

Vala

Mehmet Eribol

Aralık, 2011

Uzun süredir yazmayı düşündüğüm bir Vala Programlama Dili yazısı var. Belki dört, belki beş defa giriş yazısı yazdım; ama sanırım kuruntularımın dolayısıyla sildim. Son yazdığım yazı ise temelli bir göç sebebiyle bilgisayarın, tarafımdan şamar oğlanına dönmesi sonucu arada kaybolup gitti. :) Bu nedenle beni bu denli zorlayan bu konuyu doğaçlama yazacağım. Teknik terimler konusunda bir yanlışlık olduğunu düşünürseniz lütfen forum başlığında belirtiniz.

Vala dilinin geçmişi çok uzun sayılmaz. Bu programın ortaya çıkış sebeplerinin başında GNU/Linux sistemlerinde GTK ile program yazılmasının C ile yazılması veya yorumlanabilir diller ile yazılması zorunlu olmasındandır. C, güçlü alt yapısından dolayı, görsel araçlarda kullanılması pek zor olan bir dildir. Zira çok zaman alır ve çok fazla kod satırı gerektirir. Oysa aynı işi 3. nesil dillerle yazmak çok daha kolaydır. Ancak bu 3. nesil diller de C ile karşılaştırıldığında bellek ve işlemci konusunda çok cömert davranırlar. Diğer bir sebep ise GTK ile hem güçlü hem de hızlı bir şekilde görsel program geliştiren en iyi aracın Mono olmasıdır. Mono hem sisteme yüklediği ek paketlerden dolayı ve hem de Microsoft'un C# kökeninden türemesinden dolayı özgür yazılımcılar tarafından pek benimsenmemiştir.

Bu sebeplerden dolayı özgür yazılımcılar ve özellikle Gnome ve GTK ile program geliştirenler için C# kadar esnek bir dil gereksinimi doğmuştur. Elbette ki bu dil sadece bir alternatif olarak değil, bir gereklilik olarak sunulduğunda farkını ortaya koyacaktır. Bu farklılığı da Vala geliştiricileri, sisteme ek yük bindirmeme yolunu seçerek ortaya koymuşlardır. Vala ile yazılan bir program, sistemde C ile yazılan bir GTK programından fazla bir yük bindirmez. Vala ile yazılan bir programı kurmak için hiçbir zaman Vala paketlerini kurmak zorunda kalmazsınız. Vala bir tek paket derlemek için gereklidir ki bu da son kullanıcı için gerekli değildir. Hatta Vala ile program geliştiren yazılımcılar kaynak kodları C şeklinde sundukları için derlemek için dahi Vala'ya ihtiyaç duyulmaz.

Belirttiğim gibi Vala ile yazılan bir program, Vala derleyicisi ile aslında C kodlarına çevrilir. Yani esasında yazılan program yeni bir dil olan Vala ile değil, C ile yazılıp derlenir. Bu durum programcılara hem zamandan hem de performanstan kazanç sağlar. Bu avantajından dolayıdır ki son zamanlarda geliştirilen neredeyse tüm GTK/Gnome araçları Vala ile yazılır.

Kodlama kısmına ufak bir giriş yapmadan önce Vala ile yazılan gözde programları yazmak güzel bir izlenim bırakacaktır diye düşünüyorum.

Unity: Ubuntu'nun ses getiren Gnome çatısı Vala ile yazılmıştır. Son zamanlarda Ubuntu geliştiricileri Python yerine daha çok Vala kullanmaya yönelmişlerdir. Elbette ki Python ile yine bir çok araç geliştiriliyor; ancak gün geçtikçe yeni geliştirilen araçlar Vala ile yazılmaya başlanıyor. Tabii burada programın amacı da etkili oluyordur. Yine de performans gerektiren bir uygulama için Vala çok daha uygun bir dil olarak öne çıkıyor.

- **Globalmenu:** Yine Unity ile paralel zamanlarda geliştirilen bu araç da Vala ile yazılmıştır.
- **Shotwell:** Vazgeçilmez fotoğraf düzenleyicisi.
- **Peynir:** Her sistemde olması gereken kamera etkileşimli program.
- **Deja Dup:** Mükemmel bir yedekleme yazılımı.
- **ElementaryOs** ve **Paldo Linux** geliştiricileri. Bu dağıtımın geliştiricileri de özel araçları için Vala dilini kullanıyor.

