

Gmail üzenet küldése a Gmail megnyitása nélkül

PCLinuxOS Magazine – 2014. április

Írta: Paul Arnote (parnote)

Mint a PCLinuxOS Magazine szerkesztője gyakran kerülök olyan helyzetbe, hogy a magazin stábjával fájlokat küldözgetünk oda-vissza egymásnak e-mailben. Igen, természetesen használjuk a Google Docs-ot is, de néha a stáb egyetlen tagjának kell egy dokumentumot küldenem. Ami engem illet, szerintem erre a legjobb az e-mail.

Ugyanakkor elég macerás mindig bekapcsolni a böngészőt (vagy a levelezőt ... én nem használok levelező programot, a Gmail webes felületét szeretem használni) és elküldeni a fájlt a Gmail webes felületéről. Ha már úgyis meg kell keresnem a fájlt a fájlkezelővel, miért ne küldhetném el onnan? Nos, el tudom. Így te is tudod.



Mielőtt belefognánk, nem szállnék be a Google magán titkokkal kapcsolatos hírnevéről a vitába. Jól ismerem a Google híréit ezen a területen és mindezek ellenére választom a szolgáltatását. Megfelel az igényeimnek és nincs mit elrejtennem (legalábbis én úgy gondolom, hogy nincs). Ezek fényében a cikk a Gmail fiók SMTP szolgáltatásának levél küldésére való használatáról szól anélkül, hogy be kellene lépni, vagy megnyitni a Gmail-t. Ha akarod, ezt beállíthatod más e-mail szolgáltatáshoz, vagy szolgáltatóhoz is, azt már neked kell kitalálnod, hogy hogyan kell – azt hiszem a szolgáltatód, vagy az internetkereső segíthet ebben.

Sendmail, vagy Postfix?

Ha esetleg nem tudnád, két alapvető levélküldő program érhető el a PCLinuxOS tárolóban: a sendmail és a postfix. A mi-volt-előbb-a-tojás-vagy-a-tyúk vitához

annyit, hogy a sendmail régebb óta van a piacon. A postfix sokkal később jött, a könnyebb beállíthatóságot megcélozva, miközben megtartja a sendmail-en felnőtt felhasználók által már megszokott parancsokkal való kompatibilitást. Mindemellett a postfix gyorsabb és biztonságosabb.

Én sendmail használatával kezdtem. De hű! Nem csak komplikált volt a beállítása, de megbízhatatlannak is bizonyult. Néha működött, néha nem. Miután megírtam a szkriptemet és mindent elrendeztem, ahogy akartam, valami megakadályozta a sendmail-t az e-mail elküldésében. Vacakoltam egy kicsit néhány dologgal és sikerült a sendmail-t ismét működésre bírnom. Ugyanakkor sosem tudtam megfejteni, hogy végül pontosan mitől kezdte a sendmail ismét küldeni az e-maileket.

Így elhatároztam, hogy rövidre zárom, telepítettem postfix-et és ismét kipróbáltam. Jegyezd meg, egyszerre csak egyik, vagy másik lehet – sendmail VAGY postfix – telepítve. A postfix synaptic-os telepítése a sendmailt eltávolította. Hasonlóképpen, sendmail telepítése postfix-et eltávolítja. A több lépés ellenére a postfix beállítása könnyebbnek tűnik. Emellett sokkal stabilabbnak is éreztem és bármikor indítottam működött. Szerencsére nem kellett a szkriptet átírnom, mivel mind a sendmail, mind a postfix a levelek küldésére ugyanazokat a parancsokat használja.



Gmail üzenet küldése a Gmail megnyitása nélkül

A sendmail-hez és a postfix-hez rövidített instrukciók a send-gmail szkript megjegyzéseiben olvashatók. A szkript és az eredeti cikkek, ahol az instrukciókat találtam, a magazin honlapjáról [letölthetők](#). A szkriptnek a magazin hasábjain megjelentetett változatából ezek a megjegyzések, helytakarékossági okokból hiányoznak. Mindazonáltal a cikkben végigmegyünk ezeken az instrukciókon.

Csak nyugodtan próbáld ki a sendmail vonalat is, ha akarod – és elég bátor vagy. Ugyanakkor, noha kevesebb lépést igényel, sokkal összetettebb program, mint a postfix. Az utóbbi eredetileg arra hivatott, hogy egyszerűsítse a folyamatot, és sikerült. Meggyőződésed érdekében a sendmail instrukciókat lentebb leírtam. Mindenkinek más a tapasztalata.

A sendmail és a postfix beállításának utasításainak azonosnak kell lennie, függetlenül a használt asztali környezettől.

