

İŞLETİM SİSTEMLERİNE GİRİŞ - 2

Kaynakların Paylaşımı (Resource Sharing)

Sistem, sistem kaynaklarını belli bir hiyerarşi içinde kullanıcının hizmetine sunar. Bir işletim sisteminde paylaşılan kaynaklar ise şunlardır: Merkezi işlem birimi, Bellek, Giriş/Çıkış birimleri gibi donanım elemanları, Dosyalar ve klasörler, yazılımlar; uygulama programları, çeşitli programlar.

Kaynak paylaşımı yapılmasının nedenleri 4 faktör söylenebilir:

- Maliyet: Her kullanıcıya birbirinden bağımsız kaynak sağlamak zordur.
- Birinin geliştirdiği bir programı veya uygulamayı diğerleri de kullanabilir
- Aynı veri tabanı birden fazla kullanıcı tarafından kullanılabilir
- Bir programın birden fazla kullanıcı tarafından kullanılarak depolama birimlerinden tasarruf sağlanır ve geçersiz kaynak kullanımının önüne geçilebilir

İşletim sisteminin paylaşım işlemini gerçekleştirebilmek için ise şu görevleri yerine getirmesi gerekir:

- a. Giriş/Çıkış İşlemleri: Kullanıcıdan bağımsız donanıma bağlı değişen G/Ç işlemlerini yerine getirme.
- b. Bellek İşlemleri: Makinedeki fiziksel bellekten farklı olarak kullanılan soyut bir bellek imkanı sunmalıdır.
- c. Dosya Sistemi: Soyut makinelerin çoğu program ve verilerin saklanması için bir dosya sistemi içerir. Kullanıcının bu bilgilere ulaşımı, adresler yerine sembolik isimlerle sağlanır.
- d. Koruma ve hata kontrolü: Kullanıcıların birbirlerinin alanlarına müdahale etmeleri önlenmeli ve kişisel bilgilerin güvenli bir şekilde saklanması için koruma sağlanmalıdır.

- e. Program kontrolü: Soyut bir makine kullanıcıya program ve işlemler üzerinde işlem yapmaya izin verir.
- f. Etkileşim: İşletim sistemi kullanıcılar arasında karşılıklı etkileşim olanağı verir.

Kaynakların farklı kullanıcılar veya programlar arasında paylaşılması farklı teknikler kullanılarak yapılabilir;

- Aynı anda çalışma (Concurrent execution): Bir bilgisayarın birden fazla programı aynı anda çalıştırmasıdır. Sistemde aynı anda farklı aktivitelerin yer almasıdır. Mesela birden fazla kullanıcı G/Ç işlemi yapabilmeli belleği kullanabilmelidir. Bu durumda işlerin senkronize edilmesi gerekir. Çünkü aslında sistemde gerçekleştirilecek işlemler aynı anda yapılmamakta, sıralı olarak yapılmaktadır. Yani birden fazla program mantıksal olarak aynı anda çalışmakta iken fiziksel olarak sıralı olarak çalışmaktadır. Örnek olarak çoklu programlama yaparken MİB'ni programların paylaşması verilebilir.
- Paralel çalışma (Paralel execution): Bir bilgisayarın birden fazla programı gerçekten aynı anda çalıştırmasıdır. Yani birden fazla program hem mantıksal hem de fiziksel olarak aynı anda çalışmaktadır.