Elbette oldukça uzun bir listemiz var; ancak bu yazılımlar bile Vala dilinin gücünü gösterir nitelikte.

Şimdi ilk programımızı yazabiliriz.

```
1 int main () {
2     stdout.printf("Merhaba Dünya");
3     return 0;
4 }
```

Gördüğünüz gibi birkaç satır kod ile ekrana "Merhaba Dünya" yazdırdık. Şimdi bu kodları "a.vala" (Vala uzantılı olmalı.) olarak bir dizine kaydedin ve terminalden o dizine gidip

```
1 valac a.vala
```

komutunu verin. Bu komutla birlikte aynı dizinde "a" adındaki dosyayı "./a" komutu ile çalıştırın. Terminalde yazıyı göreceksiniz. Unutmadan "valac" komutu için depodan Vala paketini kurmalısınız. :) Bu komutlar elbette ki C veya C++

kullanıcıları için yabancı değil; ancak diğerleri için açıklamak gerekirse "int main()" ile ana fonksiyonumuzu tanımlıyoruz. Bu, programın bu kısımdan başlayacağına işaret eder. "stdout.printf" komutu ise ekrana yazdırma komutudur ve "return 0" komutu ise programdan çıkılmasını sağlar. "return" komutunu bu konu içinde detaylıca açıklamaya gerek olmadığını düşünüyorum.

Yine bu komutların C kodlarına nasıl dönüştürüldüğünü görmek için "-C" parametresiyle derleyin (Aslında bu parametre derleme yapmaz, sadece C koduna çevirir.). Eğer kodlarınızda kütüphane var ise ve bunun C kütüphanesine dönüştürülmesini istiyorsanız "-H" parametresini kullanabilirsiniz. C diline çevrilen kodlar gözünüzü korkutmasın. Vala dönüşüm yaptırdığı için çok karmaşık ve çok fazla komut oluşturur. Ancak ne kadar basit uygulama olursa olsun "glib" kütüphanesini kullanır, zira gücünü bu kütüphaneden alır.

Basit bir GTK uygulamasını Vala ile yazıp derleyelim.

```
1 using Gtk;
2
3 int main (string[] args) {
4     Gtk.init (ref args);
5     var window = new Window ();
6     window.title = "İlk Gtk+ programımız";
7     window.border_width = 10;
8     window.window_position = WindowPosition.CENTER;
9     window.set_default_size (350, 70);
10    window.destroy.connect (Gtk.main_quit);
11    var button = new Button.with_label ("Tıkla!");
12    button.clicked.connect (() => {
13        button.label = "Teşekkürler";
14    });
15    window.add (button);
16    window.show_all ();
17
18    Gtk.main ();
19    return 0;
20 }
```

Bu kodları derlemek için;

```
1 valac a.vala --pkg gtk+-3.0
```

komutu kullanılır. Elbette ki bu GTK3 içindir. Bu programda gtk2 veya gtk3 için bir fark yoktur. Derlemeden önce bilmeniz gereken ise GTK ile program geliştirmek için "gtk-devel" paket(ler)ini kurmanızdır. Aynı şekilde aynı dizin içerisindeki "a" dosyasını "./a" komutu ile çalıştırabilirsiniz. Dikkat ettiyseniz eğer "using Gtk" satırını ekledik. Bu satır ile sisteme GTK kütüphanesinin kullanılacağını belirtmiş oluyoruz. Eğer kendi yazdığınız bir Vala veya C kütüphanesini kullanacaksanız aynı şekilde "--pkg" parametresinden sonra ismini yazarak kullanabilirsiniz.

"Gtk.init" satırı her GTK uygulaması için gereklidir. Diğer satırları basitçe açıklamak gerekirse; kullanacağınız her GTK nesnesini bir değişken olarak tanımlamalısınız. "var window = new Window();" satırı ile "window" adında bir "Window" nesnesi tanımlamış olduk. Tanımlamış olduğumuz bir "Window" nesnesinin özelliklerini bu "window" değişkenini kullanarak belirteceğiz.

```
1 window.title: Pencere başlığını ,
2 window.border_width: Pencere kenarlarının genişliğini ,
3 .window_position: Ekrandaki konumunu,
4 .set_default_size: Varsayılan ebatlarını ,
5 .destroy.connect: Kapatma düğmesine tıklandığında işletilecek fonksiyonu değiştirebilmemize/düzenlememize
   mümkn verir .
6
7 var button = new Button.with_label ("Tıkla!"); bu satırda ise bir buton tanımladık ve tanımladıktan sonra
   bir özelliğini de değiştirdik." with" eki ilk tanımlamada kullanılan bir komut. Değişkeni bir defa
   tanımladıktan sonra farklı bir yerde herhangi bir özelliğini değiştirmek için ".with" komutuna gerek
   yoktur.
8
9 button.clicked.connect (() => {
```

```
10     button.label = "Teşekkürler";
11 });
```

