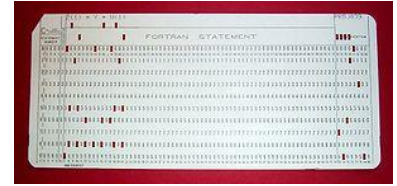


Operativsystem

Historia
(1.2 i boken)

1940-talet



- Inga operativsystem
- Programmering via mekaniska brytare, bit för bit, i maskinkod
- Senare kom hålkort in i bilden, programmen matades via hålkort
- Senare utvecklades assembler, med engelsk-liknande förkortningar
- 2:a världskriget hade stor inverkan på utveckling av datamaskiner
 - Operationsanalys
 - Beräkningar för kärnfysik
- transistorn uppfanns 1947!
- Howard Aiken, John von Neuman, Konrad Zuse...

1950-talet

- GM Research Laboratories utvecklade tidigt på 50-tal ett operativsystem för IBM 701, första??
- Tidiga system körde ett "jobb" åt gången, med operatör inblandad
 - Behov av "batch"-system
- Batch-system
 - Jobben lästes in på t.ex. magnetisk media, varefter jobben kördes i en serie (batch)
- Programmerare var ofta fortfarande tvungna att själva kontrollera systemets resurser

1960-talet

- Batch-system, men man använder systemet mer effektivt genom körning av flere job "samtidigt"
 - I/O-intensiva jobb
 - CPU-intensiva jobb
- Kallas även multiprogrammerbara system
 - "Degree of multiprogramming" – hur många job som samtidigt kan hanteras
- IBM S/360 - 1964
 - Hårdvarukompatibel
 - OS/360 – operativsystem för kompatibilitet

1960-talet

- Flera samtida användare
 - CTSS (Compatible Time-Sharing System) - MIT
 - Project MULTICS - MIT, GE, Bell Labs
 - Misslyckades, mycket p.g.a. dålig marknadsföring
 - Dock stort inflytande på dagens operativsystem
 - Språk för UNIX: BCPL, B, C
 - Bl.a. UNIX bottnade i MULTICS
 - MINIX (Tanenbaum)
 - » Linux
- Kostnader
 - CPU-timme på mainframe: \$500
 - Timlön för programmerare: \$4

1970-talet

- Tidsdelande system (timesharing) börjar bli standard
- tidiga persondatorer
- TCP/IP-standarden börjar bli allmänt använd
- LAN kommer för nätverk
- Enkryptering börjar bli relevant

1980-talet

- Persondatorer (PC)
 - IBM 1981
 - Macintosh 1984
- GUI i operativsystemet
 - X-windows (MIT)
 - Macintosh
- Distribuerade system – klienter och servrar

1990-talet

- WWW
 - ”Distribuerade system”
- Nätverksstöd blev standard i operativsystem
- Användarvänligheten ökar
- Objekt-orienterade operativsystem
- Open-source-rörelsen

