

Az OSCAR-forradalom

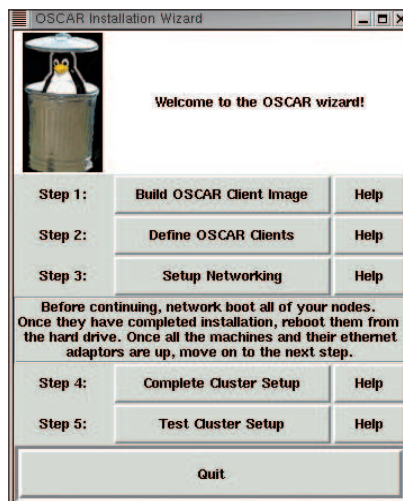
Az Open Source Cluster Application Resource (OSCAR) története és céljai.

Tavalyelőtt áprilisban rendezték meg az első OSCAR értekezletet az Oak Ridge National Labtól egy kőhajításnyira található szállodában. Vegyes közönség gyűlt össze, képviseltették magukat az állami kutatóintézetek, az egyetemek és az ipar. Az összejövetel célja az volt, hogy az abban az időben CCDK-nak (Community Cluster Development Kit) nevezett, később az OCG-vé (Open Cluster Group) alakuló csoportról és első projektjükéről, az OSCAR-ról beszéljessenek.

A résztvevők a feladatot összetevőire bontották, és minden összetevőhöz „cárt” (vezért) és „nyafogókat” (érdekelte feleket) választottak. A cárok feladata az adott összetevőhöz tartozó csoport vezetése volt, a nyafogóknak pedig elég gyakran és hangosan kellett nyafogniuk, hogy a dolgok menetrend szerint valósuljanak meg a csoport igényeinek megfelelően. Már az első találkozón, ahol megválasztották a cárokat és a nyafogókat, látszott, hogy az OSCAR fejlesztése minden eddigi program fejlesztésétől eltérő lesz. Végül is hol található nálunk egy olyan másik projektet, amelyben az IBM, a Dell, az SGI és az Intel szorosan együttműködve dolgozik egy nyílt forrású megoldáson egy olyan óriási érdeklődés övezte témában, mint a géptelepek?

Az OSCAR ötlete vacsora közben jutott eszébe az Intelnél kutató **Dr. Timothy Mattson**-nak és az Oak Ridge National Labnál kutató **Dr. Stephen Scott**-nak. Éppen egy a DOE által szponzorált géptelepekkel foglalkozó értekezleten vettek részt az Argonne National Labnál, és az foglalkoztatta őket, hogyan lehetne a linuxos géptelepeket a nagy-közönséggel elfogadtatni. Rájöttek, az a gond, hogy a nem programozók számára túlságosan nehéz a saját géptelep összeállításának feladata. A *How to Build a Beowulfhoz* (Sterling et. al.) hasonló könyvek sokat segíthetnek a számítástechnikához értőknek az alapelvek megértésében és az első géptelep megépítésében, azonban maradtak még nyomasztó gondok. Rengeteg kódot kell letölteni, amelyek megbízhatósága, támogatottsága, összeférhetősége és leírása

nem egyforma. Bizonyos csomagok leírása elavult vagy ellentmondásos adatokat tartalmazott. Számos Linux-terjesztés közül lehetett választani, és mindegyik úgy különböztette meg magát a másiktól, hogy egy kicsit másképp működtek benne a dolgok. Ez azt jelentette, hogy bizonyos parancsok működése eltért egymástól, vagy más csomagokat kellett telepíteni a szolgál-



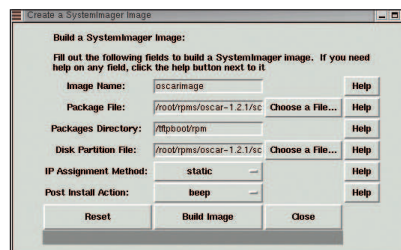
1. kép Az OSCAR-varázsló

tatás megfelelő működéséhez. Ráműtettek, hogy a gond abból adódik – mivel mindenki a saját géptelepét próbálta felépíteni azért, hogy rátegye a kezét az olcsó géptelepekkel végzett számítások piacára –, hogy minden géptelep a nulláról építettek fel. Kell lennie bizonyos gazdaságosságnak, egy helyre kell gyűjteni a legjobb programokat, eljárásokat és leírásokat, egységesíteni a különböző géptípusokra szánt csomagokat, és mindezt ingyenesen elérhetővé kell tenni. Az OSCAR projekt alapelve, hogy nem programozók is könnyen hozzanak létre géptelepeket.

