I'm not a robot

Fedora commands list

Linux, often misunderstood as a complex operating system, may not be entirely accurate. While it can seem daunting at first, getting familiar with Linux can make it challenging to return to Windows systems. The powerful nature of Linux commands in controlling the PC, coupled with its clean user interface, makes it difficult to switch back to older systems. For developers and beginners alike, a comprehensive guide on Linux/Unix command line is necessary. This guide covers basic and advanced commands, including file and directory operations, file permission commands, file compression and archiving, process management, system information, networking, and more with examples and descriptions. Moreover, this guide includes frequently used Linux shortcuts such as Bash shortcuts, Nano shortcuts, VI & Vim Shortcuts Commands. It provides a solid foundation on Linux OS commands and insights into practical applications. By the end of this guide, you will have a basic understanding of Linux/Unix Commands and how they make development easy for developers. Linux is an open-source UNIX-like operating system (OS) that sits between applications and hardware, making connections between software and physical resources that do the work. Basic Linux commands are crucial in working with the Linux operating system. This guide covers the most important Linux commands, from basics to advanced level. Tips on how to practice and learn Linux commands are also provided. It is useful for both beginners and experienced professionals. -r directory destination copies the directory “directory” and its contents to the specified destination. cp file.txt destination copies the file “file.txt” to the specified destination. mv file.txt new_name.txt renames the file “file.txt” to “new_name.txt”. mv file.txt directory moves the file “file.txt” to the specified directory. touch file.txt creates an empty file named “file.txt”. cat file.txt displays the contents of the file “file.txt”. head file.txt shows the first 10 lines of the file “file.txt”. head -n 5 file.txt displays the first 5 lines of the file “file.txt”. tail file.txt shows the last 10 lines of the file “file.txt”. tail -n 5 file.txt displays the last 5 lines of the file “file.txt”. ln -s source_file link_name creates a symbolic link named “link_name” pointing to “source_file”. find /path/to/search -name “*.txt” searches for all files with the extension “.txt” in the specified directory. File permissions on Linux and Unix systems control access to files and directories, granting read, write, and execute permissions to three categories of users: owner, group members, and everyone else. chmod grants read, write, and execute permissions to the owner of a file, chmod u+rwx file.txt for example. chown changes the owner of a file, chown user file.txt for example. chgrp changes the group ownership of a file, chgrp group file.txt for example. umask sets default file permissions, umask 022 for example. tar creates or extracts archive files, tar -czvf archive.tar.gz files/ for example. gzip compresses files, gzip file.txt for example. zip creates compressed zip archives, zip archive.zip file1.txt file2.txt for example. process management commands allow you to monitor and control running processes on the system. Commonly used process management commands include: #### ps - Displays running processes. Examples: - `ps aux`: Shows all running processes with detailed information. - `top`: Monitors system processes in real-time. #### kill - Terminates a process. Examples: - `-9 PID`: Forcefully kills a process with the specified process ID. - `pkill process_name`: Terminates processes based on their name. #### pgrep - Lists processes based on their name. Example: `pgrep process_name` lists all processes with the specified name. #### grep - Searches for specific patterns or regular expressions in text files or streams and displays matching lines. Examples: - `-i`: Ignores case distinctions while searching. - `-v`: Inverts the match, displaying non-matching lines. Example: `grep -i "hello" file.txt` searches for "hello" in file.txt, ignoring case. #### System Information Commands - Gather system information using various commands. Examples: - `sud uname -a`: Prints all system information. - `df -h`: Displays disk space usage in a human-readable format. - `du -sh directory/`: Provides the total size of the specified directory. #### Networking Commands - Manage and troubleshoot network connections using various commands. Examples: - `ifconfig`: Displays network interface information. - `ping google.com`: Sends ICMP echo requests to "google.com" to check connectivity. In Linux, various commands are available for managing network connections, file transfers, and user management. Some common commands include: * `socket information.ss -tuln` to show all listening TCP and UDP connections * `ssh` to securely connect to a remote server * `scp` to securely copy files between hosts * `wget` to download files from the web * `curl` to transfer data to or from a server Additionally, Linux offers several input/output (IO) redirection commands for managing processes and commands: * Redirecting standard input (`cmd < file`) * Redirecting output (`cmd > file`) * Redirecting error output (`cmd 2> file` or `cmd 2>&1`) * Discarding output (`cmd > /dev/null`) * Appending output (`cmd >> file`) Environment variables can be set using the `export` command, and their values can be displayed using `echo` or accessed using `env`: * Setting environment variables: `export VARIABLE_NAME=value` * Displaying environment variable values: `echo $VARIABLE_NAME` User management commands allow administrators to create, modify, and manage user accounts on the system: * Showing currently logged-in users: `who` * Creating new user accounts: `sudo adduser username` * Displaying user information: `finger` * Removing user accounts: `sudo deluser USER GROUPNAME` or `sudo userdel -r username` * Locking passwords: `sudo passwd -l username` * Switching to another user account: `su - username` These commands provide a comprehensive set of tools for managing Linux systems, ensuring efficient and secure administration. Adding a user to a group without removing them from their current groups, there are numerous shortcuts commands in Linux that can boost productivity. The most common ones include Bash, Nano, and VI/Vim shortcuts