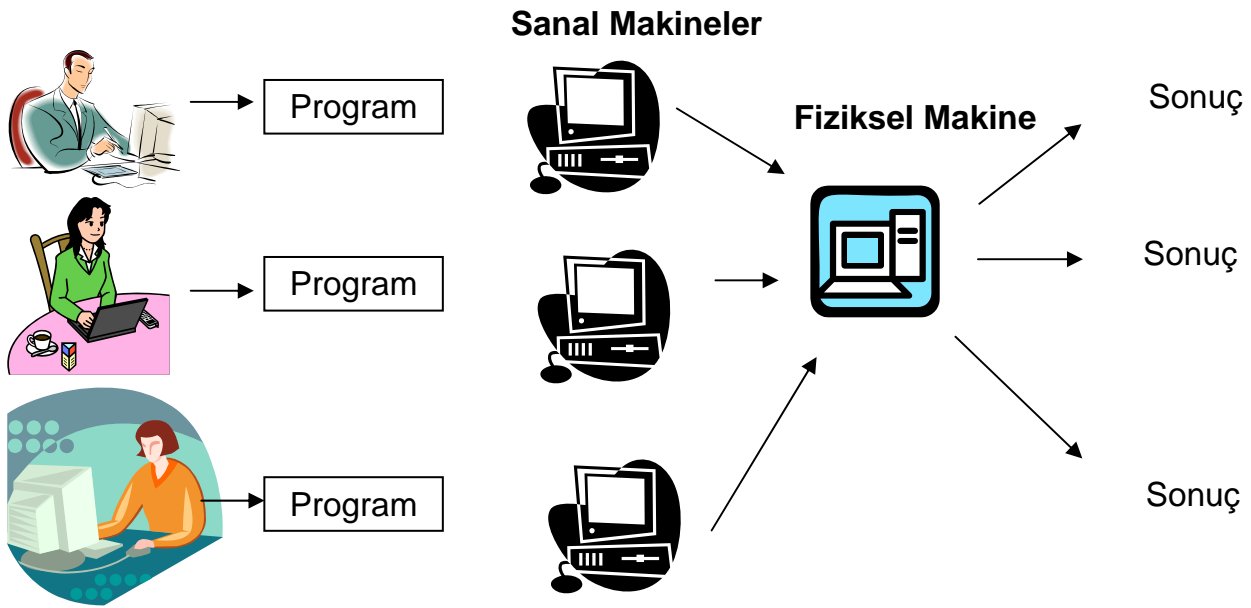
Aynı anda veya paralel çalışma sırasından bir bilgisayar kaynaklarını paylaştırmak zorundadır. Bu paylaşımlar iki türlü olabilir;

- Saydam paylaşım (Transparently sharing): Kaynakların paylaşımının işletim sistemi tarafından yürütmesidir. Burada kullanıcı kaynakların paylaştırıldığından habersizdir. Bu paylaşım türünün gerçekleştirilebilmesi için soyut makineler kullanılmaktadır.

- Açık paylaşım (Explicit sharing): İşlemlerin genel kaynakları kendi politikalarına göre kullanmalarıdır. İşletim sistemini kullanıcıya makinedeki kaynakları paylaşmasına da izin vermektedir.

Soyut/Sanal bir makine (Abstract Machine) ve Saydam Paylaşım

İşletim sisteminin ikinci görevi ise kullanıcıya veya kullanıcılara yalnızca kendine tahsis edilmiş bir makine varmış gibi çalışma ortamının sağlanmasıdır. Her bir soyut makine fiziksel makinenin bir simülasyonudur. Bir fiziksel makine birden fazla soyut makineyi aynı anda çalıştırabilir. Her program kendi soyut makinesinde çalışır. Bu işlemleri yapabilmek için fiziksel makine donanımlarını saydam paylaşım tekniğini kullanarak soyut makineler arasında paylaşır. Soyut makine tarafından çalıştırılan programa genellikle 'process (işlem)' denilmektedir.