Az első találkozó

Az Oak Ridge-i első értekezlet résztvevői a következők voltak: Tim Mattson és Stephen Scott, az OCG vezetői; **Gabriel Bonner** az SGI-től; **Dave Lombard** az MSC.Software-től; **Rob Pennington** az NCSA-tól; **Greg Lindahl**, aki jelenleg a

Conservative Computersnél dolgozik; **Ken Briskey** és én az IBM-től; **Greg Astfalk** a HP-től; végül **Clay Taylor** az MPI Software Technológiestől. Nem sokkal az első értekezlet után a Veridian-tól **Braahn Mann** csatlakozott a csapathoz, magával hozva a párhuzamos ütemezésben szerzett tapasztalatait. Csatlakozott továbbá **Jeremy Enos** és **Neil Gorsuch** az NCSA-tól (akik az SSH-t



2. kép SystemImager-lenyomat készítése

valósították meg az OSCAR-ban), illetve **Mike Brim** az Oak Ridge National Labtól (aki a legtöbb egységesítő parancsfájlt írta, és a csomagolásban vesz részt). Az utóbbi időben csatlakozott az OSCAR projekthez **Jeff Squyres** és **Brian Barrett** az Indiana Egyetemről a LAM/MPI képviselőként. A sokszínű csoport három alapvető pontban egyezett meg:

1. A géptelepek nagyteljesítményű számításokban való széles körű elterjedését gátolja, hogy nincsenek olyan elfogadott programcsomagok, amelyek elég megbízhatóak, és az átlagos felhasználók is elég könnyen használhatók lehetnek őket.
2. A csoport a programok terjesztésénél a nyílt forrású modellt választja. Minden hozzájárulást ingyenesen kell nyilvánosságra hozni – lehetőleg forráskód formájában és a Berkeley nyílt forrású felhasználási szerződés pontjainál összhangban.
3. A csoport céljai úgy valósíthatók meg a legjobban, ha a géptelepek tudományának úttörői által sok év kemény munkájával felhalmozott tudásra építve a legismertebb módszereket terjesztik.

Miután ezeket az alapelveket kőbe vésték, a csoport a géptelepeket alkotó összetevőket az „oszd meg és uralkodj”

elvért követve szedte össze. Az egyes összetevőkért felelős csoportok eldöntötték, hogy melyek az egyes összetevők-höz a legjobban ismert nyílt forrású megoldások, majd az adatokat az egész csoportnak átadták. Az egyes összetevőkre kidolgozott legjobb megoldások együttesen elfogadható megoldást kínáltak az egész géptelepre. Az összetevők kidolgozása után is fáradságos és időigényes munka volt az egész rendszer összeépítése, ezt az Oak Ridge National Labben végezték Mike Brim és *Brian Luethke* vezetésével. Tőlük függetlenül dolgozott a Dell tesztlőcsapata *Jenwei Hsieh, Tau Leng* és *Yung-Chin Fang* vezetésével. Erőfeszítéseiknek köszönhetően a személyes találkozásokon alapuló és a távolból végzett összeépítési bulikon az OSCAR olyasmivé vált, amit már érdemes volt megosztani a közösséggel.