Az alapok lerakása sendmail-lel

1. lépés: telepítsd a sendmail Synaptic-ból

2. lépés: rendszergazdaként add ki a következő parancsokat parancssorból:

```
mkdir -m 700 /etc/mail/authinfo
```

```
cd /etc/mail/authinfo
```

```
echo 'AuthInfo: "U:root" "I:az_email:címed" "P:a_jelszavad"'  
> gmail-auth
```

(az I: után az e-mail címed, és a P: után a jelszavad kell legyen)

```
makemap hash gmail-auth < gmail-auth
```

3. lépés: továbbra is root-ként szerkeszd meg a sendmail.mc fájlt (/etc/mail-ben van). A következő sort PONTOSAN az első MAILER-t tartalmazó sor elé illeszd be. (A következő sorokat NE a sendmail.mc fájl legelső sorába írd!)

```
define(`SMART_HOST', `[smtp.gmail.com'])dnl  
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl  
define(`ESMTP_MAILER_ARGS', `TCP $h 587')dnl  
define(`confAUTH_OPTIONS', `A p')dnl  
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN  
PLAIN')dnl  
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5  
CRAM-MD5 LOGIN PLAIN')dnl  
FEATURE(`authinfo', `hash -o /etc/mail/authinfo/gmail-  
auth.db')dnl
```

4. lépés: továbbra is root-ként építsd újra a sendmail beállításait a következő paranccsal:

```
make -C /etc/mail
```

5. lépés: root-ként indítsd újra a sendmail szolgáltatást:

```
/etc/init.d/sendmail reload
```

6. lépés: ha mindent jól csináltál teszteld. Az alábbi parancs terminálba beírásával kapnod kell magadtól egy e-mailt:

```
echo "This is a test" | mail -s "Testing" [email-címed@domained.com]
```

Az e-mailcím körüli zárójeleket távolítsd el!

Az alapok lerakása postfix-szel

1. lépés: a következő kilenc (9) sort root-ként kell végrehajtani.

2. lépés: nyisd meg a /etc/postfix/main.cf fájlt és add a következő parancsokat a fájlhoz:

```
relayhost = [smtp.gmail.com]:587  
smtp_sasl_auth_enable = yes  
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd  
smtp_sasl_security_options = noanonymous  
smtp_tls_CAfile = /etc/postfix/cacert.pem  
smtp_use_tls = yes
```

3. lépés: csinálj egy /etc/postfix/sasl_passwd fájlt.

4. lépés: a sasl_passwd fájlodban a következőknek kell lennie:

```
[smtp.gmail.com]:587          FELHASZNÁLÓNÉV@gmail.com:JELSZÓ
```

5. lépés: állítsd be az engedélyeket és a postfix-et a sasl_passwd fájl használatára:

```
chmod 400 /etc/postfix/sasl_passwd  
postmap /etc/postfix/sasl_passwd
```

6. lépés: töltsd le a cacerts.pem-t a <http://curl.haxx.se/ca/cacert.pem>-ről és helyezd el a /etc/postfix-ben. Jobb billentyűvel kattints a hivatkozáson és a

menüből válaszd a „fájl mentése mint ...”-et. Ellenkező esetben a böngésződ meg akarja nyitni (tanúsítvány), ahonnan nem tud továbblépni.

7. lépés: nyisd meg a `/etc/postfix/aliases` fájlt. Keresd meg a sort, amiben postmaster: root olvasható a „root”-ot cseréld le VALÓS felhasználóra (a felhasználóneved a gépen). Az én aliases fájlomban a 15. sorban van. Ugyanezt tedd meg a 80. sorban is, a root-ot felhasználóra cserélve.

8. lépés: futtasd a newaliases parancsot, majd indítsd újra a postfix-et a service postfix restart paranccsal.

9. lépés: INDÍTSD ÚJRA A SZÁMÍTÓGÉPEDET!

10. lépés: ha mindent jól csináltál, teszteld. Az alábbi parancs terminálba beírásával kapnod kell magadtól egy e-mailt:

```
echo "This is a test" | mail -s "Testing" [email-címed@domained.com
```

Az e-mailcím körüli zárójeleket távolítsd el!

A send-gmail szkript

Most az alapok már megvannak – a Gmail SMTP szerverével kommunikálhat akár a sendmail, akár a postfix – elkészíthetjük a szkriptet, ami kezeli a felhasználói bementet. A felhasználói oldal adatbemeneteinek bevitelére szolgál párbeszédablakhoz a Zenity-t választottam, mivel egyszerű és könnyen használható.

