OPERATING SYSTEM

Process Concept:

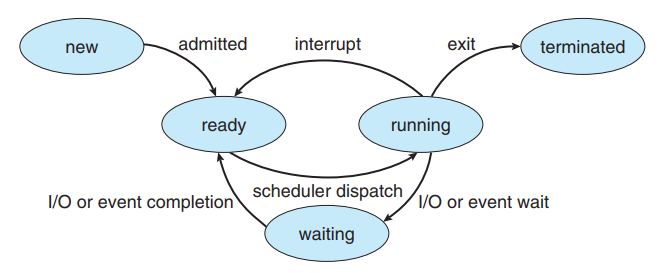
A question that arises in discussing operating systems involves what to call all the CPU activities. A batch system executes jobs, whereas a time-shared system has user programs, or tasks. Even on a single-user system, a user may be able to run several programs at one time: a word processor, a Web browser, and an e-mail package. And even if a user can execute only one program at a time, such as on an embedded device that does not support multitasking, the operating system may need to support its own internal programmed activities, such as memory management. In many respects, all these activities are similar, so we call all of them processes. Informally, as mentioned, a process is a program in execution.

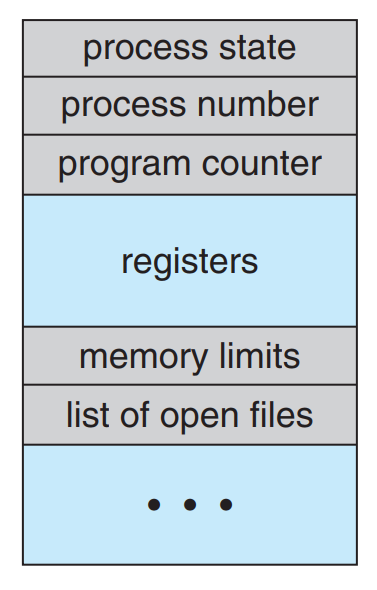
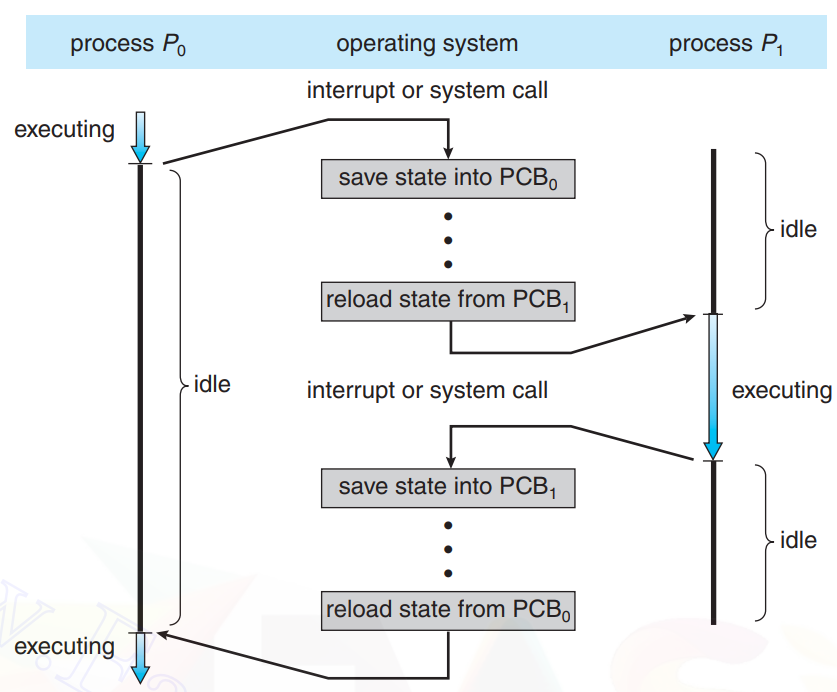
**Process State:**

As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. A process may be in one of the following states:

* **New:** The process is being created.
* **Running:** Instructions are being executed.
* **Waiting:** The process is waiting for some event to occur (such as an I/O completion or reception of a signal).
* **Ready:** The process is waiting to be assigned to a processor.
* **Terminated:** The process has finished execution.

**Process Control Block:**

****Each process is represented in the operating system by a process control block(PCB)—also called a task control block.

* **Process State:** The state may be new, ready, running, waiting, halted, and so on.
* **Program counter:** The counter indicates the address of the next instruction to be executed for this process.
* **CPU registers:** the registers vary in numbers and type, depending on the computer architecture. They include accumulators, index registers, stack pointers, and general-purpose counter, this state information must be saved when an interrupt occurs, to allow the process to be continued correctly afterward.
* **CPU scheduling information:** This information includes a process priority, pointers to scheduling queues, and any other scheduling parameters.
* **Memory management information:** This information may include such items as the value of the base and limit registers and the page tables, or the segment tables, depending on the memory system used by the operating system.
* **Accounting information:** This information includes the amount of CPU and real time used, time limits, account numbers, job or process numbers, and so on.
* **I/O status information.** This information includes the list of I/O devices allocated to the process, a list of open files, and so on.

**Threads:**

The process model discussed so far has implied that a process is a program that  
performs a single thread of execution. For example, when a process is running  
a word-processor program, a single thread of instructions is being executed.  
This single thread of control allows the process to perform only one task at  
a time. The user cannot simultaneously type in characters and run the spell  
checker within the same process, for example. Most modern operating systems  
have extended the process concept to allow a process to have multiple threads  
of execution and thus to perform more than one task at a time. This feature  
is especially beneficial on multicore systems, where multiple threads can run  
in parallel. On a system that supports threads, the PCB is expanded to include  
information for each thread. Other changes throughout the system are also  
needed to support threads.

Process Scheduling:

The objective of multiprogramming is to have some process running at all  
times, to maximize CPU utilization. The objective of time sharing is to switch the  
CPU among processes so frequently that users can interact with each program while it is running. To meet these objectives, the process scheduler selects an available process (possible from a set of several available processes) for program execution on the CPU. For a single processor system, there will never be more than one running process. If there are more processes, the rest will have to wait until the CPU is free and can be rescheduled.

**Scheduling Queues:**

As processes enter the system, they are put into a job queue, which consists  
of all processes in the system. The processes that are residing in main memory  
and are ready and waiting to execute are kept on a list called the ready queue.  
This queue is generally stored as a linked list. A ready-queue header contains  
pointers to the first and final PCBs in the list. Each PCB includes a pointer field  
that points to the next PCB in the ready queue.