A programcsomag

Közel egy évet vett igénybe, míg az OSCAR próbaváltozatát Dallasban 2000 novemberében bemutathatták az SC2000 kiállításon, az Oak Ridge National Lab standján. A próbaváltozat eltérő kiszolgálótelepen futott, amelyet a Dell és az SGI szállított. Röviddel ezután bejelentették az első kiadást, és sikeresen szerepeltek vele a 2001 februárjában megrendezett New York-i LinuxWorld Expón az Intel standján. Azóta is folyamatosan fejlesztik az OSCAR programcsomagját, amely jelenleg az alábbi összetevőkből áll:

- **Linux-telepítés:** SIS (rendszer telepítő csomag)
A SIS egy nyílt forrású géptelep-telepítő eszköz, amely a LUI (linuxos segédprogram a géptelep telepítéséhez) és a népszerű SystemImager egyesítésével jött létre. A SIS-t az IBM-es *Michael Chase-Salerno* és *Sean Dague* fejlesztette, és az 1.2.1-es OSCAR-változatban jelent meg. Az utóbbi időben a SystemImager készítője, *Brian Finley* a Bald Guy Software-től is látogatja az OSCAR értekezleteket, és ráhajt az ingyen sörre.
- **Biztonság:** OpenSSH
Ez a biztonságos kapcsolatok létrehozásának leggyakoribb módja linuxos környezetben. Az OpenSSH programcsomag kezeli a biztonságos kapcsolatokat, a kiszolgálóoldali SSH-szolgáltatásokat, a biztonsági kulcsok létrehozását és sok más egyebet, amire a számítógépek közötti biztonságos kapcsolathoz szükség van.

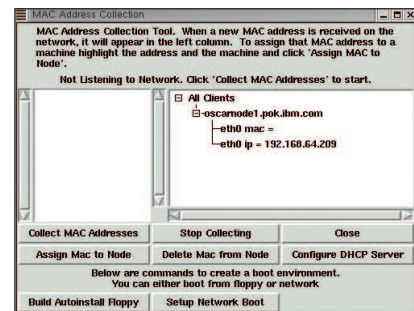
- **A géptelep karbantartása:** az egész géptelepre kiterjedő karbantartó műveletekre az OSCAR a Cluster Command and Control (C3) nevű csomagot használja, amelyet az Oak Ridge National Labnál dolgozó *Stephen Scott* és az East Tennessee állami egyetemen tanuló *Brian Luethke* fejlesztette ki. Az C3 olyan benyomást kelt, mintha egyetlen rendszeren dolgoznánk, egy parancs az egész géptelepre hatással van. A C3 telepítve marad a géptelep csomópontjain, hogy a felhasználók és a rendszergazdák később is használhassák.
- **Programozó környezet:** Message-Passing Interface (MPI – üzenetküldő felület) és Parallel Virtual Machine (PVM – párhuzamos virtuális gép). A géptelepek felhasználói általában maguk írják a géptelepen futtatandó programokat. Több különböző megközelítés létezik a programírára e környezetben. A leggyakoribb módszer az üzenetküldő könyvtárak használata. Jelenleg az OSCAR által telepített fordítóprogramok és a matematikai könyvtárak a Linux-terjesztésből jönnek. A LAM/MPI és az MPICH is elérhető az OSCAR 1.1 óta.
- **Terheléselosztás:** Portable Batch System (PBS) a Veridiantól és a Maui Scheduler (a Maui High Times Computing Center fejlesztése). A géptelep idejének beosztásához valamilyen terhelés- és feladatelosztás szükséges. A Maui az OSCAR feladatelosztója – ez osztja el az erőforrásokat és dönt az ütemezésről. A PBS a feladatkiszolgáló, illetve -indító, a feladatok indításán és leállításán kívül a feladatsorokat is kezeli.

Az MSC.Software és az OSCAR

Az MSC.Software Corporation Systems divíziója által kifejlesztett MSC.Linux-terjesztés különösen fontos az OSCAR elfogadtatása szempontjából. Röviddel az OSCAR 1.0-s változatának megjelenése után az MSC.Software bejelentette saját megoldásait a géptelepekre, ez volt az MSC.Linux Version 2001 operációs rendszer. Ez a 2001-es ajánlat nagyrészt az OSCAR-on alapult, és az első olyan kereskedelmi termék volt, amely az OCG munkája nyomán jött létre. *Joe Griffin* az MSC.Software-től Webmin felületet adott a LUI-hoz (a LUI az első OSCAR géptelep-telepítő eszköz volt), amely alacsony szintű LUI-parancsokat készített a csomópontok számára. Ennek a felületnek a segítségével könnyen