Először is íme a szkript:

```
1.#!/bin/bash
2.3.# send-gmail.sh
4.#
5.# Egy meghatározott felhasználóknak parancssorból, vagy Thunar
6.# felhasználói menüből fájlküldésre a sendmail-t használó szkript.
7.# A szkript Zenity szövegbeviteli párbeszédablakkal felszerelve, hogy
8.# végigvezesse a felhasználót az e-mail küldésén.
9.#
10.# A Postfix rész ezekből a cikkekből származik:
11.# https://rtcamp.com/tutorials/linux/ubuntu-postfix-gmail-smtp/
12.# és footstep11-től a PCLinuxOS fórumról.
13.# http://www.pclinuxos.com/forum/index.php/topic,124750.msg1043028.html#msg1043028
14.#
15.# Sendmail-rész ezen a cikken alapul:
16.# http://linuxconfig.org/configuring-gmail-as-sendmail-email-relay
17.#
```

```
18.# Parancssori használat: send-gmail.sh [útvonal/fájlnev csatolandó
fájl]
19.# Thunar felhasználói menü használat: send-gmail.sh %f or send-
gmail.sh %n
20.#
21.# Copyright © 2014, Paul Arnote. All Rights Reserved.
22.# As originally published in The PCLinuxOS Magazine, April 2014.
23.# Read the article for more complete information on how to set up.
24.# http://pclosmag.com/download.php?f=2014-04.pdf
25.# Released under the GPLv2 license.
26.#
27.# You are free to alter, change or distribute this work, provided
28.# you do so while retaining this notice, and so long as your
derivative
29.# work is also released to the public under any of the GPL licenses.
Any
30.# distribution of this work must also comply with the terms of the
GPLv2
31.# license, and no charge may be made for this work, whatsoever.
32.#
33.# A postfix és sendmail telepítő információk eltávolítva magazinban
nyomtatott
34.# szkriptváltozatból helyszűke miatt. Ugyanakkor elérhetőek a magazin
honlapjáról
35.# letölthető változatban itt:
36.#
37.# http://pclosmag.com/Misc-Downloads/send-gmail.sh.txt
38.# Ne feledd a .txt fájlkiterjesztést eltávolítani és a fájlt
39.# végrehajthatóvá tenni (chmod +x send-gmail.sh).
40.#
41.#
42. hasCC=0
43. hasBCC=0
44.#
45. function getSubject {
46. Subject=`zenity --title="Send Gmail" --entry -text="Add meg a
tárgyat:"`
47.     if [ $? == 1 ]; then
48.         exit
49.     fi
50.     if [ `echo ${#Subject}` -lt 3 ]; then
51.         zenity --title="Send Gmail" --question --text="Nincs
tárgy megadva.\n\nAkarsz tárgyat megadni?" --ok-label="Igen"
--cancel-label="Nem"
52.         if [ $? == 1 ]; then
53.             zenity --title="Send Gmail" --info --
text="Kilépés. Nincs tárgy."
54.             exit
55.         else
56.             getSubject
57.         fi
58.     fi
```

