

Linux bevezető ismeretek

1. Bevezetés

Nagyon fontos, hogy a lentebb szereplő parancsokat kipróbáljuk! Erre több lehetőségünk adódik.

- van a saját számítógépünkön valamilyen telepített linux verzió és azt használjuk,
- letöltünk egy live cd-t és azt használjuk, pl.: PuppyLinux(<http://skami.homelinux.org/>)
- eddig nem volt, de szeretnénk a gépünkre linuxot is; egy kis leírás, bár nem a legfrissebb, de jól használható: <http://hogyan.org/debian-5-lenny-telepites>
- a putty programmal belépünk valamely egyetemi szerverre (users, turdus) és ott dolgozunk;

2. Egy kis történelem

2.1. A kezdetek

A számítógépek az ötvenes évektől a nyolcvanas évek elejéig a kutatók eszközei voltak és igazából a tömegek előtt el voltak zárva. Több kutató használt egy nagy gépet, a fejlesztéseiket megosztották egymással, igazi kis közösségek jöttek így létre. A kor egyik legendás gépe volt a PDP10-es, ezen az ITS nevű operációs rendszer futott, ennek továbbfejlesztésén dolgozott Richard Stallman is (lásd később). A PDP10 fejlesztését azonban a gyártója a nyolcvanas évekre abbahagyta, az intézeteknek újabb gépek után kellett nézniük. Ezek már más operációs rendszert futtattak, melyek nem voltak szabadok, már ahhoz is titoktartási szerződét kellett aláírniuk, ha egy futtatható másolatot akartak. Azaz tilos lett egymásnak segíteni, az eddig együttműködő közösségek felbomlottak, nem oszthatták meg egymással fejlesztéseiket.

2.2. A GNU

Richard Stallmann ezt az új helyzetet nem tudta elfogadni, elhatározta egy új, teljesen nyílt operációs rendszernek a megírását, 1983 táján létrehozta a GNU projektet, hogy terveit megvalósítsa. (A GNU jelentése: GNU is not UNIX). Ekkor fogalmazta meg a GNU kiáltványt, melyet teljes terjedelmében például a gnu.hu oldalon olvashatunk el.

Ekkor fogalmazta meg a szabad szoftverekkel kapcsolatos alapelveit:

- a program szabadon használható bármilyen célra.
- a programot bárki szabadon módosíthatja igényei szerint.
- a programot bárki továbbadhatja akár ingyen akár pénzért.
- a program módosított verziói szintén szabadon terjeszthetők.

Ezen elveket a jogászok számára is elfogadható formába kellett önteni, így jött létre egy különleges licenc, a GPL (General Public Licence). Teljes szövege magyarul szintén a gnu.hu oldalon olvasható. Ekkoriban jött létre a szabad szoftvereket támogató alapítvány, az FSF (Free Software Foundation). Céljukról, működésükről többet olvashatunk pl.: az fsf.hu oldalon.

Már csak egy valami hiányzott: egy olyan kernel (rendszermag, azaz maga az operációs rendszer), melyen futtatni lehetne a GNU programokat.

2.3. Linus Torvalds

1991-ben egy finn egyetemista, Linus Torvalds épp egy új projekten kezdett el dolgozni, egy új, szabad operációs rendszeren, melyben ki akarta javítani az oktatásra akkoriban előszeretettel használt MINIX operációs rendszer hibáit, hiányosságait. Azaz adott volt egy kernel (Linus munkája) alkalmazások nélkül és adott volt egy alkalmazás gyűjtemény (GNU) kernel nélkül. Nem kellett sok idő ahhoz, hogy egymásra találjanak, így született meg a Linux, amit helyesen GNU/Linux-nak kellene neveznünk. Pár év, és megjelentek az első disztribúciók: kernel és a rengeteg GNU alkalmazások közül néhány összeépítve egy jól használható rendszerré. (pl.: Debian: 1993. augusztus 16.)

3. Disztribúciók

A Wikipédiából, a szabad enciklopédiából:

A Linux-disztribúció (röviden: distro vagy disztró) olyan összeállítás, mely egy felhasználásra kész Linux vagy GNU/Linux alapú operációs rendszert, és ahhoz tartozó, válogatott programokat tartalmaz.

Operációs rendszerekkel kapcsolatban a disztribúció (más néven distro vagy disztró) szó alatt egy gondosan egybeválogatott, általában rendszermagból és felhasználói programok csomagjaiból álló terjesztést értünk.

