

11. Hafta: İŞLETİM SİSTEMİ ÇEŞİTLERİ

Bu haftayı çalıştıktan sonra;

- İşletim sistemi nedir? Ne işe yarar ?
- İşletim sisteminin genel olarak yapısı ve bileşenleri
- İşletim sistemi çeşitleri nelerdir ?
- En yaygın kullanılan işletim sistemleri nelerdir?

Konularında bilgi sahibi olacaksınız.

11.1 İşletim sistemi nedir? Ne işe yarar ?

İşletim Sistemi (OS), bilgisayar kullanıcısı ile bilgisayar donanımı arasındaki bir arayüzdür.

İşletim sistemi; dosya yönetimi, bellek yönetimi, süreç yönetimi, giriş ve çıkışı işlemleri ve disk sürücüler ve yazıcılar gibi çevresel aygıtları kontrol etme gibi tüm temel görevleri gerçekleştiren bir yazılımdır.

11.2 İşletim Sistemlerinin Genel Yapısı ve Bileşenleri

Piyasada birbirinden farklı kesimlere hitap eden çok çeşitli işletim sistemleri bulunmasına rağmen, bir işletim sistemi kavramsal olarak üç grupta toplanabilecek bileşenlerden oluşur:

- * Kullanıcı arayüzü (grafik kullanıcı arayüzü ve/veya komut satırı yorumlayıcısı [“kabuk” da denir])
- * Çekirdek
- * Dosya yönetim sistemi

Bunlardan kısaca bahsetmek gerekirse,

Kabuk (Shell): İşletim sisteminin kullanıcıya gözüken yüzüdür. Komut yorumlayıcısı olarak da adlandırılmaktadır. Kullanıcı ile çekirdek arasındaki arabirim olarak da tanımlanabilir. Bilgisayar açıldıktan bir süre sonra komut satırı (prompt) görüntülenir. Kullanıcı tarafından komut satırına girilen komutlar bilgisayar tarafından işleme konulur. Bu noktada kabuk (Shell) olarak adlandırılan yazılım devreye girer. Öncelikle komutun geçerliliğini inceler, kullanıcının ne yapmak istediğini çözümler ve bu iş için gerekli olan programları belleğe yükler. Macintosh bilgisayarların çıkışı ve ardından Microsoft Windows yazılımının geliştirilmesinden beri insanlar

arayüz denildiğinde otomatik olarak grafiksel kullanıcı arayüzünü (GUI – grafik kullanıcı arayüzünü) düşünmektedir.

Çekirdek (Kernel): İşletim sisteminin kalbidir. Uygulamalar ve donanım seviyesindeki bilgi işlemleri arasında bir köprü görevi görür. Çekirdeğin görevleri sistemin kaynaklarını yönetmeyi de kapsamaktadır. Genellikle çekirdek, işletim sisteminin temel bir elemanı olarak, yazılımın fonksiyonunu yerine getirebilmesi için kontrol etmesi gereken kaynaklar için düşük seviye soyutlama katmanı sağlayabilir. İşletim sistemi görevleri, tasarımları ve uygulanmalarına göre farklı çekirdekler tarafından farklı şekillerde yapılır. Sistem açılırken belleğe yüklenir ve sistem kapatılıncaya kadar ana bellekte kalır. Çekirdek, ayrıcalıklı bir işlem seviyesi olan **Kernel Mode**'da çalışır. Bu işlem seviyesinde çalışan kod, donanıma ve sistemin tamamına tam yetki ile müdahale edebilir. Uygulama programları ise, **User Mode** denilen daha kısıtlı bir işlem seviyesinde çalışır. User mode'da çalışan uygulamalar, çekirdeğin ya da başka uygulamaların bellek bölgelerine erişemezler, bazı kısıtlanmış işlemci komutlarını çalıştıramazlar ve donanım kaynaklarına doğrudan müdahale edemezler.

Çekirdeğin en önemli fonksiyonları:

- Belleği yönetmek
- Bilgisayara ulaşimleri kontrol etmek
- Dosya sistemini oluşturup, korumak
- İnterruptları (kesme) kullanmak
- Hataları kontrol etmek
- Girdi-çıkı birimlerini çalıştırmak
- Bilgisayarın kaynaklarını (işlemci, girdi-çıkı birimleri gibi...) kullanıcılar arasında dağıtmak.

Birçok kaynakta işletim sisteminin çekirdek olduğu ileri sürülmektedir. Bunun nedeni, işlevselliği bakımından en önemli işletim sistemi ögesi olmasıdır.

İşletim sistemleri ilk bilgisayar neslinden itibaren kullanılmaktadır ve zamanla gelişmeye devam etmektedir. İşletim sistemlerinin genel yapısı ve bileşenleri özetlendikten sonra bu bölümde işletim sistemi çeşitleri ve geçmişten günümüze piyasada kullanılan işletim sistemleri anlatılacaktır.

11.2. İşletim Sistemi Çeşitleri

11.2.1. Eski Sistemler (Mainframe)

- a) 1950'lerdeki sistemlerdir. Tek kullanıcılarıdır. İşletim sistemi kullanılmamıştır. Örn: Eniac da işletim sistemi yoktu
- b) Sadece tek bir iş: Aynı anda sadece bir iş yapılmaktaydı. Kısacası doğrusal çalışıyorlardı.
- c) Bilgi girişi ve çıkışı için kartlar kullanılmaktaydı.

11.3.2. Toplu İşlem İşletim Sistemleri (Batch Operating Systems):

Toplu işlem işletim sisteminin kullanıcıları, bilgisayarla doğrudan etkileşime giremezlerdi. Her kullanıcı işini delikli kartlar gibi çevrimdışı bir aygıtta hazırlar ve bilgisayar operatörüne sunardı. İşlemeyi hızlandırmak için, benzer ihtiyaçlara sahip işler bir araya toplanır ve grup halinde çalıştırılırdı. Programcılar programlarını operatöre bırakır ve operatör de benzer gereksinimleri olan programları gruplar halinde sıralardı.

- a) Monitor programları geliştirilerek otomatik olarak birden fazla işin arka arkaya yapılması sağlandı. Bir işe örnek olarak “şu programı çalıştır” veya “şu programı derle” komutları verilebilir.
- b) Kart okuyucular kullanılmaktaydı.

Sistemin dezavantajları:

- Kullanıcı ve iş arasında etkileşim eksikliği.
- Mekanik I / O cihazlarının hızı CPU'dan daha yavaş olduğu için CPU genellikle boştaydı.
- İşler için istenilen önceliği sağlamak zordu.

11.3.3. Zaman paylaşımli sistemler (Timesharing Systems)

Zaman paylaşımı, çeşitli terminallerde bulunan birçok kişinin aynı anda belirli bir bilgisayar sistemini kullanmasını sağlayan bir tekniktir. Zaman paylaşımı (Time sharing) veya çoklu görev (multi tasking), çoklu programlamanın (multi programming) mantıksal bir uzantısıdır. İşlemcinin aynı anda birden fazla kullanıcı arasında zamanı paylaşması, zaman paylaşımı (time sharing) olarak adlandırılır.

- a) Diskteki veya bellekteki farklı işleri aynı anda yapabilmektedir.
- b) Bu işleri yaparken bir donanımı belirli zamanlarda farklı işleri yapılması için tahsis ederdi. Örneği MİB farklı işleri aynı anda yapmıyor ama bu işlerin belli görevlerini yerine getiriyordu. Örneğin aynı anda 3 iş yapması gerekiyorsa öncelikle 1. işin ilk görevini sonra 2. işin ilk görevini ve 3. işin ilk görevini yapıyordu ve bu şekilde devam ediyordu. Kullanıcı bu işleri aynı anda yaptığını zannetse de aslında bu işlerin belli görevleri arasında değişim yapıyordu. İşte bu özelliğe ‘Çok görevli (Multitasking)’ denilmektedir. “Çoklu programlama (multiprogramming)” sistemleri ise aynı anda birden fazla programın çalıştırılması anlamına gelmektedir. Bu sistemlerde MİB zamanlaması (scheduling), bellek yönetimi gibi işlemler sistem tarafından yapılmaktaydı.
- c) Kart okuyucu olmadan kullanıcı ile iletişimde bulunabilmektedir.

Multiprogrammed Batch Systems ile Time-Sharing Systems arasındaki temel fark,

Multiprogrammed batch sistemleri söz konusu olduğunda, amacın işlemci kullanımını en üst düzeye çıkarmak iken, Time-Sharing Systems'de amaç yanıt süresini en aza indirmektir. CPU tarafından aralarında geçiş yapılarak birden fazla iş yürütülür, işler arasındaki geçişler çok sık gerçekleşir. Böylece kullanıcı anında yanıt alabilir. Örneğin, bir işlem işletilirken, işlemci her kullanıcı programını kısa bir hesaplama süresinde veya kuantum hesaplamayla yürütür. Yani, n kullanıcı mevcutsa, her kullanıcı bir zaman kuantumu alabilir. Kullanıcı komutu gönderdiğinde, yanıt süresi en fazla birkaç saniyedir.

İşletim sistemi, her kullanıcının iş talebine en kısa sürede yanıt verebilmek için CPU planlamasını ve çoklu programlamayı kullanır. Öncelikle toplu sistemler olarak tasarlanan bilgisayar sistemleri, zaman paylaşım sistemlerine dönüştürülmüştür.

Zaman Paylaşımı işletim sistemlerinin avantajları aşağıdaki gibidir -

Hızlı yanıt avantajı sağlar.

Yazılımın kopyalanmasını önler.

CPU boşa kalma süresini azaltır.

Zaman paylaşımli işletim sistemlerinin dezavantajları aşağıdaki gibidir -

Güvenilirlik sorunu.

Kullanıcı programlarının ve verilerinin güvenliği ve bütünlüğü sorunu.

Veri iletişimi sorunu

11.3.4. Dağıtık İşletim Sistemi (Distributed Operating Systems)

Dağıtık işletim sistemleri birden fazla makinaryı tek ve entegre bir makina olarak davrandırmak amacıyla tasarlanmıştır. Ağ işletim sistemlerinden (Örnek linux, windows) farklı olarak dağıtık işletim sistemi kullanıcıları ortamda birden fazla bilgisayarın bulunduğu farkında değildir.

Ağ işletim sistemlerinde her kullanıcının oturum açabildiği yerel bir işletim sistemi varken dağıtık işletim sistemlerinde tek sistem imajı kullanıcılara dağıtılmış durumdadır. Kullanıcılar, komutlarını çalıştıran işlemcilerin ve dosyalarını saklayan dosya sunucularının sayısından veya yerinden haberdar olmazlar. Sıradan kullanıcılar için dağıtık işletim sistemi, sıradan bir time-sharing işletim sistemi gibidir. Kullanıcıların, dağıtık işletim sistemini kullanırken time-sharing işletim sistemiyle arasındaki farkı anlayamaması dağıtık işletim sisteminin başarısının kanıtıdır. Dağıtık işletim sistemlerinde temel amaç paralellik,transparanlık ve performanstır.

Dağıtık işletim sistemlerinin avantajları aşağıdaki gibidir -

Kaynak paylaşım tesisi ile, bir sitedeki bir kullanıcı, başka bir sitedeki mevcut kaynakları kullanabilir.

Elektronik posta yoluyla veri alışverişi hızlıca yapılabilir.

Dağıtılmış bir sistemde bir site başarısız olursa, kalan siteler potansiyel olarak çalışmaya devam edebilir.

Müşterilere daha iyi hizmet.

Ana bilgisayardaki yükün azaltılması.

Veri işlemedeki gecikmelerin azaltılması.

Amoeba, Plan 9, Chrous, Mungi dağıtık işletim sistemi örnekleridir. Amoeba hem dağıtık hesaplama (farklı farklı projeler üzerinde çalışan birden fazla kullanıcı), hem paralel hesaplama (paralel satranç oynamak için 50 cpu kullanan tek kullanıcı) uygulamaları için tasarlanmıştır. Dağıtık hesaplama ve paralel hesaplama gerektiren alanlar için kullanılabilir olmuştur. (Örneğin, yapay zeka)

Plan 9 tam adıyla Plan 9 from bell labs Unix kurucuları bell labs tarafından 1980 ortalarından 2002 yılına kadar geliştirilen dağıtık işletim sistemidir.(bkz. dennis ritche,rob pike,ken thompson,dave presotto,phil winterbottom) Bu işletim sistemi geliştirilirken utf-8 character encoding'i de geliştirilmiştir.

11.3.5. Network İşletim Sistemleri (Sunucu İşletim Sistemleri)

Birden fazla bilgisayarın kaynakların paylaşılması amacıyla birbirine bağlandığı sisteme network denir. Bu ağ sistemini yöneten bilgisayarlara sunucu ve sunucu üzerinde çalışan, aynı zamanda da tüm ağı yönetebilme kapasitesine sahip işletim sistemine ise Network İşletim Sistemi (NOS: Network Operating System) denir.

Bilgi işlem uzmanlarına altyapı üzerinde maksimum kontrol gücü sağlayan sunucu işletim sistemleri aynı zamanda yüksek düzeyde kullanılabilirlik sunar. Sahip olduğu idari araçlar sayesinde pek çok görevin otomatik bir şekilde yapılabilmesini destekleyerek hem işletmenin maliyetini düşürür hem de işletmeye güçlü ve güvenli bir sunucu ortamı sağlar.

Ağ işletim sistemlerinin avantajları aşağıdaki gibidir -

Merkezi sunucular oldukça kararlıdır.

Güvenlik, sunucu tarafından yönetilir.

Yeni teknolojilere ve donanıma yükseltmeler sisteme kolayca entegre edilebilir.

Sunuculara uzaktan erişim, farklı konum ve sistem türlerinden mümkündür.

Ağ işletim sistemlerinin dezavantajları aşağıdaki gibidir -

Bir sunucu satın almanın ve çalıştırmanın yüksek maliyeti.

Çoğu işlem için merkezi bir konuma bağımlılık.

Düzenli bakım ve güncellemeler gereklidir.

Ağ işletim sistemi örnekleri arasında Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, BSD gibi sistemler bulunur.

11.3.6. Gerçek Zamanlı İşletim Sistemleri (Real Time operating System) RTOS ve LynxOS

1. Gerçek zamanlı işletim sistemleri, kaynak yönetimi, iletişim, hassas zamanlama, senkronizasyon ve planlama için destek sağlarlar[1]. RTOS, görevlerle ilişkili bir son tarihi vardır ve bu tarihi geçirmeden görevlerini tamamlaması gerekmektedir[2]. RTOS deterministik, hizmet kalitesi (QoS) ve son zaman (deadline) olmalıdır. Deos (DDC-I), embOS (SEGGER), FreeRTOS (Amazon), Integrity (Green Hills Software), Keil RTX (ARM), LynxOS (Lynx Software Technologies), MQX (Philips NXP / Freescale), Nucleus (Mentor Graphics), Neutrino (BlackBerry), PikeOS (Sysgo), SafeRTOS (Wittenstein), ThreadX (Microsoft Express Logic), µC/OS (Micrium), VxWorks (Wind River), Zephyr (Linux Foundation) popüler gerçek zamanlı işletim sistemleridir.

***Deterministik:** Bir girdiye her zaman aynı çıktıyı üretmek*

***Hizmet Kalitesi (QoS):** Bir ağın gecikme ve hata oranları, ağın performansı*

***Son Zaman (Deadline):** Görevin tamamlanması gereken son zaman*

2. Gerçek Zamanlı İşletim Sistemi Türleri

2.1. Zor RTOS (Hard RTOS)

Görevin son tarihi sıkı şekilde ele alınır, görev belirtilen sürede başlayıp belirtilen sürede bitirilmesi gerekmektedir. Bu tip RTOS'lar görevin son tarihine bağımlıdır ve görev gecikmeye girerse sistem bozulur. Bu yüzden bu tarz sistemler gecikmelere tahammül edemezler. Uçak sensörü, uzay araçları gibi.

2.2. Sağlam Gerçek Zamanlı (Firm RTOS)

Görevin son tarihinde sonlanmamış olması büyük bir tehlike arz etmez fakat kalitede büyük bir düşüş istenmeyen etkilere neden olabilir. Multimedya uygulamaları gibi.

2.3. Yumuşak RTOS (Soft TROS)

Bu sistemlerde, gecikme kabul edilebilir bir durumdur fakat istenilen bir durum değildir. Ses ve video dağıtım yazılımları örnek verilebilir.

3. Gerçek Zamanlı İşletim Sistemi Özellikleri

Görev Önceliği (Task Priority): En çok ihtiyaç duyulan görevi tanımlama ve kaynağı elde etmek için denetimi gerçekleştirme

Görevler Arası İletişim Mekanizması (Task Communication Mechanism) Birden fazla görevin iletişim kurması ve veri bütünlüğünü sağlaması için senkronizasyon mekanizması gerekir.

Çoklu Görev (Multitasking): Birden fazla görevin aynı anda yürütmeye olanak sağlar

Belirleyici Davranış (Deterministic behaviour): Süreçler ve kesintiler (interrupts) belirli bir süre içinde ele alınır

Tanımlanmış Yığın Kullanımı (Defined stack usage): Her görev için tanımlanmış bir yığın alanı tahsis edilir

Sistem Yönetimi (System management): Kaynak yönetimi yerine, uygulama geliştirmeye olanak tanır

4. Gerçek Zamanlı İşletim Sistemleri Neden Tercih Edilir

- *Önceliğe dayalı programlama sunar*
- *Daha az hafıza kullanımı*
- *Az kaynak tüketimi*
- *Hızlı yanıt süreleri*
- *API servisi sağlarlar*
- *Zaman kaybı olmaksızın süreçlere odaklanır*
- *Modüler geliştirmeye teşvik eder*
- *Görev tabanlı tasarımı vardır*

5. Gerçek Zamanlı İşletim Sistemleri Hakkında Bazı Yanlışlar

- *RTOS çok hızlıdır! Bir göreve hızlı yanıt vermek hızlı oldukları anlamına gelmez*
- *Tüm RTOS'lar aynıdır! Aslında daha önce bahsettiğimiz gibi Zor ve Yumuşak RTOS olmak üzere farklı yapılar mevcuttur.*

- CPU ek yükü kullanımı fazladır! Bir RTOS CPU'nun %1-%4 zamanını kullanır.
- RTOS tasarımı bilim yoktur! Gömülü sistemlerin karşılaştığı sorunlar üzerine çalışırken geliştirilmiştir.
- Çok fazla elektrik tüketir! Kullanılan mimariye ve donanımlara göre değişkenlik gösterir.[2]

6. Gerçek Zamanlı İşletim Sisteminde Kullanılan Bazı Terimler

- **Görev (Task):** Sistem işlevlerini birlikte sağlayabilen bir dizi
- **İş (Job):** İşlemciye atanan küçük iş parçası
- **İşin Serbest Bırakılma Zamanı (Release time of a job):** İşin uygulamay hazır hale geldiği zaman
- **İşin Yürütme Süresi (Execution time of a job):** İşin bitmesi için geçen süre
- **İşin Son Tarihi (Deadline of a job):** İşin tamamlanması gereken zaman
- **İşlemciler (Processors):** Aynı zamandaki aktif kaynaklar
- **İşin Yanıt Süresi (Response time of a job):** İşin başlayıp bittiği anda geçen süre
- **Mutlak Bitiş Tarihi (Absolute deadline):** Göreceli işin son tarihi

