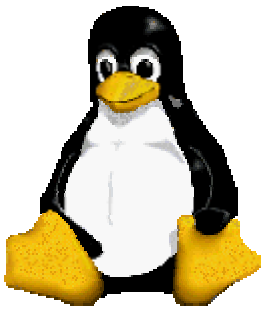
LINUX ALAPISMERETEK



Verzió: 1.0pre

Szerzõ: Zsoltino (Szalai Zsolt)

Copyright, licence és a felhasználás feltételei

Kapcsolatbalépés a szerzõvel: zsoltino@freemail.hu

Utolsó módosítás dátuma: 10. 10. 2003. by Zsoltino (Szalai Zsolt)

TARTALOMJEGYZÉK

01.	Copyright, licence és a felhasználás feltételei	2
02.	Tartalomjegyzék	3
03.	Előszó, avagy mijaza Linux?	4-6
04.	A Linux és én - a kezdetek	7
05.	GPL, General Public Licence	8
06.	Töredezettség-mentesítés	9
07.	A disztribúciók és azok mérete	10
08.	A Linux erőforrásigénye	11
09.	Verziószámok	12
10.	Windowsos programok, játékok	13
11.	GUI, azaz Graphical User Interface, X szerver, ablakkezelők	14
12.	A parancssor és a GUI	15
13.	A parancssor hatékonysága	16
14.	File hierarchia	17-18
15.	Néhány Linux parancs rövid leírása	19-24
16.	Wildcard, azaz a joker karakterek	25
17.	A tabulator gomb használata	26
18.	A rendszer biztonsága, jogosultságok	27-31
19.	Átirányítások	32-34
20.	Linkek	35-36
21.	Aliasok	37-38
22.	Változók, környezeti változók	39-40
23.	A Linux hatékonysága, példa munkára, scriptek	41-43
24.	Komolyabb script írása	44-47
25.	Programtelepítés parancssorból, fordítás	48-50
26.	Modem	51
27.	Modemes internet, csatlakozás, levelezés, mailserver	52
28.	Hálózat otthon	53-54
29.	NVIDIA driver telepítése X-hez	55
30.	Kernel optimalizálás, fordítás	56-57
31.	A lilo visszaállítása	58-59

ELŐSZÓ, AVAGY MIJAZA LINUX?

Ez a dokumentum neked szól, ha azt fontolgatod, hogy ezt az operációs rendszert szeretnéd használni, kipróbálni, de félsz nekivágni, mert előítéletekkel rendelkezel, illetve tévhitben élsz vele kapcsolatban. Továbbá akkor is, ha komolyabban fontolgatod a használatát, és a számítástechnika is érdekel, meg akarod ismerni a számítógép működését, a saját számítógépedet, meg akarsz ismerkedni a Linux igazi erejével és a szinte határtalan lehetőségekkel, amiket biztosít számodra ki akarod használni.

Te, akinek fogalma sincs, hogy mi is az a Linux, vagy aki a Linux használatát fontolgatod, mert eleged van:

- a vírusokból,
- a fagyásokból,
- az állandó újraindítgatásokból program telepítése után is,
- a lassúságból,
- a „magától" tönkremenő dolgokból,
- „A program szabálytalan műveletet hajtott végre és kilépett a rendszerből" üzenetekből,
- „Általános védelmi hiba, a rendszer leáll" üzenetekből,
- „A rendszer nem stabil. Zárja be..." üzenetekből,
- „A probléma megoldásához lépjen kapcsolatba a rendszergazdával" üzenetekből, holott ez otthon többnyire te vagy,

nos, elsősorban neked szól ez az első fejezet.

A Linux egy, gyakorlatilag az Interneten született, nyílt forrású unix operációs rendszer. Ez azt jelenti, hogy a program forráskódja bárki számára hozzáférhető, nem hétpecsétes titok, bárki átírhatja, fejlesztheti, megváltoztathatja, használhatja. Ennek következménye – nem mellékesen – hogy ingyen van...
A finn származású Linus Benedict Torvalds egyetemista korában megkapta az akkoriban legújabb csúcstechnikának számító 386-os processzorát. Annak képességeit, új technológiáját viszont nem tudta kihasználni egyetlen akkori operációs rendszer sem. Elkezdett hát csinálni egy sajátot. Csak úgy magának. Aztán már az elején fogta, és kihajította a netre, hogy akit érdekel az nézegesse. Nézegették is, és sokan beszálltak a fejlesztésébe. Rohamléptekkel fejlődött, egyre több programozó szállt be, és manapság milliók használják, fejlesztik.

Hogy hogy néz ki? Változatos, nagyon szép grafikus felülete van, ikonokkal, ablakokkal, stb. Windowsban ugye egyféle „kezelőfelület" van (az explorer, bár csináltak egy LiteStep nevűt is, ami nem nagyon ismerős senkinek), ugyanaz az asztal, ugyanaz a menü minden gépen, (ezt hívják ablakkezelőnek) maximum a színeket, ikonokat, háttérképet tudod megváltoztatni. Linuxban sokfajta ablakkezelő van, változatos, egyéni ízlésednek megfelelőt választhatsz. Tehát sokkalta szebb grafikus kezelőfelületek vannak rá, és sokfajta.

Ha egy windowst fel tudsz telepíteni magadnak, akkor egy Linux-ot is. Ha nem, akkor is. Berakod a CD-t, (Pl. UHU-Linux) legegyszerűbb, ha bejelölöd, hogy mindent szeretnél feltelepíteni, és vársz, míg feltelepítődik. Majd a legvégén egyetlen egyszer újraindul, és már használhatod is. Ekkor kb. 1-2 ezer program lesz hozzá feltelepítve, csak győzd végignézegetni.
Teljesen magyarul beszél, akár a windows. Furcsa lesz a telepítésnél, hogy nem kell sokszor sok CD-t cserélgetni, (Videókártya CD-je, hangkártya CD-je, alaplap CD-je, Office CD-je, más programok CD-je, stb.), és nem kell sokszor ujraindítgatni. A végén egyszer, magától újraindul és kész.

Ugyanúgy a "start menüben" találod a programokat, illetve az asztalon az ikonokat. Ugyanúgy csak rákattintasz és megy a program. Ja, itt biztos megy a program.

Biztonságos rendszer.

Röviden: Be kell jelentkezned. Meg kell adnod a felhasználói nevedet és a jelszavadat, és csak azután tudod használni. Más rajtad kívül nem fér hozzá a dolgaidhoz. Több felhasználót is meg lehet adni, ha apu, anyu, testvér, férj, gyerekek is használni akarják. Ők a saját felhasználói nevükkel illetve jelszavukkal tudnak bejelentkezni. Nem kell félni, hogy a gyerek, testvér, stb. véletlenül letöröl valamit, elállítja, tönkreteszi, mert nem tudja. Csak a saját dolgait tudja piszkálni. Hja, te magad sem tudod tönkretenni a rendszert.
Továbbá nem kell félned a vírusoktól, mert nincsenek, illetve a windowsos vírusok nem működnek.
Itt jegyzem meg: Egyetlen rendszergazda van a gépen, a felhasználó neve: root. Jelszavát szintén beállítod telepítésnél. Persze rendszergazdaként SOHA nem szabad és nem is kell használni a számítógépet. Erre csak speciális feladatokkor van szükség, pl. programtelepítés, új hangkártya vagy más hardware telepítése, meg ilyenek.

Stabil rendszer.

Azaz nem kell félned, hogy véletlenül lefagy, mert nem fog. Egyszerre akárhány programot futtathatsz, még CD-t is írhatsz, akár filmnézés, letöltés stb. közben, nem fogja elírni a CD-t.

Mindent meg tudsz vele csinálni.

Ha átlagos felhasználó vagy, akkor többnyire alábbiakat csinálod a géppel: Zenehallgatás, filmnézés, internetezés, chat, szövegszerkesztés, email, játék, CD-írás, képnézegetés. Kb. mindent fel is soroltam, átlagember ezekre használja a gépet.

Zenehallgatás (zeneCD, vagy mp3), windowssal többnyire gondolom a WinAmpot használtad. Itt az XMMS nevű programot fogod. Tuti, hiszen gyakorlatilag teljesen ugyanaz.
Film, (DVD vagy CD, divx, stb.) windowsban lehet, hogy több fajtát is használsz. Itt mindent le tudsz játszani az mplayer nevű lejátszóval. Akármilyen filmet, és DVD-t is. Még a hibásakat is, nem fog akadozni, stb.
Internetezés. Az explorer helyett többfajta közül választhatsz. Van Opera, Mozilla, sőt Netscape is, meg még számtalan egyéb.
Chat. Van többfajta, pl. xchat.
Szövegszerkesztés. Az Office'97, Office'2000 vagy OfficeXP helyett az OpenOffice-t. Szinte ugyanúgy néz ki. Tudja kezelni a windowsos dokumentumokat is. Ez ugyanolyan office, van táblázatkezelője, szövegszerkesztője, adatbáziskezelője, stb. Még ugyanúgy is néz ki, csak van egy-két plusz menüpontja is.
E-mailre is számtalan program van. Ugyanolyan kényelmesek, vagy könnyen használhatóak mint Outlook és társai.
Játék. Sok játék van. A windowsos játékok nem mind mennek, de sok megy. De sok Linuxos játék is van. Komoly játékok is, nemcsak passziánsz meg ilyenek.
CD írás. Szintén sok CD író program közül választhatsz.

A Linux felkínálja a parancssori munka lehetőségét. Van olyan, hogy terminálablak. Mint Windowsban a DOS ablak (Start menü, Programok, MS-DOS parancssor). Ugye itt lehet parancsokat írni és sokmindent csinálni. A parancssor hatalmas lehetőséget kínál, e dokumentum segítségével a használatát el tudod sajátítani.

Persze ahhoz, hogy a Linux-szal elboldogulj, NEM KELL ismerned a parancsokat és a parancssori munkát. Elég, ha tudsz olvasni és kattintani az egérrel. Mint látod fentebb, ugyanannyit kell tudnod, mint Windowsban. Kb. annyira lesz ismeretlen, mint amikor életedben először ültél a Windows elé. Nem tudod, hogy mit hol keress, hova kell kattintani, hol az a program, hol vannak a beállítások. Illetve még annyira sem, csak lesz egy pár újdonság. Pl. hogy néha sokmindent kiír, nem csak egy kék csíkot látsz haladni balról jobbra. Nem titkolja, hogy hogy működik. Igaz, ebből eleinte nem sokmindent fogsz érteni, de ez nem is fontos, lehet, hogy nincs is szükséged rá.

Mikor életedben először ültél a Win9x előtt, minden ment, mindent tudtál?

Naugye. De elkezdtél kattogtatni, majd rájöttél, megtanultad kezelni. Így megy ez a Linuxszal is. A win után annyival könnyebb a dolgod, hogy valamit már sejtesz. Valami beállításokat kell keresni a start menüben, sejted, hogy kb. merre lehet, stb...
Csak más, máshogy néz ki, mutat olyanokat, ami kínai, de legalább kiír bizonyos infót. Meg sokkal több a lehetőség. Fura, hogy nem barmolhatod szét a rendszert, (jogosultságok). Kicsit más az egész de könnyen elsajátítható, ha akarod.

Tehát manapság nem kell szakértőnek és gurunak lenni ahhoz, hogy használd. Ez nagy tévedés. Ha elboldogulsz egy Windows-sal, el fogsz egy Linuxszal is. Továbbá az is nagy tévedés, hogy csak szervernek jó, asztali gépnek nem.

Hja, a Linux és a rengeteg program teljesen ingyenes, nem kell érte fizetni. Boltban is lehet ugyan vásárolni, szép dobozban, könyvecskével, mindennel, tizenezer forintért, de teljesen ugyanazt le is lehet tölteni az internetről ingyen, illetve bárki másolhatja, továbbadhatja. A sok program, játék mind mind ingyenes. Ezzel szemben a windows XP legalább 50,000.-Ft, az Office XP kb 50-100.000,-Ft, és még a sok egyéb program... pontosan nem tudom, az árakat, nem is érdekel, csak tippeltem.

Röviden ennyi, remélem mostmár valamennyire el tudod képzelni, hogy miről is van szó. Ha kedvet kaptál, használd, szerezz be pl. egy UHU-Linux CD-t, akárhonnan, internetről is letöltheted (www.linuxforum.hu, www.uhulinux.hu), rakd fel, próbáld ki, használd. Ha komolyabban akarsz foglalkozni vele, akkor olvass tovább.

A parancssori munkát sem nehéz elsajátítani, csak venni kell a fáradtságot, és tanulni, gyakorolni, olvasni. Gondolom a DOS használata sem ment az egyik pillanatról a másikra... ha egyáltalán használtál DOS-t. Csak meg kell tanulni a parancsokat. Ehhez pedig csak gyakorolni kell, hogy megmaradjon.

Ha ezt a dokumentumot elolvasod, gyakorolsz, a példákat pötyögöd, a feladatokat megoldod, akkor menni fog a parancssori munka is, és kezded megismerni a rendszert, a lehetőségeit. Ehhez viszont szükséges, hogy legyen egy Linuxod, továbbá egy nyitott terminálablakod ahol gyakorolsz, hogy lásd is amiről szó van, mert úgy lehet tanulni.
Tehát a továbbiakban parancssoros kezelést tanulunk, mert a GUI-t (Graphical User Interface, azaz Grafikus Felhasználói Felület) nem érdemes. Kattogtatni mindenki tud.

A LINUX ÉS ÉN - A KEZDETEK

Én tudod hogy kezdtem? Kaptam kb. 1999-ben egy Mandrake 7.0 CD-t, és haver belőtte nekem. Még nagyon döcögős volt, ugyanis, akkor még nem minden volt autodetect, nemcsak berakom a telepítő CD-t, katt, katt és kész.

Szóval felment, de nem minden ment, amit én csináltam akkoriban a winnel, illetve bonyolultnak találtam. Persze még fent volt a win. De még nem nagyon volt magyar nyelvű Office sem, az xawtv nem húzta szét teljes képernyőbe a TV-t, hanem középen nagy széles fekete keretben ment, mert nem tudta kifeszíteni, stb., úgyhogy nem nagyon foglalkoztam vele. De eltelt egy év, és besokalltam. Mármint a wintől, a fagyásoktól, a túl rövid uptimetól (3 nap, és muszáj reboot a winnél). Na ekkor szóltam havernak, Linux friss feltelepítés, meg vettem új, nagy vinyót, (40G volt akkoriban) Rakjunk rá Linuxot, de már újat, és mondtam, hogy lője be ami nem akar véletlenül menni. Hja, magyar doksik sem olyan nagy számban voltak. Fel is ment a Linux, mondom játék miatt a végére rakjunk egy wint. Lamerek voltunk mindketten, a win nem megy fel asszem az első 8Gb-on túl (1024 cylinder). Így hát nem lett win. Rá voltam kényszerítve a Linuxra, de - gondoltam - mostmár inkább nem játszok mindennel és megtanulom kezelni, használni.
Ekkor olvasgattam az angol doksikat, és a papírra leírt parancsokat gyakorolgattam. Meg cimborám magyarázott itt linkekről, meg scriptekről, meg átirányításokról, de nem úgy, mint én itt fogok, hanem úgy, hogy ilyen is van, és ha érdekel nézz utána, meg valami számomra még majdnem kínai magyarázatokat adott néha. Meg állandóan azt hajtogatta, ha valami bánatom van ne zaklassam, meg hogy ezt már kérdeztem és hogy RTFM. Ez egy mozaikszó, a Read The Fucking Manual szavak rövidítése (Olvasd El A Ki*aszott Kézikönyvet)... Ez azért van, mert a Linux agyon van dokumentálva. Csak nagyon sok az angol doc, és még magyar szinte sehol nem volt akkoriban. De manapság már sok a magyar is. Azóta az évek során begyûjtöttem kb 130 MEGA magyar doksit folyamatosan. Ha egy parancsról infó kell (milyen kapcsolót kell megadni, hogy ezt meg ezt meg tudd csinálni, akkor ott a manual:

man parancs

pl.:

man ls

és már ad is pár oldal doksit az ls programról.

Vagy, ha tudod már kb., hogy mit akarsz, akkor:

parancs --help

Ekkor ad egy rövidebb helpet.

Szóval telt az idő, és RTFM-eltem, mint állat, ugye végre vadásztam magyar doksikat is.
Elmagyarázta haver, hogy hogy kell lefordítani valamit, leírtam papírra, és gyakoroltam. És nézegettem a rendszert, mi hol van, mik ezek a fura könyvtárak, stb. Namost, neked sokkal könnyebb dolgod van/lesz, mint nekem volt.

GPL, GENERAL PUBLIC LICENCE

Maga a Linux és a programok majdnem 100%-a szabadon terjeszthető, másolható. Pénzt munkadíjként lehet csak kérni, pl. másolás, de nem a programokért és a Linuxért.
Szerzői jog védi természetesen, ez a General Public License (GPL), mely licence szerződés számodra legérdekesebb lényege, hogy ha pénzt kér érte valaki (MAGÁÉRT A LINUXÉRT ill. a nyílt forrású, GPL által védett programokért), AKKOR követi el a Szerzői és szomszédos jogok megsértése bűncselekményt... A GPL lényege, és a Linuxé is, hogy nyílt forrású, azaz magát a nyers programforrást – mely legtöbbször C nyelven íródik – terjesztik, amit szabadon lehet továbbterjeszteni, módosítani, stb. Természetesen előre lefordított formában (futtatásra készen) is terjesztik, pl. rpm, deb, uhu, stb. csomagok (wineseknek: a la setup.exe)
Ha nem vagy biztos valamiben, akkor csak nézd meg a letöltött/beszerzett progi licenseszerződését, ha látod benne azt, hogy GPL, akkor lehet vinni ingyért, nyugodtan másolhatod, használhatod, terjesztheted. Ilyen egyszerű ez.

Hja, felmerülhet a kérdés, hogy ha már ingyen lehet vinni, meg a programokat is másolni, átírni, akkor minek a szerzői jog egyáltalán? Miért van erre szükség? Egyszerű. Mert eleinte nem védte szerzői jog a nyílt forrású, ingyenes programokat. És számtalanszor megtörtént, hogy a programozó, aki szabadidejében (többnyire magának) megírt egy jó programot, rengeteg munkát, szabadidejét belefektetve, és ingyen közreadta, hogy segítsen másokon, bizonyos cégek/személyek „ELLOPTÁK” a programot (forráskódot). Ez úgy történt, hogy a cég átírta a program forrását, vagy kódrészletet szedett ki belőle és beleírta a saját programjába, majd levédte, és pénzért árulta egy általa kitalált néven. Csak az a baj, hogy azt valójában nem ők írták, az egy olyan program volt, ami addig ingyenes volt, nyílt forrású. És ugye innentől fogva az ingyenes, nyílt forrású software pénzes, zárt forrású software lett. A pénzt (jogdíjat) a cég zsebelte be, és nem az, aki dolgozott vele. Ha nem lenne nyílt forrású, azaz a szerző nem adná közre a program forrását, csak a lefordított, már futtatható programot, azt már nem, vagy csak nagyon nagyon nehezen lehetne visszafejteni, hogy át lehessen írni, bele lehessen nyúlni. Mivel egy software úgy fejlődik a legjobban és a leggyorsabban, ha más, hozzáértők hibát (bug) találnak benne akkor szólnak, esetleg még be is segítenek, ezért nyíltnak kell lennie a forrásnak, hogy a másik programozó már ki is tudja javítani, módosítani, illetve besegíteni a fejlesztésbe. Tehát az nem jó megoldás, hogy a forráskódot nem adom, csak magát a programot rakom ki a netre ingyen elvihetőként (zárt forrású, ingyenes, azaz freeware program), mert nagyban hátráltatja a fejlesztést, más nem tud segíteni. Viszont így védtelen az ellen, hogy „ellopják”.
Erre kellett kitalálni valamit, ezért jött létre a GPL, hogy megvédje a nyílt forrású programokat és készítőiket.

