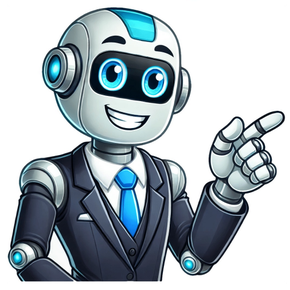
I'm not a robot



As we all know Linux is a multitasking and multi-user system. So, it allows multiple processes to operate simultaneously without interfering with each other. Process is one of the important fundamental concepts of the Linux OS. A process is an executing instance of a program that carries out different tasks within the operating system. What is a Process in Linux?In Linux, a process is a running instance of a program. When you execute a program, it becomes a process, an independent, executing entity with its own memory space. Each process is assigned a unique identifier, the Process ID (PID). Processes are fundamental to the functioning of the operating system and play a crucial role in multitasking, allowing the computer to execute multiple tasks concurrently. List Running Processes in Linux We can use multiple commands to list the running processes in Linux like ps, top, htop, and atop commands in Linux. We can also have a combination of commands to list the running processes in Linux. 1. The `ps` Command to List Running Processes in Linux The `ps` command, which stands for "process status," is like a computer tool that helps you see what's happening inside your Linux computer. Imagine your computer is doing several things simultaneously, like running different programs or apps. These are the processes and the `ps` command lets you take a quick look at them. When you use it without any special instructions, it shows you the processes that are connected to the window or screen you are currently using. But here's where it gets interesting: you can make the ps command show you exactly what you want to know by giving it special instructions, called options. These options let you customize the information you see, like finding out which programs are using the most computer power or checking what a specific user is doing. So, while it can give you a basic overview, the ps command's real strength is in letting you choose exactly what details you want to see about the processes on your computer. Syntax of `ps` Command in LinuxThe ps command provides a snapshot of the current processes on your system. The basic syntax is as follows: ps [options] Without any options, `ps` displays information about the processes associated with the current terminal session. However, to harness the full potential of the `ps` command, various options can be used to customize the output. Options for `ps` Command to List Running Processes in LinuxSome commonly used options List all ruining processes for all users. Lists all processes on the entire system, offering a complete overview of running tasks and programs. List all processes except session leaders (instances where the process ID is the same as the session ID) and processes not associated with a terminal. Lists all processes except session leaders, providing a filtered view of processes running on the system. Lists all processes except those that meet specific user-defined conditions. Displays the hierarchy of processes in a visual ASCII art format, illustrating parent-child relationships. Presents the output in the jobs format, providing detailed information such as process ID, session ID, and command. Lists all processes associated with the current terminal, aiding in focusing on tasks related to a specific terminal. Only lists running processes, useful for monitoring system performance. Expands the output to include additional information like CPU and memory usage. Specifies a username, listing processes associated with that user. Includes processes without a TTY, showing background processes not tied to a specific terminal session. 1) Simple process selection : Shows the processes for the current shell - ps list running processes in LinuxResult contains four columns of information. Where, PID - the unique process ID TTY - terminal type that the user is logged into TIME - amount of CPU in minutes and seconds that the process has been running CMD - name of the command that launched the process. Note - Sometimes when we execute ps command, it shows TIME as 00:00:00. It is nothing but the total accumulated CPU utilization time for any process and 00:00:00 indicates no CPU time has been given by the kernel till now. In above example we found that, for bash no CPU time has been given. This is because bash is just a parent process for different processes which needs bash for their execution and bash itself is not utilizing any CPU time till now. 2) View All Running Processes in Linux. To view all running processes, use either of the following options with the `ps` command: ps -A ps -A option to view all running processes in linux ps -e ps -e option to view all running processes in linux 3) List Processes Not associated with a Terminal in Linux View all processes except both session leaders and processes not associated with a terminal. ps -a ps -a option to list running processes in linuxNote - You may be thinking that what is session leader? A unique session is assign to every process group. So, session leader is a process which kicks off