

# Szövegfájlból sorok törlése gyorsan, egyszerűen.

PCLinuxOS Magazine - 2025. április

Írta: Paul Arnote (parnote)

Nemrég (úgy február vége felé, vagyis tényleg nem rég), nekiláttam az „utazósnak” nevezett laptopom frissítésének. Ez egy kis Lenovo Thinkpad X230. Amikor még dolgoztam, a hátizsákomban cipeltem.

A laptopon PCLinuxOS Xfce futott az első telepítése, azaz úgy öt vagy több évvel ezelőtt óta, nagyon jó szolgálatot tett nekem. Amikor először telepítettem a PCLinuxOS-t erre a gépre, egyszerűen csak elfogadtam a régi telepítő által adott összes „alapértéket”. Ebbe beletartozott a root könyvtár „alapbeállítás” szerinti 20 GB-ra korlátozott mérete.

Úgy gondoltam, megfelelő lesz.

Tévedtem.

A régi telepítő alapértelmezett értékeit azért fogadtam el, mert az utazó laptopnak „picinyke” SSD-je volt. A benne lévő SSD csak 128 GB méretű volt, amit én a használatra kényelmesnél sokkal kisebbnek éreztem. Mindig tartottam attól, hogy gyorsan megtöltöm az SSD-t. Az összes többi PCLinuxOS-rendszeremen 30 GiB, vagy 40 GB méretű root partíciót készítettem.

Egy nap, amikor még dolgoztam (2022 végén mentem nyugdíjba), a munkahelyemen nekifogtam az úti laptopom frissítésének. Megkaptam azt az üzenetet, amitől tartottam, hogy a root partíció megtelt.

Nekiláttam hát az úti laptopomon igazából szükségtelen „dolgok” eltávolításának. Sikerült annyi helyet felszabadítanom, hogy a frissítés lefusson.

Ezután el is felejtettem az esetet ... egészen addig, amíg február végén meg nem ismétlődött. Minthogy addig az aktuális frissítések probléma nélkül lefutottak, a



Temel készítette kép a Pixabay-ről

munkahelyen tapasztalt gond feledésbe merült. Mivel a 2022-ben tapasztalt hiba idején minden szükségtelen dolgot eltávolítottam, így gyakorlatilag nem maradt semmi, amit most törölhettem volna, hogy további helyet teremtsék. Bosszúsan beleástam magam és a végén az SSD-n lévő rendszert tönkretettem.

Nem titok, hogy úgy 16 éve ülök a „szerkesztői” székekben. Ezek alatt az évek alatt a PCLinuxOS szinte minden aspektusáról írtam már. Emellett nem vagyok kezdő, ami a PCLinuxOS telepítését illeti, sok-sok alkalommal csináltam már korábban. Nem csupán a saját számítógépeimen, hanem a barátaim és a család számítógépein is. Vagyis, nem túlzás állítani, hogy elég jól eligazodom a PCLinuxOS telepítésében.

Majdnem azóta, amióta csak megvan az utazós laptopom, mindig terveztem, hogy új SSD-t rakok bele. Ám, valamilyen érthetetlen okból (számomra is érthetetlen), az összes eddig gyűjtött tapasztalatomat sutba dobva elkövettem a legbördületesebb kezdő hibát. Akár egy teljesen kezdő, egyszerűen fejest ugrottam a dologba.

## Szövegfájlokból sorok törlése gyorsan, egyszerűen

Hadd emlékeztessenek néhány dologra, Először, a partíciók átméretezése előtt készíts biztonsági másolatot a fájljaidról. Most, mivel ezen az úti laptopon semmi olyan nincs, ami nélkül ne tudnék meglenni, átugrottam ezt a lépést. Higgs nekem, ha azt mondom, hogy az adatvesztés veszélye partíció átméretezésekor NAGYON IS valós. És nem ez hiúsította meg a próbálkozásaimat. Nem. A második dolog, amit tudni kell, hogy a partíciók átrendezését (GParted-del) CSAK Live-futás alatt szabad csinálni ..., amit nem tettem. A végén a fájlrendszer helyett az SSD-en egy nagy katyvasz kaptam. (Semmi gond, újraformázom az SSD-t és másra fogom felhasználni.)



*OpenClipart-Vectors képe a Pixabay-ről*

Szóval az lett a vége, hogy a pici SSD-t egy nagyobbra kellett cserélnem. Sokkal korábban, amikor szert tettem a laptopra, akkor kellett volna megtennem. Amikor jó áron látok SATA3-as SSD meg szoktam venni azokat (persze, amikor van rá fedezet), így van pár „használatra váró” darab. Fogtam egy 512 GiB-es SSD-t és leváltottam vele a 128 GiB-es SSD-t. Ez a része nagyon könnyen ment.