3.1. Disztribúciók közötti leggyakoribb különbségek

Disztribúciókat legtöbbször az különbözteti meg, hogy milyen célközönségnek és milyen feladatra készítik őket, így mindenki megtalálhatja a neki leginkább megfelelőt. Így léteznek olyanok, melyek lehetőséget nyújtanak arra, hogy szinte az összes konfigurálási lehetőséget egy grafikus felületen végezzük el és vannak olyanok is, amelyek megkövetelik, hogy a felhasználó mindent a konfigurációs állományok szerkesztésével állítson be a saját ízlésének megfelelően. Egyes disztribúciók célja, hogy mindig a lehető legfrissebb szoftvereket szállítsa, míg mások jól kitesztelt, stabil, ám emiatt kissé elavult csomagokat szállítanak. A legtöbb disztró adott közönséget céloz meg: profi vagy kezdő felhasználókat, adminisztrátorokat, „buherátorokat”, kevés memóriával rendelkező vagy csak CD-t tartalmazó gépeket stb. Néhány disztró a grafikus környezetet, míg mások inkább a

karakteres konzolt támogatják.

További fontos különbség, hogy milyen csomagkezelőt használnak az adott terjesztésben. A könyvtárstruktúra általában hasonló módon van felépítve, viszont kisebb különbségek adódhatnak e tekintetben is, extrém esetekben teljesen eltérő felépítést is alkalmaznak a disztribútorok (pl.: GoboLinux). A disztrók egyik fő jellemzője az egyes programcsomagok installálásának, eltávolításának és frissítésének megkönnyítése és támogatása (lásd még: APT, RPM). A csomagkezelők a rengeteg feltelepíthető program karbantartását, frissítését, telepítését, stb teszik könnyebbé: például a GNU/Debian 6 esetében majd 30000 különböző program közül válogathatunk, így szinte biztosan megtaláljuk a felmerült feladataink megoldásához szükséges szofvereket e bőséges választékban. Az egyes programok, csomagok pontos verzió számmal vannak ellátva, egy-egy program megfelelő működéséhez szükség lehet más programokra is, azaz függőségei lehetnek. Ezen függőségek (lehetőleg automatikus) feltelepítését is a csomagkezelők végzik.

Hardvertámogatás terén is adódhatnak különbségek, viszont alapvetően mind ugyanazt a kernelt használják, így elviekben ha egy disztribúció alatt egy hardver működik, akkor az bármely más, az adott architektúrát támogató disztribúció alatt is működésre bírható. Vannak céldisztribúciók is, például kifejezetten tűzfal vagy router üzemelésére. Megkülönböztethetjük őket az alapján is, hogy server, desktop vagy embedded felhasználásra szánják.

A disztrók nagy részének készítői komolyan veszik a biztonsági problémákat, és az ismert hibák javításait rövid időn belül elérhetővé teszik disztrójuk csomagfrissítési módszerének segítségével

Nagy eltérések vannak a disztrók kiadásai között eltelt időben; egyes disztrók fix ciklust alkalmaznak (például 6 hónaponként egy új kiadás), más disztróknál nincs kötött kiadási ciklus. Léteznek kereskedelmi terjesztések.

Léteznek vállalati és otthoni/kisirodai disztribúciók is.

Nem mindegyik disztró ugyanazt a kernel verziót használja, továbbá sok disztró saját igényeinek megfelelően módosítja a hivatalosan kiadott, ún. vanilla kernelt.”

A nagyobb és ismertebb disztribúciók (a teljesség igénye nélkül)

- blackPanther OS, magyar Linux-disztribúció
- UHU-Linux, magyar Linux-disztribúció
- Debian GNU/Linux
- Ubuntu, Kubuntu, Xubuntu
- Mandriva
- PCLinuxOS
- Red Hat Linux
- Fedora
- CentOS
- openSUSE
- Slackware

- Gentoo
- Arch Linux
- Knoppix, Damn Small Linux, Live CD-ként való futtatásra tervezve
- CrunchBang Linux
-
-
-

Egy kis érdekesség: <http://upload.wikimedia.org/wikipedia/commons/8/8c/Gldt.svg>

4. Könyvtárszerkezet

Linux alatt egy "tökéletes" fa struktúrába van szervezve a teljes könyvtárszerkezet, (azaz ne számítsunk C, D... meghajtókra!) Mindennek az alapja a "/" jellel jelölt gyökérkönyvtár más néven root. Ez minden Linux alapja, ebből ágaztatható le a teljes szerkezet.

