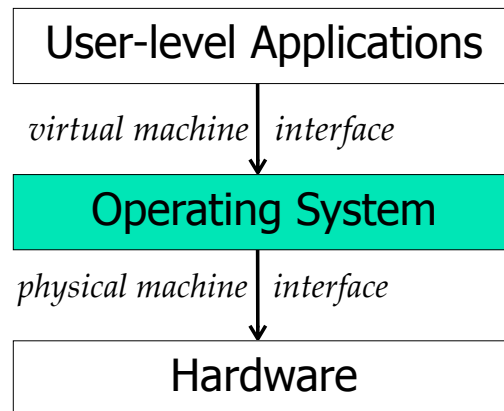
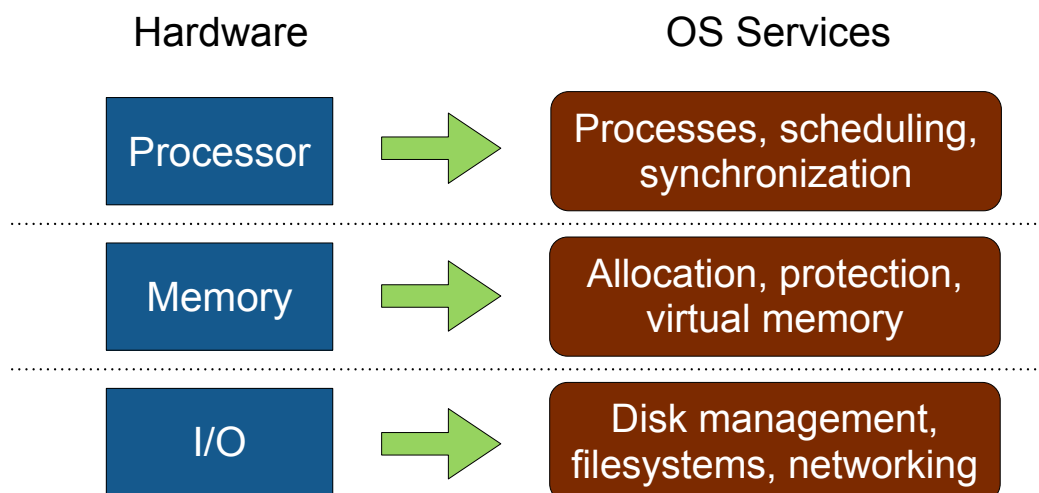


