

# Helyspórolás mini-HOGYAN

Írta Guido Gonzato, <<mailto:guido@ibogeo.df.unibo.it>>

v1.0.1, 1999. április 7.

Ez a mini-HOGYAN elmagyarázza, hogyan lehet a lehető lehető legkisebb helyre beszorítani egy Linux rendszert. Főleg notebook felhasználóknak hasznos.

## Contents

<b>1 Bevezetés</b>	<b>1</b>
<b>2 Szoftverkövetelmények</b>	<b>1</b>
<b>3 Az eljárás</b>	<b>2</b>
3.1 A kernelforrás eltávolítása . . . . .	2
3.2 Az alkalmazások ritkítása . . . . .	2
3.3 A bináris fájlok méretének csökkentése . . . . .	3
3.4 Tömörítés az upx programmal . . . . .	3
3.5 Gzippejük, amit lehet . . . . .	3
<b>4 Egy valódi példa</b>	<b>4</b>
<b>5 Befejezés</b>	<b>4</b>
5.1 Szerzői jogok . . . . .	4
5.2 Jogi felelősség . . . . .	4
5.3 Magyar fordítás . . . . .	5

## 1 Bevezetés

Kaptam egy notebookot előtelepített Windows 95 operációs rendszerrel, és felraktam rá egy Linuxot is. Összenyomtam a Windows partícióját 500 MB-ra, 240 MB maradt a Linux partíció számára. Ez elég kevésnek tűnik, de mégis sikerült egy eléggé teljes Red Hat 4.1 alapú Linux rendszert rátennem, némi trükközés árán, amivel sok helyet spóroltam meg.

Ha helyre van szükségünk, a következő fejezetekben leírt tanácsok megfogadásával elég sok lemezhelyet szabadíthatunk fel. Felhívom a figyelmet, hogy senki ne engem okoljon, ha valami tönkremegy. Jópár olyan programot kell majd használni, amelyek nálam klasszul működtek, de természetüknél fogva veszélyesek. Én megmondtam...

## 2 Szoftverkövetelmények

Szükség lesz:

- egy működő Linux rendszerre (bármelyik verzió megfelel);
- a széles körben elterjedt **gzip** tömörítőprogramra, vagy a **bzip2**-re, ami letölthető a <ftp://sunsite.unc.edu/pub/Linux/utils/compress> webhelyről, és jobban tömörít, mint a **gzip**, de lassabb és rengeteg memóriát használ;
- az **upx** végrehajtható állományokat tömörítő programra, amelynek a honlapja a <http://www.nexus.hu/upx/> és a <http://wildsau.idv.uni-linz.ac.at/mfx/upx.html> webhelyeken található;
- és a <ftp://sunsite.unc.edu/pub/Linux/libs/compression> webhelyről letölthető **zlibc** csomagra, amelyet **zlibc-X.X.tar.gz**-nek hívnak, ahol **X.X** a legújabb verzió.

Vannak más végrehajtható állományokat tömörítő programok is. Az egyik a **gzexe** (felejtjük el), egy másik a **tcx**, ami talán a legjobb volt, mielőtt az **upx** megjelent, egy harmadik a **tzx**, ami elvileg jobb, mint a **tcx**. A helyzet az, hogy mindent tönkretett, amikor kipróbáltam egy tartalék számítógépen, bár nem vagyok biztos benne, hogy mindent jól csináltam. A legjobb, ha az **upx**-et használjuk, mert megbízható, hatékony és kényelmes.

Léteznek olyan kernel foltok, amelyek Stacker-szerű röptömörített fájlrendszert hoznak létre, de jelen pillanatban egyik sem a stabilitásáról és a megbízhatóságáról híres. Biztos, ami biztos, tartózkodjunk ezek használatától.

## 3 Az eljárás

### 3.1 A kernelforrás eltávolítása

A kernel forrása több mint 20 MB helyet foglal, megfontolhatjuk a letörlését. Ha így döntenénk, fordítsunk egy kernelt egyszer és mindenkorra, ami a gépünkre kell, és szabaduljunk meg a forráskódtól. Azért legyünk óvatosak.

A kernelforrás eltávolítása *nem* ajánlatos, ha nem vagyunk biztosak benne, hogy a gépünk megfelelően be van konfigurálva. Emellett a kernel **#** fejlécfájljai szükségesek a C programok fordításához. Gondoljuk meg kétszer!

Ha a törlés mellett döntünk, *akkor se* töröljük le az **include/linux** könyvtárstruktúrát, hacsak nem vagyunk biztosak benne, hogy soha nem fogunk alkalmazásokat fordítani a gépen.

### 3.2 Az alkalmazások ritkítása

Döntsük el, hogy melyik programra van *igazán* szükségünk. Néhány alkalmazás felesleges lehet. Például: biztos, hogy kell az **emacs**? Használhatjuk helyette a **jed** szövegszerkesztőt.

