TECMINT
#1 LINUX BLOG

Q ☰ Menu

5 Most Frequently Used Open Source Shells for Linux

Aaron Kili | Last Updated: September 2, 2023 | Read Time: 4 mins
| Top Tools | 6 Comments

The **shell** is the command interpretor in an operating system such as Unix or GNU/Linux, it is a program that executes other programs. It provides a computer user an interface to the Unix/GNU Linux system so that the user can [run different commands](#) or [utilities/tools](#) with some input data.

When the shell has finished executing a program, it sends an output to the user on the screen, which is the standard output device. For this reason, it is referred to as the "**command interpretor**".

The shell is much more than just a command interpretor, it is also a programming language of its own with complete programming language constructs such as conditional execution, [loops](#), variables, functions, and many more.

That is why the Unix/GNU Linux shell is more powerful compared to the Windows shell.

In this article, we shall take a look at some of the top most used open-source shells on Unix/GNU Linux.

1. Bash Shell

The Bash Shell, or simply "Bash", stands for "Bourne Again SHell". It's an enhancement of the original Bourne Shell (sh) and was introduced in 1989 by Brian Fox.

Over the years, Bash has become one of the most popular and widely used command-line interpreters on [many Linux distributions](#) and macOS (until Catalina, after which it was replaced by zsh).

Bash provides users with the ability to run commands, navigate the file system, and run scripts. Users can harness features like [command history](#), tab completion, and scripting capabilities to automate tasks and streamline workflows.

The Bash Shell has played a pivotal role in the evolution of open-source computing, offering a consistent interface for users to interact with their systems.

```
BASH(1)                  General Commands Manual                  BASH(1)

NAME
       bash - GNU Bourne-Again SHell

SYNOPSIS
       bash [options] [command_string | file]

COPYRIGHT
       Bash is Copyright (C) 1989-2013 by the Free Software Foundation, Inc.

DESCRIPTION
       Bash  is  an  sh-compatible  command language interpreter that executes
       commands read from the standard input or from a file.  Bash also incor-
       porates useful features from the Korn and C shells (ksh and csh).

       Bash  is  intended  to  be a conformant implementation of the Shell and
       Utilities portion  of  the  IEEE  POSIX  specification  (IEEE  Standard
       1003.1).  Bash can be configured to be POSIX-conformant by default.

OPTIONS
       All  of  the  single-character shell options documented in the descrip-
       tion of the set builtin command can be used as options when  the  shell
Manual page bash(1) line 1 (press h for help or q to quit)
```

Bash Shell

2. Tcsh/Csh Shell

The Tcsh (pronounced "**tee-shell**") and Csh (pronounced "**see-shell**") are Unix-based command-line shells that have been around for decades. Csh, or the C shell, was developed by Bill Joy in the late 1970s at the University of California, Berkeley.

It introduced several features and scripting syntax that were inspired by the C programming language, which made it popular among programmers familiar with C.

Tcsh is an enhanced version of the original Csh, which stands for "Tenex C Shell", with "Tenex" being a reference to the TENEX operating system.

Tcsh incorporates all the features of Csh along with additional improvements, such as command-line editing, job control, and command completion.

Linux Commands and Tools

Pssh – Execute Commands on Multiple Remote Linux Servers

Autojump – Quickly Navigate Directories and Linux File System

How to Transfer Files Between Two Computers using nc and pv Commands

11 Lesser Known Useful Linux Commands

How to Delete User Accounts and Their Home Directories in Linux

Both shells have been influential in the Unix world, with many users appreciating their unique scripting syntax and features. However, over time, other shells like bash have gained more popularity due to their extensive features and widespread adoption.

