Name, Last name, Student #:

01.06.2017

1st	2nd	3rd	Grade/Toplam
4	4	3	PLO / PÖÇ

Karadeniz Technical University

Faculty of Engineering

Department of Computer Engineering

COM/BİL 2000 Final Exam

1. Express **in one sentence** the basic advantage of using interrupt initiated data transfer over data transfer under program control without an interrupt. (33 points) **/*/** Kesme ile başlatılan veri aktarımının kesmesiz program denetimi ile yapılan veri aktarımına göre temel avantajını **bir cümlede** ifade ediniz. (33 puan)

Veri aktaracak cihazın hazır olup olmadığını kontrol etmesi gerekmediği için daha az kaynak kullanması

2. It is necessary to transfer a 32KB data block from the buffer of a magnetic disk to a memory section starting from address $34AC_6$ by means of 16-bit DMA transfer. **/*/** Bir manyetik disk tamponundan 32 kB'lık bir blok verinin $3A4C_6$ adresinden başlayan bir bellek kısmına 16-bit DMA ile aktarılması gerekmektedir.

   a. Give the initial values that the CPU must transfer to the DMA controller. (33 points) **/*/** CPU'nun DMA denetleyiciye aktarması gereken başlangıç değerlerini veriniz. (33 puan)

Başlangıç adresi kaydedicisine 3A4C değeri ve blok boyu kaydedisine 32 kB / 2 (16 bitlik DMA bir aktarımda 2 byte aktarır) 16384 yazılır.

   b. Give the step-by-step account of actions taken during the input of the first two words. (34 points) **/*/** İlk iki kelimenin okunması sırasında yapılan eylemleri adım adım veriniz. (34 puan)

DMA denetleyici adres yoluna 3A4C adresini çıkarır.
" " yazma işareti ile 16 bitlik veriyi belleğe yazar.
" " adres kaydedicisini 2 artırır.
" " blok boyu kaydedicisini 1 azaltır.
" " adres yoluna 3A4E adresini çıkarır.
" " yazma işareti ile 16 bitlik veriyi belleğe yazar.
" " adres kaydedicisini 2 artırır.
" " blok boyu kaydedicisini 1 azaltır.

3.a. Assembly language runs extremely fast even though it is tedious to code with limited number of registers and instructions. That is why, it is reserved for critical sections of an algorithm which needs to respond very fast. On the other hand, high level programming languages run slower (compared to Assembly) but they are easier to learn and code. Both high level and low level languages can be used together by calling each other's functions as an external function for improved efficiency. Explain **in one sentence**, the reason why **fast call** convention is preferred by most of the modern compilers to pass arguments from a high level language (eg. C++) to a function coded in Assembly. (5 points) **/*/** Sınırlı sayıda kaydedici ve emirleri ile Assembly dilinde kodlama çok zor olmasına karşı çalışma hızı çok yüksektir. Bu nedenle bir algoritmanın çok hızlı cevap vermesi gereken kritik kısımları için ayrılmıştır. Diğer yandan yüksek seviyeli programlama dilleri, (Assembly diline kıyasla) daha yavaş çalışır fakat öğrenmesi ve kodlanması daha kolaydır. İyileştirilmiş verim için hem yüksek seviyeli hem de düşük seviyeli diller, diğer dildeki fonksiyonları dışta (external) tanımlanan fonksiyon olarak çağırarak birlikte kullanılabilirler. Yüksek seviyeli dillerden (örn. C++) Assembly dilinde kodlanmış bir fonksiyona argumanları geçirmek için neden modern derleyicilerin çoğunda **hızlı çağırma** (fast call) kuralının tercih edildiğini **bir cümlede** açıklayınız. (5 puan)

Argümanları yığına iterek çağırmaya göre daha hızlı olduğu için

b. Fill out Intel x86/x64 architecture register names corresponding to **arguments** and **return value** of functions used by 32/64-bit compilers with **fast call** convention in the table below. (5 points) **/*/ Hızlı çağırma** kuralını kullanan 32/64-bit derleyicilerin, argümanlar ve fonksiyon geri dönüş değeri için kullandıkları Intel x86/x&4 mimarisi kaydedici isimlerini aşağıdaki tabloda doldurunuz. (5 puan)

	X64 reg. / kay.	X86 reg. / kay.
Return value / Geri dönüş değeri	rax	eax
1st Argument / 1. Argüman	rcx	ecx
2nd Argument/ 2. Argüman	rdx	edx
3rd Argument / 3.Argüman	r8	r8d
4th Argument / 4. Argüman	r9	r9d
4+ Argument / 4+ Argüman	Stack	stack

} x86 için derlenirse kaydedici üzerinden aktarılmaz, yığına itilir.

C. Code an x64 assembly function to return the minimum of two 64-bit integers passed as arguments. (5 points) **/*/** Arguman olarak aktarılan iki 64-bit tamsayıdan küçük olanı geri döndüren bir assembly dili programı kodlayınız. (5 puan)

```
cmp ecx, edx
jg L1
mov eax, ecx
jmp L2
L1: mov eax, edx
L2: ret
```

veya

```
cmp ecx, edx
jl L1
mov eax, edx
jmp L2
L1: mov eax, ecx
L2: ret
```