

Labor: Linux 3

Jelmagyarázat

A mérési utasításban az egyes lépésekben szükséges választásokat illetve beírandó parancsokat **kék félkövér** szöveg jelzi. Ahol ezekre a választásokra vagy parancsokra csak hivatkozás történik, ott **fekete félkövér** betűtípust használunk. A csillaggal megjelölt feladatok elvégzése a kiválóan megfelelt minőség feltétele.

Bevezetés

A Linux 3-as labor feladatai függetlenek a korábbi Linux 1-2 laboroktól. Ebben a laborban egy grafikus felülettel rendelkező asztali Linux operációs rendszert telepítünk majd és ezen vizsgáljuk a rendszer felépítését és komponenseit, illetve azok testreszabhatóságát.

1. Feladat: A Linux Mint 19.1 "Tessa" telepítése

E feladat célja, hogy egy új virtuális gépbe telepítsünk egy asztali Linux operációs rendszert, amiben a későbbi feladatok során dolgozni fogunk. A telepítésre kerülő operációs rendszer a Linux Mint nevet viseli: ez egy Ubuntu alapokra épülő disztribúció, mely alapértelmezetten a Cinnamon nevű asztali környezetet használja.

Érdeemes megjegyezni már itt az elején, hogy az "asztali" és a "szerver" Linux között, illetve az egyes disztribúciók között a Linux kernel tekintetében nincs különbség, mindössze a telepítésre kerülő és az alapértelmezett programokban térnek el ezek. Míg a korábbi laborokon használt Ubuntu Server telepítő nem telepítette a grafikus környezetet, addig egy Ubuntu Desktop telepítő ugyanazzal a Linux kernellel az asztali környezetet is telepíti. Az elérhető disztribúciók között nagyon széles a választék, illetve a telepítést követően is tetszőlegesen testre szabhatjuk és átkonfigurálhatjuk a rendszert, ahogy azt nemsokára látni fogjuk.

Ebben a feladatban azért a Linux Mint-re esett a választás, mert az ebben alapértelmezetten megtalálható Cinnamon felület hasonlít talán leginkább a Windows rendszereken megszokotthoz, így Windows alapokkal talán ez kezelhető a legkönnyebben.

1.1. Virtuális gép létrehozása

Első lépésben hozzuk létre a virtuális gépet, amiben dolgozni fogunk:

- Töltse le a [telepítő ISO fájlt innen](#), ha az nincs még meg a használt számítógépen! (Az IB141-142-es laborgépeken megtalálható az ISO fájl.)
- Hozzon létre egy új virtuális gépet a VMWare Player programban!
- Adja meg a telepítő ISO fájlt az **Installer disc image file (iso)** mezőben!
- Válasszon **Linux** operációs rendszert és azon belül **Ubuntu 64-bit-et**!

- Adjon **Linux 3** nevet és mentse a virtuális gépet a laborgépeken a D:\ meghajtó alá egy saját Neptun kódjáról elnevezett mappába!
- Hozzon létre egy új lemezkép fájlt 20 GB mérettel!
- Nyissa meg a **Customize virtual machine** ablakot, és állítson be legalább 2 GB memóriát és 2 processzormagot a rendszer számára!
- Indítsa el a virtuális gépet!

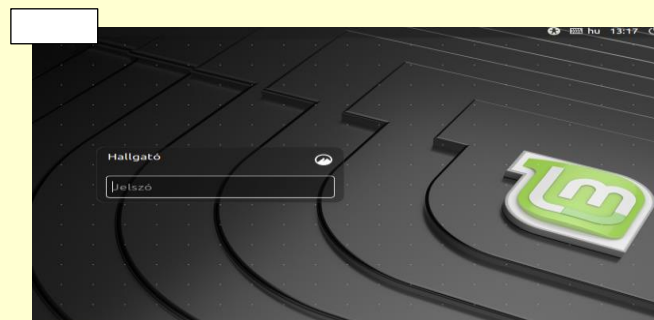
1.2. Az operációs rendszer telepítése

Mivel a virtuális gépben egyedül a telepítő CD bootolható, ezért a rendszer automatikusan arról tölt be. A Linux disztribúcióknál általánosan megszokott megoldás a telepítőt egy LiveCD formájában elérhetővé tenni, így maga a rendszer telepítés és bármilyen elköteleződés nélkül is könnyen kipróbálható. Mivel a rendszert Live CD formában futtatva a változtatásaink újraindításakor elvesznek, illetve a teljesítmény se ugyanaz mint a merevlemezről futtatva, ezért telepítsük fel az operációs rendszert a virtuális gép egyelőre üres merevlemezére!

- Dupla kattintással indítsa el az asztalon található **Install Linux Mint** programot!
- Válasszon nyelvet! A leírás során mi a magyar változatra fogunk hivatkozni, de nyugodtan használhat angol nyelvet is, ha azt preferálja!
- Válasszon billentyűzet kiosztást!
- Engedélyezze a **harmadik féltől származó szoftverek telepítését!**
- Ebben az új és üres virtuális gépben válassza a "**Merevlemez törlése, és Linux Mint telepítése** opciót", majd hagyja jóvá a lemez javasolt partícionálását! Ha olyan számítógépre telepítenénk, ahol van más operációs rendszer is, akkor nagyobb körültekintéssel kellene eljárunk, de szerencsére a telepítő is segít ilyenkor.
- Válassza ki a **Budapest** időzónát!
- A felhasználó és a gépnév beállításánál adja meg a következő lehetőségeket (**ezeektől eltérni tilos!**):
 - Az Ön neve: **Hallgató**
 - A számítógép neve: **az Ön neptun kódja**
 - Felhasználói név: **hallgato**
 - Jelszó: **oprepass**
 - A többi beállításon ne változasson!
- Várja meg, amíg a telepítő végez a fájlok másolásával, majd indítsa újra a virtuális gépet!

2. kérdés

Készítsen egy képet a bejelentkező képernyőről, ahol a rendszer a jelszavát kéri!



- Jelentkezzen be a megadott **oprepass** jelszóval!

