

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
Telafi sınavı – 3 Ocak 2014

Öğrenci Adı:

Numarasi:

Soru 1. Aşağıdaki biri 32bit diğeri 8bit olan iki işaretsiz sayıyı toplayınız.

a=0x00235500 b=0x8D a+b=?

```
leal (%eax,%eax,2), %eax
sall $2, %eax
```

```
t <- x+x*2
return t << 2;
```

Soru 2. Yukarıdaki örnekte %eax yazmacı 12_{10} ile çarpılmaktadır. Benzer biçimde %eax yazmacını 19_{10} ile çarpan kodu yazınız.

Soru 3. %eax yazmacını 21_{10} ile çarpan kodu yazınız.

Soru 4. Aşağıda onlu tabanda verilen sayıların ikili tabanda karşılıklarını yazınız.

[illegible]

Normalized Values Condition: $\text{exp} \neq 000 \dots 0$ and $\text{exp} \neq 111 \dots 1$ $E = \text{exp} - \text{Bias}$	Denormalized Values Condition: $\text{exp} = 000 \dots 0$ $E = -\text{Bias} + 1$	Bias = $2^{k-1} - 1$	
---	---	--	--

Soru 5. Yukarıdaki bilgiler kapsamında 14_{10} sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız.

Soru 6. Yukarıdaki bilgiler kapsamında $15_{10}/16_{10}$ sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız.

Soru 7. Aşağıdaki komut kümesini uyguladığınızda sonuç ne olur? Şekli güncelleyiniz.

%eax		123	0x124	
%edx		456	0x120	
%ecx			0x11c	
%ebx			0x118	
%esi			0x114	
%edi		0x120	0x110	
%esp		0x124	0x10c	
%ebp	0x104	Rtn adr	0x108	
			0x104	
			0x100	

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

Soru 8. Aşağıdaki “**call**” komutunu uyguladığınızda elde edilen değerlerle şeklin sağ tarafını güncelleyiniz.

804854e:	e8 3d 06 00 00	call	8048b90 <main>
8048553:	50	pushl	%eax

			call 8048b90
0x110		0x110	
0x10c		0x10c	
0x108	123	0x108	123
0x104		0x104	
%esp	0x108	%esp	
%eip	0x804854e	%eip	

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
1. Ara sınav – 2 Aralık 2013

Öğrenci Adı:

Numarasi:

Soru 1. C programla dilinde “short int *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 “short int i;” tanımındaki i 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 işaretsiz sayıyı toplayınız.

a=0x00235500 b=0xFC a+b=?

Soru 4. Aşağıdaki biri 32bit diğeri 8bit olan iki işaretsiz sayıyı toplayınız.

a=0x00235500 b=0x8C a+b=?

```
leal (%eax,%eax,2), %eax
sall $2, %eax
```

```
t <- x+x*2
return t << 2;
```

Soru 5. Yukarıdaki örnekte %eax yazmacı 12_{10} ile çarpılmaktadır. Benzer biçimde %eax yazmacını 17_{10} ile çarpan kodu yazınız.

Soru 6. %eax yazmacını 20_{10} ile çarpan kodu yazınız.

Soru 7. Aşağıda onlu tabanda verilen sayıların ikili tabanda karşılıklarını yazınız.

5.25	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td></tr></table>						.				
					.							
10.5	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td></tr></table>						.				
					.							
5.125	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td></tr></table>						.				
					.							
5.875	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td></tr></table>						.				
					.							

Normalized Values Condition: $\text{exp} \neq 000 \dots 0$ and $\text{exp} \neq 111 \dots 1$ $E = \text{exp} - \text{Bias}$	Denormalized Values Condition: $\text{exp} = 000 \dots 0$ $E = -\text{Bias} + 1$	Bias = $2^{k-1} - 1$	
---	---	--	--

Soru 8. Yukarıdaki bilgiler kapsamında 12_{10} sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız.

.....

.....

.....

Soru 9. Yukarıdaki bilgiler kapsamında $13_{10}/16_{10}$ sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız.

.....

.....

.....

Soru 10. Aşağıdakilerden hangisi yanlıştır?

- a) `movl (%ebp),%eax`
- b) `movl %ebp,%eax`
- c) `movl %ebp,(%eax)`
- d) `movl (%ebp),(%eax)`
- e) Hiçbiri

Soru 11. Aşağıdaki komut kümesini uyguladığınızda sonuç ne olur? Şekli güncelleyiniz.

%eax	
%edx	
%ecx	
%ebx	
%esi	
%edi	
%esp	
%ebp	0x104

	Address
123	0x124
456	0x120
	0x11c
	0x118
	0x114
0x120	0x110
0x124	0x10c
Rtn adr	0x108
	0x104
	0x100

```

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

```

Soru 12. Aşağıdaki “**call**” komutunu uyguladığınızda elde edilen değerlerle şeklin sağ tarafını güncelleyiniz.

```

804854e: e8 3d 06 00 00    call 8048b90 <main>
8048553: 50                pushl %eax

```

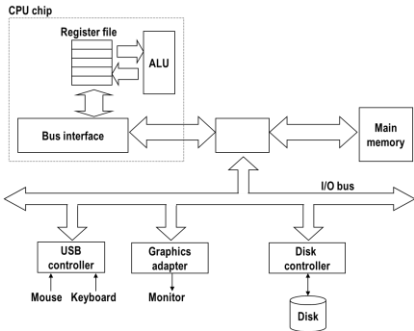
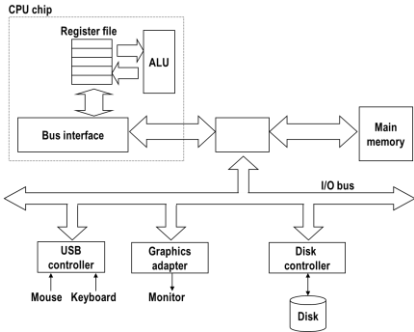
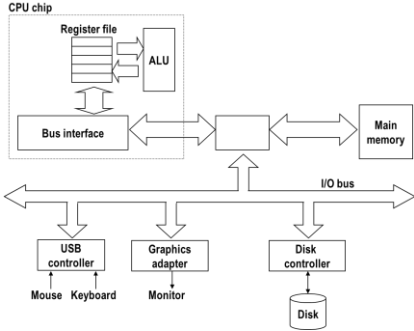
		call 8048b90	
0x110		0x110	
0x10c		0x10c	
0x108	123	0x108	123
0x104		0x104	
%esp	0x108	%esp	
%eip	0x804854e	%eip	

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
2. Ara sınav – 30 Aralık 2013

Öğrenci Adı:

Numarası:

Soru 1. Doğrudan bellek erişim (DMA: *Direct Memory Access*) yöntemini kullanarak bir disk sektörünün okunmasına ilişkin adımları aşağıdaki şekiller üzerinde gösteriniz. Her adımı bir/iki cümle ile açıklayınız.



Soru 2. Zamansal Yerellik (*Temporal Locality*) kavramını bir/iki cümle ile açıklayınız.

.....

.....

.....

.....

Soru 3. Konumsal Yerellik (*Spatial Locality*) kavramını bir/iki cümle ile açıklayınız.

.....

.....

.....

.....

Soru 4. %99 “hit” oranıyla yapılan veri erişimleri %95 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 5. Aşağıda verilen iki kod kesimi derlenip (*compile*) bağlandığında (*link*) ne sonuç elde edilir? (“deg” adlı değişken kapsamında cevaplayınız.)

p1.c

```
int deg=5;

p1() {
}
```

p2.c

```
int deg=6;

p2() {
}
```

.....

.....

.....

.....

Soru 6. Aşağıda verilen iki kod kesimi derlenip (*compile*) bağlandığında (*link*) ne sonuç elde edilir? (“deg” adlı değişken kapsamında cevaplayınız.)

p1.c

```
int deg;

p1() {
}
```

p2.c

```
int deg;

p2() {
}
```

.....

.....