Nonetheless, Tcsh and Csh remain an integral part of Unix's shell history.

```
TCSH(1)                    General Commands Manual                    TCSH(1)

NAME
       tcsh - C shell with file name completion and command line editing

SYNOPSIS
       tcsh [-bcdefFimnqstvVxX] [-Dname[=value]] [arg ...]
       tcsh -l

DESCRIPTION
       tcsh  is  an enhanced but completely compatible version of the Berkeley
       UNIX C shell, csh(1).  It is a command language interpreter usable both
       as an interactive login shell and a shell script command processor.  It
       includes a command-line editor (see The command-line editor),  program-
       mable word completion (see Completion and listing), spelling correction
       (see Spelling correction), a history mechanism (see  History  substitu-
       tion),  job  control  (see Jobs) and a C-like syntax.  The NEW FEATURES
       section describes major enhancements of tcsh over  csh(1).   Throughout
       this  manual, features of tcsh not found in most csh(1) implementations
       (specifically, the 4.4BSD csh) are labeled  with  `(+)',  and  features
       which are present in csh(1) but not usually documented are labeled with
       `(u)'.

 Manual page tcsh(1) line 1 (press h for help or q to quit)
```

Tcsh Shell

3. Ksh Shell

The KornShell, commonly known as ksh, is a Unix shell that was developed by David G. Korn at AT&T Bell Laboratories in the early 1980s.

It combines features from both the C shell (csh) and the Bourne shell (sh), aiming to provide the best of both worlds. Over the years, ksh has been praised for its

advanced scripting capabilities, making it a favorite among many system administrators and power users.

One of the standout features of ksh is its associative arrays, a feature that wasn't common in shells of its time. It also introduced the select construct, which makes menu generation in scripts straightforward.

Another significant advantage is its command-line editing and history capabilities, which were later adopted by other shells like bash.

```
KSH(1)                     General Commands Manual                     KSH(1)

NAME
       ksh, ksh93 - KornShell, a command and programming language

SYNOPSIS
       ksh  [  ±abcefhikmnoprstuvxBCDP ] [ -R file ] [ ±o option ] ... [ - ] [
       arg ... ]
       rksh [ ±abcefhikmnoprstuvxBCD ] [ -R file ] [ ±o option ] ...  [ - ]  [
       arg ... ]

DESCRIPTION
       Ksh  is  a command and programming language that executes commands read
       from a terminal or a file.  Rksh is a restricted version of the command
       interpreter  ksh;  it is used to set up login names and execution envi-
       ronments whose capabilities are more controlled than those of the stan-
       dard  shell.   Rpfksh  is a profile shell version of the command inter-
       preter ksh; it is used to to execute commands with the attributes spec-
       ified by the user's profiles (see pfexec(1)).  See Invocation below for
       the meaning of arguments to the shell.

   Definitions.
       A metacharacter is one of the following characters:
Manual page ksh(1) line 1 (press h for help or q to quit)
```
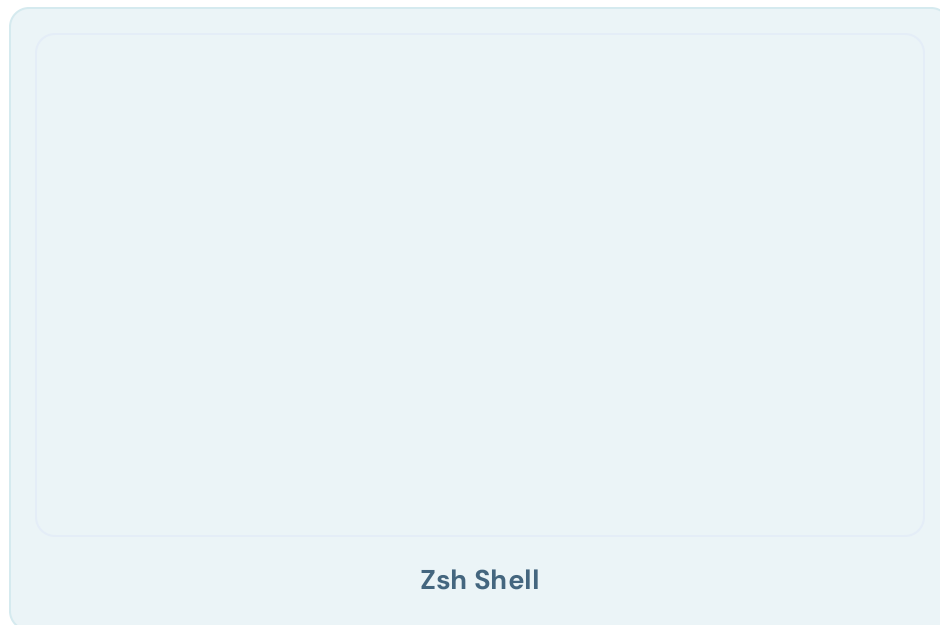
Ksh Shell

4. Zsh Shell

The Zsh shell, often referred to simply as "Zsh" is a powerful Unix shell that has gained remarkable popularity among programmers and system administrators.

