Programozás Gtkdialog-gal, harmadik rész

PCLinuxOS Magazine – 2014. június

Írta: Peter Kelly (critter)

Eddig megnéztük a szöveget, vagy választási lehetőséget felkínáló widget-eket, de gyakran lehet a felhasználó részéről bevitelre is.

A beviteli widget

Ez a legegyszerűbb, de rugalmas módja felhasználói bevitelnek, a felhasználó kap egy dobozt, amibe egyetlen sor szöveget írhat be.

Megjeleníthetsz valami alap szöveget, elrejtheted a szöveget jelszószerűen, korlátozhatod a szöveg hosszát és aktívvá teheted a beviteli dobozt, így a felhasználó azonnal elkezdhet gépelni, felülírva az alapszöveget. Amennyiben a beírt szöveg hosszabb a doboz méreténél, akkor görgetni fogja, hogy a kurzor mindig látható maradjon. A szöveg jöhet fájlból, illetve mehet oda. Ikonokat is megjeleníthetsz, beleértve a gtkdialog ikonjait is, a szöveg elején, végén, vagy mindkét helyen. Az ikonokhoz tartozhat elemleírás, ami megjelenik, ha ráállsz és az elemleírásban html tag-ek használhatók a karakterek jellemzőinek, pl. szín, vagy kiemelés, módosítására.

Természetesen magának a widget-nek a megjelenése is állítható. Lehet sima, vagy keretes, beállított szélességű és magasságú, látható, vagy láthatatlan, illetve aktív, vagy inaktív. A szöveg szükség esetén automatikusan frissülhet, vagy másik widget törölheti. A beviteli widget-ek reagálnak a jelzésekre. Egyszerű widget-nek nem túl gyenge!

Íme működés közben.

```
#!/bin/sh

MY_DIALOG='
<window title="Agent Authentication"
width-request="300"
resizable="false">
   <vbox>
      <frame Agent ID & Passphrase>
         <vbox>
            <hbox>
               <vbox homogeneous="true">
                  <entry activates-default="true">
                     <default>'$USER'</default>
                  </entry>
                  <entry has-frame="false"
                 visibility="false"
                 invisible-char="63"
                    primary-icon-name="password"
                    primary-icon-tooltip-markup="Enter your
                      <span fgcolor='"'red'"'>
                    <b><i>Passphrase</i></b></span>">
                    <default>password</default>
                   </entry>
                 </vbox>
                 <vseparator></vseparator>
                 <vbox>
                     <pixmap>
                        <visible>"true"</visible>
                        <input file>"fbi-image.png"</input>
                     </pixmap>
                  </vbox>
               </hbox>
            </vbox>
         </frame>
         <hbox homogeneous="true">
            <button>
               <label>Submit Credentials</label>
               <action>EXIT:ok</action>
            </button>
         </hbox>
      </vbox>
</window>
'
export MY_DIALOG

gtkdialog --program=MY_DIALOG
```

Az előző kódban néhány sor elég hosszú, ezért több részre kellett vágnom. Ez megengedhető, de a szöveg tördelése sortörést vihet be, ami szándékozottól eltérő szövegmegjelenést adhat. Arra is ügyelj, hogy a nyitó és záró jelölők illeszkedjenek, mert a Gtkdialog ebben nem nagyon nézi el a hibákat.

Az első beviteli dobozban az 'activates-default' jellemző értéke 'true' (igaz). Ezzel indításkor aktívvá válik. Az alapszöveg a USER változóból jön. A '$' jelzi a widget-nek, hogy a USER tartalmát jelenítse meg és ne a szót magát. A rendszer itt tárolja az operációs rendszerbe bejelentkezett felhasználók neveit. Ez a sor elnyomható, üres felhasználói mezőt megjelenítve.

A következő beviteli doboz sokkal érdekesebb. Eltávolítottam a keretét, így simán jelenik meg a párbeszédben. Ezután a 63-as ASCII karakter használatára utasítottam (kérdőjel) a jelszóként beírt karakterek elrejtésére. Az elsődleges ikon bal oldalt van és a rendszer ikonkészletéből való. A rendelkezésedre álló ikonkészlet függvényében ez változhat. Az ikonhoz tartozó eszköztipp sima szövegként indul, majd jelölők hatására betű pirosra, félkövérre és dőltre változhat, a szükséges kiemelés függvényében. Beírtam alapszövegként a „password” szót, így láthatod a maszkolás hatását. Rendesen üresen szokás hagyni és a felhasználó tölti ki.

