



Linux Alapok

Készítette: Kun Attila

Tartalomjegyzék

0. Bevezető.....	3
1. Az operációs rendszerek csoportosítása.....	3
2. Linux alapfogalmak.....	4
3. Telepítés, bootloader.....	7
4. Csomagkezelés.....	7
5. Rendszer felépítése.....	8
6. Alapvető parancsok.....	13
7. Grafikus felületek.....	16
8. Terminálok, parancssor, alias, előzmények.....	19
9. Könyvtárszerkezet, csatolások (mount).....	21
10. Felhasználókezelés.....	22
11. Felhasználókezelés – Rendszergazda.....	26
12. Csomagkezelés, processzkezelés.....	27
13. Fájlrendszer, jogosultságok.....	31
14. Ha akarsz a Linuxszal még foglalkozni.....	33

Források: Órai jegyzetek, illetve az alábbi linkek:

[https://m-info.hu/sites/default/files/oktatas/LINUX_OS_alapok_dia_\(2017\).pdf](https://m-info.hu/sites/default/files/oktatas/LINUX_OS_alapok_dia_(2017).pdf) (mint váza a doksinak)

<https://wiki.hup.hu>

https://hu.wikipedia.org/wiki/File_Allocation_Table

<https://hu.wikipedia.org/wiki/Rendszermag>

<http://linuxkezdoknek.hu/>

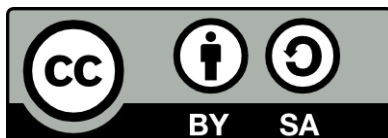
<https://www.tldp.org/LDP/lame/LAME/linux-admin-made-easy/shadow-file-formats.html>

<https://szit.hu> -ról egy-két téma (amit tartalmazott a pdf)

http://szabadszoftver.kormany.hu/wp-content/uploads/sysadmin_I.pdf

Minden forrás nagyrészt GPL vagy CC-BY-SA licenccel rendelkezik, így ezt a pdf-et a CC-BY-SA módon adom tovább. Lehet természetesen tovább fejleszteni ;)

A dokumentum LibreOffice 6.2-ben készült.



0. Bevezető

Úgy gondoltam, hogy kéne már egy normális Linuxos útmutató azok számára, akik vagy középiskolában tanulják informatika szakon, vagy éppen most kezdik el használni, és nem értik csínját-bínját a rendszernek. Ez a dokumentum nagyjából az alapokat lefedi, ami a Linuxos rendszereket illeti. Természetesen biztos, hogy van olyan dolog, ami más-más disztribúciónál máshogy fordul elő pl. fájlok esetében, de a legtöbbször a valóságot tükrözi. A sok forrás azt jelzi, hogy nem csak egy forrásból kell kiindulni, hanem nagyon is bele kell mélyedni, hogy tisztán lássuk a dolgokat, utána kell nézni, hogy mennyire tükrözi a valóságot az adott dolog (akár éles/virtuális rendszer alatt is kipróbálni az adott dolgot, és ha lehet máshol is nézzük meg, hogy tényleg így van-e). Nagy köszönet thottee kollégának a PenguinPit közösségből, aki sokat segített a dokumentum létrehozásában.

1. Az operációs rendszerek csoportosítása

Az operációs rendszer lehet:

- ▶ felülete lehet grafikus (GUI) vagy parancssoros (CLI)
- ▶ betöltött szerepe szerint: szerver, desktop
- ▶ felhasználók száma szerint: egyfelhasználós, többfelhasználós
- ▶ folyamatkezelés szerint: single task, multi task

előnye a karakteresnek: gyors, rugalmas parancsbevitel, automatizálás (szkriptek végrehajtása), a grafikus magot nem kell kihasználnia, így bármilyen gépen jól működik

- ▶ **single task**: 1 feladat / program tud futni (felhasználói)
- ▶ **multi task**: több felhasználói program is tud futni
 - preemptív: az op. rendszer elvehet erőforrásokat
 - kooperatív: csak akkor működik, ha mindenki közreműködik
- ▶ **egy/többfelhasználós**: lehet több felhasználói fiókot csinálni, de régebben bárki garázdálkodhatott a számítógépben, magyarul nem volt privát szférája senkinek. Mostanában már nem lehet hozzáférni mások könyvtárához, de ezt a fájlrendszernek kell támogatnia (FAT16, FAT32 így kizárt, viszont az EXT2,3,4; NTFS támogatja). Ami még fontos, hogy hálózaton keresztül egyidőben több felhasználó is használhatja a rendszert.

Ezek alapján az OS-ek csoportosítása:

DOS, illetve a ráalapuló Windows-ok (1.01-től ME-ig bezárólag): karakteres(a Windows az egy grafikus felület itt), egyfelhasználós (lehet azt mondjátok, hogy lehet már itt több fiókot létrehozni, igen, de itt csak egy felhasználóval lehet adott időben belépve lenni). Szerver verziója nincsen a DOS-ban található biztonsági rések miatt. Fizetni kellett érte.

Windows NT-szerű rendszerek (Windows NT 3.x – 10): grafikus, többfelhasználós (itt már a felhasználók mappája titkosítva van, így nem férhetsz hozzá mások mappájához a megfelelő

jogosultság nélkül), van szerver és asztali verzió is (pl. a Home és a Pro verziók), fizetős, mint a DOS-ra alapuló Windowsok voltak.

Linux: karakteres, többfelhasználós, azaz több felhasználó lehet a hálózaton egyszerre, a mappák titkosíthatóak, jogosultságok beállíthatóak, szabadon testreszabható (programkínálat, illetve felület terén). Ingyenes, nem kell fizetned érte (a legtöbb Linux-alapú rendszerre, kivételt képez a Red Hat, amiért igen is kell fizetni). Szerver és asztali felhasználásra szánt verzió is van.

Összefoglaló kérdések:

1. Milyen lehet egy operációs rendszer?
2. Mi a karakteres felület előnyei?
3. Mi az, hogy többfelhasználós? Hogyan működik?
4. A Linux mitől tér el a többi rendszertől?

2. Linux alapfogalmak

Miért Linux? Linus Torvalds megmutatta, hogy a MINIX nem elérhető az x86-os architektúrára, így átírta a x86-os gépekre a kernelt.

1991-ben Linus Torvalds, másodéves hallgatója volt a Helsinki Egyetem számítástechnikai tudományok karának. Linus autodidakta hacker volt, saját operációs rendszert szeretett volna írni. Elhatározta, hogy ír egyet. Hogy miért? A 80386-os processzor védett módú, feladat-váltó lehetőségeit szerette volna felfedezni. Ez körülbelül 1991 nyarának elején lehetett.

GPL A GNU General Public License egy széles körben használt szabad szoftver licenc, melyet eredetileg Richard Stallman írt a GNU Project részére. A GNU GPL célja, hogy garantálja a szabad szoftver másolásának és terjesztésének szabadságát, ezáltal biztosítva a szoftver szabad felhasználhatóságát minden felhasználó számára.

A Linuxot körbevették shellekkel (héj) - parancsértelmezőkkel, és rálehet építeni az X Window rendszert is

Disztribúciók = Kiadások

Legfontosabb disztribúciók:

1. Red Hat Enterprise Linux

A Red Hat Enterprise Linux az egyik legnagyobb sikerű Linux disztribúció, amit a Red Hat fejleszt kereskedelmi felhasználásra, hozzáköthető például az RPM csomagkezelő rendszer.

2. Slackware

A legrégebbi létező Linux disztribúció. Az első kiadása 1993-ban jelent meg, az akkoriban elterjedt SLS Linux-ot alapul véve, akkor még hajlékonylemezekon. A kialakítása során elsődleges szempont volt az egyszerű használhatóság, a stabilitás, és a biztonság. Nincs automatikus függőség kezelése. Kibocsátása óta célkitűzése a leg-Unix-szerűbb disztribúcióvá válni.

3. Debian

Debian az egyik legrégebbi disztribúció, 1993. augusztus 13-án jelentette be Ian Murdock. A Debian a Red Hathez képest nem kereskedelmi célú, önkéntesek vesznek benne részt.

Nagyon sok disztribúciónak az alapja (pl. Ubuntu, Knoppix). A Debian a Debora (Ian felesége) és Ian nevéből származik.

4. Ubuntu

Egyik legnépszerűbb Debian-alapú disztribúció, amit a Canonical Ltd. fejleszt. 2004-ben jelent meg az első kiadása, a 4.10 és azóta 6 havonta jön ki újabb, illetve minden páros év 4. hónapjában egy hosszan támogatott (Long-Term Support) verziója¹.

5. Arch Linux

Az Arch Linux egy függetlenül disztribúció haladóbb Linux felhasználóknak. Pacman csomagkezelőt használ. Rolling-release modellje van, ami azt jelenti, hogy folyamatosan frissül a nagyobb kiadások helyett.

6. openSUSE

Az openSUSE a nagy múltú SUSE Linux folytatása, és egy olyan projekt, ami arra alapul, hogy hirdessék a Linux használatát, ezzel biztosítva egy ingyenes, teljes körű disztribúciót a felhasználóknak.

7. Gentoo

A Gentoo Linux egy sokoldalú, gyors disztribúció, ami Portage csomagkezelőt használ. A többi disztribúcióhoz képest az egész rendszert forrásból kell fordítani, így kicsit időigényes lehet a telepítés (akár órákat is igénybe vehet rendszertől függően).

Grafikus felületek:

- ▶ KDE, Gnome, MATE, Xfce, Cinnamon, LXDE, stb.
- ▶ Asztali menedzselők: KDM(utódja az SDDM), GDM, LightDM

Linux kabalája Tux, a pingvin, amit Alan Cox tervezett és Larry Ewing rajzolt.
(<https://www.maketecheasier.com/the-origin-of-the-penguin-tux/>)



RPM és DEB-alapú rendszerek

RPM: Red Hat, Fedora, CentOS, openSUSE stb.

DEB: Debian, Ubuntu, Linux Mint, MX Linux stb.

repository: tárolók; programok, driverek találhatók a disztribúcióhoz

csomagkezelő: a függőségi fát megoldja, ha inkompatibilitási gond van

Magyar disztrók:

- ▶ UHU-Linux
- ▶ BlackPanther OS
- ▶ Magyarcs (bár még most csak egy Arch Linux telepítő magyarul, mégis már tekinthető disztribúciónak)

¹ Jelenleg 18.04-es a legfrissebb LTS disztribúció.



ubuntu



debian



gentoo linux™



UHU-Linux



redhat.

Összefoglaló kérdések:

1. Mikor és miért jött létre a Linux?
2. Mi a GPL?
3. Melyik disztribúció fizetős?
4. Melyik disztribúció használja a Pacman-t?
5. Melyik disztribúció fordítja a programokat forrásból?
6. Mik a repositoryk?
7. Milyen DEB-re alapuló disztribúciókat ismersz?
8. Milyen magyar disztrók léteznek?