Gmail üzenet küldése a Gmail megnyitása nélkül

```
59.}
60.
61.function getMessage {
62.Message=`zenity --title="Send Gmail" --entry --text=
"Add meg az e-mailed szövegét:"`
63.  if [ $? == 1 ]; then
64.    exit
65.  fi
66.  if [ `echo ${#Message}` -lt 5 ]; then
67.    zenity --title="Send Gmail" --question --text="Nincs
üzenetszöveg megadva.\n\nnaz üzenet szövegét megadod?" --ok-label="Igen"
--cancel-label="Nem"
68.    if [ $? == 1 ]; then
69.      zenity --title="Send Gmail" --info --
text="Kilépés. Nincs üzenetszöveg."
70.      exit
71.    else
72.      getMessage
73.    fi
74.  fi
75.}
77.function getRecipient {
78.TO=`zenity --title="Send Gmail" --entry -text="Add meg a címzett
címét:"`
79.  if [ $? == 1 ]; then
80.    exit
81.  fi
82.  if [ `echo ${#TO}` -lt 10 ]; then
83.    zenity --title="Send Gmail" --question -text="Helytelen, vagy
rossz elsődleges címzett megadva.\n\n Akarsz címzettet meghatározni?" --
ok-label="Igen" --cancel-label="Nem"
84.    if [ $? == 1 ]; then
85.      zenity --title="Send Gmail" --info -text="Kilépés. Nincs
konkrét címzett."
86.      exit
87.    else
88.      getRecipient
89.    fi
90.  fi
91.}
93.function getCC {
94.  zenity --title="Send Gmail" --question -text="Akarsz másolatot
küldeni\n valakinek?" --ok-label="Igen" --cancel-label="Nem"
95.  if [ $? == 1 ]; then
96.    let 'hasCC = 0'
97.  else
98.    CC=`zenity --title="Send Gmail" --entry --text="Add mega
másolat címzettjét:"`
99.    let 'hasCC = 1'
100.  fi
101.  if [ "$hasCC" == "1" ]; then
102.    if [ `echo ${#CC}` -lt 10 ]; then
```

```
103.      zenity --title="Send Gmail" --question --text="A másolat
címzettje hibás, vagy nincs.\n\nAkarsz címzettet megadni?" --ok-
label="Igen" --cancel-label="Nem"
104.      if [ $? == 1 ]; then
105.        let 'hasCC = 0'
106.      else
107.        getCC
108.      fi
109.    fi
110.  fi
111.}
112.
113.function getBCC {
114.  zenity --title="Send Gmail" --question -text="Küldesz titkos
másolatot\n más címzettnek?" --ok-label="Igen" --cancel-label="Nem"
115.  if [ $? == 1 ]; then
116.    let 'hasBCC = 0'
117.  else
118.    BCC=`zenity --title="Send Gmail" --entry --
text="Add meg a titkos másolat címzettjét:"`
119.    let 'hasBCC = 1'
120.  fi
121.  if [ "$hasBCC" == "1" ]; then
122.    if [ `echo ${#BCC}` -lt 10 ]; then
123.      zenity --title="Send Gmail" --question --text=
"Titkos másolat hiányzó vagy hibás címzése.\n\n Akarsz címzettet
megadni?" --ok-label="Igen" --cancel-label="Nem"
124.      if [ $? == 1 ]; then
125.        let 'hasBCC = 0'
126.      else
127.        getBCC
128.      fi
129.    fi
130.  fi
131.}
132.
133.  if [ $# -lt 1 ]; then
134.    zenity --title="Send Gmail" --info --text="Használat:\n\t
send-gmail.sh [útvonal/fájlnev]\n\nThunar felhasználói művelet:\n\t
send-gmail.sh %f\n\t\tVAGY\n\t\tsend-gmail.sh %n\n\n\n          Nincs fájl
megadva. Kilépés."
135.    # Xdialog --title "Send Gmail" --infobox "Használat:\nsend-
gmail.sh [útvonal/fájlnev]\n\nThunar felhasználói művelet:\nsend-gmail.sh
%f\nVAGY\n send-gmail.sh %n\n\nNincs fájl megadva. Kilépés." 0 0 0
136.    exit
137.  fi
139. FILE1=`zenity --title="Send Gmail" --entry --entry-text=$1 --text=
"Írd be a fájlnevet és az útvonalat\nna csatolandó fájlhoz:"`
140.  if [ $? == 1 ]; then
141.    exit
142.  fi
143.  if [ -f $1 ]; then
```

Gmail üzenet küldése a Gmail megnyitása nélkül

```
144. FILE=$FILE1
145. fi
146. if [ -z $1 ]; then
147. zenity --title="Send Gmail" --info --text="Kilépés. Nincs
csatolandó fájl megadva."
148. exit
149. fi
150.
151.# Ha a tömörített fájlt a tömörítendő könyvtárban akarod elhelyezni,
akkor
152.# használd a $1/$1.tar.gz tar parancsot mind a parancsban, mind a
fájlváltozónál.
153.# Ez az alapbeállítás.
154.#
155.# Ha a tömörített fájlt a tömörítendő könyvtár szülőkönyvtárába
akarod rakni
156.# használd a „$1.tar.gz” mind a tar parancsban mind a fájl
változójában (vagy
157.# csak távolítsd el az első „$1/”-et a tar parancsból és a
158.#fájlváltozóból).
159.
160. if [ -d $1 ]; then
161. tar -zcvf $1/$1.tar.gz $1
162. FILE=$1/$1.tar.gz
163. fi
164. if [ ! -e $1 ]; then
165. zenity --title="Send Gmail" --info -text="A fájl nem található.
Kilépés."
166. exit
167. fi
168.
169. getSubject
170. getMessage
171. getRecipient
172. getCC
173. getBCC
174.
175. if [ "$hasCC" == "0" ] && [ "$hasBCC" == "0" ]; then
176. echo ${Message} | mailx -s "$Subject" -a $FILE $TO
177. exit
178. fi
179.
180. if [ "$hasCC" == "1" ] && [ "$hasBCC" == "0" ]; then
181. echo ${Message} | mailx -s "$Subject" -a $FILE -c $CC $TO
182. exit
183. fi
184.
185. if [ "$hasCC" = "0" ] && [ "$hasBCC" == "1" ]; then
186. echo ${Message} | mailx -s "$Subject" -a $FILE -b $BCC $TO
187. exit
188. fi
189.
```

```
190. if [ "$hasCC" == "1" ] && [ "$hasBCC" == "1" ]; then
190. echo ${Message} | mailx -s "$Subject" -a $FILE -b $BCC -c $CC $TO
192. exit
193. fi
194.
195. exit 0
```

