

Linux 1-2. Labor – Előzetes ismeretek

Bevezetés

Az Operációs Rendszerek tárgy Linux laborjainak célkitűzése, hogy gyakorlati tapasztalatokat nyújtsanak egy Linux operációs rendszer telepítéséről, néhány tipikus szolgáltatás beüzemeléséről valamint egyszerű adminisztrátori feladatok végrehajtásáról. Jelen dokumentum a laborok elvégzéséhez szükséges háttérismereteket foglalja össze, egyben a jobb szélre igazítva megadja a lehetséges beugró kérdéseket is.

Az operációs rendszer telepítése

A laboron a VMWare Player programot fogjuk használni egy virtuális gép létrehozására, mely ingyenesen letölthető alkalmazás a VMWare honlapjáról¹. A Player alkalmazáson belül egy futó virtuális gépbe kattintva az magához vonja a billentyűzet és az egér kezelését: amennyiben a host operációs rendszerbe szeretnénk visszatérni, akkor alapértelmezetten a **Ctrl+Alt** billentyűkombinációval szabadíthatjuk fel a beviteli eszközöket, hogy azok újra a host számára is elérhetőek legyenek.

Hogyan lehet a billentyűzetet és az egeret a virtuális gépből a host gépnek visszaadni?

Az operációs rendszert a laboron hagyományos módon egy ISO lemezkép használatával fogjuk telepíteni. A telepítési útmutatóban néhány alapvető döntésnél csak a helyes választás szerepel, azonban ezek háttere időnként összetettebb magyarázatokat igényel, melyekre a következő alfejezetekben térünk ki.

Nyelv választás

A legtöbb korszerű operációs rendszer ma már tud magyar nyelven, azonban a labor során mégis az eredeti angol nyelvű változatot telepítjük majd, aminek oka kettős: egyrészt a magyar fordítások általában nem 100%-osak, vagyis egy magyar nyelvű felületen is rendre előbukkannak angol üzenetek és szövegek, ugyanis a fordítók állandó versenyfutásban vannak a fejlesztőkkel. Másrészt a fontosabb indok az, hogy az angol nyelven megjelenő hibaüzenetekre és egyéb visszajelzésekre azonnal rá tudunk keresni az interneten, és jó eséllyel megoldást is gyorsan találunk a problémánkra. Ha ugyanezen hibaüzenet magyar változatát először angolra kell fordítani, akkor az esetleges fordítási pontatlanságokon jelentős időt veszíthetünk az angol beállítás használatához képest. E megfontolások alapján a laboron az operációs rendszert angol nyelven telepítjük és használjuk majd.

Milyen előnyei vannak az operációs rendszer angol nyelven történő telepítésének?

A billentyűzet nyelvének kiválasztása már kevésbé egyértelműen eldönthető kérdés: alapvetően igaz az, hogy a magyar billentyűzeten az ékezetes betűk jóval kényelmetlenebb helyre szorították a speciális karaktereket, mint ahol azok az eredeti angol kiosztásban szerepeltek – így általánosságban

¹ <https://my.vmware.com/web/vmware/downloads>

elmondható, hogy a legtöbb rendszeradminisztrációs művelethez kényelmesebb az angol kiosztás használata. Viszont azt is el kell ismerni, hogy az angol kiosztás megszokása – különösen egy magyarul feliratozott klaviatúrán – eleinte kényelmetlen, és némi betanulási időt vesz igénybe, így a laboron a billentyűzetkiosztás választása egyéni preferencia alapján lehetséges.

Miért kényelmesebb az angol billentyűzetkiosztás használata gyakorlott felhasználók számára a konzolban?

Automatikus frissítések

Az asztali operációs rendszereknél már régen megszokhattuk, hogy a frissítéseket a rendszer automatikusan telepíti mindaddig, amíg ezt esetleg explicite ki nem kapcsoljuk. Az Ubuntu szerver telepítése során meglepő lehet, hogy a telepítő alapértelmezetten az automatikus frissítések kikapcsolását ajánlja fel, és a frissítéseket nekünk kell engedélyezni. A jelenség háttere a következő: a szerver operációs rendszereket alapvetően nagy megbízhatóságra tervezik, vagyis fontos, hogy akár adminisztrátori beavatkozás nélkül is megbízhatóan évekig működjön egy ilyen rendszer. E logika mentén az egyes frissítések telepítése az adminisztrátor feladata, ami azzal az előnnyel jár, hogy a frissítések bármilyen mellékhatása (például növekvő lemezterület használat vagy esetleges ritka kompatibilitási problémák) esetén azonnal helyben van a helyzetet kezelni képes kompetens adminisztrátor. Megbízhatósági szempontból tehát a frissítések telepítésére tekinthetünk egyfajta kockázatként, ami egy már megbízhatóan működő rendszer életébe avatkozik be, így indokolt, hogy minden frissítésről az adminisztrátor döntsön.

