

Why can htop beat Task Manager?

There is a better application than Task Manager to handle processes and it runs on command line interface.

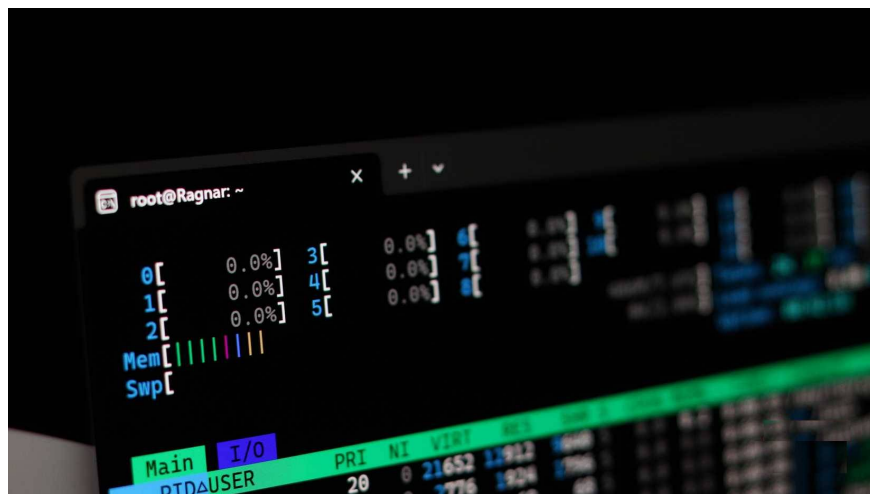
If you've ever spent any time managing open programs and processes on Windows, you're probably familiar with Task Manager . It's one of the most popular Windows utilities, and you can use it for everything from checking your system's specs and resources to closing unresponsive programs.

You can easily identify and fix high memory usage using Task Manager and system health. However, there is a better application than Task Manager for handling processes and it runs on the command line interface.

Why is htop superior to Task Manager?

More data, fewer clicks

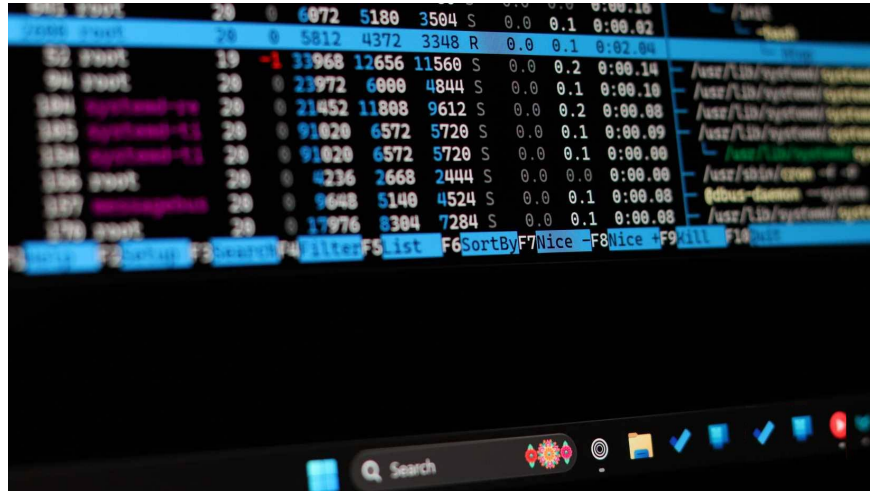
Windows' Task Manager works perfectly fine and offers enough control for most Windows users who want to monitor their computer's performance. However, htop has a few advantages that make it better for power users and anyone who wants more control over their system.



Unlike Task Manager's simple graphs, htop uses a color-coded system that tells you exactly what's going on with your system resources. Green bars show normal user processes using the CPU, blue shows low-priority threads, red shows system and kernel processes, and orange shows interrupt processing time.

