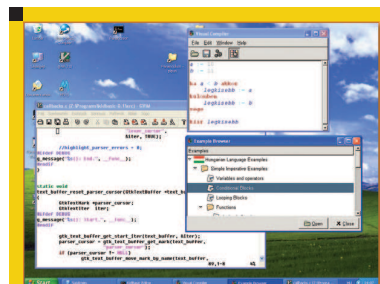


Alkalmazásfejlesztés a GTK+ programkönyvtárral (1. rész)

Napjainkban a legtöbb felhasználó olyan alkalmazásokat keres, amik grafikus felhasználói felülettel (GUI) is fel vannak szerelve. Éppen ezért a programozással hobbiból vagy hivatásszerűen foglalkozók körében igen keresettek a grafikus eszköztárak. Cikksorozatunkban a GTK+ programkönyvtár segítségével végzett alkalmazásfejlesztés fontosabb elemeit mutatjuk be azon olvasóink számára, akik a programozás alapjaival tisztában vannak és szeretnének modern, felhasználóbarát grafikus alkalmazásokat készíteni.

A GTK+ (*Gimp Toolkit*) programkönyvtár – ahogyan neve is mutatja – grafikus eszköztárat biztosít a programozó számára *GUI* alkalmazások készítésére. Persze több programkönyvtár is létezik hasonló szolgáltatásokkal, amelyek közül érdemes megfontoltan választanunk. A programkönyvtárak napjainkban olyan sokrétűek, nagyok és bonyolultak, hogy az alapos megismerésükhöz évek kemény munkájára van szükség, ezért nem lehet túlbecsülni a választás jelentőségét. Mik tehát a *GTK+* programkönyvtár előnyei? A programkönyvtár teljes egészében C programozási nyelven íródott, ezért hatékony, gyors és viszonylag kis memóriaigényű a többi, hasonló tudású eszköztárral összevetve. Ennek köszönhető, hogy a segítségével készített alkalmazások gördülékenyen futnak, nem akadoznak és nem idegesítik fel feleslegesen a felhasználókat. Abban, hogy a felhasználók jelentős része gyorsabbnak tartja a *Linuxot* a *Windows* különféle változatainál, valószínűleg komoly szerepe van, hogy a *Windows* grafikus eszköztára sokszor hosszú másodpercekre érzéketlenné válik a felhasználó ingereire, különösen ha sok adatot megjelenítő lista, ikonmező van a képernyőn. A *GTK+ C nyelv*en íródott, ennek ellenére támogatja a *C++*, *Objective-C*,

Guile/Scheme, *Perl*, *Python*, *TOM*, *Ada95*, *Free Pascal*, és *Eiffel* nyelvű alkalmazásfejlesztést is, ráadásul objektumorientált programozási módszertan (*OOP*) alapján készült, ami a *GUI* készítésére különösen alkalmas. Az olvasók közül bizonyára sokan felkapják a fejüket e kijelentésre, hiszen nem szokás a C nyelvet az *OOP* módszertannal egy lapon emlegetni. Valóban, a *C nyelv* nem támogatja az *OOP* módszereket, viszont nem is teszi lehetetlenné. A *GTK+* alapjait biztosító *G* programkönyvtár teljes értékű keretrendszert ad az osztályok, az öröklődés és a többalakúság megvalósítására, a *GTK+* pedig ki is használja ezeket a lehetőségeket, így az objektumorientáltság ez esetben nem üres frázis, hanem következetesen végigvitt tervezési alapelv. A *GTK+* másik nagy erőssége az, hogy többféle operációs rendszeren is futtatható. Ennek köszönhető az, hogy a *GTK+* és *G* programkönyvtárakkal készített alkalmazások nem csak *Linuxot* futtató számítógépeken, de például a *Windows* különféle változatain, a *Solaris* rendszereken és *MacOS* felett is használhatók. Aki tehát *GTK+* segítségével fejleszt *Windows* programot is tud írni (1. ábra). Ne becsljük alá a grafikus alkalmazások hordozhatóságát! Sokan próbáltak már olyan programfejlesztő környeze-