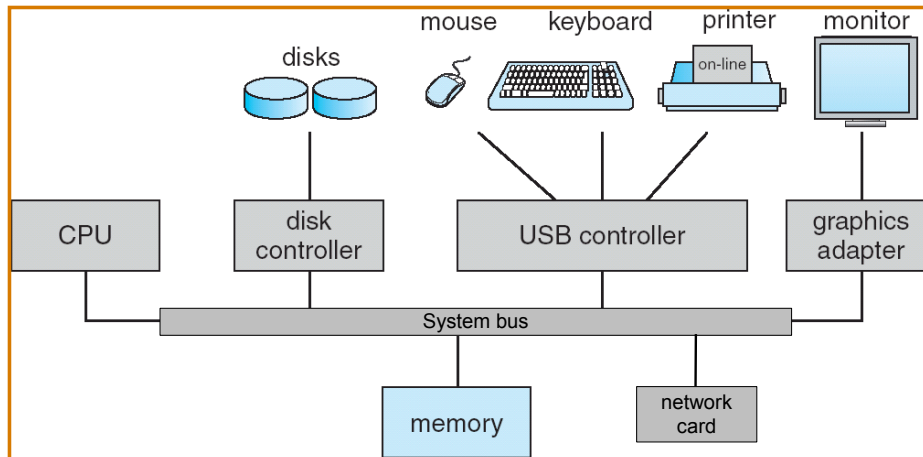
Recap: The OS Abstraction



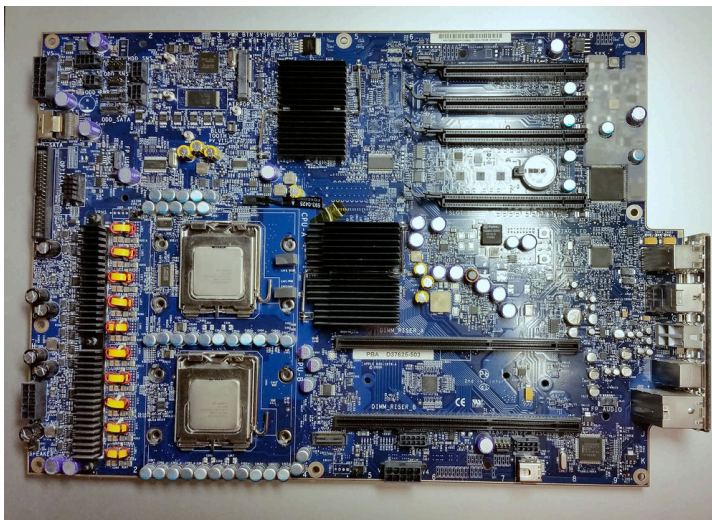
OS Resource Management



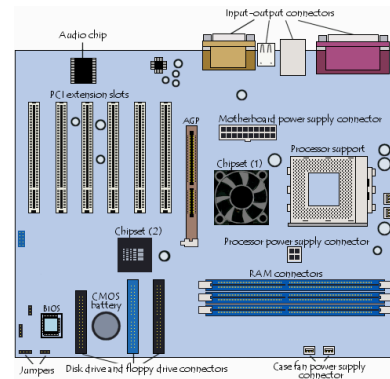
Today: OS and Computer Architecture



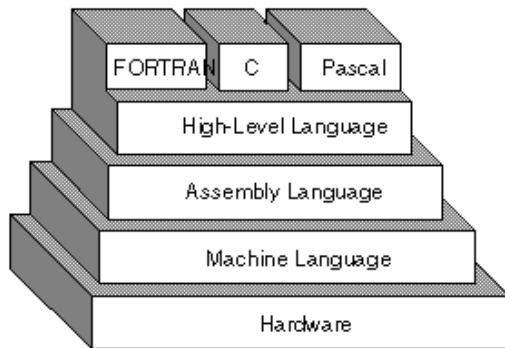
Computer Architecture



Logic board



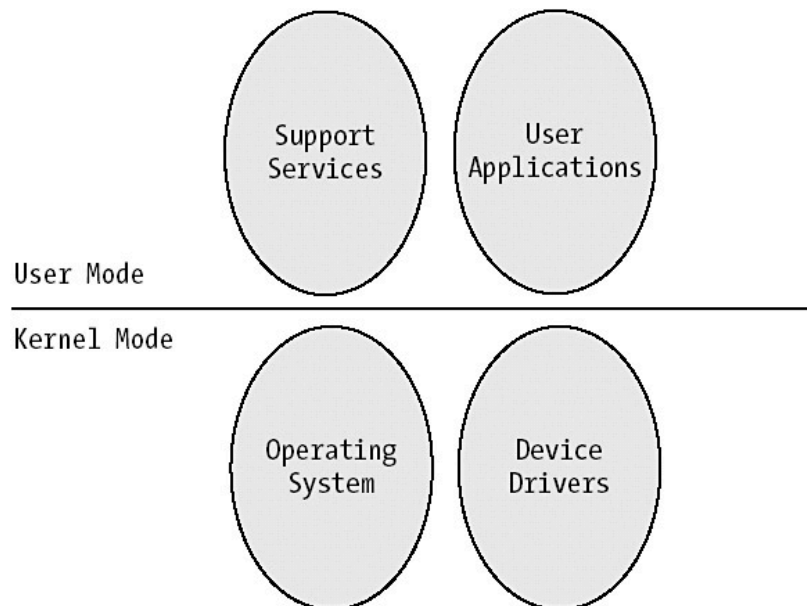
Assembly Language



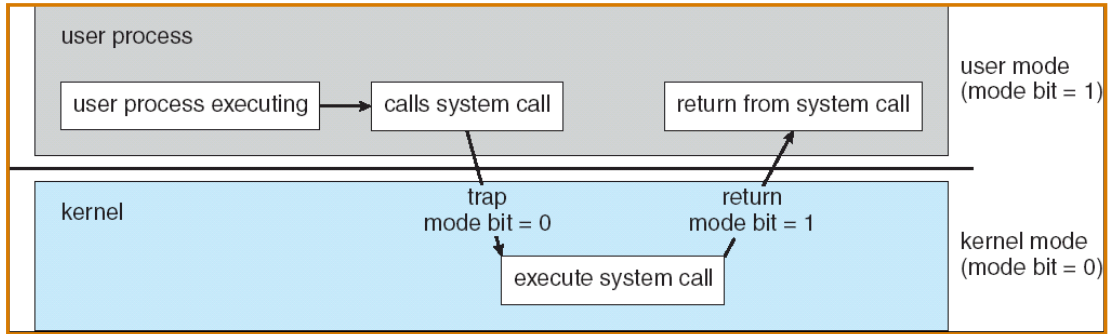
; Example of IBM PC assembly language
; Accepts a number in register AX;
; subtracts 32 if it is in the range 97-122;
; otherwise leaves it unchanged.

```
SUB32 PROC ; procedure begins here
    CMP AX,97 ; compare AX to 97