3. Telepítés, bootloader

- ▶ hagyományos telepítés
- ▶ grafikus vagy CLI

telepítés lehet CD, DVD, USB-ről, vagy hálózatról is

Lehet telepíteni virtuális gépként is (VMware, Virtualbox)

Bootloaderek (rendszer betöltők): régebben LILO, ma GRUB, boot paramétereket lehet megadni, hogy melyik rendszer induljon, akár memória tesztet is lehet végrehajtani, illetve hardver tesztet, ami megnézi, hogy a hardvereknek hibája van-e



Hova telepítsük?

A bootloadert MBR esetén magára a merevlemezre lehet telepíteni pl. /dev/sda (első merevlemez), GPT esetén meg meglehet adni melyik partíció legyen a „BIOS boot” (pl. /dev/sda3 – első winchester 3. partíciója), így oda fog kerülni a bootloader, ahová bejelöltük.

A GRUB képes mindenféle rendszert bebootolni, de a Windows telepítése a Linux telepítését törli, még a GRUB-ot is (illetve, ha a Windows 10 teljes frissítést csinál a gépen, a bootloadert felülírja)

Az X szerver leképzi a felülről érkező kéréseket a videokártyának, szolgáltatásokat nyújt a grafikus ablakkezelő részére.

Összefoglaló kérdések:

1. Honnan telepíthetjük a rendszert?
2. Mik a bootloaderek, mire használhatóak?
3. Hova tudjuk telepíteni a bootloadert?
4. Miért nem jó, ha Windows-zal használjuk együtt a Linuxot?

4. Csomagkezelés

Csomagok az összes programot összefoglalják, ami Linuxos programot használ.

`sudo apt update` – csomaglista frissítése Ubuntu alatt

hozzáadhatóak repositoryk a csomaglistához

dpkg helyett legtöbbször apt-t használunk már Debian-alapú Linux alatt.

program / csomag telepítése:

`apt install <program> / dpkg -i <csomag>`

dpkg-nál a csomagoknál nem írja a függőséget, külön kell felrakni azt, ami nincs meg

a dpkg, rpm, apt, apt-get, stb. parancsoknak van man-juk (man = manual, használati útmutató)

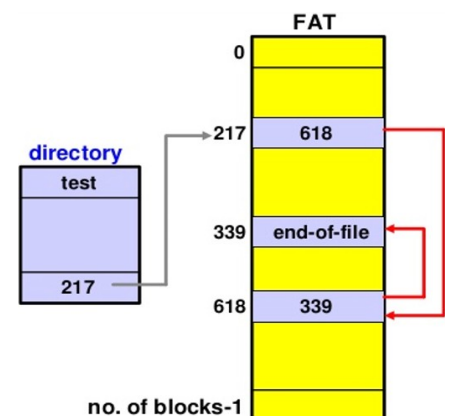
5. Rendszer felépítése

- ▶ Alapvetően konzolos
 - DOS is hasonló volt
 - Windows parancssor
- ▶ Van grafikus felület (GUI)

Fájlrendszer

- ▶ Nincs betűjelzés, mint a DOS/Windows esetén → Minden a / alatt van
- ▶ Csatolja a külső tárolókat (mount)
 - Ma már a Windows is tudja
 - az /etc/fstab állomány a partíciók automatikus csatolási pontjait és a hozzá tartozó beállításait tartalmazza.
 - fdisk vagy fdisk -l (partíciók kilistázása)
 - a /mnt alatt vannak a felcsatolt eszközök (bár ez nem annyira igaz, mert bárhol lehetnek felcsatolva, a legtöbb mai disztribúcióban a gyakran automatikusan felcsatolásra kerülő eszközök (pl. pendrive) a /media-ba kerülnek, de az igaz, hogy a /mnt-t azért hozták létre, hogy oda lehessen felcsatolni partíciókat)

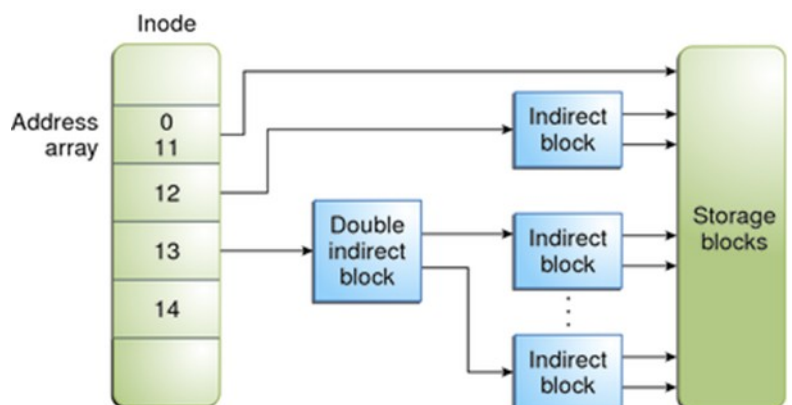
A **FAT**-tábla bejegyzéseket tartalmaz minden cluster-hez a tárolólemez egy folyamatos területén. Minden bejegyzés tartalmazza vagy a következő cluster számát egy fájlban, vagy fájlvége, nem használt lemez terület ill. a lemez speciálisan fenntartott területének jelzését. **Csak soft-linket tud.**



EXT-fájlrendszer: Inode-táblát használ.

Az inode-ban van a fájl leírása és az adatblokkok helye, a fájl neve nélkül. A fájlhoz pontosan egy inode tartozik. Nagyobb méretű fájl esetén akár több katalógusbejegyzés is tartozhat, ami hordozza a fájl nevét. Az inode tartalmazza **a nevet, leírást, létrehozás időpontját, jogosultságokat,**

tulajdonost(UID), csoportot(GID), és azt is, hogy mennyi hivatkozás van rá. (a FAT-tal szemben sokkal többre képes) **Soft-linket és hard-linket is tud, ami az egyik legnagyobb előnye a ext fájlrendszernek a FAT-tal szemben.**



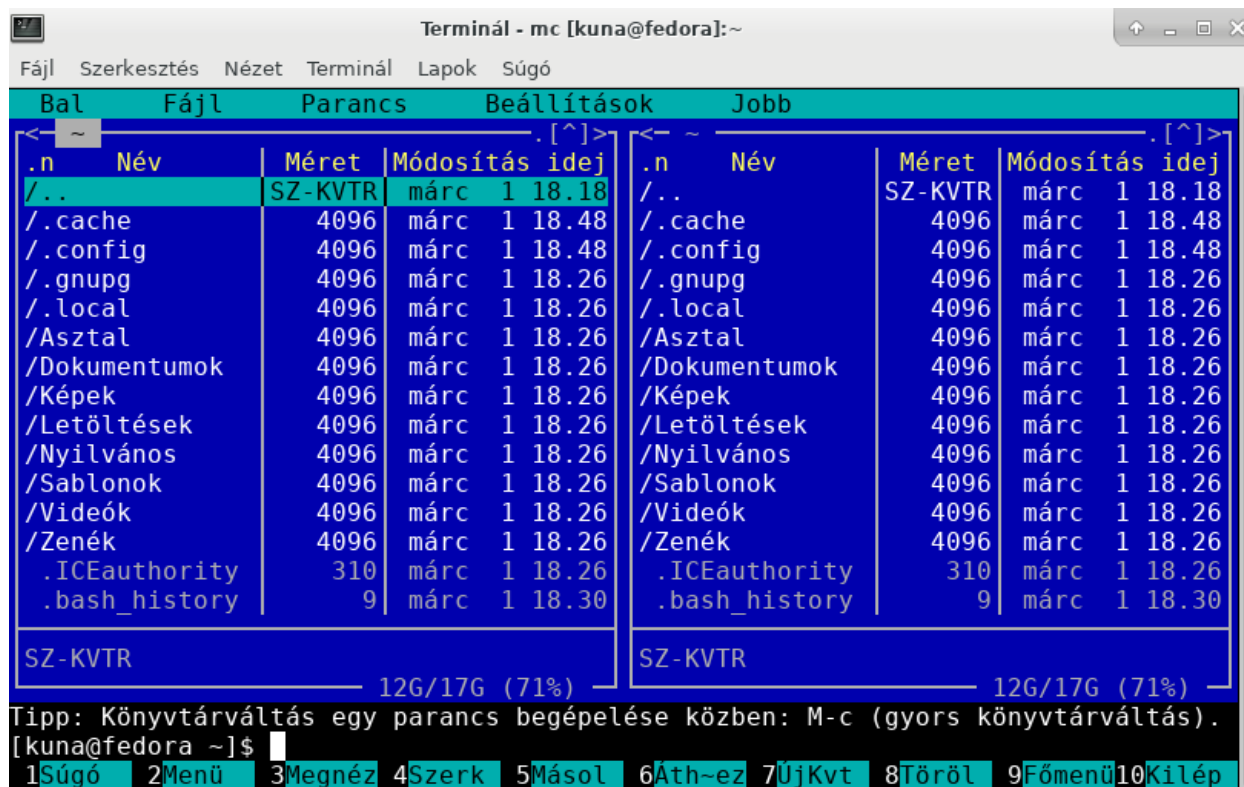
- ▶ Soft-link: parancsikon, ha törlődik az eredeti, akkor törlődik, vagy éppen hibát ír ki, hogy nem találja

- Hard-link: A fájlnek egy másolata, de ha az eredetit változtatjuk, akkor a másolat tartalma is változik, viszont csak akkor tűnik el véglegesen a fájl, ha az eredetin kívül az összes másolat törlődik.

Fájlrendszer tulajdonságai

- Az engedélyeket minden fájlhoz külön csatolja
- A hozzáférés a saját tulajdonosára vagy csoportokra korlátozza
- olvasás, írás, végrehajtási engedélyek

Fájlkezelők



- mc (Midnight Commander) - parancssoros
- Gnome Commander - grafikus
- illetve a felületekhez járó fájlkezelők: thunar, nautilus, rox-filer, dolphin, pcmanfm, stb. (<https://www.tecmint.com/top-best-lightweight-linux-file-managers/>)
- **Fájlkezelés:** fájl műveletek

Kernel = rendszermag, az OS „lelke”

perifériák kezelése, fájlrendszer kezelése, könyvtárstruktúra megvalósítása, processzor irányítása

típusai:

Monolitikus: A rendszermag egyetlen, megbonthatatlan egységet képező modulból áll. Régebbi Linuxokra (2.0-nál régebbi) és UNIX rendszerekre jellemző.

hibrid: pl.: Windows – jogosultságok beállíthatók. A hibrid kernelek alapjában véve olyan mikrokernelek amelyekben néhány „nem létfontosságú” kódrészletet átmozgattak a felhasználói

szintről (userspace) a kernel szintre (kernel space) azért, hogy az kevesebb absztrakciót használva, gyorsabban fusson.