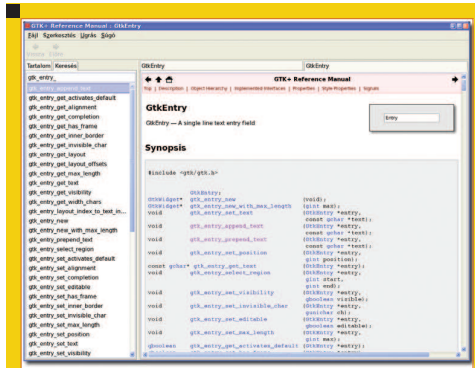
1. ábra GTK+ segítségével készített alkalmazások Microsoft Windows XP felett

tet létrehozni, ami *GUI* programok fejlesztésére alkalmas és több operációs rendszert is támogat, használható megoldást azonban annál kevesebbek sikerült készíteniük. A *GTK+* programkönyvtár legnagyobb előnye azonban az egyszerű és logikus felépítése, ami lehetővé teszi, hogy viszonylag rövid idő alatt elsajátítsuk a használatát. A programkönyvtárat profi fejlesztők írják és szabad szoftverként sokan tökéletesítik, akik következetesen betartják a fejlesztés irányelveit, ezért nem nehéz megtanulni a használatát. Nyilván sokan megütköznek e kijelentés hallatán. Egyszerűen megtanulható programkönyvtár *GUI* programok fejlesztésére? Több ezer függvénnyel? C nyelven megírt objektumhierarchiával?

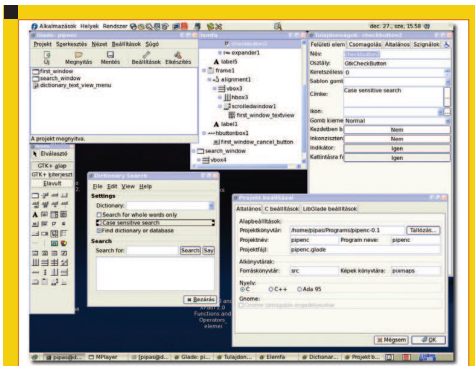
Osztályok százaival? Az egyszerűség nyilván viszonylagos, a **GTK+** a programozó számára könnyebben kiismerhető, mint a hasonló tudású eszköztárak, ezért nyugodtan mondhatjuk, hogy egyszerű. A **GTK+** ráadásul szabad szoftver, ami lehetővé teszi, hogy elolvassuk, sőt akár módosítsuk is a forrását. Ha egy átlagos alkalmazás esetében ennek nem is tulajdonítunk nagy a jelentőséget – mondván, hogy az átlagos alkalmazást átlagos felhasználók használják – a programkönyvtárak esetében azonban különösen fontos a nyílt forrás. A programkönyvtárat nyilván programozók használják, akik szívesen vetnek néha egy-egy pillantást a forrásra, hiszen a legjobb dokumentáció maga a program.

Az első lépések

A programozáshoz a programkönyvtárakon és a fejláományokon kívül okvetlenül szükségünk lesz a programkönyvtárak dokumentációjára, amelyet legkönnyebben a `devhelp` program segítségével böngészhetünk. Maga a `devhelp` elég primitív program, már néhány perc használat után is komoly késztetést érzünk arra, hogy megnyissuk a forrását és egy kicsit továbbfejlesszük, gyorskereső szolgáltatása azonban gyakorlatilag nélkülözhetetlenné teszi. A program ablakának bal oldalán található beviteli mezőbe (2. ábra) a keresett függvény vagy adatszerkezet nevének részletét is beírhatjuk, a találatok listájából pedig kiválaszthatjuk, hogy melyik találat részletes dokumentációjára vagyunk kíváncsiak. Ezt a lehetőséget a kezdő programozó igen intenzíven használja az első néhány hónapban, de a későbbiekben is hasznos lehet hiszen sokszor nem emlékezünk a szükséges függvény teljes és pontos nevére. A munka során legtöbbször használt eszköze azonban a **Glade** lesz, a **GTK+** svájci biciskája, aminek használatát mindenképpen meg kell ismernünk. E programmal az alkalmazásunk grafikus felhasználói felületét grafikus felhasználói felületen szerkeszthetjük, ami meggyorsítja és megkönnyíti a **GUI** létrehozását és a későbbi módosítást is lehetővé teszi. A különféle képernyőelemek a **GTK+** megfelelő



