**Hacettepe Üniversitesi**

**Bilgisayar Mühendisliği Bölümü**

**BBM341 Sistem Programlama**

**Genel sınav – 14 Ocak 2015**

**Öğrenci Adı:** ...........................................

**Numarası:** ...........................................

**Soru 1.** C programla dilinde “double \*p;” tanımındaki p değişkeninin boyu x86-64 mimarisi için nedir?

a) 8bit b) 16bit c) 32bit d) 64bit e) Hiçbiri

**Soru 2.** C programla dilinde “double p;” tanımındaki p değişkeninin boyu x86-64 mimarisi için nedir?

a) 8bit b) 16bit c) 32bit d) 64bit e) Hiçbiri

**Soru 3.** Aşağıdaki biri 32bit diğeri 8bit olan iki işaretli sayı için işlemin sonucunu hesaplayınız.

a=0x00235590 b=0x81 a-b=?

**Soru 4.** %ebp yazmacında başlangıç adresi bulunan ve 32 bitlik sayılardan oluşan bir dizinin bir elemanına %eax yazmacındaki değeri aktaran komutu yazınız. Dizinin kaçıncı elemanına erişileceği bilgisi %esi yazmacındadır.

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

|  |  |  |  |
| --- | --- | --- | --- |
| **Normalized Values**  **Condition: exp ≠ 000…0 and exp ≠ 111…1**  **E = exp – Bias** | **Denormalized Values**  **Condition:**  **exp = 000…0**  **E = –Bias + 1** | **Bias = 2k-1 - 1** |  |

**Soru 5.** Yukarıdaki bilgiler kapsamında 12010 ve 12110 sayılarını 8 bitlik kayan noktalı sayı olarak kodlayınız.

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**Soru 6.** Aşağıdaki komut kümesini uyguladığınızda sonuç ne olur? Şekili güncelleyiniz.

|  |  |
| --- | --- |
|  | **movl 8(%ebp), %edx**  **movl 12(%ebp), %ecx**  **movl (%edx), %ebx**  **movl (%ecx), %eax**  **movl %ebx, -8(%edx)**  **movl %eax, -8(%ecx)** |
|  |  |

**Soru 7.** Konumsal Yerellik (*Spatial Locality*) kavramı kapsamında aşağıdaki kod kesimini etkin çalışan bir kod olarak değerlendirir misiniz? Neden?

**int sum\_array\_cols(int a[M][N])**

**{**

**int i, j, sum = 0;**

**for (j = 0; j < N; j++)**

**for (i = 0; i < M; i++)**

**sum += a[i][j];**

**return sum;**

**}**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**Soru 8.** %2 “miss” oranıyla yapılan veri erişimleri %95 “hit” oranıyla yapılanlara göre kaç kat daha iyidir? Ön bellekten erişim zamanını 1 birim, ana bellekten erişim zamanını 100 birim alınız.

a) 2 kat b) 3 kat c) 4 kat d) 6 kat e) Hiçbiri

**Soru 9.**  Bağlayıcı (Linker) birden fazla dosyayı biraraya getirip hedef dosyayı oluştururken dosyalar arasındaki referansları çözmede nasıl bir veri yapısı kullanır. Açıklayınız.

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**Soru 10.**  Aşağıdaki kod kesimi uygulandığında çıktısı ne olur?

|  |  |
| --- | --- |
| **void fork4()**  **{**  **printf("L0\n");**  **if (fork() == 0) {**  **printf("L1\n");**  **if (fork() == 0) {**  **printf("L2\n");**  **fork();**  **}**  **}**  **printf("Bye\n");**  **}** | **……………………………………………………**  **……………………………………………………**  **……………………………………………………**  **……………………………………………………**  **……………………………………………………**  **……………………………………………………**  **……………………………………………………**  **……………………………………………………**  **……………………………………………………**  **……………………………………………………** |

**Soru 11.**  İşletim sistemi bir göreve “*signal*” gönderdiğinde görevin gösterebileceği tepki türlerini açıklayınız.

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**Soru 12.** Aşağıdaki fonksiyon için derleme işlemi yaparak simgesel dille (Intel 32 bit mimari) komutları yazınız. Yığıt yapısını çiziniz.

**int arith(int x, int y)**

**{**

**return x+y;**

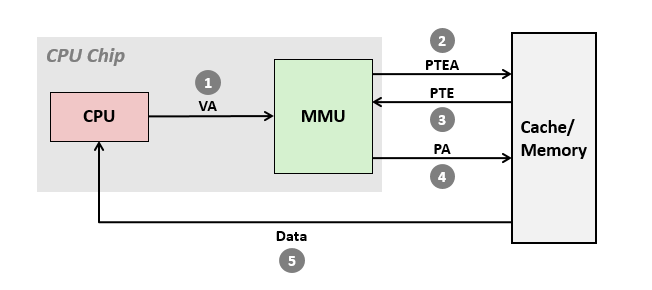
**}**

**Soru 13.** Soru 12’deki fonksiyonu çağıran kod kesimi için simgesel dille (Intel 32 bit mimari) komutları yazınız.

**Soru 14.**  **movl %ebx, (%edx)** komutu için verinin ön bellekte bulunamaması (*miss*) ve bulunması (*hit*) durumlarını içerecek şekilde veriye erişimin adımlarını çizim üzerinde gösteriniz ve birkaç anahtar sözcük kullanarak açıklayınız. (sorgula, adresle, oku, yaz, vb.)

|  |  |
| --- | --- |
| **“Hit”** | **“Miss”** |
|  |  |

**Soru 15.**  Görüntü bellek (Virtual Memory) yönetimi kapsamında aşağıdaki şekildeki adımları birer cümle ile açıklayınız.

****

**(1)…………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**(2)…………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**(3)…………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**(4)…………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**(5)…………………………………………………………………………………………………**

**……………………………………………………………………………………………………**

**……………………………………………………………………………………………………**