Operativsystemets Uppbyggnad

Mittuniversitetet
MID SWEDEN UNIVERSITY

Operativsystem

- Ett program som kontrollerar exekveringen av applikationsprogram
- Fungerar som interface mellan applikationer och hårdvara

Mittuniversitetet
MID SWEDEN UNIVERSITY

Operativsystems uppgift

- Bekvämlighet
  - Gör datorn mer användarvänlig
- Effektivitet
  - Gör att datorns systemresurser utnyttjas mera effektivt
- Utvecklingsmöjligheter
  - Effektiviserar utveckling, testning, och introduktion av nya systemfunktioner

Mittuniversitetet
MID SWEDEN UNIVERSITY

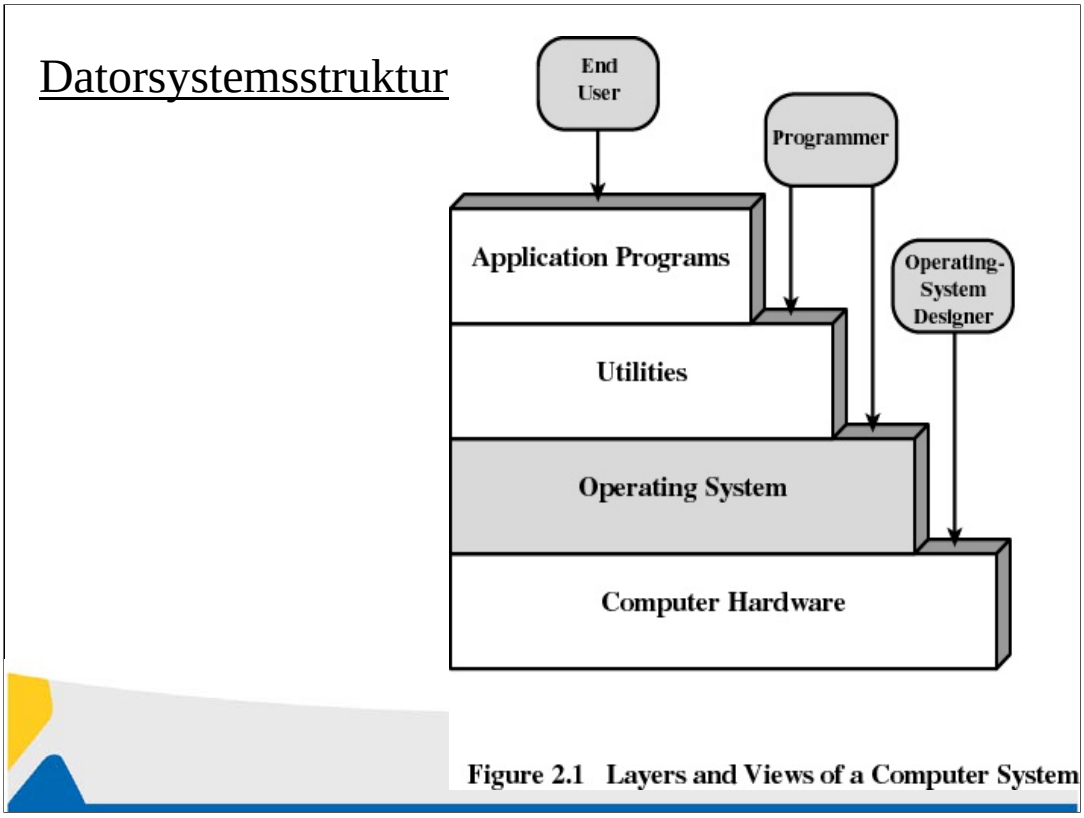


Bild 4: Observera att Operativsystemet (OS) och ev. utilities erbjuder API (Application Programmers Interface) gentemot programmerare samt att OS ligger mellan en applikation och datorns hårdvara. I moderna OS är det omöjligt att adressera någon hårdvara direkt. Alla anrop måste gå via OS.

Bild 5: OS döljer detaljerna om hårdvaran och erbjuder standardiserade interface. Ex access till en ström av data behandlas som fil oavsett om den kommer från floppy, hårddisk eller I/O-enhet. Med systemaccess menas hantering av tillgång till samtliga resurser. Dvs. processor, minne, lagringsmedia…

Operativsystemets Tjänster

- Feldetektering och felhantering
  - Interna och externa hårdvarufel
    - Fel på minne
    - Fel på enhet
  - Programfel
    - Aritmetiskt overflow
    - Access till förbjudna minnesadresser
  - Operativsystemet kan inte utföra av applikation begärd tjänst

Mittuniversitetet
MID SWEDEN UNIVERSITY

Operativsystemets tjänster

- Accounting Logg
  - Statistik över utnyttjandet av resurser
  - övervaka prestanda
  - Används för att spåra problem och hitta lösningar på dessa
  - Fördela kostnader på användare, vid stordatordrift

Mittuniversitetet
MID SWEDEN UNIVERSITY

Operativsystem

- Fungerar som annan mjukvara
  - Är ett program som exekveras, i bakgrunden
- Operativsystemet överlämnar kontrollen (lånar ut) av processorn till applikations-programmen

Mittuniversitetet
MID SWEDEN UNIVERSITY

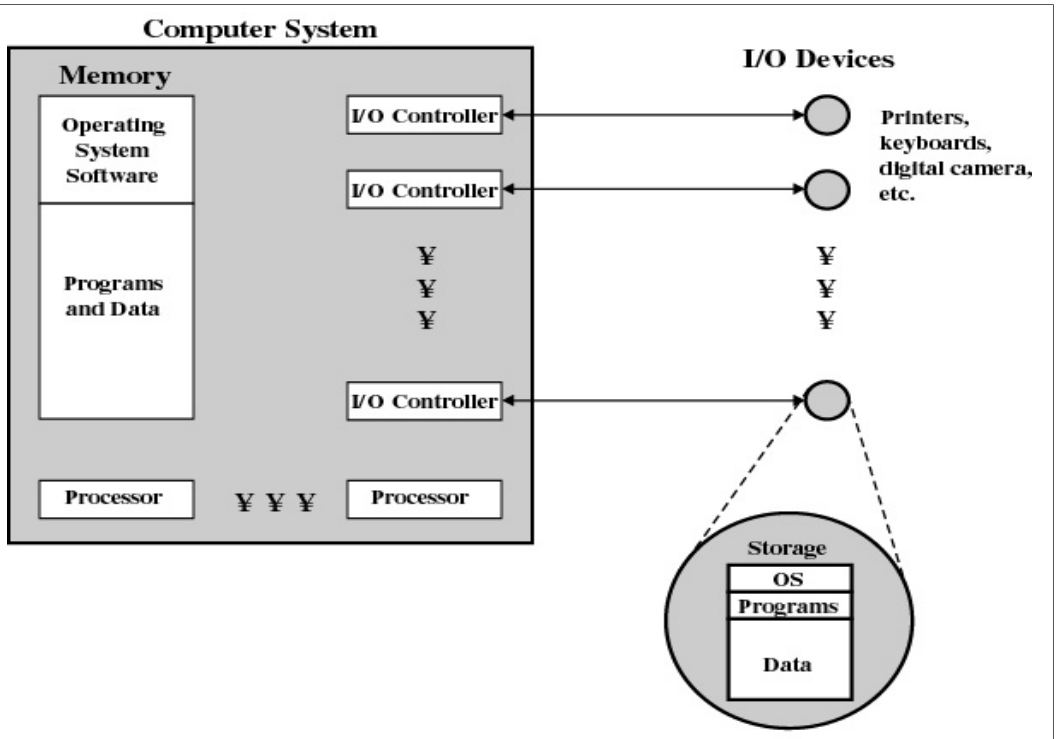


Figure 2.2 The Operating System as Resource Manager

Kernel/nucleus

- En del av operativsystemet som alltid ligger i primärminnet
- Innehåller de oftast använda funktionerna
  - Processkontroll
  - Filkontroll
  - Hårdvarukontroll
  - Övervakning

Mittuniversitetet
MID SWEDEN UNIVERSITY

Operatingsystemet Förändras

- Hårdvaran uppgraderas och nya typer av hårdvara tillkommer
- Nya tjänster (ex. Nätverk)
- Buggfixar

Mittuniversitetet
MID SWEDEN UNIVERSITY

Operatingsystemens Utveckling

- Seriell exekvering
  - Inget operativsystem
  - datorn körs från en konsol med indikerings-lampor och strömbrytare, inmatnings-enheter, och skrivare
  - Schedulering m.h.a. bokningslista
  - Före exekveringen måste kompilator och källprogram laddas, programmet kompileras, kompilerat program sparas, sedan laddas och länkas.