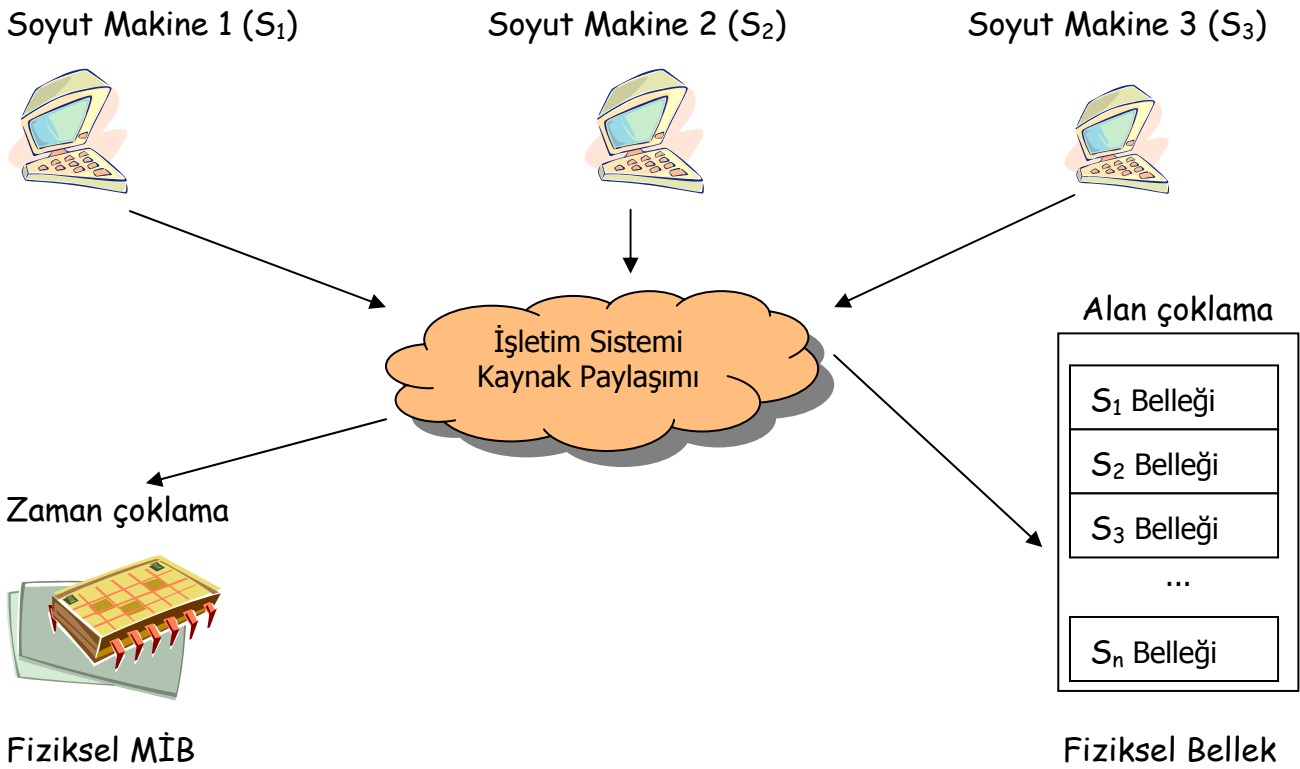
Soyut makinelerin oluşturulması için saydam paylaşım tekniğinin 2 tür paylaşımı kullanılmaktadır;

- a. Alan çokluma paylaşımı (Space-multiplexed sharing): Bir kaynak bir veya daha fazla bölüme ayrılır, ardından her bir bölüm bir işlem'e atanır. Bu

paylaşım türüne, bellek ve hard disk'in belli bölümlere ayrılarak her bölümün farklı işlemler için ayrılması örnek olarak verilebilir.

- b. Zaman çoklama paylaşımı (Time-multiplexed sharing): Bu paylaşım türünde bir kaynak bölümlere ayrılmaz, bunun yerine kaynak bir işlem tarafından belli bir süre kullanılır yani o işleme adanır ardından diğer işlem bu kaynağı belli bir süre kullanır. Örneğin MİB'ni işlemlerin belirli süreler için kullanılması bu duruma bir örnektir.

