Operating Systems

Thomas Schwarz, SJ

History

- Early computers
  - User load their program and run it
  - Assemblers, compilers, and linkers are software modules used to transform to machine code
- As performance increases:
  - Useful proportion = Time to use a computer : Time to load a computer becomes less
- Another machine program: Run queue that selects which e.g. magnetic tape to load next job

History

- Example: Leo 1 (Lyons Electronic Office)
  - Modeled on Cambridge EDSAC built by 1951
  - First computer dedicated to commercial business applications
  - Multiple Input/Output (I/O) buffers: paper readers and punches, punched card readers and punches, + (later) magnetic tape
  - Processed daily orders, payroll, inventory

History

- IBM 701 System Monitor (1955):
  - Checks running jobs and resource consumption
- GM-NAA I/O General Motors: First Operating System (OS) [1956]:
  - Provides Batch processing
    - Automatically execute a new program once current job has finished
  - Built using shared functions that provide access to various I/O devices
  - "**Resident**" program: Runs always and calls user jobs

History

- SHARE Operating System (SOS) 1959:
  - Allows sharing of programs
- Early OS:
  - Diversity: Each computer type had its own OS
  - 1960s: IBM concentrates on System/360 series
  - Decides on a single OS but:
    - Encounters many difficulties
      - Ends up with at least four OS

History

- Time sharing:
  - Sharing of a computer among many users **at the same time**
  - Needs the capability to connect multiple terminals to the same machine
  - Computer spends a small time slice on each user
    - State of interaction with user is stored after slice is up and reloaded when it is that user's turn
  - Implemented after many tries in the early 1970s

History

- Time sharing:
  - Includes first security problem:
    - How to protect one user from accessing another user's resources (files, programs)
    - To prevent unintentional damage
- Central idea: Permissions: what rights has a user over an entity
  - Dealt with access control lists: who has rights on this object
  - Dealt with capabilities: what rights on what objects has this entity

Unix

- Unix (AT&T Bell Labs, late 1960s / UC Berkeley):
  - A simpler version of Multics: less performance problems
  - Written in C : a higher level language, so it is easy to port
  - Ancestor of a large family of OS including MacOS and Linux

Dos / Windows

- 1980: IBM is worried about loosing out the personal computer market such as it lost the mini-computer market
- Pushes the IBM PC
  - Selects Intel 8086 because it was available
  - Needs an OS
    - Decides to find an outside provider
      - because "money is in hardware, not software"

Dos / Windows

- Disk Operating System (DOS)
  - Single user system targeting Intel 8086 using floppy disks
    - Developed through six different versions
- Windows (Graphical User Interface)
- Slow development towards a consumer facing and a server facing market

Linux

- Torvalds 1991
  - Based on the Minimal Unix -- MINIX teaching operating system
  - 1983: Stallman: "free" software foundation with GNU General Public Licence
    - Free: can be modified
    - Can be commercial

OS Functions

- Modern Operating System Functions
  - Multi-tasking:
    - Programs can run concurrently
      - Each program receives a slice of CPU time
      - Program stops when slice is up or when a long lasting command is issued (such as getting data from a disk, waiting for IO)
      - Programs can be put to sleep and OS wakes them up when certain conditions are true (e.g. network packet arrived)
      - Programs can have different priorities

OS Functions

- Multi-user
  - OS can maintain separation between different users
  - OS mediates privileges:
    - Access rights
      - Access Control Lists
      - Capabilities

OS Functions

- Real-time OS
  - Mechanisms to guarantee programs to finish in time
  - E.g.: Apollo 11 Guidance Computer

OS Components

- Kernel
  - Random-access memory (RAM)
    - Kernel decides which process can us which memory
    - How to deal with lack of memory

OS Components

- Input/Output Devices
  - Keyboards, mouses, storage, printer, USB devices, network adapters, displays
  - Programmed with device drivers
  - Kernel mediates requests from applications to access I/O devices

