

## Hogyan térjünk át Linuxra lépésről lépésre? (3. rész)

A fájlkezelés alapjai, avagy nyissuk fel a motorháztetőt!

**S**orozatunknak ebben a részében egy igazi mumushoz érkeztünk el. Talán egyik területet sem fed nagyobb homály, mint a Linux fájlkezelését. A dolog már-már érthetetlen, hiszen a történet nagyjából harminc évvel ezelőtt kezdődött. Ekkor jelentek meg ugyanis az első Unixok, amelyek már akkor is ugyanezt az elvet követték, amelyet a Linux is szinte teljes mértékben átvett és a kezdetektől fogva alkalmaz. Jelen cikkben ezt a fájlkezelési rendszert szeretném bemutatni legalább olyan részletességgel, hogy kényelmesen meg tudjuk oldani a sorozat további részei során felmerülő, fájlokkal kapcsolatos feladatokat. Mivel a témakör meglehetősen átfogó, ki szeretném ragadni belőle az alapvető részeket, amelyeken keresztül bemutatható a unixos állományelgondolás – ennek segítségével viharmentes vizek felé kormányozhatjuk rendszerünket. Ezenkívül szükségünk lesz egy-két fájlműveletet végző eljárásra annak érdekében, hogy élőben láthassuk, mi is a valós eredménye annak, amiről itt értekezünk. Kicsit tömörebben szólva: olyan kérdésekről lesz szó, amiket az eddigi használat során a SuSE grafikus felülete és maga a telepítő többnyire elrejtett a szemünk elől, ám ahhoz, hogy továbbhaladhassunk a mindennapi használat felé, feltétlenül szükségünk van a válaszokra.

### Hová lett a C meghajtóm?

...kérdesztek ijedten a frissen Linuxot telepített felhasználók. Már *Einstein* is megmondta annak idején, hogy az anyag nem vész el, legfeljebb átalakul. A mondás tökéletesen igaz a mi helyzetünkre is, a titok nyitja a unixos könyvtárszerkezet-kezelésben található. Linux alatt ugyanis a könyvtárszerkezetet megtestesítő faszkelet hagyományosan egygyökerű. Ha jobban szemügyre vesszük eddig használt rendszerünket, megfigyelhetjük, hogy ott is ugyanilyen könyvtárszerkezettel dolgoztunk, ám a kiindulópont mindig egy meghajtó volt, így tehát az ott használt fának annyi gyökere volt, ahány meghajtóval a rendszer rendelkezett. Mostani összeállításunk esetében azonban a rendszer határoz meg számunkra egy gyökeret (ez a / könyvtár), s minden későbbiekben használt hálózati vagy helyi meghajtó könyvtárszerkezete csak ez alatt a gyökér alatt fakadhat. Mindezt úgy oldjuk meg, hogy az egyes meghajtókat, amiket eddig használtunk, befűzzük (mount) az új fájlrendszer egy meghatározott pontja alá, s ezek után tartalmát úgy érhetjük el, hogy belemegyünk abba a bizonyos könyvtárba, ahová az adott meghajtót előzetesen befűztük. Belelépve a könyvtárba észre sem vesszük, hogy egy másik meghajtóra léptünk volna át, olyan, mintha csak az adott könyvtár tartalmát böngésznének. A művelet tehát a felhasználó számára teljesen átlátszó, a működés részleteit a rendszer teljesen elrejtí, így ha nem ismerjük, hogy ki mikor mit hova fűzött be, meg sem tudjuk állapítani, hogy milyen fájlrendszeren tartózkodunk éppen. Nem látjuk, hogy helyi lemezzészről vagy hálózati meghajtóról van-e szó, hajlékonylemez, PenDrive vagy CD-ROM-e az, amit olvasunk pillanatnyilag olvasunk, s azt sem tudjuk azonnal, hogy az adott eszközön milyen típusú fájlrendszer van (természetesen a csatlakoztatásnál olyan nevekkkel hivatkozunk a tartalomra,

amelyekből azonosítható, hogy melyik eszközt használjuk). Ez az egységesen kezelt felépítmény az elmúlt jó pár évben már bizonyította hatékonyságát, ám az kétségtelen, hogy először hozzá kell szokni, s csak azután élvezhetjük igazán a használatát.

