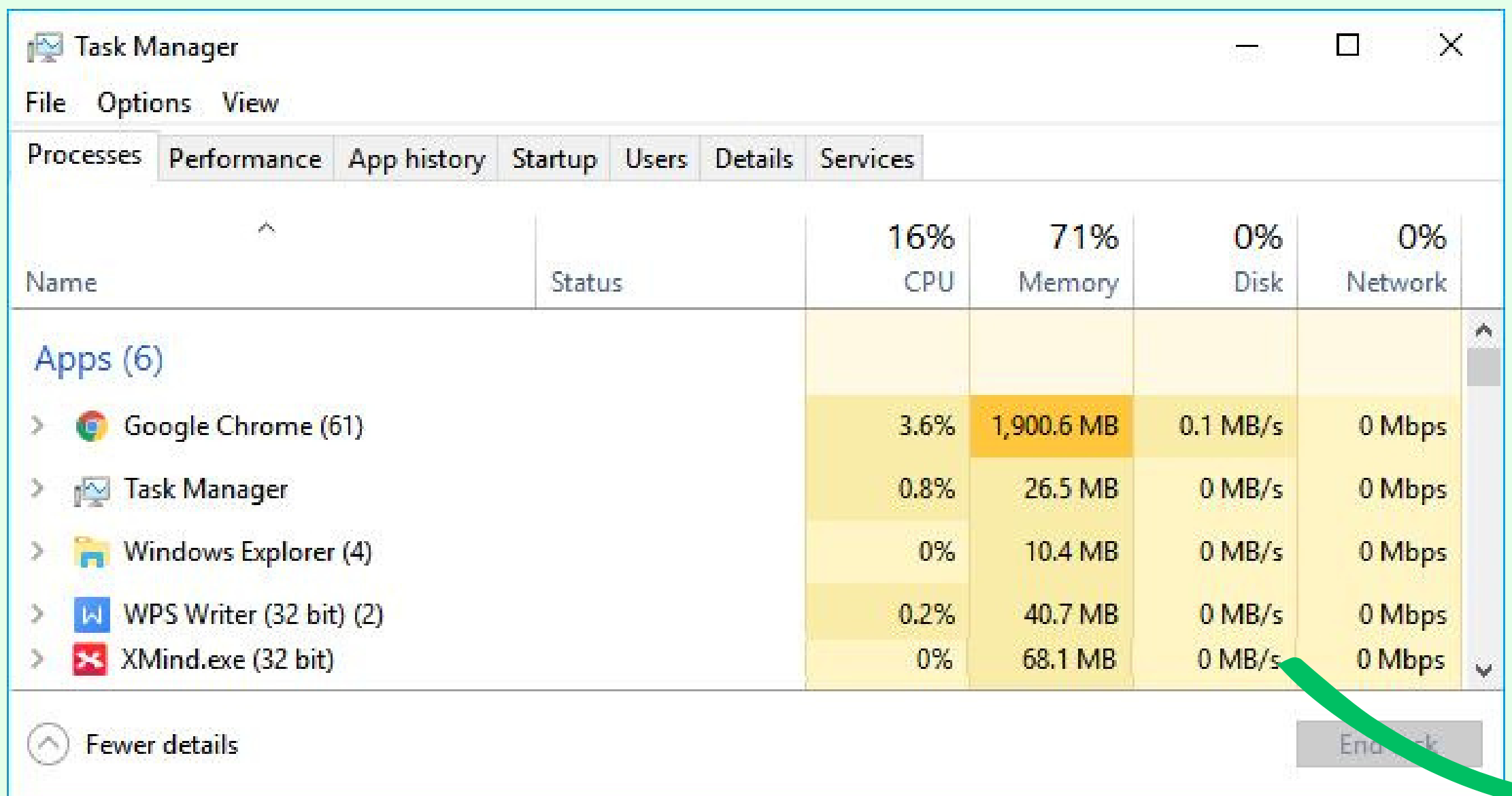




Omar Benazza

Network Automation Engineer

How to check CPU and Memory usage on Linux?

