Hátránya a biztonsági kockázat, amely nem csekély. A névszerver feladata a névfeloldás, tehát ha egy kérés érkezik hozzá, akkor megadja a kérésben szereplő doménnévhez tartozó IP címet (vagy fordítva).

A névfeloldás működési elve

A névfeloldás a betörési statisztikák szerint nem nevezhető biztonságosnak, ezért célszerű lehet a szolgáltatás kettéosztása egy minimális adatmennyiséget tartalmazó nyilvános (külső) és egy csak belülről elérhető belső szolgáltatásra. Az ábrán példán keresztül mutatjuk be a névfeloldás folyamatát.



1. A helyi hálózat gépe a *keresett.idegen.hu* gép IP címét szeretné megtudni. Megkérdezi a belső névszervertől,ez van megadva (resolv.conf).

2. A belső névszerver megkérdezi a külsőt (forward beállítva).

3. A külső névszerver egy root névszerverhez fordul, ahonnan megtudja az idegen.hu domént kiszolgáló névszerver IP címét. A root névszerverek az */etc/bind/db.root* fájlban vannak.

4. Miután a külső névszerver megtudta, hogy melyik névszervertől kell megkérdeznie a keresett névhez tartozó IP számot, ezt meg is teszi. Továbbítja a belső névszervernek, ez pedig a helyi hálózaton annak a gépnek, amelyik a kérdést feltette.

Saját névszerver létrehozása

Az UHU-Linux disztribúció alapból telepíti a névszolgáltatáshoz szükséges komponenseket, ezzel tehát nem kell foglalkoznunk. Saját névszolgáltatásunkat a szerveren konfiguráljuk, illetve helyezzük üzembe.

1. Először rootként megszerkesztjük az **/etc/named.conf** állományt úgy, hogy a végére beillesztünk egy zóna és egy fordított zóna bejegyzést. A zóna neve szabadon választható, legyen most **sajat,** az alhálózat pedig, amit ki kell szolgálnunk 192.168.1.0.

```
// This is primary configuration file for the BIND DNS server named.
// A névszolgáltatás alapkönyvtára, javasolt áthelyezni.

options {

directory "/etc/bind";
      // query-source address * port 53;
      // allow-query { 192.168.108.0/24; localhost; };

//azok a névszerverek, akiktől kérdezünk
forwarders {
195.56.234.124;
};
};
// reduce log verbosity on issues outside our control
logging {
      category lame-servers { null; };
      category cname { null; };
```

```
};
// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};
// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};
zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};
zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};
// Saját zóna bejegyzései

zone "sajat." {
            type master;
            file "/etc/bind/db.sajat";
};

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

A **type master** bejegyzés elsődleges névkiszolgálót jelöl, másodlagos névszervernél ez *type slave* lenne. Ha al-doméneket is szeretnénk, annak a neve így nézhetne ki: *iroda.ceg.hu*, *tanterem.ceg.hu* stb.

2. Létrehozzuk rootként a szerveren az ***/etc/bind/db.sajat*** zóna állományt. A szériaszám kivételével másodpercekben megadott időintervallumokat látunk.

```
;Saját zónaállomány

$TTL  86400
@     IN    SOA   szerver.sajat.    root.sajat. (
            200411011   ; Serial
                28800   ; Refresh
                 7200   ; Retry
               604800   ; Expire
                86400 ) ; Negative Cache TTL
;-----------------------------------------------
@     IN    NS    szerver.sajat.
;-----------------------------------------------
szerver     IN    A     192.168.1.200
```

Az egyes bejegyzések értelméről az alábbi táblázatban kapunk felvilágosítást.

Bejegyzés	Jelentés
;	Megjegyzés
TTL	Time To Live, a zónabejegyések ennyi ideig maradhatnak más szerverek gyorsítótárában
@	Helyettesíti a doménnevet, ami most *helyi.hu*
SOA	Start Of Authority, zóna kezdete
Serial	Sorozatszám, év(4), hónap(2), nap(2), sorszám(2)
Refresh	Frissítés (most 7 nap), a slave szerver ennyi idő múlva kérdezi meg a mastertől a sorszámot
Retry	Várakozási idő, ha a frissítés nem sikerült
Expire	Ha a master nem válaszol, mennyi ideig szolgáltatunk?
NS	Name Server, névszerver rekord
MX	Mail eXchanget, levelezőszerver rekord
A	Address (cím rekord) doménnév - IP cím összerendelés
CNAME	Segítségével a hosztokat más néven is nevezhetjük

3. Létrehozzuk rootként a szerveren fordított zóna adatállományát. Ez biztosítja a cím-név hozzárendelést.

```
;Fordított saját zónaállomány

$TTL 86400
@     IN          SOA   szerver.sajat.    root.sajat. (
            200411011   ; Serial
                28800   ; Refresh
                 7200   ; Retry
               604800   ; Expire
                86400 ) ; Negative Cache TTL
;-------------------------------------------------
@     IN    NS    szerver.sajat.
;-------------------------------------------------
200   IN    PTR   szerver.sajat
```

A **PTR** jelentése PoinTer Rekord, azaz mutató bejegyzés. Ez mutatja meg, hogy egy IP címhez milyen doménnév tartozik.

4. Szerkesszük meg a szerveren az /etc/resolv.conf állományt, adjuk meg:

```
search sajat
nameserver  127.0.0.1
```

5. A konfigurációs állományok megszerkesztése után indítsuk el a **named** szolgáltatást a szerveren, az UHU Vezérlőpulttal, vagy rootként parancssorból:

```
root@szerver:~# service named start
```

A vezérlőpultban való named indítás esetén a számítógép újraindítása esetén is elindul a névszolgáltatás. Ha a **service** parancsot használjuk szívesebben, akkor gépeljük be az */etc/runlevel.d/custom/named.service* fájlba:

```
Runlevels=2345
```

6. Ellenőrizzük, hogy a névszerverünk valóban működik. Ehhez először a szerveren adjuk ki a **dig** parancsot **-x** kapcsolóval egy olyan IP számra, amelyik szerepel a névszerver bejegyzései között (most csak a szerver):

```
user@szerver:~$ dig -x 192.168.1.200

;; QUESTION SECTION:
;200.1.168.192.in-addr.arpa.  IN      PTR

;; ANSWER SECTION:
200.1.168.192.in-addr.arpa. 86400 IN PTR   szerver.sajat.1.168.192.in-addr.arpa.

;; AUTHORITY SECTION:
1.168.192.in-addr.arpa. 86400 IN      NS      szerver.sajat.

;; ADDITIONAL SECTION:
szerver.sajat.            86400   IN      A       192.168.1.200

;; Query time: 11 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Nov  1 21:28:06 2004
;; MSG SIZE  rcvd: 113
```

7. További vizsgálat lehetséges az **nslookup** parancs segítségével. Az **nslookup** egy promptot biztosít a különböző lekérdezések számára, ahová doménneveket vagy IP számokat gépelhetünk be. Ezekre a program valamilyen adatokkal válaszol. Ennek egy részlete a következő oldalon tekinthető meg.

```
user@szerver:~$ nslookup

> szerver.sajat
Server:    127.0.0.1
Address:   127.0.0.1#53

Name:   szerver.sajat
Address: 192.168.1.200
```

KÉRDÉSEK

1. Milyen fájlok szerkesztése, illetve létrehozása szükséges egy egyszerű névszolgáltatás beindításához?
2. Ismertesse a zónaállományok felépítését!
3. Ismertesse a névfeloldás működési elvét! Készítsen sematikus ábrát egy névfeloldási folyamat szemléltetésére!
4. Névszerverek esetén mit jelent a forward fogalma?
5. Milyen parancsokkal lehet ellenőrizni a névszerver helyes működését?

Elektronikus levelezés: email

Az elektronikus levelezés az Internet egyik leggyakrabban használt szolgáltatása. A levelek továbbítását két (vagy több) program együttesen végzi. Az ügyféloldalon elhelyezett Mail User Agent (MUA) arra szolgál, hogy a felhasználó a levelét megszerkessze, majd átadja továbbításra a Mail Transport Agentnek (MTA). Az MTA feladata a levél továbbítása a címzett felé. Ha a küldő és a címzett nem ugyanazon a szerveren rendelkezik postafiókkal, akkor a távoli szerverrel való kapcsolat felépítését is az MTA végzi.

Az alapvető levelezési feladatok ellátásához UHU-Linux operációs rendszerben a következők szükségesek:

- Névszolgáltatás (DNS), ezt már kialakítottuk,
- Levéltovábbító program (MTA). Erre a célra a **postfix** levelezőszervert választjuk, ez már telepítéskor felkerül az UHU-Linux szerverre,
- Levelező kliens, itt a **nail, mutt** és **evolution** programokat is használni fogjuk.
-

Levelező rendszer telepítése

A **postfix** levelező szerver, illetve a nail, mutt és evolution kliensek az alaprendszerrel együtt már fel vannak telepítve virtuális szerverünkre.

A postfix konfigurálása

A konfigurációs fájl ***/etc/postfix/main.cf***, ezt kell szükség szerint rootként módosítanunk, természetesen a virtuális szerveren.

Változtatásokat a konfigurációs állományban fájlban csak ott eszközöljünk, ahol ez feltétlenül szükséges. Az alapértelmezett beállításokat az /etc/main.cf.default fájl tartalmazza.

Minden beállítás az alábbi sémát követi: *változó = érték.* Egy példa konfigurációs állomány:

```
# Levelezőszerverünk teljes nevét a myhostname változóban tároljuk.
myhostname                              =                              szerver.sajat

# Doménnevünk a mydomain változóba kerül.
mydomain                                  =                                  sajat

# Minden levél amit felhasználóink küldenek ilyen domén névvel kerül kiküldésre.
# Ha a  domén sajat, a szerver neve szerver, és a user nevű felhasználó küld egy
# levelet, akkor az a következő formában kerül elküldésre: user@szerver.sajat
myorigin                                =                                $myhostname

# Az összes hálózati csatoló címén fogadunk leveleket.
# Figyelem! Ha ezt a bejegyzést megváltoztatjuk, akkor a postfix démont a stop, majd start
# paraméterekkel újra kell indítanunk.
inet_interfaces                       =                      all

# Levél érkezhet a teljes internetes nevünkre, domain nevünkre, mail.domain nevünkre, és a
# localhost.domain nevünkre. Jelen esetben: mail.sajat,  sajat,  mail.sajat, lo
# calhost.sajat. A localhost fontos, ha kifelejtjük problémáink lehetnek.
mydestination   =   $myhostname,   localhost.$mydomain,   $mydomain,   mail.$mydomain

