



Utazás a Postfix körül (2. rész)

Az előző részben megismertük a Postfix szerkezetét, a levélfogadást és -továbbítást. Az elmélet után hasznos, ha tudásunkat a gyakorlatban is kipróbáljuk.

Első lépésként – mint mindig – telepítjük az alkalmazást. Erre két lehetőségünk is nyílik: vagy egy csomagkezelőt használunk, vagy pedig forráscsomagból rakjuk fel az alkalmazást. A csomagkezelővel telepített változatok előnyei közé sorolhatjuk, hogy sok beépített foltot (patch) tartalmaznak vagy az alapfordítástól eltérő kapcsolókkal fordították (szerepel benne például a MySQL- vagy OpenLDAP-támogatás). Ekkor ugyanis nem kell feltenni az adott program header és include fájljait, amelyek a terjesztésekben általában a *csomagnév-dev* csomagban szerepelnek, például: *openldap-dev*. Másik előnye, hogy így szükségtelen fordítóprogramokat telepíteni gépünkre, ami külön hasznos, ha csak egy kicsit is gondolunk a biztonságra. Elvégre minden bit segít! A forráscsomagokból telepített változatok kedvező tulajdonsága, hogy az alkalmazások közül mindig a legfrissebbet használhatjuk, valamint a C fordító segítségével finomhangolni tudjuk a binárist, illetve csak olyan támogatás kerül a binárisba harmadik fél programjához, amelyet szeretnénk. Így a postfix telepítésekor nem kell felraknunk például az *openldap* csomagot, mert nem igényeljük, hogy az LDAP-alapú címtárszolgáltatásokat használja. Megoldás lehet, ha mi magunk készítünk deb vagy rpm csomagot a forrásból, esetleg egy másik gépen fordítjuk le a forrást, és a célgépre a binárisokat csupán átmásoljuk. Mi azonban most a legegyszerűbb megoldásokhoz folyamodunk: először megnézzük a csomagkezelővel történő telepítést a Debian- és RedHat-rendszeren.

Debian (vagy deb-alapú):

```
# apt-get install postfix
vagy
```

```
# dpkg -i postfix_0.0.19991231p111-1.deb
RedHat (vagy rpm-alapú):
```

```
# rpm -Uvh postfix-20010202-4.i386.rpm
```

A Debian Potatóban láthatóan még a régebbi változat szerepel, és a RedHat sem tud lépést tartani *Wietse Venemával*. A postfix legújabb változata a *snapshot-20010714*. A forrásból ezt fogjuk a lehető legegyszerűbb módon telepíteni. Először is látogassunk el a postfix honlapjára, a <http://www.postfix.org> címre, keressük meg a *Downloads* pontot és válasszunk kiszolgáltatót. Két lehetőség áll előttünk: vagy az utolsó hivatalos – jelenleg a Postfix Release 20010228 névre hallgató –, vagy a pillanatfelvétel-változatot (snapshot) – a cikkírás időpontjában Postfix Snapshot 20010714 – töltsük le. Ha ezt megtettük, csomagoljuk ki a forrást a következő módon (feltételezzük, hogy a fájl neve *snapshot-20010714.tar.gz*):

```
# tar xvfz snapshot-20010714.tar.gz -C
/usr/local/src
```

Ez az */usr/local/src* könyvtár alatt egy *snapshot-20010714* nevű könyvtárat hoz létre. Amikor ez is megtörtént, akkor lépünk be a könyvtárba és olvassuk el a számunkra érdekes README fájlokat. Mivel jelenleg az alaprendszert telepítjük, ezt a lépést kihagyhatjuk. A következő lépés a forrás lefordítása bináris állománnyá. A műveletet a szokásostól eltérően nem a *configure* paranccsal kezdjük, hanem egyenesen a *make* parancshoz fordulunk – mindenfajta kapcsoló és beállítás nélkül:

```
# make
```

Most megvárjuk, amíg az összes forrásfájl lefordítódik, majd a postfix futtatásához szükséges felhasználókat adjuk hozzá a rendszerhez:

```
# useradd -d /var/spool/postfix -s /bin/false
postfix
```

Ezzel hozzáadtuk a rendszerhez a postfix nevű felhasználót, aki nem tud bejelentkezni, saját könyvtárának pedig a program által használt könyvtárat adtuk meg, bár megadhattunk volna nem létező könyvtárat és héjat is (a */bin/false* is csak akkor létezik, ha szerepel a */etc/shells* fájlban). Mivel nem mindenki által írható (world-writeable) sort fogunk használni a nem SMTP-kapcsolaton keresztül érkező levelek fogadására, egy *maildrop* csoportot is létre kell hoznunk:

```
# groupadd maildrop
```

Ezután kiadjuk a

```
# make install
```

parancsot, ez elkezd a telepítést. Ezzel egyenértékű a következő parancs:

```
# sh INSTALL.sh
```

ugyanis a *make install* is ezt a parancsállományt hívja meg.

