

1. <b>Communication Primitives</b>	semaphores message passing monitors
2. <b>CPU</b>	Hardware circuit within a computer that carries out instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions.
3. <b>CPU Pipeline</b>	a set of data processing elements connected in series, where the output of one element is the input of the next one. The elements of a pipeline are often executed in parallel or in time-sliced fashion; in that case, some amount of buffer storage is often inserted between elements
4. <b>Interprocess Communication</b>	<p>a set of programming interfaces that allow a programmer to coordinate activities among different program processes that can run concurrently in an operating system. This allows a program to handle many user requests at the same time. Since even a single user request may result in multiple processes running in the operating system on the user's behalf, the processes need to communicate with each other. The IPC interfaces make this possible. Each IPC method has its own advantages and limitations so it is not unusual for a single program to use all of the IPC methods.</p> <p>IPC methods include pipes and named pipes; message queueing; semaphores; shared memory; and sockets.</p>
5. <b>Interrupt Handler (First level)</b>	<p>The operating system instructs the hardware to jump to the first level interrupt handler whenever an interrupt occurs; there may be several FLIHs if interrupts are 'vectored' (i.e. different interrupts jump to different places); needs to save the machine state.</p> <p>The first level interrupt handler (FLIH) is the part of the operating system which is responsible for responding to signals both from the outside world (interrupts) and from within the computing system itself (error traps and system calls). We shall refer collectively to both types of signal as interrupts, and use the adjectives 'external' and 'internal' to distinguish between them where necessary. The function of the FLIH is twofold:</p> <ol style="list-style-type: none"> <li>1. to determine the source of the interrupt</li> <li>2. to initiate service of the interrupt</li> </ol>
6. <b>Kernel</b>	<p>A component of the Operating System.</p> <p>The kernel is a computer program that constitutes the central core of a computer's operating system. It has complete control over everything that occurs in the system. As such, it is the first program loaded on startup, and then manages the remainder of the startup, as well as input/output requests from software, translating them into data processing instructions for the central processing unit. It is also responsible for managing memory, and for managing and communicating with computing peripherals, like printers, speakers, etc. The kernel is a fundamental part of a modern computer's operating system</p> <p>Contents of the Kernel</p> <ol style="list-style-type: none"> <li>1. First Level Interrupt Handler</li> <li>2. Low Level Scheduler (Dispatcher)</li> <li>3. Communication Primitives</li> </ol> <p>3 Types of Kernel</p> <ol style="list-style-type: none"> <li>1. Monolithic</li> <li>2. Microkernel</li> <li>3. Hybrid</li> </ol>
7. <b>Low Level Scheduler</b>	This performs process start-up and switching. Start-up is a special case of switching. Process start-up involves: allocate and initialise a Process Control Block (PCB); add the PCB to the list of runnable processes; update other system tables.
8. <b>Process</b>	An instance of a computer program that is being executed. It contains the code, and its current activity (state).

9. **Processes** Process

**vs**

**Threads**

Each process provides the resources needed to execute a program. A process has a virtual address space, executable code, open handles to system objects, a security context, a unique process identifier, environment variables, a priority class, minimum and maximum working set sizes, and at least one thread of execution. Each process is started with a single thread, often called the primary thread, but can create additional threads from any of its threads.

Thread

A thread is the entity within a process that can be scheduled for execution. All threads of a process share its virtual address space and system resources. In addition, each thread maintains exception handlers, a scheduling priority, thread local storage, a unique thread identifier, and a set of structures the system will use to save the thread context until it is scheduled. The thread context includes the thread's set of machine registers, the kernel stack, a thread environment block, and a user stack in the address space of the thread's process. Threads can also have their own security context, which can be used for impersonating clients.

---

10. **Processor** the logic circuitry that responds to and processes the basic instructions that drive a computer. Also called the CPU or Central Processing Unit

---

11. **Scheduler** process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy.

3 Types of Scheduler

Long Term

Short Term

Medium Term

---

12. **Thread** a collection of instructions which together form a specific task. Eg. Single-threaded, multi-tasking operating systems

Each processor provides a separate "thread of control" within a single program in memory.

---