2000-talet

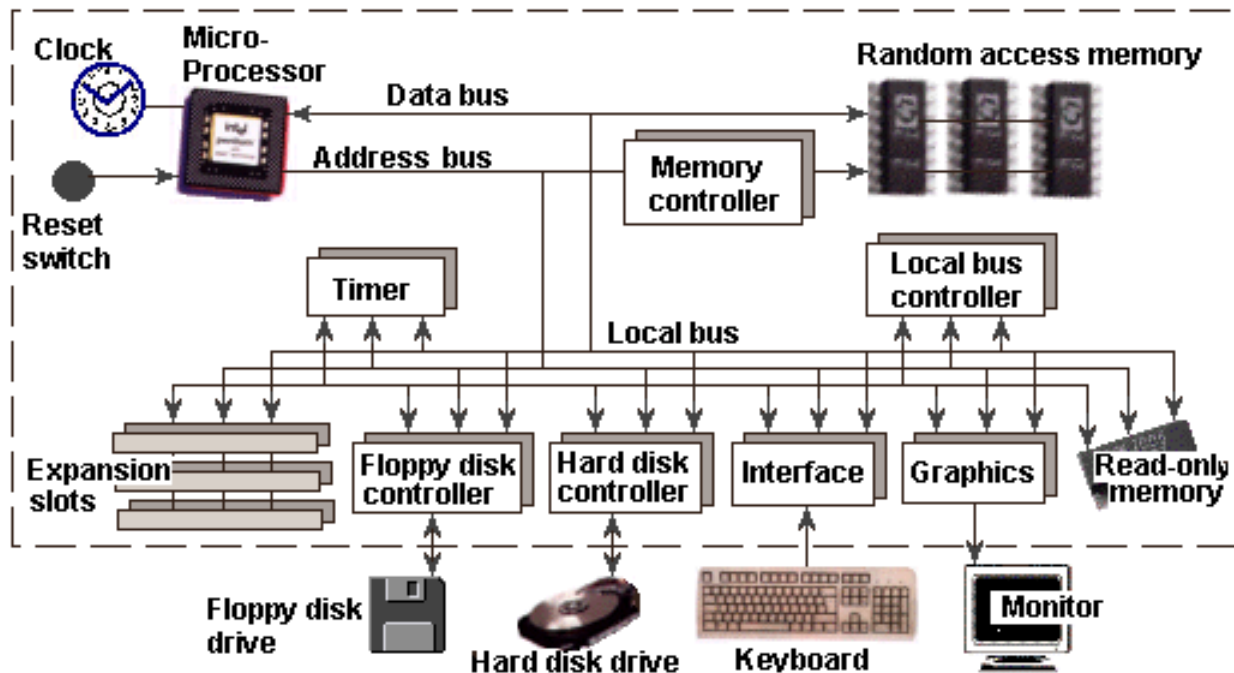
- Middleware: länkar två applikationer
 - T.ex. mellan web-server och databassystemet
- Distribuerade system
 - T.ex. virtuella butiker på nätet
 - Förmedlar varor från något annat system
- Standardiserade användargränssnitt i operativsystem
- Operativsystem kliver in i många nya apparater, såsom set-top-boxar, mobiltelefoner, bilar, routrar, basstationer

Operativsystem

Hårdvara
(1.4 i boken)

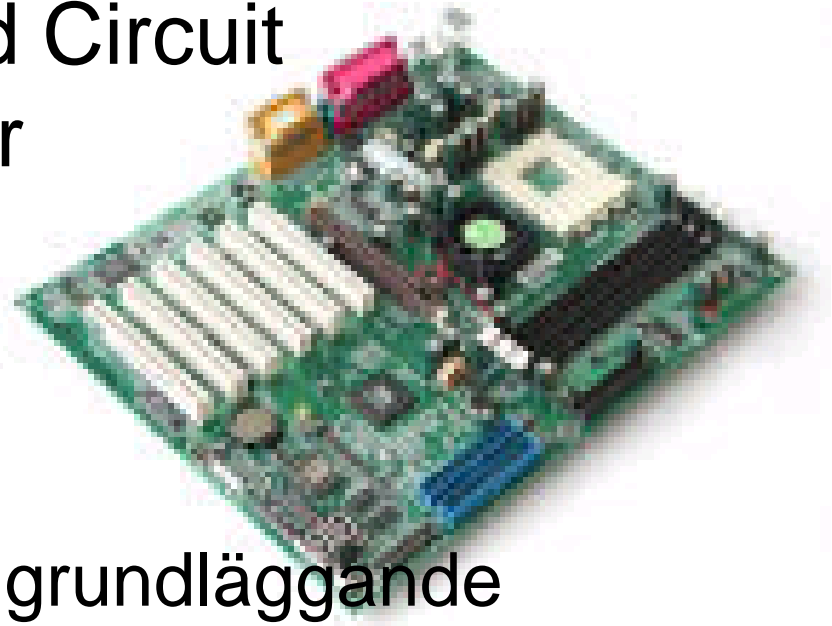
Hårdvaruöversikt

- Generellt: CPU:n, minnet och I/O-enheter är uppkopplade till varandra via en buss (t.ex. PCI)