Buradaki komutlar ise tanımladığımız button nesnesinin "clicked" olayından "connect" metodunu çağırıyor. Bu olaylar ile tanımlanmış bir fonksiyon çağırabileceği gibi örnekte gösterilen şekilde aynı yerde de bir fonksiyon oluşturulup çalıştırılabilir.

"window.add(button);" bu komut GTK için gerekli bir komuttur. "Window" nesnesi GTK'de kullanılan neredeyse tüm nesnelerin atası gibidir. Bu nedenle tüm nesnelere bu komut pencereye yerleştirmeliyiz. Ve tabii ki "window.show_all();" komutu ile de tanımladığımız pencereyi de ekranda gösteriyoruz.

Basit olarak yaptığımız bu programı derleyip çalıştırdığımızda ekranda bir düğme(buton) barındıran bir pencere olacak. Buton üzerine tıkladığımızda "Tıkla" yerine "Teşekkürler" yazacak.

Şimdilik Vala yazısını burada sonlandıralım. Sonlandırırken ufaktan Genie'den de bahsedelim :)

Genie:

Vala esasında bir yorumlayıcı. Vala uzantısı ile yazılan kodları C kodlarına çeviriyor. Vala şekil olarak C# diline yakındır ve aslında amacı da GTK için C#(mono) kadar gelişmiş bir dil ortaya çıkarmaktır. Ancak yazım dilini her ne kadar diğer nesne yönelimli dillere benzetse de asıl yaptığı iş, bu kodları C diline çevirmektir. Eğer bu kodları C diline çevirebiliyorsa yazımı çok daha basit olan pythonvari komutları neden C diline çeviremesin? Geliştiriciler de kendilerine bu soruyu sormuş olmalıdır ki pythonvari bir dil olan Genie'yi de geliştirmişler. Fazladan hiçbir gereksinimi yoktur. Sadece kodları derlemek için Vala ihtiyacı vardır. Vala'dan farklı olarak bu dosyayı "gs" uzantısı ile kaydediyoruz. Bu uzantı sayesinde Vala derleyicisi bu kodların bir Genie kodu olduğunu algılar ve ona göre derleme yapar. Henüz pek fazla belge yok; ancak elimizde python gibi yazım dili basit, C kadar performanslı bir dil var. Makine diline çevrildiği ve C kütüphanelerinden başka da bir gereksinimi olmadığı için bu dil de sisteme ek yük bindirmez.

Ufak bir program yazalım;

```
1  init
2  print "merhaba"
```

Bu kodları a.gs olarak kaydedip aynı dizinde "valac a.gs" komutunu verip "./a" komutu ile de çalıştırabilirsiniz. Gördüğünüz gibi çok basit, tıpkı Python gibi. "init" C dilindeki main fonksiyonu görevini yürütür ve varsayılan olarak fonksiyon içerisindeki kodlar "init" komutunun başlangıç noktasından bir tab aralıkla başlar(8 karakter boşluk). Ancak eğer programın en başında "[indent=2]" gibi bir komut eklenmişse satır boşluğu olarak iki karakter boşluk bırakılır. Bu açıklamayı Python kullanıcıları daha iyi anlayacaktır.

Not: Bu yazıdan Python dilinin iyi olmadığı anlaşılmalıdır. Python yorumlanabilir bir dil olduğundan birçok yerde derlenebilir dillerden çok daha esnek ve çok daha gereklidir. Zira yorumlanabilir bir dil derlenme zorunluluğu olmadığından yapılan değişikliği anında yansıtır. Oysa derlenebilir diller ise zorunlu olarak tekrar derlenmelidir. Bu nedenle birçok uygulama, eklentiler için Python ve Ruby gibi dillerden büyük destek alır.