

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

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

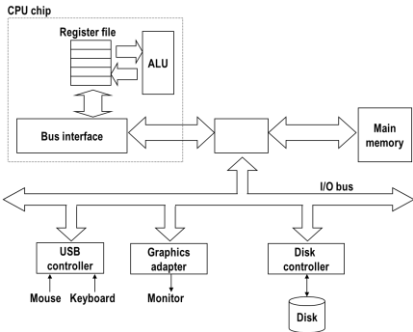
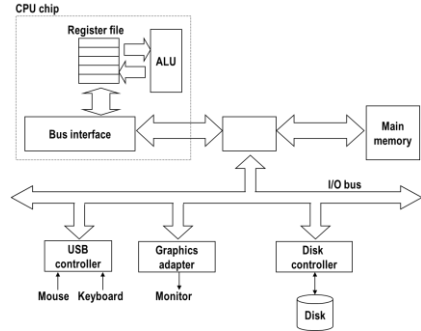
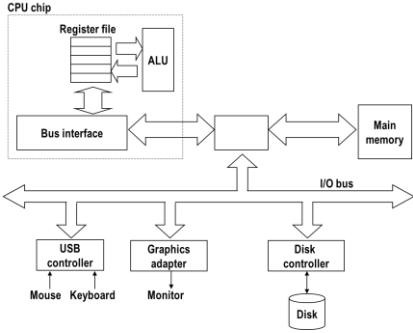
	Address
123	0x124
456	0x120
	0x11c
	0x118
	0x114
0x120	0x110
0x124	0x10c
Rtn adr	0x108
	0x104
	0x100

```

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.

.....

.....

.....

.....

.....

.....