Programozás Gtkdialog-gal: Elsö rész

PCLinuxOS Magazine – 2014. április

Írta: Pete Kelly (critter)

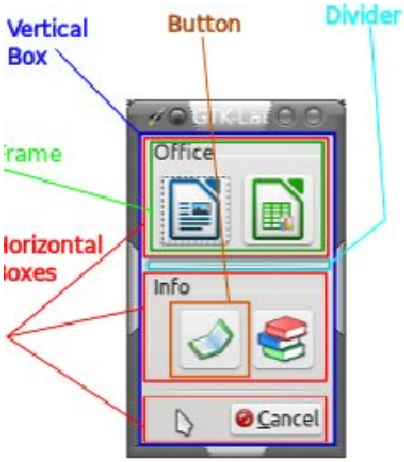
A PCLinuxOS alapkiépítésével alkalmazások százait kapod. A tárolókon keresztül további ezrekhez férsz hozzá. Ez nagyszerű, de valójában csak egy tucatot használsz rendszeresen, miközben sokkal több van telepítve.

Bármelyik kedvenc alkalmazásomhoz a hozzáférés azt jelenti, hogy megnyitom a menüt és számos almenüt, hogy megtaláljam, vagy készítek egy billentyű-parancsot, amire sosem emlékszem. Természetesen készíthetnék hivatkozást is az asztalra, de a tiszta asztalt szeretem. Emellett az asztalt általában az aktuális alkalmazás úgyis takarja.

A probléma megoldására gtkdialog-nak hívott programmal írtam egy kis alkalmazást. Ez a kis program párbeszédet jelenít meg, benne minden alkalmazáshoz egy gomb. Az egyes gombokhoz tartozik egy ikon és az alkalmazást leíró eszköztipp, ami az egérmutató hatására jelenik meg. További előnye, hogy a párbeszéd bármikor megjeleníthető az egérmutató helyénél. A gombok egyikére kattintva a párbeszéd bezáródik és elindul a kiválasztott alkalmazás.

A párbeszéd elemei, a gombok, a feliratok, az elválasztók, stb. úgynevezett widget-ek. A program elkészítéséhez egy bash szkript-et kell készíteni, amit nem kell, hogy megérts – ugyanakkor hasznos, ha érted, csakúgy mint azokat az egyszerű text-eket, amik sztanza-kban (pár sor) írják le a megjeleníteni kívánt widget-eket. A stanzák jelölő tag-eket használnak a gtkdialog-gal való együttműködésre. Néhány példán keresztül ezeket a tag-ek elég jól megértheted.

A megfelelő kialakítás érdekében meg kell értened a szkriptben lévő szöveg struktúráját. Kezdjük egy egyszerű, 2x2-es párbeszéddel, benne egy Cancel gombbal. A folyamat, mérettől függetlenül azonos lesz. Egy sor vízszintesen és függőlegesen elhelyezett téglalapot tartalmaz, amiket Lego-ként kapcsolhatsz egymáshoz. Ezekbe helyezed el a widget-eket gombként. A szöveges stanzá-id leírják a widget megjelenését és feladatát.



A párbeszédablakban a függőleges négyzetekkel kezdjük (kék Lego építőkockák), adjunk hozzá 3-at vízszintesen (vörös téglák), 2 vízszintes elválasztóval (lila téglák, csak egyet emeltem ki a képen). A 2 felső vízszintes négyzethez kereteket adtunk (zöld téglák), amik segítenek leírni és csoportosítani a tartalmat. Ezek a keretek tartalmazzák a felhasználói gombokat (barna téglák) és végül az utolsó négyzetben vízszintesen helyezünk el egy gombot, amivel lehetőségünk van kilépni anélkül, hogy bármit is kiválasztottunk volna.

Gtkdialog-ot megjelenítő kód lehet akár ilyen egyszerű is:

```
#!/bin/bash
export MY_DIALOG='<vbox><button cancel></button></vbox>'
gtkdialog --program MY_DIALOG
```

Ez az szkript a fenti egyszerű párbeszédet jeleníti meg. A gombra rákattintva eltünteti. Mit vártál?

