Od procesora do high level programskih jezika: Kako gledamo klipove mačaka na jutjubu?

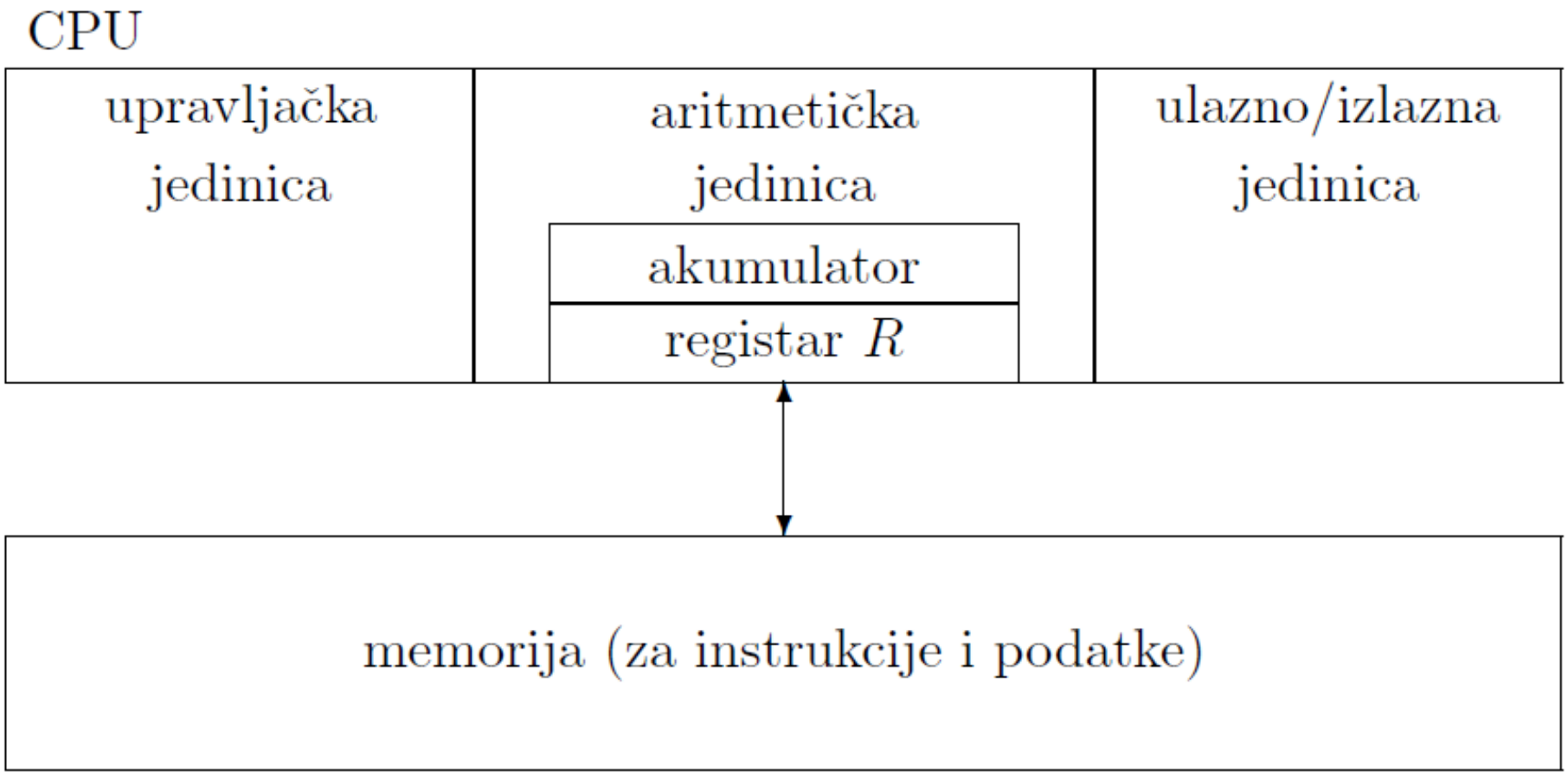
Lazar Mitrović

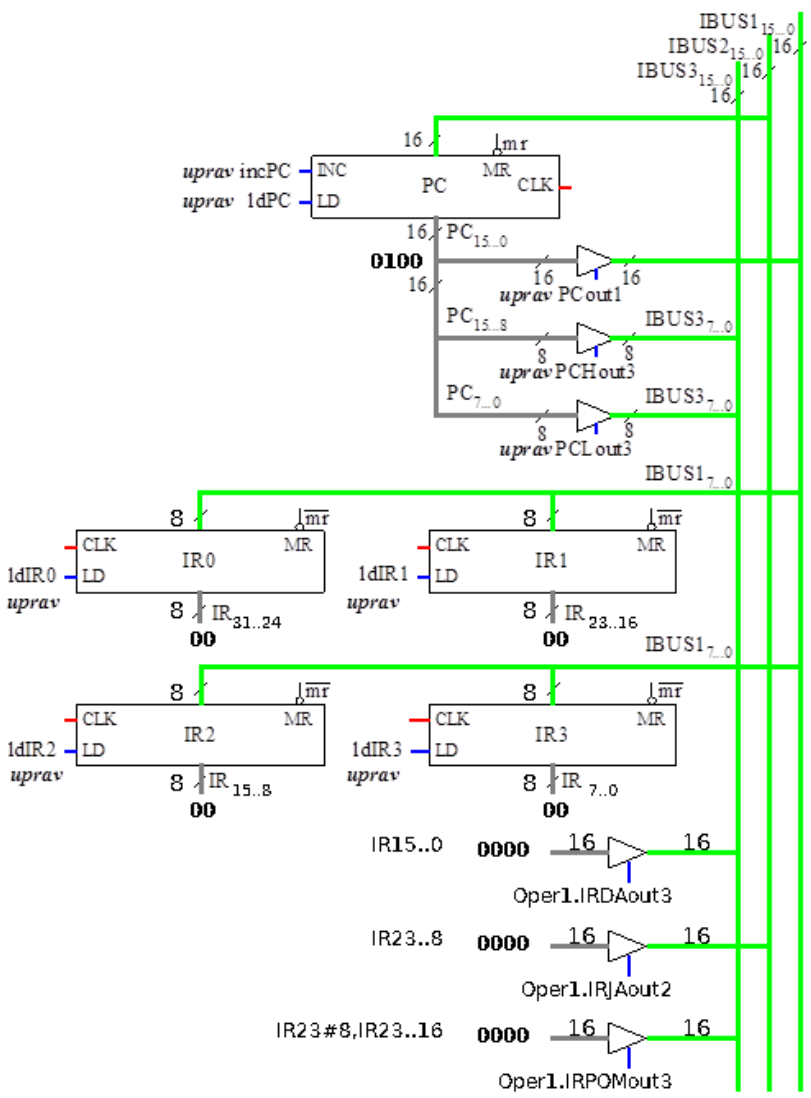
Matematička gimnazija
NEDELJA INFORMATIKE v5.0

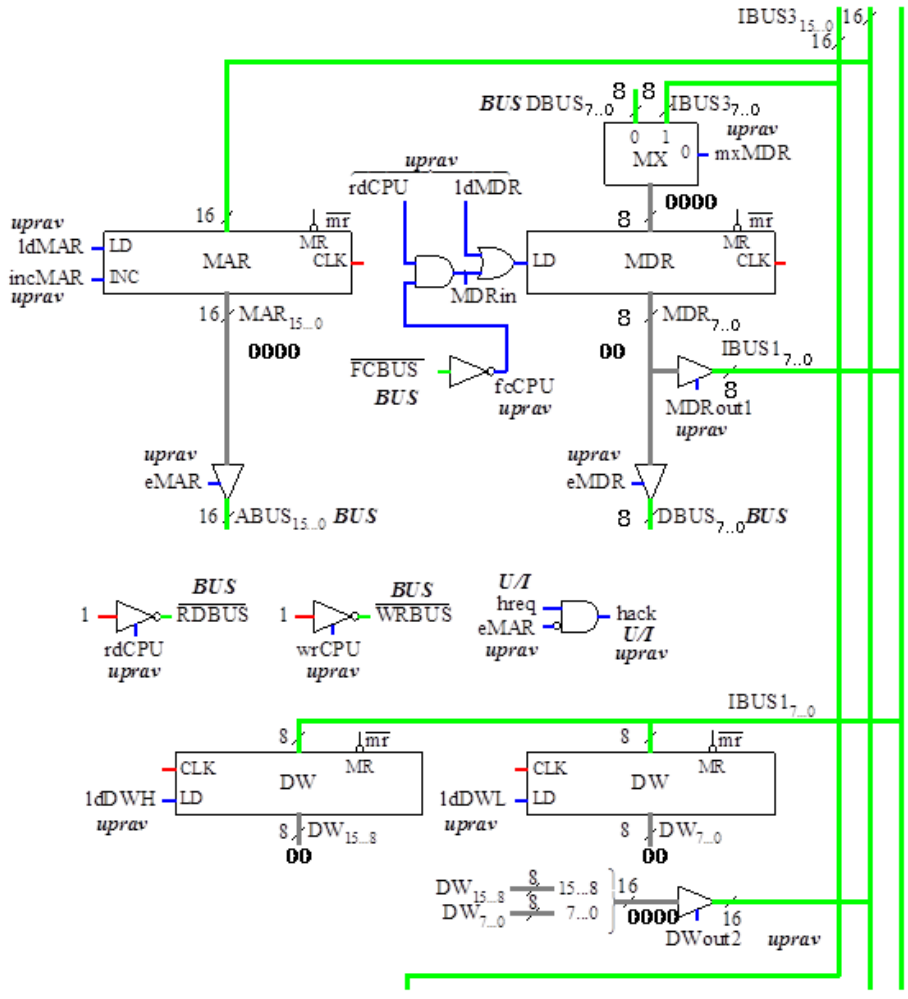
18. decembar 2018.

Rekapitulacija

- Nastavak prošlogodišnjeg predavanja (gledaj NI v4.0 materijale!)
- Bottom up pristup
- Cilj - minimalizovati prazninu između top-down i bottom-up


CPU
upravljačka jedinica
aritmetička jedinica
akumulator
registar R
ulazno/izlazna jedinica
memorija (za instrukcije i podatke)