Mittuniversitetet
MID SWEDEN UNIVERSITY

1940

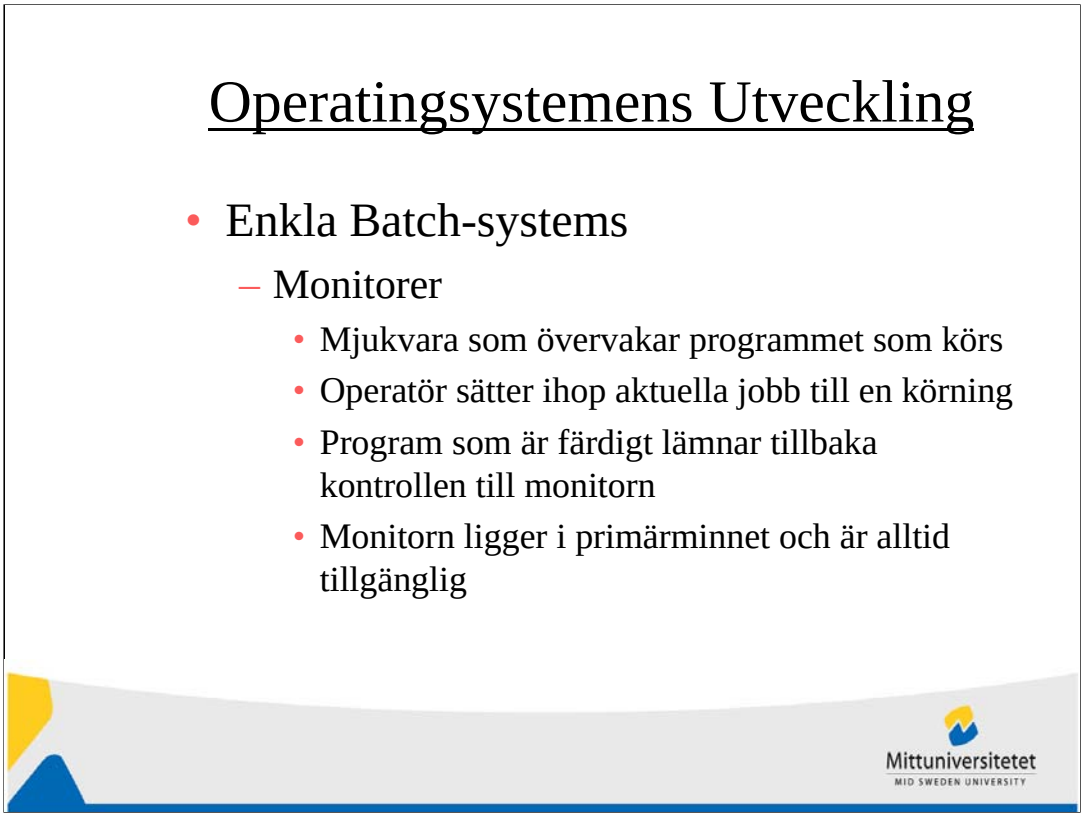


Bild 13: Från början var hårdvaran dyr men personalkostnaden liten. Målet var att maskinen/processorn skulle utnyttjas så effektivt som möjligt.

Job Control Language (JCL)

- Speciell typ av programmeringsspråk
- Innehåller instruktioner till monitorn
  - Vilken kompilator som ska användas
  - Vilket data som ska användas

Mittuniversitetet
MID SWEDEN UNIVERSITY

Instruktioner typ load run..

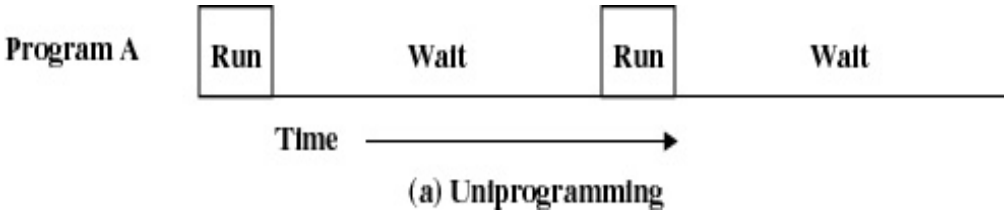
Hårdvarukontroll

- Skyddat minnesområde
  - Tillåter inte att något program använder det minne som monitorn ligger i
- Timer
  - Ser till att monitorn får tillbaka kontrollen även om ett program skulle låsa sig

Mittuniversitetet
MID SWEDEN UNIVERSITY

Ett Program i Taget

- Processorn måste vänta på att I/O-operationer ska bli klara innan den kan fortsätta



(a) Uniprogramming

Läsa data från fil 1,5 ms

Exekvera 100 instruktioner 0,1 ms

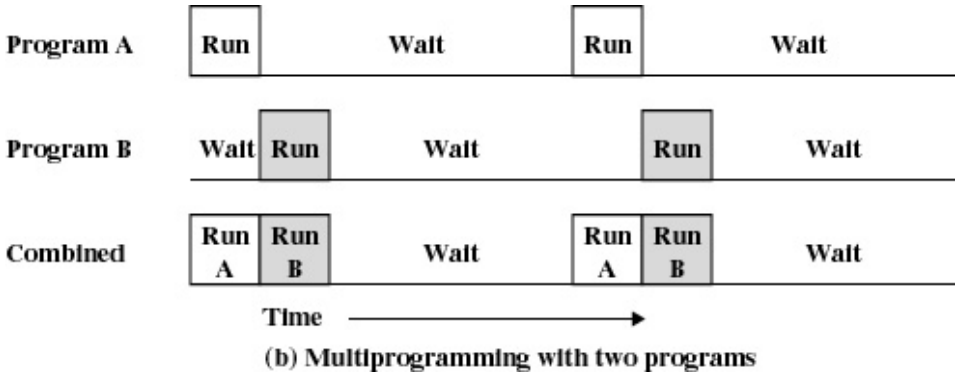
Skriva data till fil 1,5 ms

Totalt 3,1 ms

CPU utnyttjande 0,1/3,2=3,2%

Multiprogrammering

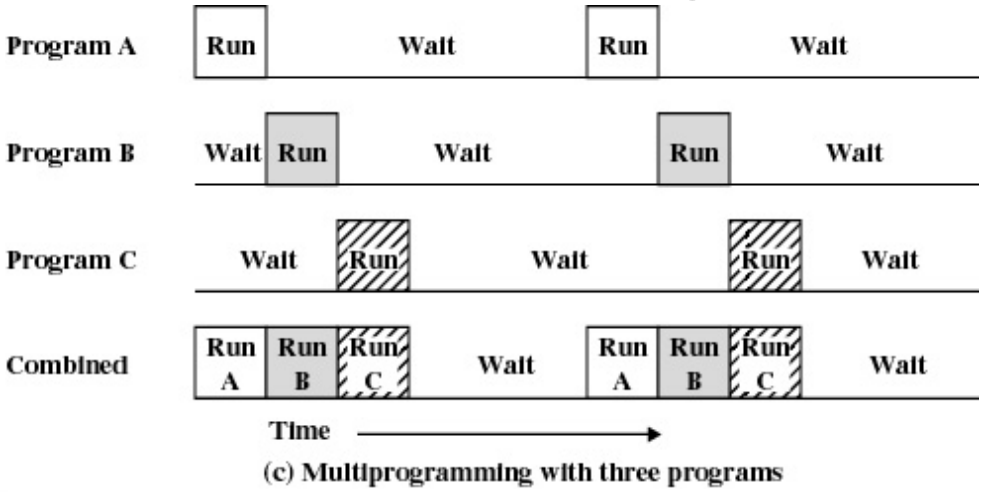
- Under tiden som ett jobb väntar på I/O, kan processorn köra ett annat jobb



Mittuniversitetet
MID SWEDEN UNIVERSITY

Bild 17-: Från början var jobben av Batch-typ dvs. ingen interaktion med användare. För att klara flera samtidiga interaktiva jobb krävs Time sharing med preemptive multitasking. Dvs. en process har inte exklusiv tillgång till processorn tills den är färdig. Den kan avbrytas av OS för att släppa in andra processer.

