

A Linux operációs rendszer kormányzati felhasználásának lehetőségei, nemzetközi tapasztalatok elemzése

Szerzők:

**Györkő Zoltán
Lippai Gergő
Littvay Levente
Varga Csaba Sándor**

Nyelvi lektor: Kósa Lajos

Készítette: BalaBit IT Kft.

2002. április 26.

Jelen tanulmány a Miniszterelnöki Hivatal Informatikai Kormánybiztosság felkérésére készült. A tanulmány felhasználói jogaival a KITKH (Kormányzati Informatikai és Társadalmi Kapcsolatok Hivatala) mint az IKB jogutódja rendelkezik.

1. VEZETŐI ÖSSZEFOGLALÓ	4
1.1. CÉLOK	4
1.2. HATÓKÖR	5
1.3. KÖVETKEZTETÉSEK	6
2. TÖRTÉNELMI ÁTTEKINTÉS	8
2.1. SZABAD SZOFTVER ÉS A LINUX TÖRTÉNETE	8
3. KORMÁNYZATI TRENDEK ÉS FELHASZNÁLÁSOK	12
4. SIKERTÖRTÉNETEK	16
5. SZOFTVERFEJLESZTÉS – SZERVEZÉSI MODELL	18
6. GAZDASÁGI MODELL	19
6.1. PIACI MODELL A FEJLESZTŐKSZÁMÁRA (ELŐNYÖKÉS HÁTRÁNYOK)	19
6.1.1. <i>Informatikai Oldalról Egyértelműen Látható Előnyök</i>	19
6.1.2. <i>A szabad szoftver Hátrányai az Informatikai oldalról</i>	21
6.2. ELŐNYÖKÉS HÁTRÁNYOK FELHASZNÁLÓI OLDALRÓL	21
6.2.1. <i>Felhasználók számára egyértelmű előnyök</i>	21
6.2.2. <i>szabad szoftver Hátrányai</i>	23
6.3. VERSENYHELYZET ÉS EGYÜTTMŰKÖDÉSEGYENSÚLYA	24
6.3.1. <i>A szabad szoftver projekteken belüli együttműködés</i>	24
6.3.2. <i>A szabad szoftver projekteken belüli verseny</i>	24
6.3.3. <i>A szabad szoftver projektek közti együttműködés</i>	25
6.3.4. <i>A szabad szoftver projektek közti verseny</i>	25
7. FINANSZÍROZÁSI MODELLEK	26
7.1. NEM FINANSZÍROZOTT SZABAD SZOFTVER PROJEKTEK	26
7.2. KÜLSŐLEGFINANSZÍROZOTT SZABAD SZOFTVER PROJEKTEK	26
7.2.1. <i>Állami finanszírozás</i>	26
7.2.2. <i>Cégektől származó anyagi támogatás</i>	27
7.3. BELSŐVAGY BEVÉTEL-ALAPÚ FINANSZÍROZÁS	27
7.3.1. <i>Szaktudásra építő szervezet</i>	27
7.3.2. <i>Kódra építő szervezet</i>	28
7.3.3. <i>Egyedi licenck</i>	28
7.3.4. <i>Márka teremtés</i>	28
7.4. BELSŐFELHASZNÁLÁSRA KÉSZÍTETT SZOFTVER	29
8. COTS VERSUS SZABAD SZOFTVER	30
8.1. TCO (TOTAL COST OF OWNERSHIP)	30
8.2. IT BIZTONSÁG	32
9. VERSENYHELYZET TANULMÁNYOZÁSA ÉS JAVÍTÁSA	34
9.1. FÜGGŐSÉGIHELYZET VIZSGÁLATA	34
9.2. MONOGÁMIA	35
9.3. KOMPATIBILITÁS ÉS ÖSSZEFÉRHETŐSÉG	35
9.4. MULTIK, KONTRA HELYI KIS- ÉS KÖZÉPVÁLLALKOZÓK	35
9.5. COTS BEFOLYÁSOLÁSA	35
10. JOGI KÖRÜLMÉNYEK TANULMÁNYOZÁSA	37
10.1. SZERZŐIJOG ÉS LICENCEK	37
10.2. SZABADALMAK PROBLÉMÁI	39

11. MEGVALÓSÍTÁS	40
11.1. VIZSGÁLATI SZEMPONTRENDSZER	40
11.1.1. Funkcionális definíció	40
11.1.2. projektháttér	40
11.1.3. Funkcionalitás	42
11.1.4. Dokumentáltság/kezelhetőség	42
11.1.5. Erőforrás igény	43
11.1.6. Sikertörténet	43
11.1.7. Konklúzió	43
11.2. KLIENS ANALÍZIS (MUNKAÁLLOMÁSOK)	43
11.2.1. Desktop környezet/Disztribúciók	43
11.2.2. Levelezés – kliensek	46
11.2.3. Irodai programcsomagok használata	49
11.2.4. Böngésző	54
11.3. KISZOLGÁLÓK	57
11.3.1. Levelezés – lista szerverek	57
11.3.2. Fájl kiszolgálás, PDC	59
11.3.3. Web kiszolgálók	62
11.3.4. Monitorozás	66
11.3.5. Központi napló feldolgozás	68
11.3.6. Backup, archiválás	69
11.3.7. Gyorsítótár	71
11.3.8. Fax szerver	72
11.3.9. Behívópont	74
11.3.10. Nyomtató szerver	75
11.3.11. Tűzfal	78
11.3.12. Virtuális Magánhálózatok – VPN	80
11.3.13. Adatbázis szerverek	81
11.4. EGYÉB LEHETŐSÉGEK	83
11.4.1. SMS és MCommerce Linux alatt	83
11.4.2. Kriptográfia és a fájlrendszer	85
11.4.3. VMWare	86
11.4.4. Személyi asszisztens (PDA-PIM)	87
11.4.5. Beléptető rendszer	89
11.4.6. HA, Load Balance	90
11.4.7. Network IDS	91
11.4.8. Sebezhetőség felmérés (scannerek)	93
11.4.9. Linux mint Router (WAN eszközök)	94
11.4.10. Tanúsítványkiadó (CA) funkciók	96
11.4.11. Távoktatás és tanulmány-koordináció	98
11.5. FEJLESZTÉSI LEHETŐSÉGEK	99
11.5.1. Központi címtár	99
12. KONKLÚZIÓ	101
13. KIEGÉSZÍTŐ ANYAGOK, MELLÉKLETEK	102
13.1. FORRÁSANYAGOK	102
13.2. 1. MELLÉKLET: LICENCEK	103
13.3. 2. SZ. MELLÉKLET: NÉHÁNY SZOFTVER DOKUMENTÁCIÓJÁNAK ELÉRHETŐSÉGE	115

1. Vezetői összefoglaló

1.1. Célok

A 2000 – 2001-es évet az áttörés évének nevezhetjük a szabad szoftver Mozgalom és annak legjelesebb képviselője, a Linux szempontjából. Egyes szakértők már évek óta hangoztatják, hogy a nyílt forráskódú, ingyenes eszközök használata nem csak lelkes egyetemisták múltó fellángolása – ahogy ezt a monopóliumukat féltő szoftvergyárak állítják –, hanem az informatikai ipar egy új, erős és versenyképes trendje. Ezt az állítást támasztja alá az is (ami miatt tulajdonképpen a közelmúltat az áttörés éveinek nevezhetjük), miszerint a legnagyobb cégek, meglátván a Linuxban az üzleti lehetőséget, teljes erőbedobással támogatni kezdték azt. A neves hardvergyártók szinte mindegyike (IBM, HP, Compaq, SGI stb.) Linux-alapú kis- és középkategóriás szerver családot dobott piacra (illetve mainframe-et, ld. IBM zSeries). A szoftverfejlesztő cégek ennek megfelelő eszközöket állítottak csatarendbe: a Borland előállt a Delphi programozási nyelv linuxos megfelelőjével a Kylixszal, az Oracle pedig kijelentette, néhány éven belül a PC-architektúra és a Linux lesz 100%-ban a kiszolgáló rendszerek alapja (és ezt alátámasztotta terméktámogatásainak Linuxra való kiterjesztésével is). A konkurens SUN megvásárolta és termékpalettájába integrálta a Red Hat Linux-alapú Cobalt szervercsaládot, valamint úgy döntött, hogy a Solaris operációs rendszer grafikus felületét adó CDE ablakkezelőt a nyílt forrású GNOME-ra cseréli.

Az LWN.net portál a Dow Joneshoz (BUX-hoz) hasonló tőzsdei indexet hozott létre a Linuxban érdekelt cégek valós tőzsdei teljesítményének mérésére. Az index jelenleg 12 olyan céget tart nyilván, amelyek elsődleges üzleti tevékenysége a Linuxban merül ki, és 18 olyat, amelynek termékeiben és/vagy bevételeiben az nagy szerepet játszik (csak az érdekesség kedvéért megjegyzendő, hogy például az Oracle még nincs köztük).

A szabad szoftver mozgalom és a Linux terjedésének trendszerűségét bizonyító fenti tények is nagyban hozzájárultak jelen tanulmány megszületéséhez.

A Miniszterelnöki Hivatal Informatikai Kormánybiztossága, felismervén a nyílt forráskódú eszközök informatikában betöltött pozíciójának jelentőségét, az „elektronikus kormányzat” irányelvének megfelelően ezen a téren is szerepet kell, hogy vállaljon.

Az eddigi két sikeresnek mondható projekt, az UHU Linux disztribúció fejlesztésének elindítása és az OpenOffice.org honosításával párhuzamosan (amely projektek eredményei talpkövei lesznek a továbblépésnek), szükség van a szabad szoftver modellel, valamint annak gazdasági hátterével kapcsolatos ismeretterjesztésre. Ezeken túlmenően a felhasználók, az adminisztrátorok, valamint az informatikai vezetők számára is tanulságos lehet, ha a vállalati környezetben használatos szoftverek, eszközök, nyílt forrású megfelelőit, még ha

nem is teljes körűen, de rendszerezetten számba vesszük.

1.2. Hatókör

Jelen tanulmány elsődleges célja, hogy áttekintést adjon a szabad szoftver Mozgalom egészéről, a nyílt forrású fejlesztési projektek mozgatórugóiról, valamint a felhasználásukból származó előnyökről. Másodlagos célként funkciójuk alapján rendszerezi a tipikus vállalati környezetben használatos informatikai eszközöket és bemutatja azok legnépszerűbb nyílt forráskódú megvalósításait. E tanulmánynak nem célja minden megvalósítás számba vétele, és az sem, hogy azokat teljes körűen, minden részletre kiterjedően elemezze. A csokorba gyűjtés értelme az, hogy az olvasó információt szerezzen a szabad szoftver közösség által előállított eszközökről és megbizonyosodjon azok reális felhasználhatóságáról.

A tanulmány második fejezete összefoglalja a szabad szoftver Mozgalom és ezen belül a Linux történetét.

A harmadik fejezet példákat ad a világ más országai, az EU szabad szoftverek támogatására indított kezdeményezéseiből, a kormányzati szintű felhasználásokról, míg a negyedik fejezet a hazai sikertörténetek közül emel ki néhányat.

A ötödik fejezet röviden ismerteti Eric S. Raymond, a szabad szoftver közösség egyik vezéralakjának, a nyílt forrású projektek felépítéséről szóló, alapvető írását, a Katedrális és Bazár-t.

A hatodik fejezet részletesen bemutatja a szabad szoftver gazdasági modelljét. Az elemzés kitér a nyílt forráskódú eszközök használatának előnyeire és hátrányaira, mind a fejlesztőket érintő informatikai aspektusból, mind pedig felhasználói szempontból. Ugyanezen fejezet vizsgálja a projektek mozgatórugóit a projekteken belüli és közti „versenyhelyzet-egyensúly” elv figyelembevételével.

A hetedik fejezet összefoglalja a nyílt forrású fejlesztési projektek finanszírozási modelljeit. Elmagyarázza, miként tud működni egy olyan folyamat, melyet senki sem finanszíroz, illetőleg rámutat, milyen forrásból kaphatnak közvetlen anyagi, valamint közvetett szakmai támogatást a fejlesztők.

A nyolcadik fejezet összehasonlítja a COTS (Commercial Off The Shelf Software – azaz kereskedelmi szoftver) termékeket és a szabad szoftvereket két igen fontos szempontból. Az első a TCO, azaz teljes bekerülési és üzemeltetési költségek szempontja. A tanulmánynak nem célja, hogy összegszerű előnyt állapítson meg bármely oldalon, kizárólag általános következtetések levonását segíti elő.

A második szempont az IT-biztonság kérdése, amely napjainkban egyre inkább előtérbe kerül.

A kilencedik fejezet a versenyhelyzet szempontjából hasonlítja össze a zárt forrású kereskedelmi termékek és a szabad szoftverek gazdasági szerepét. Megmutatja, hogy a zárt szabványok használata monopóliumok kialakulásához vezethet, valamint ronthatja az egyes termékek kompatibilitásának szintjét. Ugyanitt kifejtjük, miért támogatja a szabad szoftver modell a helyi kis- és középvállalatokat a multinacionális cégekkel szemben és miért szükséges a kormányzati beszállítók befolyásolása. A tizedik fejezet a szabad szoftverek magyarországi felhasználásával

kapcsolatos szerzői jogi összefüggéseket vizsgálja. A fejezetnek nem célja a pontos jogi állásfoglalás, de kiderül, hogy erre mindenképp szükség van.

A tizenegyedik fejezet a másodlagos célokkal kapcsolatos vizsgálatokat tartalmazza, azaz a tipikus informatikai felhasználások, szoftverek nyílt forrású megfelelőit mutatja be. Az alkalmazásokat funkcionalitásuk szerint három nagy kategóriára osztottuk. Az elsőbe csoportosítottuk a munkaállomások szoftvereit, amelyekkel a nem informatikai képzettségű felhasználóknak a legtöbb kapcsolata van. A második csoportba kerültek a vállalati munkakörnyezet háttérét adó kiszolgáló alkalmazások, míg a harmadikba az egyedi, vagy a fentiek egyikébe sem sorolható megoldások. A szoftvereket egységes szempontrendszer szerint vizsgáltuk, mely alkalmas arra, hogy bizonyítsa: az adott szabad szoftver méltó és érett a komoly, tömeges felhasználásra.

1.3. Következtetések

A tanulmány elsődleges következtetése, hogy a szabad szoftverek kormányzati szintű felhasználása támogatandó kezdeményezés, hosszú távon az alábbi előnyöket biztosítja:

- általános TCO-csökkenés a informatikai rendszerekben
- helyi kis- és középvállalkozások fokozottabb támogatása a multinacionális cégekkel szemben
- szoftver-monopólium kialakulásának megelőzése
- az informatikai rendszerek minőségének általános javulása a nyílt szabványok használata révén
- az informatikai rendszerek biztonságának javulása a nyílt szabványok és a nyílt forráskódú fejlesztési modell révén
- a kormányzati, állami és költségvetési intézmények informatikai eszköz ellátottságának és az ezek által nyújtott szolgáltatások minőségének javulása a kiszolgáló hardverpark (munkaállomások, szerverek) drasztikus megújítása nélkül is

E tanulmány által levont következtetéseket támasztja alá a nemzetközi, más államok kormánya által támogatott kezdeményezések megjelenése, illetve az ezekre és a piaci igényekre reagáló vezető informatikai cégek üzletpolitikája is.

A szabad szoftverek felhasználása az általános trendet követve növekszik mind az üzleti, mind a költségvetési szférában. Ahhoz, hogy utóbbiban ez ne csak alkalomszerű és bátortalan egyéni kezdeményezésekre történjen, egyértelmű kormányzati állásfoglalásra és támogatási rendszerre van szükség.

Ezek kialakítása komoly és összetett feladat, mely pontos tervezést igényel.

Az alábbiakban azokat a legfontosabb teendőket foglaltuk össze, amelyek a továbblépéshez szükségesek.

- Jelen tanulmány szerzői joggal foglalkozó fejezeteiben kifejtettük, hogy az
-

első és legfontosabb teendő az érvényben lévő magyar jogrend és a szabad szoftver licencek licencek kompatibilitásának vizsgálata. Ennek eredményei alapján, ha kell, akár törvénymódosítással is, megalkotható lenne a kormányzat által hivatalosan támogatott szabad szoftver licenc. A fentiekhez szükséges átfogó jogi tanulmánnyal Magyarország úttörő lehetne egy, az Európai Unió által már sokszor és sokféleképpen megfogalmazott irányelv útján.

- A magyar jogrendnek is megfelelő nyílt forráskódú szoftverek licencének megalkotása után a kormányzat által finanszírozott szoftverfejlesztésekben javasolt ennek használata, valamint a nyílt szabványok és platform-függetlenség támogatásának kötelezővé tétele.
- Javasoljuk nyílt forráskódú, teljesen magyar nyelvű Linux disztribúció kormányzati szintű támogatását. Itt szóba jöhet magyarított külföldi (pl.: SuSE, Red Hat, Mandrake) és teljesen magyar terjesztés (UHU Linux projekt) is. Az előbbiek magasabb fejlettségi szintje, míg utóbbi fejlesztési irányának kézben tarthatósága, illetve teljesen magyar nyelvű támogatása jelent előnyt. A preferált eszközök kiválasztása után szükséges az ehhez kapcsolódó szakkönyvek, oktatási segédanyagok és egységes felhasználói tanfolyam-, valamint vizsgarendszer kidolgozása. Amint ezek mind rendelkezésre állnak, a személyiszámítógép-forgalmazóktól elvárható, hogy – akár jogi szankciók kilátásba helyezésével – komplett gépet csak előre telepített, jogtiszt operációs rendszerrel értékesítsenek. Ez a lépés nagy mértékben csökkentené a hazai, az EU átlagot jóval meghaladó illegális szoftver használatot.
- Javasoljuk forráshiányos, informatikailag kritikus helyzetben lévő területeken pilot projektek indítását, melyek célja az adott intézmény rendszereinek szabad szoftver-alapúra való átállása. A pilot projektek eredményei megfelelő gyakorlati tapasztalatot nyújtanának a szabad szoftverek felhasználásának előnyeiről és nehézségeiről. A fentieket első körben olyan, relatíve izolált intézményekben érdemes megvalósítani, ahol az informatikai rendszerek integráltsági foka viszonylag alacsony. Ennek tipikusan megfelelő területek az oktatási intézmények és önkormányzatok.
- Az egészségügyben világ a más országait tekintve a szabad szoftverek teljes mértékben elfogadottak, ezért az ezek nyújtotta egészségügyi-informatikai lehetőségeket a terület szakértőit bevonva érdemes mélyen átvizsgálni.

Oktatási intézmények esetén kezdő lépésnek javasoljuk – a tanulmányban szereplő ausztrál példa alapján – olyan magyar Linux-alapú laborok létrehozását, melyek hardverszükségletét a jelenleg már elavultnak tekintett kis teljesítményű PC-k biztosítanák. Ennek megvalósításához érdemes a vékony kliens technológiát igénybe bevenni.

Természetesen a javasoltak nem fedik le a továbblépési lehetőségek teljes körét, de felvázolják a követendő irányelveket.

2. Történelmi áttekintés

2.1. szabad szoftver és a Linux Története

Ahhoz, hogy a szabad szoftver közösség kialakulását, a Linux és a nyílt forráskód történetét megértsük, meglehetősen régmúlt időkre (Microsoft előtti időkre), a UNIX rendszerek tündöklésének és bukásának idejére kell visszatekintenünk.

A szabad szoftverek és a Linux története több szálon bonyolódik és összefonódik a – történetét tekintve nem kevésbé érdekes – internettel.

Az első, talán legfontosabb szál az 1970-es évekre, az Amerikai Egyesült Államok keleti partvidékére nyúlik vissza. Itt élt akkoriban Richard M. Stallman (ismertebb nevén "rms"), aki 1971-től az MIT AI Lab (Massachusetts Institute of Technology Artificial Intelligence Lab, azaz a Massachusetts-i Tudományos Intézet Mesterséges Intelligencia Laborja) munkatársa volt. Abban az időben az MIT AI Lab projektje egy Digital PDP-10 típusú számítógépen futtatott egy ITS (Incompatible Timesharing System) nevű rendszert, Stallman feladata pedig ennek fejlesztése volt. Akkoriban a "szabad szoftver" fogalom még semmit sem jelentett, mivel minden szoftver szabad volt, bárki használhatta, átírhatta, nem volt sem licenc, sem trademark jelzés. Stallman megszállottságára jellemző, hogy munkahelyét otthonának, az ott dolgozókat pedig családtagjának tekintette, önmagát pedig igazi hackernek (klasszikus értelemben a hacker nem számítógépkalóz, vagy betörő, mint ahogy azt a mai képzavaros időkben mindenhol sulykolják, hanem olyasvalaki, aki szeret programozni, és élvezni, hogy ért hozzá).

Gazdasági és egyéb körülmények hatására a MIT-ben a 80-as évek közepére a PDP-10-es gépeket a Digital zárt forrású operációs rendszerével kezdték el használni – a Stallman által favorizált – ITS helyett.

Stallman önérzetét ez mélyen bántotta, és úgy döntött – a közösség, a felhasználók érdekeit szem előtt tartva –, hogy felépít egy operációs rendszert, a rendszermagtól kezdve egészen az olyan felhasználói programokig, mint például a levelező kliensek. Az alapelv a szabadságjogok biztosítása volt: az új rendszer legyen ingyenes, nyílt forráskódú, mindenki lássa, amit használ, ha úgy érzi, javítani kell rajta, megtehesse, aki segítségre szorul, az megkaphassa, aki módosított rajta, és ettől a program jobb lett, az terjeszthesse, és legfőképpen mindenki számára szabadon használható legyen.

Az új rendszer neve GNU lett, ami egy rekurzív rövidítés, és Stallman humorát dicséri: "GNU is Not Unix", azaz a "GNU Nem Unix" (a rekurzivitás használata egy elterjedt programozási módszer).

1984-ben Stallman kilépett a MIT-től, hogy a munkaviszonya ne befolyásolhassa a GNU szabad mivoltát (eme lépés nélkül a MIT igényt támaszthatott volna a GNU tulajdonjogára). A kilépés ellenére Stallman megtarthatta MIT-beli irodáját, ahol az

ezután következő években lakott, és használhatta a labor számítógépeit is.

A következő évek alatt Stallmann létrehozta azokat szabadszoftver-eszközöket, amelyek egy szabad operációs rendszer megírásához szükségesek, így az ő nevéhez fűződik többek között a GCC (GNU C Compiler) fordító, valamint a GNU Emacs editor megalkotása.

Stallmannak azonban szembe kellett néznie (legalább) két nem számítástechnikai jellegű problémával is, ezek közül az egyik a licencelés. Licenc nélkül bárki szabadon beépítheti a GNU-forrásokat (mivel a forráskód szabadon hozzáférhető) a saját kereskedelmi termékébe, ezáltal zárta téve azt. Ezért ki kellett dolgoznia egy olyan licencet, amely megvédi a GNU nyújtotta szabadságjogokat.

A licenc neve GNU General Public License, azaz GNU GPL lett. Ez a licenc igen kedvelt a fejlesztések körében, mert az általa képviselt és garantált nyitottság hozzájárul a projektek feldarabolódásának (forking) elkerüléséhez, valamint jogi védelmet nyújt vitás esetekben, amelyekben egyébként csak az emberi erkölcs lehetne mérvadó.

A másik „világi” probléma anyagi jellegű: a programozók, akik a GNU megalkotásában segítettek, nem pénzért tettek. Nem vártak el semmiféle ellenszolgáltatást, azonban érthető volt, nem mindenki engedheti meg magának, hogy csak lelkesedésből dolgozzon. Stallmannak rendelkezésére állt a MIT-beli iroda mint lakhely, azonban nem mindenkinek adatik ilyen lehetőség. Ezért hívták életre a Free Software Foundationt (szabad szoftver Alapítvány), amelynek célja a szabad szoftver mozgalom támogatása.

Stallman az általa kidolgozott szabad szoftver modellt, illetve az idealizált életformát szinte mániákus megszállottsággal védte, védi ma is, és bár pontosan ebből kifolyólag meglehetősen sok (természetesen békés) vitája van a közösség egyéb résztvevőivel, a mozgalom egyik legfontosabb vezéralakjának tekinthető. (Honlapja: <http://www.stallman.org/>)

A GNU rendszer fejlesztése tehát szépen haladt, egyre többen csatlakoztak a programozókhoz, előre tervezett és meglepetésszerű új projektek kezdődtek és fejeződtek be, azonban egy dolog még hiányzott a teljességhez.

Egy operációs rendszer legfontosabb eleme a rendszermag (kernel), amely a programok és a számítógépben található hardver közötti kommunikációt oldja meg. E nélkül nem működik semmi, de a GNU-nak nem volt használható rendszermagja. A kezdetektől fejlesztett kernel neve HURD (korábban Alix), mely igen lassan épült és még ma is távol áll a készről.

A GNU projektnek kapóra jött, hogy Linus Torvalds – egy finn fiatalember – a 90-es évek elején létrehozta a Linux rendszermagot.

Linus a Helsinkii Egyetem diákja (ma a Transmeta nevű amerikai cégnek dolgozik), elégedetlen volt Andrew S. Tannenbaum Minix kernelével, ami teljesen jogos is

volt, hiszen Tannenbaum, az Amszterdami Szabad Egyetem professzora (aki Stallmannal is kapcsolatba került C fordítóprogramja kapcsán), több informatikai alpmű szerzője, kizárólag oktatási célból írta a Minix kernelt.

Linus tehát nekifogott, és az adott eszközökkel (egy Intel 386 processzorral szerelt számítógéppel) létrehozott egy ingyenes, UNIX-szerű kernelt. Amikor úgy érezte, hogy a kód többé-kevésbé szalonképes, GNU GPL alatt közzétette az akkor még (jóindulattal) félkész Linux kernelt. Az újításra nagy igény mutatkozott, és rövid idő alatt rengeteg fejlesztő csatlakozott a mozgalommá terebélyesedő csoporthoz, hogy jobbá, használhatóbbá tegyék a Linux kernelt. A kezdeti, kizárólag Intel architektúrán való futtathatóság rég a múlté. Ma a Linux az x86-on kívül képes futni Compaq Alpha AXP, Sun SPARC és UltraSPARC, Motorola 68000, PowerPC, PowerPC64, ARM, Hitachi SuperH, IBM S/390, MIPS, HP PA-RISC, Intel IA-64, DEC VAX, AMD x86-64 architektúrákon is. Ezt a kernelt használja a mai napig a GNU rendszer, melyet azonban leginkább Linuxnak hívnak, a megfelelőbb GNU/Linux helyett.

Linus a kezdetek óta személyében fogja össze a linuxos társadalmat, Stallman mellett az „új világ” legnagyobb alakja. Sikerének titka a természetes úton kialakult, ám igen jól szervezett fejlesztői gárda. A GNU/Linux sikerének titka pedig végső soron megint csak visszavezethető a GPL-licenclésre: mivel a Linux ingyenes, szabad, valamint nem kötődik egyetlen céghez sem, de facto standarddá válhat. Ennek jelei már mutatkoznak a szerverek piacán, valamint a disztribúciók világában. A Linux elterjedésének nem feltétlenül kell egy teljesen új világrend eljövételét jelentenie, ahol mindennek ingyenesnek kell lennie (mint ahogy sok gyártó fél ettől), egyszerűen csak a kereskedelmi termékek szintje eggyel feljebb ugrik: a platform adott, stabil, megfelelő, erre lehet fejleszteni a megoldásokat. A platform neve pedig GNU/Linux. Ellenpéldáért nem kell messzire menni: a UNIX jó kezdés után hamar hanyatlásnak indult, helyét fokozatosan átveszi a Linux; az ok: a sok gyártó sokféle UNIX rendszere nem fér meg egymás mellett, zártak, drágák.

A szabad szoftver történetének másik meghatározó eleme a GNU/Linux mellett a BSD disztribúciók családja. A BSD története a 80-es évekre nyúlik vissza. Ekkoriban a University of California at Berkeley (Berkeley Egyetem, Kalifornia) Computer Science Research Group nevű csoportosulása létrehozott egy saját UNIX terjesztést, amit BSD UNIX-nak neveztek el (a BSD a "Berkeley Software Distribution" szavakból képzett mozaikszó). A BSD sokáig AT&T licenc alatt volt elérhető, mígnem a 80-as évek végén meg nem született az egyik első nyíltforráskód-licenc, a BSD. Sajnos azonban a BSD sokáig az AT&T-hez kötődött, mert a rendszer sok fontos része kereskedelmi termék volt. Ezt a problémát a 90-es évek elején sikerült kiküszöbölni, a 386/BSD-vel, amely már nem tartalmazott kereskedelmi elemeket, így létrejött egy teljesen BSD-licenclésű, azaz szabad platform.

Nem sokkal a 386/BSD megjelenése után létrejött a NetBSD csoport, mely fő

céljával a 386/BSD továbbfejlesztését tűzte ki, valamint a minél több platformon való elérhetőséget és támogatást. Fő terjesztési útvonaluk az internet volt, 1998-ig nem adták ki a disztribúciót.

A FreeBSD pár hónappal a NetBSD után készült el, célja kizárólag a PC-architektúra támogatása, valamint némileg egyszerűbbé tenni a használatot. Ezen felül a FreeBSD-ben rendelkezésre áll Linux-emuláció is, vagyis a Linuxra írt programokat lehet futtatni ezen a platformon is.

A legfrissebb BSD ág az OpenBSD, mely a NetBSD-ről szakadt le. Céljuk a biztonság mindenek felett való növelése, valamint az egyszerűbb kezelhetőség megvalósítása.

A BSD elterjedését azonban valamelyest visszavetette egy, az AT&T (illetve annak egyik leányvállalata) által 1992-ben kezdeményezett per, melyben az AT&T azt állította, hogy egy BSDI nevű cég (amely BSD-vel és annak kereskedelmi szintű támogatásával tervezett foglalkozni) megsértette az AT&T szabadalmait. A per végül 1993-ban oldódott meg azzal, hogy a Novell felvásárolta a felperes vállalatot. Fontos megjegyeznünk, a BSD licencek általában nem felelnek meg az FSF által megfogalmazott elvárásoknak, mivel a forráskód vagy annak módosítása akár zárttá is tehető.

3. Kormányzati Trendek és Felhasználások

Az alábbi fejezet kifejezetten a szabad szoftverek kormányzati szintű felhasználásairól szól, illetve külön kiemeli azon kormányzati vagy akár nemzetközi intézkedéseket, amelyek a szabad szoftverek elterjedését ösztönözik.

A szabad szoftverek elsősorban a fejlődő országokban nyújtanak komoly lehetőségeket, hiszen itt igazán költségérzékeny az informatikai büdzsé, emellett szorgalmas szakember vagy ambiciózus tanulni vágyó fiatal mindig akad. További előnyt jelent pl. a Linux közismerten alacsony hardverigénye, mivel ezekben az országokban a számítógépes ellátottság, illetve a felhasználók által birtokolt átlagos processzorteljesítmény messze elmarad a modernnek nevezhetőtől.

A fent vázolt állapottal szemben inkább a fejlett országok azok, ahol a szabad szoftverek támogatása állami szinten, szervezett módon zajlik.

Az Amerikai Nemzetbiztonsági Szövetség (NSA) már teljes egészében tudja alkalmazni a szabad szoftver modellt, és maga is beszállt a biztonságtechnikai szoftverek fejlesztésébe. Ennek eredménye a 2001 szeptember-októberében világot látott Security-Enhanced Linux, melyet most a Red Hat rendszerrel használ az NSA. A Linux nem csak az NSA-ban van jelen: az IBM adatai szerint a szövetségi kormánzatnak szállított linuxos rendszerek száma majdnem exponenciálisan emelkedik.

Forrás:

- http://www.linux.hu/article.php?id=14&LinuxHu_Session=5a9d192c3d80761111fdb3c9b264dda0
- <http://index.hu/tech/szoftver/opensource2/>

A felértékelődött biztonsági elvárásoknak tudható be az is, hogy a német adatvédelemmel foglalkozó szövetségi ügynökség szerződést kötött 3 céggel egy valóban biztonságos levelező rendszer létrehozására. A most készülő Sphinxet első körben a Kmailbe, az egyik legelterjedtebb grafikus, és a Muttba, az egyik legelterjedtebb karakteres levelezőprogramba szándékoznak beépíteni.

A Német Gazdasági Minisztérium jelentős pénzforrásokat biztosított a GnuPG kriptográfiai szoftver fejlesztéséhez.

A Bundestag felvette középtávú tervei közé informatikai rendszerének teljesen Linux-alapúvá tételét.

Forrás:

- http://www.linux.hu/article.php?id=16&LinuxHu_Session=5a9d192c3d80761111fdb3c9b264dda0
- <http://www.gnupg.org/aegypten/>
- http://www.linux.hu/article.php?id=237&LinuxHu_Session=5a9d192c3d80761111fdb3c9b264dda0

Franciaországban már rendelet is született arról, hogy ahol csak lehet, szabad

szoftvereket alkalmazzanak, továbbá a kormány kötelezővé tette a nyilvános szabványok használatát is. Franciaországban nem ez volt az első lépés a szabad szoftverek irányába, például a Francia Bányászati Tanács már régen a szabadszoftver-licencre épülő Mioga extranet eszközkészletet használja, továbbá a kulturális, a nemzetvédelmi és az oktatási tárca is átállt a Linux használatára.

Forrás:

- http://www.linux.hu/article.php?id=237&LinuxHu_Session=5a9d192c3d8076111fdb3c9b264dda0
- <http://index.hu/tech/szoftver/opensource2/>

Dániában egyre több önkormányzat veszi fontolóra a szabad szoftverek használatát. Ennek – bevallásuk szerint – fő oka, hogy a Microsoft a szoftverlízingelés erőltetésével arra kényszerítette az önkormányzatokat, hogy akkor is vásárolják meg a legújabb operációs rendszert és irodai programcsomagot, ha beérnek a korábbiakkal is. Ez és a nemrég bekövetkezett 30%-os licencár-emelés komoly ellenérzéseket váltott ki a hivatalnokokból. Ők úgy gondolják, a magasabb díjakért nem kapnak – nekik hasznos – további szolgáltatásokat, ezért első körben (2002 májusában) a StarOffice 6-os szoftvert vezetik be, amely munkaállomásonként mintegy 100 eurós megtakarítást jelent majd. Amennyiben a megoldás beválik, további változások várhatók.

Forrás:

- http://www.linux.hu/article.php?id=500&LinuxHu_Session=5a9d192c3d8076111fdb3c9b264dda0

Az angol kormány is letette voksát a szabad szoftverek mellett. 2001 decemberében készült el az irányvonalakat meghatározó kormányzati döntés első vázlata, amely egyértelműen a szabad szoftvereket részesíti előnyben. A szabályzat elsősorban a szabad forráskódú szoftverek minél szélesebb körben történő elterjesztésében hivatott segídeni a kormányzati és a nyilvános IT-szektorban. Fontos vizsgálati szempont az összeférhetőség (kompatibilitás) és a nyílt szabványok támogatása. A szabályzat eredetileg az Európai Unió eEurope – An Information Society for all című stratégiájára épül, amely célul tűzte ki, hogy 2002-re az Európai Közösség és a tagállamok a szabad szoftverek használatát részesítik előnyben a nyilvános szektorban és az e-kormányzatban.

Forrás:

- http://www.linux.hu/article.php?id=315&LinuxHu_Session=5a9d192c3d8076111fdb3c9b264dda0
- http://www.govtalk.gov.uk/rfc/rfc_document.asp?docnum=429

Az Európai Közösség Informatika-társadalmi Igazgatóság szabad szoftverekkel foglalkozó munkacsoportot hozott létre, amely 2000 áprilisára készítette el Európa szabad és nyílt forráskódú szoftverek adta lehetőségeiről szóló tanulmányát. Az elkészült anyag nagy segítséget nyújtott jelen dokumentum elkészítésében is.

Forrás:

- <http://eu.conecta.it/paper.pdf>

A fenti ajánlás szellemében készült el az Európai Unió nyilvános forráskódú webportál-szoftvere, a YIHAW is.

Forrás:

- http://www.linux.hu/article.php?id=237&LinuxHu_Session=5a9d192c3d8076111fdb3c9b264dda0

A távol-keleti példák közül Kínát érdemes külön kiemelni. Az IBM legnagyobb megrendelői között tartja számon a kínai postát, leginkább a Linux-alapú szervercsalád területén. Peking városának önkormányzata 2001 decemberében szerződést írt alá hat kínai szoftvercéggel, hogy irodai szoftvercsomagot és operációs rendszert szállítson a városnak. A hat cég között volt a legsikeresebb Kínai Linux disztribúciót, a Red Flag-et, fejlesztő cég is. A hetedik, visszautasított pályázó a Microsoft volt.

Forrás:

- http://www.linux.hu/article.php?id=381&LinuxHu_Session=5a9d192c3d8076111fdb3c9b264dda0
- <http://index.hu/tech/szoftver/opensource2>

Korea kormányának központi beszerző hivatala a HancomLinuxszal írt alá szerződést 120.000 Linux-alapú irodai munkaállomás 2002-ben történő leszállításáról.

Forrás:

- http://www.linux.hu/article.php?id=406&LinuxHu_Session=5a9d192c3d8076111fdb3c9b264dda0

A szabad szoftverek az oktatásban is számos sikert könyvelhettek el az elmúlt években. A Mexikói Red Escolar program kereteiben minden iskola egy Linux-alapú kiszolgálót és 7-27 db Linuxos munkaállomást kapna, melyeken a GNOME munkafelület futna. A választás többek közt azért esett a GNOME munkafelületre, mert a projekt alapítói mexikói állampolgárok. A kormány számításai szerint 600.000 – 1.000.000 darab számítógépre lesz szükségük a jövőben és a Linux használata körülbelül 104 millió USD (közel 30 milliárd forint) megtakarítást jelent. A projekt pilot része már sikerrel fut San Luis Potosí-ban.

Forrás:

- http://www.govtalk.gov.uk/documents/QinetiQ_OSS_rep.pdf

Természetesen az oktatásban gyakran önálló kezdeményezésre jönnek létre kisebb sikeres projektek. Egy ausztrál iskola nemrégiben újrahasznosította a régi, már szinte használhatatlan 386-os gépeit, és egy szerver segítségével vékonykliens-technológiára építő, könnyen adminisztrálható, teljesen működőképes rendszert alkotott.

Forrás:

- http://www.govtalk.gov.uk/documents/QinetiQ_OSS_rep.pdf

Több hasonló kezdeményezésről tudunk a hazai oktatásban is. Sokan arról számolnak be, hogy a tanárok meggyőzése volt a legnehezebb feladat, de amint hozzászórtak az új rendszerhez, még meg is szerették azt. Ezzel szemben néhol ma is, jobb híján egy már szinte kihalt rendszert, a DOS-t oktatják. Az ausztrál iskolában alkalmazott modell Magyarországon is beválna, a már használhatatlannak vélt számítógépekkel, ezek újrahasznosításával az informatikai színvonalat komoly pénzek megtakarítása mellett lehetne javítani.

A szabad szoftverek egészségüggyel kapcsolatos fejlesztései meglepően széles körben terjedtek el világszerte. Az Egyesült Államok kormánya által támogatott DHCP (Decentralized Hospital Computer Program, nem összetévesztendő az IP-címek elosztásáért felelős DHCP-protokollal) projektet 1982-ben indította a veteránok ügyeivel foglalkozó tárca. A tárca 1994-re 173 egészségügyi központot, 389 klinikát, 131 öregek otthonát és 39 lakóotthont üzemeltetett a program koordinálásával. 1995-ben a DHCP megtisztelő jelölést kapott a Smithsonian díjra, amit az informatikai lehetőségek legjobb kihasználásáért osztanak ki. A rendszert azóta több tucat intézmény implementálta, és a nyílt fejlesztés ma is aktívan folyik.

Forrás:

- <http://www.va.gov/dhcp.htm>

A Medical Record Document Type Definition tanulmány egy olyan szabad (XML) szabvány kifejlesztésének lehetőségét vizsgálta, amely lehetővé teszi az orvosi akták biztonságos elektronikus mozgását és hordozhatóságát.

Forrás:

- http://www.govtalk.gov.uk/documents/QinetiQ_OSS_rep.pdf

Az angol NHS (National Health Service) által készített tanulmány az EU és a helyi kormányzat irányelveinek megfelelően az egészségügyi informatikai eszközök nyílt forráskódú fejlesztésére biztat. A tanulmány részletesen kifejti a kereskedelmi termékek használatának hátrányait és a szabad szoftverek előnyeit. Ez a dolgot alapul szolgálhat egy, a hazai viszonyokat feltérképező anyaghoz is.

Forrás:

- http://www.nhsia.nhs.uk/def/pages/features/i_250202.asp

A fejlett és fejlődő országokban létrejött összefogások és a bemutatott példák bizonyítják, hogy a szabad szoftver Mozgalom, a Linux és más nyílt forráskódú szoftverek terjedése nem futó fellángolás, hanem az informatika fejlődési irányát követő erőteljes trend.

4. Sikertörténetek

Informatikai szakértők hazánkban még gyakran találkoznak azzal a problémával, hogy az az ügyfél, akinél Linux-alapú rendszert szeretnének bevezetni, idegenkedik ettől a gondolatától. Az ellenérzéseket legtöbbször az okozza, hogy az ügyfél azt hiszi, ha linuxos rendszert használ, akkor valamilyen bátor, újszerű, de kockázatos tettet hajt végre. A Linux-felhasználók Magyarországi Egyesülete ezért a 2001-2002-es évben folyó ismeretterjesztő előadásainak tematikájába felvette a „Sikertörténetek” című fejezetet. Az előadások visszhangja minden esetben pozitív és a hallgatóság számára meglepő volt. Az előadásokat tartó szakember a linuxos közösséghez intézett e-mailben kért olyan példákat, ahol szabad szoftvereket használnak vállalati környezetben. Az elért eredmény minden várakozást felülmúlt, közel száz eseteleírást kapott. A példák között több is akadt, ahol a beküldő nem engedélyezte a publikálást, mivel a Microsoft-alapú kiszolgáló rendszert felsőbb engedély nélkül, a felhasználók számára észrevehetetlenül cserélte Linuxra, és akadt olyan példa is, ahol attól féltek, hogy a MS-szerver cseréje miatt a Microsoft-vizionteladó felbontja a kliensekre vonatkozó kedvező keretszerződéseket.

A sikertörténetek alapja minden esetben az, hogy a Linux-terjesztések 1500-2000 programot fognak csokorba, és az egyik legismertebb, fejlesztőket és felhasználókat összefogó fórum, a www.sourceforge.net jelen pillanatban közel 37.000 GPL/BSD és egyéb, nyílt fejlesztési politikával rendelkező projektet tart nyilván.

A beérkezett eseteleírások között számtalan profitorientált cég és jó pár olyan iskola vagy non-profit intézmény is szerepelt, melyek említésre méltók jelen tanulmány keretein belül.