# A mynetworks változóban tároljuk, hogy mely alhálózatokról fogadunk el leveleket
# továbbításra, azaz relay-ezünk. Fontos, hogy a localhost címe (127.0.0.0/8) is benne
# legyen. Az újabb postfixekben már beállíthatjuk a host típusú relay-ezést, tehát akár
# egy hosztra is lekorlátozhatjuk a levélküldés engedélyezését!
mynetworks                =               192.168.1.0/24,               127.0.0.0/8

#          Ha          központi          levelezőszerver          is          van:
#                    relayhost                       =                     $mydomain
# relayhost = átjaró.saját.domain
# relayhost = [egy.ipcim.is.jó]
# Virtuális felhasználók létrehozása, vagy mail aliasok, pl.teszt@szerver.sajat
# mellé teszt.elek@szerver.sajat készítése. Az alias-okat az /etc/aliases
```

```
# fájlban helyezzük el, majd lefuttatjuk a newaliases parancsot.
#                 alias_maps                  =                 hash:/etc/aliases
#               alias_database                =                 hash:/etc/aliases

# Spam- és vírusszűrés az átmenő leveleken. A filtert a master.cf file-ban kell
#                                                                    definiálni
#                                  content_filter                                =

#  „Maildir” típusú portafiók létrehozása az azt kérő felhasználók részére
# home_mailbox = Maildir/

# Felhasználók postafiókméretének korlátozása [kB]. Ha a postafiók megtelik, akkor
#    sem    a    feladó,    sem    a    címzett    nem    kap    hibaüzenetet
#                               mailbox_size_limit                             =

# Maximális küldhető levélméretet korlátozása [kB]
message_size_limit =
```

A konfigurációs állomány szerkesztése után indítsuk újra a postfix démont:

```
root@szerver:~# service postfix stop
Postfix levelezőszerver (postfix) leállítása          [ OK ]

root@szerver:~# service postfix start
Postfix levelezőszerver (postfix) indítása            [ OK ]
```

Levelezőszervert fogunk a **sajat** tartományban üzemeltetni, ezt a névszerverben regisztrálni kell .

Az **/etc/bind/db.sajat** állományba írjuk be az MX bejegyzést, illetve ha mail.sajat címre is akarunk levelet kapni, akkor a megfelelő alias sort is el kell ebben az állományban helyezni. Az /etc/bind/db.sajat állomány számunkra fontos részlete a változásokkal:

```
;-------------------------------------------------------
@       IN      NS         szerver.sajat.
        IN      MX      10 szerver.sajat.
;-------------------------------------------------------
szerver IN      A       192.168.108.200
mail    IN      CNAME   szerver
```

Töltessük be újra a démonnal a névszerver konfigurációs állományait:

```
root@szerver:~# service named reload
```

Levélküldés a levelező szerveren regisztrált felhasználóknak

A szerveren hozzunk létre root-ként egy *teszt* nevű felhasználót, és adjunk neki jelszót (min 6 karakter):

```
root@szerver:~# man useradd
root@szerver:~# useradd -m teszt
root@szerver:~# passwd teszt
New UNIX password:
Retype new UNIX password:
```

A kliensről felhasználóként ssh segítségével jelentkezzünk be a szerverre, és írjunk egy levelet a *teszt* felhasználónak (a levélírást Ctrl+D billentyűkombinációval fejezzük be). Az UHU-Linux a ***nail*** nevű levelező programot alapból telepíti, erre ***mail*** néven van szimbolikus link:

```
user@szerver:~$ mail teszt@mail.sajat
Subject: teszt
Ez egy teszt.
EOT
```

Jelentkezzünk be be a szerverre kliensünkről **teszt** felhasználóként:

```
melegak@mk4:~$ ssh teszt@192.168.108.200
Jelszó:
Üdvözöllek, új felhasználó!
```

```
Új leveled van a /var/mail/teszt fájlban.
```

Olvassuk el a teszt felhasználó leveleit a /var/mail/teszt állományban (részlet):

```
teszt@szerver:~$ cat /var/mail/teszt
From melegak@szerver.sajat  Thu Jan 20 09:19:55 2005
Return-Path: <melegak@szerver.sajat>
X-Original-To: teszt@mail.sajat
Delivered-To: teszt@mail.sajat
Date: Thu, 20 Jan 2005 09:19:53 +0100
To: teszt@szerver.sajat
Subject: teszt
User-Agent: nail 10.6 11/15/03
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
From: melegak@szerver.sajat

Ez egy teszt.
```

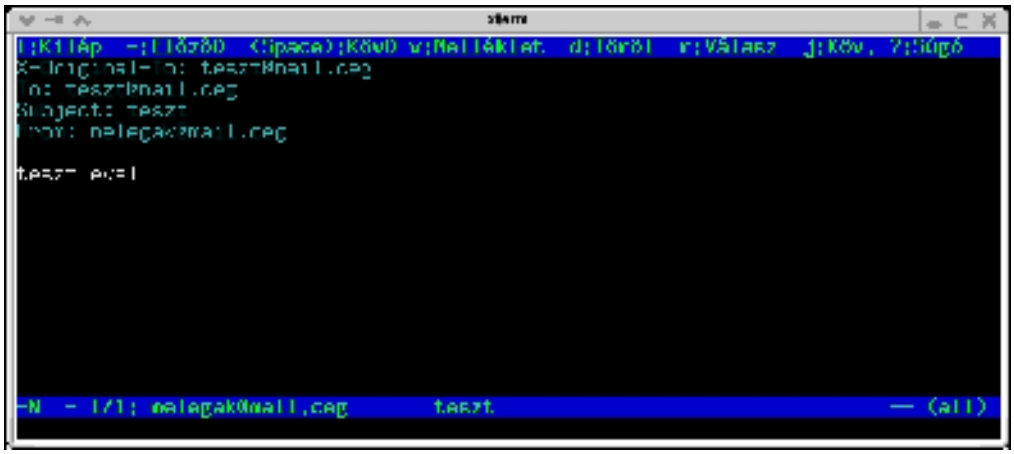
A postfix levelező szerver a leveleket a /var/mail könyvtárba kézbesíti, ahol minden felhasználó szöveges állományként is olvashatja. A kliensről **teszt** felhasználóként jelentkezzünk be a szerverre, és olvassuk el a levelünket egy egyszerű levelező programmal:

```
teszt@szerver:~$ mail
mailx version nail 10.6 11/15/03.  Type ? for help.
"/var/mail/teszt": 1 messages 1 new
>N  1 melegak@szerver.sajat  Thu Jan 20 09:19  18/568   teszt
?
```

A kérdőjel után gépeljük be a levél sorszámát, és máris olvashatunk. Nem működik jól a rendszer, ha a teszt felhasználóként **su teszt** paranccsal jelentkezünk át. A helyes parancs ilyenkor **su – teszt,** így biztosítva lesz a megfelelő környezet.

A mutt levelezőprogram

Indítsuk el **teszt** felhasználóként **mutt** levelezőprogramot! Megkérdezi, hogy létrehozza-e a /home/teszt/Mail könyvtárat. Válaszoljunk igennel. A levél kiválasztása és ENTER után elolvashatjuk a levelünket.



Ábra 10

A mutt egyébként egy nagy tudású levelező kliens, számos lehetőséggel. Egyik előnye, hogy karakteres felületen is üzemeltethető. Kezelése némi ismeretet igényel, célszerű először a súgó tanulmányozása.

Levelek kézbesítése: procmail

A **procmail** olyan alkalmazás, amely a levelező kiszolgálótól átveszi a leveleket, majd ezeket a kívánt feltételek alapján az általunk megadott levelesládába kézbesíti, és szűrni is tudja a leveleket. A procmail már telepítéskor felkerül az UHU-Linux operációs rendszerre, külön elindítani sem kell, a postfix **main.cf** állományában hivatkozunk rá.

Konfigurálni a felhasználó saját (home) könyvtárában létrehozott **.procmailrc** nevű állomány szerkesztésével lehet. A szerveren

teszt felhasználóként hozzuk létre a fájlt saját könyvtárunkban, a következő tartalommal:

```
teszt@szerver:~$ cat > .procmailrc
SHELL=/bin/bash
MAILDIR=~/Mail
LOGFILE=$HOME/procmail.log
:0
* ^From:.*user
/dev/null
:0
* .*
$MAILDIR/Inbox
```

A szerkesztés lezárása: Ctrl+D

A bejegyzések értelme:

Bejegyzés	Jelentése
SHELL	Ez a változó tárolja a használt héjprogramot
MAILDIR	Ez a változó tárolja felhasználó levelezőkönyvtárát
LOGFILE	A naplófájl helye, ahová a procmail folyamatosan bejegyzi a levélforgalmat.
:0 * ^From:.*user /dev/null	Az user nevű felhasználótól érkező leveleket egyből kidobja.
:0 * .* $MAILDIR/Inbox	Minden levelet, ami nem az user felhasználónak szólt az Inbox állományba kézbesíti.

Hozzuk létre felhasználóként a ~/Mail mappát (ha még nincs ilyen):

```
teszt@szerver:~$ mkdir Mail
```

Ha ezután írunk levelet a teszt felhasználónak, akkor az a /home/teszt/Mail könyvtárban létrehozott Inbox nevű állományba lesz kézbesítve.

A mutt konfigurálása

A **mutt** beállításához a teszt felhasználó saját könyvtárában létre kell hozni a **.muttrc** állományt, amiben megmondjuk a levelező programnak, hogy ezen túl hol keresse a leveleket tartalmazó állományokat:

```
user@szerver:~$ cat > .muttrc
set folder=~/Mail
set editor="mcedit"
set charset="iso-8859-2"
set hostname="szerver.sajat"
set sort=date-received
set spoolfile=~/Mail/Inbox
set tmpdir=~/tmp
```

A szerkesztés lezárása: Ctrl+D

A **/home/teszt/tmp** könyvtárat „kézzel” kell létrehozni. A konfigurációs bejegyzések értelme:

Bejegyzés	Jelentése
set folder	Kiinduló könyvtár, később állítjuk be
set editor	A levelek írásához használt szövegszerkesztő

Bejegyzés	Jelentése
set charset	A levelek írására-olvasására beállított karakterkészlet
set hostname	A szerver teljes doménneve
set sort	Levelek rendezési sorrendje
set spoolfile	A bejövő levelek helye
set tmpdir	A piszkozatok helye, létre kell hoznunk

Tanácsok a mutt használatához

A **mutt** számtalan billentyűparanccsal rendelkezik, ezek közül csak néhánnyal ismerkedünk meg.

Postafiókot a **c** billentyűvel válthatunk, ezután **?** leütésével választhatunk a postafiókok között.

Levelet törölni a **d** billentyű leütésével lehet, ez egy *D* betűvel megjelöli a törlendő leveleket, amelyek véglegesen akkor törlődnek, ha kilépünk a levelező programból.

Levelek letöltése a munkaállomásra

Az elektronikus levelezés általában nem egy belső hálózaton folyik (bár ez sem ritka egy vállalaton belül), hanem az egész Interneten. Ha a levelező szerverre belépve olvassuk a leveleinket, annak vannak hátrányai. Ha csak modemes kapcsolatunk van, akkor ez elég sokba kerül. Szintén lassíthatja a folyamatot, ha a levelezéshez valamilyen webmail programot használunk.

A mai gyakorlat szerint jobb megoldás, ha levelező programunk a munkaállomáson fut, és a levelező szerverünkre csak addig kell csatlakoznunk, míg letöltjük a leveleket. Ezek olvasására, megválaszolására már ráérünk később, amikor már megszakítottuk a modemes kapcsolatot.

A levelek letöltése a szerverről például POP3 vagy IMAP protokoll segítségével történhet. A POP3 minden levelet teljes terjedelmében letölt a gépünkre, és alkalmazásakor eldönthetjük, hogy a levelező szerveren is megőrizzük, vagy azonnal töröljük a levelet. Ellentétben a POP3-mal az IMAP először csak a levelek fejléceit küldi el, lehetővé téve, hogy a levelek letöltésükről külön-külön dönthessünk. Cserébe postafiókunk tárigénye a levelező szerveren megnő, a telefonszámlánkkal együtt.

Először nézzük meg, hogy van-e POP3 szolgáltatás a szerveren. Használjuk erre a szerveren kiadott **nmap** parancsot:

```
user@szerver:~$ nmap szerver.sajat

Starting nmap 3.77 ( http://www.insecure.org/nmap/ ) at 2005-01-20 23:24 CET
Interesting ports on 192.168.108.200:
(The 1659 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
53/tcp  open  domain
631/tcp open  ipp
```

A lekérdezés eredményeként látjuk, hogy POP3 szolgáltatás pillanatnyilag nem áll rendelkezésre. POP3 szolgáltatást biztosít a ***cyrus-imapd*** csomag, ezt kell telepítenünk a virtuális szerverre. Szükséges még a cyrus-sasl csomag is, ami viszont már az UHU-Linux telepítésekor felkerül az operációs rendszerre.

```
root@szerver:~# apt-get install cyrus-imapd
```

Beállítások

A **cyrus** működéséhez konfigurációs fájlokat kell módosítanunk, illetve szolgáltatásokat kell elindítani vagy újraindítani.

1. A kliensen rootként az **/etc/resolv.conf** állományba írjuk be első helyre a saját névszerverünket, hogy legyen a lokális hálózatunkon névfeloldás:

```
nameserver    192.168.1.200
```

2. Az ***/etc/postfix/main.cf*** állományban helyezzük megjegyzésbe a procmail-re, és engedélyezzük a cyrus-ra vonatkozó bejegyzést:

```
#mailbox_command = /usr/bin/procmail
mailbox_transport  =  cyrus
```

3. Indítsuk el a szerveren a **saslauthd** és **cyrus** szolgáltatást, indítsuk újra **postfix** levelező szerverünket, és ellenőrizzük a **pop3** és **imap** protokollokat:

```
root@szerver:/etc# service saslauthd start
SASL AUTH Daemon (saslauthd) indítása

root@szerver:/etc# service cyrus start
Cyrus IMAP/POP3 server (cyrus) indítása          [ OK ]
root@szerver:/etc# service postfix restart
Postfix levelezőszerver (postfix) leállítása     [ OK ]
Postfix levelezőszerver (postfix) indítása       [ OK ]
```

```
root@szerver:/etc# nmap 192.168.1.200
```

```
PORT    STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
53/tcp   open  domain
110/tcp  open  pop3
143/tcp  open  imap
631/tcp  open  ipp
993/tcp  open  imaps
995/tcp  open  pop3s
```

4. A kliens gépen indítsuk el az **Evolution** levelező programot, és készítsünk az **Eszközök** > **Beállítások** paranccsal egy *user@szerver.sajat*, és egy teszt*@szerver.sajat* postafiókot.
5. Írjunk a felhasználók nevében leveleket, és töltsük le a kliensre az **Evolution** levelezőprogram segítségével.

Cyrus postafiókok a szerveren

Amikor az Evolution levelező programmal bejelentkezünk a levelező szerverre, akkor automatikusan létrejön a felhasználó postafiókja. A cyrus által kezelt postafiókok helye a /var/spool/imap könyvtáron belül van. Például a teszt felhasználó postafiókját listázva ezt látjuk:

```
root@szerver:~# ls -a /var/spool/imap/t/user/teszt/
. .. 1. cyrus.cache cyrus.header cyrus.index Sent SPAM Trash
```

A teszt felhasználónak küldött levelek külön, számozott állományokban helyezkednek el. Egy levél listázása után a következőt látjuk (részlet):

```
root@szerver:~# cat /var/spool/imap/t/user/teszt/1.

Subject: teszt
From: Melega =?ISO-8859-1?Q?K=E1lm=E1n?= <melegak@melega>
Reply-To: melegak@sajat
To: teszt@szerver.sajat
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
Date: Thu, 03 Feb 2005 20:53:48 +0100
Message-Id: <1107460428.3246.11.camel@kliens>
Mime-Version: 1.0
X-Mailer: Evolution 2.0.3

teszt
```

A felhasználó postafiókját a szerveren is elkészítheti, elég a cyradm programmal egy egyszerű bejelentkezés:

```
user@szerver:~$ cyradm localhost
IMAP Password:
localhost.localdomain> exit
```

Az elektronikus levelezés természetesen ennél sokkal bonyolultabb feladat, hiszen a hitelesítés, titkosítás, vírusellenőrzés, spamszűrés, adatbázis támogatás problémaköre ebben a leírásban még csak szóba sem került.

Weblapok közzététele: Apache

Az **Apache** webszerver népszerűségére jellemző, hogy Linux és BSD környezetben szinte egyeduralkodó, de a Windows szerverek egy részén is ez fut. Gyors és megbízható, el van látva megfelelő kommunikációs modulokkal, amelyek lehetővé teszik a kapcsolódó szolgáltatásokkal való együttműködést (php, mysql, perl stb).

UHU-Linux használata esetén az **Apache** nem része az alaprendszernek, telepítenünk kell a szerverre:

```
root@szerver:~# apt-get install apache
root@szerver:~# apt-get install apache-mod*
```

A második parancs telepíti az összes modulokat, a php csomaggal együtt. A telepítés után indítsuk el rootként a szolgáltatást az alábbi paranccsal:

```
root@szerver:~# service apache start
httpd: Could not determine the server's fully qualified domain name, using
192.168.1.200 for ServerName
```

Indítsunk el a kliensen egy böngészőt, és gépeljük be a beviteli mezőbe a következőt:

```
http://szerver.sajat
```

Megjelenik az alapértelmezett nyitóoldal:

Ábra 11

Az Apache-nak semmi köze sincs az apache indiánokhoz. A név egy szójáték „a patchy server”, aminek jelentése: foltozott szerver.

Apache fogalmak

Virtualhost: ugyanazon a hoszton több virtuális webszerver is futhat. Például a www.sajat és a www.honlapom webhelyet ugyanaz a webszerver szolgálhatja ki, de ez kívülről két különböző webhelynek látszik. A virtuális webszervereknek futhatnak különböző (*IPVirtualHosting*), vagy azonos IP címen is (*NameVirtualHosting*). Ha a doméneket külön IP címre tesszük, akkor nagy forgalom esetén később szétválaszthatjuk külön gépekre is a webhelyeket.

SSL: *Secure Socket Layer* (Biztonságos csatorna): a kliensgépen futó böngésző és a webszerver között titkosított adatcserét biztosít. Ez azért hasznos, mert így nem lehet a vonalat lehallgatni.

User authentication: felhasználó azonosítás. Egyes oldalakra csak bizonyos felhasználók léphetnek be, amit valamilyen azonosítással lehet megoldani. Ez leggyakrabban egy név és jelszó. Az azonosítás LDAP protokoll (címtárkezelés), vagy adatbázis segítségével is történhet.

Modulok: Csak azokat a részek töltődnek be a szerverprogram indításakor, melyek számunkra szükségesek. Így erőforrást takarítha-tunk meg. Az Apache modulok használatuk módjában is eltérnek a kernel moduloktól, mert csak induláskor tölthetőek be, futás közben nem. Ilyenkor az Apache démont (httpd) újra kell indítanunk.

Az Apache kofigurálása

UHU-Linux operációs rendszerben a szerkesztendő Apache konfigurációs állományok: ***/etc/apache/httpd.conf*** és az ***/etc/apache/main.conf.***

A httpd.conf állományt szerkesztése

```
# A konfigurációs - lock és a pid - fájlokat tartalmazó könyvtárak

ServerRoot                                                    "/etc/apache"
LockFile                                          "/var/lock/apache/accept.lock"
PidFile                                                  "/var/run/apache.pid"


# Az az idő másodpercben, ami alatt egy átvitelnek be kell fejeződnie. Időtúllépés
esetén  az átvitelt sikertelennek minősíti a szerver

Timeout                                                                     300


# A szerver Timeout után bontja HTTP kapcsolatot, ha egy kért anyagot elküldött a
kliensnek. Értéke On vagy Off lehet. A kliens kérheti a kapcsolat bontásának mel-
lőzését, On beállításnál a szerver elfogadja a kérést

KeepAlive                                    On


# Megadhatjuk, hogy hány anyag küldéséig tartja fenn a kapcsolatot a szerver. Nul-
la megadásával ezt a funkciót kikapcsolhatjuk, ami viszont nem javasolt

MaxKeepAliveRequests 500


# A két küldési folyamat közötti idő beállítása másodpercben. Ha eltelik, akkor a
szerver lebontja a kapcsolatot. Csak KeepAlive On beállításnál érvényesül

KeepAliveTimeout                                  60


# A következő opció lehetővé teszi, hogy több különböző virtuális szervert működ-
tessen különböző neveken egy Apache webszerveren. A szerverhez küldött kérésben
nemcsak a kért anyag azonosítója, hanem a hívott szerver neve is szerepel, az
Apache eldönti, hogy melyik virtuális szerver részére jött az üzenet

Namevirtualhost                                              192.168.1.200:80
#                    Namevirtualhost                    *:443


# További konfigurációs állományok helye

Include                                             /etc/apache/mods/loaders/*.load
Include                                                /etc/apache/mods/conf/*.conf
Include                                                   /etc/apache/conf.d/*.conf
Include                                                       /etc/apache/main.conf
Include                                                  /etc/apache/vhosts/*.vhost
```

A main.conf állományt szerkesztése

Az Apache démont rootként kell indítani, mert az első 1024 port (privilegizált portok) csak root jogokkal érhetőek el, és a naplófájlok is csak így írhatók. Biztonsági szempontból viszont fontos, hogy egyébként ne legyen root joga az Apache-nak.