2. ábra A devhelp gyorskereső szolgáltatása



3. ábra A glade ablakai

függvényeinek hívásával előállíthatók és a képernyőn megjeleníthetők, az ilyen program írás és karbantartása azonban igen unalmas és lassú, érdemes a **Gladedel** automatizálni. A **Glade** programból jelenleg két – egymástól jelentősen különböző – változat is elérhető. A régebbi a `glade-2` paranccsal indítható (3. ábra). Ez a változat képes elmenteni a szerkesztett felhasználói felületet **XML** formátumban és képes az `autoconf` és az `automake` programokkal támogatott projektet létrehozni, fel van szerelve egy **C** nyelvű kódgenerátorral. Az újabb – `glade-3` paranccsal indítható – változatban nincsen kódgenerátor, csak az **XML** állomány létrehozására használható. Az első lépések megtételéhez a `glade-2` alkalmasabb, mert igen könnyen készíthetünk teljes értékű alkalmazást a segítségével, nem kell a projekt beállítóállományával és az **XML** állománnyal bábélni. Valójában a `glade-2` olyan egyszerű, hogy néhány sorban leírható hogyan készíthetünk alkalmazást a segítségével.

A `glade-2` indítása után a menü segítségével hozzunk létre és állítsunk be egy új projektet. Adjuk meg a projekt nevét és a projektkönyvtárat, ahová az állományok kerülni fognak. A többi beállítással az első kísérletek során nem is kell foglalkozni.

Második lépésként a **paletta** ablakban válasszuk a legelső ikont, hogy egy új ablakot hozzunk létre. Az ablakba – szintén a paletta segítségével – ízlés szerint helyezzünk el vízszintes és függőleges dobozokat, azokba pedig szintén tetszés szerint válasszunk a paletta különféle képernyőelemei közül. Amikor úgy gondoljuk, hogy itt az ideje a programunk kipróbálásának, a **Glade** főablakának segítségével mentjük a projektet, majd az **elkészítés** gomb segítségével hozzuk létre a projekt automatikusan készített állományait. A következő lépésként a **Glade** által létrehozott és a projekt főkönyvtárban elhelyezett `autogen.sh` héjprogramot kell futtatnunk. Ez a program az `automake` és `autoconf` programcsomagok segítségével létrehoz néhány állományt, majd szíveségből futtatja az ugyanitt elhelyezett `configure` programot is.

Ezek után az alkalmazásunkat a szokásos módon, a `make` futtatásával lefordíthatjuk, a `make install` paranccsal pedig telepíthetjük. A forrásprogram terjesztésre alkalmas csomagolt változatát szintén a megszokott módon, a `make dist` parancs kiadásával hozhatjuk létre.

A **Glade** a projekt alapkönyvtárban létrehoz egy `.glade` névvezető XML formátumú állományt a felhasználói felület leírásával. Ezt az állományt a későbbiekben megnyithatjuk a **Glade** segítségével, módosíthatjuk, a forrásprogramot pedig a **létrehoz** gombbal frissíthetjük. A **Glade** az automatikusan módosított függvényeket és a programozó által készített programsorokat külön állományokban tárolja, így egy kis figyelemmel elkerülhetjük, hogy a munkánk kárba vesszen.

Pere László
(pipas@gamma.ttk.pte.hu)