### Meghajtók befűzése a rendszerbe

Az eddigi használat során ilyesmire nem volt szükségünk, hiszen a rendszer anélkül végezte el helyettünk ezt a műveletet, hogy észrevettük volna. Legelőször a telepítés során találkoztunk vele, ahol a lemezzész kialakítását végeztük. Az egyes lemezzész mellett ugyanis találtunk egy olyan jellemzőt, hogy *Csatolási pont*, amely a rendszer egy könyvtárára mutatott. Alapértelmezés szerint a windowsos lemezzész esetében a /Windows/C, D, E könyvtárba fűzte be a rendszer a régi C, D, E stb. meghajtókat. A telepítés után a munkaasztalon találtuk a kötetcímké után elnevezett meghajtó-ikonokat, amelyekre kattintva megtekinthettük a tartalmukat. Biztos vagyok abban is, hogy mindenki rájött arra, hogy a CD-meghajtó ikonjára kattintva elérte annak tartalmát, ám amikor ki akarta venni a CD-t, az semmiképp sem adta magát. Ennek az az oka, hogy a rendszerbe behelyezett minden lemezt vagy hajlékonylemezt ugyanúgy be kell fűzni, mielőtt használatba vennénk, s eltávolításuk előtt le kell választani (unmount) őket a fájlrendszerrel. A CD-t is csak azután vehetjük ki a meghajtóból, miután leválasztottuk. Ezt úgy tehetjük meg (mint ahogy sokan meg is tették), hogy a CD ikonra a jobb gombbal kattintva a menüből kiválasztjuk a *Leccsatlakoztatás* menüpontot. Előfordulhat, hogy hibaüzenetet kapunk, ami a legtöbb esetben azért következik be, mert a lemez valamelyik fájlját használjuk: épp lejátszunk egy MP3-at a CD-ről, vagy a böngészőben nyitva van egy ablak, amely a CD tartalmára mutat. Zárjuk be azokat az alkalmazásokat, amelyek valamilyen módon használják a fájlokat, majd próbálkozzunk újra. Látni fogjuk, hogy a művelet ezúttal hibaüzenet nélkül lezajlik. Létezik néhány megoldás arra nézve is, hogy ne kelljen

### Megjegyzés

A hajlékonylemez leválasztás nélkül is eltávolíthatjuk, legfeljebb néhány hibaüzenetet kapunk utána, ám akad egy nagyobb baj is: elképzelhető, hogy a ráírt adatok nem lesznek rajta a lemezen. Ennek az az oka, hogy a fájlok a lemeze írás előtt egy átmeneti tárbba kerülnek, hogy utána csoportosítva gyorsabban írassuk a lemeze, s ha leválasztás nélkül vesszük ki a lemezt a meghajtóból, elképzelhető, hogy még nem történt meg a kötegelt lemezeírás. Ha viszont kiadjuk az `unmount` parancsot, biztosak lehetünk benne, hogy a rendszer az átmeneti tárbba található adatokat is ráírja, így nem ér minket adatvesztés.

minden esetben ki-befűzögetni a CD-eket, lemezeket, ezekről a sorozat későbbi cikkeiben esik majd szó, most azonban kanyarodjunk vissza egy kicsit a befűzéshez.

## Új merevlemezem van, mit tegyek vele?

Tegyük fel, hogy a gépben található merevlemezfiókba (mobirack) új meghajtót helyeztünk, s látni szeretnénk a tartalmát. Sokat nem kell aggódnunk, hiszen a rendszer indításakor a SuSE eszközfelismerő programja felkínálja a lehetőséget, hogy az újonnan behelyezett lemezt beállítsuk. Ha igent mondunk, a rendszer elindítja a lemezszerkezelő programot, ahol új lemezszerkezetet hozhatunk létre, s a *Csatlakoztatási pont* mezőben megadhatjuk, hogy fájlrendszerünk mely pontjához szeretnénk csatlakoztatni az új lemezszerkezt. Meglévő lemezszerkezet esetében természetesen nem kell újat létrehozni. Ha például le szeretnénk másolni a hordozható merevlemezben található adatokat, amelyet épp most hoztunk az ismerősünk gépéből, az elinduló lemezszerkezelő programban csak adjuk meg a csatlakoztatási pontot a felismert lemezszerkezet mellett, s a művelet befejezése után azonnal egy, a többihez hasonló meghajtóikont kapunk az asztalon. Ezek után a lemezszerkezet elérhető.

