Processes

Eric McCreath

Processes

Processes are useful as they effectively allow the computer system to perform many tasks at once. This is useful for a variety of reasons including:

- Facilitates multiÂ-programming which improves CPU utilisation. Because while programs are waiting on I/O events, other programs may run.
- Facilitates timeÂ-sharing.(Each user gets a slice on the CPU resource.)
- The user may run different programs at once. (text editor, database, calculator, compiler, etc.)
- Helps modularity in program design.

Processes

- A process is a program in execution.
- To provide multiprogramming the operating system will maintain multiple independent program contexts. The operating system will manage the simultaneous execution of programs. A process is program in one of these execution contexts.
- Processes are the basic unit of work in an operating system.
- A process is an active entity.

Information about a process

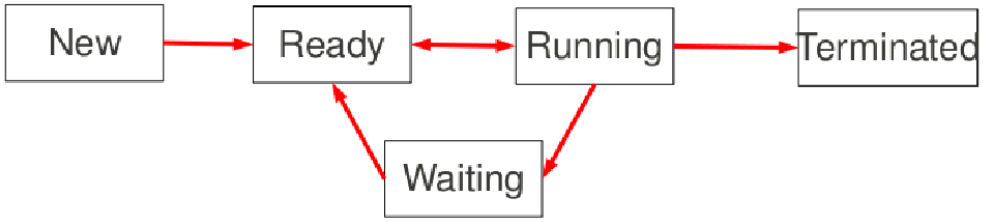
A process is a program in execution. It is described by the following information:

- Program Code (Referred to as the text).
- Program Counter (or instruction pointer). The next instruction in the text to execute.
- Scheduling information.
- Stack. This contains the local variables of the program.
- Registers.
- Data Section. The global variables.
- IO information. (e.g. open files)
- Process State.
- Accounting information. This may be used to help record and control the resources used by this process.

Process State

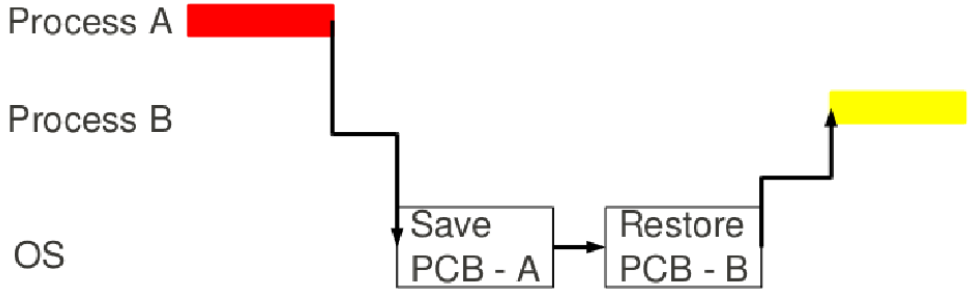
A process may be in one of the following states:

- Running : The process is assigned to a CPU and executing instructions.
- Waiting : The process is waiting for an I/O device or a signal.
- Ready : The process is ready to run.
- New : The process is just starting.
- Terminated : The process has completed execution.



Context Switch

A Context Switch changes the process a CPU is executing. When an OS performs a context switch from process A to process B the current state of process A is saved in the PCB for A and process B's state is restored from the PCB for B.



Theads

A thread is the basic unit of execution within process. Simple programs normal just have one thread of execution running. However, it is sometimes useful for a program to have multiple units of execution within the program. An OS many provide this resource (kernel threads), or it may be done within user space (user-level threads).

A process (or task) consists of resources such as: open files, the text, data section, signals, and a set of threads.

Each thread, sometimes called a lightweight process (LWP), consists of a program counter, a register set, and a stack space.

A traditional process is called a heavyweight process.

heavyweight process = task with one thread.

Threads

A thread is the basic unit of CPU utilization. A group of threads combine together to form a task/process. These threads share text, data, open files, capabilities (protection), etc