Multiprogrammering, multitasking



(c) Multiprogramming with three programs

Mittuniversitetet
MID SWEDEN UNIVERSITY

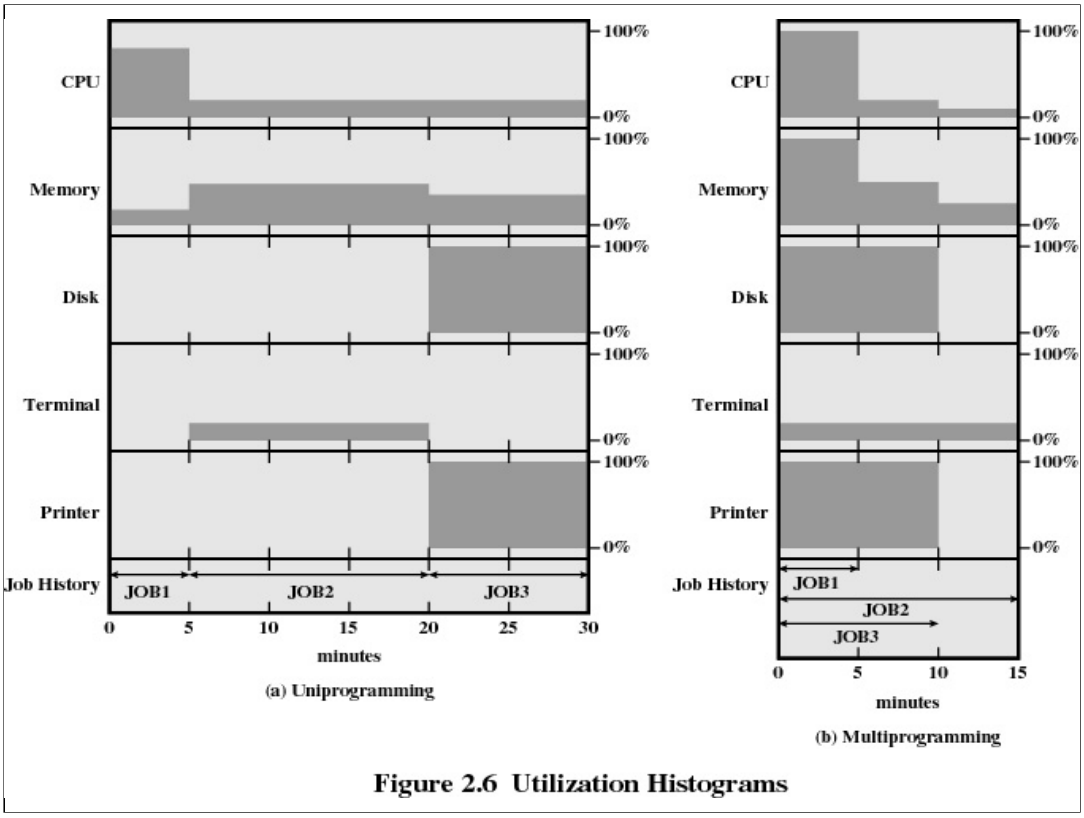


Figure 2.6 Utilization Histograms

Exempel

	JOBB1	JOBB2	JOBB3
Typ av jobb	Tung beräkning	Tung I/O	Tung I/O
Varaktighet	5 min.	15 min.	10 min.
Minneskrav	50K	100 K	80 K
Diskaccess?	Nej	Nej	Ja
Terminal	Nej	Ja	Nej
Utskrift?	Nej	Nej	Ja

Mittuniversitetet
MID SWEDEN UNIVERSITY

Jämförelse Uni/Multitasking

	Uniprogramming	Multitasking
Processor anv.	22%	43%
Minnes anv.	30%	67%
Disk anv.	33%	67%
Printer anv.	33%	67%
Total tid	30 min.	15 min.
Throughput	6 jobs/hr	12 jobs/hr
Genomsnitt svarstid	18 min.	10 min.

Mittuniversitetet
MID SWEDEN UNIVERSITY

Time Sharing

- Använder multitasking för att hantera flera samtidiga interaktiva jobb
- Processorns tid delas mellan flera användare
- Flera samtidiga användare har tillgång till systemet via terminaler

Mittuniversitetet
MID SWEDEN UNIVERSITY

Batch Multitasking Jämfört Med Time Sharing

	Batch Multitasking	**Time Sharing**
Principal objective	Maximize processor use	Minimize response time
Source of directives to operating system	Job control language commands provided with the job	Commands entered at the terminal

Mittuniversitetet
MID SWEDEN UNIVERSITY

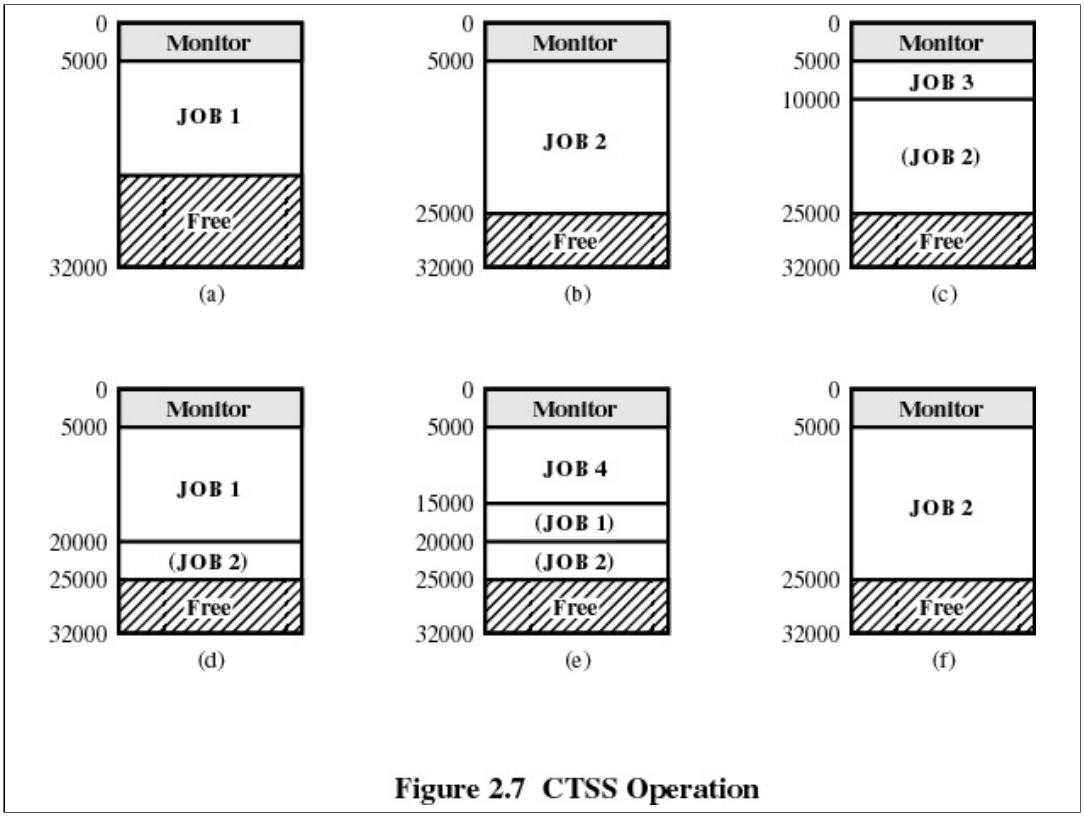


Figure 2.7 CTSS Operation

Bild 24: När maskinerna blir billigare och människorna ”dyrare” prioriterar man annorlunda, från att hålla processorn sysselsatt till att interagera med användare. Dessutom att göra arbetet effektivt för människorna. Man får ännu bättre utnyttjande av processorn vid multiprogrammering.