1.3. Illesztőprogramok telepítése

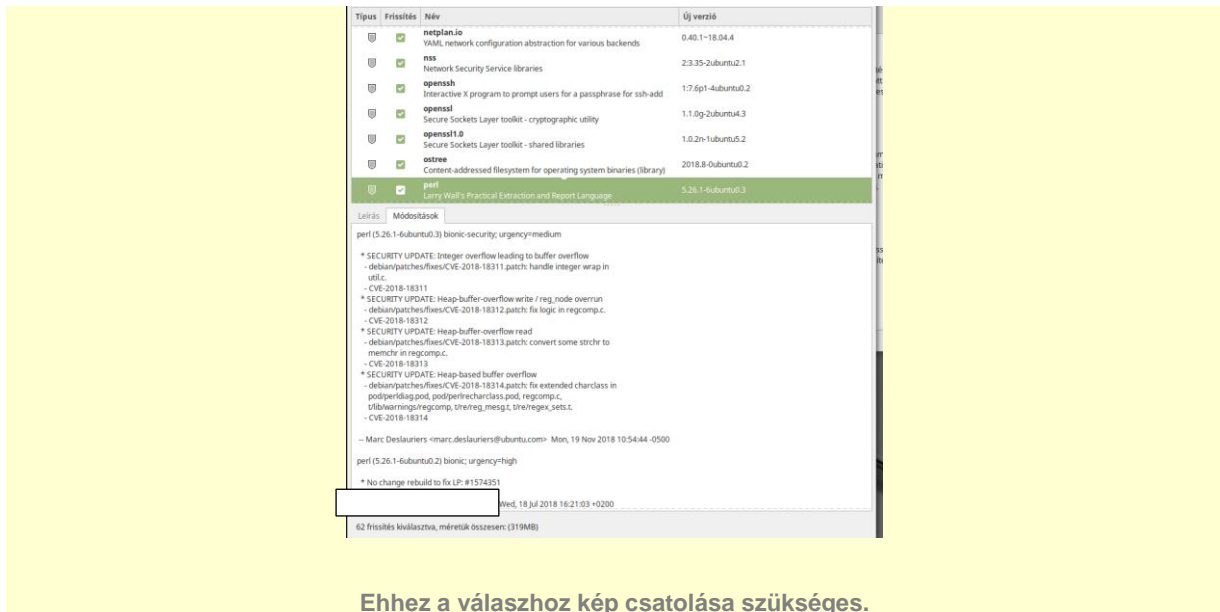
- VirtualBox használata esetén:
 - A rendszer a jobb felső sarokban figyelmeztet, ha még nincs betöltve minden szükséges illesztőprogram a grafikus kártyához, ezért egyelőre szoftveres megjelenítési üzemmódban működik
 - Ha lát ilyen figyelmeztetést, az azt jelenti, hogy a grafikus kártyán hardveresen gyorsított feladatokat is a CPU terhelésével oldja meg.
 - Első lépésben erre keresünk majd megoldást!
 - Belépés után az üdvözlőképernyő fogadja: nyissa meg ezen az **Első lépések** fület!
 - Indítsa el az Illesztőprogramkezelőt a hozzá tartozó **Indítás** gombra kattintással! Adja meg a szokásos **oprepass** jelszót, hogy a program root jogokkal futhasson.
 - A gyorsítótár frissítése után a rendszer felajánlja, hogy illesztőprogramot telepíthet a videokártyához - válassza a **virtualbox-guest-x11** opciót, majd kattintson a **Változtatások alkalmazása** gombra!
 - Ha kész a telepítés, akkor kapcsolja ki a virtuális gépet!
 - A virtuális gép kikapcsolt állapotában nyissa meg a VirtualBox-ban a virtuális gép beállításait, és a Grafikus beállítások oldalon pipálja be a 3D gyorsítás engedélyezését!
- VMWare Player esetén:
 - A VMWare Player-ben futó virtuális gép nem mindig hagyja átméretezni az ablakot, így előfordulhat, hogy 800x600-as felbontással indul el a vendég operációs rendszer.
 - Ha nagyon kicsi a felbontás, akkor nyissa meg a rendszer menüben (bal alsó sarokban a Start menü helyén) a Beállítások almenüben található Kijelző programot és állítson be egy kényelmesebb felbontást!

1.4. Frissítések telepítése

- A következő lépésben telepítsük az elérhető frissítéseket!
 - Ehhez használjuk az induláskor automatikusan megjelenő üdvözlőképernyő Első lépések fülét! (Ha esetleg bezárta volna már az üdvözlőképernyőt, akkor újra megnyithatja a rendszer menüből, ha a menüt megnyitva elkezdí begépelni az **Üdvözlőképernyő** szöveget!)
 - Görgessen le a Frissítéskezelő részig és kattintson az **Indítás** gombra!
 - **Okézza** le a megjelenő információkat, majd a lista felett az eszköztáron a **Telepítés** gombra kattintással indítsa el a frissítések telepítését! Első körben a program csak magát a frissítéskezelőt frissíti majd, újraindul.
 - A második körben egy sokkal hosszabb listát látunk a frissíthető komponensekről. Egyesével dönthetünk a komponensek frissítéséről, és a döntéshez az egyes elemeket kijelölve az alsó részen kapunk tájékoztatást. Válasszunk ki egy tetszőleges elemet a listából (pl. openssl) és nézzük meg az alsó részen a **Leírás** illetve a **Módosítások** fűlek tartalmát!

3. kérdés

Készítsen képet egy tetszőleges frissítésre váró csomagra vonatkozó Módosítások fűlről!



Ehhez a válaszhhoz kép csatolása szükséges.

- Telepítsünk minden listázott frissítést az eszköztár **Telepítés** gombjára kattintva!
 - A csomagok telepítési folyamatát mutató csúszka alatt kattintsunk a **Részletek** feliratra, amikor a rendszer túljutott már a letöltésen és a telepítést végzi!
 - Az így megjelenített konzolon a korábbi laborokról már jól ismert **apt upgrade** parancs kimenetét láthatjuk, tehát a GUI felület háttérében ugyanaz a parancs fut, amit konzolról mi magunk is elindíthatunk volna kézzel.
 - Ha elkészült a frissítések telepítése zárja be a segédprogramot! A frissítések kapcsán sose kényszerít bennünket a rendszer arra, hogy újraindítsuk a számítógépet - ha olyan komponenseket telepített, melyek csak a rendszer újraindításával tölthetők be (pl. kernel frissítések), akkor azok a következő újraindításig még a korábbi verzióban futnak.

1.5. A megjelenés testreszabása

- Az üdvözlőképernyő Első lépések lapján tovább görgetve lefelé lehetőségünk van választani a tálca (Linux terminológiával panel) megjelenési módjai között az **Asztali kiosztás** fejezetben: a **Hagyományos** kiosztás a Windows 7 és korábbi, az **Újszerű** kiosztás a Windows 8 és későbbi asztal megjelenítését követi. Válasszon tetszése szerint ezek közül!
- Tovább görgetve lefelé az üdvözlőképernyő Első lépések lapján a Rendszerbeállítások blokkhoz érkezünk. Indítsa el a **Rendszerbeállítások** programot az **Indítás** gombra kattintva!
- Fussa végig az elérhető beállításokat, ha szeretne kísérletezhet a különböző beállításokkal (Effektusok, Háttérképek, Témák, stb.)!
- Görgessen a Rendszerbeállítások képernyő aljára és nyissa meg a **Szoftverforrások** pontot!

