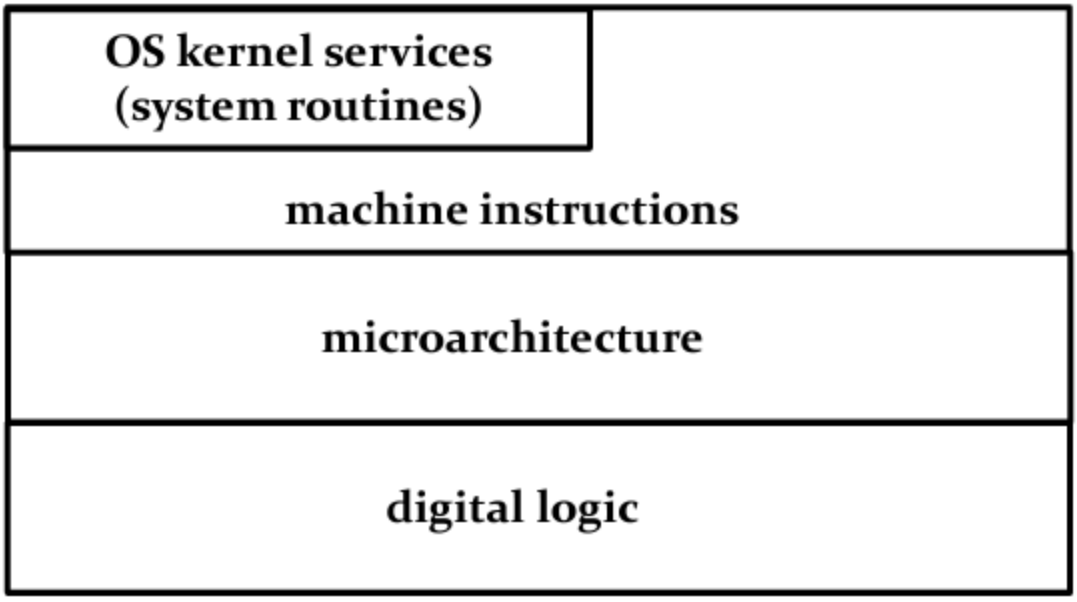
Introduction

Definition of OS

- The operating system controls the hardware and coordinates its use among the various application programs for the various users.
- A general definition is that the operating system is the one program running at all times on the computer — usually called the kernel. Along with the kernel, there are two other types of programs: system programs, which are associated with the operating system but are not necessarily part of the kernel, and application programs which, include all programs not associated with the operation of the system.

Low-level application programming interface



Exemplary concepts introduced by OS

- OS introduces concepts such as:

- Files (filesystem). Processor can only make disk controller read or write a disk sector, but the concept of files doesn't exist on the CPU level.
- Processes. CPU is only aware of the next instruction to be executed, OS introduces the concept of a process.

Tasks of OS

- Defining the user interface (shell)
- Providing a file system
- Providing environment to run programs
  - mechanisms of actual code execution, safety
  - mechanisms of process synchronization, communication
- Managing the I/O devices
- Providing procedures for handling basic errors

Resource Management

- Resource management consists of:
  - Resource allocation
  - Planning optimal allocation
  - Access authorization, protection
  - etc.
- Resources managed by OS:
  - Processor
  - Memory
  - I/O devices
  - File system

Classification of Operating Systems

- With respect to processing:
  - Interactive systems (on-line systems):
    - Users directly interact with OS
    - Task execution starts immediately after submission
    - Most popular type
  - Batch systems (off-line systems)
    - Task execution can be delayed
    - Multiple users can use it at the same time
- With respect to the number of programs executed
  - Single-tasking systems
    - It is not possible to start a new user program until previous task has finished
  - Multi-tasking/Multiprogramming systems
    - Multiple programs can be executed at the same time
- With respect to number of users supported
  - Single-user systems
    - All of computer’s resources are accessed by only one user
  - Multi-user systems
    - Many users can access computer’s resources
    - Examples: UNIX, Linux, MS Windows Server
- Other types:
  - Real-time operating systems
  - Distributed systems
  - Server, mainframe systems

- Embedded systems

OS program execution

- Von Neumann architecture:
  - Programs and data are stored in the same memory.
  - Instructions are executed one after another
- Instruction cycle (FDX-cycle, fetch decode execute) — sequence of actions that the processor performs to execute each machine instruction in a program.
- Typical FDX-cycle is divided into:
  - Instruction loading phase
  - Operand loading phase
  - Instruction execution phase
  - Operand storing phase
  - Checking for interrupts
- Interrupts:
  - An interrupt is a state of computer system in reaction to an asynchronous event (an event which happens independently of processing and its precise moment of occurrence cannot be predicted)
  - It consists in the processor automatically saving its state of execution (context switch), and starting execution of an interrupt handler.
  - Sources of interrupts:
    - External interrupts — raised by external devices
    - Software interrupts — raised by executed program with special instruction (interrupt call)
    - Diagnostic interrupts (exception handling) — raised to handle an exceptional state of processing like software errors, or hardware errors.