1. példa. Adjuk ki a következő utasítást: `ls /`

Hasonló listát kell látnunk:

```
bin boot cdrom dev etc home lib lost+found media mnt opt proc root sbin
sys tmp usr var vmlinuz
```

Ezek a főkönyvtárak majdnem minden Linuxban változatlanul megvannak, leszámítva talán a /cdrom-ot és /media-t. A /media egy újabb "találmány", ide kerülnek a cserélhető médiák. Nézzük, melyikben mi található:

bin, sbin: A bin könyvtárakban - nem meglepő módon - futtatható bináris állományok csücsülnek. Több bin könyvtár is található ezen kívül, például a /usr/bin és a /usr/sbin. Bár ez nem törvényszerű, de általában a bin könyvtárakban a minden felhasználó által elérhető állományok kerülnek az sbin könyvtárakba pedig olyan rendszereszközök, melyeket például csak rendszergazdák használnak. Az értelméről később, a Path kapcsán ejtünk pár szót. A /bin és /sbin az alaprendszerhez szükséges programokat tartalmazza, a felhasználó által telepített programok a /usr/bin /usr/sbin alá kerülnek.

boot: a boot könyvtárban található a bootnál fontos fájlok: általában a rendszermag (kernel), illetve Grub rendszerbetöltő esetén annak konfigurációs állománya is. A gyökérben található még egy vmlinuz fájl is (esetenként bzImage), mely egy ún. szimbolikus link a /boot/vmlinuz-ra azaz a rendszermagra. A szimbolikus link egy hasznos megoldás Linux alatt: ha egy fájlra több helyen is szükségünk van, elég ehgy helyen tárolni azt, és a többinél csak egy mutatót létrehozni az adott fájlra, így onnan is ugyanúgy elérhető, de nem foglal kétszer helyet a vinszeszteren.(lásd később)

cdrom: szintén egy szimbolikus link, általában a /media/cdrom könyvtárra. Ez utóbbi alá csatolódik be a CD meghajtó egység.

dev: Linux alatt fájlkon keresztül érünk el mindent a CD-vel kezdve, a hangon át, az egész. ezek a speciális eszközfájlok találhatóak ebben a mappában.

etc: Az etc könyvtár a gyűjtőhelye a különböző programok globális konfigurációs fájljainak. Ellentétben a Windowsos registry megoldással Linux alatt minden konfigurációs állomány egyszerű szövegfájlba van mentve, aminek nagy előnye, hogy az állományok akkor is egyszerűen elérhetők, ha a rendszer egyébként használhatatlan. Természetesen emellett az egyes programok felhasználó specifikus beállításokkal is rendelkeznek, ezeket a home könyvtárakban tárolja a rendszer, rejtett mappákban.

home: ezalatt a könyvtár alatt található a felhasználói könyvtárak, az adott könyvtár alatt a teljes dűlési joga van, ezen az egy könyvtáron kívül azonban leginkább csak olvasási joga van alából.

lib: a lib könyvtár alatt már a rendszer részei lapulnak: library fájlok, kernel modulok, stb.

lost+found: egy speciális könyvtár, jelen esetben egy ext3 típusú fájlrendszerrel szerelt partícióról van szó, ez a könyvtár nem is a Linux, mint inkább a fájlrendszer része. media: rendszerfüggő a dolog, általában a /media könyvtár alá kerülnek befűzésre a CD/DVD eszközök, pendrive illetve a floppy. Röviden: a cserélhető médiák.

mnt: a másik "betűzőgetős" könyvtár. Ez alá a könyvtár alá kerülnek (általában) befűzésre a fix partíciók. Mivel ebben a könyvtárstruktúrában nincs kiemelt "volume" egy egy meghajtónak, mint Windows alatt a C:, D:, stb., így egy-egy eszközt tetszőleges helyre befűzhetünk a fájlrendszerbe. Különösen praktikus ez például home könyvtár esetén: ha kinőjük az e célra fenntartott partíciót, és veszünk egy új vinszesztert, egyszerűen csak rámásoljuk anyagainkat, letöröljük az eredeti példányt, majd befűzzük a /home könyvtár alá az új adathordozót.