Mikrokernel: Nem egyetlen részben van megoldva a kernel, hanem több apró programban.

Moduláris: A moduláris kernel a monolitikus kernel problémáit igyekszik orvosolni. A kernel bizonyos részeit - fizika és logikai – és az ebből keletkezett modulokat, amik főképpen meghajtó programok, a memóriába be lehet tölteni. A kernel méretét spórolja ugyan, de a kettő együtt egyre és egyre nagyobb (mérete az relatív, attól függ, mennyi drivert szeretnél hozzáadni, általában 100 MB körüliek a komplett kernelek, de ha minimalista vagy, akkor lehet akár 10 MB alatt is). A mai Linux rendszereket már ide soroljuk be (pontosabban monolitikus moduláris a Linux kernel). A kernel modulok a /lib/modules/kernel-verzió mappába találhatók.

A kernel helye: A /boot könyvtárban van, van a vmlinuzhoz és az initrd-hez soft-link a gyökérben

Fájlrendszerek

- ▶ A fájlrendszereknél hatalmas szerepe van a tömbösítésnek Linuxnál
- ▶ Virtuális fájlrendszerek (a rendszer számára)
 - SMBFS, SSHFS, NFS, stb.
- ▶ támogatott fájlrendszerek
 - ext2, ext3, ext4 (Linux saját fájlrendszere)
 - ReiserFS, Btrfs, ZFS, XFS
 - FAT32, NTFS
 - stb.

Könyvtárszerkezet

- ▶ Hierarchikus
 - könyvtárak
 - fájlok

(egyszintű könyvtár: Commodore mágneslemez; mágnesszalag)

- ▶ Elérési út
 - Root gyökér: /
 - pl: /home/joe/myfile.txt
- ▶ Csatolt (mount) tárolóeszközök
 - pl. /mnt/usb
- ▶ Csatolás: `mount /dev/sdc -t ISO9660 /mnt/cdrom` (ha CD-olvasót akarunk használni)
`mount /dev/sdb -ta /mnt/[név]` - a -ta jelző az automatikus felismerést jelenti
az sda,sdb, sdc... a merevlemezeket jelölik, a legtöbb modern disztró automatikusan a /media/cdrom-ba csatolja

- ▶ lecsatlakoztatás: `umount /dev/sdb`

Főbb mappák

- ▶ **/bin, /sbin**
 - futtatható, bináris állományok (az `sbin`-es programokat csak rendszergazda futtathatja), nem szükséges fájlkiterjesztés
- ▶ **/boot**
 - rendszer betöltésének állományai: GRUB, LILO, kernel (`/boot/vmlinuz`)
- ▶ **/cdrom, /media, /mnt**
 - média becsatolása: USB, Floppy, CD/DVD
- ▶ **/dev**
 - az eszközöket (mindet) fájlkon keresztül érjük el
pl: `/dev/dsp` (hangkártya) `/dev/cdrom` (CD)
- ▶ **/etc**
 - konfigurációs fájlok
 - rendszergazda éri el
 - globális konfigurációs hely
- ▶ **/home/**
 - felhasználók mappája
 - felhasználó fájlljai
 - beállítások, konfigurációs fájlok, mappák
 - pl.: `/home/peter`
`/home/peter/dokumentumok`
- ▶ **/lib**
 - megosztott objektumok, libraryk
 - mint windows esetén a DLL-ek
 - valamint a kernel moduljai
- ▶ **/lost+found**
 - sérült fájlok
 - helyreállíthatók
- ▶ **/proc**
 - az éppenfutó műveletek – fájlként sorszámozva, illetve információk a rendszerről (processzor, memória stb.)
 - példaparancs (processzorinfó): `cat /proc/cpuinfo`
- ▶ **/tmp**

- ideiglenes fájlok helye
- rendszer és felhasználó szinten

► /usr

- felhasználói fájlok helye
- telepített programok
- library-k
- források (kernel): /usr/src
- dokumentációk: /usr/share/doc, /usr/doc
- ikonok, és így tovább.

► /var

- szolgáltatások gyűjtőkönyvtára
- átmeneti tárolók (cache): /var/cache
- naplófájlok (log): /var/log/maillog
- levelezési fiókok (mailbox): /var/mail/root
- www könyvtár: az apache alapértelmezett weblap könyvtára (/var/www)

Felhasználók

- rendszergazda: **root**
- felhasználók: a „root” felhasználó hozza létre alapvetően
- csoportokba szervezhetők a felhasználók
- jogosultságok adhatók a fájlrendszeren
 - pl: `/home/joe/myfile.txt 26 joe joe -rwxrwxrwx`
- Az első felhasználó is root jogokkal rendelkezik (akit a telepítéskor megadtál) – de ez disztribúció függő, pl. Debian esetén a felhasználónak nincsen root joga, és sudo sincsen.

Összefoglaló kérdések(4-5. fejezet):

1. Milyen csomagkezelő parancsot használunk Debian-szerű rendszerek alatt?
2. Mit tartalmaz a /etc/fstab?
3. Mi a INODE/EXT-fájlrendszer előnye a FAT-táblával szemben?
4. A Linux hová sorolható be kernel tekintetében?
5. Hogyan csatolhatunk és csatolhatunk le egy eszközt/partíciót?
6. Mit tartalmaz a /etc mappa?
7. Mi található a /proc-ban?
8. A /usr mappába mik vannak?
9. Mi az alapértelmezett felhasználó neve?

6. Alapvető parancsok

a promptban lehetsz rendszergazda

- `[root@ubuntu ~]#` root jogosultság
- `[user@ubuntu ~]$` felhasználói jogosultság

Parancsok:

- ▶ `<parancs> --help`: kiírja az adott parancsra a kapcsolókat
- ▶ `cd`: change directory – könyvtár változtatása (→ `cd ..` = egy könyvtárral visszalépés)
- ▶ `pwd`: print working directory – kiírja, hogy melyik mappába tartózkodsz
- ▶ `w`: megmutatja, hogy ki van bejelentkezve, mit csinál (`w <felhasználónév>` esetén kiírja az adott felhasználó tevékenységét)
- ▶ `who`: ez csak a dátumot és az IP-t írja ki
- ▶ `ls`: fájlok kilistázása
 - `ls -a`: kilistázza a rejtett állományokat
 - `ls -A`: hiányzik a `.` és a `..` (. a jelenlegi mappára utal, a `..` pedig az előző könyvtárra)
 - `ls -l`: részletes könyvtárlista
 - `ls -al`: a fentebbi parancsok alapján: részletes könyvtárlista + rejtett fájlok + `.` és `..`
- ▶ `exit`: a shell-be bejelentkezett felhasználót kilépteti, és a shellt zárja be
- ▶ `sudo <parancs>`: rendszergazdaként hajtható végre a parancs
- ▶ `mv`: move, mozgatás A-ból B-be: szintaxis: `mv <forrás> <cél>`, akár átnevezésre is használható.
- ▶ `mc`: midnight commander parancsa
- ▶ `touch` fájl: fájl nevű állomány létrehozása
- ▶ `echo` valami: a beadott „valami” adatot az echo saját standard kimenetére irányítja.
- ▶ `nano` fájl: fájl nevű állomány szerkesztése
- ▶ `cat` fájl: fájl nevű állomány kiírása, amiből ki is lép
- ▶ `less` fájl: fájl nevű állomány kiírása, de lehet benne lépkedni, q-val lehet a szöveges megtekintőből kilépni
- ▶ `rm`: remove, törlés
 - `rm -r`: rekurzív, azaz mindenre hat a törlés
 - `rm -f`: force, azaz erőltetett, argumentumoktól függetlenül törlődik a fájl/mappa
 - `rmdir`: mappa törlése, bár az `rm -rf` egyszerűbb parancs rá.
- ▶ `mount`: eszköz csatolása (pl. `mount /dev/sdb -ta /mnt/usb`)
- ▶ `umount`: eszköz lecsatolása (pl. `umount /mnt/usb`)
- ▶ `ln`: link/hivatkozás készítése (hard-link)
 - `ln -s`: soft-link/symlink készítése (a különbség a soft- és hard-link között az 5. fejezetben le van írva)

- ▶ df = disk free = melyik partíció mennyi szabad hely van
- ▶ du = disk usage = lemezhasználat; meg kell adni paramétereket, mert az összes könyvtárat fogja különben kilistázni
 - `du /home` esetén a /home könyvtárnak a tartalmát írja ki (mármint hogy mennyi helyet foglal a vinszeszteren egyes mappái)
 - `du -h` kiírja, hogy mennyi kB, MB, GB-ot foglal (ha nem rakjuk ezt ide akkor bájtba írja ki)
 - `du -s` szummáz (összes mappát egybe veszi egyként kiírja hogy mekkora helyet foglal)
- ▶ free = szabad memória és szabad swap hely
- ▶ top (fekete-fehér) / htop (színes) = kiírja az aktuálisan futó appokat (mint Windowsnál a Feladatkezelő)

uname = rendszerről ír ki infót, -a kapcsolóval részletesen ír ki a kernellel kapcsolatba infót

```
Linux fedora 4.20.12-200.fc29.x86_64 #1 SMP Mon Feb 25 16:16:18 UTC 2019
x86_64 x86_64 x86_64 GNU/Linux
```

pl. ezt írja ki Fedora 29 alatt, fedora a gépnév ebben az esetben

- ▶ uptime = mióta megy a rendszer, és mekkora terheltséggel
- ▶ netstat = hálózati statisztika
- ▶ man = A parancs manual oldalait nyitja meg, rövid, tömör, célratoró leírás
- ▶ info = ugyanaz, mint a man, csak Red Hat-es rendszereknél jellemzőbb

Hálózatkezelés

- ▶ ifconfig = ip cím, hálózati beállítás, küldött, fogadott csomagok mennyisége
- ▶ iwconfig = ugyanaz mint az ifconfig, csak itt vezeték nélküli hálózatok beállítására van
- ▶ ping (egy ip-cím/weboldal megpingelésére) : ping www.google.hu ; ping -c 3 www.google.hu (a -c kapcsoló után meglehet adni hogy hányszor küldjön csomagot az adott oldanak)
- ▶ ifup = interface up – scriptet futtat le mikor aktiválódik pl. ifup eth0 = eth0 interface felkapcsolódik)
- ▶ ifdown = interface down – ugyanaz mint az ifup, csak itt az interfacet lekapcsoljuk, pl. ifdown eth0
- ▶ ifstatus - interface állapota
- ▶ ifstat – monitorozza az interface-ek letöltési/feltöltési sebességét

Felhasználókezelés

- ▶ adduser/useradd – felhasználó létrehozása
- ▶ userdel – felhasználó törlése

- ▶ `chmod` - a fájlok-, könyvtárakra vonatkozó jogokat állíthatjuk be (részletesen a 29-30. oldalon van róla több infó)
- ▶ `chown` - tulajdonos módosítása
- ▶ `usermod` - felhasználó módosítása

csomagkezelési parancsok: `dpkg`, `rpm`, `apt`, `yum`, `dnf` stb.