Kezdeként a szkript két változót határoz meg, a hasCC-t és a hasBCC-t, illetve nullázza azokat. Ezután függvények sorozatán keresztül állítja be a párbeszédablakot, amit a Zenity vezérel. Ezek a párbeszédablakok teszik lehetővé az üzenet szövegének, a címzettnek, az üzenet tárgyának, további másolatok (CC) és titkos másolatok (BCC) címzettjeinek meghatározását. Az első három adatelem az e-mail kötelező kelléke, de nem minden e-mailnek lesz CC, és BCC címzettje

Minthogy a CC és BCC címzett opcionális, különleges helyzetet jelentenek. Ezek megváltoztatják az e-mailt küldő parancssort is. A két változó szerepe ekkor jelentkezik. Ha nincs CC, vagy BCC cím, akkor a hasCC és hasBCC értéke 0 marad. Ugyanakkor, ha a felhasználó az e-mail másolat és titkos másolatként el akarja küldeni, akkor a megfelelő érték 1-re változik. Ezen változóértékek határozzák meg, hogy melyik parancsot használja a sendmail e-mail küldésre (erre való az utolsó négy feltételes függvény a szkript végén).

Egy másik dolog, ami kell a fájlküldéshez az a fájlnev, vagy útvonal és fájlnev átadása parancssori paraméterként. Emlékezz arra, hogy eredetileg a céloom az volt, hogy a fájlkezelőből tudjak közvetlenül e-mailt küldeni böngésző, vagy levelezőkliens megnyitása nélkül. A valóságban a szkript úgy indul, hogy egy párbeszédablakot nyit meg, ahol meghatározható a fájlnevet, vagy a teljes elérési útvonalat. Ha a fájlt a fájlkezelődben kiválasztottad és a szkriptet úgy állítottad be, hogy a jobb billentyűs felhasználói menüből függvényként meghívható, akkor a fájlnevnak a bemeneti mezőben már meg kell jelennie. De egy kicsit később bővebben erről.

Ezután meghívja az egyes függvényeket, egyenként moduláris stílusban, lehetőséget adva a felhasználónak, hogy párbeszédablakokban adja meg az adatokat. Amikor válaszolsz a legutolsó párbeszédablakban (BCC címzettje), a fájlt elküldi az e-mail címzettjeinek.