other processes. The process ID of first process of any session is similar as the session ID. 4) List All The Processes Except Session Leaders in Linux A session leader is a process that initiates other processes. View processes except session leaders: ps -d ps -d option to list running processes in linux 5) List All Processes Except those that Fulfill the Specified Conditions (negates the selection) Example - If you want to see only session leader and processes not associated with a terminal. Then, run ps -a -N OR ps -a --deselect 6) List All Processes Associated with this Terminal in Linux ps -T ps -T option to list running processes in linux7) View All Processes Owned By You Processes i.e same EUID as ps which means runner of the ps command, root in this case - ps -x ps -x option to list running processes in linux 8) Process selection by list Here we will discuss how to get the specific processes list with the help of ps command. These options accept a single argument in the form of a blank-separated or comma-separated list. They can be used multiple times. For example: ps -p "1 2" -p 3,4 9) Select Processes by Command Name This selects the processes whose executable name is given in cmdlist. There may be a chance you won't know the process ID and with this command it is easier to search. Syntax : ps -C command_name Example : ps -C systemd ps -C option to list running processes in linux 10) Select by group ID or name The group ID identifies the group of the user who created the process. Syntax: ps -G group_name ps --Group group_name Example : ps -G root View by group id : ps -g 1 View processes by process ID : Syntax: ps p process_id ps -p process_id ps --pid process_id Example: ps p 27223 You can view multiple processes by specifying multiple process IDs separated by blank or comma - Example : ps -p 1 904 27223 Here, we mentioned three process IDs - 1, 904 and 27223 which are separated by blank. 2. The `top` Command to List Running Processes in LinuxIn Linux, the `top` command is a dynamic and interactive tool that provides real-time information about system processes. It offers a comprehensive view of running processes, system resource utilization, and other critical system metrics. This article explores how to effectively use the top command to monitor and manage processes. Launching topTo launch the top command, open a terminal and simply type: top list all processes running in Linux in top Process-related information including: PID: Process IDUSER: Owner of the processPR: PriorityNI: Nice valueVIRT: Virtual memory usageRES: Resident set size (non-swapped physical memory used)SHR: Shared memoryS: Process status (S: Sleeping, R: Running, I: Idle)%CPU: Percentage of CPU usage%MEM: Percentage of memory usageTIME+: Total CPU timeCOMMAND: Command or process name 3. The `htop` Command to List Running Processes in Linux`htop` is an interactive process viewer for Linux that provides a visually appealing and feature-rich alternative to the traditional `top` command. It allows users to monitor and manage system processes in real-time with an easy-to-use interface. This guide demonstrates how to utilize `htop` to list running processes on a Linux system. Installing htopBefore using htop, ensure it is installed on your system. Use the package manager relevant to your Linux distribution: Debian/Ubuntu: sudo apt-get install htop Red Hat/CentOS: sudo dnf install htop Launching htopTo launch htop, open a terminal and type: htop list all processes running in Linux in htop Process-related information including: PID: Process IDUSER: Owner of the processPRI: PriorityNI: Nice valueVIRT: Virtual memory usageRES: Resident set size (non-swapped physical memory used)SHR: Shared memoryS: Process status (S: Sleeping, R: Running, I: Idle)CPU%: Percentage of CPU usageMEM%: Percentage of memory usageTIME+: Total CPU timeCommand: Command or process name4. The `atop` Command to List Running Processes in Linux`atop` is a powerful and interactive system and process monitor for Linux that provides detailed information about system resources and processes. This guide illustrates how to use the `atop` command to list running processes and monitor system activity. Installing htopBefore using atop, ensure it is installed on your system. Use the package manager relevant to your Linux distribution: Debian/Ubuntu: sudo apt-get install atop Red Hat/CentOS: sudo dnf install atop Launching atopTo launch atop, open a terminal and type: atop list all processes running in Linux in atop Process-related information including: PID : Process IDSYSCPU : System CPU TimeUSRCPU : User CPU TimeVGROW : Virtual Memory GrowthRGROW : Resident Memory GrowthRUID : Real User IDEUID : Effective User IDST : Process StateEXC : Exit CodeTHR : Number of ThreadsS : Process StatusCPUNR : CPU NumberCPU : Overall CPU UsageCMD : Command or Process Name5. The `pgrep` Command to Find Process IDs in LinuxThe `pgrep` command in Linux is a useful tool for finding and listing process IDs (PIDs) based on various criteria. It simplifies the process of identifying running processes, especially