### Egy kis háttéradat

A lemezszerkezelő valójában egy előlap (frontend) valamilyen lemezszerkezelő programhoz, és a lemezek befűzését végző `mount` rendszerhez. A meghajtóbejegyzések a `/etc/fstab` beállításfájlban találhatóak, ebbe bárki belenézhet. Rendszerindításkor a benne található bejegyzéseken végigfut a `mount` parancs, és amelyiknél engedélyezve van az önműködő befűzés, azt csatlakoztatja. A `mount` parancs természetesen kézzel is futtatható, a használata egyszerűsítve valahogy így néz ki:

```
mount -t <fájlrendszer típusa> <eszközkezelő fájl neve> <csatlakoztatási pont>
```

Az így befűzött meghajtók azonban a következő rendszerindításkor nem csatlakoznak önműködően, ehhez a bejegyzést rögzíteni kell a `/etc/fstab` fájlban.

## A könyvtárszerkezet

Nem árt, ha tudjuk, hogy mit hol találunk. Ilyen megfontolásból szeretném a főbb könyvtárak rendeltetését bemutatni – természetesen a teljesség igénye nélkül.

**/bin:** itt található az alaprendszer indítható állományai a hétköznapiakban leggyakrabban használt parancsok (például: `date`, `chmod`, `chown`).

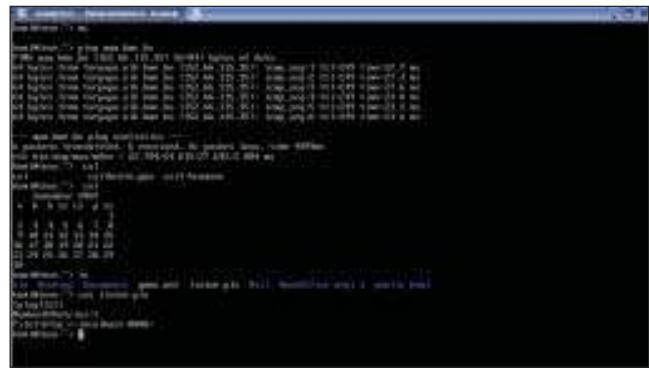
**/boot:** ebben a könyvtárban vannak a rendszerindításhoz nélkülözhetetlen állományok, köztük maga a rendszermag – a rendszertöltő állományai, beállításai is itt lelhetők fel.

**/dev:** ez a könyvtár csupa különleges fájlt tartalmaz, itt vannak ugyanis az egyes eszközök képviselő fájlok, a `dev` könyvtárban belül kategóriákba rendezve; ha valamelyik alkalmazás meg akar szólítani egy eszközt, akkor valójában az eszköz kezelőállományához fordul, amelyeket egytől egyig itt talál meg.

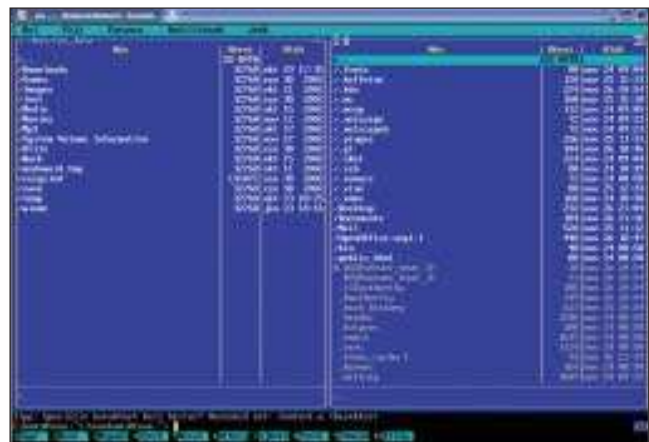
**/etc:** szinte kivétel nélkül itt helyezkednek el a rendszerkövetkező alkalmazások beállításfájlljai; a legtöbb esetben, amikor valamit testre kell szabnunk, ebben a könyvtárban kell keresnünk az állítási lehetőségeket.

**/home:** s különböző felhasználók saját könyvtárai találhatóak itt, mindenkihez tartozik egy; ezen a könyvtáron belül minden felhasználó azt csinálja, amit csak akar – fájlokat hozhat létre, törölheti őket stb.