3. kép Ügyfelek hozzáadása a SystemImager-lenyomathoz



4. kép MAC-címek összegyűjtése a SystemImager-lenyomathoz

megadhatók a géptelep csomópontjai és az egyes csomópontokra telepítendő erőforrások. Az OCG egyik eredeti célja az volt, hogy a vállalatok felismerjék a nyílt OSCAR programcsomag értékét, és saját zárt vagy nyílt programcsomagot építsenek az OSCAR köré. Ez esetben ugyanis megmenekülnek a géptelep felépítésének aprólékos munkájától, nem nekik kell az alapvető szerkezeteket kitalálni, és sokkal jobban összpontosíthatnak azokra a világszínvonalú fejlesztésekre, amelyekkel a saját megoldásukat megkülönböztethetik a többiekétől.

A közös munka

Mint más szerteágazó nyílt forrású projektnél, itt is a kezdetektől világos volt, hogy a társaság nem feltétlenül tud egy időben és egy helyen együtt dolgozni. Az utazási költségek egyszerűen túl magasak voltak, és ilyen sok ember idejének egyeztetése is nehézséget jelentett. A munka irányítására a csoport hetente telefonértekezletet tartott, használták a levelezőlistát, és tanácskozások vagy kiállítások alkalmával néha összejöttek. Szemtől szemben a fejlesztők a negyedévente tartott „összekovácsolási” bulikon találkoztak, az egyik ilyen az Intelnél volt az oregoni Hillsboróban, a másik az NCSA-nél Illinoisban. Hogy az egység az összejövetelek között is megmarad-

jon, bevezették az úgynevezett elosztott egységbe rendező buli intézményét, angol rövidítéssel a DIP napot. A DIP napokon minden résztvevő, akinek van géptelege, idejét az OSCAR-nak szenteli. Mindenki letölti az OSCAR csomagjait, telepíti, futtatja, és a hibákat jelenti a csoportnak. A DIP napokon a programozóktól elvárják, hogy a hibákat azonnal javítsák, így a kódot rövid időn belül egymás után sokszor ki tudják próbálni. A DIP nap alatt több konferenciabeszélgetésre is sor kerül, megtárgyalják az eredményeket, új feladatokat tűznek ki, és meghatározzák a feladatok fontosságát sorrendjét. A DIP napokkal és a személyes találkozásokkal irányított OSCAR projekt nagy előrehaladást tett a megbízhatóság és a tudás terén.

Az első benyomások az OSCAR-ról

Az első dolog, amit az OSCAR fájljának kicsomagolása után észrevehettünk, hogy készítői alapos munkát végeztek. Kiterjedt leírás szól a telepítésről, a rendszerkövetelményekről, a felhasználói szerződésről (GPL) és az OSCAR mögött meghúzódó elméletről. Létezik gyorsítópályó tanfolyam a türelmetleneknek és egy teljes leírás is. Azt is megfigyelhetjük, hogy semmi egyebet nem kell letölteni, minden benne van az OSCAR egyetlen *tar* fájljában. Az OSCAR a géptelegek hagyományos szemléletén alapul: egy kiszolgáló és N számítógépes csomópont van. A kiszolgáló felelős a csomópontok telepítéséért, ütemezéséért és megfigyeléséért. A gépteleg csomópontjainak egyformáknak kell lenniük, azaz ugyanazt a Linux-terjesztést és ugyanazt a változatot kell használniuk. A felhasználó először az `install_cluster` parancsot adja ki, ami sok mindent elvégez. Létrehozza a szükséges könyvtárakat, kezeli az NFS-t és az `xinetd`-t, telepíti a LAM/MPI-t, a C3-at, a PBS-t, a Maui-t, az OpenSSH-t, a SIS-t, a Perl-t, a SystemImagert és az MPICH-t, frissíti a különböző profilokat és beállító parancsfájlokat, végül elindítja az OSCAR-varázslót.

Ha minden jól megy, kellemes meglepetés ér bennünket: ez az OSCAR-varázsló. Az OSCAR-csapat úgy érezte, hogy a varázsló az OSCAR egyik megkülönböztető jegye lesz a linuxos géptelegek megoldásai között. A varázsló célja egyértelmű – követni kell az utasításokat, és a gépteleg fájdalommentesen telepíthető. A varázsló minden lépésének van egy belépési és egy kilépési feltétele. Ha a kilépési feltétel teljesül, az OSCAR egy sikerüzenetet ad vissza, ami azt jelzi, hogy biztonságosan a következő

lépésre lehet ugrani. A varázslót követve a **Build OSCAR client image** gomb megnyomásával a második panelre érünk, neve **Create a SystemImager Image** panel.