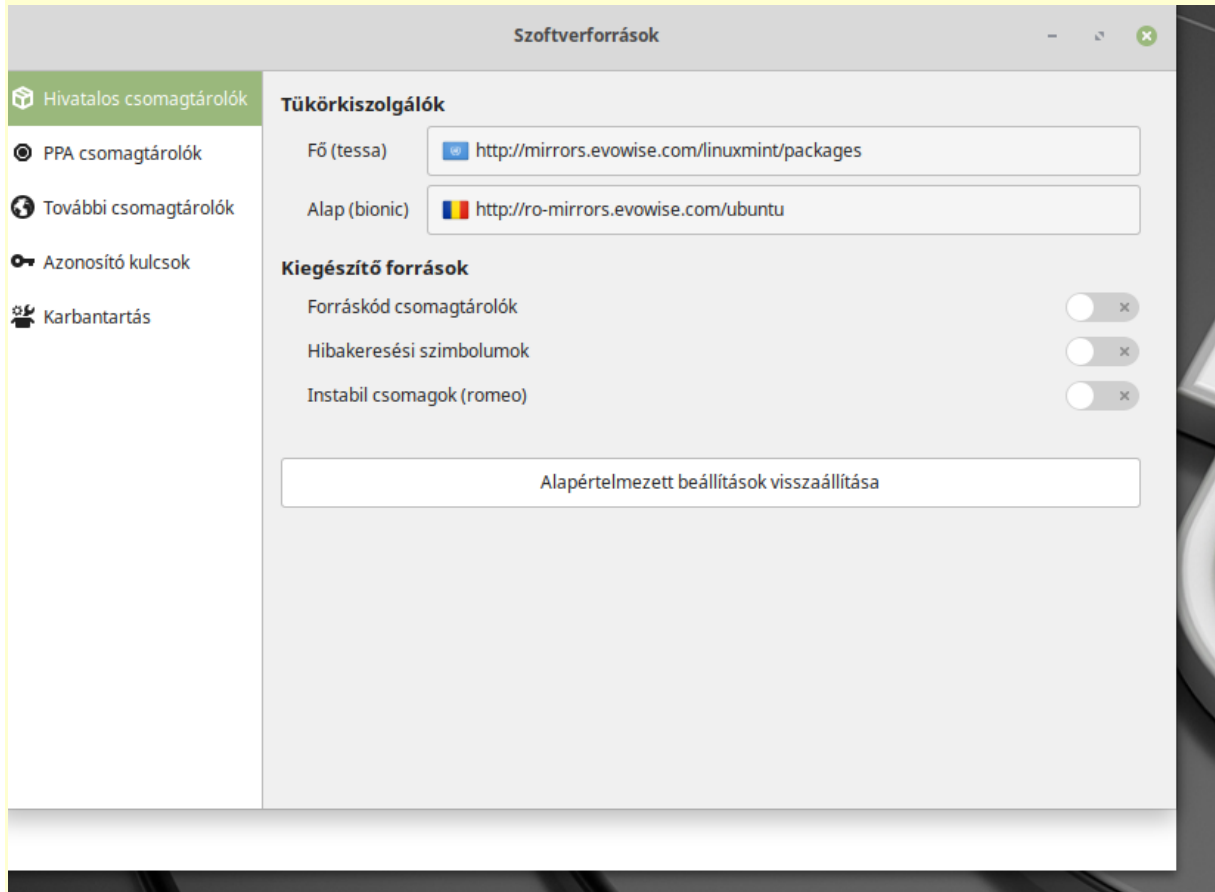
1.6. Szoftverforrások beállítása

- A rendszer két fő tükörszolgáltatót használ a csomagok telepítésére: A **Fő (Tessa)** kiszolgáltató a Linux Mint csomag adatbázisa, az **Alap (Bionic)** a rendszer alapjául szolgáló Ubuntu csomagok forrásának tükörszerverét állítja be.
- Kattintson az egyes alapértelmezett tükörszerverekre és állítsa be a legnagyobb letöltési sebességgel kijelzett alternatívát, hogy a csomagokat ne a tengerentúlról töltsse le a rendszer!

- Kattintson az eszköztár **OK** gombjára, hogy a rendszer betöltse ezeket a forrásokat! (A szerverek grafikus felületen történő beállítása ekvivalens a `/etc/apt/sources.list.d/official-package-repositories.list` fájl kézi szerkesztésével majd a **sudo apt update** parancs kiadásával.)

4. kérdés

Készítsen egy képet a Szoftverforrások program ablakáról a leggyorsabb tükörkiszolgálók beállítása után!



Ehhez a válaszhoz kép csatolása szükséges.

2. feladat: Ismerkedés a telepített szoftverekkel

A legtöbb asztali Linux disztribúció számos nyílt forráskódú programot előre telepítve tartalmaz.

2.1. A rendszerrel együtt telepített alkalmazások

Tekintse át a képernyő bal alsó sarkából elérhető Menüből elérhető programokat! A legtöbb hétköznapi feladatra (böngészés, levelezés, szövegszerkesztés, stb.) talál előre telepített alkalmazást a rendszerben.

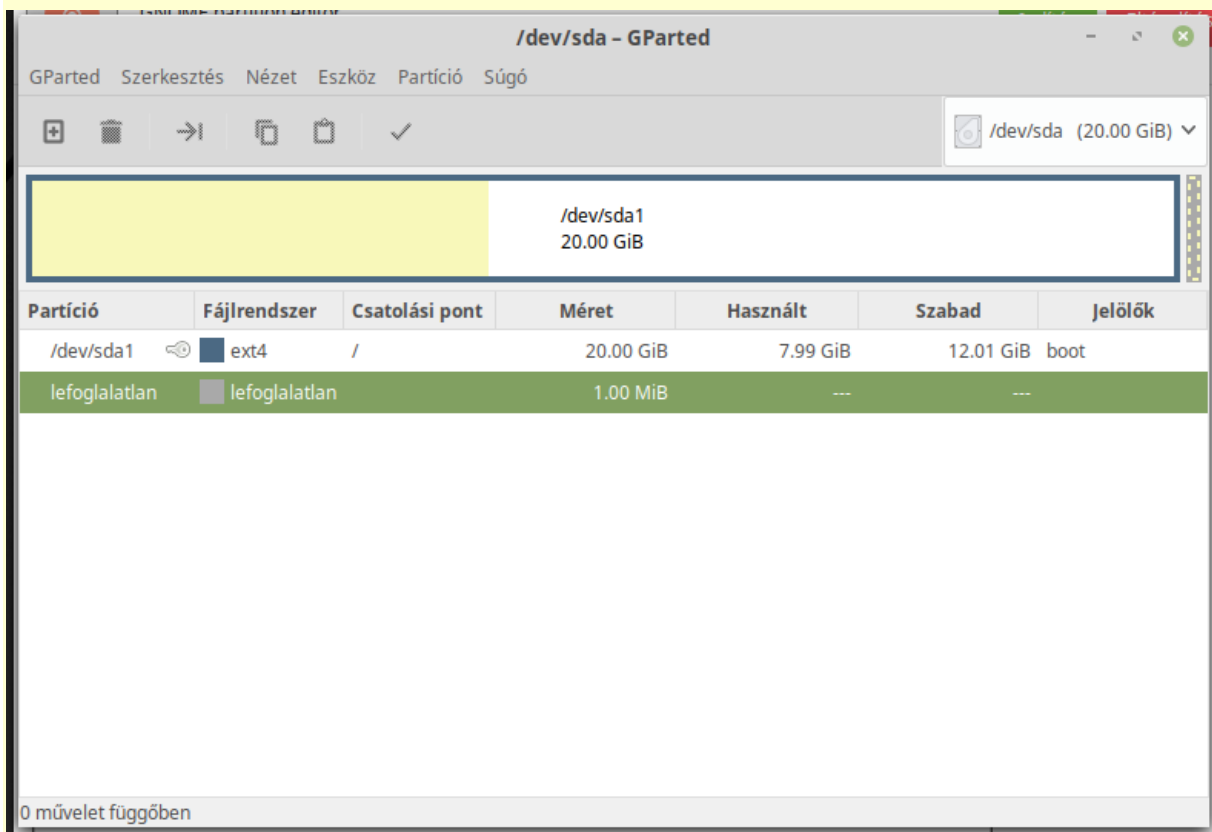
2.2. A Szoftverkezelő használata

Ha további asztali alkalmazásokat szeretnénk telepíteni (pl. Skype kliens), akkor erre használhatjuk a **Szoftverkezelőt**, mely a rendszer menüben az **Adminisztráció** pont alatt érhető el.

- Nyissa meg a **Szoftverkezelőt**!
- Telepítse a **Gparted** nevű partíciókezelő programot a **Szoftverkezelő** használatával a **Telepítés** gombot használva!
- Indítsa el az alkalmazást az **Indítás** gombbal! A program az elérhető egyetlen lemez (/dev/sda) egyetlen partícióját mutatja (/dev/sda1).

5. kérdés

Készítsen egy képet a Gparted felületéről!



Ehhez a válaszhoz kép csatolása szükséges.