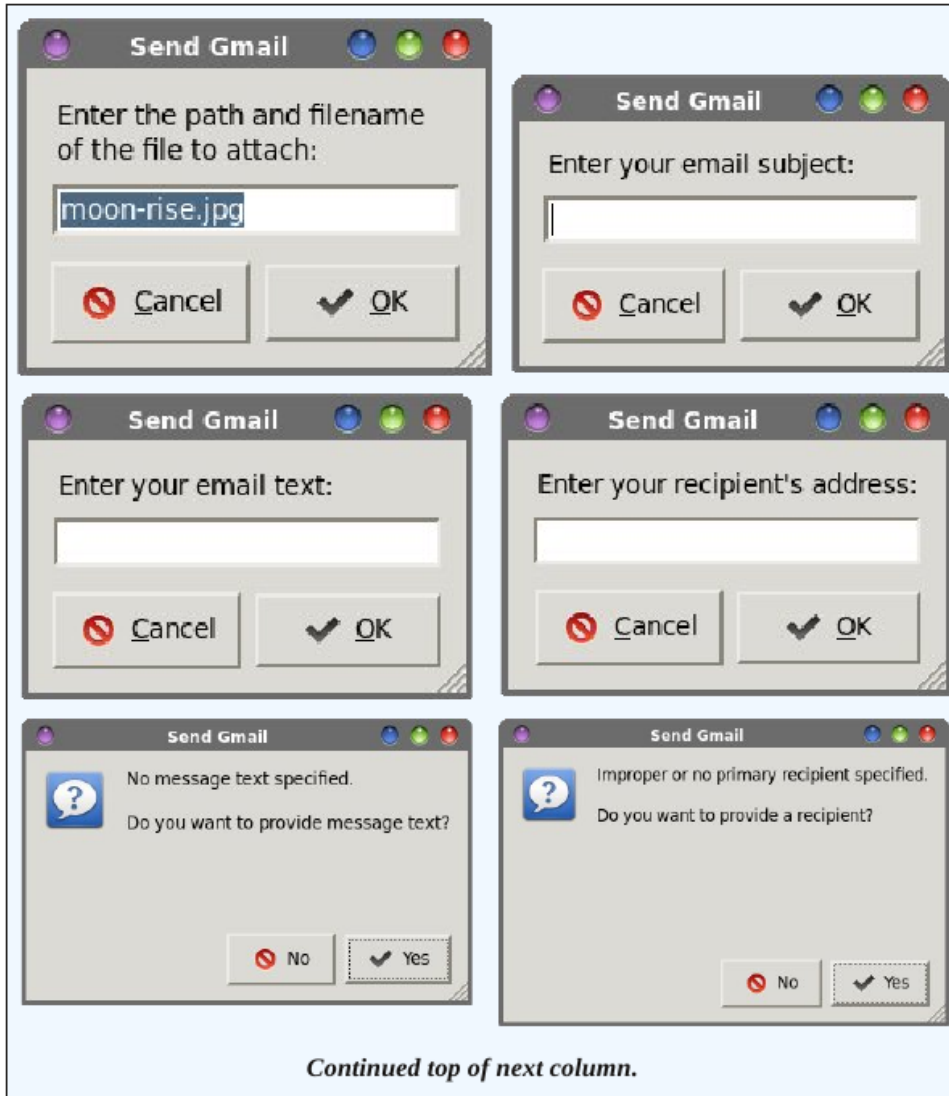
Amit kapsz

A send-gmail szkript képes 19 Zenity párbeszédablakot megjeleníteni, de szükségtelen és redundáns megjeleníteni az összes fajta párbeszédablakot. Miközben a szkriptben nincs túl sok hibaellenőrzés, leellenőrzi az adatok jelenlétét, vagy hiányát. Egyes adatok jelenléte (mint a CC, vagy BCC címzett)

Gmail üzenet küldése a Gmail megnyitása nélkül

megváltoztatja az e-mailt küldő parancsot. Hasonlóképpen, ha egy 10 karakternél rövidebb e-mailcímet írsz be (ez az elsődleges címzetre vonatkozó korlátozás), három karakternél rövidebb tárgy, vagy öt karakternél rövidebb üzenet esetén lehetőséget ad az információ újbóli megadására, vagy a kilépésre.

Itt láthatsz kilencet a leggyakoribb párbeszédablakok közül:



Hogyan használd

A send-gmail szkript teljes értékű bash szkript, amit parancssorból el lehet érni. Amikor parancssorból fut, akkor is láthatod a grafikus párbeszédablakokat, amikben megadod a bementi adatokat. Ne feledd, a fájlnévet (vagy a teljes útvonalat és fájlnévet) parancssori paraméterként kell megadnod.

Azonban emlékezz az eredeti célunkra – használható legyen az aktuális fájlkezelődből. Szerencsére sok modern fájlkezelő lehetővé teszi a működés kiterjesztését a jobb billentyűs felhasználói menün keresztül. Legcélszerűbb a szkriptet egy, az útvonalleírásban szereplő könyvtárban elhelyezni, az egyszerűbb alkalmazhatóságért.

KDE alatt létrehozatsz egy KDE szervizmenüt. Ezzel a témával már foglalkoztunk a PCLinuxOS Magazine egy korábbi számában. A leírás hosszadalmas volta miatt (noha nem nehéz alkalmazni), csak hivatkozom a folyamatot leíró [korábbi cikkre](#). A végrehajtandó parancs `send-gmail.sh %f`.

Ugyanakkor a bejegyzésnek az alábbiakban leírtakhoz hasonlóknak kell lennie. Gondoskodj arról, hogy `send-gmail.desktop`-ként mentsd a `~/.kde4/services` alá, ha azt akarod, hogy adott felhasználó érje el, vagy a `/usr/share/kde4/services` alá, ha szeretnéd, hogy a számítógép összes használója elérje.

```
[Desktop Entry]
Type=Service
Name=Send Gmail
Encoding=UTF-8
ServiceTypes=KonqPopupMenu/Plugin
Actions=SendGmail
MimeType=application/octet-stream
Icon=gmail
```

```
[Desktop Action SendGmail]
Name=SendGmail
Exec=send-gmail.sh %f
Icon=gmail
```

Xfce alatt beállíthatod Thunar felhasználói műveletnek. A parancs lehet akár `send-gmail.sh %f`, vagy `send-gmail.sh %n` (mindkettő működik ... `%f` a teljes elérési út és a fájlnev, míg a `%n` csak a fájlnev). A Megjelenés feltételei fülnél hagyd a fájlmintát az alapérték *-on, és helyezz minden fájltypus elé kijelölést. Ezután, amikor jobb billentyűvel kattintasz egy fájlhoz hozzáfűzi a kimenő e-mailedhez és elküldi az általad meghatározott címzett(ek)nek.

Mate esetén, ami alapvetően a Gnome 2.x folytatása, beállíthatod a szkriptet a Caja jobb billentyűs felhasználói menüjéből használatra. Csak rakd a szkript egy másolatát a `~/.config/caja/scripts-be`. Tedd a fájl végrehajthatóvá. Ha jól csinálod a „send-gmail.sh” megjelenik a Caja jobb billentyűs felhasználói menüjében a szkriptek között. Most, ha jobb billentyűvel kattintasz egy fájlhoz és kiválasztod a Script → `send-gmail.sh-t`, az adott fájl mellékletként el lesz küldve a meghatározott címzett(ek)nek.