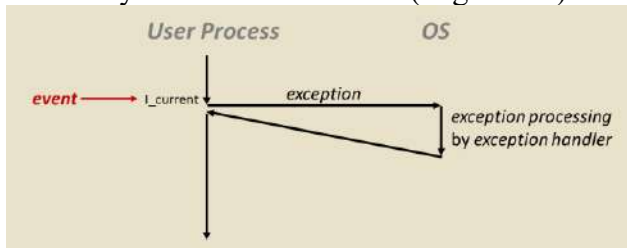
.....

.....

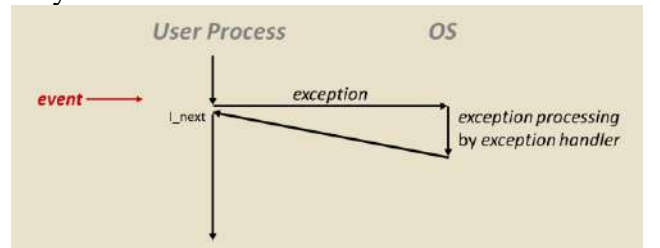
Soru 7. Aşağıdakilerden hangisi bağlayıcı (Linker) kullanımının olumlu yanı değildir?

- a) Modüler gerçekleştirime izin verir.
b) Derleme zamanını kısaltabilir.
c) İşletim zamanını kısaltabilir.
d) Yazılım kitaplıklarının yer gereksinimini azaltabilir.
e) Hiçbiri.

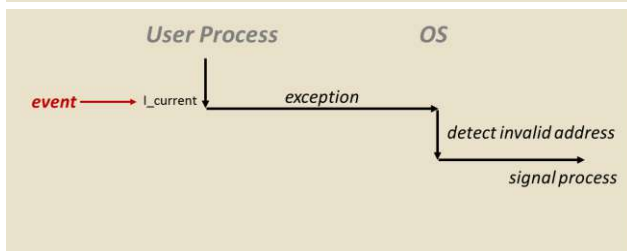
Soru 8. Sayfa bulunamadı hatası (Page Fault) sonucunda aykırı durum nasıl ele alınır?



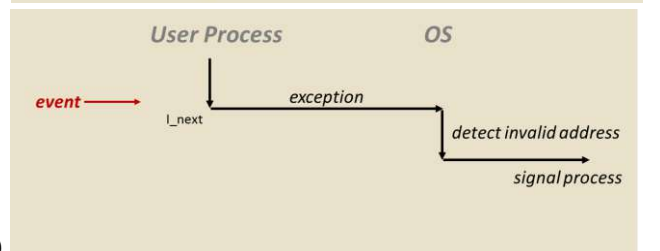
a)



b)



c)



d)

e) Hiçbiri.

Soru 9. Aşağıdakilerden hangi ifade A, B ve C görevleri için yanlıştır?



- a) A ve B eşzamanlıdır (*Concurrent*).
- b) A ve C eşzamanlıdır (*Concurrent*).
- c) A ve D eşzamanlıdır (*Concurrent*).
- d) B ve C eşzamanlıdır (*Concurrent*).
- e) B ve D eşzamanlıdır (*Concurrent*).

Soru 10. Aşağıdaki kod kesimi uygulandığında çıktısı hangisi olamaz?

```
void fork3()
{
    printf("G0\n");
    fork();
    printf("G1\n");
    fork();
    printf("G2\n");
    fork();
    printf("Bye\n");
}
```

- | | | | | | | | | | |
|----|-----|----|-----|----|-----|----|-----|----|-----|
| a) | G0 | b) | G0 | c) | G0 | d) | G0 | e) | G0 |
| | G1 | | G1 | | G1 | | G1 | | G1 |
| | G1 | | G2 | | G1 | | G2 | | G1 |
| | G2 | | G1 | | G2 | | G1 | | G2 |
| | Bye | | Bye | | Bye | | Bye | | G2 |
| | G2 | | G2 | | G2 | | G2 | | G2 |
| | Bye | | Bye | | Bye | | G2 | | G2 |
| | Bye | | Bye | | Bye | | Bye | | Bye |
| | G2 | | Bye | | Bye | | Bye | | Bye |
| | Bye | | G2 | | Bye | | Bye | | Bye |
| | Bye | | Bye | | G2 | | Bye | | Bye |
| | Bye | | Bye | | Bye | | Bye | | Bye |
| | G2 | | G2 | | G2 | | G2 | | Bye |
| | Bye | | Bye | | Bye | | Bye | | Bye |
| | Bye | | Bye | | Bye | | Bye | | Bye |

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
Bütünleme Sınavı – 31 Ocak 2014

Öğrenci Adı:

Numarasi:


```
leal (%eax,%eax,2), %eax
sall $2, %eax
```

```
t <- x+x*2
return t << 2;
```

Soru 1. Yukarıdaki örnekte %eax yazmacı 12₁₀ ile çarpılmaktadır. Benzer biçimde %eax yazmacını 6₁₀ ile çarpan kodu yazınız.

Soru 2. Aşağıda onlu tabanda verilen sayıların ikili tabanda karşılıklarını yazınız.

0.25	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td><td></td></tr></table>							.					
						.								
0.5	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td><td></td></tr></table>							.					
						.								
0.875	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td><td></td></tr></table>							.					
						.								
0.125	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td><td></td></tr></table>							.					
						.								

Normalized Values Condition: $\text{exp} \neq 000 \dots 0$ and $\text{exp} \neq 111 \dots 1$ $E = \text{exp} - \text{Bias}$	Denormalized Values Condition: $\text{exp} = 000 \dots 0$ $E = -\text{Bias} + 1$	Bias = $2^{k-1} - 1$	
---	---	--	--

Soru 3. Yukarıdaki bilgiler kapsamında 1_{10} sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız.

Soru 4. Aşağıdaki örnekte iki sayının arasındaki farkı bulan fonksiyonun C programlama dili ve x86 simgesel dili ile kodlamaları verilmiştir. Siz de üç sayının en büyüğünü bulan fonksiyon için kodlamaları C programlama dili ve x86 simgesel dili ile yapınız.

```
int absdiff(int x, int y)
{
    int result;
    if (x > y) {
        result = x-y;
    } else {
        result = y-x;
    }
    return result;
}
```

```
absdiff:
    pushl %ebp
    movl %esp, %ebp
    movl 8(%ebp), %edx
    movl 12(%ebp), %eax
    cmpl %eax, %edx
    jle .L6
    subl %eax, %edx
    movl %edx, %eax
    jmp .L7
.L6:
    subl %edx, %eax
.L7:
    popl %ebp
    ret
```

Soru 5. Soru 4’de verilen örnek için yığıt yapısını çiziniz.

Soru 6. Soru 4’deki çözümünüz için yığıt yapısını çiziniz.

Soru 7. Soru 4'deki çözümünüzde geliştirdiğiniz kodu çağırın kesimi x86 simgesel dili ile kodlayınız.

Soru 8. Aşağıdaki kod kesimi uygulandığında olası bir çıktısını veriniz.

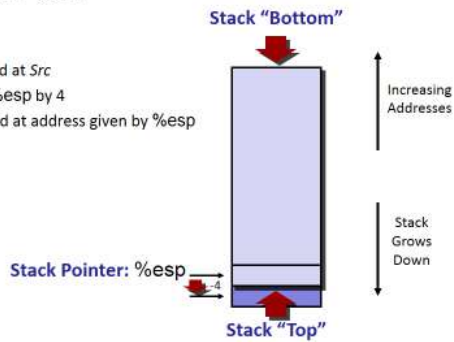
```
void fork4()
{
    printf("L0\n");
    if (fork() != 0) {
        printf("L1\n");
        if (fork() != 0) {
            printf("L2\n");
            fork();
        }
    }
    printf("Bye\n");
}
```

Soru 9. Görev anahtarlama (*Context Switching*) sırasında uygulanan adımları listeleyiniz.