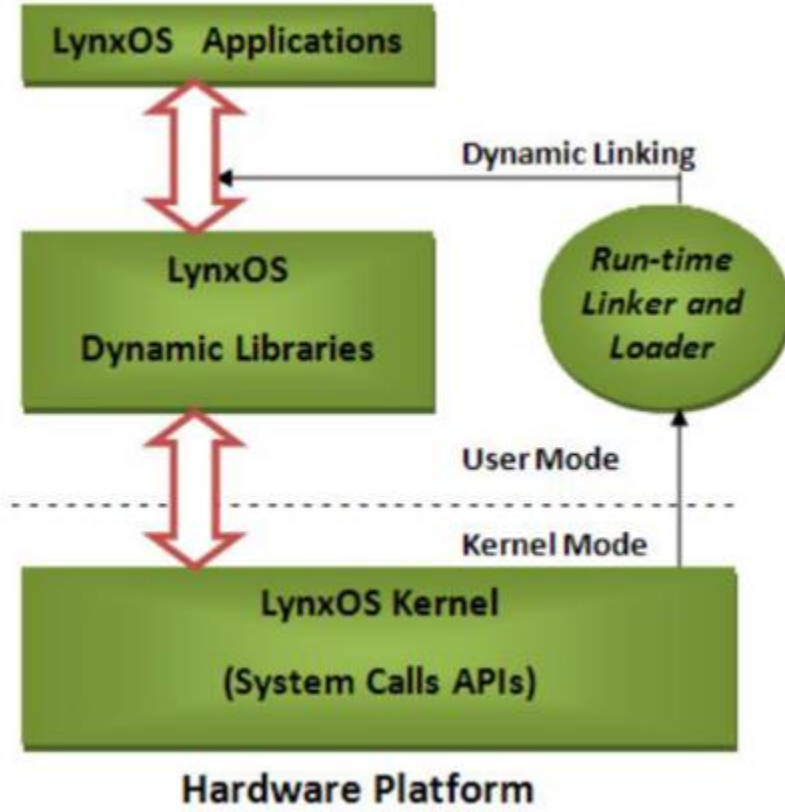
7. Genel Amaçlı İşletim Sistemleri ile Gerçek Zamanlı İşletim Sistemleri Farkları

Genel Amaçlı İşletim Sistemleri	Gerçek Zamanlı İşletim Sistemleri
Masaüstü, dizüstü için kullanılırlar	Yalnızca gömülü sistemler
Süreç bazlı planlama	Zamana dayalı çizelgeleme
Gecikmeler önemli görülmez	Mikrosaniye bile olsa gecikmeler önemli
Öncelikli görevlendirme yoktur	Önceliklendirme mekanizması vardır

8. LynxOS

8.1. LynxOS Gerçek Zamanlı İşletim Sistemi

İlk olarak 1986 yılında tanıtıldı. Lynx Yazılım Teknolojileri tarafından geliştirilen, havacılık, askeri, uzay ve savunma uygulamaları için zor gerçek zamanlı bir işletim sistemidir. ARM ve PowerPC mimarilerini desteklemektedir[3]. Kanıtlanmış güvenlik, güvenilirlik geçmişine sahiptir. Kullanan kurumlar arasında NASA, Boeing, General Dynamic ve ABD Silahlı Kuvvetleri bulunmaktadır[4].



LynxOS Mimarisi [5]

Önleyici bir işletim sistemidir ve zaman açısından kritik sistemlerde, görevlerin anında biteceğini garanti eder. Ağa bağlı sistemlerde çok iş parçacıklı olarak çalışabilen bir sistemdir. Bellek yönetim birimi (MMU) kullanımı gereklidir ve güvenlik olarak iyidir ancak esnek olmadığı için olumsuz olarak görülmektedir[5].

8.2. LynxOS Hakkında Detaylar

LynxOS, kabuğunun UNIX benzeri olduğu için daha fazla RAM ihtiyacı hissetmektedir. Araçları kullanma konusunda diğer gerçek zamanlı işletim sistemlerine göre daha özgürlükçü bir yaklaşım izlemektedir. LynxOS'da bellek yönetimi devre dışı bırakılamadığından sistemi daha güvenli çalıştırma imkanı sunmaktadır.

8.3. LynxOS Özellikleri

- **CPU Planlama:** Dinamik zaman dilimlemeyi kullanılır
- **Hafıza Yönetimi :** Geliştiriciyi bellek yönetimi kullanmaya zorlar
- **Boyut:** Optimize edilmiş boyuta sahip olması gereken sistemler için uygun değildir
- **Güvenlik:** MMU kullanımı nedeniyle son derece güvenlidir
- **Bellek Ayırma:** Bellek ayırma algoritmalarını gizli tutar
- **Esneklik:** MMU kullanımı gerektiğinden iyi bir esnekliğe sahip değildir