Rajtunk múlik, hogy mit tartunk meg. Néhány általános ötlet:

- A **gcc** elég nagy csomag. Ez szükséges a kernel újrafordításához, és minden olyan alkalmazás lefordításához, amelynek nincs előre lefordított Linux bináris változata. Természetesen akkor is szükségünk lesz rá, ha a saját C vagy Fortran (**f2c**-vel vagy **g77**-tel) programjainkat akarjuk lefordítani, törlés előtt gondoljuk át, hogy fogunk-e valaha programokat fejleszteni.. Más fordítóprogramok, pl. az **lcc** elég jók, de nem érik el a **gcc** szintjét. Azt javaslom, tartsuk meg!
- Az **X11** szörnyen nagy, de jó ha megvan. Ha nem tudunk megglenni nélküle, legalább próbáljuk a méretét a lehető legkisebbre leszorítani: csak a megfelelő **X** szerver program, csak egy egyszerű ablakkezelő, csak egy **xterm**, ne telepítsük a 100 dpi-s betűkészleteket, stb.

- A  $\text{\TeX}$  és kiegészítő csomagjai szintén nagyon nagyok. Ha nem használjuk a  $\text{\LaTeX}$  csomagot, és megelégszünk a plain  $\text{\TeX}$  csomaggal, sok helyet spórolhatunk. Nem kell használnunk az X11 alatti nézegetőket (`xdvi` és `ghostview`), mert használhatjuk a `dvitty`, a `dvivga` és a hasonló programokat. Ha nincs szükségünk az X11 alatti nézegetőkre, akár maga az X11 is feleslegessé válhat.
- A játékok sosem létfontosságúak.

### 3.3 A bináris fájlok méretének csökkentése

Kezdjük el csökkenteni a bináris fájlok méretét! Lépünk be az `/usr/bin` könyvtárba, és adjuk a következő parancsot:

```
machine:/usr/bin# strip *
```

Ez eltávolítja a bináris fájlokba beágyazott szimbólumokat. Ismételjük meg ezt a lépést az `/usr/X11R6/bin/` könyvtárban is, és az összes olyan könyvtárban, ahol végrehajtható fájlok lehetnek. Ne feledjük megkeresni a  $\text{\TeX}$  és a `gcc` binárisait. Ha nem akarjuk tönkretenni a rendszerünket, akkor *ne* futtassuk ezt a parancsot az `/sbin`, a `/bin`, vagy az `/usr/sbin/` könyvtárakban!

### 3.4 Tömörítés az upx programmal

Telepítsük az `upx` programot, és olvassuk el a dokumentációját. Utána lépünk be az `/usr/bin` könyvtárba és adjuk ki az `upx *` parancsot. Ez minden végrehajtható fájlra tömörít, a `suid`-osakat is (a `tcx` ezt nem tenné). Ismételjük meg ezt a lépést a többi fent említett könyvtárban.

Ne feledjük tömöríteni a végrehajtható fájlokat, amikor új csomagot telepítünk!

### 3.5 Gzipeljük, amit lehet

Van egy csomó fájl, amit egyszer és mindenkorra tömöríthetünk. Kezdjük az `/usr/doc/` könyvtárral! Lépünk bele és adjuk ki a következő parancsot:

```
machine:/usr/doc# find . -type f -exec gzip -9 {} \; 2> /dev/null
```

Ne feledjük tömöríteni a dokumentációt, amikor új csomagot telepítünk!

Ismételjük meg ezt a lépést a  $\text{\TeX}$  dokumentációját tartalmazó könyvtárra (az én rendszeremen ez az `/usr/lib/texmf/texmf/doc/`). Ha *teljesen biztosak* vagyunk benne, akár le is törölhetjük ezeket a könyvtárakat.

Telepítsük a `zlibc` csomagot és fordítsuk le. Az olyan rendszereken, mint az enyém, a fordítás megszakad, mert a fordító nem találja a (statikus) `libc` könyvtárat. Semmi gond, fogunk találni egy `uncompress.o` nevű fájlt, és ez az, amire szükségünk van. Másoljuk ezt a fájlt az `/usr/local/lib/` könyvtárba, és adjuk hozzá a következő sort az `/etc/profile` fájlhoz:

```
export LD_ELF_PRELOAD=/usr/local/lib/uncompress.o
```

Ezek után nemcsak a dokumentációt tömöríthetjük a `gzip` programmal, hanem az adatfájlokat is, az őket használó alkalmazások mégis tudják majd használni őket. Elvileg ennek trükknek működni kellene a legtöbb alkalmazásra, de gyakorlatilag ez nem feltétlenül van így. Én nem értem el különösebb sikert ezzel a trükkel.