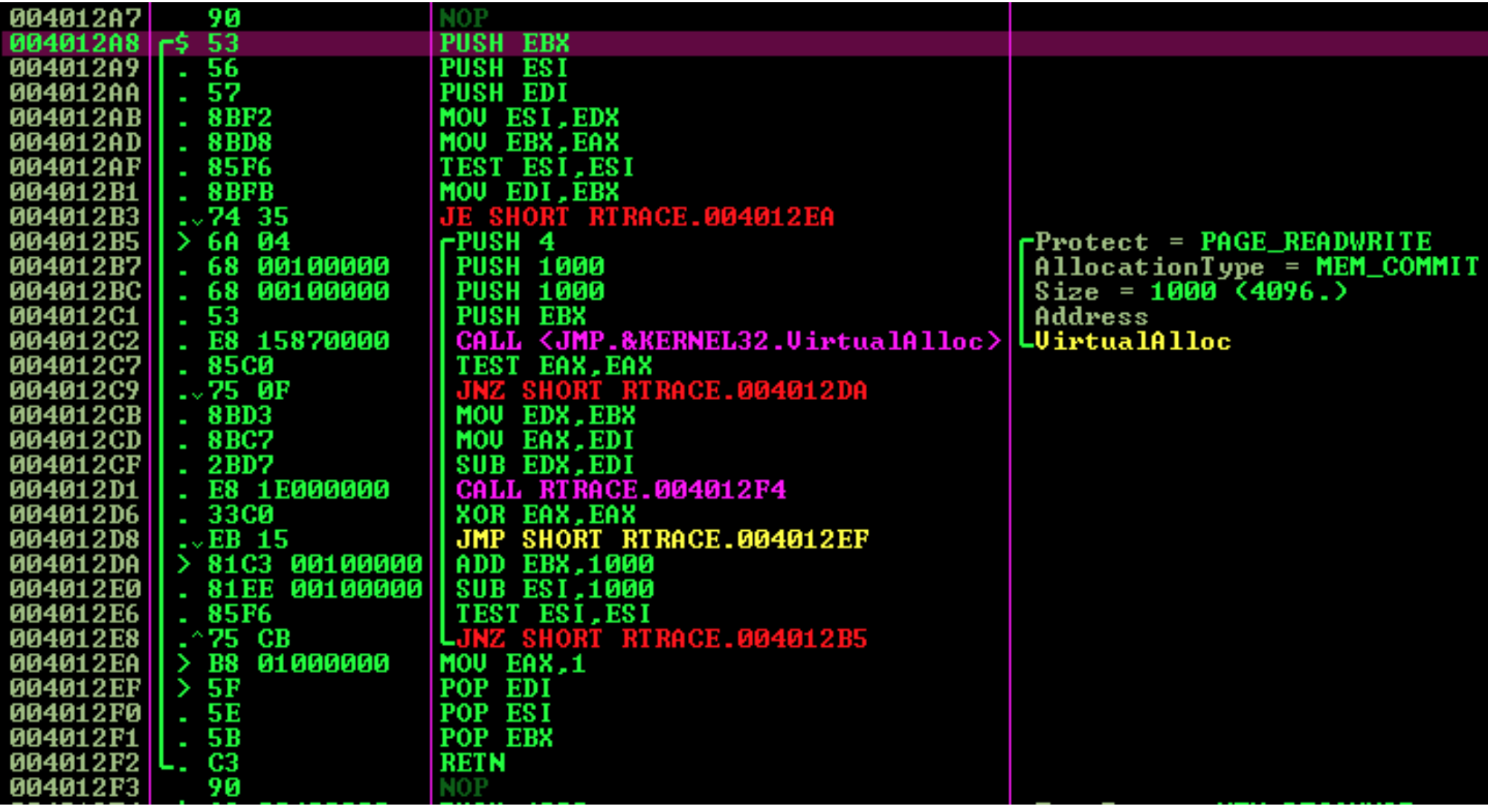

IBUS1 15..0
IBUS2 15..0
IBUS3 15..0
16
uprav incPC
uprav ldPC
INC
LD
PC
MR
CLK
mr
PC 15..0
0100
uprav PCout1
PC 15..8
IBUS3 7..0
uprav PCHout3
PC 7..0
uprav PCLout3
IBUS1 7..0
ldIR0
ldIR1
ldIR2
ldIR3
IR0
IR1
IR2
IR3
IR 31..24
IR 23..16
IR 15..8
IR 7..0
00
IR15..0
IR23..8
IR23#8,IR23..16
0000
Oper1.IRDAout3
Oper1.IRJAout2
Oper1.IRPOMout3

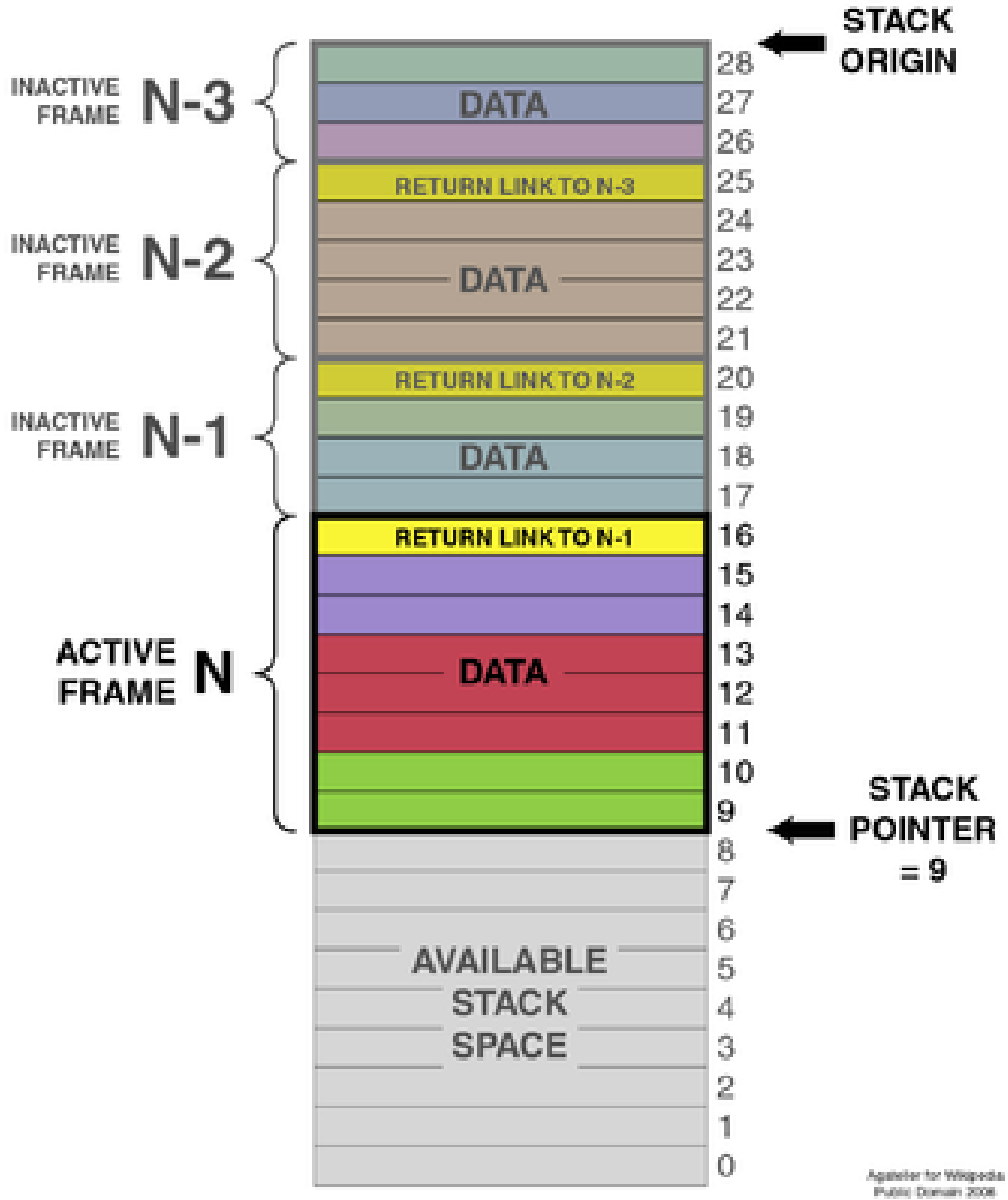

IBUS3 15..0
BUS DBUS 7..0
IBUS3 7..0
uprav
mxMDR
MX
0000
rdCPU
ldMDR
ldMAR
incMAR
MAR
MDR
MDRin
MAR 15..0
MDR 7..0
FCBUS
BUS
fcCPU
uprav
IBUS1 7..0
MDRout1
eMAR
eMDR
ABUS 15..0 BUS
DBUS 7..0 BUS
RDBUS
WRBUS
rdCPU
wrCPU
U/I
hreq
hack
ldDWH
ldDWL
DW
DW 15..8
DW 7..0
DWout2

$$\mathbf{n!} = \mathbf{n}\ (\mathbf{n}-1)\ (\mathbf{n}-2)\ \ldots\ 2\ 1 = \prod_{\mathbf{i}=1}^{\mathbf{n}} \mathbf{i}$$

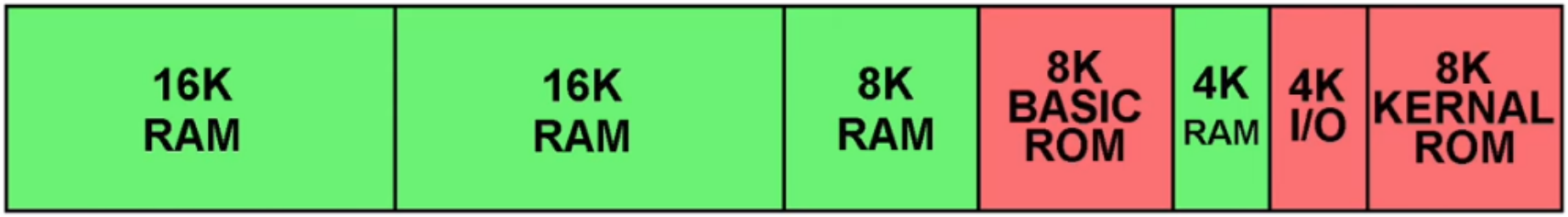
```
        N=1
        F=2
        ORG 8
DALJE:  IN N
        BGT 0, N, KRAJ
        JSR NF
        OUT F
        BEQ N, N, DALJE
KRAJ:   STOP
```

```
NF:     MOV F, 1
        BEQ N, 0, GOTOVO
PETLJA: MUL F, F, N
        SUB N, N, 1
        BGT N, 0, PETLJA
GOTOVO: RTS
```