Erről jut eszembe, a „Bízd a hackerre” című film nagy igazságokat tartalmaz, nem egy teljesen kitalált fikció... sajnos van valóságalapja...

Enyhén szólva kissé leegyszerűsítve magyaráztam el a GPL-t de ebből már érthető a lényeg.

TÖREDEZETTSÉG-MENTESÍTÉS

3 és fél éve, mikor még winnyózom volt, akkor hallottam utoljára ezt a szót, most hogy felemlítjük, kezd derengeni valami...

Linuxos filerendszereket NEM KELL töredezettség-mentesíteni, ezt a fogalmat akár el is felejtheted.

A DISZTRIBÚCIÓK ÉS AZOK MÉRETE

Namost tisztázni kell a különbséget a Linux, és a Linux disztribúció fogalmak között.

Valójában maga a Linux alig 1MB, többnyire csak 700, 800 KiloByte, hogy egy kicsit összezavarjalak. A Linux valójában maga a kernel (magyarul: rendszermag). Persze azzal még nem tudsz mit kezdeni progik nélkül. De minden más csupán programok tömege. Egy másik nyílt forrású Unix rendszer is csak kb. a rendszermagban különbözik a Linux-tól. Ugyanazok illetve hasonló progik, parancsok vannak (lehetnek) melléfordítva. Persze nem TELJESEN csak ennyi, de a lényeg és az érthetőség kedvéért...
Tehát a Linux az a rendszermag, de köznyelven a rendszermag és egy csomó program összessége, azaz amit éppen használsz.

A disztribúció (terjesztés) pedig nem más, mint a Linux és sok program összeválogatva egy CD-re (vagy többre). Bizonyos cégek (Mandrake, UHU, SuSe, stb...) fogják a Linuxot (ugye a rendszermag), és rengeteg programot, összevadásszák (helyetted) a NET-ről, és írnak hozzá egy grafikus telepítőprogramot (nem kézzel kell telepítened a programokat egyenként), csak kattintanod kell párat és magától feltelepülnek. Meg írnak hozzá szép grafikus beállító programot, hogy könnyebben tudd beállítani (mint a windowsban a vezérlőpult). Ezeket CD-kre pakolják, és boltban árulják, plusz kirakják a CD-ket az internetre ingyen letölthetőként.

A disztribek tizenezer forintért boltban megkaphatók, de ott sem a Linuxért és a sok progiért kérik a lét, hisz' az bűncselekmény, hanem az összeállításért (disztribúció). A szép dobozért, a könyvecskéért, doksikért, a CD-k anyagáráért, és a belefektetett munkáért, meg saját programokért, pl. a telepítőprogramért, a beállító programért (menudrake, harddrake, yast, UHU-vezérlőközpont, stb.) és a support-ért (támogatás).

Ha a wint feltelepíted, az ugye még semmire sem jó. Arra még ugye telepítesz WinAmp-ot, office-t, képnézegetőt, médialejátszót, kodeceket, meg még istentudja mit, istentudja honnan összebogarászva.
Mikor egy disztribet feltelepítesz, legyen az UHU, Mandrake, akármi, akkor mindezt megcsinálja helyetted a gép, te csak bejelölöd, hogy mit akarsz feltelepíteni, szinte minden rajta van a telepítő CD-n. Nem kell innen-onnan összeszedni, külön feltelepíteni. Illetve nagy eséllyel megtalálható a CD-k valamelyikén az a progi, amit fel akarsz rakni. Ezért olyan nagy a "Linux".
És ugye mikor win alá is feltelepíted azt a sok progit, akkor kompletten az is több giga is lehet, nem igaz?
Úgyhogy ha így vesszük, akkor természetes dolog a Linuxos disztribeknél, hogy 1-2, vagy több gigás lesz a "Linux", hiszen pl. a Mandrake 9.0-hoz (ha jöl dereng) több, mint 5.000 programot mellékelnek... Nem csoda, ha a full telepítés több gigabyte...

A LINUX ERÕFORRÁSIGÉNYE

Intel Celeron 433Mhz processzor, 128MB RAM.

Nem túl sok, manapság a 3000Mhz-es procik korszakában már elég gyenge gépnek számít.
Filmek, nagy felbontású divx, DVD, Office, játékok stb. Természetesen megy rajta minden. Nyilván, ha a windowsos játék 2GHz-es procival, rakat memóriával szaggat, akkor ezen a kis gépen el sem indul, de pl. Starcraft, Half-Life, Medal Of Honor, Return To Castle Wolfenstein qrvajól elfut. Továbbá a 2D-s sok 3D-s Linuxos játék is.

Ha Linuxod van ezen a kis gépen, akkor nagyon jól lehet CD-t írni, internetről letölteni, mpeg-et kódolni divx-be, CD olvasóban lévő audió CD-t grabbelni mp3-ba. Namost - mondhatnád - ezt még Windowssal is meg tudom csinálni ezen a gépen.
Viszont most jön a lényeg: Mindezt EGYSZERRE, egy időben.

Hihetetlen?

Pedig ez volt régebben a konfigurációm, úgyhogy tapasztalatból mondom. Sőt, dobok rá még egy lapáttal: MINDEKÖZBEN, hogy ne unatkozzak, DVD minőségű filmet néztem a vinyóról... EZEN a gépen. És a film sem akadozott.

A Linux 386-ra íródott. Tehát ha nem kell neked KDE3, Gnome, meg ilyen erőforrás-zabáló cuccok, (van számtalan más gyönyörű grafikus ablakkezelő, pl. WindowMaker) akkor a fenti kis 433Mhz-es Celeron-nál kisebb gépen is elmegy, sokkal jobb sebességgel, mint egy Windows.

VERZIÓSZÁMOK

Megjegyzésként elöljáróban elmondom, hogy Linuxban is, mint máshol az újabb verziójú program nem biztos, hogy jobb. (Többnyire, 99%-ban persze igen.) Élő példa a gcc, (GNU C Compiler) azaz a C fordító program. Nem mindig kell ész nélkül frissítgetni. Ha valami nagyon jól megy, akkor nem muszáj csak azért lecserélni, mert az újabb.

Pl. a gcc (és bármely progi) verziószámát többnyire a "-V" opcióval (NAGY v betű!), illetve a --version opcióval lehet megnézni. Tehát nyiss egy terminálablakot, és írjad:

gcc --version majd ugye ENTER.

Sz'al mint látod, Linuxban 3 számból áll a verziószám. Pl.: 3.2.2

Namármost az első szám a fő verziószám, ez a példában 3-as. A másodiktól kezd érdekes lenni, ez, ha PÁROS, akkor stabil verzió, ha páratlan (pl. 3.1.2), akkor nem stabil, teszt, béta, fejlesztői, vagy akárhogy hívjuk. A harmadik kb. annyit jelent, hogy hanyadik változat.

WINDOWSOS PROGIK, JÁTÉKOK

A Windows emulátor (wine, wineX) már elég sok windowsos progit le tud futtatni, DE NEM MINDET.
Tehát ha neked kell office (word, excel, rajzoló, powerpoint stb) akkor ott az Openoffice, ami ugyanaz, csak kicsit többet is tud a dózerosnál. Természetesen tökéletesen kezeli a Word 6. 97, 2000, XP, stb. doksikat is. Ha kell WinAmp, ott az XMMS. Tök ugyanaz, csak a neve nem. Ha kell akármi, ott a Linuxos megfelelője. Ha ezek után mégis valami wines progit akarsz futtatni, az gondolom a játék. Megy Starcraft, Half-Life, stb, az emulátorral. Rögtönzött lista:

http://www.lokigames.com/products/

Játékok felsorolása:
Postal Plus
Rune: Halls of Valhalla
Tribes 2
Sid Meier's Alpha Centauri with Alien Crossfire expansion pack
Heavy Metal: F.A.K.K.2
MindRover
Rune
Kohan: Immortal Sovereigns
Deus Ex
Soldier of Fortune
Unreal Tournament
Descent3
Civilization: Call to Power
Heavy Gear II
SimCity 3000 Unlimited
Eric's Ultimate Solitaire
Heretic II
Heroes of Might and Magic III
Myth II: Soulblighter
Railroad Tycoon II

Ezek biztosan mennek még Linux alatt, és nem a gagyi játék kategória. Továbbá.
Linuxon, nem emulátorral, hanem rendesen pl. Return to Castle Wolfenstein, Medal of Honor, hogy egy-két kedvencemet is említsem.

A Linuxos átiratot ne úgy képzeld el, hogy ott a boltban a polcon a játék, és rá van írva, hogy ez a Linuxos. (Bár nem lehetetlen, hogy ilyen is van.) A síma windowsos játékot megveszed, és letöltöd/beszerzed a Linuxos átiratát, ami csupán egy bináris (futtatható állomány). Ezek kicsi file-ok, nem kell megilyedni. Nem az egész játékot kell átírni, hanem csak a futtatható részét kell a programozóknak átportolni, a többi ugyanaz.

Továbbá Doom 1-2 Linuxos verzió, hogy a régiek közül is legyen itt példa. Quake-ek, Linuxos verzió, nem emulátorral. Unreal Tournament Linuxos verzió, nem emulátorral.

Vannak továbbá a kifejezetten Linuxos játékok. Ott van pl. a Tux Racer, Cromium (űrhajós, lövöldözős), TORCS (autó szimulátor) FlightGear (Repülő szimulátor) America's Army, Enemy Territory, Urban Terror (fps), Foobillard (3D biliárd) stb. mint 3D-s gamék, nem éppen a gagyi kategória, akkor pl. Pingus (Lemmings klón), akkor Civilization klón, stb, stb... csak néhány kiragadott példa… Hja, ezek is természetesen ingyenesek.

Végkövetkeztetés: AZ nem tud játszani Linuxos számítógépen aki NEM AKAR.

GUI (Graphical User Interface), AZAZ GRAFIKUS FELHASZNÁLÓI FELÜLET, X SZERVER, ABLAKKEZELÕK

Természetesen a Linuxnak van grafikus felülete, mint a winnek, és a mai disztribúciókban természetesen automatikusan szép grafikus képernyőkön dolgozunk. Ez az alaphelyzet. Tehát a Linux „kinézete” nagyon szép, sokkal szebb mint akármelyik dózeré.

Lássuk viszont a részleteket, miről is beszélünk. A grafikus felületet nem azért találták ki, hogy egerészni és kattogtatni tudjon az emberfia, hanem azért, mert vannak grafikus programok, melyeknek értelemszerűen grafikus környezetet kell teremteni.

Az ABLAKKEZELŐKET találták ki azért, hogy kattogtatni lehessen.

Na, most jól megkavartam a dolgokat, mi?

Szóval a grafikus kezelő, mint szinte minden a Linuxban szerverként működik. Ezért hívjuk grafikus szervernek, másnéven X szervernek a grafikus kezelőfelületet. De, úgymond "szűzen" az X-et, ha elindítanád, egy natúr, fehér alapon sűrű szürke pöttyözött, teljes képernyős grafikus felületet látnál, egyetlen nagy X egérpointerrel, amivel így símán semmit sem tudsz kezdeni.
Kattogtathatsz, pötyöghetsz a billentyűzeten, semmi. Ez csak egy grafikus felületet ad, egy grafikus környezetet teremt, melynek segítségével tudnál grafikus alkalmazásokat használni.
De kissé kellemetlen, nem felhasználóbarát lenne, ha minden egyes programot csak konzolon pötyögve tudnál elindítani. Ezért találták ki az ablakkezelőket. Windows alatt többnyire csak egy van, ezt úgy hívjék, hogy Windows Explorer (nem összetéveszteni az Internet Explorerrel!) De emlékeim szerint csináltak egy másikat, aminek neve LiteStep, de ezzel a legtöbben nem találkoztak a win felhasználók közül. Linuxra azonban nagyon sokféle ablakkezelő van, pl. Afterstep, IceWM, KDE, Gnome, Blackbox, WindowMaker és még számtalan fajta, ízlések, pofonok és hardvarekapacitás kérdése. Olyat választasz, amilyet csak akarsz, nem csak egyféle van, és ha tetszik, ha nem azt használod és kész, mint a dózerben.

Szóval úgy működik a rendszer, hogy elindítja a grafikus szervert, és a konfig filejában (.xinitrc) meghatározott programokat, legfőképpen valamilyen ablakkezelőt. Így te csak azt látod, hogy szép grafikus képernyő jelenik meg előtted, ikonokkal.

A PARANCSSOR ÉS A GUI

Természetesen a mai Linux disztribek a parancssor teljes kizárásával is működnek, nem úgy, mint a régi időkben. Úgyhogy akit nem érdekel komolyabban a rendszer, annak lehetőségei, csupán szeretne egy biztonságos, stabil rendszeren elvegetálni, annak nincs szüksége a parancssorra. Más kérdés, hogy a parancssor használatának lehetősége mindig adott. Csak már szép grafikus felületen, ezt pedig terminálablaknak hívják.
Szinte mindent sokkal gyorsabban és hatékonyabban lehet elvégezni, parancssor segítségével, és rengeteg lehetőséget lehet kiaknázni, amit egy GUI kezelőfelület képtelen biztosítani. Nem azért, mert lusták a programozók és nem csinálják meg, hanem azért, mert lehetetlen, képtelenség, és nincs értelme sem. Minden egyes kis parancs, pl. a legalapvetőbbek, mint az ls (könyvtár listázás) is egy egy különálló program, aminek rengeteg kapcsolója van. Képtelenség, és leginkább felesleges GUI kezelőfelületet írni hozzá. Pl. írd be:

```
ls -h
```

és csodálkozz, mennyi kapcsoló van, mennyit tud a kis ls parancsocska.
Pl. ott a cdrecord.
Ez egy GUI (grafikus felület) nélküli, parancssoros cd író program, és megfelelő kapcsolókkal, paraméterekkel lehet CD-t írni vele.
Nos, ehhez pl. létezik GUI, a neve pedig gcombust. Illetve van más is, én ezt használom.
Visszatérve, a legtöbb programnak rengeteg kapcsolója van. A parancsok nem mások, mint programok, amiket a program nevének beírásával, és ha szükségünk van rá, esetlegesen paraméterezéssel lehet futtatni. Ha a grafikus kezelőfelületét nézed mondjuk a cdrecordnak úgy, hogy nem ikonról, hanem parancssorból indítod a gcombust CD-író programot, (megnyitod a terminálablakot és beírod, hogy gcombust, majd enter) és át-át pillantasz a terminálablakra (ALT+TAB, mint dózerben), akkor rájössz, hogy a jól ismert kis kockákba kattintva kipipálva az történik, hogy a cdrecord program paramétereit aktivizálod, csak nem beírással, hanem kattintással. Pl. a legördülő menüben kiválasztod, hogy 8x-osan akarsz írni. Ez gyakorlatilag annyit csinál, hogy megadja a cdrecordnak a "-speed 8" paramétert. Ez is szépen látszik a terminálablakból indított gcombust esetén.

De pl. a cdíró program egy olyan program, aminek van értelme grafikus kezelőfelületet írni. De a legtöbbnek (pl. az ls) nincs. Hisz az ls-nek is életünk során kb. egy-két kapcsolóját használjuk, illetve kissé macerás és időigényes lenne kattogtatva különböző módszerekkel listázni a könyvtárakat...

A PARANCSSOR HATÉKONYSÁGA

Egymás mellett 2 gép, egyiken win, másikon Linux. Haverral egyszerre Matrix (feliratos) CD berak, én a nyitott terminálablakba beírtam:

mplayer /mnt/cdrom/Matrix.avi

A TABULATOR gomb használatával ez pillanatok műve, ugyanis a tabulátor gomb intelligens szövegkiegészítést csinál. Tehát a fenti sort kb. 5 másodperc beírni, ha lassan gépelsz és használod a tabulátor billentyűt. Az mplayer pedig auto betöltötte a sub-ot, feliratfile-t (ha nem ugyanaz a neve, akkor nem, akkor külön meg kell adni neki).

Ezzel szemben a haver katt a BS Player ikonjára, megnyílik... Katt a file gombra... Na kb. eddig jutott, én már néztem a filmet. Ezután neki még hátra volt, hogy megnyitás egérrel csúszka-húzogatás, tallózgatás, stb.

Persze ha Linux alatt én GUI-s (grafikus) mplayert indítottam volna, abban ugyanúgy egerésznem kellett volna, csúszkát húzogatni, stb., ugyanaz a helyzet ebből a szempontból, mintha wint használnál.
Nálam pl. mindig előttem van egy terminalablak. Például sokszor - igaz, hogy kint van az xmms (alias WinAmp) ikonja is az asztalon, de - mikor berakom az mp3cd-t, beírom, hogy "xmms /mnt/cdrom/*" és már bent is az összes mp3 a filelistában, úgy nyílik az xmms. Mennyivel gyorsabb, mint katt, megnyílik, filelista törlése, tallózás, kattogtatás stb.

Ha az ember az egeret arra használja, amire eredetileg ki lett találva, akkor furamód tud dolgozni is. Egyszer-s-mindenkorra, a számítógép fő adatbeviteli eszköze és irányító perifériája NEM az egér, hanem a BILLENTYŰZET. Az egérrel való munka (kivéve, amire kitalálták, pl. rajzprogramok, játék, stb. stb...) iszonyatosan lelassítja a munkát.
Ezért is parancssori munka tanulása közben próbálom bemutatni a rendszert. Tanuljunk egy kis számítástechnikát is. Nézzünk mélyebben a rendszerbe, ismerjük meg a lelkivilágát. Mostanra már sokmindent megtudtunk, kipróbálgattunk a rendszerünkön, kb. értjük, tudjuk hogy gőzgép, de nézzük meg hogy ugyan mi hajtja.

FILE HIERARCHIA

Egy kis alapozás. Sajnos sokan még csak DOS-t sem láttak, és – köszönet és hála Bill Gatesnek, áldassék a neve – a SZÁMÍTÁSTECHNIKÁRÓL halovány fogalma nincs a win felhasználóknak, ráadásul ennek ellenkezőjét HISZIK.

Azt sem tudják, hogy mi az a file hierarchia. A lényeg az, hogy a winchesteren az adatok, fileok hierarchikusan tárolódnak. Minden file egy könyvtárban van. Namost a legfőbb könyvtár a főkönyvtár. Minden itt található, ebből ágazik el. Olyan, mint egy fa, ezért van, hogy fa struktúraként is szokták emlegetni, és fejreállított faként kell elképzelni mint pl. a családfát. Azaz inkább mint a fa gyökerét. Pont ezért a főkönyvtárat szokták gyökérnek vagy gyökérkönyvtárnak is nevezni. Ezért is lett a rendszergazda neve Linuxban root, azaz gyökér, de NEM pejoratív értelemben, hanem szimbólikusan, mert ő a fa gyökere, mindenhez és minden könyvtárhoz hozzáfér. A gyökérkönyvtár winnél és DOS-nál a c: volt. Ez volt a jele. Ebben vannak könyvtárak, azokban alkönyvtárak, megint alkönyvtárak... és persze maguk a fileok a könyvtárakban. DOS-nál és winnnél, mikor az elérési utat adjuk meg, a \ jellel választjuk el a könyvtárat. Linuxban pedig a / jellel, akár az internetnél. Annyi még a különbség, hogy Linuxban a gyökérkönyvtár jele is a / jel, hogy jól bele lehessen kavarodni.
Tehát winben meg DOS-ban így írnánk:

c:\akarmi\valami.txt Linuxban pedig:

/akarmi/valami.txt

Tehát, ha parancs után, vagy hivatkozásnál, ha ELŐRE rakunk / jelet, akkor az a gyökérkönyvtárat jelenti. Ha nem rakunk előre / jelet, akkor az azt jelenti, hogy az aktuális könyvtáron belül, ahol épp állunk.
Tehát, ha a home könyvtárunkban állunk, azaz a /home/pistike könyvtárban, akkor ha ki akarjuk listázni pl. a doc könytárunkban (ami itt van, a /home/pistikében) lévő telefonkonyv file-t (feltéve, hogy van doc nevű könyvtárunk, és abban ilyen nevű dokumentumunk), akkor így írjuk (az ls parancs segítségével):

ls doc/telefonkonyv

Ilyenkor a rendszer TUDJA, hogy éppen hol vagyunk, és azt – úgymond – automatikusan “eléírja”, nekünk nem muszáj. Tehát mintha azt írtuk volna, hogy

ls /home/pistike/doc/telefonkonyv

Próbáljuk ki, így is menni fog, természetesen.
Ugyanis minden oprendszer úgy dolgozik, hogy csak TELJES elérési úttal “látja” a fileokat, ill. könyvtárakat.

