



Néhány érv az Xdialog mellett és ellen

Grafikus felhasználói felületek könnyedén és gyorsan héjprogramozással.

Az Xdialog fejlesztése azzal a céllal indult, hogy olyan programot hozzanak létre, amellyel a dialog vagy a cdialog adta lehetőségeket használó parancsfájlokat módosítás nélkül grafikká varázsolhatjuk. Az Xdialog az említett alkalmazások összes elemét kezeli és ezek mellett számos újat is bevezet. Az utóbbiak között említhetjük a fájl- és könyvtárválasztót, a fanézetet, a súgó gombot – és még sorolhatnám. További előny, hogy az Xdialog a GTK+ eszközkészletet használja, így a segítségével létrehozott grafikus parancsfájlnak alkalmazkodhat a felhasználó által választott asztaltémához. A projekt honlapja a <http://xdialog.dyns.net/> címen érhető el.

Beszerezés és telepítés

Az 57. CD Magazin/Xdialog könyvtárban található a legfrissebb változat, amely a cikk írásának pillanatában a 2.1.1-es. A forrás *.tar.gz* és *.tar.bz2* formában tömörítve, továbbá SRPM formátumban érhető el. Több felépítésre található előfordított RPM csomag is. A *.deb* csomagok egy kicsit lemaradtak az RPM-hez képest, ugyanis a 2.0.6-os volt közülük a legújabb. Az én Debian 3.0 r1 változatomban megtalálható 2.0.5-ös óta a napló tanúsága szerint nem történt sok változás, ezért a kipróbáláshoz is ezt használtam. A forrásból történő telepítés sem egy ördögösség, a megszokott `./configure && make && make install` biztos eredményre vezet.

Általános használata

```
xdialog [<GTK+ opciók>] [<általános opciók>]
↳ [<átmeneti opciók>] <dobozopció> ...
```

vagy

```
xdialog <különleges opció>
```

Az első esetben az [<általános>] [<átmeneti>] <doboz> sorozat tetszőleges számban ismétlődhet egyetlen Xdialog-sorban – ezt hívják dobozláncolásnak. Az általános és átmeneti beállítások (option) elhagyhatók, de utolsóként minden Xdialog-sorban egy dobozbeállításnak kell szerepelnie.

- Egy <általános beállítás> minden utána következő <dobozbeállítás>-ra érvényes, amíg egy másik

<általános beállítás> ki nem oltja a hatását. Ezek a beállítások az ablak címére, a gombok stílusára és hasonlókra vonatkoznak.

- Az <átmeneti beállítások> csak a közvetlenül utánuk következő <dobozbeállítás>-ra érvényesek. Ezek a beállítások többnyire az adott doboz különleges tulajdonságaira vonatkoznak.
- A <doboz beállítás> határozza meg, hogy az Xdialognak milyen elemet kell megjelenítenie. Legalább három értéket vár: egy szöveget, valamint a rajzolendő ablak magasságát és szélességét képpontban megadva.

Amikor a felhasználó egy ablakot bezár, az Xdialog azt az eredményt egy vagy több karakterlánc formájában a szabványos hibacsatornára küldi. Egy általános beállítással ez a szabványos kimenetre is átirányítható. Az Xdialog visszatérési értéke hordozza magában az információt, hogy a felhasználó hogyan zárta be az utolsó ablakot. Ez lehet 0, ekkor az *Ok* vagy *Igen* gombra kattintott. 1 esetén a *Mégsem* vagy *Nem* gombot választotta.

1. lista Mi a neved?

```
#!/bin/bash

DIALOG="/usr/bin/xdialog"
CIM="Mi a neved?"

NEVED=`$DIALOG --title "$CIM" --stdout
↳ --inputbox "Hogy hívnak?" 0 0`

case $? in
  0)
    $DIALOG --title "$CIM" --msgbox
    ↳ "Szia $NEVED!" 0 0
    ;;
  1)
    $DIALOG --title "$CIM" --msgbox
    ↳ "Talán legközelebb..." 0 0
    ;;
  255)
    echo "Kilépés."
    ;;
esac
```

255-ös visszatérési értéket kapunk, ha valami hiba történt vagy ha az alkalmazást az ablakkezelő rendszeren keresztül zárták be. További értékek is lehetségesek: a `--help` és `--wizard` átmeneti kapcsolók esetén. Dobozok láncolásakor az első olyan doboznál, amelynél a visszatérési érték nem nulla, a lánc megszakad.