A szöveget egyszerű szövegszerkesztőben megírhatod, de az olyan programozói szerkesztő mint a kate, SciTE, vagy vim sokkal kényelmesebb. A szöveget mentés után végrehajthatóvá kell tenni. Kattints jobb billentyűvel a fájlkezelődben a fájlon, válaszd ki a tulajdonságokat, engedélyeket és jelöld be a végrehajtható négyzetet. Alternatív lehetőség, hogy terminálban beírod type **chmod u+x parbeszedem**, vagy aminek hívtad.

Ha jobban megnézzük ezt a rövid szkriptet, ezt látjuk:

1. sor: megmondja a rendszernek, hogy bash-sel dolgoztatjuk fel a szkript-et.

2. sor: rakja be a szöveget, ami a programunk, egy változóba, vagy tartóba, ezt MY_DIALOG-nak hívom és exportálja, hogy a gtkdialog alkalmazás elérhesse. Te hívhatod, aminek akarod.

A szöveg közli a gtkdialog-gal, hogy rajzoljon egy függőleges négyzetet (a <vbox> tag teszi) a párbeszédbe, és helyezzen el egy Cancel gombot a vbox-ba (a <button> tag teszi). A Cancel gomb egyike az előre definiált gtkdialog gomboknak. A többi a Yes, a No, az OK és a Help. A </button> és a </vbox> tag-ek zárják le a button és a vbox definícióját, szükség szerint.

3. sor: végül meghívja a gtkdialog alkalmazást, utasítva a MY_DIALOG változóba rakott szöveges parancs használatára. Vedd észre a „program” szó előtti két kötőjelet.

Ha összetettebb szkriptet akarsz haszni, sokkal több szöveget is írható a 2. sorban a két idézőjel közé, de ennek két hátránya is van. Először is, nehéz lesz karbantartani, vagy módosítani, másodszor a shell szkript-rész eléggé korlátozott.

A következőkben olvasható a szkript átírva, hogy ezt a problémát kezelje. Nem kell megértened a szürkével szedett shell szkript-részt, az csak a gtkdialog „csomagolására” szolgál. Csupán az idézőjelek közötti szöveg az érdekes nekünk, és ezt is egyszerűsítjük.

```
#!/bin/bash
DISPLAY=":0"
exec 9>/tmp/pid
if ! flock -n 9  ; then
    zenity --warning --text="GTK-Launcher már fut!";
    exit 1
fi
export MY_DIALOG='
<vbox>
    <button cancel>
    </button>
</vbox>'
gtkdialog --program=MY_DIALOG
rm -f /tmp/pid
exec 9>&-
```

Behúzást alkalmaztam a szövegben, hogy elkülönítsem a különböző widget-eket, amitől a követő kód egy kicsit egyszerűbb lesz. Még jobb ötlet, szintaxis kiemelést ismerő szövegszerkesztő használata. A legtöbb szövegszerkesztők mint a kate, scite, vagy gvim ismeri, ugyanakkor keresni kellett olyat, amelyik a gtkdialog-típusú kódokat támogatja alapból. A munkakörnyezetben én általában nyelvként HTML-t, vagy XML-t állítok be, ami általában segíti a kiemelést. Nem tökéletes, de segít.

Akiket érdekel, a shell csomagoló szkript-rész így működik:

DISPLAY=”:0” közli az X windows rendszerrel, hogy melyik képernyőn kell a párbeszédet megjeleníteni, végül is a Linux többfelhasználós rendszer. Erre akkor lehet szükség, amikor a szkriptet más alkalmazás ablakából akarod indítani, vagy más bejelentkezés alól.

A kód további része arra való, hogy a szkript egy már futó példányát érzékelje az által, hogy nyitott átmeneti fájlt keres a szkript indításakor, és attól függően cselekszik. Én itt a zenity-t hívom meg figyelmeztető üzenet megjelenítésére, amihez itt készítettem egy másik gtkdialog-ot. Az ok egyszerű, működik és azonos stílusú fejlécet használok a többi szkriptben is, ami jó programozói gyakorlat.