Moderkort

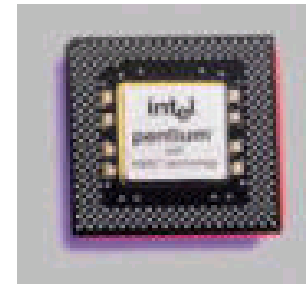
- Centrala PCB (Printed Circuit Board) i en bordsdator
- Flere chips, bl.a. för BIOS (Basic Input Output System)
 - Chipset (innehållande grundläggande arkitekturkomponenter)
 - Front Side Bus, avbrottshantering, minnesstyrning, nätverksadapter, hårddiskadapter, ljud, grafik



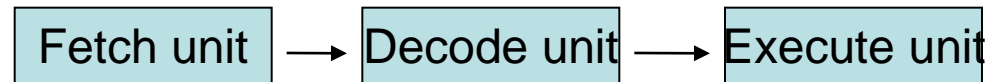
Busstyper

- En buss är en samling elektriska kopplingar
 - databuss
 - adressbuss
- Front Side Bus (FSB)
 - Kopplar bl.a. processorn till minnet
- Peripheral Component Interconnect (PCI) buss
 - Hårdvaruenheter, såsom t.ex. ljudkort
- Accelerated Graphics Port (AGP)
 - För att koppla grafikkort till systemet

