



Fenerbahçe Üniversitesi
BLM 205 – Nesneye Yönelimli Programlama
FBCPU Compiler
Proje İçeriği

Veriliş Tarihi: 13.12.2020

Teslim Tarihi ve Yeri: 15.01.2021, Ders Saatlerinde, Elektronik olarak

1. Tanım:

Fenerbahçe İşlemcisi (FBCPU) isminde bir işlemcinin çalışması için ihtiyaç duyduğu kendine özel assembly dili gereksinimlerini karşılamak için daha yüksek seviyeli bir dilden (C gibi), FBCPU assembly'sine dönüşüm yapan bir araç gerçekleştirilecektir.

2. Proje Ekibi:

Proje 4 kişiye kadar olan ekiplerden oluşacaktır. Her bir proje ekibinin bir sorumlusu olacaktır. Öğrenciler 1, 2, 3 veya 4 kişilik kendi proje ekiplerini ve proje sorumlusunu belirlemelidirler.

Ekiplerin kurulması ve proje sorumlusunun belirlenmesi en geç **18.12.2020** tarihine kadar tamamlanmalıdır. Ekip sorumluları, Teams'te açılmış olan "Proje Ekip Sorumluların Takımlarını Bildirmesi" başlığının altına, ekip üyelerinin isimlerini göndermelidirler.

3. Kullanılacak Araçlar:

Proje kapsamında, ödev ve LAB'larda kullanılan Microsoft'un derleyicisi olan Visual Studio Community kullanılacaktır.

4. Tasarım Gereksinimleri

FB-CPU'nun desteklediği işlemler Tablo 1'de verilmektedir.

Tablo 1. FB-CPU ISA (Instruction Set Architecture)

Komut Adı	Görevi
LOD ADDR	Yükleme (Load), Bellekteki verilen adresin içerisinden değeri alıp, ACC saklayıcısına yerleştirir. $ACC = *(ADDR)$
STO ADDR	Kaydetme (Store), ACC'nin içerisindeki değeri alıp, bellekte verilen adrese yazar. $*(ADDR) = ACC$
ADD ADDR	Bellekteki verilen adresteki değeri alır, ACC ile toplayıp, ACC'nin üzerine yazar. $ACC = ACC + *(ADDR)$

SUB ADDR	Bellekteki verilen adresteki değeri alır, ACC ile çıkartıp, ACC'nin üzerine yazar. ACC = ACC - *(ADDR)
MUL ADDR	Bellekteki verilen adresteki değeri alır, ACC ile çarpıp, ACC'nin üzerine yazar. ACC = ACC * *(ADDR)
DIV ADDR	Bellekteki verilen adresteki değeri alır, ACC ile bölüp, ACC'nin üzerine yazar. ACC = ACC / *(ADDR)
JMP SAYI	PC = Sayı olur.
JMZ SAYI	ACC'ın değeri 0 ise, verilen sayı değerini PC'e atar, değilse işlem yapmaz.
NOP	No Operation, hiçbir işlem yapılmaz.
HLT	Uygulama durur

İşlemci 10 adet komutu desteklemektedir.

Verilecek yüksek seviyeli C'e benzer bir dildeki ifadeleri FBCPU Assembly Set'ine dönüşüm yapılmalıdır. Yüksek seviyeli dilde desteklenecek özellikler;

- Başlangıç Sabit Değer (A = 5)
- Atama işlemi (A=B)
- Toplama (A = B + C)
- Çıkarma (A = B – C)
- Çarpma (A = B * C)
- Bölme (A = B / C)
- Atlama:

ETIKET:

...

...

goto ETIKET;

- Eşit Değil Kontrolü (A = B == C)
- Eşit Değilse Kontrolü (A = B != C)

Aritmetik işlemler en fazla 2 elemanlı olacaktır. Yani tek seferde A + B + C işlemi olmayacaktır.

5. Gerçekleme Yöntemi:

FBCPU'nun desteklediği komutlar ile yapılabilecek örnek problem setleri aşağıdaki linklerde verilmektedir.

[http://www.levent.tc/files/courses/introductions to computer engineering/labs/lab5/BLM101 lab5 von neum ann mimarisi kontrol komutlari.pdf](http://www.levent.tc/files/courses/introductions%20to%20computer%20engineering/labs/lab5/BLM101_lab5_von_neum_ann_mimarisi_kontrol_komutlari.pdf)

[http://www.levent.tc/files/courses/introductions to computer engineering/labs/lab5/BLM101 lab5 von neum ann mimarisi kontrol komutlari.pdf](http://www.levent.tc/files/courses/introductions%20to%20computer%20engineering/labs/lab5/BLM101_lab5_von_neum_ann_mimarisi_kontrol_komutlari.pdf)

[http://www.levent.tc/files/courses/introductions to computer engineering/labs/lab6/BLM101 lab6 von neum ann mimarisi fonksiyonlar.pdf](http://www.levent.tc/files/courses/introductions%20to%20computer%20engineering/labs/lab6/BLM101_lab6_von_neum_ann_mimarisi_fonksiyonlar.pdf)

Burada bulunan algoritma sorularından en az 5 tanesinin C kodunu yazınız. Bu kodlar compiler'ı test ederken referans alınacaktır.

