Linux és open source

# Mi a linux

Kezdve az okostelefonoktól az autókban, szuper számítógépeken és háztartási készülékeken át otthoni számítógépekig és vállalati szerverekig, a Linux operációs rendszer mindenhol megtalálható.

A Linux már az 1990-es évek közepétől jelen van, és azóta világméretű felhasználói bázist ért el. Valójában a Linux mindenütt ott van: a telefonban, a termosztátban, az autóban, a hűtőben, és a televízióban. Emellett működteti az internet nagy részét, a világ legjobb 500 szuperszámítógépét és a világ tőzsdéit.

De azon kívül, hogy a világ minden táján elterjedt platform a számítógépek, szerverek és beágyazott rendszerek futtatásához, a Linux az egyik legmegbízhatóbb, legbiztonságosabb és legproblémamentesebb operációs rendszer is.

## Mi a linux?

Ahogyan a Windows, az iOS és a Mac OS, a Linux is egy operációs rendszer. Az egyik legnépszerűbb platform a bolygón, az Android is a Linux operációs rendszeren működik. Az operációs rendszer olyan szoftver, amely kezeli az összes hardver erőforrást a számítógépen vagy laptopon. Egyszerűen fogalmazva az operációs rendszer kezeli a kommunikációt a szoftverek és a hardverek között. Az operációs rendszer (OS) nélkül a szoftver nem működne.

A Linux operációs rendszer több elkülönülő részből áll:

* Bootloader - Ez a szoftver kezeli a számítógépének indítási folyamatát. A legtöbb felhasználó számára ez egyszerűen csak egy felugró ablak, amely végül eltűnik, és elindítja az operációs rendszert.
* Kernel - Ez az az elem, amelyet valójában 'Linux'-nak nevezünk. A kernel a rendszer magja, és kezeli a CPU-t, a memóriát és a perifériás eszközöket. A kernel az operációs rendszer legalacsonyabb szintje.
* Init rendszer - Ez egy alrendszer, amely inicializálja a felhasználói teret, és felelős a daemon-ok (háttérszolgáltatások - például nyomtatás, hang, ütemezés stb.) irányításáért. Az egyik legelterjedtebb init rendszer a systemd, amely egyben az egyik legvitatottabb is. Az init rendszer kezeli az indítási folyamatot, miután az elsődleges bootolást megcsinálta a bootloader (pl. GRUB vagy GRand Unified Bootloader).
* Daemon-ok - Ezek a háttérszolgáltatások (nyomtatás, hang, ütemezés stb.), amelyek vagy az indítás során indulnak el, vagy miután bejelentkeztünk az asztali környezetbe.
* Grafikus szerver - Ez az alrendszer jeleníti meg a grafikát a monitoron. Gyakran hivatkoznak rá X szerverként vagy egyszerűen csak X-ként.
* Asztali környezet - Ez az a rész, amellyel a felhasználók ténylegesen interakcióba lépnek. Sok asztali környezet közül választhatunk (GNOME, Cinnamon, Mate, Pantheon, Enlightenment, KDE, Xfce, stb.). Minden asztali környezet tartalmaz beépített alkalmazásokat (például fájlkezelőket, konfigurációs eszközöket, webböngészőket és játékokat).
* Alkalmazások - Az asztali környezetek nem tartalmazzák az összes elérhető alkalmazást. Ahogyan a Windows és a macOS, a Linux is több ezer magas minőségű szoftvercímet kínál, amelyek könnyen megtalálhatók és telepíthetők. A legtöbb modern Linux disztribúció (ezekről lentebb lesz szó) tartalmaz App Store-szerű eszközöket, amelyek egyszerűsítik és centralizálják az alkalmazások telepítését. Például az Ubuntu Linux rendelkezik az Ubuntu Szoftverközponttal (régen GNOME Szoftver), amely lehetővé teszi, hogy gyorsan keressünk az alkalmazások ezrei között egyetlen központi helyen.

## Miért használjunk Linuxot?

Ez az a kérdés, amit legtöbben feltesznek. Miért érdemes teljesen más számítógépes környezetet tanulni, amikor az operációs rendszer, amelyet a legtöbb asztali számítógép, laptop és szerver mellékelve kap, tökéletesen működik?

Ahhoz, hogy erre a kérdésre válaszoljunk, tegyünk fel egy másik kérdést. Az a jelenleg használt operációs rendszer valóban "teljesen jól" működik? Vagy néha olyan akadályokkal nézünk szembe, mint a vírusok, kártékony szoftverek, lassulás, összeomlások, költséges javítások és licensz díjak?

Ha küzdesz az említett problémákkal, akkor a Linux lehet az ideális platform számodra. A Linux az egyik legmegbízhatóbb számítógépes ökoszisztéma a bolygón. Ezt a megbízhatóságot párosítsd az ingyenes használattal, és megkapod a tökéletes megoldást egy asztali platformhoz.

Igen, nullás belépési költség... azaz ingyenes. Linuxot telepíthetsz annyi számítógépre, amennyire szeretnél, anélkül, hogy bármit is fizetnél a szoftverekért vagy a szerver licenszért.

Vegyük például egy Linux szerver árát a Windows Server 2016-hoz képest. A Windows Server 2016 Standard kiadásának ára 882,00 USD (közvetlenül a Microsofttól vásárolva). Ez nem tartalmaz ügyfél hozzáférési licenszet (CAL) és más olyan szoftverek licencét, amelyek futtatásához szükséged lehet (például adatbázis, webszerver, levelezőszerver stb.). Például egyetlen felhasználói CAL, a Windows Server 2016-hoz, 38,00 USD-ba kerül. Ha például 10 felhasználót kell hozzáadnod, az további 380,00 dollárt jelent a szerver szoftverek licenszére. A Linux szerverrel mindez ingyenes és könnyen telepíthető. Valójában egy teljes webkiszolgálót telepíteni (ami tartalmaz egy adatbázis szervert) mindössze néhány kattintással vagy paranccsal történik (pl Easy LAMP Server).

Ha a nullás költség nem lenne elegendő, mi a helyzet egy olyan operációs rendszerrel, amely problémamentesen működik, amíg használod? Van aki közel 20 éve használ Linuxot mind asztali, mind szerverplatformként, és nem tapasztal problémát zsarolóvírussal, kártékony szoftverrel vagy egyéb vírusokkal. A Linux általában jóval kevésbé sebezhető az ilyen támadásokkal szemben. A szerver újraindítása csak akkor szükséges, ha a kernel frissítésre kerül. Egy Linux szerver esetében nem szokatlan, hogy évekig működjön újraindítás nélkül. Ha követed a rendszeres ajánlott frissítéseket, a stabilitás és megbízhatóság gyakorlatilag garantált.

## Open source

A Linux is az open source licensz alatt van. Az open source az alábbi alapelveken alapszik:

* A program futtatásának szabadsága bármilyen célra.
* A program működésének tanulmányozásának és megváltoztatásának szabadsága annak érdekében, hogy azt tegye, amit a felhasználó szeretne.
* A másolatok terjesztésének szabadsága annak érdekében, hogy segíteni lehessen a többi felhasználót.
* A módosított változatok mások számára történő terjesztésének szabadsága.