Amikor az `Xdialog`ot különleges beállítással használjuk, semmilyen értéket nem vár el, mindössze adattal szolgál a szabványos hibacsatornán. Próbáld ki, mi történik, ha az `Xdialog`nak nem adsz át más értéket, csak a `--print-version!`

Szia, világ!

Minden program itt indul. Lássuk, mire képes a `--msgbox` dobozbeállítás!

```
$ xdialog --msgbox
'Szia világ!' 0 0
```

Ne feledd, hogy az `Xdialog` program nevének az első betűje nagybetű. A `--msgbox` három értéket vár: az első az a szöveg, amely megjelenik a dobozban. A továbbiak az ablak magasságát és szélességét jelölik ki, ezek értékeként adhatasz 0-t, ekkor az ablak önműködően akkora lesz, hogy a szöveg éppen beleférjen. Ha a szöveg sortörést tartalmaz, az az ablakban is ugyanúgy fog megjelenni. Ezenkívül a `'\n'` különleges karaktert használhatod a sortörések kijelölésére (1. kép, lásd a 71. oldalon).

Mi a neved?

Ezek után lássunk egy valódi alkalmazást! Egy olyan parancsfájlt fogunk írni, amelyik megkérdezi a felhasználó nevét, majd kiírja a képernyőre (lásd az 1. listán és a 2. képen). Lássuk sorról sorra a parancsfájl működését! Az első sor mindössze a futtatandó parancsértelmezőt határozza meg – ez régi barátunk, a Bourne Again Shell. Az `Xdialog` nevét érdemes egy változóban tárolni. Előfordulhat, hogy úgy döntünk, nincs szükség a grafikus csillogásra, és megelégszünk a szöveges üzemmód nyújtotta menüvezérelt felületekkel. Ekkor vissza kell térnünk a `dialog` vagy a `cdialog` használatához. Szerencsére ezek többé-kevésbé megfeleltethetőek egymásnak, a futtatandó program nevét azonban így is át kell írunk. Ezt könnyebb egyetlen helyen megtennünk, mint az egész programban, ezért fontos az előrelátás.

Mivel több ablakot használunk a programban és szeretnénk ragaszkodni az egységes kinézethez, az ablakok címét is egyetlen változóban tároljuk. A következő sor már az `Xdialog`hoz fordul a név bekéréséhez. Az ismerős ``` műveletjelet (operator) használjuk, ami az idézőjelek között szereplő parancsot futtatja és a kimenetét az egész kifejezés helyére helyettesíti be. Az `Xdialog --stdout` általános beállítása lehetővé teszi az alapértelmezésben

a szabványos hibacsatornára érkező adatok átirányítását, így a felhasználó által begépelte adat az ablak bezárása után a szabványos kimeneti csatornán fog szerepelni, s így a `$NEVED` változó tárolja. Ebben az új `Xdialog`-sorban egy új általános kapcsolót láthatasz, a `--title-t`, amellyel az ablak címét írhatjuk át.

A kérdésfeltevés után meg kell vizsgálnunk, hogy milyen módon zárta be az ablakot a felhasználó. Az `Ok` gomb megnyomása esetén a programnak másként kell eljárnia, mint a `Mégsem` gomb választásakor, ezért a `?` különleges változó értékét vizsgáljuk meg. Ez a változó a `bash`-ben az utoljára futtatott program visszatérési értékét tartalmazza. A fentebb említettek alapján választjuk szét az eseteket: normál esetben a név beírását az `Ok` gombra

történő kattintás követi – ekkor egy üzenetdobozban köszöntjük a felhasználót. A `Mégsem` megnyomása esetén az udvariatlan felhasználótól névtelenül köszönünk el. Ha pedig a bal felső sarokban látható `X`-szel zárta be az ablakot, akkor csak a szabványos kimenetre írunk ki egy üzenetet.