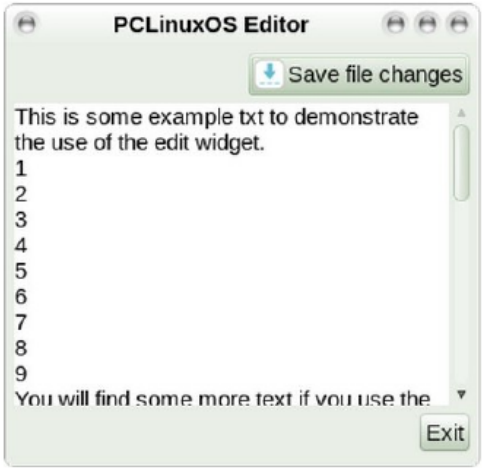
A kód többi része olyan, amit korábban már láttunk és a párbeszéd formálására, dekorálására való.

A szerkesztő widget

Ezt akkor használjuk, amikor többsoros szöveggel dolgozunk. A szöveg beírható billentyűzetről egyesen a dobozba, vagy létező fájlból betölthető. Igény szerint mentheted fájlba is. A widget-ben megjeleníthető mind vízszintes, mind függőleges gördítősáv a szöveg elhelyezéséhez. Támogatja a beillesztést, vagy felülírás módot, a sortörést és szövegellenőrzést.

A következő kis példa bemutatja a widget alapvető alkalmazását.

```
#!/bin/sh

touch myfile

EDIT_DIALOG='
<window title="PCLinuxOS Editor" resizable="true">
     <vbox>
          <hbox>
       <button image-position="0">
              <label>"Save file changes"</label>
              <input file stock="gtk-save"></input>
              <action function="Save">edit1</action>
       </button>
          </hbox>
          <hbox>
       <edit wrap-mode="2" vscrollbar-policy="1">
              <output file>myfile</output>
              <variable>edit1</variable>
              <width>300</width>
              <height>200</height>
              <input file>myfile</input>
       </edit>
          </hbox>
     </vbox>
</window>
'
export EDIT_DIALOG

gtkdialog --program=EDIT_DIALOG
```



A fentiek nagy részével a korábbi példákban már foglalkoztunk, de az „edit” (szerkesztés) jelölőknél meghatározott két jellemző némi magyarázatot igényel. A sortörésnél a következő négy érték egyike használható: 0 nincs sortörés, 1-es a sort a látható szövegterület végénél töri, ha kell a szóval együtt, 2-es csak szó végénél töri a sort és 3-as a 2-es erősebb változata, ami figyelembe veszi az egyes nyelvekben megtalálható különleges írásjeleket, mint a kötőjeleket, angol szöveggel is biztonságosan használható. Hasonlóképpen a 0, 1 és 2 a vscrollbar-policy különleges jelentéssel bír: 0 - mindig; 1 - csak szükség esetén; 2 - soha sem jelenik meg a függőleges görgetősáv. A vízszintes görgetősávot a hscrollbar-policy szabályozza. A 'Save file changes' gomb a mentést hajtja végre, a szerkesztő ablak aktuális tartalmát az 'output file' jelölővel meghatározott nevű fájlba mentve.

A Terminal widget

A következő példa újra felhasználja az előző példákban bemutatott kódok nagy részét, lehetővé téve egy hasznos eszköz gyors létrehozását, a kerék újra feltalálása nélkül. Lesz egy terminálunk és egy szerkesztő ablakunk. Ha a terminálba parancsot gépelünk be, végrehajtja. Azonban, ha a parancsokat átirányítjuk a 'results' (eredmények) fájlba a > (felülírás), vagy a >> (hozzáfűzés) jeleket használva, akkor a szerkesztő ablak automatikusan felismeri a változásokat és frissíti a fájl tartalmát. A parancs kimenete folyamatosan a rendelkezésünkre áll szerkesztésre. A 'save file changes' gombra kattintva a szerkesztett fájlt, az eredeti 'results' fájl felülírásának elkerülése érdekében, amire esetleg még szükséged lehet, a 'results.new' nevű fájlba menti.

```
#!/bin/sh

touch results

EDIT_DIALOG='
<window window_position="2"
title="PCLinuxOS Editor"
resizable="true">
      <vbox>
          <hbox>
       <button image-position="0">
              <label>"Save file changes"</label>
              <input file stock="gtk-save"></input>
              <action function="Save">edit1</action>
       </button>
          </hbox>
          <hbox>
       <terminal has-focus="true" argv0="/bin/bash">
              <height>"10"</height>
              <width>"80"</width>
       </terminal>
          </hbox>
          <hbox>
       <edit wrap-mode="0"
          hscrollbar-policy="1"
          vscrollbar-policy="1"
          auto-refresh="true">
                  <variable>edit1</variable>
                  <width>300</width>
                  <height>500</height>
                  <input file>results</input>
                  <output file>results.new</output>
       </edit>
          </hbox>
          <hbox>
       <button>
              <label>Exit</label>
<input file stock="gtk-quit"></input>
              <action>EXIT:ok</action>
       </button>
          </hbox>
      </vbox>
</window>
'
export EDIT_DIALOG

gtkdialog --program=EDIT_DIALOG
```