- **Linux-terjesztő központ:** A szabad szoftver közösség büszkesége az ftp.fsn.hu néven ismert letöltő hely, ahonnan mindig gyorsan és zökkenőmentesen lehet elérni a legfrissebb Linux-terjesztéseket. Nem is olyan régen még 2-3 gépen szolgálta ki egész Magyarországot és Európa egy részét. Adatforgalma megdöbbentő, napi 400 Gbyte, havi kb. 15 Terabyte, amely egy közepes internet-szolgáltató havi forgalmával egyenlő. Az eleinte Debian GNU/Linux-alapú rendszert később teljesítményokokból FreeBSD-re cserélték. A gépet egy egyetemista álmodta és valósította meg. Talán jelen pillanatban is ez a legnagyobb Linux-tükör Magyarországon.
- **Hógyészi Hunyadi Általános Iskola, Zeneiskola és Óvoda:** Kis költségvetésű iskolaként 486-os gépeket kezdett diplomamunka gyanánt Linuxra átalakítani egy visszajáró öregdiák. Az iskola minden gépén és minden célra Debian/GNU Linuxot használnak a mai napig, és a Linux természetesen bekerült a tananyagba is.
- **Tapolca Liftvezérlés:** Az alábbi példában az emberi találékonyság párosul

a Linuxban megbúvó számtalan fantasztikus lehetőséggel. 9 db 10 emeletes házban 18 db non-stop működő liftet vezérelnek egy darab 486-os Linux-szerverrel. A megbízható monitorozás és a Linux 99,99%-os rendelkezésre állásának köszönhetően az ott lakó 4000 ember ritkán kényszerül csak – az amúgy egészséges – gyaloglásra.

- **Vas Megyei Bíróság:** egyike azon hivataloknak, ahol minden feladatra, szerver- és kliens-megoldásra is Linux-rendszereket használnak. A számtalan előny mellett (megbízhatóság, biztonság, magas rendelkezésre állás) több olyan esetről is beszámoltak, ahol a pályázaton nyert gépek mellé adott MS-termékek visszamondásával újabb gépekhez jutottak, plusz költség nélkül.
- **Habar Kft., Szarvas:** A két telephellyel rendelkező cég pár Linux-alapú szerverrel, minimális személyzettel teljesen automatizált pulykanövelvét hozott létre. A rendszer automatikusan gondoskodik a 80.000 szárnyasról: monitorozza a környezetet (pl. páratartalom, hőmérséklet) és szabályozza az etetést, itatást.
- **OpenOffice.org/hu:** Egy lelkes, a szabad szoftverek híveiből álló csapat 3 nap alatt a GNU-alapfilozófiát követve az interneten keresztül honosította az Openoffice.org terméket. A nyílt forrású OpenOffice.org magyar felültével és a magyar helyesírás-ellenőrzővel mindenképpen méltó ellenfele lehet az irodai programcsomagok terén szinte egyeduralkodó más office-termékeknek.

Rengeteg történetet tudnánk itt felsorolni (és számuk napról napra nő), de terjedelem hiányában erre sajnos nincs mód. Következtetésként talán levonható, hogy a Linux és GNU együtt alkotja a dinamikus jövőt, melynek előretörése lassítható ugyan, de eljövetele megállíthatatlan.

5. Szoftverfejlesztés – Szervezési modell

A kereskedelmi és szabad szoftverek fejlesztési modellje közti különbséget már sokan próbálták megmagyarázni, talán a legsikeresebb ezen a téren Eric S. Raymond volt, aki maga is egy sikeres szabad szoftver projekt vezéregyénisége. Raymond a középkori építészeti gyakorlatokhoz hasonlítja a szoftverfejlesztést. Ennek egyik módja egy központilag tervezett, a kivitelezésben erős kézzel kontrollt gyakorló, tiszta irányvonalakat szabó módszer. Ezt a katedrális építéshez hasonlítja, ami ugyancsak komoly koordinációt igényelt a középkorban. A másik véglet az elsőként kaotikusnak tűnő, de mindenképpen saját működési rendszerrel rendelkező, szabad információáramlásra alapuló, informális kommunikációra építő rendszer, amely talán a középkori bazárhoz hasonlít. A bazárban mindenki tudja, hogyan kell viselkedni, és aki már ismeri ezt a környezetet, az, a már említett informális csatornákon keresztül, rögtön megtudja, hol lehet a legfinomabb almát kapni. Ez a szoftverfejlesztésben sincs másképpen.

Mindkét modell működőképes, de Raymond tehetségesen domborítja az utóbbi rendszer előnyeit. Ezek közé tartozik, hogy a szabad információ és forráskód lehetővé teszi a fejlesztési hibák gyors napvilágra kerülését. Elsősorban emiatt a jó tulajdonság miatt a „bazárelven” működő fejlesztési struktúrákat állandóan fejlesztik, és az új fejlesztések nagyon hamar napvilágot látnak. Sőt, az ilyen módszerrel készülő programok gyakran rendelkeznek egy stabil verzióval és egy éppen fejlesztés alatt állóval, amely a hibajavítás fázisában van. Ezáltal a felhasználó eldöntheti, számára mi a fontos: a legfrissebb funkciók használata vagy a 100%-os üzembiztonság. Ez a két külön kiadás gyakran egy, a fejlesztést gyorsító erőként is hat, mivel az ötletek akármilyen forrásból érkehetnek, és azok kipróbálására a gyakorlatban is lehetőség van. Ha egy ötlet bevált, akkor belekerül a stabil verzióba, ha nem, akkor elvetik.

Egyértelmű, hogy szabad szoftverek esetében mindkét modell jól használható, ezzel szemben a klasszikus piaci termékekénél csak a fejlesztés „katedrális”-megközelítése lehetséges. Vannak központilag szervezett szabad szoftverek, de a bazármodell elterjedtebb, hiszen ilyen programok fejlesztésébe szívesebben szállnak be külső fejlesztők is. A már említett informális kommunikációhoz a világhálón működő levelezőlisták és fórumok adnak jó alapot. A fejlesztők vonzódása ezen projektekhez a demokratikus felépítéssel magyarázható, hiszen mindenkinek lehetősége van olyan szinten beépíteni magát a környezetbe, hogy elismerést szerezzen társaitól. Az elismerés egyben a projekt irányításába való nagyobb beleszólást is jelenti, amely lehetőség nem összehasonlítható a fentről küldött feladatok megoldásának szűrkeségével.

6. Gazdasági Modell

E fejezet célja, hogy átfogó képet adjon a szabad szoftverek előnyeiről és esetleges hátrányairól a kereskedelmi, zárt forráskódú termékekkel szemben. A fejezet első részében fejlesztői oldalról közelítjük meg a problémát. Az előnyök bemutatása során fény derül többek közt arra, hogy miért is lett sikeres a szabad szoftver modell, míg a hátrányokról szóló rész leírásként fog szolgálni a szabadszoftver-fejlesztőket érintő főbb kihívásokról.

6.1. Piaci modell a fejlesztők számára (előnyök és hátrányok)

6.1.1. Informatikai oldalról egyértelműen látható előnyök

A szabadszoftver-fejlesztés teljes mértékben a piaci kereslet-kínálat elvét követi. Kereskedelmi termék létrehozásánál kizárólag gazdasági döntés, hogy egy lassan induló vagy éppen nem az elvárások szintjén teljesítő szoftver fejlesztését megszüntessék vagy felfüggeszsek. Ebben az esetben ritkán veszik figyelembe azon felhasználókat, akik már megvásárolták a terméket, és számítanak annak hibajavításaira, továbbfejlesztésére. A másik alternatíva, hogy a fejlesztő cég azért viseli el az irreálisan nagy kiadásokat, mert hisz a termékében. A tisztán (radikálisan) szabad szoftverek esetében nincs kiadás, és a projekt csak addig él, amíg fejlesztői nem unják meg. Mivel szabad szoftverről van szó, ezért a fejlesztés probléma nélkül folytatódhat, ha valaki átvállalja annak menedzselését.

A szoftvergyártók nem a direkt értékesítésből szerzik bevételeik nagy részét (nem közvetlenül a végfelhasználóknak árulják termékeiket), hanem a viszonteladói, disztribútori csatornáikon keresztül. Ebben az értékesítési láncban megjelennek a rendszerintegrátorok is, akik a vevő számára általában olyan komplett megoldást szállítanak, amelynek csak egy összetevője az adott szoftver. Triviális megállapítás, de a rendszerintegrátorok nagy mértékben tudják növelni versenyképességüket és profitjukat a szállított eszközök beszerzési árának csökkentésével. Az informatikai üzletben trendnek mondható, hogy a szolgáltatás és rendszerintegrációs jellegű bevételek (másik oldalon költségek) egyre nagyobb hányadát teszik ki a beruházásoknak. Az összetett feladatokat megvalósító és jó minőségű informatikai rendszer felépítésénél a szabad szoftver használata két összefüggő okból is előnyt jelent. Egyrészt az ügyfél által igényelt testre szabhatóság és rugalmasság a forráskód birtokában szinte 100%-os, másrészt ezeket a kisebb módosításokat a szoftver gyártójának lassú és drága apparátusának bevonása nélkül házon belül is elvégezhetjük.

A szabad szoftver eddig ismeretlen mértékű mozgásteret biztosít azon informatikusok számára, akiknek megoldást kell találniuk egy adott problémára. A szabad szoftver ingyen működésbe helyezhető és kipróbálható. Így nem kell licenceket megvásárolni, pénzt kiadni drága programokra, amikről használati tapasztalat után kiderülhet, nem teljesítik elvárásainkat, nem beszélve a demo-, vagy kölcsönkapott kereskedelmi termékekről, melyek gyakran okoznak szerzői jogi problémákat. Ez az azonnali implementáció az informatikai megoldáskeresés folyamatának időigényét töredékére csökkentheti.

A szabad szoftver közösség volt az első, amely felfedezte a forráskód-újraterjesztés pozitívumait. Az összgazdasági produktivitás (feladatmegoldás sebessége) exponenciálisan nő, ha nem kell mindenkinek újraterjesztetni a már más körülmények között elkészült és ugyanarra a feladatmegoldásra képes programmodulokat.

A szabad szoftver közösség a többség álláspontja szerint igen tehetséges, bár nem ritkán munkájukért fizetséget nem kapó programozókból áll. Ezt a közösséget olyan emberek alkotják, akik nem elsősorban a gyors és egyszerű meggazdagodás lehetőségét keresik, hanem hosszú távon szeretnék alkotásaikkal a biztos megélhetésüket megalapozni. A gyors karrierhez általában a menedzsmentbe való betörés biztosít sima utat, nem a szoftverfejlesztés, ezért gyakran a legtehetségesebb, legtöbb tapasztalattal rendelkező és legfegyelmesebb, a kódalkotás iránt legelhivatottabb programozók a szabad szoftver mozgalom kulcsemberei.

A szabad szoftver közösség háttérét adó programozók száma jóval nagyobb, mint bármely kereskedelmi terméket gyártó cég szakembereié. A szabad szoftverek fejlesztésének egyik alapvető eleme a "**release early, release often**" elv, amely szerint a programozó minél korábban tegye a nagyközönség számára elérhetővé az általa fejlesztett eszközt, a benne található hibáktól és funkcionális hiányosságaitól függetlenül, illetve gyakran bocsásson ki új verziót. Ezen elv hatására a közösség más érdeklődő tagjai és részt tudnak venni a fejlesztésben, a felhasználók visszajelzései által pedig rendkívül gyorsá válik az eszköz tesztelése, hibamentesítése. A szabad szoftver projektek sikerességének egyik kulcsa a **kritikus tömeg** megszervezése, azaz ha az adott eszközt elég sokan használják, az maga után vonja további fejlesztők és esetleges szponzorok belépését is. A korai kibocsátás ténye és a kritikus tömeg megszervezésének igénye pedig szükségessé teszi a kibocsátott verziókkal kapcsolatos bejelentett hibák pontos követését és a gyors hibajavításokat. Ezt fejlett hibajavító és követő eszközök alkalmazásával lehet megoldani. A szabad szoftver közösség már rendelkezik olyan, mindenki által könnyen használható, az internetes kommunikációs csatornákat rugalmasan kezelő "bugtracking" (hibafigyelő, hibakövető) rendszerekkel, melyeket COTS termékgyártó is próbált már implementálni (általában kevés sikerrel.) Ez utóbbi

bizonyosságaként érdemes a biztonsági hibákra specializálódott levelező listákon a hiba bejelentése és a hibajavítás megjelenése közt eltelt időt megfigyelni. (A nyílt forráskódú termékek gyártói – kevés kivétellel – több nagyságrenddel gyorsabbak.) A szabad szoftverek jó minőségének és funkcionális teljességének biztosítéka, hogy ha egy felhasználónak valamely tulajdonságra szüksége van, gyakran saját maga fejleszti bele azt az adott eszközbe, az elkészült módosítást pedig elküldi a szoftver eredeti fejlesztőjének, azzal a céllal, hogy tegyék bele a fő verzióba, így verzióváltás esetén nem neki kell foglalkoznia saját módosításainak illesztésével.

A világ egyetemének és kutatási intézeteinek projektjei javarészt szabad szoftver alapokra épülnek, illetve előszeretettel hasznosítanak újra már megírt szabad kódot. Azok a diákok, akik tanulmányaik alatt bensőséges kapcsolatba kerültek a szabad szoftverekkel, gyakran ezt az irányt képviselik későbbi munkájuk során is. Sokan ezzel magyarázzák a Linux egyre növekvő népszerűségét az üzleti szférában.

6.1.2. A szabad szoftver hátrányai informatikai oldalról

A szoftverfejlesztőknek alternatív bevételi forrás után kell nézniük, mivel a szoftver mint termék (licenccij) mostantól nem jelent bevételt. Ezek a források lehetnek az eszközzel kapcsolatos szolgáltatások (integráció, testre szabás, támogatás), dokumentációk és könyvek kiadása, oktatási tevékenység stb., éppen ezért a modell nehezen alkalmazható a piac egyes területeire, mint például a játékprogramok piacára, ahol szinte kizárólag a licenccijből származnak bevételek.

Befektetők számára még mindig nem világos, miként működik a szabad szoftver modell, éppen ezért a befektetési kedv egyelőre alacsony.

Külső támogatással nem rendelkező szabadszoftver-termékeknek nincs közvetlen marketing-költségvetése.

Ezen termékeknek nincs (illetve ha van, akkor alacsony) fejlesztési költségvetése, ezért a felhasználóbarát funkciók és a széles körű dokumentáció a termék megjelenéséhez képest viszonylag későn jelennek meg. Mivel a klasszikus értelemben vett szabad szoftver projektek résztvevői lelkes fejlesztők egyértelmű projektvezetés és irányvonal nélkül, viták nehezíthetik meg a résztvevők életét, amely vita a legvégső esetben a projekt elágazásához (fork) is vezethet.

6.2. Előnyök és hátrányok felhasználói oldalról

6.2.1. Felhasználók számára egyértelmű előnyök

Egy informatikai beruházásban a bekerülési ár csak töredéke a hosszú távú üzemeltetési költségnek, mégsem elhanyagolható, hogy a szabad szoftvert nem kell megvásárolni. Nagy mennyiségben a szoftverlicenc komoly kiadásnak számít, főleg

a géppark bővítésénél és az upgrade-nél. Ehhez hozzájárul az is, hogy a kereskedelmi terméket gyártó cégek gyakran a saját munkájuk megkönnyítése érdekében az új verziók megjelenése után nem sokkal megszüntetik a támogatást a régebbi verziókra, ezzel is kényszerítve a felhasználót a fejlettebb verziók megvásárlására.

A kereskedelmi szoftverek árpolitikájára általában jellemző, hogy „ami olcsó, azt biztos nagy mennyiségben kell megvásárolnunk (minden gépre), amiből pedig nem kell sok (pl. szerveralkalmazások), annak ára a kliensek számától függ”. Ezt tovább súlyosbíthatja az is, ha a licencköltséget nem lehet időben elosztani (pl. lízing), hanem egy összegben kell kifizetni.

Nem elhanyagolható az a költség sem, amelyet egy, a kereskedelmi licenceinket nyilvántartó adminisztratív és technikai komponenseket egyaránt igénylő rendszer kialakítására kell költenünk, nem beszélve a büntetésről, amit ennek hiányában könnyen a nyakunkba varrnak.

A szabad szoftver modell már a fejlesztés korai fázisban a felhasználásra ösztönöz (release early, release often), ez pedig elősegíti a hibák kiszűrését, és biztosítja, hogy mire a fejlesztők a programot a stabil jelzővel látják el, az valóban így is működjön.

A nyílt forráskódú fejlesztések nagy előnye a nyílt szabványok támogatása (pl. IETF RFC-k vagy az XML dokumentumformátum), ami kereskedelmi (zártforrású) termékeknél nem minden esetben tapasztalható.

A nyílt forráskód következménye az is, hogy a sikeres eszközök jóval portábilisabbak operációs rendszer tekintetében, mint a COTS-termékek. A nagyszámú szakértő-felhasználó tipikusan különböző operációs rendszerekért rajong, és addig nem nyugszik, amíg a szükséges eszközt a neki megfelelőre le nem fordította. Ebben segíti őt a nyílt szabványok használata. COTS-termék esetében ezt a portabilitást a cég szakembereinek ismeretei, valamint a piaci igények kielégítéséhez feltétlenül szükséges minimum határozzák meg.

A nyílt forráskód lehetővé teszi, hogy házon belül kezeljünk esetleges szoftverhibákat és biztonsági réseket. A biztonsági rések hibajavításai általában minimális forráskódbeli változtatást igényelnek (és a javítás javasolt módját gyakran már maga a hibabejelentő megteszi), mégis heteket kell várunk erre egyes kereskedelmi termékek esetében, nem beszélve a hibás működés javításáról, amelyet verzióváltásonként kb. félévente kapunk meg.

COTS-termékeknél bevett gyakorlat a termékkapcsolás. Ekkor egy bizonyos termék csak akkor működik (teljes funkcionalitással), ha a gyártó egy másik termékét is megvásároljuk. A szabad szoftvereknél is létezik ez a kapcsolás (nevezzük függőségnek), de itt biztosak lehetünk benne, hogy nem gazdasági,

hanem kizárólag technikai okokból van kapcsolat. Továbbá a szabad szoftver függősége ugyancsak szabad, tehát annak beszerzése és telepítése sem jelent külön költséget.

Szabad szoftverek esetén nem csak egy bizonyos forrásból lehet hozzájutni terméktámogatáshoz és továbbfejlesztéshez. Ez a tény biztosít a támogatásra egy bizonyos redundanciát (ha az eredeti meg is szűnik, más források rendelkezésre állnak). További előny, hogy ha több cég verseng a szoftvertámogatás lehetőségéért, akkor ez a piac szabályai szerint az árakat is lefelé viszi (COTS esetén alsó határt szab a gyártó által a hivatalos támogatásért kért ár).

Amennyiben helyi támogatás nem elérhető egy bizonyos szabad szoftverre, de erre igény lenne, akkor a piac szabályai tisztán tudnak érvényesülni, és a támogatás meg fog jelenni egy új cég színeiben. Mivel egy szabad szoftverről minden információ adott, egy szakember probléma nélkül meg tudja érteni annak működését és támogatást tud nyújtani (és nem csak azokat a kérdéseket tudja megválaszolni, amit már ő is megkérdezett egyszer a szoftver gyártójától). A kereslet tehát meg tudja teremteni a professzionális támogatást. Hazánkban az informatikai oktatás relatíve magas színvonala, továbbá a szabadszoftver-szakértők és felhasználók népes taborát tömörítő több szervezet (LME, BSD, FSF.hu, FSN.hu) elhivatottsága, valamint egy, már a 3. évfolyamát megért, Linuxszal foglalkozó informatikai szaklap léte is ezt támasztja alá.

A szabad szoftverek köré általában teljes internetes közösségek tömörülnek, amelyek levelezőlistákon és fórumokon többnyire a COTS-terméktámogatásnál gyorsabban nyújtanak segítséget egy-egy probléma megoldására. Ez a segítség akár a forráskód módosítását is jelentheti, ami hagyományos termékek támogatóitól nem várható el.

A belső kezdeményezés keretében indított szoftverfejlesztéseket érintő problémák jellemzően a fejlesztési módszertan hiányából fakadnak. Kis cégek esetén ezeket a feladatokat általában a rendszer üzemeltetését végző szakemberek kapják, akik a leggyorsabb és legkönnyebb megoldásokat választják. Tipikusnak mondható az is, hogy egy-egy ilyen alkalmazott távozása után az adott szoftvert (a forráskód minősége miatt) más továbbfejleszteni nem képes. Természetesen ennek elkerülésére leginkább a megfelelő szoftverfejlesztési módszertan (projektvezetés) kiválasztása és használata alkalmas, de sokat segíthet, ha a fejlesztők tudják: az általuk írt szoftver nyílt forráskódú lesz, mivel így motiválja őket a szakmai elismerés lehetősége is, valamint az is, hogy tudják, mások is láthatják, milyen minőségben dolgoznak.

6.2.2. A szabad szoftver hátrányai

COTS-szoftver esetén a gyártó alapvető érdeke, hogy támogatást nyújtson a termékéhez. Szabad szoftver esetén ehhez először a kritikus tömeg, majd a piaci igény elérése szükséges.

Mint már említettük, a felhasználóbarát funkciók és dokumentációk lassabban készülnek el.

A szabad szoftver bevezetésénél gyakran ütközhetünk olyan félreértésekbe, melyek a modell komplexitásából és a COTS-cégek által terjesztett, a szabad szoftverek lejáratására irányuló valótlán információkból származnak. Ezek a szabad szoftver modell ismeretében teljes mértékben tisztázhatók. A leggyakoribb félreértések az alábbiak:

- Ami ingyenes, az nem megbízható!
- Senkit sem lehet perelni, ha szoftver problémákat okozott.
- Mivel nem én fizetem a fejlesztőket, semmi ráhatásom a munkájukra.
- Mivel nem pénz, hanem hírnév hajtja a fejlesztőket, ezért kiszámíthatatlanok. Ha találnak egy jobb projektet, akkor nem tudom továbbfejlesztetni az eszközt.
- A szabadszoftver-fejlesztők nem értik a piaci szükségleteket (például visszafelé kompatibilitás, szoftverintegráció fontossága).

6.3. Versenyhelyzet és együttműködés egyensúlya

A legmagasabb produktivitást a versenyhelyzet és együttműködés egyensúlya teremti meg. Ennek lényege, hogy míg a különböző projektek versengenek az erőforrásokért, piaci elfogadottságért, addig a forráskód megosztásával egymást segítik a fejlesztésben. Az alábbi fejezet ezeket a folyamatokat vizsgálja részletesebben.

6.3.1. A szabad szoftver projekteken belüli együttműködés

A projekteken belüli együttműködés általában felülmúlja a COTS-készítésen belüli kommunikációt és együttműködést. Ennek oka, hogy COTS-fejlesztési projekteken gyakran még a programozók között is korlátozott az információ-áramlás az üzleti titkok védelmének érdekében. A programozó a kapott információ alapján saját kreativitását kevésbé használva végrehajtja az adott feladatot, és nem törekszik a legjobb megoldás megtalálására. Ehhez az is hozzájárul, hogy a megkapott, korlátozott információ nem elég a magasabb szintű szemléletmódhoz.

Ez a szabad szoftver projekteken teljesen másként működik. Ott a fejlesztési feladatokat a programozók maguk vállalják és nem fentről várják az utasításokat, éppen ezért érdekelték és aktívak a kommunikációban. Emellett szerepet játszik az is, hogy a kommunikáció elsősorban a fórumokon és a jól archivált levelezőlistákon folyik, tehát szabadon kereshető és hozzáférhető bárki által.

6.3.2. A szabad szoftver projekteken belüli verseny

A szabad szoftver projektek általában egy érdemi hierarchia köré épülnek. Azok kapják a legtöbb elismerést és befolyást, akik a legtöbbet tették a projektért. Egyebek közt ez is oka, hogy a fejlesztők aktívan figyelik a kommunikációs

csatornákat, ellentétben a COTS-fejlesztőkkel. Továbbá nagyon fontos felismerni azt a lehetőséget, ahogy egy cég vagy cégközeli programozó szerepvállalása a szabad szoftver projektben komoly befolyást biztosít.

6.3.3. A szabad szoftver projektek közti együttműködés

A szabad szoftver alapelveiből ered, hogy mindenki hozzáfér a forráskódhoz. Amint egy új ötlet előtérbe kerül az adott szoftver új verziójában, arról a többi projekt fejlesztője – ha szemfüles – azonnal értesül, és beépítheti saját programjába. Ez természetesen előnye és hátránya is a szabad szoftver modellnek, de ez hozza létre azt az egyensúlyt, ami miatt a szabad szoftver projektek nagy százaléka olyan gyorsan fejlődik. (Természetesen ezt az előnyt kihasználják a COTS-fejlesztők is, amikor olyan nyílt forráskódú részleteket használnak fel, amelyeknek a licence megengedi a zárt forrásúvá tételt.)

6.3.4. A szabad szoftver projektek közti verseny

Világunkban, ahol az erőforrások végesek, a projektek közti verseny természetes jelenség. A klasszikus szabadszoftver-fejlesztés esetében nem elsősorban pénzért hanem piaci elismerésért, tehetséges programozókért folyik a harc. Ez a verseny biztosítja a termék jó minőségét, mivel ha a felhasználói kör csökken, akkor a hibajavítók köre, a visszacsatolás ereje is csökken. Az általános érdeklődés csökkenésével a programozók is inkább átpártolnak más projektekbe, és ez, feltéve, ha nem veszi át egy másik csoport a fejlesztést, megpecsételheti a projekt jövőjét. Az itt kifejtett versenyhelyzet és együttműködés természetes folyamata biztosítja a szabad szoftver projekteknek a magas hatékonyságot. Mivel mindenki nyílt kártyákkal játszik, ezért szükségtelen mindent többször feltalálni, megtervezni és kivitelezni.

7. Finanszírozási modellek

A szabad szoftverek finanszírozási modellje gyökeresen eltér a COTS-szoftverekétől. A COTS-szoftverek nagyrészt az eladott példányok által keletkezett bevételéből szerzik meg pénzügyi forrásaik túlnyomó részét. Ez a szabad szoftverek esetében nem igaz, mert nincs eladott darabszám utáni bevétel, ezért a szabad szoftver projektekben más finanszírozási formákat kell, hogy kidolgozzanak. Mint már tudjuk, a klasszikus szabad szoftver projektek kiadásai nem összehasonlíthatók a COTS-cégek kiadásaival (nincs menedzsment-fizetés, a fix kiadások eleinte nem mérvadóak, nem kell stábot fizetni stb.), ettől függetlenül ha a projekt elér egy bizonyos szintet, a kiadások növekedése elkerülhetetlen.

Négy alapvető finanszírozási elvet lehet megkülönböztetni a szabad szoftverek esetében. Ezek a nem finanszírozott, a külsőleg és belsőleg finanszírozott, valamint a belső felhasználásra készített szoftvermodellek.

A fejezetben kifejtett elvek természetesen elméletiek, így a gyakorlatban ritkán jelennek meg önmagukban.

7.1. Nem finanszírozott szabad szoftver projektek

Mivel a szabad szoftver projektek kiadásai minimálisak, egy projektet problémamentesen elindíthat egy ember vagy egy kis csoport is a számukra már rendelkezésre álló infrastruktúrával, elvégre semmi másra nincs szükségük, mint lelkes, számítógéppel és internet-kapcsolattal rendelkező programozókra és érdeklődő felhasználókra. A programozók természetesen önként, ellenszolgáltatást nem várva végzik ilyenkor a munkát. Ez a legjellemzőbb formája a szabad szoftver projekteknek. Elsősorban kisméretű kezdeményezéseknél működnek, de van példa arra, hogy hatalmas projektek is ezt az elvet követik. Ilyen például a GNU/Linux kernel, a Debian disztribúció és a három szabad BSD rendszer is.

7.2. Külsőleg finanszírozott szabad szoftver projektek

Egy projekt természetesen formálisan is megjelenhet a gazdasági térben. Ez általában cég vagy non-profit szervezet formájában valósul meg. Ekkor már jogilag sincs semmi akadálya annak, hogy a projekt külső finanszírozást kapjon, amely származhat állami forrásokból és cégektől is.

7.2.1. Állami finanszírozás

Kutatási, illetve nagy informatikai projektek támogatásában az állam szerepvállalása előnyös és gyakori. Ennek legelterjedtebb módja az egyetemek és egyéb kutatási intézetek költségvetéséből történő finanszírozása. Második helyen az általános informatikai fejlesztésekre fordítható keretösszegekből, pályázati úton nyerhető és szabad szoftver fejlesztésére fordítható támogatás áll. Természetesen az is járható út, hogy az állam közvetlenül támogat olyan projektet, melynek

megvalósulásához komoly (pl. gazdasági) érdeke fűződik, ennek tipikus esete, amikor finanszírozásával hozzájárul egy szabad szoftver honosításához.

7.2.2. Cégektől származó anyagi támogatás

Közvetett támogatásnak nevezhetjük azt, amikor egy cégnek szüksége van egy bizonyos szabad szoftver továbbfejlesztésére, hogy azt saját céljaira jobban fel tudja használni.

Ebben az esetben az adott cég a szükséges fejlesztéseket vagy elvégezteti házon belül, vagy kiadja alvállalkozóknak (és fizet érte). A végén az elkészült fejlesztések visszakerülnek szabad terjesztésbe, így a projekt implementálni tudja azokat. Ezt tette például a Corel Corporation, amikor a WordPerfect és CorelDraw programokat alakította Linux-komforttá. A fejlesztéshez szükségük volt egy olyan eszközre, amellyel Microsoft Windows-alapú szoftvereket lehet Linux alatt futtatni. A lehetséges megoldások közül a Wine-t választották, de nem voltak megelégedve a nyílt forráskódú eszköz funkcionalitásával, így házon belül allokáltak szakembereket ennek továbbfejlesztésére. Amint elérték a nekik megfelelő szintet, eredményeiket megosztották a Wine projekt programozóival, akik ezeket sikeresen integrálták saját alkalmazásukba.

A cégektől származó közvetett támogatás (azaz fizetett programozók részvétele) egy szabad szoftver fejlesztési projektjében gyors előrelépést hozhat a minőségbiztosítás, a dokumentációk és a finanszírozó preferenciájába tartozó funkcionalitások megvalósításában.

Közvetlen támogatásnak nevezhetjük, amikor egy cégnek érdekében áll, egy projekt fejlődése, és ezért anyagilag segíti azt vagy annak kulcsfontosságú szakembereit. Ezt tette például az O'Reilly & Associates könyvkiadó, amikor támogatta a Perl programnyelv fejlesztőit. A kiadónak érdeke volt a programnyelv fejlődése, hiszen sok eddigi és majdan e témában kiadandó könyvét csak így tudta értékesíteni. A VA hardvergyártó cég a Linux-rendszermag (kernel) projektet támogatta annak érdekében, hogy a kernel teljesen kompatibilis legyen az általuk gyártott eszközökkel.

7.3. Belső vagy bevételalapú finanszírozás

A bevételalapú finanszírozás alapja, hogy egy informatikusokból (szakértőkből) álló szervezet értékesít egy szabad szoftverrel kapcsolatos szolgáltatást és/vagy meghatározott irányba fejleszti tovább a forráskódot. A szolgáltatások a szervezetben felhalmozott szaktudásra vagy a már meglévő forráskódra épülnek.

7.3.1. Szaktudásra építő szervezet

Az értékesített termék tulajdonképpen az az adott szabad szoftverről felhalmozott ismeretanyag, amivel a szervezet rendelkezik. Ennek leggyakoribb megnyilvánulásai természetesen a rendszerintegrációs és támogatási szolgáltatások.

Ezt a stratégiát bármely – szakértőket tömörítő – szervezet alkalmazhatja, mivel a forráskód nyíltsága teljes rálátást biztosít a szoftver működésére. Kereskedelmi (zárt forrású) termék esetében a szolgáltatásokat nyújtó cég csak annyi információval rendelkezhet, amennyit tapasztalati úton és a gyártó által adott tanfolyamokon megszerzett, amely tudás a forráskód ismeretéhez képest lényegesen kevesebb.

Mivel a forráskódot mindenki megismerheti, a nyújtott szolgáltatások minőségét és árait az egészséges piaci verseny tartja reális egyensúlyban.

7.3.2. Kódra építő szervezet

Az a szervezet, amely saját fejlesztésű szabad szoftverre alapozza szolgáltatásait előnyre tesz szert konkurensaival szemben, hiszen elsőként jelenik meg a piacon, míg a többieknek meg kell ismerniük a termék működését.

A piaci előny további növelésének eszköze lehet az is, ha a fejlesztő cég a termékkel kapcsolatban ún. duál licenclést alkalmaz, azaz egyazon termékét zárt licenccel és szabad szoftverként is elérhetővé teszi. Ez a módszer a klasszikus szabad szoftver szemlélettel ellentétes ugyan, mégis nagy népszerűségnek örvend napjainkban. Az így eljáró szervezetek általában az alábbi két megoldást választják:

1. A fejlesztő által kiadott zárt forrású termék funkcionálisan bővebb, mint a nyílt forrású. Természetesen a szabad szoftver közösség a nyílt forráskódú verzió alapján maga is kifejlesztheti a hiányolt funkciókat, de ez időbe kerül.
2. A szoftverfejlesztő a legújabb verziót minden esetben zárt forrású termékként teszi közzé, de azokat a régebbi verziókat, amelyeket már nem támogatna, kiadja nyílt forrású változatban.

Fontos, hogy az ilyen jellegű duál licenclésekre kizárólag a szoftver szerzői jogainak tulajdonosa (az eredeti megalkotó) jogosult, a többieket köti a szabad szoftver felhasználói licencszerződése.

7.3.3. Egyedi licencek

A fent említett duál licenclés egy másik példája lehet az alábbi.

Tegyük fel, egy COTS-gyártó zárt forrású termékbe szeretne beépíteni egy olyan szabad szoftvert, amelynek licence tiltja ezt. A COTS gyártója megkeresi az eredeti fejlesztőt (a szerzői jogok tulajdonosát), és megfelelő juttatások ellenében megvásárolja tőle a szoftvert – bezárható licenccel.

7.3.4. Márkateremtés

Nem ritka, hogy egy cég szabad szoftvereket gyűjt össze, azokat további funkciókkal kiegészíti, és így új terméket hoz létre. Erre tipikus példa a kereskedelmi Linux disztribúciók esete, melyek nem állnak másból, mint nyílt forrású programokból. Az értéknövelt tulajdonságuk tulajdonképpen a csomagolás, a jobb dokumentáció és a könnyen kezelhető telepítő és felhasználói felület. A stratégia sikerességét bizonyítja, hogy az USA-ban a Linux és a Red Hat Linux

disztribúció fogalma szinte teljesen összenőtt. A márkateremtő cégek természetesen szintén a támogatásból profitálnak.

7.4. Belső felhasználásra készített szoftver

Ebbe a kategóriákba olyan szoftverek tartoznak, amelyeket egy cégen belül saját használatra fejlesztettek ki, mivel a COTS-termékek nem feleltek meg az igényeknek, vagy drágábbnak bizonyultak a saját megvalósításnál. A cég szakemberei fejlesztés közben vagy annak végeztével megnyitották a szoftver kódját, hogy élvezzék a szabad szoftver modell előnyeit: külső programozási és hibajavítási segítségre tehetnek szert. Nem kizárt az sem, hogy egy ilyen szoftver nagy népszerűsége tesz szert és széles körben használatos lesz, mint a Cisco nyomtatási rendszere, a CEPS, amely a kód megnyitása óta sikeres szabad szoftver projektté vált.

8. COTS versus szabad szoftver

8.1. TCO (Total Cost of Ownership)

Ha a gazdasági modell eddig nem nyújtott volna kielégítő választ arra, miért is jó az egy szervezet számára, ha szabad szoftvereket használ, más oldalról is megközelítjük a problémát. TCO (Total Cost of Ownership), vagyis a bekerülési és üzemeltetési költség egy nemzetközileg elfogadott gazdasági mérce a szoftverek vagy esetleg szoftver-hardver termékekkel járó effektív kiadások mérésére. Fontos, hogy a TCO nem méri az üzemeltetésből nyert előnyöket, csak a kiadásokat (általánosságban nem is lehetne az előnyöket vizsgálni, hiszen azok nagyrészt a kivitelezés minőségétől függenek). E fejezetben kifejtjük a szabad szoftver bekerülési és üzemeltetési költségeinek struktúráját.

A TCO magában foglalja a beszerzési, kivitelezési, beüzemelési és működtetési költségeket a használat időtartamára leosztva. Ezt a folyamatot a következő stádiumokra bontottuk: információszerzés, beszerzés vagy elkészítés, kivitelezés, képzés, karbantartás, megszüntetés és új rendszerre való áttérés.

Információszerzés

Ez a beszerzési folyamat legkorábbi szakasza, itt kell eldöntenünk, melyik az ideális a lehetséges megoldások közül. Az információszerzés folyamata két nagyobb lépésre bontható; egyrészt ki kell derítenünk, hogy problémánk megoldására létezik-e termék, és ha igen, akkor a létezőkről a legtöbb információt kell beszerezni a döntés megkönnyítéséhez. A megoldások megtalálása (*létezik-e*) kereskedelmi termék esetén könnyebb, hiszen azokkal a szakújságok többet foglalkoznak, nem beszélve a reklámokról és a szóróanyagokról. Ha az on-line információhalmazt is figyelembe vesszük (honlapok, tematikus keresők), ez a különbség jelentősen csökken.

Ezzel szemben a szabad szoftverekről szerzett információ hitelesebb, mint egy COTS-termékről annak gyártójától vagy forgalmazójától szerzett tudás. Egy újságcikk esetében nehéz megállapítani, hogy azt, a szóban forgó termék fejlesztői, mint bújtatott reklámot tették-e közzé, vagy tényleg hiteles értékelést látunk-e. A szabadszoftver-felhasználókkal könnyen lehet közvetlen kapcsolatot teremteni, amely általában pontos, reális válaszokhoz vezet, akár több független forrásból is. További előny, hogy a szabad szoftvert saját magunk is letesztelhetjük minden külön költség nélkül.

Beszerzés vagy elkészítés

Ha már létezik az előző stádiumban felállított kritériumainknak megfelelő eszköz, akkor beszerzése a forráskód letöltésére korlátozódik. Amennyiben nincs ilyen termék, nos, ez esetben a programozóké a terep. A szabad szoftverek beszerzése esetében mindenképpen nagy anyagi előnyt jelent, hogy nem jelentkezik felhasználónkénti vagy példányonkénti költség.

Amennyiben még nem létezik kész megoldás a problémánkra és szoftvert kell fejlesztenünk, akkor érdemes körülnézni, hogy létezik-e olyan nyílt forrású eszköz, amely megfelelő alapot biztosít a munkánkhoz. A szabad szoftver fejlesztési modell ezentúl a külső programozó és hibakereső erőforrások könnyű bevonhatóságával is kecsegtet.

Kivitelezés

A kivitelezést nehéz általánosságban vizsgálni, mivel szoftvert típustól is függ. A szabad szoftverek COTS termékekhez képest nehéz telepíthetősége az installációs technológiák fejlődésével, és a terjesztések (disztribúciók) előretörésével rohamosan csökken.

Ezzel szemben a szabad szoftverek sokkal jobban testeszedhetők, ideértve azt is, hogy ha egy adott rendszeren forráskódból fordítjuk, sokkal optimalizáltabb működést kapunk.

Természetesen bonyolult és komoly szakértelmet igénylő rendszerek telepítése esetén az igénybe vett külső szakemberek költségében nem jelentkezik eltérés, akár szabad, akár COTS-termékről van szó.

Képzés

Mint már említettük, a szabad szoftverek egyik hátránya, hogy a felhasználói dokumentációk lassabban készülnek el. Ezzel szemben egy sikeres projekt esetén a közösség által írt HOGYANOK, GYIK-ek, levelezőlista-archívumok messze több információt szolgáltatnak, mint egy átlagos kereskedelmi termék dokumentációja, továbbá mint majd láthatjuk, az ilyen sikeres projektek köré gyorsan szerveződnek kommersz szolgáltatásokat nyújtó cégek, amelyek természetesen oktatással is foglalkoznak.

A házon belüli fejlesztések esetében lehetséges, hogy ha más körökben is elterjed a szoftver, az áttételesen jobb dokumentációhoz vezet.

Karbantartás

Ez a TCO legmérvadóbb része, és ez az a szekció, amelyben a szabad szoftverek a leginkább nyeresre állnak vetélytársaikkal szemben. Mivel a forráskód rendelkezésre áll és annak felhasználása nem ütközik jogi problémákba, így a hibajavítások és fejlesztések azonnal implementálhatók. Ennek zárt szabványok esetében mindenképp a szoftverfejlesztőn keresztül kell történnie, ami az esetek túlnyomó részében hosszú folyamat.

Megszüntetés és új rendszerre való áttérés

Egy informatikai rendszer életciklusának utolsó fázisa, amikor leváltásra kerül. A TCO-elemzések gyakran elfelejtkeznek erről a pontról, pedig kulcsfontosságú az adódó költségek miatt. Ha konkrétan a rendszer megszüntetését vizsgáljuk, annak költsége nyilván nem jelentős. Ezzel szemben fontos a régi rendszer által felhalmozott és az új rendszerben is szükséges adatok átmentése, konverziója. A szabad szoftverek azzal az óriási előnnyel rendelkeznek e téren, hogy a nyílt szabványokra épülő adatformátumokhoz fejleszthetünk bármilyen konverter eszközt, míg zárt formátum esetén ez problémát jelenthet. A zárt adatformátumot

vissza kell fejteni, ami hibák nélkül szinte lehetetlen, továbbá nagyon pénz- és időigényes feladat.

8.2. IT-biztonság

A szabad szoftver közösség és a COTS-termékeket gyártó kör között állandó vita, hogy a forráskód nyíltsága jót tesz-e vagy árt az IT-biztonságnak. Jelen fejezetnek nem célja ezt eldönteni, csak felsorakoztatni mindkét fél érveit, illetve objektív, külső információ-forrásokat bevonni a vitába.

A Microsoft mint legnagyobb operációsrendszer-gyártó a szabad szoftver modell legnagyobb kritikusa, több médiumon keresztül a következő érveket sorakoztatta fel a szabad szoftverek ellen és a saját módszere mellett:

- ♦ Mivel a biztonságért felelős szakembereknek teljes munkaidőben és fizetésért kell tenniük a dolgukat és kizárólag a biztonsággal foglalkoznak, a szabad szoftvereket használó adminisztrátorok hátrányban vannak. Az ő feladatuk nehezebb, mert minimum figyelniük kell a szóban forgó szoftverekkel foglalkozó fórumokat és levelezőlistákat, de nem árt az sem, ha a programkódban kutatva hibákat keresnek, hiszen ha ők nem teszik ezt meg, akkor más sem fogja.
- ♦ A nyílt forráskód önmagában is egy biztonsági rés – figyelmeztet a Microsoft –, mivel ez megadja a lehetőséget különböző rosszindulatú embereknek, hogy kihasználható hibákat találjanak, amelyeket aztán nem hoznak nyilvánosságra.
- ♦ A szabadszoftver-fejlesztők álláspontja szerint a "több szem többet lát" elv alapján a kódban minden hibát meg tudnak találni, ha elegendően keresik ezeket. A tapasztalat azt mutatja, hogy nyílt forrású szoftverben biztonsági hiányosság csak addig maradhat rejtve, amíg egyszer ki nem használják. Ha ez megtörtént, a javítás a lehető leggyorsabban elkészül.
- ♦ Szerintük a zárt szoftverek esetében sincs ez másképp, csak ha ott a hibát nem egy a gyártó cégen belüli személy fedezi fel, akkor az a szabad szoftverek fejlesztői légkörének hiányában gyakrabban vezet visszaéléshez.

Egyes cégek álláspontja szerint a szabad szoftveres fejlesztés megadja a lehetőséget a „hátsó kapuk” beépítésére, amivel később az egykori fejlesztő bármikor visszaélhet. Ez ugyanígy igaz a zárt forráskódú termékekre is, csak ott még az esély sincs meg, hogy ezeket megtaláljuk (jó példa erre a piacvezető táblázatkezelő szoftverben található repülőgép-szimulátor, amely ártatlan ugyan, de feleslegesen növeli a szoftvert és lassítja működését).