A legtöbb szerver esetében azonban igaz, hogy nem áll készenlétben folyamatos adminisztrátori szolgálat, és ezekben az esetekben a megbízhatósági kockázatnál jóval nagyobb veszélyt jelent az esetleges támadható felületek nyújtotta biztonsági kockázat, ezért többnyire indokolt a frissítések automatikus telepítése. Mivel ez utóbbi a gyakrabban előforduló eset, ezért a telepítés során explicite be fogjuk kapcsolni az automatikus frissítéseket.

Milyen megfontolások miatt ajánlja fel a rendszer alapértelmezetten kikapcsolva az automatikus frissítéseket?

A bootloader telepítése

A labor során a GRUB boot loader telepítésekor komolyabb körültekintés nélkül jóváhagyhatjuk a boot loader beírását a lemez Master Boot Record részére. Ezt jelen esetben azért tehetjük meg, mert a frissen telepített lesz a számítógép egyetlen operációs rendszere. Ha olyan számítógépre telepítenénk ahol van másik (például egy már létező Windows) telepítés, akkor jóval körültekintőbben kellene eljárunk annak érdekében, hogy továbbra is bootolható maradjon a többi operációs rendszer. Általában a GRUB felismeri ezeket, és megfelelően állítja be őket, azonban ilyen esetben fontos a körültekintés.

Az operációs rendszer használata

A konzolos felület

A labor során végig konzolból dolgozunk majd, legyen az magának a létrehozott Ubuntu szervernek a karakteres konzolja, vagy egy SSH kapcsolat. Napjainkban minden asztali számítógépen illetve mobil eszközön grafikus felülettel találkozunk, azonban a Unix szerverek üzemeltetése során továbbra is nagyon kedvelt a karakteres felület, mely a tipikus rendszergazdai feladatok ellátása során a következő előnyökkel jár:

- A konzolos felület minimális erőforrás igényű, vagyis a szerveren nem köt le a (csak ritkán használt) grafikus felület felesleges erőforrásokat, ráadásul emiatt a terminál felület magas terhelés mellett is jól használható tud maradni.
- A konzolos felület távolról alacsony sávszélességű kapcsolat mellett is kényelmesen használható, még akkor is, amikor egy grafikus felület elérése már zavaróan akadogna.
- A parancsértelmező nagyon sok funkcióval és segédprogrammal támogatja a hatékony munkát: könnyen szerkeszthetők konfigurációs fájlok, egyszerűen megfigyelhetők és szűrhetők napló fájlok, egymás után kapcsolható az egyes programok kimenete, így egyetlen sorban komplex műveletek is elvégezhetőek bármilyen extra program telepítése nélkül, és még hosszan lehetne sorolni.
- Minden a konzolon kiadott parancs használható shell script programban is, vagyis mindaz, ami kézzel a konzolon végrehajtható azonnal futtatható program formájában le is menthető a lemezre és később bármikor előhívható. Ráadásul a shell maga is támogat programozási műveleteket (elágazások, ciklusok, stb.), így számos ránézésre komplex program is shell script formájában került elkészítésre.

Természetesen vannak feladatok, amikre a konzolos hozzáférés nem alkalmas: nem lehet grafikus szerkesztési feladatokat elvégezni, hatékonyan szöveget és multimédiás tartalmat szerkeszteni, és még egy sor más feladatot elvégezni, amire valóban alkalmasabb egy grafikus felhasználói felület – szerencsére azonban ilyen feladatokkal ritkán találkozni szerver környezetben.

Miért van létjogosultsága a konzolos felületnek
Unix szerverek adminisztrálása során?

Hasznos segítség a konzol használata során, hogy az egyes parancsok vagy fájl elérési utak begépelése közben a **TAB** billentyűvel a shell kiegészíti az adott parancsot, amennyiben a kiegészítés egyértelmű. A korábban kiadott parancsokat a terminálon a fel/le billentyűk hívják elő, míg a kiadott parancsok listájában keresni a **Ctrl+R** kombinációval lehet. E funkciók használatával jelentősen gyorsítható az egyes parancsok bevitele a terminálon keresztül.