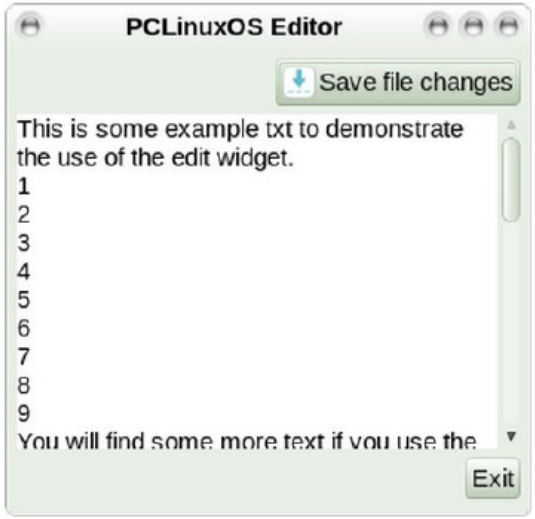
Menük

A menüket a gtkdialog-ban egy, vagy több menü widget-et tartalmazó menüsáv widget segítségével állítják elő. Mindegyik menü widget menüelem widget-eket, menüelválasztó widget-eket és más, beágyazott menüelem widget-et tartalmaz.

A mi menürendszerünk elég egyszerű. Két menüelemet tartalmaz: 'File' és 'Help'. A fájl menüben lesz egy 'Save', a záró 'Quit' menüelemtől egy szeparátor menüelválasztó widget-tel elválasztva. A 'Help' menüben egyetlen 'About' menüelem lesz (ahol a felelősségvállalásról nyilatkozunk). Minden menühöz egy

ALT billentyűvel kombinált billentyűkód és minden menüelemhez egy a CTRL billentyűvel kombinált gyors elérést rendelünk hozzá. A billentyűkódot a menüben aláhúzott karakter jelzi, a gyors elérést a menüelemben szöveg írja le. Minden menüelemhez szöveg és ikon is egyaránt tartozik.

Ez nem tűnhet túl bonyolultnak, de hála a gtkdialog-nak, ezt a példát sablonként használva elég összetett menürendszereket lehet létrehozni minimális programozói tudással.