Gyakoroljunk! Nyissuk meg a terminalablakot. Ugye ilyenkor automatikusan a saját home könyvtárunkban állunk, és villog a cursor. A rendszeren ez az a hely, ahol dolgozunk, ahova a cuccainkat pakoljuk, szerkesszük, stb., azaz ahol dolgozunk. Tehát mintha ez lenne a „c: meghajtónk”.

1. Nézzük meg, hogy mi van a saját home könyvtárunkban. Listázzuk ki. Írd be, hogy ls, majd bökj egy entert. Nem nagyon sok minden lehet, ha még nincs sok cuccod (doksi, kép, film, zene, stb.)
2. Hozzunk létre egy doc könyvtárat, ha nincs, a dokumentumaink tárolására. Írjad: mkdir doc (Parancs után mindig ENTER-t nyomunk, ezt mostmár nem fogom mondani)
3. Újra listázzuk ki a home könyvtárunkat az ls paranccsal, láthatjuk, hogy ott a doc könyvtár. Lépjünk bele a cd parancs segítségével. Írjad: cd doc

4. Hozzunk létre egy telefonkonyv nevű dokumentumot ide, a doc könyvtárunkba, és írjunk bele legalább 4-5 nevet és mellé telefonszámot! Ehhez most használjuk a joe nevű szövegszerkesztőt. Használatát később tanuljuk részletesebben. Írjad: joe telefonkonyv
   Ekkor megnyitja a még üres telefonkonyv nevű dokumentumot (hiszen még nincs ilyen). Írjunk bele pár nevet, telefonszámot. Mentsük el a Ctrl+K, majd egy X gomb megnyomásával! Először a Ctrl billentyűt nyomva tartjuk, és nyomunk egy K betűt. Majd elengedjük a két gombot, és rábökünk az X gombra. Tehát ha az van írva, hogy Ctrl+K, az azt jelenti, hogy egyszerre kell lenyomni.
   Ha kiírta, hogy „File telefonkonyv saved.” Akkor jól csináltunk mindent.

Na, 4 parancsot máris tudunk. ls-sel listázunk, cd-vel mászkálunk a könyvtárak között, és mkdir paranccsal könyvtárat tudunk létrehozni, a joe-val meg szöveget szerkeszteni. Menjünk vissza a home könyvtárunkba a cd paranccsal. Pl. cd /home/pistike
Most próbálgassuk az ls parancs segítségével a fenti példákban szereplőket.

Ja, sok programnak nincs olyasmi kimenete, mint pl. az ls-nek, hogy ugye várunk tőle pl. egy listát. Lásd pl. az mkdir-t. Biztosan észrevetted, hogy nem írt ki semmit, miután beírtad, hogy mkdir doc, és böktél egy entert. Kicsit szokatlan lehet a win után, hogy ha valamit megcsináltál, nincs taps, fütty, ováció, húdeügyesvagy, stb...
A Linux olyan, mint a tökéletes "alkalmazott". Szó nélkül maradéktalanul végrehajtja a parancsokat. Kicsit szűkszavú. Nem ugat feleslegesen, csak azt csinálja amit mondasz neki. Viszont ha hülyeséget csináltál, illetve gond van, akkor azt közli.

Szóval miután egy parancsot beírtál, és entert ütöttél, ha nem ír ki semmit, akkor az annyit jelent, hogy végrehajtotta, minden a legnagyobb rendben. :)
Miután ezeket a feladatokat sikeresen megoldottuk átverekedtük rajta magunkat kedhetünk ráérezni az ízére... Pl. milyen gyorsan létrehoztunk egy könyvtárat. Nem kellett előbb az egérrel új mappa ikonra kattintani, majd bele még egyet, billentyűzeten bepötyögni, majd egéret fogni újra, rákattintani, stb.

NÉHÁNY LINUX PARANCS (és egyebek) RÖVID LEÍRÁSA

Most, hogy kezdünk megbarátkozni a parancssorral, és hogy tudjunk dolgozni, tanuljunk meg néhány parancsot. Ehhez nyissunk egy terminálablakot, és a példákat pötyögjük szorgalmasan, gyakoroljunk, játszunk velük. Hja, ugyanúgy, mint a windowsban, az ALT+TAB gombok használatával gyorsan és könnyedén tudunk terminálablak és az office (ha azzal olvasod ezt a dokumentumot) között váltani. Na, lássuk. Tehát elolvasol egy parancsot, leírását, utána pötyögjed befele, majd ENTER.

ls – (l)i(s)t, azaz lista, könyvtár listázása

Mondom írjad. Mondo... nem érted?! Azt mondtam írjad! Tudom, hogy most csináltad az előbb, de azt mondtam, hogy írjad és kész. Nem sumákolunk!

ls -l – (l)ong, azaz "hosszú", teljes listát ad, Ez már érdekesebb.
Jogosultságok, file tulajdonosa, csoport, dátum, fileméret stb.
Majd tanuljuk, hogy mi micsoda.

file – fájlról megállapítja, hogy milyen tipusú

Szintaxisa (használata): file filenév

pl: file doc/telefonkonyv
file /home/pistike/doc/telefonkonyv

Ha nem pistike a felhasználó neved, és pistikét írsz, megharaplak!

Értelemszerűen, ha lehetne...

Szóval ekkor megmondja, hogy a telefonkonyv egy natúr szöveges dokumentum. Linuxban felesleges is a kiterjesztés használata. A Linuxot az marhára nem érdekli. Szokjunk is le róla! A KITERJESZTÉS HASZNÁLATA CSAK NEKÜNK FONTOS, MERT MI NEM TUDJUK, HOGY MI IS AZ. De gondolom pl. a telefonkonyv nevű fileról is tudjuk, hogy micsoda anélkül, hogy oda lenne írva a végére, hogy .txt. Mostmár nem fogok ilyet írni, hogy elérési úttal. Legyen mostmár ez magától értetődő, hogy bármit megadhatunk teljes elérési úttal, ill. ha a fenti telefonkonyv file mondjuk a /home/pistike/valami könyvtárban lenne, mi pedig éppen a home könyvtárunkban (a /home/pistike könyvtárban) állunk, akkor nyilván meg kell mondani a rendszernek, hogy az nem itt van, hanem a valami nevű könyvtárban.
Tehát mi a /home/pistike-ben vagyunk, akkor elég INNEN tovább indulni:

file valami/telefonkonyv (mint fent, file doc/telefonkonyv)

Kezdő koromban sokat szívtam vele, hogy a példánál maradva így írtam a hivatkozást:

file /doc/telefonkonyv

Feltűnik a hiba? Először nem értettem miért írja ki, hogy nincs ilyen file vagy könyvtár. (a "/" a sor elején, vagy elérési út elejére írva ugye a gyökérkönyvtár jele, (winben és DOS-ban a c:\ volt) és ott nyilván nincs doc könyvtár.)

ln – (l)i(n)k linkek létrehozása – majd tanuljuk, hogy mire jó
Szintaxisa (használata): elsősorban soft link, ln -s mit hova

pl: ln -s /home/pistike/doc/telefonkonyv /home/pistike/doc/telefonkonyvlink

Írjad ezt is. Majd listázd ki a doc könyvtáradat. A linkelést később tanuljuk részletesen.

du – (d)isk (u)sage, lemezhasználat, megmondja, hogy az adott file/ok/ vagy könyvtár/ak/ mennyi helyet foglalnak a lemezen.

szintaxisa: du micsodáé

du paraméter nélkül kiadva, file vagy könyvtár megjelölése nélkül az éppen aktuális könyvtár méretét mondja meg
du -h (h)uman, azaz "emberi" formátumban adja meg a méretet, (megabite, gigabite, kilobite)

pl. du -h /home/pistike – pistike homekönyvtárának méretét mondja meg

whatis – mi az az... egymondatos információ,

használata: whatis program_neve
pl. whatis ls (mi az az ls, és megmondja)

echo – "visszhang", kiírja a paraméterként megadott szöveget.

használata: echo "szöveg"

Írjad: echo "Írjad!"

Szignálok (mert most jutott eszembe, azért írom ide.

ctrl+c kill szignál, megöli az éppen előtérben futó processt (programot)
ctrl+d filevége szignál, befejezi a program futását
ctrl+z megszakítja (nem lelövi!) a program futását (pillanatmegállító)

Majd tanuljuk, hogy mikor mire jó.

grep – nagytudású keresőprogram, szövegfileban, vagy fileokBAN keresünk vele.
NEM rekurzív, magyarul alkönyvtárakban NEM keres! De ha akarjuk, megfelelő paraméterrel, ugye mindent... (ez a -r paraméter)

Szintaxisa: grep mit_keressen hol_keresse

pl.: grep Tóth doc/telefonkonyv

A -i paraméterrel nem veszi figyelembe a kis és nagybetű különbséget, pl.

grep -i tóth doc/telefonkonyv

megtalálja a Tóth, TÓTH szavakat is, nemcsak a tóth-ot.

find – talál, keresőprogram, fileokAT keres, talál. Vagy nem talál, ugye...
REKURZÍV, azaz alkönyvtárakban is keres.

Általában használatos szintaxisa: find hol_keressen -name filenév

pl.: find /home/pistike -name telefonkonyv – megmondja, hogy talált egy ilyen nevű file-t a doc könyvtárunkban

Azért általában, mert általában egy file-t keresünk vele. Persze Joker karaktert is lehet használni:

find /home/pistike -name telefo* (megtalálja a „telefo” kezdetű fileokat. Most leginkább csak a telefonkonyv-et, mert más nincs.

tar – file összefűző, annyit csinál, hogy egy fileba fűzi a megadott file(ok)at/könyvtára(ka)t. Olyan mint a tömörítő, de NEM tömörít, csak egybe fűzi a sok file-t, a mérete ugyanakkora lesz, mint kibontva. Viszont kitömöríteni tud, megfelelően paraméterezve.

használata:

kicsomagolás:

tar xfvz filenév.tar.gz
tar xfvz filenév.tgz

Annyit említenék meg, ami fontos lesz a BEtömörítésnél, hogy nem véletlenül jelezzük úgy a file típusát, hogy tar.gz, vagy tgz. Mert először tarral össze van fűzve, majd gzippel be van tömörítve. Innen tudjuk, hogy mi a file, mit kell vele csinálnunk.

tar xfvj filenév.tar.bz2

ua, csak a Bzip2 tömörítő programmal tömörítettet csomagolja ki.
Itt jegyzendő meg, hogy több paraméter megadását lehet egybe is írni egy kötőjel után.
Mint jól látható ebben az esetben is.
Viszont mint jól látható, a tar kivétel is, kivételesen NINCS kötőjel a paraméter(ek) előtt!

Az előbbi paraméterek értelmezése:

x = e(X)tract, kicsomagolás
f = nem jut eszembe :-)
v = (V)erbose, bőbeszédű
z = un(Z)ip-pelje is
j = bzip2-vel tömörítettet csomagolja is ki

Keríts egy akármilyen tar.gz-t. Pl. Linuxod CD-jéről. Indítsd el a Midnight Commandert (mc), gondolom használtál már filekezelőt (Norton, Volkov, Windows Commanderek). Csinálj a home könyvtáradba egy gyakorlas nevű könyvtárat, és másolj bele egy akármilyen tar.gz-t. Lődd le az mc-t.
Menj bele (ha nem ott vagy) a gyakorlas nevű könyvtárba (cd gyakorlas). Listázd ki, hogy lásd, ott a tar.gz (ls). És tömörítsd ki, ahogy fent láttad. Utána ha kész, megint listázd ki. Hagyd így, majd legvégül nyomj egy sima cd parancsot (home könyvtáradba visz, ugye).

su - (s)et (u)ser - felhasználó változtatása

használata: su anonymous

(majd kéri anonymous nevű felhasználó jelszavát.)

visszalépés: exit vagy logout, ekkor ismét saját magunk leszünk.

Paraméter nélküli kiadása automatikusan root lesz, tehát mintha su root-ot írtunk volna. Próbáljuk ki. Utána logout, vagy exit. Ne maradjunk root!

pwd - (p)rint (w)orking (d)irectory, azaz az aktuális könyvtár kiírása, ahol épp állunk.

cp - (c)o(p)y - másolás

használata: cp mit hova

pl.: cp doc/telefonkonyv doc/katalogusok

Ekkor a doc könyvtárból átmásolja a telefonkonyv file-t a doc/katalogusok könyvtárba. Tehát ugyanoda, csak más néven.

mkdir - (m)a(k)e (dir)ectory - könyvtár létrehozása

használata: mkdir konyvtarnev

pl.: mkdir konyvtar

Mondtam már, hogy írjad! Tudom, hogy már tanultuk, de írjad. Utána egy ls parancsot is nyomjál.

rmdir - (r)e(m)ove (dir)ectory - könyvtár letörlése

pl.: rmdir konyvtar

Megint nyomj egy ls-t.

Ekkor letörli az aktuális könyvtárban lévő konyvtar nevű könyvtárt HA üres. Mivel az, letörölte.

Pl2.: rmdir gyakorlas

Mivel nem üres, itt játszottunk kicsomagolást, és van benne cucc, ezért szól, és nem engedi törölni, csak a megfelelő paraméter megadása után.

De mivel Linuxban egy feladatot többféleképpen, több paranccsal is meg lehet oldani, én ezt "rm -rf gyakorlas" parancs kiadásával csinálom. Persze ezzel vigyázni kell, mert ha volt a gyakorlas könyvtárban valami, ha nem, az egészet szó nélkül eltünteti. És Linuxban NEM LEHET VISSZAHOZNI TÖRÖLT FILE-t! Biztonsági okok, ugye... Úgyhogy óvatosan az rm paranccsal. Nem kell nagyon megijedni, csak ÉSSZEL.

rm (r)e(m)ove - file törlés
-r (r)ecursive, alkönyvtárakkal együtt
-f (f)orce, azaz erő, mindenképp töröl mindent, könyvtárat is!

használata: rm mit

pl.:

rm gyakorlas/az_a_tar.gz_amit_oda_raktal

rm * mindent töröl az éppen aktuális könyvtárban, - ugye a csillag egy joker karakter, MINDEN-t helyettesít. Tehát Mindent törölni fog, kivéve az alkönyvtárakat és a benne lévő fileokat. Ezt most ne próbáld ki.

rm -rf gyakorlas/* letörli a gyakorlas könyvtáron belül lévő alkönyvtárakat fileokat, az égvilágon mindent, kérdés nélkül. Írjad. Aztán írjad: ls gyakorlas

rm -rf / na ez az a parancs, amit soha nem szabad kiadni, pláne nem rootként... (az egész gyökérkönyvtárat (vinyódat) letörli kérdezés nélkül), tán ne próbáld ki.

cd -(c)hange (d)irectory - könyvtárváltás, akár DOS-ban, teljesen ugyanaz.

használata: cd hova

pl.:
cd /usr/bin belép a /usr/bin könyvtárba (nyomjál ls-t is, hogy láss.)
cd / a gyökérkönyvtárba lép (ls, megint, ismerkedjünk a rendszerrel)
cd így, paraméterek nélkül kiadva hazavisz a saját home könyvtárunkba, azaz mintha azt írtuk volna be, hogy cd /home/pistike

Írjad ezt is: cd /root

Permission denied - hozzáférés megtagadva (ez nem parancs, hanem hibaüzenet) Csak mert most eszembe jutott, és kezdők nagyon sokszor találkoznak vele. Mint ahogy most te is.

man - (man)ual, azaz kézikönyv

használata: man programnév

pl.: man ls ad pár oldal dokumentációt az ls programról

ps - (p)roces(s) futó programok (processek) listája

használata: ps

pl.: ps a (a)ll, azaz mind, az összes futó program listája

top – rendszertesztelő/figyelő program, érdemes nézegetni. Olvasd a man oldalát (man top) infóért, hogy mi micsoda)

df – (d)isk (f)ree , azaz szabad lemezterületet mondja meg a meghajtókon.

df -h (h)uman, megabyteban, gigabyteban mondja, próbáld ki.

cat – con(cat)enate fájlokat fűz össze, és kiírja a standard kimenetre (képernyő)

haszálata pl.: cat doc/telefonkonyv ekkor kiömleszti a képernyőre a telefonkonyv file tartalmát.

more – oldalankéti megjelenítés

less – ua. mint a more, csak jobb, kényelmesebb.

Pl. less doc telefonkonyv és már olvasgathatod is a telefonkonyv dokumentumodat

head – feje, azaz fájl „fejének”, első sorainak megjelenítése
tail – farka, azaz fájl „farkának”, utolsó sorainak megjelenítése

használatuk, példa:

head doc/telefonkonyv
tail doc/telefonkonyv

Kiírja az első ill uccsó néhány sorát a telefonkonyv doksidnak.

head -2 doc/telefonkonyv
tail -2 doc/telefonkonyv – az első ill. uccsó 2 sorát írja ki

wc – Neeeeeem, nem a budi... Hanem (w)ord (c)ount, azaz szó számoló, megszámolja hogy a paraméternek megadott fileban hány szó van.

Pl.: wc doc/telefonkonyv

mount – fájlrendszer csatlakoztatás /mountolás/ a rendszerünkhöz.

használata rootként: mount mit hova

A floppy az fd0 a Linuxban. A cdrom többnyire cdrom. Ezek a /dev könyvtárban vannak, tehát úgy kell rájuk hivatkozni, hogy /dev/fd0 ill. /dev/cdrom. Tehát:

mount /dev/fd0 /root/teszt

Ezt most ne próbáld. Mint látható a floppyt most a root a saját homekönyvtárának teszt nevű könyvtárába csatlakoztatta, nem pedig a megszokott /mnt/floppy-ba. A root ugye MINDENT csinálhat...

Felhasználóknak csak arra van joguk, hogy a szabványos, előre kijelölt helyre mountoljanak, ők így használják:

mount hova (a hova egyúttal egyenlő az eszközzel is)
pl: mount /mnt/floppy
mount /mnt/cdrom

Ezt se próbáld. Ugyanis a legtöbb disztribben már automount, vagy supermount van, ami egy egetverő baromság és lehetséges biztonsági rés is... Ezerszer el fogom mondani, hogy a Linux biztonságos rendszer, persze ha lamer módon lyukakat hagyunk, akkor nem biztos, hogy a Linux a szar... A legtöbb disztribben (sajna) supermount van, úgyhogy ezt a parancsot nem nagyon fogod használni.

touch – megérint, file-t lehet vele”megérinteni”, létrehozni. Teljesen

mezei (szöveg)file-t hoz létre, ami teljesen üres.

pl.: touch doc/telefonkony

Írjad, PONT így. Nem írtam el. Aztán ls doc, hogy lássad.

setcd – cd sebességének beállítása (ha feltelepítetted, alapból nem biztos, hogy fent van, csak hasznos cucc, azért jutott eszembe)

használata:

setcd -x n /dev/cdrom n=természetes szám, a kívánt sebesség

pl.: setcd -x 8 /dev/cdrom – nyolcszorosra leveszi a cd sebességét (nem zúg filmnézés vagy zenehallgatás közben)

Ha nincs ilyen progid, akkor szerezd be, hasznos.

Na, most hogy ilyen sokmindent kipróbáltunk, gyakoroltunk és megtanultunk, máris vérbeli Linux buherátornak érezhetjük magunkat, de sajnos ez még nem így van. Most vagyunk kb. azon a szinten, mint amikor windowsban sikerült elsajátítanunk a kettős kattintás végrehajtását! ;-)
Na, azért nem kell annyira lelombozódni, hanem tovább olvasni ezt a dokumentumot, és tanulni. Fog ez menni, az egyik legnehezebb részén túlestünk.