Hasznos beállítások

Most csupán néhány fontosabb beállítást említenék meg, bár ezeknél jóval több létezik – mind megtalálhatod őket a kényelmesen használható HTML alapú leírásban. A `dialog` és `cdialog` parancsfájlokban a `--backtitle <szöveg>` (általános) a beállítással a háttérre, azaz magára a terminálra lehetett fejléct kiírni. Grafikus felület alatt a háttérre nem illik írni, de a megfeleltethetőség érdekében a beállítást meg kellett tartani. A szöveg ezzel a beállítással jelenik meg az ablakban, majd egy elválasztójelel után továbbiak következnek.

```
--allow-close | --no-close (általános)
```

A `--allow-close` alapértelmezett beállítás, az ablaknak az ablakkezelőn keresztül történő bezárását teszi lehetővé. Ebben az esetben az `Xdialog` 255-ös visszatérési értékkel ér véget. A `--no-close` megadása esetén az `Xdialog` figyelmen kívül hagyja az ablakkezelő bezárt eseményeit. Megjegyzendő, hogy az ablak a töröl esemény kiváltásával így is bezárható.

```
--no-buttons (átmeneti)
```



2. lista Könyvtárfa mentése

```
#!/bin/bash

TITLE="könyvtárfa"

MSG=""
Ez a program egy könyvtárról készít egy listát.
1. lépés: válaszd ki a könyvtárat
2. lépés: válaszd ki az új adatbázist
"
xdialog --title "$TITLE" --noclose --stdout
↳--msgbox "$MSG" 0 0

BACKTITLE="Válaszd ki a könyvtárat"
DIR=`xdialog --backtitle "$BACKTITLE"
↳--title "$TITLE" --no-close --stdout
↳--no-buttons --dselect "" 0 0`
if [ 1 -eq $? ]; then
    exit 1
fi

f=1
while [ 1 -eq $f ]; do
    BACKTITLE="Válaszd ki a célfájlt"
    FILE=`xdialog --backtitle "$BACKTITLE"
↳--title "$TITLE" --no-close --stdout
↳--fselect "" 0 0`
    if [ 1 -eq $? ]; then
        exit 1
    fi

    if [ -f $FILE ]; then
        xdialog --title "$TITLE" --no-close
↳--stdout --default-no --yesno
↳"A cél fájl létezik. Felülírjam?" 0 0
        if [ 0 -eq $? ]; then
            f=0
        fi
    else
        f=0
    fi
done

updatedb --localpaths="$DIR"
↳--output="$FILE"; echo "XXXX" |
xdialog --title "$TITLE" --no-close --stdout
↳--no-buttons --infobox "Kérlek, várj,
↳amíg elkészülök..." 0 0
```

Ezzel a beállítással lehet elérni, hogy ne jelenjen meg az *Ok* vagy a *Mégsem* a `--infobox` dobozban, illetve a *Könyvtár létrehozása*, a *Fájl törlése* és a *Fájl átnevezése* gombok a `--fselect`, `--dselect` dobozokban.

`--default-no` (átmeneti)

A *Nem*, illetve a *Mégsem* gomb lesz az alapértelmezett, vagyis ami előre ki lett választva.