IA32 Stack: Push

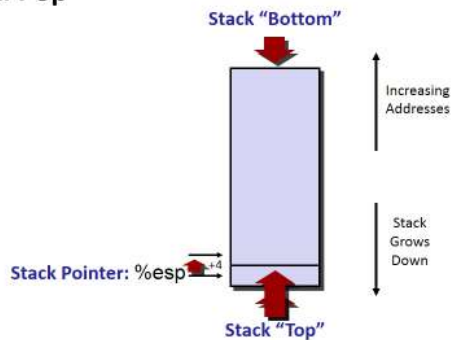
■ `pushl Src`

- Fetch operand at `Src`
- Decrement `%esp` by 4
- Write operand at address given by `%esp`



43

IA32 Stack: Pop



45

Procedure Control Flow

- Use stack to support procedure call and return

■ Procedure call: `call label`

- Push return address on stack
- Jump to *label*

■ Return address:

- Address of the next instruction right after call
- Example from disassembly

```
804854e: e8 3d 06 00 00  call 8048b90 <main>
8048553: 50
```

- Return address = `0x8048553`

■ Procedure return: `ret`

- Pop address from stack
- Jump to address

47

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
Genel sınav – 13 Ocak 2014

Öğrenci Adı:

Numarasi:

Soru 1. C programla dilinde “char *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 “char p;” tanımındaki p değişkeninin boyu x86-32 mimarisi için nedir?

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

Soru 3. Aşağıdaki biri 32bit diğeri 8bit olan iki işaretsiz sayıyı toplayınız.

a=0x00235500 b=0x80 a+b=?

```
leal (%eax,%eax,2), %eax
sall $2, %eax
```

```
t <- x+x*2
return t << 2;
```

Soru 4. Yukarıdaki örnekte %eax yazmacı 12_{10} ile çarpılmaktadır. Benzer biçimde %eax yazmacını 36_{10} ile çarpan kodu yazınız.

Soru 5. Aşağıda onlu tabanda verilen sayıların ikili tabanda karşılıklarını yazınız.

10.25	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td><td></td></tr></table>							.					
						.								
7.5	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td><td></td></tr></table>							.					
						.								
8.875	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td><td></td></tr></table>							.					
						.								
15.125	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td><td></td></tr></table>							.					
						.								

Normalized Values Condition: $\text{exp} \neq 000 \dots 0$ and $\text{exp} \neq 111 \dots 1$ $E = \text{exp} - \text{Bias}$	Denormalized Values Condition: $\text{exp} = 000 \dots 0$ $E = -\text{Bias} + 1$	Bias = $2^{k-1} - 1$	<p>Diagram illustrating the IEEE 754 floating-point format structure:</p> <ul style="list-style-type: none"> s: Sign bit (1 bit) exp: Exponent field (4-bits) frac: Fraction field (3-bits)
---	---	--	---

Soru 6. Yukarıdaki bilgiler kapsamında 30_{10} sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız.

Soru 7. Aşağıdaki komut kümesini uyguladığınızda sonuç ne olur? Şekli güncelleyiniz.

%eax	
%edx	
%ecx	
%ebx	
%esi	
%edi	
%esp	
%ebp	0x104

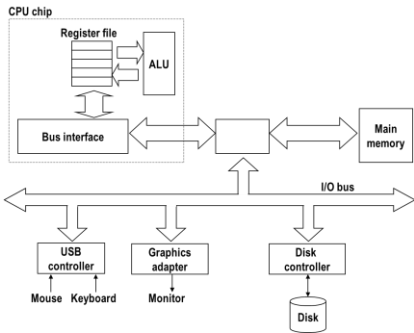
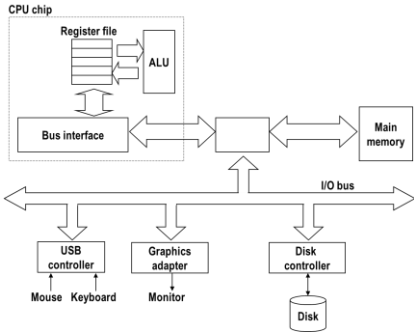
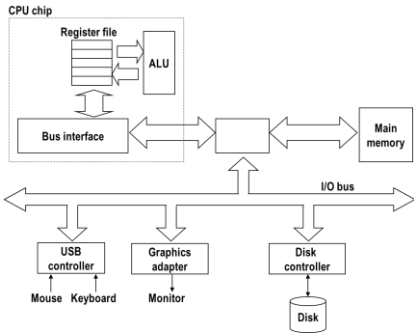
Address	
123	0x124
456	0x120
	0x11c
	0x118
	0x114
0x120	0x110
0x124	0x10c
Rtn adr	0x108
	0x104
	0x100

```

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

```

Soru 8. Doğrudan bellek erişim (DMA: *Direct Memory Access*) yöntemini kullanarak bir disk sektörüne yazmaya ilişkin adımları aşağıdaki şekiller üzerinde gösteriniz. Her adımı bir/iki cümle ile açıklayınız.



Soru 9. Aşağıdaki kod kesimi uygulanmadan önce ilgili verilerin ön bellekte olmadığını varsayınız. Bu kod kesimi uygulanırken veriler ön belleğe getirilecektir. Bu erişimlerde Zamansal Yerellik (*Temporal Locality*) Konumsal Yerellik (*Spatial Locality*) kavramlarına birer örnek veriniz.

.....

.....

.....

.....

.....

```
sum = 0;
for (i = 0; i < n; i++)
    sum += a[i];
return sum;
```

Soru 10. %98 “hit” oranıyla yapılan veri erişimleri %95 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 11. Aşağıdaki kod kesimi uygulandığında çıktısı hangisi olamaz?

```
void fork4()
{
    printf("L0\n");
    if (fork() != 0) {
        printf("L1\n");
        if (fork() != 0) {
            printf("L2\n");
            fork();
        }
    }
    printf("Bye\n");
}
```

- | | | | | | | | | | |
|----|-----|----|-----|----|-----|----|-----|----|-----|
| a) | L0 | b) | L0 | c) | L0 | d) | L0 | e) | L0 |
| | L1 | | Bye | | L1 | | Bye | | Bye |
| | L2 | | L1 | | Bye | | L1 | | L1 |
| | Bye | | L2 | | Bye | | Bye | | Bye |
| | Bye | | Bye | | L2 | | Bye | | L2 |
| | Bye | | Bye | | Bye | | L2 | | Bye |
| | Bye | | Bye | | Bye | | Bye | | Bye |

Soru 14. Aşağıdaki kod kesimi uygulandığında çıktısı ne olur? Kısaca açıklayınız.

```
void fork10()
{
    pid_t pid[N];
    int i;
    int child_status;
    for (i = 0; i < N; i++)
        if ((pid[i] = fork()) == 0)
            exit(100+i); /* Child */
    pid_t wpid = wait(&child_status);
    printf("Child %d terminated with exit status %d\n",
        wpid, WEXITSTATUS(child_status));
}
```

.....

.....

.....

.....

.....

.....

Soru 15. İşletim sistemi bir göreve “*signal*” gönderdiğinde görevin gösterebileceği tepki türlerini açıklayınız.

.....

.....

.....

.....

.....

.....

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
1. Ara sınav – 12 Kasım 2014

Öğrenci Adı:

Numarası:

Soru 1. C programla dilinde "float 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 "float *f;" tanımındaki i 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 sayıyı toplayınız.

a=0x00123400

b=0xFF

a+b=?

0x001233FF

00123400
+ FFFFFFFF
1 001233FF

Soru 4. Aşağıdaki biri 32bit diğeri 8bit olan iki işaretli sayıyı toplayınız.

a=0x00123400

b=0x7F

a+b=?

0x0012347F

00123400
+ 0000007F
0012347F

leal (%eax,%eax,2), %eax
sall \$2, %eax

t <- x+x*2
return t << 2;

Soru 5. Yukarıdaki örnekte %eax yazmacı 12₁₀ ile çarpılmaktadır. Benzer biçimde %eax yazmacını 72₁₀ ile çarpan kodu yazınız.

leal (%eax,%eax,8), %eax
sall \$3, %eax

Soru 6. %eax yazmacını 48₁₀ ile çarpan kodu yazınız.

leal (%eax,%eax,2), %eax
sall \$4, %eax

Soru 7. Aşağıda onlu tabanda verilen sayıların ikili tabanda karşılıklarını yazınız.

