

CSE 102 Bahar 2024 - Bilgisayar Programlama Ödevi 10

Son teslim tarihi 17 Mayıs 2024 saat 23:59

Bu ödev için bir assembly dili çalıştırma simülatörü oluşturacaksınız. Program assembly dili kod dosyasının çıktısını kullanıcıya gösterecektir. Bu görev için komut seti "*instructions.txt*" dosyasında verilmiştir, formatını değiştirmeyin. Komut setindeki her işlemin bir opcode'u, tipi, formatı ve açıklaması vardır ve komut seti toplam 30 işlemten oluşmaktadır. Buna ek olarak, her bir komutun açıklaması ödevin sonunda verilmiştir.

Komut işleme için struct işlemlerini kullanmanız ve verilen tüm işlemleri verimli bir şekilde işlemek için bir algoritma tasarlayarak koşullu ifadelerin kullanımını en aza indirmeniz gerekir.

Bölüm I. [20 puan] Simülasyon Yapısının Oluşturulması

Programınız, kullanıcıdan ".asm" dosya uzantısına sahip olması gereken montaj kodu dosyasının adını girmesini isteyerek başlamalıdır. Dosyayı işledikten sonra, program işlemin çıktısını kullanıcıya göstermelidir. Kullanıcı girdisinin doğruluğundan emin olun ve kullanıcı '0' yazarak çıkmayı seçene kadar programın çalışmasını sürdürün.

Montaj simülatörünün gereklilikleri aşağıda açıklanmıştır:

- A ve B olarak gösterilen toplam 2 yazmaç vardır. Her biri 8 bit saklayabilen char dizileri olarak uygulanmalıdırlar.
- Assembly kodunun satır numarasını içeren ve programın mevcut konumunu izleyen bir Program Sayacı (PC). Bu sayaç 16 bit saklayabilmelidir. Program çalıştıkça bir sonraki komutu işaret etmek için artırılmalıdır.
- Her bir komut 24 bit uzunluğunda olacak şekilde mevcut komutu tutmak için bir Komut Kaydı (IR) gereklidir. Her komut verildiği gibi ikili forma dönüştürülebilir:
 - İşlemin opcode'unu satırın başına yerleştirin.
 - Her bir sayıyı ikili forma dönüştürün.
 - Kayıtları bellek adresi ile temsil eder.

3 - A	10000101
-------	----------

Güncel Talimat: ADDI A 2

Talimat Kaydı: 00000010 10000101 00000010

(İpucu. Boşlukların göz ardı edilmesi gerekir, bunlar size kolaylık sağlamak için yerleştirilmiştir. Bu nedenle, IR içindeki gerçek değer 000000101000010100000010'dur)

- Bellek, her bir bellek konumu (kelime olarak adlandırılır) 8 bit uzunluğunda olacak şekilde bir yığın olarak yapılandırılmalıdır. Bellek adresleri satır numaralarına karşılık gelebilir (1, 2, 3, 4, vb.). Belleğin üst kısmı bir assembly programını saklamak ve okumak için ayrılırken, alt kısmı değişkenler ve diğer işlemler için tahsis edilebilir. Belleğin boyutu tüm bu bilgileri saklamak için yeterli olmalıdır.

Bölüm II. [10pts] Ondalık Sayıları İkiliye ve İkiliyi Ondalığa Dönüştürme

Ondalık sayıları ikili forma dönüştürmek için `int to_binary(int decimal_number)` adında bir fonksiyon yazın. Bu fonksiyon girdi olarak bir tam sayı almalı ve ikili gösterimini döndürmelidir. Tüm sayılar 8 bit olarak gösterilmelidir.

İkili sayıları ondalık forma dönüştürmek için `int to_decimal(char binary_number[])` adında bir fonksiyon yazın. Bu fonksiyon 8 bit uzunluğunda bir char sayı dizisini girdi olarak almalı ve ondalık gösterimini döndürmelidir.

Bölüm III. [70pts] Simülasyon Çalıştır

Program akış döngüsü açıklanmış ve aşağıda bir örnek verilmiştir:

1. **Komut getirme:** Her komut PC kullanılarak bellekten satır satır okunmalı ve yukarıda açıklanan formatta IR'ye yerleştirilmelidir.
2. **Kod çözme talimatı:** Komutun program tarafından anlaşılması gerekir ve bu döngü, opcode ve kullanılan kayıtların belirlenmesi gibi hangi işlemin gerçekleştirilmesi gerektiğinin belirlenmesini içerir.
3. **Talimat çalıştır:** Talimat tarafından belirtilen işlemlerin simülatör tarafından yapılması gerekir.
4. **Sonucu saklayın:** Sonucun belleğe kaydedilmesi gerekir.

Girdi Dosyası:

```
1  LDAI A 100
2  LDA A M[A]
3  LDAI B 101
4  LDA B M[B]
5  SHR A 2
6  SHL B 2
7  EKLE A B
8  PRI A
9  ÇIKIŞ
```

Terminal:

```
Enter the filename: example_code.asm
Result of the execution: 25
Do you want to execute another assembly code: Press (1) for yes, Press (0) for exit: 0
```

Başlangıçtaki ve Sondaki Hafıza:

Hafıza - Başlangıç	
1	00011101
2	01100110
3	01100100
4	00010101
5	01100110
6	00000101
7	00011101
8	01100111
9	01100101
10	00010101
11	01100111
12	00000110
13	00010000
14	01100110
15	00000010
16	00001111
17	01100111
18	00000010
19	00000001
20	01100110
21	01100111
22	00011010
23	01100110
24	00000000
25	00011101
26	00000000
27	00000000
...	...
100	00000101
101	00000110
102-A	00000000
103-B	00000000
104-PC	00000000
105-PC	00000001
106-IR	00010101
107-IR	01100110
108-IR	01100100
...	...