Végülis valljuk be, leginkább nem nehéz volt, hanem csupán ismeretlen, teljesen új dolgok.

Gyakorolgassuk párszor a parancsokat. Hozzunk létre könyvtárakat, törölgessük le, nézzük meg mennyi helyet foglalunk, nézzük meg milyen filejaink és könyvtáraink vannak, hozzunk létre gyakorlásként szövegfileokat, keresgéljünk bennük, keressünk rájuk, stb. Majd mehetünk is tovább.

WILDCARD, AZAZ A JOKER KARAKTEREK

A rendszer által használt joker karakterek.

Van ugye a „*”, ami az abszolút joker karakter, mindent helyettesít.

Tehát keresünk:

find –name telefo*

akkor megtalálja a "telefo" kezdetû összes filet, a „telefo” után bármi állhat, akárhány betû, szám, bármi. Vagy akár semmi. Mint látod, megtalálta a telefonkony és a telefonkonyv nevű fileodat is.
Írjad: find –name telefonkony*

Mint látod, semmi különbség, tehát így is megtalálta mindkettőt.

Akkor van a "?".

Ez egyetlen, bármilyen karaktert helyettesít. Előző példával:

find –name telefonkony?

Érdekes, nem? Csak a telefonkonyv nevű file-t találta meg.

Tehát megtalálja a "telefonkony" kezdetû olyan file-t, ami után még EGY bármilyen karakter VAN. (mert egy kérdőjelet raktunk).

Mint látod, most MUSZÁJ, hogy álljon utána egy karakter, ez a "telefonkony" nevû file-t NEM találta meg. Értelmesebb példa, ha mondjuk nem tudjuk, hogy a file kis- vagy nagy betûvel kezdődik-e, akkor:

find –name ?elefonkonyv

Na, ez utóbbit nagyon nagyon ritkán fogod használni, a *-ot annál inkább.

A TABULATOR GOMB HASZNÁLATA

Nem kell megijedni a hosszú parancsok begépelésétől. Csak használni a mágikus TABULATOR billentyűt. Elkezded írni a parancsot/elérési utat, és csak csapkodni kell a TAB-ot, kiegészíti, amíg tudja. Szövegkiegészítő funkciója van a TABULATOR gombnak.

Például bele akarunk menni a cd-n lévő egyeb, azon belül a képek, azon belül a sajat, hazibuli, szex könyvtárba. Ezt ugye a

cd /mnt/cdrom/egyeb/kepek/sajat/hazibuli/szex

paranccsal tehetjük meg. Persze a tabulátor gomb használatával alig kell gépelnünk. Valahogy így:

[TAB] = tabulator gomb leütése

cd /mn[TAB]c[TAB]eg[TAB]ke[TAB]sa[TAB]hazi[TAB]s[TAB]

a fenti majd' egy soros parancsot így pillanatok alatt be lehet gépelni, alig kellett írni valamit. Elkezded írni a dolgokat, annyit kell csak begépelni, hogy egyértelmű legyen a gépnek.

Írjad:

touch doc/hosszufilenev
touch doc/hosszuezis

Ki ha akarjuk listázni a doc könyvtárban lévő hosszufilenev file-t, és csak annyit írunk be, hogy:

ls doc/h

majd nyomunk egy TAB-ot, akkor kiegészíti az "osszu”-val, és megmutatja, hogy két file-t is talált, ami ugyanígy kezdődik, ezért kell neki még karakter, mert tovább nem tudja kiegészíteni, mert a hosszufilenev és a hosszuezis is „hosszu”-val kezdődik, ezért nem egyértelmű még.
Ekkor még leütsz egy f betűt, majd TAB, és már folytatja is, hozzárakja, hogy "ilenev ", (szokozt is!!!) hiszen már egyértelmű, hogy mit akarunk.

Nehéz magyarázni, ezért is jó, ha csinálod, látod. Egyszerű, és baromira kényelmes, próbáld ki.

A RENDSZER BIZTONSÁGA, JOGOSULTSÁGOK

A rendszer biztonsága egyrészt magára a felépítésére, másrészt arra az alaposan kidolgozott rendszerre épül, ami a hozzáférési jogokat szabályozza.
A rendszer a magra, a kernelre épül, ez a lelke az egésznek. Pici, nincs 1 MB, ami bootkor a memóriába töltődik. Mindent a kernel intéz a rendszerben, semmilyen program nem képes semmire direkt, a kerneltől semmi nem veheti el a vezérlést, a hardverhez is kizárólag a kernel fér, semmi más. Minden kernelhívásokkal történik. Ha pl. egy mentésnél a program írni akar a merevlemezre, akkor meghívja a kernelt, közli vele, hogy kéne a vinyó, mert ki akar írni vmit. A kernel eldönti, hogy ki a soros, és ha a progi sorbakerül, kiírja amit kér. Tehát ha véletlenül lefagy egy program, az nem tudja magával rántani az egész rendszert, mert nem a kernel fagy ki. Persze ha egy progi nagyon szarul van megírva, akkor megtörténhet, semmi sem kizárt, de ennek esélye gyakorlatilag nulla. A rendszer pörög tovább, és szépen ki lehet lőni a beragadt progit.

A rendszer nem valós idejű multitask (többfeladatos) rendszer, ami azt jelenti, hogy VALÓJÁBAN nem EGYSZERRE fut több program, csak úgy látszik. Fizikailag úgy működik, hogy a kernel bizonyos időintervallumban futtatja a programokat, időosztásos alapon. Nem lehet pontosan megmondani, de kb. 100 (???) egységre osztja a másodpercet. Az óra ketyeg, az egyik progi megkapja a rendszert egy egységnyi ideig. Amint ez letelt, a progi „leáll”, és a másik kerül sorra. És így tovább. Ha végig ért, kezdődik elölről, ismét megkapja az első, és így tovább. Ez sokkal jobb megoldás, mintha valós időben egyszerre futna minden, ez is növeli a stabilitást.

A biztonság másik részéért az alaposan kidolgozott jogosultságrendszer felel.

Már tudjuk, hogy ls -l paranccsal tudunk hosszú listát kérni, valahogy így néz ki:

```
drwxr-xr-x      pistike    user       2002-11-13 10:30  doc
-rw-r--r--      pistike    user       2003-02-02 11:10  akarmi.mp3
-rwxr-xr-x      pistike    user       2001-12-31 12:12 *script
```

Kérj egyet, hogy lásd élőben. Nézzük először a példa második és harmadik oszlopot. A második oszlop a file tulajdonosát jelzi, aki éppen pistike. A harmadik a csoportot jelzi, itt user. Pistike a user csoportba tartozik.

Namost. Az első oszlop a fileok jogosultságai. 3 különböző file-t látsz itt a példában.
Az első, a doc, ez egy könyvtár. Ez az első oszlop legelső jelöléséből látszik, ott egy "d" (mint directory) betű. A többinél nincs semmi, ki van húzva, tehát „sima” fileok.

Ezt követően jön a 9 darab jogosultság jelölés, amit hármasával kell nézni. Az első három a tulajdonosra vonatkozó jogosultságok, a második három a csoportra vonatkozó, a harmadik három pedig a bárki másra vonatkozó jogosultságok.

Nézzük az akarmi.mp3 file-t. -rw-r--r-- a jogosultsága, ez leosztva:

Filetípus	tulaj	csoport	más
-	rw-	r--	r--

File típusa NEM különleges, azaz sima, mezei file. A csoportokon belül ugye 3 jogosultság van vagy nincs. "r", mint (r)ead, azaz olvasási az első helyen. Jelen esetben mindenki tudja olvasni, mert mindhárom helyen van „r” jog. A második helyen van a "w" mint (w)rite, azaz írási jogosultság. Mint látjuk csak a tulaj tudja írni, az ugyanabba a csoportba tartozók sem, és bárki más sem, ki van húzva. A harmadik helyen van vagy nincs az "x", mint e(x)ecutable, azaz futtatási jogosultság. Mint látható, senki nem tudja

futtatni. Az „x” helyén szintén kötőjel van, ki van húzva. Mivel ez egy zene, így nincs is szükség rá, hogy futtassuk. Felmerülhet a kérdés, hogy mi van, ha rakok rá mégis futtatási jogot?
Semmi. Szuverén jogod. Persze csak akkor, ha a te tulajdonod a file. Csak nincs értelme, mert utána ha lefuttatod, a Linux le is futtatja, aminek egy hibaüzenet az eredménye, ki fogja írni, hogy ez nem futtatható file, azaz nem ELF bináris. Mielőtt kérdés lenne (E)xecutable (L)inux (F)ile, ha jól tévedek.

Van a példában egy harmadik file is, mely előtt egy csillag is látszik. Ez plusz infó, a futtatható fileok előtt szerepel, hogy könnyebben lássuk. A jogosultságokból látszik, hogy ez futtatható állomány.

Hogy bonyolítsuk egy kicsit a dolgot, már úgyis felmerült legalább két kérdés. Miért van a könyvtáron futtatási jog? A könyvtár nem bináris, nem is script. Azért van rajta, mert amikor entert bökünk rá az mc-ben vagy belelépünk, akkor futtatjuk. Az mc-ben sokkal érthetőbb, ugye entert akkor nyomunk vmire, ha futtatni akarjuk. És a könyvtárra entert kell nyomni.
Ha nincs rajta futtatási jog, akkor nem tudunk belelépni. Majd kipróbáljuk.

Jogok osztogatása és fosztogatása közben MINDHÁROM csoportra oda kell figyelni. Hiszen, ha mondjuk a tulajdonostól, pistikétől elvesszük a könyvtárról a futtatási jogot, attól még pistike mindig bele tud lépni. Miért? Mert pistike a user csoportba tartozik, a filet pedig a user csoport tudja futtatni. Tehát van pistikének futtatási joga. Ha megvonjuk a csoport futtatási jogát is, még mindig bele tud menni pistike, hiszen az other, azaz bárki másra, bárkire, akárkire vonatkozó jogosultságokban még mindig ott van a futtatási jog, és akárhogy is nézzük, pistike ugye akárki.

Na akkor a futtatási jogot tisztáztuk. Olvasási jog. Annyi az érdekessége, hogy az olvasás nem szó szerint értendő, mert ugye ki akarna olvasni mondjuk mp3 file-t? Olvasni általában szövegfilet szoktunk. Az olvasás inkább azt jelenti, hogy meg tudjuk-e hallgatni azt az mp3-at, meg tudjuk-e nézni azt a filmet, magyarul látjuk-e a file-t, használhatjuk-e?

Írási jog. Szintén nem szó szerint értendő. Nyilván mp3-at nem nagyon szoktuk szó szerint írni, mondjuk szövegszerkesztővel, de pl. zeneszerkesztővel átbuherálhatjuk. Írás helyett találóbb kifejezés a módosítás.
Ha törlünk valamit, akkor módosítjuk a könyvtár tartalmát amiben az található... Nem a file-ét, amit töröltünk, hanem a könyvtárét, amiből törlünk. Ha nincs írási jogunk egy könyvtárra, akkor annak tartalmát nem tudjuk módosítani, ergo nem tudunk benne filet létrehozni, vagy abba valamit bemásolni, abból törölni.

Itt még valamire oda kell figyelni. Hogy ne legyen olyan egyszerû a dolog, van mondjuk a saját home könyvtáradban egy file, ami a root tulajdona, legyen történetesen egy szövegfile. Mondom legyen. Csináld.

```
su
(root jelszó)
touch valami
exit
```

Kérd le a hosszú listáját!

```
ls -l valami
```

Ilyesmi lesz:

```
-rw-r-r--   root  root  03-01-01  12:21 valami
```

Ezt a file-t, mivel rád a 3., azaz a bárki másra vonatkozó jogosultságok vonatkoznak, így csak olvashatod. Akárhogy is erőlködsz, nem tudom módosítani (írni) a file tartalmát, vagy annak jogosultságait.

Próbáld:

```
joe valami
```

írkálj, majd mentsed (CONTROL+K, majd X, ugye).

Nem ment. Mert nem atiéd, és nincs rá írási (módosítási) jogod. DE! Töröld le!

rm -f valami

Listázd ki (ls -l valami), tényleg eltűnt.

Minden zokszó nélkül le tudod törölni, max. szól a rendszer, hogy a file írásvédett.
Na akkor hogy is van ez? Nincs rá írási jogod, de mégis tudod törölni? Igen. Teljesen logikus. :-)
Olyan könyvtárban van, ami az TE tulajdonod, és írási jogod is van rá. A home könyvtárad és a benne lévő könyvtárak ugye mind ilyenek, mind a TE tulajdonod. És van jogod őket írni, módosítani. Ellentétben a valami nevû file-lal.
Dehát nem az enyém a file!!! - sipákolsz. Nem számít. :-) A unix rendszerek alapfilozófiája, hogy MINDEN file. A könyvtár is. Ebből következően a home könyvtárad is. És mikor a home könyvtáradban tevékenykedsz, (ami egy file) módosítod ANNAK tartalmát. Ha letörlöd belőle a valami nevû dokumentumot, NEM AZT módosítod, nem írsz bele, teljesen változatlanul hagyod, módosítatlanul, hisz nincs jogod változtatni, módosítani (azaz írni)... Hanem a /home/pistike nevû könyvtár tartalmát módosítod a törléssel...

Ezekután teljesen érthető, hogy miért is nem tudjuk userként szétbarmolni a rendszert.

Ha nyomnál egy "rm -rf /"-t a gyökérkönyvtárra userként, akkor azzal csupán annyit érnél el, hogy a home könyvtáradat legyalulod, minden más megmarad és egy csomó Permission Denied hibaüzenetet kapsz még ajándékba. A rendszer vígan megy tovább, ha az asszony, a gyerek a gép elé ül, bejelentkezik, ugyanúgy használja a gépet. Csak magaddal cseszel ki... Persze ha rootként adod ki ezt a parancsot, akkor ugye picit más a helyzet...

Root az egyetlen (optimális esetben) isteni hatalommal rendelkező felhasználó. Szó szerint bármit, mindent megtehet, amit csak akar. A root bármilyen file-lal vagy könyvtárral is azt tesz amit akar.

Akkor állítsunk file jogosultságot.

Először kezdjük a tulajdonossal és a csoporttal.

File tulajdonosát és csoportját unix rendszerben KIZÁRÓLAG a root tudja megváltoztatni. User csak jogosultságot változtathat. Aki létrehozott egy file-t, az mindörökre annak a tulajdonát képezi, - természetesen mint mindig - kivéve, ha isten (root) máshogy kívánja.
Tehát bármennyire is akarod, nem adhatod át a file-t például a barátnődnek. Odaadhatod neki, de a tulajdonosa akkor is te maradsz. Erre csak a root képes.

Tulajdonost a chown (change owner), azaz tulajdonos változtatás paranccsal teheted meg, rootként.

Szintaxisa:

chown kinek_adok mit_adok

pl.

chown cukorfalat akarmi.mp3

Természetesen ha nincs az asszony cukorfalat felhasználói néven a rendszeren, akkor nem fog menni, nem létező usernek nem lehet...

Fontos opciója a "-R", ha rekurzívan akarunk átadni több könyvtárat valakinek.

pl.: chown -R cukorfalat doc

Így a doc könyvtárat a benne található fileokkal együtt adjuk át.

A csoport megváltoztatására a chgrp (change group) parancs szolgál. Ugyanúgy műxik, mint a chown. Ezt kb. soha nem fogod használni.

A jogosultságok megváltoztatására a chmod parancs szolgál. "+"-szal adunk, "-"-szal elveszünk. pl. chmod +r akarmi.mp3, akkor olvasási jogot adunk a file-ra a tulajdonosnak.
Ha csoportot és mást akarunk, akkor meg kell mondani még, hogy mire is akarjuk, de ezt a módszert én nem szeressem, úgyhogy hagyjuk. Egy igazi buherátor az alábbi módon változtat jogosultságokat:

chmod 644 akarmi.mp3

Ez "-rw-r--r--" jogosultságot eredményez. Tetszik?

Na, az a lényeg, hogy - mint látod - három szám kell, az első szám állítja a tulajdonosra vonatkozó, a második a csoportra, a harmadik pedig a bárki másra vonatkozó jogosultságot. Ez az alábbi - általános iskola 1. osztály - összeadási művelettel számíthatod ki:

r=4
w=2
x=1

Ha nincs jog, akkor az r, vagy a w vagy az x értéke 0.

Kis fejszámolás:

rw-r--r-- jogosultság belövése, miért, mitől 644.
Szétbontom, hogy jobban lehessen látni:

r w -	r - -	r - -
4+2+0	4+0+0	4+0+0
6	4	4

Tulaj olvashatja? Igen, az ugye 4. Írhatja? Igen, az 2. Futtathatja? Nem, akkor az 0. 4+2+0=6. Nehéz szülés volt.

Csoport olvashatja? Igen, akkor az 4. Írhatja? Nem, akkor az 0. Futtathatja? Nem, akkor az is 0. 4+0+0=4. Bárki más ua., 4+0+0=4.

Tehát 644. Ugye milyen pofonegyszerű?

Scriptre egyszerűen "chmod 700 script_neve"-t szoktam nyomni, nem nehéz ezekután kitalálni: rwx------, én mindent csinálhatok vele, más semmit. Szövegfilera a 600 érdemes (-rw------- írhatom, olvashatom, másnak meg bakfitty.)

Érdemes pár szót áldozni a sticky bit-re. Bár szinte soha nem használjuk. Ez a pár év alatt nekem még nem volt rá szükségem.
A sticky bitet ha rárakod egy futtatható filera, akkor futtatáskor ugye betöltődik a memóriába. Viszont ha lelövöd, akkor azt követően is a memóriában tartja, így következő induláskor úgy indul, mintha már futna, nem kell rá várni. Tehát arra jó, hogy egy progi minél gyorsabban induljon el. Ha könyvtárra rakod, akkor abba a könyvtárba bárki írhat (a többi jognak is stimmelnie kell), de mindenki csak a saját tulajdonában lévőt törölheti. Tehát az egyik előző példában, ha az a root tulajdonú valami nevû szövegfile mondjuk egy olyan könyvtárban lett volna, amin sticky bit van, akkor viszont nem tudad volna törölni. Ilyen jog van pl. a /tmp könyvtáron. Abban vannak ugye a rendszer ideiglenes file-jai, bejegyzései, a felhasználók által futtatott programok bejegyzései, meg ilyenek. Namost ugye a /tmp-t ezért nem árt, ha tudjuk írni, és abból törölni. De az sem árt, ha egy másik felhasználó nem tudja törölni az általunk, vagy az általunk megnyitott program által létrehozott file-t. (de szép mondat volt, adok is egy piros pontot magamnak).

Ami még fontos, és érdeklődésre tarthat számot az a setuid és a setgid jogosultság beállítása.
Na, ezek azok, amikkel gyönyörû biztonsági lyukakat tudunk gyártani

rendszerünkben.
A setuid az a (Set) (U)ser (ID)entification rövidítése. Lényege, hogy ha ráteszed egy futtatható file-ra, akkor az a TE jogosultságaiddal, és nem az azt futtató felhasználó jogosultságaival fog futni. Namost akkor van gáz, ha mindezt root-ként tesszük. Pl. Az egyik legnagyobb lyuk az mplayerre rátenni a setuid root jogot. Persze még nagyobb elmebetegség gyanánt pl. Az xtermre. Ezt azért mondom, hogy érthetőbb legyen. Tehát ha az xtermre setuid root jogot raknál, és felhasználóként futtatná bárki, akkor root jogokkal tudna mindent csinálni a terminálablakban. Bárki, aki megnyitja az xterm nevű terminálablakot.

Setuid root jognak pl. A passwd parancsnál van értelme. Ugye a jelszavaink a /etc/shadow és /etc/shadow- fileokban vannak tárolva, amire egy usernek nyilvánvalóan nincs írási joga. De nem ártana, hogy a user meg tudná változtatni a saját jelszavát. Mivel írni kell a /etc/shadow file-t, ezért ez nem sikerülne, mert arra csak a root képes, de a setuid root joggal mikor elindítjuk, akkor a program futása alatt root jogokkal fut, mintha azt a root indította volna el, tehát tudja írni a file-t.