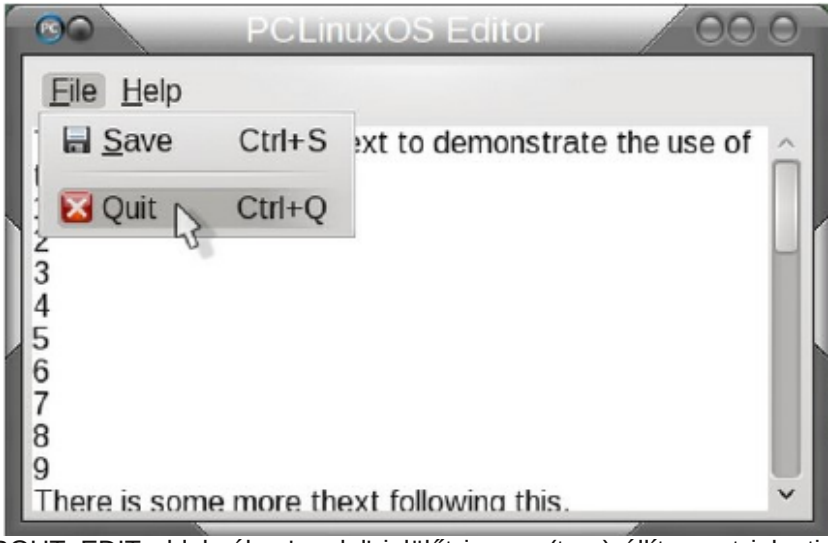



```
#!/bin/sh

touch myfile

ABOUT_EDIT='
<window modal="true"
    image-name="system-help.png"
    window_position="1">
<vbox>
   <pixmap height-request="80">
      <input file>"/home/me/gtk/pcloslogo.png"</input>
   </pixmap>
   <text height-request="50" use-markup="true">
       <label>"<b><span size='"'x-large'"'
           color='"'blue'"'>Edit
        </span></b>"
        </label>
   </text>
   <text width-request="150"justify="2">
       <label>"A text editor written for PCLinuxOS Magazine
          by Me"</label>
   </text>
   <text height-request="50"justify="2">
        <label>"me@my-email.mag"</label>
   </text>
   <button ok>
       <action>closewindow:ABOUT_EDIT</action>
   </button>
</vbox>
     <variable>ABOUT_EDIT</variable>
     <action signal="focus-out-event">
        presentwindow:ABOUT_EDIT</action>
</window>'

EDIT_DIALOG='
<window title="PCLinuxOS Editor"
     image-name="pclinuxos.png"
     resizable="true"
     window_position="1">
<vbox>
     <menubar>
       <menu label="_File" use-underline="true">
         <menuitem stock-id="gtk-save" accel-key="0x53"
                 accel-mods="4">
           <action function="Save">edit1</action>
         </menuitem>
         <menuitemseparator></menuitemseparator>
         <menuitem stock-id="gtk-quit" accel-key="0x51"
                 accel-mods="4">
            <action>exit:Quit</action>
         </menuitem>
       </menu>
       <menu label="_Help" use-underline="true">
          <menuitem stock-id="gtk-about" accel-key="0X48"
                   accel-mods="9">
          <action>launch:ABOUT_EDIT</action>
          </menuitem>
       </menu>
     </menubar>
     <hbox>
       <edit wrap-mode="2" vscrollbar-policy="1">
           <output file>myfile</output>
           <variable>edit1</variable>
           <width>400</width>
           <height>200</height>
           <input file>myfile</input>
        </edit>
       </hbox>
     </vbox>
```

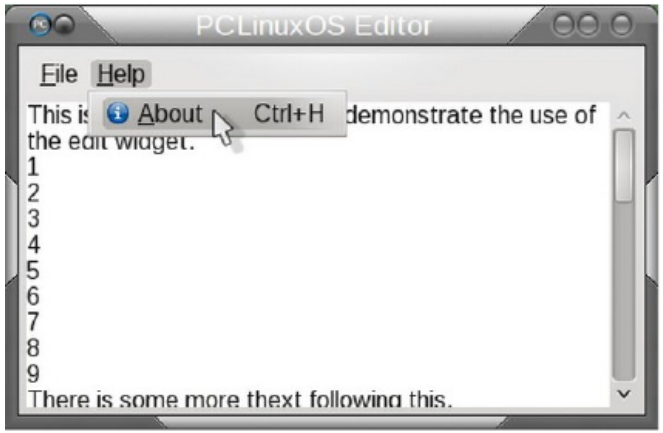
```
</window>
'
export EDIT_DIALOG
export ABOUT_EDIT
gtkdialog --program=EDIT_DIALOG
```

Ezúttal vannak itt egyebek is. Két szkript írja le a megjelenítendő ablakot. Az egyik az EDIT_DIALOG nevű, ez lesz a fő ablak, a másikat pedig ABOUT_EDIT-nek hívják, ami csak akkor jelenik meg, amikor a Help → About menüelemre kattintunk. A fő ablak a korábban már használt szerkesztő, amihez menüsort adtunk, illetve a 'save fájl changes' gombot eltávolítottuk, mivel a továbbiakban nem kell. A menü 'useunderline' jelölője állítja be, hogy billentyűparancs a 'label' jelölőben az aláhúzás után szereplő karaktert használja. A menüelem widget-ben a gyors elérés billentyűkombinációját két jelölő segítségével állítja be: 'accel-key' adja meg a használni kívánt karakter nagybetűs ASCII kódját, az 'accel-mods' értékét úgy határozzák meg, hogy 1 a Shift-et, 4 a Ctrl-t és 8 az Alt-ot jelenti. Ezek az értékek összeadhatóak úgy, hogy 9 (1+8) jelenti a Shift+Alt-ot. Ez a menüelemnél automatikusan létrehozza a megfelelő szöveget.