commands. **Bash Shortcuts Commands** Navigation: * Ctrl + A: Move to the beginning of the line * Ctrl + E: Move to the end of the line * Alt + B/Ctrl + B: Move back one word/character * Alt + F/Ctrl + F: Move forward one word/character * Ctrl + P/Ctrl + N: Go to the previous/next command in history Editing: * Ctrl + U: Cut/delete from the cursor position to the beginning of the line * Ctrl + K: Cut/delete from the cursor position to the end of the line * Ctrl + W: Cut/delete the word before the cursor * Ctrl + Y: Paste the last cut text * Alt + B/Ctrl + F: Move back/forward one word **Nano Shortcuts Commands** File Operations: * Ctrl + O: Save the file * Ctrl + X: Exit Nano (prompt to save if modified) * Ctrl + V: Scroll down one page * Ctrl + Y: Uncut/restore the last cut text Navigation: * Alt + W: Search and replace a string in the text * Alt + R: Repeat the last search * Alt + \: Go to a specific line number * Ctrl + 6: Mark a block of text for copying or cutting * Ctrl + J: Justify the current paragraph **VI/Vim Shortcuts Commands** Command: * cw: Change the current word. Deletes from the cursor position to the end of the current word and switches to insert mode. * dd: Delete the current line. * x: Delete the character under the cursor. * r: Replace the character under the cursor with a new character entered from the keyboard. * i: Switch to insert mode before the cursor. * ESC: Exit from insert or command-line mode and return to command mode. Note: These shortcuts are subject to change based on the specific Linux distribution being used. Here's the rewritten text: Enter insert mode at your cursor position.:wSave the file.vJump into visual mode to select text.xDelete the character under your cursor.:qQuit Vim without saving changes.yCopy the selected text.ddDelete the current line.:q!Quit and discard any unsaved changes.dDelete the highlighted text.yyCopy the current line.:wq or [Tex]:x Save, then quit Vim.pPaste the copied or deleted text below the current line.:s/old/new/gReplace all occurrences of “old” with “new”. uUndo the last change.:set nu Display line numbers. Ctrl + RRedo your previous undo. In conclusion, Linux is a widely used operating system for development, and as a developer, you should be familiar with basic commands. This Cheat Sheet covers essential commands like creating directories, file compression, process management, system information, networking, and more. Organized and categorized, this resource helps developers quickly find the right command for specific tasks, enhancing productivity and efficiency. By using this cheat sheet, developers can successfully complete projects. Don’t miss our other Python data science cheat sheet for Scikit-Learn, Bokeh, Pandas, and Python basics. There’s a command to list all available commands, but it might be unfamiliar. To access the command line, use Ctrl+Alt+T (or navigate through your menu system). Once you’re there, type compgen -c | more to view every available command. Use the space bar to scroll down the lengthy output. Note that this utility has a broad definition of what constitutes a command, including punctuation symbols used in longer commands. Modern terminal emulators allow scrolling back up to review previous output. When you find the desired command, type q once to return to your prompt. Using Bash Shell Builtin Commands for System Management Given article text here Open terminal program by selecting Applications→Accessories→Terminal (System→Terminal in KDE) or using Konsole under KDE. If frequently used, right-click on Terminal option and add launcher to panel for easier access. Automatic command line appears when logged in through SSH login screen. Standard shell prompt shows user name, computer being used, and current working directory (~). To enter commands, type them and press Enter to execute. Editing keys are available for line positioning, character deletion, and text insertion. Table 4-1 summarizes useful editing keys. Scroll through command history using up and down arrow keys or search for previous commands with Ctrl-R. Superuser account, root, provides privileged access but should be used carefully on single-user computers as Fedora assumes user expertise. Using Root Privilege Safely in Linux When using the root account, incorrect commands can cause more damage than similar commands executed in a normal account. Directly logging in as a root user is safer when only necessary, as using the su (switch user) command to gain temporary access is recommended. Always avoid entering commands as root unnecessarily. The shell prompt will change to end in a pound sign (#) instead of a dollar sign ($) when you are in root mode. When entering Linux commands, it's essential to be aware that many commands won't output an error message if everything goes well. For example, trying to remove a non-existent file with the rm command won't display any errors. The ls command allows users to combine options with positional arguments, which can be placed before or after other arguments. For example, the -w option (width of output) can be combined with the -a and -l options in any order, as long as the number immediately follows the -w option. Virtual Terminals (VTs) play a crucial role in Fedora's configuration, allowing users to log in using a character-mode display even when the graphical user interface is running. There are 12 virtual terminals configured for character-mode login, with VT7 used for graphical login. To switch between virtual terminals, users can press Ctrl-Alt and the function key corresponding to the desired terminal. The type, which, and whereis commands provide information about program locations. The type command displays detailed location information, including whether a command is hashed (stored in the shell for quick reference). Whereis shows multiple locations for a command, as well as its manpage documentation. There are several shells available on your system, including csh (tcsh), bash (sh), ksh, and zsh. You can quickly switch between them by typing the shell's name followed by a dollar sign: $csh. To return to your original shell, press Ctrl-D or type "exit". If you want to permanently change your default shell, use the chsh command: $chsh. Enter your password, then select the new shell from the list of options provided. You can view the list of available shells by using the -l option with the chsh command.