Felhasználók és jogkörök

A rendszer telepítésekor létrehozuk az első felhasználót, azonban Ubuntu esetében ez egy korlátozott jogosultságú felhasználó. Míg egyes más Unix variánsok esetén engedélyezett a legmagasabb jogokkal rendelkező root felhasználó nevében belépni a rendszerbe, addig az Ubuntu filozófiája szerint minden felhasználónak korlátozott jogkörrel kell belépnie, hogy csak a szükséges műveletekhez kérjen root jogosultságot. E korlátozás háttere az, hogy a terminálon root

jogosultsággal dolgozva viszonylag könnyen lehet komoly kárt tenni a rendszerben nagyon kis hibát elkövetve is. Példának vegyük az alábbi két parancsot:

- **rm -rf /tmp/folder**
- **rm -rf / tmp/folder**

A két parancs mindössze egyetlen alig látható szóköz karakterben tér el egymástól, viszont amíg az első az átmeneti fájlokat tartalmazó **tmp** könyvtár **folder** alkönyvtárát törli, addig utóbbi a teljes gyökérkönyvtárát megsemmisíti, illetve ezt követően kísérletet tesz az aktuális könyvtárban a **tmp/folder** alkönyvtár törlésére. Utóbbi parancsot korlátozott felhasználó szerepkörben kiadva egy egyszerű hozzáférés megtagadva hibával megússzuk, míg root jogosultságokkal máris telepíthetjük újra a rendszert.

Milyen megfontolásból nem támogatja a rendszer, hogy root jogú felhasználóként dolgozzunk benne?

Az előbbieken túl további szempont, hogy mivel a legtöbb Unix rendszeren a root felhasználó root néven található meg, ezért az SSH szerverek folyamatos automatizált támadás alatt állnak, mely támadások célja a root felhasználó nevében valamilyen könnyen törhető jelszóval bejutni a rendszerbe. E jelenségre adott válaszként az SSH szerverek többsége még azokon a rendszereken is megtiltja a root felhasználóval történő belépést, ahol ez a felhasználó egyébként szabadon használható a rendszer adminisztrálására.

Miért tiltják az SSH szerverek a root felhasználó belépését?

Az Ubuntu rendszerbe tehát csak a telepítés során létrehozott felhasználónkkal tudunk belépni, akinek a fenti megfontolások alapján korlátozott a jogköre. Minden olyan esetben, amikor root jogokra van szükségünk, akkor ezt explicite kérnünk kell: erre szolgál a **sudo** parancs. A **sudo** első futtatásakor újra bekéri a felhasználónk jelszavát (amivel megbizonyosodik arról, hogy valóban mi tartózkodunk a terminálnál), majd az utána azonos soron megadott parancsot immáron root jogosultsággal futtatja. A **sudo** használatához az adott felhasználónak a megfelelő felhasználói csoport tagjának kell lennie, vagyis a **sudo** használata csak az ilyen módon lekorlátozott felhasználói kör számára elérhető.

Milyen paranccsal hajthat végre korlátozott jogú felhasználó root jogot igénylő feladatokat?

Fájl jogosultságok

A Unix rendszerek hagyományosan egy egyszerű jogosultság rendszert használnak, melyek három csoportra osztják a felhasználókat. Adott a fájl tulajdonos felhasználója (egy felhasználó), adott a fájl tulajdonos csoportja (egy tetszőleges felhasználói csoport – nem kell megegyeznie a tulajdonos felhasználó csoportjával), illetve mindenki más az egyéb csoportba tartozik. Minden egyes csoportra megadható három jogosultság: az írási jog, az olvasási jog, illetve a végrehajthatósági jog (programok esetén a futtathatóság, könyvtárak esetén a könyvtárba belépést jelenti). Az egyes fájlokról ezeket az információkat legegyszerűbben az **ls** parancs szolgáltatja:

```
[eredics@morgo log]$ ls -l
total 13588
drwx----- 2 root root 4096 Nov 1 03:50 aide
-rw----- 1 root root 2564 Jul 10 2012 anaconda.ifcfg.log
-rw----- 1 root root 19705 Jul 10 2012 anaconda.log
-rw----- 1 root root 38359 Jul 10 2012 anaconda.program.log
-rw----- 1 root root 96749 Jul 10 2012 anaconda.storage.log
-rw----- 1 root root 131134 Jul 10 2012 anaconda.syslog
-rw----- 1 root root 26752 Jul 10 2012 anaconda.xlog
-rw----- 1 root root 4949 Jul 10 2012 anaconda.yum.log
drwxr-xr-x 2 root root 45056 Mar 8 00:00 atop
drwxr-x--- 2 root root 4096 Mar 4 09:47 audit
drwxr-x--- 2 bacula bacula 4096 Feb 19 2015 bacula
-rw-r--r-- 1 root root 1994 Feb 1 16:28 boot.log
-rw----- 1 root utmp 218496 Mar 8 00:49 btmp
-rw----- 1 root utmp 24696 Mar 1 03:09 btmp-20160301.gz
drwxr-xr-x 2 clam clam 4096 Mar 1 03:09 clamav
-rw----- 1 root root 227823 Mar 8 08:15 cron
```

A fenti példában a log könyvtárban adtuk ki az `ls -l` parancsot. Látszik, hogy a legtöbb fájlnak a root felhasználó (3. oszlop) és az azonos nevű root csoport (4. oszlop) a tulajdonosa, azonban más felhasználók és csoportok is megjelennek. A legtöbb fájlra csak a tulajdonos felhasználónak van írási és olvasási joga, míg az **atop** könyvtárra a tulajdonos csoportnak és mindenki másnak is olvasási és listázási joga van. A **bacula** könyvtárra a tulajdonosnak teljes joga, a tulajdonos csoportnak olvasási és listázási joga van, míg a többi felhasználó nem fér ehhez hozzá.