Compatible Time.sharing System, MIT 1961.

Avbrott 0,2 s swap till disk

Största Fördelarna Med Ett Operativsystem

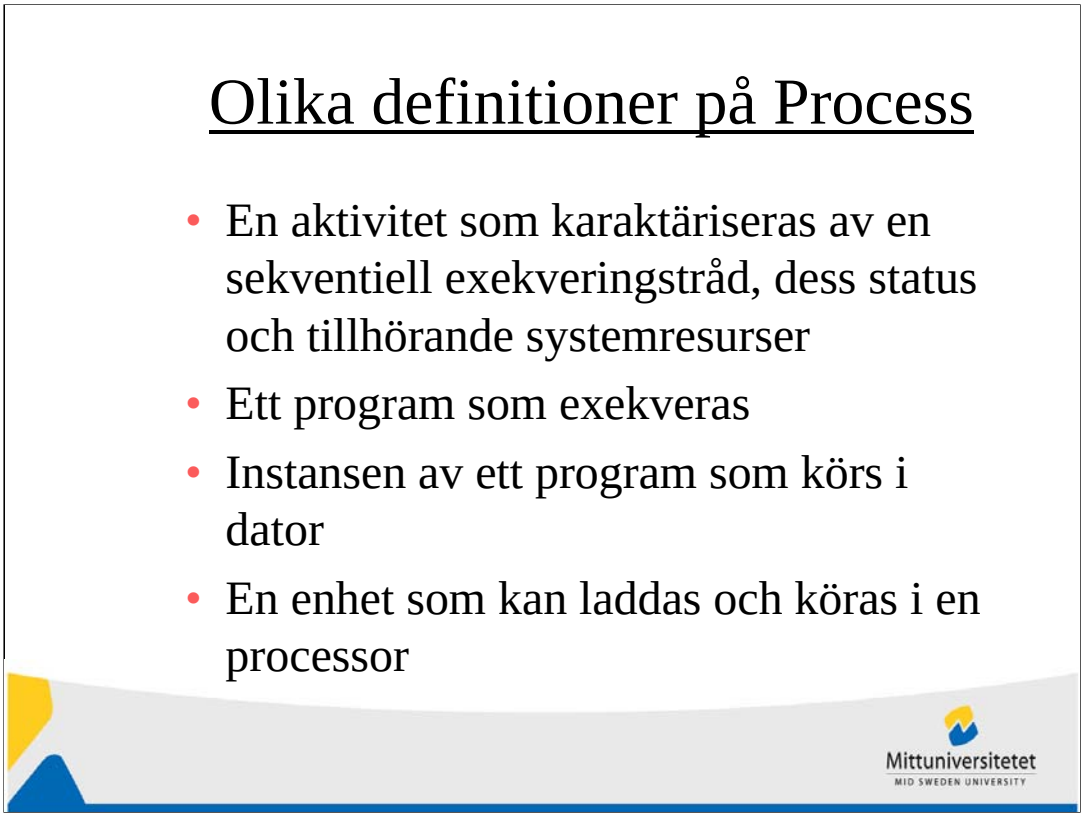
- Processer (1960)
- Minneshantering
- Skydd av data, datasäkerhet
- Schedulering och resurshantering
- Strukturering av systemet

Mittuniversitetet
MID SWEDEN UNIVERSITY

-61 CTSS < 5000 36bitas ord

-75 Multics 20 milj instr

Windows 2000 32 miljoner rader kod



Processer

Bild 26: Processbegreppet kom först 1960. Definition av en process enligt SiS är:

”*En serie av händelser tjänande ett visst ändamål eller främjande visst resultat vid databehandling*”.

Observera att program och process **inte** är samma sak.

Ett program är en statisk följd av instruktioner. Medan en process är den dynamiska aktivitet, inkl. variabler och samtliga kontrollstrukturer, när ett program körs. Processen ändrar hela tiden status

Design Av Operativsystem

- Synkronisering
  - En process som väntar på en I/O-enhet väntar tills den får en signal om att data finns i buffert
- *Mutual exclusion*
  - Två processer kan inte samtidigt få tillgång till samma resurs
- *Nondeterminate program operation*
  - Resultatet från en process får bara bero på indata till processen och inte vilka övriga processer som körs samtidigt
- *Deadlocks*

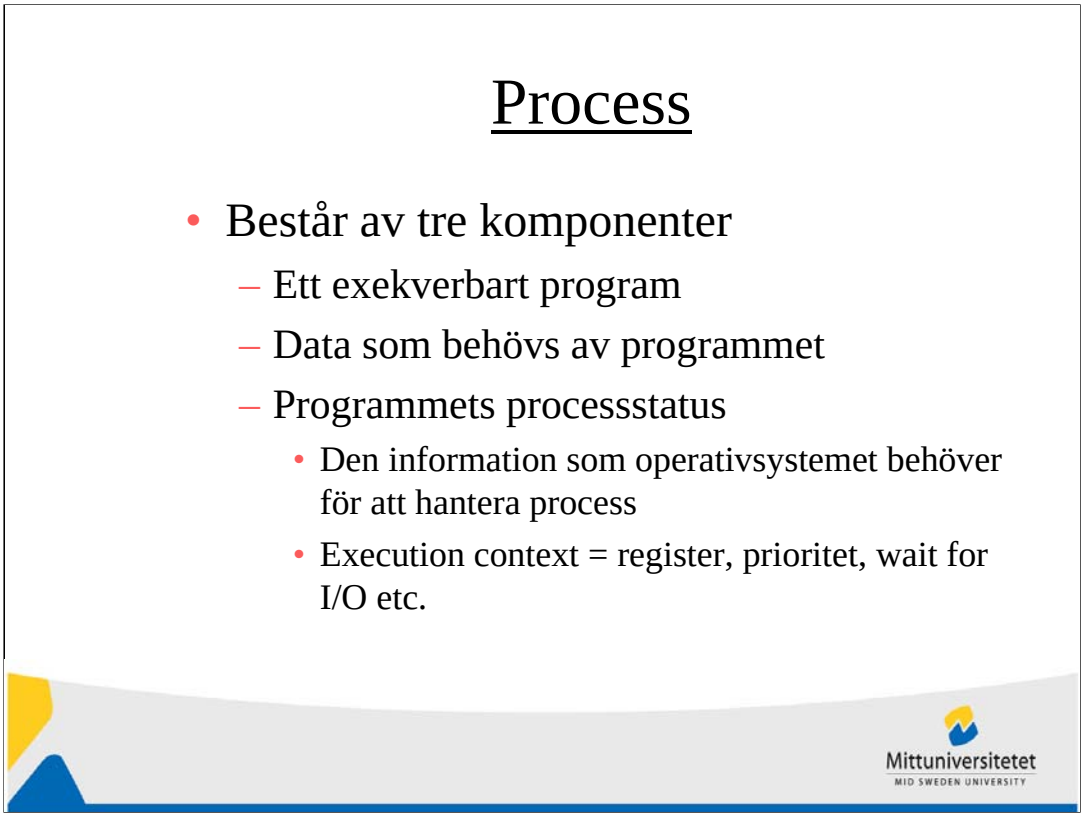
Mittuniversitetet
MID SWEDEN UNIVERSITY

Mutual Exclusion

Bild 27: Två processer A och B, som exekverar ”samtidigt” försöker ändra värdet på en variabel, (som från början har värdet 10). A ska lägga till 3 och B ska dra ifrån 5. Båda hämtar värdet 10, men sedan beror det på vem som sparar sist om slutvärdet blir 13 eller 5.