egyéb parancsok:

- ▶ `clear` - terminált lepucolja
- ▶ `chroot` - root könyvtár módosítása
- ▶ `kill` - folyamat kilövése PID alapján (`ps -A` kiírja azt)
 - `kill -9` - minden alszálat megöl
 - `killall` - listának az összes elemét, de a `kill -9 @(...)` egyszerűbb
- ▶ `halt` - a gép megáll, de a tápja megy (ATX v2-es tápoknál)
- ▶ `shutdown` - a gép teljesen leáll - `sudo init 0`
- ▶ `reboot` - a gép újraindul - `sudo init 6`
- ▶ `/bin/bash` - parancsértelmező
- ▶ `service` - gépen futó szolgáltatásokat lehet vele indítani/újraindítani
- ▶ `systemctl` - ugyanaz mint a `service`
- ▶ `cron`: a gép nélküled dolgokat csinál (időközönként, indításonként, leállításonként), a `cron` daemon futtatja ezeket (óránként, naponta, hetente...)
- ▶ `crontab`: a felhasználó sajátja
 - `crontab -e`: `crontab` szerkesztése

`/etc/skel` mappa: új felhasználónak a home könyvtárába kerülnek a mappában levő fájlok (`skel` = skeleton, csontváz)

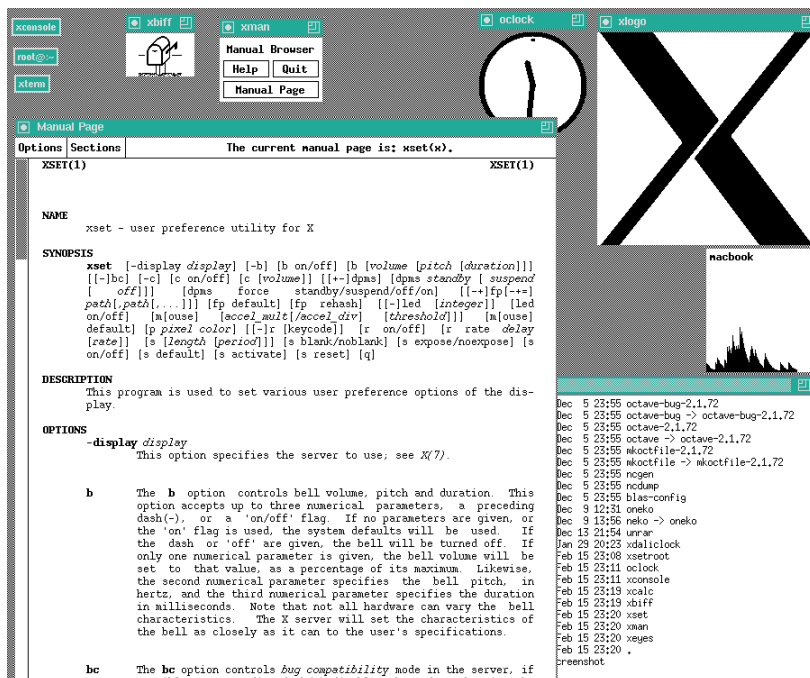
`/etc/sudoers` fájl: azok a felhasználók, akik tudják a `sudo` parancsot használni. (13. fejezetben részletesebben elmagyarázva)

Összefoglaló kérdések:

1. Melyik parancs mondja meg, hogy melyik mappába vagy?
2. Mire jó a `w` parancs?
3. A `htop` parancs mit mutat meg?
4. Hogy lehet egy mappát teljesen kitörölni minden tartalmával együtt?
5. A du mit jelenít meg? Melyik kapcsolóval tudjuk elérni azt, hogy normálisan, ne bájtbá jelenítse meg az adatmennyiséget?
6. Hogyan tudsz egy hálózati interface-t felkapcsolni?
7. Melyik paranccsal tudod lekérdezni egy parancs/programnak az összes kapcsolóját(illetve részletes útmutatóját)?
8. Melyik paranccsal tudod a jogosultságokat megváltoztatni?
9. Hogyan tudod a gépet leállítani? (legalább két parancs!)
10. Milyen paranccsal tudsz felhasználót létrehozni, és honnan szedi a plusz fájlokat az új home mappába alapértelmezetten?
11. Melyik paranccsal tudsz egy adott folyamatot kilőni minden más alfolyamatával együtt?

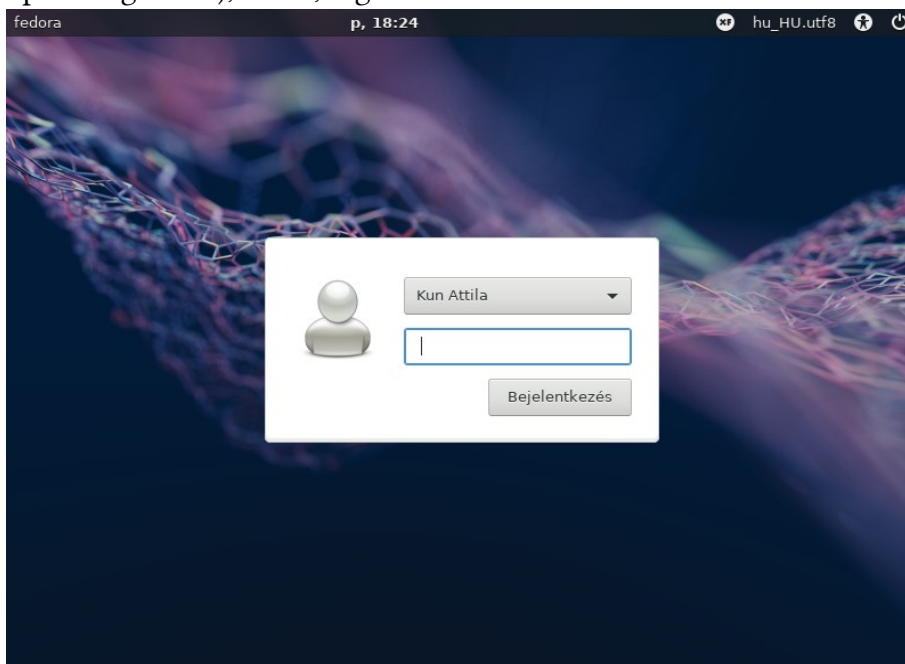
7. Grafikus felületek

Linuxon a grafikus megjelenítést az X Window System biztosítja. Az **X Window System** helyett néha azt mondjuk egyszerűen, hogy **X**, vagy **X11**. Az X11 az utolsó verzió alapján kapta a nevét. Az X11 az utolsó verzió alapján kapta a nevét. Az X Window System kezeli a grafikus megjelenítést, a billentyűzetet és az egeret. Viszont nem nyújt semmilyen grafikus felületet.



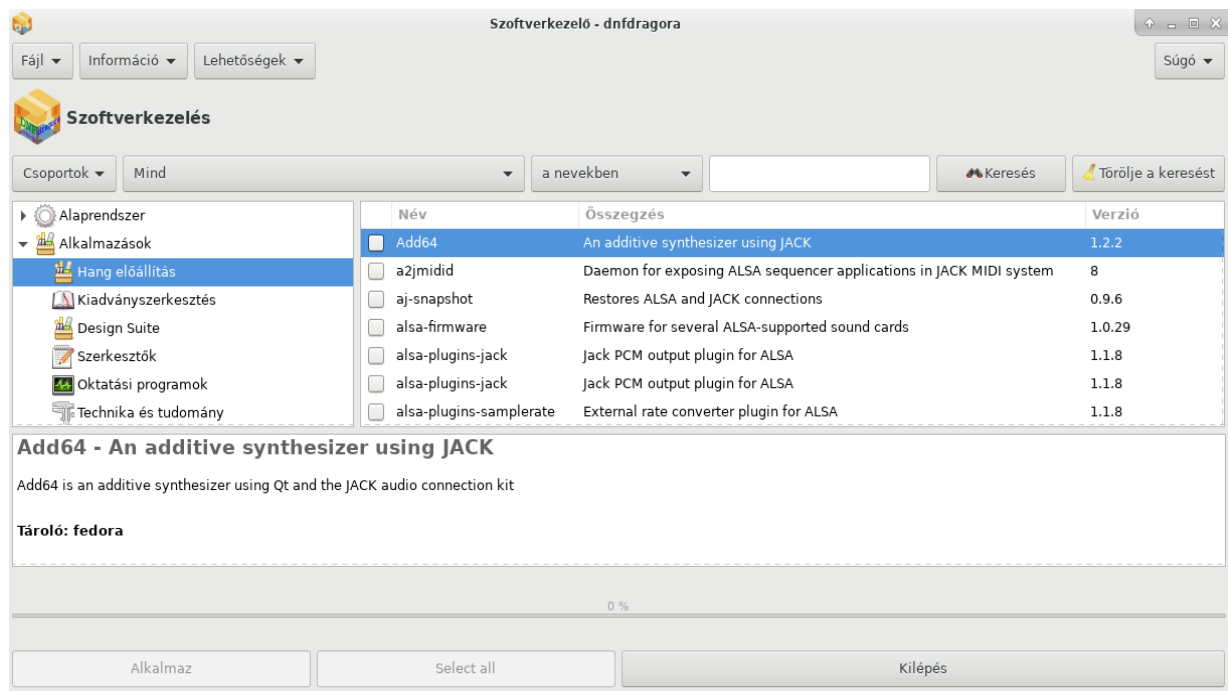
Az X felülete

Bejelentkezés lehet grafikus felületen is:
XDM(X Desktop Management), GDM, LightDM stb.



Fedora bejelentkező felülete LightDM-mel

Minden egyes operációs rendszernek megvan a saját maga grafikus csomagkezelője, Debian-alapú rendszerek alatt megtalálható a Synaptic csomagkezelő, Fedora alatt a dnfdragora, Arch alatt a Pamac vagy az Octopi, stb.