```
# Ennek a felhasználónak és csoportnak a jogait kapja meg az Apache

User                                                                    www
Group                                                                   www

# Erre az e-mail címre küldi az Apache a rendszerrel kapcsolatos üzeneteket.
# Az alapbeállítás root@localhost.

ServerAdmin                                                melegak@szerver.sajat

# A hivatkozások kiegészítésének engedélyezése

UseCanonicalName                                                        On

# A weblapok gyökérkönyvtára:

DocumentRoot                                                  "/var/www/html"

# Az  Options FollowSymLinks a szimbolikus linkek követését engedélyezi.
# A korábbi beállítások felülbírálását az  AllowOverride None nem engedélyezi.

<Directory                                                              />
                                    Options                  FollowSymLinks
                                      AllowOverride                     None
</Directory>

# A felhasználók saját weboldalaikat a /home/felhasználó/public_html mappában
# helyezhetik el, amelyeket a webről a domainnév/~felhasználónév címen érhetik el

UserDir                                                          public_html

# Ha egy URL-ben csak könyvtárnevet adunk meg, akkor a szerver az itt
# felsorolt fájlokat keresi a megadott sorrendben: index.html, index.php

DirectoryIndex                        index.html                    index.php

# A hozzáférési jogokat a .htaccess fájlban lehet megadni.

AccessFileName                                                      .htaccess

# „On” esetén az IP cím helyett a naplófájlokba doménnév kerül,
# ami viszont jelentősen lassítja a rendszert.

HostnameLookups                                                         Off

# Naplófájl helye, amely a hibákat tartalmazza, root tulajdonában van.

ErrorLog                                             /var/log/apache/error.log
```

Virtuális kiszolgáló létrehozása

A virtuális kiszolgáló egy olyan webszerver, amelyik más doménnéven érhető el, de ugyanazon az Apache webszerveren fut. Legyen most a virtuális webszerverünk elérhetősége **zeus.sajat**

Létrehozása lépésenként a következő:

1. Hozzuk létre és szerkesszük meg az **/etc/apache/vhosts/zeus.sajat.vhost** fájlt:

```
root@szerver:~# cat > /etc/apache/vhosts/zeus.sajat.vhost
```

```
<VirtualHost zeus.sajat:80>
        ServerAdmin     melegak@szerver.sajat
        DocumentRoot    /var/www/zeus
        ServerName      zeus.sajat
        ErrorLog        /var/log/zeus/error.log
        CustomLog       /var/log/zeus/acces.log common
</VirtualHost>
```

2. Hozzuk létre a szükséges könyvtárakat és írjuk meg a Zeus virtualhost nyitóoldalát, a ***/var/www/zeus/index.html*** fájlt:

```
root@szerver:~# mkdir /var/www/zeus /var/log/zeus
```

```
root@szerver:~# cat > /var/www/zeus/index.html
<HTML>
<BODY>
Ez itt a Zeus virtuális kiszolgáló honlapja!
<BODY>
<HTML>
```

3. Jegyezzük be alias-ként a névszerver ***/etc/bind/db.sajat*** zónaállományába a **zeus** nevet a névfeloldás biztosítására:

```
root@szerver:~# cat >> /etc/bind/db.sajat
zeus  IN    CNAME szerver
```

4. Aktiváljuk újra a névszervert, és ellenőrizzük a helyes működését:

```
root@szerver:~# service named reload
Névszerver (named) konfigurációs fájljainak újraaktiválása...[ OK ]
```

```
root@szerver:~# ping -c 1 zeus.sajat
```

5. Indítsuk újra az Apache webszervert:

```
root@szerver:~# service apache restart
```

6. Nézzük meg a weboldalunkat egy böngészővel! Gépeljük be a kliensen egy terminálba:

```
user@kliens:~$ links
```

ESC leütése után Fájl > URL-re ugrás, gépeljük be a címet:

```
http://zeus.sajat
```

Azt fogjuk tapasztalni, hogy a virtualhost által szolgáltatott weboldal fog megjelenni. Ha viszont virtuális hosztot definiálunk, akkor az eredeti webszerver nem fog látszani, csak a virtuális hosztok.

KÉRDÉSEK

1. Mi webszerver, és milyen fogalmak kapcsolódnak hozzá?
2. Mi a virtuális kiszolgáló, és milyen célt szolgál?
3. Sorolja fel az Apache webszerver konfigurációs állományait!
4. Hogyan bírálhatja felül egy felhasználó a webszerver beállításait?

FELADAT

Oldjuk meg, hogy az előzőekben létrehozott **zeus.sajat** virtualhost mellett elérhető legyen www.sajat néven az Apache által szállított eredeti weboldal is!

MEGOLDÁS

1. A virtuális szerverünkön másoljuk át a **zeus.sajat.vhost** fájlt **www.sajat.vhost** néven ugyanabba a könyvtárba, így nem kell annyit gépelnünk:

```
root@szerver:~# cd /etc/apache/vhosts/
root@szerver:/etc/apache/vhosts# cp zeus.sajat.vhost www.sajat.vhost
```

2. Szerkesszük át rootként a **www.sajat.whost** fájlt úgy, hogy az eredeti weboldal jellemzőire mutasson:

```
<VirtualHost www.sajat:80>
        ServerAdmin     melegak@szerver.sajat
        DocumentRoot    /var/www/html/
        ServerName      www.sajat
        ErrorLog        /var/log/apache/error.log
        CustomLog       /var/log/apache/access.log common
</VirtualHost>
```

3. Módosítsuk ismét az /etc/bind/db.sajat állományt:

```
root@szerver:~# cat >> /etc/bind/db.sajat
www     IN      CNAME   szerver
```

4. Aktiváljuk a névszervert, indítsuk újra a webszervert, és próbáljuk ki egy böngészővel, hogy mi történik a http://www.sajat és http://zeus.sajat begépelésekor. Ha mindent jól csináltunk, akkor tökéletesen fog működni mindkét weboldal.

Portál kialakítása

Az Interneten számos olyan PHP alapú ingyenes portálrendszer érhető el, amelyek segítségével igényes kivitelű saját weboldal, vagy például távoktatási rendszert hozhatunk létre. A telepítéshez általában működő Apache webszerver, PHP és MySQL (vagy postgresql) adatbázis szerver szükséges.

A PHP beüzemelése

Az előző fejezetben már telepítettük az Apache webszervert és annak moduljait. A függőségek miatt települt a PHP csomag is. A PHP működésének kipróbálásához készítsük el a /var/www/html/index.php állományt:

```
root@szerver:~# cat > /var/www/html/index.php
```

```
<?php
print "Hello, ez itt a PHP nyitólapja";
?>
```

A PHP kipróbálásához gépeljük be a böngészőbe:

```
http://www.sajat/index.php
```

Ha a PHP működik, akkor a „Hello, ez itt a PHP nyitólapja” szöveg jelenik meg a böngésző ablakban.

A MSQL szerver telepítése és használatba vétele

Győződjünk meg róla, hogy a mysql csomag telepítve van-e a rendszerünkre:

```
root@szerver:~# dpkg -l mysql
No packages found matching mysql.
```

Telepítsük a mysql csomagokat a következő paranccsal:

```
root@szerver:~# apt-get install mysql*
```

Indítsuk el a mysqld szolgáltatást parancssorból:

```
root@szerver:~# service mysqld start
Mysql adatbázis szerver (mysqld) indítása     [ OK ]
```

Ha azt szeretnénk, hogy a számítógép újra indításakor is legyen mysql szerverünk, akkor adjuk meg a futási szinteket :

```
root@szerver:~# echo Runlevels=2345 >/etc/runlevel.d/custom/mysqld.service
```

A mysql kliens segítségével jelentkezzünk be a MySQL szerverre, ezzel ellenőrizzük a működését:

```
root@szerver:# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9 to server version: 4.0.17

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Jelszó nélkül be tudtunk jelentkezni, ez biztonsági szempontból a továbbiakra nézve nem túl bíztató. A `mysql>` prompt után parancsokat adhatunk ki. Jelentkezzünk ki a karakteres kliensből:

```
mysql> exit
Bye
root@szerver:#
```

Mint az előbbiekben meggyőződtünk róla, a mysql telepítése után az adatbázishoz bárki jelszó nélkül, root jogokkal hozzáférhet. En-

nek az állapotnak a megváltoztatásával kezdjük az adatbázis szerverünk használatba vételét. Először a mysql root részére jelszót adunk meg (ne keverjük a rendszerünk adminisztrátorával):

```
root@szerver:~# mysqladmin -u root password 'kamion'
```

Ha ezután akarunk bejelentkezni, akkor ez jelszó nélkül már nem fog menni:

```
root@szerver:~# mysql
ERROR 1045: Access denied for user: 'root@localhost' (Using password: NO)
```

A bejelentkezés jelszóval (ez a root csak a mysql programra vonatkozik):

```
root@szerver:~# mysql -u root -p
Enter password:
mysql>
```

Mysql webes admnisztráció: phpMyAdmin

A mysql szerver adminisztrálásához számos grafikus felület készült (például mysqlcc, phpMyAdmin, knoda). Működik a weszerverünk a PHP is telepítve van, ezért most a phpMyAdmin programot választjuk. Letöltése például innen lehetséges:

```
http://www.phpmyadmin.net/home_page/
```

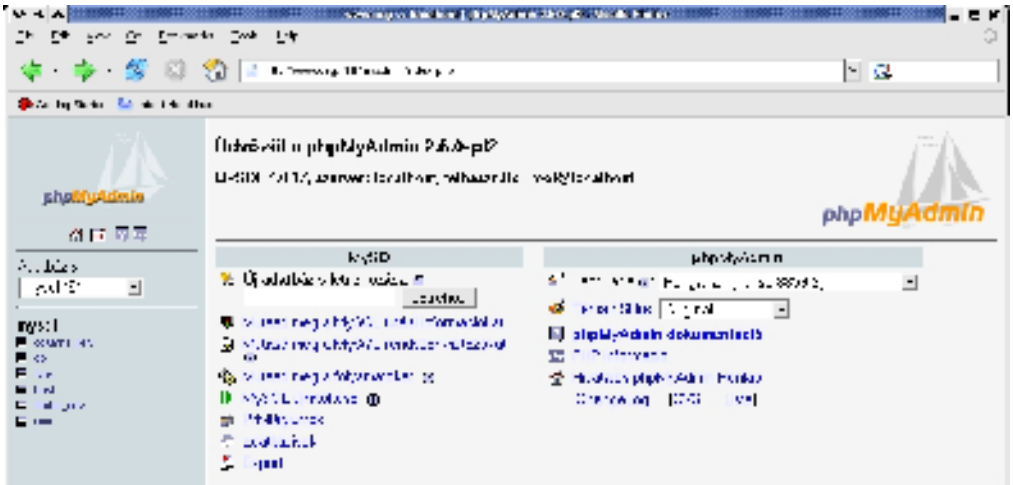
Letöltés után csomagoljuk ki a letöltés könyvtárában állva a fájlt, majd a kicsomagolt könyvtárunk tartalmát mozgassuk át a /var/www/html/PHPmyadmin könyvtárba:

```
root@szerver:~# tar -xzf phpMyAdmin-2.6.0-pl2.tar.gz
root@szerver:~# mv phpMyAdmin-2.6.0-pl2 /var/www/html/PHPmyadmin
```