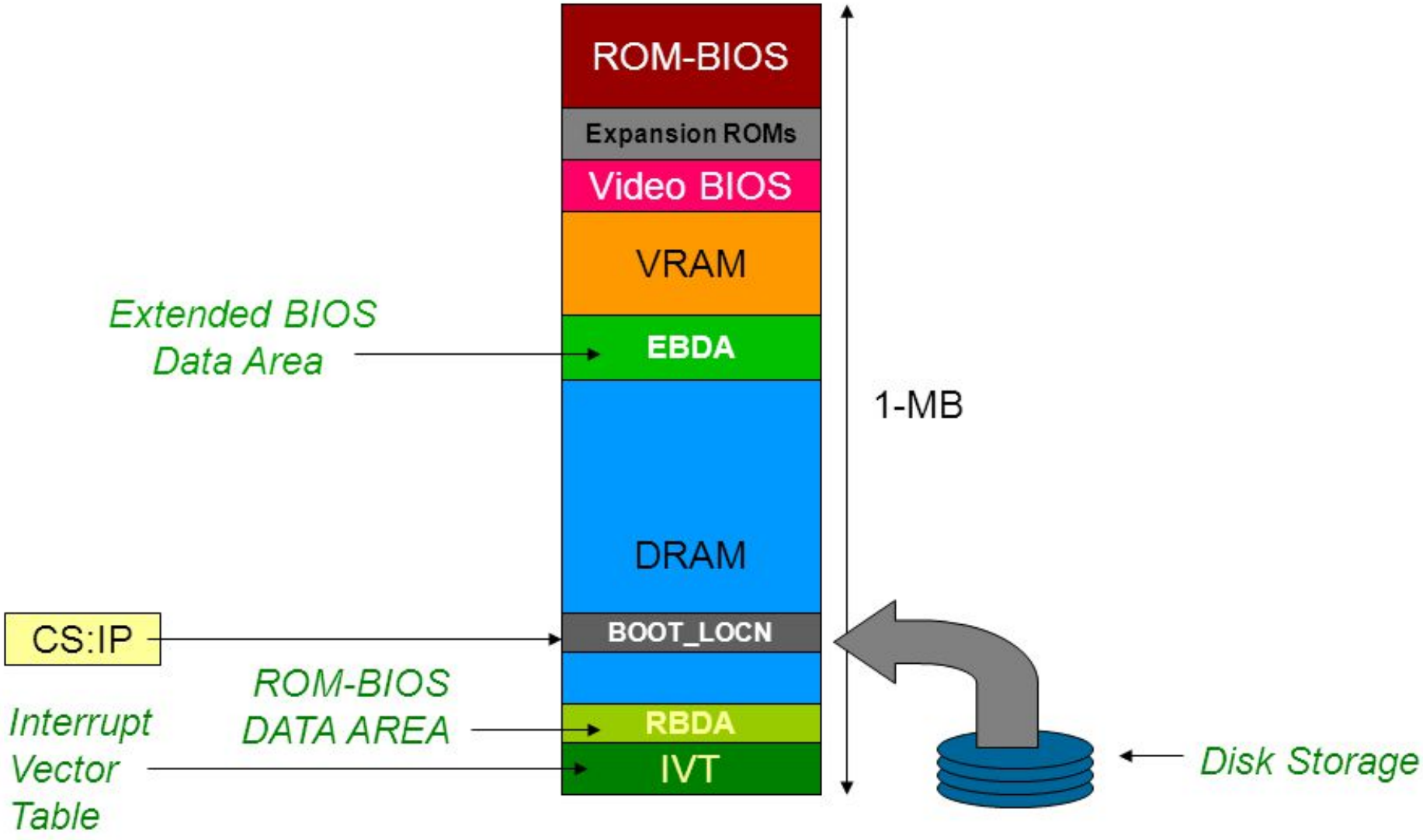

004012A7 90 NOP
004012A8 $ 53 PUSH EBX
004012A9 . 56 PUSH ESI
004012AA . 57 PUSH EDI
004012AB . 8BF2 MOV ESI,EDX
004012AD . 8BD8 MOV EBX,EAX
004012AF . 85F6 TEST ESI,ESI
004012B1 . 8BFB MOV EDI,EBX
004012B3 . 74 35 JE SHORT RTRACE.004012EA
004012B5 > 6A 04 PUSH 4
004012B7 . 68 00100000 PUSH 1000
004012BC . 68 00100000 PUSH 1000
004012C1 . 53 PUSH EBX
004012C2 . E8 15870000 CALL <JMP.&KERNEL32.VirtualAlloc>
004012C7 . 85C0 TEST EAX,EAX
004012C9 . 75 0F JNZ SHORT RTRACE.004012DA
004012CB . 8BD3 MOV EDX,EBX
004012CD . 8BC7 MOV EAX,EDI
004012CF . 2BD7 SUB EDX,EDI
004012D1 . E8 1E000000 CALL RTRACE.004012F4
004012D6 . 33C0 XOR EAX,EAX
004012D8 . EB 15 JMP SHORT RTRACE.004012EF
004012DA > 81C3 00100000 ADD EBX,1000
004012E0 . 81EE 00100000 SUB ESI,1000
004012E6 . 85F6 TEST ESI,ESI
004012E8 . 75 CB JNZ SHORT RTRACE.004012B5
004012EA > B8 01000000 MOV EAX,1
004012EF > 5F POP EDI
004012F0 . 5E POP ESI
004012F1 . 5B POP EBX
004012F2 . C3 RETN
004012F3 90 NOP
Protect = PAGE_READWRITE
AllocationType = MEM_COMMIT
Size = 1000 (4096.)
Address
VirtualAlloc



Commodore 64 Memory Range



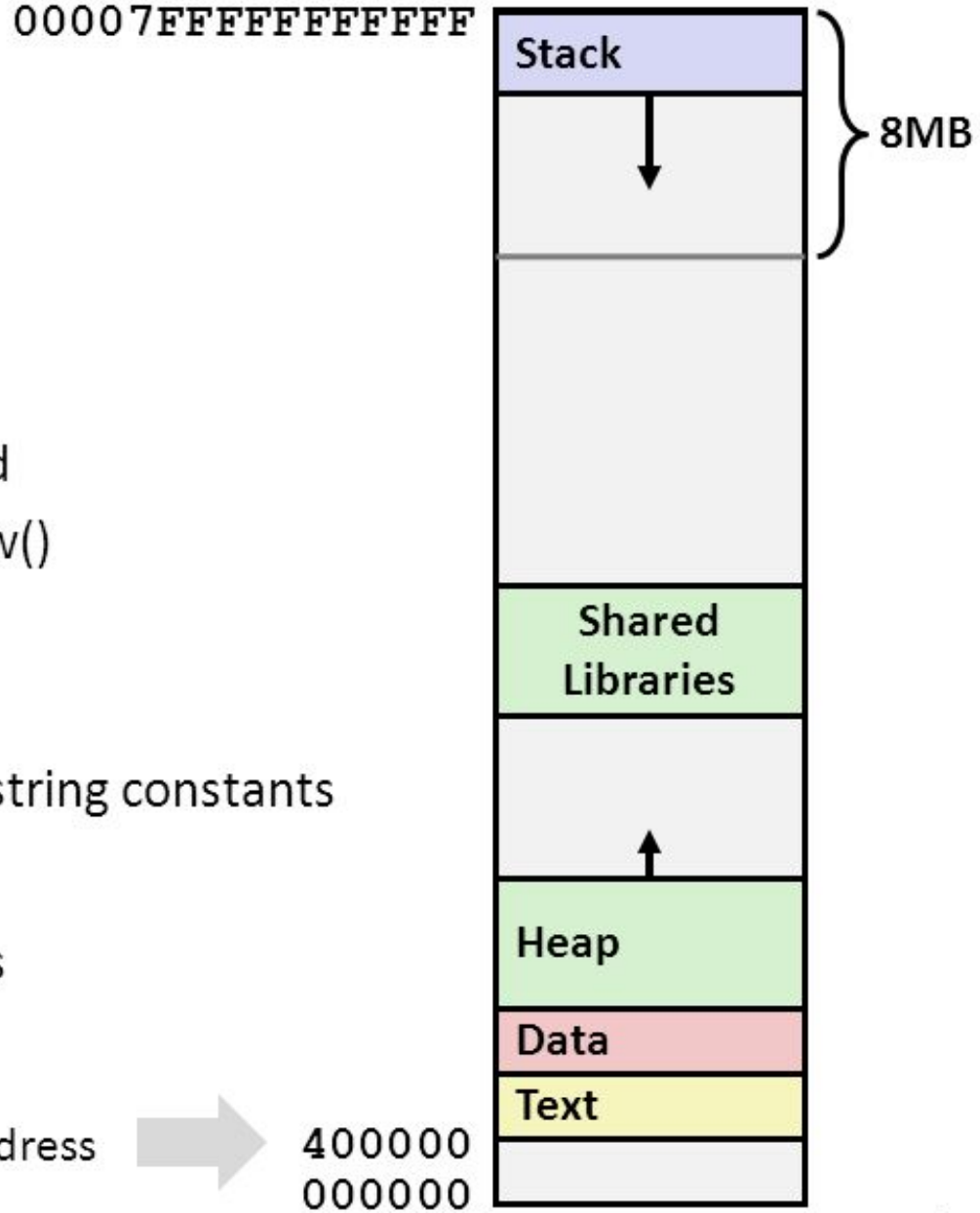
X86 ‘real-mode’ memory



x86-64 Linux Memory Layout

not drawn to scale

- **Stack**
  - Runtime stack (8MB limit)
  - E. g., local variables
- **Heap**
  - Dynamically allocated as needed
  - When call malloc(), calloc(), new()
- **Data**
  - Statically allocated data
  - E.g., global vars, `static` vars, string constants
- **Text / Shared Libraries**
  - Executable machine instructions
  - Read-only