    JL DONE ; if less, jump to DONE
    CMP AX,122 ; compare AX to 122
    JG DONE ; if greater, jump to DONE
    SUB AX,32 ; subtract 32 from AX
DONE: RET ; return to main program
SUB32 ENDP ; procedure ends here
```

Protection: User and Kernel Mode



System Calls



Traps



Memory Addresses

Trap Vector

0: 0x00080000	Illegal address
1: 0x00100000	Memory violation
2: 0x00100480	Division by zero
3: 0x00123010	System call

⋮

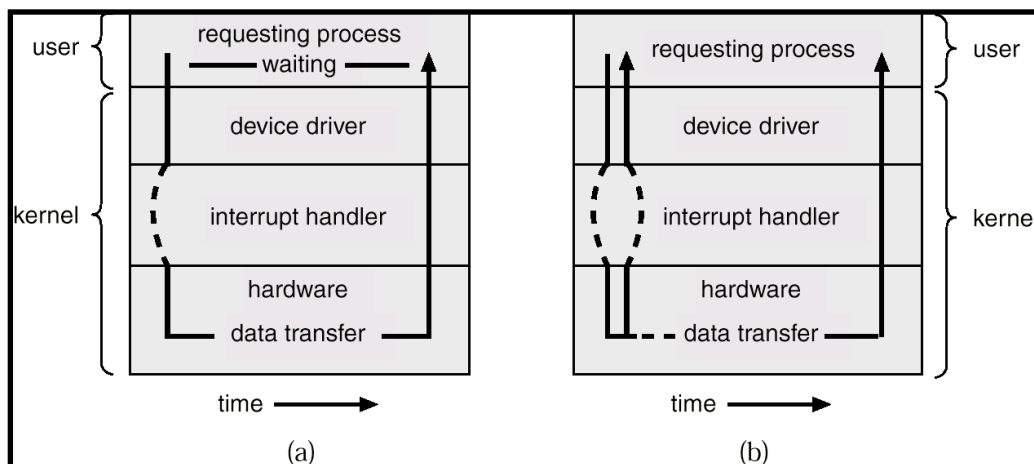
I/O Control & Interrupts



Interrupt
Vector

0: 0x2ff080000	keyboard
1: 0x2ff100000	mouse
2: 0x2ff100480	timer
3: 0x2ff123010	disk 1