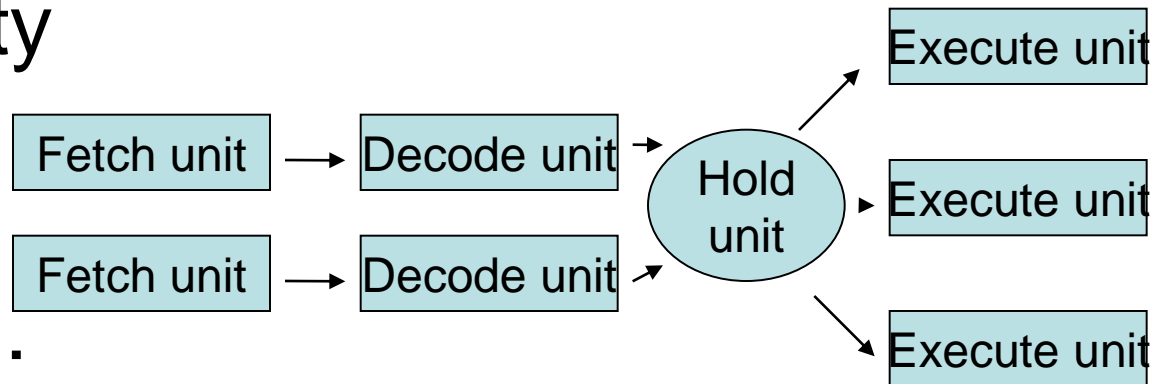
CPU



- Register: PC, SP, generella register, PSW
- Pipeline

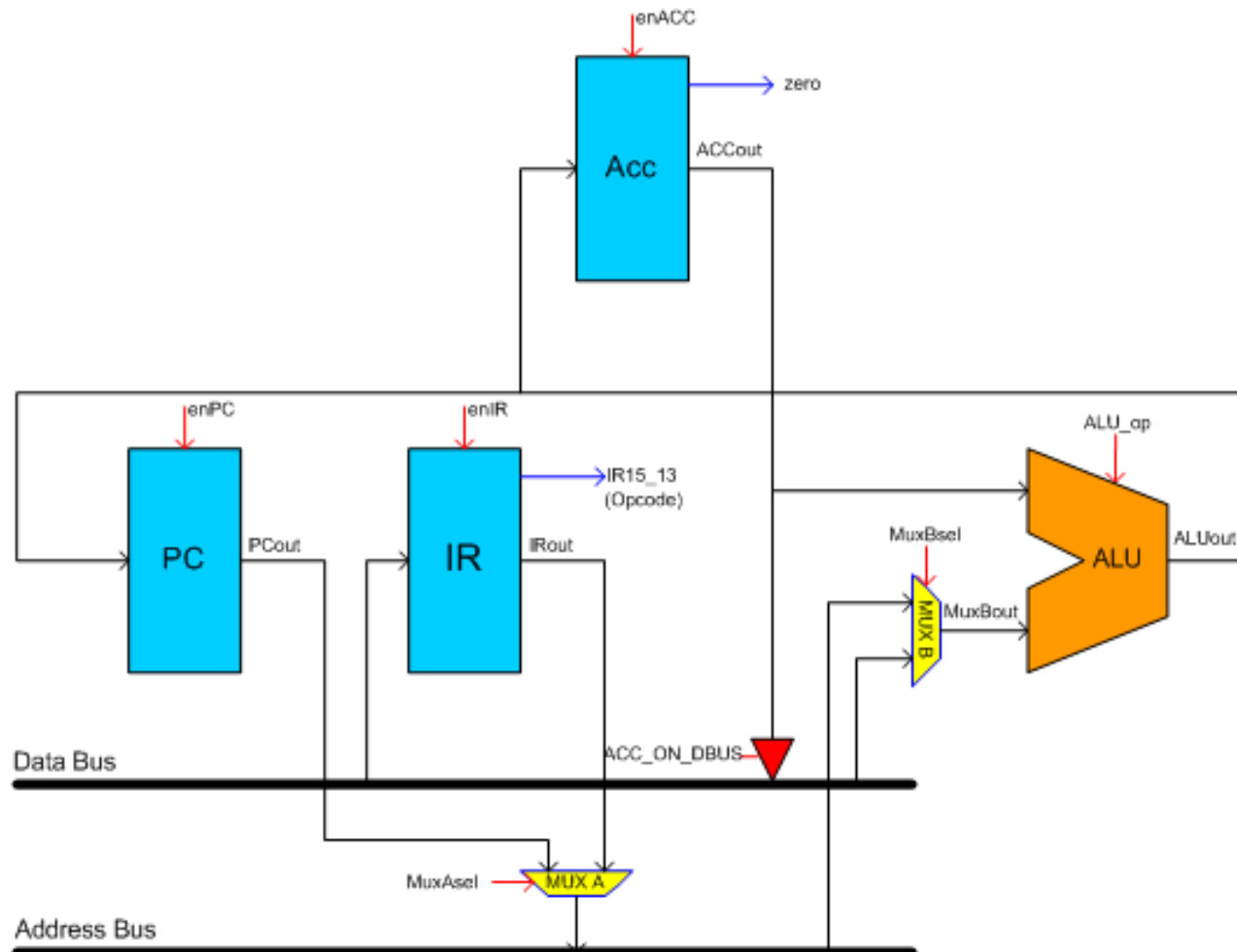


- Superscalarity



- CPU tillstånd:
 - Minst 2: Kernel mode, user mode

CPU - simplifierat



Minnestyper

	Söktid	Storlek (typiskt)	Pris (€/GB)
Register	1 ns	1 KB	???
Cache	2 ns	1 MB	50000
Primärminne	10 ns	4GB-16GB	10
Magnetisk skiva	10 ms	1 TB	0,10
SSD skiva	0,1 ms	50-200 GB	1
Optisk skiva	500 ms	5 GB	0,20