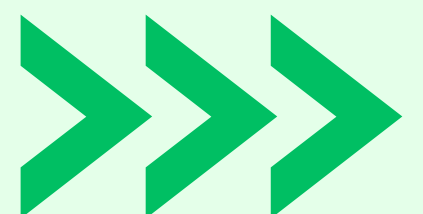


The screenshot shows the Windows Task Manager Performance tab. At the top, it displays overall system usage: 16% CPU, 71% Memory, 0% Disk, and 0% Network. Below this, a table lists running applications with their individual resource usage. A green arrow points from the 'End Task' button at the bottom right to the 'Memory' column of the table.

Name	Status	16% CPU	71% Memory	0% Disk	0% Network
Apps (6)					
> Google Chrome (61)		3.6%	1,900.6 MB	0.1 MB/s	0 Mbps
> Task Manager		0.8%	26.5 MB	0 MB/s	0 Mbps
> Windows Explorer (4)		0%	10.4 MB	0 MB/s	0 Mbps
> WPS Writer (32 bit) (2)		0.2%	40.7 MB	0 MB/s	0 Mbps
> XMind.exe (32 bit)		0%	68.1 MB	0 MB/s	0 Mbps

Fewer details

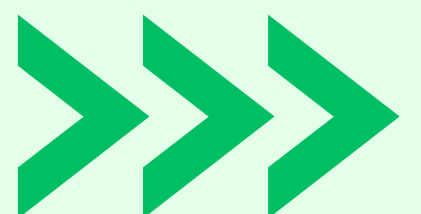
End Task



The commande: “Top”

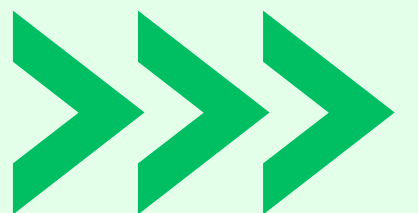
```
top - 17:49:58 up 2:24, 1 user, load average: 2.08, 3.70, 3.37
Tasks: 314 total, 2 running, 312 sleeping, 0 stopped, 0 zombie
%Cpu(s): 7.0 us, 2.3 sy, 0.0 ni, 77.5 id, 13.2 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7710.8 total, 1584.8 free, 3498.5 used, 2627.6 buff/cache
MiB Swap: 2048.0 total, 1840.2 free, 207.8 used. 2839.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10891		20	0	1125.2g	348948	142580	S	25.0	4.4	2:42.33	msedge
4471		20	0	33.1g	180828	98152	R	18.8	2.3	1:18.98	msedge
15643		20	0	1125.0g	303032	129768	S	18.8	3.8	0:45.21	msedge
2439		20	0	5654740	241412	80560	S	6.2	3.1	2:39.96	gnome-shell
1	root	20	0	168836	13232	7472	S	0.0	0.2	0:04.41	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-event+
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kth+
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kt+
14	root	20	0	0	0	0	S	0.0	0.0	0:00.11	ksoftirqd/0
15	root	20	0	0	0	0	I	0.0	0.0	0:02.55	rcu_preempt



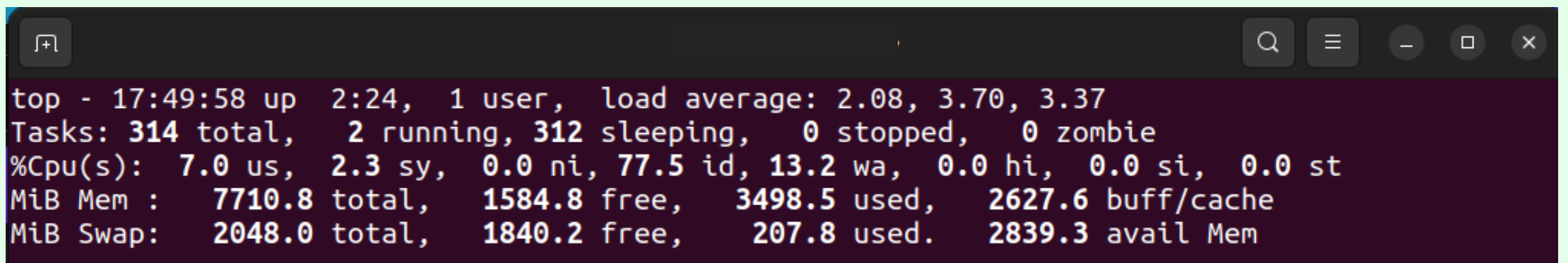
Display Information

It provides a dynamic and continuously updated display of system-related information, including CPU usage, memory usage, running processes, and more.

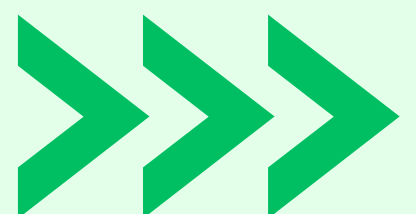


Header Information

The top of the top display contains various header lines that provide summary information about system performance. These headers typically include details such as system uptime, load averages, CPU usage statistics, and memory usage statistics.

A screenshot of a terminal window showing the output of the 'top' command. The window has a dark background with light-colored text. The output provides a summary of system performance, including uptime, load averages, task counts, and CPU/memory usage statistics.