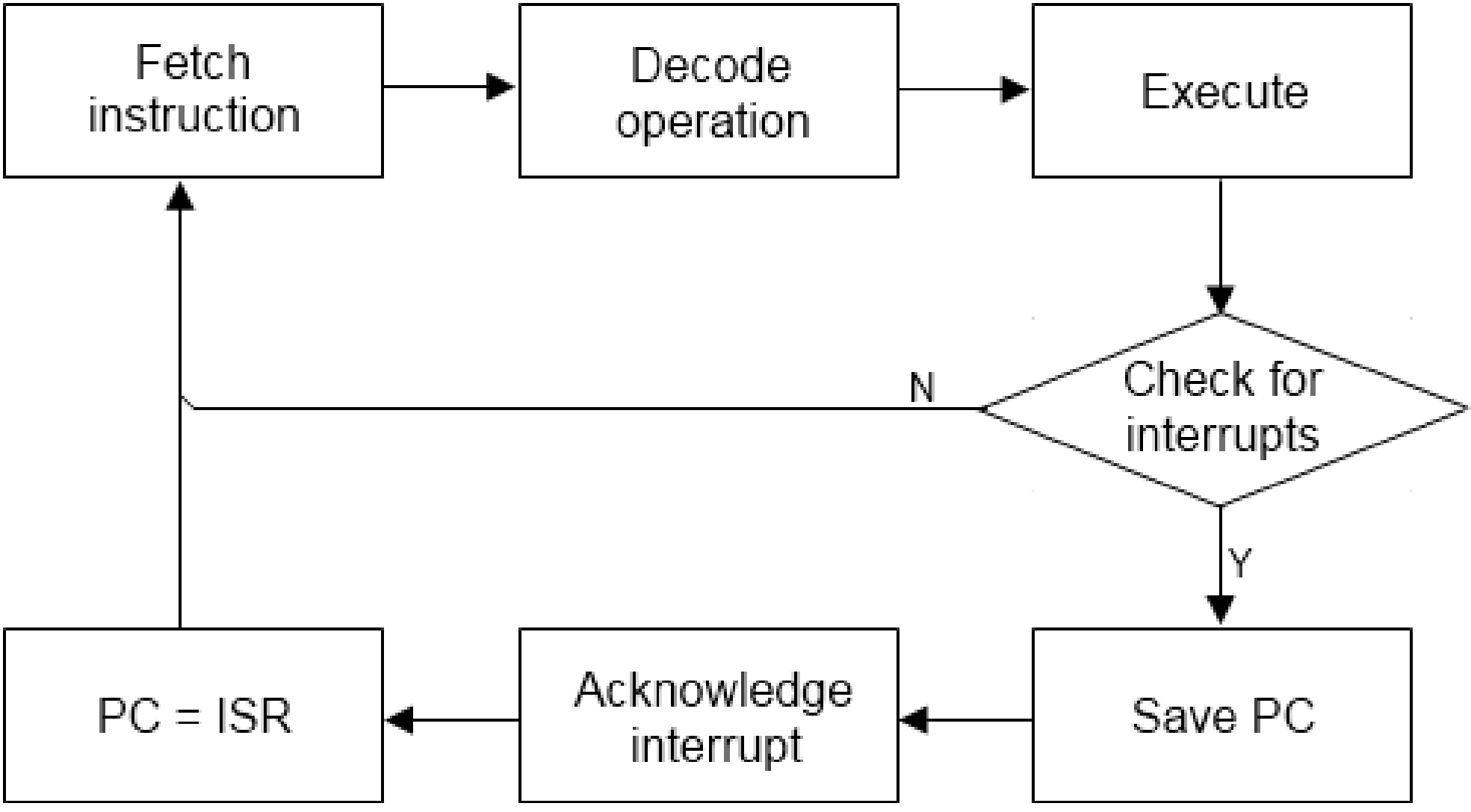


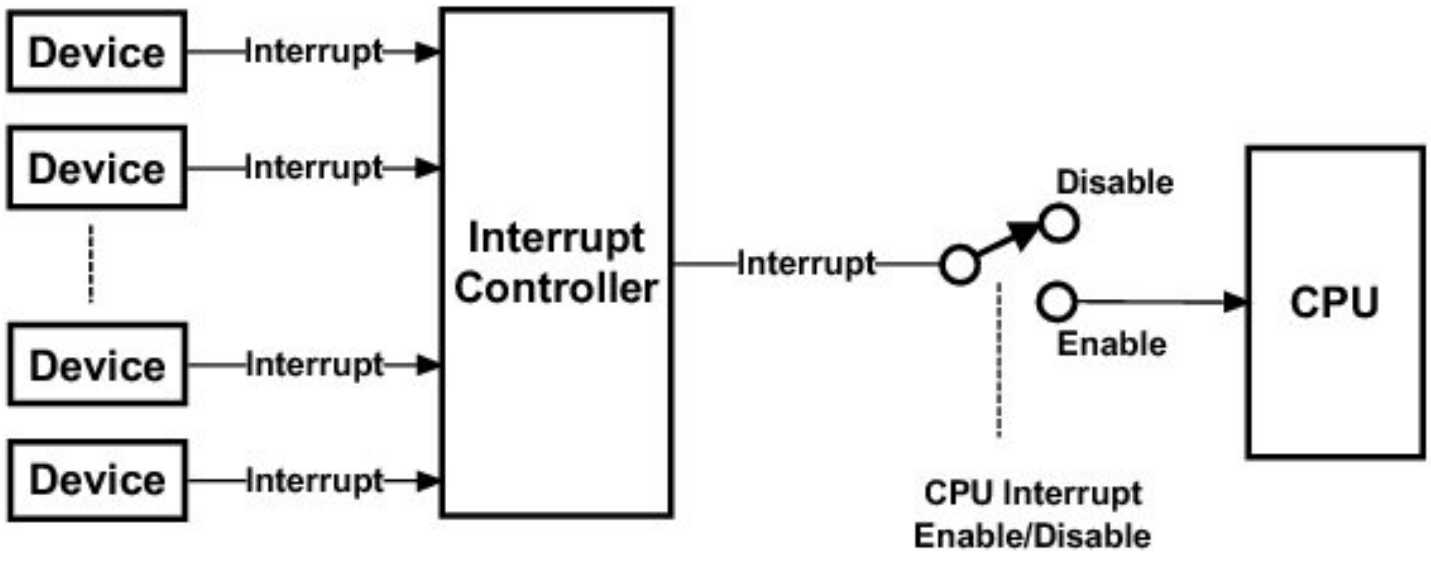
x86 Operation Modes

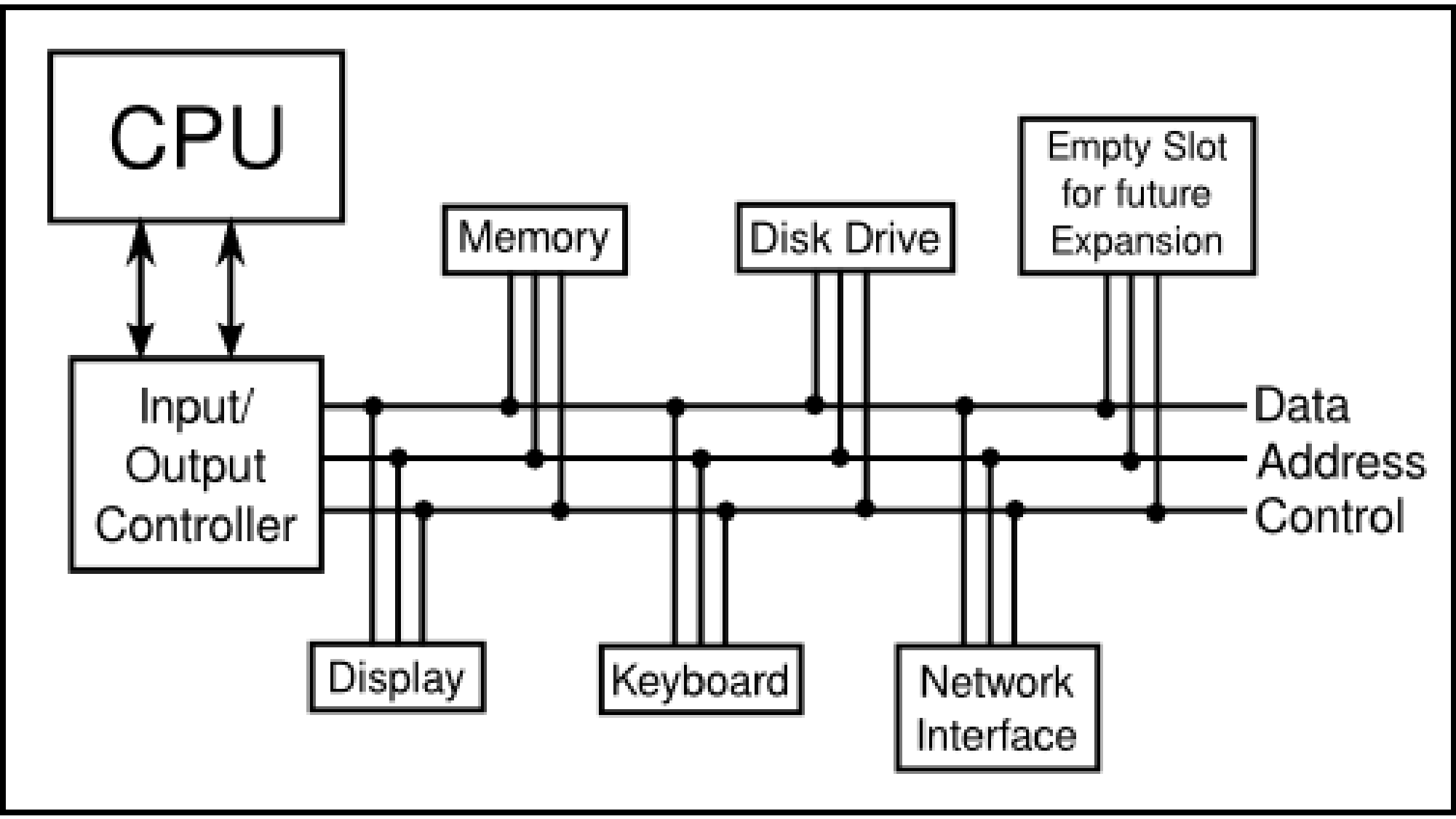
- **Real Mode** (= real address mode)
  - Programming environment of the 8086 processor
  - 8086 is a 16-bit processor from Intel

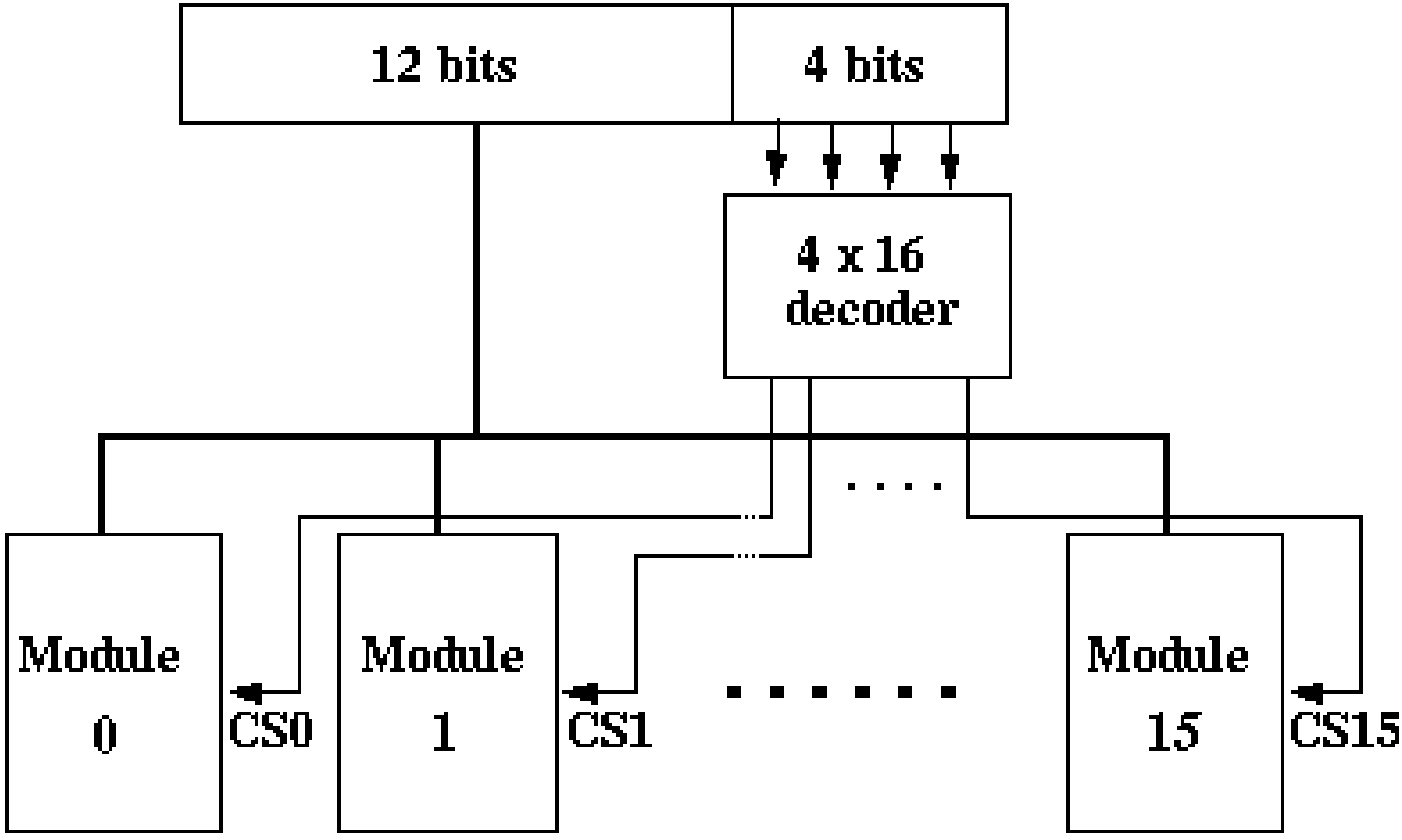
- **Protected Mode**
  - Native state of the 32-bit Intel processor
  - 32-bit mode

- **IA-32e mode (Intel) or Long mode (AMD)**
  - 2 sub modes: Compatibility mode and 64-bit mode
    - Compatibility mode is enabled by the operating system **on a code segment basis**. It means that a single 64-bit OS can support both 64-bit applications running in 64-bit mode and legacy 32-bit applications running in compatibility mode.


Fetch instruction
Decode operation
Execute
Check for interrupts
N
Y
Save PC
Acknowledge interrupt
PC = ISR

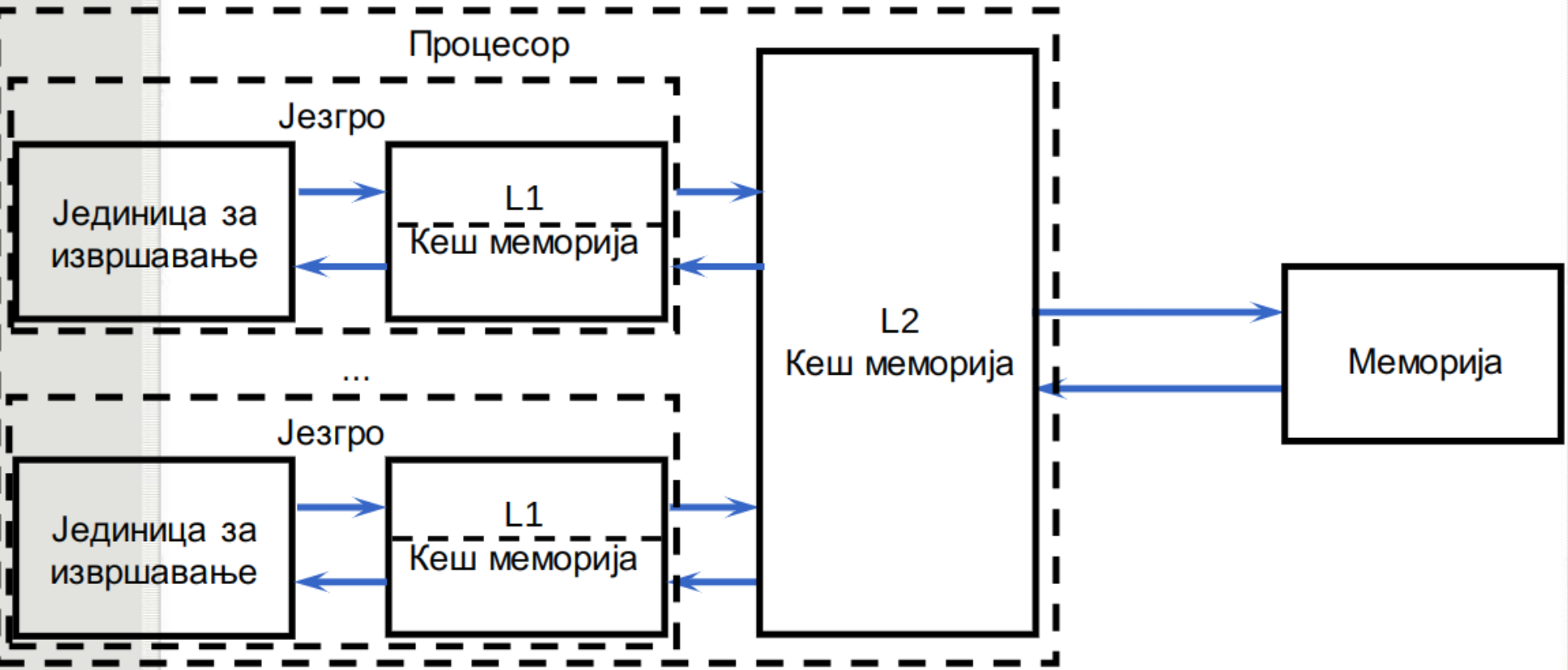

Device
Interrupt
Device
Interrupt
Device
Interrupt
Device
Interrupt
Interrupt Controller
Interrupt
Disable
Enable
CPU
CPU Interrupt Enable/Disable