Born as an extension to the Bourne Shell, Zsh incorporates features from other shells like Bash, ksh, and tcsh, offering a rich set of features and improved interactive use.

One of Zsh's standout features is its robust auto-completion capabilities. Unlike other shells, Zsh's auto-completion is context-aware, suggesting relevant commands, file paths, and even Git branches, making the user experience smoother and more intuitive.

Additionally, Zsh is highly customizable using the Oh My Zsh framework, users can easily apply themes and plugins, enhancing both the aesthetic and functional aspects of their terminal environment.

Zsh Shell

5. Fish

Fish Shell is a modern, user-friendly command-line shell that aims to provide an interactive and intuitive interface

for users.

Unlike traditional shells like Bash or Zsh, Fish offers smart suggestions out of the box, making it easier for both beginners and experienced users to navigate and execute commands.

One of the standout features of [Fish](#) is its auto-suggestions, which provide real-time feedback based on your command history, allowing for quicker command execution.

Fish also boasts a vibrant and colorful interface, with syntax highlighting that makes it easier to read and understand commands. Its web-based configuration allows users to customize their shell experience without delving into complex configuration files.

Additionally, Fish has an extensive and growing collection of community-contributed plugins and themes, enhancing its functionality and appearance.
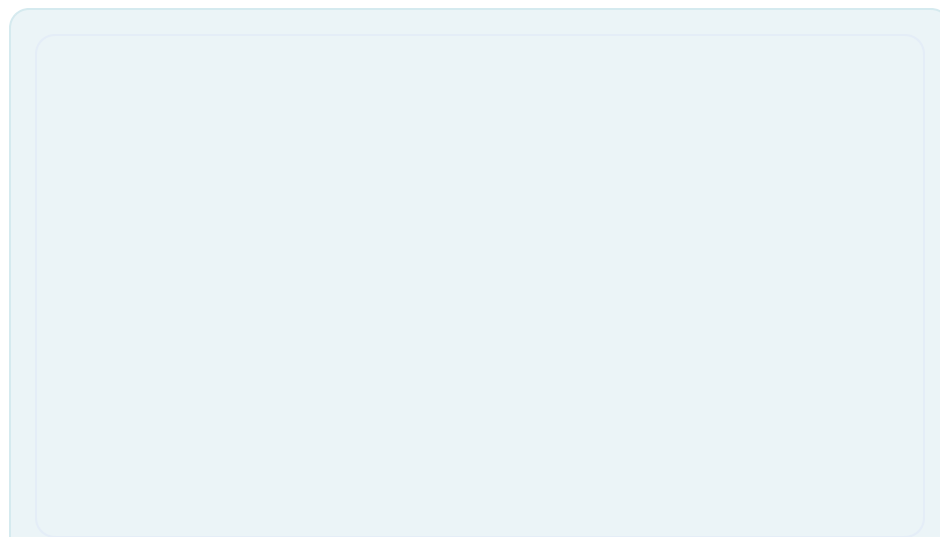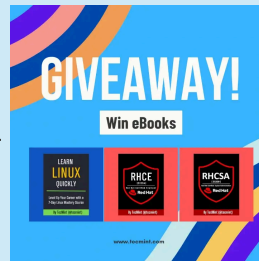
Fish Shell

Summary

These are not the only shells available in Unix/GNU Linux, but they are among the most commonly used, aside from those that come pre-installed with various Linux distributions.

I hope you find this article useful. For any additional information, please don't hesitate to post a comment.

Aaron Kili

Aaron Kili is a Linux and F.O.S.S enthusiast, an upcoming Linux SysAdmin, web developer, and currently a content creator for TecMint who loves working with computers and strongly believes in sharing knowledge.