**/lib:** más alkalmazások számára különböző szolgáltatásokat kínáló állományokat találhatunk itt; a segítségükkel elérhető,



1. kép Konsole – A KDE X-terminálja



2. kép Így néz ki a Midnight Commander

hogy ugyanazokat az előre megírt eljárásokat több alkalmazás is használja; nagyon hasonlóak a Windowsnál megszokott DLL-állományokhoz

**/proc:** ez egy látszólagos fájlrendszer; gépünk alkatrészeiről találunk itt mindenféle adatot, a legtöbb alkalmazás is innen olvassa ki a gép tulajdonságait; mindegyik itt található fájl egy szöveges állomány, amely minden rendszerindítás során újra létrejön.

**/tmp:** a rendszer működése során keletkező átmeneti állományok tárolására szolgál, így rendet lehet tartani az egyes alkalmazások háza táján, mivel mindenki egy közös helyre szemetel, amit a rendszer hatékonyan karbantarthat.

**/usr:** itt található a felhasználói alkalmazások állományai; ha egy általános célú csomagot telepítünk, annak a célterülete szinte minden esetben e könyvtáron belülre mutat.

## Jogosultságok

Ezennel a fájlrendszerrel tisztában vagyunk annyira, amennyire nekünk, kezdő felhasználóknak szükségünk van, ám most egy másik fontos tényezőt kell szemügyre vennünk, amely azért fontos, hogy kellő hozzáértéssel hozzunk létre fájlokat, mozgassuk, másoljuk őket stb.

Mint már említettem, a Linux többfelhasználós rendszer, ami azt jelenti, hogy egy időben egyszerre többen is használhatják – akár a helyi gépen, akár valamilyen terminálon vagy hálózati kapcsolaton keresztül. Ahhoz azonban, hogy akár a véletlen, akár a szándékos károkozást vagy adatlopást megakadályozzuk, gondoskodni kell arról, hogy mindenki csak olyan állományokhoz férjen hozzá, amelyek a rendeltetészerű használathoz szükségesek. Egy kicsit

bővebben ez azt jelenti, hogy egy átlagos felhasználónak nem kell látnia mások állományait, neki ahhoz semmi köze, és a rendszer működéséhez szükséges állományokat sem szabad megváltoztatnia! E feladatok megoldására létezik az úgynevezett jogosultságrendszer, amely szinte egyidős a fent vázolt fájlrendszerrel, fájlstruktúrával, és egyébként szorosan kapcsolódik hozzá.

## Nézzük, hogy is van ez!

A rendszer összes állományának – beleértve a befűzött meghajtókon találhatóakat is – vannak bizonyos jellemző tulajdonságai. Itt természetesen nem a méretre és a formátumra gondolok, hanem azokra a pluszadatokra, amelyek azt az ismeretet hordozzák, hogy egy fájljal egy adott felhasználó milyen műveleteket végezhet. Ezek közül számunkra most három oktális szám válik fontossá. E három szám alapján ismeri a rendszer a felhasználók jogosultságait az adott fájlra, s különböztet meg három felhasználói réteget: a tulajdonost, a csoportot és mindenki mást. Mindhárom rétegre vonatkoznak bizonyos jogosultságok (mindegyik esetében ugyanazok közül lehet választani), s a hozzáférni kívánó felhasználó alapján dől el, hogy ki mit csinálhat, kire mi vonatkozik. Ez egy kissé zavaros, ugye? Ne aggódjunk, hamarosan mindenki megvilágosodik!

### Az a mágikus oktális szám

Ez a nyolc variáció alkotja a Unix-rendszerek állományain végrehajtható műveletek halmazát. A dolog talán a kettes számrendszer használatával érthető meg legjobban. Minden helyi értéknek egy művelet felel azokon a helyeken, ahol a művelet engedélyezett, a helyi értéken képviselt értékek összeadódnak. Alapvetően háromféle jogosultság van: olvasás (r), írás (w) és futtatás/böngészés (x). Ez az utóbbi jogosultság fájljok esetében futtatásra vonatkozik, könyvtárak esetében pedig arra, hogy az oda belépő felhasználó lekérhesse a könyvtár tartalmának a listáját. Ezeket egymás mellé írva azt kapjuk, hogy: `rwX`. Ez maga az oktális szám emberi formátumban való megtestesülése. Az első betű helyi érték szerint 4-et, a második 2-t, a harmadik 1-et jelent. Ezek után, ha egy állományra az adott jogosultságcsoportból olvasási és írási jog van, azt egy 6-os szám jelképezi, ha csak olvasható, azt egy 4-es. Ilyen számból három van, s ez mondja meg az egyes csoportokhoz kötött jogosultságokat, amiket aztán a rendszer értelmez.