CPU
Memory
Disk Drive
Empty Slot
for future
Expansion
Input/
Output
Controller
Data
Address
Control
Display
Keyboard
Network
Interface

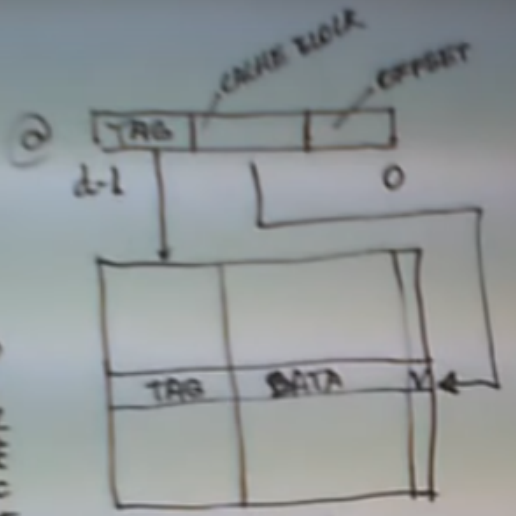


Memory Interleaving
(2^{16}=64 K words, 16 modules)

Једноставна(?) организација



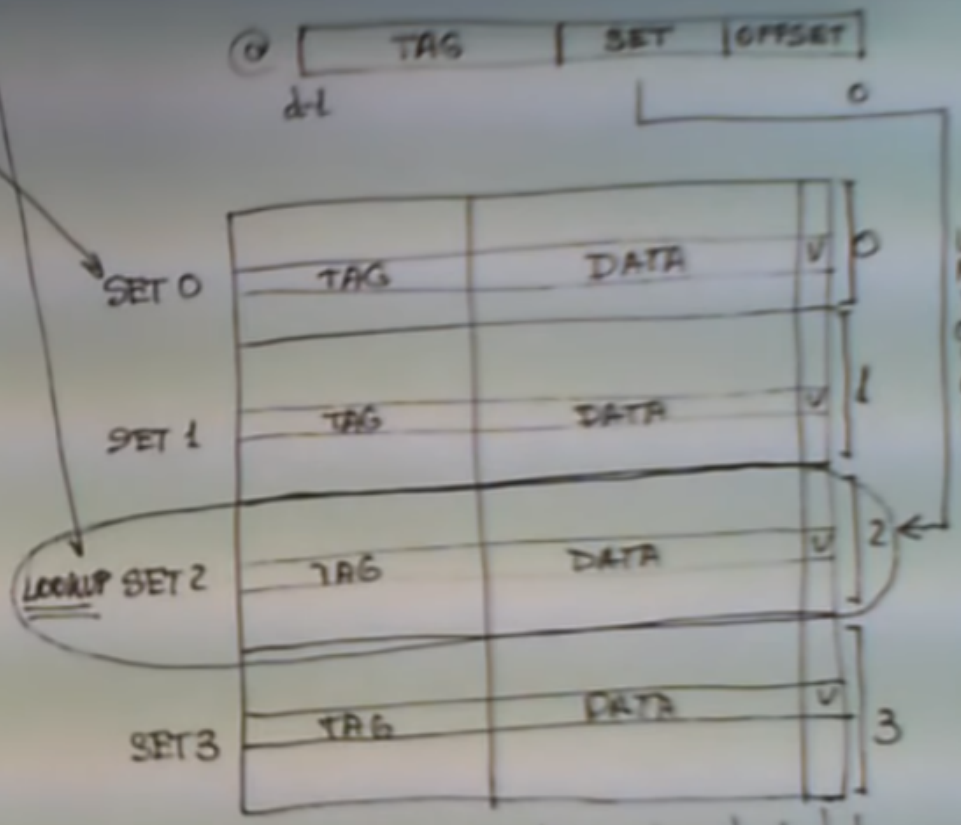
SET ASSOCIATIVE CACHE



DIRECT

1 memory block
⇓
1 single possible cache location

MUX

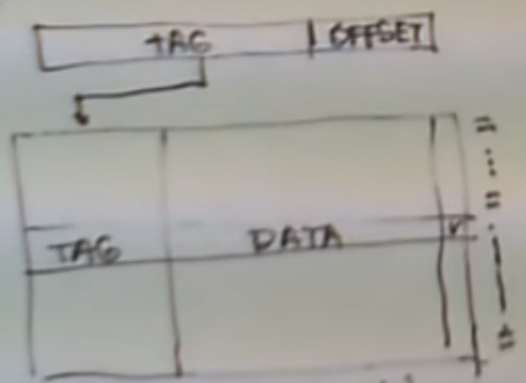


1 Memory block ⇒ 1 single set but to any row on that set

Total data = #Sets × #Blocks/set × Size of Block

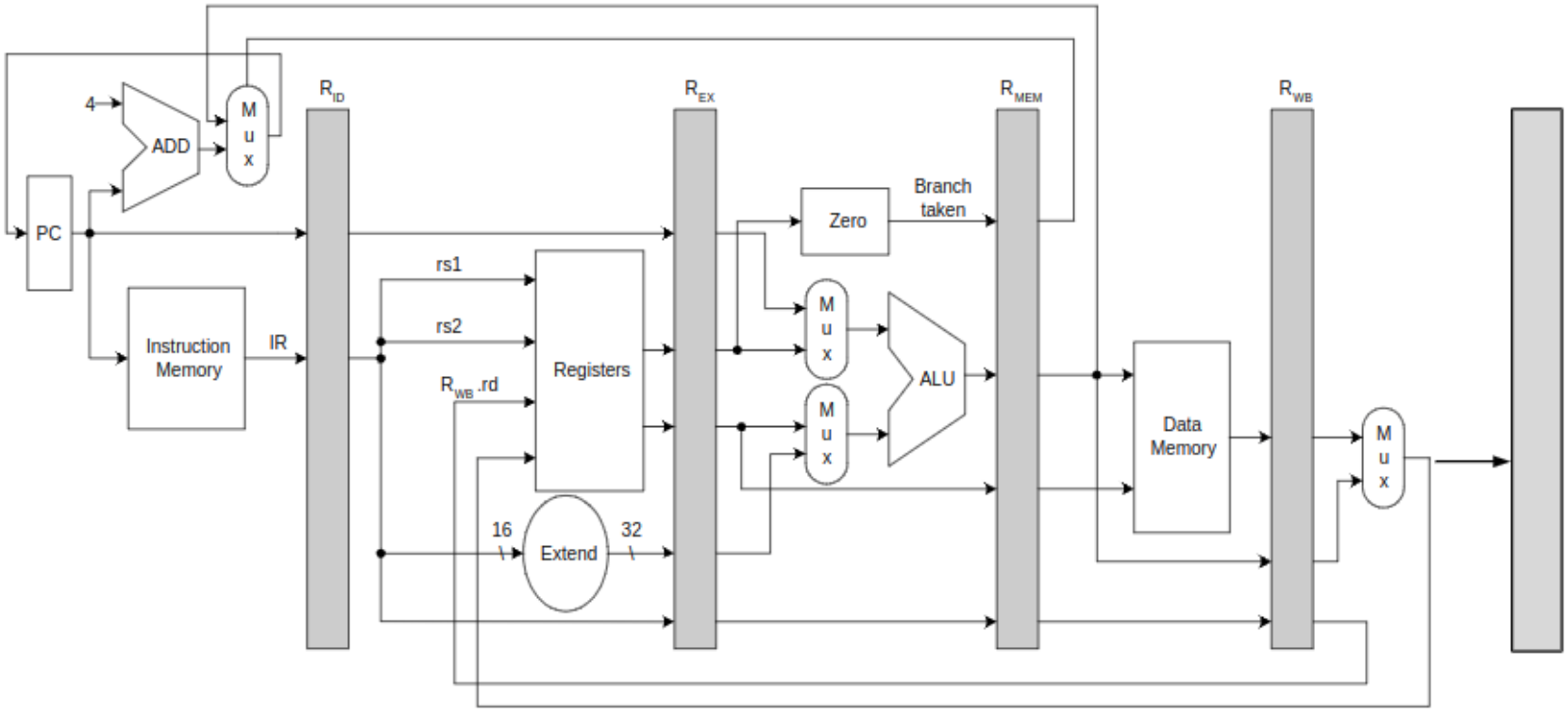
SIMPLER LOOKUP & STILL FLEXIBLE PLACEMENT

ASSOCIATIVE



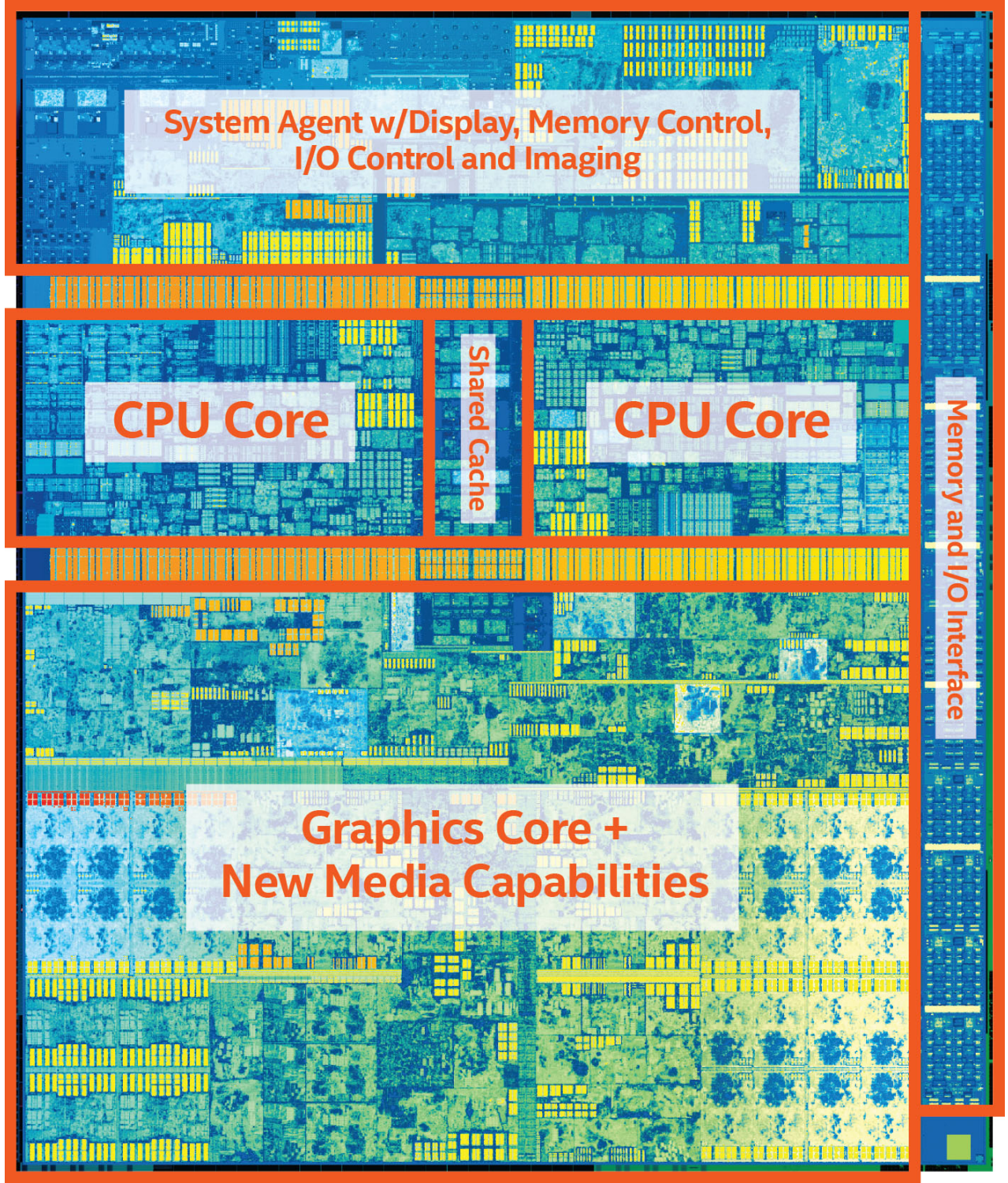
2/W ⇒ LOOKUP TAG IN ALL BLOCKS!