Lépjünk be a `/var/www/html/PHPmyadmin` könyvtárba, és módosítsuk a `config.inc.php` nevű állomány alábbi sorait:

```
$cfg['PmaAbsoluteUri'] = 'www.sajat/PHPmyadmin';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'kamion';
```

A kliensen böngészőbe begépelve: „http://www.sajat/PHPmyadmin” megjelenik az adminisztrációs felület:



Ábra 12

A moodle e-learnig portál kialakítása

A PHP alapú portálrendszerek közül most ennek az önálló tanulásra is alkalmas rendszernek a beüzemelésével foglakozunk. Először letöltjük a telepítő csomagot innen:

`http://moodle.org`

Virtuális webkiszolgáló

Készítsünk a moodle számára egy virtuális kiszolgálót az Apache webszerveren. A szerverünkön másoljuk át a `/etc/apache/vhosts/www.sajat.vhost` állományt `moodle.sajat.vhost` néven:

```
root@szerver:~# cd /etc/apache/vhosts
root@szerver:~# cp www.sajat.vhost moodle.sajat.vhost
```

Szerkesszük át rootként a `moodle.sajat.whost` fájlt úgy, hogy a leendő moodle weboldalra mutasson:

```
<VirtualHost moodle.sajat:80>
        ServerAdmin     melegak@szerver.sajat
        DocumentRoot    /var/www/moodle
        ServerName      moodle.sajat
        ErrorLog        /var/log/moodle/error.log
        CustomLog       /var/log/moodle/access.log common
</VirtualHost>
```

Készítsük el a szükséges könyvtárakat:

```
root@szerver:~# mkdir /var/www/moodle
root@szerver:~# mkdir /var/log/moodle
```

Módosítsuk az /etc/bind/db.sajat állományt a névfeloldás biztosítása érdekében:

```
root@szerver:~# cat >> /etc/bind/db.sajat
moodle    IN      CNAME   szerver
```

Indítsuk újra a névszervert és a webszervert:

```
root@szerver:~# service named restart
root@szerver:~# service apache restart
```

Moodle adatbázis létrehozása

A moodle adatbázist elkészíthetnénk a phpMyAdmin segítségével is, de most a karakteres klienst használjuk. Először bejelentkezünk a mysql szerverre, majd létrehozzuk a `moodle` adatbázist, használatba vesszük, azután létrehozzuk a `moodleadmin` nevű felhasználót, `kamion` jelszóval és a megfelelő jogokkal a `moodle` előtaggal kezdődő táblákhoz. Végül kilépünk a mysql kliensből.

```
root@szerver:~# mysql -u root -p
Enter password:

mysql> CREATE DATABASE moodle;
Query OK, 1 row affected (0.00 sec)

mysql> USE moodle
Database changed

mysql> GRANT SELECT, INSERT, CREATE, UPDATE, DELETE ON moodle.*
    -> TO moodleadmin@localhost IDENTIFIED BY 'kamion';
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

Moodle konfigurálása

Töröljük a /var/www/moodle könyvtárat (hogy ne zavarjon a tartalma), majd álljunk abba a könyvtárba, ahová letöltöttük a `moodle-latest-14.tgz` csomagot, csomagoljuk ki, és mozgassuk a helyére:

```
root@szerver:~# rm -r /var/www/moodle
root@szerver:~# tar -zxf moodle-latest-14.tgz
root@szerver:~# mv moodle /var/www
```

A következőkben belépünk a `/var/www/moodle/` könyvtárba, és létrehozzuk a `config.php` állományt, egy mintafájl másolásával:

```
root@szerver:~# cd /var/www/moodle/
root@szerver:/var/www/moodle# cp config-dist.php config.php
root@szerver:/var/www/moodle# mcedit config.php
```

Módosítjuk a `/var/www/moodle/config.php` állományt:

```
$CFG->dbtype  = 'mysql';       // mysql használata
$CFG->dbhost  = 'localhost';   // helyi adatbázisszerver
$CFG->dbname  = 'moodle';      // adatbázis neve
$CFG->dbuser  = 'moodleadmin'; // adatbázis adminisztrátor
$CFG->dbpass  = 'kamion';      // adminisztrátor jelszava
$CFG->prefix  = 'mdl_';        // moodle táblák kezdete
$CFG->wwwroot = 'http://moodle.sajat';   // webes elérés
$CFG->dirroot = '/var/www/moodle';       // moodle könyvtára
$CFG->dataroot = '/var/www/moodledata';  // adatkönyvtár
```

Létrehozzuk a /var/www/moodledata könyvtárat, és a /var/www/moodle könyvtárral együtt az /etc/apache/main.conf állományban megadott felhasználó és csoport tulajdonába adjuk. Ezután mindkét könyvtárra 0777 jogot állítunk be:

```
root@szerver:~# mkdir /var/www/moodledata
root@szerver:~# chown -R www:www /var/www/moodle
root@szerver:~# chown -R www:www /var/www/moodledata
root@szerver:~# chmod -R 0777 /var/www/moodle
root@szerver:~# chmod -R 0777 /var/www/moodledata
```

Apache és PHP konfigurálása

Az `/etc/apache/main.conf` állományban állítsuk be, hogy az Apache először mindig `index.php` nevű állományt keressen és állítsuk be a magyar ékezeteket kezelő karakterkészletet:

```
DirectoryIndex  index.php  index.html index.html.var
AddDefaultCharset ISO-8859-2
```

Módosítani kell a PHP beállításait is, ezt az `/etc/php.ini` állományban tehetjük meg. A következő bejegyzések lényegesek:

```
memory_limit = 16M
magic_quotes_gpc = 1
magic_quotes_runtime = 0
file_uploads = 1
session.auto_start = 0
session.bug_compat_warn = 0
```

A beállítások végeztével indítsuk újra a webszervert:

```
root@szerver:~# service apache restart
```

Moodle telepítése

A moodle PHP alapú telepítővel rendelkezik, ennek indításához gépeljük be a böngészőbe:

```
http://moodle.ceg
```

1. Elindul a telepítő, ha ez mégsem történne meg, akkor meg lehet próbálni a `http://moodle.sajat/install.php` begépelését.
2. Megkérdezi, hogy létrehozza-e az adattáblákat: `Yes`.
3. Kiírja, hogy milyen táblákat hozott létre: `Continue`.
4. A következő képernyőn váltsunk nyelvet: Magyar és országot: Hungary, a többit később is beállíthatjuk, ezután: `Save changes`.
5. A telepítő nyelve ezután magyarra vált. Egy pár információs képernyő következik, a válaszunk mindegyiknél: `Folytatás`.
6. Elérkezünk az oldal beállításához, itt adjuk meg az oldal teljes és rövid nevét, utána: Változások mentése. Végül adjuk meg az adminisztrátor jelszavát.
7. A moodle oktatási portál felépítését ezzel befejeztük.

KÉRDÉSEK

1. Általában milyen feltételek teljesülése esetén lehet webportált építeni?
2. Ismertesse a mySQL adatbázis-szerver használatba vételének lépéseit!
3. Milyen lehetőségei vannak a phpMyAdmin nevű alkalmazásnak?
4. Mit jelen az Apache konfigurációs állományában a `DirectoryIndex` kifejezés?

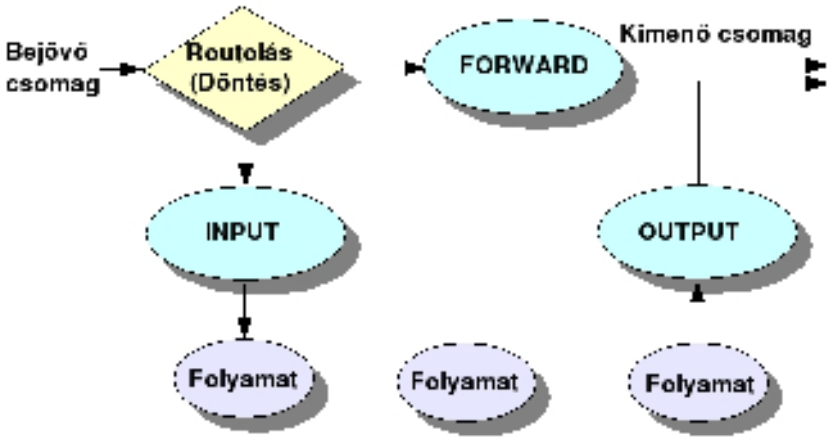
Védekezzünk a betörők ellen: tűzfal (firewall)

A tűzfalak segítségével védekezhetünk az illetéktelen behatolók ellen. A Linux operációs rendszerben van egy kernel szintű csomag–szűrő tűzfal, a netfilter. Ezt az `iptables` parancs segítségével lehet konfigurálni. Több grafikus frontend is rendelkezésre áll a beállí–tások megkönnyítése céljából.

Ellenőrizzük a tűzfalbeállítást a szerveren:

```
root@szerver:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Az iptables úgynevezett láncokból áll, nevük: **INPUT**, **FORWARD** és **OUTPUT**. A felépítést az alábbi ábra mutatja:



Ábra 13

1. A hálózati interfészen beérkező csomag célcímét megnézi a kernel, és megpróbálja kézbesíteni. Ezt nevezzük routolásnak, vagy útválasztásnak.

2. Ha a csomag célcíme ezen a gépen van, akkor a „router” az **INPUT** lánc felé irányítja. Ha ez átengedi (ACCEPT), akkor feldolgo–zásra kerül egy futó folyamat (processz) segítségével.

3. Ha a kernelben engedélyezve van a csomag továbbítás (FORWARD) és a csomag egy másik hálózati interfészre lett címezve, akkor a csomag a **FORWARD** lánc felé halad. Ha ez átengedi (ACCEPT), akkor a kernel továbbítja a kimenetre.

4. Ha egy futó folyamat akar csomagot továbbítani a gépről kifelé, akkor az **OUTPUT** láncon keresztül teszi, persze ha az átengedi (ACCEPT).