Az új SSD-re csináltam egy 200 MiB-es partíciót az EFI boot-nak (a laptopban van EFI, de van annyira régi, hogy NE LEGYEN Secure Boot benne), egy 50

GiB-osa root partíciót, egy 8 GiB-es swap partíciót és a „maradék” helyet a /home partíciómnak hagytam. A telepítés elég simán ment, egy két apróbb akadálytól eltekintve, amiket gyorsan leküzdöttem.

```
<action>
  <icon>terminal</icon>
  <name>Open Terminal</name>
  <submenu></submenu>
  <unique-id>1694383311350259-1</unique-id>
  <command>exo-open --working-directory %f --launch TerminalEmulator</command>
  <description>Open Terminal window here</description>
  <range></range>
  <patterns>*</patterns>
  <directories/>
</action>
<action>
  <icon>gksu-root-terminal</icon>
  <name>Open Root Terminal</name>
  <submenu></submenu>
  <unique-id>1694383311350271-2</unique-id>
  <command>gksu xfce4-terminal</command>
  <description>Open Root Terminal window here</description>
  <range></range>
  <patterns>*</patterns>
  <directories/>
</action>
```

*Részlet az uca.xml fájlomból*

Az sem titok, hogy nagy Xfce-rajongó vagyok. RENGETEG Thunar Egyéni műveletem van, amiket szeretek az ÖSSZES Xfce-t futtató gépemre felrakni. Nos, a Thunar uca.xml fájlja az, ami tárolja a trükköket, lehetővé teszi a Thunar Egyéni műveletei futtatását. Valamikor, talán az Xfce 4.14-től kezdődően (pontosan nem emlékszem), az Xfce-fejlesztők minden Thunar felhasználói műveletbe „beillesztettek” egy új adatmezőt. Az adatmezőt <unique-id> ... </unique-id>-nek hívják. Életemre, véget nem érő kutatás után sem látom, vagy értem az okát, hogy ezt miért tették. Tudom ugyanakkor, hogy az uca.xml fájlban a sort változatlanul

hagyva nem viheted át az uca.xml fájlot egy másik Xfce-rendszerbe ... hacsak nem veszed a fáradságot, és el nem távolítod ezt a sort minden egyes Thunar Egyéni művelet leírásából.

Van a fájloknak egy meghatározott csoportja, amit a gépeimre felrakok, amikor (legalábbis nekem) „új” gépre telepítek. Viszem a magazin elrendezésére szolgáló fájlokat (alapvetően fontosak nekem), a saját bash-szkriptjeimet és – a hibátlan – uca.xml fájlot. Ha kézzel csinálnám NAPOKBA telne átvinni. Ezért, pár éve véletlenül felfedeztem, hogy ha végigmegyek az uca.xml fájlmon és törlöm a <unique-id> ... </unique-id> sort minden Egyéni művelet leírásából, akkor úgy működik, ahogy kell. A Thunar automatikusan létrehozza, ha nincs ott. Ez lehetővé teszi számomra, hogy az uca.xml fájlot mozgassam a PCLinuxOS Xfce-t futtató különféle gépeim között.

Tehát, nekiláttam az uca.xml fájlom átvitelére a fő gépemről az utazós laptopra. Ez azt jelenti, hogy megnyitottam az uca.xml fájlt szerkesztőben és kézzel töröltem az összes <unique-id> ... </unique-id>-t tartalmazó sort. A teljes folyamat úgy 15 percet vett igénybe, ami SOKKAL gyorsabb, mintha egymás után beléptem volna az egyes Egyéni műveletek tulajdonságaiba, de ettől ez még így is nagyon unalmas folyamat volt. Miközben csináltam töprengtem. Van tipped, hogy mire gondoltam? Igen. Nyertél: „Kell, hogy legyen egy ennél gyorsabb módszer!”

### A megmentő bash-szkript

Nos, ekkor döntöttem el, hogy csinálok egy bash-szkriptet, ami megcsinálja helyettem és ez az, amiről a cikk valójában szól.

Ha CSAK a sor eltávolítására akartam volna szkriptet írni, nagyon egyszerű lett volna (egyszer már kitaláltam, hogyan csináljam). Igazából, annyira egyszerű lett volna, hogy még csak szkript sem kellett volna hozzá. Csak a megfelelő parancsot a Thunar Egyéni műveletbe be írbi paraméterként. De készítettem egy bash-szkriptet, ami lehetővé teszi alkalmazását BÁRMILYEN szövegfájlra (sima szövegfájlra). Ki tudja? Lehet, hogy valamikor jól jöhet egy ilyen eszköz, zavaró

szövegsorok eltávolítására más szövegfájlokból. Eldöntöttem, hogy inkább ebbe az irányba megyek.