Each tutorial at TecMint is created by a team of experienced Linux system administrators so that it meets our high-quality *standards.*

Join the [TecMint Weekly Newsletter](#) (More Than 156,129 Linux Enthusiasts Have Subscribed)

Was this article helpful? Please [add a comment](#) or [buy me a coffee](#) to show your appreciation.

Related Posts

11 Best Self-Hosted Alternatives to Google Photos

41 Must-Have Free Open Source Applications for 2024

16 Open Source Cloud Storage Software for Linux in 2024

5 Best PDF to Word Converters for Linux

4 Best Tools for Creating Fillable PDF Forms on Linux

11 Best PDF Editors to Edit PDF Documents in Linux

💬 6 Comments

Leave a Reply

Joe

August 14, 2021 at 2:41 am

Two shells everyone will sooner or later come across and are therefore worth knowing about are dash and busybox.

dash is what many Linux distros use as their default /bin/sh-shell, because it is very small and so executes small and simple shell scripts quickly. Being able to write dash-conform shell scripts means these will work for many distributions and even most UNIX-derivatives and this can be considered a core skill for everyone who works with Linux and UNIX systems.

busybox is less of a shell, but an entire collection of commands included in a shell-like environment. It is found on many embedded systems as a replacement for a full shell and other executables because it packs many of these into a single executable to save space. Although it makes for a very sparse shell environment does know busybox mean one can find their way quickly and easily around on even the smallest of computer systems and it is worth knowing about.

Reply

Dell printer Support Number

July 9, 2018 at 9:22 pm

Linux is the most powerful operating system written in C language by Linus Torvalds. The architecture of Linux consists of several layers such as Hardware in the core then kernel, shell, and the outermost layer is user application that is the operating system.

Reply

Aaron Kili

July 10, 2018 at 12:37 pm

@Dell printer Support Number

Many thanks for sharing this useful information.

Reply

Nikola Yanev

July 18, 2016 at 4:07 pm

you missed ssh jail shell

Reply

Aaron Kili K
July 18, 2016 at 4:40 pm

@Nikola Yanev

Thanks for your feedback, I will find out more about it and add it to the list above.

Reply

CodeObiect
May 5, 2016 at 6:56 pm

Thanks, maybe I try somone, although you could describe a little bardziejz some examples.

Reply

Got Something to Say? Join the Discussion...

Thank you for taking the time to share your thoughts with us. We appreciate your decision to leave a comment and value your contribution to the discussion. It's important to note that we moderate all comments in accordance with our [comment policy](#) to ensure a respectful and constructive conversation.

Rest assured that your email address will remain private and will not be published or shared with anyone. We prioritize the privacy and security of our users.

Name *

Email *

Save my name, email, and website in this browser for the next time I comment.

Post Comment

Linux Server Monitoring Tools

How to Create a Centralized Log Server with Rsyslog in CentOS/RHEL 7

How to Monitor Docker Containers with Zabbix Monitoring Tool

Setting Up Real-Time Monitoring with 'Ganglia' for Grids and Clusters of Linux Servers

Icinga: A Next Generation Open Source 'Linux Server Monitoring' Tool for RHEL/CentOS 7.0

SARG – Squid Analysis Report Generator and

Learn Linux Tricks & Tips

2 Ways to Create an ISO from a Bootable USB in Linux

8 Parted Commands to Manage Disk Partitions in Linux

3 Ways to Delete All Files in a Directory Except One or Few Files with Extensions

How to Force cp Command to Overwrite without Confirmation

How to Find Out List of All Open Ports in Linux

How to Boot into Single User Mode in

Best Linux Tools

11 Best GitHub Alternatives to Host Open Source Projects

10 Best Google Drive Clients for Linux in 2023

Top 6 Command Line Music Players for Linux Users

5 Best Command Line HTTP Clients for Linux

7 Best Email Clients for Linux Systems

5 Online Tools for Generating and Testing Cron Jobs for Linux

Internet Bandwidth Monitoring Tool

CentOS/RHEL 7

Trickle – Control Network Traffic Bandwidth Of Applications in a Linux