5.5 = 000101.100000
10.25 = 001010.010000
15.825 = 001111.110100
5.3 = 000101.010001

yuvarlama yoksa
yuvarlama varsa
01010

Normalized Values Condition: $\text{exp} \neq 000 \dots 0$ and $\text{exp} \neq 111 \dots 1$ $E = \text{exp} - \text{Bias}$	Denormalized Values Condition: $\text{exp} = 000 \dots 0$ $E = -\text{Bias} + 1$	Bias = $2^{k-1} - 1$	
---	--	-----------------------------	--

Soru 8. Yukarıdaki bilgiler kapsamında 30_{10} sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız.

$$30_{10} = 11110_2 \quad 1.1110 \quad E = 4$$

$$\text{exp} = 4 + 7 = 11$$

Soru 9. Yukarıdaki bilgiler kapsamında $15_{10}/16_{10}$ sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız.

$$15_{10}/16_{10} = 0.1111 \quad 1.111 \quad E = -1$$

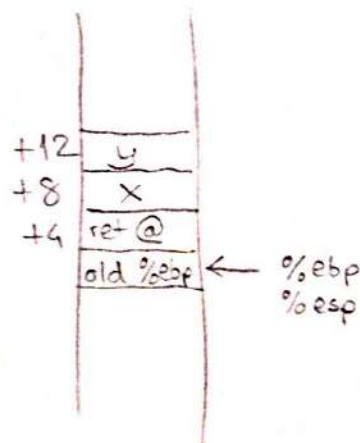
$$\text{exp} = -1 + 7 = 6$$

Soru 10. %ebp yazmacında başlangıç adresi bulunan ve 32 bitlik sayılardan oluşan bir dizinin bir elemanını okuyup %eax yazmacına aktaran komutu yazınız. Dizinin kaçınıcı elemanına erişileceği bilgisi %esi yazmacındadır.

`movl (%ebp, %esi, 4), %eax`

Soru 11. Aşağıdaki fonksiyon için derleme işlemi yaparak simgesel dille (Intel 32 bit mimari) komutları yazınız.

```
int arith(int x, int y)
{
    return x+y;
}
```



arith:

```
pushl %ebp
movl %esp, %ebp
movl 8(%ebp), %eax
addl 12(%ebp), %eax
popl %ebp
ret
```

Soru 12. Soru 11'deki fonksiyonu çağıran kod kesimi için simgesel dille (Intel 32 bit mimari) komutları yazınız.

X: .long 1

Y: .long 2

```

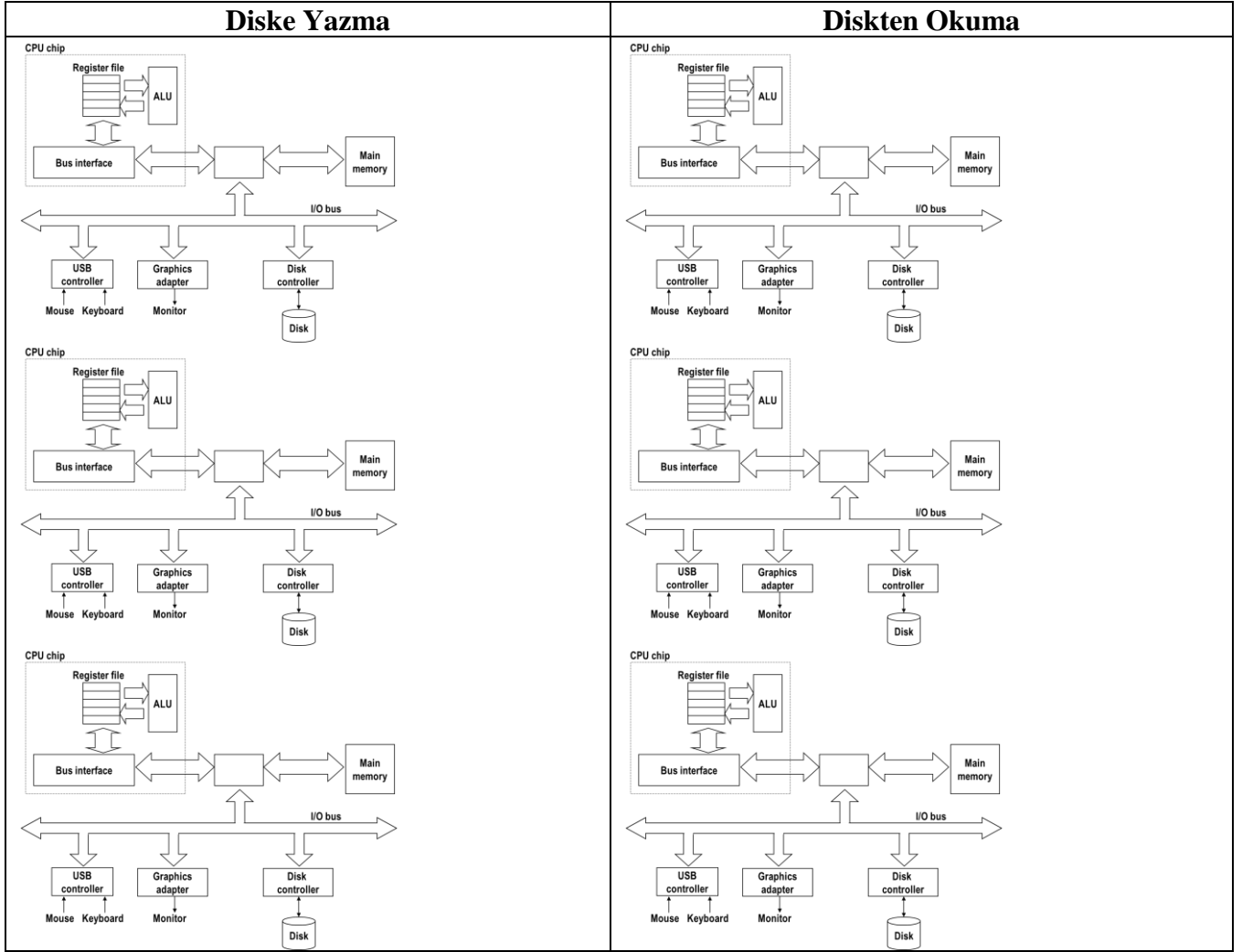
:
movl y, %eax
pushl %eax
movl x, %eax
pushl %eax
call arith
```

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
2. Ara sınav – 24 Aralık 2014

Öğrenci Adı:

Numarası:

Soru 1. Doğrudan bellek erişim (DMA: *Direct Memory Access*) yöntemini kullanarak bir disk sektörünün erişimine ilişkin adımları aşağıdaki şekiller üzerinde gösteriniz.



Soru 2. *Zamansal Yerellik (Temporal Locality)* kavramı aşağıdaki kod kesiminde hangi verilere erişim için söz konusudur, bir/iki cümle ile açıklayınız.

```
sum = 0;
for (i = 0; i < n; i++)
    a[i] = b[i] + c[i];
return sum;
```

.....

.....

.....

.....

Soru 3. *Konumsal Yerellik (Spatial Locality)* kavramı aşağıdaki kod kesiminde hangi verilere erişim için söz konusudur, bir/iki cümle ile açıklayınız.

```
sum = 0;
for (i = 0; i < n; i++)
    a[i] = b[i] + c[i];
return sum;
```

Soru 4. %99 “hit” oranıyla yapılan veri erişimleri %91 oranıyla yapılanlara göre kaç kat daha iyidir? Ön bellekten erişim zamanını 1 birim, ana bellekten erişim zamanını 100 birim alınız.

Soru 5. Aşağıda verilen iki kod kesimi derlenip (*compile*) bağlandığında (*link*) ne sonuç elde edilir?

p1.c

```
int deg=5;

p1() {
}
```

p2.c

```
int deg;

p1() {
}
```

Soru 6. Bağlayıcı (Linker) birden fazla dosyayı biraraya getirip hedef dosyayı oluştururken dosyalar arasındaki referansları çözmede nasıl bir veri yapısı kullanır. Açıklayınız.