Alltså kan man inte tillåta att flera processer okontrollerat jobbar med samma data samtidigt.

Mer om detta i senare kapitel.



Execution Context / Context Switch

Bild 28, 29: Programmets processtatus vid en given tidpunkt kallas **Execution Context**

och när en process lämnar plats för en annan sker ett Context Switch.

Notera att varje process har sitt eget minnesområde innehållande context, kod och data.

Process

- Varje process har sitt eget minnesområde
- En övre och nedre gräns, så att OS kan kontrollera otillåten access
- Anrop till adress i annan process ska generera felmeddelande

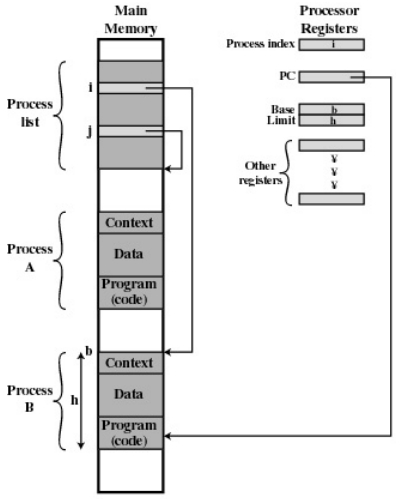
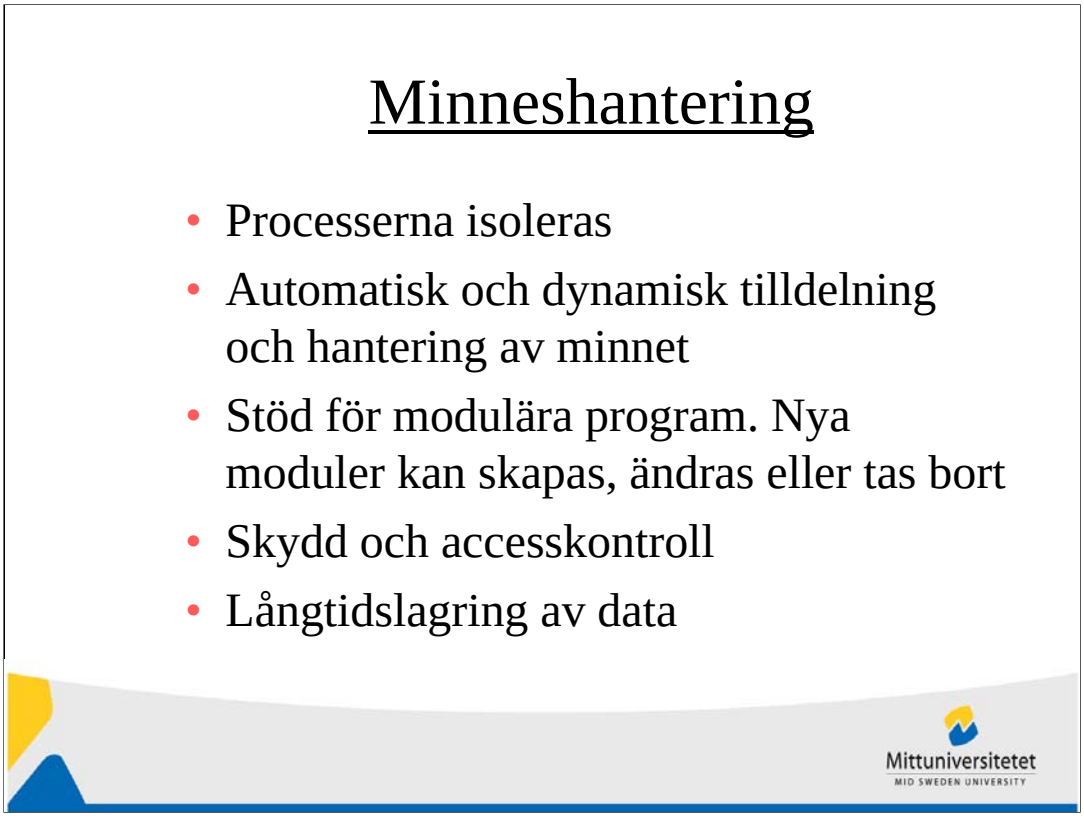


Figure 2.8 Typical Process Implementation

Mittuniversitetet
MID SWEDEN UNIVERSITY

Kan ni se vilken process exekverar för ögonblicket?



Minneshantering

Bild 30: Minneshantering sker dynamiskt. Program kan begära mer minne efter hand.

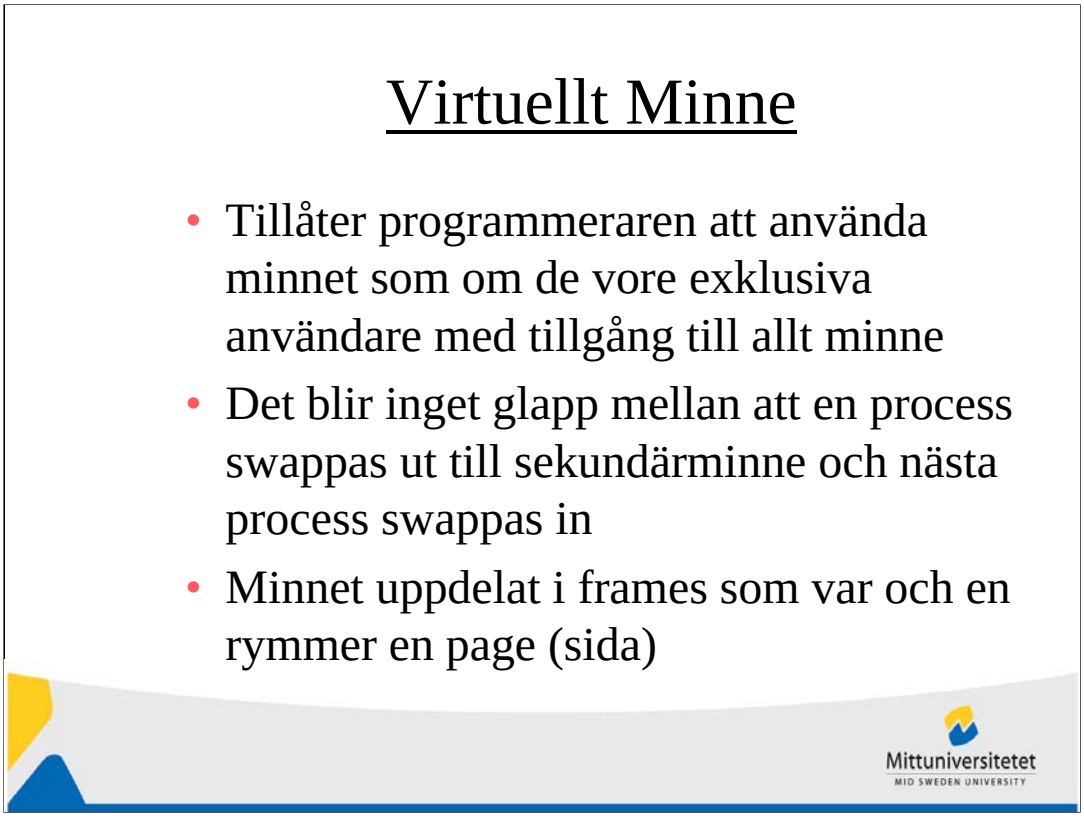


Bild 31-: Grunderna för virtuellt minne. En process består av ett antal sidor (pages).