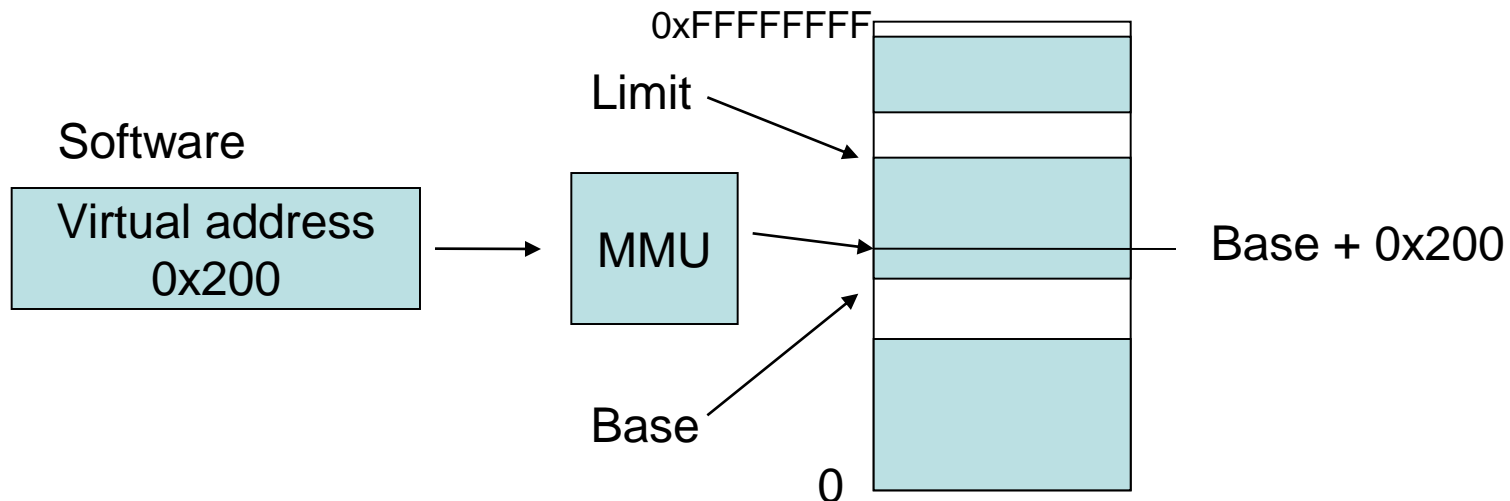
Typer av primärminne:

RAM: SIMM, DIMM, CMOS

ROM: EPROM, EEPROM

Primärminne

- Datorsystemet (dvs. hårdvara + OS) måste hantera multiplexing i “rymden”:
 - 1. Hur skydda program från varandra
 - 2. Hur se till att program kan flyttas i minnet



