Chapter – 6 MONITORING AND MANAGING LINUX PROCESSES

Objectives:

- Get information about programs running on the system so that you can determine status, resource use, and ownership, so you can control them.
- Use Bash job control to manage multiple processes started from the same terminal session.
- Control and terminate processes that are not associated with your shell, and forcibly end user sessions and processes.
- Describe what load average is and determine processes responsible for high resource use on a server.

LISTING PROCESSES

DEFINITION OF A PROCESS

A *process* is a running instance of a launched, executable program. A process consists of:

- An address space of allocated memory
- Security properties including ownership credentials and privileges
- One or more execution threads of program code
- Process state

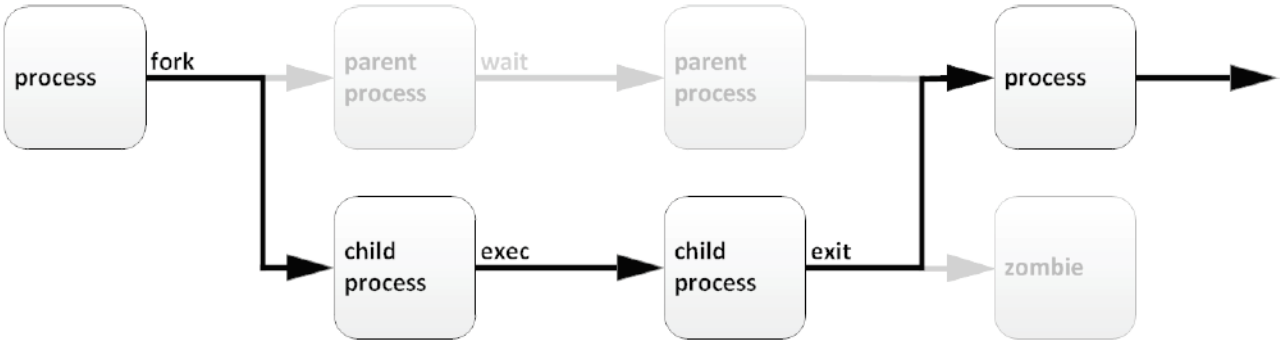

Figure: 6.1 Process Life Cycle

DESCRIBING PROCESS STATES

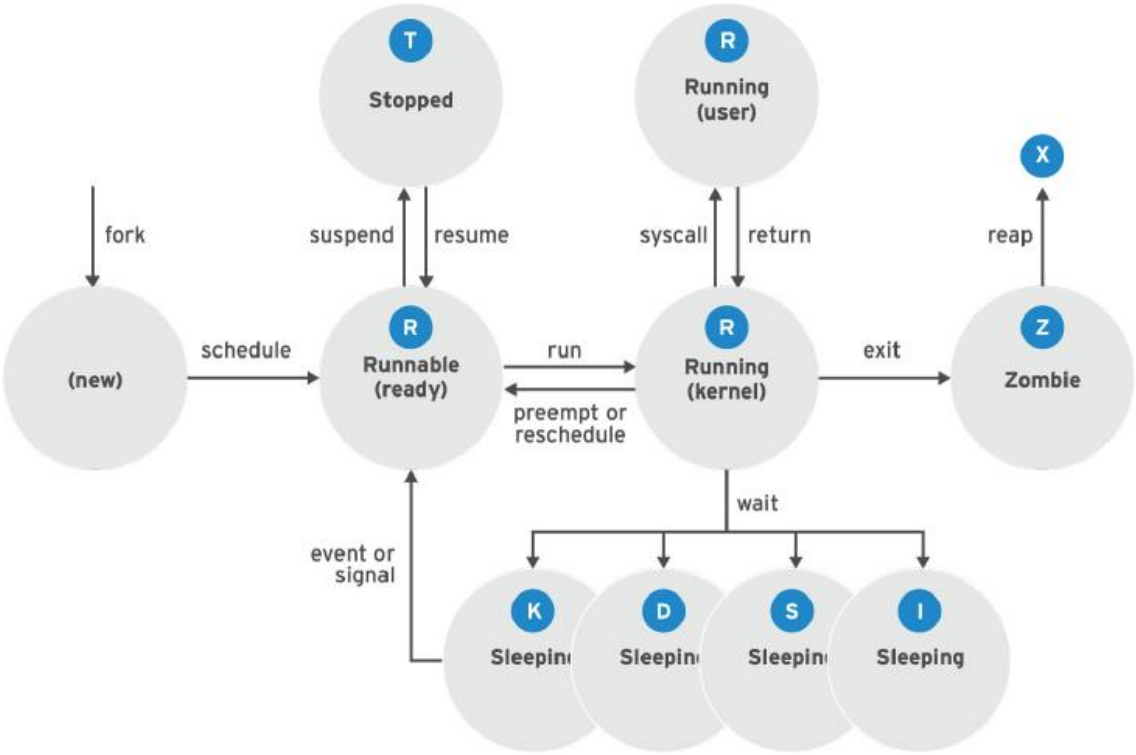

Figure: 6.2 LINUX Process States

Linux Process States

NAME	FLAG	KERNEL-DEFINED STATE NAME AND DESCRIPTION
Running	**R**	TASK_RUNNING: The process is either executing on a CPU or waiting to run. Process can be executing user routines or kernel routines (system calls), or be queued and ready when in the *Running* (or *Runnable*) state.
Sleeping	**S**	TASK_INTERRUPTIBLE: The process is waiting for some condition: a hardware request, system resource access, or signal. When an event or signal satisfies the condition, the process returns to *Running*.
	D	TASK_UNINTERRUPTIBLE: This process is also *Sleeping*, but unlike **S** state, does not respond to signals. Used only when process interruption may cause an unpredictable device state.
	K	TASK_KILLABLE: Identical to the uninterruptible **D** state, but modified to allow a waiting task to respond to the signal that it should be killed (exit completely). Utilities frequently display *Killable* processes as **D** state.
	I	TASK_REPORT_IDLE: A subset of state **D**. The kernel does not count these processes when calculating load average. Used for kernel threads. Flags TASK_UNINTERRUPTABLE and TASK_NOLOAD are set. Similar to TASK_KILLABLE, also a subset of state **D**. It accepts fatal signals.