Biztosan felmerült már néhány emberben, hogy mit jelent a futtathatóság. Ezekben a rendszerekben nem a kiterjesztés alapján dől el, hogy egy adott állomány indítható-e, hanem ez is a fájl egy tulajdonsága.

Példaképpen hozzunk létre egy fájlt, majd mindenkinek adjunk hozzá olvasási jogot, magunknak pedig írási/olvasási. Ehhez indítsunk el egy parancssort (például a Konsole a KDE menü mellett a panelen), majd adjuk ki a következő parancsokat:

```
cd ~
touch probafajl
chmod 644 probafajl
```

Ez a bizonyos `chmod` parancs szolgál arra, hogy a segítségével megváltoztassuk egy fájl jogosultságait. Legegyszerűbb használatában a folyamatosan emlegetett három oktális szám egymás után írásával adhatunk újabb jogokat a fájljal, könyvtárak.

Haladjunk az elejéről, s nézzük a tulajdonos jogait! Minden állománynak, könyvtárnak van egy tulajdonosa, aki egy létező felhasználó a rendszerben (a tulajdonos általában az, aki az adott fájl létrehozta). Az első oktális szám rá vonatkozik. A fájl tulajdonosa a többi jogosultságtól függetlenül azt tehet az állománnyal, ami itt engedélyezve van.

A következő szám a csoport jogaira vonatkozik. Minden fájlhoz tartozik ugyanis egy felhasználói csoport, mint ahogy minden egyes felhasználó is benne van egy neki megfeleltetett csoportban (például a felhasználók csoportjába tartozik Józsi). Ha egy olyan felhasználó akar hozzáférni a fájlhoz, aki az adott csoportban benne van, az az oktális számnak megfelelő műveleteket végezheti el a fájljal.

Végül az utolsó szám az egyéb, avagy „kívülálló” kategóriára vonatkozik. Mindenki, aki nem tagja a csoportnak és nem is tulajdonosa az állománynak, az itt lefektetett jogokat birtokolja. Igen ám, de ezek közül egy jogosultság csakis akkor érvényes, ha vagy a tulajdonosi, vagy a csoportra vonatkozó ugyanezen jog engedélyezett.

Ezeket a jogosultságokat csakis a rendszergazda (root), vagy az adott fájl/könyvtár tulajdonosa változtathatja meg – ha ez nem így volna, az egész védelem nem érne semmit, hiszen bárki felhatalmazhatná magát egy fájl használatára.

Ide kapcsolódik még az a kérdés is, hogy hogyan adható át egy fájl tulajdonjoga egy másik felhasználónak vagy hogyan változtatható meg a fájlhoz rendelt csoport? A válasz felelős, ugyanis a tulajdonjog senkire sem ruházható át, a hozzárendelt csoportot viszont a fájl tulajdonosa megváltoztathatja. Ezek a szabályok nem vonatkoznak azonban a rendszergazdára: ő bárkinek átadhatja bárki fájljait és bármelyik fájl csoportját megváltoztathatja.

Minden ilyen változtatás a `chown` paranccsal történik, amelynek használata a következő:

```
chown: <felhasználó>:<csoport>
-><fájl vagy könyvtár>
```

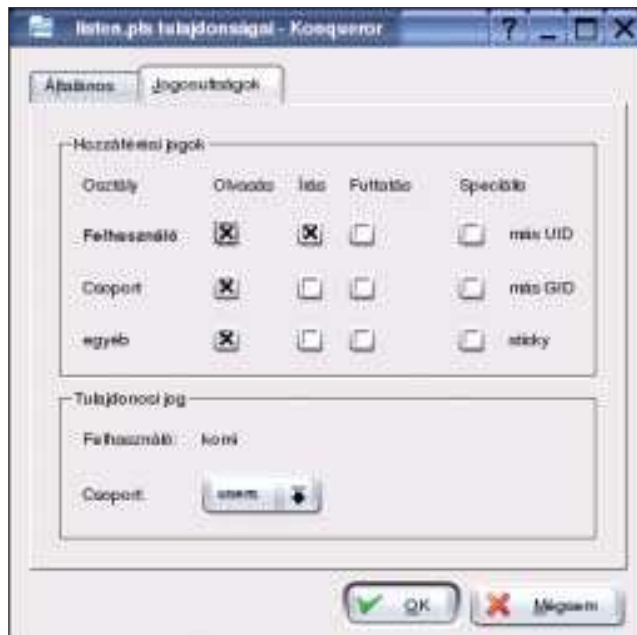
## Egy kis könnyítés...