```
top - 17:49:58 up 2:24, 1 user, load average: 2.08, 3.70, 3.37
Tasks: 314 total, 2 running, 312 sleeping, 0 stopped, 0 zombie
%Cpu(s): 7.0 us, 2.3 sy, 0.0 ni, 77.5 id, 13.2 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7710.8 total, 1584.8 free, 3498.5 used, 2627.6 buff/cache
MiB Swap: 2048.0 total, 1840.2 free, 207.8 used. 2839.3 avail Mem
```

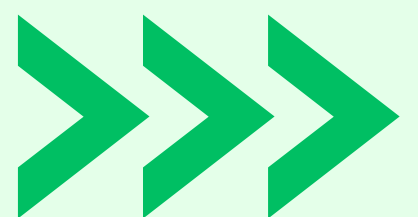


Process List

Below the header information, top displays a list of processes currently running on the system.

Each process entry includes information like process ID (PID), user, CPU usage, memory usage, and more.

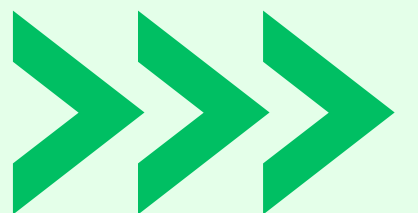
The list is sorted by default based on CPU usage, but users can change the sorting criteria interactively.



Interactive Interface

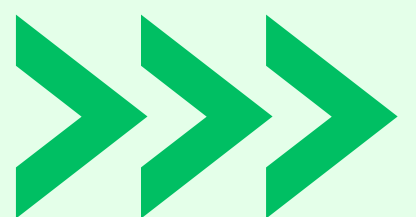
It has an interactive text-based interface that allows users to view and manipulate the displayed data.

Users can sort processes by various criteria, change the update interval, and send commands to processes directly from the top interface.