Ezek a pontok kulcsfontosságúak a Linux platform létrehozásában együttműködő közösség megértéséhez. Kétségtelen, hogy a Linux egy olyan operációs rendszer, amely "az emberek által, az emberekért" készül. Ezek az alapelvek is fő tényezők abban, hogy miért választják sokan a Linuxot. A szabadságról, az használat szabadságáról és a választás szabadságáról szól.

## Mi az a disztribúció?

A Linuxnak számos különböző verziója van, hogy megfeleljen bármilyen felhasználótípusnak. Az új felhasználóktól a hardcore felhasználókig találsz egy "flavor"-t a Linuxból, ami illik az igényeidhez. Ezeket a verziókat disztribúcióknak nevezik (rövidítve "distro"). Majdnem minden Linux disztribúció ingyenesen letölthető, kiégethető lemezre (vagy USB kulcsra) és telepíthető (annyi gépre, amennyire csak szeretnéd).

## Népszerű Linux disztribúciók:

* Linux mint
* Manjaro
* Debian
* Ubuntu
* Antergos
* Solus
* Fedora
* Elementary OS
* Openuse

Minden disztribúciónak más a megközelítése az asztali felülettel. Néhányuk nagyon modern felhasználói felületet választ (például a GNOME vagy az Elementary OS Pantheon), míg mások egy hagyományosabb asztali környezetet használnak (az openSUSE a KDE-t használja).

A Distrowatch weboldalon megnézheted a legjobb 100 disztribúciót.

Szerver disztribúció területen a következők közül választhatsz:

* Red Hat Enterprise Linux
* Ubuntu Server
* CentOS
* SUSE Enterprise Linux

A fent említett szerver disztribúciók közül néhány ingyenes (például az Ubuntu Server és a CentOS), és néhány nem (például a Red Hat Enterprise Linux és a SUSE Enterprise Linux). Az árral rendelkező verziókhoz támogatás is tartozik.

Az, hogy melyik disztribúciót választod, az attól függ, hogy:

* Mennyire jártas vagy a számítógép használatában?
* Inkább modern vagy hagyományos asztali felületet szeretnél?
* Szerver vagy asztali számítógép?

Ha számítógépes készségeid viszonylag szerények, érdemes kezdőbarát disztribúciót választanod, mint például a Linux Mint, Ubuntu, Elementary OS vagy Deepin. Ha a készségeid átlagosnál magasabb szintűek, választhatod a Debian vagy Fedora disztribúciót. Ha azonban már gyakorlatilag mestere vagy a számítógép és rendszer adminisztrációnak, akkor a Gentoo disztribúció lehet számodra megfelelő. Ha igazi kihívásra vágysz, akkor pedig létrehozhatsz saját Linux disztribúciót is a Linux From Scratch segítségével.

Ha kizárólag szerver disztribúciót keresel, akkor el kell döntened, hogy szükséged van-e asztali felületre, vagy kizárólag parancssoros interfészen keresztül szeretnél dolgozni. Az Ubuntu Server nem telepít grafikus felületet. Ez két dologgal jár: a szervered nem lassul le a grafikus felület betöltése miatt, és kiterjedt ismeretekre van szükséged a Linux parancssoron keresztüli használatához. Azonban telepíthetsz grafikus felületet az Ubuntu Serverre egyetlen parancs kiadásával, például a "*sudo apt-get install ubuntu-desktop*" paranccsal. Rendszergazdák továbbá fontolóra vehetnek egy disztribúciót azáltal is, hogy figyelembe veszik a funkciókat. Szeretnél olyan szerver-specifikus disztribúciót, amely már az alapoktól mindent tartalmaz, amire szükséged van a szerveredhez? Ha igen, akkor a CentOS lehet a legjobb választás. Vagy inkább egy asztali disztribúciót választanál, és ahogy szükséged van rá, hozzáadod a szükséges elemeket? Ha igen, akkor a Debian vagy az Ubuntu Linux is megfelelő lehet a számodra.

## Telepítés

Sokak számára az operációs rendszer telepítése nagyon ijesztő feladatnak tűnhet. Hihetetlen, de a Linux az összes operációs rendszer közül az egyik legegyszerűbben telepíthető. Valójában a legtöbb Linux verzió kínál olyan úgynevezett "Live disztribúciót", ami azt jelenti, hogy futtathatod az operációs rendszert CD/DVD vagy USB flash meghajtóról úgy, hogy nem hajt végre változtatásokat a merevlemezen. Teljes funkcionalitást kapsz anélkül, hogy kötelező lenne a telepítés. Miután kipróbáltad, és eldöntötted, hogy szeretnéd használni, egyszerűen duplán kattints az "Install" ikonra, és kövesd az egyszerű telepítési varázslót.

Általában a telepítési varázslók követik az alábbi lépéseket (bemutatjuk az Ubuntu Linux telepítését):

1. Előkészítés: Győződj meg róla, hogy a géped megfelel a telepítés követelményeinek. Ezenkívül lehet, hogy megkérdezi, hogy telepíteni szeretnél-e harmadik féltől származó szoftvert (például MP3 lejátszáshoz szükséges plugineket, videokodekeket stb.).
2. Vezeték nélküli beállítás (ha szükséges): Ha laptopot (vagy vezeték nélküli hálózattal rendelkező gépet) használsz, kapcsolódni kell a hálózathoz annak érdekében, hogy letölthess harmadik féltől származó szoftvereket és frissítéseket.
3. Merevlemez kiosztás: Ezen a lépésen kiválaszthatod, hogy hogyan szeretnéd telepíteni az operációs rendszert. Szeretnéd Linuxot telepíteni egy másik operációs rendszer mellé (ezt "dual booting"-nak nevezik), használnád az egész merevlemezt, frissítenéd egy meglévő Linux telepítést vagy telepítenéd egy meglévő Linux verzió helyére.
4. Helyszín: Válaszd ki a helyet.
5. Billentyűzetkiosztás: Válassz billentyűzetkiosztást a rendszeredhez.
6. Felhasználói beállítás: Állítsd be a felhasználónevedet és a jelszavadat.

Ennyi. Miután a rendszer befejezte a telepítést, indítsd újra, és készen állsz. Ha részletesebb útmutatóra van szükséged a Linux telepítéséhez, nézd meg a "How to Install and Try Linux the Absolutely Easiest and Safest Way " című cikket, vagy töltsd le a Linux Foundation PDF útmutatóját a Linux telepítéséhez.

## Szoftverek telepítése Linuxra

Ahogyan maga az operációs rendszer könnyen telepíthető, úgy a programok is könnyedén telepíthetők. A legtöbb modern Linux disztribúció tartalmaz egy olyan alkalmazásboltot, amelyet sokan úgy ismernek, mint app store. Ez egy központosított hely, ahol a szoftvereket keresni és telepíteni lehet. Az Ubuntu Linux (és sok más disztribúció) a GNOME Szoftverre támaszkodik, az Elementary OS az AppCenterre, a Deepin a Deepin Szoftverközponra, az openSUSE pedig az AppStore-ra, míg néhány disztribúció a Synaptic-ra támaszkodik.

A sok név ellenére ezeknek az eszközöknek mindegyike ugyanazt teszi: központi helyek a Linux szoftverek kereséséhez és telepítéséhez.