??? **???** **???** **???** **and** R6, *R1*, *R7*



sub *R5*, *R1*, *R8*

	1	2	3	4	5	6	7	8	9	10
lw *R1*, 32(*R6*)	IF	ID	EX	MEM	WB					
add *R4*, *R1*, *R7*		IF	ID	stall	EX	MEM	WB			
sub *R5*, *R1*, *R8*			IF	stall	ID	EX	MEM	WB		
and R6, *R1*, *R7*				stall	IF	ID	EX	MEM		

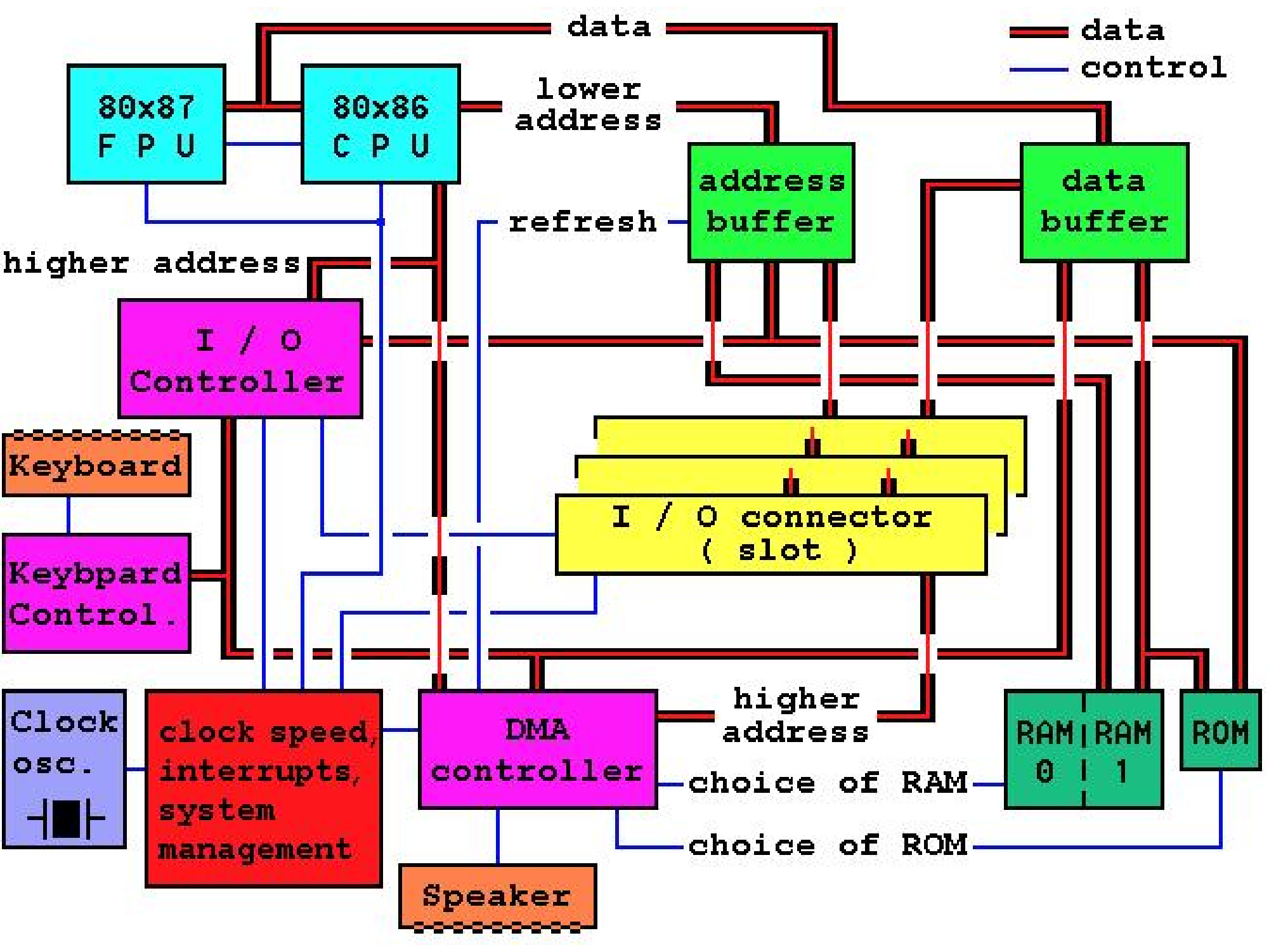

System Agent w/Display, Memory Control, I/O Control and Imaging
CPU Core
Shared Cache
CPU Core
Memory and I/O Interface
Graphics Core + New Media Capabilities

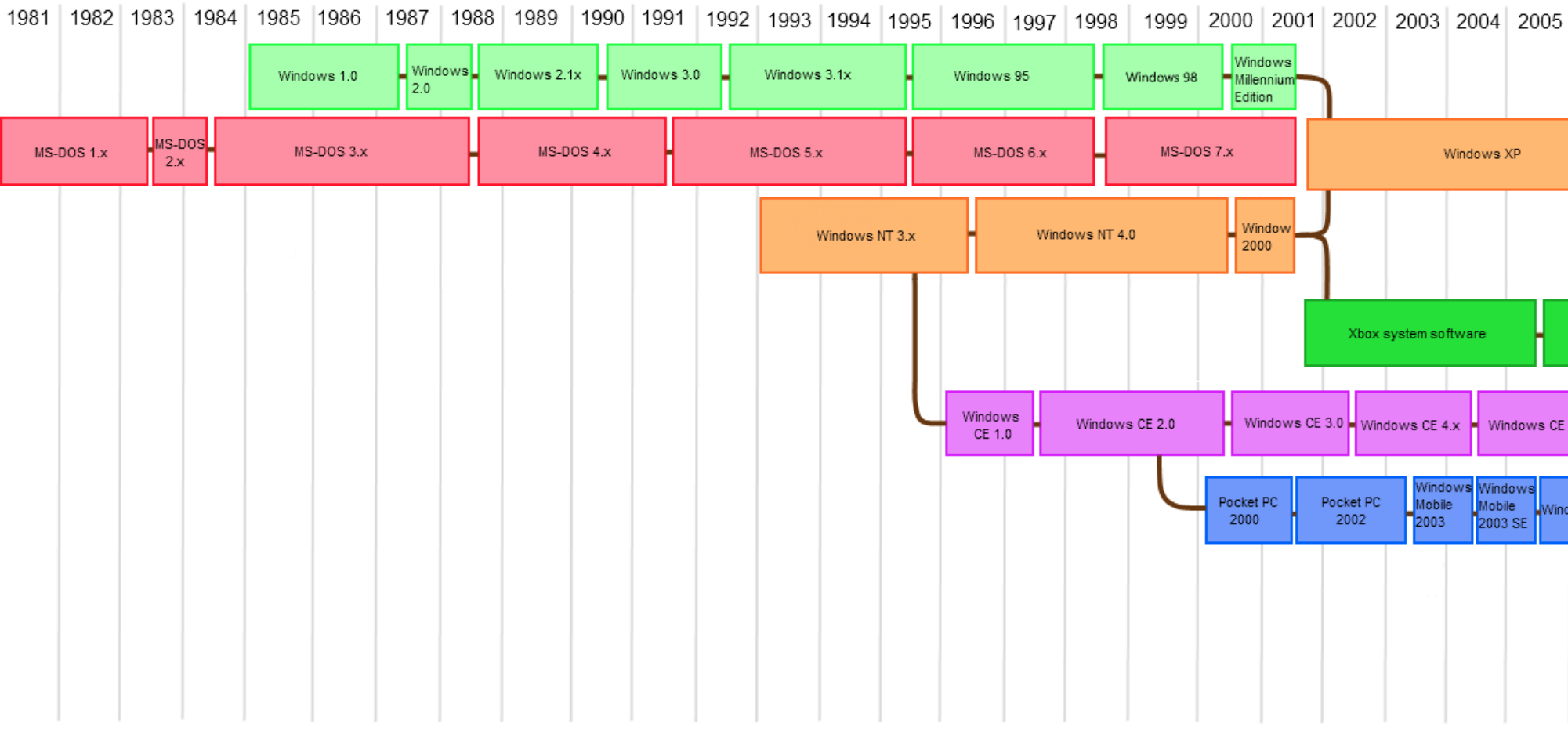
x86

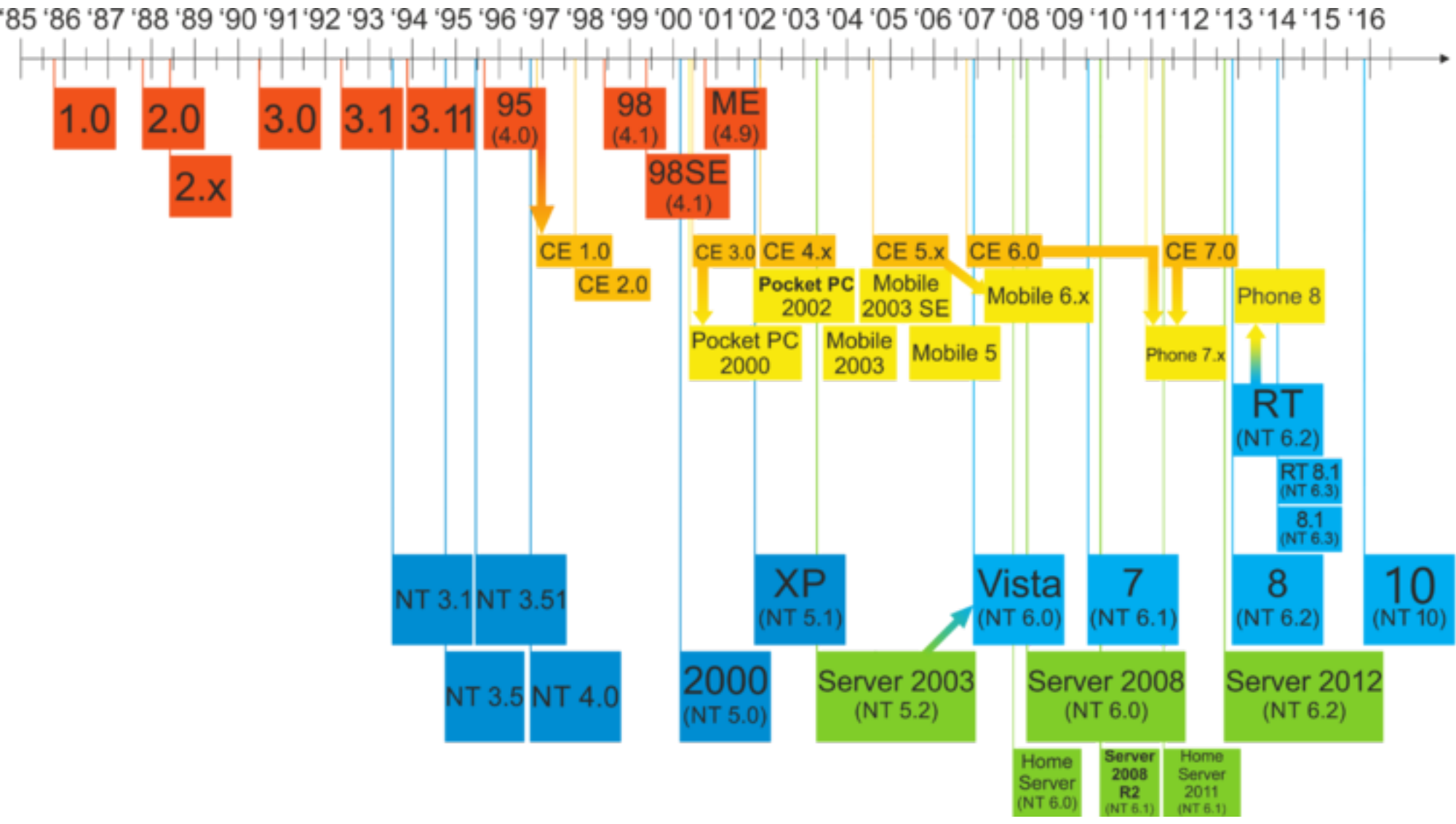
- 8086 - začetnik - 1979.
- 8088 -u IBM PC, 8 bitna magistrala
- 80286 (286) - 24 bitne adrese + pipeline + protected mode - 1982.
- 80386 (386) - 32 bita (4GB) - 1985
- 80486 (486) - integrisan FPU, L1 cache - 1989.
- Pentium - superskalar, 64bitna magistrala, Branch prediction, FDIV bug - 1993.
- Pentium 2,3,4
- AMD64 (i pokušaj IA64) - 2003.