3. lista Könyvtárfa böngészése

```
#!/bin/bash

TITLE="könyvtárfa"

BACKTITLE="Válaszd ki az adatbázist!"
FILE=`xdialog --backtitle "$BACKTITLE"
↳--title "$TITLE" --no-close --stdout
↳--no-buttons --fselect "" 0 0`
if [ 1 -eq $? ]; then
    exit 1
fi

echo `xdialog --title \"$TITLE\" --stdout
↳--treeview \"$FILE\" 0 0 0 \\` >/tmp/dir$$
i=2; IFS=""

PROGRESS="Fa készítése..."
locate -d "$FILE" \* | while read LINE; do
    while [ "" == "`echo "$LINE" | cut -d/
↳-f$i`" ]; do
        i=`expr $i - 1`
    done
    until [ "" == "`echo "$LINE" | cut -d/
↳-f$i`" ]; do
        echo -n "`echo "$LINE" | cut -d/
↳-f-$i`\" >> /tmp/dir$$
        echo -n "`echo "$LINE" | cut -d/
↳-f-$i`\" >> /tmp/dir$$
        echo " off `expr $i - 2` \\"
↳>> /tmp/dir$$
        i=`expr $i + 1`
    done
    echo -n "."
done | xdialog --title "$TITLE" --progress
↳"$PROGRESS" 0 0 `locate -d "$FILE" \* |
↳wc -l`

echo " ;" >> /tmp/dir$$
./tmp/dir$$
rm /tmp/dir$$
```

Fontosabb dobozok

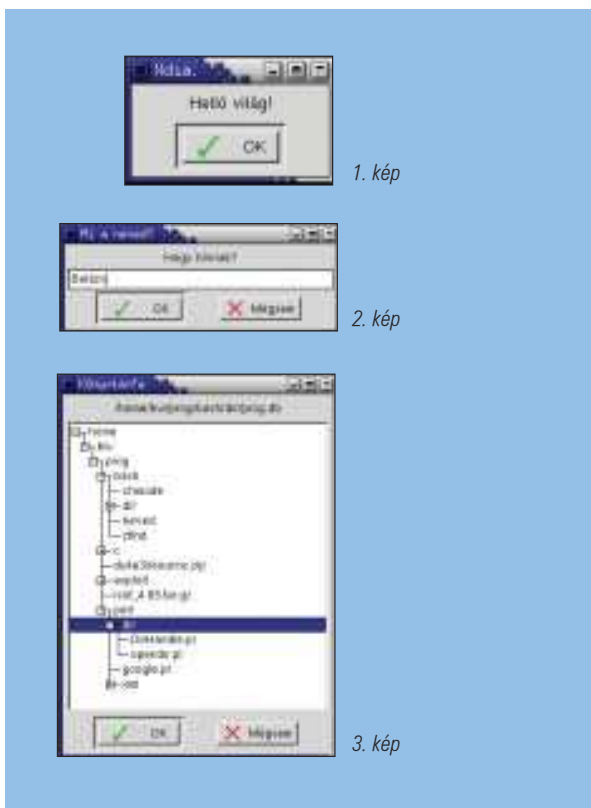
Az alábbiakban tárgyal dobozok jóval többre képesek, mint amennyit itt elárulok róluk. Nem mutatom be az összes kapcsolójukat és helyhiány miatt nem teszek említést az összes átmeneti beállításról sem, amelyekkel használhatók.

`--yesno <szöveg> <magasság> <szélesség>`

A `--yesno` mindössze annyiban különbözik a `--msgbox`-tól, hogy egy *Ok* gomb helyett egy *Igen* és egy *Nem* gombot rak ki a szöveg kiírása után.

`--infobox <szöveg> <magasság> <szélesség> [<idő>]`

Hasonlít a `--msgbox`-ra, de egy, a `--no-buttons` gombok



nélküli ablakhoz jutunk. Továbbá egy nem kötelezően megadandó időértékkel meghatározhatjuk, hogy hány ezredmásodpercig éljen az ablak, utána önműködően bezárul. Ha az időérték 0 vagy elhagyjuk, a doboz másként viselkedik: a szabványos bemenetről fogad üzeneteket.

```
--progress <szöveg> <magasság> <szélesség>
↳ [<max pont>]
```

Ez egy olyan doboz, amelyben egy állapotjelző csík található. A szabványos bemenetről fogad üzeneteket. A <max pont> által jelzett számú pontot vár és ennek megfelelően növeli a csíkot. A <max pont> elhagyása esetén alapértelmezésben 100-at vesz. Pontok helyett számokat is küldhetünk neki a szabványos bemeneten keresztül. Az ablak akkor zárul be, ha a szabványos bemenetre állományvége jel érkezett, vagy a pontok száma, illetve az átadott szám átlépte a <max pont> értékét.

```
--treeview <szöveg> <magasság> <szélesség>
↳ <listahossz> <címke1> <elem1> <állapot1>
↳ <mélység1> ...
```