5. Minden egyéb esetben a csomag eldobásra kerül (DROP)

Pillanatnyilag nincs csomagszűrés, ellenőrizzük a kliensől a szerver nyitott portjait:

```
user@kliens:~$ nmap szerver.sajat
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-11-05 02:46 CET
Interesting ports on szerver.sajat.108.168.192.in-addr.arpa (192.168.108.103):
(The 1650 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
110/tcp  open  pop3
443/tcp  open  https
631/tcp  open  ipp
3306/tcp open  mysql
```

Az nyitott portok megfelelnek a szerveren pillanatnyilag futó démonok szabványos kommunikációs portjainak. Amennyiben például névfeloldást kifelé nem akarunk szolgáltatni, akkor teljesen felesleges az 53-as portot kifelé nyitva hagyni.

UHU-Linux esetében a **Guarddog** (őrkutya) nevű grafikus előtét már telepítéskor felkerül a rendszerre.

Indítsuk el a menüből a tűzfalbeállító programot a szerveren: `Internet > Tűzfalak > Guarddog`, a root jelszó begépelése után megjelenik a kezelőfelület.

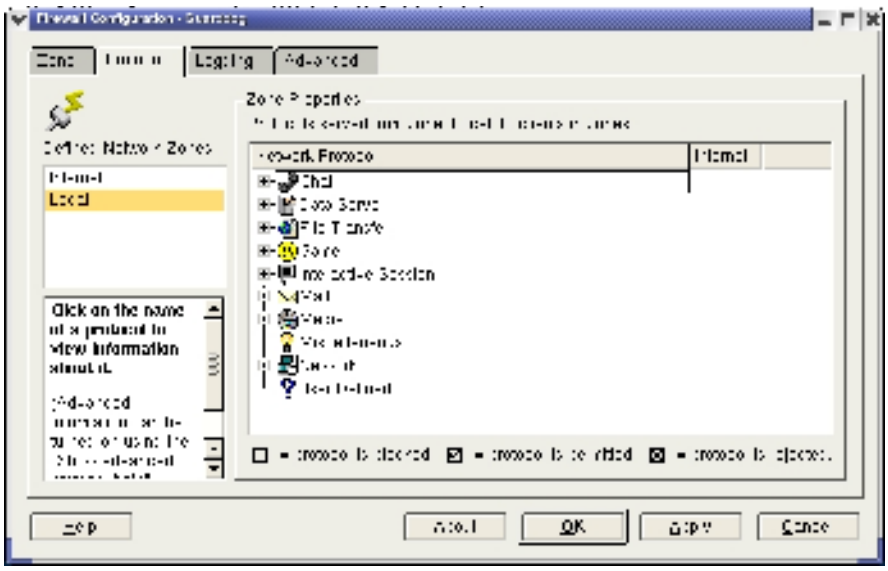
A Guarddog elindítható akkor is, ha a szerveren nem is működik grafikus munkakörnyezet. Ilyenkor a kliens grafikus terminájáról ssh-val bejelentkezünk a szerverre, su paranccsal root jogokat szerzünk, majd innen indítjuk a tűzfal frontendet:

```
root@szerver:~# guarddog
```

Természetesen ez a módszer csak akkor működik, ha az X forwarding a szerver és a kliens oldalon is engedélyezve van.

A hálózati zónák közül a **Local** határozza meg, hogy a szerverről milyen szolgáltatást nyújthatunk a helyi hálózat felé, az **Internet** zóna beállítása pedig a bejövő forgalom szűrésére alkalmas.

Ellenőrizzük, hogy azok a szolgáltatások, amelyeket kifelé engedélyezni fogunk, fussanak a szerveren: webszolgáltatás (apache), levelezés (postfix, cyrus), távoli elérés (ssh), névszolgáltatás (bind). Ha a Guarddog grafikus kezelőfelületét a kliensről indítottuk, akkor a Local zónában mindenképpen engedélyezzük az ssh szolgáltatást, mert különben kizárjuk magunkat a szerverről a tűzfal aktiválása után.



Ábra 14

Váltsunk át először a `Protocol` fülre, majd a a Local zóna Interactiv Session kategóriában engedélyezzük az SSH szolgáltatást. Végül sz `Apply` gombra kattintva minden egyéb forgalmat letiltunk kifelé. Ellenőrizzük a nyitott portokat az **nmap** paranccsal a kliensről:

```
user@kliens:~$ nmap szerver.melega

Starting nmap 3.77 ( http://www.insecure.org/nmap/ ) at 2005-02-04 09:39 CET
Note: Host seems down. If it is really up, but blocking our ping probes, try
-P0Nmap run completed -- 1 IP address (0 hosts up) scanned in 3.032 seconds
```

Az nmap most nem működik, hiszen a lokális hálózat felé a tűzfalon tiltva van a ping, illetve a http protokoll is.

Engedélyezzük a Local zóna kiválasztása után azokat a protokollokat, amelyeken kifelé szolgáltatni akarunk:

Csoport	Protokoll	Jelentése
File Transfer	http	Web szolgáltatás
Interactive Session	SSH	Secure Shell, távoli bejelentkezés
Mail	POP3	Levelek letöltése a szerverről
Mail	SMTP	Levelek küldése
Network	DNS	Névfeloldás
Network	Ping	Hálózat ellenőrzése

Ezután **Apply**, ellenőrizzük ismét a nyitott portokat a kliensről az **nmap** paranccsal, majd próbáljuk ki, hogy az engedélyezett szolgáltatások elérhetők-e.

```
melegak@kliens:~$ nmap szerver.melega

PORT     STATE  SERVICE
22/tcp   open   ssh
25/tcp   open   smtp
53/tcp   open   domain
80/tcp   open   http
110/tcp  open   pop3
8000/tcp closed http-alt
8080/tcp closed http-proxy
```

A tűzfal kikapcsolása az **Advanced** fül **Disable Firewall** kapcsolóval lehetséges, ezután **Apply**. Ilyenkor az összes használt portunk megnyílik, és nincs tűzfalunk.

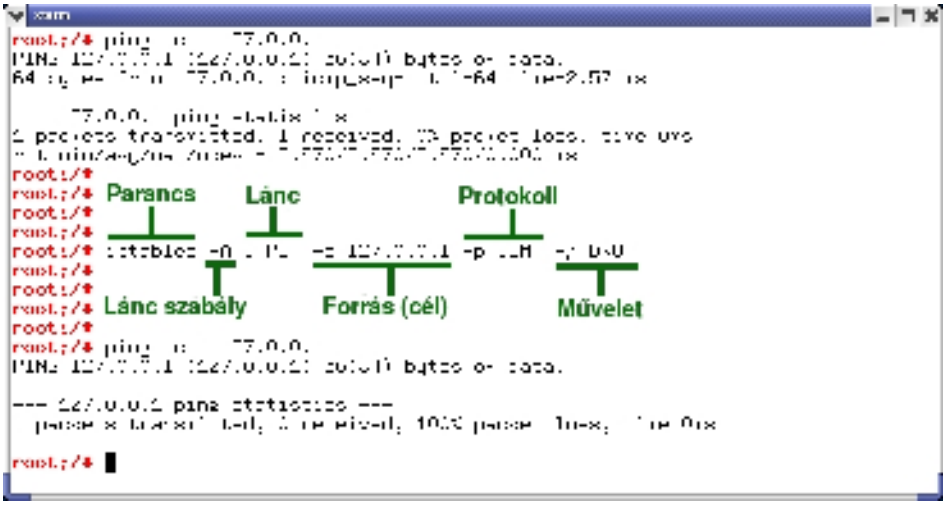
Ábra 15

A Guarddog az **/etc/rc.firewall** futtatható állományt szerkeszti, ebben hozza létre a régebbi kernelhez szükséges ipchains, illetve az újabb kernelekkel együttműködő **iptables** bejegyzéseket.

A tűzfal (Netfilter) beállítása parancssorból

A tűzfalak finomhangolásához szükséges lehet a parancssori beállítás, ezért röviden ezzel is foglalkozunk.

Minden operációs rendszer létrehoz egy visszacsatolt (loopback) interfészt, aminek egységesen 127.0.0.1 az IP címe. Így futtathatók és tesztelhetők az IP protokollt igénylő alkalmazások (pl. böngésző). A következő példában bemutatjuk, hogy a tűzfal segítségével hogyan akadályozzuk meg a loopback interfész pingelhetőségét.



Ábra 16

Az első **ping** után visszaérkezik a csomagra a válasz. Az **iptables** parancs után látjuk, hogy a **ping** hatástalan, mert a tűzfal kiszűri. Ha a parancsot egy mondatban fogalmazzuk meg, akkor jelentése e következő:

„ Add hozzá a bemeneti (INPUT) lánchoz azt a szabályt, ami a 127.0.0.1 címről érkező (forrású) ICMP protokollal rendelkező (a ping ezt a protokollt használja) csomagokat eldobja (DROP). A következőkben sorra vesszük ezeket a fogalmakat.

Láncműveletek

Szabály	Magyarázat
-N	Új lánc létrehozása
-X	Üres lánc törlése
-P	Irányelv megváltoztatása beépített láncon
-L	Egy lánc szabályainak listázása
-F	A lánc összes szabályának törlése
-Z	Csomag és byte-számlálók nullázása

Lánc szabályok műveletei

Szabály	Jelentés	Magyarázat
-A	Add	Új szabály hozzáfűzése a lánchoz
-I	Insert	Új szabály beszúrása egy láncba, adott pozición
-R	Replace	Egy szabály helyettesítése egy másikkal
-D	Delete	Szabály törlése a láncból

Lánc típusok

Lánctípus	Magyarázat
INPUT	Bemeneti lánc, az illeszkedő csomagot egy folyamatnak küldi
FORWARD	Továbbító lánc, illeszkedő csomagot egy másik interfészre küldi
OUTPUT	Kimeneti lánc, illeszkedő csomagot folyamattól hálózatra küldi
privat	Egyéni lánc, felhasználó definiálja, kisbetűvel írandó

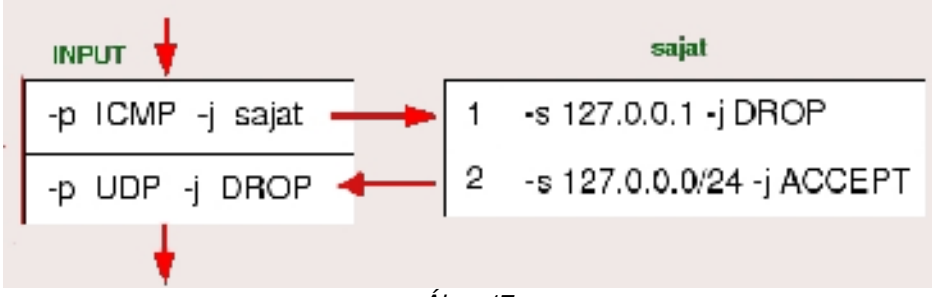
Szűrési beállítások

Kapcsoló		Jelentése	Példa
-s	--source	Forráscím	-s 192.168.108.0/24 vagy -s www.sajat
-d	--destination	Célcím	-d 127.0.0.1
-p	--protocol	Protokoll	-p TCP vagy -p tcp vagy -p 80
-i	--in-interface	Bejövő interfész	-i eth0
-o	--out-interface	Kimenő interfész	-o eth1
-f	--fragment	Töredék csomag	Szabályok a töredékcsomagokra