A **SystemImager** panel célja, hogy olyan fájlrendszert hozzon létre a kiszolgálón, amely később minden ügyfélre telepítve lesz. Az **Image Name** mező lehetővé teszi, hogy a felhasználó több **SystemImager** lenyomatot készítsen, mindegyiket egyedi névvel ellátva. A **Package File** mezőben adhatjuk meg, hogy milyen csomagokat telepítünk az ügyfelekre. Az OSCAR által nyújtott alapbeállítás a legtöbb felhasználó igényeit kielégíti. A **Packages Directory** mezőben adhatjuk meg az RPM-csomagok forrását, végül a **Disk Partition File** mezőben lehet testreszabni a lemez felosztását. Az OSCAR erre is kínál alapbeállításokat, IDE- és SCSI-meghajtókra egyaránt. A **Build Image** gomb megnyomása után megindul az ügyféllelennyomat készítése a kiszolgálón. Ha ez kész, továbbléphetünk a varázsló második pontjára, az OSCAR-ügyfelek megadásához.

Az **Add Clients** panelen a felhasználó megadhatja az új ügyfelekhez rendelendő IP-címtartományt. Minden egyes ügyfélhez egy lenyomatnév tartozik, amelyet az **Image Name** mezőbe kell írni. Egy ügyfélcsoportot IP-címtartomány formájában adhatunk meg, hogy mindegyik ügyfélnek azonos legyen az alhálózati maszkja és az alapértelmezett átjárója. Az **Addclients** gomb az ügyfél beállítófájlját készíti el a SIS számára. Ha ez megvan, lépünk tovább a varázsló harmadik pontjára, a hálózati beállításokhoz.

A **Setup Networking** panelen a gépteleg

csomópontjainak MAC-címeit gyűjtjük össze. Ha a csomópont a hálózatról rendszert tud indítani (PXE), egyszerűen rendeljük hozzá a MAC-címhez az ügyfelet, és kapcsoljuk be. Ha csomópont nem ismeri a PXE-t, készíteni kell egy **SystemImager** rendszerindító lemezt a **Build Autoinstall Floppy** gomb segítségével. A MAC-címek összegyűjtése után nyomjuk meg a **Configure the DHCP** gombot, és kapcsoljuk be a csomópontokat, hogy megkezdődjön a Linux telepítése. A csomópontok telepítése után minden csomópont elkezd idegesítően és kitaratóan sípolni, jelezve, hogy el kell távolítani a rendszerindító lemezt vagy ki kell kapcsolni a PXE-t, és a csomópontot újraindítani a merevlemezről. Miután mindegyik újraindult, a csomópontok készen állnak a **Complete Cluster Setup** gomb megnyomására. Ez a művelet egyezteteti az időt és lefuttat néhány csomagfüggő telepítés utáni parancsfájlt. A **Test Cluster Setup** gomb hatására rövid feladatok futnak le – kipróbálva az ütemezőt és a párhuzamos könyvtárakat. Ha a gépteleg telepítése elkészült és minden működik, a próbaparancsfájlokat futtathatjuk, melyek a gépteleg általános egészségi állapotát mérik fel. A `test_install` parancsfájl ellenőrzi, hogy a PBS és a Maui ütemező be van-e állítva és fut-e, a C3 eszközök telepítve vannak-e, és a gépteleg készen áll-e párhuzamos feladatok végrehajtására.