2.3. A csomagkezelő használata

A Szoftverkezelő segítségével egyszerűen telepíthető a legtöbb gyakori alkalmazás az összes szükséges függőségével együtt. Ha nagyobb kontrollt szeretnénk a telepített csomagok felett, akkor erre a célra a **Csomagkezelő** szolgál, melyet a rendszer menü Adminisztráció pontja alól érhet el.

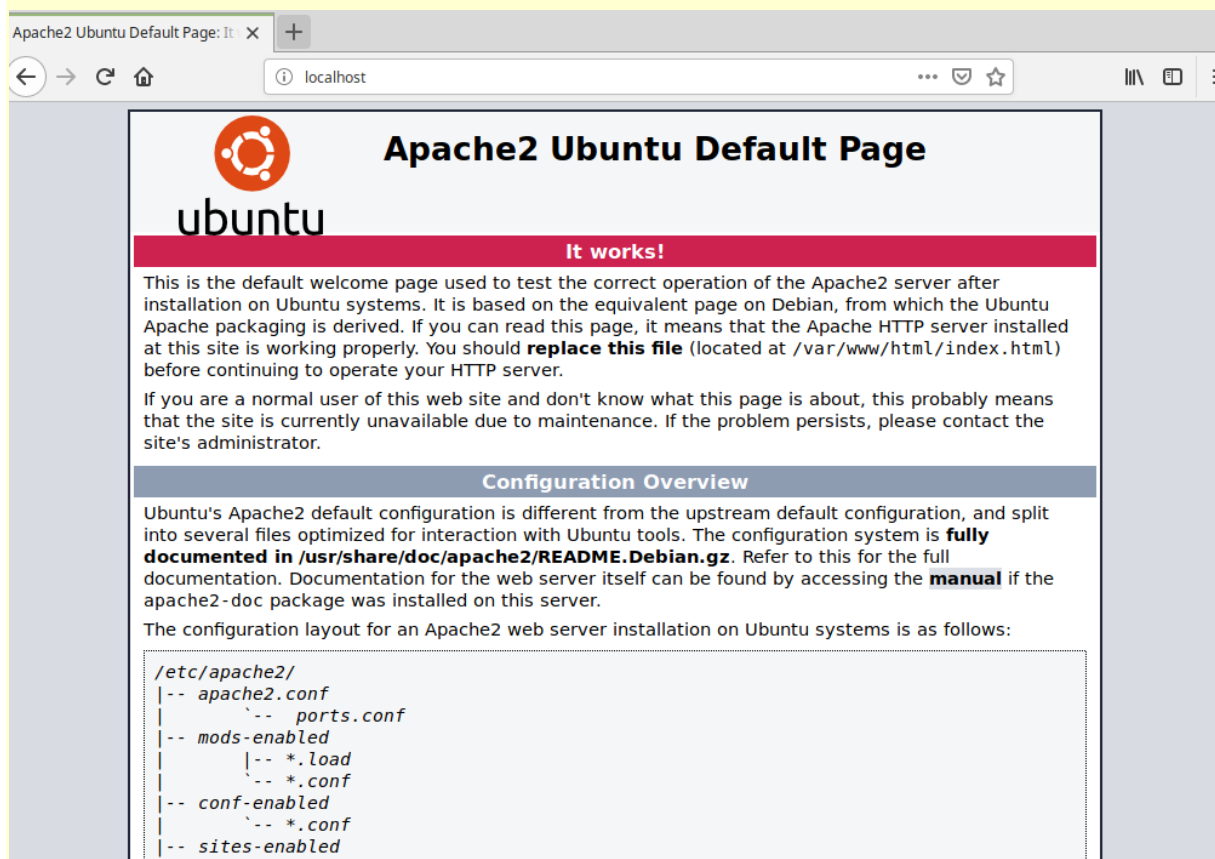
- Nyissa meg a **Synaptic csomagkezelőt**!
- A **World Wide Web** kategóriában válassza ki az **apache2** csomagot telepítésre dupla kattintással!

- A program kijelzi a függőségeket, melyeket szintén telepíteni kell: a **Kijelölés** gombra kattintással hagyja jóvá ezeket!
- Az előző lépésekben csak kijelöltük telepítésre a csomagokat. A telepítés elvégzéséhez kattintson az eszköztár **Alkalmaz** gombjára!

A folyamat részleteit a **Részletek** feliratra kattintva megjelenítve ismét csak az **apt** parancs ismerős kimenete fogad bennünket. A grafikus felületen végrehajtott lépésekkel ekvivalens lett volna a **sudo apt install apache2** parancs kiadása a terminálban. Bármelyik megoldást is választottuk, az telepítés eredményeként a gépünkre került az Apache 2 webservert, amit a korábbi laborokban is használtunk - ha webes fejlesztési feladatunk lenne, most a saját gépünkön is tudnánk tesztelni pontosan olyan körülmények között, mint ahogy az a szerveren fut majd.

6. kérdés

Nyisson egy Firefox böngészőt és nyissa meg a <http://localhost> oldalt, hogy a frissen telepített webservert kezdőlapjáról képet tudjon készíteni!



Ehhez a válaszhoz kép csatolása szükséges.

2.4. A fájlkezelő használata

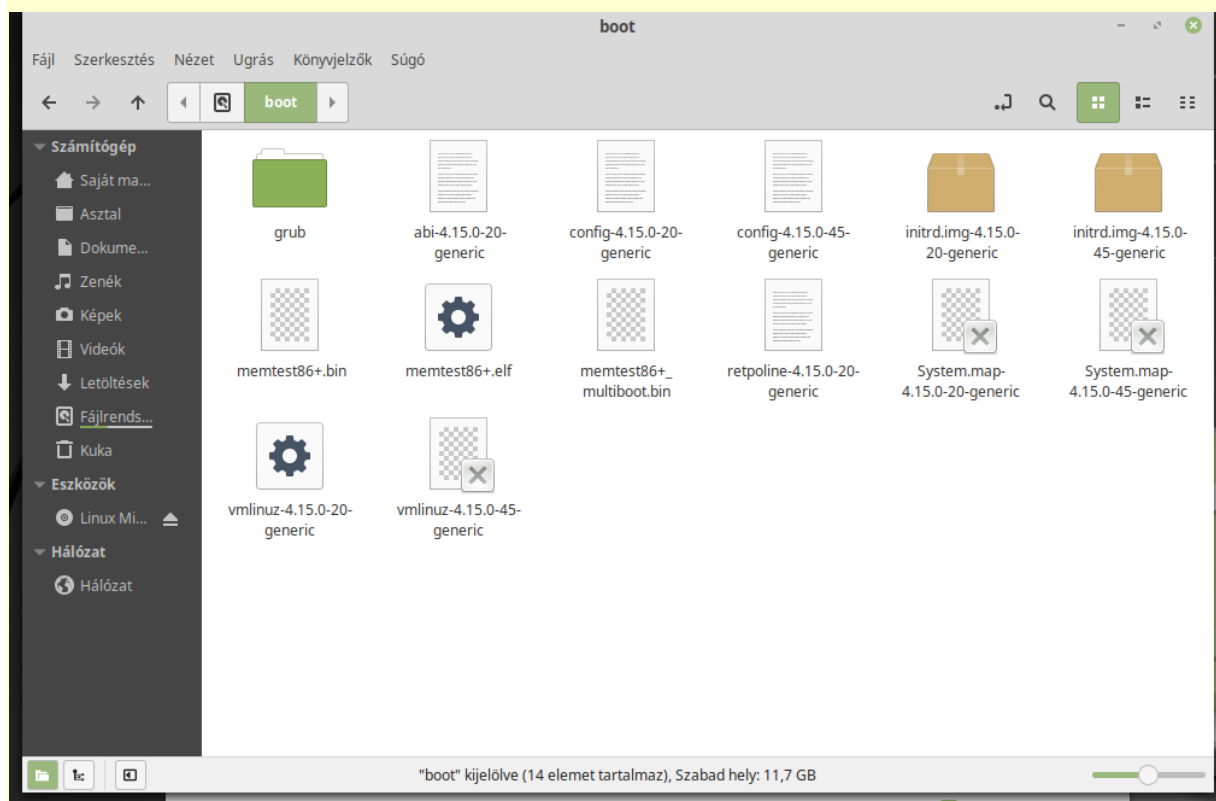
Nyissa meg az asztalon található **Saját mappa** könyvtárat! Ezzel a hallgató felhasználó home könyvtára (a /home/hallgato mappa) tartalma jelenik meg a fájlkezelőben.