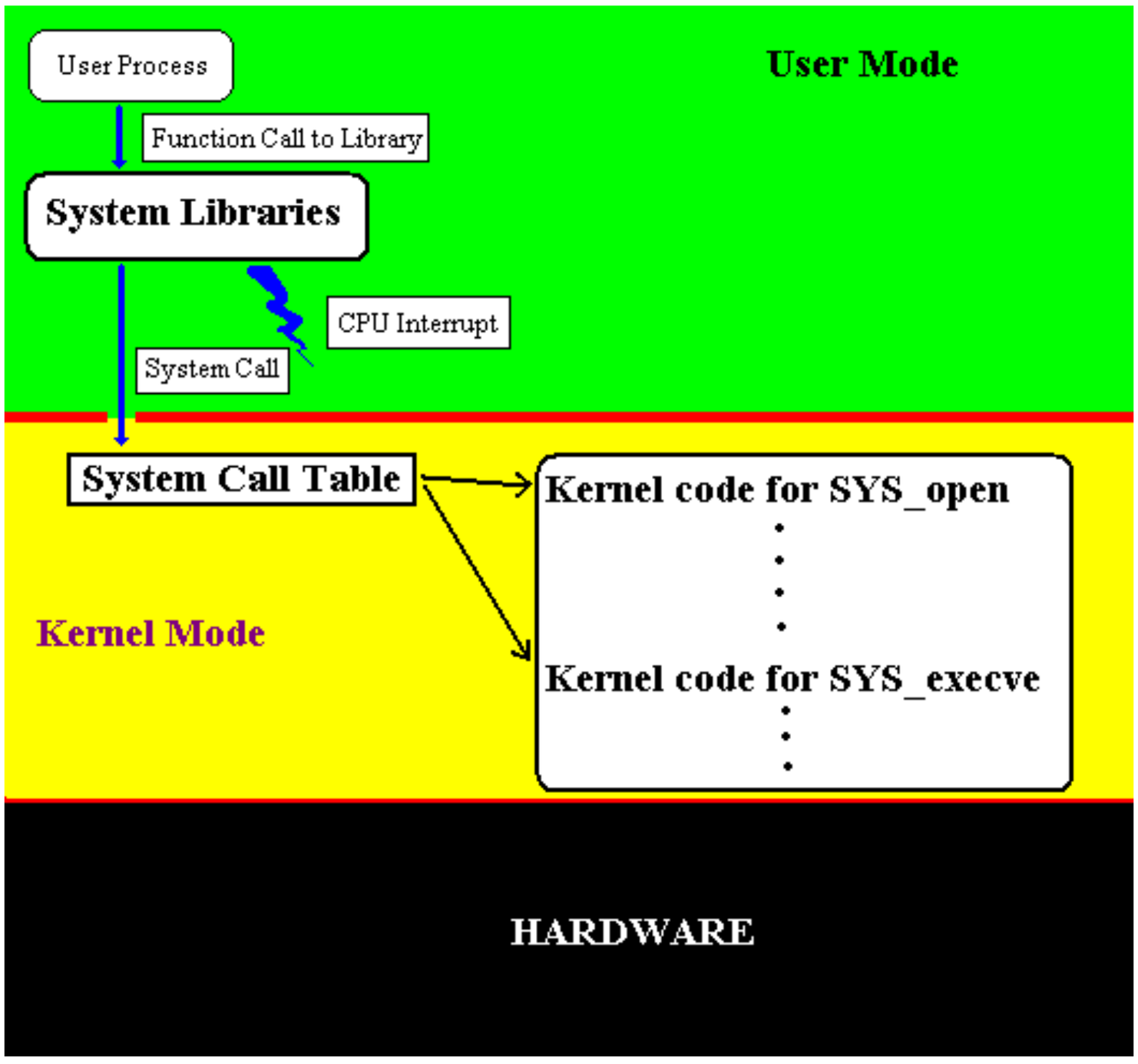
OS Components

- Memory Management
  - Only kernel has full access to the memory system
  - Paging / Segmentation:
    - Memory is divided in abstract pages (of 4KB)
    - Pages can be in DRAM or temporarily in disk
- Virtual Addressing:
  - User process uses virtual addresses that kernel translates to actual addresses
  - Allows kernel control of who can access what memory