A Save menüelem a mentést alkalmazza az 'edit1' nevű widget-re. A Quit menüelem egyszerűen kilép az alkalmazásból, míg az About menüelem indítja az ABOUT_EDIT párbeszédablakot.



Az ABOUT_EDIT ablaknál a 'modal' jelölőt igazra (true) állítva azt jelenti, hogy mielőtt használni tudnád ismét a szülő ablakot, ezt be kell zárni. A 'focus-out-event' jelzés teszi elérhetetlenné a szülő ablakot. Az ablak további része egy vbox-ot takar, amiben különféle szöveges és grafikus elemek találhatók, egy ablakbezáró gombbal egyetemben.



A Listázó és Haladásjelző widget-ek használata

Ez a két widget teljesen független egymástól, de a példában együtt használom.

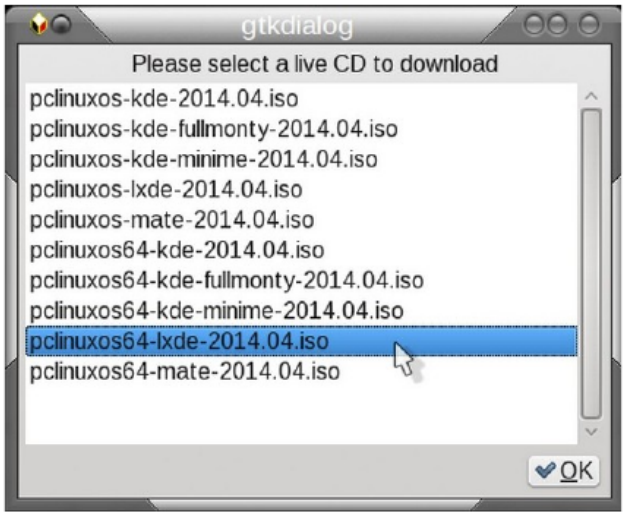
A listázó widget egy szöveges lista elemeit veszi és párbeszédablakban megjeleníti, engedve a egyikük kiválasztását. Megadható, hogy legyen-e görgetősáv és az vízszintes, vagy függőleges-e és indításkor melyik listaelemen legyen a kiemelés. A példában az első elem mutatását (item 0) választottam, de azután egy másik fájlt raktam ki a képernyőképre.

Ebben a példában az egyik PCLinuxOS tárolókból online veszem a listát. A lista a disztribúció pillanatnyilag letölthető összes live CD-jét tartalmazza. Az ezt

végrehajtó kód – attól tartok – nem kezdő szintű, de mint minden ebben a példában, ahhoz kell, hogy megértsd, és így képes legyél alkalmazni.

Amikor a felhasználó kiválasztja az egyik ISO fájlt, a példa szkript letölti, mutatva a letöltés állását százalékosan egy előrehaladást jelző widget-ben. Ha a letöltés megszakadna valamiért, onnan folytatható, ahol a kapcsolat megszakadásakor megállt, az elejétől való újrakezdés szükségessége nélkül. Tartalmaz még egy egyszerű opciót az md5sum ellenőrzőösszeg-fájlok listába történő felvételére, így azok is letölthetők. Íme a szkript.