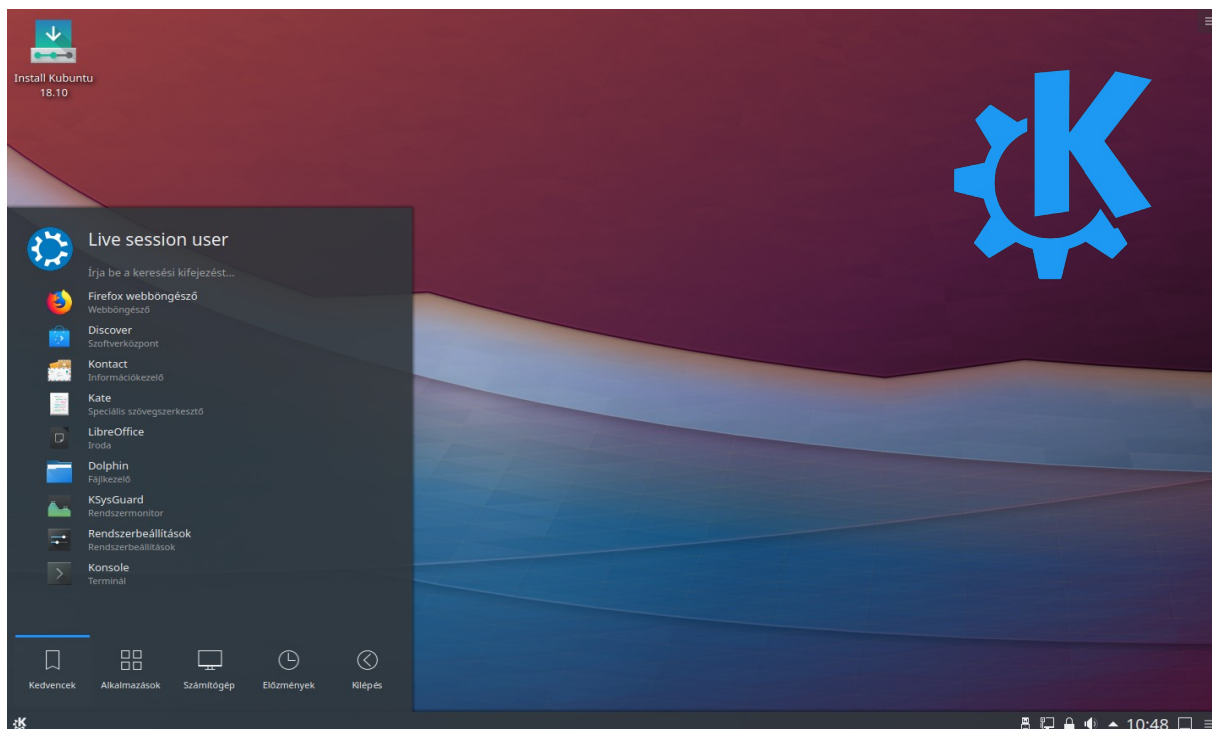
Fedora alatt

Asztali felületek (Desktop Environment) + logóik:

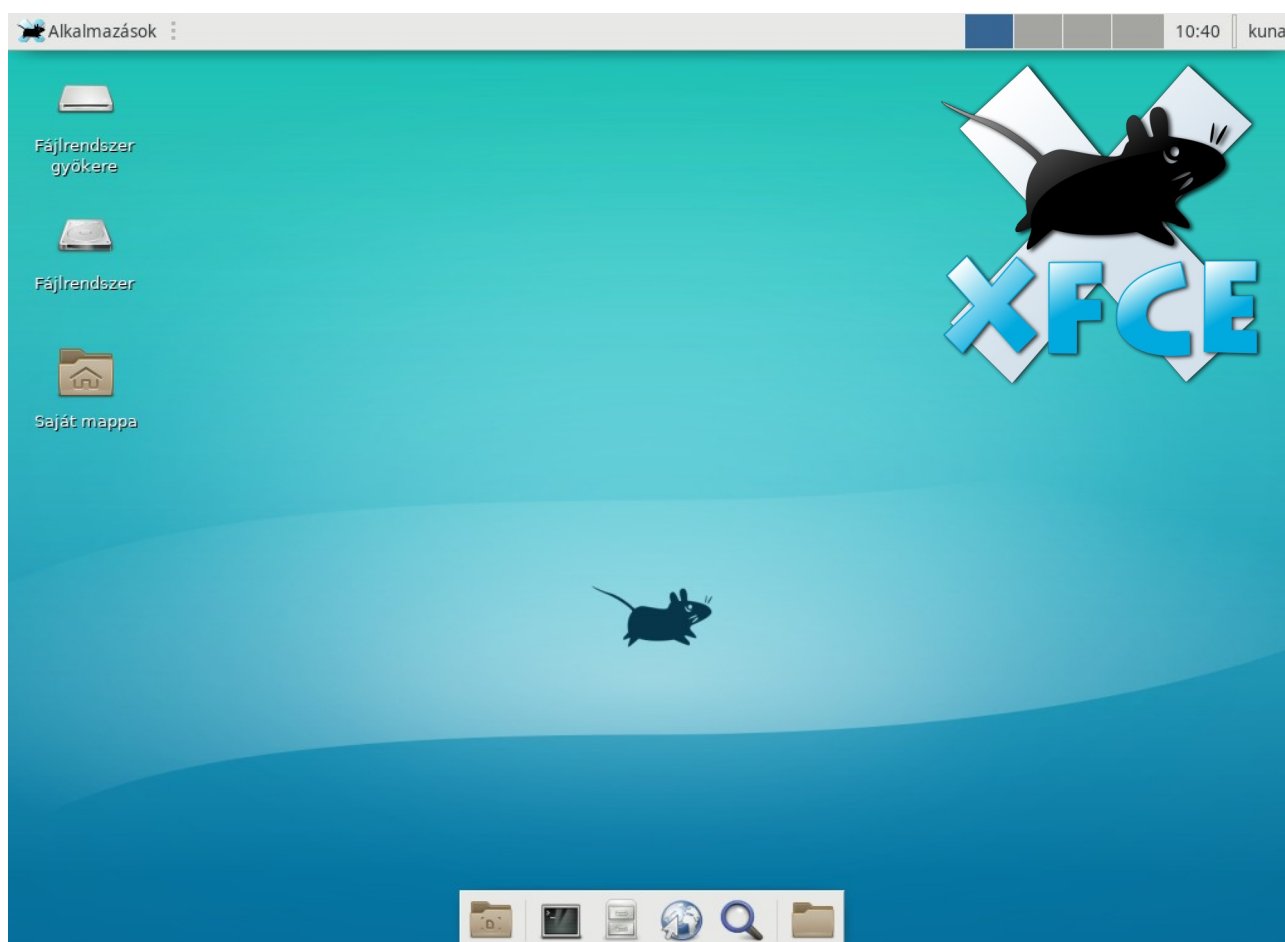


Gnome 3.32 felület (alkalmazások menü)

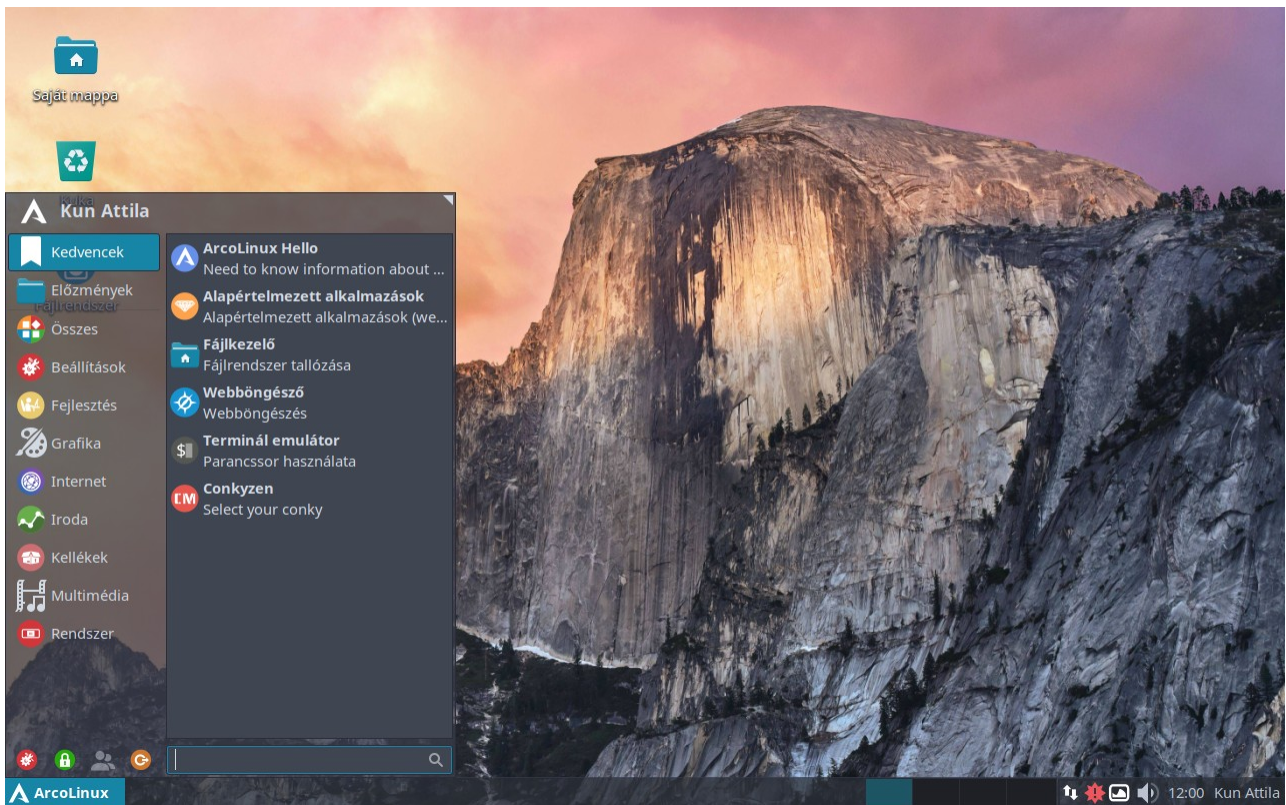




KDE Plasma 5-ösnek az asztala (Kubuntu 18.10 alatt)



XFCE 4.12-es asztal (alap kinézete)



XFCE 4.12-es felület témázva ArcoLinux alatt

Van még a Cinnamon, meg a MATE felület, amik még említésre méltók, mindkettő Gnome-származék. (https://wiki.archlinux.org/index.php/desktop_environment - összes jelenleg elérhető asztali felület) A te döntésed, melyiket szeretnéd telepíteni Linux alá, a testreszabhatóság határa pedig a csillagos ég.

Kompozitáló ablakkezelők pl.: Compiz, Compton, Xfwm, Marco, KWin.

Az ablakkezelő lényegében az ablak műveletekért felel (nagyít, kicsinyít, mozgat), a kompozitor az effektusokért, pl. kicsinyítéskor elhalványulva tűnik el, stb.

Működés közben váltható az ablakkezelő és a felület (bár utóbbinál először fel kell rakni, és utána kilépni a jelenlegi környezetből).

Összefoglaló kérdések:

1. Mi az X11? Mire való?
2. Milyen grafikus bejelentkezési lehetőség van?
3. Debian alapú rendszerek alatt milyen grafikus csomagkezelő van, illetve Arch alatt?
4. Az XFCE alatt hol helyezkedik el a tálca/panel?
5. Mire jó a kompozitáló ablakkezelő?