A szkriptemnek „strip-line.sh” nevet adtam. A következőkben láthatod a szkript sorait. Az egész 471 byte hosszúságú, ezért nem adok letöltési helyet. Rövid lévén öt perc (vagy kevesebb) kell a begépeléséhez.

```
#!/bin/bash
```

```
# This script will remove any line from a text file with the specified text in it.
```

```
#
```

```
# by Paul Arnote
```

```
# Published in The PCLinuxOS Magazine, April 2025 issue
```

```
# This script is released under a GPL 2.0 (or higher) license
```

```
#
```

```
# This script takes a single text file filename as its sole parameter.
```

```
textfile=$(zenity --entry --title="Text File Line Deleter" --text="Enter the text to remove:")
```

```
if [ $? == 1 ]; then
```

```
exit
```

```
fi
```

```
sed -i '/$textfile/'d' $1
```

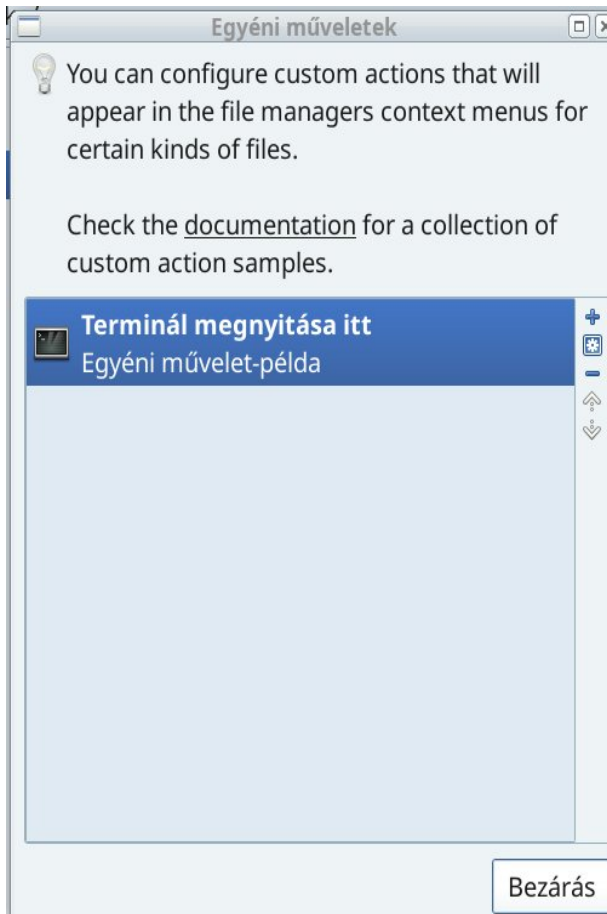
Akár sok más szkriptem, ez is futtatható parancssorból, vagy Thunar Egyéni műveletként is. Ha parancssorból indítod, csak a szövegfájl nevét kell megadni, amiből sorokat akarsz törölni.

A bash-szkriptjeimet a /home partícióm egy bin nevű könyvtárban tartom. A /home/bin könyvtár benne van az útvonalamban, így az operációs rendszer könnyen megtalálja azokat anélkül, hogy a bash-szkriptek teljes útvonalát meg kellene adni, amikor futtatni kell.

Akár a többi bash-szkript esetében az első sor a szokásos bash „indító” karaktorsor. Azt hét sor leírás követi.

A „textfile”-al kezdődő sor indítja a Zenity párbeszédablakát, ahol megadható, hogy milyen szöveget keressen. Esetemben, az uca.xml fájlnál az összes <unique-id> sort kell megkeresni és törölni. Az „if” teszi lehetővé, hogy a szkript és a Zenity párbeszédablak tisztán kilépjen, ha a „Cancel” (mégsem) gomb lett kiválasztva a párbeszédben.

Az utolsó, „sed”-del kezdődő sorban történik a „mágia”. Az „i” paraméter mondja meg a sed-nek, hogy közvetlenül (inline) szerkesszen, ami a bash nyelvén a fájl szerkesztését jelenti annak megnézése nélkül (amíg egyébként normálisan, egy szövegszerkesztőben megnyitva csinálnád). A Zenity a párbeszédben megadott (\$textfile) szövegre keres és találat esetén az egész sort törli („/d” paraméter).