Ez a doboz az átadott elemekből egy fát épít, amely legfeljebb 24 szint mélységű lehet. A <címken> <elemn> <állapotr> <mélységn> sorozat tetszőleges számban ismétlődhet. Egy elem mélysége az előzőnél legfeljebb eggyel lehet nagyobb, kisebb viszont akár mennyivel. A <listahossz> mondja meg, hogy hány soros legyen az az ablak, amelyben a fát ki kell rajzolni; 0 megadása esetén ez

önműködő lesz. Az *Ok* megnyomása révén a szabványos kimeneten megjelenik a fa kijelölt elemének a címkéje.

```
--fselect <állomány> <magasság> <szélesség>
--dselect <könyvtár> <magasság> <szélesség>
```

Ezekkel a dobozokkal a GTK+ állományválasztóját érhetjük el: egy állomány- vagy könyvtárnevet választhatunk ki velük. Az <állomány> és a <könyvtár> beállítások az alapértelmezést jelölik ki. A --no-buttons segítségével tüntethetők el a *Könyvtár létrehozása*, a *Fájl törlése* és a *Fájl átnevezése* gombok.

A hegycsúcs

Most látni fogod, hogy mennyire egyszerűen lehet összetett feladatokat megoldani az Xdialog használatával. Két parancsfájlt fogunk írni: az első egy tetszőleges könyvtár teljes szerkezetét menti (2. lista), a második képes az első által készített adatbázisokból betölteni (3. lista) a szerkezetet és egy fát készíteni belőle (lásd a 3. képen).

Ha ismered a bash-t és figyelmesen követted az eddigieket, ezek a sorok nem jelentenek meglepetést. Az első a könyvtár és az adatbázis megkérdezése után leellenőrzi, hogy az állomány létezik-e, majd egy türelemre felszólító üzenet mellett elkészíti a könyvtárlistát. A második egy kicsit összetettebb: az adatbázis ismeretében a locate paranccsal az összes bejegyzést kiszedi belőle, s ezeket soronként feldolgozva egy átmeneti állományban előállítja az Xdialog-parancssort. Ezt egy csővezetékkel oldjuk meg, az állapotjelző csík táplálásához azonban még szükség van egy csővezetékre.

Miért ne?

Eddig csak az Xdialog mellett szóló érvekkel hozakodtam elő, ezért joggal megkérdezheted, hogy mikor fogok végre hibát keresni benne. Épp az előbb tettem meg. Ekkora méretű alkalmazásokat már nem gazdaságos bash-ben megoldani. Véletlenül se próbálj meg például CD-katalógust készíteni a fenti programokkal, mert napokig is eltarthat, mire elkészíti a fát. Az Xdialog telepítő parancsfájlnak tökéletes megoldás. Ha fontos a csillogás, készíthetsz hajlékonylemez-formázó programot és szép grafikus kabátot húzhatsz rá. Látni kell viszont a héjprogramozás korlátait: az előbbi feladat megoldása lassú és nem is szép. Egy szint elérése után nem kerülhető el, hogy megtanuld a grafikus felhasználói felületek készítésének a módjait. Oldd meg a feladatot C++-ban, Javában, vagy – ha közel állnak hozzád a parancsnyelvek – Tcl/Tk-ban.

Remélem, sikerült bemutatnom az Xdialog nyújtotta lehetőségeket. Ha te magad, a barátod vagy a felettesed szereti a grafikus felületet, és egy egyszerű, parancsfájllal önműködővé tehető feladat áll előtted, az Xdialog a te barátod.



Fülöp Balázs (admin@guardware.com)

18 éves, imádja a Túró Rudit, a Debian Linuxot és a teheneket. Kedvenc írója Slawomir Mrodek. Leginkább a számítógépes hálózatok biztonsága érdekli. A BME VIK műszaki informatikus szak hallgatója.