Endast de sidor som för ögonblicket behövs, läses in i primärminnet. Minnet är uppdelat i ”frames” med samma storlek som en sida.

Sidorna läses in i ledig frame. Om ingen ledig finns skrivs en upptagen över.

En process vet inte från början var (i vilken frame) sidorna kommer att ligga. Varje process behöver därför en lista som visar om en sida finns i

minne och i så fall i vilken frame.

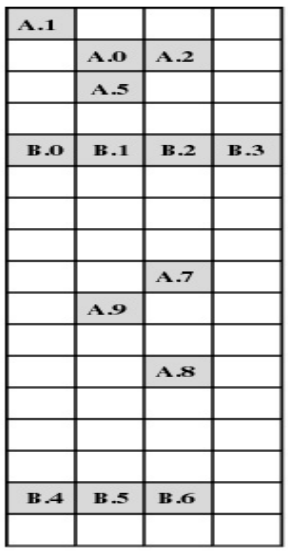
Filsystem

- Möjliggör långtidslagring
- Data sparas som filobjekt

Paging

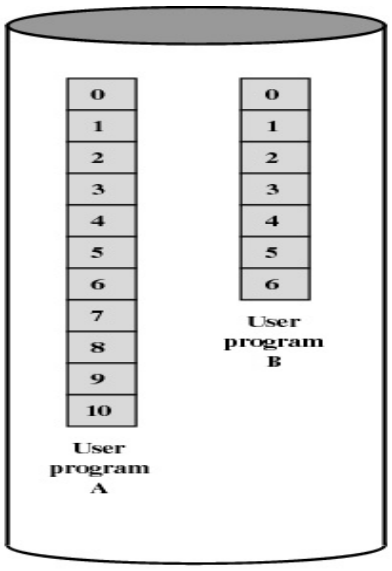
- Processer kommer att ligga på ett antal “sidor”(pages) med fast storlek
- Virtuell adress består av ett sidnummer och en offset inom sidan
- Sidorna kan ligga var som helst i primärminnet
- *Real adress*-fysisk adress i primärminnet

Mittuniversitetet
MID SWEDEN UNIVERSITY



Main Memory

Main memory consists of a number of fixed-length frames, equal to the size of a page. For a program to execute, some or all of its pages must be in main memory.



Disk

Secondary memory (disk) can hold many fixed-length pages. A user program consists of some number of pages. Pages for all programs plus the operating system are on disk, as are files.

Figure 2.9 Virtual Memory Concepts

Adresser i Virtuellt Minne

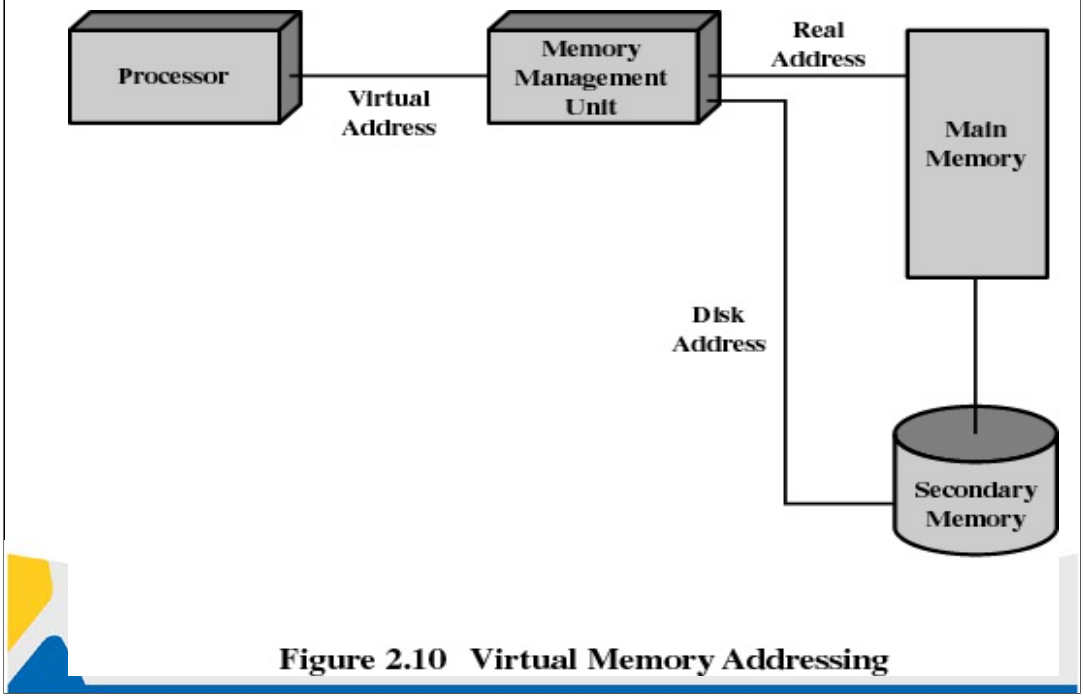


Figure 2.10 Virtual Memory Addressing

Memory Management Unit

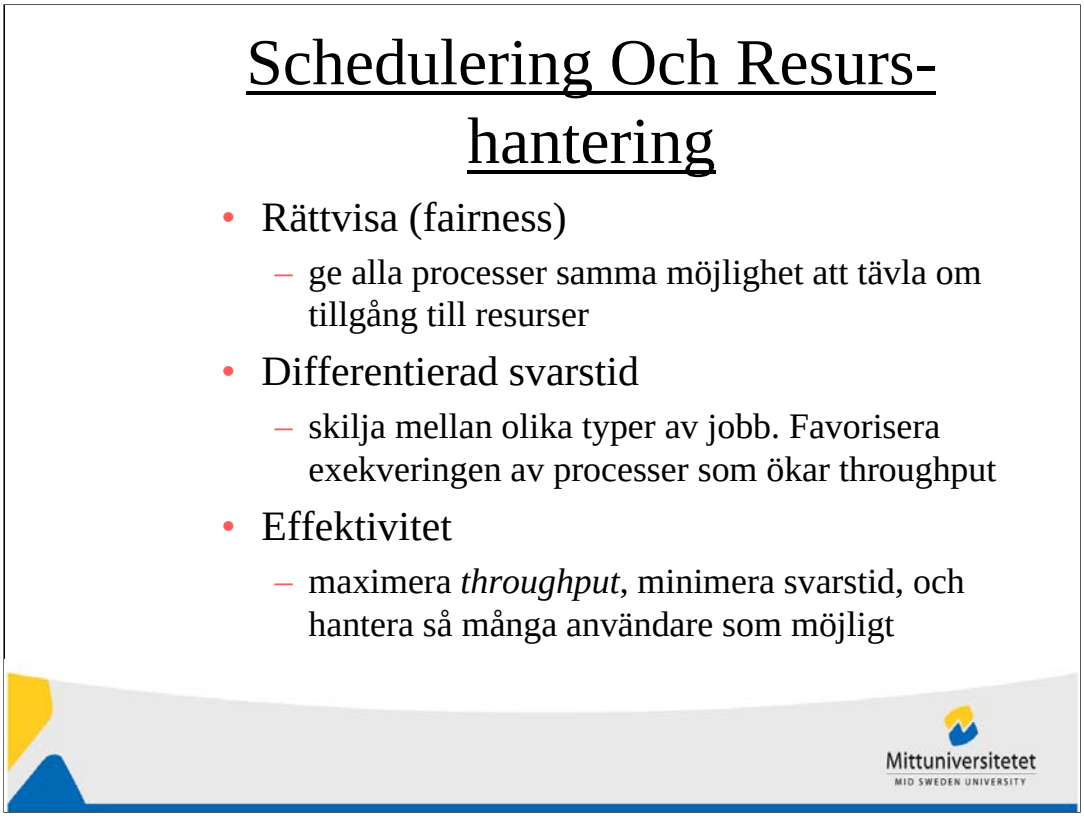
MMU översätter från virtuell adress (önskad sida) till var den ligger i minnet. Finns den

inte i minnet måste den först läsas från disk

Skydd För Data Och Datasäkerhet

- Accesskontroll
  - Reglerar användares access till systemet
- Flödeskontroll
  - Reglerar dataflödet inom systemet och till/från användare
- Certifiering
  - Garanterar att access och flödeskontroll sker enligt specifikationer

