

2023/1

BLM3061 Mikroişlemci Sistemleri ve Assembly Dili

Ödev 1

Verilen bir haritada 1 piksel kalınlığında ve 4 komşuluğa göre çizilmiş yollar bulunmaktadır. Haritada a) sadece 1 beyaz komşusu olan, b) 3 beyaz komşusu olan ve c) 4 beyaz komşusu olan pikseller önemli piksellerdir ve düğüm noktası olarak isimlendirilirler, ayrıca 2 beyaz komşusu olan pikseller geçiş yolu olup hangi iki düğüm arasında bağlantı olduğunu belirlemek haricinde önemsizdir. Harita PGM P5 formatta resim dosyasından okunacak olup, int tipinde sayı olarak bir matris şeklinde tanımlanmaktadır. Beyaz ile işaretli yollar 255 değeri ile siyah zemin ise 0 değeri ile kodlanmıştır. C++ tarafında resim okuma, konsola yazdırma, konsoldan okuma gibi yüksek seviye fonksiyonları hazır olarak içeren, assembly tarafında ise a) (50 puan) girdi olarak harita verisinin adresi ile haritanın en ve boy değerlerini alan ve sonuçta kaç adet düğüm olduğunu ve bu düğümlerin indislerini döndüren inline assembly bloğu ve b) (50 puan) hangi düğümler arası doğrudan bağlantı olduğunu döndüren (adjacency matrix) assembly fonksiyonu içeren kodu tamamlayınız.

Not:

- Beyaz ile işaretli yollar harita üst, alt, sağ veya sol kenarlarından en az 1 piksel içeriden geçmektedir.
- Düğüm sayısı en fazla 99 adet olacaktır.
- Resim dosyasını okuyan fonksiyonda dinamik olarak hafıza tahsisi yapıldığı için matris satırları hafızada ardışık olarak yerleşmemiş olabilir.
- Assembly kodunuzu 32 bit x86 mimarisi ile uyumlu olacak şekilde yazınız.
- Verilen README dosyasını inceleyiniz: kurulum, derleme, çalıştırma ve hata ayıklama için gerekli adımlar burada verilmiştir. Verilen adımlar 64 bit Intel mimarisinde kurulu 64 bit linux os üzerinde 32 bitlik derleyici ve araçların kurulup çalıştırılmasına yöneliktir.
- Kodunuzu 64 bitlik linux işletim sistemi üzerinde derleyiniz.
- Inline assembly ve assembly fonksiyonunu 32 bit Intel mimarisi ile uyumlu olacak şekilde yazınız.
- 32 bit Intel mimarisinde dolaylı hafıza erişimi için ebp, ebx, esi ve edi yazmaçlarının kullanımına dikkat ediniz.
- 32 bit Intel mimarisinde linux işletim sisteminde fonksiyona parametre aktarımı sağdan sola doğru yığına yerleştirme ile sağlanmaktadır.
- 32 bit Intel mimaride varsayılan olarak eax yazmacı üzerinden fonksiyon değer döndürür.
- 32 bit Intel mimaride fonksiyon çağrılarında geri dönüş için yığına 4 byte'lık tek dönüş değeri atılır.
- int veri tipi 4 byte genişliğindedir.
- Derleme ara çıktısı olarak assembly kodu da üretilmektedir, bu ara çıktıyı inceleyiniz.
- Inline assembly bloğu için aşağıdaki formatı kullanın:

```
asm(  
    " mov ebx, [ebp+16] ;"  
    ...  
);
```

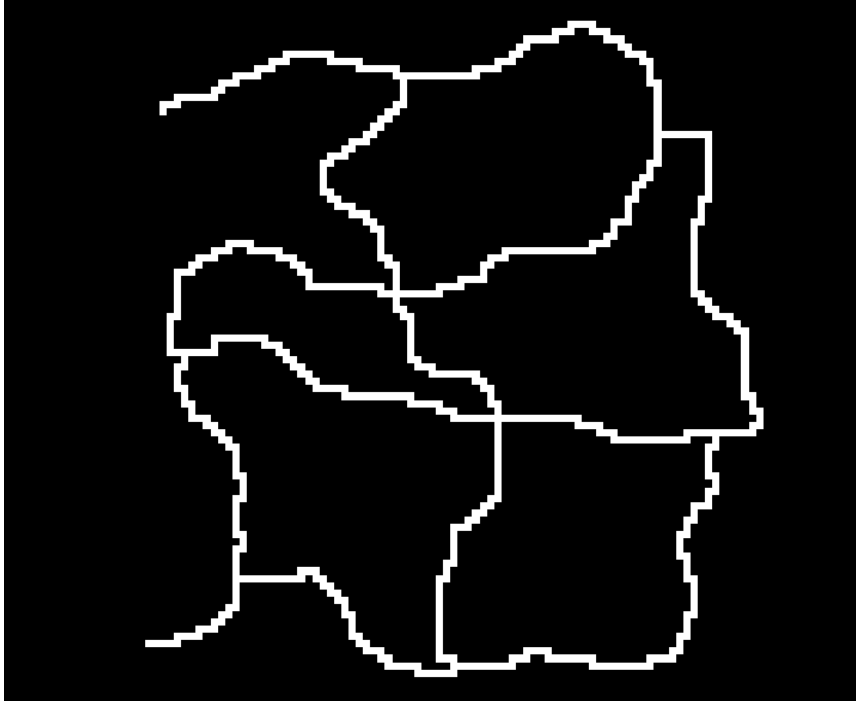
- Inline assembly ile fonksiyondan değer döndürmek için fonksiyon içinde bir yerel değişken tanımlanıp, assembly tarafında yerel değişkene değer atanarak yerel değişkenin döndürülmesi sağlanabilir. Tanımlanan yerel değişken ebp-4 adresindedir.

```

int findNodes(int n_, int m_, int **resim_org_, int* index_i_, int* index_j_)
{
    int a;
    asm(...
        "mov [ebp-4], 1;"
    );
    return a;
}

```

Aşağıdaki 100x120 piksel boyutlarındaki bir harita (test1.pgm) için örnek çıktıya (düğüm isimlendirmesi uygulamaya göre farklılık gösterebilir) yer verilmiştir:



DugumSayisi: 10

DugumIndisleri:

i:11 j:57

i:16 j:24

i:19 j:92

i:41 j:56

i:49 j:27

i:58 j:70

i:60 j:100

i:80 j:34

i:89 j:22

i:92 j:64

DugumBaglantiMatrisi:

0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1
0	1	1	0	1	0	0	1	0	0
0	0	0	1	0	1	0	1	0	0
0	0	0	0	1	0	1	0	1	0
0	0	0	0	0	1	0	0	0	0
0	0	0	1	1	0	0	0	1	1
0	0	0	0	0	1	0	1	0	1
0	0	1	0	0	0	0	1	1	0

Teslim Şekli:

Microsoft Teams üzerinden açılacak olan Ödev 1 arayüzünden yükleme yapılmalıdır. Mail ile iletilen cevaplar kesinlikle dikkate alınmayacaktır.

Yüklenecek Dosyalar:

- Cevabınızı içeren güncellenmiş main.cpp ve findAdj.asm dosyaları dahil tüm kaynak kodları
- Tüm test dosyaları
- Projenin derleme çıktısı olan *.o, *.S ve çalıştırılabilir map2adj dosyası
- Kodunuzu anlattığınız ve demo yaptığınız en fazla 3 dk'lık video

Son Teslim Zamanı:

25/12/2023 09:00