opt: a hivatalos leírás szerint külsős programok települnek ebbe a könyvtárba, de a rendszerek nagy részén üresen áll...

proc: Itt találhatóak az éppenfutó műveletek -fájlként leképezve, sorszámozva, illetve információk a rendszerről: processzorról, memóriáról, stb. Iszonyú mennyiségű hasznos információt talál itt az avatott kéz.

root: A rendszergazda (root) felhasználói könyvtára

sys: A másik számunkra homályos kérdés. 2.6-os kernellel együtt jelent meg ez az újfajta eszközkezelési metódus, ebben a könyvtárban található meg a sysfs számára egy komplett fa. Szintén egy kincsesbánya, de egy átlag felhasználó ritkán téved erre.

tmp: Az egyes programoknak szükségük van/lehet átmeneti fájlokra. Ezek kerülnek ide. Ez a másik olyan könyvtár, amely alapértelmezettben írható minden felhasználó számára.

usr: Ez alatt a könyvtár alatt található minden. Persze ez így kicsit túlzónak hat, de majdnem igaz: az usr könyvtár alatt található a telepített programok nagy része, hagyományból ide szokott az ember fia forrásokat pakolni (/usr/src), és azt lefogatni. Itt találhatóak a dokumentációk, itt találhatóak az ikonok nagy része, sorolhatnánk a végtelenségig...

var: Szintén számos szolgáltatás gyűjtőkönyvtára. Itt találhatóak a naplófájlok, egyes programok hosszabb ideig tárolt, mégis átmeneti fájljai, alapértelmezettben a felhasználói levélboxok, stb.

2. példa. *Nézzünk bele az egyes könyvtárakba: adjuk ki a következő utasítást (utánuk ENTER): ls /bin (után ls /boot, ls /home,...)*

3. példa. *Gépeljük be majd nyomjunk ENTER-t: cat /proc/meminfo*

(Gondolom már kiderült, hogy a ls parancs kilistázza az adott könyvtár bejegyzéseit)

5. Jogosultságok

A linux disztribúciókban található egy kitüntetett felhasználó, a rendszergazda, a telepítés során jön létre, a neve: root. Neki mindenhez joga van, bármit törölhet, bármit megnyithat, létrehozhat felhasználót, stb. Az ő általa indított programok az ő jogaival futnak, egy szándékosan vagy véletlenül megváltoztatott program a root jogaival futva komoly károkat tud okozni. Ezért a legtöbb disztribúcióban létre kell hozni már a telepítéskor egy korlátozott jogú felhasználót, akinek az adataival belépve korlátozott jogokkal tudunk dolgozni. Ez így biztonságos!

4. példa. *Írjuk be: cat /etc/passwd*

A kapott hosszú lista első oszlopa a rendszerünkön lévő felhasználók neveit tartalmazza, a sajátunkat is ott kell látnunk.

(Talán kiderült már: a cat utasítással szöveges fájlok tartalmát lehet kilistázni) A lentebb bemutatásra kerülő jogoknak igazi jelentősége a több felhasználó által használt rendszerek esetében van (pl.: users és turdus szerverek), ha egy gépet csak egyedül mi használunk, a jogosultságok állítgatása nem lesz annyira fontos. Minden felhasználó valamilyen csoportnak is tagja (akár többnek is), mindenkinek van egy alapértelmezett csoportja (elsődleges csoport), ez Debian rendszeren megegyezik a felhasználó nevével, a felhasználó létrehozásakor jön létre, az új felhasználó egyből belekerül.

5. példa. *Adjuk ki a következő utasítást: cat /etc/group*

A kapott lista első oszlopa a rendszerünkön lévő csoportok neveit tartalmazza. A Linux fájlrendszere tárolja a fájl tulajdonosának azonosítóját a fájlhoz tartozó csoportokat és a hozzáférési jogosultságot is. A hozzáférési jogosultságok ábrázolásához egy 3 jegyű számból álló kódot használ, amit fájlmodnak nevezzünk.