Milyen három csoportban szabályozható a fájlok felhasználói hozzáférése?

A jogosultságok beállításakor általában egy háromjegyű számot használunk, például úgy, hogy: **chmod 644 boot.log**. Az itt látható három egymást követő számjegy a három felhasználói csoportnak felel meg: az első a tulajdonos jogait kódolja, a továbbiak pedig a tulajdonos csoport és a többiek jogait állítják be. Az egyes számjegyek 3 biten reprezentálják az adott csoport hozzáférési jogát:

- a legmagasabb helyiérték (4-es érték) az olvasási jog,
- a következő (2-es érték) az írási jog,
- végül az utolsó (1-es érték) a végrehajtási jog.

Ha tehát egy csoportnak csak olvasási jogot szeretnénk adni, akkor az adott számjegy 4 lesz, ha írási és olvasási jogot is, akkor $4+2 = 6$, ha minden jogot, akkor $4+2+1 = 7$ lesz. Ez alapján látható, hogy az előző **chmod 644 boot.log** parancs éppen a fenti képen látható jogosultságokat állítja be a boot.log fájlra.

Hogyan számíthatók ki a fájlok jogosultság beállításához használható három számjegyű kód számjegyei?

A csomagkezelő használata

Unix rendszereken az alkalmazások telepítése az esetek döntő többségében napjainkban már csomagkezelő segítségével történik. Nyilván továbbra is adott a lehetőség az egyes programokat forráskód formájában minden függőségükkel együtt letölteni, majd az adott rendszeren lefordítani, azonban általában jóval kényelmesebb és elégséges a már lefordított alkalmazást csomag formájában a csomagkezelő használatával letölteni és telepíteni.

A Debian alapú rendszerek, mint amilyen az Ubuntu is, az APT (Advanced Package Tool) csomagkezelőt használják. Az ezzel kapcsolatos három legfontosabb parancs a következő:

- **apt update:** Ez a parancs a csomagkezelő helyi adatbázisát frissíti a központi szerverekről. Lefuttatásával minden elérhető csomagról a legfrissebb információk állnak majd rendelkezésre helyben, vagyis a helyi rendszer e parancs futtatásával értesülhet az elérhető új csomagokról vagy frissítésekről.
- **apt upgrade:** Ez a parancs telepíti az elérhető frissítéseket. Általában az apt update után futtatva a telepített csomagok legfrissebb verziójának telepítésére használatos.
- **apt install:** Ezzel a paranccsal telepíthető egy új csomag a rendszerre. A csomagkezelő a telepítés során figyel arra, hogy minden szükséges szoftver követelmény vagy már elérhető legyen a rendszerben, vagy a telepítési folyamat során szintén telepítésre kerüljön.

Mivel a rendszerre szoftverek telepítése alapvetően adminisztrátori feladat, azért az **apt** parancsok használatához root jogosultság kell, vagyis mindig a **sudo** paranccsal használjuk őket.

Milyen paranccsal telepíthető szoftver a csomagkezelő segítségével a rendszerre?

Mi a különbség az apt update és az apt upgrade között?

Alapvető UNIX ismeretek

Az alábbiakban néhány alapvető ismeretet foglaltunk össze, melyekre a labor elvégzéséhez szükség lesz. A beugróban rákérdezhetünk ezeknek a parancsoknak és könyvtáraknak a pár szavas szerepére.

Fontosabb parancsok

Alapvető fontosságú parancsok:

- **man** (manual): megjeleníti a paraméterként megadott parancs használati utasítását
- **cd** (change directory): könyvtár váltása
- **ls** (list): könyvtár tartalmának megjelenítése
- **cp** (copy): fájl vagy könyvtár másolása
- **mv** (move): fájl vagy könyvtár mozgatása
- **rm** (remove): fájl vagy könyvtár törlése
- **less:** szöveges fájl tartalmának megjelenítése
- **nano:** szöveges fájl szerkesztése
- **sudo:** az őt azonos soron követő parancs root joggal futtatása

Mire szolgál a *** parancs?

Fontosabb könyvtárak

Néhány fontosabb könyvtár:

- **/home:** a felhasználók saját könyvtára, ahol a személyes fájljaik, levelezésük, beállításaik tárolódnak
- **/etc:** a rendszer konfigurációs állományait tartalmazó könyvtár
- **/var/log:** a rendszernaplók tárolására szolgáló könyvtár

Milyen célra szolgál a *** könyvtár?