Ekkor kérdésekkel kerülünk szembe, ezek nagy részére alkalmazhatjuk a felkínált lehetőségeket. Nézzük sorban:

install_root: [/] – Arra kíváncsi, hogy mi lesz a gyökérkönyvtár.

A Unix-alapú rendszereken ez a / (perjel), azaz a gyökérkönyvtár.

tempdir: [/usr/local/src/snapshot-20010714] – Ez adja meg, hogy a telepítés közben szükséges átmeneti tárolóhely hol legyen.

config_directory: [/etc/postfix] – A beállítófájlok helye.

daemon_directory: [/usr/lib/postfix] – A Postfixet alkotó demónok helye.

command_directory: [/usr/sbin] – Hol lesznek azok a parancsok, amelyek a Postfixet vezérlik.

queue_directory: [/var/spool/postfix] – A Postfix sorai, ahol a levelek feldolgozás közben tárolódnak.

sendmail_path: [/usr/lib/sendmail] – A Postfix Sendmail programja.

newaliases_path: [/usr/bin/newaliases] – A Postfix Newaliases

programja (az utóbbi kettő használatáról még később szólunk).

mailq_path: [/usr/bin/mailq] – A mailq program helye a rendszerben.

mail_owner: [postfix] – A Postfix sorainak (queue) tulajdonosa.

Ha az előbb nem Postfix-felhasználót adtuk hozzá, akkor annak a nevét kell ide beírni, akit héj és saját könyvtár nélkül létrehoztunk.

setgid: [no] – Mindenki által írható (world-writeable) könyvtár, amelybe a helyileg – a Postfix Sendmail binárisán keresztül – küldött levelek kerülnek vagy pedig a könyvtár által megadott csoporthoz tartozik. Ha az adott csoport tudja csak írni a könyvtárat, akkor úgynevezett csoportjoggal (setgid) rendelkező program alkalmazására nyílik lehetőségünk. Ehhez a Sendmail program a Postdrop programot használja a levelek beillesztésére. Ajánlott a szigorított postázás, ezért adtuk hozzá a rendszerhez a telepítés elején a *maildrop*-csoportot. Tehát itt adjuk meg a *maildrop*-csoportot, a képernyőn pedig a következőket kell látnunk:

```
setgid:[no] maildrop
manpages: [/usr/local/man]
```

Ha mindezekkel végeztünk, a telepítés befejeződik és a

2. lista Az /etc/shells tartalma

```
# /etc/shells: valid login shells
/bin/bash
/bin/csh
/bin/sh
/usr/bin/ksh
/usr/bin/tcsh
/bin/sash
/bin/zsh
/bin/false
```

Az /etc/adduser.conf vonatkozó része

```
# /etc/adduser.conf: 'adduser' configuration.
# See adduser(8) and adduser.conf(5) for full
# documentation.
# The DSHELL variable specifies the default
# login shell on your
# system.
DSHELL=/bin/false
```

```
# postfix start
```

parancsokkal tudjuk indítani az alkalmazást. Ne ijedjünk meg! Amikor legelőször indítjuk el a rendszert, a postfix az addig nem létező könyvtárakat a neki fenntartott helyen hozza létre – ez alapértelmezésben a /var/spool/postfix –, ezek alkotják majd a sorokat. Ezt a folyamatot mutatja be az 1. lista (a 15. CD Magazin/Postfix könyvtárában található).

A postfix minden indulásnál ellenőrzi, hogy léteznek-e a számára szükséges könyvtárak. Amennyiben nem, akkor létrehozza őket. Ha elindult a rendszer és egyetlen naplófájlban sem jelent meg hiba – főleg a /var/log/messages és /var/log/mail.log fájlakat nézzük át –, akkor állítsuk le a rendszert addig, amíg be nem állítjuk. Nézzük meg, hogy mi lesz a célunk:

- A felhasználókat felvesszük a rendszerre, de héj nélkül.
- Beállítjuk a kiszolgáló tulajdonságait: a nevét, illetve hogy melyik névtartománynak (domain) nyújt szolgáltatást.

Az első feladatot könnyű végrehajtani, Debian-rendszeren meg kell keresni a /etc/shells fájlt és egy sorba írjuk bele a következőt: /bin/false – ezzel felvettük az érvényes héjak közé. Elértük, hogy bár a felhasználó szerepel a rendszeren, tehát levelet is tud fogadni, nem tud a rendszerre belépni. A második lépés, hogy átszerkesztjük a /etc/adduser.conf fájl tartalmát. Keressük meg a DSHELL=/bin/bash sort és cseréljük ki azt az alábbira: DSHELL=/bin/false

Ha ezzel végeztünk, elkezdhetjük beállítani a rendszert. A démonok viselkedését a master.cf fájl tartalma határozza meg, mely a sorozatunk előző részében ismertetett master demont irányítja. A kiszolgáló tulajdonságait alapértelmezésben a /etc/postfix/main.cf-ben találjuk meg, most ezzel foglalkozunk.