Hafıza - Son	
1	00011101
2	01100110
3	01100100
4	00010101
5	01100110
6	00000101
7	00011101
8	01100111
9	01100101
10	00010101
11	01100111
12	00000110
13	00010000
14	01100110
15	00000010
16	00001111
17	01100111
18	00000010
19	00000001
20	01100110
21	01100111
22	00011010
23	01100110
24	00000000
25	00011101
26	00000000
27	00000000
...	...
100	00000101
101	00000110
102-A	00011001
103-B	00011000
104-PC	00000000
105-PC	00011100
106-IR	00011101
107-IR	00000000
108-IR	00000000
...	...

Opcode	Tip	Biçim	Açıklama
00000001	G	EKLE A B	A ve B kayıt değerlerini toplar, ardından sonucu A'da saklar
00000010	T	ADDI A I	A yazmacını ve verilen sayıyı toplar, ardından sonucu A'da saklar.
00000011	U	ADDM A M[B]	A yazmacını ve B bellek adresinde saklanan bir sayıyı ekler ardından sonucu A'da saklar.
00000100	G	VE A B	A ve B kayıt değerlerini toplar ve sonucu A'da saklar.
00000101	T	VEI A I	A yazmacını ve verilen sayıyı kaydeder ve ardından sonucu A'da saklar.
00000110	U	VEM A M [B]	A yazmacını ve B bellek adresinde saklanan bir sayıyı gösterir ardından sonucu A'da saklar.
00000111	G	VEYA A B	A ve B kayıtları arasında veya işlemi yapar, ardından A ile sonuçlanır.
00001000	T	ORI A I	A yazmacı ile verilen sayı arasında veya işlemi yapar, sonra sonucu A'da saklar.
00001001	U	ORM A M[B]	A yazmacı ile A yazmacının içindeki değer arasında işlem yapar. bellek adresi B, ardından sonucu A'da saklar.
00001010	G	SUB A B	A ve B kayıtları arasında çıkarma işlemi yapar, ardından sonucu A'da saklar.
00001011	T	SUBI A I	A yazmacı ile verilen yazmaç arasında çıkarma işlemi yapar. sayısını hesaplar, ardından sonucu A'da saklar.
00001100	U	SUBM A M[B]	A yazmacı ile değer arasında çıkarma işlemi yapar B bellek adresinin içinde, ardından sonucu A'da saklar.
00001101	T	INC A	A kaydını 1 artırır.
00001110	T	ARALIK A	B kaydını 1 azaltır.
00001111	T	SHL A I	Kaydedici A değerini I kez sola kaydırır.
00010000	T	SHR A I	A değerini I katına kaydırın.
00010001	G	BRE A B	A ve B eşitse A satırında dallanma.
00010010	G	BRN A B	A ve B eşit değilse A satırında dallanma.
00010011	G	J A B	A + B olan belirli satır numarasına atlayın.
00010100	G	NOR A B	A ve B kayıtları arasında ne işlem yapar, ardından A ile sonuçlanır.
00010101	U	LDA A M[B]	B bellek adresindeki değeri A kaydına yükler.
00010110	U	STR A M[B]	Değeri A yazmacından B bellek adresine depolar.
00010111	G	SWP A B	A ve B kayıtlarını değiştirir.
00011000	G	DAHA AZ A B	A'nın B'den küçük olup olmadığını kontrol eder.
00011001	G	GRT A B	A'nın B'den büyük olup olmadığını kontrol eder.
00011010	G	PRI A	A kaydının değerini yazdırır. Değerin şu değere dönüştürülmesi gerekir ondalık sayı.
00011011	T	PRII I	I değerini yazdırır. Değerin ondalık sayıya dönüştürülmesi gerekir Numara.
00011100	U	PRIM M[A]	A bellek adresindeki değeri yazdırır. Değerin şu olması gerekir ondalık sayıya dönüştürülür.
00011101	T	LDAI A I	I değerini A kaydına yükle

00011110	-	ÇIKIŞ	Program sona erer.
----------	---	-------	--------------------

ÖNEMLİ NOTLAR:

- Ödevinizi öğrenci kimliğiniz (StudentID.zip) ile adlandırılmış bir **zip dosyası** olarak gönderin ve bu dosya şunları içermelidir:
 - YourStudentID.c dosyası
 - Giriş dosyalarının ve terminalin ekran görüntülerini içeren YourStudentID.pdf dosyası.
- stdio.h ve string.h dışında herhangi bir kütüphane kullanmayın.
- Çıktı biçimi verildiği gibi olmalıdır, değiştirmeyin.
- Çalışmanızı verilen "gcc --ansi your_program.c -o your_program" komutu ile derleyin.
- Bu ödev için son teslim tarihinden sonra bir demo oturumu düzenlenecektir. Demo oturumları için zaman çizelgesi **19 Mayıs'ta** Teams üzerinden sizinle paylaşılacaktır. Tercih ettiğiniz zaman dilimlerini doldurmak için **iki gününüz** olacak. **Zaman dilimi seçimi için son tarihten sonra hiçbir mazeret kabul edilmeyecektir.**
- Sorularınız ve sorunlarınız için bana her zaman **e-posta yoluyla** (incikaramahmutoglu@gtu.edu.tr) ulaşabilir veya beni İnsan Bilgisayar Etkileşimi Laboratuvarı'nda bulabilirsiniz. (No: 120) 7 Mayıs 2024 ve 14 Mayıs 2024 tarihlerinde saat 13:30'da mesai saatleri içinde ve 14:30'da.