Ez bizonyára soknak tűnt így elsőre, de azzal szeretném megnyugtani a kedves felhasználókat, hogy ezt azért nem kell állandóan használnunk, de egy „Hozzáférés megtagadva” üzenet esetén azonnal tudjuk a hiba okát, s orvosolni tudjuk azt – rendszerismeretünkhöz elengedhetetlen ezeknek a tényeknek az ismerete. Azért nem kell mindent kézzel beállítani, az állományoknál a grafikus felület is kínál számunkra beállítási lehetőségeket. Kattintsunk rá valamelyik meghajtóikonra, majd válasszunk egy tetszőleges fájlt vagy könyvtárat, s kattintsunk rá az egér jobb gombjával, majd válasszuk a legelső **Tulajdonságok** menüpontot. Eredményül egy panelt kapunk, amelyen a második fül a jogosultságokra vonatkozik. Itt a fent tárgyaltakat állíthatjuk be igen kényelmes módon, feltéve, ha van hozzá jogunk. Emellett van néhány parancssoros eszköz, amelyek nemcsak a jogosultságkezelést könnyítik meg, de a fájlműveleteket is segítik úgy általában. Ilyen eszköz a közismert Midnight Commander, amelynek ismertetése előtt azonban egy kis kitérőt kell tennünk.

## A parancssor használata

A Unix-rendszerekről általában elmondhatók, hogy parancssorban rendkívül erősek, tehát igen hatékonyan lehet használni a rendszerünket szöveges utasításokon keresztül is. Ha jobban belegondolunk, tulajdonképpen a legtöbb grafikus szolgáltatás is ezekre a parancssori elemekre épül, azzal a különbséggel,





3. kép Jogosultságok megtekintése a Konqueror segítségével

hogyan ott fizikailag nem mi gépeljük be a parancsokat.

Az egész folyamatot tekintve a dolog úgy zajlik, hogy bejelentkezéskor mindenkihez hozzárendelődik egy héj (shell), s utána ezen a héjon belül tevékenykedhetünk, kihasználva a rendszerünk nyújtotta szolgáltatásokat. (Ez a fajta héj-hozzárendelés természetesen a távoli felhasználás során is érvényes.) A grafikus munkakörnyezet ezt ugyan elrejtja előlünk, de a háttérben ott van, olyannyira, hogy egy parancssoros ablak formájában akár elő is hozhatjuk. Ebből egyébként tetszőleges számút futtathatunk egymás mellett, mindehhez csak az kell, hogy valahonnan (a bejelentkezéskor kapott héjből) el tudjuk indítani őket. Számunkra az alapértelmezett program, amely biztosítja a parancssort, nem más, mint a KDE egyik alkalmazása, a Konsole, amelyet a panelen a KDE menü mellett találunk. Elindítva egy a windowsos parancssornak megfelelő linuxos parancsértelmezőt kapunk, ám ennek az erejével, tudásával össze sem hasonlítható redmondi társa (ennek többnyire hagyomány- és szemléletbeli okai vannak). Annak érdekében, hogy mindez hihető legyen, sorozatunk egy későbbi részében egy teljes cikket fogunk áldozni a linuxos parancssor, azon belül is a bash nevű parancsértelmező program bemutatására. Egy valamirevaló linuxos szakember élete nagy részét ilyen parancsértelmezők között tölti, s legfeljebb a weblapokat nézegeti grafikusan – nem árt tehát, ha fokozatosan hozzászokunk (eleinte csak a gondolathoz) a parancssorok jelenlétéhez, aktív használatához.

