



Shell, Bash, and CLI

.NET

*A command-line interface (CLI) is an operating system **Shell** that uses alphanumeric characters typed on a keyboard to interactively provide instructions and data to the operating system.*

[HTTPS://EN.WIKIPEDIA.ORG/WIKI/SHELL_\(COMPUTING\)](https://en.wikipedia.org/wiki/Shell_(computing))

What is a Shell?

[https://en.wikipedia.org/wiki/Shell_\(computing\)](https://en.wikipedia.org/wiki/Shell_(computing))
https://en.wikipedia.org/wiki/Microsoft_Windows

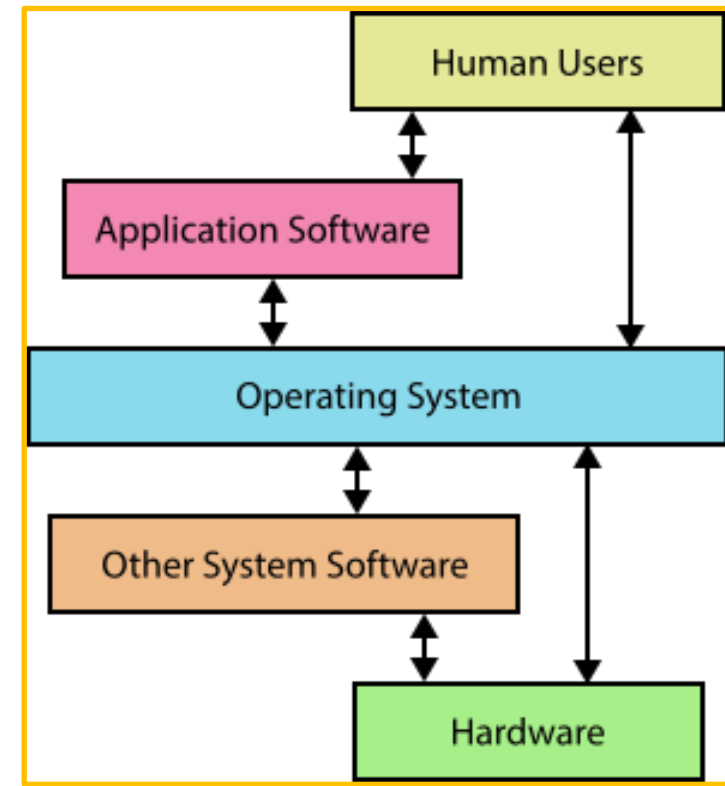
To understand a **Shell**, first, you need to understand what an Operating System (OS) is.

- An OS is the foundational program in a computer.
- It is responsible for managing the computers registers, processors, memory, file naming/management, etc.
- You can gain access to the OS and manually manage your computer using a **Shell** application.

A **Shell** application acts like the shell of a clam. A **Shell**:

1. prompts users for input,
2. interprets the input, and then
3. handles output from the underlying operating system.

Using correct keywords and commands, a **Shell** program gives you limited access to interact with the OS through the **kernel API**. The **kernel API** is also used by applications that run on the computer.



What is a Shell?

[https://en.wikipedia.org/wiki/Shell_\(computing\)](https://en.wikipedia.org/wiki/Shell_(computing))
https://en.wikipedia.org/wiki/Microsoft_Windows

There are only two different variations on a **Shell**.

- Command-Line Shell – All interaction is through text with syntax rules.
- Graphical User Interface – Interaction is done with a mouse using images representing files, applications, etc.

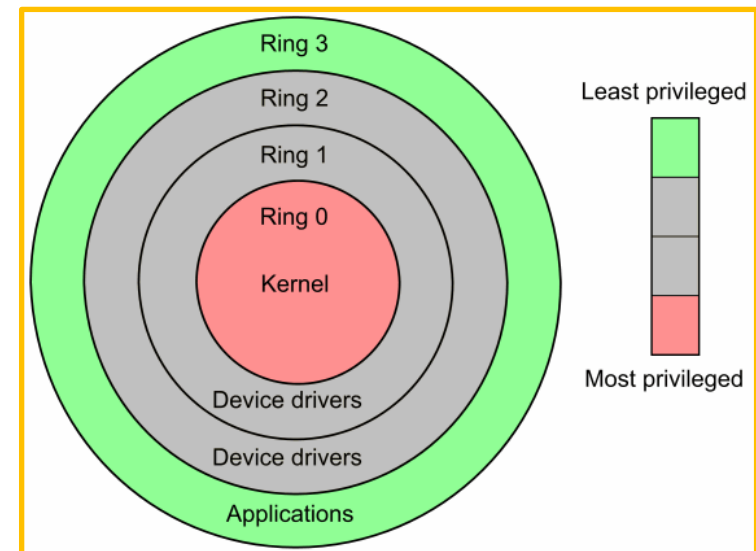
We use the **Graphical User Interface** version of our computers **Shell** when we use our mouse to drag and drop, copy and paste, or select files or applications to open.