Stopped	**T**	TASK_STOPPED: The process has been *Stopped* (suspended), usually by being signaled by a user or another process. The process can be continued (resumed) by another signal to return to *Running*.
	T	TASK_TRACED: A process that is being debugged is also temporarily *Stopped* and shares the same **T** state flag.
Zombie	**Z**	EXIT_ZOMBIE: A child process signals its parent as it exits. All resources except for the process identity (PID) are released.
	X	EXIT_DEAD: When the parent cleans up (*reaps*) the remaining child process structure, the process is now released completely. This state will never be observed in process-listing utilities.

LISTING PROCESSES

The ps command is used for listing current processes.

CONTROLLING JOBS

Bg

Fg

Jobs

KILLING PROCESSES

The KILL command is used to kill the process.

Fundamental Process Management Signals

SIGNAL NUMBER	SHORT NAME	DEFINITION	PURPOSE
1	HUP	Hangup	Used to report termination of the controlling process of a terminal. Also used to request process reinitialization (configuration reload) without termination.
2	INT	Keyboard interrupt	Causes program termination. Can be blocked or handled. Sent by pressing INTR key combination (**Ctrl**+**C**).
3	QUIT	Keyboard quit	Similar to SIGINT, but also produces a process dump at termination. Sent by pressing QUIT key combination (**Ctrl**+\).
9	KILL	Kill, unblockable	Causes abrupt program termination. Cannot be blocked, ignored, or handled; always fatal.
15 *default*	TERM	Terminate	Causes program termination. Unlike SIGKILL, can be blocked, ignored, or handled. The "polite" way to ask a program to terminate; allows self-cleanup.
18	CONT	Continue	Sent to a process to resume, if stopped. Cannot be blocked. Even if handled, always resumes the process.
19	STOP	Stop, unblockable	Suspends the process. Cannot be blocked or handled.
20	TSTP	Keyboard stop	Unlike SIGSTOP, can be blocked, ignored, or handled. Sent by pressing SUSP key combination (**Ctrl**+**Z**).