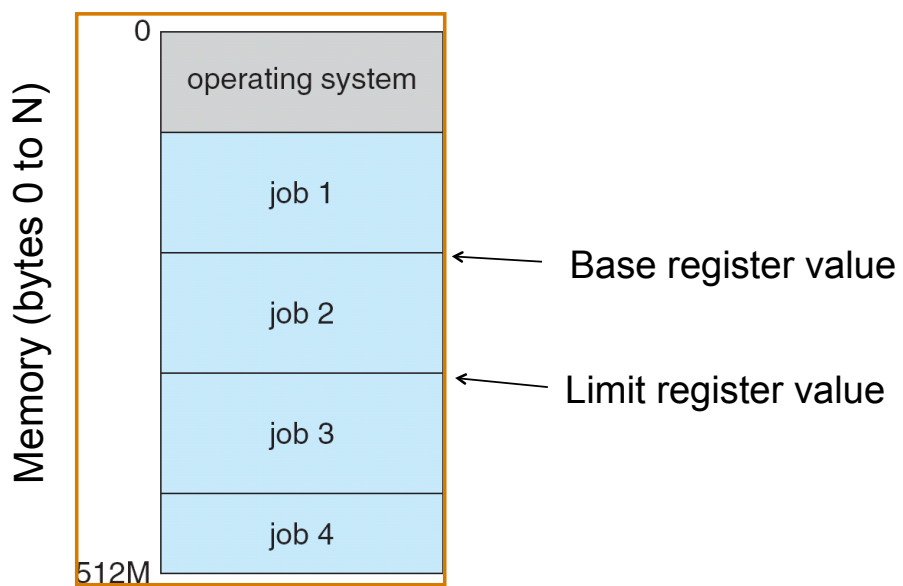
Synchronous & Asynchronous I/O



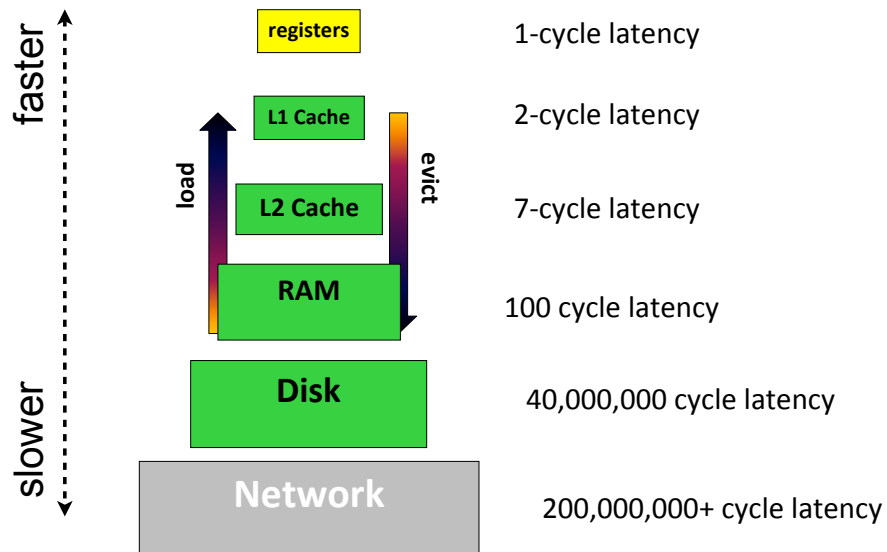
Hardware Timer



Memory Protection



Caches in the Memory Hierarchy



Summary of Architecture Support

OS Service	Hardware Support
Protection	Kernel/user mode, protected instructions, base/limit registers
Interrupts	Interrupt vectors
System calls	Traps and trap vectors
I/O	Interrupts
Scheduling, error recovery, accounting	Timer
Synchronization	Atomic instructions
Virtual memory	Translation look-aside buffers