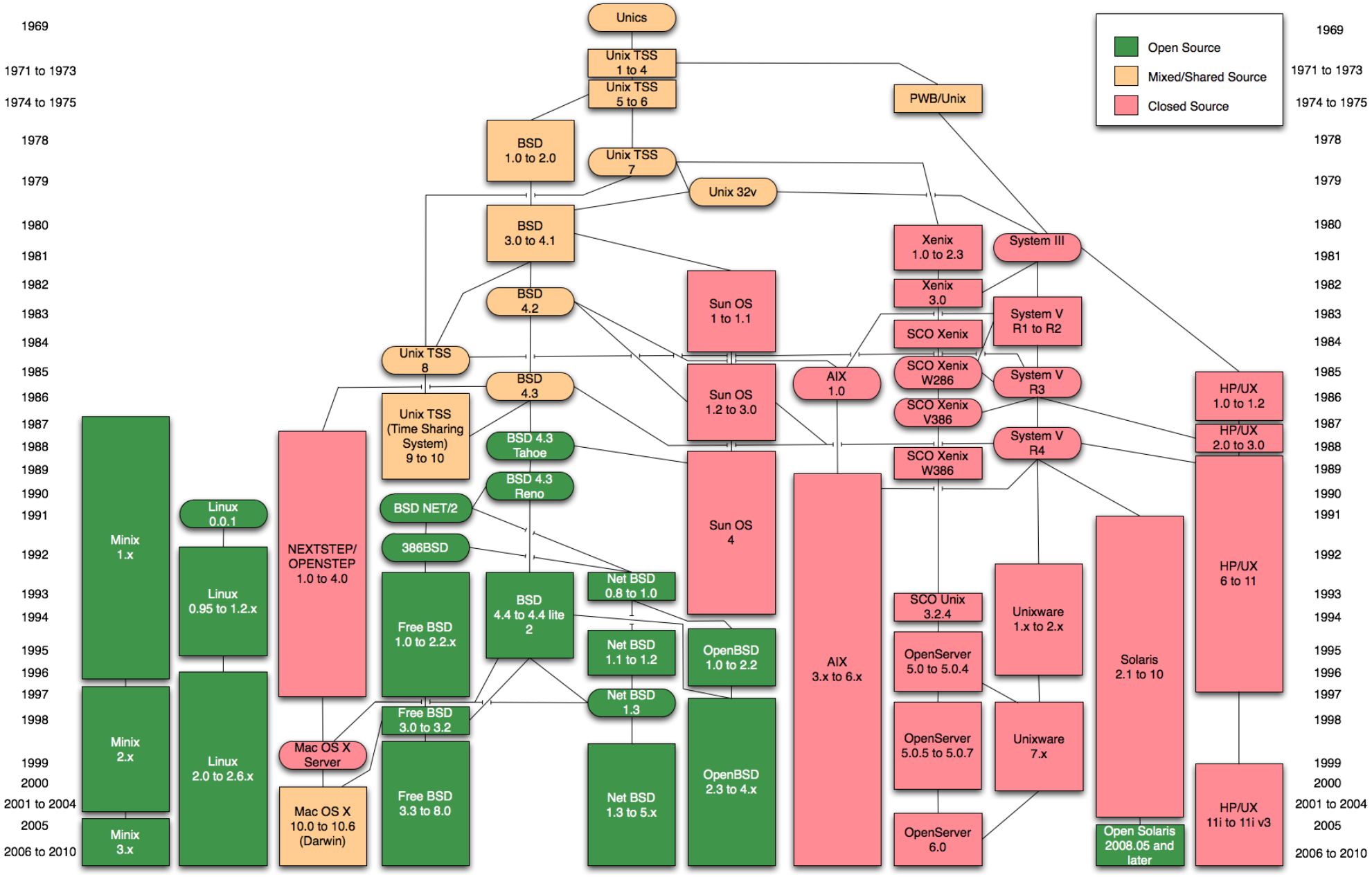
x86

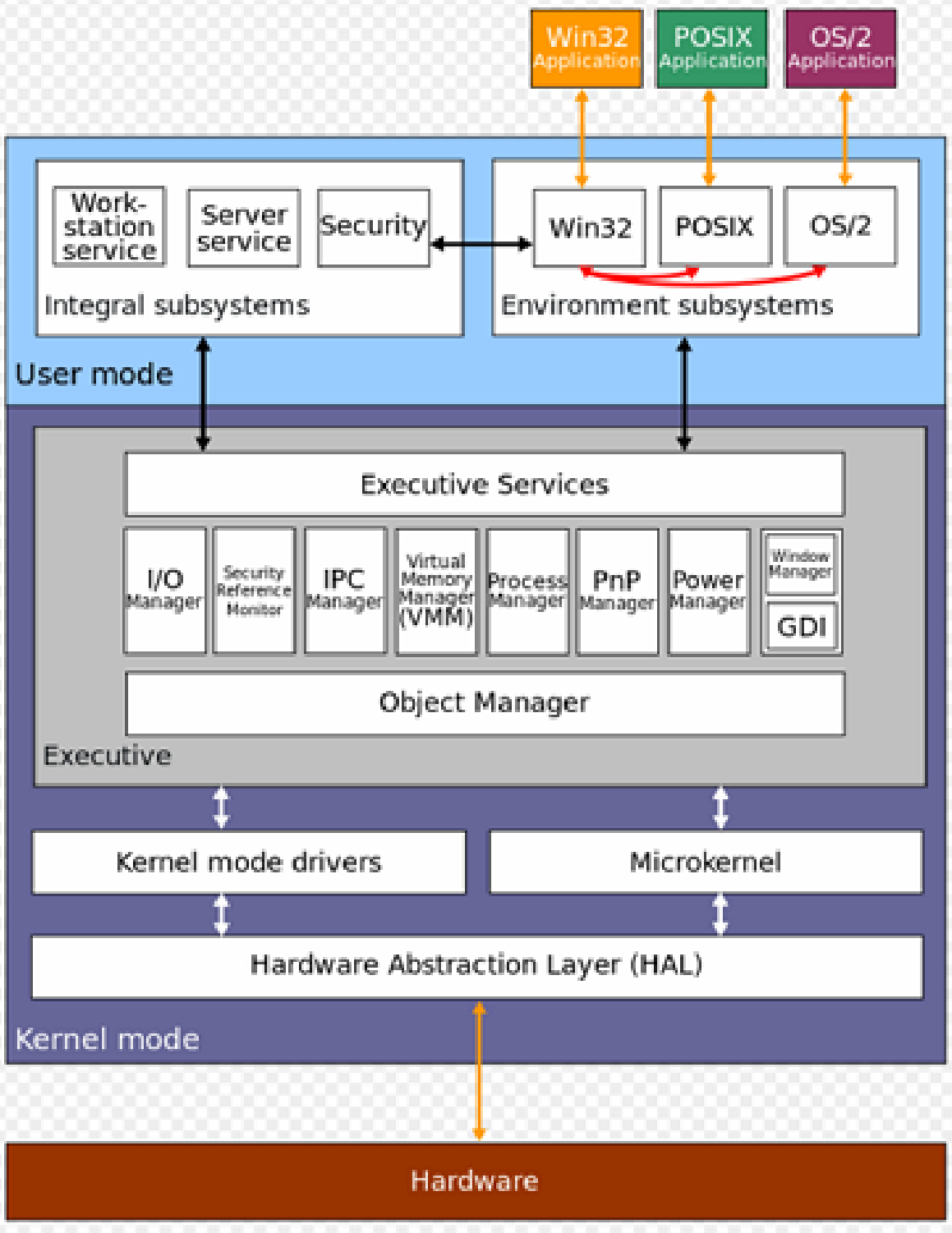
- AMD Athlon 64 X2 - prvi dual core - 2005.
- Intel Core 2 Duo - 2006.

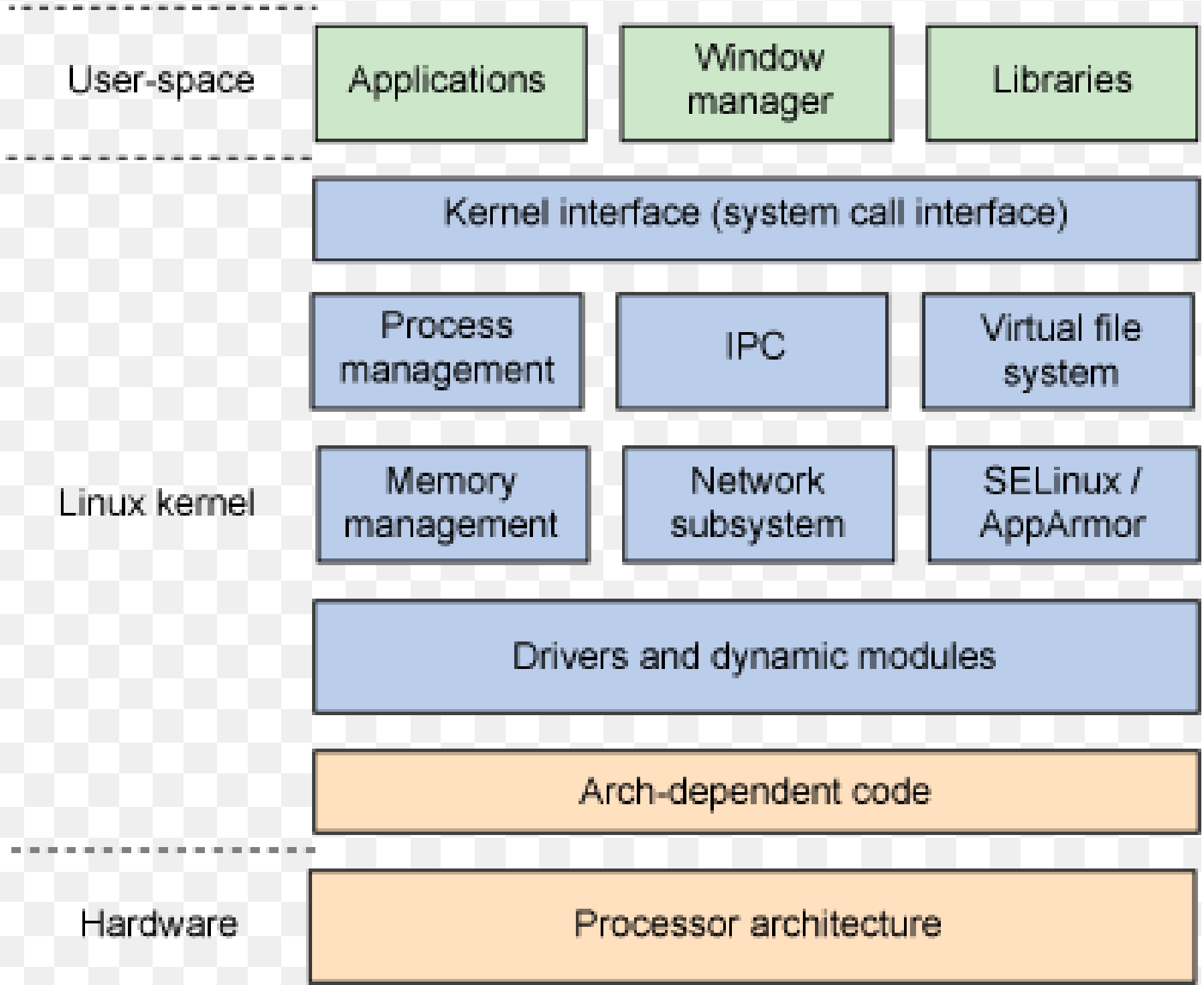

data
data
control
80x87
F P U
80x86
C P U
lower
address
address
buffer
data
buffer
refresh
higher address
I / O
Controller
Keyboard
Keybpard
Control.
I / O connector
(slot)
Clock
osc.
clock speed,
interrupts,
system
management
DMA
controller
higher
address
choice of RAM
choice of ROM
RAM
0
RAM
1
ROM
Speaker