Nézzünk két különböző eszközt, hogy szemléltessük, milyen könnyű akár a parancssoros telepítés is. A bemutatott példák Debian alapú disztribúciókra és Fedora alapú disztribúciókra vonatkoznak. A Debian alapú disztribúciók a szoftverek telepítéséhez az *apt-get* eszközt használják, míg a Fedora alapú disztribúciókhoz a *yum* eszközre lesz szükség. Mindkettő hasonlóan működik. A bemutatáshoz az *apt-get* parancsot használjuk. Tegyük fel, hogy telepíteni szeretnél a wget nevű eszközt (ez egy hasznos eszköz, amelyet a fájlok letöltésére lehet használni a parancssorból). A wget telepítéséhez az apt-get használatával a következő parancsot kell kiadnod:

*$sudo apt-get install wget*

A sudo parancsot hozzáadod, mert a szoftver telepítéséhez superuser jogokra van szükség. Hasonlóan, ha ugyanezt a szoftvert egy Fedora alapú disztribúcióra szeretnéd telepíteni, először váltasz szuperhasználóvá a "*su*" parancs kiadásával és a root jelszó megadásával, majd kiadod ezt a parancsot:

*$yum install wget*

Ez minden, amire szükség van a szoftver telepítéséhez egy Linux gépen. Nem olyan nagy kihívás. Emlékezz vissza az előzőleg említett Easy Lamp Server telepítésre. Ez egyetlen parancs kiadásával történik:

*$sudo taskel*

Teljes LAMP (Linux Apache MySQL PHP) szervert telepíthetsz szerver vagy asztali disztribúcióra. Tényleg ilyen egyszerű.

# Open source

## Definition

Open Source Software (OSS) olyan szoftver, amely a forráskódjával együtt terjesztik, így elérhetővé válik a felhasználásra, módosításra és terjesztésre az eredeti jogokkal együtt. A forráskód a szoftver része, amelyet a legtöbb számítógép felhasználó soha nem lát; ez az a kód, amelyet a számítógép programozók manipulálhatnak annak érdekében, hogy irányítsák, hogyan viselkedjen egy program vagy alkalmazás. A forráskódhoz hozzáféréssel rendelkező programozók módosíthatják a programot úgy, hogy hozzáadnak, megváltoztatnak vagy kijavítanak olyan részeket, amelyek nem működnek megfelelően. Az OSS általában olyan licencet tartalmaz, amely lehetővé teszi a programozók számára, hogy módosítsák a szoftvert úgy, hogy legjobban megfeleljen igényeiknek és ellenőrizzék a szoftver terjeszthetőségét.

## Történet

Az az elképzelés, hogy a forráskódot ingyenesen elérhetővé tegyék, 1983-ból származik, Richard Stallman, az MIT programozója informálisan alapította meg egy ideológiai mozgalomból. Stallman úgy gondolta, hogy a szoftvereknek elérhetőnek kell lenniük a programozók számára, hogy módosíthassák úgy, ahogy akarják, annak érdekében, hogy megértsék, megismerjék és fejlesszék azt. Stallman saját licencével, a GNU Public License-szel kezdett el ingyenes kódot közzétenni. Ez az új megközelítés és ideológia a szoftverfejlesztő társadalomban elfogadottá vált, és végül az Open Source Initiative (OSI) létrehozásához vezetett 1998-ban.

## Open Source Initiative

Az Open Source Initiative (OSI) az open source szoftverek és közösségek népszerűsítésére és védelmére jött létre. Röviden az OSI központi információs és szervező egységként működik az open source szoftverek számára. Szabályokat és irányelveket biztosít arra vonatkozóan, hogy hogyan lehet használni az OSS-t, együttműködni OSS-sel, valamint kódengedélyezési információkat, támogatást, meghatározásokat és általános közösségi együttműködést nyújt az open source használatának és kezelésének megértéséhez és etikus megközelítéséhez.

## Hogy működik

Az open source kód általában egy nyilvános szerveren (repository) van tárolva és nyilvánosan megosztva. Bárki hozzáférhet a szerverhez, hogy használja a kódot, vagy hozzájáruljon a szoftver projekt tervezésének és funkcionalitásának javításához.

Az OSS általában egy disztribúciós licencet is tartalmaz. Ez a licenc meghatározza, hogy a fejlesztők hogyan használhatják, tanulmányozhatják, módosíthatják és hogy terjeszthetik a szoftvert. Az OSS-en végzett módosítások esetén és az alapján, hogy mi és hogy lett megváltoztatva, a licence megmondja, hogy az eredményként létrejött szoftvereket ingyenesen elérhetővé kell e tenni vagy nem.

A Synopsys Black Duck® KnowledgeBase adatai alapján az öt legnépszerűbb licenc a következők:

* MIT licenc
* GNU General Public License (GPL) 2.0 - ez szigorúbb és előírja, hogy a módosított kódot nyilvánosan elérhetővé kell tenni
* Apache License 2.0
* GNU General Public License (GPL) 3.0
* BSD License 2.0 (3-Clause, New or Revised) - ez kevésbé szigorú

Az OSS-ben fejlesztett szoftverek például a következők lehetnek:

* GNU/Linux
* Mozilla Firefox
* VLC médialejátszó
* SugarCRM
* GIMP
* VNC
* Apache webszerver
* LibreOffice
* jQuery

## Bug free?

Rövid válasz: nem.

Az open source szoftverek több fél módosításait és javításait tartalmazzák, így elkerülhetetlen, hogy minőségi, teljesítménybeli és biztonsági hibák legyenek. Azonban a kódhoz hozzájárulók széles köre azt is jelenti, hogy a hibákat gyorsabban azonosítják és javítják.

Függetlenül attól, hogy milyen típusú szoftverről van szó - nyílt forráskódú vagy kereskedelmi - mindig lesznek kódhibák. A fő különbség az, hogy ki felelős a hibák javításáért; a kereskedelmi szoftverek esetében a szállítók felelősek, míg az open source szoftver esetében a felhasználó felelős.

## Előnyök és hátrányok

### Nyílt forráskódú szoftver előnyei:

* Nyílt forráskódú szoftver ingyenes.
* A nyílt forráskód rugalmas; a fejlesztők megvizsgálhatják, hogyan működik a kód, és szabadon módosíthatják a program vagy alkalmazás hibás vagy problémás részeit, hogy azok jobban megfeleljenek az egyedi igényeiknek.
* A nyílt forráskód stabil; a forráskódot nyilvánosan terjesztik, így a felhasználók hosszú távú projektekhez használhatják, mivel tudják, hogy a kód készítői valószínűleg nem szüntetik meg egyszerűen a projektet vagy nem hagyják, hogy lepusztuljon.
* A nyílt forráskód ösztönzi az ötletességet; a programozók előre létező (például más OSS) kódot használhatnak a szoftver javításához és akár saját innovációkat is kitalálhatnak.
* A nyílt forráskód mögött beépített közösség áll, amely folyamatosan módosítja és javítja a forráskódot.
* A nyílt forráskód nagyszerű tanulási lehetőségeket kínál az új programozók számára.