Műveletek a csomagokkal

Művelet	Magyarázat
-j ACCEPT	Csomag engedélyezése
-j DROP	Csomag eldobása
-j REJECT	Csomag eldobása hibaüzenet küldéssel
-j LOG	Csomagforgalom naplózása
-j QUEUE	Csomag várakozási sorba állítása

Saját lánc beillesztése



Ábra 17

A saját lánc nevét mindig kisbetűvel írjuk, legyen a neve „privat”. A példában hivatkozunk az INPUT láncban egy saját láncra, tiltjuk a bemeneti láncon a localhost pingelését és az UDP protokollt. A fenti ábra szerinti beállítások, és a saját lánc létrehozásának lépései a szerveren a következők:

1. A tűzfal pillanatnyi állapotának lekérdezése, ennek kimenete a Guarddog használata után eléggé hosszú lehet:

```
root@szerver:~# iptables -L
```

2. Írunk a szerveren egy olyan szkriptet, amelyik először törli az összes szabályt, majd mindent engedélyez a beépített láncokon:

```
root@szerver:~# cat > /sbin/iptables.reset

#!/bin/bash

# /sbin/iptables.reset
# root jogokkal futtasd ezt a szkriptet!

# Töröljük az összes szabályt és az üres láncokat
iptables -F
iptables -X
iptables -Z

# Mindent engedélyezünk a beépített láncokon
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

CTRL+D a fájl szerkesztésének befejezéséhez, majd a megfelelő jogok beállítása:

```
root@szerver:~# chmod 0700 /sbin/iptables_reset
```

3. Futtassuk a szerveren a szkriptet, majd ismét nézzük meg a láncok pillanatnyi állapotát:

```
root@szerver:~# iptables.reset
root@szerver:~# iptables -L
```

4. Saját lánc létrehozása és szabályok hozzáadása. A localhost tiltva lesz, a localdomain elérése a localhost kivételével engedélyezett:

```
root@szerver:~# iptables -N privat
root@szerver:~# iptables -A privat -s 127.0.0.1 -j DROP
root@szerver:~# iptables -A privat -s 127.0.0.0/8 -j ACCEPT
```

5. Saját láncra hivatkozás az INPUT láncban, az ICMP protokollra vonatkozóan. Ezt a protokollt használja a ping parancs. Végüla az UDP protokollok tiltása:

```
root@szerver:~# iptables -A INPUT -p ICMP -j privat
root@szerver:~# iptables -A INPUT -p UDP -j DROP
```

6. Ellenőrizzük a szerveren, hogy a localhostot lehet-e pingelni:

```
root@szerver:~# ping localhost
PING localhost.localdomain (127.0.0.1) 56(84) bytes of data.

--- localhost.localdomain ping statistics ---
502 packets transmitted, 0 received, 100% packet loss, time 501335ms
```

7. Listázzuk ki az érvényben lévő tűzfalszabályokat a szerveren:

```
root@szerver:~# iptables -L

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
privat     icmp --  anywhere             anywhere
DROP       udp  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain privat (1 references)
target     prot opt source               destination
DROP       all  --  localhost.localdomain  anywhere
ACCEPT     all  --  127.0.0.0/8          anywhere
```

Tűzfal a gyakorlatban

A következő iptables script olyan átjáróként működő szerver esetén is alkalmazható, ahol egy publikus IP címet tartalmazó hálózati interfészen jön be az Internet, és nem publikus IP címmel rendelkező hálózati kártyán keresztül szolgálunk ki egy belső alhálózatot. Indítási paraméterei: **start | stop | open**

```
#!/bin/bash

# [VÁLTOZÓK BEÁLLÍTÁSA] ------------------------------------------------------

# Az 'iptables' parancs helyének tárolása "IPTABLES" változóban
IPTABLES="/usr/sbin/iptables"

# Hibaüzenet, ha nincs az adott helyen "iptables" parancs
if ! [ -x $IPTABLES ] ;
then echo $IPTABLES "- nem létező parancs";
exit 1;
fi

# A külvilág felé néző hálózati interface lekérdezése,
# ezen van az alapértelmezett átjáró beállítva:
DEF_IFACE=`route|grep default|awk '{print $8}'`

# A külvilág felé néző hálózati interface IP címének megállapítása:
DEF_IP=`ifconfig $DEF_IFACE|grep inet|awk '{print $2}'|cut -d: -f2`

# Privát IP címünk:
PRIV_IP="192.168.1.200"

# Privát IP címmel rendelkező hálózat(ok), amelyek rajtunk
# keresztül csatlakoznak:
NAT_FOR_NET=`echo $PRIV_IP | cut -d. -f1,2,3`.0/24
echo -n "A privát hálózat: $NAT_FOR_NET"

# Engedélyezett INPUT portok minden interfészen:
ACCEPT_TCP="ftp,ssh,smtp,www,https,pop3"
ACCEPT_UDP="ftp"

# További engedélyezett INPUT portok a privát interfészen:
```

```
ACCEPT_TCP_PRIV="domain"
ACCEPT_UDP_PRIV="domain"

# Válasz a ping-re: igen=1, nem=0
PING_REPLY=0

# Maximális ping sűrűség:
MAX_PING_SPEED=5/s

# Erre a ping-re válaszolunk: ping -s 1357 IP-címünk
MAGIC_PING=1357

# A ping csomagok maximális mérete:
MAX_PING_SIZE=84

# Tiltott hosztok, ezekkel nem állunk szóba:
IGNORE_HOSTS=""

# Tiltott portok, ezeken nem kommunikálunk:
IGNORE_PORTS_TCP=""
IGNORE_PORTS_UDP=""

# Ha valaki megkísérel egy tiltott portra csatlakozni, elbújunk
# előle, igen=1 nem=0 (alapértelmezett kernel nem tartalmazza)
HIDE_FROM_HOSTS=0
HIDE_INTERVAL=600

# Sikertelen csatlakozási kísérletek logolása:
LOG_ILLEGAL=0
LOG_ILLEGAL_TCP="0:1024"
LOG_ILLEGAL_UDP="0:1024"

# Átjáró vagyunk? igen=1, nem=0:
FORWARD_CONNECTIONS=1

# Publikus IP címmel rendelkező hálózat, amely rajtunk
# keresztül csatlakoznak az Internetre:
FORWARD_FOR_NETWORKS=`echo $DEF_IP | cut -d. -f1,2,3`.0/24

# Publikus IP címmel rendelkező, rajtunk keresztül csatlakozó
# gépeken kívülről elérhető portjai:
```

```
FORWARD_PORTS_TCP=""
FORWARD_PORTS_UDP=""

# Az új bejövő kapcsolatok logolása: igen=1, nem=0
LOG_LEGAL=1

# Amennyiben a kernel nem támogatja a multiport opciót, akkor
# átjavítandó "--dport"-ra, a portlistákban meg a vesszőket
# szóközre kell cserélni!
USE_MULTIPORT="-m multiport --dports"


# [TűZFAL SZABÁLYOK] ------------------------------------------------------

# Paraméterek szerinti elágazás: start | stop | open
case „$1” in

# Start ág kezdete, tűzfal aktiválása
start)
echo -n „...Tűzfal indítása"

# Régi szabályok és saját láncok törlése, számlálók nullázása:
$IPTABLES -F
$IPTABLES -X
$IPTABLES -Z

# Alapértelmezés: befelé semmit, kifelé bármit megengedünk
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT ACCEPT

# Loopback felület engedélyezése:
$IPTABLES -I INPUT -i lo -j ACCEPT

# Meglévő kapcsolatokkal összefüggő csomagok engedélyezése:
$IPTABLES -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
# A tiltott csomagok eldobása:
for bad_hosts in $IGNORE_HOSTS;
 do $IPTABLES -A INPUT -s $bad_hosts -j DROP;
done
for port_number in $IGNORE_PORTS_TCP;
 do $IPTABLES -A INPUT -p tcp --dport $port_number -j DROP;
done
for port_number in $IGNORE_PORTS_UDP;
 do $IPTABLES -A INPUT -p udp --dport $port_number -j DROP;
done

# Pingelhetőség beállítása:
if (( PING_REPLY == 1 ));
then
$IPTABLES -A INPUT -p icmp --icmp-type echo-request -m length --length `echo $MAX_PING_SIZE | awk '{print $0+1}'`:65535 -j DROP
$IPTABLES -A INPUT -p icmp --icmp-type echo-request -m limit --limit $MAX_PING_SPEED -j ACCEPT
else
 if (( $MAGIC_PING > 0 ));
 then
 $IPTABLES -A INPUT -p icmp --icmp-type echo-request -m length --length `echo $MAGIC_PING | awk '{print $0+28}'` -j ACCEPT;
 fi;
$IPTABLES -A INPUT -p icmp --icmp-type echo-request -j DROP
fi;

# Bármilyen egyéb ICMP csomagot elfogadunk:
$IPTABLES -A INPUT -p icmp --icmp-type ! echo-request -j ACCEPT

# A tűzfal által elfogadott csomagok logolása:
#if (( LOG_LEGAL == 1 )); then
#for port_number in $ACCEPT_TCP;
#do $IPTABLES -A INPUT -d $DEF_IP -p tcp -m state --state NEW $USE_MULTIPORT $port_number -j LOG --log-prefix "[FW_NEW_TCP] ";
#done
#for port_number in $ACCEPT_UDP;
#do $IPTABLES -A INPUT -d $DEF_IP -p udp -m state --state NEW $USE_MULTIPORT $port_number -j LOG --log-prefix "[FW_NEW_UDP] ";
#done
#fi
```

```
# A tűzfal által minden interfészen elfogadott csomagok szabályai:
for port_number in $ACCEPT_TCP;
do $IPTABLES -A INPUT -p tcp -m state --state NEW $USE_MULTIPORT $port_number -j ACCEPT;
done

for port_number in $ACCEPT_UDP;
do $IPTABLES -A INPUT -p udp -m state --state NEW $USE_MULTIPORT $port_number -j ACCEPT;
done

# A tűzfal által a privát interfészen elfogadott további csomagok szabályai:
for port_number in $ACCEPT_TCP_PRIV;
do $IPTABLES -A INPUT -p tcp -s $NAT_FOR_NET -m state --state NEW $USE_MULTIPORT $port_number -j ACCEPT;
done

for port_number in $ACCEPT_UDP_PRIV;
do $IPTABLES -A INPUT -p udp -s $NAT_FOR_NET -m state --state NEW $USE_MULTIPORT $port_number -j ACCEPT;
done