İşletim istemi bu iki paylaşım türünü beraber kullanabilir. Örneğin MİB'i için zaman çoklama, hard disk ve bellek için ise alan çoklama paylaşım türlerini kullanabilir. Bu paylaşımlar sayesinde aynı andan birden çok program çalışabilmektedir kısacası çoklu programlama özelliği kullanılabilir. N tane soyut makine olduğunu düşünürsek;



Şekil 2. Çoklu programlama (Multiprogramming)

Çoklu programlama bir işlemcinin dolayısı ile bir bilgisayar sistemin performansını arttırır.

Açık paylaşım (Explicit sharing)

İşlemlerin genel kaynakları kendi politikalarına göre kullanmalarıdır. Bu paylaşımında zaman veya alan çoklama paylaşımı yapılırsa da dikkat edilmesi gereken 2 önemli nokta vardır;

- **Kaynak yalıtımı (Resource isolation):** Sistem yerleşim politikasına göre kaynaklara ulaşımı ayırabilmelidir. İşletim sistemi, bir soyut makine tarafından kullanılan kaynağa diğer yetkisiz işlemlerin ulaşmasını engellemelidir. Örneğin bellek yalıtım mekanizmasında, bellek belli bölümlere ayrılarak her bir bölüm ayrı bir soyut makine tarafından kullanılmaktadır. Bu durumda sistem, bir soyut makineye ayrılmış bir bellek bölümüne başka bir soyut makinenin müdahale etmesini engellemelidir. Aynı şekilde MİB'nin işlemler tarafından sıralı olarak paylaşılmasında da, bir işlem MİB'ni kullanırken diğer işlemlerin bellek alanlarına MİB yoluyla müdahale etmemelidir.
- **İşbirliği yaparak paylaşım (Cooperatively sharing):** Sistem işlemlerin kaynakları istenildiği takdirde işbirliği yaparak paylaşmalarına izin verebilmelidir. Örneğin bir işlem diğer bir işlemin sonucunu kullanabilmeli veya belirli bir bellek bölgesindeki bilgiler iki işlem tarafından kullanılması gerektiğinde buna izin verebilmelidir. Fakat bu paylaşım son derece dikkatli yapılması gerekmektedir.

Kaynak paylaşımını yapmak ile görevli işletim sistemi yazılımı bu paylaşımı algoritma veya başka bir hata (bug) yüzünden gerçekleştiremediğinde ise hatalar oluşmaktadır. Oysa güvenilir ve gelişimi tamamlanmış bir işletim sisteminde bu tür hataların olması beklenmez. Güvenilir bir işletim sisteminde donanım kaynaklarının doğru bir biçimde paylaşılması ve birbirinden yalıtılması gerekmektedir. Bir işletim sisteminin soyutlamaları, sistem çağrı arayüzü (system call interface) olarak da bilinen

işletim sisteminin arayüzü kullanılarak gerçekleştirilebilir. Tüm sistem yazılımları bir uygulama programlama arayüzü - API (Application Programming Interface) yoluyla ulaşılabilir. API; bir yazılım veya sistem yazılım parçasının programlama arayüzüdür. Veritabanları, VBasic editörü gibi. Programcılar uygulama programlama arayüzlerini kullanırlarken, işletim sistemi, sistem çağrı arayüzünü kullanır. Uygulama programlama arayüzleri sistem yazılım arayüzlerine ulaşarak bunların kullanıcı tarafından kullanılabilmesini sağlar. Microsoft Windows sistem çağrı arayüzünü "Win32 API" olarak adlandırmıştır.

Uygulama yazılımı, sistem yazılımı ve işletim sistemi arasında bir hiyerarşi vardır. İşletim sistemi, yazılım-donanım arayüzünü kullanarak işletim sistemi arayüzüne, sistem yazılımı, işletim sistemi arayüzünü kullanarak API'ye ve uygulama yazılımı da API'yi kullanarak insan-bilgisayar arayüzünde gerçekleşecek olan yazılımı oluşturur.

İnsan-Bilgisayar Arayüzü

(Human-Computer Interface)



API

Uygulama Yazılımı

İşletim Sistemi Arayüzü

Sistem Yazılımı
(Soyut Kaynaklar)

Yazılım-Donanım Arayüzü

Güvenilir İşletim
Sistemi (Soyut
Kaynaklar)

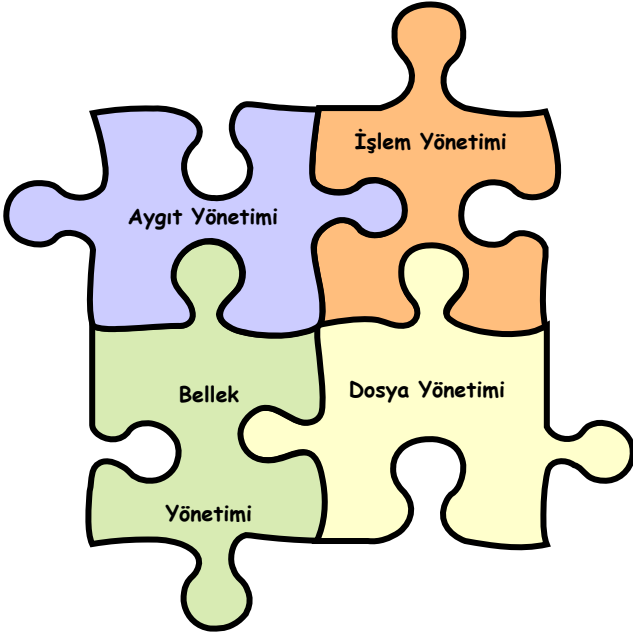
Donanım Kaynakları



Şekil 3. Uygulama yazılımı, Sistem yazılımı ve İşletim Sistemi

Bir İşletim Sisteminin Fonksiyonları Açısından Mantıksal Yapısı

Bir işletim sisteminin, temel olarak 4 bileşeni bulunmaktadır;

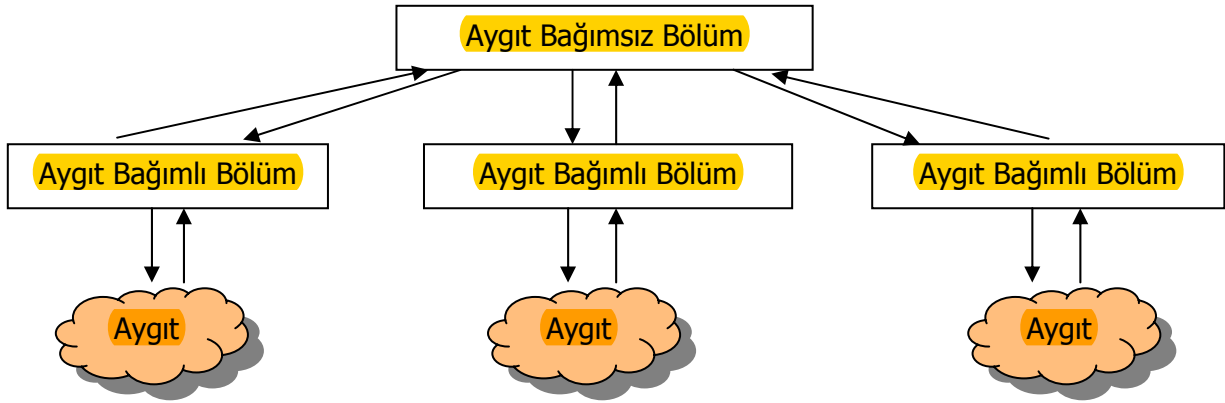


- Aygıt yönetimi (Device management)
- İşlem yönetimi (Process management)
- Bellek yönetimi (Memory management)
- Dosya yönetimi (File management)

Aygıt yönetimi (Device management)

Bir işletim sistemi donanım aygıtlarının yönetiminden sorumludur. Birçok işletim sistemi yazıcı, disk gibi donanım aygıtlarına genelde aynı şekilde yönetirken işlemci ve belleğin yönetiminde farklı yaklaşımlar kullanmaktadır.