Namost disztribekben szándékosan a felhasználóbarátabb rendszer kialakítása miatt vannak hagyva biztonsági lyukak. Pl. Setuid root van a halt parancson. Ismerősebb lehet, ha úgy mondom: poweroff parancs. Ha így sem, akkor mondom, hogy a poweroff parancs beírására a számítógép kikapcsol. (Amikor a számítógép kikapcsolása gombra kattintasz, az erre a parancsra mutat). Ez ugye elég gáz, pl. Lógsz a neten, épp töltesz le, meg CD-t is grabbelsz, CD-t írsz, és a haver neten bejelentkezik, hogy a gépeden lévő home könyvtárából az előre megbeszélt cuccot letöltse, de te mondjuk nem értál rá vagy elfelejtetted bemásolni neki. Erre felmérgeli magát, és kib.szásból beírja, hogy poweroff... A gép meg kikapcsol, letöltés megszakad, grabbelés is, CD-írás, stb, stb... Nesze neked. Tehát a rendszert lelőni nem az összes felhasználónak, hanem csak a rootnak lenne szabad tudni... Ez csak egy példa volt...

Hja, a setgid. Set Group Identification. Ua. Csak a program annak a csoportnak a jogaival fut, amelyik csoportba tartozó rárakja valamelyik filera. Szintén egy kb. soha nem használt dolog...

Gyakoroljunk!

1. A home könyvtárunkat listázzunk ki, hosszú listát kérve (ls -l). Nézzük meg a doc könyvtárunk jogosultságait.
2. Állítsuk át a doc könyvtárunk jogosultságait a chmod paranccsal és a paraméternek a megfelelő számok megadásával úgy, hogy mi tudjuk írni és olvasni, más pedig semmit ne tudjon csinálni vele (-rw-------). Ellenőrizzük le hosszú lista ismételt kérésével.
3. Próbáljunk belemenni a doc könyvtárba (cd doc). Mi történik és miért?
4. Állítsuk át a doc könyvtárunk jogosultságát a chmod parancs és megfelelő számok megadásával úgy, hogy mi mindent tudjunk vele csinálni (írni, olvasni és futtatni) majd ellenőrizzük le. Próbáljunk belemenni. Mi történt, és miért?

ÁTIRÁNYÍTÁSOK

A unix rendszerek filozófiája, hogy minden file. Mint tudod, a file az a cucc, amiket pl. a MC-ben látsz. Vagy az ls parancs hatására, ott vannak a könyvtárakban. Ez eddig tudott dolog. Talán még az is, hogy a könyvtárak is tulajdonképpen fileok, amikben sok kisebb file van. Beszéltünk róla az előbb. De mivel minden file, file a billentyűzet is. File a képernyő is. File a nyomtató is, a vinyó is, a memória is, az égvilágon minden. És ha az ember megszokja, rájön ennek nagyon nagy hasznára.

Ahhoz, hogy belevágjunk, tudnod kell, hogy a programok úgy működnek, hogy egy fileból olvasunk és egy másik fileba írunk.
Minden proginak van egy bemeneti csatornája, és egy kimeneti csatornája. Gondolom ez világos, pl. egy szövegszerkesztő, vagy maga a rendszer standard bemenete nyilván a billentyűzet, és a standard kimenete nyilván a képernyő. Bepötyögünk valamit, és az kiíródik a képernyőre. Gondolom ez teljesen érthető. Viszont, hogy bonyolítsuk egy kicsit, minden proginak (és a rendszernek) valójában 2! standard kimeneti CSATORNÁJA van. A standard kimenet, és a standard hibacsatorna. Ebből nem sokat veszel észre alapesetben, hiszen mindkettő a képernyőre íródik ki.

```
stdin  = standard in    (alap bemenet)         -------> billentyűzet     0
stdout = standard out   (alap kimenet)         -------> képernyő         1
stderr = standard error (alap hibacsatorna)    -------> képernyő         2
```

Az a 0, 1 és 2 a csatornák jelei. Ezt használjuk ezek átirányításához.

Namost az átirányítás is kétféle, a változatosság kedvéért. Átirányíthatsz akármit FILE-ba, ennek jele a ">" és a "<". Illetve a másik lehetőség, hogy egy csövön (úgy hívják, hogy pipe) keresztül, melynek jele a "|", programba irányíthatsz valamit.

Nézzük először a fileba irányítást, gyakorlati, hasznos példán keresztül.

Csinálj a doc könyvtáradba egy proba nevű könyvtárat (mkdir)!
Hozz létre benne egy teszt nevű üres szövegfile-t (touch)!

Tehát a doc könyvtáradban van néhány szövegfile már, meg mostmár egy proba nevű alkönyvtár és abban egy teszt nevű szövegfile.

Keress rá a doc könyvtáradban lévő fileokban a grep paranccsal mondjuk a Tóth szóra. (Vagy amilyen nevű személynek a telefonszámát beírtad, olyanra ami van.

```
grep Tóth doc/*
```

Mint tanultuk, a * az minden-t jelent, tehát a doc könyvtáradban lévő összes fileban fog így keresni. Erre megtalálja azt a file-t, amelyikben szerepel, és kiírja azt a sort.
DE! Hibaüzenetet is kapsz, történetesen, hogy a proba az egy könyvtár.

```
Grep: doc/proba: Is a directory
```

Most próbáld ki, hogy rekurzívan keresel benne. (-r paraméter). Amint látod, most annyi változott, hogy nincs hibaüzenet, hiszen most rekurzívan kerestél, keresett a doc könyvtárad összes (most egyetlen proba) alkönyvtárában lévő fileokban is. De nem akarsz rekurzívan, mert tudod, hogy amit keresel, az a doc könyvtáradban lévő fileok egyikében van, nincs szükséged, hogy alkönyvtáraiban is keressen, hiszen be tudod jobban határolni. Így viszont – sok alkönyvtárnál – ez sok (felesleges és zavaró) hibaüzenetet eredményez. Ezért irányítsuk át a hibacsatornát a képernyő helyett egy hiba nevű fileba. Írjad:

```
grep Tóth doc/* 2>hiba
```

Mint látod, az eredmény sikeres, hiszen hibát nem írt ki.

Ezt követően bármilyen szövegszerkesztővel ill. megjelenítővel (less, more, cat, joe) nézd meg, hogy ha volt valamiféle bánata (volt neki, mint tudjuk), mi is volt az.

Most csináljuk meg ennek ellenkezőjét! Azt akarjuk, hogy az eredmény ne a képernyőre íródjon ki, hanem irányítsuk egy file-ba. Akkor nem a kettes csatornát, hanem az 1-est kell átirányítanunk. Írjad:

```
grep Tóth doc/* 1>grep_kimenet
```

Ekkor a hiba megjelenik a képernyődön, de az eredmény nem. Azt a grep_kimenet nevű fileban találod. Nézd meg. (cat, joe, less, stb.)

Viszont, van alapbeállítás (default), az átirányításoknál ez nem más, mint az 1-es csatorna. Tehát ha nem mondjuk meg, hogy melyiket irányítsa át, akkor alapesetben az egyest, az stdout-ot. Töröld le a grep_kimenet nevű file-t (rm grep_kimenet), majd írjad:

```
grep Tóth doc/* >grep_kimenet
```

Mi történt? Ugyanaz, mint az előbb, a hiba ugyanúgy a képernyőn van, az eredmény viszont a grep_kimenet nevű file-ban. Nézd meg, benne van a Tóth. Hagyd meg, ne töröld le.

Mint windowsban illetve DOS-ban, itt is van egy úgynevezett feketelyuk. Te nagy eséllyel még csak nem is hallottál róla, ha nem használtál DOS-t.
Mint a neve is mutatja, ami belemegy, azt nyomtalanul, maradéktalanul, visszavonhatatlanul elnyeli. Ez DOS-ban a "null" volt. Egyszerűség kedvéért itt is az. Csak persze ez is egy file, mint Linuxban minden, nevezetesen a /dev könyvtárban található, és szintén null a neve. Teljes elérési úttal tudunk rá hivatkozni: "/dev/null".

Ez nagyon hasznos tud lenni. Pl. a fenti példa, annyi különbséggel, hogy nem érdekel bennünket a grep hibaüzenete. Nem vagyunk kíváncsiak arra, hogy sír, mert könyvtárat is talált, hiszen nagyon is tisztában vagyunk vele. És mellesleg éppen zavaró is, úgyhogy egyszerűen átirányítjuk a hibacsatornát a /dev/null-ba.

```
grep Tóth doc/* 2>/dev/null
```

Nem is olyan nehéz, igaz?

Namost, keressünk rá egy - a telefonkonyv fileban szereplő - másik névre, (olyanra, amit írtál bele, pl. Nagy). Írjad:

```
grep Nagy doc/* >grep_kimenet
```

Nézd meg a grep_kimenet tartalmát! Az előbb benne volt a Tóth. Most mi történt? Csak a Nagy van benne. Ez azért van, mert felülírta a file-t. Eeegen, csak így szó nélkül. Úgyhogy figyeljünk oda, hogy mit csinálunk. Ha létező filenevet adunk meg, akkor felül fogja írni, a régi eltűnik, és az újonnan létrehozott file lesz helyette, új tartalommal.

De mi akarjuk, hogy a Tóth Pista telefonszáma is a listában legyen. Ne írja felül, hanem rakja bele azt is. Természetesen van erre is megoldás. Most ugye csak a Nagy van benne. Írjad:

```
grep Tóth doc/* >>grep_kimenet
```

Nézd meg a grep_kimenet tartalmát. Szépen hozzáfűzte, ott van a Nagy alatt a Tóth is. Tehát pontosabban úgy lehet fogalmazni, hogy a > Átirányítja egy fileba, a >> meg BELEirányítja egy létező fileba, hozzáfűzi a végére.

Programok kimenetét más programnak is átadhatjuk, hogy onnan vegye a bemenetet, ne pl. a billentyűzetről. Ezt pedig a cső, vagy pipe

segítségével tehetjük meg, a | jellel.

Tanultuk a cat parancsot.

Mivel Linuxban minden file, így a standard bemenet, azaz a billentyűzet is. Tehát a cat nemcsak pl. egy szövegfile tartalmát tudja kiömleszteni a képernyőre, hanem ez alapján a billentyűzetét is. Írjad:

```
cat
```

Enter után eltűnik a kurzor, nincs prompt. De tudsz írni, hát írjad:

```
Ez egy próba.
```

Majd enter. Amint látod, a bemenetről (ami Linux és unix filozófia szerint szintén egy file) olvasott adatot ugyanúgy kiírta a képernyőre, mint mikor egy szövegfile-lal csináltad az előbb.

Hogy megszakítsd a program futását, nyomj egy CTRL+C-t.

Most csináljunk egy szövegfile-t, de ne szövegszerkesztővel, hanem a cat-tal. Hogyan csinálod? Hát persze… a képernyő helyett egy új szövegfileba kell irányítani a cat kimenetét. Írjad:

```
cat >vasarlas
```

Majd szépen egymás alá:

```
Zöldalma
Borsó
Tejföl
Cukor
Almalé
```

Végén nyomj CTRL+D-t, hogy kultúráltabb legyen, ne megszakítsuk a program futását, hanem szépen leállítsuk.
Nézd meg cat-tal a vasarlas file tartalmát (cat vasarlas)! Kiírja ugyanígy, igaz? Most a cat kimenetét adjuk át a sort nevű programnak, és majd annak kimenetét irányítsuk a vasarlas nevű fileba, mely ennek hatására, mint már tudod felülíródik.

```
cat | sort >vasarlas
```

Írjad megint:

```
Zöldalma
Borsó
Tejföl
Cukor
Almalé
```

Majd zárd be a programot a CTRL+D-vel. Most az történt, hogy a cat nem a képernyőre, de nem is egy fileba írta a kimenetét, hanem átadta a sort nevű programnak. A sort kimenetét pedig átirányítottuk egy vasarlas nevű fileba. Nézzük mi történt, nézd meg a vasarlas file tartalmát!

```
Almalé
Borsó
Cukor
Tejföl
Zöldalma
```

A sort program ugyanis a bemenetére érkező sorokat ABC sorba rendezi, ez a dolga. A bemenete a cat által küldött rendezetlen lista, a kimenete pedig a már ABC sorba rendezett.

LINKEK

Nem, nem a kocsmai cimboráidról lesz szó, hanem egészen másról.

Internetet illetve intranetet használóknak nem lehet idegen a fogalom. Magyarul azt jelenti, hogy kapcsolat, de a hivatkozás jobb kifejezés rá. Amikor a neten fent vagy egy honlapon, akkor látsz egy másik honlapot, egy linket, pl. www.pistikehonlapja.hu. Arra kattintva megjelenik ugye pistike honlapja. Namármost Pistike honlapja valójában, fizikailag nem azon a gépen, szerveren van, hanem csak úgy látszik. Az valójában egy másik helyen, gépen van, csak hivatkozik rá, rámutat, olyan, mintha ott lenne, de valójában nem ott van.

Példa.

Valami úgy jó, ahogy van, nem akarjuk átnevezni, de szükség van arra, hogy más néven vagy helyen lássuk ugyanazt.
Erre a problémára (is) találták ki a linkeket.

Viszont van hard link illetve soft link, hogy bonyolítsuk. Életed során 99,9%-ban softlinkeket fogsz használni.

Értelmes példa mondjuk az xmame nevű emulátor. A netről letöltöttem hozzá mondjuk a Warrior Of Fate meg a Mortal Kombat 2 játéktermi játékokat, meg még egy rakatot. Ezek neve WarriorOfFate.zip, MortalKombat2.zip. Ezek a játékok az emulátorhoz, mellesleg ROM-oknak hívják őket. Igen, be vannak tömörítve, jól látod, de az emulátor így használja.

Ugye tárolnom kell őket valahol, valahova csak rakni kell, ezért behajítottam az emulator/xmame/jatekok könyvtáramba. Itt gyűjtöm az xmame-s játékokat. Igen ám, de az xmame nevű emulátor a .xmame/images könyvtárban keresi a ROM-okat (a fenti játékfilokat). Kérdezhetnéd, hogy akkor miért nem oda rakom? Nos azért, mert - csak hogy ne legyen olyan egyszerű - ráadásul nem ám akármilyen néven keresi őket, hanem pl a Mortal Kombat 2-t mk2r14.zip néven, máshogy nem is találja meg, a Warriors Of Fate-et meg wof.zip néven, és így tovább. (Ezekről, hogy melyiknek mi a neve listát kapunk a programtól, de nem ez a lényeg.).

Ismét mondhatnád: Egyszerű, akkor pakoljam át a ROM-jaimat a .xmame/image könyvtáramba, és nevezzem át olyanra, ahogy az emulátor akarja. Egen...
De nekem az, hogy wof.zip nem sokat mond, a rák tudja mi az. A WarriorsOfFate.zip-ről tudom, hogy az a Warriors Of Fate nevű játék.
Szóval nem ártana, ha ránézésre tudnám, hogy az a játék micsoda. Meg ott maradna a helyén, az emulátorok között, pont ezért van könyvtár emu-s cuccoknak. Szeretem, ha rend van a gépemen. Tehát ez nem jó megoldás. Win alatt nincs is megoldás, max. másolással, és akkor 2x van meg ugyanaz, más néven, rengeteg helyet elfoglalva. Linux alatt ez egy link létrehozásával megoldható, és hely sem fogyott a vinyón (pár byte csupán).

Tehát mindent hagytam az emulator/xmame/jatekok könyvtáramban, az összes játékot, és belinkeltem a .xmame/images könyvtáramba olyan néven, ahogy az emulátornak jó.

Tehát:

```
ln -s ~/emulator/xmame/jatekok/MortalKombat2.zip ~/.xmame/images/mk2r14.zip
```

Ugye a ~ az a /home/zsoltino, mint már tudod. És így tovább. Ezzel olyan, mintha ott is ott lenne a file, ráadásul más a neve is. Viszont a mérete az mindössze néhány byte-ocska. Már csináltál te is egy linket, ott van a doc könyvtáradban. Ha hoszú listát kérsz, jobban látod, még mutatja is, hogy mire is van linkelve, nézd csak meg (ls -l doc)

Gyakoroljunk!

1. A doc/telefonkonyv fileodra csináltál soft linket, telefonkonyvlink néven.
2. Listázd ki hosszú listával (ls -l) a doc könyvtáradat, és nézd meg a telefonkonyvlink file-t.
3. Nyissunk mc-t, és nézzük meg a telefonszamok nevű file-t, mi a különbség a többi filehoz képest, miből is látod, hogy az egy link, Majd töröld le.
4. Lépj a saját home könyvtáradba. Linkeld be ide soft linket készítve a doc könyvtáradban lévő telefonkonyv file-t telszamok néven!
5. Kérj hosszú listát róla. Írjad: cat doc/telefonkonyv, majd cat telszamok. Ugye mindkettő ugyanaz. Szerkeszd az új telszamok nevű fileodat a joe-val, írj bele még egy nevet és telefonszámot, majd mentsd el a CTRL+K, majd X lenyomásával.
6. Nézd meg cat-tal a doc könyvtárad telefonkonyv filejának tartalmát! Azt tapasztalod, hogy te a home könyvtárad telszamok fileját szerkesztetted, ami egy link, tehát valójában ezt a file-t szerkesztetted, itt van benne az új név és szám.

A hard linkről annyit, hogy gyakorlatilag a linkelt file másolata. Tehát a mérete is ugyan akkora, minden. Olyan, mintha simán másolatot készítettünk volna. De mivel ez egy link, a tartalma ugyanaz mint a másiké, akkor is, ha szerkesztettük. Tehát ha az eredeti (vagy akár a link) tartalmát megváltoztattuk, a másiké is ugyan úgy változni fog. Mintha egyszerre szerkesztettük volna mindkettőt. Ugyanaz, mint a soft linknél. Viszont, ha soft link esetében letöröljük az eredeti file-t, akkor a linkünk nem létező filera hivatkozik, hibaüzenet lesz, ha olvasni akarjuk. Ellentétben a hard link-kel. Ha letöröljük az eredetit, akkor a hard link ugyanúgy használható, hiszen az gyakorlatilag egy másolat. Készíts egyet, és nézd meg listával, hosszú listával, mc-ben is. Ellentétben s soft linkkel, itt nem látod, hogy az valójában egy link. Még a file mérete is ugyanakkora, természetesen.

ALIASOK

Lehet, hogy emlékszel a DOS-ban a dir parancsra. Ezzel lehetett könyvtárat listázni, ugyanúgy, mint Linuxban az ls-sel. Írjad:

```
dir
```

Nagy eséllyel működik, ugyanúgy könyvtárlistát kapsz. Hogy ez hogy lehet? Nos, akkor beszéljünk egy kicsit az aliasokról. Nem a sajátjainkról, hanem a rendszer aliasairól. Pötyögjed:

```
alias
```

Majd enter. Erre néhány sorban valami hasonlónak kell lennie:

```
alias cd..='cd ..'
alias dir='ls'
alias ls='ls -F --color=auto'
alias...........stb....
```

Gondolom a példát látva mostmár teljesen érthető. Azért megy a DOS-os dir parancs, mert az az ls aliasa.