Az OSCAR jövője

A cikk írásának idején az OSCAR 1.2.1 a legfrissebb, amely Red Hat Linux 7.1 alatt fut. Az MSC.Linux Version 2001 az OSCAR 1.1-en alapul. Az 1.0 és 1.1 kiadá-

Kapcsolódó címek

- A LUI projekt honlapja ➔ <http://oss.software.ibm.com/loi>
- Az MSC.Linux terjesztés letöltési helye ➔ <http://www.msclinux.com>
- OCG- és OSCAR-előadások és -cikkek ➔ <http://www.csm.ornl.gov/oscar>
- Az OCG honlapja ➔ <http://www.openclustergroup.org>
- Az OSCAR projekt honlapja ➔ <http://www.sourceforge.net/projects/oscar>
- A System Configurator projekt honlapja ➔ <http://www.sourceforge.net/projects/systemconfig>
- A SystemImager projekt honlapja ➔ <http://www.sourceforge.net/projects/systemimager>
- A System Installation Suite projekt honlapja ➔ <http://www.sourceforge.net/projects/sisuite>
- A SystemInstaller projekt honlapja ➔ <http://www.sourceforge.net/projects/systeminstaller>
- A Tennessee Oak Ridge Cluster projekt honlapja ➔ <http://www.epm.ornl.gov/torc/TORCHomepage.htm>
- A TORC projekt honlapja ➔ <http://www.epm.ornl.gov/torc/TORCprojectspage.htm>

sok eléggé népszerűek voltak a közösség tagjai között – körülbelül 25 000-szer töltötték le a SourceForge-ről. Jelenleg az az OSCAR legnagyobb gondja, hogy viszonylag kevés fejlesztőjének időt és energiát kell áldoznia az olyan új programcsomagok beépítésére, amelyeket az emberek az OSCAR-ban látni szeretnének. Az OSCAR 2.0 projekt már jó úton halad, és ennek a változatnak az egyik hangsúlyos pontja, hogy az összetevőket szabványos API-kon keresztül lehessen bővíteni, így bárki hozzáteheti majd a nyílt forrású csomagját az OSCAR-hoz. Maga az OCG is nő. Amióta Tim Mattson az Inteltől egyéves szabadságra távozott, Jeff Squyres az Indiana Egyetemről vette át a vezetést az OSCAR 2.0 felépítése és egységessége témakörében. *Ibrahim Haddad* az Ericcsontól (sokat ír a Linux Journalba is) olyan érdekes ötlettel érkezett a társasághoz, amelyek lehetővé tennék, hogy az OSCAR közel olyan megbízható legyen, mint a telekommunikációs alkalmazások. *Jim Garlick* a Lawrence Livermore National Lab képviselőjében szintén csatlakozott a társasághoz, magával hozva a nagy géptelepek méretezéséről szerzett tapasztalatait.

Felületek és terjesztések

A legelső OSCAR értekezleten a felek megegyeztek, hogy az OSCAR egyetlen Linux-terjesztéstől vagy felülettől sem függhet. Ennek ellenére az OCG mostanáig főleg a Red Hat és az MSC.Linux terjesztésekre összpontosított az IA-32 felületen. 2002-ben a látókör tágulni fog. A SIS OSCAR-ba építésének az volt a célja, hogy minden RPM-en alapuló terjesztést támogathassanak, ezek a SuSE, Turbolinux, Red Hat, MSC.Linux és a Caldera. A későbbiekben a deb-eken alapuló terjesztések támogatása is megvalósul (pl.: Debian). Ráadásul a SIS felépítése lehetővé teszi, hogy más felületekre is könnyen átvigyék. Az NCSA máris rendelkezik egy Itaniumon futó OSCAR próbával, és az Oak Ridge a Red Hat 7.2-t próbálja ki. Mivel az API nyílt, és az OSCAR számos terjesztés alatt és nagyszámú felületen képes futni, ebben az évben nagy fellendülés várható.

Záró gondolatok

Az OSCAR hatása a linuxos géptelepeket használó közösségre több szempontból is vizsgálható. A legnyilvánvalóbb előny, hogy az OSCAR hasznos segéd-

eszköz a géptelep működtetéséhez, és a legkülönbébb gyártók rendszereiben is használható. Sok nehézséget elkerülhetünk a segítségével, mert a szükséges programokat nem kell különböző webhelyekről összeszedgetni. Ez tényleg olyan megoldás géptelepek létrehozására, amelyet a nem programozó felhasználó is képes használni. A felszín alatt az OCG egyre növekvő szervezetét találjuk, amelyben részt vesznek az állami kutatóintézetek, az egyetemek és az ipar képviselői, hogy együtt új nyílt forrású megoldást fejlesszenek Linuxra. Visszatekintve számára a talán legfontosabb hosszú távú hozzájárulás a közönség az, hogy a nyílt forrás jobbá tételében a közös munkálkodás új formáit fejlesztették ki.