### Nyílt forráskódú szoftver hátrányai:

* A nyílt forráskódú szoftver használata és elfogadása nehezebb lehet az egyszerű beállítások és a barátságos felhasználói felületek hiánya miatt.
* A nyílt forráskódú szoftverek kompatibilitási problémákat okozhatnak. Amikor OSS-sel próbáljuk illeszteni a szabadalmaztatott hardvereket, gyakran szükség van olyan speciális illesztőprogramokra, amelyek általában csak a hardver gyártójától érhetők el.
* A nyílt forráskódú szoftverek felelősségi problémákat okozhatnak. Míg a kereskedelmi szoftvereket teljesen a szállítók irányítják, a nyílt forráskódú szoftverek ritkán tartalmaznak garanciát, felelősséget vagy jogi védelmet. Ez azt jelenti, hogy az OSS fogyasztója felelős a jogi kötelezettségek betartásáért.
* A nyílt forráskódú szoftverek nem várt költségeket okozhatnak a felhasználók képzésében, az adatok importálásában és a szükséges hardver beállításában.

## Security

Bár a nyílt forráskódú szoftverek számos előnyt kínálnak, új szintet hoznak a szoftver kockázatkezelésébe. Kritikus fontosságú, hogy az OSS-t használó szervezetek, vagy az OSS-t tartalmazó kódbázisokkal rendelkező szervezetek valóban megértsék, mi van a kódjukban, hogy hatékonyan kezeljék és biztosítsák azt. Ha többet szeretnél megtudni a nyílt forráskódú kockázatokról és arról, hogyan lehet azokat csökkenteni, itt van néhány lépés, amit megtehetsz:

* Olvasd el az éves "Open Source Security and Risk Analysis" (OSSRA) jelentést, hogy megértsd az aktuális állapotot a nyílt forráskódú sebezhetőségek és kockázatok terén.
* Keress olyan szervezeteket és tanácsadó cégeket, amelyek segíthetnek az open source kockázatok azonosításában és kezelésében a saját szervezetedben.
* Keressetek cikkeket, blogokat és webináriumokat az interneten, amelyek tippeket és legjobb gyakorlatokat (best practice) kínálnak a nyílt forráskóddal járó kockázatok kezeléséhez.
* Vedd fel a kapcsolatot open source kockázatkezelés területén jártas szakemberekkel, hogy személyre szabott tanácsot és iránymutatást kapj a szervezet egyedi igényeihez.
* Fontoljátok meg az automatizált megoldások bevezetését a nyílt forráskódú kezelés és biztonság terén, hogy segítsenek azonosítani és megoldani a sebezhetőségeket, valamint megfeleljenek a nyílt forráskódú licenszekkel kapcsolatos követelményeknek.

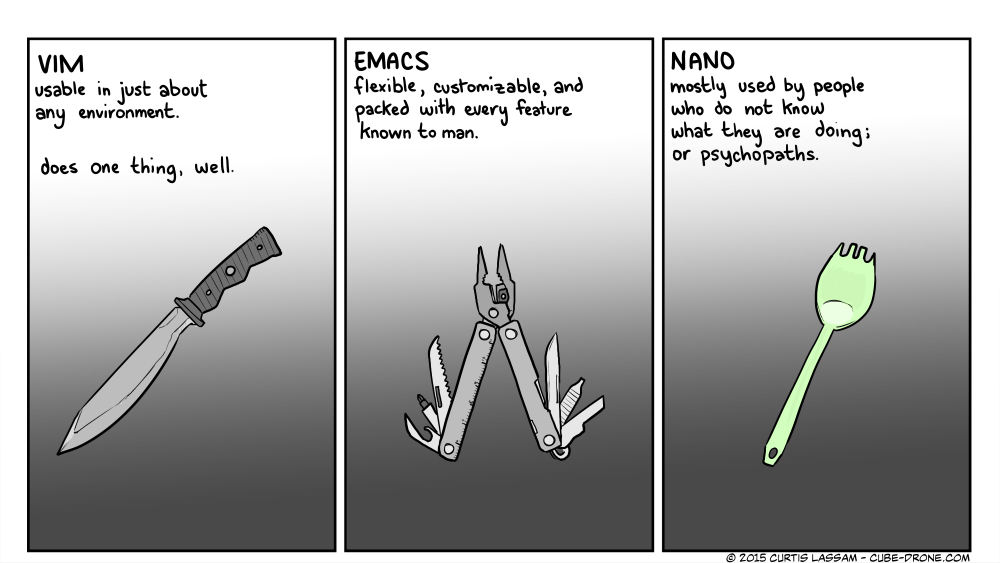
Ezekkel a lépésekkel többet megtudhatsz a nyílt forráskódú kockázatokról, és megteheted a szükséges lépéseket a kockázat csökkentése érdekében, biztosítva a szervezet szoftverének biztonságát és megfelelőségét.

# Linux használat

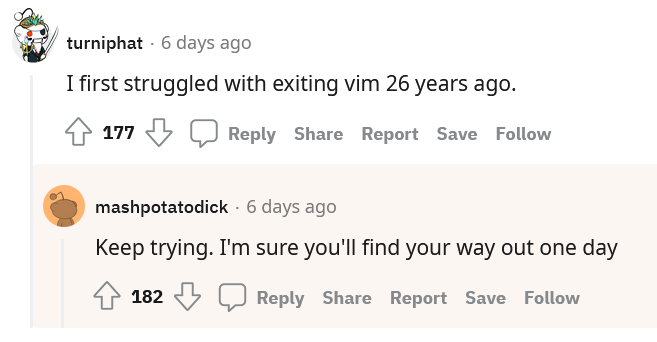
## Alap parancsok

* cd: Váltás a mappák között (Change Directory). A "cd" parancs használatával tudsz navigálni a fájlrendszereken belül és átjutni egyik mappából a másikba.
* ls: Listázza a mappa tartalmát (List). A "ls" parancs használatával megnézheted az aktuális mappa tartalmát, vagy megadott mappában található fájlokat és almappákat.
* rm: Törli a fájlokat vagy mappákat (Remove). A "rm" parancs segítségével törölheted a fájlokat és mappákat a fájlrendszerből.
* cp: Másolja a fájlokat vagy mappákat (Copy). A "cp" parancs használatával fájlokat és mappákat másolhatsz egyik helyről a másikra.
* mv: Áthelyezi a fájlokat vagy mappákat (Move). A "mv" parancs segítségével áthelyezheted a fájlokat és mappákat egyik helyről a másikra.
* pwd: Az aktuális mappa útvonala (Print Working Directory). A "pwd" parancs kiírja az aktuális mappa elérési útvonalát.
* mkdir: Új mappa létrehozása (Make Directory). A "mkdir" parancs segítségével új mappákat hozhatsz létre a fájlrendszerben.

## Szövegszerkesztés











A Vim és a Nano két népszerű szövegszerkesztő a Linux rendszerekben. Mindkettőt használhatod a szövegfájlok szerkesztésére a terminálban vagy a grafikus környezetben.

### Vim

**Fájl megnyitása Vim-ben:**

$vim file.txt

Ez megnyitja a file.txt nevű fájlt a Vim-ben.

**Mód váltása:**

A Vim három alapvető móddal rendelkezik:

* Bemeneti mód (insert mode): A szöveget ebben a módban tudod szerkeszteni.
* Normál mód (normal mode): A navigáció és parancsok adása ebben a módban történik.
* Parancssor mód (command mode): Speciális parancsok adása ebben a módban történik.

A Normál módra való váltáshoz nyomd meg az Esc billentyűt.