Első szám a saját (user) jogot

Második szám a csoport (group) jogot

Harmadik a többiek (others) jogait rögzíti

A saját jog alatt a fájl tulajdonosának jogait értjük, legtöbb esetben ő az adott fájl vagy könyvtár létrehozója is. Minden komponens a következő részekből áll:

1. Saját r w x | 4 2 1

2. Csoport r w x | 4 2 1

3. Többiek r w x | 4 2 1

r (Read): Olvasási jog (vagyis az adott fájl ezáltal olvasható)

w (Write): Írási jog (az adott fájl ezáltal válik írhatóvá)

x (Executable): Végrehajtási jog (Futási)

5.1. Jogosultságok megváltoztatása

Egy fájl tulajdonosi (hozzáférési) jogait csak a fájl tulajdonosa, vagy a rendszergazda tudja megváltoztatni. `chmod +|-<mód> <fájlnev>`

Chmodnál meg kell határozni az alábbiakat:

Adunk vagy elveszünk jogot (+ : adunk, - : elveszünk)

Kinek/kitől (saját, csoport, mások (ugo))

Milyen jogot adunk (r w x / 4 2 1)

u – saját (User)

`chmod u+w munka.tar.gz` Saját magunknak írási jog

g – csoport (Group)

`chmod o+x munka.tar.gz` Másoknak futtatási jog

o – mások (Other)

`chmod o+x,u+w munka.tar.gz`

a - mindenki (All)

`chmod 777 munka.tar.gz` Mindenkinek minden jog, ugyanaz mint:

`chmod a+rwx <fájlnev>`

`chmod 700 munka.tar.gz` Csak nekem van jogom mindenhez, ugyanaz mint

`chmod u+rwx,g-rwx,o-rwx <fájlnev>`

Fájlok esetében a végrehajtási jognak csak a futtatható fájlknál van jelentőségük (bináris állományok, scriptek). Könyvtárak esetén az olvasási jog azt jelenti, hogy elolvashatja a fájl neveit az adott könyvtárban, az írási jog jelenti, hogy a könyvtárban állományt, könyvtárat hozhatunk létre, míg a futtatási jog megengedi a belépést a könyvtárba.

5.2. Alapértelmezett jogok

Amikor egy fájl létrehozunk, akkor az a jogosultságoknak egy alapértelmezett csoportjával fog rendelkezni. Pl.: Létrehozunk egy üres fájl:

`$touch akarmi`

`$ls -la akarmi`

`rw-r--r--`

A létrehozáson kívül, hogy bármit is tettünk volna mint tulajdonosok, alapértelmezés szerint írási és olvasási joggal, a csoportba és a nem csoportba tartozók pedig csak olvasási joggal rendelkeznek. Ennek az az oka, hogy az operációs rendszer a fájl létrehozásakor a 022 maszkot alkalmazza. Egy adott állomány létrehozásakor alapértelmezésben senki sem kap futtatási jogot. Az alapértelmezett maszk lekérdezése:

`$umask`

`-> 022`

Könyvtárak létrehozása esetén a 777-ből vonódik ki a mask, azaz alapértelmezetten egy könyvtár 755 jogokkal jön létre. Fájlknál a 666-ból vonódik ki a mask, így 644 jogokkal jönnek létre a fájljaink.

6. Parancsok

Lásd a mellékelt `Linuxparancsoksegdlet.doc` fájlt.

7. Átírányítás

Bejelentkezés után egy program várja az utasításaidat, hogy azokat végrehajtsa, ez a shell. Több fajtája van, pl.: `sh`, `bash`, `csh` ... A számunkra szükséges szinten mindegyik egyformán működik. A bejelentkezés után a villogó prompt jelzi, hogy a shell várja az utasításainkat. A fentebb bemutatott parancsok után nézzünk valami különlegességet, amivel egész összetett feladatokat is megpróbálhatunk végrehajtani a shellel. Ez az átírányítás.

Egy programnak általában egy bemenete (`stdin` – általában a billentyűzet) van és két kimenete: `stdout` – szabvány kimenet, általában a képernyő, `stderr` – szabvány hibakimenet, szintén általában a képernyő. Lehetőség van bármelyik átírányítására fájlba illetve fájlból.

6. példa. *Szeretném tudni, hogy a gépemen mely felhasználó home könyvtára mennyi helyet foglal. Ehhez kiadhatom a*
`du -hs /home/` parancsot.*

Ez a képernyőre listázza az eredményt, amit kértem. Igen ám, de lehet, hogy ezt szeretném megőrizni, hogy meg tudjam mutatni az érintetteknek, esetleg szeretném magamnak automatikusan, időről időre elküldeni e-mailben. A parancs kimenetét átírányítjuk egy fájlba, így nem vész el.