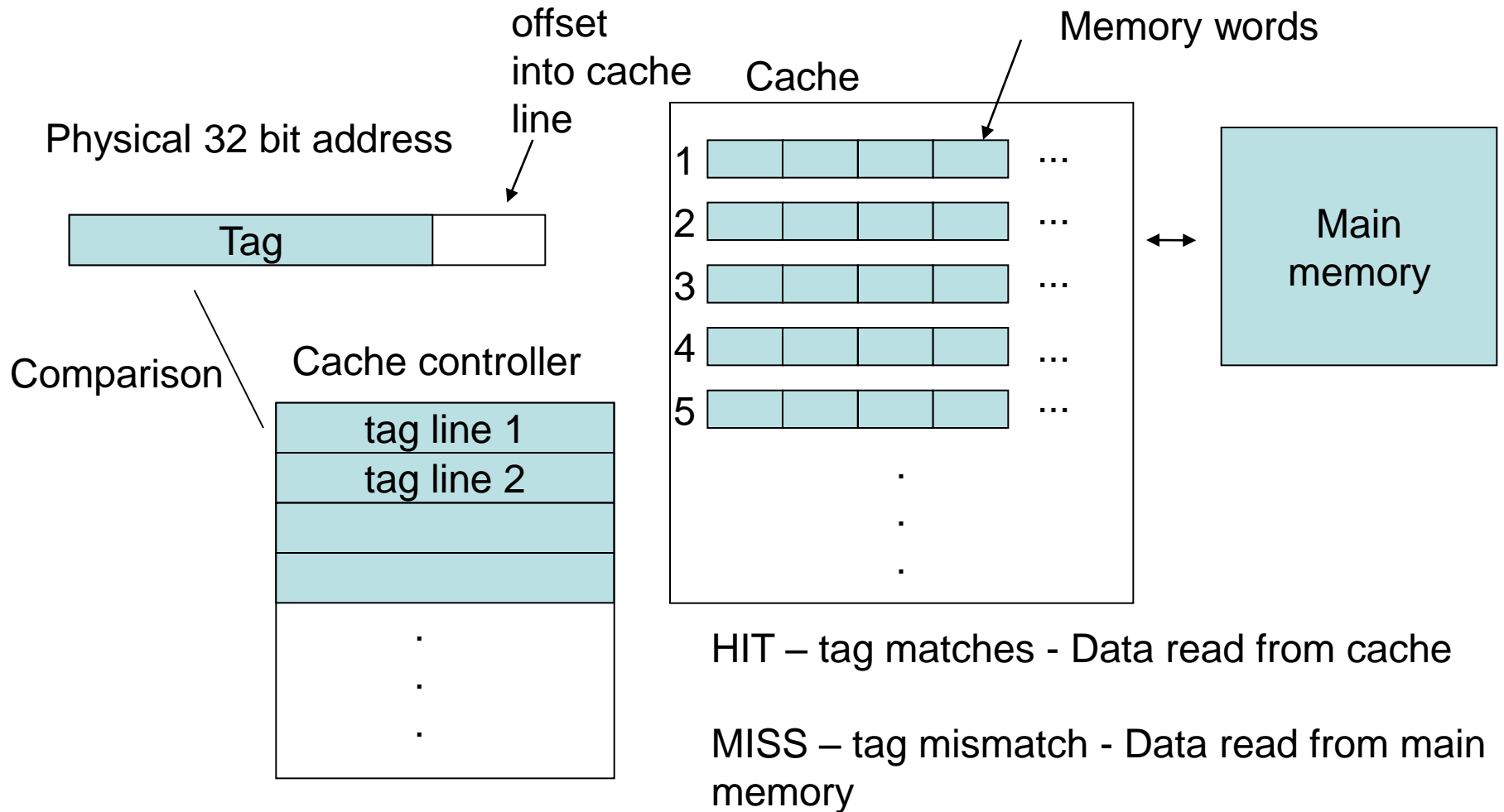
Hårdvaru-cache

- CPU är mycket snabbare än primärminne
 - CPU väntar många klockcykler på data
- Lokala egenskaper vid minnesaccess:
 - Framtida minnesreferenser sker ofta i närheten av de nuvarande (t.ex. programkod eller räckor med data)
- ✂ → Mindre och snabbare minnesenheter innehållande senast använda kod/data
- Cache-linjer innehållande senast använda bytes (och bytes i närheten av dessa)

Typer av hårdvaru-cache

- Fully mapped – fullt mappad
 - En rad från primärminnet sparas alltid på samma stället i cache-minnet
- Fully associative – fullt associativ
 - En rad från primärminnet kan sparas varsomhelst i cache-minnet
- N-way associative – N-vägs associativ (det normala)
 - En rad i primärminnet kan sparas i någon av N stycken cacherader (t.ex. 2-vägs cache → en rad i primärminnet kan sparas i två alternativa rader i cache-minnet)

Hårdvaru-cache i praktiken



DMA

- Direct Memory Access
- De flesta I/O-operationer transporterar data mellan centralminnet och I/O-enheter
 - Detta sköttes länge genom det som kallas Programmet I/O (PIO)
 - Processorn måste se till att varje byte förflyttas från minne till I/O-enhet
- DMA tillåter direkt kopiering av data från I/O-enhet till minnet, utan intervention från processorn

I/O-enheter

- Skivenhet (IDE, EIDE, SCSI (Ultra320: 320 Mb/s), SATA I: 1,5 Gb/s, SATA II 3 Gb/s)
 - (E)IDE – Enhanced Integrated Drive Electronics
 - SCSI – Small Computer System Interface
 - (S)ATA – (Serial) Advanced Technology Attachment
- (Floppy)
- Skärm
- Tangentbord, mus,
- Seriell (115 kb/s), Parallell
- USB (1.1 (“Full speed”): <12 Mb/s, 2.0 (“High speed”): 480 Mb/s, 3.0 specs (Aug 2008): 4.8 Gbit/s)
- Firewire (400, 800 Mb/s, nya versioner 1.6 Gbit/s, 3.2 Gbit/s)
- Nätverk (Ethernet 1 Gbit/s, 10 Gbit/s, (andra))

Hårdvarustöd för OS

- För att kunna implementera OS bra, behövs stöd från hårdvaran:
- Processor:
 - User mode, Kernel mode, privileged instructions
- Skyddning samt hantering av minnet
 - Gränser för minnesområdet, virtuellt minne
- Avbrott och exeptioner
 - I/O-enheter kan "meddela" systemet att något har hänt
- Timers och klockor
 - För att t.ex. med jämna mellanrum skapa avbrott
- Bootstrapping
 - System i BIOS för att överhuvudtaget få något program att snurra i en dator