FBCPU Compiler C'nin tüm özelliklerini desteklemeyecektir. Geliştirilen compiler aracına verilecek örnek giriş ve çıkış Tablo 2'de verilmektedir. Bu tablodaki verilen syntax gibi yazdığınız C kodlarını düzenleyip, compiler aracınıza besleyebilirsiniz.

Tablo 1. FB-CPU ISA (Instruction Set Architecture)

A = 5;	0: LOD 40
B = 10;	1: MUL 41
C = B * A;	2: STO 42
	40: 5
	41: A
	42: 0

Yapılan compiler aracının 2 argüman girişi olacaktır. İlk argüman giriş yüksek seviyeli dil dosyası adı ve ikinci argüman ise içinde assembly kodları bulundurulacak çıkış dosyasının ismidir.

Örnek:

donusturucu.py giris.txt cikis.txt

donusturucu.py giris.txt cikis.txt

Sabit sayılar için 40. adresten sonra yerleştirme yapabilirsiniz.

6. Notlandırma ve Proje Teslimi:

Bu başlık "FBCPU Assembler" projesinin teslimi ve notlandırılması hakkında bilgiler içermektedir.

6.1. Notlandırma:

Projenin iki ana değerlendirme kriteri vardır. Her iki kriter 50 şer puandır.

İlk kriter "FBCPU Assembler" 'nin **doğru çalıştırılması ve objeye dayalı programlama dili yaklaşımlarının kullanılmış olmasıdır.**

İkinci kriter ise **Proje Teslim Dokümanı ve Sunumdur.**

- **Proje Teslim Dokümanı:**

Öğrenciler, proje raporlarını verilen "Proje Teslim Dokümanı" 'nın içeriğini doldurarak yapacaklardır.

Proje Teslim Dokümanı:

http://levent.tc/files/courses/object_oriented_design/projects/BLM205_proje_teslim_dokumani.docx

Proje teslim dokümanı tamamlanıp PDF'e dönüştürüldükten sonra, PDF'in sonuna Latex (Genellikle bilimsel çalışmaların yazımında kullanılan, görsel olarak Word ile yazılmış dokümanlara göre çok daha başarılı olan bir doküman oluşturma dilidir) ile tasarlanmış özgeçmişler eklenecektir. Özgeçmişin minimumda kullanılan diller, yabancı diller, projeler (kısa açıklamaları ile), kullanılan kütüphaneler (varsa, örn. OpenCV), not ortalaması, github adresi, youtube adresi bilgileri verilmelidir.

Latex ile özgeçmiş oluşturma:

http://levent.tc/files/courses/object_oriented_design/projects/BLM205_latex_ozgecmis_olusturma.pdf

Latex ile özgeçmiş oluşturulduğunda PDF uzantılı dosyalar elde edilecektir. Proje teslim dokümanının PDF'inin arkasına eklemek için PDF birleştirici araçları (https://www.ilovepdf.com/merge_pdf gibi) kullanılmalıdır. İşlem sonucunda tek bir PDF dosyası elde edilecektir.

- **Proje Sunumu:**

Powerpoint üzerinde ortalama 5 dakika (4-6 dakika arası) sürecek bir sunum hazırlayarak kayıt etmelidirler. Kayıt işlemi, cep telefonu veya bilgisayar ekran kayıt yazılımları (Screen-Recorder, Bandicam vb...) ile yapılabilir.

Sunum, ekip üyeleri içinden biri tarafından, projenin nasıl yapıldığı, nasıl çalıştığı vb.. konularının powerpoint slaytları üzerinden anlatılırken kaydedilmesi ile olmalıdır. Sunum video'sunda powerpoint slaytları okunabilir ve konuşmacının sesinin anlaşılır olması gerekmektedir. Powerpoint slayt görünüm tasarımı istenildiği gibi yapılabilir.

Proje ekibinin tamamı, notlarını bu değerlendirmeye göre alırlar.

6.2. Teslim:

Projenin teslimi için aşağıdaki adımların gerçekleştirilmesi gerekmektedir. İstenen dosyaları sadece proje ekip sorumlusunun getirmesi, Teams ve Github (Çok yaygın bir açık kaynak kod paylaşım platformudur)'a yüklemelidir.

Teams'te açılmış olan "Proje Teslim" sayfasına aşağıdaki dosyaların yüklenmesi gerekmektedir.

- Proje kaynak kodları
- Hazırlanan powerpoint sunum dosyası (.ppt uzantılı dosya)
- Proje Teslim Dokümanı (Word formatında yüklenmelidir)
 - Dokümanın alt başlıkları doldurulmalıdır.
 - Kaydedilen powerpoint sunum video'su youtube'a yüklenip, adresi, dokümanın sonuçlar bölümündeki açılmış yere link'i yazılmalıdır (Video'nun herkes'e görünür olmamasını istiyorsanız, youtube'a yükledikten sonra liste dışı seçeneğini seçerek, sadece link'e sahip olan kişilerin görmesini sağlayabilirsiniz).
 - Teams'e yüklenen tüm dosyalar (Proje kaynak kodları, ppt uzantılı sunum dosyası ve Proje Teslim Dokümanını (PDF formatında)), github.com sitesine üye olup, yüklenip, dokümanın sonuçlar bölümündeki yere link'i yazılmalıdır.