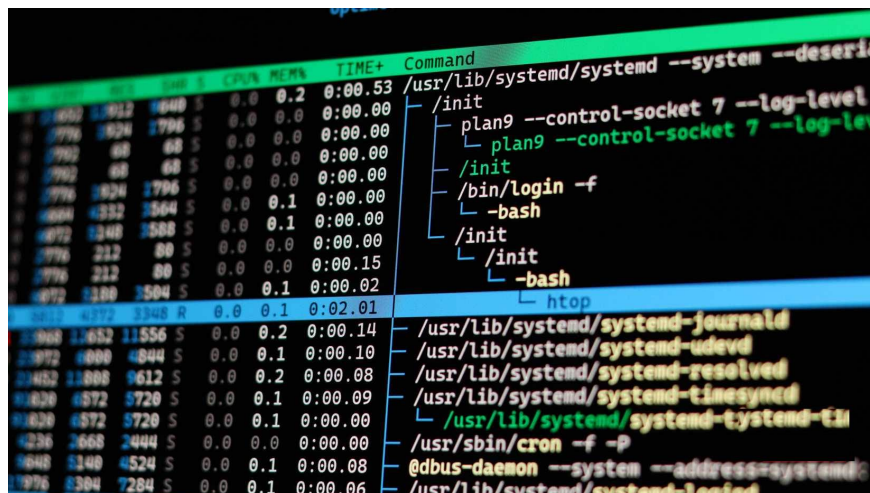
Similarly, green indicates active memory pages, blue indicates buffer pages, and orange indicates cached data. These visual cues allow you to diagnose problems immediately without having to dig through menus and wonder if the process you're about to close is system-critical.

However, color coding isn't all htop has to offer. It's incredibly flexible and easy to use for a terminal application. First, you get full mouse support even though htop is a command line tool. You can click on processes to select them, scroll through the list with the mouse wheel, and even click the function buttons at the bottom instead of pressing keyboard shortcuts.



Htop also supports both horizontal and vertical scrolling. This solves a significant problem with command line tools, where long command names or text strings get truncated. htop allows you to scroll horizontally to see the entire command line of any process. This is especially useful when you are trying to determine exactly what is running for processes with long parameters.

You also get much more advanced process management with htop. If you want to kill a misbehaving process, just select it and press **F9**. If you need to adjust the priority of a process, **F7** and **F8** let you raise or lower it instantly. Accessing this is like a secret menu that shows you everything running on your PC.



You can even view processes in a hierarchical tree structure, showing the parent-child relationship between them, using the **F5** key. This makes it easy to understand which process spawned which process—something Task Manager doesn't do elegantly.

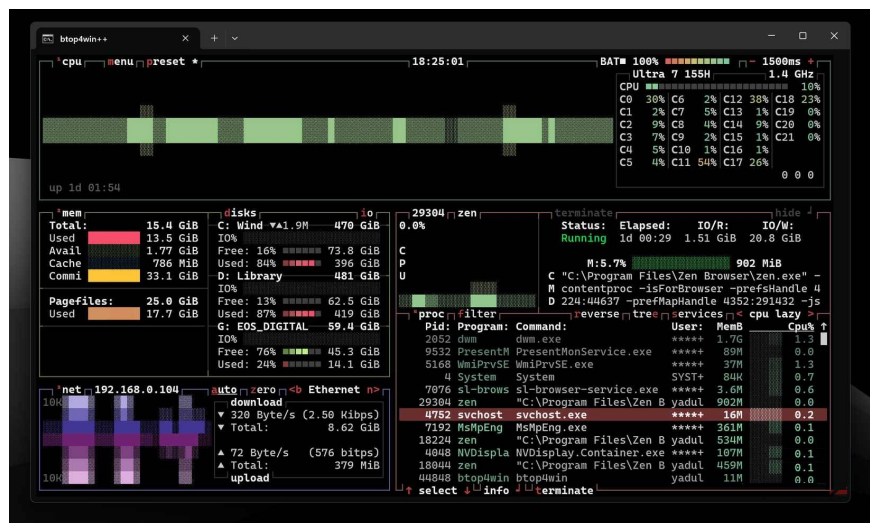
Using htop on Windows

Yes, it runs on Windows too.

Although htop was originally a Linux-exclusive tool, Windows users are no exception. Several ports and alternatives bring similar functionality to Windows. And if you want the real htop experience, you can run native htop on Windows using Windows Subsystem for Linux (WSL).

For ports, you should use btop4win. It is a free, open source, feature-rich port of btop++, which is itself a modern C++ successor to htop with even more features. It runs on Windows 10 version 1607 and above and works great in Windows Terminal. You can also install it using Winget by running the following command.

```
winget install aristocratos.btop4win
```



If you want to use native htop, make sure you have WSL installed on Windows first. For most users, this is as simple as running this command.

```
wsl --install
```