Maskinspråk, assembler

- En dator (läs CPU) förstår enbart maskinspråk
 - Processorspecifikt
 - I grunden 1:or och 0:or:
 - 1300042774
1400593419
1200274027
- Assembler (assembly language)
 - Engelsk-liknande förkortningar motsvarande maskinspråket
 - LOAD BASEPAY
ADD OVERPAY
STORE GROSSPAY

Tolkar och kompilatorer

- Maskinspråk / assembler tar lång tid att skriva
 - Högnivå-språk utvecklas (t.ex. C/Java/Visual Basic, C++...)
- Typer av högnivåspråk
 - Tolkade språk (Visual Basic, Python, PHP)
 - Tolkas i samband med exekveringen till mer primitiva byggstenar
 - Kompilerade språk (C/C++/Fortran)
 - Översätts på förhand med hjälp av kompilator till maskinspråk
 - Byte-kod (Java)
 - Kompilerat språk, där "maskinspråket" är Javas virtuella maskin (JVM)
 - Varje byte-kod-instruktion måste dock mappas till den aktuella hårdvaran

Kompilering & Länkning

- Kompilering
 1. Analys av källkod -> tokens
 2. Parser -> abstrakt träd av konstruktioner
 3. Generering av mellankod -> assembler
 4. Kodoptimering -> assembler
 5. Kodgenererare -> maskinkod
- Länkning
 - Integrering av maskinkod från olika bibliotek till en exekverbar helhet

Firmware

- Ofta skrivet som mikroprogrammering, dvs ett lager under det egentliga maskinspråket
- Enkla, fundamentala instruktioner för att implementera själva maskinspråket
- Även den mjukvara som är resident i ett system (t.ex. digitalkamera / printer / nätverksenhet)

Operativsystem

Modeller

(1.7 i boken)

Monolitisk arkitektur

- Varje del är innesluten i kerneln och kan direkt kommunicera med andra delar
 - T.ex: filsystemet kan direkt kalla på minnesallokatorn
- Kerneln körs i obegränsat (CPU) tillstånd:
 - Alla operationer är tillåtna (och felaktiga är ödesdigra för systemet...)
- OS/360, VMS, Linux kan kategoriseras som monolitiska operativsystem

Arkitektur i lager

- Komponenter som utför liknande funktioner läggs i lager
- Lager på lägre nivå erbjuder service för lager på högre nivå
- Gör systemet mer modulärt – klarare design
- Sämre prestanda jämfört med ett monolitiskt system

Mikrokernelarkitektur

- Själva kärnan erbjuder mycket begränsad mängd service
 - minneshantering
 - IPC
 - processsynkronisering
- Annan service är utanför kärnan
 - processhantering
 - nätverk
 - filsystem
- Mycket modulära
- Intermodulär kommunikation – minskad prestanda i systemet

Virtuella maskiner

- Ett tidsdelande (timesharing system) kan erbjuda en exakt kopia av den existerande hårdvaran
 - De olika virtuella maskinerna kan köra olika operativsystem
 - Systemanrop fångas först av det egna operativsystemet, som sedan när det i sin tur utför ett kommando, fångas av den virtuella maskinen
 - Exempel: VMware (www.vmware.com), Xen (www.xen.org), VirtualBox

Nätbaserade distribuerade system

- Processer kan accessera resurser som finns på andra oberoende datorer i ett nätverk
- Ofta klient/server-modellen
- Exempel: Sun NFS

Monolitisk vs. microkernel

- Monolitisk

Fördelar	Nackdelar
<ul style="list-style-type: none">• Prestanda	<ul style="list-style-type: none">• Mera utsatt• Statisk

- Microkernel

Fördelar	Nackdelar
<ul style="list-style-type: none">• Modularitet• Klar design	<ul style="list-style-type: none">• Prestanda

Men...

- WinNT (ursprungligen microkernel design)
 - Rör sig mot monolitisk (grafikprestanda)
- Linux (monolithic design)
 - Tar i bruk funktioner som gör kernelen mer modulär, och viss mån liknar en microkernel
 - Kerneltrådar (servers)
 - Kernelmoduler (kernel modularity)