Az eszköztár **Ugrás a szülőmappára** gombját (felfelé nyíl, az eszköztár harmadik gombja) duplán megnyomva a fájlrendszer gyökérkönyvtárába jutunk.

A **boot** könyvtárba belépve a rendszer indulásához szükséges fájlokat látjuk: itt találhatóak a telepített kernel vmlinuz-* néven (több is lehet telepítve, de egyszerre természetesen csak egy verzió indul el a rendszer indulásakor), a bootolás során a kernel által használt kezdeti fájlrendszer (initrd.img*), a memtest86 alkalmazás (mellyel operációs rendszer betöltése nélkül tesztelhetjük a fizikai memóriát hibák után kutatva), illetve a grub könyvtár, ami az alapértelmezett Linux bootloader-t tartalmazza.

7. kérdés

Készítsen egy képet a /boot könyvtár tartalmáról!



Ehhez a válaszhoz kép csatolása szükséges.

3. A rendszer működésének mélyebb megértése

A Linux kernel kényelmesen elérhető felületet biztosít a programok felé: a Linux alapvető filozófiája szerint a rendszerben minden fájl, így a fájlrendszerbe ágyazott módon érhetünk férhetünk hozzá a legtöbb állapotváltozóhoz illetve adhatunk parancsokat a kernelnek. E megközelítés előnye, hogy tetszőlegesen egyszerű programok (például egyszerű szöveges shell scriptek) is hozzáférhetnek ezekhez a változókhöz.

Nyisson egy **Terminál** alkalmazást a rendszer menüből!

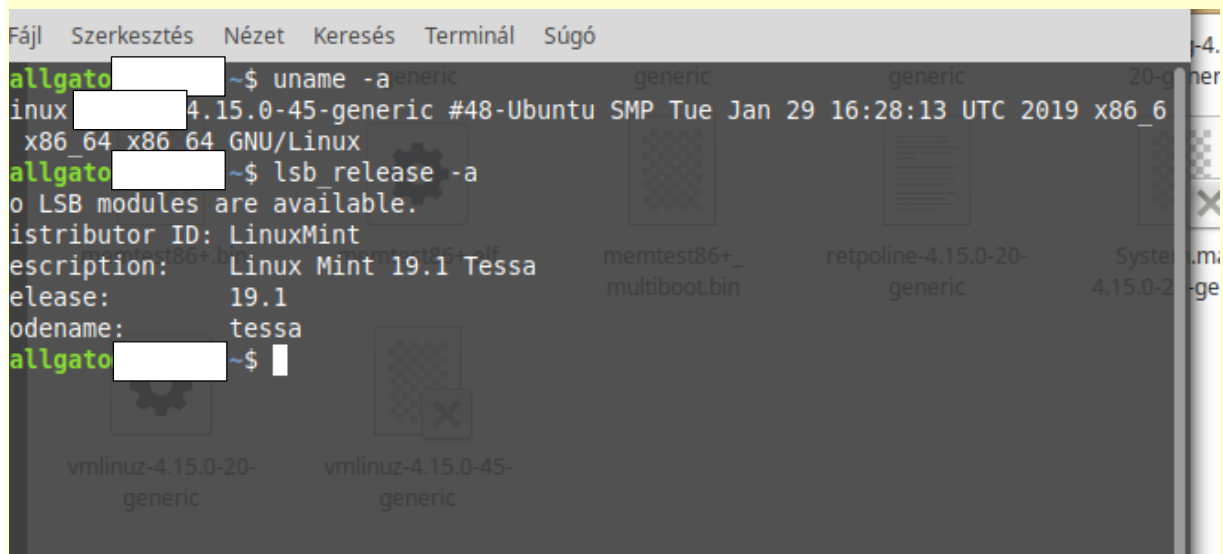
3.1. A kernel és a disztribúció verziója

A kernel verziójának meghatározásához adja ki az **uname -a** parancsot! A megjelenített kernel verzió valószínűleg nem a legfrissebb verzió amit a **/boot** könyvtárban látott: ha ugyanis az 1.4-es feladatban lefutott frissítés óta még nem indította újra a virtuális gépet, akkor a rendszer még a telepített és nem a legfrissebb kernellel fut.

Az éppen használt Linux disztribúció nevének és verziójának kiírásához adja ki az **lsb_release -a** parancsot!

8. kérdés

Készítsen képet az **uname** és az **lsb_release** parancsok kimenetéről!



```
Fájl Szerkesztés Nézet Keresés Terminál Súlyó
allgato [redacted] ~$ uname -a
Linux [redacted] 4.15.0-45-generic #48-Ubuntu SMP Tue Jan 29 16:28:13 UTC 2019 x86_64
x86_64 x86_64 GNU/Linux
allgato [redacted] ~$ lsb_release -a
No LSB modules are available.
Distributor ID: LinuxMint
Description: Linux Mint 19.1 Tessa
Release: 19.1
Codename: tessa
allgato [redacted] ~$
```

Ehhez a válaszhoz kép csatolása szükséges.

Ahogy korábban már említettük, a kernel az állapotváltozóit a fájlrendszer részeként jeleníti meg. A fájlrendszer gyökérkönyvtárában található **/dev**, a **/proc** és a **/sys** könyvtárak alatt található könyvtár hierarchia nem létezik a lemezen, ezeken a helyeken az elérési útvonalakon a kernel állapotváltozóit érhetjük el.

Adja ki a **cat /proc/version** parancsot! A kernel a fájlrendszerben ezen a ponton teszi közzé a verzióját, ebből dolgozik a korábban használt **uname** parancs is.

Adja ki a **cat /proc/cmdline** parancsot! Itt láthatja, hogy a **grub** a kernel betöltésekor milyen paramétereket adott át az induláshoz: a **BOOT_IMAGE** változóban szerepel a kernel fájl neve, a **root** paraméter a felcsatolandó fájlrendszer UUID-jét (Universally Unique Identifier) tartalmazza, illetve néhány további paraméter is látható itt.

9. kérdés

Készítsen képet a **cat /proc/version** és a **cat /proc/cmdline** két parancs kimenetéről!

```
hallgato@~$ cat /proc/version
Linux version 4.15.0-45-generic (buildd@lgw01-amd64-031) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #48-Ubuntu SMP Tue Jan 29 16:28:13 UTC 2019
hallgato@~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-4.15.0-45-generic root=UUID=a1bbd164-ef39-4702-945a-2851496c555c ro quiet splash
hallgato@~$
```

Ehhez a válaszhoz kép csatolása szükséges.