**Szöveg szerkesztése:**

* A Normál módban használhatod az nyilakat és hasonlókat, hogy navigálj a szövegben.
* A i billentyű megnyitja a Bemeneti módot, ahol tudsz szöveget írni.
* A x billentyű törlő műveletet hajt végre (a kurzor alatt lévő karaktert törli).

**Fájl mentése és bezárása:**

* A Normál módban használd az :w parancsot a fájl mentéséhez.
* A :q parancs bezárja a fájlt.
* Ha módosítottad a fájlt, de nem mentsz, használd a :q! parancsot a bezáráshoz erőltetve.

### Nano

**Fájl megnyitása Nanoban:**

$nano file.txt

Ez megnyitja a file.txt nevű fájlt a Nanoban.

**Szöveg szerkesztése:**

* Használd az nyilakat a navigáláshoz.
* Írj szöveget a kijelölt helyre.

**Fájl mentése és bezárása:**

* A Ctrl + O billentyűkombinációval tudod menteni a fájlt.
* A Ctrl + X billentyűkombinációval tudod bezárni a szövegszerkesztőt.

Mind a Vim, mind a Nano rendelkezik további parancsokkal és lehetőségekkel, de ezek a legfontosabbak a gyors kezdéshez. Válaszd azt a szerkesztőt, amely a legkényelmesebb számodra, és gyakorolj, hogy megtanuld az alapvető műveleteket!

## File jogosultságok

A fájl jogosultságok a Linux rendszerek biztonsági modelljének alapvető elemei. Meghatározzák, hogy ki és hogyan férhet hozzá a fájlokhoz és könyvtárakhoz a rendszeren.

Az "*ls*" parancs a "*-l*" (hosszú listázás) opcióval megnézhetjük a Linux fájlokra vonatkozó metaadatokat (adatokról szóló adatok), ideértve a fájlengedélyeket is.

$ ls -l

drwxr-xr-x. 4 root root 68 Jun 13 20:25 tuned

-rw-r--r--. 1 root root 4017 Feb 24 2022 vimrc

Ebben a példában két különböző listát látunk. Az "*ls -l*" kimenet első mezője a metaadatok egy csoportja, amely tartalmazza a fájlengedélyeket. Itt vannak a vimrc lista elemei:

* Fájltípus: -
* Engedélybeállítások: rw-r--r--
* Kibővített attribútumok: dot (.)
* Felhasználó tulajdonos: root
* Csoport tulajdonos: root

A fenti példában a vimrc fájl egy normál fájl, amelynek fájltípusa - (azaz nincs különleges típusa).

A tuned lista egy d, vagyis könyvtár típusú fájlhoz tartozik. Vannak más fájltípusok is, de ez a két leggyakoribb. A rendelkezésre álló attribútumok a használt fájlrendszertől függnek. Pl a Red Hat Enterprise Linux 7, 8 és 9 esetén az alapértelmezett fájlrendszer formátuma az XFS.

A rw-r*--r–* karakterlánc valójában három különböző engedélykészlet kifejezése:

* rw-
* r--
* r--

Az első engedélykészlet a fájl tulajdonosára vonatkozik. A második engedélykészlet a fájl tulajdonosának csoportjára vonatkozik. A harmadik engedélykészletet általában "others"-nek nevezik. Minden Linux fájlhoz tartozik egy tulajdonos és egy csoport.

Amikor az engedélyeket és a felhasználókat betűkkel ábrázolják, azt nevezik szimbolikus módnak. A felhasználók esetében az "u" a felhasználó tulajdonosra, a "g" a csoport tulajdonosra és az "o" a "másokra" utal. Az engedélyek esetében az "r" a "read" (olvasás), a "w" a "write" (írás) és az "x" az "execute" (végrehajtás).

Amikor a rendszer a fájl engedélyeit nézi, hogy meghatározza, milyen információkat nyújtson neked, amikor a fájllal interakcióba lépsz, egy sor ellenőrzést végez:

1. Először megvizsgálja, hogy te vagy-e a fájl tulajdonosa. Ha igen, akkor a tulajdonos felhasználó engedélyt használja, és további ellenőrzések nem fognak megtörténni.
2. Ha nem te vagy a fájl tulajdonosa, akkor a csoporttagságodat ellenőrzi, hogy megnézze, a tulajdonos csoportba tartozol-e. Ha igen, akkor a tulajdonos csoport engedélyt használja, további ellenőrzések nem történnek meg.
3. A "másoknak" vonatkozó engedélyeket akkor alkalmazzák, amikor az a fiók, amely a fájllal interakciót folytat, nem a fájl tulajdonosa és nem is a fájl csoport tulajdonosának tagja. Más szóval, a három mező egymással kizárólagosak: nem tartozhatsz több mező engedélybeállításai alá egy fájl esetében.

Az engedélyek túlmutatnak azok, hogy ki férhet hozzá egy fájlhoz. Minden felhasználó kap egy kifejezést, amely tartalmazza a három alapvető típusú engedélyt. A fenti példában a fájl tulajdonosának az alábbi engedélyei vannak:

rw-

Az engedélykifejezés minden karaktere jelzi, hogy adott engedély adott esetben engedélyezett-e vagy sem. A fenti példában az olvasás (r) és írás (w) engedélyezettek a fájlon. Azonban a végrehajtás (x) nem engedélyezett, ezért az engedélykifejezésben - jel található. Az engedély ezen mezője le van tiltva.

Tekintsük meg a tulajdonos csoport engedélyeit ebben a példában:

r--

Az olvasás (r) engedély a csoport tagjainak meg van adva, azonban az írás és a végrehajtási engedélyek nincsenek megadva.

Amikor a Linux fájlengedélyek számokkal kerülnek ábrázolásra, azt numerikus módnak nevezik. Numerikus módban egy háromjegyű érték reprezentálja a specifikus fájlengedélyeket (például 744.) Ezeket oktális értékeknek nevezik. Az első számjegy a tulajdonosi engedélyekre, a második a csoportengedélyekre, a harmadik pedig más felhasználókra vonatkozik. Minden engedélyhez numerikus érték van hozzárendelve:

* r (read): 4
* w (write): 2
* x (execute): 1

Pl egy 744 engedélyérték esetén az első számjegy a felhasználóra, a második a csoportra, a harmadik pedig más felhasználókra vonatkozik. Az egyes felhasználók értékét összeadva megkaphatod a fájlengedélyeket.

Például egy fájl olvasási, írási és végrehajtási engedélyekkel rendelkezhet a tulajdonos számára, és csak olvasási engedéllyel minden más felhasználóra, az így néz ki:

* Owner: rwx = 4+2+1 = 7
* Group: r-- = 4+0+0 = 4
* Others: r-- = 4+0+0 = 4

Az eredmények háromjegyű értéket eredményeznek: 744.

Beszéltünk arról, hogy hogyan lehet megtekinteni a fájlengedélyeket, kikre vonatkoznak és hogyan lehet olvasni, milyen engedélyek engedélyezettek vagy letiltottak. De mi valójában mit csinálnak a fájlengedélyek?

### Read(r)

Az olvasási engedélyt a fájl tartalmának elérésére használjuk. Pl egy olyan eszközzel, mint a "*cat*" vagy "*less*", hogy megjelenítsd a fájl tartalmát. Használhatsz egy szövegszerkesztőt, mint például a "*Vi*" vagy "*view*", hogy megjelenítsd a fájl tartalmát. Az olvasási engedély szükséges ahhoz is, hogy másolatot készíts egy fájlról, mivel a fájl tartalmához hozzáférésre van szükséged annak másolásához.