when searching for multiple instances or processes associated with specific attributes. Basic UsageTo find the PID of a process using pgrep, you can use the following syntax: pgrep [options] pattern For instance, use the following command to search for the `systemd` process: pgrep systemd Finding process ID Example: Using the PID 291 provides information on the systemd process: Finding process ID ConclusionIn this article we discussed how to list running processes in Linux . Processes are vital for multitasking and are identified by a unique Process ID (PID). Various commands like ps, top, htop, atop, and pgrep help list and manage these processes. For instance, ps provides a snapshot of processes, while top and htop offer real-time monitoring. atop gives detailed insights, and pgrep finds processes based on criteria. These commands empower users to understand, monitor, and control processes in a Linux system, ensuring efficient multitasking and resource management. Use the Bash shell in Linux to manage foreground and background processes. You can use Bash's job control functions and signals to give you more flexibility in how you run commands. We show you how. All About Processes Whenever a program is executed in a Linux or Unix-like operating system, a process is started. "Process" is the name for the internal representation of the executing program in the computer's memory. There is a process for every active program. In fact, there is a process for nearly everything that is running on your computer. That includes the components of your graphical desktop environment (GDE) such as GNOME or KDE, and system daemons that are launched at start-up. Why nearly everything that is running? Well, Bash built-ins such as cd, pwd, and alias do not need to have a process launched (or "spawned") when they are run. Bash executes these commands within the instance of the Bash shell that is running in your terminal window. These commands are fast precisely because they don't need to have a process launched for them to execute. (You can type help in a terminal window to see the list of Bash built-ins.) Processes can be running in the foreground, in which case they take over your terminal until they have completed, or they can be run in the background. Processes that run in the background don't dominate the terminal window and you can continue to work in it. Or at least, they don't dominate the terminal window if they don't generate screen output. We'll start a simple ping trace running. We're going to ping the How-To Geek domain. This will execute as a foreground process. ping www.howtogeek.com We get the expected results, scrolling down the terminal window. We can't do anything else in the terminal window while ping is running. To terminate the command hit Ctrl+C. Ctrl+C The visible effect of the Ctrl+C is highlighted in the screenshot. ping gives a short summary and then stops. Let's repeat that. But this time we'll hit Ctrl+Z instead of Ctrl+C. The task won't be terminated. It will become a background task. We get control of the terminal window returned to us. ping www.howtogeek.com Ctrl+Z The visible effect of hitting Ctrl+Z is highlighted in the screenshot. This time we are told the process is stopped. Stopped doesn't mean terminated. It's like a car at a stop sign. We haven't scrapped it and thrown it away. It's still on the road, stationary, waiting to go. The process is now a background job. The jobs command will list the jobs that have been started in the current terminal session. And because jobs are (inevitably) processes, we can also use the ps command to see them. Let's use both commands and compare their outputs. We'll use the T option (terminal) option to only list the processes that are running in this terminal window. Note that there is no need to use a hyphen - with the T option. jobs ps T The jobs command tells us: [1]: The number in square brackets is the job number. We can use this to refer to the job when we need to control it with job control commands. +: The plus sign + shows that this is the job that will be acted upon if we use a job control command without a specific job number. It is called the default job. The default job is always the one most recently added to the list of jobs. Stopped: The process is not running. ping www.howtogeek.com: The command line that launched the process. The ps command tells us: PID: The process ID of the process. Each process has a unique ID. TTY: The pseudo-teletype (terminal window) that the process was executed from. STAT: The status of the process. TIME: The amount of CPU time consumed by the process. COMMAND: The command that launched the process. These are common values for the STAT column: D: Uninterruptible sleep. The process is in a waiting state, usually waiting for input or output, and cannot be interrupted. I: Idle. R: Running. S: Interruptible sleep. T: Stopped by a job control signal. Z: A zombie process. The process has been terminated but hasn't been "cleaned down" by its parent process. The value in the STAT column can be followed by one of these extra indicators: