**HAFTA:1**

İşletim Sistemleri: Bilgisayar donanımı yöneten program.

BİLGİSAYAR SİSTEMİ: Donanım, İşletim Sistemi, Programlar, Kullanıcılar

SİSTEM AÇISINDAN: Kaynak kullanımını planlayan bir yazılım.

İŞLETİM SİSTEMİ: Sürekli çalışır ve kernel olarak adlandırılır.

MOBİL İŞLETİM SİSTEMİ: KErnel ve Middleware yapısına sahiptir.

MİDDLEWARE: Uygulama geliştiricilere ek özellikler sağlayan yazılımlardır.

BOOTSRAP:

* Bilgisayar çalışmaya başladığında çalışan program.
* Oldukça basit yapıda ROM (Read-Only-Memory) ya da EEPROM(Electrically Eraseble Programmable Read-Only-Memory) içinde olabilir.
* İşletim sisteminin kernel yerini bilir ve hafızaya yükler.

FİRMWARE: Bootstrap programlarına verilen ad.

* Bilgisayarın tüm yapısını başlatır.

Bazı servisler kernel dışından çalışabilir. (System daemons, system processes)

UNİX İLK PROCESS : İNİT

Bilgisayar sistemlerinde bir **event(olay)** olması **interrupt(kesme)** ile bildirilir.

Yazılımlar **system cal** çağrılarını başlatarak **interrupt** başlatır.

CPU interrupt alıyor ve çalıştığı yeri bırakıp oraya gidiyor.

Kesme yapılan yerin hafızası biliniyor service routine’nin belirlediği başlangıç adresidir.

**Service Routine’nin** belirlediği yere gidip geldikten sonra önceki yere gider ve çalışmaya devam eder.

İnterrupt routine adresleri **pointer** ile gösterilir.

VON NEUMANN: Komutlar **fetch** ile hafızadan alınır **execute** ile yürütülür.

VOLATİLE : Enerji kesildiğinde veri gidiyor. RAM,CACHE,REGİSTER

SECONDARY STORAGE : İkincil hafıza hdd,sdd

DEPOLAMA YAPISI:

* Register
* Cache
* Main memory (RAM)
* Solid State Disk
* Magnetice disk
* Optical Disk
* Magnetic Tapes
* Yukarı çıktıkça hız artar depolama oranı düşer.

NON-VOLATİLE: Solid-State Disk, Magnetic Disk, Magnetic Tape: Veriyi kalıcı saklar.

SCSI(Small computer system interface): Küçük ölçekli bilgisayar sistem arayüzü .

DEVİCE DRİVER: Her cihaz denetleyicisi için bir tane vardır.

MODERN BİLGİSAYARLAR **DMA(Direct Memory Access)** İLE VERİ AKTARIMI YAPAR.

TEK İŞLEMCİLİ SİSTEMLER:

* Komut kümesindeki tüm komutlar bir işlemci tarafından çalıştırılmaktadır.
* İşletim sistemi tarafından yönetimi yapılmaktadır.
* Bu sistemler özel olmak üzere, özel işlemcilere sahip olabilirmektedir.
* Disk denetleyicisi işlemcisi , CPU’dan gelen istekleri yönetir.

ÇOK İŞLEMCİLİ SİSTEMLER:

* Multiprocesser system, parallel systems, multicore systems.
* İki ya da daha fazla CPU vardır.Bus,clock , memory ve çevre birimlerini paylaşırlar.
* Sunucular, masaüstü, dizüstü, server, mobil cihazlar.
* Yüksek hız; işlemci sayısı artarsa hızlanma oranı artar denemez.
* Ekonomik ölçeklendirme: Çevre birimlerini ortak kullanırlar.
* Yüksek Güvenilirlik: Bir hata sistemi çökertmez:

GRACEFUL DEGRADATİON: Performans azalması.

ASİMETRİK ÇOK İŞLEMCİLİ SİSTEMLER: (AMSP)

Her işlemci tek bir işe atanmış. Bir işlemci tarafından yönetilirler.

SİMETRİK ÇOK İŞLEMCİLİ SİSTEMLER:(SMP)

* Her işlemci işletim sistemindeki tüm işleri yapabilir. Tüm işlemciler eş düzey çalışır (PEER).
* Her işelmci kendi register ve cahe yapısına sahiptir. Ancak hafıza paylaşılır.

UMA: (Uniform memory access) : Ram’e erişim süresi tüm işlemcilerde **aynı** olan model

NUMA : (Nonuniform memory access) : Ram’e erişim tüm işlemcilerde **farklı** olan model.

MULTİCORE SİSTEMLER: Bir chip üzerinde birden fazla işlemci yapısı.

* Her core kendi register ve önbelleğe sahiptir fakat hafıza paylaşılır.

CLUSTERED SİSTEMLER: Bağımsız iki veya daha fazla sistemden oluşur.

* Bu sistemler depolamayı paylaşır ve LAN(Local area) üzerinden haberleşir.
* Loosely Coupled (gevşek bağlı)
* Hata olursa diğer düğümün görevini bir başkası alır.
* Bir program parçalara bölüştürülür ve çalışrırılır (parallelization)
* DLM(distributed-lock manager)

MULTİPROGRAMMİNG: İşletim sisteminin birden fazla programı çalıştırabilmesidir.

JOB POOL: İş havuzu, yapılacak işleri tutar.

PROCESS: Hafızaya yüklenen ve çalıştırılan program.

CPU SCHEDUELİNG: Hafızada çok fazla process varsa yapılır.

JOB SCHEDUELİNG: Hafızaya alınmak için bekleyen çok process varsa yapılır.

İNTERRUPT DRİVEN: Modern işletim sistemi kesilmelerle sürülür.

TRAP-EXCEPTİON : Yazılım tarafından üretilen bir interrupt. İşletim sistemini tetikler.

KULLANICI MODU: User mode ==1

KERNEL MODEU (0): İşletim sistemi uygulama tarafında bir istek aldığında buraya gelip gider.

MODEBİT: Hangi modda çalışıldığına karar verilir.

PROGRAM: Pasif varlıktır – passive entity PROCESS: Aktif varlıktır.

PROGRAM COUNTER: Komut adresi tutar.

İŞLETİM SİSTEMİ HAFIZA YÖNETİMİ:

* Hafızanın hangi kısmının kullanıldığı ve kimin taradaından kullanıldığı.
* Hangi process’in hafızaya alınacağı.
* Hafızaya boş alan tahsisi veya serbest bırakılması.

ÜÇÜNCÜL HAFIZA: CD,DVD,Manyetik Tape

* Cache belleklerde erişim adrese göre registerlerda isme göre yapılır.

YIĞIN : LIFO

KUYRUK: FIFO

FİREWALL: IP,port,WEB,İçerik filtreleme yapabilir.

TCP/IP : Transmission Control Protocol/ Internet Protocol

FTP/NFS: Sadece dosya erişimine yönelikleridir. (File Transfer Protocol/Network File System)

PAN :(Personal Area Network)

P2P: İstemci ve sunucu ayrımı yoktur. Her birim aynı yetki ve kapasiteye sahiptir. Aaaaa Blockchain

**HAFTA:2**

İŞLETİM SİSTEMİ: Uygulama programlarının çalışması için ortam sağlar.

* İşletim sistemlerinin kendi servisleri ve ortak servisleri vardır.

KULLANICI ARAYÜZÜ:

* CLI► Xommand Line İnterface, metin komutları alır.
* Batch(batch interface) ► Komutlar dosya içeriğine sağlanır.
* GUI(Graphical user interface)

PROGRAM ÇALIŞTIRMA:

Sistem bir programı yükleme , çalıştırma ve sonladırma işlemlerini yapar.

I/O İŞLEMLERİ:

I/O cihazları yönetimi sağlanır.

* Dosya erişimlerini düzenleyen işlevleri sağlarlar.

İLETİŞİM: Bir process’in başka bir process ile haberleşmesi.

* İletişim **hafıza(shared memory)** kullanılarak yapılabilir.

HATA DENETİMİ: İşletim sistemi hataları düzenler, sınıflandırır, izler, raporlar ve düzeltir.

COMMAND INTERPRETER: (Komut Yorumlayıcısı)- SHELL

* Kernel kısmında bulunabilir.
* Windows ve UNIX ise ayrı bir program derler.
* Create, delete, list ,print , copy, execute işlemleri yapılabilir.

GUI:

* İlk GUI 1973 yılında Xerox Alto bilgisayarlarda kullanıldı.

SYSTEM CALLS: Servislerin işletim sistemi tarafından kullanılabilmesi için arayüz sağlar.

* Dosya okumak, içeriği okumak, kapatmak (ayrı ayrı sistem çağrılarıdır).
* Dosya bulunamadı, dosya okunamaz (sistem çağrları)
* Uygulama geliştirici API kullanarak işletim sistei fonksiyonlarını kullanır.
* İşletim sistemi fonksiyonları kernel’daki sistem çağrılarını başlatır.
* CreateProcess() komutu kernelda NTCreateProcess() çağrısıını çağırır.

SİSTEM ÇAĞRI TÜRLERİ:

* Process Control
* Dosya Yönetimi
* Cihaz Yönetimi
* Bilgi Bakımı
* İletişim
* Koruma

PROCESS CONTROL:

* Program çalışması end() ya da abort() şekilde bitebilir.
* Debugger tarafından hatalar incelenir ve bug düzeltilir.
* Load() program çalışması için gerekli, execute() yürütülme.
* Acquirelock() kilitleme releaselock() serbest bırakma
* MS-DOS(single tasking işletim sistemi)
* FreeBSD(Berkeley Software Distribution) multi-tasking işletim sistemidir.