Soru 7. Kod içindeki soruları yanıtlayınız.

```
...

/* a) Aşağıdaki satırda ne yapılmaktadır? */
addvec = dlsym(handle, "addvec");
if ((error = dlerror()) != NULL) {
    fprintf(stderr, "%s\n", error);
    exit(1);
}

/* b) Aşağıdaki satırda ne yapılmaktadır? */
addvec(x, y, z, 2);
printf("z = [%d %d]\n", z[0], z[1]);

/* c) Aşağıdaki satırda ne yapılmaktadır? */
if (dlclose(handle) < 0) {
    fprintf(stderr, "%s\n", dlerror());
    exit(1);
}
return 0;
}
```

- a)
- b)
- c)

Soru 8. Kod içindeki soruyu yanıtlayınız.

```
void fork11()
{
    pid_t pid[N];
    int i;
    int child_status;
    for (i = 0; i < N; i++)
        if ((pid[i] = fork()) == 0)
            exit(100+i);
    for (i = N-1; i >= 0; i--) {
        /* Aşağıdaki satırda ne yapılmaktadır? */
        pid_t wpid = waitpid(pid[i], &child_status, 0);
        if (WIFEXITED(child_status))
            printf(" Falan filan %d\n");
        else
            printf(" Yine falan filan %d\n");
    }
}
```

Soru 9. Aşağıdaki kod kesimi uygulandığında çıktısı ne olur?

```
void fork4()
{
    printf("Falan \n");
    if (fork() != 0) {
        printf("Filan\n");
        if (fork() != 0) {
            printf("Yalan\n");
            fork();
        }
    }
    printf("Görüşürüz.\n");
}
```

Soru 10. Aşağıdaki kod kesimi kapsamında P1() çağırıldığında çağırıldığında sonuç ne olur?

```

jmp_buf env;

P1()
{
    P2(); P3();
}

P2()
{
    if (setjmp(env)) {
        printf("Falan Filan \n");
    }
}

P3()
{
    longjmp(env, 1);
}

```

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
Genel sınav – 14 Ocak 2015

Öğrenci Adı:

Numarasi:

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 kaçınıcı elemanına erişileceği bilgisi %esi yazmacındadır.

Normalized Values Condition: $\text{exp} \neq 000 \dots 0$ and $\text{exp} \neq 111 \dots 1$ $E = \text{exp} - \text{Bias}$	Denormalized Values Condition: $\text{exp} = 000 \dots 0$ $E = -\text{Bias} + 1$	Bias = $2^{k-1} - 1$	 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">s 1</div> <div style="text-align: center;">exp 4-bits</div> <div style="text-align: center;">frac 3-bits</div> </div>
---	---	--	--

Soru 5. Yukarıdaki bilgiler kapsamında 120_{10} ve 121_{10} sayılarını 8 bitlik kayan noktalı sayı olarak kodlayınız.

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

%eax	
%edx	
%ecx	
%ebx	
%esi	
%edi	
%esp	
%ebp	0x104

Address
123
456
0x120
0x124
Rtn adr

```
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.

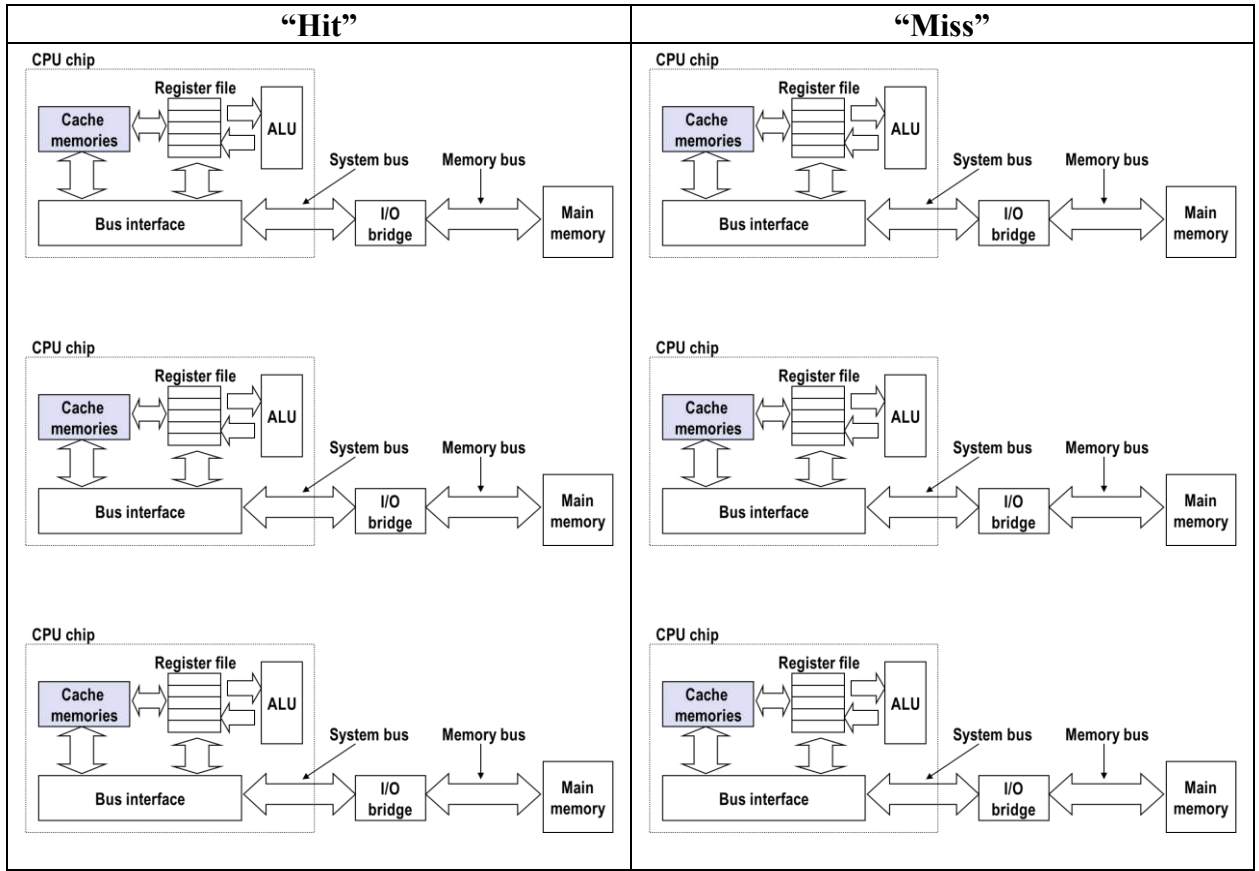
.....
.....
.....
.....
.....
.....
.....
.....
.....

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

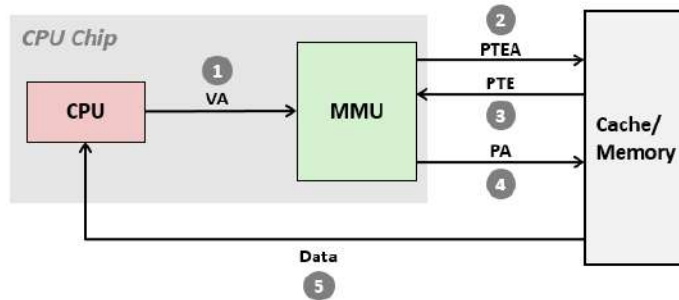
```
int arith(int x, int y)
{
    return x+y;
}
```

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.)



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



- (1)
- (2)
- (3)
- (4)
- (5)

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
1. Ara sınav – 3 Kasım 2015

Öğrenci Adı:

Numarasi:

Soru 1. C programla dilinde “float *p;” tanımındaki p değişkeninin boyu x86-64 mimarisi için nedir? (1 puan)

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

Gerekçe:

Soru 2. Aşağıdaki biri 32bit diğeri 8bit olan iki işaretsiz sayıyı toplayınız. (1 puan)

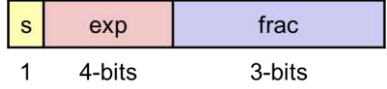
a=0x00FFFFFF b=0xFA a+b=?

Çözüm:

Soru 3. Aşağıda onlu tabanda verilen sayıların ikili tabanda karşılıklarını yazınız. (1 puan)

25.25	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td></tr></table>									.				
								.							
50.50	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td></tr></table>									.				
								.							
15.15	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td></tr></table>									.				
								.							
0.30	=	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>.</td><td></td><td></td><td></td><td></td></tr></table>									.				
								.							

Çözüm:

Normalized Values Condition: $\text{exp} \neq 000\dots 0$ and $\text{exp} \neq 111\dots 1$ $E = \text{exp} - \text{Bias}$	Denormalized Values Condition: $\text{exp} = 000\dots 0$ $E = -\text{Bias} + 1$	Bias = $2^{k-1} - 1$	
---	--	--	---

Soru 4. Yukarıdaki bilgiler kapsamında 60_{10} sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız. (1 puan)

Çözüm:

Soru 5. İzleyen C programını GCC ile derlediğimizde aşağıdaki kodu elde ediyoruz. C programındaki boşlukları doldurunuz. (2 puan)