Oké, nézzük meg a 2x2-es indítóeszközt előállító kódunkat, a szürkével kiemelt részt kihagyva ez olvasható (a MY_DIALOG változót átneveztem LAUNCHER-re). A szövegszín a párbeszéd rajzában szereplőknek felel meg.

```
export LAUNCHER='
<window window_position="2" title="GTK-Launcher" icon-name="fork"
resizable="false">

<vbox>
  <hbox>
      <frame Office>
            <hbox>
            <button tooltip-text="Word Processor">
```

```
                    <input file>/usr/share/icons/
                         libreoffice4.2-writer.png</input>
                    <action>libreoffice4.2 --writer
                            &</action>
                    <action>rm -f /tmp/pid</action>
                    <action>EXIT:ok</action>
                  </button>
                  <button tooltip-text="Spreadsheet">
                    <input file>/usr/share/icons/
                         libreoffice4.2-calc.png</input>
                    <action>libreoffice4.2 --calc
                            &</action>
                    <action>rm -f /tmp/pid</action>
                    <action>EXIT:ok</action>
                  </button>
                </hbox>
      </frame>
</hbox>
<hbox>
      <hseparator width-request="140"></hseparator>
</hbox>
<hbox>
      <frame Info>
                <hbox>
                  <button tooltip-text="Address Book">
                    <input file>/usr/share/icons/
                           kaddressbook.png</input>
                    <height>40</height>
                    <action>kaddressbook &</action>
                    <action>rm -f /tmp/pid</action>
                    <action>EXIT:ok</action>
                  </button>
                  <button tooltip-text="PCLinuxOS
                                        Documentation Portal">
                    <input file>/usr/share/icons/
                              bookcase.png</input>
                    <height>40</height>
                    <action>pclos-docuportal &</action>
                    <action>rm -f /tmp/pid</action>
                    <action>EXIT:ok</action>
                  </button>
                </hbox>
      </frame>
</hbox>
<hbox>
      <hseparator width-request="140"></hseparator>
</hbox>
<hbox>
      <button cancel>
        <action>rm -f /tmp/pid</action>
        <action>EXIT:Cancel</action>
      </button>
  </hbox>
</vbox>
</window>'
gtkdialog -program=LAUNCHER
```

Itt egy kicsit csaltam, hogy a szöveg az oldalra kiférjen, de az ikonokhoz hasonló külső elemek meghatározásakor neked a teljes (fájl) elérési címet kell használnod.

Láthatod, hogy a következetesen alkalmazott behúzások és szintaxis kiemelések segítik a kód könnyebb olvashatóságát. A Lego-analógiával élve ismét, vannak piros <hbox>, barna <button>, zöld <frame> és így tovább, építőköveink, tégláink. Egyszerűen összepattintjuk a szkript összeállításához.

Legfölül található egy <window... kezdetű sorunk.

Ez állítja be a képernyőn megjelenő gtkdialog ablakot.

window_position="2" Position 2 jelzi, hogy a kurzor körül központosítva induljon.*

title="GTK-Launcher" A title (cím) jelenik meg az ablak címsávjában.

icon-name="fork" Az itt megadott ikon jelenik meg az ablak címsorában és az indítósávban.

resizable="false" A párbeszéd nem méretezhető át.

**Window_position 1 esetén az (aktuális) ablakot közepére rendezi. Position 0-nál az alapképernyőn jobbra lent lesz, de ez X és Y eltolás meghatározásával módosítható, ha meghatározott helyre kell rakni.*

Az ablak widget-nek csak egy leszármazottja lehet. Emiatt, általában vbox, vagy hbox widget-tel indítunk, amiknek lehet több leszármazott widget-je is.

Ezután egy hbox-ot tartalmazó vbox jön, ami tartalmaz egy „Office” című keretet. Ebben van egy további hbox két gombbal.

Kiemeltem azt a stanzát, ami teljes egészében leírja az első gombot.

A gomb fölé húzva az egeret, „Szövegszerkesztő” felirat jelenik meg. A használt ikon a /usr/share/icons/libreoffice4.2-writer.png. Rákattintva három művelt indul:

1. Elindul a LibreOffice writer. Az & utasítja a szkriptet a writer indítása után a folytatásra, nem várva, amíg a writer befejezi futását, amitől az indítót nem lehetne tovább használni.