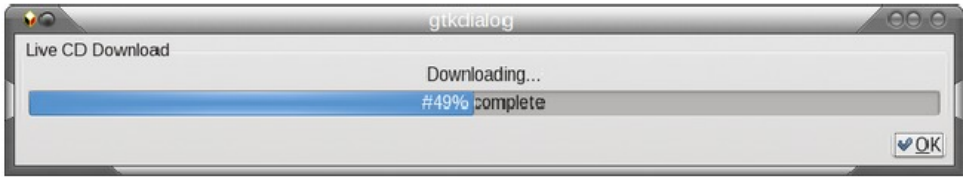
```
#!/bin/bash

# set the repository to use
REPO=http://ftp.nluug.nl/pub/os/Linux/distr/pclinuxos/pclinuxos/live-cd/

# get the html source of the download page from the repo
wget -q -O /tmp/dl_list $REPO

# filter out all but the names of the iso files
cat /tmp/dl_list | sed -n '/"pclinuxos/p' | awk -F"\"" '{ print $6 }'
| sed -n '/iso$/p' > /tmp/iso_list

# or use the line below to include the name of the md5sum checksum files
#cat /tmp/dl_list | sed -n '/"pclinuxos/p' | awk -F"\"" '{ print $6 }'
> /tmp/iso_list

export LIST_DIALOG='
<window window_position="1">
     <vbox>
          <text>
               <label>Please select a live CD to
                   download</label>
          </text>
          <list selected-row="0" vscrollbar-policy="0">
               <width>400</width>
               <height>250</height>
               <variable>LIST1</variable>
               <input>cat /tmp/iso_list</input>
         </list>
         <hbox>
        <button ok>
             <action function="command">echo $LIST1>
                          /tmp/dl_iso</action>
             <action>exit: ok</action>
                  </button>
             </hbox>
        </vbox>
</window>'
```




```
export PROGRESS_DIALOG='
<window window_position="1">
      <vbox>
            <frame Live CD Download>
                  <text>
                        <label>Downloading...</label>
                  </text>
                  <progressbar>
                        <input>wget -c '"$REPO"'`
                              cat /tmp/dl_iso` 2>&1| sed -u
                              "s/.* \([0-9]\+%\)\ \+\([0-9
                              .]\+.\) \(.*\)/\1\n#\1
                              complete/"
                        </input>
                        <action>exit:ok</action>
                  </progressbar>
            </frame>
            <hbox>
                  <button ok>
                  </button>
            </hbox>
      </vbox>
      <variable>PROGRESS_DIALOG</variable>
</window>
'
```

A példa menüben két párbeszéd szkriptet használtunk, az EDIT_DIALOG és ABOUT_EDIT -et. Az

```
# show the dialog to choose a file to download
gtkdialog --program=LIST_DIALOG
# show the download progress dialog
gtkdialog --program=PROGRESS_DIALOG

# clean up - remove all of our temporary files
rm -f /tmp/dl_list
rm -f /tmp/dl_iso
rm -f /tmp/iso_list
# all done
```



EDIT_DIALOG kódja, amikor kell, indítja az ABOUT_EDIT párbeszédet. A két dialógus együtt látható a képernyőn, mivel ez tűnt legcélszerűbbnek. Ebben a példában mihelyst kiválasztottuk a fájlunkat a listából, tovább nincs már szükség a LIST_DAILOG-ra, ezért bezárjuk azt és a PROGRESS_DIALOG párbeszéd kódjára térünk át. Ennek az eljárásnak az előnye, hogy egyszerűsíti, könnyebben olvashatóvá és más alkalmazásokban újrahasználhatóvá teszi a kódunkat.

Ahogyan működik:

Az első sortól eltekintve, ami jelzi a rendszenek, hogy bash héjat használunk a következő kód fordítására, az összes kettős kereszttel „#” kezdődő sor megjegyzés. Sajnos a Gtkdialog-ban a kommentálás csak bizonyos trükkökkel lehetséges, amik ebben a fázisban csak összezavarnák a dolgokat.

Először meghatározzuk a tároló URL-jét (webcím ezúttal), amit a REPO változóban akarunk használni. Ez segít a kódot egy kicsit kevésbé töredezetté tenni.

Ezután a wget paranccsal letöltjük a live CD tárolójának html forráskódját. Ez tartalmazza az összes letöltésre elérhető fájl nevét. A wget parancs nagyon hasznos ilyen helyzetekben, számos opcióval, amik testre szabhatóvá teszik a működését. Esetünkben a -q opciót alkalmazzuk, ezzel utasítjuk a csendes működésre, és hogy a kimenete csak a letöltött fájl legyen, minden zavaró üzenet nélkül. Az -o opció mondja meg a wget-nek, hogy a letöltés kimenetét az ideiglenes /tmp/dl_list-be rakja. Végül, megmondjuk a wget-nek, hogy a REPO változó tartalmát használja annak meghatározására, hogy a fájlt honnan töltse le.

Most dobjuk el a html kódok nagy részét és csak a nekünk szükségeseket tartsuk meg. Az első, a cat parancsot tartalmazó sor csak az „.iso”-ra végződő fájlneveket hagyja meg. A következő két sor, ami itt megjegyzésbe van téve, beveszi az md5sum fájlokat is a listába. Add hozzá, vagy vedd el a kettős keresztet igény szerint. Az újraértelmezett lista a /tmp/iso_list ideiglenes fájlba kerül mentésre. Ha érted, hogyan működik – jó. Ha nem, egyszerűen használd.