```

1  int test(int x, int y) {
2      int val = _____;
3      if ( x<-3 ) {
4          if ( _____ )
5              val = _____;
6          else
7              val = _____;
8      } else if ( _____ )
9          val = _____;
10     return val;
11 }
```

```

x at %ebp+8, y at %ebp+12
1  movl    8(%ebp), %eax
2  movl    12(%ebp), %edx
3  cmpl    $-3, %eax
4  jge     .L2
5  cmpl    %edx, %eax
6  jle     .L3
7  imull   %edx, %eax
8  jmp     .L4
9  .L3:
10  leal    (%edx,%eax), %eax
11  jmp     .L4
12  .L2:
13  cmpl    $2, %eax
14  jg      .L5
15  xorl    %edx, %eax
16  jmp     .L4
17  .L5:
18  subl    %edx, %eax
19  .L4:
```

6. 7. ve 8. Sorular için:

```
1  int proc(void)
2  {
3      int x,y;
4      scanf("%x %x", &y, &x);
5      return x-y;
6  }
```

Yukarıdaki C kodunu GCC ile derlediğimizde aşağıdaki kodu elde ediyoruz.

```
1  proc:
2      pushl   %ebp
3      movl    %esp, %ebp
4      subl    $40, %esp
5      leal    -4(%ebp), %eax
6      movl    %eax, 8(%esp)
7      leal    -8(%ebp), %eax
8      movl    %eax, 4(%esp)
9      movl    $.LC0, (%esp)   Pointer to string "%x %x"
10     call    scanf
11     Bu noktadaki yığıt çerçevesini çiziniz.
12     movl    -4(%ebp), %eax
13     subl    -8(%ebp), %eax
14     leave
15     ret
```

“proc” fonksiyonunun aşağıdaki yazmaç değerleri ile başladığını varsayarak 6. 7. ve 8. soruları cevaplayınız.

Register	Value
%esp	0x800040
%ebp	0x800060

“proc” fonksiyonu satır 10’da “scanf” fonksiyonunu çağırıyor ve “scanf” girdi olarak 0x46 ve 0x53 değerlerini okuyor. “%x %x” dizgesinin 0x300070 hafıza adresinde bulunduğunu varsayın.

Soru 6. Üçüncü satırda %ebp yazmacına hangi değer atanır? Dördüncü satırda %esp yazmacına hangi değer atanır? (1 puan)

Çözüm:

Soru 7. Yerel deęişkenler olan x ve y deęişkenleri hangi adreslerde saklanmıřtır? (1 puan)

Çözüm:

Soru 8. “scanf” fonksiyonu döndükten sonra “proc” için olan yığıt görünümünü tamamlayınız ve yanına kısaca açıklayınız. Yığıtın “proc” tarafından kullanılmayan kesimlerini belirtiniz. (2 puan)

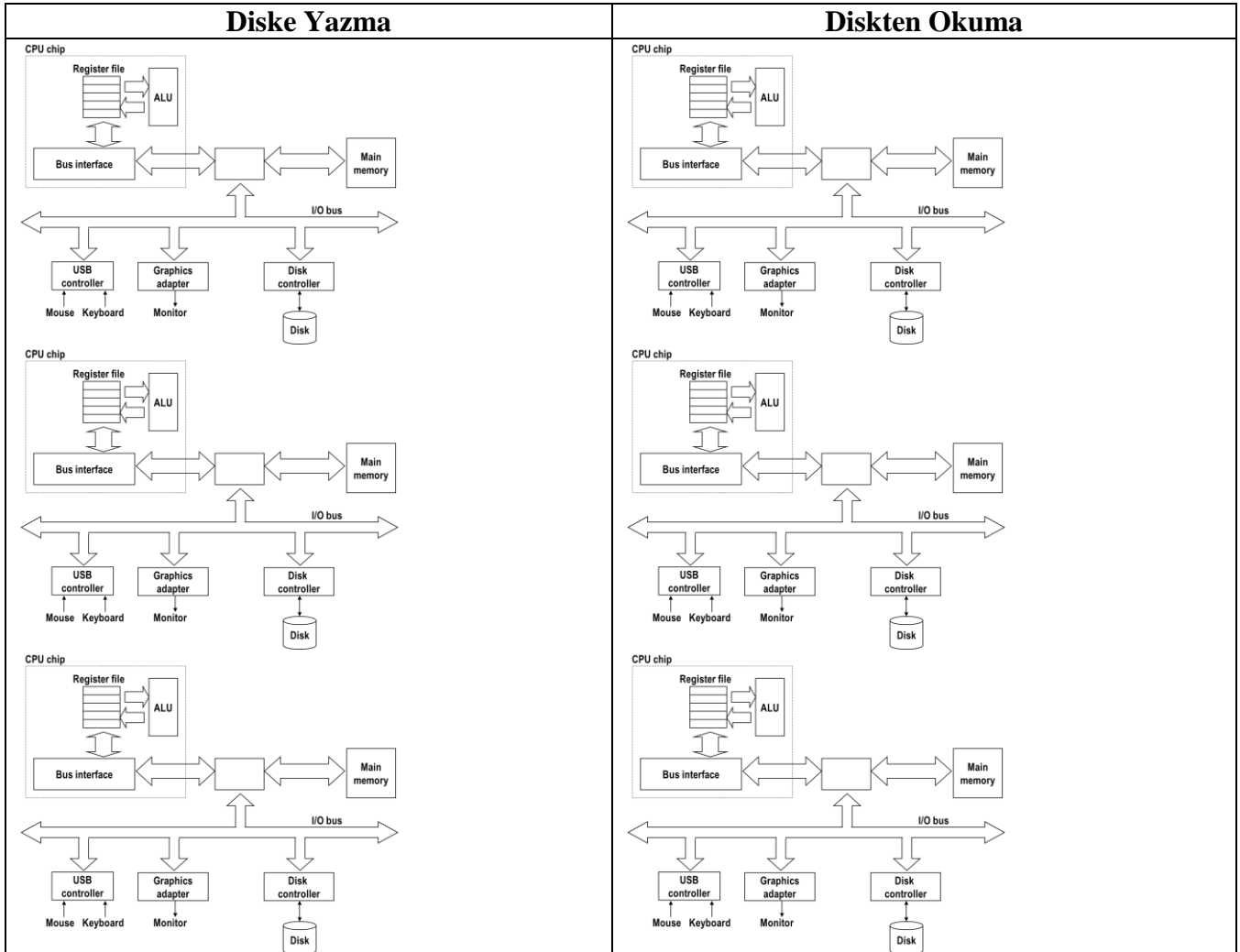
0x80003C	
0x800038	
0x800034	
0x800030	
0x80002C	
0x800028	
0x800024	
0x800020	
0x80001C	
0x800018	
0x800014	

Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
2. Ara sınav – 1 Aralık 2015

Öğrenci Adı:

Numarası:

Soru 1. (10 Puan) Doğrudan bellek erişim (DMA: *Direct Memory Access*) yöntemini kullanarak bir disk sektörünün erişimine ilişkin adımları aşağıdaki şekiller üzerinde gösteriniz.



Soru 2. (10 Puan) Zamansal Yerellik (*Temporal Locality*) kavramı aşağıdaki kod kesiminde hangi verilere erişim için söz konusudur, bir/iki cümle ile açıklayınız.