3.2. Memóriahasználát

Adja ki a `cat /proc/meminfo` parancsot! A kimenetben a memória és a swap állapotával kapcsolatos minden fontos állapotinformáció megtalálható. Mivel ezekre az esetek többségében nincs mind szükség, ezért a `free` nevet viselő program jobban formázva és csak a lényegét emeli ki (adja ki a `free` parancsot is)!

10. kérdés

Készítsen képet a `cat /proc/meminfo` és a `free` parancs kimenetéről!

```
Active: 1004292 kB
Inactive: 443224 kB
Active(anon): 870448 kB
Inactive(anon): 310372 kB
Active(file): 193844 kB
Inactive(file): 132852 kB
Unevictable: 32 kB
Mlocked: 32 kB
SwapTotal: 969960 kB
SwapFree: 922856 kB
Dirty: 100 kB
Writeback: 0 kB
AnonPages: 1132064 kB
Mapped: 277504 kB
Shmem: grub 48020 kB
Slab: 98468 kB
SReclaimable: 48500 kB
SUnreclaim: 49968 kB
KernelStack: 11212 kB
PageTables: 37004 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
WritebackTmp: 0 kB
CommitLimit: 1978600 kB
Committed_AS: 4307344 kB
VmallocTotal: 34359738367 kB
VmallocUsed: 0 kB
VmallocChunk: 0 kB
HardwareCorrupted: 0 kB
AnonHugePages: 0 kB
ShmemHugePages: 0 kB
ShmemPmdMapped: 0 kB
CmaTotal: 0 kB
CmaFree: 0 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize: 2048 kB
DirectMap4k: 169856 kB
DirectMap2M: 1927168 kB
DirectMap1G: 0 kB

hallgato@~$ free
total        used        free      shared  buff/cache   available
Mem:    2017280    1433000    160996    48020    423284    380544
Swap:   969960      47104    922856
```

Ehhez a válaszhoz kép csatolása szükséges.

A `free` parancs kimenetében feltűnhet, hogy nem csak egy számot kaptunk (a szabad memória méretét), hanem több paraméterünk is van, amik nem triviálisak. Nézzük meg ezeket alaposabban:

- Total: a fizikai memória teljes mérete

- Used: a folyamatok által lefoglalt és használt memória mennyisége
- Free: szabad (sem mire nem használt) memória
- Shared: a folyamatok között megosztott memóriaterület
- Buff/cache: a lemez gyorsítótárazására használt terület
- Available: a szükség esetén felszabadítható memória mértéke

Könnyű belátni, hogy minden olyan memória, amit a rendszer sem mire nem használ, a felhasználó szempontjából veszteség: hiszen ha a szabad memóriát bármilyen gyorsítótárazásra fel lehet használni, akkor érdemes is erre használni, mivel így a felhasználó számára a rendszer gyorsul anélkül, hogy bármelyik futó folyamat ennek kárát látná. Ezért van az, hogy a rendszer igyekszik a lemez gyorsítótárazásával (a legutóbb a lemezzől beolvasott fájlok memóriában tartásával) gyorsítani a működést. Ha a programok memóriát igényelnek a kerneltől, akkor először a Free keretből kapnak, majd szükség esetén a buffereket felszabadítva az Available értékig adható további memória.

A **Swap** soron látható a lemezen elérhető memória cserehely mérete és kihasználtsága. Adja ki a **swapon --summary** parancsot, mellyel listázható a lemezen erre a célra allokalált helyek listája: alapértelmezetten a rendszer jelenleg a /swapfile fájlt használja, de allokalhatnánk erre a célra további fájlokat és külön lemezzartíciókat is.

A kernel a régen használt memórialapokat hajlamos a lemezre menteni annak érdekében, hogy az éppen futó és újonnan induló programok számára mindig kellő mennyiségű szabad memória álljon rendelkezésre a gyors működés érdekében. Azt, hogy a kernel a memória hány százalékos kihasználtsága felett kezdi el használni a swap területet egy swappiness nevű kernel paraméter határozza meg. Értéke a **cat /proc/sys/vm/swappiness** parancs kiadásával kérhető le!

11. kérdés

Készítsen képet a **cat /proc/sys/vm/swappiness** előző parancs kimenetéről!

```
hallgato@ [redacted] :~$ cat /proc/sys/vm/swappiness
60
hallgato@ [redacted] :~$
```

Ehhez a válaszhoz kép csatolása szükséges.

Az alapértelmezett 60-as érték azt jelenti, hogy amíg a fizikai memória 60%-a szabad, addig a kernel egyáltalán nem nyúl a lemezhez. Ha a szabad memória mértéke kevesebb mint 60%, akkor kezdi el a legrégebben használt lapokat a lemezre mozgatni.

3.3. Lemezkezelés

A rendszer által érzékelt lemezek listázásához adjuk ki a **sudo fdisk -l** parancsot! A rendszer a virtuális gépben beállított egyetlen lemezt /dev/sda néven ismeri fel, melyen egyetlen partíció található /dev/sda1 néven.

A lemezek és a partíciók elnevezése nem véletlenül hasonlít a fájlnevek elérési újtára: a /dev (virtuális, a kernel által karbantartott) könyvtárban valóban létezik sda és sda1 nevű fájl, pontosabban device node, ami a fájlrendszeren keresztül elérhető így.

Az egyes partíciókon található fájlrendszerek helye a "/" fájlrendszer gyökér alatt attól függ, hogy azokat hova kötjük be a mount parancs segítségével. A **mount** parancsot kiadva láthatjuk az éppen felcsatolt összes fájlrendszert és azok csatolási helyét (mount point-ját)! A /dev/sda1 sort kikeresve látjuk, hogy az a fájlrendszer gyökérkönyvtára "/", a proc nevű virtuális fájlrendszer a /proc könyvtár alatt érhető el és így tovább.

A rendszer indulásakor felcsatolt fájlrendszerek a /etc/fstab fájlban találhatóak, ennek tartalmát a **cat /etc/fstab** paranccsal írathatjuk ki. Mindössze két releváns sort látunk majd: a /dev/sda1 (mely itt UUID-jével szerepel) kerül felcsatolására a "/" gyökér alá, illetve a /swapfile nem kerül sehova csatolásra ("none"), mert a típusa swap.

12. kérdés

Készítsen képet a **cat /etc/fstab** kimenetéről!

```
hallgato@ [redacted] :~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=a1bbd164-ef39-4702-945a-2851496c555c / ext4 errors=remount
-ro 0 1
/swapfile none swap sw
0 0
hallgato@ [redacted] :~$
```

Ehhez a válaszhoz kép csatolása szükséges.

Mivel a kernel számára nincs különbség a /dev alatti device node-ok és a fájlrendszer tetszőleges más pontjai között, ezért semmi akadály, hogy egy partíciót vagy akár teljes lemezt fájlba másoljunk vagy onnan visszaállítsunk a sztenderd **cp** parancs segítségével (vagy a kicsit jobban paraméterezhető ezért jobb teljesítményű **dd** parancs használatával). Sőt, akár tetszőleges fájl használhatunk fizikai lemez helyett, a rendszer számára ez se jelent különbséget.

Ez utóbbi lehetőségre egy játékos példa:

- Hozzunk létre egy üres, csak nullákat tartalmazó 10 MB méretű fájlt a következő parancs kiadásával: **dd if=/dev/zero of=/tmp/disk-file bs=1M count=10**
- Ennek hatására létrejött egy 10 MB méretű disk-file névre hallgató fájl a /tmp könyvtárban, amit ellenőrizhetünk is: **ls -lh /tmp/disk-file**
- Hozzunk létre EXT4 fájlrendszert a fájlon belül pont úgy, mintha egy lemezre mutató device node lenne: **mkfs.ext4 /tmp/disk-file**
- A rendszer tartalmaz egy /mnt üres könyvtárat a fájlrendszer gyökerében. Csatoljuk fel az előbb létrehozott fájlrendszert oda: **sudo mount /tmp/disk-file /mnt**
- A felcsatolt fájlrendszereket és az azokon elérhető szabad helyet a következő parancs jeleníti meg: **df -h**
 - Jól láthatjuk, hogy a /mnt alatt egy majdnem 9 MB-os kötet van felcsatolva, ahol még közel 8 MB szabad helyünk van.