A LIST_DIALOG kód a widget-ek szokásos nyelvezetét használja, amit már ismerünk, de tartalmazza az új listázó widget-et is. A megjelenítendő elemlistát a kód <input> jelölőjéből veszi, ami esetünkben a 'cat /tmp/iso_list' parancs, ez egyszerűen elküldi a fájl tartalmát a list widget-nek. A list widget a LIST1 nevet kapta és így a pillanatnyilag kiválasztott elemek listáját tartalmazó változó a LIST1.

Kattintás az OK gombra a LIST1 tartalmát menti az ideiglenes /tmp/dl_iso fájlba és ezután bezárja a LIST_DIALOG-ot. A szkript eztán meghívja a gtkdialog alkalmazást a PROGRESS_DIALOG-gal, mint futtatandó szkripttel.

Ismét csak van egy a szokásos widget-ekből összerakott egyszerű párbeszédünk, amiből csak a progressbar widget új. A haladásjelző widget-nek 0-100 között folyamatosan változó bemenet kell. E bemenet érdekében ismét a wget parancshoz nyúlunk. Most az elég okos -c opciót használjuk, ami megmondja, hogy amennyiben ez egy letöltés folytatása, akkor onnan folytassa, ahol abbamaradt a letöltés. Ez egy nagyon hasznos opció, amikor nagy ISO fájlokat töltünk le. A REPO változónkat ismét használatba vesszük, mint a letöltés forrását és a /tmp/dl_iso fájlt a letöltendő fájl neveként. A kód további része ebben a sorban arra való, hogy az aktuális letöltés százalékát kivéve mindent eltávolítson, hogy a haladásjelző csúszkájának helyzete frissíthető legyen és az aktuális helyzet megjeleníthető legyen.

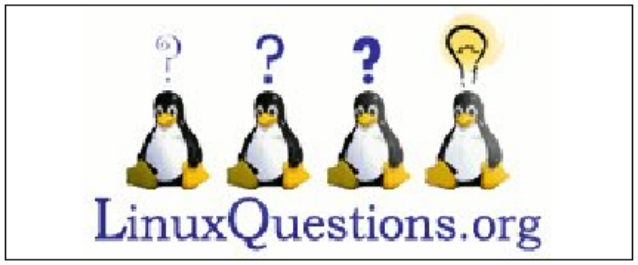
A fájl letöltését követően a három időközben létrehozott ideiglenes fájlt eltávolítja.

Ha úgy döntesz, hogy az ellenőrző összegfájlt letöltöd a nagy .iso fájllal együtt, és valóban szükség van rá, íme így használd. Mindkét fájlnak az aktuális könyvtárban kell lennie.

```
$ md5sum -c pclinuxos64-lxde-2014.04.md5sum
pclinuxos64-lxde--2014.04.iso: OK
```

Ha OK-t kapsz, akkor az .iso fájlt biztonságos használni, ha valami mást kapsz, akkor a letöltéssel valami elromlott.

```
$ md5sum -c pclinuxos64-lxde-2014.04.md5sum
pclinuxos64-lxde-2014.04.iso: FAILED
md5sum: WARNING: 1 computed checksum did NOT match
```

Ezek a példaprogramok távolról sem tökéletesek. Sosem akartak tökéletesek lenni, de működnek. Ez az, amire a gtkdialog widget-ek valók. Arra szolgálnak, hogy grafikus alkalmazást készíts gyorsan és minimális programozási tudással. Megírtam és teszteltem a letöltő alkalmazást két órán keresztül, de nem tudtam további két napot a finomításával szánni. Az alkalmazás hasznos lehet, de mivel feltehetően évente két-három alkalommal használnám, így megfelelő lehet. A Gtkdialog widget kis olyan feladatokra szolgáló eszközök készítésére való, amikre egyébként sokkal nagyobb alkalmazások kellenének, és azok valószínűleg nem úgy működnek, mint ahogy azt szeretnénk.

Ha elmented és készítesz egy hasznos alkalmazást, ami minden szükséges hibaellenőrzést végrehajt és más nyelveket használó felhasználók számára a nyelvi illesztést tartalmazza (Linux alatt éppenséggel elég egyszerű), akkor akár terjesztheted is. Sok ilyen elem van, ami a tárolókban elérhető.