### A Midnight Commander

Ez egy klasszikus, meglehetősen idős, karakteres módú fájlkezelő alkalmazás, nevéből kikövetkeztethetően a DOS-os Norton Commanderhez, a DOS Navigatorhoz vagy éppen a windowsos Total Commanderhez hasonló feladatot lát el. Tisztában vagyok vele, hogy nincs az az ember, aki ne használna ilyen osztott paneles fájlkezelő megoldást. Én Linux alá ezt a programot javaslom, hiszen kis memóriáigénye és karakteres módú megjelenése miatt bárhol alkalmazható. Mindenekelőtt telepítsük az alkalmazást, amelyhez indítsuk el

a YaST vezérlőközpontot, s ott a *Szoftver* kategóriából választjuk a *Szoftver telepítése és eltávolítása* lehetőséget (mint azt az előző cikk csomagkezelésről szóló részében megtanultuk). Itt a csomagcsoportokon belül a *File Utilities* kategóriában található *mc* csomagra lesz szükségünk. Ha valamilyen okból kifolyólag nem találunk meg, válasszuk a keresés lehetőséget, s keressünk rá az *mc* kifejezésre. A telepítés után nincs is szükségünk másra, mint elindítani egy parancssort, s kiadni az *mc* parancsot.

Indítás után elénk tárul a már jól megszokott felépítésű fájlpanel-képernyő, így már talán otthonosabban érezhetjük magunkat a rendszerünkben. A részletes megismerést az olvasóra bízom, itt most csak néhány témába vágó szolgáltatását szeretném kiemelni. Először is itt van ugye a fájlműveletekhez nyújtott támogatás (másolás, törlés, mozgatás stb.), másodsor pedig a segítségével viszonylag átlátható formában állíthatjuk be a fájlok jogosultságait.

Tegyük is egy próbát, s keressük meg az előbbi példában létrehozott *probafajl* nevű állományunkat, álljunk rá a kurzorral, majd nyomjuk le a CTRL+X billentyűkombinációt, ezután nyomjuk le a C billentyűt. Azonnal láthatjuk a jogosultságkezelő ablakot, ott pedig a jól megismert három kategóriát: *tulajdonos*, *csoport*, *mások*. Mindegyik mellett ott az emlegetett ír, olvas, futtat/belép jogosultság, a mi dolgunk csak annyi, hogy a szóközbillentyű segítségével beikszeljük azt a jogot, amire szükségünk van, így nem kell binárisan vagy épp oktálisan számolgatnunk a beállítandó jogosultságnak megfelelő kódot.

Ha ezzel megvoldnánk, maradjunk továbbra is a megadott állományon, s most nyomjuk le a CTRL+X kombinációt, majd az O billentyűt. Ekkor a tulajdonlást kezelő ablak jelenik meg, ahol tulajdonosként a listából kiválasztva egy másik csoportot rendelhetünk a fájlhoz, rendszergazdaként pedig ugyanezen a módon a fájl tulajdonosát is megváltoztathatjuk.

### Összegzés

Most végre elmondhatjuk magunkról, hogy valamelyest konyvitunk a linuxos fájlkezeléshez, s ha nem is ismerünk minden apró trükköt, a mindennapi feladatok során már egészen jól elboldogulunk ezekkel az ismeretekkel. Nem kell azonban minden bitet részletesen ismernünk, a gyakorlatban az itt elmondottakra kezdőként olyan helyzetekben lehet szükségünk, ha mondjuk kézzel, s nem csomagból telepítünk egy alkalmazást, s esetleg olyan könyvtárba kell azt másolni, ahová csak a rendszergazdának van írási joga. Ebben az esetben egyetlen teendőnk van: be kell jelentkezni rendszergazdaként, s úgy elvégezni az előírt műveletet. Hasonló gond léphet fel bizonyos fájlok létrehozásánál törlésénél, vagy egyéb olyan alkalmazásoknál, amik egy olyan fájl szeretnének használni, amelyre nincs is olvasási joguk. Ekkor, ha ismerjük a fájlt, egy huszárvágással megoldhatjuk a problémákat, s idővel a parancssor egyre fokozódó használatával eljutunk oda, amikor már mindenféle külső segítség nélkül, csupán az addig összegyűjtött ismeretanyag felhasználásával képesek leszünk minden nehézséget elhárítani.

*Kellemes gyakorolást!*



**Komáromi Zoltán**

(komi@kiskapu.hu)

23 éves, a BME hallgatója,

mellette PHP-programozóként dolgozik.

Kedvenc területe a multimédia.