### Write(w)

Az írási engedély lehetővé teszi, hogy módosítsd a fájl tartalmát. Az írási engedély lehetővé teszi az is, hogy használd a redirect (>) vagy append (>>) shell operátorokat a fájl tartalmának módosításához. Írási engedély nélkül a fájl tartalmának módosítása nem megengedett.

### Execute(x)

A végrehajtási engedély lehetővé teszi, hogy végrehajtsd a fájl tartalmát. Általában a végrehajtható dolgok parancsok vagy fordított bináris alkalmazások. A végrehajtási engedély lehetővé teszi valakinek, hogy futtassa a Bash shell scripteket, Python programokat és számos interpretált nyelvet.

Végrehajtási engedély nélkül is van lehetőségünk végrehajtani a fájl tartalmát. Például használhatsz egy interpretert, amelynek van végrehajtási engedélye, hogy egy fájlt olvasson utasításokkal. Egy példa lenne egy Bash shell script meghívása:

$ bash script.sh

A futtatott file a "bash". A script.sh fájl olvasható a Bash interpreterrel, és a benne lévő parancsokat végrehajtja a bash. A Linux rendszerben gyakran vannak alternatív módszerek a feladatok elvégzésére.

## Könyvtár jogosultságok

A könyvtár típusú fájlok “d” attribútummal vannak megjelölve. Koncepciós szinten az engedélyek ugyanúgy működnek, de a könyvtárak ezeket a műveleteket másként értelmezik.

### Read(r)

Ahogy a rendes fájlok esetén, ez az engedély lehetővé teszi a könyvtár tartalmának olvasását. Ez azt jelenti, hogy megtekintheted a könyvtárban tárolt tartalmakat (vagy fájlokat). Ez az engedélyhez szükséges ahhoz, hogy pl az "*ls*" parancs működjön.

### Write(w)

A rendes fájlokhoz hasonlóan ez lehetővé teszi valakinek a könyvtár tartalmának módosítását. Amikor a könyvtár tartalmát megváltoztatod, vagy fájlokat adsz hozzá a könyvtárhoz, vagy fájlokat veszel ki a könyvtárból. Ezért ez szükséges a könyvtárban a fájlok mozgatásához (*mv*) vagy eltávolításához (*rm*). Az írási engedély szükséges új fájlok létrehozásához (a "*touch*" vagy egy fájl-átirányítási operátor használatával) vagy fájlok másolásához (*cp*) a könyvtárba.

### Execute(x)

Az engedélyek ezen mezője a könyvtárak esetében teljesen más. Gyakorlatilag arról van szó, hogy hozzáférést nyújt a könyvtárhoz. Az végrehajtási engedély a könyvtárban megengedi, hogy megnézd a kibővített információkat a könyvtárban tárolt fájlokról (például az "*ls -l*" használatával), emellett lehetővé teszi, hogy munkakönyvtárrá tegyük (a "*cd*" használatával).

A végrehajtási engedély hiánya egy könyvtárban furcsa módon korlátozhatja a többi engedélyt. Például hogyan adhatsz hozzá egy új fájlt egy könyvtárhoz (a "write" engedély felhasználásával), ha nem férhetsz hozzá a könyvtár metaadataihoz az új fájl információinak tárolásához? Nem tudsz. Ezért az általános gyakorlat, hogy a könyvtár típusú fájlok általában megadják az "execute" engedélyt legalább az egyiknek a tulajdonos felhasználó, a tulajdonos csoport vagy a többi felhasználó közül.

## File hozzáférések módosítása

A fájl- és könyvtár-engedélyeket a "*chmod*" parancsal módosíthatod, ami a "change mode" rövidítése. A fájlengedélyek módosításához numerikus módban beírod a "*chmod*" és az általad kívánt oktális értéket, például 744, mellett a fájl nevét. Szimbolikus módban a fájl neve mellé megadod egy felhasználóosztályt és az engedélyeket, amelyeket meg szeretnél adni nekik. Például:

$ **chmod** ug+rwx example.txt

$ **chmod** o+r example2.txt

Ez ad a felhasználónak és a csoportnak olvasási, írási és végrehajtási engedélyeket, és csak olvasási engedélyt ad a többi felhasználónak. Szimbolikus módban a "chmod u" a felhasználó tulajdonos engedélyeire vonatkozik, a "chmod g" a fájl csoport tulajdonosainak engedélyeire vonatkozik, a "chmod o" a fájl csoportján kívüli más felhasználók engedélyeire vonatkozik. Az összes felhasználó esetén használd a "chmod a" parancsot.

Esetleg meg akarod változtatni magát a fájl tulajdonosát. Ezt megteheted a "*chown*" parancs segítségével. Hasonlóképpen, a "*chgrp*" parancs segítségével megváltoztathatod egy fájl csoport tulajdonát.

## Speciális hozzáférések

A fájlokhoz és könyvtárakhoz különleges engedélyek is rendelkezésre állnak, amelyek a standard engedélybeállításokhoz kiegészítő jogokat biztosítanak.

* A SUID a felhasználó hozzáférés speciális engedélye, a fájl mindig úgy fut, mintha a tulajdonosa futtatná, függetlenül attól, hogy ki adja ki a parancsot.
* Az SGID lehetővé teszi, hogy egy fájlt a fájl csoport tulajdonosaként hajtsanak végre; egy a könyvtárban létrehozott fájl a csoport tulajdonába kerül. Ez akkor hasznos, ha a könyvtárat több csoporttag együtt használja, mivel az összes tag hozzáférhet az új file-okhoz és futtathatja is őket.
* A "sticky bit" egy könyvtár szintű speciális engedély, amely korlátozza a fájl törlését, azaz csak a fájl tulajdonosa törölheti a könyvtáron belül lévő fájlt.

## Keresés

Kétségtelen, hogy a find és a grep parancsok nagyon hasznosak és gyakran használtak a Linux rendszerekben. A find segítségével keresést végezhetsz a fájlrendszerben, míg a grep lehetővé teszi a szöveges fájlokban és a kimenetben a keresési minták megtalálását.

### Find

A find parancs használatával kereséseket végezhetsz a fájlrendszerben, és megtalálhatod a megadott kritériumoknak megfelelő fájlokat és könyvtárakat.

**Alapvető keresés a jelenlegi mappában:**

$find . -name "filename"

Ebben az esetben a find parancs a jelenlegi mappában keresi meg az összes olyan fájlt és könyvtárat, amelynek neve "filename".

**Rekurzív keresés egy adott mappában és alkönyvtáraiban:**

$find /path/to/directory -name "filename"

Ez a példa a /path/to/directory mappában és az összes alkönyvtárában keresi meg az összes olyan fájlt és könyvtárat, amelynek neve "filename".

**Keresés fájl méret alapján:**

$find /path/to/directory -size +10M

Ez a parancs a /path/to/directory mappában keresi meg az összes olyan fájlt és könyvtárat, amely nagyobb, mint 10 megabyte (M).

**Keresés fájl típusa alapján:**

$find /path/to/directory -type f

Ez a példa csak a /path/to/directory mappában keresi meg az összes fájlt.