FİLE MANAGEMENT:

* Create() delete()
* Open() close()
* Read() write()
* Fileattributes() set file attributes()
* Copy() move()

CİHAZ YÖNETİMİ

* Farklı kaynak ister
* Fiziksel kaynaklar sanal cihazlar
* Request() release()

İLETİŞİM:

* Message-passing model; mesajlar prosesler arasınsında doğrudan ve doalylı gönderilir.
* Her processing bir process adı ve ID değeri vardir.
* Shared-Memory Model: Hafızadan çeker

KORUMA:

* Set permissin() get permission()
* Allow user() denyuser()

MEKANİZMA: Nasıl yapılacağını belirler- time kullanılacak

Kurallar(Policies) : Ne yapılacağını- timer süresi ne kadar olacka

ASSEMBLY : İlk işletim sistemini yazan dil

MİKROKERNEL: 1980 Yıl MACH işletim Sistemi

Kerneli küçültmek amaçlanmıştır.

WİNDOWS: Mikrokerneli kullanır

LİNUX: Monolithic yapıdadır, dinamik olarak güncellenebilir.

CRASH : Hafızada ortaya çıkan hata.

**HAFTA:3**

PROCESS : İşi,PC değerini ve CPU Register değerini barındırır.

İçerir; Stack ,Data Section, Heap

NEW: Process oluşturur,

RUNNİNG: Komutları çalıştırır

WAİTİNG: Birini bekliyor

READY: Hazırım gönder.

TERMİNATED: Bittim.

PCB : Process Control Block

PROCESS SCHUDELER: Bir processi çalıştırmak için seçer.

JOB QUEUE: Tüm işlerin bulunduğu yer

READY QUEUE: Hafıza alınan işler.

I/O BOUND: I/O işlemleri için daha fazla

CPU- BOUND: Cpu için daha fazla

CONTEXT SWITCH: CPU’nun bir işten öbürüne geçmesi.

0 : Child Process

>1 : Baba Process

**HAFTA:4**

THREAD: PC,Register ve Stack sahiptir.

FAYDALAR:

* Cevap Verebilirlik artar
* Kaynak payalaşımı- ekstra gerek yok
* Ekonomik
* Ölçeklenebilirlik

PARALLELİSM: Birden fazla görevin eş zamanlı yapılması ifade eder

CONCURRENCY: Birden fazla görev arasında kısa aralıklarla geçiş yapılarak birlikte ilerletilmesini ifade eder.

CORE:İŞLEMCİ=CPU=PROCESS=THREAD

AMDAHL KURALI: Core sayısı artarsa sistemdeki performasn artar.

ZORLUKLAR:

* Zamanlama tutturmak
* Farklı belleklerin gerekmesi
* Denge kurulması
* Veriyi bölme
* Veri bağımıslığı
* Test ve debug işlemi

DATA PARALLELLSİM:

Aynı veri kümesine ait parçaların core’lara dağıtılıp aynı tür işlemin eşzamanlı yürütülmesine odaklanır.

TASK PARALLELLİSİM:

Core’lara görevlerin dağıtılmasına odaklanır.

MANY TO ONE: Çok sayıda kullanıcı thread var sadece bir tane kernel thread var.

ONE TO ONE: Bir kullanıcı thread = Bir kerne –l thread.

MANY TO MANY: Çok sayıda kullanıcı thread Çok sayıda kernel thread

PTHREADS: Thread kütüphanesı - Linux

WİNDOES THREAD: Windoes thread kullanılır.

JAVA THREAD: Java kullanır

**HAFTA:5**

Aynı anda iki process kritik bölüme erişmemeli.

MUTUAL EXCLUSİON: Bir process kritik bölümü kullanırsa diğer processler elleşemez.

PROGRESS: Hiçbir process kritik bölümde değilse birine izin verilir.

BOUNDED WAİTİNG:

Bir sınır değeri vardır.

RACE CONDİTİON: İki process’in aynı anda istek atması.

SEMAFOR: Wait() Signal()

**HAFTA:6**

**HAFTA: 7**

HAFIZA: Allocate() free()

Semafor: wait() signal()

Mutex Kilitlemesi: acquire() release()

FİZİKSEL KAYNAKLAR: Yazıcı hafıza

MANTIKSA KAYNAKLAR: Semafor, Mutext

DEADLOCK ŞARTLARI:

* Mutual Exclusion : kaynak bende istek yapıyorsun
* Hold and wait: kaynak bende seni bekliyorum.
* No preemotion: Kaynak kafadan boşaltılmaz önce process salsın.
* Circual Wait: her process diğerini bekler, döngüse bekleme.

DİKDÖDÖRTGEN: Kaynaklar

DAİRE: Process

DEADLOCK-PREVENTİON: Kaynak isteklerini sınırlandırır.

DEADLOCK-AVOİDANCE: İstiyorsun ama ne kadar süreliğine?