8. Terminálok, parancssor, alias, előzmények

- ▶ Terminál = parancssor → shell = parancsértelmező
- ▶ Szöveges üzemmódú, DOS-szerű
- ▶ különböző terminálok vannak

- Gnome-terminál, Xfce4-terminal, xterm, stb.
- ▶ Sok parancsértelmező van (pl **bash**, fish, ksh, zsh stb.)
- lehet a terminálokat csinosítani, erre példa Linux Mind Zoli videója (<https://www.youtube.com/watch?v=gkZaukcebG0>) – zsh telepítése
- ▶ parancsok adhatóak ki

parancsok listája: <http://letix.hu/> <http://szit.hu/doku.php?id=oktatas:linux>

- ▶ alias: álnév
- ▶ alias: egy adott álnévvel tudunk rövidíteni egy-egy összetettebb parancsot vagy saját parancsot hozhatunk létre.
- ▶ Definiálás helye (bash shell esetén): ~/.bashrc
- ▶ Globális alias-ok: /etc/bashrc

Előzmények

- ▶ Fel, le nyilakkal újra lehet előhívni
- ▶ lapozni a terminálon a ctrl+page up, ctrl+page down billentyűkkel lehet
- ▶ a ~/.bash_history a bejelentkezés előtti parancsokat tartalmazza

Környezeti változók: az operációs rendszerről vagy programjainkról tárolnak információkat. A környezeti változók a rendszer indulása után automatikusan beállításra kerülnek.

A PATH az egy környezeti változó ami megmondja a shellnek, hogy hol keresse a futtatandó fájlokat.

- ▶ Listázásuk: `printenv`, `set`
- ▶ lehet egyedileg lekérdezni: `printenv PATH`
- ▶ beállításuk: `set VAR1="valami"` a VAR1 az egy tetszőleges változó
- ▶ megszüntetésük: `unset var1`
- ▶ megjelenítésük: `echo $VAR1`
- ▶ hogy a beállítás megmaradjon: `export VAR1="valami"`
- ▶ saját parancsot lehet a /home/bin mappa alá csinálni, persze csak úgy működik, ha a mappát a PATH környezeti változóhoz hozzáadtuk: `export PATH=$PATH:~/bin`, és hogy a .bashrc-be fixáljuk: `echo "export PATH=$PATH:~/bin" >> ~/.bashrc`
thottee elmagyarázta ezt a weboldalán részletesebben, hogy mit csinálunk konkrétan: http://linuxkezdoknek.hu/articles.php?article_id=33 (a cikk második fele)

Súgó: help, segítség a parancs használatához

- ▶ parancs: man, info
- használata: `man <kapcsolók> <parancs>`
`info <kapcsolók> <parancs>`


```
Terminál - kuna@sparky: ~
Fájl Szerkesztés Nézet Terminál Lapok Súgó

TOP(1)                                User Commands                                TOP(1)

NAME
    top - display Linux processes

SYNOPSIS
    top -hv|-bcEHiOss1 -d secs -n max -u|U user -p pid -o fld -w
    [cols]

    The traditional switches '-' and whitespace are optional.

DESCRIPTION
    The top program provides a dynamic real-time view of a running
    system. It can display system summary information as well as a
    list of processes or threads currently being managed by the Linux
    kernel. The types of system summary information shown and the
    types, order and size of information displayed for processes are
    all user configurable and that configuration can be made persis-
    tent across restarts.

    The program provides a limited interactive interface for process
    manipulation as well as a much more extensive interface for per-
    sonal configuration -- encompassing every aspect of its opera-
    Manual page top(1) line 1 (press h for help or q to quit)
```

Összefoglaló kérdések:

1. Mi a shell?
2. Milyen parancsértelmezőket ismersz?
3. Mi az alias?
4. Hol tárolódnak a saját parancsértelmeződ beállításai (bash shell esetén)?
5. a ~/bash_history mit tartalmaz?
6. Mi a környezeti változó? Hogyan lehet beállítani?
7. Hogy lehet egy fájlba fixálni a környezeti változó tartalmát?

9. Könyvtárszerkezet, csatolások (mount)

► könyvtárszerkezet: fájlok csatolására kialakított struktúra.

► Archiválás, tömörítés a tar programmal

● Listázás

`tar -tvf file.tar`: megmutatja a mentes.tar tartalmát

`tar -tvfz file.tar.gz`: megmutatja a mentes.tar.gz tartalmát

`tar -tvfj file.tar.bz2`: megmutatja a mentes.tar.bz2 tartalmát

`tar -tvvf mentes.tar`: részletes listázás, jogok, tulajdonos etc.

● Betömörítés

`tar -cvf file.tar /eleresi/ut`: az elérési útvonal alatt levő adat tar-ba tömörítése.

`tar -cvfz file.tar.gz /eleresi/ut`: -"- gzip-be tömörítése.

`tar -cvfj file.tar.bz2 /eleresi/ut`: -"- bz2-be tömörítése.

● Kitömörítés

`tar -xvf file.tar`: Kitmöríti a tar-t és egy /file mappába teszi

`tar -xvzf file.tar.gz`: Kitömöríti a gz-t, majd a tar-t, és egy /file mappába teszi

`tar -xvfj file.bz2`: bzip2-t tömöríti ki

10. Felhasználókezelés

- Funkciók: felhasználó hozzáadása, törlése, jelszóadás, módosítás, stb.

- Felhasználó hozzáadása: `adduser`, `useradd`

Ubuntu alatt az `adduser`:

```
root@ubuntu:/home/kuna# adduser valaki
"valaki" felhasználó létrehozása...
"valaki" (1001) új csoport létrehozása...
"valaki" (1001) felhasználó létrehozása "valaki" csoporttal...
A(z) '/home/valaki' saját könyvtár létrehozása...
Fájlok másolása innen: '/etc/skel'
Adja meg az új UNIX jelszót:
Írja be újra a UNIX jelszót:
passwd: password updated successfully
Changing the user information for valaki
Enter the new value, or press ENTER for the default
    Full Name []: valaki
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Az adatok helyesek? [I/n] i
root@ubuntu:/home/kuna# _
```

- a `useradd` parancs:

`useradd [opciók] felhasználónév`

opciók:

- `-u szám` : felhasználói azonosító megadása (UID)
- `-c`: megjegyzés
- `-g csoport`: csoport tagság beállítása
- `-d /home/user`: A felhasználó HOME mappáját állítjuk be `/home/user` -re.
- `-s /bin/bash`: Alapértelmezésként a bash shell-t kapja bejelentkezéskor.
- `-G csoport1, csoport2`: vesszőkkel elválasztva, ha egyéb csoportnak is tagja a felhasználó
- `-m -k /home/letozo <felhasználónév>`: egy létező felhasználó mintájára építi fel az új user home-ját.
- A `-m` kapcsoló a home könyvtár létrehozására utal, a `-k` pedig hogy honnan másoljuk az alapértelmezett fájlokat (pl. `/etc/skel`)

így a parancs pl:

```
Useradd -c "teszt fiók" -u 1001 -g users -G info,rgazda -d /home/teszt -s /bin/bash -m -k /etc/skel teszt
```

- Felhasználó információi: `getent`

`getent passwd <felhasználónév>`

A informatika csoportról szeretnénk informálódni: `getent group informatika`

- Felhasználói adatok helye: `/etc/passwd`

► az /etc/passwd fájl tartalma:

```
janos:x:1000:1000:Nagy János:/home/janos:/bin/bash
```

1. Felhasználónév.
2. Titkosított jelszó.
3. Felhasználói azonosító szám (uid).
4. Csoportazonosító (gid).
5. Teljes név, illetve egyéb leírás.
6. Home könyvtár.
7. Milyen shell fusson bejelentkezéskor.

► A /etc/shadow fájl tartalma például:

```
smithj:Ep6mckr0LChF.:10063:0:99999:7:::
```

A mezők kettőspontontként:

1. Felhasználónév
2. Titkosított jelszó (legalább 8-12 karakter hosszú)
3. Utolsó jelszómódosítás 1970. január 01-je óta napokban számolva
4. Minimum ennyi napnak kell elteltie az újabb jelszóváltoztatásig (a 0 azt jelzi, hogy bármikor meglehet változtatni)
5. Maximum ennyi ideig érvényes a jelszó (a 99 999 azt jelzi, hogy a felhasználó változatlanul hagyhatja a jelszót évekig)
6. Figyelmeztetés ennyi nappal a jelszó lejáratára előtt
7. Ha lejár a jelszó, a felhasználó nem léphet be
8. 1970. január 01. után ennyi nappal járt le a jelszó
9. Egy fenntartott mező jövőbeli használatra

► Felhasználói szabályok módosítása: chage

A chage parancs segítségével szabályozhatjuk egy-egy felhasználónak mikor jár le a jelszava.

Mikor jár le joska jelszava? (chage -l joska példakimenete):

```
Utolsó jelszóváltás          : febr 28, 2019
Jelszó lejár                 : soha
Jelszó inaktív               : soha
Hozzáférés lejár            : soha
A jelszómódosítások közti legkevesebb nap: : 0
A jelszómódosítások közti legtöbb nap:    : 99999
A jelszó lejáratára előtt figyelmeztetés napok száma : 7
```

99 999 nap körülbelül 273 év

- ▶ A joska felhasználó jelszava járjon le 10 nap múlva
`chage -M 10 joska`
- ▶ Mikor járjon le a jelszava?
`chage -E "2019-12-31" joska`
- ▶ Inaktív napok után lezárás
- ▶ joska-nak 10 inaktív nap után lezárjuk a jelszavát.
`chage -I 10 joska`
`chage --inactive 10 joska`

- ▶ **Felhasználói jelszó: passwd**
 - Saját jelszó módosítása:
▶ `passwd`
 - Rendszergazda jogosultsággal módosítható bárkinek a jelszava:
▶ `passwd felhnev`
- ▶ Jelszó tárolása: `/etc/shadow`
- ▶ Jelszó generálása: `makepasswd`

- ▶ Felhasználói adatok módosítása: `chfn`
 - A finger csomag szükséges.
 - Pl.:
`# chfn -f János janos`
`# chfn --full-name "Nagy János" janos`
 - **Kapcsolók:**
 - f, --full-name – teljes név
 - o, --office -- irodai szobaszám
 - p, --office-phone -- irodai telefonszám
 - h, --home-phone -- otthoni telefonszám
- ▶ Felhasználó törlése: `deluser`, `userdel`
- ▶ A `/etc/deluser.conf` segítségével érdemes szabályozni. Részletekért nézzük meg a `deluser.conf(5)` kézikönyvet.
- ▶ Pl.:
 - `deluser --remove-home`
 - `deluser --remove-all-files`
 - `deluser --backup`
 - `deluser --backup-to`

- ▶ **Felhasználói csoportok kezelése**
 - Csoport létrehozása: `addgroup`, `groupadd`
- ▶ A csoportadatok helye: `/etc/group`