13. kérdés

Készítsen egy képet a `df -h` kimenetéről a disk-file felcsatolása után!

```
hallgato@~:~$ df -h
Fájlrendszer  Méret  Fogl.  Szab.  Fo.%  Csatol. pont
udev          954M   0      954M   0%    /dev
tmpfs         197M   1,3M   196M   1%    /run
/dev/sda1     20G    7,7G   11G    42%   /
tmpfs         985M   888K   985M   1%    /dev/shm
tmpfs         5,0M   4,0K   5,0M   1%    /run/lock
tmpfs         985M   0      985M   0%    /sys/fs/cgroup
tmpfs         197M   28K    197M   1%    /run/user/1000
/dev/sr0      1,9G   1,9G   0      100%  /media/hallgato/Linux Mint 19.1 Cinnamon 6
4-bit
/dev/loop0    8,7M   172K   7,9M   3%    /mnt
hallgato@~:~$
```

Ehhez a válaszhoz kép csatolása szükséges.

3.4. A /dev fájlrendszer

A /dev könyvtár alatt található a kernel által létrehozott device-node-ok. Szinte minden eszköz itt látható, ami a számítógéphez kapcsolódik, így a lemezek (pl. /dev/sda), a partíciók (pl. /dev/sda1), a fizikai terminálok (tty*), stb. Van itt néhány speciális elem is:

- Az előbbi feladatban használtuk a disk-file csupa nullával való kitöltésére azt, hogy a /dev/zero-ból tetszőleges számú nulla érték kimásolható anélkül, hogy valaha is a fájl végére érnénk.
- A /dev/random hasonlóan használható, csak nulla helyett pszeudovéletlen értékeket ad vissza, így például ha egy fájlt véglegesen és visszaállíthatatlanul törölni szeretnénk a lemezeiről, akkor először érdemes a /dev/random-ból származó értékekkel felülírni - így a tartalma jó eséllyel sose lesz már visszaállítható ellentétben az egyszerű törléssel, amikor a fájl tartalmának bájtjai a lemezen maradnak, csak a fájlrendszer rájuk mutató inode-ját töröljük.
- A /dev/null egy végtelen nyelő, vagyis tetszőleges program kimenetét ideirányítva az nyom nélkül és a lemezen hely foglalás nélkül eltűnik, ami hasznos lehet ha például a konzolra feleslegesen író alkalmazás kimenetétől szeretnénk megszabadulni.

3.5. A /proc fájlrendszer

Minden futó folyamathoz részletes információk érhetők el a /proc virtuális fájlrendszer alatt. Próbaként indítsuk el a következő parancsot: `tail -f /var/log/syslog`

A `tail` parancs az `-f` kapcsolóval folyamatosan olvassa a paraméterként megadott rendszernapló fájlt, és amint egy új sor kerül annak a végére, megjeleníti azt. (A programból majd a Ctrl+C kombinációval tudunk kilépni, de egyelőre hagyjuk futni még!)

Mivel a futó `tail` program elfoglalja a meglévő terminálunkat indítson egy második terminál alkalmazást a rendszer menüből, és adja ki a következő parancsot: `ps axw | grep tail`

A **ps** parancs listázza az összes futó folyamatot, a kimenetét pedig a csővezetéken (pipe |) keresztül megkapja a grep, ami csak azokat a sorokat mutatja, melyek tartalmazzák a paraméterként kapott "tail" szót. A kimenetben keresse meg a "tail -f /var/log/syslog" végződésű sort, és az első oszlopból olvassa le a folyamat azonosítóját (Process ID vagy röviden PID). A továbbiakban a parancsokon belül **{PID}** formában hivatkozunk majd az itt leolvasott számra, vagyis minden **{PID}** helyére helyettesítse majd be azt!

Először lépünk át megfelelő könyvtárba a **cd /proc/{PID}** parancs kiadásával! A rengeteg elérhető fájl közül néhány érdekesebb:

- **cat limits**: a folyamatra érvényes korlátozások listázása (maximális CPU idő, egyszerre nyitott tartható fájlleírók, stb.)
- **cat status**: a folyamat részletes állapot információja (memória adatok, umask, swap használat, stb.)
- **cat maps**: a folyamat által használt programkönyvtárak a megosztott memóriában
- **ls -l root**: a folyamat által használt fájlrendszer gyökérkönyvtárára mutató szimbolikus link (jelen esetben ez a valós fájlrendszer gyökérkönyvtár a "/" lesz, de biztonsági megfontolásból egy folyamat bezárható egy tetszőleges mappába is)
- **ls -l fd**: a folyamat által nyitott fájlleírók szimbolikus link formájában

14. kérdés

Készítsen képet az **ls -fd** kimenetéről!

```
összesen 0
lrwx----- 1 hallgato hallgato 64 febr 19 15:56 0 -> /dev/pts/1
lrwx----- 1 hallgato hallgato 64 febr 19 15:56 1 -> /dev/pts/1
lrwx----- 1 hallgato hallgato 64 febr 19 15:56 2 -> /dev/pts/1
lr-x----- 1 hallgato hallgato 64 febr 19 15:56 3 -> /var/log/syslog
lr-x----- 1 hallgato hallgato 64 febr 19 15:56 4 -> anon_inode:inotify
hallgato@ [redacted] :/proc/5805$
```

Ehhez a válaszhoz kép csatolása szükséges.

Egy pillanatra érdemes alaposabban szemügyre venni a nyitott fájl leírókat! Még a legegyszerűbb folyamat is legalább három nyitott fájl leíróval rendelkezik:

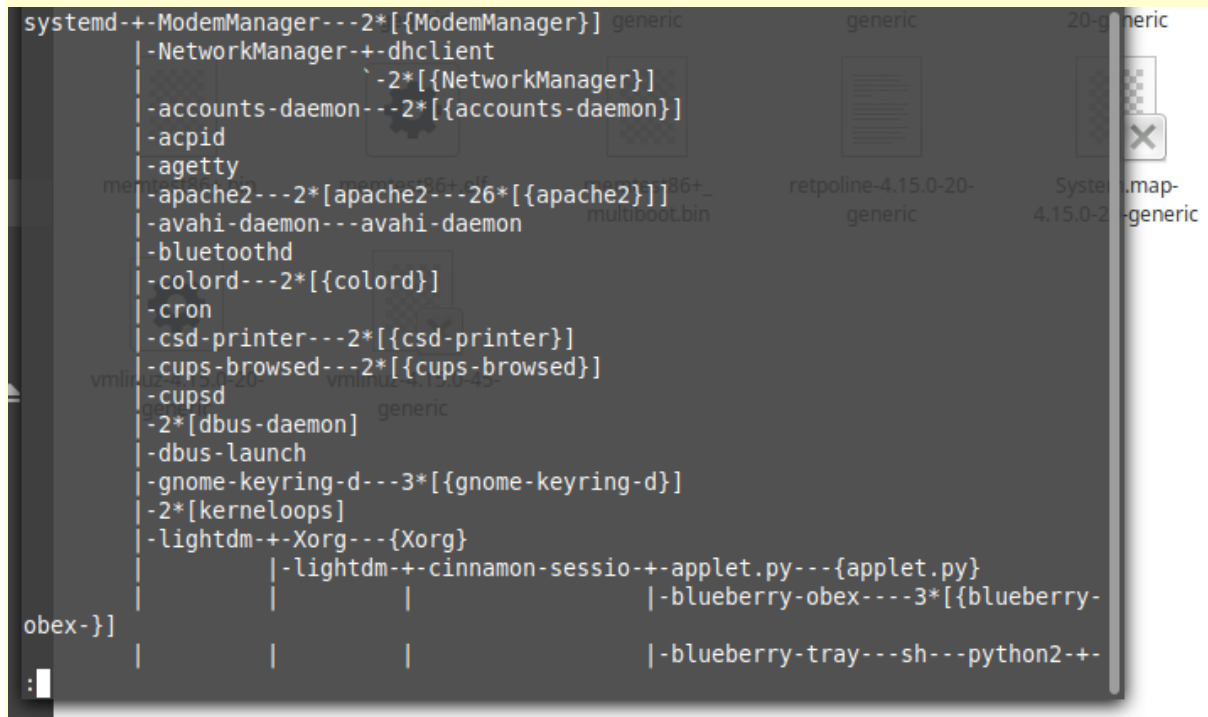
- 0 = standard input - ez minden alkalmazás alapértelmezett konzolos bemenete, innen olvas a C gets() függvénye, a Java System.in objektuma vagy a Python sys.stdin objektuma,
- 1 = standard output - ez minden alkalmazás alapértelmezett szöveges kimenete, ide ír a C printf() függvénye, a Java System.out objektuma vagy Python sys.stdout objektuma,
- 2 = standard error - erre a kimenetre küldik a programok a futás közbeni hibaüzeneteket, például a különböző kezeletlen kivételekről itt jelennek meg üzenetek.

A fentiek felül a futó **tail** programunk nyitva tartja még a /var/log/syslog fájlt 3-as sorszámmal (hiszen innen olvassa az új sorokat), és van egy 4-es sorszámu inode:inotify leírója is, melyen keresztül az operációs rendszertől kap jelzéseket arról, hogy új adat került a syslog fájl végére.

A /proc fájlrendszeren található adatokból dolgozik a **pstree** parancs, mely a folyamatokat leszármazás alapján fa struktúrában ábrázolja. Adja ki a **pstree | less** parancsot!

15. kérdés

Készítsen képet a pstree | less kimenetéről!



```
systemd+-+ModemManager--2*[{ModemManager}] generic
|-NetworkManager+-+dhclient
|   |-2*[{NetworkManager}]
|-accounts-daemon--2*[{accounts-daemon}]
|-acpid
|-agetty
|-apache2--2*[apache2--26*[{apache2}]]
|-avahi-daemon--avahi-daemon
|-bluetoothd
|-colord--2*[{colord}]
|-cron
|-csd-printer--2*[{csd-printer}]
|-cups-browsed--2*[{cups-browsed}]
|-cupsd
|-2*[dbus-daemon]
|-dbus-launch
|-gnome-keyring-d--3*[{gnome-keyring-d}]
|-2*[kerneloops]
|-lightdm+-+Xorg---{Xorg}
|   |-lightdm+-+cinnamon-sessio+-+applet.py---{applet.py}
|   |-blueberry-obex----3*[{blueberry-
obex-}]
|   |-blueberry-tray---sh---python2+-+
```

Ehhez a válaszhoz kép csatolása szükséges.

A kirajzolt fa struktúrát a **less** program bemenetére irányítottuk a | karakter használatával, így a fel-le gombokkal mozoghat benne, kilépni pedig a **q** lenyomásával tud majd!

Jól látható, hogy minden folyamat őse a **systemd**: ez indul el először és ennek a PID-je az 1-es minden esetben. A grafikus környezet úgy indult el, hogy a **systemd** elindította a **lightdm** ablakkezelőt, ami elindította az **Xorg** névre hallgató X szervert, és megjelenítette a **lightdm** bejelentkező képernyőjét.

A sikeres belépés után indult el a **cinnamon-session**, ami a Cinnamon ablakkezelőt indította el, és ez az az asztali felület amit látunk magunk előtt bejelentkezés után. Minden itt látható program részletes adatai elérhetők a PID-jük kikeresése után a /proc fájlrendszeren.

4. Gondolkodtató kérdések* (csak a kiválóan megfelelt értékeléshez)

Az alábbi kérdésekre csak akkor kell válaszolnia, ha a laborra kiválóan megfelelt értékelést szeretne. A normál megfelelt értékeléshez elég eddig a pontig eljutnia! A kérdések megválaszolásához először próbáljon önállóan gondolkodni, de ezt követően nem tilos internetes kereséssel keresni a választ! Minden esetben gondolja át, értse meg, és saját szavaival megfogalmazva indokolja a választ!

*4.1. Stdin, Stdout, Stderr

Ahogy korábban láttuk, a folyamatoknak alapértelmezetten mindig van legalább három nyitott fájl leírójuk: a Standard Input, Standard Output és a Standard Error. Ezek közül az első kettő érthető (mindig kell valahonnan tudnia olvasni a programnak, illetve írnia valahova), azonban a harmadik nem ennyire triviális. Ha konzolon futtatjuk a programot, akkor a Standard Error ugyanazon a konzolon jelenik meg, mint a Standard Output.

16. kérdés

Mi értelme van a Standard Error kimenet létezésének? Miért nem írhatók a hibák egyszerűen a Standard Output-ra?

Stdin - ez a fájlkezelő, amit a folyamat olvas, hogy információt kapjon tőled

Stdout - a folyamat normál információt ír a fájlkezelőre

Stderr - a folyamat hibajelentéseket ír be a fájlkezelőbe

A hibaüzenet parancssorból történő futtatás esetén úgy tűnik, mintha az STDOUT-ra menne, azonban a gép egy külön i/o csatornát tart fenn a hibaüzenetek számára, ez az STDERR

*4.2. Lemezek azonosítása rendszerben

Az **fdisk** parancs kimeneténél és a `/dev` könyvtárban is azt láttuk, hogy a lemezeket `/dev/sdX` ($X=A,B,\dots$), a partíciókat `/dev/sdXY` ($Y=1,2,\dots$) formában azonosította a rendszer. A lemezek nevében az első két "sd" betű azt jelzi, hogy ezek IDE/SATA/SCSI lemezek, helyettük például virtualizált lemez esetén "vd" állna. A sorszám az adott lemez sorszáma a rendszerben.

Amikor a rendszer által automatikusan felcsatolt fájlrendszereket néztük a **cat /etc/fstab** kimenetében, akkor az újabb rendszereken mégsem a `/dev/sdX` formában hivatkozunk a lemezre és a fájlrendszerre, hanem azoknak az UUID-jét használjuk.

17. kérdés

Miért UUID-vel hivatkozunk a lemezekre és nem /dev/sd* formában a /etc/fstab fájlban?

Egyértelmű azonosítás miatt.

*4.3. Memória és swap

A **free** parancs kimenetében a Mem soron 6 oszlopban van érték, a Swap soron viszont csak háromban. Miért nem értelmezhetőek vagy relevánsak a hiányzó oszlopok a Swap esetében?

18. kérdés

Miért nem relevánsak bizonyos oszlopok a free kimenetében a swap-re nézve?

Mert a Swap soron a lemezen elérhető memória cserehely mérete és kihasználtsága látható. Ami shared vagy buff/cache vagy available az nem cseretárhely.