Cinnamon-nál a Nemo a fájlkezelő, a Nautilus egy leszármazottja. A szkriptet beállíthatod a jobb billentyűs felhasználói menüben használatra. Egyszerűen helyezz egy másolatot a `~/.gnome2/nemo-scripts` könyvtárba és tedd futtathatóvá. Ezután a `send-gmail.sh` szkriptnek meg kell jelennie a Nemo jobb billentyűs felhasználói menüjében, a Szript menüpont alatt. Válaszd ki az e-mail mellékletként küldeni tervezett fájl jobb billentyűvel rákattintva és válaszd ki az új menüelemünket. A kiválasztott fájlod elrepül a címzett(jeid)nek.

LXDE felhasználók a PCFileMan fájlkezelővel eléggé rosszul jártak a szkriptnek a PCFileMan-on belüli használatát illetően. A PCFileMan fejlesztői csak most kezdték lehetővé tenni a felhasználói kiterjesztések használatát. Ugyanakkor ezeknek a kiterjesztéseknek C nyelven kell íródniuk, gcc-vel lefordítva. Egyelőre csak annyit tudok mondani, hogy csak a C-ben írott és lefordított kiterjesztések engedélyezettek. Még ezek az információk is elég hiányosak. Remény az van (a remény hal meg utoljára, nemde?), hogy a fejlesztők könnyebb, felhasználóközelibb kiterjesztéseket készítenek majd a PCFileMan-hoz, amikor a Gtk+ alapú Qt alapúra váltanak.

A PCFileMan-t használóknak csak egyetlen lehetőségük van (még így is több, mint ami korábban volt). Az egér jobb billentyűjével kattintanak az elküldendő fájlra, vagy könyvtárra. Kiválasztják a helyi menüből a Megnyitás ezzel-t. Kattintanak a „Felhasználói parancssor” fülre a nyíló párbeszédablakban. Beírják a `send-gmail.sh %f-et` (feltéve, hogy a szkriptet egy, az útvonalban lévő könyvtárba mentették; ellenkező esetben a fájlnevet a szkript teljes elérési útvonalával ki kell egészíteni) a parancsbeviteli ablakba, majd OK-ra kattintanak. Tudom, ez sok lépés és könnyebb lehet a fájl elküldeni böngészőből, vagy levelezőből.

Emellett a nyakatekert megoldás mellett lehet még nyitni terminált a küldeni tervezett fájl, vagy könyvtárat tartalmazó könyvtárban, és futtatni a `send-gmail` szkriptet a parancssorból. Legyél pontos a küldeni tervezett fájl, vagy könyvtár, illetve a könyvtár nevét illetően. Valóban sokkal könnyebb lehet a „Felhasználói parancssor”-t használni, a korábban megtárgyalt módon.

A címzettek beviteli párbeszédablakainak bármelyikében (elsődleges címzett, CC másolati címzett, vagy BCC rejtett másolati címzett) több e-mail címet is beírhatsz, egyszerűen vesszővel elválasztva egymástól. Például a `pclinuxos.mag@gmail.com,yourfriend@somedomain.net,someoneyouknow@thisdomain.com` a fájl mindhárom címzettnek elküldi.

A szkript alapból egyetlen fájl melléklet küldésére lett beállítva. Természetesen kiterjeszthető több fájl melléklet kezelésére, miközben kikerüli a fájlok unalmas felsorolását és egyenkénti hozzáadását az e-mailhez mellékletként. Ha több fájl

akarsz küldeni egy e-mail mellékleteként, legjobb becsomagolni azokat egy tömörített archív fájlba (.tar.gz, .zip, .bz2, .xz stb.) és küldeni az archív fájlt a címzett(jeid)nek.

Ugyanakkor van egy kivétel. Ha egy fájlok tartalmazó könyvtárát választasz ki a címzett(ek)nek elküldésre, a send-gmail.sh szkript automatikusan létrehoz egy .tar.gz fájlt, a kiválasztott könyvtáron belül azonos néven, majd elküldi az adott archív fájlt a címzett(jeid)nek. Az archív fájlban a küldendő könyvtáron belüli tárolásával a dolgokat egyszerűen és rendben tartjuk. Azonban, ha az archív fájlt az archiválendő könyvtár szülőkönyvtárában akarod tárolni, könnyen átírhatsz a szkriptet. Egyszerűen változtasd meg a 161. és 162. sorokat az alább olvasható formára:

```
tar -zcvf $1.tar.gz $1
FILE=$1.tar.gz
```