1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
Windows 1.0
Windows 2.0
Windows 2.1x
Windows 3.0
Windows 3.1x
Windows 95
Windows 98
Windows Millennium Edition
MS-DOS 1.x
MS-DOS 2.x
MS-DOS 3.x
MS-DOS 4.x
MS-DOS 5.x
MS-DOS 6.x
MS-DOS 7.x
Windows XP
Windows NT 3.x
Windows NT 4.0
Window 2000
Xbox system software
Windows CE 1.0
Windows CE 2.0
Windows CE 3.0
Windows CE 4.x
Windows CE
Pocket PC 2000
Pocket PC 2002
Windows Mobile 2003
Windows Mobile 2003 SE

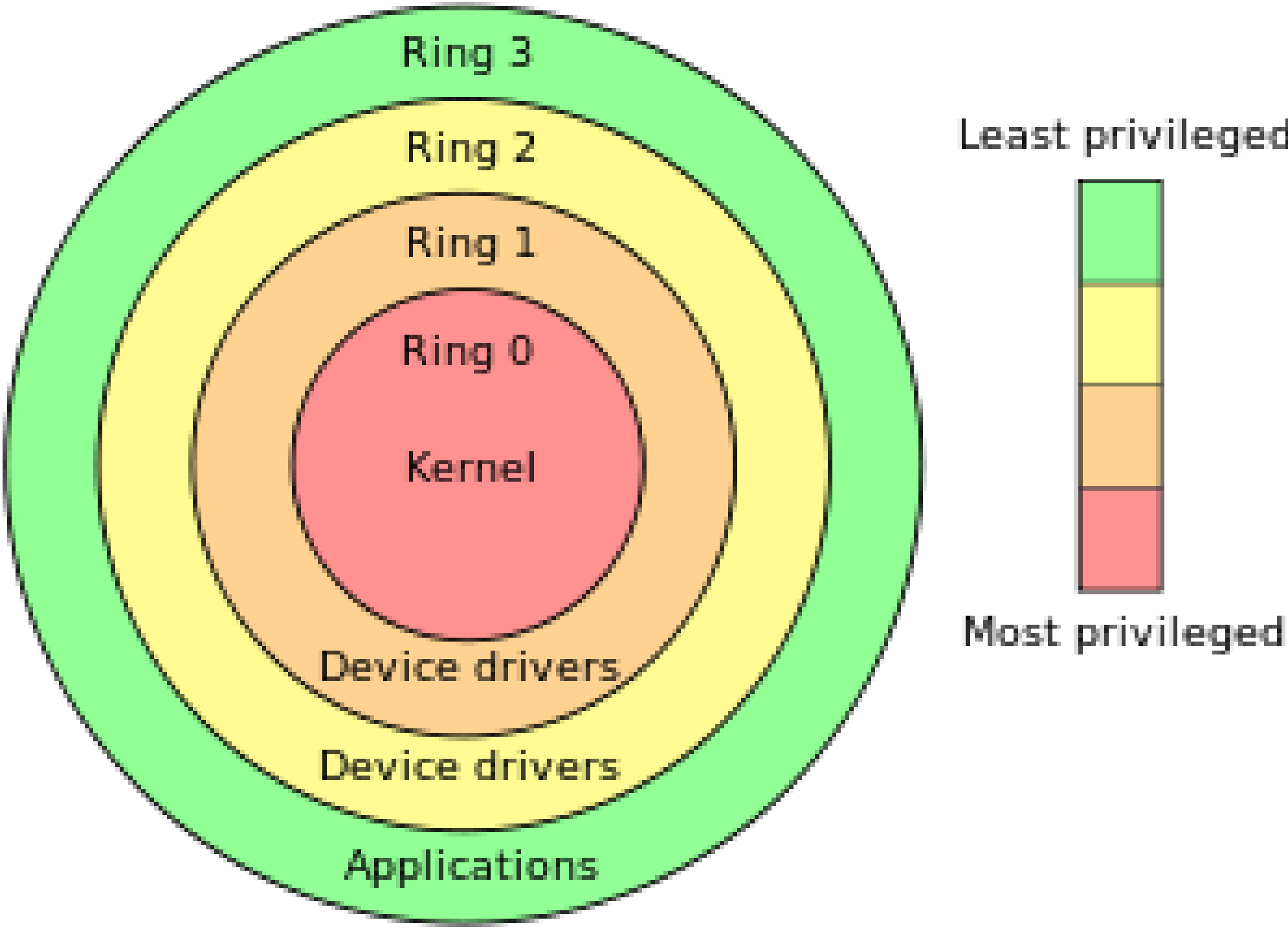

'85 '86 '87 '88 '89 '90 '91 '92 '93 '94 '95 '96 '97 '98 '99 '00 '01 '02 '03 '04 '05 '06 '07 '08 '09 '10 '11 '12 '13 '14 '15 '16
1.0
2.0
2.x
3.0
3.1
3.11
95 (4.0)
98 (4.1)
98SE (4.1)
ME (4.9)
CE 1.0
CE 2.0
CE 3.0
CE 4.x
CE 5.x
CE 6.0
CE 7.0
Pocket PC 2002
Mobile 2003 SE
Mobile 6.x
Phone 8
Pocket PC 2000
Mobile 2003
Mobile 5
Phone 7.x
RT (NT 6.2)
RT 8.1 (NT 6.3)
8.1 (NT 6.3)
NT 3.1
NT 3.51
XP (NT 5.1)
Vista (NT 6.0)
7 (NT 6.1)
8 (NT 6.2)
10 (NT 10)
NT 3.5
NT 4.0
2000 (NT 5.0)
Server 2003 (NT 5.2)
Server 2008 (NT 6.0)
Server 2012 (NT 6.2)
Home Server (NT 6.0)
Server 2008 R2 (NT 6.1)
Home Server 2011 (NT 6.1)


1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2010
Open Source
Mixed/Shared Source
Closed Source
Unics
Unix TSS 1 to 4
Unix TSS 5 to 6
PWB/Unix
BSD 1.0 to 2.0
Unix TSS 7
Unix 32v
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
BSD 4.2
Xenix 3.0
Sun OS 1 to 1.1
System V R1 to R2
SCO Xenix
Unix TSS 8
BSD 4.3
SCO Xenix W286
AIX 1.0
System V R3
HP/UX 1.0 to 1.2
Unix TSS (Time Sharing System) 9 to 10
Sun OS 1.2 to 3.0
SCO Xenix V386
HP/UX 2.0 to 3.0
BSD 4.3 Tahoe
System V R4
BSD 4.3 Reno
SCO Xenix W386
Minix 1.x
Linux 0.0.1
BSD NET/2
Sun OS 4
NEXTSTEP/ OPENSTEP 1.0 to 4.0
386BSD
AIX 3.x to 6.x
HP/UX 6 to 11
Net BSD 0.8 to 1.0
SCO Unix 3.2.4
Unixware 1.x to 2.x
Linux 0.95 to 1.2.x
Free BSD 1.0 to 2.2.x
BSD 4.4 to 4.4 lite 2
Net BSD 1.1 to 1.2
OpenBSD 1.0 to 2.2
OpenServer 5.0 to 5.0.4
Solaris 2.1 to 10
Net BSD 1.3
Free BSD 3.0 to 3.2
OpenServer 5.0.5 to 5.0.7
Unixware 7.x
Minix 2.x
Linux 2.0 to 2.6.x
Mac OS X Server
OpenBSD 2.3 to 4.x
Free BSD 3.3 to 8.0
Net BSD 1.3 to 5.x
OpenServer 6.0
HP/UX 11i to 11i v3
Minix 3.x
Mac OS X 10.0 to 10.6 (Darwin)
Open Solaris 2008.05 and later


Win32 Application
POSIX Application
OS/2 Application
Work-station service
Server service
Security
Integral subsystems
Win32
POSIX
OS/2
Environment subsystems
User mode
Executive Services
I/O Manager
Security Reference Monitor
IPC Manager
Virtual Memory Manager (VMM)
Process Manager
PnP Manager
Power Manager
Window Manager
GDI
Object Manager
Executive
Kernel mode drivers
Microkernel
Hardware Abstraction Layer (HAL)
Kernel mode
Hardware


User-space
Applications
Window manager
Libraries
Kernel interface (system call interface)
Process management
IPC
Virtual file system
Linux kernel
Memory management
Network subsystem
SELinux / AppArmor
Drivers and dynamic modules
Arch-dependent code
Hardware
Processor architecture

```
void main(void)          /* This really IS void, no error here. */
{                        /* The startup routine assumes (well, ...) this */
/*
 * Interrupts are still disabled. Do necessary setups, then
 * enable them
 */
        time_init();
        tty_init();
        trap_init();
        sched_init();
        buffer_init();
        hd_init();
        sti();
        move_to_user_mode();
        if (!fork()) {          /* we count on this going ok */
                init();
        }
```


Ring 3
Ring 2
Ring 1
Ring 0
Kernel
Device drivers
Device drivers
Applications
Least privileged
Most privileged

```
(gdb) list
1
2
3 int main()
4 {
5    int i;
6    for(i=0; i < 10; i++) // Loop 10 times.
7    {
8       puts("Hello, world!\n"); // put "Hello World" to the output.
9     }
10    return 0; // Tell OS the program exited without errors.
(gdb) disassemble main
Dump of assembler code for function main:
0x00000000004004f4 : push rbp
0x00000000004004f5 : mov rbp,rsp
0x00000000004004f8 : sub rsp,0x10
0x00000000004004fc : mov DWORD PTR [rbp-0x4],0x0
0x0000000000400503 : jmp 0x400513
0x0000000000400505 : mov edi,0x40060c
0x000000000040050a : call 0x4003f0
0x000000000040050f : add DWORD PTR [rbp-0x4],0x1
0x0000000000400513 : cmp DWORD PTR [rbp-0x4],0x9
0x0000000000400517 : jle 0x400505
0x0000000000400519 : mov eax,0x0
0x000000000040051e : leave
0x000000000040051f : ret
End of assembler dump.
```

Kolko ostane vremena

- Video akceleratori (3dfx VooDoo, PowerVR) + OpenGL/DirectX/Vulkan - vidi NI v4.0
- Cytrix vs Quake
- RISC vs CISC (i x86 alternative)
- Kompajleri, lekseri, linkeri (vidi NI v2.0)
- Interpreterski i kompajlerski programski jezici
- Soketi, mrežni stek, drajveri
- Kodeci i video kompresija (HW vs SW)