```
int sum_array_cols(int a[M][N])
{
    int i, j, sum = 0;
    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum += a[i][j];
    return sum;
}
```

.....

.....

.....

.....

Soru 3. (10 Puan) *Konumsal Yerellik (Spatial Locality)* kavramı aşağıdaki kod kesiminde hangi verilere erişim için söz konusudur, bir/iki cümle ile açıklayınız.

```
int sum_array_cols(int a[M][N])
{
    int i, j, sum = 0;
    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum += a[i][j];
    return sum;
}
```

Soru 4. (10 Puan) %99 “hit” oranıyla yapılan veri erişimleri %93 oranıyla yapılanlara göre kaç kat daha iyidir? Ön bellekten erişim zamanını 1 birim, ana bellekten erişim zamanını 50 birim alınız.

Soru 5. (20 Puan) Aşağıdaki C kaynak kodunda M ve N değişmezlerinin #define ile tanımlandığını varsayınız.

```
1  int mat1[M][N];
2  int mat2[N][M];
3
4  int sum_element(int i, int j) {
5      return mat1[i][j] + mat2[j][i];
6  }
```

Yukarıdaki C kaynak kodu derlendiğinde aşağıdaki liste elde edilmiştir. Buna göre tersine mühendislik yöntemini kullanarak M ve N değişmezlerinin değerini hesaplayınız.

```
    i at %ebp+8, j at %ebp+12
1  movl 8(%ebp), %ecx
2  movl 12(%ebp), %edx
3  leal 0(%ecx,4), %eax
4  subl %ecx, %eax
5  addl %edx, %eax
6  leal (%edx,%edx,8), %edx
7  addl %ecx, %edx
8  movl mat1(,%eax,4), %eax
9  addl mat2(,%edx,4), %eax
```

M =

N =

Soru 6. (20 Puan) Aşağıda bir önbelleğin içeriği gösterilmiştir. Yazan tüm sayılar onaltılık sistemdedir. Her adres bir bayt uzunluğundaki bellek konumuna karşılık gelmektedir (4 baytlık değil). Adresler 13 bit uzunluğundadır. Ön bellekte 8 tane küme (S=8) vardır. Blok uzunluğu 4 bayttır (B=4). Her kümede iki satır vardır (E=2).

2-way set associative cache												
Line 0							Line 1					
Set index	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3
0	09	1	86	30	3F	10	00	0	—	—	—	—
1	45	1	60	4F	E0	23	38	1	00	BC	0B	37
2	EB	0	—	—	—	—	0B	0	—	—	—	—
3	06	0	—	—	—	—	32	1	12	08	7B	AD
4	C7	1	06	78	07	C5	05	1	40	67	C2	3B
5	71	1	0B	DE	18	4B	6E	0	—	—	—	—
6	91	1	A0	B7	26	2D	F0	0	—	—	—	—
7	46	0	—	—	—	—	DE	1	12	C0	88	37

- A. Aşağıdaki şekil her bit bir kutucuğa gelecek şekilde adres formatını göstermektedir. Kutucukları hangi alanların hangi amaçla kullanıldığını gösterecek şekilde işaretleyin. Kutucuklara ön bellek blok ofseti için CO, ön bellek küme indeksi için CI, ön bellek etiketi (tag) için CT yazınız.

12	11	10	9	8	7	6	5	4	3	2	1	0

- B. Bir program 0x1239 adresindeki bayta erişmek istiyor. Adresi ikilik sistemde yazınız.

12	11	10	9	8	7	6	5	4	3	2	1	0

Aşağıdaki alanları buna göre doldurunuz:

Önbellek blok içi adresi (offset):

Önbellek küme indeksi:

Önbellek etiketi (tag):

Önbellek hit/miss?

Önbellekten dönen değer:

- C. Bir program 0x1E1A adresindeki bayta erişmek istiyor. Adresi ikilik sistemde yazınız.

12	11	10	9	8	7	6	5	4	3	2	1	0

Aşağıdaki alanları buna göre doldurunuz:

Önbellek blok içi adresi (offset):

Önbellek küme indeksi:

Önbellek etiketi (tag):

Önbellek hit/miss?

Önbellekten dönen değer:

Soru 7. (20 Puan) Aşağıdaki “struct” için bileşenlerin başlangıç adreslerini (offset) ve veri yapısının toplam uzunluğunu tabloya yazınız.

```
struct P4 { short w[4]; char *c[4] ; int i};
```

A. 32 bitlik mimari için:

w	c	i	Toplam uzunluk

B. 64 bitlik mimari için:

w	c	i	Toplam uzunluk

Hacettepe University
Computer Engineering Department
BBM341 Systems Programming
1st Midterm – 27 November 2018

Name :

ID No :

Question 1. (10 points) Place the value 0x10 into the memory starting from the address 0x100. (32-bit data)

Big Endian

```
0x100  0x101  0x102  0x103
```

--	--	--	--	--	--	--	--

Little Endian

```
0x100  0x101  0x102  0x103
```

--	--	--	--	--	--	--	--

Question 2. (10 points) Apply shift operations in the following table.

Shift Operation	Arithmetic/Logical	Value (8 bits)	Result (8 bits)
Shift left 4 positions	Arithmetic	10101111	
Shift left 4 positions	Logical	10101111	
Shift right 4 positions	Arithmetic	10101111	
Shift right 4 positions	Logical	10101111	

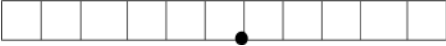
Question 3. (10 points) For the unsigned/signed 8 bit integers, perform the operations below.


Signed	Unsigned	Signed	Unsigned
$\begin{array}{r} 11101111 \\ + 11101111 \\ \hline \end{array}$	$\begin{array}{r} 11101111 \\ + 11101111 \\ \hline \end{array}$	$\begin{array}{r} 11101111 \\ - 11101111 \\ \hline \end{array}$	$\begin{array}{r} 11101111 \\ - 11101111 \\ \hline \end{array}$

Question 4. (10 points) Encode the base 10 values in base 2.

10.25 = 

10.50 = 

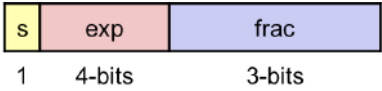
10.15 = 

10.30 = 

```
leal (%eax,%eax,2), %eax
sall $2, %eax
```

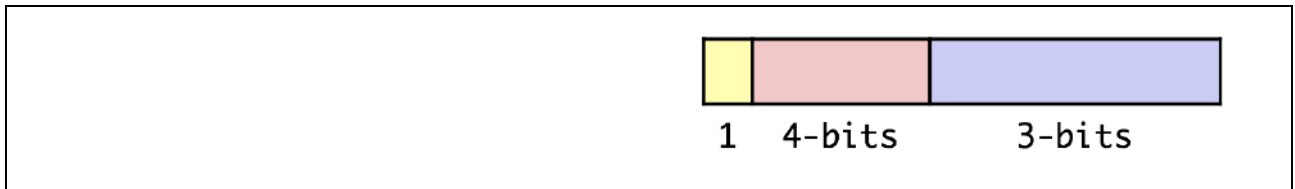
```
t <- x+x*2
return t << 2;
```

Question 5. (10 points) Write the code that multiplies %eax by 25. Use only “leal” and/or “shift” instructions.

Normalized Values Condition: $\text{exp} \neq 000 \dots 0$ and $\text{exp} \neq 111 \dots 1$ $E = \text{exp} - \text{Bias}$	Denormalized Values Condition: $\text{exp} = 000 \dots 0$ $E = -\text{Bias} + 1$	Bias = $2^{k-1} - 1$	
---	---	-----------------------------	---

Question 6. (10 points) Based on the above information encode 0.1_{10} as floating point number.

Solution:



Question 7. (10 points) Based on the above information encode 0.00000001_{10} as floating point number.

Solution:



Question 8. (10 points) Perform the rounding operations.

	\$0.80	\$0.50	-\$0.50
Towards zero			
Round down ($-\infty$)			
Round up ($+\infty$)			
Nearest Even (default)			

Question 9. (10 points) Fill in the table with proper addressing mode information (“Imm”, “Reg” or “Mem”).

	Source	Destination
movl (%eax), %ecx		
movl (%eax), \$0xf		
movl \$0x30, %eax		
movl %eax, 0xf		
movl \$147, (%eax)		

Question 10. (10 points) For the following C function, give the assembly listing for the 32-bit architecture, and draw the stack view while the “if statement” is being evaluated. Do not forget to depict where %esp and %ebp are pointing.