## 4 Egy valódi példa

Az alábbiakban olvasható, hogy hogyan alkalmaztam a fent említett eljárásokat a saját gépemre. A kezelés előtt a `df` szerint 398.798 db. 1024 bájt méretű blokkot használtam.

- Nem távolítottam el a kernelforrást és a kernel fejlécfájljait.
- eltávolítottam egy csomó alkalmazást és az összes játékot, de meghagytam az X11-et, az X11 fejlesztőkészletet, a C és Fortran fejlesztőkészletet, a Tcl/Tk-et, a hálózati segédprogramokat és még néhány egyéb alkalmazást. A `df` szerint ekkor 244.668 blokk volt foglalt.
- Lefuttattam az `upx` programot az `/usr/bin`, az `/usr/X11R6/bin`, az `/usr/lib/texmf/bin/i586-linux` és az `/usr/lib/gcc-lib/i386-linux/2.7.2.1` könyvtárakban. 226.270 foglalt blokkra csökkent a felhasznált terület.
- Tömörítettem a dokumentációt az `/usr/doc` és az `/usr/lib/texmf/texmf/doc` könyvtárakban. 198.745 foglalt blokkra csökkent a felhasznált terület.

Összefoglalva, 398.798 blokkal kezdtem, és 198.745 blokkal fejeztem be. Gondoljunk arra, hogy mennyi mindent tárolhatunk azon a 200.000 megspórolt blokkon! Még több helyet felszabadíthattam volna, ha a `bzip2` programot használom a `gzip` helyett.

Átlagos esetben, ha az elejétől fogva óvatosak vagyunk, és csak a szükséges alkalmazásokat telepítjük, a végrehajtható állományok és a dokumentáció tömörítése kb. 20 MB helyet szabadít fel. Egy notebook gépen ez "életet menthet".

## 5 Befejezés

### 5.1 Szerzői jogok

A Linux HOGYAN dokumentumok szerzői joga a megfelelő szerzőké, hacsak másképp nem rendelkeznek. A Linux HOGYAN dokumentumokat szabad részben vagy egészben másolni és terjeszteni bármilyen fizikai vagy elektronikus adathordozón, de ezt a szerzői jogokra vonatkozó megjegyzést minden másolathoz mellékelni kell. Az üzleti célú terjesztés megengedett és bátorított, de a szerző szeretne tudomást szerezni minden ilyen terjesztésről.

Minden Linux HOGYANon alapuló fordítás, származtatott mű vagy összefoglaló munka ezen szerzői jogi kötelek alá kell essen. Ez azt jelenti, hogy nem szabad olyan HOGYANon alapuló művet létrehozni, amely további megkötéseket tartalmaz a terjesztésre. Bizonyos feltételek esetén ezek a szabályok figyelmen kívül hagyhatóak, erről a Linux HOGYAN koordinátor ad felvilágosítást a lent olvasható címen.

Röviden megfogalmazva, szeretnénk, ha az információ elterjedne minél több csatornán, de szeretnénk megtartani a HOGYAN dokumentumok szerzői jogát, és szeretnénk értesülni bármilyen terjesztési tervről.

Kérdésekkel a Linux HOGYAN koordinátorát, Tim Bynumot lehet megkeresni a `linux-howto@metalab.unc.edu` drótpostacímen.

### 5.2 Jogi felelősség

A Helyspórolás mini-HOGYANt Guido Gonzato írta ( [<REMOVE\\_MEguido@ibogeo.df.unibo.it>](mailto:REMOVE_MEguido@ibogeo.df.unibo.it) ).

Erre a dokumentumra nem vonatkozik semmilyen garancia. Nagyon igyekeztem, hogy hibátlan legyen, de mindenki csak a saját felelősségére használhatja az itt közölt információkat. Semmilyen esetben sem lehet engem felelőssé tenni az itt leírtak felhasználásából eredő károkért.

A visszajelzéseket örömmel veszem. Bármilyen kérdést, kérést, javaslatot szívesen meghallgatok.

Élvezzük a Linuxot és az életet!

Guido =8-)

### 5.3 Magyar fordítás

A magyar fordítást *Tímár András* <[mailto:timar\\_a@freemail.hu\\_NO\\_SPAM](mailto:timar_a@freemail.hu_NO_SPAM)> készítette. A lektorálást *Daczi László* <[mailto:dacas@freemail.hu\\_NO\\_SPAM](mailto:dacas@freemail.hu_NO_SPAM)> végezte el (2002.11.11). Bármilyen fordítással kapcsolatos észrevételt a *linuxhowto@sch.bme.hu* <[mailto:linuxhowto@sch.bme.hu\\_NO\\_SPAM](mailto:linuxhowto@sch.bme.hu_NO_SPAM)> címre küldjete. A dokumentum legfrissebb változata megtalálható a *Magyar Linux Dokumentációs Projekt* <<http://tldp.fsf.hu/index.html>> honlapján.