Akár a sajátunk, vagy a Mátrixban Mr. Anderson aliasa Neo volt, bár ezt mi inkább nicknek hívjuk. Akármit be lehet állítani így, amit csak akarsz. Pl. ha beírunk ilyen parancsokat, hogy:

```
alias torold='rm'
alias lista='ls'
alias konfiguralj='./configure'
alias forditsd='make'
alias installald='make install'
alias stb....
```

akkor máris ért a gép magyarul. :-) Persze ennek semmi értelme, de jó példa volt. Pl. nálam van ilyen, hogy

```
alias google='links www.google.com'
alias freshmeat='links www.freshmeat.net'
alias mutt='TERM=color_xterm mutt'
```

Tehát értelmes példa aliasokra. A links egy text alapú bőngésző, hogy érthető legyen. A mutt pedig a levelezőprogramom (text alapú, nem grafikus), és ha a TERM környezeti változót előtte color_xterm-re állítom, akkor szines lesz a mutt. Hja, és az alias prioritást élvez. Tehát van ugye Midnight Commanderünk. Az mc paranccsal indul. Írjad:

```
alias mc='ls -l'
```

Índítsd el a Midnight Commandert (mc parancs). Hát nem sikerült. Egy szép hosszú listát kaptál. Hiszen - igaz, hogy van mc parancs és program - de az mc most egy alias, ami prioritást élvez. Aliast „törölni” pofonegyszerű, írjad, mert nem lesz mc-d:

```
unalias mc
```

Tehát unalias és utána a beállított alias.

Az, hogy bepötyögted a parancsokat, azzal létre is hoztad az aliasokat. De csak ideiglenesen. Ha azt akarod, hogy ezek az aliasok megmaradjanak a következő bootolás után is, vagy a másik terminálablakban is működjenek, akkor nincs más dolgod, mint a home könyvtáradban található .bash_profile fileba írod be ezeket a parancsokat. Akkor is, ha nincs ilyen fileod.

Ez a saját profilod, tehát a te egyedi beállításaid. Ez lefut amikor bejelentkezel a rendszerbe.

Ha viszont globális beállításnak akarod beállítani az aliast (magára a rendszerre, az összes felhasználóra vonatkozzon), akkor a /etc/profile vagy akár a /etc/aliases fileba kell beírnod. Mondanom sem kell, hogy ez utóbbi fileokba rootként, hiszen rendszeradminisztrálásról van szó. Ennyit az aliasokról.

VÁLTOZÓK, KÖRNYEZETI VÁLTOZÓK

Beszélgessünk kicsit a változókról, elsősorban a rendszer változóiról, a környezeti változókról. Van ugye a ".", meg a "..". Ezek is tulajdonképpen változók, a "." az aktuális könyvtárat jelenti, a ".." pedig az előző könyvtárat. Ellenőrizd mc-ben, legfelül mindig a ".." van, arra bökve az entert tudsz visszafele menni a könyvtárból.
Folytassuk. A HOME is ismerős számunkra, ez a változó a home könyvtárt tartalmazza, teljes elérési úttal ("/home/pistike"). A "~" környezeti változó is ismerős, az dettó ugyanez, csak rövidebben. A TERM változó a terminálbeállításokhoz kell, ez mondja meg, hogy mikor egy terminált használunk, akkor milyen terminálbeállítások vonatkozzanak rá. Ez általában xterm, esetleg color_xterm, konzol esetén pedig Linux. Van ugye a BASH, ami általában a /bin/sh. A DISPLAY az a képernyő. ez többnyire :0.0, azaz a mi képernyőnk. Ennek a változónak az átállításával lehet pl. megmondani, hogy melyik képernyőn "fusson" egy adott progi. Pl. otthon helyi hálózat, öcsém nézegeti a másik gépet. Én épp vmit csináltam, azt mondja, hogy hallgatni akar zenét a másik gépemen. Mondom azon a gépen nincs zene sem, hangkártya sincs benne. Mondom hagyjál most békén, és megnyitottam a gépemen lévő xmms-t, de előtte a displayt átállítottam, hogy:

```
export DISPLAY=ANNAK_a_gepnek_a_neve:0.0
```

majd utána futtattam az xmms-t (ugyanebből a terminálablakból), és megjelent nála az előtte lévő gépen az xmms kezelőfelülete. Az én gépemen futott, az én hangkártyámmal és hangfalammal, de ő vezérelte amazon a gépen amannak a gépnek az egerével, billentyűzetével... Tetszik? Ugye milyen jó, hogy minden file?

Az EDITOR és a VISUAL változó arra jó, hogy pl. egy progi text szövegszerkesztőt használ, akkor megnézi, hogy ezeknek mi a tartalma, és azt a progit használja editornak ai itt be van állítva. Ilyen pl. a mutt levelezőprogram, ami a kedvencem. Nálam mindkettő tartalma a joe szövegszerkesztő, ami szintén a kedvencem. A LESSCHARSET a less karakterkészletét, a LOGNAME a login nevet, a HOSTNAME a gép nevét tartalmazza. A HISTSIZE a history méretét tartalmazza (hány begépelt parancsot jegyezzen meg), tudod, mikor nyomkodod a felfele nyilat, hogy az előbb, vagy azelőtt, vagy még azelőtt begépelt parancsot ne kelljen újra bepötyögni... Ha még nem tűnt volna fel, ilyet is lehet.
Az LS_OPTIONS is fontos, na azért olyan szép színesek a terminálablakban a file-ok, (zöld a futtatható, kék a könyvtár, stb.) mert itt be van lőve, hogy az legyen. A MAIL tartalmazza a levelesládádat, ebből a változóból olvassa ki a levelezőprogid, hogy ugyan merre van a postaláda... Az OSTYPE tartalmazza, hogy éppen milyen OS fut a gépen. Többnyire linux-gnu. A PATH talán az egyik legfontosabb, és ugye ismerős is. Na, ezért nem kell mindig tudni, hogy melyik parancs, progi hol van. Ezért elég beírni, hogy mc, nem pedig, hogy /usr/local/bin/mc, ezért tudja a rendszer "magától", hogy hol vannak a progik futtatható állományai, mert van ez a változó. Kb. Ilyesmit tartalmaz:

```
/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin:/home/pistike/bin:...
```

Nem folytatom, mert nem fér ki. Most már gondolom érthető az is, hogy miért kell a scriptjeinket a home könyvtárunkban lévő bin könyvtárba hajítani... Mert ha nem ott van, vagy nem a pathban beállított könyvtárban, akkor ott nem nézi a rendszer, következésképpen nem találja és nem fogja tudni futtatni, csak akkor ha megadjuk az elérési utat is. A PWD is fontos, ez az aktuális könyvtárunk, tehát ua. mint a ".". A pwd (print working directory) parancs nem csinál mást, minthogy kiolvassa ennek a változónak a tartalmát. Nem is tudom van-e egyszerûbb és kisebb program a gépen... A SHELL-ben a shell (parancsértelmező) van belőve, (/bin/sh) a TZ a TimeZone, azaz az időzóna, az USER a júzer, jelen esetben te, ugye, az UID a júzerID, azaz az azonosítószámod, a TMP és a TMPDIR pedig a saját temp (ideiglenes) könyvtárad. Ezek a legfontosabbak, persze van több is. Hogy mi van most nálad belőve?

Pötyögjed:

```
set
```

És nem történik semmi. Ja, elfelejtettél entert nyomni. Nyomd. És szépen le is szalad a képernyőről. Ismerkedj vele. Egyes változók tartalmát ugye az

```
echo $változó_neve
```

paranccsal tudjuk megnézni. Beállítani pedig, mint a fenti példában, az export paranccsal.

```
export DISPLAY=stronghold:0.0
```

és bármit megnyitok, a stronghold nevû gépre kötött monitoron jelenik meg. Nálam pl. volt egy script a stronghold nevû szerveremen, ami felcsatlakozott a netre (modem). És közben nyitott egy wm appot, egy kis benchmark „ikont" ami a ppp kapcsolatot, forgalmat monitorozza, de az a moonlight-on, azaz az én monitoromon jelent meg. A szerveremen ugye monitor sincs... Csupán ennyi volt a connect nevû scriptem tartalma:

```
wvdial
export DISPLAY=moonlight:0.0
wmnet -ppp0
```

Felcsatlakozik, megmondom, hogy hol a képernyő, mert ott aztán nincs, majd indítja a kapcsolatfigyuzó progit. Mivel luxuskörülmények közt élünk, és annyi terminálablakot nyitunk, amennyit csak akarunk, ezért fontos megemlíteni, hogy mivel a terminálablak egy új shell-t nyit, új parancsértelmezőt, ezért csak abban az ablakban érvényes a beállítás, ahol beállítottuk... Ha maradandót, véglegeset akarunk, és hogy minden ablakban érvényes legyen a beállítás, akkor a profilunkban kell belőni (ugye .bash_profile). Nálam itt be van lőve pl. az EDITOR, a VISUAL környezeti változók. Ne feledjük, hogy a .bash_profile-ban történt változtatások csak akkor lépnek érvénybe, hogyha újra bejelentkezünk. Ezt az aliasoknál elfelejtettem mondani, bocsesz. Tehát ha csak a .bash_profile-ba írjuk be az aliast, vagy valamit, akkor az csak a következő bejelentkezésnél műxik, ha meg csak az egyik terminálablakban, akkor csak abban az ablakban.
De gyors logout, majd login, és már érvényesek is.

A sima változók abban különböznek a környezeti változóktól, hogy azokat mi használjuk akármire. Változónak értéket így adunk:

```
Változó_neve="változó értéke"
```

Pl írjad.: maidátum="1900. február 35."

Ezután ugyanúgy, mint a környezeti változó értékét, kiírathatjuk a képernyőre, felhasználhatjuk:

```
echo „A mai nap $maidátum"
```

Változót „törölni" pedig az: unset változó_neve

paranccsal tudunk, írjad: unset maidátum.

A LINUX HATÉKONYSÁGA, PÉLDA MUNKÁRA, SCRIPTEK

Egy számítógép a parancssor használatával a leghatékonyabb. A Linuxnak is ott mutatkozik meg igazán az ereje. Nyissunk egy terminálablakot. Feladat:

CD-ről audiót grabbeljünk mp3-ba és konvertáljunk mpeget divx-be. E hosszadalmas művelet közben unalmunkban nézzünk filmet.

Linux alatt ez 2 jól megírt script segítségével pillanatok műve. A script a DOS .bat kiterjesztésű filejainak Linuxos megfelelője. Csúnyán foglamazva. Tehát azért célszerű a használatuk, mert csak egy rövid parancsot kell begépelnünk, ami végrehajtja a bele írt, akár többszáz utasítást.
Jó, jó, de hogy csináljunk ilyet? Hát a következőképpen.

Kedvenc TEXT ALAPÚ, NEM OFFICE szövegszerkesztő elővesz (pl. joe), majd új dokumentumot készítünk. A szöveges dokumentumunkat úgy írjuk, mintha az a parancssor lenne. Tehát parancs beírása, enter. Köv. parancs beírása, enter. És így tovább. Annyi különbséggel, hogy tagolhatjuk is. Naszóval.

Legyen a grabbelő script neve mondjuk grab, stílusosan. Ez a cdparanoia és a lame nevű programok segítségével fogja nekünk grabbelni és mp3-ba kódolni a zenecdnket. Legrabbeli, majd 160 kbit/seccel kódolja mp3ba a fileokat, ill. törli a feleslegessé vált wavokat. Tehát írjad, hogy:

```
joe grab
```

és írhatod is bele a következőket:

```
cd $1
cdparanoia -B &&

for WAVFILE in * ; do
lame -b 160 $WAVFILE $WAVFILE.mp3
done

rm -f *wav
```

Mentjük ugye a CTRL+K, majd az X gomb leütésével.

Mielőtt elszörnyedsz, értelmezzük a scriptet.

Az első sor nem csinál mást, mint a cd paranccsal belép abba a könyvtárba, amit az egész scriptnek paraméterként fogsz megadni annak használatakor. Ez ugye azért kell, mert nem árt, ha oda rakja a majdani mp3-akat, ahova te akarod, amit mondasz neki. A $ azt jelzi a rendszernek, a parancsértelmezőnek (shellnek), hogy változóról van szó, most a változó neve 1, azaz az első paramétert olvassa be.

Azért kell változót használni, mert megadsz egy könyvtárnevet, hogy ide nyomja az mp3akat, legközelebb lehet, hogy másik könyvtárnevet adsz meg a scriptnek, ezért ugye a könyvtárnév változik, az egy változó, mindig más.

A második sor csupán a cdparanoia nevű programot futtatja le, megfelelő paramétereivel. Létrejönnek a megadott könyvtárban a leszedett zenék .wav formátumban. Az && azt jelzi a rendszernek a parancs végén, hogy csak akkor menjen tovább, ha gond nélkül lefutott a parancs.

A következő 3 sort direkt külön vettem, hogy jól látható legyen. Ez egy finom kis for ciklus.
Első sora annyit mond, hogy a WAVFILE változóba olvassa be az összes file-t

(*) egymás után, és amíg értéke VAN, bármi, de van, - jelen esetben pl. track01.wav, track02.wav stb. lesz a grabbelés után - addig csinálja (do) a következő sorban lévő dolgokat.
Ami jelen esetben nem más, mint a lame nevű program, megfelelően paraméterezve. Ez konvertálja át a wavokat mp3-ba. Látható, hogy változók vannak filenevek helyett. Ez megint azért van, mert szeretnénk a könyvtárban lévő összes wavot átkonvertálni, és ugye azok neve mindig más. (track01.wav, track02.wav...stb.) A ciklus azért ciklus, mert addig fut, forog körbe, amíg a WAVFILE változó értéke valami. Amint az utolsót, mondjuk pl. a track22.wav file-t (az utolsót) is átkonvertálta és nincs több, a WAVFILE változóba nem tud mit beolvasni, akkor kilép a ciklusból, azaz végrehajtja a ciklus utolsó, 3. sorát ami a done, azaz készen van. Majd ugye mivel kész, kilép a ciklusból, és végrehajtja az ezután szereplő parancsokat, ha vannak. Ugye most van.
Jön az uccsó sor, ami azt csinálja, hogy az összes wav végű file-t letörli ebben a könyvtárban, tehát kitakarít maga után, és csak a kész mp3-ak maradnak meg.

A második script a konvertáló script, melynek neve legyen mondjuk konvert, stílszerűen. Írjad: joe konvert, és írjad bele a következőket:

```
mencoder $* -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=900 -oac mp3lame
-lameopts vbr=3 -o $*.avi
```

(Ez EGY sor, csak nem fér ki)

Mentjük ismét.

Ez nem csinál mást, mint az Mplayer/MEncoder páros segítségével, tehát az mplayer kódoló részével, a mencoder segítségével átkonvertálja a paraméternek megadott filmfile-t, legyen az mpg, akármi, divx-be. Azért olyan hosszú a sor, mert ugye sok paramétert lehet a mencodernek (és az mplayernek is) megadni, igény szerint. RTFM, és onnan ki lehet nézni, hogy mennyi sokmindent lehet vele csinálni. Szintén változót kell használnunk, mert ha beírnánk, mondjuk a macskafogó.mpg-t a $* változók helyére, akkor a szkript értelemszerűen használhatatlan lenne mondjuk a hullajó.mpg konvertálására.

Na, kész is a két script. De a mentés után ez még valójában nem script, hanem két sima szövegfile. A tanult módon:

```
chmod 700 grab konvert
```

paranccsal adjál rájuk futtatási jogot (mint látható, egyszerre több file jogosultságát is tudjuk módosítani), ekkor már script. De hiába írod be, hogy grab, nem találja. Mert nem olyan könyvtárban van, ami a PATH-ban is benne van, hiszen itt van a home könyvtáradban. Hajítsd őket a saját bin könyvtáradba. Akár a commanderrel is lehet, vagy az mv paranccsal. Ha nincs bin könyvtár a home könyvtáradban, akkor nem kell pánikba esni, hanem csinálni kell egyet. Ha belehajintottad, és még mindig nem működik, pl. No such file or directory üzenettel boldogít, akkor ellenőrizd a PATH környezeti változód tartalmát (echo $PATH), nézd meg, hogy van-e benne /home/pistike/bin, vagy $HOME/bin, vagy ~/bin. Nagy eséllyel nincs. Ja, szépen látszik, hogy : van a könyvtárak között. Így kell egymás után megadni a könyvtárakat. Szóval ebben az esetben kénytelen leszel exportálni hozzá. Ugye export-tal egy változó értékét állítod be. Ez felülírja régi értékét, tehát most, mint látod elég sok minden benne van. Ha csak annyit írsz (most még ne írd):

```
export PATH=/home/pistike/bin:
```

CSAK ez lesz benne, viszont eltűnt az összes többi, ergo szinte semmi parancs nem fog működni. Hogy ez ne történjen meg, hozzá kell írni, pluszban. Logikusan, nem pedig felülírni. De borzasztó sok cucc van, most be kell gépelni mindet??? Dehogy. Hiszen azok most a PATH változóban benne vannak, és hivatkozhatunk is rá ($PATH). Tehát helyesen, és írjad is:

```
export PATH=$PATH:/home/pistike/bin:
```

Tehát mintha beírtuk volna azt a sokmindent, ami eddig már volt. Ellenőrizd is (echo $PATH). Szépen látszik, hogy ott van az eredeti rész, mert a

parancs kiadásakor behelyettesítette, és a végén ott az új is. Gyorsan írjad bele a .bash_profile file-odba is, hogy holnap, mikor bekapcsolod a gépet ne szentségelj...

Tehát kész a két scriptünk. E két szkript segítségével a fenti feladatok (grab mp3-ba, filmkonvertálás és filmnézés a konvert közben) pillanatok alatt elvégezhető a következőképpen:

Csinálsz a zenének egy zenecd könyvtárat a home könyvtáradban (mkdir zenecd), majd a filmek könyvtáradban lévő Valami Amerika c. filmet nézed, miközben grabbelsz és konvertálod a filmek könyvtáradban lévő macskajaj.mpg-t ugyanoda, divxbe:

```
mplayer filmek/V_Amerika.avi & grab zenecd & konvert filmek/macskajaj.mpg
```

Ugye milyen egyszerű és gyors? Kb 5 másodperc bepötyögni, (tabulátor gomb hasznos) és a feladatot végrehajtottuk, csak várni kell, közben meg élvezzük a filmet.

Azért értelmezzük, mert ismétlés a tudás jó édes...

Egy sorban több parancsot is kiadhatunk, ha ; vagy & vagy && jelet teszünk közéjük. Azért kell egy sorba írni, mert egyszerre akarjuk elindítani.

A parancsokat itt "&"-del választjuk el egymástól. Ez annyit jelent, hogy és, tehát ezt és ezt és ezt is futtassa, egyszerre, egy időben. Ha && raknánk, akkor az azt jelentené, hogy a következő programot csak akkor futtassa, ha az előző hiba nélkül lefutott. Ez most nem célszerű, hiszen pont azt akarjuk, hogy egyszerre történjen.

Az első ugye elindítja a filmet, hogy a hosszadalmas grabbelés és konvertálás közben ne unatkozzunk. A második a sorban elindítja a grabbelést a zenecd könyvtárba ömlesztve a cuccot. A harmadik a sorban pedig elindítja a macskajaj.mpg konvertálását divxbe.

Nos, ez ugye Linux alatt, parancssorból 5 mp, és már dolgozik a gép, nézzük is a filmet.

Most gui-val (grafikus programokkal, egerészéssel), windows alatt ez valahogy úgy néz ki, hogy katt sajátgép, jobbklikk, új, mappa, mappanév bepötyög. Easy CDDA Extractor progi megnyit, egerészés, kattogtatás, lassan elindul. Projectdivx megnyit, tallózgatás, egerészés kattogtatás stb, és ha még nem fagyott ki a win akkor megpróbálhatjuk megnyitni a BS Playert, egerészés, kattogtatás, tallózgatás... Ha még mindig nem fagyott ki a win vagy egyáltalán elindul a player, (csodálkoznék) akkor már akár 3-5 perc múlva élvezzük a filmet, ha nem szaggat (megint csodálkoznék.)

Linux alatt ugyanez a játék, ugyanennyi idő, ha grafikus kattogtatással és egerészéssel szórakozunk, annyi különbséggel, hogy nem fagy ki, és a film sem akad.

Tehát mint látjuk, a parancssor nagyon hatékony, villámgyors, és egyszerű. Mármint neked még a script megírása nem annyira, hanem a használata. Hiszen hogy is lenne egyszerűbb grabbelni meg konvertálni, mint:

```
grab melyik_könyvtárba
konvert melyik_filmet
```

KOMOLYABB SCRIPT ÍRÁSA

Sokmindent tanultunk. Már tudunk scriptet is írni, láttunk olyat, hogy változót kért be, ezzel paramétert is adhattunk neki. De a lehetőségek messze továbbmutatnak.