Aygıt yönetiminin aygıt bağımlı ve bağımsız olmak üzere iki bölümü bulunmaktadır. Bağımlı olan bölüm'e aygıt sürücüsü (device driver) de denilmektedir. İşletim sisteminin her bir aygıt için kullandığı ayrı bir sürücü vardır. Aygıt yönetiminin bağımsız olan bölümü ise aygıt bağımlı bölümün yürüteceği yazılım ortamını temsil etmektedir. Örneğin aygıt bağımsız alan, sistem çağrı arayüzündeki çağrıları aygıt sürücüsüne iletmektedir. Aygıt bağımsız alan genellikle aygıt yönetiminin küçük bir bölümüdür, büyük bölümü sürücülere ayrılmıştır.



Şekil 4. Aygıt yönetimi

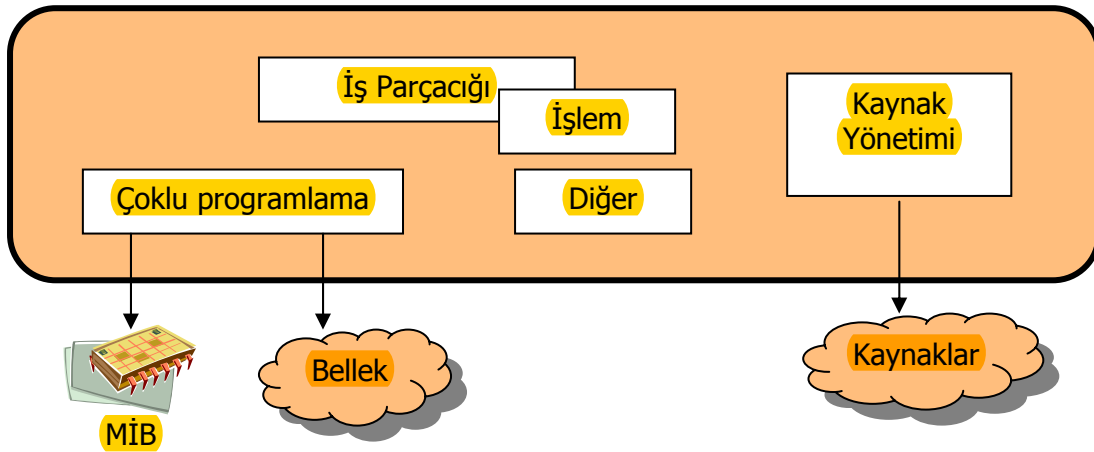
Aygıt yönetiminin bu şekilde ikiye ayrılması ile bilgisayara yeni bir donanım eklemek çok kolay hale gelmiştir. Öncelikle işletim sistemi tasarımcısı, aygıtın hangi bölümünün bağımlı ve hangisinin bağımsız olacağını belirler. Bağımsız olan bölüm temel işletim sistemi içerisinde, bağımlı olan bölüm ise aygıt sürücüsü içerisinde uygulamaya geçirilir. Bu; aygıt yönetiminin bağımsız bölümünün; bir aygıtta okuma ve/veya yazma işlemlerini yürüten sistem çağrılarını içerdiği anlamına gelmektedir. Örneğin yazıcı sürücüsü bir yazıcı ile ilgili tüm yazılımları içermektedir. Bu yazıcı bilgisayara bağlanarak sürücüsü yüklendiğinde aygıt yönetiminin bağımsız olan bölümü işletim sisteminin içerisinde hali hazırda bulunduğu için bu yazıcı kullanıcı tarafından hemen kullanılabilir.

İşlem Yönetimi (Process Management)

Birçok işletim sistemi işlem ve iş parçacığı (Thread) ve kaynak yönetimini birlikte ele almaktadır. İş parçacığı, bilgisayarda en düşük kaynağa ihtiyaç duyan bir program parçasıdır. Genellikle bir işlem ile birlikte kullanılır. İlgili işleme ayrılmış disk alanı, dosyalar, bellek gibi kaynaklarını kullanarak çalışır. Çoklu kullanım

(multithreading) ise bir işlemin birden fazla iş parçacığına bölünerek aynı anda çalıştırılmasıdır. Bir işlem birden fazla iş parçacığından oluşmaktadır.

