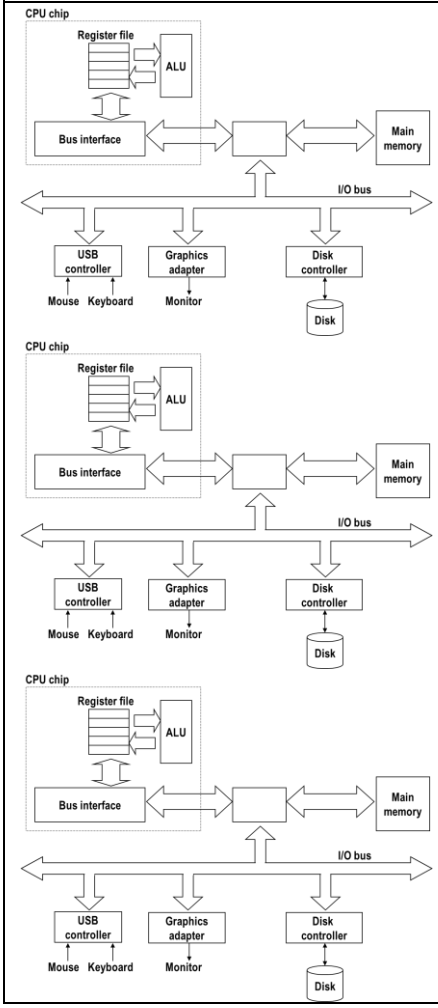
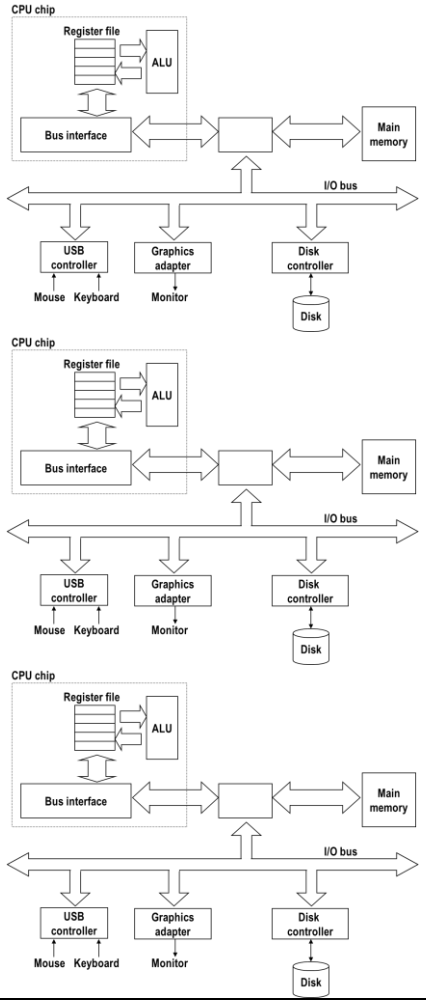
Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BBM341 Sistem Programlama
2. Ara sınav – 24 Aralık 2014

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

Diske Yazma	Diskten Okuma
	

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);
}
```