Arra is volt már példa, amikor zárt forrású terméket (Interbase adatbázis-kezelő) tettek nyílt forrásúvá, és szoftver felépítése után érdeklődők rögtön kiszúrtak egy komoly hátsó kaput, amit az eredeti fejlesztők „felejtettek benne” kényelmi céllal.

A biztonság objektív összehasonlításához kevés tudományos vagy akár tényeken alapuló információ áll rendelkezésünkre, továbbá a meglévő statisztikákkal is vigyázni kell, mert könnyen félreértelmezhetőek.

A J.S. Wurzler cég, amely talán elsőként kínál biztosítást crackerek ellen, 25%-os felárat számol fel, ha a biztosítandó rendszer Windows NT-n fut. Az árkézésük egyéni elbírálás alapján történik, és állításuk szerint semmi különbséget sem tesznek szabad szoftverek és COTS-szoftverek között.

A BugTraq egy, a sebezhetőségeket nyilvántartó levelezési lista és adatbázis naprakész adatokat közöl felhasználói szoftverekről és operációs rendszerekről. Statisztikáikból nehéz következtetni bármire is, mivel abban nem jegyzik a biztonsági rés komolyságát (növekvő sorrendben pl. DoS, lokális jogosultság expanziója, távoli jogosultság expanziója) és a hibajavítások megjelenésének gyorsaságát.

Egy ehhez hasonló brit tanulmány az ő 2000. évi adatait idézi, mely szerint a Red Hat Linuxban 85 sebezhetőséget találtak, amíg a Windows NT-ben 144-et. Természetesen az adatok nem teljesen összehasonlíthatóak, mert a szabad szoftver modell több lehetőséget ad az ilyen fajta sebezhetőségek felismerésére, illetve a Red Hat kategóriákkal nagyobb mennyiségű felhasználói programot tartalmaz, mint a Windows NT.

Ugyanezen a brit tanulmány szót ejt a vírusokról is. A 2000-es évben több mint 60.000 Windows, 40 Macintosh, 5 COTS-Unix és 40 Linux-alapú vírusról számoltak be. A Windowst érintő sebezhetőségi statisztikák kiugróan magas értékére több magyarázat is adható, de az egyértelmű, hogy a Windows a legvonzóbb célpont a víruskészítők számára. Ennek oka a Windows széles körű használata, a rendszerkezelés szabad hozzáférése a felhasználók és általuk futtatott programok számára, és a rendszerbe épített olyan automatizált folyamatok (makrók, szkriptek), amelyek vírushordozóként üzemelhetnek.

Crackerek további megnyilvánulásai közé tartozik bizonyos honlapok feltörése és azok nyitóoldalának megváltoztatása. A „betörési” eseményekből készült statisztikák azt mutatják, hogy az ilyen sikeres támadások célpontjai az estek 60%-ban Windows Nt-n futó Microsoft IIS szerverek, amelyek egyébként a webkiszolgálóknak csak 25%-át teszik ki. Ez az aránytalanul magas támadási szám viszont félrevezető, hiszen figyelembe kell vennünk, hogy a crackereknek kihívást jelentő nagyvállalati és kormányzati honlapok nagy része IIS-t futtat.

A biztonsági statisztikák alapján tehát még nem lefutott a mérkőzés. Mindkét típusú rendszerre igaz, hogy egy hozzáértő rendszergazda gondosan frissítő kezében biztonságossá tehető, azonban véleményünk szerint ez IIS esetén nagyságrendekkel több munkát jelent.

9. Versenyhelyzet tanulmányozása és javítása

9.1. Függőségi helyzet vizsgálata

Jelen fejezet célja a szoftverfüggőség kialakulásának és következményeinek elemzése.

A függőség akkor alakulhat ki, ha egy szervezet oly mértékben kapcsolja hozzá működését egy bizonyos szoftvertermékhez, hogy nélküle nem is tudna létezni. Mivel az informatikai rendszereknek is van életciklusa, és ennek velejárója, hogy egy szoftver vagy hardver „előregszik”, ezért már a tervezésnél figyelembe kell vennie a más rendszerre való átállás lehetőségeit.

Vegyünk egy példát. Egy kormányzati szervnek van egy ügyviteliadatbázis-rendszere, amely nyilvántartja az elmúlt 10 év tranzakcióit. Ez a rendszer nagyon lassú és nagyon régi, nehéz kezelni és már nem is működik a munkaállomásnak újonnan vásárolt számítógépeken. Ez egyértelműen jelzi, az idő előrehaladtával az üzemeltetési költségek hatványozottan emelkednének, így itt az ideje a rendszer leváltásának.

Az igény nyilván az, hogy egy új, modern adatbázist hozzunk létre, amely gyors és az új munkaállomásokkal is együttműködik. Ez mind szép és jó, de általában szükséges a régi adatok új adatbázisba való konvertálása. Ha az adatok struktúrája nem nyílt szabványoknak megfelelő, a kormányzati szerv már is beleütközött a függőség problémájába. A probléma megoldására, ebben a helyzetben véges számú lehetősége van:

- Szakemberekkel visszafejteni az adatformátumot. Ez költséges és hosszú folyamat.
- Esetleg beviheti újra az adatokat. Ez általában az adatok mennyisége miatt nem megvalósítható.
- Ha létezik még az eredeti terméket előállító cég, akkor őket lehet megkeresni, de ezzel a piaci verseny elmarad.
- Legköltségesebb esetben párhuzamosan működteti mindkét rendszert.

A fentiek egyike sem igazán ideális megoldás.

A kormányzatnak különösen nagy figyelmet kell fordítania arra, hogy az intézmények azonos funkciójú rendszerei közti kommunikáció lehetőségét meg tudja teremteni. Ha a kormányzat által használt adatformátumok, kommunikációs protokollok stb. nincsenek jól definiálva, akkor a fent említettek megvalósítása kudarcba fulladhat. A kormányzat erre vonatkozó előrelátását igazolja, hogy a tanulmány keletkezésének időpontjában több olyan projekt is folyik, amely az informatikai rendszerek egységesítését szolgálja. Ennek két (talán legnagyobb) példája a kormányzati címtár és az egységes iratkezelési (iktatási) rendszerek kialakítása.

9.2. Monogámia

Természetesen a szoftverfüggőségének lehetőségét felfedezték a kereskedelmi termékek gyártói is. Felismervén az integrációs szolgáltatások piacának nagyságát, a függőségi helyzet kiaknázásával rövid távon tudnak profitálni, míg az „upgrade-ek” megvásárlása hosszú távon hoz hasznot. Ha a kormányzat mint legnagyobb informatikai felhasználó konkrét termékek és nem szabványok használatát ösztönzi, akkor nagyban hozzájárul a szoftver-monogámia kialakulásához.

9.3. Kompatibilitás és összeférhetőség

Az informatika talán legerősebb trendje a teljes körű integrációt célozza meg. Ezt próbálják meg kisajátítani maguknak a szoftver-monogámiaira törekvő cégek. Ezzel szemben az informatikában és elektronikában számos példa mutatja, hogy a nyílt szabványok jobban elterjedtek, mert mindenki által használhatóak, így implementálhatóak voltak. (Vegyük például az egyébként rosszabb minőségű VHS videoszabványt, amely nyíltsága folytán győzte le a Betát. A Beta szabványt feltalálója szabadalmaztatta és egyedül próbálta meg terjeszteni, így az ára a versenyhelyzet és a tömeges alkatrészgyártás hiánya miatt magas maradt. Ilyen körülmények között zárt szabvánnyal csak egy monopóliumnak lett volna esélye. Ezzel analóg, hogy a kommunikációs protokollok esetében az ISO által támogatott OSI 7-layer modell megvalósítása (MAP és TOP) megbukott a szabad TCP/IP-vel szemben.

A szabad formátumok és szabványok előtérbe helyezése a kormányzaton belüli és az állampolgárokkal való kommunikációban hosszú távon meg fogja hozni gyümölcsét, hiszen a fejlesztési költségek ezáltal lényegesen csökkennek, valamint a szoftver-monogámia kialakulására jóval kisebb az esély.

9.4. Multik kontra helyi kis- és középvállalkozók

A szabad szoftver modell sajátossága, hogy a helyi kis- és középvállalkozásokat támogatja a multinacionális cégekkel szemben.

Ennek oka leginkább az, hogy mint már elemeztük, a szabad szoftver támogatása kizárólag a piac valós igényei alapján jelenik meg. Ennek felmerülése esetén viszont a kisvállalkozások kisebb reakcióideje előnyt jelent számukra. Ezen felül a multinacionális cégek esetén ritkább (bár az informatikai ipar trendjei ezt szorgalmazzák), hogy „katedrális jellegű” termékeikből kiindulva olyan mértékben testre szabható megoldásokat kínáljanak, mint egy szabadszoftver-alapokon építkező vállalkozás.

9.5. A COTS befolyásolása

A szoftvergyártó multinacionális vállalatok konkurenciával szembeni legnagyobb

fegyvere a zárt szabványok használata. Ebben az esetben a gyártó az általa fejlesztett szoftverekbe olyan szabványokat, eljárásokat épít be, amelyeket saját maga dolgozott ki, és azok leírását nem osztja meg mással. Ezzel az eljárással nagy mértékben akadályozza, hogy más gyártók olyan eszközöket készítsenek, amelyek kompatibilisek az ő termékével, ezzel is erősítve egy területre vonatkozó monopóliumát. Például egy igen nagy piaci részesedéssel rendelkező irodai programcsomag által generált állomány-formátum felépítését nem teszik közzé, és ezért kizárólag az adott termékkel lehet helyesen olvasni és létrehozni ilyen állományt. A formátumok közti átjárhatóság elérésére már sok kísérlet történt, ezidáig sikertelenül. A zárt szabványokkal szemben kizárólag az úgynevezett visszafejtés – reverse engineering – vethető be, amely a világ számos országában illegális, de hatásfoka meglehetősen alacsony.

A COTS-termékeket gyártó cégek befolyásolásának legjobb eszköze, ha a kormányzat a nyílt szabványok használata mellett teszi le a voksát. Az informatika érintett területeiről készülő részletes tanulmányok segítségével könnyen kiválasztható a támogatandó nyílt szabvány.

A befolyásolás legradikálisabb eszköze, ha a kormányzati beszállítókat a nyílt szabványok használatára kötelezik (vagy legalább az azokkal való kompatibilitásra). Érdemes olyan fejlesztőkkel kezdeni, akik új szoftvert készítenek. Persze egy multinacionális cég esetén ez már rövid és nem egyszerű folyamat, de csak így lehet megakadályozni, hogy egyes gyártók egy teljes országot érintő monopóliumot tudjanak kialakítani.

A monopólium ellen való küzdelem másik fegyvere a szabad szoftverek terjesztése a felhasználói oldalon. Ennek legjobb módszere, ha kötelezzük a hardverforgalmazókat, hogy kizárólag operációs rendszerrel együtt értékesítsék a számítógépeket. Ennek természetesen alapvető feltétele egy magyarul „beszélő”, szabadszoftver-jellegű operációs rendszer megléte. A SuSE, a Mandrake és a Red Hat már majdnem teljesen honosított, de készül az UHU Magyar Linux terjesztés is.

A szabad operációs rendszerrel együtt adott hardver nagy mértékben visszaszorítaná a szoftvertolvajlás mértékét is Magyarországon.

Ez természetesen nem jelenti azt, hogy a szoftverek másolása teljesen meg fog szűnni. A rendszer, amellyel a számítógépeket szállítják, letörölhető, és helyére telepíthető bármi, de véleményünk szerint ha valaki a boltból működőképes rendszerrel kapja komputerét, az nagy százalékban megtartja, amit rajta talált. Az előbbiekben kifejtett gyakorlat a legális programhasználat terjesztésén felül az árakat két irányból is csökkenti. Egyrészt a nagyobb versenyhelyzet kialakításával, másrészt a hardverrel szállított operációs rendszer fogalmával (OEM gyakorlat).

10. Jogi körülmények tanulmányozása

E fejezet hivatott felhívni a figyelmet a szabad szoftverek használata esetében felmerülő alapvető jogi problémákra. A szoftverek esetében komoly figyelmet kell fordítanunk a szellemi tulajdon, ezen belül a szerzői jogok védelmére. Ez a szabad szoftverek esetén sincs másképp, mivel azok létezési körülményei gyakran nincsenek szinkronban a jogi normákkal.

10.1. Szerzői jog és licencek

A szerzői jog a szoftverek legáltalánosabb jogvédelmi formája. A szerzői jog alapvető hozzáállása az, hogy a szoftvert felhasználni, terjeszteni csak a szerző által kinyilvánított engedély birtokában és annak feltételeivel összhangban lehet. Ezeket a feltételeket definiálják a végfelhasználói szerződések, azaz licencek. A kereskedelmi licencek jellemzően az alábbiakat szabályozzák:

- felhasználás feltételei (pl. egy telepítés, maximális kliensszám, 2 évre korlátozott engedély stb.);
- terjesztés tiltása (másolás, továbbértékesítés);
- visszafejtés tiltása, ... stb.

A szabad szoftverek és a kereskedelmi termékek licencei merőben eltérőek, de a felhasználói jogok szabályozása a lényege az előbbinek is. A legfontosabb különbség, hogy egy szabadszoftver-licenc olyan feltételeket állít, amelyekkel a szoftverrel kapcsolatos szabadságjogok megőrzése a cél. Ezek a felhasználás, a terjesztés, a forráskód-hozzáférés és -módosítás joga köré csoportosulnak. Ide tartoznak azok a kötelezettségek is, melyek szerint a továbbterjesztett (akár módosított) változatnak is kell tartalmaznia az eredeti szerzőre vonatkozó copyright- (szabad szoftver esetén copyleft) információkat.

Több szervezet is létezik, amely definiálta a szabad szoftver fogalom által megkövetelt minimális jogokat. Ilyen például a Debian Free Software Guidelines (DFSG) vagy az ennek keretein belül működő Open Source Initiative (OSI), illetve a GNU projekt. A legismertebb licenceket ld. az 1. sz. mellékletben.

Természetesen a fenti definícióknak nem megfelelő licencek között érdemes rangsort állítanunk a „szabad” és „kevésbé szabad” besorolásnak megfelelően.

A kereskedelmi termékekhez legközelebb álló licencek a „freeware” kategóriába tartozók, melyek felhasználása ingyenes, de a forráskód nem elérhető, ezáltal a módosítás sem lehetséges (pl. Acrobat Reader).

A második kategóriát alkotják a már nyílt forráskódú, de más jogaiban korlátozott eszközök (pl. Pine levelező, amelynek módosítása saját használatra lehetséges, de a módosítottat továbbterjeszteni nem lehet).

A licencek harmadik és legérdekesebb típusa az, amely a felhasználónak tulajdonképpen a legnagyobb szabadságot adja, mivel megengedi a forráskód módosításának vagy egészének zárttá tételét. Ez viszont nem felel meg a fenti

definícióknak, mivel nem biztosítja tovább az alapvető jogok szabadságának folytonosságát. Utóbbiakat építik be termékeikbe előszeretettel a zárt forrású szoftvereket fejlesztő cégek.

A licenceknek a már említettekén kívül még természetesen számos típusa létezik, amelyek a jogosultságok szinte összes lehetséges változatát felölelik.

A licenctípusok közül természetesen a szerző saját ízlése szerint választhat, és felhasználhat az említett szervezetek által definiáltakat (pl. GPL), vagy írhat sajátot is (pl. Sun Industry Standards Source License (SISSL) – az OpenOffice.org licence). Természetesen ez a választás nem könnyű, de ha a magyar kormány leteszi voksát a szabad szoftverek mellett, akkor irányelvben, megfelelő vizsgálat után, érdemes választania egy licenctípust, amelyet a kormányzati fejlesztőknek követniük kell majd. Az angol kormánynak hasonló tanulmányt készítő csoport a Mozilla böngésző licencét, a Mozilla Public License-t ajánlotta, mivel szerintük ez a legkomfortosabb az üzleti célú felhasználással.

Magyarországon a helyzet nem ilyen egyszerű, mivel az elmúlt pár évben történt ugyan némi előrelépés a szerzői jogok nyugati országokkal való harmonizációjával, de precedens hiányában a meglévő szabályok értelmezése zavaros, továbbá arról sincs tudomásunk, hogy létezne a jogi helyzettel foglalkozó tanulmány. Ilyen tanulmányokra azért lenne szükség, mert a jogi alapértelmezések gyakran oly mértékben térnek el a szabad szoftver filozófiájától, hogy ez a szabad szoftverek létjogosultságát is veszélybe sodorhatja. Bár a „licenconosítás” (a licencek helyi joghoz való igazítása) nem ismeretlen fogalom, és ezzel majdnem minden probléma kiküszöbölhető – ha komoly összeférhetlenségekre derül fény, akkor akár törvénymódosítás is indokolt lehet. Ezekből következően felhasználandó licenctípust felelősen ajánlani kizárólag a jogi helyzet tisztázása után annak ismeretében érdemes.

A licencek közti különbségeket tehát általában az képezi, hogy a szerző milyen fontosságot tulajdonít:

- a szoftverszabadság hosszú távú biztosításának;
- a morális jogok védelmének (pl. a szerző(k) kötelező feltüntetése);
- a tulajdonjogok védelmének (esetünkben tulajdonjogoknak számítanak a szerző(k) által fenntartott általános jogokon felüli privilégiumok);
- a más (akár nem szabad) licencekkel való összeférhetőségnek vagy összeférhetlenségnek.

A szerzői joggal kapcsolatban érdekes problémát vet fel a kompatibilitás megteremtésének igénye. Volt már rá precedens, hogy bírói döntés kötelezett egy szoftvergyártót arra, hogy mély belátást biztosítson a többi fejlesztőnek a szoftver működésébe. Ezzel a felek jogvédett információkhoz jutottak, de ezek az információk lehet, hogy a helyi jogszabályok alapján nem is estek a szerzői jog védelme alá. Egyes országok pontosan meghatározzák, hogy a szoftverek esetében

mi kerül a szerzői jog védelme alá és mi nem. Egy ilyen eset nemzetközi jogi problémákat is felvet.

10.2. Szabadalmak problémái

A szerzői jog nem az egyetlen mechanizmus, amely a szoftvereket mint alkotásokat védheti. Az utóbbi időben, elsősorban az Egyesült Államokban, a szoftver és a szoftveres megoldások szabadalmi kezdnek elterjedni. De miért is okoz ez problémát?

Kis méretű algoritmusok szabadalmaztatása esetén azok újra „feltalálása” – de inkább megalkotása – szinte elkerülhetetlen. Sajnos a gyakorlat azt mutatja, hogy az adott szervek gyakran jegyeznek be triviális dolgokat. Az Egyesült Államokban a Novellnek a Y2K probléma sokak szerint legkézenfekvőbb megoldását sikerült szabadalmaztatnia, egy olyan eljárást, amit valószínűleg több ezer programozó feltalált és alkalmazott, bár azóta a bíróság felülvizsgálta döntését e szabadalom ügyében, azonban a jogi procedúra nem kevés állami és magánpénzbe került.

A szoftver- és algoritmus-szabadalom monopóliumok kialakulásához és megerősödéséhez vezethet, így a szabad szoftver modellel nem összeegyeztethető. Mivel hazánkban ilyen jellegű szabadalomosztály nincs, ebből a szempontból könnyebb helyzetben lennénk, mint az USA. Sajnos az utóbbi időben egyéb szabadalmi osztályokba tartozó, de hasonló területet érintő bejegyzések jelentek meg. Ilyen például az SMS direktmarketing, aminek használata jelenleg jogdíjköteles.

(ref. W. Neville Holmes. The Evitability of Software Patents. *IEEE Computer* 33(3):30-34, 2000. március)

11. Megvalósítás

A tanulmány további részében a kormányzatban használt jellemző informatikai alkalmazások nyílt forrású alternatíváit mutatjuk be.

A vizsgált alkalmazásokat három nagy csoportba osztottuk a funkcionalitásuknak megfelelően. Az első csoportba azok az alkalmazástípusok tartoznak, melyeket az asztali gépeken leggyakrabban használunk, mint például a különböző kliensprogramok (levelező, böngésző), irodai programcsomag stb.

A második kategóriába kerültek az informatikai folyamatok működéséhez elengedhetetlenül szükséges háttérrel biztosító kiszolgáló eszközök (webkiszolgálók, fájlserverek stb.). Végül külön tárgyaljuk azokat a megoldásokat, amelyek nem sorolhatók egyértelműen a fenti két csoport egyikébe sem (pl. beléptető rendszer).

A fent leírt csoportokon belül alkalmazástípusokat definiáltunk, és ezen belül vizsgáltuk az alkalmazástípus konkurens megvalósításait (pl. böngészők: Mozilla, Opera).

A vizsgált alkalmazástípusok szinte mindegyikéről elmondható, hogy több konkurens nyílt forrású megvalósításuk is létezik. Jelen tanulmánynak nem célja, hogy a különböző megvalósítások mindegyikét számba vegye vagy azokat teljes körűen összehasonlítsa. A lehetséges változatok közül azokat vizsgáljuk, amelyek a nyílt forráskódot támogató közösségben a legismertebbek, legelterjedtebbek.

Minden megoldást megegyező szempontok szerint tanulmányoztunk. Ennek a módszernek ugyanaz az előnye, mint a hátránya, a szempontrendszer egységessége segít az egy alkalmazástípusba tartozó különböző megvalósítások összehasonlításában, de az is nyilvánvaló, hogy teljesen eltérő céllal létrehozott eszközöket (más-más alkalmazástípusok) nehéz egységesen értékelni.

11.1. Vizsgálati szempontok rendszere

11.1.1. Funkcionális definíció

A definíció röviden elmagyarázza, hogy mi az adott alkalmazás szerepe az informatikai rendszerben, és meghatározza a funkcionalitási minimumot. Ez utóbbit a leggyakrabban használt és szükséges funkciókból vezetjük le. Az ideális (teljes) funkcionalitás megállapításához szükséges lenne minden egyes alkalmazástípus kereskedelmi és nyílt forrású változatainak beható vizsgálatára és összehasonlítására, de erre ezen tanulmány keretei között nincs lehetőség.

11.1.2. Projektháttér

A nyílt forrású szoftverfejlesztési projektek fontos jellemzője a projektháttér, amelynek összetevőiből következtetni lehet a termék megbízhatóságára,

funkcionális teljességére, az elérhető támogatásra és a szoftver jövőjére is.

Ebben a részben az alábbiakat fejtjük ki:

az adott szoftver története röviden összefoglalva;

a szoftverfejlesztési projekt állapota, amely lehet:

- fejlesztés alatt – a szoftverfejlesztés még az elején tart, a program nem stabil, funkcionálisan hiányos;
- stabil – a szoftverből (eszközből) létezik kibocsátott stabil verzió (1-nél nagyobb verziószám), a nyílt forrású közösségben már ismert és használt, de funkcionalitásában még fejlesztésre szorul (nem teljesíti a funkcionális minimumot);
- stabil és teljes – a funkcionálási minimumot vagy annál többet teljesítő, sikeres nyílt forrású szoftver;
- stabil, már nem fejlesztett – olyan szoftver, melynek fejlesztése valamilyen okból (forráshiány, más projekt) már egyáltalán nem, vagy már nem szervezeten folyik;

a projektháttér felméréséhez szükséges a környezet és erőforrások felmérése; ennek összetevői:

Ismert-e vagy becsülhető-e a szoftverfejlesztő csapat nagysága?

Kereskedelmi (zárt forráskódú) szoftver beszerzés előtti értékelésének fontos része a szoftvert előállító cég gazdasági és humán-erőforrásainak ismerete. Ezzel a lépéssel csökkenthetjük annak kockázatát, hogy az általunk megvásárolt eszköz továbbfejlesztése és támogatása rövid úton megszűnik. Nyílt forrású szoftver esetén (mint a Gazdasági modell c. fejezetben olvashattuk) akár saját szakemberekkel vagy harmadik cég bevonásával is megoldhatjuk ezt a problémát. Ezen felül ebből az adatból információt szerezhetünk a projekt sikerességéről, az eszköz ismertségéről (a sok fejlesztő, aktív levelezési lista általában azt jelenti, hogy az eszköz felhasználóinak száma elérte azt a kritikus tömeget, ami ahhoz kell, hogy a projekt ne haljon el, a fejlesztők ne lépjenek ki a munkából).

Létezik-e kereskedelmi változat az adott nyílt forrású eszközből?

Ha létezik olyan kereskedelmi termék, amely egy nyílt forráskódú szoftverből építkezik, egy cég üzleti lehetőséget látott meg benne és nem sajnálta a pénzt és energiát a szoftver termékesítésére (nyomtatott dokumentáció, a kezelői felület professzionális megjelenése, marketinganyagok stb.), ennek áldásos hatásaiból általában az eredeti projekt is részesül (akár anyagilag, akár funkcionális fejlődéssel).

Létezik-e olyan cég/szervezet/állam, amely hivatalosan is támogatja/szponzorálja az adott projektet?

Ez a szempont szinte megegyezik az előzővel, de más finanszírozási elveket és módokat is figyelembe vesz.

Milyen jellegű támogatás (support) létezik a szoftverhez?

Szintén a projekt sikerességének mérője lehet az adott eszközhöz elérhető támogatás. A nyílt forrású projektek elején általában kizárólag a levelezési listák adnak segítséget, a következő lépcsőfok az, ha az eredeti fejlesztők céggként hivatalos, garantált támogatást nyújtanak. A kritikus felhasználói tömeg elérése után nem csak az eredeti cégek, hanem erre szakosodott a nyílt forráskódból élő cégek is nyújtanak támogatást.

11.1.3. Funkcionalitás

Az adott megvalósítás konkrétan milyen tulajdonságokkal (feature-rel) rendelkezik. Itt elsősorban azt emeljük ki, ha egy termék a funkcionális minimumnál többet vagy attól eltérőt nyújt.

11.1.4. Dokumentáltság/kezelhetőség

Ebben a szekcióban számba vesszük a vizsgált eszközhöz elérhető dokumentációkat és az egyéb olyan információ-forrásokat, ahonnan a felhasználók a szoftver működésével kapcsolatban tájékozódhatnak. Ezek általában az alábbiak közül kerülnek ki:

- telepítési útmutató,
- felhasználói dokumentáció,
- adminisztrátori dokumentáció,
- fejlesztési dokumentáció,
- hogyanok (man pages),
- on-line help (projekt weblapján),
- GYIK – Gyakran Ismételt Kérdések (FAQ),
- levelezési lista.

Külön említést teszünk arról, ha a fent említett dokumentációk magyar nyelven is elérhetők.

A kezelhetőséget nehéz objektíven mérni, ez mindenképpen függ a felhasználótól és a kezelőfelülettől is. Mindenképpen meghatározzuk, hogy egy eszköz felhasználói felülete:

- parancssori vagy konfigurációs állományos,
- GUI:
 - natív, azaz grafikus felületen futó program (pl. gtk-s),
 - webalapú – azaz böngészőből elérhető
- egyéb vagy vegyes.

Természetesen ha a felületből van honosított, magyar nyelvű változat, akkor azt is

számba vesszük.

Az eszközök adminisztrátori felülete is fontos kérdés. A nyílt forráskódú eszközöknél a UNIX-os múltat idéző szöveges konfigurációs állomány az alapeset, ettől eltérően létezhet grafikus felület is, amely lehet

- az adott szoftver része
- vagy kifejezetten rendszer-adminisztrálásra alkotott felületek adott szoftverre vonatkozó része (pl. Webmin-modulok).

11.1.5. Erőforrásigény

Bármely szoftver esetén lényeges annak erőforrásigénye, melyet az alábbiak alapján tudunk becsülni:

- fejlesztők által megállapított minimális hardverszükségletek,
- fejlesztők által közzé tett teljesítménytesztek,
- a szoftver használatával szerzett gyakorlati tapasztalatok.

11.1.6. Sikertörténet

Ebben a szekcióban kerül megemlítésre, ha a vizsgált eszközzel valahol jelentős projektet sikeresen hajtottak végre.

11.1.7. Konklúzió

A szempontrendszer szerinti vizsgálat eredményei alapján, ha lehetséges, javaslatot teszünk a konkurens megvalósítások egyikére mint az adott alkalmazástípusnak legjobban megfelelő szoftverre.

11.2. Kliensanalízis (munkaállomások)

11.2.1. Desktopkörnyezet/disztribúciók

A **Desktopkörnyezet** az a felület, amelyet a felhasználó a monitoron maga előtt lát és nap mint nap használ. Mivel ez a felület egyfajta kiindulási alap, nagyon fontos, hogy könnyen beállítható, karbantartható és kényelmes legyen mind a felhasználó, mind pedig az adminisztrátor számára. Erről a felületről indítjuk a többi programot, mint például a levelezőt vagy a böngészőt.

Linux alatt számtalan ablakkezelő rendszer létezik. A rendszer elnevezés itt jogos, hiszen nemegyszer több száz programot és több ezer funkciót valósítanak meg. A kisebb tudású, ám egy 386-os gépen is tökéletesen gyorsan futó kezelőfelületről egészen a nagyobb, számításigényesebb felületig mindenki megtalálhatja a számára megfelelőt. Többéves csatározás után 2 nagy desktopkezelő került ki győztesen a harcból. A dobogó felső fokán a KDE – The K Desktop Environment és a GNOME osztozkodnak szinte egyenlő nagyságú felhasználói táborral.

A KDE projektet Matthias Ettrich indította 1996-ban, azzal a céllal, hogy egy komplett grafikus környezetet hozzon létre, ne csak egy újabb ablakkezelőt. A kezdetek kezdetén egy közepes tudású ablakkezelő programnak indult, ám a fejlesztésben nagyon sokan (többek között profitorientált cégek is) nagy fantáziát láttak, ezért pénzt és szaktudást fektettek bele. Az első stabil 1.0 verzió megjelenésekor már látszott, hogy kinőtte az ablakkezelő megfogalmazást. Ekkor már rendelkezett interneteléréshez és irodai munkához szükséges szinte minden komponenssel. Mára szinte minden Linux-terjesztés szerves része lett, és több millióan használják ablakkezelőnek. Rengeteg komponenst és külön programot is tartalmaz, pl. szövegszerkesztő, táblázatkezelő, prezentációkészítő, kalkulátor, levelező, fájlkezelő, internettárcsázó, képszerkesztő, számtalan játék stb. Ezeknek a programoknak köszönhetően igen nagy felhasználói táborra tett szert, és ennek köszönhetően sokan fejlesztik – körülbelül 1-2 évente növekszik egyet a stabil verzió verziószáma. **A jelenlegi stabil verzió a KDE 3.1.1.** Minden, a KDE-vel kapcsolatos információ megtalálható a <http://www.kde.org> címen, illetve egy magyar non-profit csapat gondozásában a <http://www.kde.hu> címen is. Érdeemes továbbá megnézni a <http://www.kde.org/awards.html> címen a KDE-felület által nyert díjakat. Pontosan a fenti tényeknek köszönhetően rengeteg funkcióval, komplett dokumentációval, „hogyan”-leírásokkal, levelezési listákkal és kereskedelmi támogatással is rendelkezik, akár magyar nyelven is.

A KDE főbb alkalmazásai:

- **keyword: teljes értékű szövegszerkesztő program;**
- **kspread: teljes értékű táblázatkezelő program;**
- **kpresenter: prezentációkészítő;**
- **kmail: minden igényt kielégítő levelezőkliens, amely támogatja az összes levelezési protokollt és titkosítási eljárást, valamint a több felhasználót és külön levelesláda technológiát;**
 - **kmultimedia: ebbe a programcsoportba tartozik az összes, multimédiával foglalkozó program, zenei-, film-, tévé- stb. lejátszó és -felvevő szoftverek;**
 - **kgames: számtalan színes és elgondolkodtató játék;**
 - **knetwork: internetezéshez szükséges beállító, statisztikakészítő és monitorozó programok, kezdve az egyszerű modemtől az ADSL-en keresztül a WLAN kapcsolatig.**

A programcsomag licence GNU/GPL. A programcsomagnak egyetlen negatívum róható fel: az erőforrásigénye. Még ha csak ablakkezelőnek akarjuk is használni, akkor sem árt egy Celeron 300-as gépet beszerezni minimum 64 MB RAM-mal. Persze ez a hardverkövetelmény a mai korban nevetségesen kevésnek tűnik, azonban jó tudni, hogy 486-os gépeken ne akarjunk KDE-t futtatni.

A **GNOME** a „GNU Network Object Model Environment”, azaz a „GNU Hálózati

Objektum-modell Környezet” kifejezés rövidítése. A GNU jelentése „GNU's Not Unix”, azaz a „GNU Nem Unix”. A GNOME projektet Miguel De Icaza kezdte el a Mexikói Egyetemen, később többen csatlakoztak a fejlesztéshez, majd a mexikói sulinet program keretében felkarolták, és ma már több mint 300 aktív fejlesztővel rendelkezik. A KDE komoly riválisa, és vele ellentétben már a kezdetektől a GNU irányelveket követte. Szinte kísértetiesen ugyanazokat a funkciókat lehet vele megvalósítani, mint a KDE-vel, csak kicsit más nevű programokkal és másképpen. Az egészséges versenyszellemből kifolyólag az office-rendszer a KDE-ben jobb, azonban a GNOME-hoz írt levelező kliens még az Outlook kereskedelmi változataival is felveszi a versenyt. A GNOME programcsomag része minden disztribúciónak, hiszen nem tartalmaz nem szabadszoftver-jellegű komponenst. Természetesen kaphatók hozzá kereskedelmi kiegészítések, de ezeket külön kell vásárolnunk (pl. levelezőhöz plug-in). Az utóbbi időben jelent meg a piacon a teljes mértékben kereskedelmi változat, a Ximian is. Erőforrásigénye bőven a KDE alatt van – ha a processzorteljesítményt nézzük, azonban a GNOME is szereti a bőséges memóriát. A GNOME-programok felsorolása a terjedelem miatt lehetetlen, mivel az élet minden területére megírták a megfelelő alkalmazást. Ilyen például a remek levelező, amelyről a következő fejezetben szó lesz, vagy a gnome-pilot program, amely PDA típusú gépek összekötését és szinkronizálását teszi lehetővé. Az utolsó stabil verzió a 2.2.1-es, amely bár számozásban elmarad a KDE-től, ám tudásban semmiképpen sem. Konkrét információk a <http://www.gnome.hu> és <http://www.gnome.org> oldalon találhatóak.

Konklúzió: Mindkét felület igen népszerű Magyarországon és külföldön egyaránt. Rendszerint az dönti el, hogy melyiket választjuk, hogy mekkora kapacitású gépre szeretnénk telepíteni, illetve melyiknek szimpatikusabb a felépítése, felhasználói felülete (amelyet természetesen teljesen testre szabhatunk). Mivel a két rendszer között teljes mértékű átjárás van, ezért ha KDE az ablakkezelőnk, akkor is használhatjuk a GNOME alkalmazásait és viszont.

Disztribúció vagyis terjesztés: Linux alatt minden program egyenként és csoportosan, csokorba szedve is elérhető. Sokkal egyszerűbb egy CD-n egybefogott, telepítővel rendelkező komplett rendszert használni, hiszen minden kéznél van, mint egyenként minden programot megszerezni és „kézzel” telepíteni. Ebből az alapötletből fejlődtek ki a különböző terjesztések, amelyek kényelmesebbnél kényelmesebb telepítőfelülettel és honosított programokkal rendelkeznek. Ha szükséges, letölthetjük a programokat egyenként is és lefordíthatjuk azokat gépünkre, azonban sokkal célszerűbb telepíteni egy szervezet által összeállított komplett csomagrendszert. Sok terjesztés kifejezetten profitorientált (cégszerű), azonban vannak non-profit összeállítások is. A legismertebb terjesztések: Mandrake, Red Hat, SuSE, Debian. Számuk napról napra változik, mert rengeteg cég lát fantáziát egy-egy saját terjesztés megalkotásában és forgalmazásában.

Magyarországon az egyik legismertebb végfelhasználói Linux a SuSE, amely már teljes mértékben magyar felülettel is rendelkezik. A rendszermérnökök által preferált terjesztés a Debian/GNU Linux, amely minden igényt kielégít szerveroldali megoldásokban is. Nem véletlen a GNU elnevezés, hiszen a kezdetektől kizárólag GNU szoftver komponensek kerülhetnek a „core” telepítőkészletbe. Minden területnek megvan a maga jól bevált terjesztése, így nem is csoda, ha Kínában a kormány egy saját disztribúciót készítettett „Red Flag” néven.

Hazánkban 2001-ben, állami támogatással kezdte meg egy cég az első magyar Linux-disztribúció megalkotását. Az UHU Linux fejlesztők célja, hogy könnyen telepíthető, teljes mértékben magyar nyelvű rendszert hozzanak létre. Az UHU Linuxnak várhatóan desktop- és szerverkiadása is lesz, a megvalósítás jelenleg a Release Candidate tesztállapotban van.

További információk:

Debian: <http://www.debian.hu>

Redhat: <http://www.redhat.com>

Suse: <http://www.suse.de>

UHU: <http://www.uhulinux.hu>

11.2.2. Levelezés – kliensek

Egy levelezőkliens a programok alapfunkcióján (e-mail szerkesztése, küldése, fogadása) túl számos egyéb tulajdonsággal is fel van ruházva. Ezek közé tartozhat például a hírcsoportok, a címjegyzék, több felhasználó, a titkosítás és az elektronikus aláírás kezelése is. A Linux-alapú grafikus levelezőkliensek világában is természetesen a legelterjedtebbek a két nagy munkafelület, a KDE és a GNOME saját alkalmazásai.

Ezeket túl említésre méltók még a különböző böngészők beépített moduljai, illetve a karakteres, karaktergrafikus megvalósítások is.

Fontos például, hogy az utóbbi csoportba tartozó pine és mutt kidolgozottságában és funkcionalitásában mindenképpen felveszi a versenyt konkurensével. A tanulmányban a felhasználók által jobban kedvelt grafikus alkalmazásokat vesszük górcső alá.

A GNOME két levelezőt is sajátjának tudhat. A Balsa a GNOME eredeti beépített levelezője, de az utóbbi időben feltűnt a GNOME piaci változatának, a Ximiannak fejlesztéseként világot látott Evolution, amely funkcionalitásában a vezető piaci levelezőkkel vetekszik. A KDE saját levelezője, a Kmail igen elterjedt a Linux-felhasználók körében. A böngészők moduljai közül ebben a fejezetben csak a Netscape beépített levelezőjét vizsgáljuk. Végül szemügyre vesszünk két, a böngészőktől és munkafelületektől független szoftvert, a pronto-t és a sylpheed-et.

Balsa

A Balsa egy kis méretű, egyszerű, de működőképes levelező, amelyet a Windows-alapú Eudora alapján valósítottak meg. Mint a GNOME őseleme, már régen a stabil jelzővel dicsekedhet. Ismeri az alapvető levelezési protokollokat, ezen felül rendelkezik helyesírás-ellenőrzővel. Sajnos ennek honosítási állapotáról nem sikerült tényszerű információt találnunk. A Balsa címjegyzéke zökkenőmentesen működik együtt GnomeCard-dal, és az LDAP szabvánnyal is kompatibilis. A Balsa projektet feltételezhetően halálra ítélte az Evolution megjelenése, mert annak tudása nehezen hasonlítható össze kistestvérel. A Balsa egyetlen előnye (az Evolutionhoz képest) az alacsony hardverigény, de tapasztalataink szerint pl. a sylpheed hasonló hardverkörnyezet mellett egy funkcionalitásában jobb és használatában kényelmesebb levelezési alternatívát nyújt. Ettől függetlenül a Balsát használni vagy fejleszteni vágyókat a hivatalos honlap egy levelezőlistával és egy FAQ-val várja.

Evolution

A Ximian fejlesztői egy olyan programot hoztak létre, ami könnyen veszi fel a versenyt a piacot vezető levelezési programokkal, s mára már elérte a stabil állapotot. A Microsoft Outlook-hoz, a Lotus Notes-hoz és a Symantech Act-hez hasonlóan az Evolution nem csak egy sima levelező, de tartalmaz beépített telefonkönyvet, címjegyzéket, naptárat, határidőnaplót is, valamint csoportmunkát segítő modulokat. Bár az Evolution is szabad szoftver, a Ximian kifejlesztett egy Exchange modult, amelyet kizárólag megvásárolni lehet. A kiegészítő modul segítségével az egy személyt kiszolgáló funkciókon túl az Evolution kompatibilis lesz az Outlook és a Lotus Notes backoffice kiszolgálójával is.

Az Evolution levelezője több mail accountot és több postaládát képes egyszerre kezelni, levelek szortírozását egy fejlett szűrőrendszer segíti. A beépített editor képes dinamikus üzenetformázásra s azok HTML-formátumban való továbbítására is (természetesen HTML-üzenetek fogadása is lehetséges). Beépített helyesírás-ellenőrzővel rendelkezik, de ennek honosítási stádiumáról nincs információnk. Az Evolution ettől függetlenül teljesen magyarított, kivétel ezalól a friss Exchange kiegészítő modul. Egy jól optimalizált kereső segítségével egy nagy méretű, egyébként Unix szabványokhoz hű postaládában is másodperceken belül rálelhetünk a keresett szövegre. Az Evolution ismeri az elektronikus levelezés főbb protokolljait (POP3, IMAP, SMTP), valamint a titkosításhoz szükséges PGP/GPG, SASL és SSL/TLS módszereket. Az import-export funkciókkal ellátott rendszer kompatibilis a Unix szabványokkal és a vezető levelezők állományformátumaival.

Az Evolution címjegyzéke teljes körű kapcsolat-menedzserként működik, és palm operációs rendszerű PDA-kal is szinkronizálható. A megoldás kompatibilis továbbá a Microsoft Outlook és a Lotus Notes elektromos névjegykártyáival, vCardjaival is. A címjegyzéken felül az Evolution egybe van építve egy naptárral, amely határidőnaplóként is funkcionál. A találkozószervezés funkció is elérhető,

természetesen Outlookkal és Lotus Notes-cal kompatibilisan, Palmra szinkronizálhatón. A határidőnaplón túl az Evolution egy „todo”-listát is kezel, továbbá különböző, konfigurálható figyelmeztetésekre képes (e-mail, hangjelzés stb).

Az Evolution fejlesztői bőségesen el vannak látva információval. A Ximian honlapján egy levelezőlista, egy útmutató, több válogatott folyóiratcikk és technikai dokumentum áll a fejlesztők azonnali rendelkezésére. Felhasználók számára elérhető felhasználói levelezőlista, egy FAQ és egy online tudásbázis, amelyek segítségével problémánkat biztosan orvosolni tudjuk. A Ximian programcsomag megvásárlásával ügyfélszolgálati támogatáshoz is juthatunk. Az Evolution egyetlen hátránya, hogy ellentétben a GNOME programjainak nagy részével, igen magas a hardverigénye, de egy 128 MB RAM-mal szerelt Pentium II-es (vagy hasonló szintű) gépen már zökkenőmentesen fut. Természetesen ennél kisebb konfigurációval is beéri, de mivel tudjuk, a program alapvető felhasználói funkciókat lát el, használatát ennél gyengébb gépen nem ajánljuk.

Kmail

A Kmail a KDE levelezőprogramja, ugyancsak stabil stádiumban van. Ez a program az Evolutionnel ellentétben nem hazudtolja meg munkafelületének hardverigényét. A megoldás számos professzionális funkcióval ellátott, támogatja az elektronikus levelezés megszokott szabványait, és az Evolutionhoz hasonlóan SSL/TSL OpenPGP vagy GnuPG segítségével biztonságos levelezésre is képes. Együttműködik a KDE Address Book (KAB) címjegyzékével, beépített helyesírás-ellenőrzővel van ellátva. **A Kmail mára már teljesen magyarul beszél, továbbá mivel a teljes KDE-be integrálni lehet az Ispell-helyesírásellenőrzőt (amit egyébként az OpenOffice.org is használ), a helyesírás-ellenőrzés is megoldott.** HTML-levelek írására nem alkalmas, de probléma nélkül tudja olvasni azokat. Több e-mail accountot és postaládát kezel, és fejlett levélszortírozó rendszerrel rendelkezik. Ezekon a funkciókon túl egy SPAM (nem kívánatos e-mail) védelmi megoldás is helyet kapott. A Kmail saját mailbox-formátumát használja, de természetesen az egyéb népszerű levelező-programokkal és formátumokkal kompatibilis export- és importfunkciókra is képes.

