**Daemon Definition**

|  |  |
| --- | --- |
|  | A *daemon* is a type of program on Unix-like [operating systems](http://www.linfo.org/operating_systems_list.html) that runs unobtrusively in the background, rather than under the direct control of a user, waiting to be activated by the occurrence of a specific event or condition.  Unix-like systems typically run numerous daemons, mainly to accommodate requests for services from other computers on a network, but also to respond to other programs and to hardware activity. Examples of actions or conditions that can trigger daemons into activity are a specific time or date, passage of a specified time interval, a file landing in a particular directory, receipt of an e-mail or a Web request made through a particular communication line. It is not necessary that the perpetrator of the action or condition be aware that a daemon is *listening*, although programs frequently will perform an action only because they are aware that they will implicitly arouse a daemon.  Daemons are usually instantiated as *processes*. A process is an *executing* (i.e., running) instance of a program. Processes are managed by the *kernel* (i.e., the core of the operating system), which assigns each a unique *process identification number* (PID).  There are three basic types of processes in Linux: interactive, batch and daemon. Interactive processes are run interactively by a user at the *command line* (i.e., all-text mode). Batch processes are submitted from a queue of processes and are not associated with the command line; they are well suited for performing recurring tasks when system usage is otherwise low.  Daemons are recognized by the system as any processes whose *parent process* has a PID of one, which always represents the process *init*. init is always the first process that is started when a Linux computer is *booted up* (i.e., started), and it remains on the system until the computer is turned off. init *adopts* any process whose *parent* process *dies* (i.e., terminates) without waiting for the *child* process's status. Thus, the common method for launching a daemon involves *forking* (i.e., dividing) once or twice, and making the parent (and grandparent) processes die while the child (or grandchild) process begins performing its normal function.  Some daemons are launched via *System V init scripts*, which are *scripts* (i.e., short programs) that are run automatically when the system is booting up. They may either survive for the duration of the session or be regenerated at intervals.  Many daemons are now started only as required and by a single daemon, *xinetd* (which has replaced *inetd* in newer systems), rather than running continuously. xinetd, which is referred to as a *TCP/IP super server*, itself is started at boot time, and it listens to the *ports* assigned to the processes listed in the */etc/inetd.conf* or in */etc/xinetd.conf* configuration file. Examples of daemons that it starts include *crond* (which runs scheduled tasks), *ftpd* (file transfer), *lpd* (laser printing), *rlogind*(remote login), *rshd* (remote command execution) and *telnetd* (telnet).  In addition to being launched by the operating system and by application programs, some daemons can also be started manually. Examples of commands that launch daemons include *binlogd* (which logs binary events to specified files), *mysqld* (the MySQL databse server) and *apache* (the Apache web server).  In many Unix-like operating systems, including Linux, each daemon has a single *script* (i.e., short program) with which it can be terminated, restarted or have its status checked. The handling of these scripts is based on *runlevels*. A runlevel is a configuration or operating state of the system that only allows certain selected processes to exist. Booting into a different runlevel can help solve certain problems, including repairing system errors.  The term *daemon* is derived from the daemons of Greek mythology, which were supernatural beings that ranked between gods and mortals and which possessed special knowledge and power1. For example, Socrates claimed to have a daemon that gave him warnings and advice but never coerced him into following it. He also claimed that his daemon exhibited greater accuracy than any of the forms of divination practiced at the time.  The word *daemon* was first used in a computer context at the pioneering Project MAC (which later became the MIT Laboratory for Computer Science) using the IBM 7094 in 1963. This usage was inspired by Maxwell's daemon of physics and thermodynamics, which was an imaginary agent that helped sort molecules of different speeds and worked tirelessly in the background. The term was then used to describe background processes which worked tirelessly to perform system chores. The first computer daemon was a program that automatically made tape backups. After the term was adopted for computer use, it was rationalized as an acronym for *Disk And Execution MONitor*.  On the Microsoft Windows operating systems, programs called *services* perform the functions of daemons, although the term *daemon* is now sometimes being used with regard to those systems as well.  \_\_\_\_\_\_\_\_  1 A daemon should be distinguished from a *demon*, which is an evil spirit in some religions.  Created August 16, 2005. Copyright © 2005 The Linux Information Project. All Rights Reserved. |