Első dolgunk, hogy beállítsuk, hol is található meg a levéltároló sorok, a programok és a démonok, valamint a jogosultságok. Ha szabályszerűen telepítettünk, akkor ezekhez a sorokhoz nem is kell nyúlnunk:

```
# a sorok helye
queue_directory = /var/spool/postfix
# a vezérlő programok helye
command_directory = /usr/sbin
# a postfix démonjainak helye
daemon_directory = /usr/lib/postfix
```

ki a sorok tulajdonosa, azaz ki kezeli a sorokat
mail_owner = postfix

A következő sornál érdemes elidőzni. Lehet, hogy felhasználóink egyéni fájlokat hozhatnak létre saját könyvtárukban. Ha ebben egy .forward fájlt helyeznek el, akkor a bejövő levelet az itt megadott címre tudják továbbítani vagy egy programnak továbbadni a rendszeren. Minket ez utóbbi érdekel. A következő értékénél beállított felhasználó jogosultságaival fog végrehajtódni a .forward fájlban keresztül meghívott program. Ezt biztonsági megfontolásból állítsuk minél alacsonyabb szintre. A nobody felhasználó megfelelő erre a szerepre, elvégre nem szeretnénk, hogy egy olyan parancsfájlnak adja át a felhasználó a levelet, ami például a levél tartalmát hajtja végre – abban pedig az rm -rf / szerepel és mindezt a rendszergazda jogaival teszi –, igaz? A postfix-et vagy más kiemelt jogosultságú felhasználót, illetve felhasználónevet ne állítsunk be!

```
default_privs = nobody
```

Ekkor minden héjprogram a nobody jogosultságaival fog futni, ami a .forward fájlból hívódik meg.

A következő sor adja meg a gép teljes nevét, például:

```
myhostname= mailserver.linuxvilag.hu
```

A mydomain érték pedig a tartományt adja meg:

```
mydomain= linuxvilag.hu
```

Ha helyileg (rendszeren belül) keletkezik egy levél, programból vagy felhasználó által, akkor a myorigin értékét veszi alapul. Itt már látható, hogy a Postfix az ismert értékeket más paramétereknél változóként képes kezelni. Itt a myorigin értéke a linuxvilag.hu lesz: myorigin = \$mydomain

Az inet_interfaces = all egyelőre maradhat a helyén, ugyanis ez határozza meg, hogy mely IP-címekre fogadunk el levelet. Az all mindegyiken elfogadja az SMTP-kapcsolatot. A következő számban már néhány trükköt is elsajátíthatunk.

A következő érték adja meg, hogy mely gépnevekhez tartozó leveleket dolgozzon fel a rendszer. Ne soroljuk fel a virtuális tartományokat – azok majd egy későbbiekben tárgyalandó érték feladatait képezik, amelynek használatával a következő számban ismerkedünk meg.

```
mydestination = $myhostname,
                 localhost.$mydomain
```

A tökéletes beállítás az, ha gépünknek csak ez az egy neve van.

```
alias_maps = hash:/etc/aliases
```

```
alias_database = hash:/etc/aliases
```

A fenti két adattábla közül az elsőt csak a Postfix, a másodikat más programok is alkalmazhatják. Itt éppen közös adatbázist használnak. Ha az alias fájlban bármit változtatunk, akkor ki kell adni a newaliases parancsot.

Az utolsó fontos érték, ami ahhoz szükséges, hogy egyszerű, de mégis biztonságos és gyors levelezőkiszolgálót kapjunk, a mynetworks. A mynetworks értékbe kerülnek azok a hálózatok, ahonnan a küldés engedélyezett.

```
mynetworks = 127.0.0.0/8 192.168.1.0/24
```

Ez abban az esetben igaz, ha a helyi hálózat – amely előtt a kiszolgáló található – a 192.168.0.0-s címet használja. Fontos megjegyezni, hogy a 127.0.0.0/8 címtartományt, tehát a saját gépet se hagyjuk ki! A változtatások megtétele és a Postfix elindítása után a rendszer máris használható. A következő részben egy kisebb, nem állandó kapcsolattal rendelkező iroda szükségleteinek kielégítésével és virtuális tartományok kezelésével ismerkedhetünk meg.



Deim Ágoston (ago@lsc.hu)

Kedveli a sört, szereti a futást és imádja Szabó Lőrinc verseit. Nem hisz vakon egyik rendszerben sem. Vonzódik a BSD-hez is. Tagja az LME-nek és a MBE-nek. Mottója: a gép nem lehet fontosabb az embernél.