A Kmail jól dokumentált program, a FAQ mellett egy biztonsági áttekintés is helyet kapott honlapján. A fejlesztők munkáját egy levelezőlista segíti.

Netscape 4 és 7, valamint a Mozilla

A Netscape-ről már előző fejezetünkben volt szó, ezért itt csak a levelező funkciói kapnak helyet. A Netscape Communicator 4-es sorozata egy címjegyzékkel ellátott Messenger nevű levelezőmodult is tartalmaz, amely kompatibilis az Eudora és az Outlook Express levelezőkkel, továbbá szinkronizálható a Palm operációs rendszerrel rendelkező PDA-kkal. A POP3, IMAP és SMTP protokollokon

keresztül a levelezés minden feltétele adott. A Netscape-pel készíthetők levélsablonok is. A 7-es sorozatú Netscape funkcionalitásában nem nagyon különbözik a 4-es sorozattól, viszont a cég bevallása szerint igen komolyan fejlesztették a használhatóságot. Ilyen például a postaládán belüli postaláda lehetősége is. A Netscape szabad forráskódú verziójaként induló Mozilla mára már annyira fejlett lett, hogy a Netscape 7.0-es szériájának alapját képezi. Kb. negyedévente jelenik meg egy-egy új verzió, amely használhatóságában, gyorsaságában felülmúlja az egyébként nagyon jól sikerült Netscape 7.01-et is. Mind a levelezőkliense, mind a böngészője sokoldalú, rengeteg apró és nagyobb újítással. A legutolsó stabil verzió az 1.3-as, a fejlesztés jelenleg az 1.4a-nál tart. Az 1.3-as verzióból létezik teljesen magyar nyelvű változat is, amelyben már nem csak a menük, de a leírások is magyar nyelvűek.

Pronto

Sokak által dicsérettel böngészőktől és munkafelületektől teljesen független szabad fejlesztésű stabil levelező program. Hardverigénye alacsonyabb, mint a Kmailé vagy az Evolutioné. A Pronto honlapján egy levelezőlistára iratkozhatnak fel az érdeklődők. A megoldás a Unix-alapú MBOX postaláda-szabvánnyal kompatibilis, HTML-üzenetek olvasására (de írására nem) is képes, viszont igen fejlett címjegyzékekkel láttak el. Az Ispell és az Aspell helyesírás-ellenőrzőket ámogatja (az Ispellnek magyar változata is létezik), bár tudomásunk szerint senki nem próbálta ezt még a Prontóba integrálni. A Pronto sajnos még nem beszél magyarul és honosítási szándékokról sem tudunk. Pronto az állománycsatolásokat tökéletesen kezeli, és a levelek letöltését POP3 protokollon keresztül teszi lehetővé. Bár a legtöbb levelezőben alapfunkció, azért érdemes megemlíteni az aláíráskezelést, ami lehetőséget ad rá, hogy minden levelünk alján egy előre definiált szöveget jelenítsünk meg.

Sylpheed

Végül, de nem utolsó sorban ejtsünk szót a japán fejlesztésű szabad levelezőről, a Sylpheedről. Bár ez még „unstable” státuszban van, a 0.7.5-ös változat tapasztalataink alapján teljesen üzembiztosan működik. A minimális hardverigényű, roppant gyors program igen nagy népszerűségnek örvend. A projektnek egy Sylpheed-Claws nevű mellékága is van, amely próbálja a fő fejlesztési ág szoftverét javítani, és ha egy-egy ötletük beválik, az belekerül a főágba. A sylpheed több e-mail account kezelésére képes, egyszerű, de jól használható levélszortírózó rendszerrel is ellátott, csatolt fájlokat kitűnően kezelő, roppant gyorsan fejlődő projekt. A sablonok, aláírások, beépített (csatolt) képnézegető mind hasznos funkcióknak bizonyulnak. POP3, SMTP, IMAP4 protokollokat akár biztonságos csatornán is tudja kezelni. Továbbá a sylpheed el van látva egy XML-alapú, LDAP kompatibilis címjegyzékekkel és használja a GnuPG kriptográfiai rendszert is. A Sylpheed honlapján a funkciók leírásán túl egy levelező lista várja az érdeklődőket. A Sylpheed bár több karakterformátumot jól kezel, sajnos nem rendelkezik

helyesírás-ellenőrzővel, és meg honosítása sem kezdődött meg.

11.2.3. Irodai programcsomagok használata

Mint nevéből is kiderül, ez a program, illetve program-gyűjtemények az irodai munkánk megkönnyítését szolgálják. Manapság ha valaki használ számítógépet, az valószínűleg használ, vagy legalábbis találkozott már irodai programcsomaggal, melynek fontosabb elemei:

- szövegszerkesztő,
- táblázatkezelő,
- prezentációkészítő,
- adatbázis-kezelő,

Ebben a kategóriában egyértelműen a Microsoft Office a piacvezető termék, amely régen külön terjesztett programokból (Word, Excel stb.) tevődik össze. A Microsoft Office-nak nem létezik Linuxos változata. A Microsoft történelmileg legnagyobb vetélytársa mindig a WordPerfect volt, melynek linuxos változata is elkészült, de ennek terjesztése és támogatása mára megszűnt. A Corel Wordperfect 8-as linuxos változatát – amit ingyen lehetett letölteni a Corel honlapjáról – még lehet regisztrálni, de ez a program kereskedelmi utódjával, a WordPerfect Office 2000-rel szemben, csak egy szövegszerkesztő, és nem egy komplett irodai programcsomag. A linuxos Wordperfect Office 2000-et is (bár relatíve friss szoftvernek számít) már kivonták a piaci forgalomból, továbbá a linuxos WordPerfecteknek nem létezik honosított változata.

Másik, a WordPerfecthez képest újonnan érkezett vetélytárs a StarDivision által fejlesztett StarOffice volt. A több platformot, köztük a Linuxot is támogató StarOffice-t nem régiben megvásárolta a Sun, hogy a monopolhelyzetben lévő Microsoft Office komoly ellenfelévé fejlessze.

A Sun a szabad szoftver közösséghez fordult, amikor felszabadította a StarOffice kódját, és megalapította az OpenOffice.org projektet, ami egy, a StarOffice kódján alapuló ingyenes és sok szempontból szabadnak is mondható irodai programcsomag kifejlesztését tűzte ki céljául.

A már sokat emlegetett KDE és GNOME is saját irodai programcsomagokat épített be a rendszerébe. Természetesen, mint minden egyéb KDE- és GNOME-program, a Koffice és a GNOME Office akkor is használható, ha nem KDE vagy GNOME felületet futtatunk gépünkön. A két rendszer között alapvető különbség, hogy amíg a KDE maga fejlesztette ki irodai programcsomagját, addig a GNOME Office külső projektekből építkezik.

A felsorolt programok közül nem tarjuk fontosnak a WordPerfect részletes, már említettekén túli tárgyalását, de a továbbiakban a StarOffice-t, illetve OpenOffice-t, a Koffice-t és a GNOME Office-t közelebbről fogjuk vizsgálni.

GNOME Office

A GNOME Office a honlapján található információk alapján több külső, illetve belső fejlesztésű programból tevődik össze. Főbb részei a Gnumeric táblázatkezelő, és az AbiWord szövegszerkesztő. A GNOME Office részeként tartják számon többek között a már tárgyalt Galeon böngészőt, a Balsa és Evolution levelezőt, és számos egyéb programot, mint például a Sodipodi és a Sketch vektorgrafikai szoftvert, a gfax faxprogramot, a gncash személyi-háztartási költségfigyelőt, a több adatbázissal (MySQL, PostgreSQL, Oracle stb.) is együttműködő gnome-db adatbáziskezelőt, vagy, a szabad szoftveres csodaként számon tartott GIMP fotómanipulációs programot, amely a piacvezető Adobe Photoshop-pal is felveszi a versenyt. A GNOME Office prezentációkészítője az Achtung lesz, de ennek fejlesztése még igencsak kezdeti stádiumban van. Ezeknél a programoknál a fejlesztői összetartó erő relatíve kicsi, és ez kihat a honosítási folyamatokra is. Az AbiWordöt már több távol-keleti és szláv nyelvre sikerült honosítani, de a magyar változat még nincs üzemképes állapotban. Ezzel szemben a Gnumeric és a GIMP magyar változatai már teljes egészében elkészültek. A honosítási folyamatok állapotáról nehéz információt szerezni, hiszen nincs egy átfogó GNOME Office fejlesztő és honosító csapat, hanem a külső programok állapotát kell egyesével elemezni.

AbiWord

Az AbiWord fejlesztői bőségesen ellátják információval mind a fejlesztőket, mind a felhasználókat. A projekt honlapján a FAQ-n és a használati kézikönyvön túl külön levelezőlista várja mindkét érdeklődő csoportot. Az AbiWord talán legnagyobb előnye a minimális hardverigény. A program a hivatalos álláspont szerint egy gyenge Pentium processzoron akár 16MB RAM-mal is képes futni. A szoftver legnagyobb hátránya számunkra, hogy a magyar ékezetes karaktereket még nem kezeli tökéletesen. Az AbiWord utolsó fejlesztői kiadásánál tart, és az 1.0-s, stabil változata a közeljövőben várható. Funkcionalitásában az AbiWord tudja azt, amit egy egyszerű szövegszerkesztő programnak tudnia kell, bár ennek ellentmond, hogy a táblázatok szövegbe illesztésére egyáltalán nem képes. A fejlesztők állítása szerint az AbiWord képes egyszerűbb Microsoft Word dokumentumokat importálni, viszont az ugyanezen formátumba való exportálás még csak fejlesztési tervben szerepel.

Gnumeric

A GNOME táblázatkezelője, a Gnumeric már stabil és teljesen honosított 1.0.6-os változatát akár azonnal le is tölthetjük a GNOME Office honlapjáról. Ugyanitt találhatóunk egy levelezőlistát, egy felhasználói kézikönyvet és egy FAQ-t. A Gnumeric nem dicsekedhet olyan alacsony hardverigényekkel, mint az AbiWord, de egy GNOME munkafelületet futtatni képes gépen zökkenésmentesen működni kell. A Gnumeric tudja azt, amit egy átlagos felhasználó elvár egy táblázatkezelő programtól. Sikeresen képes megnyitni a piacvezető Microsoft Excel mentéseit, és

ugyanezen program formuláinak 95%-át ismeri is. A Gnumeric saját XML-alapú adatformátumot használ, de ha el tudunk tekinteni a dokumentumformázástól, akkor akár Microsoft Excel állományba is tud menteni.

KOffice

A KOffice hardverigénye megegyezik a KDE-nél már ismertetett paraméterekkel. A KDE, mivel maga fejleszti a Koffice-t, és a GNOME Office-szal szemben nem külön fejlesztett részelemekből tevődik össze, a dokumentációkat is jól kézben tartja. A <http://www.koffice.org>-on egy felhasználói és egy fejlesztői levelezőlistát és egy FAQ-t találunk, míg a docs.koffice.org bőven tartalmaz dokumentációt. A KDE installálása után egyéb KOffice dokumentációkhoz és a súgókhöz is hozzá férünk. A szoftver honosítása a KDE-től megszokott előrehaladott stádiumban van. A KOffice a KWord szövegszerkesztőn és KSpread táblázatkezelőn túl a KPresenter prezentációkészítőből, a Kugar üzleti jelentések összeállításához segítséget nyújtó programból, a KChart grafikonkészítőből, egy Kivio flowchart, munka- és ügyvitel-tervező szoftverből és két képmanipulációs programból áll. Jelen a tanulmányban a GNOME-hoz hasonlóan csak a legfontosabb részprogramok, a KWord, a KSpread és a KPresenter funkcióit fogjuk közelebbről vizsgálni.

KWord

A KWord egy jó minőségű szövegszerkesztő program, amely teljesen magyar nyelvű, sőt az Ispell helyesírás ellenőrzőt is bele lehet építeni. A KWord tartalmaz minden olyan funkciót, amire egy átlagos felhasználónak szüksége lehet. A teljesség igénye nélkül érdemes megemlíteni a komoly dokumentumformázási lehetőségeket, a sablonokat és a beépített dokumentumszervező rendszert, amelynek segítségével tanulmányok formázása és tartalomjegyzék készítése egy pillanat alatt megtörténhet. Külön érdemes megemlíteni, hogy a Kword-öt frame-orientált felépítése alkalmassá teszi azt egyszerűbb kiadványszerkesztésre is.

KSpread

A Kwordhöz hasonlóan a KSpread is egy jó minőségű, funkcionalitásában teljesnek mondható táblázatkezelő. A program teljességét sablonok sokasága, több mint 100 matematikai és logikai formula, sorozatok kezelése, komoly formázási funkciók, mint például feltételen alapuló cellaszínezés, és egyéb külalakkezelő funkciók biztosítják.

KPresenter

A KPresenter a Microsoft PowerPoint-hoz hasonló prezentációkészítő program. A funkcionalitásában teljesnek számító KPresenter mindenképpen felveszi a versenyt Windows-alapú vetélytársával. A KPresenterhez jár egy grafikagyűjtemény és több előre definiált sablon, amelyek az előadások készítésekor hasznos eszköznek szoktak nyilvánulni. Az általánosságban elvárt funkciókat a KPresenter tudja.

Ilyenek például a háttérbeállítások, módosítható sablonok, objektumkezelés és animálás, különleges szövegformázási funkciók, fejléc- és lábjegyzet-kezelés. Hasznos funkció, hogy előadásunkat akár HTML-oldalként is menthetjük, így könnyen tudjuk publikálni az interneten. Ezen felül a KPresenter lehetőséget ad a közvetlen PostScriptbe való mentésre. A KPresenter XML dokumentumformátumot használ, és a KWordhöz, valamint a Kspreadhez hasonlóan a más termékekkel való együttműködési képességeinek leírását a docs.kde.org-on találjuk.

A StarOffice és az OpenOffice

A StarOffice a szabad szoftver közösség körében az utóbbi hónapokig a legelterjedtebb irodai programsomag volt. A német fejlesztésű programot nemrégiben megvásárolta a Sun, és megnyitotta a forráskódját. Ez az akció rengeteg programozót vonzott a létrehozott OpenOffice.org projektbe, hiszen ezáltal, a Mozillához hasonlóan, egy igen népszerű alkalmazás jobbá tételének részesei lehetnek. A Sun saját licence alatt adta ki az OpenOffice-t, amely csak részlegesen nevezhető szabadnak. A terméket bárki fejlesztheti, a fejlesztésről jelentést kell tennie a Sun-nak, és az újonnan keletkezett kód tulajdonáról a Sun javára le kell mondania. A StarOffice legutóbbi stabil és ingyenes verziója az 5.2-es, ami már a Sun gondozásában került kiadásra. Ebbe beilleszthető egy magyar helyesírás-ellenőrző, valamint a menük is magyar nyelvűek. **Az OpenOffice 1.0.1 verziója a legfrissebb stabil változat, de elődje, az utolsó béta, a 641d is teljességgel használható volt. Ezt igazolja az is, hogy e dokumentum nagy része is az OpenOffice 641-es változatával készült. A Sun kiadta StarOffice 6-os kiadását is, amely az OpenOffice-ra épül, kiegészítve azokkal a funkciókkal, amelyek forrásának megnyitása jogilag nem volt lehetséges. A StarOffice 6.0 az eredeti tulajdonosok elképzelései és az OpenOffice-szal szemben nem ingyenes.**

Mint arról már a bevezetésben írtunk, az OpenOffice honosítását a magyar szabad szoftver közösséget összefogó egyik szervezet, az FSF.hu Alapítvány aktivistái és a nekik segítő önkéntesek hajtották végre. A honosításon túl a már teljesen magyarul beszélő OpenOffice-t sikerült egy jól működő magyar helyesírás-ellenőrzővel is integrálni.

Az OpenOffice igen memóriaigényes alkalmazás, 64MB RAM alatt semmiképpen ne próbálkozzunk használatával, az ajánlott konfiguráció azonban 128 MB. A továbbiakban részletesen az OpenOffice-t vizsgáljuk, hiszen sokban nem különbözik a többi, Linuxon elérhető office-verziótól, azonban teljes mértékben honosított.

Az OpenOffice négy alapvető építőelemből áll. Ezek a Writer – szövegszerkesztő, a Calc – a táblázatkezelő, az Impress – a prezentációkészítő, és a Draw – a rajzprogram. Az OpenOffice.org-ról elmondható, hogy saját (native) szabad (XML-alapú) állományformátumokat használ, de más szoftverekkel is kompatibilis, többek között szinte tökéletesen tudja importálni a piacvezető Microsoft Office-

állományokat.

A Writer egy teljes körű szolgáltatásokat nyújtó, funkcionalitásában a Microsoft Worddel versengő és kényelmi funkciókban azt már szinte leghagyó szövegszerkesztő program. Különleges képességei közé tartozik külső levelezőkkel való könnyű integrálhatósága, fejlett szövegindexelője, valamint az automata szövegtördelési funkció. Nem csak dokumentumokat, hanem sablonokat is képes importálni és akár HTML-be is képes exportálni.

A Calc táblázatkezelő komoly versenytársa a Microsoft Excelnek, különleges képességei közül talán a Netbeans alapú felhasználói függvénykészítőjét érdemes kiemelni, hiszen ennek segítségével bármilyen testre szabott táblázatkezelő funkciót el tudunk készíteni.

Az Impress a Microsoft PowerPoint szinte minden funkcióját magába foglalja. Az Impressben lehetőségünk van emberi beavatkozást nem igénylő prezentációk készítésére, illetve kizárólag vetítőmodullal is rendelkezik. Ennek segítségével olyan számítógépen is le tudjuk játszani a prezentációt, ahova az OpenOffice nincs feltelepítve.

A Draw egy a CorelDrawhoz hasonló vektorgrafikai szerkesztő, amely rendelkezik fejlett 3D formatervezéssel, valamint egy beépített kis teljesítményű raszteres képmanipulátorral is.

Konklúzió:

Irodai programcsomagok terén igen bő a választék a szabad szoftverek között. Ezek talán egyetlen hátránya, hogy a piacvezető Microsoft Office-termékek állományformátumaival csak többé-kevésbé kompatibilisek. Ezen a téren az OpenOffice kiemelkedő teljesítményt nyújt, hiszen az ebben készült, .doc formátumban mentett anyagot a Word szövegszerkesztővel beolvasva csak néhány kisebb tördelési hibát találunk. Ennél nagyobb hiba mutatkozik viszont az Excel táblázatok makróinak értelmezésében, melyekkel talán a Gnumeric birkózik meg a legjobban.

11.2.4. Böngésző

A webböngészőt leginkább a világhálóra nyíló ablakként definiálhatjuk, melynek feladata a honlapok megjelenítése, illetve az ehhez szükséges kliensoldali programok futtatása.

A fejezetben elsődleges célunk a Linux alatt elérhető grafikus böngészők rövid összehasonlítása. A karakteres böngészők családja is igen nagy, de mivel népszerűségük egyre csökken, itt csak megemlítjük őket.

A karakteres böngészők közül a legnépszerűbb a Lynx, amely a folyamatos fejlesztéseknek köszönhetően, rengeteg funkciót támogat (pl. SSL-kapcsolat), és elmondható róla, hogy grafikus felület hiányában is teljes körű megoldást ad.

Az X11 grafikus felülettel rendelkező böngészők piacán erős konkurenciaharc alakult ki az utóbbi időben. Pár évvel ezelőtt egyetlen linuxos program volt, amely használhatónak bizonyult, a Netscape Communicator 4-es sorozata. Amikor a Netscape már végképp elvesztette a Microsoft Internet Explorer elleni harcot a Windows platformokon, váratlan üzleti döntést hozott. Megnyitotta a böngésző forráskódját, ezzel megteremtve a Mozilla projektet. **A projekt most, 5 évvel később – ahogy azt már fentebb röviden említettük – az 1.3-as változatánál tart. A Netscape természetesen nem hagyta abba a böngészőinek fejlesztését, a 4-es sorozat jelenleg 4.80-as változatnál tart, ezen kívül megjelent a Mozillára épülő 7-es sorozat is – a 6-os Netscape-széria meglehetősen sikertelensége után a 7-es széria várhatóan sokkal jobban fog szerepelni, mert a 7.01 változat alapjául a Mozilla 1.0.1 verziója szolgált.**

A KDE és a GNOME projektek is kifejlesztették a saját szoftvereiket. A Konqueror, a KDE böngészője, a Microsoft Internet Explorer mintájára egyben a felület fájlkezelőjeként is funkcionál. A Galeon a GNOME projekt része, és – hasonlóan a 6-os Netscape-hez – a Mozilla motorját használja. Nemrégiben tűnt fel a böngészők piacán az Opera, amely nagy port kavart még az Internet Explorer által uralt Windows platformokon is. Az Opera nem szabad szoftver, de az ingyenes változata letölthető a gyártó cég honlapjáról. Az ingyenes változat egyetlen hátránya, hogy a jobb felső sarokban állandóan jelen van egy – egyébként nem túl zavaró – reklámcsík.

Két funkciót érdemes külön kiemelni: minden vizsgált böngésző képes legalább 128 bites biztonságos SSL-kapcsolat létrehozására, és mindegyik képes Java-programok futtatására. A Netscape-ekben és a Mozillában beépített levelezőkliens található, a KDE Konquerorja szükség esetén a Kmailt hívja meg. A többi programban a felhasználónak kell definiálnia, hogy milyen levelezőt használjon a böngésző.

Mozilla

A projekt indulásakor sok programozót vonzott, hiszen a Netscape fejlesztői az egyik legelterjedtebb linuxos program forráskódját adták át a szabad szoftver közösségnek.

A Mozillához kapcsolódó dokumentáció leginkább az on-line segítségben, FAQ-ban, levelezőlistákban és newsgroupokban jelenik meg.

A Mozilla erőforrásigénye relatíve nagy, és bár hivatalos minimális hardverigénye sehol nincs megjelölve, a tapasztalatok szerint a normális használathoz egy MMX-es Pentium processzor és 64 MB RAM ajánlott.

A Mozillát olyan, saját licenctípus alatt terjesztik, amely a szakértők szerint ideális a piaci fejlesztők számára, de mindezek mellett magában foglalja a szabad szoftver definícióhoz megfogalmazott alapvető szabadságjogokat is. Az EU – European Environmental Agency, amely már egyértelműen letette a voksát a szabad szoftverek mellett, minden szervezeti szoftverfejlesztést a Mozilla licencének megfelelően fogad csak el.

Galeon

A már elemzett, sokak által fejlesztett, egyik legnépszerűbb munkafelület, a GNOME böngészője. A Galeon ugyancsak a Mozilla motort használja, és bár sebességtesztek szerint jobban optimalizált, mint elődje, még mindig hardverigényes. Követelményei körülbelül megfelelnek a GNOME munkafelülethez szükséges paramétereknek.

A Galeon jelenleg az 1.2 verziónál tart.

A Galeon saját weboldalán (<http://galeon.sourceforge.net/>) nem található sem telepítési útmutató, sem felhasználói dokumentum, de a GNOME szokásainak megfelelően a programban jó minőségű súgó van. A GNOME magyarországi fejlesztői gőzerővel dolgoznak a felület magyarításán, beleértve a Galeon honosítását is.

A Konqueror a KDE böngészője

A Konqueror (magyarul hódító) nevét valószínűleg a konkurens Navigator és Explorer termékekkel szembeni erőfitogtatás miatt kapta.

Mint már említettük, a KDE hardverigénye elég magas, nincs ez másként a Konquerorral sem. Legalább 64 MB RAM-mal rendelkező, Pentium II-es (erősebb K6-2 vagy Celeron) kiépítésű gép alatt nem érdemes próbálkozni használatával. Hivatalos honlapján található egy kézikönyvet, amely tartalmazza a szükséges

információkat. A fejlesztők számára adott egy levelezőlista és egy FAQ. A KDE és egyben a Konqueror honosítását párhuzamosan végzi a magyar KDE csoport, valamint a SuSE Linux hivatalos képviselője.

Netscape

A Netscape a böngészők piacán már veteránnak számít. Bár a Mozilla projekt a Netscape által felszabadított forráskódokból indult, a Netscape sem 4-es, sem 7-es változata nem számít szabad szoftvernek.

A 4-es Netscape-nek volt honosított változata, de használatát biztonsági okokból nem javasoljuk. A 4-es sorozat támogatása folyamatosan csökken, a hivatalos honlapon hosszas keresés után kizárólag egy FAQ-ra leltünk. A cég szemmel láthatóan a 7-es sorozatra koncentrál, és feltételezhetően csak a kisebb hardverigénye miatt tartja meg a régi változatot (Pentium I, 32 MB RAM). **Hátránya, hogy jogi nehézségek miatt a Netscape 7.01 magyarítása egyelőre nem készülhet el.**

A 7-es Netscape a 4-esnél jobban támogatott, a FAQ-n felül egy terjedelmes súgót is találunk a <http://www.netscape.com/> honlapon, viszont honosításáról semmilyen információt sem sikerült szereznünk. A 7-es sorozat hardverigénye hivatalosan egy 233MHz-es MMX Pentium és legalább 64MB RAM.

Opera

Az ugyancsak nem szabad, de a Netscape-hez hasonlóan ingyenesen letölthető Opera 6.11-es változatánál jár (bár ez elég meglepő, hiszen első verziója az 5-ös volt). A kereskedelmi termék, mint már említettük, mindössze abban tér el az ingyenesétől, hogy abban nincs benne az állandó reklámcsík. Az Operának nincs linuxos honosított változata (a Windows alatti verzió már „tud” magyarul). Az Operához kereskedelmi termékhez méltó telepítési útmutató, felhasználói dokumentáció, on-line help, valamint levelezési lista található a <http://www.opera.com-on>.

Az Opera legnagyobb előnye a minimális hardverigény, amely még a 4-es Netscape-énél is kedvezőbb.

Konklúzió:

Mint láthatjuk a néhány évvel ezelőtti állapot, amikor csak 2 böngésző (MS Explorer, Netscape) versenyzett a felhasználók kegyeiért, gyökeresen megváltozott. A jelenlegi eszközök mindegyike jól használhatónak és stabilnak mondható, a választást talán az egyéni igények illetve, a majdani honosítási állapot befolyásolhatja igazán.

11.3. Kiszolgálók

11.3.1. Levelezés – listaszerverek

Az interneten küldött e-mail a felmérések szerint 99%-os valószínűséggel áthalad egy vagy több Linux-alapú levelezőszerveren. Ennek oka természetesen az, hogy a levelezőszerverek 80-90 %-a biztosan valamilyen Linux-alapú GPL vagy kereskedelmi megoldás. Népszerűségük nagyrészt a teljesítmény és ár viszonyának, a jó dokumentációnak, a könnyű és távoli adminisztráció lehetőségének és a magas fokú biztonságának köszönhető. Körülbelül 50 fajta GNU és kommersz levelezőszerver-típus létezik, amely Linux-kompatibilis. A levelezőszerver fogalma összetett, és nem csak az MTA (Mail Transfer Agent) modult jelenti.

A levelezőszervereknek a következő feladatokat kell maradéktalanul ellátniuk:

- levél fogadása/továbbítása a lehető legrövidebb időn belül (lokális és távoli)
- levelek ideiglenes tárolása a cím elérhetetlensége esetén
- csoportos címek és listák kezelése
- képesség, hogy a kimenő és bejövő leveleket vírusellenőrzőnek adja át
- fejlődésűrések végrehajtásai a különböző mintaillesztések alapján
- levelek lokális és távoli elérése különböző felületekről
- statisztika készítése a terhelés és az átmenő forgalom alapján
- bármilyen nem alapfunkció könnyű megvalósíthatósága
- biztonság

Ezen felül legyen olyan rugalmas, hogy a felsoroltakat minden rendszergazda saját igényei szerint kiegészíthesse, tehát olyan applikációt használjunk, amely sok szolgáltatást tartalmaz, és könnyen bővíthető akár saját fejlesztésű programokkal is. Nézzünk a fentiek alapján néhány példát, a rendelkezésünkre álló eszközökből:

Kereskedelmi programok: Post Office, Lotus Domino stb.

GNU vagy BSD termékek: Postfix, Sendmail, Exim, Qmail stb.

Néhány évvel ezelőtt még a kereskedelmi termékek messzemenően nagyobb funkcionalitással rendelkeztek, mint a nyílt forrásúak, de ez napjainkra megváltozott. Ennek kitűnő példája a **postfix**.

A **postfixet** 1998 -ban Dr. W.Z. Venema kezdte el fejleszteni azzal a céllal, hogy az akkor egyeduralkodó sendmail (szintén nyílt forrású) levelezőszervernek konkurenciája legyen. A projektet az IBM rögtön felkarolta és finanszírozta. Első lépésben csak biztonságosabb lett a postfix, de nem tudott többet, mint a sendmail, viszont a kezdetektől sokkal kézenfekvőbb volt a karbantartása, konfigurálása és jobban skálázódott. A szakemberek szerint a postfix a jelenlegi nyílt forrású MTA eszközök közül a legbiztonságosabb. Ezt bizonyítja, hogy számos tűzfalgyártó is integrálta termékébe. Számos illesztési felületet támogat: vírusirtók, logelemzők, szűrők és konfiguráló programok csatolhatók hozzá, ezzel is segítve rendszergazdák

munkáját. A postfix felületét szinte minden neves vírusirtó és tartalomszűrő szoftver támogatja. Számtalan díjat és elismerést kapott, melyeket büszkén hirdetnek is a <http://www.postfix.org/press.html> oldalon.

Szinte minden terjesztésben megtalálható. Alkalmazhatunk hozzá parancssori felületet, de webmin-modullal is adminisztrálható. Széles körű dokumentációt a projekt hivatalos weblapja tartalmaz. A programnak több magyar nyelvű dokumentációja is létezik, illetőleg folyamatosan cikkek jelennek meg róla a magyar szaksajtóban. A licence BSD-alapú IBM licenc. Fejlődésének pontos leírása megtalálható a <ftp://ftp.porcupine.org/mirrors/postfix-release/official/postfix-1.1.6.HISTORY> oldalon.

A másik, igen nagy felhasználói táborral rendelkező levelező szerver a **sendmail**. Népszerűségének az utóbbi időben tapasztalható csökkenéséhez nagyban hozzájárult a postfix megjelenése is, valamint az, hogy ez a szoftver messzemenően rugalmasan, de viszonylag bonyolultan konfigurálható (az alapesetekre létezik webmin-modul). A programhoz írt illesztések számában felveszi a versenyt a postfix-el. Néhány éve szinte minden olyan helyen sendmailt használtak, ahol tetemes felhasználószámra vagy e-mail forgalomra lehetett számítani (pl. internet-szolgáltatók). A dokumentáció túlnyomó többsége csak angol nyelven elérhető, ám rengeteg magyar cég ad rá hivatalos támogatást, és több magyar kiadó is jelentetett meg könyvet a témában. A sendmailről bővebb információ <http://www.sendmail.org> lapon található.

A levelezőszerverek megépítése után szinte azonnal megjelent a levelezőlisták üzemeltetésének igénye. Eleinte egyszerű „alias” technológiával éltek, azaz egy virtuális cím mögé felírták a tagok e-mail címét. Azonban több olyan körülmény is felmerült, amely szükségessé tette, hogy listaszervereket használjunk:

- a le- és feliratkozókat egy adminisztrátornak kell kezelnie (nagy tagság esetén kényelmetlen)
- nem lehet egyszerűen megoldani, hogy ne minden beküldött levél menjen ki automatikusan (moderált lista igénye)
- aliasra nem lehet anonim írni
- nem volt egységes archívum, ahol minden átment levelet vissza lehetett keresni

Az igények növekedése megkövetelte olyan listaszerverek megalkotását, amelyek előre beállított feltételek szerint gyakorlatilag mesterséges intelligenciaként viselkednek. Így egy jól beállított, 40.000 fővel működő lista is tud komolyabb emberi beavatkozás nélkül önállóan üzemelni.

Az első listaszoftverek közé tartozott a **Majordomo** nevű program, amely a nehezen konfigurálható felületével a kőkemény UNIX-irányvonalat követte. Azonban a szakma hamarosan felébredt, és megjelent az első webről konfigurálható

szoftver, a **Mailman**, amelynek fejlesztésében számos magyar szakember is részt vett. A mai napig az egyik leghasznosabb és legkönnyebben beállítható listaszerver, pontos magyar dokumentációval és GPL-licenccel. Mivel a mailmant python programozási nyelven írták meg, könnyen továbbfejleszhető. A felhasználói felület is magyarított, és sok helyen így is használják. Az utolsó stabil verzió a 2.1-es, amelyről információ a <http://www.list.org/> címen érhető el.

Konklúzió: Nyílt forrású levelező- és listaszerver-szoftverekből nagy választékot találunk, melyek között megtalálhatjuk az igényeinknek megfelelőit. A kereskedelmi termékek és a szabadszoftver-megoldások nyújtotta funkcionalitás szinte azonos. A postfixhez megírt kiegészítő eszközökkel licenccíjmentesen tudjuk megvalósítani azokat az „újdonságokat”, melyekért kereskedelmi termék esetén 1-2 dollárt fizetünk mailboxonként.

11.3.2. Fájlkiszolgálás, PDC

A mai irodák, munkahelyek többségében egynél több számítógép található, nagy százalékban hálózatba kötve. Ilyen esetekben természetes igény, hogy az egyik számítógépen tárolt állományhoz egy másik számítógépről is – természetesen a jogosultságok figyelembevételével – hozzá lehessen férni, lehetőleg a hálózaton keresztül. Fejlett irodai környezetben gyakran nincsenek felhasználóhoz kötve a számítógépek. Azonosítójának és jelszavának beírásával bárki bármelyik munkaállomásról képes elérni a hálózaton keresztül – jól megszokott – munkakörnyezetét, saját és hálózati beállításait. Ebben a hálózattal megosztott környezetben rengeteg dolgot kell biztosítanunk a helyes működéshez: például módosítás alatt lévő állományokhoz való hozzáférés letiltását (locking), mert ez adatvesztést okozhat. Felügyelnünk kell a nyomtatók megosztásakor a hálózat (a munkaállomások) és a nyomtatót kezelő program közti kapcsolatot, sorrendbe rendezni a kéréseket („nyomtatási sor”), illetve gyorstárazni az egy időben felmerülő feladatokat. Ilyen és ehhez hasonló feladatok elvégzésére szolgál az állománykiszolgáló, illetve a PDC (Primary Domain Controller – elsődleges domainvezérlő).

Linux alatt gyakorlatilag az összes elterjedtebb állománykiszolgálási protokoll kezelhető, mindegyiknek megvan a maga szerverprogramja: Apple környezetben (AppleTalk protokoll) a Netatalk, Microsoft környezetben (SMB, más néven NetBIOS, NetBEUI protokoll) a SaMBa, Novell környezetben az IPX, illetve tiszta UNIX-os környezetben az NFS (Network File System).

SaMBa

A SaMBa (<http://www.samba.org/>) létét egy ausztrál szakembernek, bizonyos Andy Tridgellnek köszönhetjük. Ő volt az, aki 1991-ben, a canberrai Ausztrál Állami Egyetem informatikai laboratóriumának PhD-hallgatójaként először szembesült az MS-DOS és a DEC Pathworks protokolljának összekapcsolásával,

illetve az ebből származó problémákkal. Amikor megírta első SMB-kliensét, még nem is tudta, mi az a NetBIOS. Az első, 0.1 verziót 1992-ben adta ki, ezt követte kb. egy hónapon belül a „Server 0.5” és a „Server 1.0”. Itt Tridgell részéről megállt a fejlesztés, mivel nem volt rá többé szüksége. A szoftvert magára hagyta egy ftp-szerveren a következő üzenettel: „Aki úgy érzi, vegye át a projektet”. 1992 novemberében, mikor megismerkedett a Linux operációs rendszerrel, újra előkerült a probléma, mivel Tridgell otthoni hálózatán egy Windowst futtató gépet kellett a linuxos géppel összekötni. 1993. december 1-jén Tridgell bejelentette a „NetBIOS for UNIX” projektet, bizonyítékául annak, hogy többen érdeklődtek a program iránt. 1995 májusában körülbelül 1400 tagja volt a levelezőlistának, és két hónapon belül 5000 letöltést számolt az ftp-szerver. A GPL-licencű SaMBa igazi nyílt forrású sikertörténet, és népszerűségének növekedése ma is töretlen. Jelenleg a **2.2.7a** verziónál tart, és bár teljesen stabilnak és kiforrottnak mondható, természetesen folyamatosan fejlesztik. A fejlesztők kemény magja 20 emberből áll, akik a világ minden tájáról folyamatosan adják tudásukat, fejlesztéseiket a projekthez.

Mivel a SaMBa gyakorlatilag egyeduralkodó a linuxos állománykiszolgálás terén, rendkívül alaposan dokumentált. Hivatalos dokumentációkon (gyűjteményük elérhető: <http://us1.samba.org/samba/docs/>) kívül a Linux Documentation Project is rendelkezik SaMBa HOGYAN-nal (www.linuxdoc.org/HOWTO/SMB-HOWTO.html), amely magyar nyelven is elérhető például a <http://legjobb.kom.org/linux/> címen.

Ugyancsak könnyű támogatást szerezni hozzá levelezőlistákon (melyek hivatalos gyűjteménye elérhető a <http://us1.samba.org/samba/archives.html> címen), internetes fórumokon, Usenet hírcsoportokon (pl. comp.protocols.smb). A SaMBa-hoz kivételesen sok hivatalos (cégszintű) támogatás létezik, erről a projekt weboldalán lehet bővebb, a pillanatnyi helyzetnek legnagyobb eséllyel megfelelő információt szerezni.

A SaMBa-ra épülő kereskedelmi termékek száma sem elhanyagolható. A gyártók között olyan nagy nevek is szerepelnek, mint a Hewlett-Packard, az SGI (ex-Silicon Graphics), a Cobalt Networks, a SCO (Santa Cruz Operation – a SCO UNIX gyártója), valamint a VMware. A termékskála pedig a különböző, nem Linux-alapú operációs rendszerekre való átültetésektől kezdve, a teljes megoldást nyújtó állománykiszolgáló és egyéb kombinált funkciókkal bíró irodai szervermegoldásokig terjednek. A teljes lista a <http://us1.samba.org/samba/vendors/> oldalon tekinthető meg. Sajnos ezeknek a cégeknek nagy része Magyarországon nincs jelen, ezért a listán látható meggyőző bekezdések többnyire csak a program és a projekt életképességét és megbízhatóságát bizonyítják.

A hivatalos honlap tanúsága szerint üzleti célú terméktámogatás Magyarországon is elérhető.

A SaMBA népszerűségének alapja az, hogy az eredeti SMB protokolltól kicsit ugyan eltérve, de megpróbálták elérni a legnagyobb átjárhatóságot a Microsoft megoldásaival, mivel ez utóbbiak piaci dominanciája meghatározó.

Mint látjuk, az állományok és meghajtók megosztása csak egy része a SaMBA által nyújtott funkcióknak. Az operációs rendszeren és a kiszolgálószoftverben is állítható felhasználói jogosultságokkal rugalmasan hozhatunk létre egyéni, csoportos és „vendég”-hozzáféréseket.

A szoftver által megvalósított további funkció a nyomtatók megosztása. Ez azt jelenti, hogy egyrészt lehetőség van a fizikailag is a kiszolgálóra kötött nyomtatók elérhetővé tételére, másrészt SMB-n keresztül távoli nyomtatók is (akár Windows alatt megosztott, akár a linuxos távoli nyomtatók) elérhetők.

A SaMBA a főbb PDC-funkciókat (elsődleges tartományvezérlő) is ellátja: teljes értékű Domain Master, WINS-szerver. Megadható a linuxos rendszertől különböző felhasználói adatbázis, illetve beállítható a szerveren ún. „domain logons” is. Ebben az esetben ha egy felhasználó feljelentkezik a hálózatra, a SaMBA kiszolgáló automatikusan szolgáltat neki egy kis parancsfájlt, amit a felhasználó gépe lefuttat. Így belépéskor hálózati meghajtók csatlakozhatnak, az idő szinkronizálható, és gyakorlatilag bármit lehet futtatni a munkaállomáson. Természetesen ezt is csoport vagy felhasználó szerint szabályozhatjuk.

Windowsos felhasználó számára transzparens ez a megoldás, az ő szemszögéből az állománykiszolgáló ugyanúgy jelenik meg, mint bármelyik más munkaállomás vagy szerver a hálózaton. Felhasználói oldalról nézve tehát semmiféle fennakadást nem okoz a program használata, sőt inkább elősegíti a produktivitást, mivel működési stabilitása jobb, és rugalmasabban szabályozhatók a jogosultságok is.

A SaMBA természetesen minden műveletet naplóz, így könnyebb az esetleges beállítási hibákat megtalálni vagy az eseményeket visszakeresni.

A programcsomag nagy előnye, hogy Linux alá nem csak a kiszolgáló modul létezik, hanem kliensprogram is elérhető: mind parancssorból, mind grafikus felületen tudunk keresni a windowsos hálózaton, tudunk nyomtatókat és megosztásokat csatlakoztatni rendszerünkhöz, sőt tudunk WinPopUp által olvasható üzeneteket küldeni és fogadni.

A SaMBA beállítása, linuxos szoftverről lévén szó, szöveges beállítófájlok szerkesztésével történhet. A konfigurációs állományok felépítése nagyon egyszerű, átlátható, maga a beállítófájl is külön kézikönyvvel (HOGYAN-nal) rendelkezik.

A felhasználói oldalhoz hasonlóan ma már számos grafikus adminisztrátori felület is rendelkezésre áll. KDE alá elérhető a KSamba (<http://www.kneschke.de/projekte/ksamba>), mely varázslók segítségével könnyíti meg a megosztások létrehozását. Érdemes még megemlíteni a HP által fejlesztett, szabadon elérhető B+BSamba Admin Tool programot (elérhető a http://jazz.external.hp.com/src/samba/bbsat_index.html címen). Ezen kívül létezik

még webes konfiguráló felület is, az egyik a sok célra használatos Webmin (www.webmin.com/wemin), amelynek létezik SaMBa-konfiguráló modulja, a másik pedig a SWAT (Samba Web Administration Tool).

További érdekes és hasznos tartozék a perl-cgi-alapú SMB2WWW (<http://www.samba.org/samba/smb2www/smb2www.html>), amellyel a SaMBa által megosztott állományainkat html-formátumba konvertálhatjuk, így webböngésző segítségével tállózhatunk köztük. Ezekon kívül létezik még JAVA-alapú beállító felület (<http://www.eatonweb.com/samba/>), Emacs-mód (<http://savage.iut-blagnac.fr/projets/developpement/gsm/>), GTK-alapú adminisztrációs felület (<http://savage.iut-blagnac.fr/projets/developpement/gsm/>), X-alapú felület (<http://www.public.iastate.edu/~chadspen/>), de akár TCL-alapú böngésző is (<http://www.highlitemedia.de/liveserver/>).