Linux Journal 2002. június, 98. szám



Richard Ferri
az IBM Linux Technology Centerben vezetőprogramozó. Nyílt forrású linuxos géptelepeken dolgozik.

© Kiskapu Kft. Minden jog fenntartva

Linux-parancsok: szerelem első látásra

Amikor parancsokról beszélünk, ez minden alkalommal a parancsfelületen való munkát jelenti. Ez az a bizonyos parancsjel (\$), ami nagyon sok parancshéjnal közös. Amikor rendszergazdaként jelentkezünk a rendszerbe, jobbra eltérő parancsjelet (prompt) láthatunk. Az ilyenkor használatos jelnek (#) igen sok elnevezése ismeretes: Észak-Amerikában „pound sign” (fontjel) vagy „hash mark” a neve, angol barátaink azt mondják, ez egy „octothorp”, mások tic-tac-toe táblának nevezik (magyarul pedig kettős kereszt vagy rács a neve – a ford.). Én parancsjelnek (*root prompt*-nak) fogom nevezni. Végül is nem az elnevezés fontos, csak egy olyan parancsjelet szeretnénk, amivel elkezdhetjük a gyakorlást. Ha grafikus felületen vagyunk, kattintsunk a termináblak ikonra, hogy elinduljon a terminál- (vagy héj-) program. A KDE-felhasználók egy Konsolot indítanak, míg a Gnome-felhasználók valószínűleg egy színes *xterm*-et. Ha a terminálemulátorok számát találomra próbálnánk megbecsülni, nagy valószínűséggel kevesebbre gondolnánk, mint ahányféle valójában létezik. Használható a tiszteletre méltó *Xterm*, a *Konsol*, az *rxvt* és *Eterm*, hogy csupán néhányat említsek.

Parancsok, amelyeket szeretni érdemes

date Dátum és idő megjelenítése.
who Az adott pillanatban bejelentkezett felhasználók.
whoami Ki is vagyok valójában?
tty Az adott munkaállomást azonosítja.
echo Hello, ello, llo, lo, o, o, o ... (visszhang).
finger Egy felhasználót azonosít. Megállapít róluk néhány dolgot.
last Ki jelentkezett be utoljára és még mindig bent van-e?

Állománykezelés

Hadd áruljam el a számítógépek, operációs rendszerek és az ezeket körülvevő egész iparág nagy titkát: minden dolog adat, és minden, amit a számítógépekkel teszünk, legvégső célunk az ebből nyerhető információt tekint. A fájlok tulajdonképpen raktárai ezeknek az információknak. Az adatok ügyes kezelése, karbantartása, élni és visszaélni velük – még húsz év múlva is ez lesz a számítástechnika központi kérdése. A következő dolog, amiről beszélni szeretnék, az a három fájl, ami felett a legtöbbször siklik át a tekintetünk: a szabványos bemenet, a szabványos kimenet és a szabványos hiba. E „fájlok” kezelésében szerzett jártasság bámulatos rugalmasságot fog biztosítani, amikor mindennapi munkánkat kell majd végeznünk.

ls Állományok listázása (*LiSt*)
cat Állományok összefűzése (*conCATenate*).
sort Egy állomány (vagy kimenet) tartalmának rendezése (*SORT*).
uniq Rendezés után az egyedi (*UNIQue*) sorokkal tér vissza.
wc Szavak megszámlálása (*Word Count*). A sorok, szavak és karakterek számát adja eredményül.
cp Fájlok másolása (*CoPy*).
mv Fájlok mozgatása (*MoVe*) vagy átnevezése.
rm Egy állomány eltávolítása (*ReMove*) vagy törlése.
more Lehetővé teszi nagy szöveges állományok könnyű lapozását.
less Mint a *more* parancs, de komolyabb hozzáállással.

Részlet Marcel Gagné: Linux-rendszerfelügyelet című könyvéből