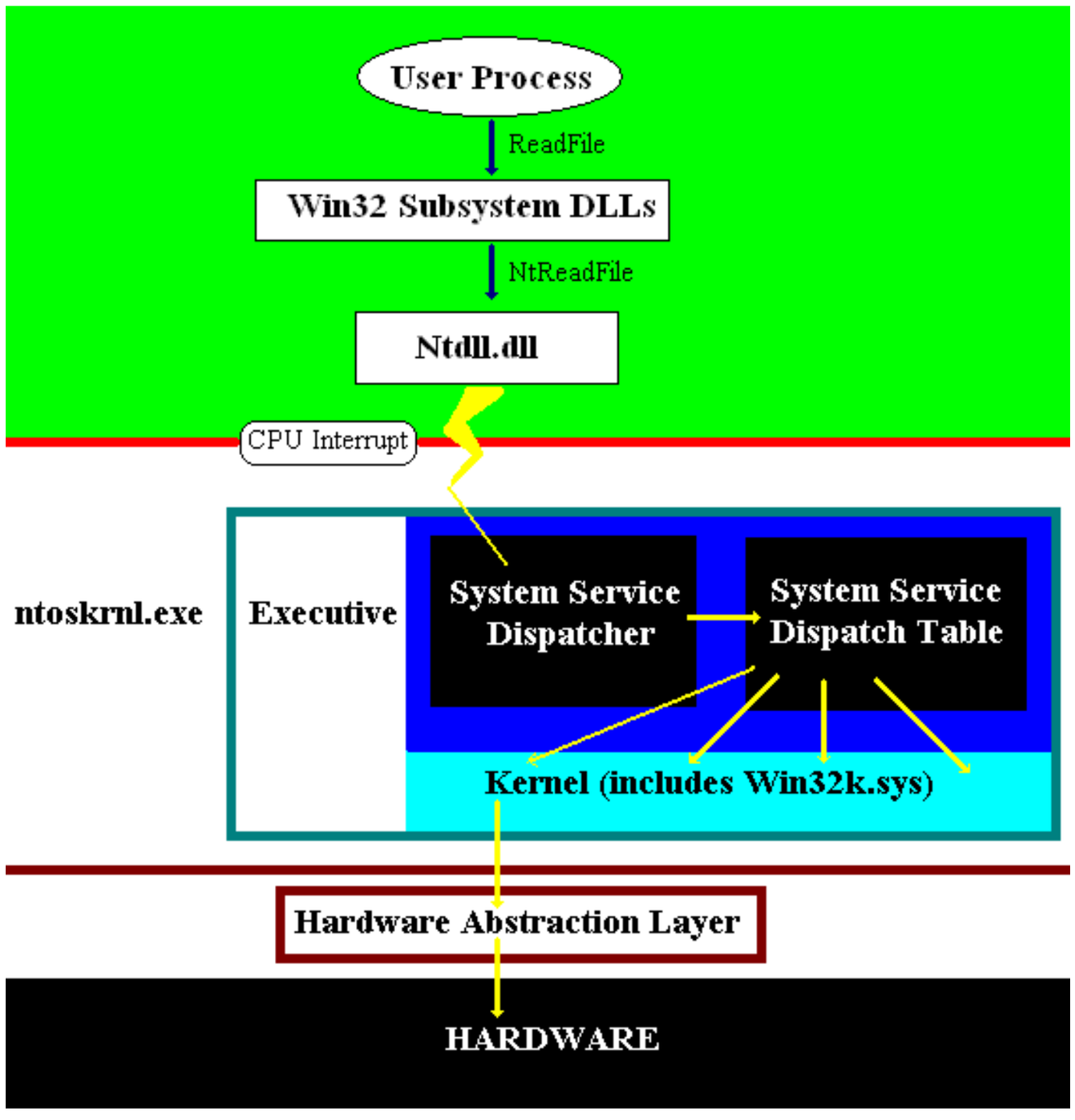
OS Components

- System calls:
  - A user process request a service from the OS kernel
  - Cannot call this service directly
  - Creates a system call through a systems library
  - Call places parameters in a register, then uses a hardware instruction that is a CPU interrupt
  - Control passes to the kernel
  - Kernel does the job
  - Restarts the user process

OS Components



OS Components



OS Components

- Process Management
  - User application run one or more process
  - Each process consists of one or more threads - continuous strands of execution
    - Modern CPUs allow several threads to execute in parallel
- OS determines which process can run

OS Components