Mittuniversitetet
MID SWEDEN UNIVERSITY



Schedulering

Bild 37: shedulering är alltså den funktion som väljer vilken process som ska exekveras

Operativsystemets huvuddelar

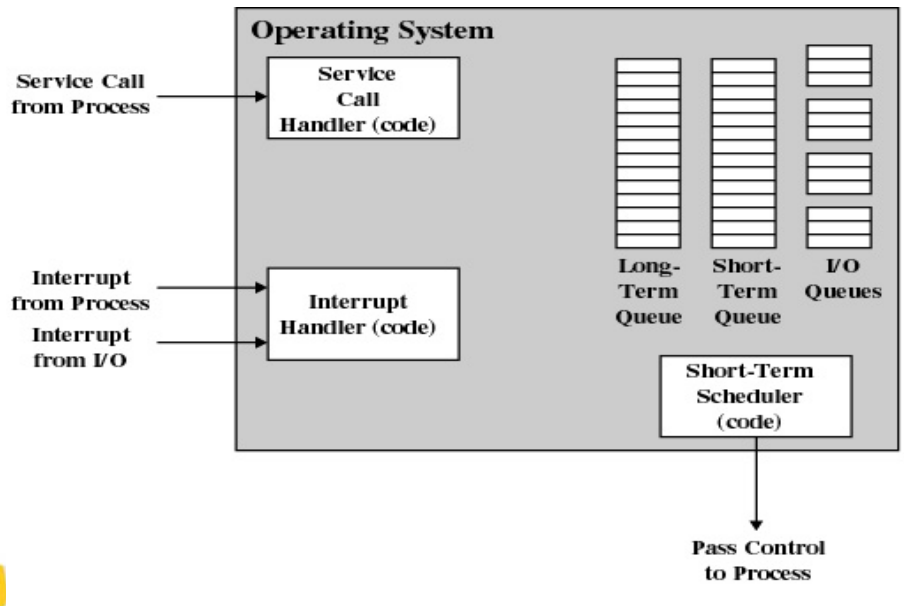


Figure 2.11 Key Elements of an Operating System for Multiprogramming

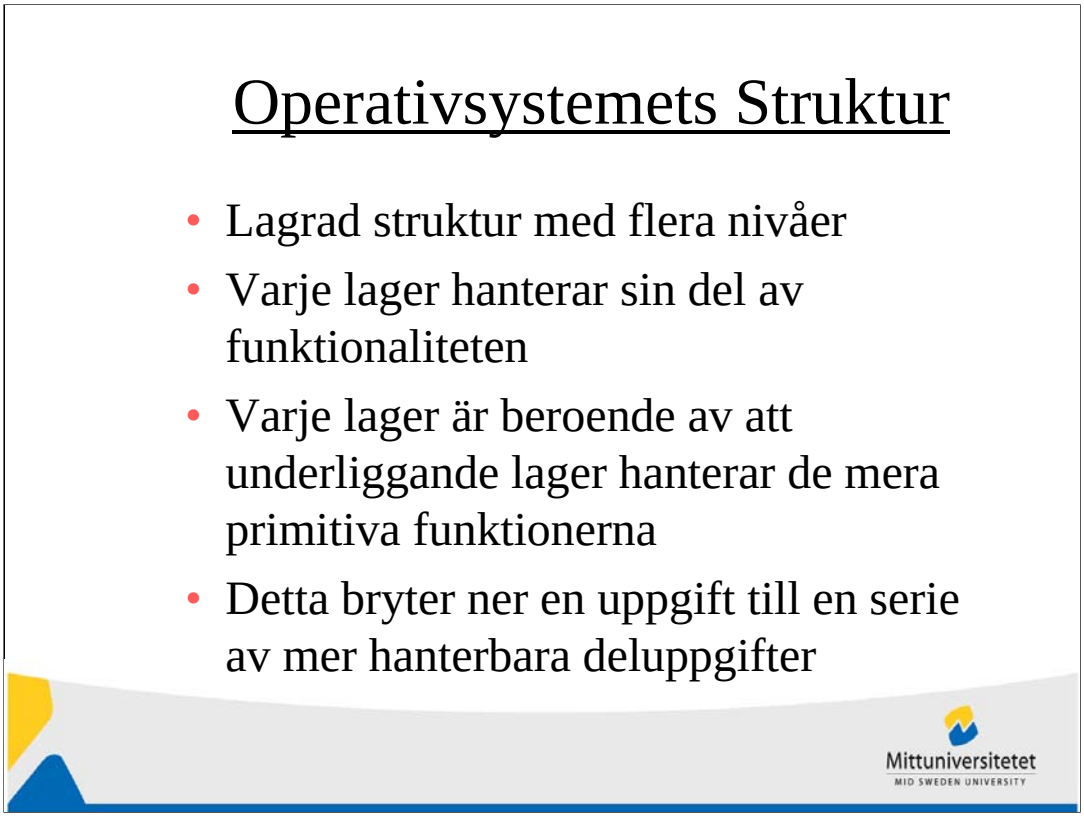


Bild 39-: Observera den hierarkiska strukturen.

Operating System Design Hierarchy

Level	Name	Objects	Example Operations
13	Shell	User programming environment	Statements in shell language
12	User processes	User processes	Quit, kill, suspend, resume
11	Directories	Directories	Create, destroy, attach, detach, search, list
10	Devices	External devices, such as printer, displays and keyboards	Open, close, read, write
9	File system	Files	Create, destroy, open, close read, write
8	Communications	Pipes	Create, destroy, open. close, read, write

Mittuniversitetet
MID SWEDEN UNIVERSITY

Operating System Design Hierarchy

Level	Name	Objects	Example Operations
7	Virtual Memory	Segments, pages	Read, write, fetch
6	Local secondary store	Blocks of data, device channels	Read, write, allocate, free
5	Primitive processes	Primitive process, semaphores, ready list	Suspend, resume, wait, signal

Mittuniversitetet
MID SWEDEN UNIVERSITY

Operating System Design Hierarchy

Level	Name	Objects	Example Operations
4	Interrupts retry programs	Interrupt-handling	Invoke, mask, unmask,
3	Procedures	Procedures, call stack, display	Mark stack, call, return
2	Instruction Set	Evaluation stack, micro-program interpreter, scalar and array data	Load, store, add, subtract branch
1	Electronic circuits	Registers, gates, buses, etc.	Clear, transfer, activate, complement

Mittuniversitetet
MID SWEDEN UNIVERSITY

Moderna Operativsystem

- *Microkernel*
  - Kerneln innehåller endast ett fåtal grundläggande funktioner
    - adresshantering
    - *interprocess Communication (IPC)*
    - Grundläggande schedulering

Mittuniversitetet
MID SWEDEN UNIVERSITY

Ett stort opsys som innehåller allt i en process

I stället liten kernel och processer som ger tjänster. Servers kan anpassa efter behov.