```
void compare(int x, int y)
{
    if(x > y)
        return x;
    else
        return y+1;
}
```

Hacettepe University
Computer Engineering Department
BBM341 Systems Programming
2nd Midterm – 18 December 2018

Name :

ID No :

Section :

Question 1. (20 points) Assume the following:

- The memory is byte addressable.
- Memory accesses are to **1-byte words** (not to 4-byte words).
- Addresses are 13 bits wide.
- The cache is two-way set associative ($E = 2$), with a 4-byte block size ($B = 4$) and eight sets ($S = 8$).

A. The contents of the cache are as follows, with all numbers given in hexadecimal notation.

2-way set associative cache

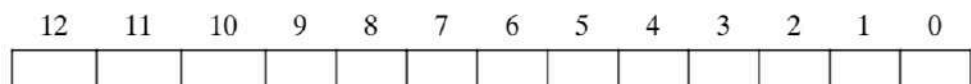
Set index	Line 0						Line 1					
	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3
0	09	1	86	30	3F	10	00	0	—	—	—	—
1	45	1	60	4F	E0	23	38	1	00	BC	0B	37
2	EB	0	—	—	—	—	0B	0	—	—	—	—
3	06	0	—	—	—	—	32	1	12	08	7B	AD
4	C7	1	06	78	07	C5	05	1	40	67	C2	3B
5	71	1	0B	DE	18	4B	6E	0	—	—	—	—
6	91	1	A0	B7	26	2D	F0	0	—	—	—	—
7	46	0	—	—	—	—	DE	1	12	C0	88	37

The following figure shows the format of an address (one bit per box). Indicate (by labeling the diagram) the fields that would be used to determine the following:

CO The cache block offset

CI The cache set index

CT The cache tag



Suppose a program running on the machine references the byte at address 0x0E38. Indicate the cache entry accessed and the cache byte value returned **in hex**. Indicate whether a cache miss occurs. If there is a cache miss, enter “–” for “Cache byte returned.”

B. Address format (one bit per box):

12	11	10	9	8	7	6	5	4	3	2	1	0

C. Memory reference:

Parameter	Value
Cache block offset (CO)	0x_____
Cache set index (CI)	0x_____
Cache tag (CT)	0x_____
Cache hit? (Y/N)	_____
Cache byte returned	0x_____

Question 2. (10 points) List the differences between DRAM and SRAM.

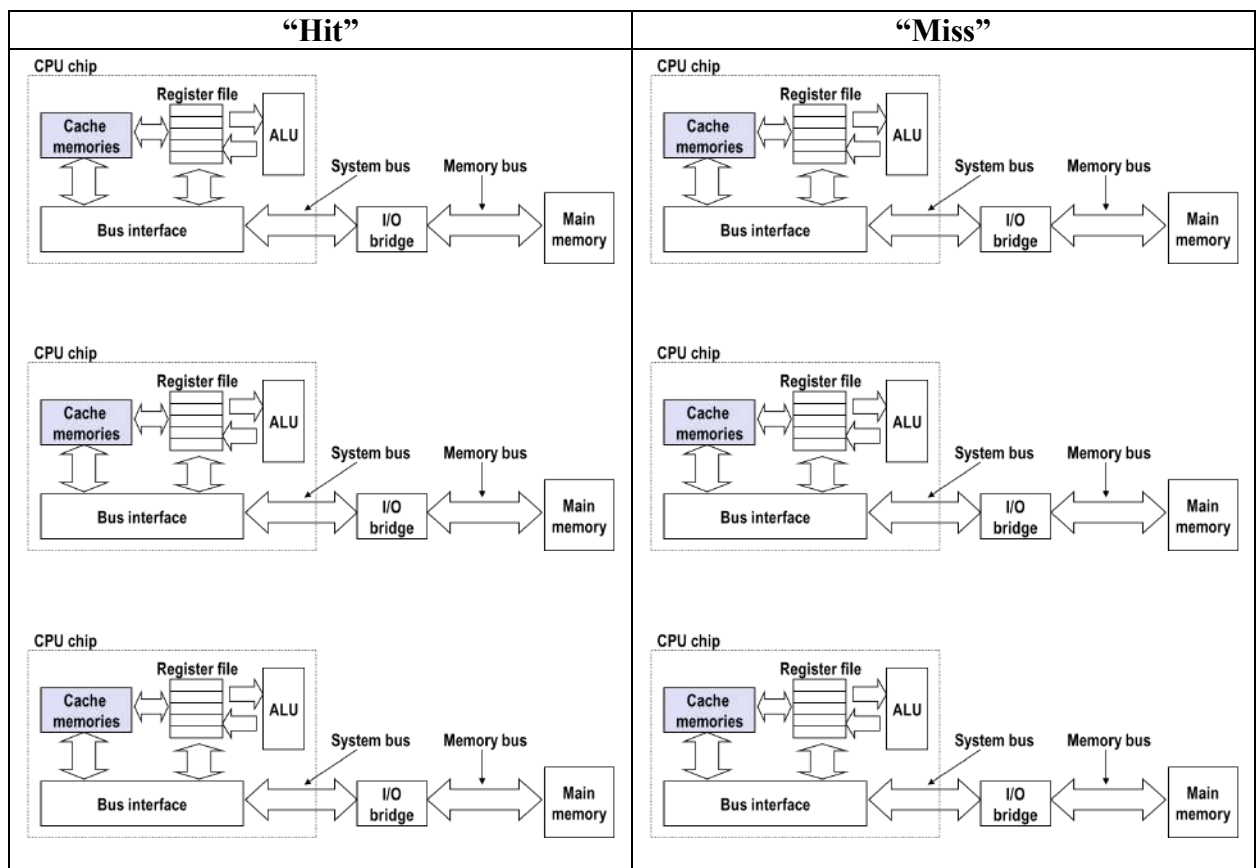
Question 3. (10 points) If the hit rate increases from 95% to 100%, how much improvement can be observed in the average access time? Take cache hit time of 1 cycle and miss penalty of 100 cycles.

Question 4. (20 points) For the following matrix multiplication algorithm, explain how the algorithm benefits from Spatial and Temporal Localities. (For both instruction and data references)

```
float A[100][100], B[100][100], C[100][100];
int i, j, k, m, n, p;
...
for (i = 0; i < m; i++)
    for (j = 0; j < n; j++)
        for (k = 0; k < p; k++)
            C[i][j] += A[i][k] * B[k][j];
...
```



Question 5. (10 points) For `movl (%edx), %ebx` instruction, visualize data access steps for both *miss* and *hit* cases. Use arrows to depict dataflow, addressing, entry checks etc., and use keywords such as **check**, **address**, **read** and/or **write** to identify each step.



Question 6. (20 points) Using the following information fill in the content and addressing mode columns in the table below.

<u>Address</u>	<u>Content</u>	<u>Register</u>	<u>Content</u>
0x100	0xFF	%eax	0x100
0x104	0xAB	%ecx	0x1
0x108	0x13	%edx	0x3
0x10C	0x11		

<u>Operand</u>	<u>Content</u>	<u>Addressing Mode</u>
%eax
0x104
\$0x108
(%eax)
4(%eax)
9(%eax,%edx)
260(%ecx,%edx)
0xFC(,%ecx,4)
(%eax,%edx,4)

Question 7. (10 points) List the differences between SSD and Rotating Disks.