7. példa. *Gépeljük be: `du -hs /home/* > foglalas.dat`*

Ez a parancs szabványos kimenetét átírányította a `foglalas.dat` fájlba. Ha ez eddig nem létezett, akkor létrehozta a shell, ha létezett, akkor a tartalmát kitörli és csak az új tartalom lesz benne.

Ha nem szeretnénk kitörölni a már létező fájlunk már létező tartalmát, akkor a következő képpen járhatunk el:

8. példa. *`du -hs /home/* >> foglalas.dat`*

Remélem, hogy a kétféle jel között látjuk a különbséget.

Igen ám, de ha a fenti parancsot kiadjuk, akkor egy sor hibaüzenet is keletkezhet, hiszen a legtöbb felhasználó home könyvtárába nincs jogunk „belekukkantani”. lehetőségünk van arra, hogy a szabványos hibakimenetet is átírányítsuk egy fájlba, ekkor a monitoron nem jelennek meg.

9. példa. *Írjuk be: `du -hs /home/* > foglalas.dat 2>foglalas.hiba`*

Igen, a 2 szám a hibacsatorna azonosítója, a `2>foglalas.hiba` utasítással irányítottuk a hibaüzeneteket egy másik fájlba. Ha nincs szükségünk a hibákra a fájlban és a képernyőn sem akarjuk látni, akkor egy különleges fájlba, a semmibe is átírányíthatjuk a hibakimenetet.

10. példa. `du -hs /home/* 2> /dev/null`

A bemenet átirányítása annyit jelent, hogy a programom a bemenetét nem a billentyűzetről kapja, hanem egy fájlból. Egyszerű példa: az előzőleg elkészített foglalas.dat fájlból keressük meg azon sorokat, melyekben gigabájtnyi helyfoglalásról van adat.

11. példa. `grep < foglalas.dat G`

Bizonyos mintát (G) tartalmazó sorokat keres a grep (man grep) utasítás, de bemenetét az adott fájlból veszi.

Újabb hasznos lehetőség a csövezés (pipe), amikor az egyik parancs kimenete alkotja a másik bemenetét. Ennek jele a |. Pl.: irassuk ki a passwd állomány sorait abc szerint rendezve!

12. példa. `cat /etc/passwd |sort`

A cat program kilistázza a fájl tartalmát, ez átadódik a sort program bemenetére (man sort).

8. Feladatok

1. feladat. *Nézz utána, hogy mit csinál az ncal parancs!*

2. feladat. *A hét milyen napján születted?*

3. feladat. *Mekkora helyet foglalsz a users.itk.ppke.hu szerveren?*

4. feladat. *Hozd létre a következő könyvtárstruktúrát a saját könyvtáradon belül!*

`./szulok/apa`

`./szulok/anya`

5. feladat. *Hozz létre egy fájlt (akár üreset is lehet) az apa alkönyvtáron belül! (touch, esetleg nano, esetleg cat,...)*

6. feladat. *Másold át az anya alkönyvtárba!*

7. feladat. *Írasd ki egy fájlba az elmúlt 10 percben módosított fájlok neveit a munkakönyvtáradon belül! (find parancs)*

8. feladat. *Fűzd hozzá a fájl végéhez az aktuális dátumot! (date és átirányítás)*

9. feladat. *Módosítsd az előző fájl jogait, hogy neked csak írási jogod, másoknak (csoport, egyéb) pedig semmilyen joga ne legyen!*

10. feladat. *Próbáld meg a tartalmát kilistázni! (pl.: cat)*

11. feladat. *Szerezz információkat az od programról! (man, keresők,...)*

12. feladat. *Add ki a következő utasítást!*

`echo ō|od -t x1`

Értelmezd az eredményt!

Tartalomjegyzék

1. Bevezetés	1
2. Egy kis történelem	1
2.1. A kezdetek	1
2.2. A GNU	1
2.3. Linus Torvalds	2
3. Disztribúciók	2
3.1. Disztribúciók közötti leggyakoribb különbségek	2
4. Könyvtárszerkezet	4
5. Jogosultságok	6
5.1. Jogosultságok megváltoztatása	7
5.2. Alapértelmezett jogok	7
6. Parancsok	8
7. Átirányítás	8
8. Feladatok	9