This command will install Ubuntu as the default WSL installation. Once done, you can install htop as follows

```
sudo apt install libncursesw5-dev autotools-dev autoconf automake build-essential
```

```
root@Ragnar: ~
<3>WSL (641 - Relay) ERROR: CreateProcessParseCommon:909: getpwuid(1000) failed 17
root@Ragnar:~# sudo apt install libncursesw5-dev autotools-dev autoconf automake build-essential
```

```
root@Ragnar: ~
Setting up libncurses-dev:amd64 (6.4+20240113-1ubuntu2) ...
Setting up cpp-13 (13.3.0-6ubuntu2~24.04) ...
Setting up gcc-13-x86-64-linux-gnu (13.3.0-6ubuntu2~24.04) ...
Setting up gcc-13 (13.3.0-6ubuntu2~24.04) ...
Setting up cpp (4:13.2.0-7ubuntu1) ...
Setting up g++-13-x86-64-linux-gnu (13.3.0-6ubuntu2~24.04) ...
Setting up gcc-x86-64-linux-gnu (4:13.2.0-7ubuntu1) ...
Setting up gcc (4:13.2.0-7ubuntu1) ...
Setting up g++-x86-64-linux-gnu (4:13.2.0-7ubuntu1) ...
Setting up g++-13 (13.3.0-6ubuntu2~24.04) ...
Setting up g++ (4:13.2.0-7ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.10ubuntu1) ...
Setting up libheif1:amd64 (1.17.6-1ubuntu4.1) ...
Setting up libgd3:amd64 (2.3.3-9ubuntu5) ...
Setting up libc-devtools (2.39-0ubuntu8.6) ...
Setting up libheif-plugin-aomdec:amd64 (1.17.6-1ubuntu4.1) ...
Setting up libheif-plugin-libde265:amd64 (1.17.6-1ubuntu4.1) ...
Setting up libheif-plugin-aomenc:amd64 (1.17.6-1ubuntu4.1) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for install-info (7.1-3build2) ...
root@Ragnar:~# apt install htop
```

```
root@Ragnar: ~
0[ 0.0%] 3[ 0.0%] 6[ 0.0%] 9[ 0.0%] 11[ 0.0%] 14[ 0.0%] 17[ 0.0%] 20[ 0.0%]
1[ 0.0%] 4[ 0.0%] 7[ 0.0%] 10[ 0.3%] 12[ 0.0%] 15[ 0.0%] 18[ 0.0%] 21[ 0.0%]
2[ 0.0%] 5[ 0.0%] 8[ 0.0%]
Mem| 461M/7.47G Tasks: 25, 20 thr, 0 kthr, 1 running
Swp| 0K/2.00G Load average: 0.04 0.07 0.02
Uptime: 00:05:23

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
2596 root 20 0 5772 4596 3436 R 0.3 0.1 0:00.12 htop
1 root 20 0 21652 12908 9640 S 0.0 0.2 0:00.49 /usr/lib/systemd/systemd
2 root 20 0 2776 1924 1796 S 0.0 0.0 0:00.00 /init
7 root 20 0 2792 68 68 S 0.0 0.0 0:00.00 plan9 --control-socket 7
8 root 20 0 2792 68 68 S 0.0 0.0 0:00.00 plan9 --control-socket 7
9 root 20 0 2776 1924 1796 S 0.0 0.0 0:00.00 /init
52 root 19 -1 33968 12600 11504 S 0.0 0.2 0:00.09 /usr/lib/systemd/systemd
94 root 20 0 23972 6000 4844 S 0.0 0.1 0:00.09 /usr/lib/systemd/systemd
104 systemd-re 20 0 21452 11808 9612 S 0.0 0.2 0:00.08 /usr/lib/systemd/systemd
105 systemd-ti 20 0 91020 6572 5720 S 0.0 0.1 0:00.06 /usr/lib/systemd/systemd
154 systemd-ti 20 0 91020 6572 5720 S 0.0 0.1 0:00.00 /usr/lib/systemd/systemd
156 root 20 0 4236 2668 2444 S 0.0 0.0 0:00.00 /usr/sbin/cron -/systemd
157 messagebus 20 0 9648 5140 4524 S 0.0 0.1 0:00.06 @dbus-daemon --s/systemd
170 root 20 0 17976 8304 7284 S 0.0 0.1 0:00.06 /usr/lib/systemd/systemd
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit
```

Once installed, you can run btop4win or htop by typing btop4win or htop into the Windows Terminal or the WSL Ubuntu terminal. Btop4win has a nicer interface, which may be easier to understand if you have never used a command line tool before. However, htop is equally easy to understand if you read the interface carefully.

The controls for both tools are fairly intuitive, but htop's helpful function key guide at the bottom will get you up and running faster. For btop4win, check out the official GitHub repository for any help you need. The repository also has a help menu with all the keyboard shortcuts, which you can access by pressing the **H** key while the tool is open.

You finished reading the article "**Why can htop beat Task Manager?**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