Konklúzió:

Megállapítható tehát, hogy a SaMBa felülről kompatibilis a Windows-alapú eszközökkel, ennek következtében jól illeszkedik a hibrid irodai környezetbe. Számos jó referenciája és teljes dokumentáltsága, valamint a linuxos rendszerek stabilitásából, biztonságából fakadó előny az, amely a kereskedelmi termékek méltó ellenfelévé emeli.

11.3.3. Webkiszolgálók

Az internet legnépszerűbb felületének nevezhetjük a World Wide Web-et, melynek „motorja” a webszerver.

Technikai oldalról nézve a webszerver mint kiszolgáló igen egyszerű feladatot lát el: a kliensek (böngészők, pl. Netscape Navigator) felől érkező kéréseket feldolgozza, és a kérésben szereplő állományokat elérhetővé teszi („visszaküldi”) az azt igénylő kliensnek megjelenítésre.

Napjainkban az e-commerce előretörésével nem kell bizonygatnunk ezen eszköz jelentőségét. Az üzletfelekkel történő kapcsolattartás egyik legfontosabb eleme az internetes honlap, mivel a befektetett összeg és a hatékonyság aránya kimagaslóan jó: gyakorlatilag korlátlan mennyiségű információ bocsátható rendelkezésre igen olcsón, nagyon hatékonyan (célozhatóan, testre szabhatóan) és nagyon gyorsan. Ezért is természetes elvárás a honlappal szemben a korszerűség, állandó elérhetőség és nem utolsó sorban a biztonság. Ezen okok miatt fontos, hogy a kiszolgálón megbízható és erőteljes webszerver-szoftver fusson.

Természetesen a statikus honlapok kiszolgálásán túl egy webszerver már szerveroldali alkalmazásokat (cgi, php, asp, stb.) is futtat és biztonságos kapcsolaton (SSL) kommunikál.

Apache

A Netcraft angol internet konzultációs cég felmérése szerint (<http://www.netcraft.com/>) a világ webkiszolgálóinak piaca az alábbiak szerint

oszlík meg.

Az Apache nyílt forrású termék a teljes piac **63%-át** uralja. A második legnagyobb résztvevő a Microsoft IIS, valamivel több, mint 25%-ot tudhat magáénak. A többiek közül csak a Netscape termékek és ezek származékai, valamint az NCSA webservere szerepel számszerűsíthető eredménnyel.

Az Apache fő platformja a Linux. Erre kezdték fejleszteni, és bár ma már Windows-platformra is áthelyezték (és természetesen mint Linux-szoftver, Unixokon is fut), webservert szolgáltatást ma Linuxon a legcélszerűbb nyújtani.

Az Apache webservert története egészen 1993-94-ig, az internet hőskoráig nyúlik vissza. Akkoriban az internet legtöbb oldalát egy nyílt forrású webservert, az amerikai University of Illinois National Centre for Supercomputing Applications (az Illinoisi Egyetem Nemzeti Szuperszámítóközpontja) falai között fejlesztett HTTP daemon (HTTPd) szolgálta ki. A vezető fejlesztő Rob McCool volt, ám a projekt nyílt volta miatt sokan adták tudásukat a programhoz. 1994 közepén azonban Rob McCool elhagyta az NCSA-t, a HTTPd fejlesztése lelassult, mert nem volt olyan vezető egyéniség, aki a munkát kézben tartotta, irányította volna.

Brian Behlendorf, a HotWired internetes magazin munkatársa és Cliff Skolnick, a HotWired által biztosított sávszélességet és McCool 486-os számítógépét használva létrehozta egy levelezőlistát a kallódó HTTPd-fejlesztők összefogására. 1995 februárjára összeállt az Apache Group nyolcfős magja. Az első hivatalos Apache változat, amely az NCSA HTTPd 1.3-ra épült, 1995 áprilisban jelent meg, 0.6.2 verziószámmal. Az indulás után egy évvel az Apache átvette a vezetést az NCSA HTTPd-jétől a webserverek toplistáján (a HTTPd fejlesztését időközben újraindították, majd ismét – végleg – leállították, és a felhasználóknak az Apache-ra való áttérést ajánlották). 1995. december 1-jén hosszas tesztelés eredményeként kiadták az 1.0 verziószámú Apache web szervert. 1999-ben az Apache Group tagjai létrehozták az Apache Software Foundation-t (Apache Szoftver Alapítvány), hogy hivatalos szervezeti és anyagi háttérrel biztosítsanak az Apache projektnek, így biztosítva annak jövőjét. Bár a szűk körű „kemény mag” még mindig létezik, jelenleg a világon több ezer fejlesztő segíti a szoftvert.

Az Apache – a kereskedelmi termékeket is beleértve – a rendelkezésre álló legsokoldalúbb webservert. **Jelenleg a 2.0.43-as verziónál, amely hosszas bétatesztelés után a közelmúltban megjelent 2.0-s széria legfrissebb tagja.**

A szoftver erősségeit tételesen felsorolni szinte képtelenség, ezért a teljesség igénye nélkül vagyunk kénytelenek néhány erényét kiemelni:

- Kompatibilitás: az Apache HTTP/1.1-kompatibilis, továbbá kezeli az interneten fellelhető szabványok és „kvázi-szabványok” túlnyomó többségét.
- Modularitás: az Apache moduláris felépítése lehetővé teszi, hogy bármilyen új funkcionalitást egy újabb modul betöltésével valósítsunk meg. Az alapkiépítésbe beletartozó modulok mellett egy sor külső modul is rendelkezésre áll, illetve (mivel igen jól dokumentált) magunk is írhatunk hozzá.

- Átjárhatóság a platformok között: az Apache több platformon is rendelkezésre áll: a UNIX-kompatibilitás megoldott, és ma már létezik Windows 98/NT/2000 változata is (bár a fejlesztők figyelmeztetése szerint nem olyan stabil, mint a linuxos változat).
- Rendkívüli megbízhatóság, biztonság: a nyílt forrású modellből adódóan az Apache hibái igen hamar kiderülnek, és nagyon hamar ki is javítják őket. Emellett felépítése, szerkezete folytán is magas fokú biztonságot nyújt a betörési kísérletekkel szemben. A világ legnagyobb terhelést kiálló webservert-farmjai szinte kivétel nélkül Apache-alapúak (pl. www.google.com).
- Ingyenes: szintén a nyílt forrás és a GPL-licenc következménye, hogy ár/teljesítmény aránya felülmúlhatatlan.
- Terméktámogatás: népszerűségéből következően kevésbé valószínű, hogy olyan probléma merülhet fel, amelyet még nem tárgyaltak ki a levelező listákon.
- Cégszerű támogatás: számtalan külföldi és magyar cég termékpalletáján szerepel.
- Dokumentáltság: az Apache igen jól használható és minden részletre kiterjedt dokumentációval rendelkezik, ahogy az el is várható egy hat éve fejlesztett szoftvertől. **Az Apache Software Foundation internetes oldalán (www.apache.org) a stabil 2.0-s széria, de a korábban nagyon elterjedt 1.3-s széria teljes dokumentációja is elérhető.** A weboldal ezen kívül egy FAQ segítségével teszi könnyebbé a webservert telepítését, fenntartását. A weboldalon kívül a „hivatalos” Linux HOWTO-gyűjteményben (Linux HOGYAN-ok), a Linux Documentation projekt weboldalán (www.linuxdoc.org) is elérhetők kifejezetten a szoftverrel kapcsolatos HOGYAN-ok.

Az Apache konfigurálása szöveges konfigurációs állományok szerkesztésével történhet. Mivel a program teljesen dokumentálva van, sőt magában a konfigurációs állományban is igen sok segítség található, a szerver beállítása tulajdonképpen nem sokkal több egyszerű „másolás-beillesztés”-művelet sorozatánál. Aki ettől idegenkedik, az hasznosnak találhatja az Apache-hoz rendelkezésre álló grafikus konfigurációs felületeket. Ezek közül van, amely kizárólag az Apache beállítására készült, ilyen például a Tcl/Tk-alapú Comanche (www.comanche.org), vagy a GTK+-alapú Mohawk (<http://everythinglinux.org/Mohawk/index.html>), de van olyan is, amelyik egy általánosabb Linux-adminisztrációs felület részeként érhető el, pl. a LinuxConf (<http://www.solucorp.qc.ca/linuxconf/>) vagy a Webmin (www.webmin.com/webmin).

Az Apache-nak (és általában a Linux-alapú webservereknek) nincsenek nagy igényei a rendszer-erőforrások szintjén. Egy kis látogatottságú honlapot akár egy régi, 486-alapú géppel is kezelhetünk; a szűk keresztmetszet nem a kiszolgáló sebessége lesz, hanem a hálózat. Természetesen nagyobb terhelés (pl. memóriaigényes adatbázis-alkalmazások) és nagyobb megbízhatóság-igények esetén célszerű gyorsabb gépre és nagyobb sávszélességre építeni.

Roxen

Bár az Apache abszolút piaci egyeduralmát nem veszélyezteti, az egyik érdekes alternatíva webservert választására a Roxen Internet Software (www.roxen.com) által fejlesztett Roxen WebServer (jelenlegi verziószáma 3.3). Bár a Roxen WebServer önmagában is megálló GNU GPL licenclésű program, a Roxen IS egy teljes körű kereskedelmi alkalmazás-szerver, a Roxen a saját Roxen Platform-jának részeként is szállítja. Sőt egyes funkciói (pl. a valósidejű naplózásos funkciók) csak így, a teljes terméket megvásárolva érhetőek el, ami bizonyos esetekben zavaró lehet.

Maga a Roxen WebServer egy meglehetősen robusztus, stabil szoftver. Az Apache-hoz hasonlóan moduláris felépítésű, sőt moduljai menet közben is elindíthatók (nem mintha ennek hiánya az Apache-nál gondot okozna). Elvileg saját modulok is készíthetők hozzá némi licenclésű fejében, mivel a használt nyelv, a Pike, szintén a Roxen IS kereskedelmi célú fejlesztése.

A szerver (és a Roxen Platform) igazi erősségének a dinamikus tartalmak kimagaslóan jó kezelése mondható. Az ingyenes webserververben alapkiszolgálásban megtalálhatók az ingyenes adatbázisokhoz (mSQL, mySQL, PostgreSQL) szükséges csatlakozófelületek. A teljes Platformban további, kereskedelmi adatbázisok kezeléséhez szükséges fejlesztőkörnyezet is rendelkezésre áll. Emellett a 2.0-s széria már széles körű WAP-támogatással büszkélkedhet, valamint erős dinamikus grafikai támogatást is magáénak mondhat.

Az ingyenesen rendelkezésre álló szervernek része továbbá egy FTP-szerver, egy Proxy-szerver, valamint egy tükörszerver (mirror) is. Képes biztonságos (126/128 bites SSL-en keresztüli) kapcsolatok kezelésére, teljesítménye összevethető az Apache-ével.

Konfigurálása webes felületen történik, az Apache-hoz képest annyi előnnyel, hogy a webes beállító felület az alaprendszer részeként áll rendelkezésre, automatikusan feltelepül, ezzel is javítva a használhatóságot.

Dokumentáltsága azonban sajnos hagy némi kívánnivalót maga után. Természetesen a kereskedelmi termék, a Roxen Platform megvásárlása esetén jár a (telefonos) terméktámogatás, azonban ez az ingyenes termék használóit nem vigasztalja (valamint az USA-n kívülieket sem): sajnos nincs saját Usenet hírcsoportja, igaz, levelezőlistákon és fórumokon segítenek a problémák megoldásában. Hátrány ugyanakkor az „ingyenes terméktámogatás” esetén, hogy sokkal kevesebb használója van, mint az Apache-nak, ezért nehezkesebb fórumot találni problémáink megoldására. Írásos segédanyag sajnos nincs túl sok hozzá, a természetszerűleg rendelkezésre álló hivatalos telepítési útmutatón kívül alig találunk nem hivatalos HOGYAN-okat, így a Roxenhez elérhető tudás a közelébe sem érhet az Apache esetében tapasztalható mennyiségnek.

Caudium

A Caudium webservert (www.caudium.net) a Roxen továbbfejlesztéseként hozták

létre (úgynevezett „fork” eredményeként). A leválás fő okai voltak, hogy a fejlesztők aggályosnak tartják a Roxen IS üzletpolitikáját, úgy látják, hogy a Roxen WebServer-t túlságosan alávetik a Roxen Platform által képviselt üzleti célnak. Hiányolták továbbá a 2.0-s szériában ígért (de nem megvalósított) kompatibilitást a korábbi verziókkal, valamint szerintük a 2.0 verzió teljesítményt tekintve is visszalépés volt a korábbiakhoz képest.

Ezért döntöttek úgy, hogy a Roxen WebServer 1.3 verzióját alapul véve belefognak saját fejlesztésükbe. Jelenleg a projekt az 1.2.16 verziónál tart, stabilnak mondható. A Caudium projekt célja a Roxen 1.3 továbbfejlesztése a teljesítmény növelésével, a visszafelé és a Roxen 2.0-val való kompatibilitás megtartásával, valamint a Roxenben mindig problémát jelentő dokumentáció-kérdés rendezése, és természetesen a biztonsági és egyéb hibák javítása és a nyílt szabványok elősegítése.

A Roxen WebServer-hez viszonyítva a Caudium igen jól dokumentált; a fejlesztői dokumentáció a hivatalos honlapon érhető el, míg kellően részletes nem hivatalos telepítési HOGYAN áll rendelkezésre az interneten, igaz, mindez csak angol nyelven. Az Apache-útmutatók mennyiségétől természetesen a Caudium is jócskán elmarad.

A szoftver konfigurálása a Roxenhez hasonlóan egy, a Caudium által generált webes felületen történik (amit a Roxen 1.3-hoz képest még csiszolgattak). Teljesítménye a Roxenhez hasonlóan (Roxen-kódból fejlesztették a sebesség szem előtt tartásával) megfelelő, összevethető az Apache teljesítményével, erőforrás-igénye is hasonló.

Konklúzió:

A fentiekben említettekén kívül számtalan webszerver-megoldás létezik, mindegyik más és más tulajdonságokban kiemelkedő. Az Apache egyeduralmával küzdő Roxen-felhasználók pontosan azt róják fel az előbbinek, amit a többség előnynek értékel: összetett, hatalmas funkcionalitással rendelkező eszköz. Ebből következik, hogy webszerverekből is az csak igényeinket pontosan felmérve kell és szabad választanunk.

11.3.4. Monitorozás

Szinte minden hálózatban vagy szerveren – legyen az nagy vagy kicsi – fontos az adminisztrátor számára, hogy rendszerének működését különböző szempontok alapján nyomon tudja követni. Fontos az is, hogy hiba, rendellenesség esetén a figyelő, monitorozó eszköz a rendszergazdának riasztást küldjön, illetve a lehető legtöbb és legváltozatosabb paraméterek ellenőrzése megfelelő alapot adjon a döntéshez.

A monitorozó programok segítségével hálózatunkat, hálózati (internetes) összeköttetéseinket, a hálózatra kötött szervereinket, routereinket, switcheinket, printereinket, klienseinket, az ezeken futó szolgáltatásokat, illetve a gép egyéb

belső paramétereit (hálózati forgalom, memóriahasználat, CPU-terhelés, lemezterület, futó processzek, belépett felhasználók stb.) tudjuk ellenőrizni. Nem elhanyagolható az sem, hogy az információk begyűjtése valamilyen biztonságos, megbízhatónak tekintett kapcsolaton keresztül történjen. A hibakeresésen és megelőzésen kívül a monitorozás további területe a statisztikakészítés lehetősége.

Az alkalmazások naplózása és monitorozása között helyezkednek el a logelemző programok. Ezek feladata, hogy a naplóállományokat előzetesen feldolgozzák, ezzel könnyítve meg az adminisztrátor munkáját, miközben azokban hibára vagy biztonsági eseményre utaló jeleket keres. A logelemző szoftverek általában tanítható módon, mintaillesztés alapján dolgoznak.

A szabad szoftver közösség által fejlesztett monitorozó programcsomagok közül ki kell emelnünk a NetSaint szoftvert (<http://www.netsaint.org>). A NetSaint lehetőséget nyújt teljes körű hálózati monitorozásra, felhasználva plug-in-eken alapuló felépítését, azaz minden ellenőrizendő paraméterhez egy plug-in tartozik. Természetesen nekünk is van lehetőségünk saját plug-in-ek fejlesztésére, ilyen módon a rendszer testre szabhatósága határtalan. A NetSaint könnyű távoli használatának érdekében rendelkezésünkre áll egy dinamikus webes felület, ahonnan minden információt – igényeink szerint rendezve – megkaphatunk. A webes felületről lehetőségünk van az egyes monitorozó szolgáltatások ki- és bekapcsolására, valamint státuszuk megtekintésére, vagy azok teljes körű konfigurálására. Természetesen a webes felülethez való hozzáférést autentikációhoz köthetjük, és titkosított, SSL-es csatornában is használhatjuk (sőt mindezek beállítása erősen javasolt). Rendelkezésünkre áll továbbá a monitorozás archívuma esetleges jelentések készítéséhez vagy egyes incidensek visszakereséséhez. A NetSaint csomag másik fontos funkciója és feladata a riasztások küldése, azaz ha valamely általunk megfigyelt paraméter értéke az előre beállítható határokon kívül esik, a program riasztást küld a rendszergazdának e-mailben, sms-ben vagy személyhívóra. A NetSaint csomag egy kiforrott, multifunkcionális, aktívan fejlesztett, jól dokumentált hálózati monitorozó eszköz, amelynek a használata főleg nagy rendszerekben javasolt, hiszen méltán felveszi a versenyt a piacon kapható kereskedelmi megoldásokkal.

Statisztikák, grafikonok készítéséhez az egyik legjobban használható eszköz az MRTG (<http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>). A projekt eredeti célja routerek átmenő forgalmának monitorozása volt, de a kezdeti feladatát már régen túlhaladta. Az MRTG-vel lehetőségünk van grafikon készítésére és azok automatikus, folyamatos frissítésére, illetve weben keresztüli publikálására. Az MRTG az SNMP (Simple Network Management Protokoll) segítségével gyűjti be a számára szükséges adatokat a megfigyelt eszközöktől. Emellett mi is definiálhatunk bemenetet az MRTG számára, biztosítva ezzel, hogy tulajdonképpen bármit ábrázolhatunk. A beépített, illetve a külső help- (segéd) programokkal lehetőségünk van routereink, egyéb aktív hálózati elemeink terheltségének, forgalmának nyomon

követésére, de felhasználhatjuk szervereink CPU-kihasználtságának megjelenítésére is. Érdemes megjegyezni, hogy a fentiekén kívül az MRTG-nek – flexibilitása és jó konfigurálhatósága révén – sok más felhasználási területe is van, mint például napelemek termelésének, a hőmérséklet, páratartalom változásának, hullámvész és apály-dagály mértékének nyomonkövetése.

Logelemzésre, logfeldolgozásra jól használható eszköz a logcheck (<http://www.psionic.com/>), amely mintaillesztésekkel készíti elő az adminisztrátor számára a naplóállományokat emberi elemzésre. A logcheck testvére az fwlogwatch (<http://cert.uni-stuttgart.de/projekts/fwlogwatch/>), amely kifejezetten a tűzfalak naplóbejegyzéseinek feldolgozására szolgál, és például képes felfedezni a portscaneket. Nem szabad elfelejtenünk azonban, hogy ezek az eszközök nem helyettesítik, csak megkönnyítik az emberi intelligenciára épülő, kegyetlen, de szükséges rendszergazdai feladatot, a logolvasást.

Konklúzió:

A hálózat- és alkalmazás-monitorozó programok használata nagyban megkönnyíti a rendszergazda életét és javíthatja eszközeink rendelkezésre állását, valamint növelheti a teljes rendszer biztonsági szintjét, a grafikonok segítségével pedig nyomon követhetjük rendszerünk erőforrás-felhasználását és tartalékait, ami nagyon fontos információ a rendszer esetleges bővítése esetén.

11.3.5. Központi napló-feldolgozás

Ha egy számítógépnek közlendője van, akkor egy egyszerű munkaállomás esetén kiírja azt a képernyőre. Ez a technika azonban a kiszolgáló célra használt, dedikált számítógépeknél nem alkalmazható, hiszen ezek előtt nem ül folyamatosan adminisztrátor.

Ilyen esetekben, amikor fontos esemény történik, az nem a képernyőre, hanem egy naplóállományba íródik, melyet az adminisztrátor rendszeresen áttekint.

A keletkezett naplókat rendszeresen elemezni kell, hogy az esetleges betörési kísérletek, illetve a számítógép meghibásodásával kapcsolatos esetek napfényre kerüljenek. További fontos feladat, hogy a naplókat archiválni kell annak érdekében, hogy a későbbiekben bármikor bizonyítékként felhasználhatóak legyenek.

Nagy számú kiszolgáló esetén a naplók rendszeres elemzése és archiválása gépenként nehezen megoldható, ezért érdemes erre egy központosított infrastruktúrát kialakítani. Ennek keretében minden naplózást támogató eszköz a helyi napló mellett minden üzenetet a keletkezésükkor továbbít a naplózási központba. Az on-line naplóátvitelnek nagy előnye, hogy kompromittáció (betörés) esetén az eredeti naplók a központban is megtalálhatóak, így azok lokális módosítása esetén is van autentikus mentésünk.

Az infrastruktúra alapvető eleme egy dedikált naplófeldolgozó gép, ami a hálózatról érkező üzeneteket tárolja és figyeli. Kritikus esemény észlelése esetén azonnal riasztást küldhet az operátornak.

A tárolt üzenetek szalagos egységre vagy más hordozható médiára történő archiválása is a központban történik.

A naplók elemzése jóval hatékonyabb, mert központosítottága folytán elemzéskor megvan a lehetőségünk események közti korreláció felfedezésére, valamint képet kaphatunk a teljes rendszer működéséről.

Az integráció eszköze a UNIX-szervereken régóta alkalmazott BSD syslog protokoll, melyet a UNIX-alapú rendszereken kívül számos hálózati eszköz támogat, a Microsoft szerverekhez készült operációs rendszerei pedig felkészíthetők a használatára.

syslog-ng

A syslog-ng 1998 eleje óta fejlesztett szoftver, melynek célja, hogy az akkor még meglehetősen nehézkes UNIX naplózó rendszert jobba tegye. Ennek érdekében támogatja a naplók TCP-n történő átvitelét, az üzenetek tartalma alapján történő szűrést, valamint a célfájlok nevében a makrókiegészítést.

Ezek a tulajdonságok, valamint az, hogy a syslog-ng ma már számos UNIX-platformot támogat, teszik ideálissá a syslog-ng-t központi naplózó kiszolgálónak. A program saját dokumentációján kívül a hálózaton számos nyelven és helyen található syslog-ng dokumentáció, valamint sikertörténet.

Támogatás tekintetében sincs problémánk, a syslog-ng magyar fejlesztésű szoftver, így a fejlesztők közvetlenül is elérhetőek.

A syslog-ng jelenlegi 1.4-es stabil verziója a harmadik stabil változat a sorban, azonban a jelenlegi 1.5-ös fejlesztői sorozat eredménye **is elérhető**.

msyslog

Az msyslog újabb keletű szoftver, felépítésében jobban hasonlít az eredeti syslog megvalósításokra, legalábbis konfigurációja tekintetében

Szintén támogatja a TCP-alapú logtovábbítást, valamint tartalom alapján történő szűrést. Egyedi lehetősége, hogy a naplók digitális aláíráson alapuló hitelesítését, illetve – hasonló módon – a titkosított naplótovábbítást is támogatja (ezeket a funkciókat a közeljövőben a syslog-ng-be is implementálni tervezik).

Az msyslog spanyol fejlesztésű szoftver, és mivel újabb, ezért kevésbé elterjedt, azonban mindenképpen ígéretes fejlesztés.

11.3.6. Backup, archiválás

Ki ne találkozott volna már tönkrement merevlemezzel? Ha megfelelően szervezzük adataink tárolását, valamint felkészülünk rá, hogy bármelyik adattároló megsérülhet, elveszhet, ellophatják, akkor nem érhet minket meglepetés. Az adatmentés tervezéséhez legalább az alábbiakat kell figyelembe kell vennünk.

- Mindig legyen mentésünk és archívumunk: a mentés önmagában kevés, hiszen nagy adatmennyiségnél 1-2 hét vagy hónap alatt újra felhasználjuk médiát, így az adataink is felülíródnak. Gyakran nem elég az, ha kizárólag a legfrissebb adataink vannak meg, ezért a napi mentésen kívül érdemes minden hónapban archiválni a teljes adatmennyiséget és azt megőrizni.
- A mentések gyakoriságát és mélységét az adatok pótolhatósága határozza meg.
- Minden mentést próbáljunk ki még üzemidőben, ezzel biztosítva, hogy véletlenül se maradjon le fontos adat a mentett struktúrából.
- Alaposan vizsgáljuk meg, milyen médiára mentünk, mivel mindnek van előnye és hátránya is. A merevlemezre való mentés a legkényelmesebb, de legkevésbé biztos megoldás (porra, mágneses mezőre érzékeny mozgó alkatrészekből áll). CD-re archiválni a lemez kapacitása miatt kényelmetlen. A legjobb megoldás, ha DAT-, vagy az újabb, gyorsabb technológiára épülő DLT-kazettára mentünk. A kazettás mentésnek nagygépes rendszerekben nagy múltja van, a kazetta megfelelő bánásmód mellett 10 évig minden gond nélkül visszaolvasható.
- Minimalizáljuk a mentéshez szükséges emberi beavatkozást: Fontos, hogy a mentés teljesen magától, katasztrófabiztosan fusson a megadott időközönként. Ha emberi beavatkozásra van szükség, azt automatikus eszközökkel ellenőrizzük (pl. a megfelelő számú kazetta került-e az egységbe).
- A mentés ne legyen fontosabb, mint a gép működése: a mentés feladatát a legkevésbé terhelt időszakban hajtsuk végre, ha a mentés valamilyen okból félbe szakad, az ne gátolja a rendszert az üzemszerű működésben.
- Mindig legyen katasztrófatervünk: a katasztrófatervnek legfontosabb eleme, hogy lépésről lépésre tartalmaznia kell a rendszer visszaállításához szükséges munkafolyamatot.

A rendszergazdák Linux alatt a mentést gyakran saját, egyszerű eszközökkel valósítják meg. Az egyik legismertebb eljárás, hogy TAR vagy CPIO programmal mentjük az adatainkat és írjuk ki szalagra. Ennek módszernek 15-20 éves nagygépes hagyománya van, és saját programok (kisebb eszközök) készítését igényli. Ebben az esetben minden apró dologra nekünk kell figyelniünk! Például: a fájlrendszer kezel-e 2 GB feletti fájl méretet? A mentő programot nekünk kell megalkotnunk és gondoskodnunk arról is, hogy a mentendő adat biztonságos csatornán eljusson arra a gépre, ahova szántuk. Ilyen jellegű mentéssel csak az próbálkozzon, akinek jelentős shellprogramozási tapasztalata van.

Egyszerűbb módszer, ha az adataink mentését olyan szoftverre bízunk, amely mindezekre figyel és az adatkommunikációt is magára vállalja. Számos olyan adatmentő szoftver létezik, amely kliens-szerver kommunikációt valósít meg, valamint ezek grafikus felületén összeállíthatjuk, hogy mit akarunk menteni, milyen

gyakorisággal, kit értesítsen stb.

Afbackup:

Egy konzolos GPL program, amely kliens-szerver architektúra-alapú mentést tesz lehetővé. Segítségével több száz gép anyagát menthetjük automatizált módon egy szerverre, valamint a központi adminisztráció is megvalósítható.

A projekt weboldala: <http://www.afbackup.com>

Amnada:

BSD-licenclésű program, amelynek koordinálását a sourceforge.net oldalon végzik, elérhetősége: <http://www.amanda.org>. Ugyan csak konzol alól elérhető program, amely szinte minden mentéshez szükséges lehetőséget tartalmaz. Jelenleg 11 aktív fejlesztője van.

Konklúzió: Mindig a cél döntse el, hogy melyik terméket választjuk az adott mentéshez. A választás előtt érdemes mindegyik ismert lehetőséget felmérni, valamint érdemes olyan mentési programot beszerezni, amely, még ha fizetős is, akkor is kínál lehetőséget az átalakításoknak, fejlesztéseknek, így nem kizárt, hogy pontos és jó mentést tudunk kialakítani a GNU elsőnek „fapadosnak” tűnő eszközeivel. Gondoljunk csak arra, hogy jelenleg egyik kereskedelmi termék sem kínál lehetőséget arra, hogy egy mobiltelefonot kötve a mentő gép soros kapujára értesítsen minket a mentés állapotáról. Erről később a mobil-lehetőségeknél még szót ejtünk.

11.3.7. Gyorsítótár

Ahogy a reális áron megvásárolható internet-hozzáférések sebessége gyorsul, egyre kevesebb szó esik a gyorsítótárakról. A gyorsítótárak – webcache, proxy – feladata a már egyszer lekért tartalom (weblap) tárolása és gyors visszaadása. Mi értelme ennek? A felhasználók böngészési szokásaira jellemző, hogy nagy százalékban ugyanazon oldalakat nézegetik, így a cache használatával jelentősen tehermentesíthetjük bérelt vonalunkat, csökkenthetjük a mért hálózati forgalmat (ami forgalomkorlátozott vonal esetén konkrét megtakarításban jelentkezik). A másik ok, ami miatt nem tudunk a proxykról lemondani, hogy a legtöbb úgynevezett transzparens technológiával dolgozik.

A gyorsítótárakkal rengeteg funkciót megvalósíthatunk:

- korlátozhatjuk vele a nem kívánt internetes címek látogatását
- korlátozhatjuk az internet-hozzáférést felhasználói autentikáció vagy IP cím alapján
- szűrhetünk vele internetes tartalmat (speciális szűrők közbeiktatásával)
- felhasználhatjuk statisztika készítésre, pl. a vállalatnál milyen tartalmat néznek inkább a dolgozók, mely vonalat kell bővítenünk

A nyílt forráskódú eszközök közül egyértelműen a legnépszerűbb és legtöbbet

használt webcache proxy szoftver a SQUID. A SQUID teljes mértékben GPL licenccelésű, fejlesztőinek száma több száz, legutóbbi stabil verziója a 2.5-ös, **de már fejlesztés alatt van a 2.6-os, sőt a 3.0-s verzió is.** Dokumentáció terén az egyik legjobban ellátott program lévén nem nagyon létezik olyan más proxyprogram Linux alá, amely ekkora felhasználói táborral rendelkezik. Igazán nagy lendületet a program fejlesztésének a SULINET programok adták, Magyarországon szinte mindegyik internetszolgáltató használja a célból, hogy a nemzetközi sávszélességét óvja. A programot könnyű beállítani és karbantartani (konfigurációs állományban), a végfelhasználói felülete 100%-ban magyar, illetve létezik hozzá webmin-modul is. A szabad szoftver közösség által nyújtott támogatás (support) igen magas színvonalú, rengeteg cikket, hogyant, esettanulmányt írtak róla. A SQUID ugyanolyan produktivitással használható egy 2-4 fős iroda hálózatán, mint egy 40.000 fős internet-szolgáltatónál. Gyakorlatilag nem létezik olyan előforduló probléma, amelyet valaki már ne oldott volna meg, és az ne lenne valamelyik levelezési listán dokumentálva. UNIX-os platformon egyetlen komoly riválisa létezett a SQUID programnak, a Netscape által gyártott proxyszerver, amely magas ára miatt ma már nem tud labdába rúgni.

Konklúzió: A gyorsítótárak esetében nem érdemes választási lehetőségeket taglalni, hiszen ténylegesen csak a SQUID program kerülhet ki győztesen, tudása, üzemeltethetősége és ár/teljesítmény aránya alapján.

11.3.8. Faxeszerver

A nagyobb irodákban a faxok nagy része fölöslegesen elvesztegetett papír: a dokumentumokat kinyomtatják, majd elfaxolják, ezután jobb esetben „faxolva” pecséttel egy iratgyűjtőbe teszik, sokszor azonban csak egyszerűen megsemmisítik.

Természetesen vannak dokumentumok, amelyek aláírást igényelnek, ám a kinyomtatott állományok jó része nem ilyen. Ilyen esetekben célszerű a faxeszerver használata kimenő faxokhoz: a fax egyszerű nyomtatóként jelenik meg a felhasználó számára, aki „kinyomtatja” rá a dokumentumot, ezzel elküldve a kiszolgálóra, amely elküldi azt. Ez egyrészt sokkal kényelmesebb (nem kell elzarándokolni a nyomtatóig, majd sorban állni a faxgépnél, nem kell ismerni a faxgép kezelésének titkait stb.), másrészt sok olyan szolgáltatás is megvalósítható, amely hagyományos fax esetében nem lehetséges. Bejövő faxok természetesen szintén kezelhetők faxeszerverrel, és ha mindez egy Linux alapú megoldással történik, akkor a stabilitás, sebesség is biztosítva van.

Linux platformon a „de facto standard” fax-szerver neve HylaFax (www.hylafax.org). Ötletgazdája és megírója Robin Schaufler, aki 1990-ben fogott hozzá a kliens-szerver architektúra megírásához; azóta több, mint száz fejlesztő adott hozzá lényegi módosításokat. A HylaFax forráskódja nyílt, a szoftver

szabadon elérhető, a név azonban 1991 óta a Silicon Graphics tulajdona, akik nem gátolják a szabad fejlesztést, terjesztést.

A HylaFax egy kiszolgálón futó szerverből és a hozzá kapcsolódó kliensekből áll. Grafikus kliensprogram gyakorlatilag minden platformra (Linux, Windows stb.) létezik. Feladata, hogy elküldje a szervernek a faxolandó dokumentumot (a hálózaton tömörítetten utazik az adat), amit a szerver a fax modemén keresztül továbbít.

Természetesen a HylaFax nem egyszerűen küld és fogad, rengeteg extra szolgáltatást nyújt. Legfőbb előnyei, szolgáltatásai:

- Kimenő és bejövő faxok naplózása, mentése. Ez egyenértékű azzal, amikor a papír alapú elküldött faxot lefűzik, azzal a különbséggel, hogy ez kereshető, valamint róla biztonsági mentés készíthető.
- Bejövő faxok automatikus nyomtatása.
- „Fax-e-mail átjáró”: a bejövő faxok e-mailben való kiküldése. Az e-mail tartalmának beállítási lehetősége bő:
 - Szimpla értesítés a fax érkezteről, valamint egy FTP-hivatkozás, ahol le lehet tölteni a faxot (célszerűen a HylaFaxot futtató számítógépre telepített FTP-szerveren). A fax .tiff képfórmátumban kerül tárolásra, melyet a legegyszerűbb grafikai program is meg tud jeleníteni.
 - Értesítés, valamint a fax csatolt fájlként való elküldése. Ez esetben .tiff-, PostScript- és PDF-formátumokba történhet a konvertálás.
- „E-mail-fax átjáró”: lehetőség van arra, hogy a felhasználók ne csak „nyomtassanak” a faxkiszolgálóra, hanem e-mailben küldjék el a faxolandó adatot.
- Több telefonvonal kezelése (természetesen több modem szükséges hozzá).
- Prioritás. A fax a kevésbé fontos küldendő dokumentumokat a telefonos csúcsidőn kívül küldi el, ezzel egyrészt kisebb telefonszámla érhető el, másrészt a fontos faxok elől nem veszi el a telefonvonalat.
- „Fax broadcasting”, vagyis a szerver a faxot több számra küldi el egyszerre. (Ennek analóg megvalósítása az, amikor a felhasználó, sorra kerülve, fél napig „ráül” a faxgépre, és súlyos látáskárosodás árán kézzel beütögeti a számokat egyenként, és újra meg újra elküldi a faxot).
- Lehetőség „faxbank” használatára.

Az alapértelmezett küldési metódus a fax „nyomtatása”. Ezt egyrészt kliensprogramokkal lehet megvalósítani (a platformok: UNIX, Windows, Mac, IBM AS/400), melyek pl. Windows alá létrehoznak egy hétköznapi nyomtatónak tűnő eszközt. Természetesen ilyen kliensprogramok ingyenesen letölthetők, akár pl. a HylaFax weboldaláról is. Lehetőség van ezen kívül „fax printer” SAMBA fájlkiszolgálón való létrehozására is, valamint léteznek platform független JAVA-alapú kliensek is.

A HylaFax konfigurálása ugyan nem nehéz feladat (telepítése során kellően

interaktív), sajnos azonban grafikus felület nem áll rendelkezésre a szerver beállításához (természetesen ez csak a szerverrészre igaz). Telepítési segédlete érthető és részletes, továbbá létezik hozzá támogatás is, mind ingyenes formában (levelezőlista), mind pedig kereskedelmi termékként (ez ügyben is célszerű a hivatalos weboldalt felkeresni).

Gépigénye minimális, az egyetlen megkötés a Class 1 modem, amelynek (lévén faxolásról van szó) nem is kell 56kbps sebességűnek lennie, egy 14kbps-os is megteszi (a fax ennél úgysem használ ki többet). Ettől eltekintve akár egy 486 processzorral bíró gép is elég a kiszolgálónak, pár száz MB-os merevlemezzel (ha helyben kívánjuk tárolni a kimenő/bejövő faxokat, az természetesen több helyet igényelhet), minimális (pl. 16 MB) memóriával.

Mindazonáltal célszerű lehet a faxkiszolgálót integrálni egy másik (pl. állomány-) kiszolgálóval, ez esetben – terheléstől függően – az erőforrásigény nagyobb lehet.

Konklúzió

A HylaFax hardverigényéből adódóan sokkal olcsóbb megoldás, mint egy hasonló kapacitású célközpont, viszont tudásában és rugalmasságában fölényesen lekörözi azokat.

11.3.9. Behívópont

A behívópontok feladata, hogy telefonvonalon keresztül kapcsolatot biztosítson akár az internettel, akár egy intézmény belső hálózatával.

Kapcsolt vonalon digitális adatátvitel kialakítására a következő lehetőségeink vannak:

- analóg kapcsolt vonal: mindkét oldalon szükséges egy analóg digitális átalakító eszköz (modem), maximális átviteli sebesség 33600 bit/sec
- analóg kapcsolt vonal, szolgáltató oldalán digitális végberendezéssel (ISDN). A kliensoldalon szükséges egy olyan modem, mely ezt az aszinkron felállást támogatja, maximális átviteli sebesség: ügyfél felé 56000 bit/sec, felfelé 33600 bit/sec
- digitális kapcsolt vonal (ISDN): mindkét oldalon szükséges egy interfészártya, mely a számítógépet az ISDN hálózathoz kapcsolja. Elérhető sebesség csatornánként 64kbit/sec, melyekből 2 összefogható.

Behívópont Linux-alapú megvalósításához szükséges szoftverek rendelkezésre állnak, azonban nagy mennyiségű vonal kiszolgálásához már különböző hardvereszközökre is szükségünk van.

Az analóg modemek soros csatolón keresztül kapcsolódnak a számítógéphez,

melyből egy PC-n 2 található. Nagy mennyiségű analóg modem használata (több, mint 16) esetén szükséges egy olyan intelligens kártya, mely a soros portokat közvetlenül a számítógép rendszerbuszára csatlakoztatja.

ISDN-behívópont alkalmazása esetén, amennyiben analóg hívásokat is ki akarunk szolgálni 56000 bit/sec sebességgel, akkor intelligens, modemet is tartalmazó adapterre van szükségünk. Ezek a kártyák általában 1-4 BRI (2x64kbit) vagy 1-2 PRI (30x64kbit) ISDN-csatlakozás kiszolgálására képesek.

mgetty

Az mgetty kimondottan modemek kezelését célul kitűző szoftver, és ennek keretében támogatja a faxok fogadását, hangok rögzítését, valamint adatkapcsolat kialakítását. Flexibilis konfigurációja miatt ideális behívópontok létrehozására. Többek között arra is képes, hogy az adatkapcsolat kezdetén a ppp által küldött szekvenciát felismerje, és ez alapján transzparensen használható szöveges bejelentkezés nélkül.

Az mgetty kiforrott szoftver, funkcióinak többségét évek óta rengeteg helyen használják. Jó dokumentáció és könnyű konfigurálhatóság jellemzi. Kereskedelmi támogatása hazánkban is könnyen megoldható.

pppd

A Linuxkernel PPP-megoldása támogatja az IP-, az Ipv6-, valamint az IPX/SPX-csomagok átvitelét. Hitelesítésként protokollszinten támogatja a PAP- és a CHAP-protokollokat, valamint az ezekhez kapcsolódó Microsoft-kiterjesztéseket is.

A pppd külső szkriptek segítségével könnyen integrálható más szoftverekkel. A kapcsolat kiépülésekor lehetőség van a sorban álló levelek kiküldésére vagy letöltésére, elindíthatjuk a betárcsázott ügyfél számára a számlázást.

Ezek a tulajdonságok lehetővé teszik, hogy a belső hálózatunkhoz kapcsolt vonali hozzáférést adjunk, de egy teljesinternet szolgáltató igényeinek is megfelel.

11.3.10. Nyomtatószerver

A számítástechnika hőskorában született nagyra törő és optimista jóslatok szerint a jövőben (azaz ma) a papír teljesen el fog tűnni, ahogy mindenki átszokik a dokumentumok számítógépen való megtekintésére, digitális aláírást használunk majd stb. Ma már látható, hogy ez az érvelés szinte minden szempontból megbukott: a hagyományos aláírás továbbra is szinte egyeduralkodó, a felhasználók nem hajlandók naphosszat a monitorral rontani a látásukat. Röviden: a világ továbbra is papíralapú maradt, a különbség, hogy sokkal gyorsabban sokkal nagyobb mennyiségű papírt vagyunk képesek elhasználni, köszönhetően az új technológiáknak. Kézenfekvő élelmes megoldás a költséghatékonyság növelésére a nyomtatók hálózatba kötése, melynek több módja lehet. Megoldható egyszerű megosztással, egy „nem megosztásra teremt” munkaállomáson. Ezzel több a baj, mint az öröm, hibás protokoll-implementációtól, lefagyó operációs rendszertől

kezdve az egyszerű felhasználói bakiig. A másik végtet a professzionális, nagyteljesítményű hálózati nyomtató beszerzése. Ez technikailag nem más, mint egy beágyazott célszámítógép, ami csak nyomtatással foglalkozik, közvetlenül a hálózatra csatolva, így kiküszöbölve az őt vezérlő számítógép hibáiból fakadó fennakadásokat, tömören végtelenül stabil. Eme megoldás hátránya igen magas költségeiben rejlik: drága a szerkezet maga, drága az alkatrész, drága a szervizelés, miközben a legtöbb irodában nem tudják kihasználni a megoldás előnyeit, egyszerűen a feladat jellegéből fakadóan („nincs rá szükség”). Az arany középút, amely ötvözi a két végtet jó tulajdonságait, a Linux alapú nyomtatószerver: teljesítménye igényhez szabható, mert gyakorlatilag bármilyen nyomtató használható hozzá, stabil, megbízható, mivel a nyomtatás nem kényszermegoldásként, hanem célszoftverrel (illetve célszoftverek együttműködéseként) van megoldva, és igen olcsó, mivel megvalósításához elég egy hétköznapi nyomtató egy hétköznapi PC-vel, amin egy nem hétköznapi, ingyenes operációs rendszer fut: a GNU/Linux.