The **Command-Line Shell** is a text only interface. The **Command-Line Interface** is referred to as a **CLI**. It is very powerful when combined with the many special keywords built into the Operating System that help us manage different aspects of our computers.

A **Shells** privileges are somewhat limited when run normally, but most have an Administrator Mode. Command Prompt is the Windows Shell. When opening it, you can select “Run as Administrator”. This will allow you to perform actions that are normally prohibited by the OP.

```
C:\> Command Prompt
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MarkC>_
```



Different CLI Shells

<https://opensource.com/resources/what-bash>

<https://snipcademy.com/linux-command-line-environment>

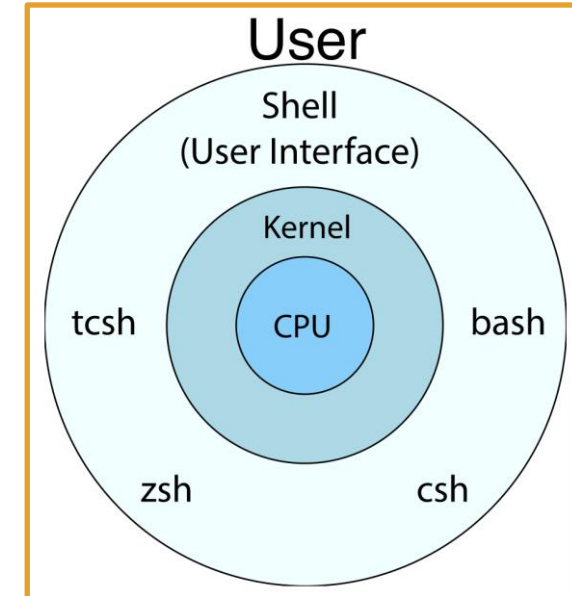
<https://devhints.io/bash>

Since a **Shell** is just another application, it can be replaced with a different, compatible, **Shell** application.

There are many different **Shell** applications. Every OS has a **Shell**. You can learn about them [here](#).

- The most popular **Shell** is **Bourne Again SHell** (bash).
- **Bash** is natively installed on Linux machines (like Macintosh).
- **Bash**, like every **Shell**, has its own keywords and syntax.

```
/usr/bin/bash --login -i
MarkC@MarkCMoore ~
$
```



Emulators

<https://en.wikipedia.org/wiki/Emulator>

<https://snipcademy.com/linux-command-line-environment>

“Emulation” is the ability of a computer program to imitate another program. An **emulator** typically enables the host system to run software designed for the guest system.

At Revature, we will use an emulator of **bash** called **GitBash** for the duration of this course.

- **GitBash** is a product of **Git**.
- **GitBash** emulates the **bash CLI**.
- **GitBash** allows you to use **Git** and **bash** commands in the same text-based interface.
- Download **GitBash** [HERE](#).

We can use **GitBash** on a Windows computer to interact with [GitHub.com](https://github.com) using **Git** commands while also interfacing with the Windows OS **kernel API** in the same **CLI** window.



```
/usr/bin/bash --login -i
TMPDIR=/tmp
NUMBER_OF_PROCESSORS=8
ProgramW6432=C:\Program Files
COMSPEC=C:\WINDOWS\system32\cmd.exe
APPDATA=C:\Users\MarkC\AppData\Roaming
SHELL=/usr/bin/bash
TERM=xterm
IntelliJ IDEA Community Edition=C:\Program Files\Jet
2019.3.1\bin;
WINDIR=C:\WINDOWS
MINGW_CHOST=x86_64-w64-mingw32
ProgramData=C:\ProgramData
SHLV=1
PLINK_PROTOCOL=ssh
ACLOCAL_PATH=/mingw64/share/aclocal:/usr/share/acloc
PROGRAMFILES=C:\Program Files
MANPATH=/mingw64/local/man:/mingw64/share/man:/usr/l
e/man
ORIGINAL_TEMP=/tmp
ORIGINAL_TMP=/tmp
ALLUSERSPROFILE=C:\ProgramData
TEMP=/tmp
DriverData=C:\windows\System32\Drivers\DriverData
MSYSTEM=MINGW64
```

Common Bash Commands

<https://devhints.io/bash>

<https://github.com/LeCoupa/awesome-cheatsheets/blob/master/languages/bash.sh>

Command	Explanation
ls	Lists the files in the current directory.
cd [~]	Change directory to the Home directory. [] means optional.
cd ..	Changes directory to one level up.
mkdir <dirname>	Creates a new directory with the given name.
touch <filename>	Creates a file with the given name. Updates the last accessed date to the current date.
pwd	Prints the path to the current directory to the CLI.
rm <name>	Deletes the named file.
rmdir <dirname>	Deletes the named directory.
up-arrow	Cycle through all previous commands in reverse order.
cd <dirname>	Move into the named directory.