# Nem engedélyezett csomagok logolása:
if (( LOG_ILLEGAL == 1 ));
then
 for port_number in $LOG_ILLEGAL_TCP;
 do $IPTABLES -A INPUT -p tcp -m state --state NEW --dport $port_number -j LOG --log-prefix "[Illegális TCP] ";
 done
 for port_number in $LOG_ILLEGAL_UDP;
 do $IPTABLES -A INPUT -p udp -m state --state NEW --dport $port_number -j LOG --log-prefix "[Illegális UDP] ";
 done
fi

# A tiltott csomagot küldő gép elfogadott csomagjai
# sem érhetik el a gépünket:
if (( HIDE_FROM_HOSTS == 1 ));
then
$IPTABLES -A INPUT -m state --state NEW -m recent --set -j $DROP
$IPTABLES -I INPUT -m recent --update --seconds $HIDE_INTERVAL -j $DROP
fi
```

```
# IP forward és NAT szabályok:
if (( $FORWARD_CONNECTIONS  == 1 ));
then
 for DEST_NET in $FORWARD_FOR_NETWORKS;
  do
   for port_number in $FORWARD_PORTS_TCP;
   do $IPTABLES -A FORWARD -o ! $DEF_IFACE -d $DEST_NET -p tcp -m state --state NEW $USE_MULTIPORT $port_number -j ACCEPT;
   done
   for port_number in $FORWARD_PORTS_UDP;
   do $IPTABLES -A FORWARD -o ! $DEF_IFACE -d $DEST_NET -p udp -m state --state NEW $USE_MULTIPORT $port_number -j ACCEPT;
   done
   $IPTABLES -A FORWARD -i ! $DEF_IFACE -s $DEST_NET -m state --state NEW -j ACCEPT
  done

 $IPTABLES -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

 for NAT_NET in $NAT_FOR_NET;
  do $IPTABLES -t nat -A POSTROUTING -o $DEF_IFACE -s $NAT_NET -j MASQUERADE;
  $IPTABLES -A FORWARD -i ! $DEF_IFACE -s $NAT_NET -m state --state NEW -j ACCEPT
  done
 echo 1 > /proc/sys/net/ipv4/ip_forward
else
 echo 0 > /proc/sys/net/ipv4/ip_forward
fi

# Vége a start ágnak
;;

# [STOP ÁG] ------------------------------------------------------------------
# Stop ág kezdete
stop)
echo -n „...A tűzfal zárva, kivéve ssh”

$IPTABLES -F
$IPTABLES -A INPUT -p tcp -s $NAT_FOR_NET -m state --state NEW --dport ssh -j ACCEPT;
$IPTABLES -A OUTPUT -p tcp -d $NAT_FOR_NET -m state --state NEW --dport ssh -j ACCEPT;
# Stop ág vége
;;
```

```
# [OPEN ÁG] ------------------------------------------------------------------
# Open ág kezdete
open)
echo -n „...A tűzfal nyitva!”

$IPTABLES -F
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT
$IPTABLES -P OUTPUT ACCEPT
# Open ág vége
;;

# Többágú elágazás vége
esac

echo "...Rendben"
```

A szkriptre futási jogot kell adnunk, és root felhasználóként kell futtatnunk. Ennél természetesen sokkal bonyolultabb iptables szkriptek is léteznek, rengeteg található belőlük az Interneten.

Open LDAP

Az LDAP a "Lightweight Directory Access Protocol" (könnyű címtár elérési protokoll) kifejezés rövidítése. A címtár egy olyan adatbázis, amelyben az információt a gyakrabban olvassák, mint írják. A címtár képes az információk sokszorosítására az elérhetőség , és a rendelkezésre állás javítása érdekében, miközben a válaszidő csökken. A többszörözött címtár információk egyes példányai között átmeneti eltérés megengedett, de a többszörözések (replica) végül szinkronba kerülnek.

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 teszi fel a kérdéseit. A szerver megválaszolja a kérdést, vagy egy mutatót ad vissza, 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.

Az LDAP démon: slapd

Az LDAP szerver démon neve **slapd** több háttér adatbázis használatát támogatja. Az elsődlegesen támogatott típus a BDB, ami egy nagy teljesítményű tranzakciós háttér adatbázis.

A slapd kofigurálása

A konfigurációs állomány: /etc/openldap/slapd.conf

```
# $OpenLDAP: /etc/openldap/slapd.conf
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.

# Az alapséma leírásokat tartalmazó fájl megadása

include         /etc/openldap/schema/core.schema

# Define global ACLs to disable default read access.
# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
# Olyan keresési kérelem, amelyet a helyi adatbázis nem tud kiszolgálni, át lesz
# irányítva az LDAP kapun (389) keresztül a root.openldap.org szerverre.

#referral       ldap://root.openldap.org

pidfile         /var/slapd.pid
argsfile        /var/slapd.args

# Általános elérés biztosítása. Minden bejegyzésre érvényes (minden más, használ
# ható adatbázis hozzáférés korlátozás után).

# access to dn.base="" by * read
# access to dn.base="cn=Subschema" by * read
# access to *
#       by self write
#       by users read
#       by anonymous auth
#
# Ha nem meg hozzáférési szabályokat, akkor alapértelmezett
# a „mindenki olvashat mindent” direktíva

#
# rootdn can always write!


#######################################################################
# ldbm database definitions
```

```
#######################################################################

# Új BDB típusú adatbázis definíció kezdetét jelzi

database        bdb

# A suffix opció határozza meg a kérések DN toldalékát. Több suffix sor is megad-
ható, de legalább egy szükséges minden adatbázis definícióhoz.

suffix          "dc=melega,dc=hu"

# Ez a korlátozás nélkül férhet hozzá a címtár adatbázishoz

rootdn          "cn=melegak,dc=melega,dc=hu"

# A rootdn felhasználó jelszava, ha a rootdn be van állítva egy adatbázisban.
# Kerülendő a kódolatlan jelszavak használata. A slapd számos kódolást támogat.
# A kódolt jelszó előállítása a slappasswd paranccsal történik, ez Secure Hash
# {SSHA} jelszavakat generál alapértelmezésben.
# Amennyiben SASL-t használunk, a rootpw sort ki kell szedni.

rootpw          {SSHA}ho5h3xdvY2m54SEZVrUNk42Q6GmDMWuz

# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
# A BDB adatbázis és a hozzá tartozó fájlok könyvtára

directory       /var/openldap-data

# Az adatbázis definícióinak tartalmaznia kell az szükséges indexek definícióit
# index {<attrlist> | default} [pres,eq,sub,none]
index   objectClass     eq
```

```
root@szerver:/etc/openldap# openssl req -new -x509 -nodes -out tanusitvany.pem
-keyout kulcs.pem -days 365
Generating a 1024 bit RSA private key
..............++++++
.....................................................++++++
writing new private key to 'kulcs.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:hu
State or Province Name (full name) [Some-State]:X
Locality Name (eg, city) []:Budapest
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Pataky
Organizational Unit Name (eg, section) []:Kontener
Common Name (eg, YOUR name) []:szerver.melega
Email Address []:melegak@szerver.melega
root@szerver:/etc/openldap#
```

Az LDAP csomag tartalmazza az LDIF fájlok BDB formátumba konvertálásához szükséges eszközöket.

Egy LDIF fájl általában így néz ki:

```
dn: o=TUDelft, c=NL
o: TUDelft
objectclass: organization
dn: cn=Luiz Malere, o=TUDelft, c=NL
cn: Luiz Malere
sn: Malere
mail: malere@yahoo.com
objectclass: person
```

Mint látható, minden bejegyzésnek saját azonosítója van, a DN (distinguished name; megkülönböztető név). A DN a bejegyzés nevéből áll, megtoldva a név elérési útjával vissza a címtár hierarchia csúcsáig (mint egy fánál).

Az LDAP-nál az **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 is tartozhat. Például a person bejegyzést definiálja a *person* objektumosztály, de szintén definiálják az inetOrgPerson, groupOfNames és organization objektumosztályok. Az LDAPserver objektumosztály struktúráját (sémáját) meghatározza az egyes bejegyzések számára szükséges és engedélyezett attribútumok egyéni listája.

Minden, a címtárban bejegyzett egyént a *person* objektumosztályban tárolt jellemzők csoportja ír le. Más jellemzőket is lehet ugyanabban a bejegyzésben használni:

```
givenname: Jonas
surname: Salk
mail: jonass@airius.com
```

A bejegyzéshez szükséges jellemzőket tartalmaznia kell a használt objektumosztálynak. Minden bejegyzésnek tartalmaznia kell az objectClass-t, ami a bejegyzéshez tartozó objektumosztályok listáját tartalmazza.

A slapd.conf fájl három beállítási információt tartalmaz:global (általános), backend specific (háttér-specifikus), és database specific (adatbázis-specifikus). Először az általános rész kerül beállításra, ezt követi a háttér, majd az adott adatbázisra vonatkozó rész zárja a konfig fájlt.

root@szerver:~# slappasswd -h {SSHA}

New password:

Re-enter new password:

{SSHA}ho5h3xdvY2m54SEZVrUNk42Q6GmDMWuz

```
nmap localhost

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2005-02-09 20:51 CET
Interesting ports on localhost.localdomain (127.0.0.1):
(The 1654 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
53/tcp  open  domain
389/tcp open  ldap
631/tcp open  ipp
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 3.047 seconds
```

```
root@szerver:/etc/openldap# ldapsearch -x -h localhost -b 'dc=szerver,dc=melega'
'(objectclass=*)'
# extended LDIF
#
# LDAPv3
# base <dc=szerver,dc=melega> with scope sub
# filter: (objectclass=*)
# requesting: ALL
#

# search result
search: 2
result: 32 No such object

# numResponses: 1
```

Szolgáltatások

Ábra 18

A ábra példaként a webszerver szolgáltatást végző **httpd** működését mutatja be. Ha egy böngészőben begépeljük a szerver domain nevét, akkor névfeloldás után küld egy kérést a 80-as porton a szerver felé. A http démon figyeli a számára alapértelmezett portot, és a kérésre válaszul elküldi a weboldal nyitólapját, amit a böngésző letölt, majd megjelenít szöveggel, képekkel, linkekkel együtt.

Szolgáltatás még például a névfeloldás (DNS), az IP cím automatikus kiosztása (DHCP) és még hosszasan sorolhatnánk. A későbbiekben az egyes szolgáltatásokkal kapcsolatos tudnivalókra még kitérünk. Elsősorban a gépnév megváltozása miatt indítsuk újra a virtuális szervert.

A későbbiekben az egyes szolgáltatásokat csak akkor fogjuk elindítani, ha már szükség lesz rájuk.