**Keresés módosított fájlok alapján az elmúlt X napban:**

$find /path/to/directory -mtime -7

Ez a parancs a /path/to/directory mappában keresi meg az összes olyan fájlt és könyvtárat, amelyek az elmúlt 7 napban módosultak.

### Grep

A grep parancs a szöveges fájlokban és a parancs kimenetében keresi meg a keresési mintákat.

**Keresés szöveges fájlban:**

$grep "pattern" filename

Ebben a példában a grep parancs a "pattern" mintát keresi a filename szöveges fájlban.

**Rekurzív keresés egy adott mappában és alkönyvtáraiban:**

$grep -r "pattern" /path/to/directory

Ez a példa a grep parancsot az összes pattern-ot tartalmazó sor megtalálására alkalmazza a /path/to/directory mappában és az összes alkönyvtárában.

**Keresés a fájlokban, ahol figyelmen kívül hagyjuk a kisbetűs/nagybetűs különbséget:**

$grep -i "pattern" filename

Ez a példa a filename szöveges fájlban a "pattern" minta megtalálására keres, és figyelmen kívül hagyja a kisbetűs/nagybetűs különbséget.

**Keresés a fájlban egy adott szövegrészlettel kezdődő sorok után:**

$grep -A 3 "pattern" filename

Ez a parancs a "pattern" mintát tartalmazó sor után a következő 3 sort jeleníti meg a filename szöveges fájlban.

**Keresés a fájlban egy adott szövegrészlet előtt:**

$grep -B 2 "pattern" filename

Ez a példa a "pattern" mintát tartalmazó sor előtt az utolsó 2 sort jeleníti meg a filename szöveges fájlban.

**Kombinációra lehetőség:**

$grep -r "pattern" | $find /path/to/directory -name "filename"

Ebben a példában a grep parancs a "pattern" mintát keresi azokban a szöveges fájlokban, amiket a find parancs talál

## Rendszerinformáció

**w parancs:**

A *w* parancs a felhasználók jelenlegi munkamenetének információit mutatja meg. Ez tartalmazza a felhasználók bejelentkezési nevét, munkamenet-időt, aktuális munkamenetben futó folyamatok számát és a rendszer terhelését.

**du parancs:**

A *du* parancs a fájlok és könyvtárak méretét jeleníti meg a fájlrendszerben. Alapértelmezés szerint kilobájtban (KB) mutatja az eredményt.

Használat:

$du filename

$du -h filename # Az eredmény emberi olvasható formában jelenik meg (KB, MB, stb.)

$du -sh /path/to/directory # Egy mappában található összes fájl és könyvtár méretének összege

**free parancs:**

A free parancs a rendelkezésre álló memóriamennyiséget jeleníti meg a rendszeren. Az eredmény alapértelmezés szerint kilobájtban (KB) van megadva.

Használat:

$free

$free -h # Az eredmény emberi olvasható formában jelenik meg (KB, MB, stb.)

**uname parancs:**

Az uname parancs a rendszer információit jeleníti meg, például a rendszer nevét, verzióját, hardverét és operációs rendszerét.

Használat:

$uname

$uname -a # Az összes rendszerinformációt jeleníti meg

$uname -s # Rendszer neve

$uname -r # Rendszer verziószáma

$uname -m # Hardver típusa

## Monitoring

**dmesg parancs:**

A dmesg parancs a rendszer üzenetnaplóját (kernel log) jeleníti meg, amely tartalmazza a rendszer indításakor és futása közben történt eseményekről szóló információkat.

Használat:

$dmesg

**journalctl parancs:**

A journalctl parancs a systemd naplórendszerét jeleníti meg, amely tartalmazza a rendszer eseményeit és üzeneteit.

Használat:

$journalctl

$journalctl -xe # Kiterjesztett napló bejegyzések megjelenítése

$journalctl -f # Az új naplóbejegyzéseket folyamatosan megjeleníti (követi az eseményeket)

**top parancs:**

A top parancs interaktív módban jeleníti meg a rendszer aktuális állapotát és a folyamatok tevékenységét, lehetővé téve a folyamatok kezelését és a rendszer erőforrásainak ellenőrzését.

Használat:

$top

A top parancs futtatása után megjelenik egy táblázat, amelyen láthatod a rendszer terhelését és a folyamatok fogyasztott erőforrásait. A parancs futása közben megjelenik egy folyamatosan frissülő kimenet, amelyből a q billentyű lenyomásával tudsz kilépni.

**ps parancs:**

A ps parancs az aktuálisan futó folyamatok listáját jeleníti meg, és tartalmazza azok adatait, például a PID (Process ID), a folyamat státusza és a CPU kihasználtsága.

Használat:

$ps

$ps aux # Részletes információk a folyamatokról

$ps -e # Az összes folyamat megjelenítése

## Csomagok telepítése

Az apt és a yum két különböző csomagkezelő rendszer a Linux operációs rendszerekben. Az apt az Advanced Packaging Tool rövidítése, és a Debian és az Ubuntu alapú disztribúciókban használják, míg a yum a Yellowdog Updater, Modified rövidítése, és a Red Hat és a CentOS alapú disztribúciókban használják. Ezek a csomagkezelő rendszerek lehetővé teszik a felhasználók számára, hogy könnyedén telepítsenek, frissítsenek és távolítsanak el szoftvereket az operációs rendszerükből.

### Kitérő: Sudo

A *sudo* (superuser do) egy parancssori parancs a Unix alapú operációs rendszerekben, amely lehetővé teszi a felhasználók számára, hogy bizonyos parancsokat és műveleteket szuperuser (root) jogosultságokkal hajtsanak végre. A szuperuser jogosultságok szükségesek olyan műveletek elvégzéséhez, amelyek a rendszer működésére, konfigurációjára vagy más felhasználók munkájára vonatkoznak, és amelyek általában az egész rendszerre vagy a rendszer lényeges részeire vonatkoznak.

A *sudo* használata az alapvető biztonság és rendszerintegritás megőrzése érdekében ajánlott. Azt jelenti, hogy a rendszer adminisztrátorai vagy megbízható felhasználói csak akkor kapnak szuperuser jogosultságokat, amikor szükség van rá, és csak a szükséges műveleteket hajtják végre. Azáltal, hogy nem használjuk állandóan a szuperuser jogosultságokat, minimalizálhatjuk a potenciális károk és hibák kockázatát.

A *sudo* parancs használatakor a felhasználót a rendszer által meghatározott sudoers fájlban definiált jogosultságokkal ellenőrzik. Ha a felhasználó szerepel ebben a fájlban, és jogosult a sudo parancs használatára, akkor a rendszer kérheti a felhasználó jelszavát, mielőtt engedélyezi a szuperuser jogosultságok használatát. Ezáltal biztosítja, hogy csak a jogosult felhasználók használhatják a sudo parancsot.

A sudo parancs használata általában az alábbi módon zajlik:

$sudo parancs

Például:

$sudo apt update # A csomaglisták frissítése szuperuser jogosultsággal

Vagy:

$sudo rm -rf /var/log # A /var/log könyvtár és tartalma rekuzívan törlése szuperuser jogosultsággal

Az sudo parancs használata előtt megadhatod a felhasználó jelszavadat, hogy hitelesítsd azonosságodat (kéri a linux). Ha a megadott jelszó helyes, akkor a rendszer engedélyezi a sudo parancsot.