Bir işletim sisteminin işlem yönetimi; birden fazla işlem ve iş parçasına aynı makinenin kaynaklarını paylaşmaları, işlemlerin eşzamanlı olarak çalışabilmeleri için zamanlamanın sağlanması gibi görevleri vardır. İşlem yönetimi; işlemlerin kaynaklara ulaşması sırasında nasıl bir kaynak yalıtımı yapacağı, bir kaynağı paylaşması gereken birden fazla işlem olduğunda hangi politikaları kullanarak bu kaynağı paylaşacağını gibi soruları cevaplamaya çalışmaktadır. Bunları yaparken de bellek yönetimi ile birlikte çalışarak belleğin bu işlemler, iş parçacıkları arasında paylaşılmasını sağlar.

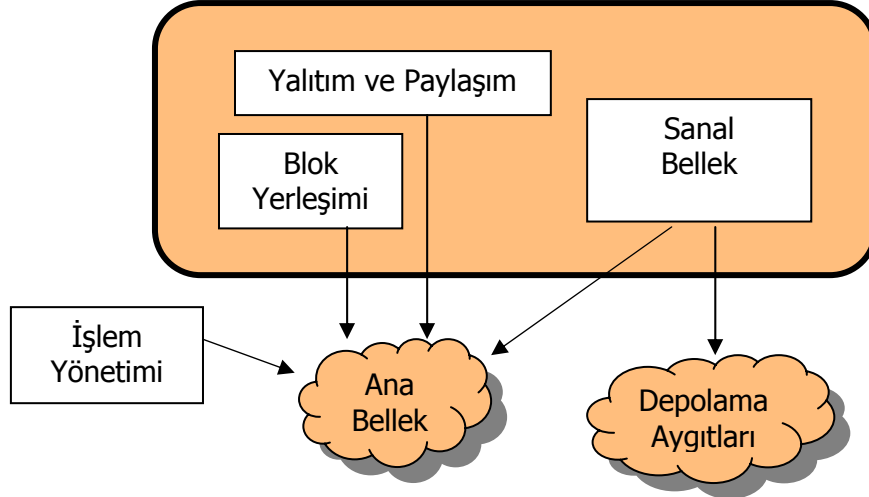


Şekil 5. İşlem yönetimi

Bellek Yönetimi (Memory Management)

Bellek yönetimi işlem yönetimi ile birlikte çalışarak ana bellekte işlemlerin yerleşimini sağlamaktadır. Her işlem bir bellek bölgesi istemekte ve bellek yönetimi de bu işlemlerin çalışması için kaynak yalıtımını da sağlayarak gerekli bellek bölümünü ayırmaktadır. Böylece bellek yönetimi bellekteki blokların paylaşılmasını için gerekli

stratejileri uygulamaktadır. Modern bellek yönetimleri sanal bellek (virtual memory) sağlayarak fiziksel bellekten çok daha büyük bir bellek alanının kullanılmasını sağlamaktadır. Eğer işletim sistemi sanal belleği destekliyorsa bellek yönetiminin bir kısmı aygıt ve dosya yönetimleri ile birlikte çalışarak belleği yönetir.



Şekil 6. Bellek yönetimi

Dosya Yönetimi (File Management)

Dosya yönetimi, bellek ve aygıt yönetimi ile birlikte çalışarak dosyaların hard disk ve CD-ROM gibi depolama birimlerine yazılmasını sağlamaktadır. İşletim sistemi bu yönetimi yapabilmek için dosya sistemlerini kullanmaktadır. Bu dosya sistemlerine örnek olarak FAT, FAT32, NTFS, EXT2 verilebilir.