Hasonlóképpen módosíthatod a tar parancsot úgy is, hogy .tar.gz-től eltérő archív típusú fájlt készítsen. A tar parancsban a „z”-t „j”-re változtatva .tar.bz2 fájlt, illetve a „z”-t „J”-re változtatva .tar.xz fájlt készíthet. A fájlkiterjesztést állítsd arra, amilyen tömörítési algoritmust akarsz választani. Szintén, ha ZIP archívumot akarsz használni (windowsos barátokkal megosztásra ... noha sok Windows eszköz van ami lehetővé teszi számukra .tar.gz fájlok olvasását, mint a WinZIP és a PeaZip), a parancssort megváltoztathatod ilyenre:

```
zip -r -x $1.zip $1.zip $1
FILE=$1.zip
```

Egy további apróság: a parancsok, amik a fájllevelekkel foglalkoznak (majdnem az egész szkript az, mivel az egész, fájlmelléklet e-maillal történő küldésére koncentrálnak) jól dolgoznak fájllevelekkel, amikben **nincsen** szóköz. Ha szoktál a fájllevelekben és könyvtárnevekben szóközt használni ... ÁLLJ! A Linux nem Windows és a Windows volt az, amelyik bevezette ezt az utálatos szokást. Sokszor

foglalkoztunk ezzel a témával a magazinban az unalomig vitatkozva. Linux fájlrendszerben NEM használhatsz szóközt a fájlnevekben. A szóköz parancssori paraméterek elválasztására szolgál, vagyis a fájlnevekben, vagy könyvtár nevében a szóköz ROSSZ ötlet, mivel a parancssor – és a bash szkriptek – a szóközt parancssori paraméter-elválasztónak látja. Ez egy előre eldöntött kérdés, hogy mi nem fogjuk a vitát itt megoldani.

Ha nem vagy képes a szóközők-a-fájl-és-könyvtárnevekben-szokásodról-lemondani, egyszerűen rakd kettős idézőjelbe a fájlneveket a szkriptben. Akkor rád fogja hagyni a dolgot. Nevezd ezt „elterelő kezelésnek”. Ha használni akarod a szkriptet és továbbra is ragaszkodsz a fájlnevekben a szóközhöz (ha kell, a kötőjel, vagy aláhúzás ugyanilyen jó), segíthet esetleg abban, hogy a változást magadtól tedd meg – ez az indoklás, hogy miért kell elállnod a fájlnevekben a szóközők használatától.

Megkötések

Jusson eszedbe, hogy a send-gmail szkriptben nincs túl erős hibaellenőrzés. A send-gmail mint olyan, nem tudja eldönteni, hogy az általad megadott e-mailcím megfelelő-e, vagy sem.

Van egy 500-as határ, amennyi e-mailt küldhetsz egy nap a G-mail SMTP szerverein keresztül. A korlátozás görgetett 24 órás időintervallumra érvényes. Tehát, ha du. 1-kor eléred a limitet, akkor a Gmail-fiókad nem lesz képes a Gmail SMTP szerverein keresztül e-mailt küldeni a következő nap du. 1-ig. Én sosem értem még el ezt a limitet (nem küldök OLYAN sok e-mailt), de ez fontos szempont lehet azoknak, akik SOK e-mailt küldenek. A Google így kerüli el, hogy az SMTP szervereket a nem kívánt e-maileket küldők és tömegesen üzletelők felhasználják. Abban az esetben, ha több e-mail-t kellene küldened, akkor jobb lehet saját SMTP szervert használni a feladatra.

Összegzés

Amint látod, a send-gmail szkript időt takaríthat meg neked, amikor fájlt akarsz küldeni e-mail mellékletként. Természetesen vannak területek, amiben továbbfejleszhető. Például nem lenne nehéz egy fájlkiválasztó párbeszédablakot megjeleníteni akkor, ha a fájlnev nincs megadva. Ugyancsak nem lenne bonyolult lehetővé tenni a felhasználónak a több fájl kiválasztását, archív fájl készíteni belőlük, majd az archívumot csatolni a kimenő e-mailhez. Az sem lenne nehéz, hogy engedjük a felhasználónak kiválasztani, milyen típusú archív fájl készüljön. Olyanra formálhatod, amilyenre akarod, kiegészítve e-mail cím érvényességének ellenőrzésével.

Ez a szkript már egyszerűsítette számomra a fájlok e-mail mellékletként történő küldését. Remélem, hogy ezt neked is megteszi. Megpróbáltam a szkriptet, amennyire csak lehet, asztali környezettől függetlenné tenni. Ha továbbfejlesztenéd a szkriptet, szeretném ha jeleznéd nekem. Küldj egy személyes üzenetet a PCLinuxOS fórumon, vagy egy e-mailt a pclinuxos.mag@gmail.com-ra.