### APT (Debian és Ubuntu):

**Telepítés:**

$sudo apt update # Frissíti a csomaglistákat

$sudo apt install package\_name # Telepíti a csomagot

**Csomag keresése:**

$apt search keyword # Keresés a csomagok között kulcsszó alapján

**Csomagok frissítése:**

$sudo apt update # Frissíti a csomaglistákat

$sudo apt upgrade # Frissíti a telepített csomagokat

**Csomag törlése:**

$sudo apt remove package\_name # Eltávolítja a csomagot, de meghagyja a konfigurációs fájlokat

$sudo apt purge package\_name # Eltávolítja a csomagot és a konfigurációs fájlokat is

## Networking

**iptables parancs:**

Az iptables parancs egy eszköz a Linux tűzfal konfigurálására és kezelésére. Segítségével beállíthatod a hálózati csomagok irányítását és szűrését, amelyek beérkeznek és elhagyják a rendszert.

Használat:

$iptables -L # Megjeleníti az aktuális tűzfal szabályokat

$iptables -A INPUT -p tcp --dport 22 -j ACCEPT # Engedélyezi a bejövő SSH csatlakozásokat

$iptables -A INPUT -j DROP # Letiltja az összes többi bejövő csomagot

**ip parancs:**

Az ip parancs az iproute2 eszközkészlet része, és a hálózati interfészek konfigurálására, illetve az IP címek és útvonalak megtekintésére használható.

$ip addr show # Megjeleníti az összes hálózati interfész IP címét és állapotát

$ip link show # Megjeleníti az összes hálózati interfész nevét és állapotát

$ip route show # Megjeleníti az aktuális IP útvonalakat

**ipconfig parancs:**

Az ipconfig parancs az Windows rendszerekben használatos parancs, amely a hálózati interfészek konfigurációját és információit jeleníti meg.

Ha Linux rendszert használsz, akkor az ip parancsot használhatod a hálózati interfészek konfigurálására és információinak lekérésére.

**ping parancs:**

A ping parancs segítségével ellenőrizheted, hogy egy adott hálózati eszköz (például egy távoli számítógép) elérhető-e és válaszol-e a hálózati csomagokra.

Használat:

$ping google.com # Pingel egy távoli szerver IP címét vagy URL-jét

**traceroute parancs:**

A traceroute parancs segítségével nyomon követheted az adatcsomagok útját a forrástól a célállomásig, és megtekintheted az útvonalon lévő minden ugrást (hop).

Használat:

$traceroute google.com # Nyomon követi az adatcsomagok útját a célszerverig

## Scriptek

### Bash

* Bash script létrehozása:
  + Hozz létre egy Bash szkriptet egy szövegszerkesztőben (például Nano vagy Vim) és mentsd el a fájlt ".sh" kiterjesztéssel. Például: myscript.sh.
* Shebang sor:
  + Kezdd a Bash szkriptet egy shebang sorral, hogy megmond a rendszernek, melyik értelmezőt használja. A leggyakoribb shebang a Bash-hoz: #!/bin/bash.
* Commentek:
  + Használj megjegyzéseket (sorokat, amelyek # karakterrel kezdődnek), hogy dokumentáld a szkripted és jobban megértsék mások.
* Változók:
  + Deklarálj változókat adatok tárolásához. A Bash változók nem típusosak, tehát nem kell megadnod a változók adattípusát.
* Függvények:
  + Definiálj függvényeket, hogy egy adott parancssort lehessen újrahasznosítani a szkriptedben.
* Vezérlési eszközök:
  + Elágazás
  + Ciklusok
* Paraméterek:
  + Kötelező paraméterek
  + Opcionális paraméterek

#!/bin/bash

# Declare variables

name="John"

age=30

# Print the variables

echo "My name is $name and I am $age years old."

**Parancsbeágyazás:**

Futtass parancsokat és tárolják az eredményt egy változóban a parancsbeágyazással: *$(...)*.

#!/bin/bash

# Get the current date and store it in a variable

current\_date=$(date)

# Print the current date

echo "Today is $current\_date"

**Bemenet olvasása:**

Használd a *read* parancsot a felhasználó bemenet olvasására és tárolására egy változóban.

#!/bin/bash

# Ask for user input

echo "Enter your name: "

read name

# Print a greeting with the user's name

echo "Hello, $name!"

**Feltételek (if-else):**

#!/bin/bash

# Get the user's age

echo "Enter your age: "

read age

# Check if the user is old enough to vote

if [ "$age" -ge 18 ]; then

echo "You are eligible to vote."

else

echo "You are not eligible to vote."

fi

**Ciklusok:**

#!/bin/bash

# Loop through numbers from 1 to 5

for i in {1..5}; do

echo "Number: $i"

done

#!/bin/bash

# Loop until the user enters 'exit'

while true; do

echo "Enter a value (type 'exit' to quit): "

read value

if [ "$value" = "exit" ]; then

break

fi

echo "You entered: $value"

done

**Függvények:**

#!/bin/bash

# Define a function

say\_hello() {

echo "Hello from the function!"

}

# Call the function

say\_hello

**Paraméterek:**

A Bash szkriptekben paramétereket tudsz használni, hogy értékeket adj tovább a szkriptnek futtatáskor. A paramétereket speciális változókkal érheted el: $1, $2, $3, és így tovább, amelyek az első, második, harmadik és további paramétereket jelentik. Itt van egy példa arra, hogyan használhatod a paramétereket egy Bash szkriptben:

#!/bin/bash

# Check if at least one parameter is provided

if [ $# -eq 0 ]; then

echo "Usage: $0 <parameter1> [<parameter2> ...]"

exit 1

fi

# Accessing the first parameter

param1=$1

echo "Parameter 1: $param1"

# Accessing the second parameter (if provided)

if [ $# -ge 2 ]; then

param2=$2

echo "Parameter 2: $param2"

fi

# Accessing all parameters in a loop

echo "All Parameters:"

for param in "$@"; do

echo "- $param"

done

Ebben a példában a szkript ellenőrzi, hogy legalább egy paramétert megadtak-e futtatáskor. Ha nincs paraméter megadva, akkor egy üzenetet ír ki a használatról és kilép. Ha egy vagy több paramétert megadtak, akkor az első és második paraméter értékét kiírja (ha meg van adva). Ezután egy ciklussal végigiterál a szkriptnek átadott összes paraméteren és kiírja azok értékét.

<https://askubuntu.com/questions/939620/what-does-mean-in-bash>

<https://stackoverflow.com/questions/3898665/what-is-in-bash>

**Bemenet:**

$./myscript.sh hello world

**Kimenet:**

Parameter 1: hello

Parameter 2: world

All Parameters:

- hello

- world

**Bemenet:**

$./myscript.sh

**Kimenet:**

Usage: ./myscript.sh <parameter1> [<parameter2> ...]

A paraméterek használata lehetővé teszi, hogy sokoldalúbbá és újrafelhasználhatóvá tedd a Bash szkriptjeidet, mivel az argumentumok értékének megadásával testreszabhatod a szkript viselkedését.

### Python

# 

# Források és további info

* <https://www.linux.com/what-is-linux/>
* https://www.redhat.com/sysadmin/linux-file-permissions-explained
* <https://www.synopsys.com/glossary/what-is-open-source-software.html>
* <https://en.wikipedia.org/wiki/Linus_Torvalds>