Linux alatt egy nyomtatószerver több rétegből áll. A legalsó szint a Linux kernel, amely támogatja az úgynevezett "Status Readback" módot (IEEE1284), amikor a nyomtató nem csak fogad, de szolgáltat is adatot (pl. a festékkazetta állapotáról), továbbá a számítógép indulásakor végig tudja keresni a párhuzamos portokat és megtalálni a nyomtatót. Az USB támogatás szintén megfelelő. A kernel fölött az első „operatív” szint maga a nyomtatódémon, amelynek feladata a kapott adatot a megfelelő kapun (pl. párhuzamos vagy USB port) kiküldeni a nyomtatónak. E fölött helyezkedik el a nyomtató szűrője, amely a neki adott adatot (pl. egy dokumentum) a nyomtató által ismert formátumba alakít (ez használja a tulajdonképpeni „meghajtót”). A legfelső szinten pedig a nyomtató megosztásáért felelős program áll, amely a hálózatról érkező feladatokat a nyomtatóra küldi. Ilyen lehet pl. a Samba. Természetesen egyes szoftverek esetében néhány szint egybekapcsolódik. Linux alá több ingyenes, GPL-licenclésű megoldás is létezik nyomtatókiszolgáló létrehozására. Ezek közül érdemes jobban megismernedni az LPRng+Magicfilter összeállítással, valamint a CUPS programmal. Mindkét esetben szükség lehet a nyomtatók Windows környezetben való megosztására: erre a SaMBa tökéletesen megfelel.

LPRng+Magicfilter

Az LPRng programcsomag elődje a kaliforniai Berkeley egyetemen fejlesztett LPR csomag, mely először a (szintén a Berkeley-n fejlesztett, illetve összeállított) 3BSD operációs rendszer részeként jelent meg 1980-ban. Az LPR több parancsból áll, melyek teljesen lefedik a nyomtatás során felmerülő szükséges funkciókat: nyomtatás állományból vagy standard bemenetről, offline nyomtatás, nyomtatási sor szerkesztése, nyomtató-adminisztrálás, nyomtatás állományba, távoli nyomtatás. Az LPRng valójában nem más, mint az LPR funkcionalitását teljesen átvevő (az LPR-rel teljesen kompatibilis) programcsomag, melyet a már nem fejlesztett LPR helyett indítottak útjára. Az LPR/LPRng eredetileg hagyományos

sor- és PostScript-nyomtatókat kezel. Napjainkban egyre több nem PostScript-szabványt használó nyomtató (pl. a Hewlett-Packard cég PCL szabványú nyomtatói), melyek szükségessé teszik specializált meghajtó programok használatát. Erre hivatott a Magicfilter valamint az Ifhp (mely igen speciális esetben, Hewlett-Packard lézernyomtatók esetében lehet hasznos). Ezek segítségével a nyomtatók nagy többségéhez (illetve azok emulációihoz) található meghajtó program. Unix/Linux környezetben a hálózati nyomtatáshoz nem is kell több dolog: az LPRng tartalmaz egy nyomtató démont (LPD) is, mely külön szabványosított kapun (az 515-ös TCP kapu) figyel, fogadja a hálózatról jövő feladatokat. Windows környezetben a hálózati megosztás SaMBA segítségével történhet.

Az LPRng alapvetően konzolos program, egy szövegalapú konfigurációs állománnyal. Magicfilter-t használva azonban konfigurálása leegyszerűsödik: egy karakteres párbeszédben választhatjuk ki a beállítandó nyomtató nevét, típusát, a használandó meghajtót. Ez utóbbiról (illetve a nyomtatónk által támogatott meghajtókról) hiánytalan információkat szerezhetünk a www.linuxprinting.org weboldalon, ahol listába foglalva megtalálható szinte minden nyomtatótípus. Természetesen létezik grafikus felület is a konfiguráláshoz (pl. a GPR), amely a parancssorhoz nem szokott felhasználóknak X felület alatt is elérhetővé teszi az LPRng funkcióit: a nyomtatók beállítását, a nyomtatási sor adminisztrálását stb.

Az LPRng, illetve a csomagot képező programok meglehetősen jól dokumentáltak, a parancsok kézikönyvén (manual) kívül létezik Printing-HOWTO (Nyomtatás-HOYGAN), mely a legfőbb nyomtatásra szolgáló programhoz nyújt segítséget.

Az erőforrásigény, mint általában Linuxos szerverek esetében, csekély. Amit érdemes figyelembe vennünk a nyomtatószerver méretezésekor, az a nyomtatandó dokumentumok mennyisége. Ha nagy dokumentumokat nyomtatunk, és a nyomtató erős terhelés alatt áll (tehát esély van arra, hogy előfordulnak „burst”-ök, amikor a nyomtató lassabban nyomtat, mint ahogy az újabb feladatok érkeznek), célszerű gyors hálózati kapcsolatot (minőségi hálózati kártya stb., mivel egyszerre több kliens próbál csatlakozni hozzá), valamint viszonylag nagy gyorstárat (tehát pl. az LPRng alapesetében nagyméretű /tmp partíciót) biztosítanunk.

CUPS

Szintén GPL-licencű a CUPS (Common UNIX Printing System – www.cups.org), melynek első hivatalos változata csupán 3 éve, 1999-ben került a piacra, sokak szerint megérett arra, hogy leváltsa az LPRng-t. Jelenlegi verziószáma 1.1, de egyes disztribúciók már ezt a megoldást választják a nyomtatás kezelésére. Nagy előnye az LPRng-megoldással szemben az, hogy webes felületen konfigurálható mind a nyomtatási sor, mind maga a nyomtatófilterek állománya.

Emellett a CUPS további előnyös újonságokat mutat fel a régebbi programokkal szemben:

- Hálózati nyomtatás esetén ún. „printer browsing”, azaz nyomtatóböngészés, így elég csak a kiszolgálón beállítani a nyomtatókat (a más szerveren találhatóakat

is), a kliensek számára azonnal elérhetőek lesznek. Sok (a hálózaton elosztott) hálózati nyomtató esetén célszerű lehet a többi nyomtatót is egy szerveren keresztül használni, mert ez esetben az egyszerű megosztott nyomtatók a gazdagép kikapcsolása esetén is „virtuálisan” jelen vannak a hálózaton; a gazdagép bekapcsolásakor a nyomtatószerver érzékeli a nyomtató rendelkezésre állását, és elkezdi küldeni a feladatokat.

- Széles körű nyomtató-támogatás. A CUPS által könnyebbé válik az új nyomtatók telepítése.
- Kezdő- és záróoldal-támogatás: a CUPS képes a dokumentum elején és végén egy-egy oldalt nyomtatni, mely (beállítástól függően) pl. a nyomtatási feladatot küldő felhasználó azonosítóját, időt, stb. tartalmazza, ezáltal könnyebb azonosítani, hogy az egyes kinyomtatott dokumentumok kihez tartoznak.
- Biztonságosabb beléptetés: a CUPS támogatja az MD5 hash-alapú autentikáció használatát
- Részletesebb feladatnaplózás

Linuxos szoftverekre jellemző módon a CUPS-nak mind telepítéséhez, beállításához, mind fejlesztéséhez (új szűrők írásához) elegendő írásos segítség áll rendelkezésre, emellett természetesen levelezőlisták, hírcsoportok és üzleti terméktámogatás is. Erről kiterjedt információ a CUPS projekt terméktámogatás oldalán (www.cups.org/support.html) található.

Beállítása ugyanakkor viszonylag egyszerű, és egyrészt a beépített webes felület, másrészt több különböző GUI is segíti. Ilyen a KUPS (kups.sourceforge.net), mely a KDE grafikus felület alatt könnyíti meg a munkát, vagy az X Printing Panel (www.phy.uni-bayreuth.de/till/xpp).

11.3.11. Tűzfal

A tűzfal feladata, hogy két vagy több, különböző biztonsági besorolású hálózati szegmens, zóna kapcsolatát biztosítsa oly módon, hogy azok között csak az adott cég informatikai biztonsági szabályzatának megfelelő forgalmat engedélyezi.

Különböző biztonsági besorolású például egy cég belső hálózata (intranet) és maga az internet. Az intranet lehet önmagában homogén – azaz egy zónába tartozó –, de ugyanakkor tovább oszthatjuk fokozottan védendő zónákra is, melyeket szintén tűzfal választ el egymástól.

A zónák elválasztásának elsődleges célja a kompromittáció kockázatának csökkentése. Egy interneten lévő számítógép feltörésének esélye megegyezik az internet jelentette kockázattal, ugyanakkor ha ezt tűzfal mögé helyezzük, a kockázatot csökkentettük, hiszen a két zóna között a kapcsolat limitált.

A cél az, hogy ha egy zóna kompromittálódik, akkor annak hatása minél kisebb valószínűséggel legyen érezhető a szomszédos zónákban. Teljes átjárás esetén a hatás is teljes körű, ha a kapcsolatot korlátozzuk, a hatás is csökkenthető.

Az IBSz az engedélyezett forgalmat illetően különböző részletességgel nyilatkozhat. Minél részletesebb az IBSz, annál jobban csökkenthető a kockázat, tehát annál biztonságosabb a rendszer.

Természetesen különböző részletességű IBSz-hez más-más eszközök szükségesek. Ezek alapján különböző tűzfal-technológiák léteznek.

Az ún. csomagszűrő tűzfalak az útválasztókból alakultak ki, a csomagok fejlécét elemezve, annak tartalma alapján döntenek el egy csomag további útját. Ezek a tűzfalak így csak az egyszerűbb IBSz-t képesek kikényszeríteni.

Az alkalmazásszintű tűzfalak ezzel szemben alkalmazási protokoll szintjén (pl. HTTP vagy POP3) értelmezik az átvitt folyamatot, és magára a folyamatra is képesek az IBSz-t kikényszeríteni. Ezekkel a tűzfalakkal részletesebb, testre szabottabb folyamatokat közvetíthetünk, azaz nagyobb biztonsági szint érhető el.

Linux IPChains/IPTables

A Linux rendszermagban a 2.0-ás sorozat óta található tűzfal-támogatás. Ez a 2.0-ban és a 2.2-ben egyszerű csomagszűrő volt (ipfwadm, illetve ipchains), a 2.4 viszont egy vadonatúj, állapotartó csomagszűrőt kapott.

A Linux rendszermag számos fejlesztő erőfeszítéseinek az eredménye, az egyes alrendszereket azonban mindig egy kisebb, zárt és elhivatott csoport fejleszti és koordinálja.

A 2.0-ás kernelben lévő ipfwadm a BSD-k ipfw nevű csomagszűrőjéből származik, így már az első kiadása is tesztelt kódbázison alapult. Az ipfwadm működés szempontjából stabil, viszont az újabb megoldások megjelenése miatt fejlesztése abbamaradt.

A 2.2-es kernel az ipfwadm helyett az ipchains nevű csomagszűrőt nyújtja, mely az ipfwadm-hez képest számos új funkciót vezetett be, azonban háttérként még mindig az ipfwadm szolgált. Ma a legtöbb Linux-alapú tűzfal ipchains-t használ, bár egyre több helyen alkalmazzák a 2.4-es sorozatban használt netfilter/iptables-t.

A 2.4-es kernelsorozatban a korábbi kódot teljesen lecserélték, és egy teljesen új, flexibilis keretrendszer jött létre netfilter néven. A netfilter fejlesztése 1999 novemberében kezdődött, és bár nem volt mentes a kezdeti problémáktól, ma már egy stabil, funkciógazdag szolgáltatást nyújt. A netfilter/iptables fejlesztése az alapvető hálózati funkciók megvalósításától külön zajlik. Az elsődleges fejlesztő csapat 5 főt számlál, de a független „bedolgozók” ennél jóval többen vannak. A projekt külön érdekessége, hogy a rendszert fejlesztő magnak magyar tagja is van.

A rendszermag csomagszűrőjének igénybe vétele a leggyakoribb Linux-felhasználások közé tartozik, így számtalan tematikus honlap, levelezőlista és fórum foglalkozik a kérdéssel. A témakörben bőven található dokumentációt mind angol, mind magyar nyelven.

A kernel funkcióira számos kereskedelmi termék épül, melyet szoftverként vagy

hardverrel integrálva (appliance) lehet megvásárolni.

FWTK

Az fwtk egy betűszó, a FireWall ToolKit rövidítése. Mint az nevéből is látszik az fwtkinább egy eszközkészlet, melyből munkával és utánajárással tűzfalat lehet kialakítani. Az fwtk alkalmazás-szintű protokoll-átjárókból áll, sajnos mivel a fejlesztése már leállt, bizonyos protokollok újabb lehetőségeit nem támogatja.

A fwtk licence nem engedélyezi a kereskedelmi felhasználást, de saját felhasználásra ingyenes.

Aktuális verziója a 2.1-es, melyet 1998-ban adtak ki. Egyenes ági fejlesztése, mint már említettük, leállt, de ezekre az alapokra építve készült el a Gauntlet nevű, népszerű kereskedelmi termék.

T.REX

Ez a termék 2000-ben debütált, és megpróbált alternatívát adni a lassan kiöregedő fwtk-nak. A T.REX szintén alkalmazás-szintű tűzfal, az egyes proxykat azonban különböző programokkal valósították meg, nem pedig egységes eszként fejlesztették.

Felhasználóként a fejlesztő cég az LPL-licencet alkalmazza, amely megengedi a program korlátozás nélküli használatát, de módosítás esetén a módosításainkat csak foltként terjeszthetjük.

A T.REX-et a kezdetektől megvásárolhatjuk kereskedelmi termékként, melyhez támogatást, illetve helyszíni oktatást is kérhetünk. Természetesen létezik nyílt levelezési lista is, melyen kérdéseinket feltehetjük.

A program dokumentációja jelenleg csak angol nyelven érhető el, a program hivatalos honlapján megtalálható

Zorp

A Zorp alkalmazás-szintű proxytűzfal, melynek első változata 2000 augusztusában látott napvilágot. Fejlesztése során a kezdetektől cél volt, hogy a lehető legnagyobb biztonságot nyújtsa a belső hálózatnak. Ennek érdekében alkalmazás-szintű átjárókat és csomagszűrést egyaránt magában foglal. A Zorp érdekessége, hogy egyedi lehetőségeket nyújt az alábbi területeken:

- **flexibilitás**
A Zorp flexibilitását a beépített programozási nyelv adja. Minden döntést testre szabhatunk, az alapértelmezett működéstől eltérhetünk.
- **modularitás**
Az egyes protokoll-átjárók összeköthetők, így képes olyan összetett protokollok teljes elemzésére, mint az SSL-be csomagolt HTTP (https).
- **mély protokollelemzés**
A protokoll teljes, RFC szerinti elemzésével a protokollt megsértő támadási

kísérletek ismeretlenül megfoghatóak.

- hitelesítés
A tűzfalon való áthaladás hitelesíthető mind egyszerű jelszó, mind erősebb hitelesítő módszerek alkalmazásával.

A Zorp szabad szoftver, azaz terjesztése a GPL licenc alapján engedélyezett. A Zorp fejlesztése folyamatos, ma már a 3. stabil változatnál tart. A szoftver teljes mértékben magyar kezdeményezés, így magyar nyelvű levelezési lista, illetve magyar szakemberek által nyújtott támogatás rendelkezésre áll.

A Zorp kereskedelmi termékként is kapható, megnövelt funkcionalitással, terméktámogatással és követéssel.

11.3.12. Virtuális magánhálózatok – VPN

A virtuális magánhálózat kriptográfiai eszközökkel védi a hálózaton átvitt adatokat, ilyen módon az adatbiztonság szempontjából megbízható hálózat határait kiterjeszti.

Titkosítást akkor alkalmazhatunk, ha adatainkat olyan közvetítő közegen visszük át, melynek lehallgathatlanságában nem vagyunk biztosak. Ha két szervezet számítógépes rendszereit össze kívánjuk kötni, akkor a következő lehetőségek állnak rendelkezésünkre:

- analóg telefonvonal/ISDN
Egy modem segítségével képesek vagyunk adatokat átvinni. Hátránya, hogy lefoglal egy távbeszélő vonalat (ha analóg), illetve lassú.
- privát bérelt vonal
Kizárólag a két intézmény közti privát kommunikációra. Hátránya, hogy költséges.
- interneten keresztül
Ha mindkét intézmény rendelkezik internet-kapcsolattal, akkor az internetet mint átviteli médiumot lehet használni. Hátránya, hogy nem garantált a kapcsolat sebessége, és hogy az adatok titkosítás nélkül kerülnek harmadik fél hálózatára.

Mindhárom esetben alkalmazhatunk VPN-t, hiszen az internet mellett akár a privát bérelt vonal, akár a kapcsolt vonal lehallgatható. A virtuális magánhálózat kialakítása során számos protokollt használhatunk. A legelterjedtebb a szabvány az IPSec, mely nyílt, tehát több gyártó terméke is támogatja.

FreeSWAN

A FreeSWAN a legteljesebb IPSec megvalósítás Linux alatt. Eredetileg a 2.0-s kernelhez fejlesztették, de ma már a 2.4-es rendszermaggal is együttműködik.

A FreeSWAN támogatja mind az automatikus kulcscserét az IKE protokollal, mind a csomagok hitelesítését/titkosítását ESP vagy AH fejlécek segítségével.

Számos dokumentum, GYIK és leírás kapcsolódik FreeSWAN-hez, ami jelzi a program közösségi elfogadottságát. Ezen termék és más IPSec-megoldások közti interoperabilitás meglehetősen jó, publikált tesztek bizonyítják, hogy együttműködik a vezető termékekkel, úgymint PGPNet, Cisco, Win2000, NetScreen stb.

(<http://www.hsc.fr/ressources/ipsec/ipsec2001/>)

Pipsecd

A pipsecd egy viszonylag kisméretű IPSec-megoldás, mely nem a rendszermagba épül, hanem felhasználói programként valósítja meg az IPSec-et. Hátránya, hogy nem támogatja az automatikus kulcscserét, így menedzsmentje meglehetősen nehézkes.

vtund

A vtund szintén alkalmas VPN csatornák kialakítására, viszont ehhez nem a szabványos IPSec-et használja, hanem egy saját, jóval egyszerűbb protokollt. Ez egyben legnagyobb hátránya is, hiszen így a csatorna másik oldalán is vtund-nek kell futnia.

11.3.13. Adatbázis-szerverek

Az információs társadalom terjedésének egyik következménye, hogy informatikai rendszereinkben mind több és több, az élet különböző területeiről származó adatot tárolunk. Az azonos típusú, nagy mennyiségű adatok rendszerezett tárolásának, csoportosításának és jó feldolgozhatóságának igényei hívták életre az adatbázis-kezelő eszközöket. Az adatbázis-kezelőknek sok típusa és nagy számú implementációja létezik, melyek céljukban, képességeikben, tudásukban, sebességükben, megbízhatóságukban és persze árukban eltérőek. A fejezet további részeiben az adatbázis-kezelő fogalma alatt a napjainkban elterjedt relációs adatbázis-kezelőt értjük.

A relációs adatbázis-kezelőket az informatika minden olyan területén igénybe vesszük, ahol hatalmas adatmennyiséggel kell dolgoznunk, de legismertebb felhasználása talán az e-business rendszerek, dinamikus honlapok tartalom-menedzsmentjéhez kötődik.

Talán a legelterjedtebb nyílt forrású adatbázis-kezelő a MySQL (<http://www.mysql.org/>). A MySQL fejlesztését egy finn és két svéd fiatalember kezdte a nyolcvanas években, akik később megalapították Svédországban a MySQL AB-t, amely a szoftver szerzői jogainak jelenlegi tulajdonosa. Az eszköz kezdetekben a szabad szoftverekétől eltérő licenccel volt elérhető (a kommersz felhasználásért kellett fizetni), de 2000 júniusa óta GNU/GPL alatt terjesztik, utolsó stabil változata a **3.23.54a**.

A termék hivatalos honlapján rengeteg sikertörténetet és referenciát olvashatunk, nem beszélve a különböző megnyert díjakról.

A MySQL gyors és egyre bővülő tudású rendszer, amely, mint a neve is mutatja, az SQL lekérdező nyelvet támogatja. Az új változat ismeri a fail-safe replikációt, az online mentést, a foreign key-eket. A hozzáférés jogosultsági szintjeit felhasználónként és adatbázisonként adhatjuk meg, a kapcsolat pedig TCP/IP, illetve Unix domain socket-alapú lehet. Szerveralkalmazás-szinten az ODBC és a JDBC protokollt támogatja, illetve egy saját protokollt is. Az adatbázis távoli menedzselését a mysql parancssori kliensen keresztül tehetjük meg, de rendelkezésünkre áll webmin-modul is. Ezek mellett további alternatívát jelent a phpmyadmin webes adminisztrációs csomag. A szoftver kiváló online dokumentációval rendelkezik az adminisztrátorok és a felhasználók számára egyaránt. A MySQL adatbázis-kezelőt beágyazhatjuk saját programjainkba, mivel főbb funkciói rendelkezésre állnak library (programkönyvtár) formájában is.

A MySQL mellett a másik meghatározó nyílt forrású adatbázis-kezelő a PostgreSQL (<http://www.postgresql.org/>). A szoftver gazdászervezete a PostgreSQL, Inc., amely más cégek mellett kommersz támogatást is biztosít. A PostgreSQL fejlesztését az Ingress adatbázis-kezelőre alapozva kezdték Berkeley-ben. A MySQL-hez hasonlóan SQL-re épül, támogatja többek között a triggereket, a foreign key-eket, a joinokat, a tranzakció-kezelést, a view-akat, az SQL92/SQL99 típusait, valamint a szabadon definiált típusokat. A PostgreSQL természetesen támogatja az indexek kezelését, ehhez az R-fát, B-fát, hash-t és Gist metódusokat használhatunk. Az adatbázishoz való hozzáférést felhasználónként, csoportonként, adatbázisonként tudjuk szabályozni. A PostgreSQL képes kihasználni a szerverben a második CPU-t, elősegítve a nagy terhelésű rendszerek kiszolgálását. Az azonosítás lehet jelszó-, Kerberos- vagy host-alapú. Az adatbázishoz a MySQL-hez hasonlóan TCP/IP-n, illetve Unix domain socket-en kapcsolódhatunk (ODBC, JDBC, illetve a saját protokoll támogatása szintén adott.) Webes megjelenést segítő a PHP, a perl, a python programozási szkriptnyelvek beépítve tartalmazzák a kapcsolódáshoz szükséges elemeket. A program használatához, adminisztrálásához bő és folyamatosan karbantartott dokumentáció érhető el a program saját weboldalán. A PostgreSQL menedzselését távolról ssh-n keresztül biztonságosan megtehetjük, de lehetőségünk van webmin-modul használatára is.

Összehasonlítva a fenti két rendszert elmondhatjuk, hogy mindkettő esetében egy-egy professzionális tudású rendszerrel állunk szemben. A PostgreSQL robusztusabb és több funkciót támogat, de valamivel lassabb társa a MySQL-nek. Webes alkalmazásokhoz – sebessége miatt – általában a MySQL kerül alkalmazásra, míg a bonyolultabb, nagyobb megbízhatóságot igénylő rendszerek esetében a PostgreSQL-t használják többen.

Fontos megjegyezni azonban azt is, hogy Linux és más szabad Unix rendszerekre

számos – szinte mindegyik – kereskedelmi adatbázis-kezelő szoftver megvásárolható.

Konklúzió:

Elmondhatjuk, hogy a nyílt forrású adatbázis-kezelők felveszik a versenyt kommersz társaikkal, sőt több helyen, főleg teljesítmény/ár arányukban messze le is hagyják őket. Ezt a kihívást megérezve jelennek meg sorra nagy nevű kereskedelmi terméket gyártó cégek adatbázis-kezelőjük linuxos változatával. A megfelelő eszköz kiválasztásánál fontos szempont, hogy az igényünkhöz legjobban alkalmazkodó változatot használjuk, azaz egy kisebb feladat esetén nem feltétlenül térül meg egy nagy, komplex, – esetenként drága – implementáció használata. Adatbázis-szerver üzembe helyezésekor mindenképpen megéri megfontolni a kiváló minőségű, már bizonyított nyílt forrású eszköz választását.

11.4. Egyéb lehetőségek

11.4.1. SMS és MCommerce Linux alatt

A XXI. század egyik nagy vívmánya a mobiltelefon. A hasznosságáról a vélemények megoszlanak, de az SMS-küldést és -fogadást mindenki fontosnak és diszkrétnek ítéli meg. Mint ahogy az élet számos területén, a számítógépes rendszerekben is nagy hasznát tudjuk látni az SMS-értesítéseknek. Szinte minden rendszermérnök, menedzser, projektvezető, titkárnő és igazgató zsebében ott lapul a mobiltelefon, azonban gyakran nagyon zavaró tud lenni egy hívás, míg egy szolid SMS érkezése rendszerint nem. Ha azonban értesíteni akarunk valakit egy fontos tárgyalásról vagy egy rendszer leállításáról, akkor emberi beavatkozásra van szükségünk – nevezetesen a titkárnőre vagy az operátorra, aki felhív minket. Ezeket a problémákat lehet áthidalni az SMS-sel és az e-mail SMS-re való átirányításával. Képzeljünk el egy olyan szituációt, ahol a munkahelyi gépünk előtt ülve a naptárprogramunkba feljegyezzük a találkozóinkat és teendőket, a számítógép pedig az adott találkozók előtt rövid szöveges üzenetben értesít minket mobiltelefonunkon. Az üzenet tartalmazza a találkozó helyét, időpontját, a partner nevét és elérhetőségeit. Így akkor sem kerülhetünk zavarba, ha ebéd közben kapjuk meg a figyelmeztető SMS-t, hiszen az minden adatot tartalmaz. Ugyancsak nagyon hasznos, ha fontosabb rendszereinket nem csak operátorok felügyelik, hanem programok, amelyek SMS-ben tudják riasztani a rendszermérnököt a kellő pillanatban (mint arról a monitorozással kapcsolatos fejezetben már beszéltünk). Mikor lehet ez hasznos? Például ha a rendszermérnök éjszakára nem kapcsolja ki telefonját, hanem csak átirányítja a hangpostájára, így SMS-ben értesülhet arról, ha a gépe „rosszul érzi” magát. Az újabb telefonokon már hívócsoportok szerint lehet beállítani az SMS hangjelzését. Mi sem kézenfekvőbb, mint hogy a szerverről jövő üzenetek hangját hangosra állítjuk, minden mást pedig halkra. Nagyon gyakran, ha megfelelő időben értesülünk a bajról, sokkal jobbak az esélyeink a javításra, mint ha

másnap reggel kész tények elé állítanak minket. Ilyen eset lehet egy illetéktelen behatolás, ahol ha időben észleljük a betörés tényét, meg tudjuk menteni a még menthetőt. Nagyon fontos azonban, hogy jól állítsuk be a figyelmeztetéseket, hiszen a sok vaklárma ahhoz vezethet, hogy a tényleges problémára már nem is figyelünk, mert nem bízunk a rendszerben.

Megvalósítás: A fenti rendszereket megvalósíthatjuk akár 20 ezer forintos minimáleszköz felhasználásával, de többmillió forint beruházásával is, pl. egy Nokia 5110 vagy 6210 telefont plusz a megfelelő soros kábelt a gépünk soros kapujára kötve a gnokii nevű szoftverrel. A gnokii egy GPL-es szoftver, amelyet a 3110-es Nokia-telefon megjelenésével egy időben kezdtek fejleszteni, a Nokia mindennemű támogatása nélkül, sőt mivel a protokoll és még jó néhány szabvány titkos, ezért inkább hátráltatásával. A gnokii alapvetően egy konzolos programnak indult, ám a későbbiek során készítették hozzá X11 felületű kezelőprogramot is. Kezeli a Nokia 51xx, 61xx, 71xx, 62xx sorozatú telefonjait, és napról napra újabb modelleket. A telefonmemória lekérdezésétől kalendáriumának szerkesztéséig sok mindent tud, így SMS üzenetek feladására is lehetőség van linuxos parancssorból. Innentől már csak egy kis program kérdése, és összeköthetjük a gnokii-t a rendszert felügyelő géppel vagy a levelezőszerverrel is.

Konklúzió: miért jobb, ha saját eszközökkel valósítjuk meg az üzenetek a telefonra küldését? Miért ne használjuk a szolgáltatók által nyújtott e-mail értesítést? Szinte mindegyik operátor kínál lehetőséget arra, hogy elektronikus leveleinket átirányítsuk a mobilunkra. Ez egy figyelmeztetés, értesítés. Ugyan sokkal olcsóbb fizetni egy szolgáltatónak havi kb. 200 Ft-ot, mint egy önálló rendszert építeni, azonban a saját rendszerünk megbízhatóbb. Miért? Ha rossz pillanatot fogunk ki és a szolgáltató leterhelt, akkor akár órákat is állhat a rendszerben az üzenet, amíg átkonvertálódik SMS-sé, de lehet, hogy sosem érkezik meg, ellenőrzési és reklamálási esélyeink csekélyek. Ha mi magunk üzengetünk, mindent tudunk naplózni és utólag ellenőrizni.

Az Nokia-telefonok és a Linux kapcsolatáról egy technikai jellegű leírás olvasható a <http://www.guska.hu/publication/> oldalon.

Ha nagyüzemi megoldásokat akarunk megvalósítani, percenként többszáz, akár párhuzamosan érkező üzenettel, vásárolhatunk GSM-terminált, vagy SMS-béreltvonalat is. Egy biztos: a jelen és a jövő a szöveges üzeneteké.

11.4.2. A kriptográfia és a fájlrendszer

Miért igyekszünk a bizalmas információinkat kriptográfiai úton mások számára olvashatatlanná tenni?

„Ez az igény nem új az emberiség számára, már Hérodotosz az ie. V. századi görög-perzsa háborúról szóló leírásában megemlíti. Demaratusz a perzsa Szúzában

élt, így hamar értesült Xerxész Spárta elleni terveiről. Ezért – noha számkivetett volt – egy fatáblára véste figyelmeztető üzenetét, melyet aztán viasszal vont be. Ezt hívják szteganográfiának, ami a görög steganos („takar”) és graphein („írni”) szavakból tevődött össze. Az elektronikus világban gyakran képek, hangok bitjei közé rejtik a szöveget, ám meg kell említeni, hogy vannak módszerek ezek felismerésére. A háború nyilván egy szélsőséges példa, ám a gyakorlat azt mutatja, hogy azért mindenki borítékba teszi a levelét. Noha a védelem gyenge, de mégis megóv egy átlagos kíváncsi szemtől.

Ezzel párhuzamosan fejlődött a kriptográfia, ami a görög kryptos („rejtett”) szóból kapta a nevét. Itt már nem magának az írásnak az elrejtése volt a cél, hanem az írás értelmének, üzenetének. Az írást valamilyen visszafordítható eljárással érthetlenné tették és azt küldték el. Az eljárás hordozta magában a kulcsot is. Ennek az előnye, hogy ha az elküldött szöveg előkerül, akkor még mindig érthetetlen maradt. A kriptográfia alapvetően két módszert használ, a helyettesítést és az átrendezést, pontosabban egy adott kulcs segítségével egyértelműen helyettesíti az üzenet betűit, illetve egy előre rögzített szabály alapján az üzenet betűit egymás között cseréli fel. Hogy most se maradjon el egy érdekes történelmi példa, az ie. VI. században a Káma-szútra könyv szerint a nőknek 45. megszerzendő tudásként a „mlecchita-vikalpa”-t kellett elsajátítani, azaz a titkosírást. Egy javasolt módszer szerint az abc betűinek véletlenszerű párosítása adta meg az új abc-t.” (idézet a oldalon található írásból)

Jól látható, hogy a titkosítás igénye nem a ma emberében merült fel először. Manapság egy merevlemezen több millió oldalnyi titkos információt is tárolhatunk, amely ha illetéktelen kezekbe kerül, súlyos visszaélésekre ad lehetőséget. A probléma keletkezhet akár egy betörésből, ahol a betörő fizikailag fér hozzá a lemezhez, szereli ki azt a számítógépből, és otthon ráérősen analizálja adatainkat. Nagyobb talán a veszély a notebook-számítógépek elterjedése miatt, hiszen rengeteg ember egyszerűen csak a gépkocsijában hagyja mindenféle őrzés nélkül a hordozható számítógépét. Ha adataink megkívánják a titkos tárolást (szinte minden személyes adatunk ilyen), akkor érdemes szétnéznünk a lehetőségek terén. Linux alatt számtalan kriptográfiai modul létezik, amely nagy biztonságot nyújt adataink tárolását illetően. Mint tudjuk, teljes biztonság nem létezik, hiszen ami 10 évvel ezelőtt szuperbiztonságosnak volt mondható, az mára már belekerült az alapszintű kriptográfiai történelemkönyvekbe. Számos olyan eszköz közül, amely nagyfokú védelmet ígér, érdemes megemlíteni két teljesen GPL-alapú, különböző felhasználási területű szoftvert.

GPG: Eleinte fájlok, később e-mailek és lemezterületek titkosítására kifejlesztett titkosító program. A pgp-ből fejlődött ki egy teljesen GPL-alapú, nagy megbízhatóságú rendszer. Minden platformon támogatott, mind a windowsos, mind a linuxos levelező programokkal tud kommunikálni, bár Linux alatt sokkal

komfortosabb. További információ a <http://www.gnupg.org/> oldalon.

Loop-AS: ez egy olyan modulrendszerű kriptográfiai fájlrendszer-támogató, amely szinte minden kernelverziót támogat lefelé is, felfelé is. Viszonylag könnyen telepíthető, akár disk image-formátumú titkosítást vagy teljes partíció titkosítását is lehetővé teszi. A loop-as modul a gpg-vel együtt is munkára bírható, így garantálja a nagyfokú megbízhatóságot.

Elérhetősége: <http://loop-aes.sourceforge.net/>

Konklúzió:

Napjainkban a biztonság mindenhol az elsődleges, olyannyira, hogy még a cégvezetők is kénytelenek elfelejteni az egyszerű jelszavak használatát, ha nem akarják, hogy bizalmas információik nyilvánosságra kerüljenek. Egyre többen tárolják belépési jelszavaikat és egyéb adataikat olyan alkalmazásokban, amelyek korlátozott hozzáférést tesznek lehetővé. Ha mindezt „megspékeljük” azzal, hogy az egész rendszerünk titkosított fájlrendszeren üzemel, akkor kicsit talán nyugodtabban aludhatunk.

11.4.3. VMware

Mi az a VMware?

Nem más, mint egy lehetőség a számítógép operációs rendszerén belül egy újabb operációs rendszer futtatására. Nem ismeretlen a nagygépes rendszerekben az a szemléletmód, hogy ha egy újabb nagyobb feladatot akarunk a géppel elvégeztetni, egy újabb virtuális gépet oszt ki nekünk. Miért is jó ez nekünk? Jó, mert ha nagy probléma adódik, akkor is csak a virtuális erőforrás erejéig érhet minket gond, vagy mert egy teljesen más operációs rendszert tudunk futtatni a sajátunk alatt. Félretéve a nagygépes igényeinket, a következő esetekben lehet hasznos Linux alatt egy virtuális gép felépítése:

- ha olyan feladatot kell teljesítenünk, amely egy csak egy adott operációs rendszeren képes futni, és az nem a Linux,
- ha egy teljesen elkülönített rendszert akarunk létrehozni a sajátunkon belül,
- ha egyszerre több operációs rendszert kell egy időben használnunk.

A fejezet címe VMware. A VMware kereskedelmi termék. Ez természetesen nem véletlen, bár létezik ingyenes megoldás virtuális gép létrehozására Linux alatt, azonban mind közül a legkönnyebben használható és konfigurálható tudással a VMware rendelkezik. Profi linuxos tudással, úgynevezett „jail” technológiával Linux alá feltelepíthetünk egy másikat, vagy akár egy BSD rendszer alá is felrakhatunk egy Linuxot, azonban ezek már nem egy egyszerű felhasználó által kivitelezhető dolgok. A VMware megvásárlása után vagy a 30 napos próbaverzió letöltése után egyszerűen tudjuk telepíteni pl. Linuxunk alá a Windows különböző verzióit, vagy akár a DOS-t is.

A VMware elérhető a <http://www.vmware.com> címen.

Konklúzió: Rengeteg olyan alkalmazás létezik, amelyet nem használunk nap mint nap, azonban néha szükségünk lehet rájuk. Ilyenkor teljes gépünk átszervezése nélkül megoldható a probléma. Egy jó példa erre, hogy ezek a dokumentációk egy olyan számítógépen készülnek, amelyen csak és kizárólag GNU/Linux üzemel, ez nem is jelent mindaddig problémát, amíg az éves nyugdíjbevallási időszak meg nem érkezik. Ekkor a nyomtatványt az APEH által biztosított Windows/DOS rendszerek alatt működő programmal kell kitölteni. Ilyenkor elegendő feltelepíteni a VMware programot és alá feltelepíteni pl. a FreeDos-t. Mivel ez csak évente egy alkalommal fordul elő, ezért nem kell az egész rendszert átszerveznünk.

11.4.4. Személyi asszisztens (PDA-PIM)

PIM: Personal Information Manager

PDA: Personal Digital Assistant

Jelen fejezet önmagunk menedzseléséről szól különböző digitális személyi titkárok segítségével.

Rohanó világunkban egyre nagyobb szerepet tölt be a digitális tér, így egyre kézenfekvőbb, hogy az elavult papír és toll helyett a többször felhasználható és számtalan funkcióval rendelkező digitális asszisztenseket használjuk. Miért jó nekünk, ha egy ilyen eszközre bízunk az adatainkat?

- rendszerint zsebben elférnek
- felismerik a kézírást vagy rendelkeznek beviteli eszközzel
- riasztással és ébresztéssel rendelkeznek
- kommunikálnak a mobiltelefonnal
- internetezni lehet velük

A felsoroltakon felül még vagy több száz hasznosabbnál hasznosabb funkcióval rendelkeznek. Itt is csak a képzelet szabhat határt, hiszen szinte mindegyik programozható, így akármilyen eszköz írható, amit aztán PDA-n is használhatunk. Több mint 20 fajta, hazánkban is kapható változat létezik PDA-ból. A legelterjedtebb fajták:

- Psion <http://www.psion.hu>
- Handspring <http://www.handspring.hu>
- Palm <http://www.palm.com>
- Agenda <http://www.agendacomputing.com>
- Compaq IPAQ

Kiválasztásuk pontosan akkora feladat, mint megválasztani egy komplett kormányzati struktúra hardverét és szoftverét. Jelen tanulmány nem tekinti feladatának a Windows CE-alapú és az egyéb tenyérgépre írt operációs rendszerrel szerelt eszközök összehasonlítását, kizárólag azokra koncentrálni, amelyek Linuxot

futtatnak vagy képesek futtatni.

PDA megválasztásánál érdemes figyelembe vennünk az alábbi szempontokat:

- Billentyűzettel rendelkező vagy billentyűzet nélküli eszközt szeretnénk? Szinte mindegyik gépen lehetséges a szövegbevitel, függetlenül attól, hogy van-e hozzá billentyűzet, vagy csak virtuális billentyűzet van; ha azonban sok adatot akarunk a PDA-val bevinni, akkor érdemes olyan eszközt választanunk, amelynek kényelmes billentyűzete van.
- Az adott operációs rendszerhez kapcsolható-e az eszköz, nevezetesen tud-e kommunikálni a Linux-rendszerrel.
- Megfelelő a hardver? Elegendő memória található benne? Rendelkezik a megfelelő csatlakozási felületekkel (USB, infra- vagy soros kapu)?

Mivel a legtöbb eszköz tudása nagyon hasonló, legfeljebb a hardver összeállításában van eltérés, ezért most kiindulási alapnak vegyünk egy Handspring Visor gépet, mivel minden eszközhöz megírták a kommunikációs szoftvert Linux alá is. Rendszerint a disztribúciónk tartalmazza azt, azonban ha nem, akkor a <http://www.pdamania.hu> magyar oldalon vagy esetleg a gyártó oldalán próbálkozzunk. A Visor és Palm modellhez a Jpilot <http://www.jpilot.org> és a <http://www.pilot-link.com> programra lesz szükségünk. Ez a két program kb. a 2-3 szorosát fogja nyújtani funkcionalitásában, mint a gyári, Windowshoz adott program. A jpilot tartalmaz kalendáriumot, jegyzetet, telefonregisztert, teendő-szerkesztőt (todo list). A program segítségével szinkronizálni lehet az adatokat a pda és a számítógép között egy kényelmes X felületű alkalmazáson keresztül. A pilot-link az úgynevezett háttér munkát végzi, azaz ténylegesen ez másolja fel az alkalmazásokat a PDA-ra, majd onnan vissza, a Jpilot csak egy „frontend”. A pilot-link lehetővé teszi továbbá, hogy parancssorból szinkronizáljunk telefonszámot, találkozót stb., így könnyedén beállíthatjuk a gépünket, hogy éjfélkor, amikor senki sem használja már, automatikusan indítson el egy backup-műveletet, amelyet a PDA fogad. Ez nagyon hasznos lehet, hiszen ha rajtunk múlik a backup, mindig csak akkor gondolunk rá, amikor már késő és elvesztek az adatok, hiszen a PDA-k olyan eszközök, amelyeknek a belső memóriája RAM típusú, tehát csak addig marad meg benne az adat, amíg az elem működőképes. Hasznos funkciót tud megvalósítani a Plucker nevű alkalmazás, amely az általunk beállított weboldalak tartalmát szinkronizálja le a PDA-ra, így akár minden reggel a munkába menet a buszon a friss internetes híreket olvashatjuk (<http://www.plkr.org>). Számtalan funkciót és feladatot megvalósíthatunk PDA és a Linux összekötésével, további ötletek találhatóak a <http://www.guska.hu/publication/cikkek/> honlapon.

Konklúzió: Rengeteg alkalmazás létezik PDA és Linux összekapcsolására. Compaq IPAQ-ra például, amely eleve Windows CE rendszerrel kerül forgalomba, lelkes programozók megírtak egy külön Linux disztribúciót, hiszen ha hardveresen jó gép, akkor miért is ne lehetne Linuxszal használni? Minden esetben a beruházás

előtt érdemes tájékozódni a lehetőségekről, vagy érdemes megvárni az eleve Linuxszal működő PDA-kat.

Az egyik ilyen projekt az Agenda nevű linuxos PDA, amelyről komplett leírás található http://www.guska.hu/publication/cikkek/agenda_vegleges.pdf oldalon. A cikk leírja, hogy az Agenda ugyan még csak kezdeményezés és nem kiforrott dolog, azonban ezen már jól felmérhető, hogy a Linux – bár csak 1-2 éve indult a PDA-piacon – hamarosan behozza a vetélytársakat.

11.4.5. Beléptető rendszer

Az alábbi esetleírás inkább illene a sikertörténetek közé, de jól példázza a Linux-rendszerek rugalmasságát.

1997-98 környékén az egyik nagy internetszolgáltatónál a sorozatos lopások miatt a cégvezetés úgy döntött, beléptető rendszert szereltet fel a bejárat ajtókra. A rendszert a rendszermérnökkel történő egyeztetés nélkül meg is rendelték. Az eszközöket hamarosan felszerelték és élesítették, minden munkatárs kapott egy kártyát, amivel ki és be tudott lépni, azonban fontos volt, hogy ha kiment az ember, akkor belépésnél is használja, különben később újra már nem tudott kilépni. A leolvasó kapuk összeköttetésben álltak egy fekete doboznak tekinthető számítógéppel. A doboz egy soros kivezetésen egy másik számítógéphez csatlakozott, és ott egy DOS alatt futó egyszerű szoftverrel lehetett újabb kártyák élesítését végezni, illetőleg figyelni, hogy ki tartózkodik házon belül

Mivel a soros kábel 3 méter hosszú volt, ezért a számítógép, amelyre a fekete doboz csatlakozott, a recepciós gépe lett; így viszont a fekete doboz adminisztrálása komoly akadályokba ütközött, mert recepciós gépét másra is kellett használni. A cég kért egy árajánlatot egy olyan szoftverre, amelyet lehet hálózaton keresztül használni. Az ajánlatban szereplő ár túl magas volt, sőt hálózaton csak naplózni tudott, adminisztrálni nem lehetett. Ezek után a probléma megoldására a cég vezetése egy belső szakembert bízott meg.