- Az /etc/group állomány kézzel is szerkeszthető (szövegszerkesztővel, mcedit) a root által.
- ▶ Minden felhasználónak saját nevével létrejön egy csoport is! Pl. **kati** user esetén egy **kati** csoport is létrejön.
- ▶ Csoporthoz adás: `gpasswd -a kati info`
 - Az info csoporthoz adjuk a kati usert
- ▶ Csoportból kivétel: `gpasswd -d kati info`
 - Az info csoportból kivesszük a kati usert
- ▶ Felhasználói csoportok kezelése (és user adatok)
 - A „usermod” parancs:
 - ▶ Felhasználói fiókok módosítása.
- ▶ A joska felhasználó felvétele az info csoportba:
 `usermod -a -G info mari`
 (A -a hatására hozzáfűzés történik, vagyis a többi csoport megmarad. Ha elhagyjuk a -a kapcsolót, akkor a többi csoport törlődik.)
- ▶ usermod parancs többi kapcsolója:
 - `-u érték user`: a felhasználó azonosítóját változtatja (UID)
 - `-g csoport user`: a felhasználó csoportját változtatja
`usermod -g info mari`
 - `-G csoport1,csoport2`: a usert több csoporthoz is rendeli
`usermod -G human,gazdasag,rgazda mari`
 - `-L user`: tiltja a felhasználó hozzáférését (nem tud belépni) `usermod -L -e 1970-01-01 mari` (a lejárat dátum bármi régebbi lehet mint az aktuális dátum) (`--lock`)
 - `-e érték`: lejárat dátum (`--expiredate`)
 - `-U user`: unlockolja/feloldja a tiltást
 - `-d user`: új home könyvtár (`--home`)
 - `-s user`: shell beállítása (`--shell`)
- ▶ Az id parancs:
 - Felhasználó adatainak lekérése.
- ▶ Csoporttag:
 - `id`
 - `id -nG`
- ▶ Felhasználói fiókok és „default” kezelése
 - Felhasználói fiók beállítások:
 - ▶ Az /etc/login.defs könyvtárban/fájlban a felhasználók számára beállítható néhány adottság. Pl.:
 - ▶ A felhasználók levelei hol tárolódnak.
`MAIL_DIR /var/mail`
 - ▶ A sikertelen bejelentkezések naplózása a /var/log/faillog fájlba.
`FAILLOG_ENAB yes`

- Az ismeretlen felhasználónevek naplózása sikertelen bejelentkezés esetén.
`LOG_UNKFAIL_ENAB no`
- A sikeres bejelentkezések naplózása:
`LOG_OK_LOGINS no`
- Jelszavak kontrollálása. Alapértelmezésként, mikor jár le, mikor változtathatja meg, mennyi legyen a jelszó minimum hossza, mikor legyen figyelmeztetés.
`PASS_MAX_DAYS 99999`
`PASS_MIN_DAYS 0`
`PASS_MIN_LEN 5`
`PASS_WARN_AGE 7`
- Felhasználók minimális és maximális azonosítója useradd esetén:
`UID_MIN 1000`
`UID_MAX 60000`
- Minimális és maximális csoportazonosítók: `GID_MIN 1000 GID_MAX 60000`
- A bejelentkezés visszautasítása ennyi sikertelen próbálkozás után: `LOGIN_RETRIES 5`
- A maximális sikertelen bejelentkezések után ennyi ideig tiltva:
`LOGIN_TIMEOUT 60`

● Felhasználói fiók alapbeállítások (defaults):

- Az /etc/default könyvtár tartalmazza.
- Pl. /etc/default/useradd

```
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

- A /etc/skel könyvtár tartalmazza az alap home könyvtár tartalmát.

Összefoglaló kérdések (9-10. fejezet):

1. Milyen programmal tudsz tömöríteni?
2. Milyen paranccsal tudsz manuálisan felhasználót hozzáadni?
3. A /etc/passwd fájl mit tartalmaz?
4. A chage mire jó?
5. Jelszót hogy tudod beállítani? Hol tárolódik el titkosított formában?
6. Felhasználói adatokat hogy tudod módosítani?
7. Hány módon tudsz hozzáadni felhasználót csoporthoz? Sorold fel őket!
8. A /etc/login.defs fájlban mik tárolódnak? (sorold fel 3 dolgot legalább!)

11. Felhasználókezelés – Rendszergazda

A rendszergazda felhasználó és funkciói

- ▶ A kiemelt felhasználó a Linux rendszerben: root

- ▶ UID: 0, GID: 0
- ▶ Mindenhez hozzáfér a rendszerben!
- ▶ A „su” parancs:
 - Felhasználó váltás és parancs futtatása:
 - Pl.:
 - Például mari felhasználóvá válhatunk:


```
su mari
```

 - ▶ Ha kötőjelet is használok, akkor a mari felhasználó teljes környezetét kapom:


```
su - mari
```
 - Egy csoportot is felvehetünk a sg parancssal. Például az info csoport felvétele:


```
sg info
```
- ▶ A sudo parancs
 - A sudo parancssal végre lehet hajtani rendszergazdai jogosultságú parancsokat. A felhasználó jelszavát kéri, ez azért fontos, hogy ne tudjon akárki módosítani a rendszeren.
 - Az Ubuntu és a Linux Mint rendszerek alapértelmezetten rendszergazdai jogokat csak a sudo-n keresztül biztosítanak, a root felhasználó nem kap jelszót.
 - A sudo beállításait fájlkból közvetlenül nem szerkesztjük. Helyette a következő parancsot használjuk:


```
# visudo
```

Beállítások → /etc/sudoers fájlban
 - Szintaxis:

```
<username> ALL=NOPASSWD: ALL
```

Példa:

```
joska ALL=NOPASSWD: ALL
```
 - A NOPASSWD nélkül kéri az adott felhasználó jelszavát.
 - Rendszergazda jogosultság beállítása:

```
joska ALL=(ALL:ALL) ALL
```

Összefoglaló kérdések:

1. Mi az UID-je a rootnak?
2. A su mire jó? Hogy használjuk?
3. A sudo miért jó? Mi a különbség a sudo és a su között?
4. Hol tudjuk beállítani a felhasználó sudo jogát?

12. Csomagkezelés, processzkezelés

- ▶ Feladata: Alkalmazások, ill. összetevőik telepítése.
- ▶ Két alapvető formátum van:
 - Az RPM alapú rendszerek (RedHat, CentOS, Fedora, openSUSE) csomagkezelője az rpm parancs. Ennek egyik burkolóprogramja a yum/dnf. SUSE alap rendszereknél yast vagy yast2 parancssal telepíthetünk.
 - A Debian GNU/Linux egyik előtétprogramja az apt csomagban található. A telepítéshez kezdetben az apt-get parancsot használhattuk. A csomagban azóta megjelent egy apt parancs is, amely az apt-get parancs egy továbbfejlesztett verziója. Az apt a **Advanced Packaging Tool** rövidítése.

- Csomagkezelés sokszor nehézkes volt a függőségek miatt, így egyszerűsítésre volt szükség.

- Van grafikus és konzolos front-end a csomagkezelőkhöz.

► **Processzkezelés:**

- A folyamat (process) egy elindított program még létező példánya.

- Az operációs rendszerek egyik legfontosabb feladata a folyamatok kezelése.

- A 0-s azonosító fenntartott, az 1-es pedig egy speciális, az ún. init process ID-je. (folyamatazonosító)

- Egy rendszeren belül egy időben nem lehet két azonos PID-ű folyamat.

- A legelső (init) folyamat kivételével minden másik folyamat úgy keletkezik, hogy egy már létező folyamat létrehozza. Az eredeti folyamatot a továbbiakban szülőfolyamatnak hívjuk (parent process, azonosítóját jellemzően PPID-ként emlegetik), a frissen létrejött neve: gyerek (child process)

- Természetesen a való élet szabályai itt is érvényesek, egy folyamat egyidejűleg lehet valamely másik folyamat gyereke, de mivel neki is lehet (akár több) gyereke, így egyben szülő-folyamat is.

- Azok a folyamatok, amelyek szülőfolyamatként a 2-s PPID-t mutatják (Egyéb UNIX és UNIX-jellegű rendszert használók számára talán furcsa lehet, mert máshol a PPID=0 szokott lenni a rendszerfolyamat jellemzője), azok a rendszermag (kernel) részei (nem található önálló bináris ezzel a fájlnevével), és csak adminisztrációs, ütemezési okokból látszanak folyamatként. Nevük: kernelszintű (vagy simán: kernel-), vagy még egyszerűbben: rendszerfolyamatok. A rendszerfolyamatokra nem érvényes semmilyen jogosultság kezelés

- A folyamatok életük során használnak valamennyi CPU-erőforrást, memóriát is, terhelik a gép IO-alrendszerét. Egy folyamat megszűnésének van egy fázisa, amikor ún. zombifolyamatként létezik a rendszerben.

- A folyamatokról információt vagy a /proc fájlrendszeren keresztül direktben, vagy pedig a ps paranccsal lehet szerezni.

- állapotok:

- R: running → fut v. futásra kész

- S: interruptable sleep → A folyamat várakozik valamilyen eseményre vagy erőforrásra és megszakítható egy jelzés által. Például egy számlálóra (sleep) vagy valamilyen Be/Ki műveletre

- D: uninterruptable sleep → A folyamat várakozik valamilyen eseményre vagy erőforrásra és nem szakítható meg jelzés által. Általában ez az állapot valamilyen be/kimeneti eszközre való várakozást jelent

- T: felfüggesztett, áll (Stopped / Traced): Ide kerül egy folyamat ha a terminálról futtatva lenyomjuk a CTRL+Z billentyűt és a háttérbe dobjuk

- X: dead / halott: nem látható elvileg ilyen állapot

- Z: zombie folyamat: megszakított, memóriában maradt.

- Lásd „man ps”

- Az első valódi folyamat az init ill. az új rendszereken a systemd, amelynek azonosítója 1. A következő elindított folyamatok mind tőle származnak, így a folyamatok összefüggésükben egy faszerű struktúrát alkotnak. → **pstree**

```
[kuna@fedora ~]$ pstree -p
systemd(1)─ModemManager(1109)─{ModemManager}(1122)
               │               │{ModemManager}(1139)
               └─NetworkManager(1182)─dhclient(1347)
                                   │{NetworkManager}(1195)
                                   └─{NetworkManager}(1197)
               └─Thunar(1736)─{Thunar}(1749)
                           │{Thunar}(1751)
               └─VBoxClient(1660)─VBoxClient(1671)─{VBoxClient}(1684)
               └─VBoxClient(1662)─VBoxClient(1668)
               └─VBoxClient(1663)─VBoxClient(1666)─{VBoxClient}(1672)
               └─VBoxClient(1664)─VBoxClient(1670)─{VBoxClient}(1674)
                                   │{VBoxClient}(1675)
               └─VBoxService(1108)─{VBoxService}(1127)
                                   │{VBoxService}(1128)
```

- A „job” fogalma: A terminálban levő munka során egy parancssorral akár több műveletet / folyamatot is lehet végrehajtani, pl:

```
$ cat valami.txt | grep alma
```

Itt 2 folyamatot indít és köztük csővezetéseket (PIPELINE) hoz létre. Ezért a terminállal történő munka számára szükséges egy másik fogalom is, ami jellemzi az indított folyamatokat.

- A job-ok követése: **jobs** parancs
- A folyamatok előtérben és háttérben futhatnak: FG, BG
- Előtérben futó folyamatokat a terminálon meg tudjuk szakítani: CTRL+C
- Előtérben futó folyamatokat a terminálon fel tudjuk függeszteni: CTRL+Z
- Háttérbe futó folyamat az & jellel, pl.: **sudo apt update &**
- Folyamatkezelés:
 - ps: folyamatok listája, állapota
 - pstree: folyamat struktúra lista
 - wait: [n] → adott számú job-ra/folyamatra vár és visszaadja annak kilépési kódját
 - top, htop, atop: dinamikusan listázza a folyamatokat.
 - nice: folyamat prioritásának (fontosság) növelése, csökkentése, az alacsonyabb szám magasabb prioritást jelez, pl.: **nice -n 3 program-neve**
 - A program prioritását futás közben is lehet növelni/csökkenteni, erre való a renice parancs: **renice -n 3 -p PID**
 - Jelzések, folyamatok megszakítása:

Egy folyamat „kívülről” megszüntethető egy ún. megszakítás (signal) adott folyamatnak elküldésével. A megszakítások teljes listája lekérdezhető: **kill -l**

A megszakításhoz szükséges a folyamat azonosítója, és hogy a megszakítást kezdeményező jogaival futó folyamatot akarjunk megszüntetni. Rendszergazda joggal természetesen mások folyamatai is megszakíthatók.
 - A killall parancssal minden folyamatot megöl, ami az adott folyamathoz tartozik pl. a **killall conky** esetében az összes conkyhoz tartozó folyamatot kilövi.

➤ Jelzések listája:

Név	Szám	Leírás
TSTP	20	Felfüggesztés, CTRL+Z
INT	2	Azonnali leállítás, CTRL+C
QUIT	3	Leáll, de takarítja állományait
KILL	9	Azonnali leállítás, kill parancs
ABRT	6	Leáll, de core dump memória kép
HUP	1	Hang up, újraindítás (config beolvasás)
TERM	15	Takarít és leáll

● PIPELINE – „csővezetékkezelés”:

- az egyik program kimenetét használja fel a másik program bemenetként
- Pl.: `ifconfig | grep eth0`

Összefoglaló kérdések:

1. Mi a két alapvető csomagkezelési formátum?
2. Mi a PID/ process ID?
3. Hogy jön létre egy folyamat? Mi az a child process?
4. A pstree mit mutat meg?
5. Mi a „job”?
6. Hogyan lehet egy programból a terminálban kilépni?
7. A nice parancs mit csinál?

13. Fájrendszer, jogosultságok

► Fájrendszer:

- LINUX rendszer alapvetően:
 - Rendszer partíció
 - Cserhely (swap) partíció, ami akkor hasznos, ha elfogy a fizikai memória, a nem használt programfájlok ilyenkor erre a partícióra átkerülnek.
- Cserhelyet azonban fájlként is adhatunk a rendszerhez menet közben. Az ilyen fájlok mérete nem változik, így töredezettség miattuk nem lép fel.
- A Linuxot több külön álló fájlszisztemre szokás telepíteni, ha azt szerverként telepítjük. Ennek oka a biztonság növelése, pl. ha a naplófájlok megtelítik a rendszert, és az külön partíción van, nem fog megállni a rendszer.
- Általában a következő könyvtárakat szokás külön partícióra tenni:
/home /var /tmp /boot /usr
- Partíciók
 - A háttértárolókat (merevlemez) partíciókra osztjuk fel, amelyeken létrehozuk a fájlszisztemet.
 - A partíciós tábla határozza meg a partíciók tárolásának módját. Kompatibilitási okokból a Linux alapértelmezetten a „DOS partíciós táblát” használja.
 - A DOS partíciós táblában 4 elsődleges partíció lehet, a többi egy kiterjesztett partícióban helyezkedhet el. A kiterjesztett partíciót további részekre oszthatjuk, ezeket logikai partícióknak nevezzük.
 - A GPT partíciós tábla korlátlan partíciót enged. Ez használatos ma már.

► Jogosultságok

- A DAC szó a Discretionary Access Control szavakból alkotott betűszó, jelentése röviden „Kizárólagos hozzáférés-vezérlés”. Ez megengedi az azonosított felhasználóknak az objektumokhoz való hozzáférést azok tulajdonságai alapján. A hozzáférés csak a tulajdonosra, illetve csak a csoportra állítható be.
- Ezeket a unixos alapjogoknak is szokás nevezni.
- Ismert még a MAC vagy Mandatory Access Control, vagy Rendelkező hozzáférés-vezérlés. Az objektumok számára hozzáférési szabályokat hozunk létre, amely minden felhasználóra (még a rootra is) érvényesek.
- A harmadik a Role-based Access Control vagy RBAC. Szerep alapú hozzáférés-vezérlés. A rendszergazda különböző szerepeket hoz létre. Az egyes objektumokhoz az adott szerepben lehet hozzáférni, az adott jogokkal.
- Alapvetően háromféle jogot különböztetünk meg:
 - Olvasás (r)
 - Írás (w)
 - Végrehajtás (x)
- A jogok háromféle felhasználónak adhatók:
 - tulajdonos
 - csoportba tartozó felhasználó
 - mindenki más (aki az előző kettőben nincs benne)
- A chmod paranccsal állíthatjuk a jogokat, ahol néha hivatkozunk a tulajdonosokra, csoportokra, illetve mindenki másra, az alábbi táblázat alapján:

tulajdonos	user	u
csoport	group	g
mások	other	o
mindenki	all	a

● Példa:

Az `ls -l` kiemerte:

```
-rw-r--r-- 1 joska joska 3184 dec 13 18.37 .bashrc
```

Típus	Tulajdonos joga	Csoport joga	Mások joga
-	rw-	r--	r--

● A típus lehet:

Tartalom	Jelentés
-	Szimpla fájl
d	Könyvtár
D	Solaris kapu (folyamatok közötti kommunikáció)
c	Karakteres eszköz (tty v. nyomtató)
b	Blokkeszköz (lemez v. CD-ROM)
l	Szimbólikus link (symlink)
s	Socket
= vagy p	FIFO (System V, Linux)

● A **chown** parancs:

- A parancsot egy állomány vagy egy könyvtár tulajdonos, illetve csoport beállítása használhatjuk.

```
chown joska.joska fajlnev
chown joska:joska fajlnev
chown joska. fajlnev
chown joska: fajlnev
```

● A **chmod** parancs:

- A jogok beállítására használható.
- Két módon adhatunk/vehetünk el jogokat. Az egyik számokkal, a másik betűkkel.
- Betűkkel először megadjuk kinek adunk vagy kitől veszünk el. Ez után +/- attól függően, hogy adunk vagy elveszünk. Utána megadjuk
- Szeretnénk a felhasználónak írási jogot adni az erdo.txt fájlra:
`chmod u+w erdo.txt`
- Adjunk a csoportnak és a tulajdonosnak minden jogot:
`chmod ug+rx erdo.txt`
- Megadás számokkal:
`chmod 770 erdo.txt`
- Jelentése:

Tulajdonos (u)	Csoport (g)	Mások (o)
421	421	000

- A 4 az olvasást, a 2 az írást, az 1 pedig a futtatást jelenti

● A **setuid**, **setgid**, **sticky** bitek:

- **SETUID**: Lehet állítani, hogy melyik felhasználó jogával tud futni a program, de ez biztonsági kockázattal jár, ezért csak akkor használjuk, ha csak így lehet a problémát orvosolni.
- **SETGID**: Beállítása esetén a program annak a csoportnak a jogaival fog futni, akinek a fájl a birtokában van, ezt könyvtárak esetén is lehet beállítani, akkor minden fájl az adott csoport tulajdonába van.
- **STICKY**: A sticky bit bekapcsolása fájlok esetén azt jelzi az operációs rendszernek, hogy a fájlt tartsa a memóriában a végrehajtás után is, hogy később gyorsabban induljon el.
 - A sticky bitet be lehet kapcsolni könyvtárak esetén is. Az ilyen bittel ellátott könyvtárban bárki írhat fájlokat (a többi jognak is rendben kell lennie), de mindenki csak a sajátját törölheti.

Összefoglaló kérdések:

1. A Linux alapértelmezetten milyen partíciókból áll?
2. A cserehely mi célt szolgál?
3. Milyen jogosultságai vannak egy fájlnak?
4. Milyen paranccsal lehet a tulajdonost megváltoztatni?
5. Hogy használjuk a chmod parancsot?
6. Mi a SETUID, SETGID?
7. Mi a sticky bit?

14. Ha akarsz a Linuxszal még foglalkozni

A [Linux Mind](#) youtube csatorna nagyon hasznos tud lenni azoknak, akik a Linux alapjait mélyebbre el akarják sajátítani, illetve programok, OS-ek használatához segítséget szeretne kapni. Van egy discordos közössége is, a PenguinPit, ahol sokan összegyűlnek, hogy tudjanak egymástól segítséget kérni, illetve hogy jól beszélhessenek ott az emberek. (<https://discord.gg/7DdbM9g>)

[Skamilinux-ot](#) is javaslom, mert ő is nagyon hasznos dolgokat mond el, sőt, akinek gyenge gépe van, pl. egy Pentium 4-esé, ővele nagyon jól jár, mert a Puppy Linuxban nagyon is jártas, illetve neki is van saját discord szervere, ahol nyugodtan tehetsz fel kérdést. (<https://discord.gg/BnJJEsk>) Ő az, akire azt lehetne tenni jelzőként, hogy „minden ami retró” vagy retró guru.

A [szit.hu](#) is egy jó hely, onnan sokat merítettem ehhez a dokumentumhoz, de nyilván sokkal több van fent.

BASH programozáshoz, illetve még több információért a <https://books.goalkicker.com/> oldalt javaslom, onnan a Bash és a Linux-os könyveket. A Git sem árt, hogy ha szeretnél programozni és a programjaidat szeretnéd publikálni a későbbiekben vagy éppen mások projektjeit szeretnéd könnyedén letölteni.

Illetve még van a <http://linuxcommand.org/tlcl.php> (The Linux Command Line) is, ami nagyon sok tudást nyújt át 555 oldalban.