Mikroişlemciler, III. Uygulama Yönergesi

Amaç: Assembly dili programları, Tümleşik Geliştirme Ortamı (Integrated **D**evelopment Environment, IDE) kullanarak geliştirme.

Hedef: Komut satırından 1. arguman olarak aldığı dosyayı açarak içeriğini 2. arguman olarak verilen dosyaya saklayan bir programın Visual Studio (VS) ile derlenmesi.

Bu uygulamada, MicroSoft'un komut satırından çalıştırılabilen derleyicilerine grafiksel kullanıcı arayüzü ile entegre bir geliştirme ortamı sağlayan VS ile program geliştirme aşamaları açıklanacaktır. VS'nun güncel sürümünü (veya Imagine aboneliğiniz yoksa ücretsiz olan Community 2017 sürümünü) kurmak için gerekli kurulum dosyasını bilgisayarınıza indirerek uygun seçenekleri ile kurunuz.

Uyarı : VS yüksek seviyeli diller için programlamayı kolaylaştıran bir geliştirme ortamı olmasına karşı, yalnız Assembly dili kaynak dosyalarını içeren bir çözüm (Solution) paketi oluşturmaya olanak vermemektedir. Bu nedenle oluşturulacak çözüm içine boş bir VC++ dosya koyarak ve bazı varsayılan proje seçeneklerini değiştirerek VS'nin Assembly dili dosyaları çözüm içinde kabul etmesi sağlanmalıdır.

VS'da yeni bir Assembly program geliştirmek için **Dosya -> Yeni -> Proje** seçildikten sonra ekranda görülen listeden **Yüklü -> Şablonlar -> Visual C++ -> Genel** altında yer alan **Boş Proje** seçeneği ile yeni bir Visual C++ dosyası çözüm paketine eklenmelidir. Dosyayı yaratırken kullanılan varsayılan dosya adı (Proje1) yerine, daha sonra çözümü açtığınızda projenizi kolay hatırlamanızı sağlayacak şekilde yaptığınız işe uygun bir ad vermeniz gerekir. Bu yönergede bir örnek proje oluşturulduğu için, boş C++ dosya adı Ornek_C olarak seçilmiştir.

VS'nun yeni oluşturduğunuz boş C++ dosyasına ekleyeceğiniz Assembly dilindeki kaynak dosyalarını Macro Assembler (masm) ile derlemesi gerektiğini belirtmek için, çözüm gezgini penceresinde oluşturduğunuz dosya üzerinde sağ tıklayarak **Derleme Bağımlılıkları** -> **Özelleştirme Oluştur** seçtiğinizde çıkan listedeki seçili olmayan **masm**'yi seçerek Tamam butonu ile onaylamanız gerekmektedir. Bu aşamadan sonra projenize ekleyeceğiniz asm uzantılı dosyalar, VS tarafından masm (seçtiğiniz hedefe bağlı olarak ml64.exe veya ml.exe) ile derlenecektir.

Projenize Assembly dilinde kaynak kodlarına ait dosyalar eklemek için, boş C++ proje dosyası üzerinde sağ tıklayarak **Ekle** -> **Yeni Öğe** seçeneği ile karşınıza çıkan listedeki VC++ **Yardımcı Dosya** altındaki **Metin Dosyası (*.txt)** türünü seçtikten sonra Assembly dilindeki dosyanızı asm uzantılı olarak adlandırarak projeye eklemeniz gerekmektedir. Bu yönergede örnek proje Assembly dosyası için Ornek_A.asm adı kullanılmıştır.

Bu yönergede örnek olarak projeye eklenecek Assembly dilindeki program, 2016 Bahar yarıyılında ödev olarak verilen Berger kodlama projesi için dağıtılan kaynak kodlardan türetilecektir. Ders sayfasında 24. sıradaki kaynakta verilen masm (arg_masm.asm dosyası) programı, komut satırından verilen giris ve çıkış dosya adlarını arguman olarak aldıktan sonra giriş dosyasını okuyarak çıkış dosyasına yazmaktadır. Verilen kaynak kodlar, öğrencilerin komut satırından arguman okuma ve dosya giriş/çıkış yapma gibi Berger kodlama dışındaki sorunlarla uğraşmaması için şablon olarak yayınlanmıştır. Bu kodun, VS'nun asm uzantılı dosyaları masm ile derlemesi için oluşturduğumuz boş C++ dosyasında bulunması beklenen main() ana programının yerini alabilmesi için **start** adlı kod adı **main** ile değiştirilmeli ve Bağlama seçenekleri içinde C++ dosyasının içermediği **main** adındaki giriş noktasının kullanılması gerektiği belirtilmelidir.

Sıralama ödevinin gelişitirilmesinde yaralanabileceğiniz şablon, not defteri ile açabileceğiniz gerekli değişiklikleri yapılmış arg_vs.asm <u>dosyası</u>nın tamamının kopyalanarak açık olan Ornek_A.asm içine yapıştırılması ile oluşturulabilir. Belirtilen Bağlama zamanı değişikliklerini yapmak için proje dosyası üzerinde sağ tıklanarak özelliklerine girildikten sonra, çıkan **Yapılandırma Özellikleri**'nin en altından ikinci satırında **MicroSoft Makro Assembler** seçeneğinin görülmesi gerekmektedir. Bu seçeneğin görülmemesi durumunda anlatılan ayarlardan bazılarının eksik ve/vaya yapılmış olması muhtemeldir. **Yapılandırma Özellikleri** -> **Bağlayıcı** -> **Sistem** içindeki boş olan **Alt Sistem**'e menüden **Konsol, (/SUBSYSTEM:CONSOLE)** seçildikten sonra uygulanmalıdır. **Yapılandırma Özellikleri** -> **Bağlayıcı** -> **Gelişmiş** altında henüz tanımlanmamış olan **Giriş Noktası**'na klavyeden **main** girildikten sonra uygulanmalı ve yine **Yapılandırma Özellikleri** -> **Bağlayıcı** -> **Gelişmiş** altındaki **Görüntü Güvenli Özel Durum İşleyicilerine** menü seçeneklerinden **HAYIR(/SAFESEHNO)** seçilerek onaylanmalıdır.

Açıklanan tüm değişiklikler yapıldıktan sonra proje derlendiğinde programda hata çıkmazsa projenin çalıştırılabilir kodu, kullanıcının Belgeler dizini içinde VS'nun oluşturduğu Projects dizini içinde oluşturduğunuz proje adlı dizin altındaki seçtiğiniz hedef mimariye bağlı olarak x86 veya x64 ve Debug veya Release dizinleri içinde oluşturulacaktır. Programınızı test etmek için, not defteri ile hazırlayacağınız ve karakterler içeren bir giriş dosyasının 1. arguman olarak verilmesi durumunda, 2. argumanda belirttiğiniz çıkış dosyasına giriş dosyasındaki tüm karakterlerin aktarılmış olması gerekir.

Kaynaklar :

https://www.codeproject.com/Articles/271627/Assembly-Programming-with-Visual-Studio http://www.deconflations.com/2011/masm-assembly-in-visual-studio-2010/ http://asmhighlighter.codeplex.com/