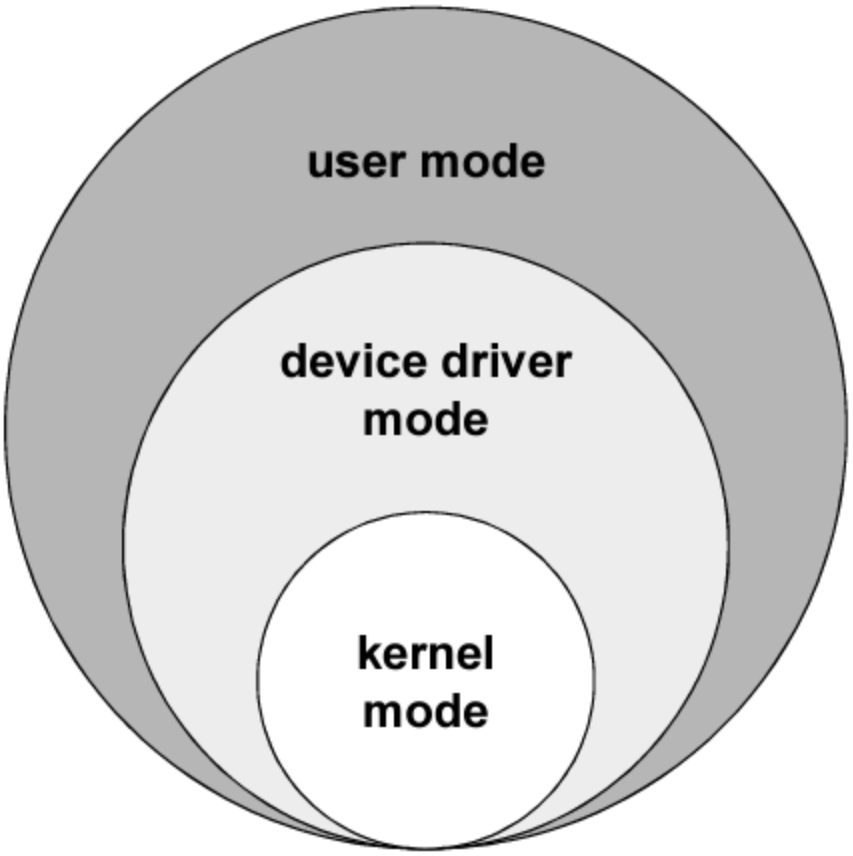
Memory protection

- Exposing physical memory to processes has several major drawbacks.
- User programs can trash the OS, intentionally or by accident
- To solve those issues the concept of an address space was introduced.
- Address space is the set of addresses that a process can use to address memory. Each process has its own address space, independent of those belonging to other processes.
- A simple way to assign each program its own address space is with using base and limit register. This method is no longer used in practice due to some better and more complicated schemes on modern CPUs.
- A CPU is equipped with two special register called base and limit registers.
- When a process is run, the base register is loaded with the physical address where its program begins in memory and the limit register is loaded with the length of the program.
- Every time a process references memory the CPU automatically check whether this address is less than the value at the limit register, if it isn't then the segmentation fault is generated. Then the value at the base address is added to this "user" address to get the true memory location which the program is trying to access.

Protection rings

- Protection rings are a mechanism to protect computer from faults and malicious behaviour.
- For example the kernel is trusted and privileged component, therefore it makes sense to give the kernel code more rights and authority.
- Protection rings distinguish unsafe, user provided code from trusted and secure code such as the kernel, or device drivers.
- In protection rings, the innermost ring, the ring 0 has the most privileges, then the ring 1 has less privileges, the ring 3 has even less privileges, and so on.

- It is possible to switch between protection rings, for example by calling a special function (like `syscall`).



CPU protection — timer interrupt

- Timer interrupt is a kind of external interrupt, fired by the timer at constant intervals of time.
- As a result, the kernel regains control of the system and may switch the context to another process.