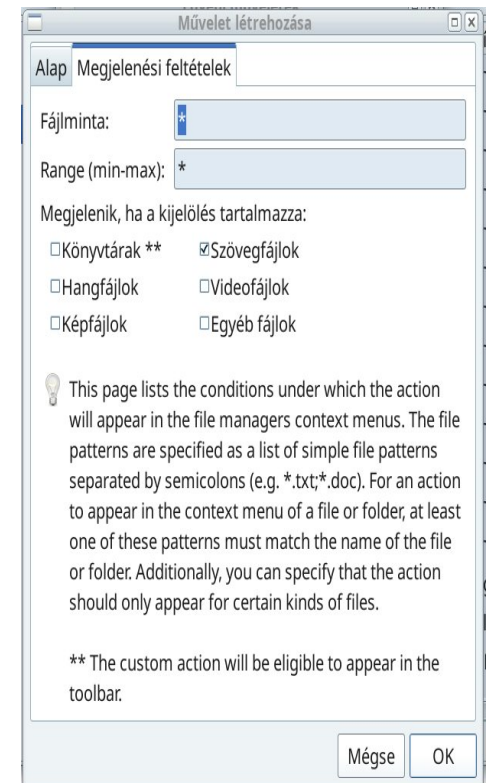
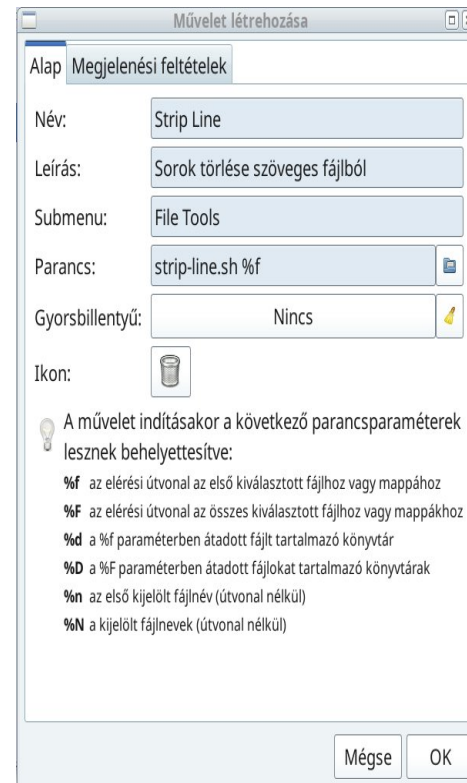
Az én esetemben, a Zenity párbeszédablakába csak a „unique-id”-t kellett beírnom. Amikor az „OK” gombra kattintok, az összes „unique-id”-t tartalmazó sort törli a szövegfájlból (és az XML-fájl nem egyéb, mint egy különlegesen formázott szövegfájl).

Elmondom, a szkript TÖKÉLETESEN működik. Azt, aminek a végrehajtása nekem kézzel, szövegszerkesztőben 15 percbe telik, **öt másodperc alatt** megcsinálja. Ha már évek óta meglett volna a szkript, akkor (összeadva) TÖBB ÓRA fárasztó munkától kímélt volna meg magam.

## A szkript használata Thunar Egyéni műveletként

Ahogy korábban említettem, a szkriptet parancssorból, vagy Thunar Egyéni műveletként futtatásra készítettem.

Tehát. Adjuk hozzá a Thunar Egyéni műveletekhez. Először a Thunar-ben menj a „Szerkesztés → Egyéni műveletek”-hez. A képhez hasonló párbeszédablak nyílik meg. Új Thunar Egyéni művelet készítéséhez csak kattints a „+” gombra, az előző képen pirossal kerettel emeltem ki.



A következő két képből az elsőnél a Thunar Egyéni művelet nevét és leírását adod meg. Én azt is meghatároztam, hogy a Egyéni műveletemet az általam létrehozott „File Tools” almenübe helyezze el. A parancs a **strip-line.sh %f** lesz. Az „%f” mondja meg a Thunar-nek, hogy a szkriptet azon a fájlra hajtsa végre, amire kattintottam. Átadja a bash-szkriptnek feldolgozásra a teljes útvonalat és a

fájlnévet. (Megjegyzés: %n is jól működne (%f helyett), de az csak a fájl nevét adja át, az útvonal nélkül.)

A második fül alatt (második kép), pipáld ki a „Szövegfájlok” melletti jelölőnégyzetet. A „Fájlmenta” és „Range” (hatókör) mezőkben hagyd meg a „\*” alapértéket. Ha ezzel készen vagy, kattints a párbeszédablak tetejénél lévő „OK” gombra. Az új Thunar Egyéni művelet a műveleteid listájának végé jelenik meg, vagy (ha úgy csináltad ahogy én, és egy almenüben helyezed el), akkor az adott almenü listájának a végén.

### Összegzés

Noha ez egy viszonylag egyszerű szkript, egy nagyon fárasztó feladatot old meg, azaz meghatározott szöveg összes előfordulásának megkeresését és az azt tartalmazó teljes sor törlését a szövegfájlból. Noha az én esetem sajátos és célirányos, megpróbáltam olyan rugalmasra készíteni, hogy más olyan szövegfájlok esetén is használható legyen, ahol ismétlődő szövegeket, vagy sorokat kell eltávolítani.



## The PCLinuxOS Magazine Special Editions!

**Get Your Free Copies Today!**