2. Törli a bash szkripttel készített ideiglenes fájlt, amit azért hoztunk létre, hogy az indítót ne lehessen több példányban futtatni. Ez akkor kell, ha az én itteni kódomat használod, vagy ha az ideiglenes fájl megakadályozná az alkalmazás további futtatását.

3. Kilép az alkalmazásból, „OK” állapotot jelentve vissza.

Ha kész a szkripted, akkor kell valamilyen eljárás a könnyű indításhoz. A Thinkpad laptopomon van egy nagy gomb, ami csak a rendszer indításánál használatos (ezt korábban a windows használta, mielőtt kilakoltattam volna). Ez egy kiváló alany. Egyszerűen lenyomom a gombot és megjelenik az alkalmazás az egérmutató helyénél. Az asztali gépemen is van egy hasonló, haszontalan gomb – a számológép gomb. Egy kis fantáziával a számológép logója hasonlít az indító alkalmazásra, tehát azt használom. A legtöbb asztalkezelőnél megoldható billentyűlenyomás összekötése alkalmazásindítással, vagy az xbindkeys-zel beállítható, lásd itt.

Tovább bonyolíthatod az alkalmazást egyszerűen ilyen stanzák hozzáadásával, mint vbox, frame, és elválasztó, szükség szerint. Egyenként add hozzá az egyik oldalról és minden hozzáadás után tesztelj, mivel bármilyen hiba, pl. hiányzó tag, a szkriptet lebéníthatja. Újabb sor ikon kell? Kattints egy piros építőkockára, egy zöld építőkockát bele, majd egy másik pirosat, amiben valami barna kerül a zöldbe, szerkeszd a stanza szövegét és készen vagy.

Minek a plusz hbox a kereten belülre? Próbáld kivenni és nézd meg mi lesz.

Az elemek méretét a gtkdialog általában automatikusan kezeli, de felülbírálhatod, ahogy én is tettem az „Info” frame esetében, hogy a gombok jobban illeszkedjenek az ikonjaikhoz és az alkalmazás általános kialakítását meghatározó vízszintes elválasztókhoz.

> Hallhattál róluk, vagy láthattál is már szkriptet, amik Zenity-t és YAD-ot használtak Gtkdialog helyett. A Zenity és a YAD történetesen két olyan program, ami a Gtkdialog parancsokat egyszerűbb használat érdekében „csomagolja”. Habár használhatod ezeket a csomagoló programokat, de pontosabban és több opcióval állíthatod be a Gtkdialog közvetlen használatával.

Természetesen azt is tudnod kell, hogy mit rakjál a stanzákba, de ezeket simán át is másolhatod, csak a tippet, az ikont és az indítandó alkalmazást cseréled ki. Működni fog, de vannak további opciók, amikkel személyesebbé teheted a párbeszédablakod. A gtkdialog leírása elég hiányos. Vannak példakódok az összes widget bemutatásához a /usr/share/doc/gtkdialog/examples könyvtárban, de lehet, hogy ehhez újra kell telepíteni a gtkdialog-ot a tárolóból. Ez eddig fellelt legrészletesebb online leírás a http://code.google.com/p/gtkdialog/w/list.

Van egy kitűnő leírás még a Puppy Linux fórumán http://murga-linux.com/puppy/viewtopic.php?t=38608 is.

Ezek a leírások és példák különösen hasznosak és érdekesek lehetnek.

Ez a kis alkalmazás csak egy kis példa arra, hogy mit érhetsz el a meglévő dolgokkal és ehhez többnyire nem kell más, mint egy kis sima szöveg és józan ész. Miközben nem valószínű, hogy lecseréled az olyan nagyobb alkalmazásokat, mint a Gimp, vagy a LibreOffice, sok mindent lehet tenni az életünk egyszerűbbé tétele érdekében a gtkdialog-hoz hasonló alkalmazásokkal.

Ez a rövid bemutató csak súrolja a felszínt. Sokkal több van még, de te azt úgyis tudod. Végül is ez a Linux.