Ger enklare kernel och flexiblare OS

Moderna Operativsystem

- Trådat
  - process är uppdelad i trådar *(threads)* som kan exekvera samtidigt
- En tråd
  - exekverbar arbetsenhet
  - exekveras sekventiellt och kan avbrytas
- En process består av en eller flera trådar

Mittuniversitetet
MID SWEDEN UNIVERSITY

Moderna Operativsystem

- Symmetriskt multiprocessorsystem
  - Hanterar flera processorer
  - Alla processorer delar samma primärminne och I/O-funktioner
  - Alla processorer kan utföra samma funktioner

Moderna Operativsystem

- Distribuerade operativsystem
  - ger illusionen av ett enda primärminne och ett primärminne
  - används för distribuerade filsystem

Moderna Operativsystem

- Objektorienterad design
  - Ger möjlighet att addera tilläggsmoduler till en liten kernel
  - Ger programmerare möjlighet att anpassa operativsystemet utan att störa systemet

Mittuniversitetet
MID SWEDEN UNIVERSITY

Windows 2000

- Utnyttjar möjligheterna i en modern 32-bitars processor
- Ger äkta multitasking i en enanvändar-miljö
- Stöder Klient/Serverarbete

Windows 2000 Arkitektur

- Modulär struktur ger flexibilitet
- Kan köras på olika hårdvaruplattformar
- Stöder applikationer skrivna för olika andra operativsystem

Mittuniversitetet
MID SWEDEN UNIVERSITY

Uppbyggnad

- Modifierad kernelarkitektur
  - Inte ren microkernel
  - Många systemfunktioner ligger utanför microkernel, men körs i *kernelmode*
- Varje modul kan tas bort, uppgraderas, eller bytas ut utan att skriva om hela systemet

Mittuniversitetet
MID SWEDEN UNIVERSITY

User mode - kernel mode.

Bild 50-56: Kernel mode är ett prioriterat läge för operativsystemet. Applikationer exekverar i usermode. Gäller alla OS.

Lagrad Struktur

- *Hardware abstraction layer (HAL)*
  - Isolerar operativsystemet från plattforms-specifika hårdvaruskillnader
- Microkerneln
  - Mest använda och mest fundamentala komponenten i operativsystemet
- *Device drivers*
  - Översätter användarens I/O-funktionsanrop till specifik hårdvaruanrop

Mittuniversitetet
MID SWEDEN UNIVERSITY

W2K *Executive* Hanterar:

- I/O
- Objekt
- Security reference monitor
- Processer och trådar
- Local procedure call (LPC) Facility
- Virtuellt minne
- Cache
- Windows/grafikmoduler

Mittuniversitetet
MID SWEDEN UNIVERSITY

Användarprocesser

- Speciella systemprocesser
  - Ex: logon och sessionshantering
- Serverprocesser
- Environment subsystems
- Applikationer för olika system
  - Win32, Posix, OS/2, Win. 3.1, MS-DOS

Mittuniversitetet
MID SWEDEN UNIVERSITY

Klient/Servermodel

- Förenklar *Executive*
  - Kan hantera olika och nya APIer
- Ökar tillförlitligheten
  - varje service körs som en separat process med sitt eget minnesområde
  - klienterna kan inte accessa hårdvaran direkt
- Enhetligt gränssnitt för kommunikation mellan program m h a LPC (Lightweight Process Communikation)
- Ger möjlighet till distribuerad computing

Mittuniversitetet
MID SWEDEN UNIVERSITY

Trådar och SMP (Symmetric Multiprocessing)

- Flera trådar kan köras samtidigt på olika processorer
- Multipla trådar i en enda process kan exekveras på olika processorer samtidigt
- Serverprocesser kan använda flera trådar
- Delar data och resurser mellan processer

Mittuniversitetet
MID SWEDEN UNIVERSITY

Symetrisk MultiProcessing

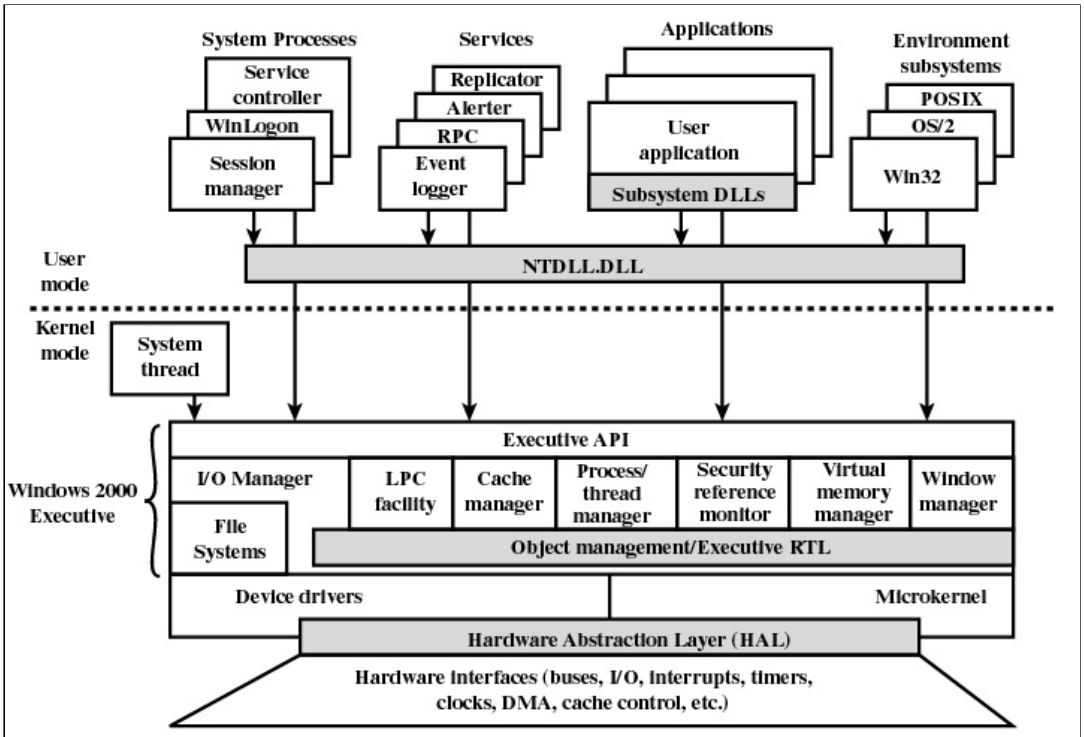


Figure 2.13 Windows 2000 Architecture

UNIX

- Hårdvaran bäddas in i operativsystemet
- Operativsystemet kallas kernel
- Till den kommer ett stort antal *services* och *interfaces*
  - *shell*
  - C-kompilator

Mittuniversitetet
MID SWEDEN UNIVERSITY

UNIX

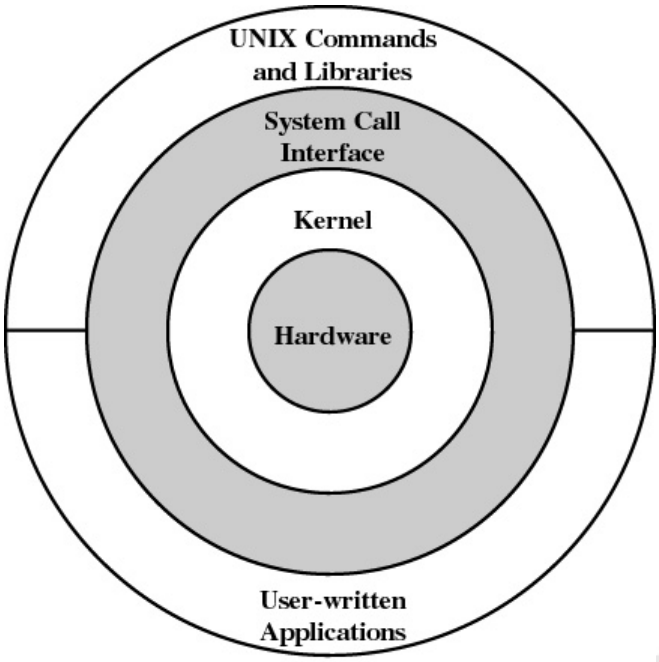


Figure 2.15 General UNIX Architecture

Moderna UNIX-system

- System V Release 4 (SVR4)
- Solaris 2.x
- 4.4BSD
- Linux