Gyakoroljunk, és tanuljunk valami újat. Van már egy telefonkonyv nevű dokumentumunk, és abba tudunk nevet, telefonszámot rögzíteni szövegszerkesztővel. Ha már sok van benne, és kell valakinek a telefonszáma, nem gond, hiszen a grep parancs segítségével pillanatok alatt tudunk benne keresni...
Sőt, az egyszerűség kedvéért rögzíteni bele új nevet és számot a tanultak alapján az echo parancs és az átirányítás segítségével még a telefonkonyv dokumentum szerkesztése nélkül is tudunk:

```
echo „Gipsz Jakab       0620200200” >>doc/telefonkonyv
```

Mennyivel kényelmesebb lenne, ha egy scriptet írnánk, ami megkérdezi, hogy új nevet és számot akarunk-e hozzáadni, illetve keresni akarjuk valaki számát vagy nevét?
Ha rögzíteni akarunk, akkor bekérné a nevet, számot, ha pedig keresni, akkor csak bekérné a keresendő szövegrészletet. Na, ezt fogod most megcsinálni.

1. újdonság: Van egy olyan parancs, ami törli a képernyőt. Ez a "clear". Tiszta logikus, nem? Próbáld ki!

Hasznos lesz, hogy nézzen is ki valahogy a dolog. Csupán esztétika miatt fogjuk használni.

2. újdonság: Azt már tudod, hogy hogy lehet kérdezni, illetve hogy kérdezzél, ugye csak ki kell íratni valamit (a kérdést) a képernyőre. (echo). Van az echo parancsnak egy "-n" kapcsolója, ami annyit csinál, hogy nem nyom a sor kiírása után entert... A design miatt ugyebár ezt jó tudni. Ez jól fog jönni majd, ha a kérdést teszi fel a script.

3. Tehát kérdezni úgy lehet, hogy az echo parancs segítségével felteszed a kérdést. Változónak értéket úgy tudsz megadni, hogy pl. VALAMI=valami. De hogy lehet billentyűzetről bekérni, nem pedig előre megadni? A "read" paranccsal. Beírod, hogy read, és egy tetszőleges nevű változónevet, amibe a választ tárolja a beolvasás után. pl. a kérdés után (új sorba, természetesen) beírod: read VALASZ, ekkor beolvassa a VALASZ nevű változóba amit bepötyög futáskor a júzer.

4. A shellünk van olyan jó, hogy enged feltételt vizsgálni, lásd a már tanult for ciklust. Most az "if, then, else" (ha, akkor, különben) ciklusról kell tanulnunk.

Szintaxisa (formája):

```
if  a megadott feltétel teljesül
    then
    csináld ezt
```

Ez ugye megnézi, hogy a feltétel igaz-e? Ha igen, akkor megcsinálja, ha nem, akkor tovább megy, anélkül, hogy megcsinálná. Persze, ha a feltételünk teljesül, és azt akarjuk, hogy az utána lévő részét a scriptnek már ne hajtsa végre, akkor ki is léptethetjük, megszakíttathatjuk a script futását, ha alá írjuk még azt, hogy:

```
    exit 1
```

Eztán tartozhat még hozzá egy else rész:

```
            else (azaz: különben)
            if ez meg ez a feltétel teljesül
            then (akkor)
            csináld ezt...
```

És ha akarjuk, ide is jöhet egy kiléptetés (exit 1). Így lehet több

feltételt is megadni.
Legvégül a ciklust le kell zárni. Ez roppant bonyolult: fi, azaz az if szó fordítva.

Na, szépen tagoltam is, célszerű, jobban áttekinthető lesz.

Minden if ciklust le kell zárni, különben végtelen ciklus lesz, és ezt hibaüzenettel díjazza a rendszer. Na, nézzünk egy példát:

Hozz létre egy baromsag nevű dokumentumot a home könyvtáradban! (joe baromsag, zongorázz bele valami blablá-t, majd mentsük. Ls-sel ellenőrizd, lásd, hogy ott van.

Aztán írd meg a pelda nevű scriptet. Írjad bele:

```
clear
echo ””
echo ””
echo ”Példaprogram”
echo ””
echo -n "Töröljem a baromsag nevű file-t? (i/n): "
read VALASZ

if [ x$VALASZ = xi ]
   then
   rm -f /home/pistike/baromsag
   echo "A file törölve."
   exit 1
        else
        if [ x$VALASZ != xi ]
           then
           echo "A file-t nem töröltem."
           exit 1
        fi
fi
```

Készen is van, mentsük. Köszi, anyukám jól van. Natehát...

Először is töröltük a képernyőt ugye a clear paranccsal. Az első két echo parancs megjeleníti az idézőjelek közötti nagy büdös semmit. Azaz szépen kihagytunk két sort, hogy nézzen ki valahogy, majd kiírattuk a képernyőre, hogy Példaprogram, majd újra kihagyatunk egy sort. A design, ugyebár.

Az ezt követő echo parancs segítségével íratjuk ki a képernyőre a kérdést. Az echo "-n" kapcsolójával elérjük, hogy szépen a sor végén villogjon a cursor, ahova a választ írjuk és nem pedig alatta. Így jobban néz ki. Már megint a design.
Ezt követően kérjük be a VALASZ nevű változóba a választ a read segítségével, ami ugye i vagy n lesz. A read parancs hatására a program futása „leáll”, adatot vár a standard bemenetről (billentyűzet), egészen ENTER leütéséig. A beérkezett adatot a megadott - jelen esetben VALASZ - változóba tárolja, és folytatja futását.

Az ezt követő sorban kezdődik a feltételvizsgálat. Ennek formája így néz ki, ahogy látod. Ugye feltételt szabunk, hogy a válasz nevű változó egyenlő az i betűvel.
Ez után jön a then, (hogy akkor, abban az esetben) majd alá, hogy mit is csináljon. Hát törölje a baromság nevű file-t. Teljes elérési úttal van ugye megadva, hiszen nem tudni, hogy éppen melyik könyvtárban is állunk, és szeretnénk, hogy bármelyikben állva működjön.
Miután ez megtörtént, ugye kiíratjuk a képernyőre, hogy a file törölve vagyon. Utána áll egy exit 1, azaz leállítjuk a scriptet, az utána lévők NE fussanak le.

Ha viszont a feltétel nem teljesül, akkor nem hajtódnak végre mindezek, hanem az else utáni részre ugrik. Itt is egy if ciklus van, mely szintén megvizsgálja a VALASZ nevű változónkat úgy, hogy ha NEM egyenlő "i" betűvel, AKKOR hajtja végre a meghatározottakat azaz csak simán kiírja, hogy nem törölt.
Tehát annó matekórán a "nem egyenlő"-t az egyenlőségjel áthúzásával adtuk

meg, itt ugye ez kissé körülményes lenne, talán alkoholos filccel... ezért ennek jele a "!=".

Majd itt is itt van az exit 1, tehát kilép. Ezután, ott ugye a „belső if ciklust lezáró fi. Majd legvégén az EGÉSZ nagy if ciklust lezáró fi. Ugye minden if ciklust fi-vel kell befejeznünk.
Adjál a scriptre futtatási jogot, hajítsd a bin könyvtáradba, és próbáld ki! Először lehetőleg n választ adjál. Ellenőrizd le, hogy tényleg nem törölte (ls). Mégegyszer futtasd, és i választ adj. Ellenőrizd le!

Remélem érthető. Ez egy példa volt. Ha kicsit elgondolkodsz és alaposan megfigyeled, rájöhetsz, hogy az else rész EBBEN a példában teljesen felesleges, nem szükséges. Miért is?
Hiszen ugye ha a feltétel teljesül, azaz a válasz i, akkor törli, és kilép, itt le is lehetne zárni fi-vel. Az azt követő rész ekkor ugye nem fut le. Utána meg elég símán odabiggyeszteni az echo „A file-t nem töröltem.” sort, hiszen az akkor csak abban az esetben fut le, ha a feltétel nem volt igaz, azaz nem i betű a válasz...
Tehát így:

```
        ...
        ...
        echo " A file-t törölve."
        exit 1 (eddig ugye ugyanaz, mint az eleje)
fi

echo " A file-t nem töröltem."
```

Az else rész nem is kell(ett volna), tehát ha a feltétel igaz, akkor hajtsa végre a törlést, írja ki, hogy törölt, majd lépjen ki, ennek végére kerül a fi, azaz a cikluslezáró. Ha a válasz nem "i", hanem pl. "n", akkor ugye a ciklus nem fut le, egyszerűen kimarad ez a rész, és ugrik a fi utáni részre, azaz kiírja, hogy nem törölt.

Na akkor hajrá. Írd meg a telefonos scriptet. A fenti minta, példa után ez a feladat rád vár, oldd meg önállóan! Mivan? Nehogy már meghátrálj. Ezek után SEMMI újdonság nincs, nem kell hozzá. Na jó, egye fene, azért segítek.

Legyen a neve mondjuk „tel”. Minél rövidebb egy parancs, annál jobb.

Hogy kezdünk hozzá? Először is ki kell találni, hogy mi a fenét is akarunk.

Hogy nézzen ki valahogy, szépítjük, képernyőtörlés, sorkihagyás. Majd hogy neve legyen a gyereknek, pl. TELEFONKÖNYV szöveg kiíratás, majd jöhet a kérdés, feltételvizsgálat, végrehajtás.

Kérdezze meg: „Rögzíteni akarsz, vagy keresni (r/k)?”
Ezt ugye pl. a VALASZ nevű változóba bekéreted, mint az előbbi példában (read VALASZ)

Vizsgálja meg hogy a válasz r betű volt-e. Azaz:

Ha a válasz r, akkor

> Írja ki pl, hogy „Kérem a nevet: ”
> Ezt kérje be pl. a NEV változóba (read NEV)
> Írja ki mondjuk, hogy „Kérem a telefonszámot: ”
> Ezt kérje be mondjuk a SZAM változóba.
> Majd a két változó tartalmát szépen rögzítse a telefonkonyv nevű dokumentumba echo „$NEV $SZAM” >>/home/pistike/doc/telefonkonyv
> Ezt már tanultuk, ugye a $ jel miatt tudja a shell, hogy a NEV meg a SZAM az változó, és nem azt írja ki az echo, hogy NEV, meg SZAM, hanem azok értékét helyettesíti be, majd a >> átirányító jel miatt nem felülírja, hanem hozzáfűzi a file-hoz. Semmi új nincs benne, csak alkalmazzuk a tanultakat.
>
> Végül lépjen ki (exit 1), majd a ciklust zárja le.

Ezután írjuk azt, hogy mi van, ha a válasz nem r, hanem k? Ezt két féleképp is lehet. Az előző ciklust nem zárjuk le, hanem else következik oda írjuk,

tehát ugyanúgy, mint ahogy a fenti példában. Vagy úgy, hogy a ciklust lezárjuk, és ismét kezdünk egy if ciklust. Ugye egyszer már megvizsgálta, ha r volt, akkor bekérte, rögzítette, kilépett. Ez volt az első. Viszont, ha nem r, akkor nem hajtotta végre és kilépett (exit 1). Hát nem írunk exit 1-et, nem léptetjük ki, ekkor folytatódik a script futása. És mégegyszer vizsgáltassuk meg vele a VALASZ nevű változó tartalmát, hogy nem-e véletlenül k betű, ha már nem r betű volt? Újabb if ciklus:

Ha a válasz k, akkor

> Írja ki: „Keresendő szöveg: ”
> Kérje be mondjuk a KERES változóba.
> Keressen rá a telefonkonyv adatbázisban, ezt ugye a
> grep -i $KERES /home/pistike/doc/telefonkonyv paranccsal
> tudjuk megcsinálni, most nem a KERES-re, hanem annak
> tartalmára fog keresni, hiszen a $ jelből tudja, hogy
> változóról van szó, és annak értékét helyettesíti be.
> Lépjen ki, zárja le a ciklust.

Meg is vagyunk. Futtatási jogot rá, bin-be behajítani, és lehet is tesztelgetni. Csinosíthatjuk, sorkihagyásokkal, a keresés elé beszúrhatjuk, hogy „A keresés eredménye:”, majd a végére, hogy „A program kilép.”, illetve az első ciklusba, beszúrhatjuk, hogy „A telefonszám rögzítve”. A fantáziádra van bízva, játssz vele! Legyen szép, kultúrált.

A fenti példa és a most leírtak alapján ezt már meg tudod írni egyedül.

Nos... Akárhogy is nézem, csináltál magadnak egy telefonkönyv programot...

Pedig nem is tudsz programozni...

PROGRAMTELEPÍTÉS PARANCSSORBÓL, FORDÍTÁS

Programtelepítést először rpm-ből csináld, az megy mindenkinek. Csak katt, hogy telepítés, és ennyi. Mandrake-nél valami Harddrake nevű program vagy akármi csinálja meg helyetted.

Meg ne kérdezd, hogy hova és hol kell kattogtatni, mert már nem tudom, term inalablakból különben is egyszerűbb és gyorsabb. pl.:

rpm -ivh /mnt/cdrom/Mandrake/RPMS/akarmilyen-program.i386.rpm

a paraméterek jelentése:

i = (i)nstall,
v = (v)erbose, azaz legyen bőbeszédű, látni is akarod, hogy mit csinál
h = (h)irtelen most nem jut eszembe. :-)

Nem kell megijedni, hogy hosszú a parancs, csak használni kell a tabulator-t.

Mellesleg a fordítás sem túl komplikált, csak első ránézésre.

A - már tanult - parancsok a következők:

su (kéri a root jelszót)

Ez azért kell, mert a programokat rootként telepítjük, fordítjuk legtöbbször. Ritka kivételek esetében nem.

mc (Midnight Commander) elindít.

Bemásolod az új progi forrását, ahova akarod, többnyire a /usr/src könyvtárban illetve abban létrehozott alkönyvtárakban illik tárolni, majd F10-zel kilépsz a commanderből úgy, hogy abban a könyvtárban állsz, ahol a forrás van).
Lehet, hogy egyes disztribek commandereiben nincs belőve, hogy ott maradjon abban a könyvtárban, ahol épp kilész a commanderből, hanem visszarak oda, ahonnan futtattad az mc-t. Erre figyelj oda, mielőtt anyázni kezdenél.

Aztán ki kell csomagolni a tar.gz-t vagy tar.bz2-t. Most legyen tar.gz.

tar xfvz ujprogi-1.0.i386.tar.gz
cd ujprogi-1.0 (mert mint látni fogod, ebbe a könyvtárba nyomja majd ki.)
./configure
make
make install
logout (vagy exit, ahogy jobban tetszik)

A logout után ismét „magunk” leszünk.

Szóval fogod a nyers programforrást, amit általában a gzip vagy jobb esetben a bzip2 programmal tömörítettek. Hogy melyikkel, azt onnan tudod, hogy a kiterjesztése tar.gz vagy tgz a gzip esetén, tar.bz2 a bzip2 esetén. A Linux nagyívben lesz*rja a kiterjesztést. Kizárólag azért használunk kiterjesztést, mert mi, felhasználók hülyék vagyunk, és így könnyebben eligazodunk a fileok közt.

Tehát nyitsz egy terminálablakot, beírod, hogy "su". Ez a (S)et(User), azaz szeretnél másik felhasználóvá átváltozni. Úgy is írhattam volna, hogy "su root". De a su parancs paraméterek nélküli kiadása automatikusan a rendszergazdává változtat. Természetesen kéri a rendszergazda jelszavát.

Namármost célszerű a Midnight Commandert megnyitni (mc), ha használtál Nortont, Volkovot, akkor nem ismeretlen a kezelése, és mindjárt

egyszerűbbek a fileműveletek.

Oda rakod az új progi forrását, ahova akarod, ahol a forrásokat tárolod ezután. Szabvány szerinti helye a /usr/src könyvtár. (az src a (S)ou(RC)es szóból jön, azaz a forrás rövidítése). Ide illik rakni az összes programforrást (tar.gz, tar.bz2, stb.) Win alatt is nagy eséllyel valami install vagy hasonló nevű könyvtárban tároltad az install cuccokat. Persze semmi sem kötelező...

Majd ezt követően ki kell tömöríteni. Ezt a tar paranccsal tudjuk, megadva neki 4 paramétert. Kivételesen nem kell kötőjel a paraméterek elé. az "x" paraméter megmondja a tarnak, hogy eXtract, azaz kitömörítés, az "f" paraméter már megint nem jut eszembe, hogy mit mond, a "v" paraméter megmondja neki, hogy Verbose, azaz legyen bőbeszédű, látni akarod, hogy mit csinál, a "z" paraméter pedig, hogy gzippel van betömörítve a progi. Mint már tudjuk.

Ha esetleg bzip2-vel (tar.bz2) lenne tömörítve, akkor a "z" helyett "j"-t kell írni. (pl. tar xfvj progi.tar.bz2)

Miután ez megvan, belépünk a cd paranccsal abba a könyvtárba, ahova kicsomagolta.

Itt találsz egy configure nevű scriptet. Kilistázod az ls paranccsal a könyvtárat, és ha ha van, futtatni kell, de hogy bonyolítsuk, Linuxban az éppen aktuális könyvtár nincs benne a PATH-ban (elérési út), biztonsági okok miatt. Tehát DOS-ban anno belement az emberke a Duke3D könyvtárba, beírta, hogy duke.exe és indult a game. Linuxban ugyanez a "Command not found" nevű hibaüzenetet eredményezné. Meg kell neki mondani, hogy HOL van a futtatható állomány. Ugye jelen esetben az aktuális könyvtárban, melynek jele a ".". Még windowsban is. A könyvtárakat Linuxban a "/"-rel és nem a "\"-rel választjuk el, tehát a parancs ezért lesz "./configure" és nem csak símán "configure".
Tehát mivel a configure script a /usr/src/progi könyvtárban van, teljes elérési úttal így hivatkozhatnánk rá és így is jó:

/usr/src/progi/configure

De a pont a /usr/src/progi-t, azaz azt a könyvtárat jelenti, ahol épp tartózkodunk, akkor helyettesítsük be helyébe a pontot:
./configure

Így kevesebbet kell gépelni. A / ugye kell, mert a pont a fenti példában NEM a /usr/src/progi/-t jelenti, csak a /usr/src/progi-t.

A feltelepített programok parancsai benne vannak a PATH-ban, azaz olyan könyvtárakban vannak, amit a rendszer "lát". Ezért parancsoknál elég csak a parancsot begépelni, nem kell neki megmondani, hogy azok hol vannak, mert azt a rendszer tudja.

Tehát ott tartunk, hogy lefutott a configure scriptünk, ami bekonfigurálta a rendszerünkre, és optimalizálta a cuccost. Ha nincs véletlenül configure script, akkor ez a lépés kimarad. Most következik a fordítás, amit a make parancs kiadásával érünk el.

A make magyar jelentése: csinálni, csináld. Tehát csak mondani kell a Linuxnak, hogy csinálja, és csinálja. Ugye milyen egyszerű?

Miután kész, meg kell mondani még neki, hogy csináljon install-t, (make install) és csinál installt. A programunk a make parancs hatására lefordult, de még itt van a forrás könyvtárban, és szegény pistike így még nem tudja használni, ezért a make install parancs nem csinál mást, mint a megfelelő helyre pakolgatja a program cuccait.

Ezt követően a rendszergazda accountról kijelentkezünk (visszaváltozunk pistikévé) a logout vagy az exit paranccsal, mert RENDSZERGAZDAKÉNT CSAK AKKOR DOLGOZUNK, HA MUSZÁJ! Pl. Program telepítése, rendszerbeállítások.

Gyakorolj sokat, és egyre több mindenre fogsz rájönni, remélhetőleg kezdenek benned kibontakozni a lehetőségek, melyek gyakorlatilag korlátlanok.