A munka körülbelül 2 hét alatt fejeződött be, a szoftverrel lehetett naplózni, adminisztrálni, webes felületen lekérdezni a statisztikákat, sőt távolról vezérelni az ajtókat.

A fenti történet jó példa arra, hogy ha alaposan felmérjük a lehetőségeket, akkor szinte határtalan dolgokat tudunk megvalósítani. Akár csak a beléptető rendszereket, a telefonközpontokat is össze lehet kapcsolni a géppel, és hasonló eszközökkel olcsón lehet rá számlázó programot fejleszteni. Érdeemes szétnézni az interneten, mert például van olyan Panasonic telefonközpont, amelyhez lelkes magyarok írtak már webes felülettel rendelkező, mysql-alapú lekérdező programot.

11.4.6. HA, Load Balance

Fürtök kiépítésének alapvetően két célja lehet: egyrészt redundáns kiépítéssel növelhetjük egy szolgáltatás rendelkezésre állását, másrészt ugyanazt a feladatot

több számítógépre osztva növelhetjük a teljes rendszer teljesítményét.

A szakirodalom ennél árnyaltabban osztályozza a fürtök típusait:

- HA (magas rendelkezésre-állást biztosító) fürtök
- Fault tolerant, hibatűrő fürtök
- load balancer, azaz terhelés-megosztó fürtök
- parallel processing, azaz párhuzamos számítási kapacitást növelő fürtök

A fürt kialakításának módja gyakran függ a konkrét alkalmazástól. Nem mindegy ugyanis, hogy a fürt háttérét adó adatokat meg kell-e osztani, és ha igen, akkor azok változási sebessége sem elhanyagolható szempont.

A fürtök kialakításához mind szoftveres, mind hardveres eszközök léteznek, és leggyakrabban ezek kombinációjára van szükség (fault tolerant megoldás).

A HA megoldáshoz általában szükséges szoftverelemek:

- heartbeat (szívdobbanás), a fürttagok egymás elérhetőségét vizsgálják vele
- fájlrendszer (naplózó vagy megosztott) – a tárolt adatok funkciója és mennyisége alapján
- a rendszer működőképességének vizsgálata (system health monitoring) – a fürttag saját működőképességét vizsgálja

A HA megoldáshoz általában szükséges hardverelemek:

- network power switch – a működésképtelen fürttagot egy működő lekapcsolhatja
- megosztott lemezalrendszer – ha az adatok szinkronizálása másképp nem megoldható
- a fürt redundáns kapcsolata (soros vagy ethernet interfészek)

Linux-HA projekt

A Linux-HA projektet 1998-ban indította Alan Robertson, hogy olyan, magas rendelkezésre állású rendszer kialakításához szükséges eszközt hozzon létre, amely könnyen konfigurálható és egyszerű. A megalkotott szoftver, a **heartbeat** első verzióját 1999 márciusában tette elérhetővé, legutolsó stabil változat a **0.4.9.2**-es. A **heartbeat** HA fürtök kialakítását teszi lehetővé, és egyszerűsége folytán olyan népszerű lett, hogy szinte az összes Linux-alapú HA megvalósításban felhasználgják.

Linux Virtual Server projekt

Az LVS projekt célja, hogy olyan terhelés-megosztó (load balancer) alkalmazást hozzanak létre, amely transzparens a felhasználók számára. A legfrissebb, 2.4-es rendszermagot támogató stabil verzióját 2002 áprilisában tették elérhetővé. A hivatalos honlap minden igényt kimerítően foglalja össze a HA-technológiákat, és kellő mennyiségű dokumentációt nyújt az LVS-modulok telepítéséhez

konfigurálásához, ugyanitt az LVS-t használó cégekről is találunk listát referenciaként. Az alkalmazás konfigurációját megkönnyítő webmin-modul szintén letölthető.

<http://www.linuxvirtualserver.org/>

Red Hat Piranha

A Piranha egy több modulból felépülő, a **heartbeat-** és a **Linux Virtual Server-**technológiákon alapuló hibatűrő és terhelés-elosztó clustermegoldás, amelyet a Red Hat Linux terjesztés sajátjaként kezel. A cluster beállítható konfigurációs állományból, parancssorból, web- (php) és X11-GTK-alapú GUI-ból. Terhelés-elosztásra az egyszerű round robin- (sorban adja a kapcsolatokat a szervereknek) és a súlyozott „legkevesebb kapcsolat”-algoritmusokat használja. A szoftverhez a Red Hat alaposságára jellemző mennyiségű dokumentációt találunk a <http://www.redhat.com/support/resources/howto/piranha/> honlapon.

Kimberlite

A Mission Critical Linux (USA) által életre keltett hibatűrő HA-megoldás. A termék kezeli a közös (megosztott) háttértárat, valamint kereskedelmi változata is létezik (MCLX-Convolo). Az utóbbihoz a cég természetesen hivatalos támogatást is nyújt. A Kimberlite egy bevált és jól dokumentált szabad szoftver, amely ideális megoldás a redundancia és a rendelkezésre állás növelésére. A Kimberlite konfigurációját webes felületen és parancssorból is elvégezhetjük.

Beowulf

1994-ben a NASA védőszárnyai alatt indult egy projekt, amely többprocesszoros nagy számító kapacitású rendszer kialakítását célozta meg. Fontos követelmény volt az is, hogy ezt a lehető legolcsóbb technológiát felhasználva ériék el. Egy 1997-es bemutatón egy 140 PC-ből álló fürtön több mint 10,9 Gigaflop/s kapacitás-teljesítményt értek el. Három évvel később a SZTAKI-ban egy 28 gépből (PC-ből) álló fürt már 30 GFlop csúcsebességre volt képes.

A Beowulf-hoz hasonló megoldás a MOSIX is, de számos hasonlóról találhatunk még információt a <http://www.topclusters.org/> oldalon.

11.4.7. Network IDS

Hálózatunk, szerverünk biztonságos üzemeltetése érdekében fontos, hogy a rendszerünk elleni támadásokat figyeljük, naplózzuk, és egy esetleges támadás esetén a megfelelő ellenlépéseket megtegyük. Az IDS-ek, avagy a betörés- és behatolás-figyelő, -felderítő rendszerek feladata pontosan ez. Működésüket tekintve két nagy csoportba oszthatjuk őket: host-IDS és hálózati IDS. Az előbbi feladata, illetve hatóköre a hoston (egy számítógépen) belüli támadások felderítése, míg az utóbbi a hálózati forgalomban próbálja azokat detektálni. A támadásra utaló jelek kiszűréséhez az IDS általában egy előre elkészített adatbázist használ, amely mintákat tartalmaz az ismert támadásokról. Ilyen detektálása esetén, a beállításoktól

függően, az IDS naplózza az eseményt és az azzal kapcsolatos információkat, riasztást küld a rendszergazdának, illetve akár meg is szakíthatja a kérdéses kapcsolatot.

A szabad szoftver közösségben népszerű hálózati betörésfigyelő rendszer a Snort (<http://www.snort.org>). A Snort egy moduláris rendszerű program, azaz a rendszergazdának lehetősége van modulok használatával rugalmasan felépíteni a behatolást figyelő szabályokat, ugyanígy van lehetősége finomhangolni a támadásokra adott reakciót, illetve megadni, hogy a támadást tartalmazó csomagokból mit és hogyan naplózzon. Behatolás észlelésénél lehetőség van az adatfolyamra minta-illesztéseket rakni. Az illesztések a már ismert támadások jellemző mintáit, kódrészleteit tartalmazzák. Az adatbázisba az adminisztrátor is vehet fel újabb elemeket, de a Snort fejlesztői, levelezőlisták és publikus adatbázisok alapján folyamatosan bővítik azt. Elmondhatjuk a Snortról, hogy egy jól dokumentált, funkciógazdag programcsomag, amely folyamatos fejlesztésének köszönhetően széles körben alkalmazott megoldás. Fontos megjegyezni, hogy a Snort esetében, hasonlóan sok más szabad szoftver projekthez, kommersz támogatás is rendelkezésre áll. A Sourcefire cég egy professzionális, nagyvállalati programcsomagot fejleszt a Snortra alapozva, azt kiegészítve.

Host-IDS-ek között sok olyan program található, amely figyeli a rendszerállományok, illetve a bináris állományok megváltozását (lecserélését), és erről riasztást ad. Emellett megtalálhatók olyan kezdeményezések, amelyek a rendszerben futó „idegen” programok, processzek felderítését hivatottak szolgálni. Ezen projektek közül kiemelkedik a LIDS (Linux Intrusion Detection System, <http://www.lids.org>), amely kernelszinten valósít meg a betörés figyeléséhez fontos funkciókat, mint például kapupásztázás-figyelő, az állományintegritás- és processzvédelmi rendszerek. Ezekon felül található még a LIDS-ben ACL támogatás és mandatory access control (MAC) is.

Konkluzió:

Nem szabad szem elől tévesztenünk, hogy a behatolásdetektáló rendszerek nem védik meg automatikusan a rendszereket a támadásoktól, azaz továbbra is szükséges a jó konfigurálás és a biztonsági hibák, illetve hibajavítások nyomon követése. Természetesen sokat segíthet egy IDS, főleg a támadások felderítésében és az azokra való gyors reagálásban. Fontos felismerni továbbá, hogy a hálózati és host-IDS-ek más-más támadási típusok ellen nyújthatnak védelmet, így ezek párhuzamos alkalmazása erősen indokolt.

11.4.8. Sebezhetőség felmérése (scannerek)

Szerverünk vagy hálózatunk biztonságos üzemeltetése során fontos szempont, hogy rendszerünket sebezhetőségi tesztek vessük alá. Ebben a helyzetben megpróbáljuk

rendszerünket a másik oldalról, azaz a támadó (cracker) szempontjából szemlélni. Az ilyen önbetörési kísérletek (penetration tests) haszna, hogy segítségével megismerhetjük rendszerünk gyenge pontjait – és megpróbálhatjuk kijavítani azokat. Fontos megjegyeznünk, hogy az automatikus scannerek segítségével csak a már ismert hibák felfedezésére van mód, ha ennél többet szeretnénk, akkor a rendszerünket érdemes szakemberekkel elvégzettetett penetration testnek vagy teljes körű rendszerauditnak alávetni. Nem szabad elfelednünk, a scanner sem ismerhet fel minden hibát. A scanner segítségével lehetőségünk van időzített és rendszeres ellenőrzések futtatására is.

Nessus (<http://www.nessus.org/>)

Hálózati biztonsági scanner, amely két részből áll. Az egyik (szerveroldal) a feltérképezés (scan) futtatásáért felelős, míg a kliensoldal feladata a szerveroldal vezérlése. A szerveroldali daemon jelenleg Unix-alapú rendszereken működik, míg a kliensoldali program elérhető Unix mellett win32 platformra is. Érdekesség, hogy létezik egy java-alapú, platformfüggetlen GUI is, ám ennek fejlesztése némileg lelassult. Lehetőségünk van természetesen a kliens- és szerveroldal egy gépen való futtatására is. A szerveroldalon a kliensek azonosításához felhasználókat tudunk definiálni, akik különböző jogosultságokkal használhatják a szervert (pl. csak megadott gépeket scannelhet csak megadott típusú pluginekkal), az autentikációhoz a kliensek jelszót vagy RSA-kulcsot használhatnak. A támadási tesztek (exploits) pluginek formájában tárolja a Nessus szerveroldali daemonja a Nessushoz fejlesztet Nasl script nyelven megírva, de természetesen lehetőségük van más nyelveken (például C-ben) saját pluginek elkészítésére. A Nessus fejlesztő csapata is folyamatosan követi a megjelent biztonsági hibákat és adja ki az újabb plugineket. A plugineket csoportokba rendezve találjuk típusuk, illetve veszélyességük alapján, így letilthatjuk azok használatát, amelyek sebezhető célpont esetén annak leállítását okozhatják. A Nessus módot ad időzített, folyamatos, rendszeres, sőt differenciális tesztek futtatására is. A differenciális tesztek segítségével rögzíthetünk egy állapotot, ezek után csak az ettől való eltérés esetén kapunk riasztást (például egy szerveren megjelent egy új nyitott port). A Nessus emellett több támadási trükköt is ismer, amivel megkerülhetők a beépített védelmi mechanizmusok.

Satan (<http://www.fish.com/satan/>)

Az első jelentős security scanner volt a „piacon”, sajnos fejlesztése jelenleg a többi programhoz képest lemaradásban van, de mindezek ellenére több projekt született meg a Satan lefektette alapokon, ilyen például a Saint vagy a Sara. A Satan program különböző Unix-rendszereken képes futni, használatához webes felület áll rendelkezésre.

Saint (<http://www.wwdsi.com/saint/index.html>)

A Satanból továbbfejlesztett security scanner, fejlesztését a Saint Corporation végzi, amely a programot különböző kereskedelmi elemekkel is bővítette. A Saint a Satan hagyományait követve szintén könnyen vezérelhető webes felületről, de

parancssorból is futtatható. A Saint adatbázisa folyamatosan és nagyon jó reakcióidővel frissül, köszönhetően a mögötte álló kommersz támogatásnak. A Saint „személyében” egy sok funkcióval felszerelt, könnyen üzemeltethető programcsomag áll rendelkezésünkre.

Nmap (<http://www.nmap.org/>)

Igazából nem nevezhető security scannernek, sokkal inkább egy kapupásztázó (portscanner) eszköz, amelyet külső programként más security scannerek is előszeretettel használnak. Jelenleg az nmap tekinthető a Unix-világban a portscannerek etalonjának. Az nmap segítségével a pásztázás teljes tárháza rendelkezésünkre áll, és bár nem tartalmaz exploit-adatbázist, így is az egyik legjobban használható eszköz.

Konkluzió:

A security scannerek alkotóinak problémája, hogy szoftverükkel egy kétélű fegyvert hoznak létre, amely segítségére van a rendszergazdáknak rendszerük biztonságosabbá tételében, de rossz kezekbe kerülve hatékony támadófegyverként is felhasználhatóak. Mindezek ellenére a szabad szoftver világában több jól használható, jól karbantartott megoldás is elérhető, melyek közül néhány még kommersz támogatással is rendelkezik.

11.4.9. A Linux mint router (WAN eszközök)

A Linux-rendszerek térhódításukat a szerverek, kiszolgálók piacán, azaz a hálózattal kapcsolatos alkalmazások terén kezdték, következik ez a Unix- és Linux-rendszerek kimagasló és megbízható hálózati támogatásából. A mai routerekkel szemben támasztott funkcionális igényeket a Linux kernel, illetve más, Linux alatt futó alkalmazások együttesen kielégítik, ezért a Linux jó teljesítmény/ár arányának, illetve a bővíthetőségének és sokrétű felhasználhatóságának köszönhetően elterjedt útválasztó megoldás.

Egy PC-kategóriás gép Linuxot és pár ingyenes, nyílt forráskódú programot futtatva képes kiváltani egy kommersz hardveres router megoldást. Miért használunk ma mégis „hagyományos” útválasztókat (routereket)?

A különbség a linuxos routerek és a célhardver-megoldások között a teljesítmény, illetve a támogatottság. Alapvetően elmondható, hogy a linuxos router-megoldásokhoz is vásárolható támogatás (szinte minden, Linuxszal foglalkozó cégnél) és elérhető dokumentáció, de ezeknek nincs akkora múltja, mint a célhardvert gyártó és támogató cégeknek.

A Linux kernel támogatja a routoláshoz szükséges funkciókat, ezen felül pluszként megtalálható benne Policy Routing (Source Routing), melynek a segítségével a statikus routingtáblát sokkal jobban, precízebben, a különleges igényeknek megfelelően tudjuk konfigurálni, továbbá elérhetünk a kernelben QoS támogatást,

állapottartó csomagszűrő funkciókat és természetesen különböző hálózati csatolók drivereit (ethernet, fast-ethernet, gigabit-ethernet, ISDN, bérelt vonal, PPP, LAPB, HDLC PPP over ethernet (ADSL) stb.). Az IPSec-alapú VPN-megoldáshoz a FreeSwan (<http://www.freeswan.org>, lásd VPN fejezet) eszközzel juthatunk hozzá.

Különböző dinamikus útválasztási protokollok támogatására is léteznek eszközök, mint például a Zebra (<http://www.zebra.org>, a RIPv1,2, OSPF, BGP4); a Gated (<http://www.gated.org>, routing); a Poptop (<http://www.poptop.org>, PPTP/L2TP Server).

Mind a kernel funkcióinak nagy része, mind pedig a user-space programok jól dokumentáltak, szabvány-kompatibilisek. A Linux-alapú routerek építése szinte minden, szabad szoftverekkel foglalkozó cég termékpalalettáján szerepel.

Nézzük meg, milyen különbségek és azonosságok vannak a célhardveres és a Linux-alapú megoldások között:

	Célhardver	Linux-alapú
Támogatott protokollok	IPv4, IPv6	IPv4, IPv6
Policy routing	Támogatott	Támogatott
Fontosabb routing-protokollok	RIPv1,2, OSPF, BGP4	RIPv1,2, OSPF,BGP-4
Quality of Service	Támogatott	Támogatott
VPN	IPSec(PreSharedKey, X.509cert, RSA-Key), L2TP, PPTP	IPSec(PreSharedKey, X.509cert, RSA-Key, OpenPGP-Key), L2TP, PPTP, SSH-TUNNEL
NAT	Támogatott	Támogatott
Protokoll-specifikus NAT	Ftp	Ftp, Irc
Packet Filter	ACL-ekel stateful és nem stateful megoldások	Stateful és nem stateful kernelszintű támogatás
Alkalmazásszintű tűzfal lehetősége	Nehezen megvalósítható	Támogatott
Időszinkronizálás	NTP, rdate	NTP, rdate
Távoli naplózás	Syslog(udp)	Syslog, syslog-ng(udp, tcp)
Távmenedzsmet, monitorozás	SNMP, HTTP, telnet, (ssh)	SNMP, HTTP(S), telnet, ssh
Modem-poll	Támogatott	Támogatott
Kapcsolódási felületek	Ethernet, bérelt vonal, ISDN, analóg vonal, PPP, SLIP, *DSL, FrameRelay, VSAT, WireLess	Ethernet, bérelt vonal, ISDN, analóg vonal, PPP, SLIP, *DSL, FrameRelay, VSAT, WireLess
Routolási teljesítmény	Mb/s – Gb/s	~ 300Mb/s (PC-n)
Megbízhatóság	Célhardver, nincs mozgó alkatrész	Mozgó alkatrész (kiváltható), PC-hardver

Linuxos routert szinte bármelyik terjesztésre alapozva lehet készíteni, de vannak

projektek, amelyek célja egy minden funkciót tartalmazó, kicsi és jól kezelhető router létrehozása. Ilyen projekt például a Linux Router Projekt (<http://www.linuxrouter.org>), amely egy floppyrezen elférő minidisztribúciót rejt, felszerelve a routerfunkciók ellátásához szükséges programokkal. Ilyen módon változtathatjuk át régi vagy új PC-ket Linux routerré.

A fentiekből is látszik, hogy a Linux alapú megoldások méltó ellenfelei a célhardvereknek. Egy kis- vagy középvállalat internetes vagy belső routelési igényeinek kielégítésére a kimagasló teljesítmény/ár aránya miatt egy Linux-alapú útválasztó kiválóan alkalmas. Tudásban, megbízhatóságban, teljesítményben is kielégíti a vele szemben támasztott legtöbb igényt. A célhardverek előnye kizárólag a nagyobb teljesítmény, amelynek maximumát Linux esetében is csak a PC-architektúra határai szabják meg, az igények és lehetőségek pontos ismeretében azonban jó választási alternatívát tud nyújtani.

11.4.10. Tanúsítványkiadó (CA) funkciók

A nyilvános kulcsú infrastruktúra (PKI) egyik alapvető eleme a tanúsítványkiadó, angol nevén certificate authority. A CA feladatai közt szerepel a tanúsítványok kiadása, a visszavont tanúsítványok listájának publikálása, valamint a kiadott tanúsítványok ellenőrizhetőségének biztosítása.

A digitális aláírás alkalmazása a gyakorlatban a következőket jelenti:

Minden entitás, legyen az személy, folyamat vagy kiszolgáló, rendelkezik egy aszimmetrikus kulcspárral. A kulcspár egyik felével – az ún. titkos kulccsal – képesek vagyunk egy dokumentumot digitálisan aláírni, illetve ugyanezzel a kulccsal a számunkra a kulcspár másik felével – az ún. publikus kulccsal – titkosított dokumentumot dekódolni.

Amennyiben két fél titkosított kommunikációt kíván folytatni, megtehetik azt, amennyiben rendelkeznek saját kulcspárral, és ugyanakkor megbízható forrásból ismerik a másik fél publikus kulcsát.

A publikus kulcs megbízható megszerzésének problémája az, ami a CA létét indokolja. A CA feladata, hogy megbízhatóan összerendelje a publikus kulcsot a hozzátartozó entitással. Ezt az összerendelést ő maga is egy digitális aláírással hitelesíti. Az entitás nevét, a hozzátartozó publikus kulcsot és az összerendelést hitelesítő aláírást együttesen tanúsítványnak hívjuk.

A publikus kulcs beszerzésének módja tehát a következő: a másik fél nyílt csatornán közli a saját tanúsítványát. A tanúsítványon szereplő aláírást ellenőrizzük, a birtokunkban lévő, általunk megbízhatónak tekintett CA-k kulcsai alapján. Ha ez az aláírás jó és a kiállítóban megbízunk, akkor a tanúsítványban szereplő publikus kulcsban is megbízhatunk.

A tanúsítványok formátumára több szabvány létezik, ma a leggyakrabban használt az ISO-szabványként elfogadott X.509 formátum.

Egy részletesebb összefoglaló a PKI-ról, a nyílt forrású PKI eszközökről a <http://ospkibook.sourceforge.net/> címen található.

OpenSSL

A OpenSSL projekt célja, hogy megvalósítson egy, a szabványoknak (SSLv2/v3, TLSv1, X.509) tökéletesen megfelelő alkalmazást. A szoftver lelke az SSLeay függvénykönyvtár, amelyet szinte minden Linux-terjesztés magában foglal és rengeteg alkalmazás is integrálja. A függvénykönyvtárra épülő parancssori programmal könnyedén létrehozhatunk egy teljes értékű saját tanúsítványkiadót, és elvégezhetjük az ennek üzemeltetéséhez szükséges tevékenységeket. A projekt honlapja bőven tartalmaz leírást a megoldás összes moduljához, beleértve a felhasználókat segítő kézikönyveket és a fejlesztőknek szükséges függvény-dokumentációkat is.

pyCA

Egy, az OpenSSL-re épülő tanúsítványkiadó megoldás, amely python nyelven írott CGI scriptekből áll. Ebből következően böngészőből könnyen elvégezhetők a CA által támogatott funkciók. A megoldás jól dokumentált (fejlesztői dokumentáció, FAQ, telepítési útmutató) viszont teljességéhez még sok minden hiányzik. A pyCA az alábbi funkciókat támogatja:

- CA hierarchia kialakítása, CRL-ek generálása
- CSR-ek generálása böngészőből (a legtöbb böngészőtípus támogatott)
- klienstanúsítványok keresése a CA-adatbázisban
- tanúsítványok online validálása
- LDAP backend-adatbázis használata
- tartalmazza a tanúsítványok előállításához szükséges szkripteket (a CA-kulcsát tároló, nem hálózatra kötött gépre)

A projekt fejlesztése egy ember kezében összpontosul, és sajnos úgy tűnik, lassan halad. Az eszköz legutolsó verziója a 0.6.6-os, amely már több mint két éves.

OpenCA

Ez a projekt tűnik jelenleg a legaktívabb és legteljesebb szabadszoftver-alapú CA-megoldásnak. A programcsomag felépítése a klasszikus PKI-elveket követve 3 fő modulból áll:

- CA-szerver: ennek feladata a tanúsítvány-kiadó titkos kulcsának tárolása, a tanúsítványok és CRL-ek nyilvántartása, generálása
- RA (Registration Authority) szerver: feladata, hogy az adminisztrátorokon keresztül kommunikáljon a CA-val (mivel ez utóbbi nincs hálózatra kötve),

valamint a felhasználók által elérhető a CRL-eket és CRT-eket tartalmazó LDAP-szerverrel

- RA Operator modul: feladata, hogy publikus felületet nyújtson a CA felhasználói számára, akik innen tölthetik le a root-tanúsítványokat, CRL-listákat, valamint itt kezdeményezhetnek tanúsítvány-igényléseket

Az OpenCA jelenleg még csak béta verzióban érhető el, de 9 fejlesztője folyamatosan jobbításán fáradozik. A szoftver dokumentációja jelenleg egy rövid FAQ-ban és levelezőlistában és hogyanokban merül ki.

Konklúzió:

A fenti projektek állapota azt mutatja, hogy van jövője a szabadszoftver-alapú CA-megoldásoknak. Mivel a CA fontos és kritikus alkalmazás mind a biztonságtechnika, mind a rendelkezésre állás tekintetében, szabad szoftver választása előtt mindenképpen szükséges lenne a lehetőségek részletesebb önálló tanulmányban való elemzése. Az elemzésnek lehetőleg magában kell foglalnia a kereskedelmi termékek által nyújtott funkcionalitásokkal való összehasonlítást, valamint szükséges lehet a termékek integritásvédelmi auditja is.

11.4.11. Távoktatás és tanulmány-koordináció

Jelen fejezetben olyan szoftverrendszereket vizsgálunk, melyek elsődleges célja, hogy informatikai eszközökkel segítse vagy akár kiváltsa a tantermi oktatási módszert. Természetesen ezt a feladatot technológiától függően rengeteg módon meg lehet oldani.

A tanulmány-koordinációs rendszerek adta lehetőségek szinte korlátlanok, mégis informatikai szempontból nézve technikai kihívást nem jelentenek. Egy ilyen rendszer főbb funkciói megvalósíthatók levelezőszerver és listaprogram, dinamikus webfelületek (adatbázis-hátérrel), fórumok, dokumentum-publikációs rendszer, kliensoldalon pedig böngésző segítségével.

Persze funkcionalitásában bővíthető a kör akár a valós idejű videó adásokig, de azt kell megállapítanunk, hogy erre is léteznek már szabad szoftveres megoldások. Természetesen ebben az esetben a kliensoldalon is szükség van a megfelelő multimédiás szoftverekre és az azokat üzemeltetni képes hardveres háttérre. Korlátokat nem a szoftveres megoldások hiánya, hanem a hardveres infrastruktúra fejletlensége állíthat fel, mivel mind az adás sugárzásához, mind vételéhez modern hardvereszközök szükségesek, nem beszélve a gyors internet-kapcsolatról.

Távoktatási rendszerek felépítésében kihívást a részfunkciókat elvégző szoftverek integrálása, azok karbantartása és üzemeltetése jelent. Az USA oktatási intézményeiben igen elterjedt Blackboard nevű rendszert gyártó és forgalmazó cég sem elsősorban a szoftvert, hanem annak karbantartását tartja termékének.

A szabad szoftver közösség által fejlesztett megoldások között találunk több sikeres

eszközt is, melyek egyetlen hátránya, hogy jelenleg magyar nyelven nem elérhetők. Ezek közül a sikeresebb projektek példaként a www.sourceforge.net-ről:

- **electure**
web- és php-alapú fejlesztés, 2-es stabil verzió, a fejlesztés koordinációját 2 fő végzi.
- **OneBook**
java-alapú fejlesztés, jelenleg *beta* állapotban, regisztrált fejlesztők száma 6
- **Ilias**
web- és php-alapú fejlesztés, stabil 2-es verzió, érdekessége, hogy ez a kölni egyetem hivatalos e-learning rendszere.

A magyar honlapok között is találunk olyat, amely az e-learning témakörrel foglalkozik, ezek közül a legjelentősebbek a MEH által támogatott <http://www.coedu.hu/> és a <http://www.tavoktatas.hu/> oldalak, melyek közül utóbbiban a már említett Ilias rendszer hivatalos támogatását tervezik.

11.5. Fejlesztési lehetőségek

11.5.1. Központi címtár

Az Informatikai Kormánybiztosság Elektronikus Kormányzat Főcsoport 2002-es évi projektjei közül nagyságában és szerepében is kiemelkedik az egységes kormányzati címtárrendszer kialakításának megkezdése. Ezt a lépést megelőzte a technológiai lehetőségeket pontosan felmérő tanulmány, illetve a tanulmány eredményei alapján kormányzati ajánlás elkészítése. Az ajánlás (amely elérhető a <http://www.ikb.hu/ekormanyzat/pdf/cimtar.pdf> oldalon) pontosan megvizsgálja a címtárrendszerek alapját képező X.500-as szabványt, valamint a címtárakkal való kommunikációra alkalmas LDAP-protokollt. A technológia ismertetése után objektív szempontok alapján elemzi ezen szabványokat megvalósító kereskedelmi termékeket.

A szabad szoftverek között is létezik olyan megoldás, amely versenyre kelhetne a kereskedelmi termékekkel. Ez az OpenLDAP, melynek 2.0.27-es verziója megérdemli a stabil jelzést. A projektnek pontos ütemterve van a megvalósítandó funkcionalitások terén, melyet számos fejlesztője követ. Az OpenLDAP különböző moduljai megvalósítják az X.500 szabvány egy részhalmazát, az LDAPv.3 protokollt, az erős autentikációt, valamint a TLSv1-alapú titkosítást. A szerverrész karbantartása a konfigurációs állomány mellett parancssorból és több, az eredeti projekttől leválva fejlesztett grafikus klienssel is megtehető (Java-, X11-gtk- és php-alapokon).

A felhasználókat FAQ, levelezési lista, adminisztrációs kézikönyv és kereskedelmi támogatás is segíti.

Konklúzió:

Ahhoz, hogy eldönthessük az, OpenLDAP megvalósítja-e a kormányzati ajánlásban szereplő feltételeket, egy, az alkalmazás ennél jóval mélyebb elemzése szükséges. Mint már láthattuk, a címtár szempontjából fontos kliensprogramok (pl. levelezők) nagy része támogatja az LDAP-hozzáférést, de kérdéses ezek kompatibilitása a kereskedelmi termékekkel. Mivel nyílt forrású eszközökről van szó, nem okozhat gondot, ha az alkalmazások által beszélt LDAP nem egyezik meg ezekkel – a probléma fejlesztéssel szabadon orvosolható.

12. Konklúzió

E tanulmány célja, hogy segítséget nyújtson a kormányzati intézmények felelős dolgozóinak, hogy átlássák az informatikai életben kialakuló, a szabad szoftvereket előtérbe helyező trendeket. Szükséges továbbá azon megoldások bemutatása, amelyek szabad szoftverként tudásban, elterjedtségben, támogatottságban felveszik a versenyt a legnépszerűbb kereskedelmi termékekkel. Az alábbiakban összefoglaljuk a kormányzatot is érintő, a szabad szoftverekkel kapcsolatos, a tanulmány által is alátámasztott következtetéseket.

- a) A Szabad Szoftver Mozgalom és a nyílt forráskódú eszközök terjedése nem egy tipikus dotcom-léggömb, amely bármikor kipukkanhat, hanem a valós piaci igényekre és motivációkra alapuló, a jövőbe mutató, gazdasági és információ-technológiai folyamat.
 - b) Ahhoz, hogy a szabad, nyílt forrású szoftverek hazai pozitív megítélése megerősödjön, a lehető leghamarabb szükséges az ezeket érintő jogi helyzet tisztázása és konszolidációja.
 - c) A jelenlegi állapotok szerint a különböző Linux-terjesztések már alkalmasak a személyi számítógépek desktopfunkcióinak ellátására. Az igazi kihívást az irodai programcsomagok használata jelenti, ezzel kapcsolatban a honosított OpenOffice.org megjelenésével a lehetőségek nyitottak.
 - d) A trendek arra utalnak, hogy a Linux néhány éven belül vezető szerepet fog betölteni az olyan beágyazott eszközökben, amelyek a modern, teljesen automatizált lakókörnyezetnek szerves részei lesznek (pl. settopboxok).
 - e) Azt semmiképpen nem látjuk megalapozottnak, hogy a kormányzat kizárólag szabad szoftverek használatát támogassa, hiszen léteznek olyan speciális alkalmazások, melyeknek még nincs nyílt forrású verziója.
 - f) Ezzel szemben szükséges, hogy a kormányzat javaslatot dolgozzon ki a zárt szabványok és monopóliumok által okozott szoftverfüggőséggel szembeni lehetséges lépésekről.
 - g) A kormányzatot érintő kockázatok közül is sok kezelhetővé válik, amint a nyílt szabványok használata előnyöket élvez a megítélt támogatások során.
 - h) Mivel a nyílt szabványok nyílt forráskódú megvalósítása gyakran gyorsítja a nyílt szabvány más implementációit is, javasolt az ilyen referencia-megoldások támogatása. Ezek mellett szükséges nyílt forrású eszközök felhasználását célzó pilot projektek indítása, első körben az informatikailag kevésbé integrált, relatíve izolált kormányzati intézményekben.
 - i) Javasolt, hogy a kormányzat a saját megbízásából fejlesztett szoftverek esetében is várja el a nyílt szabványok használatát, és a fejlesztések végeredményét a szabad szoftvereknek megfelelő licenccel vegye csak át.
 - j) A nyílt forráskódú modell új lehetőséget ad olyan, informatikailag nagy mértékben integrált közszolgálati területeken, mint az egészségügy. Javasoljuk ezen területek és a szabad szoftverek összefüggésének vizsgálatát mindkét oldali szakértők bevonásával.
-

13. Kiegészítő anyagok, mellékletek

13.1. Forrásanyagok

- Linux in the US Government
Linux Journal, February 19, 2002 by Richard Vernon
 - Linux in government: Statistics hard to pin down
CNN.com, January 17, 2001
 - What's behind IBM's Linux plans?
ITWorld
 - Linux at Sun: The Big Push
ITWorld, Februar 19, 2002
 - Analysis of the Impact of Open Source Software
UK Cabinet Office, December 11, 2001
http://www.govtalk.gov.uk/interoperability/egif_document.asp?docnum=430
 - Free Software / Open Source: Information Society Opportunities for Europe?
April, 2000
<http://eu.conecta.it>
 - Free/Libre and Open Source Software: Survey and Study
CORDIS Information Society Tech.
<http://www.cordis.lu/ist/home.html>
-

13.2. 1. Melléklet: licencek

GNU Általános Felhasználói licenc

2. verzió, 1991. június

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place – Suite 330, Boston, MA 02111-1307, USA

Jelen licenc szó szerinti sokszorosítása és terjesztése bárki számára megengedett, változtatni rajta ugyanakkor nem lehet.

Előszó

A legtöbb szoftverlicenc olyan módon van megfogalmazva, hogy korlátozza a felhasználó megosztási és módosítási jogait. Ezzel szemben, a GNU Általános Felhasználói licenc biztosítja a szabad szoftverrel kapcsolatos fenti jogokat –, hogy a szoftver minden felhasználó számára szabad legyen. Jelen Általános Felhasználói licenc a szabad szoftver Alapítvány (Free Software Foundation) legtöbb szoftverére vonatkozik, illetve azon szoftverekre, melyek alkotói azt magukra nézve érvényesnek tekintik. (A szabad szoftver Alapítvány néhány egyéb szoftvere a GNU Állományok Általános Felhasználói licencének hatáskörébe tartozik). A licenc feltételeit saját programjaira is alkalmazhatja.

Amikor szabad szoftverről beszélünk, akkor magára a szabadságra, nem pedig az árra gondolunk. Általános Felhasználói licenceink úgy lettek kialakítva, hogy lehetővé tegyék önnek a szabad szoftverek terjesztését (még úgy is, ha ebből esetleg anyagi haszna származik). A licencek biztosítják emellett a forráskódhoz való hozzáférést, akár automatikusan, akár kérésre, továbbá a szoftver módosítását, illetve egyes alkotórészeinek új, szabad programokban való alkalmazását – és a tudatot, hogy mindez nem ütközik törvénybe.

Az ön jogainak védelmében kénytelenek vagyunk olyan korlátokat felállítani, melyek megakadályozzák, hogy öntől bárki is megvonhassa azokat, vagy felkérhesse a jogokról való lemondásra. E korlátozások egyszersmind felelősséggé változnak abban az esetben, ha terjeszteni, vagy módosítani kívánja a szoftvert.

Ha például egy ilyen programról akár ingyen, akár pénzért másolatokat terjeszt, köteles ellátni a feleket ugyanazon jogokkal, amelyek önt is megilletik. Ezen kívül biztosítania kell azt is, hogy a felek egyaránt megkapják, vagy lehetőségük legyen hozzájutni a forráskódhoz. Jelen feltételeket szintén meg kell mutatnia nekik, annak érdekében, hogy megismerhessék jogaikat.

Az ön jogait két lépésben védjük: (1) a szoftver szerzői jogi védelme, és (2) e licenc felajánlása, mely törvényes felhatalmazást ad önnek a szoftver sokszorosítására, terjesztésére és/vagy módosítására.

Szintén az egyes alkotók, illetve a magunk védelmében szeretnénk, ha mindenki számára világos lenne, hogy a szabad szoftverre nincs garancia. Amennyiben a szoftver valamilyen módosítást követően kerül továbbadásra, szeretnénk, ha a felek tisztában lennének azzal, hogy amit kaptak, nem az eredeti változat, így az esetlegesen felmerülő problémák nem az alkotó érdemeit fogják csorbítani.

Végezetül, minden szabad programot szüntelenül fenyeget a szabadalmaztatás veszélye. Szeretnénk elkerülni, hogy a szabad programok terjesztői kisajátíthassák maguknak a szabadalmi licenceket, minek következtében a szoftver magántulajdonba kerülne. Ennek megelőzése érdekében

egyértelművé tettük, hogy a szabadalmi liceneket olyan módon kell kialakítani, hogy vagy bárki, vagy senki számára se tegyék lehetővé a szabad felhasználást.

Alább a sokszorosítás, terjesztés és módosítás pontos feltételei következnek.

A SOKSZOROSÍTÁS, TERJESZTÉS ÉS MÓDOSÍTÁS FELTÉTELEI

1. E licenc minden olyan programra, vagy más jellegű termékre vonatkozik, melyen megtalálható a szerzői jogtulajdonos által feltüntetett figyelmeztetés, miszerint a program terjesztése jelen Általános Felhasználói licenc feltételei alapján lehetséges. A „Program” alább bármely ilyen jellegű programra, vagy termékre vonatkozik, míg a „Programon alapuló termék” vagy a Programot, vagy az abból származtatott szerzői jogvédett terméket jelöli: másképp szólva, olyan terméket, mely tartalmazza a Programot, vagy annak alkotóelemét, szó szerint, vagy módosításokkal, és/vagy más nyelvre lefordítva. (Innentől a „módosítás” kifejezésbe a fordítás is korlátozás nélkül beleértendő). Minden licenctulajdonos megszólítása „ön”.

A sokszorosításon, terjesztésen és módosításon kívül eső tevékenységek nem képezik részét jelen licencnek. A Program futtatásának megtörténte nem esik korlátozás alá, és a Program által nyert eredmények is csak akkor esnek a licenc hatáskörébe, ha azokból a Programon alapuló termék áll össze (függetlenül attól, hogy az a Program futtatása által jött-e létre). Hogy mindez teljesül-e, az a Program funkciójától függ.

2. Önnek módjában áll a Program eredeti állapotú forráskódjáról szó szerinti másolatot készíteni, illetve terjesztetni azt bármely médiumon keresztül, feltéve, hogy jól láthatóan és megfelelő módon ellát minden egyes példányt a vonatkozó szerzői jogi, illetve garanciamentességi figyelmeztetésekkel; továbbá érintetlenül hagyja az összes jelen licencre és a garancia hiányára hivatkozó megjegyzést; és hogy a Program mellett minden egyes felet ellát az ahhoz tartozó licenc egy példányával.

A másolás során felmerülő költségeikért felszámolhat bizonyos összeget, és belátása szerint, díjfizetés fejében garanciát vállalhat a Programra.

3. Önnek lehetősége van a Program birtokában lévő példányának, vagy példányainak, illetve a Program alkotórészeinek módosítására, ezáltal pedig egy a Programon alapuló termék létrehozására, továbbá e módosítások, vagy a fentiekben (1. szakasz) meghatározott termékek sokszorosítására és terjesztésére, feltéve, hogy betartja az alábbi követelmények mindegyikét:

- a) A megváltoztatott fájlokat könnyen észrevehető jellel köteles ellátni, mely felhívja a figyelmet a változtatás tényére, és tartalmazza annak időpontját is.
- b) Minden egyes ön által terjesztett, vagy létrehozott terméket, legyen az akár egészében, akár részben a Program alkotóeleme, vagy annak bármiféle származéka, köteles teljes egészében térítésmentesen rendelkezésére bocsátani bármely harmadik félnek, az e licencben foglaltaknak megfelelően.
- c) Amennyiben a módosított program futtatáskor interaktívan olvassa be az utasításokat, úgy kell azt kialakítania, hogy ha a programot effajta interaktív használatra teljesen szokványos módon elindítják, kinyomtasson, vagy megjelenítsen egy közleményt, mely tartalmazza a vonatkozó szerzői jogi rendelkezéseket és figyelmeztet a garancia hiányára (vagy arra, hogy ön vállalja a garanciát), továbbá, hogy a felhasználók jelen feltételek szerint terjeszthetik a programot, és megmutatja a felhasználónak, miként tekintheti meg e licenc tartalmát. (Kivétel: ha a Program maga interaktív, de rendszerint nem nyomtat ki effajta

közleményt, akkor a Programon alapuló termék nem kell, hogy megfeleljen e kritériumnak).

E követelmények a módosított termék egészére vonatkoznak. Ha a szóban forgó termék azonosítható részei nem a Program származékai, és józan belátás szerint önálló, elkülönült elemeknek tekinthetők, abban az esetben jelen licenc és feltételei nem vonatkoznak rájuk, amikor önálló termékként kerülnek terjesztésre. Amikor azonban ugyanezen alkotóelemeket egy teljes, Programon alapuló termék részeként terjeszti, úgy annak jelen licenc irányelvei mentén kell történnie, melynek egyéb felhasználókat illető engedélyei felölelik a termék egészét, függetlenül attól, hogy ki alkotta annak elemeit.

E szakasznak nem célja, hogy elvitassa, esetleg megkérdőjelezze a teljes egészében ön által létrehozott termékhez fűződő jogát; sokkal inkább, hogy gyakorolja a Programon alapuló származtatott, vagy közös termékek terjesztése feletti jogot.

Mindemellett, egy másik, nem a Programon alapuló terméknek a Programmal (vagy egy Programon alapuló termékkel) való puszta összehozatala egy közös tárolási, vagy terjesztési egységen, nem helyezi a másik programot e licenc hatásköre alá.

3. Ön sokszorosíthatja és terjesztheti a Programot (vagy az azon alapuló terméket, ld. 2. szakasz) tárgyában, vagy futtatható formában az 1. és a 2. szakasz fenti rendelkezései alapján, feltéve, hogy az alábbiak közül eleget tesz egy feltételnek:
 - a) Mellékeli a hozzá tartozó, géppel olvasható forráskódot, mely az 1. és 2. szakaszban felsorolt kritériumok szerint terjeszthető egy szokványos adathordozó médiumon; vagy,
 - b) Mellékel hozzá egy legalább három év időtartamra érvényes írott ajánlatot, melyben bármely harmadik fél számára, a forrás terjesztésének fizikai költségeinél nem magasabb díj ellenében, teljes, géppel olvasható példányt biztosít a Programhoz tartozó forráskódból, mely az 1. és 2. szakaszban felsorolt kritériumok szerint terjeszthető egy szokványos adathordozó médiumon; vagy,
 - c) Mellékeli a kapott információt a Programhoz tartozó forráskód terjesztésével kapcsolatosan. (Ez a lehetőség csak nem kereskedelmi célú terjesztés esetén engedélyezett, és csak akkor, ha a programot tárgyában, vagy futtatható formában kapta kézhez egy fenti jellegű ajánlat kíséretében, ld. a b alszakaszt).