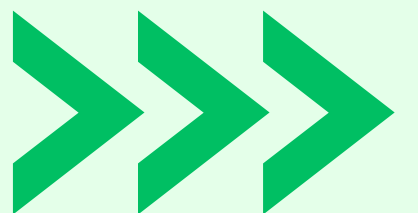
Sorting Processes

- **P**: Sort by CPU usage (default).
- **M**: Sort by memory usage.
- **N**: Sort by process name.
- **T**: Sort by total CPU time.
- **U**: Sort by user.
- **K**: Sort by process ID (PID).
- **W**: Write current configuration to the `~/.toprc` file.



Sending Commands to Processes

- **k**: kills a process by PID.
- **r**: changes a process's priority (niceness).
- **S**: changes a process's priority by PID.
- **u**: filters processes by user.
- **h**: shows a help screen with commands.

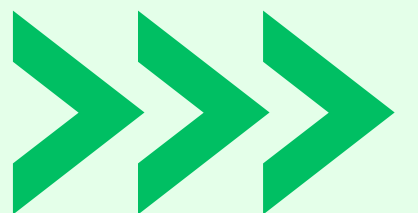


Changing the Update Interval

The default update interval for **top** is 3 seconds. You can change the update interval while **top** is running:

- Press the **d** key.
- Enter a new update interval in seconds (e.g., **5** for a 5-second update interval).
- Press Enter.

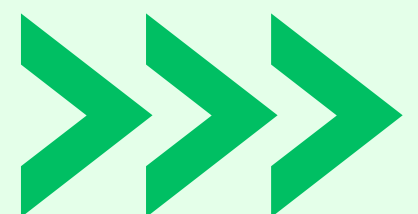
This will adjust the refresh rate to the value you entered. A longer update interval can be useful if you want to reduce the system load caused by **top**.



Configuration

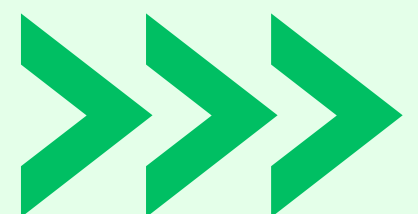