Ettől a résztől kezdve néhány rendszeradminisztrációs feladatról beszélgetünk a továbbiakban, ha érdekel és szükséged van rá, olvasd el. Ezzel véget ért az alaptanfolyam.

MODEM

A modemes problémád gondolom ott kezdődik, hogy winmodem, és ha így van, akkor már el is mondtam mindent.
Az a probléma a winmodemekkel, (a nevükön kívül), hogy a winnyóz intézi az egészet, a modem gyakorlatilag csak egy konnektor, amibe bedugod a telefonvezetéket.

Ezért kell hardware-es modemet venni, akkor nincsen gond. Viszont vigyázni kell, mert belső modemekre nagystílűen rá van írva, ill. kereskedők rábeszélnek, hogy hardware-es.
Meg kell nézni, ha nem férnek rajta a kondenzátorok meg egyéb mütyürkék, mert annyira tele a nyák cuccal, akkor nagy eséllyel tényleg hardware-es.
De a legbiztosabb, ha külső modemet veszel, az kénytelen-kelletlen hardware-es, mert máshogy nem lehet legyártani...
Bár igaz, már a legtöbb fajta winmodem támogatva van Linux alatt is, csak kicsit macerásabb és disztrib függő.

Ha mégis hardware-es modemed van (kétlem, mert azt elvileg minden disztrib kivétel nélkül auto konfigurálja, mert a jó cucc mindig támogatva van), akkor nincs modemes gondod.

Ha mégis, és soros külsõ modemed van, akkor az vagy a COM1-en, vagy a COM2-n, vagy a COM3-on, vagy a COM4-en van. Ezek Linuxban a /dev könyvtárban a ttyS0, ttyS1, ttyS2 és ttyS3 eszközök.

Egyszerûen csinálni kell rá egy linket modem néven, és kész. Mármint arra, amelyik a modem. Ha másnem próbálgasd végig. Pl.:

```
ln -s /dev/ttyS0 /dev/modem
```

stb.

MODEMES NET, CSATLAKOZÁS, LEVELEZÉS, MAILSERVER

Telefonos net, levelező szerver működése, kézi beállítása. Csatlakozáshoz ugye wvdial, levélküldéshez-fogadáshoz fetchmail, ill. a mailszervernek én exim programokat használok.

Az eximben a /etc/exim.conf-on csak az alábbi változtatásokat végeztem:

primary_hostname = moonlight.localdomain (értelemszerűen a saját géped neve kell, nem a moonlight, mert az az én gépem becsületes neve).

primary_localdomain = ..blabla.....moonlight.localdomain....blabla...

meg a revrite config részbe, hogy:

```
zsoltino@localhost.localdomain            zsoltino@freemail.hu      sfrE
zsoltino@moonlight.localdomain            zsoltino@freemail.hu      sfrE
```

Ezeket a sorokat így átírod és kész is.

A /root/.fetchmailrc fileomban pedig ennyi van:

```
set syslog
set postmaster "root"
set bouncemail
set properties ""
poll freemail.hu port 110 with proto POP3
   user "zsoltino" there with password "levelezőjelszavad" is zsoltino here
```

Ezután csak futtatni kell a fetchmailt (rootként) a háttérben és már megy is.

Felmegyek a netre a wvdial paranccsal, ami tárcsáz. A wvdial.conf-ba értelemszerűen a telefonszámot, Internet csatlakozáshoz való felhasználónevet és jelszó-t be kell írni a megfelelő sorba, és kész is a beállítás.

Nálam egy send nevű script intézi a leveleket, azt futtatom.

tartalma:

```
killall -HUP exim
```

Lelövi, pontosabban újraindítja az eximet. Ez azért kell, mert az exim ugye az offlineban megírt leveleket is próbálja kiküldeni, de az nyilván nem fog sikerülni. Ezért a leveleket tárolja, DE ha újraindul, a sikertelen küldéseket megismétli. Mivel már a neten vagyok, felcsatlakoztam, ez már sikerülni fog neki.

```
fetchmail &
```

Futtatja a fetchmailt, ugye háttérben (az "&" jel miatt).

Tehát máris kocognak ki az offlineban megírt levelek, és befele a freemail serverén lévő, nekem címzettek.

HÁLÓZAT OTTHON

Kézzel az alábbi módon kell beállítani.

Először is be kell rakni a hálókártyát, majd betölteni hozzá a kernelmodult (a drivert) Ez lehet a ne2k-pci, vagy ne, vagy bármi, attól függ milyen kártyád van. Ezt a modprobe modul_neve paranccsal lehet. Pl.:

modprobe ne2k-pci

Nyilván be kell rakni ezt a parancsot a boot scriptek egyikébe (/etc/rc.d/rcS, vagy hasonló), hogy legközelebbi bootoláskor már automatikusan működjön.

Aztán az egyik gépet kinevezed szervernek, és a helyi IP cimeket is ki kell osztani mindkét gépen. A /etc/hosts fileba be kell lőni pl.:

```
127.0.0.1   localhost.localdomain           localhost
10.0.0.1    szervergepneve.localdomain     szervergepneve
10.0.0.2    masikgepneve.localdomain       masikgepneve
```

Így már név szerint is lehet hivatkozni rá, nemcsak IP címmel.

Aztán fel kell húzni az interface-t is a hálókártyához, (itt adunk IP címet a gépnek). Amennyiben a /etc/hosts így néz ki, mint a példában, akkor

ifconfig eth0 10.0.0.1 broadcast 10.0.0.255 netmask 255.255.40.0

A másik gépen értelemszerűen a másik gép IP címe, tehát 10.0.0.2, amúgy ugyanez.
Ezeket a /etc/rc.d/rc.network vagy hasonló helyen lévő boot scriptbe is be kell írni, hogy következő bootoláskor már automatikus legyen.

Eztán a gateway-t (átjáró) kell belőni, hogy a szerveren keresztül (ez lesz az Internet átjáró) pl. hogy ki lehessen menni a netre a másik gépről is.

route add default gw 10.0.0.1 (ugye a példában a szerver a 10.0.0.1-es)

A másikon szintén belőni a gw-t, teljesen ugyanezt kell, és szintén berakni a boot scriptbe (/etc/rc.d/rc.network) is.

Kész is a hálózatunk. A gépek kommunikálnak egymással. A disztribek általában már betöltik a kernelmodult, felhúzzák az interface-t, az IP címet kiosszák (telepítésnél megkérdi), sőt a gép nevét is, és valószínű a /etc/hosts-ba már be is van lőve a saját gép, de a másik gép nyilván nincs. A GW vagy nincs belőve, vagy nem úgy, ahogy mi akarjuk.
a route paranccsal ellenőrizhetjük, ott kell látnunk a defaultot, és hogy milyen IP című gép is a default.

A default gw, ha rossz, akkor értelemszerűen a route del default gw ip_cim paranccsal tudjuk kiszedni. Helyére berakhatjuk a szerverünkét. (route add default gw ...)

Ugye nálam pl. fut a xinetd (xinetd & paranccsal), az sshd, (ssh kapcsolathoz, mert azon keresztül kommunikálok a szerverrel, azon még monitor sincs, me' minek), ezen kívül meg pl. az exim (levelezőszerver daemon) fut egy xfs egy rpc.nfsd egy smbd, egy nmbd... stb... daemonok, szóval kinek a pap, kinek a Pappné... Nálad ezek többnyire futnak, a disztribek betöltik.

Nyilvánvaló, ha az IP címet meg akarjuk változtatni permanensen, vagy a gateway-t, akkor az boot scriptekben, ahol már valószínű be vannak lőve, csupán át kell írnunk (/etc/rc.d/rc.local, rc.network, rcS, rc.stb, stb... disztribfüggő, keressük meg a grep -ri route /etc/* ill. grep -ri ifconfig /etc/* paranccsal).

Ha internetünk is van, a szervergépnek, (a példában 10.0.0.1 IP című) maszkolnia kell majd az internet kapcsolatot a host gép felé, ha pl. modemes a net, akkor a ppp0-t, ha másik hálózati kártyán keresztül, akkor az eth1-et.

Szerveren csak be kell még lőni:

iptables -t nat -A POSTROUTING -s 10.0.0.2 -o ppp0 MASQUERADE

paranccsal. Vagy ppp0 helyett eth1 pl. LAN, Wireless, internet kapcsolatok esetén. Persze szintén berakni a boot scriptbe. Mostmár mindkét gép tud netezni is.

De még nincs nameszerver. Azaz csak úgy lehet netezni, hogy pl. www.143.345.456.344 ahelyett, hogy www.babapiskota.hu. A /etc/resolv.conf-ba ezért be kell irni a szolgáltatód nameserverének (DNS szerver) IP címét:

nameserver 213.163.59.10

Ezt mindkét gépen. Készen is vagyunk.

Hja, hogy winnyóz a másik gép?! Format C:, Linux install. Wines problémák akkor nincsenek, ha nincs win. Mivel nincs win, így nincs wines probléma, tehát probléma nem is létezik. így nem is foglalkozom olyasmivel, ami nem is létezik. Egyszerű, nem? Oldd meg magad.

Na jó, annyit kell csinálni, hogy a dózeres gépen szintén, csak be kell lőni a gateway-t (pl. a 10.0.0.1), meg a saját IP címet meg a nameszervert. Asszem. De leginkább nem is érdekel.

NVIDIA DRIVER TELEPÍTÉSE X-HEZ

Letöltöd az új drivert. Futtatási jogot adsz rá, valami NVIDIA….blabla.run a neve a legújabb drivernek.

Az XF86config ill. XF86config-4 fileban a Driver sort megkeresed, valahol középtájt lesz. Hja, be kell mászni a /etc/X11 könyvtárba, itt a helye ezeknek a fileoknak. Belenézel pl. joe-val, és a Driver sort megkeresed, valahogy így kell kinéznie:

Driver "nv"

Egyszerű, az nv-t kell „nvidia”-ra átkeresztelni, mentés, kész is.

Le kell lőni az X szervert, és konzol módban kell felrakni (hiszen a grafikus felület driveréről van szó), tehát nem terminálablak, stb, hanem konzol mód.

Lefuttatod a letöltött drivert.

Továbbá, ha még mindig nincs 3D userként, akkor nézd meg rootként. Ha rootként igen, akkor jogosultsági probléma van, user nem fér hozzá a 3D kártyához, ezért a /dev/nvidia0 file jogosultságát állítsd a megfelelőre chmoddal (pl. 666).

KERNEL OPTIMALIZÁLÁS, FORDÍRÁS

Hát erről könyvet lehetne írni... Mire vagy kíváncsi? Tudsz-e alapszinten angolul? Ugye van szótárad? Van a közeledben valaki, aki segíthet, ha (nagy eséllyel) baj lesz? Tisztában vagy azzal, hogy pontosan milyen vasak vannak a gépházban? Tisztában vagy azzal, hogy mi kell neked, hálózat, NFS, kódkészlet, filerendszerek, tudod mi a fat32, iso9660, stb... és hogy kell-e neked? Tudod, hogy mit csinálsz? :-)

És még néhány kérdés. Ha egy kérdésre is NEM a válasz (kérdezhetnék még) akkor ne optimalizálj, ne forgass kernelt. Talán kivéve a van-e a közeledben valaki kivételével...

Maga a fordítás nem nagy kunszt. A kernelforrás helye a /usr/src/linux.
Na, mentsd el a régit, ide nyomd ki a letöltött új STABIL verziót.

tar xfvj kernel-2.4.21.tar.bz2

Szóval kicsomagoltuk, a /usr/src/Linux könyvtárba behajítjuk. A régiből a .config file-t átmásoljuk (rejtett file, ezt a Linux úgy jelzi, hogy "."-tal kezdődik. Ha nincs, az gáz... Akkor mindent neked kell bejelölni. De elvileg rajta kell legyen valahol a disztribúciód CD-jén. Szóval átmásoljuk mondjuk pistikonfig néven. Ezt követően jönnek a parancsok a forráskönyvtárban:

make mrproper
make menuconfig

ekkor jön a menü. Legalól van egy olyan, hogy Load Alternate Configuration File. Ebbe belemászunk, és beírjuk a konfig file-unk elérési útját.

/usr/src/linux/pistikonfig

Ekkor a konfig ua. mint a régi kernelnél. Na itt lehet elkezdeni hackelni. Akkor jó egy kernel, ha minél kisebb, ezért amit csak lehet, modulba kell forgatni. (* helyett M betű).
De ésszel kell csinálni, mert pl. ha a rendszer reiserfs filerendszeren van, akkor a reiserfs támogatást nem kéne modulba rakni, mert bootoláskor gyönyörű kernel panic felirattal találkozhatunk, és nézünk bambán.
Tehát változtatni ésszel, és az előző kernelt megtartva. Akkor boot CD-vel vissza lehet állítani a rendszert, ha valamit nagyon elbaltáztunk.
Tehát pl. hálókártya, hang, video, stb. mehet modulba, létfontosságú cuccok kernelbe.

A proci, ha nincs belőve, be kell, hogy milyen procid van, arra optimalizálja. Nyilván kernelbe... Alaplapi driver szintén.
Ami nem kell, ÉS TUDOD, HOGY NEM KELL, ki kell hajigálni. Minek 60 fajta hálókártya támogatás, ha neked csak egy db van, vagy nincs is hálókártya? Minek scsi támogatás, ha nincs scsi eszközöd és cd íród sem? Stb... stb...

Miután megvan a beállítás, mentsük el vmilyen néven a legalól lévő Save Alternate Config. segítségével, mondjuk a /usr/src/Linux/pistikonfig2 néven.

Ezt követően escape, és megkérdi, hogy mentsük-e a kernelbeállításokat? Nyilván.

Aztán a köv. parancsok:

make dep
make bzImage
make modules
make modules_install

És már kész is a legújabb kernelünk, modulokkal együtt. A kernelt az arch/i386/boot könyvtárban találjuk, bzImage néven. Ezt kell bemásolni a /boot könyvtárba valamilyen néven.

NE írjuk felül a régit, hanem (ha lilo boot managert használsz, akkor) a /etc/lilo.conf-ot szerkesszük úgy, hogy csinálunk egy plusz menüt. Az

eredeti vmi hasonló:

```
image=/boot/vmlinuz
    label=mandrake
    root=/dev/hda5
    stb...
```

Na ezt duplikáljuk, alámásoljuk, és értelemszerűen átírjuk az image sort (image = a kernel), mondjuk a vmlinuz helyett arra, amilyen néven elneveztük az új kernelt és beraktuk a /bootba. A cimkét (label) is átnevezzük mondjuk ujkernelre.

Itt jegyzendő meg, hogy BÁRMIKOR, BÁRMILYEN RENDSZERFILE-T MEGBUZERÁLUNK, ELŐTTE KÉSZÍTSÜNK BIZTONSÁGI MÁSOLATOT! Pl. a lilo.conf szerkesztése előtt mentsük el mellé lilo.conf.bak néven... (cp lilo.conf lilo.conf.bak)

Visszatérve, a többi dolgot a fileban békénhagyjuk. Kész is van, mentjük a konfig file-t, majd lefuttatjuk a lilo parancsot. Valami hasonlót kell látnunk a parancs hatására:

```
lilo

*Mandrake
 Windozer
 ujkernel
 floppy
```

Ekkor jön a reboot, és menüből az új kernelt kiválasztjuk. Ha gáz van, sebaj, ott a régi, azt választjuk a menüből.

Ha nem lilo-d van, hanem grubod (pl. UHU) vagy bármi más bootmanager, passz, még soha nem használtam.

Namármost, ha még véletlenül be is bootol a gép, akkor jönnek a gondok. Ami eddig kernelbe volt forgatva és most modulban van, pl. hang, video, hálókártya, akármi, az nem működik.

Be kell töltetni mindet, egyesével.

```
modprobe cdrom
modprobe ne2k-pci
modprobe mga
modprobe emu10k1
modprobe printer
modprobe blabla...
```

(Ez a sajátom egy része.) És láss csodát megszólal a hangkártya, stb...
stb...

Ezeket nyilván be kell lőni a boot scriptek egyikébe, hogy minden boot-kor betöltődjön már automatikusan.

LILO VISSZAÁLLÍTÁSA

Emberke winnyózt újrarakja, és nem tudja használni a Linuxot, mert eltűnt a lilo (grub, stb).
Ha a wint vagy bármilyen oprendszert felraksz, az a saját bootmanagerét rakja be a bootszektorba. Eddig a lilo volt ott, win telepít, és a win azt nyilván felülírta. A win viszont nem túl toleráns más oprendszerekkel. Nem gond, a Linux nem tűnt el, csak pillanatnyilag nem elérhető.

Vissza kell állítani a lilo-t, ami pofonegyszerű.

Bootcd (pl. mandrake 1. cd) berak, CD-ről boot, F1-gyet kell nyomni amikor mondja, "rescue" szót be kell írni amikor mondja, majd ha választási lehetőséget ad menüből, akkor a "Go to console" vagy hasonlót kell választani és máris kapunk egy konzolt. Nem kell bejelentkezni, be vagyunk, rootként.

Annyi a dolgunk, hogy egy üres könyvtárba felmountoljuk a Linuxos partíciót, és lefuttatjuk a lilo-t ARRA a könyvtárra, ahova felmountoltuk. Tehát:

Tegyük fel, hogy a Linuxos partíciónk a hda5. Ha nem tudnánk fejből, nem baj, be kell írni, hogy :

dmesg, és keresni egy olyan sort, ahol szerepel, valami ilyesmi:

hda: hda1 hda2 <hda3 hda5>

vagy valami hasonló. Nyilván a dmesg kimenete le fog futni a képernyőről, nem fog kiférni, de ez nem baj. Nyomva kell tartani a SHIFT-et, és a PGUP, PGDOWN-nal lehet lapozni a parancssoros képernyőn fel ill. le.

Ha pl. van egy win c: meg d: partíciónk, meg egy Linuxos, akkor ugye a fenti példában a hda1 a win c: a hda2 a kiterjesztett, amiben van egy hda3 (win d:) meg egy hda5. Akkor az uccsó valószínűleg a Linuxos partíciónk, a hda5.

Namost, lilo visszaállítása, ha hda5 a Linuxos partíciónk:

```
mount /dev/hda5 /mnt/disk
/mnt/disk/usr/sbin/lilo -r /mnt/disk
```

Már kész is vagyunk. Szépen egymás alá ki kell hogy írja a lilo menüjének sorait:

```
*Linux
 Windows
 Floppy
```

például.

Namost annyit kell itt megjegyezni, hogy a lilo parancsot most a rendszer nem látja alapból (nincs benne a PATH-ban) mint rendesen, ezért meg kell adni a teljes elérési útját... Disztribúciója válogatja, hogy éppen hol is a lilo. Nálam a /usr/sbin-ben. Nálad lehet, hogy a /usr/local/sbin-ben. Vagy a /sbin-ben, vagy a /usr/bin-ben,... Sajna ebben nem tudok segíteni, próbálgatni kell. De MOST, visszaállításnál elé kell írni annak a könyvtárnak a nevét, ahova felmountoltuk a Linuxos partíciónkat!!! (/mnt/disk) Mert a vinyón lévő rendszer gyökérkönyvtára nem a "/" lesz, hanem a "/mnt/disk", hiszen oda mountoltuk.

Célszerű ismerni a rendszeredet, legalább annyit megcsinálni, hogy míg nem cseszed el a win telepítéssel, előtte, ha másnem egy papírra felírni, hogy melyik is a rendszerpartíció, illetve a lilo hol is van.

Ez utóbbi, azaz bármilyen parancs helyének megkeresése a legegyszerűbb dolog a világon. Meg kell kérdezni a géptől (angolul, természetesen) hogy hol van a parancs, azaz:

whereis lilo

erre azt válaszolja (pl.), hogy

lilo: /usr/sbin/lilo

Ugye milyen egyszerű? EZ CSAK MŰKÖDŐ RENDSZERNÉL MEGY, CD-RŐL BOOTOLVA NEM!

Visszatérve: Ha a lilo lefutott, lehet rebootolni, és már műxik is.