Egy termék forráskódja azt a formátumot jelenti, melyben a módosítások a leginkább végrehajthatók. Egy futtatható termék esetében a forráskód az összes hozzá tartozó modul teljes forráskódját, továbbá bármely azzal társított kezelőfelület-definíciós fájlt, és az alkalmazás összeszerkesztésének, illetve telepítésének vezérléséhez szükséges szkripteket jelenti. Ritka kivételként, mindazonáltal, a forráskód – az alkalmazás futtatásához használt operációs rendszer fő komponensein (szerkesztő, kernel, stb.) kívül – nem kell, hogy tartalmazza a terjesztéskor (forráskódban, vagy bináris formában) megszokott összes elemet, hacsak maga a szóban forgó elem nem tartozik a futtatható termékhez.

Ha a futtatható termék, vagy tárgy kód terjesztése egy kijelölt helyen való hozzáférés biztosításával történik, akkor a forráskódhoz ugyanazon a helyen való hozzáférés lehetősége a forráskód terjesztésének minősül, annak ellenére, hogy a harmadik fél nem köteles a tárgy kóddal együtt a forráskód lemásolására.

4. A jelen licencben egyértelműen kijelölt kereteken kívül tilos a Program bármennemű sokszorosítása, módosítása, allicencelése, vagy terjesztése. Minden ezzel szembeni sokszorosítási, módosítási, allicencelési, vagy terjesztési kísérlet a jelen licencben meghatározott jogok automatikus megszűnését vonja maga után. Azok a felek, ugyanakkor, akik önön keresztül jutottak másolathoz, vagy jogosultságokhoz, nem veszítik el azokat, amíg maradéktalanul betartják e licenc előírásait.
5. Önnek nem kötelessége elfogadni e licencet, hiszen nem írta alá. Mindazonáltal semmi más nem biztosít lehetőséget önnek a Program és az abból származó termékek módosítására, vagy terjesztésére. E tevékenységek törvénybe ütköznek, amennyiben nem fogadja el jelen licenc feltételeit. Ezért hát a Program (vagy bármilyen a Programon alapuló termék) módosításával, vagy terjesztésével ön az e licencben foglaltakat a Program és minden azon alapuló termék sokszorosítására, terjesztésére és módosítására vonatkozóan önmagára nézve kötelező érvényűnek ismeri el.
6. A Program (vagy bármilyen a Programon alapuló termék) minden továbbadásának alkalmával az átvevő fél automatikusan megkapja az eredeti licenctulajdonostól a Program sokszorosítására, terjesztésére és módosítására vonatkozó licencet, jelen feltételek alapján. Ön nem állíthat fel semmilyen egyéb korlátozást, mely által az átvevő felek ehelyütt biztosított jogait csorbítaná. Mindamellet a licencben foglaltak harmadik féllel való betartatása sem az ön felelőssége.
7. Amennyiben szabadalmi jogsértés vádja éri, netán elítélik, vagy bármilyen más (nem kizárólag szabadalmi eredetű) okból kifolyólag olyan feltételeket szabnak ki önre (akár bírói rendelkezés, akár megegyezés, vagy egyéb alapján), melyek szemben állnak az e licencben foglaltakkal, azok nem mentesítik önt jelen feltételek betartása alól. Abban az esetben, ha a Program terjesztése során képtelen egyidejűleg betartani jelen licenc előírásait, illetve bármely más idevágó követelményt, úgy a Programot egyáltalán nem áll módjában terjeszteni. Ha például egy szabadalmi licenc nem tenné lehetővé a Program jogdíjmentes terjesztését azok számára, akik közvetlenül, vagy közvetve önön keresztül jutottak hozzá másolathoz, akkor az egyetlen alternatíva mellyel mindkét követelménynek eleget tehetne az volna, ha felhagy a Program terjesztésével.

Ha e szakasz bármely részlete bármilyen körülmények között érvénytelennek, vagy betarthatatlannak bizonyulna, akkor a szakasz hátralevő része szolgál irányadóul, annak egésze pedig egyéb körülmények között alkalmazandó.

Jelen szakasznak nem célja, hogy szabadalmak, vagy egyéb tulajdonjogi nyilatkozatok áthágására sarkallja önt, s nem vonja kétségbe az effajta igények helyénvalóságát sem. E szakasz kizárólagos célja az, hogy megvédje a szabad szoftverek terjesztési rendszerének integritását, amit a felhasználói licencek alkalmazása tesz lehetővé. Sokak nagylelkű hozzájárulásának eredményeként e rendszerben igen sokféle szoftver terjesztése megy végbe, ami elsősorban a rendszer következetességébe vetett bizalomnak köszönhető. Az alkotó/donor kizárólagos joga, hogy eldöntse milyen szisztéma szerint kívánja terjeszteni szoftverét, s az átvevő fél nem változtathat e döntésen.

E szakasz rendeltetése az, hogy tökéletesen egyértelművé tegye, ami a licenc hátralevő részének vélhető folyamánya.

8. Ha a Program terjesztését és/vagy használatát egyes országokban szabadalmak, vagy szerzői jogvédett kezelőfelületek korlátozzák, úgy az eredeti szerzői jogtulajdonos, aki e licenc
-

hatásköre alá helyezi a szoftvert, belátása szerint kiegészítheti azt explicit földrajzi korlátokkal, hogy a terjesztés csak a licencből nem kizárt országokban, vagy országok között mehessen végbe. Ilyen esetekben a licenc magában foglalja a fenti jellegű korlátozást, mintha csak az a licenc eredeti része volna.

9. Megtörténhet, hogy a szabad szoftver Alapítvány időről időre felülvizsgált és/vagy új verziókat bocsát ki az Általános Felhasználói licencből. E verziók szellemisége hasonló lesz jelen változathoz, ám részleteikben eltérhetnek, új problémák, új aggályok felmerülése okán.

Minden változat egyedi verziószámmal van ellátva. Ha a Program jelen licenc verziószámára hivatkozik, illetve „bármely újabb verzióra”, akkor önnek a szóban forgó változat, vagy bármely újabb a szabad szoftver Alapítvány által publikált verzió feltételeinek követésére lehetősége van. Ha a Program nem ad meg semmilyen verziószámot, úgy bármely a szabad szoftver Alapítvány által valaha publikált változat megfelel.

10. Amennyiben a Program egyes alkotórészeit egyéb, más feltételek közt terjeszhető szabad programba kívánja átültetni, úgy kérje levélben az alkotó engedélyét. Olyan szoftverek esetében, melyek szerzői joga a szabad szoftver Alapítvány tulajdonában van, írjon a szabad szoftver Alapítványnak; e gyakorlattól csak ritkán térünk el. Döntésünket egyrészt szabad szoftvereink minden származékának szabad voltában való megőrzése, másrészt a szoftverek megosztásának és újbóli felhasználásának általános szempontja fogja befolyásolni.

NINCS GARANCIA

11. MIVEL A PROGRAM LICENCE INGYENES, AZ ÉRVÉNYES JOGSZABÁLYOK ÁLTAL MEGENGEDETT MÉRTÉKIG NINCS RÁ GARANCIA. HACSAK ÍRÁSBAN NEM JELENIK MEG MÁSKÉNT, A SZERZŐI JOGTULAJDONOS ÉS/VAGY EGYÉB FELEK A PROGRAMOT MINDENFAJTA KIFEJEZETT VAGY VÉLELMEZETT GARANCIA NÉLKÜL („AS IS”) BIZTOSÍTJÁK, IDEÉRTVE, DE EZZEL EGYEBEKET NEM KIZÁRVA, A FORGALMAZHATÓSÁGRA ÉS MEGHATÁROZOTT CÉLRA VALÓ ALKALMASSÁGRA VONATKOZÓ VÉLELMEZETT GARANCIÁT. A PROGRAM MINŐSÉGÉT ÉS TELJESÍTMÉNYÉT ILLETŐ KOCKÁZATOK ÖSSZESSÉGÉT ÖN VÁLLALJA. AMENNYIBEN A PROGRAM HASZNÁLHATATLANNAK BIZONYUL, ÚGY MINDEN SZÜKSÉGES SZERVIZ-JAVÍTÁSI- ÉS KORREKCIÓS KÖLTSÉG ÖNT TERHEL.

12. SEMMI ESETBEN, HACSAK AZ ÉRVÉNYES JOGSZABÁLYOK ELLENKEZŐLEG NEM RENDELKEZNEK, VAGY ÍRÁSOS MEGEGYEZÉS NEM SZÜLETIK RÓLA, NEM TEHETŐ A SZERZŐI JOGTULAJDONOS, VAGY BÁRMELY EGYÉB FÉL, AKI A FENTIEKKEL ÖSSZHANGBAN MÓDOSÍTTJA ÉS/VAGY TERJESZTI A PROGRAMOT, FELELŐSSÉ A PROGRAM HASZNÁLATA, VAGY NEM MEGFELELŐ HASZNÁLATA KÖVETKEZTÉBEN FELMERÜLŐ KÁROKÉRT, LEGYEN AZ BÁRMILYEN GYAKORI, EGYEDI, VÉLETLENSZERŐ VAGY SZÜKSÉGSZERŰEN BEKÖVETKEZŐ KÁR (IDEÉRTVE, DE EZZEL EGYEBEKET NEM KIZÁRVA, AZ ADATVESZTÉST, VAGY ADATOK HASZNÁLHATATLANNÁ VÁLÁSÁT, ILLETVE EGYÉB ÖN, VAGY BÁRMELY HARMADIK FÉL ÁLTAL ELSZENVEDETT VESZTESÉGET, VAGY A PROGRAM MÁS PROGRAMOKKAL VALÓ INKOMPATIBILITÁSÁT), MÉG ABBAN AZ ESETBEN IS, HA A SZERZŐI JOGTULAJDONOS, VAGY EGYÉB FELEK TISZTÁBAN VOLTAK AZ EFFAJTA KÁROK LEHETŐSÉGÉVEL.

A FELTÉTELEK VÉGE

A feltételek alkalmazása az ön által készített programokra

Ha ön kifejleszt egy új programot, és azt szeretné, hogy az mindenki lehető legnagyobb hasznára váljon, annak legmegfelelőbb módja, ha szabad szoftverré teszi, melyet bárki terjeszthet és módosíthat jelen feltételek alapján.

Ha így dönt, csatolja az alábbi megjegyzéseket a programhoz. Annak érdekében, hogy a garancia hiányára egyértelműen felhívja a figyelmet, legjobb, ha minden forrásfájl elejéhez csatolja a figyelmeztetést; emellett az összes fájlban tartalmaznia kell minimálisan a „copyright” sort és egy útmutatót a teljes szöveg hollétét illetően.

Egy sor a program nevének és pár szó arról, hogy mire való.

Copyright (C) *éééé az alkotó neve*

Ez a program szabad szoftver; terjesztheti és/vagy módosíthatja a szabad szoftver Alapítvány által kiadott GNU Általános Felhasználói licenc feltételei alapján; vagy a 2., vagy (választása szerint) bármely újabb verziót használva.

E program abban a reményben kerül terjesztésre, hogy hasznos lesz, ám BÁRMIFÉLE GARANCIA NÉLKÜL; ideértve a FORGALMAZHATÓSÁG és a MEGHATÁROZOTT CÉLRA VALÓ ALKALMASSÁG vélelmezett garanciáját. További részletekért vö. a GNU Általános Felhasználói Licenccel.

A programmal együtt meg kellett, hogy kapja a GNU Általános Felhasználói Szerződés egy példányát is. Amennyiben nem így történt, írjon a szabad szoftver Alapítványnak, Inc., 59 Temple Place – Suite 330, Boston, MA 02111-1307, USA.

Adjon felvilágosítást elektronikus és hagyományos postacíméről is.

Ha a program interaktív, írasson ki vele egy rövid figyelmeztetést, ha interaktív módban elindítják:

Gnomovision 69. verzió, Copyright (C) *év az alkotó neve*

A Gnomovision-re NINCS SEMMIFÉLE GARANCIA; további részletekért gépeljen „show w”-t. Ez egy szabad szoftver és önnek lehetősége van terjeszteni bizonyos feltételek mellett; gépeljen „show c”-t a részletekért.

A „show w”, illetve „show c” hipotetikus parancsok az Általános Felhasználói Szerződés megfelelő részeit kell, hogy megjelenítsék. Természetesen az ön által használt parancsok elnevezése nem feltétlenül „show w”, vagy „show c” lesz; azokat egérekattintásokra, vagy menüelemekre is felcserélheti – attól függően, hogy mi illik inkább a programhoz.

Mindemellett, ha szükséges, kérje meg munkáltatóját (amennyiben programozóként dolgozik), vagy iskoláját, ha tanul, hogy írjon alá egy „szerzői jog lemondó-nyilatkozatot” a programhoz. Alább következnek egy példa, fiktív nevekkkel:

Yoyodyne, Inc., ezennel lemond minden szerzői jogi érdekeltségéről a James Hacker által írt „Gnomovision” programban (mely átjárókat nyit a szerkesztőkben).

Ty Coon aláírása, 1989. április 1.

Ty Coon, a bűn elnöke

Jelen Általános Felhasználói licenc nem teszi lehetővé az ön programjának beépítését szabadalmaztatott programokba. Mennyiben az ön programja egy szubrutin állomány, úgy talán hasznosabb volna, ha lehetővé tenné magántulajdonú alkalmazások azzal történő összefűzését. Ha így dönt, használja a GNU Állományok Általános Felhasználói licencét jelen licenc helyett.

GNU Szabad Dokumentációs licenc

1.1 verzió, 2000. március

Copyright (C) 2000 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Jelen licenc szó szerinti sokszorosítása és terjesztése bárki számára megengedett, változtatni rajta ugyanakkor nem lehet.

0. ELŐSZÓ

Jelen licenc célja egy olyan kézikönyv, tankönyv, vagy effajta írott dokumentum megalkotása, mely a szó szoros értelmében „szabad”: annak érdekében, hogy mindenkinek biztosítsa a szöveg sokszorosításának és terjesztésének teljes szabadságát, módosításokkal, vagy anélkül, akár kereskedelmi, akár nem-kereskedelmi úton. Másfelől, e licenc megőrzi a szerző, vagy kiadó munkája elismeréséhez fűződő jogát, s egyúttal mentesíti őt a mások által beiktatott módosítások következményei alól.

Jelen licenc egyfajta „etalonnak” tekinthető, ami nem jelent mást, mint hogy a dokumentumból származtatott munkák maguk is szabad minősítést kell, hogy kapjanak. E dokumentum egyben a GNU Általános Felhasználói licenc kiegészítőjeként is szolgál, mely egy a szabad szoftverekre vonatkozó etalon licenc.

E licencet a szabad szoftverek kézikönyveiben való használatra alkottuk, hiszen a szabad szoftver egyben szabad dokumentációt is igényel: egy szabad programot olyan kézikönyvvel kell ellátni, mely ugyanazon szabadságokat biztosítja, mint maga a program. Jelen licenc, mindazonáltal, nem korlátozódik pusztán kézikönyvekre; feltételei tetszőleges tárgy körű írott dokumentumra alkalmazhatók, függetlenül attól, hogy az könyvformában valaha megjelent-e. Mindamellet e licencet főként olyan munkákhoz ajánljuk, melyek elsődleges célja az útmutatás, vagy a tájékoztatás.

1. ALKALMAZHATÓSÁG ÉS DEFINÍCIÓK

E licenc minden olyan kézikönyvre, vagy más jellegű munkára vonatkozik, melyen megtalálható a szerzői jogtulajdonos által feltüntetett figyelmeztetés, miszerint a dokumentum terjesztése jelen licenc feltételei alapján lehetséges. A „Dokumentum” alább bármely ilyen jellegű kézikönyvre, vagy egyéb munkára vonatkozik. A lakosság minden tagja potenciális licenctulajdonosnak tekinthető, és mindegyikük megszólítása egyaránt „ön”.

A Dokumentum „Módosított Változata” bármely olyan munkára vonatkozik, mely tartalmazza a Dokumentumot, vagy annak elemeit akár szó szerint, akár módosításokkal, és/vagy más nyelvre lefordítva.

A „Másodlagos Szakasz” egy egyedi névvel bíró függelék, esetleg a Dokumentum egy megelőző szakasza, mely kizárólag a kiadóknak, vagy az alkotóknak a Dokumentum átfogó tárgy köréhez (vagy kapcsolódó témákhoz) fűződő viszonyáról szól, és nem tartalmaz semmi olyat, ami közvetlenül ezen átfogó témakör alá eshet. (Ha például a Dokumentum részben egy matematika tankönyv, úgy a Másodlagos Szakaszban nincs lehetőség matematikai tárgyú magyarázatokra). A fenti kapcsolat tárgya lehet a témakörrel, vagy a kapcsolódó témákkal való történelmi viszony, illetve az azokra vonatkozó jogi, kereskedelmi, filozófiai, etikai, vagy politikai felfogás.

A „Nem Változtatható Szakaszok” olyan speciális Másodlagos Szakaszok számítanak, melyek

ilyetén való meghatározását az a közlemény tartalmazza, miszerint a Dokumentum jelen licenc hatálya alatt lett kiadva.

A „Borítószövegek” olyan rövid szövegrészek, melyek Címlap-szöveggént, illetve Hátlap-szöveggént kerülnek felsorolásra abban a közleményben, miszerint a Dokumentum jelen licenc hatálya alatt lett kiadva.

A Dokumentum „Átlátszó” példánya olyan géppel-olvasható változatot jelöl, mely a nyilvánosság számára hozzáférhető formátumban kerül terjesztésre, továbbá melynek tartalma szokványos szövegszerkesztő-programokkal, illetve (pixelekből álló képek esetén) szokványos képmegjelenítő-programokkal, vagy (rajzok esetén) általánosan hozzáférhető rajprogramok segítségével azonnal és közvetlenül megtekinthető, vagy módosítható; továbbá olyan formátumban mely alkalmas a szövegszerkesztőkbe való bevitelre, vagy a szövegszerkesztők által kezelt formátumokba való automatikus átalakításra. Egy olyan, egyébként Átlátszó formátumban készült példány, melynek markupja úgy lett kialakítva, hogy megakadályozza, vagy eltántorítsa az olvasókat minden további módosítástól, nem tekinthető Átlátszónak. A nem „Átlátszó” példányok az „Átlátszatlan” megnevezést kapják.

Az Átlátszóság kritériumainak megfelelő formátumok között megtalálható például a markup nélküli egyszerű ASCII, a Texinfo beviteli formátum, a LaTeX beviteli formátum, az SGML vagy az XML egy általánosan hozzáférhető DTD használatával, és a standardnak megfelelő, emberi módosításra tervezett egyszerű HTML. Az Átlátszatlan formátumok közé sorolható a PostScript, a PDF, a szabadalmaztatott és csak fizetős szövegszerkesztőkkel olvasható formátumok, az olyan SGML vagy XML, melyhez a szükséges DTD és/vagy egyéb feldolgozó eszközök nem általánosan hozzáférhetők, és az olyan gépileg-generált HTML formátum, melyet egyes szövegszerkesztők hoznak létre, kizárólag kiviteli célra.

Egy nyomtatott könyv esetében a „Címlap” magát a címlapot, illetve bármely azt kiegészítő további oldalt jelöl, amely a jelen licencben definiált címlap-tartalmak közzétételéhez szükséges. Az olyan formátumú munkáknál, melyek nem rendelkeznek effajta címlappal, a „Címlap” a munka címéhez legközelebb eső, ám a szöveg törzsét megelőző szövegrészeket jelöli.

2. SZÓ SZERINTI SOKSZOROSÍTÁS

Önnek lehetősége van a dokumentum kereskedelmi, vagy nem-kereskedelmi jellegű sokszorosítására és terjesztésére, bármely médiumon keresztül, feltéve, hogy jelen licenc, a szerzői jogi figyelmeztetés, továbbá a Dokumentumot jelen licenc hatálya alá rendelő közlemény minden példányban egyaránt megjelenik, és hogy e feltételeken kívül semmi mást nem tesz hozzá a szöveghez. Nem alkothat olyan technikai korlátokat, melyek megakadályozhatják, vagy szabályozhatják az ön által terjesztett példányok elolvasását, vagy sokszorosítását. Mindazonáltal elfogadhat bizonyos összeget a másolatok fejében. Amennyiben az ön által terjesztett példányok száma meghalad egy bizonyos mennyiséget, úgy a 3. szakasz feltételeinek is eleget kell tennie.

A fenti kritériumok alapján kölcsönbe adhat egyes példányokat, de akár nyilvánosan is közzéteheti a szöveget.

3. SOKSZOROSÍTÁS NAGYOBB MENNYISÉGBEN

Amennyiben 100-nál több nyomtatott változatot tesz közzé a Dokumentumból, és annak licence feltételül szabja a Borítószövegek meglétét, úgy minden egyes példányt köteles ellátni olyan borítólappal, melyeken a következő Borítószövegek tisztán és olvashatóan fel vannak tüntetve: Címlap-szövegek a címlapon, illetve Hátlap-szövegek a hátlapon. Mindkét borítólappra egyértelműen és olvashatóan rá kell vezetnie a kiadó, vagyis jelen esetben az ön nevét. A címlapon

a Dokumentum teljes címének jól láthatóan, továbbá minden egyes szónak azonos szedésben kell megjelennie. Ezen felül, belátása szerint, további részleteket is hozzáadhat a borítólapokhoz. Amennyiben az esetleges módosítások kizárólag a borítólapokat érintik, és feltéve, hogy a Dokumentum címe változatlan marad, továbbá a borítólapok megfelelnek minden egyéb követelménynek, úgy a sokszorosítás ettől eltekintve szó szerinti reprodukciónak minősül.

Abban az esetben, ha a borítólapok bármelyikén megkövetelt szövegrészek túl hosszúnak bizonyulnának az olvasható közzétételhez, úgy csak az elsőként felsoroltakat kell feltüntetnie (amennyi józan belátás szerint elfér) a tényleges borítón, a továbbiak pedig átkerülhetnek a következő oldalakra.

Amennyiben 100-nál több Átlátszatlan példányt tesz közzé, vagy terjeszt a Dokumentumból, úgy köteles vagy egy géppel-olvasható Átlátszó példányt mellékelni minden egyes Átlátszatlan példányhoz, vagy leírni minden egyes Átlátszatlan példányban egy a módosítatlan Átlátszó példányt tartalmazó nyilvános hozzáférésű számítógép-hálózat elérhetőségét, ahonnan bárki, anonim módon, térítésmentesen letöltheti azt, egy közismert hálózati protokoll használatával. Ha az utóbbi lehetőséget választja, köteles gondoskodni arról, hogy attól a naptól kezdve, amikor az utolsó Átlátszatlan példány is terjesztésre került (akár közvetlenül ön által, akár kiskereskedelmi forgalomban), a fenti helyen közzétett Átlátszó példány még legalább egy évig hozzáférhető legyen a felhasználók számára.

Megkérjük, ámde nem kötelezzük önt arra, hogy minden esetben, amikor nagyobb példányszámú terjesztésbe kezd, már jóval ezt megelőzően lépjen kapcsolatba a Dokumentum szerzőivel, annak érdekében, hogy megkaphassa tőlük a Dokumentum esetleges felújított változatát.

4. MÓDOSÍTÁS

Önnek lehetősége van a Dokumentum Módosított Változatának sokszorosítására és terjesztésére a 2. és 3. szakaszok fenti rendelkezései alapján, feltéve, hogy a Módosított Változatot kizárólag jelen licenc feltételeivel összhangban teszi közzé, ahol a Módosított Változat a Dokumentum szerepét tölti be, ezáltal lehetőséget biztosítva annak terjesztésére és módosítására bárkinek, aki csak hozzájut egy példányához. Mindezen felül, a Módosított Változat az alábbi követelményeknek is meg kell, hogy feleljen:

- A. A Címlapon (és ha van, a borítókon) tüntessen fel egy a Dokumentumétól, illetve bármely korábbi változatától eltérő címet (melyeknek, ha vannak, a Dokumentum Előzmények szakaszában kell szerepelniük). Egy korábbi változat címét csak akkor használhatja, ha annak szerzője engedélyezte azt.
- B. A Címlapon szerzőkként sorolja fel a Módosított Változatban elvégzett változtatásokért felelős személyeket, vagy entitásokat, továbbá a Dokumentum fő szerzői közül legkevesebb ötöt (vagy mindet, ha nincsenek öten).
- C. A Címlapon a Módosított Változat közzétételéért felelős személyt tüntesse fel kiadóként.
- D. A Dokumentum összes szerzői jogi figyelmeztetését hagyja érintetlenül.
- E. Saját módosításaira vonatkozóan is tegyen közzé egy szerzői jogi megjegyzést, a többi ilyen jellegű figyelmeztetés mellett.
- F. Rögtön a szerzői jogi figyelmeztetéseket követően tüntessen fel egy közleményt, az alábbi Függelék mintájára, melyben engedélyezi a Módosított Változat felhasználását jelen licenc feltételei alapján.
- G. A fenti közleményben hagyja érintetlenül a Nem Változtatható Szakaszok és a

szükséges Borítószevegek jelen Dokumentum licencében előírt teljes listáját.

- H. Mellékelje jelen licenc egy eredeti példányát.
- I. Az „Előzmények” szakaszt, illetve annak címét szintén hagyja érintetlenül, emellett adjon hozzá egy új elemet, amely minimálisan tartalmazza a Módosított Változat címét, kiadási évét, továbbá az új szerzők, illetve a kiadó nevét, a Címlapon láthatókhoz hasonlóan. Amennyiben a Dokumentum nem tartalmaz semmiféle „Előzmények” elnevezésű szakaszt, úgy hozzon létre egyet, mely tartalmazza a Dokumentum címét, kiadási évét, továbbá a szerzők, illetve a kiadó nevét, a Címlapon láthatókhoz hasonlóan; majd ezt követően adjon hozzá egy új, a Módosított Változatra vonatkozó elemet, a fentiekkel összhangban.
- J. Ne tegyen változtatásokat a Dokumentumban megadott Átlátszó példány nyilvános hálózati elérhetőségét (ha van ilyen) illetően, vagy hasonlóképp, a Dokumentum alapjául szolgáló korábbi változatok hálózati helyére vonatkozóan. Ezek az „Előzmények” szakaszban is szerepelhetnek. Csak abban az esetben hagyhatja el egyes korábbi változatok hálózati elérhetőségét, ha azok legkevesebb négy évvel a Dokumentum előtt készültek, vagy ha maga az alkotó engedélyezi azt.
- K. Bármely „Köszönetnyilvánítás”, vagy „Ajánlások” szakasz címét hagyja érintetlenül, továbbá gondoskodjon arról, hogy azok tartalma és hangvétele az egyes hozzájárulókat, és/vagy az ajánlásokat illetően változatlan maradjon.
- L. A Dokumentum összes Nem Változtatható Szakaszát hagyja érintetlenül, úgy címüket, mint tartalmukat illetően. A szakaszok számozása, vagy bármely azzal egyenértékű jelölés nem tartozik a szakaszcímek közé.
- M. Töröljön minden „Jóváhagyás” elnevezésű szakaszt. Effajta szakaszok nem képezhetik részét a Módosított Változatnak.
- N. Ne nevezzen át semmilyen létező szakaszt „Jóváhagyás”-ra, vagy olyasmire, mely címében a Nem Változtatható Szakaszokkal ütközhet.

Ha a Módosított Változat új megelőző szakaszokat tartalmaz, vagy olyan függelékeket, melyek Másodlagos Szakaszok minősülnek, ám nem tartalmaznak a Dokumentumból származó anyagot, abban az esetben, belátása szerint, e szakaszok némelyikét, vagy akár az összeset nem változtathatóként sorolhatja be. Ehhez nem kell mást tennie, mint felsorolni a szóban forgó címeket a Módosított Változat licencének Nem Változtatható Szakaszok listájában. E címeknek határozottan el kell különülnie minden egyéb szakaszcímtől.

„Jóváhagyás” elnevezésű szakaszt csak akkor adhat a Dokumentumhoz, ha az kizárólag a Módosított Változatra utaló megjegyzéseket tartalmaz – például mások recenzióira vonatkozóan, vagy hogy egy szervezet a szöveget egy standard mérvadó definíciójaként ismerte el.

Címlap-szöveg gyanánt egy legfeljebb öt szóból álló szövegrészt adhat meg, a Hátlap-szöveg esetén pedig 25 szót fűzhet a Módosított Változat Borítószevegeinek végéhez. Bármely entitás csak és kizárólag egy Címlap- és egy Hátlap-szövegrészt adhat (akár közvetítőn keresztül) a Dokumentumhoz. Ha a dokumentum már eleve rendelkezik Borítószeveggel, akár azért, mert azt korábban ön adta hozzá, vagy mert valaki más önön keresztül gondoskodott erről, abban az esetben nincs lehetőség újabb Borítószöveg hozzáadására; a régit mindazonáltal lecserélheti, abban az esetben, ha annak kiadója egyértelműen engedélyezi azt.

A Dokumentum szerzője/i és kiadója/i jelen licenc alapján nem teszik lehetővé nevük nyilvános felhasználását egyetlen Módosított Változat támogatása, vagy támogatottsága érdekében sem.

5. KOMBINÁLT DOKUMENTUMOK

Önnek lehetősége van a Dokumentum egyéb, e licenc hatálya alatt kiadott dokumentumokkal való kombinálására a 4. szakasz módosított változatokra vonatkozó rendelkezései alapján, feltéve, hogy a kombináció módosítás nélkül tartalmazza az eredeti dokumentumok összes Nem Változtatható Szakaszát, és hogy azok mind Nem Változtatható Szakaszokként kerülnek felsorolásra a kombinált munka licencében.

A kombinált munkának jelen licenc mindössze egy példányát kell tartalmaznia, az egymással átfedésben lévő Nem Változtatható Szakaszok pedig kiválthatók egy összegzett példánnyal. Amennyiben több Nem Változtatható Szakasz szerepelne ugyanazon címmel, ám eltérő tartalommal, úgy alakítsa át minden egyes szakasz címét olyan módon, hogy mögéírja zárójelben az eredeti szerző és kiadó nevét (ha ismeri), vagy egy egyedi sorszámot. Ha szükséges, a Nem Változtatható Szakaszok címeivel is végezze el a fenti módosításokat a kombinált munka licencében.

A kombinált munkában az eredeti dokumentumok összes „Előzmények” elnevezésű szakaszát össze kell olvasztania, miáltal egy összefüggő „Előzmények” szakasz jön létre; hasonlóképp kell eljárnia a „Köszönetnyilvánítás”, illetve az „Ajánlások” szakaszok tekintetében. Ugyanakkor minden „Jóváhagyás” elnevezésű szakaszt törölnie kell.

6. DOKUMENTUMGYŰJTEMÉNYEK

Önnek lehetősége van a Dokumentumból, illetve bármely egyéb, e licenc hatálya alatt kiadott dokumentumból gyűjteményt létrehozni, és az egyes dokumentumokban található licenceket egyetlen példánnyal kiváltani, feltéve, hogy a gyűjteményben szereplő összes dokumentum esetén minden más tekintetben követi jelen licenc feltételeit, azok szó szerinti sokszorosítására vonatkozóan.

Tetszése szerint ki is emelhet egy meghatározott dokumentumot a gyűjteményből, továbbá terjesztheti azt jelen licenc feltételei alapján, feltéve, hogy a szóban forgó dokumentumhoz mellékeli e licenc egy példányát, és minden egyéb tekintetben betartja jelen licenc előírásait a dokumentum szó szerinti sokszorosítására vonatkozóan.

7. ÖSSZEFŰZÉS FÜGGETLEN MUNKÁKKAL

A Dokumentum és annak származékainak különálló, vagy független dokumentumokkal, illetve munkákkal való összefűzése egy közös tárolási, vagy terjesztési egységen, egészében nem tekinthető a Dokumentum Módosított Változatának, feltéve, hogy az összefűzés nem lesz szerzői jogvédett. Az effajta összefűzés eredményeként „összegzés” jön létre, ám jelen licenc nem érvényes az abban a Dokumentummal együtt szereplő önálló munkákra, hacsak azok nem a Dokumentum származékai.

Amennyiben a 3. szakasz Borítószövegekre vonatkozó rendelkezései alkalmazhatók a Dokumentum e példányaira, és a Dokumentum a teljes összegzésnek kevesebb, mint egynegyedét teszi ki, úgy a Dokumentum Borítószövegeit olyan módon is el lehet helyezni, hogy azok csak magát a Dokumentumot fogják át. Minden más esetben a teljes összegzés borítólapjain kell feltüntetni a fenti szövegeket.

8. FORDÍTÁS

A fordítás egyfajta módosításnak tekinthető, így hát a Dokumentum lefordított példányai a 4.

szakasz rendelkezései alapján terjeszthetők. A Nem Változtatható Szakaszok lefordítása külön engedélyt igényel a szerzői jogtulajdonostól, mindazonáltal közzéteheti a lefordított változatokat is abban az esetben, ha az eredeti Nem Változtatható Szakaszokat is belefoglalja a munkába. E licenc lefordítására ugyanezek a feltételek érvényesek, vagyis a lefordított változat csak akkor jelenhet meg, ha mellette ott van az eredeti, angol nyelvű licenc szövege is. Amennyiben eltérés mutatkozik az eredeti változat, illetve a fordítás között, úgy a licenc angol nyelvű eredetije tekintendő mérvadónak.

9. MEGSZŰNÉS

A jelen licencben egyértelműen kijelölt kereteken kívül tilos a Dokumentum bárminemű sokszorosítása, módosítása, allicencelése, vagy terjesztése. Minden ezzel szembeni sokszorosítási, módosítási, allicencelési, vagy terjesztési kísérlet a jelen licencben meghatározott jogok automatikus megszűnését vonja maga után. Azok a felek, ugyanakkor, akik önön keresztül jutottak másolathoz, vagy jogosultságokhoz, nem veszítik el azokat, amíg maradéktalanul betartják e licenc előírásait.

10. JELEN LICENC JÖVŐBELI JAVÍTÁSAI

Megtörténhet, hogy a szabad szoftver Alapítvány időről időre felülvizsgált és/vagy új verziókat bocsát ki a GNU Szabad Dokumentációs licencből. E verziók szellemisége hasonló lesz jelen változatéhoz, ám részleteikben eltérhetnek, új problémák, új aggályok felmerülése okán. Vö.: <http://www.gnu.org/copyleft/>

A licenc minden változata egyedi verziószámmal van ellátva. Ha a Dokumentum jelen licenc egy konkrét, számozott verziójára, „vagy bármely újabb verzióra” hivatkozik, úgy önnek a szóban forgó változat, vagy bármely újabb a szabad szoftver Alapítvány által (nem vázlatként) publikált verzió feltételeinek követésére lehetősége van. Ha a Dokumentum nem ad meg semmilyen verziószámot, úgy bármely a szabad szoftver Alapítvány által valaha (nem vázlatként) publikált változat megfelel.

FÜGGELÉK: A licenc alkalmazása saját dokumentumaira

Ha e licencet egy ön által írt dokumentumban kívánja használni, akkor mellékelje hozzá a licenc egy példányát, továbbá vezesse rá az alábbi szerzői jogi és licenc közleményeket, rögtön a címlapot követően:

Copyright (C) ÉV AZ ÖN NEVE.

E közlemény felhatalmazást ad önnek jelen dokumentum sokszorosítására, terjesztésére és/vagy módosítására a szabad szoftver Alapítvány által kiadott GNU Szabad Dokumentációs licenc 1.1-es, vagy bármely azt követő verziójának feltételei alapján. A Nem Változtatható Szakaszok neve SOROLJA FEL A CÍMŰKET, a Címlap-szövegek neve LISTA, a Hátlap-szövegek neve pedig LISTA. E licenc egy példányát a „GNU Szabad Dokumentációs licenc” elnevezésű szakasz alatt találja.

Ha a szövegben nincsenek Nem Változtatható Szakaszok, úgy írjon „nincs Nem Változtatható Szakasz”-t, ahelyett, hogy egyenként felsorolná azokat. Ha nincsenek Címlap-szövegek, akkor írjon „nincs Címlap-szöveg”-et, ahelyett, hogy „a Címlap-szövegek neve LISTA”, és hasonlóképp járjon el a Hátlap-szövegek esetében is.

Amennyiben a dokumentum haladó programkód-példákat is tartalmaz, úgy azt javasoljuk, hogy e példákat egy választása szerinti szabad szoftver licenc alatt közölje – mint például a GNU Általános Felhasználói licenc –, hogy lehetővé tegye a kódok szabad szoftverekben való alkalmazását.

13.3. 2. sz. melléklet: Néhány szoftver dokumentációjának elérhetősége

ABIWORD

- FAQ:
<http://www.abiword.org/>
- Felhasználói dokumentáció:
<http://www.abiword.org/support/manual/>
<http://www.abiword.org/help/en-US/>
- Fejlesztői információk:
<http://www.abiword.org/developers/>
- Levelezési lista:
abiword-dev-request@abisource.com
abiword-user-request@abisource.com?body=subscribe

A program beépített felhasználó-segédlettel és súgóval rendelkezik.

APACHE

- Installációs dokumentáció:
<http://www.freenetshop.org/docs.html>
- Felhasználói dokumentáció:
http://www.tp4.ruhr-uni-bochum.de/SoftwareDocs/apache_1.3/index.html
<http://www.auburn.edu/docs/apache/>
- Adminisztrátori dokumentáció:
<http://httpd.apache.org/docs-projekt/>
- Könyvek:
http://www.pearsonptg.com/book_detail/0,3771,0672322749,00.html
<http://www.apacheadmin.com/book/>
- Fejlesztési dokumentáció:
<http://www.apache.org/>
- Man oldalak:
<http://software.oit.pdx.edu/cgi-bin/hman-amanda?ManSection=1m&ManTopic=apache#toc1>

- FAQ oldalak:
<http://www.apache.org/foundation/faq.html>
- Levelezési lista:
Apache News and Announcements
<http://www.apache.org/foundation/maillinglists.html#foundation-announce>

Conference Announcements list
<http://www.apache.org/foundation/maillinglists.html#foundation-apachecon>

Foundation member discussions
<http://www.apache.org/foundation/maillinglists.html#foundation-members>

Board of Directors list
<http://www.apache.org/foundation/maillinglists.html#foundation-board>
+
<http://www.apache.org/foundation/maillinglists.html>

BALSA

- FAQ, Felhasználói segédlet:
<http://balsa.gnome.org/faq.html>
<http://www.linuxplanet.com/linuxplanet/reports/1978/2/>
- Levelezési lista:
<http://balsa.gnome.org/list.html>
balsa-list@gnome.org
- Installációs segédlet:
<http://balsa.gnome.org/cvs-help.html>

CAUDIUM

- FAQ:
<http://caudium.net/faqs/>
 - Levelezési listák:
<http://caudium.org/lists/>
 - Fejlesztési információk, fejlesztői közösség:
<http://caudium.org/contrib/>
 - Dokumentációk (User, administrator, installation)
<http://caudium.net/cdocs/>
-

CUPS

- FAQ, online help:
<http://www.cups.org/faq.html>
- Adminisztrátori dokumentáció:
<http://www.cups.org/sam.html>
<http://www.cups.org/sam.pdf>
- Felhasználói dokumentáció:
<http://www.cups.org/sum.html>
<http://www.cups.org/sum.pdf>
- Fejlesztői dokumentáció:
<http://www.cups.org/spm.html>
<http://www.cups.org/spm.pdf>
<http://www.cups.org/sdd.html>
<http://www.cups.org/sdd.pdf>
- Levelező listák:(cups.announce, cups.bugs, cups.cvs cups.development)
<http://www.cups.org/newsgroups.php>

HYLAFAX

- HOWTO:
<http://www.hylafax.org/howto/index.html>
- FAQ:
<http://www.hylafax.org/HylaFAQ/index.html>
- Levelezőlisták:
<http://www.hylafax.org/mailling-lists.html>
- User guide, on-line dokumentáció:
<http://www.hylafax.org/overview.html>
- Installation guide:
<http://www.hylafax.org/setup.html>
- Fejlesztési információk
hylafax-devel@hylafax.org

KMAIL

- FAQ:
-

<http://docs.kde.org/3.0/kdenetwork/kmail/faq.html>
<http://docs.kde.org/2.2.2/kdenetwork/kmail/faq.html>

- Dokumentáció:
<http://docs.kde.org/3.0/kdenetwork/kmail/>
<http://docs.kde.org/2.2.2/kdenetwork/kmail/>
- Levelezőlisták:
<http://lists.kde.org/?l=kde-kmail&r=1&w=2>
kmail@mail.kde.org.
- Felhasználói segédlet:
<http://kmail.kde.org/>
- Installációs segédlet, online segédlet:
<http://www.kde.org/install-source.html>

LPRNG

- HOWTO
<http://www.lprng.com/LPRng-HOWTO/LPRng-HOWTO.html>
- User guide:
<http://www.lprng.com/LPRng-HOWTO/LISA98.ppt>
- Administrator guide:
<http://www.lprng.com/PrintingCookbook/index.html>
- Levelezési listák:
<http://www.findmail.com/list/lprng>
<http://arthur.ucsd.edu/LPRng-Archive/>
<http://www.lrz-muenchen.de/services/netzdienste/email/email-archive/>

MUTT

- FAQ:
<http://www.fefe.de/muttfaq/faq>
 - Man oldalak:
http://www.mutt.org/doc/man_page.html
 - Felhasználói dokumentáció:
<http://www.mutt.org/doc/manual/>
<http://www.mutt.org/doc/manual.txt>
<http://mutt.blackfish.org.uk/>
-

- Levelezési listák:
 - * mutt-announce@mutt.org -- Announcements.
 - * mutt-users@mutt.org -- General Discussion.
 - * mutt-users Archive (AIMS)
 - * mutt-users Archive (mail-archive.com)
 - * mutt-users Archive (Egroups)
 - * mutt-dev@mutt.org -- Development Community.
 - * mutt-dev Archive (AIMS)
 - * mutt-dev Archive (Egroups)
 - * mutt-po@mutt.org -- Translation Issues.

- További információk:

<http://www.mutt.org/>

OPENOFFICE.ORG

- FAQ:
 - http://www.openoffice.org/FAQs/main_faq_new.html
 - <http://www.openoffice.org/FAQs/mostfaqs.html>
 - <http://www.openoffice.org/FAQs/asian.html>
 - http://www.openoffice.org/FAQs/build_faq.html
 - <http://whiteboard.openoffice.org/doc/>
 - <http://api.openoffice.org/faq.html>
 - <http://udk.openoffice.org/faq.html>
 - <http://xml.openoffice.org/faq.html>
 - Levelezési listák:
 - http://www.openoffice.org/mail_list.html#projekt
 - Telepítési fordítási útmutatók:
 - http://www.openoffice.org/dev_docs/source/build_linux.html
 - http://www.openoffice.org/dev_docs/source/build_solaris.html
 - http://www.openoffice.org/dev_docs/source/build_mac.html
 - http://www.openoffice.org/dev_docs/source/build_windows.html
 - Felhasználói útmutató:
 - http://whiteboard.openoffice.org/doc/HOW_TO/index.html
 - http://whiteboard.openoffice.org/doc/Samples_Templates/index.html
 - http://whiteboard.openoffice.org/doc/user_faq/index.html
 - Levelezési listák:
 - <http://whiteboard.openoffice.org/servlets/projektMailingListList>
-

- Fejlesztéssel kapcsolatos és egyéb további információk:
<http://www.openoffice.org>