Users can customize the **top** display and behavior by modifying its configuration file, typically located at **/etc/toprc**.

This allows users to set default sorting orders, column orders, and other preferences.



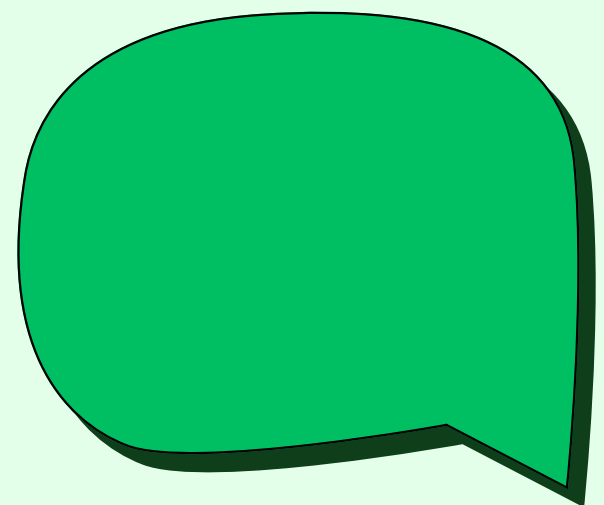
Exit

To exit top, you can press the 'q' key, which will quit the program and return you to the command prompt.



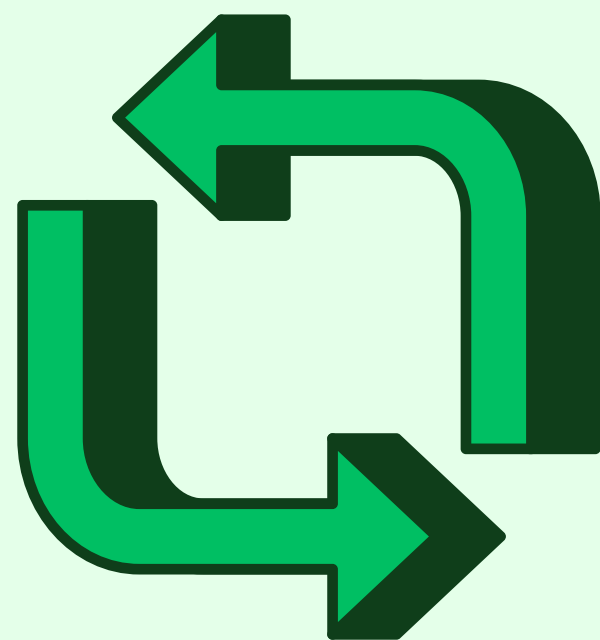
Questions?

Comment.



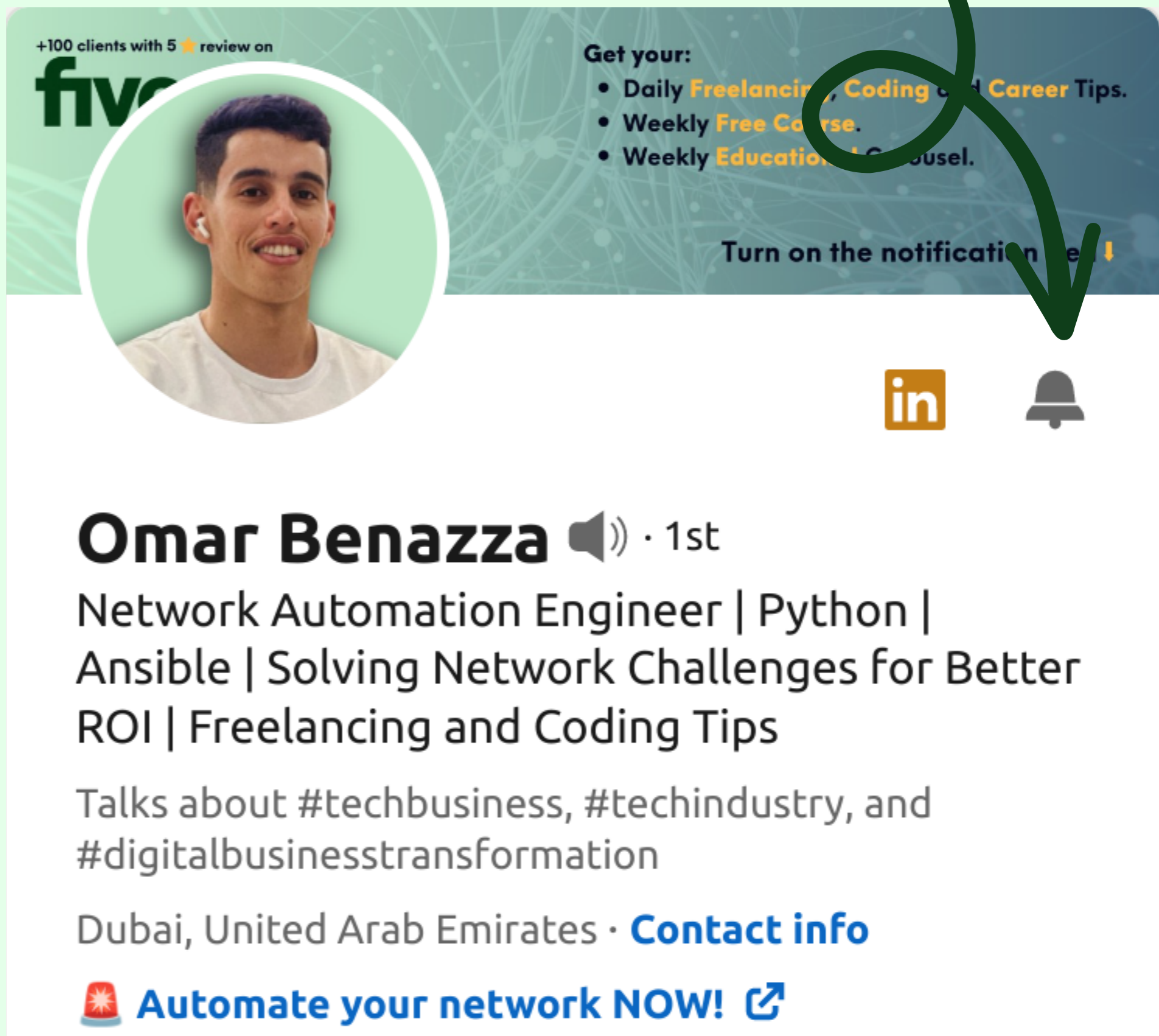
Helpful?

Repost.



Follow me for more!

Don't miss any posts – turn on notifications!



+100 clients with 5 ★ review on **five**

Get your:

- Daily **Freelancing**, **Coding** and **Career** Tips.
- Weekly **Free Course**.
- Weekly **Education** & **Counsel**.


Turn on the notification bell 🔔

Omar Benazza 🔊 · 1st

Network Automation Engineer | Python | Ansible | Solving Network Challenges for Better ROI | Freelancing and Coding Tips

Talks about #techbusiness, #techindustry, and #digitalbusinessstransformation

Dubai, United Arab Emirates · [Contact info](#)

 [Automate your network NOW!](#) 