

How to define main characteristics of computers, make declarative diagnostics, monitoring and troubleshooting with using simple mshell script and local on-premise LLMs.

mshell:

(<https://www.appservgrid.com/paw92/index.php/2025/03/30/mshell-new-linux-shell-for-ai-and-mathematics/>) + local LLMs or Linux LLM distributed evaluation framework),
ubuntu 24.04 LTS.

ollamaNexec and ollamaN are internal commands for mshell – Linux shell for AI and mathematics. Here are examples of simple ms scripts to define main characteristics of computers, make declarative diagnostics and troubleshooting of functionality.

You can install separate user on the user's system, (container, farm, k8s/ k3s nodes, etc.) with a shell like mshell and it'll be ready to go with couple lines of code. Here are examples how to create simple scripts for routine tasks and diagnostics.

Content:

- 1.** Navigation tip: accessible modes to use LLM for diagnostics and troubleshooting at environments.
 - 2.** Mshell command for information support example1a
 - 2'.** Result example1a
 - 3.** Mshell script example1b without comments for commands from LLM for declarative run (example 1b)
 - 4.** Results of script execution for example1b
 - 5.** Mshell script example1c with comments from LLM (example1c)
 - 6.** Result of script execution for example1c
 - 7.** Mshell script example2a without comments for commands from LLM for declarative run (example2a)
 - 8.** Result of script execution for example2a
 - 9.** Mshell script example2b with comments from LLM (example2b)
 - 10.** Result of script execution for example2b
 - 11.** Examples3 of executing a single command to determine the basic parameters of computer and using mshell scripts.
 - 12.** Organization of loops in scripts for the implementation the process of diagnostic automation and other routine tasks (examples4).
 - 13.** Conclusion
-

1. Navigation tips: accessible modes to use local and on-premise LLMs or (Linux LLM evaluation framework) for diagnostics and troubleshooting at environments.

- Getting information support on using OS commands, logs, writing Python code and scripts for monitoring and diagnostics (example1a).
- Declarative executing of os commands or code without any comments from LLM (example1b, example2a – monitoring and diagnostics with LLM)
- Declarative executing of os commands or code with comments from LLM (example1c, example2b – troubleshooting with LLM)
- Executing of os commands, their combinations, codes, shell scripts based on LLM (examples3)
- You can use the monitoring and diagnostic commands directly from the script or call llms to execute them. In these examples, all command executions are done by LLMs.
- The code execution during execution occurs after the message "Executing command block via mshell:" is displayed and they are highlighted in **blue color**.

2. Mshell command for information support (example1a)

```
/home/igor > ollama1 'what options has df ubuntu linux command and build mermaid diagram with explanation for options'
```

2'. Result of mshell command for information support (example1a)

Sure! The `df` command in Ubuntu Linux is used to display disk space usage on filesystems. Below are the most commonly used options along with a Mermaid diagram that explains each option.

Options of the `df` Command

1. `--human-readable`

- Displays the size in a more human-readable format (e.g., K for kilobytes, M for megabytes, G for gigabytes).

2. `--print-type`

- Prints the type of each filesystem.

3. `--inodes`

- Shows inode usage rather than disk space usage.

4. `--exclude-type=FILESYSTEM`

- Excludes filesystems of the specified type(s) from being listed.

5. `--include-type=FILESYSTEM`

- Includes only filesystems of the specified type(s) in the list.

6. `--sync`

- Forces a file system check before printing information.

7. `--total`

- Prints a line giving the total across all filesystems.

8. `--local`

- Lists only local filesystems.

9. `--all`

- Lists all filesystems (default).

10. `-T`

- Print the type of each filesystem.

11. ****P****

- Use the POSIX standard format.

Mermaid Diagram

```
```mermaid
```

```
graph LR
```

```
A[df] --> B[-h, --human-readable]
```

```
B --> B1["Displays the size in a more human-readable format (e.g., K, M, G)"]
```

```
A --> C[-T, --print-type]
```

```
C --> C1["Prints the type of each filesystem."]
```

```
A --> D[-i, --inodes]
```

```
D --> D1["Shows inode usage rather than disk space usage."]
```

```
A --> E[-x FILESYSTEM, --exclude-type=FILESYSTEM]
```

```
E --> E1["Excludes specified filesystem types from output."]
```

```
A --> F[-X FILESYSTEM, --include-type=FILESYSTEM]
```

```
F --> F1["Includes only specified filesystem types in output."]
```

```
A --> G[--sync]
```

```
G --> G1["Forces a sync before printing info."]
```

```
A --> H[--total]
```

```
H --> H1["Adds a total line summarizing all filesystems."]
```

```
A --> I[-l, --local]
```

```
I --> I1["Lists only local filesystems."]
```

```
A --> J[-a, --all]
```

```
J --> J1["Includes dummy and all filesystems."]
```

```
A --> K[-P]
```

```
K --> K1["Use POSIX output format."]
```

```
```
```

Explanation

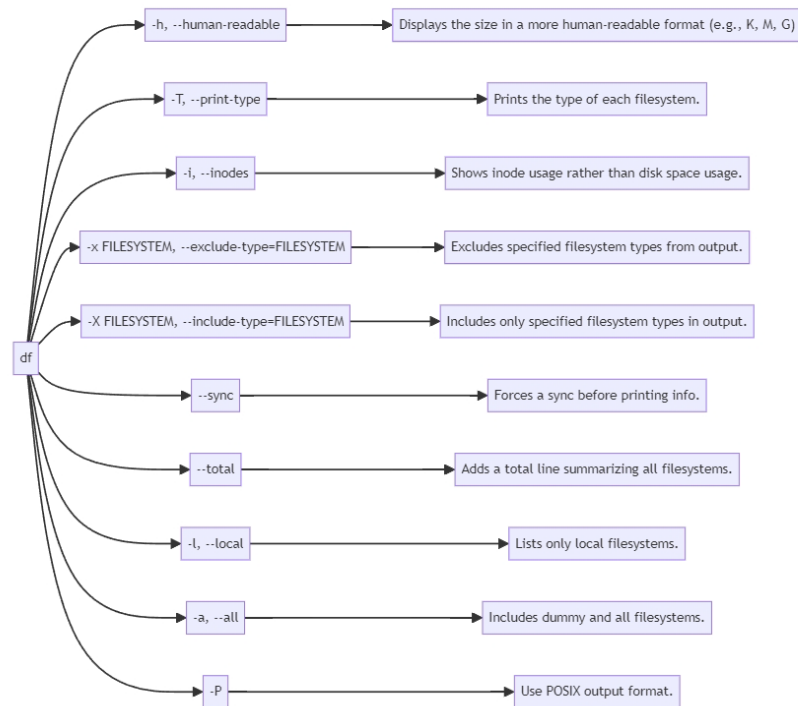
- **--h, --human-readable**: This option makes the output more readable by converting sizes like bytes to kilobytes (K), megabytes (M), gigabytes (G), etc.
- **--T, --print-type**: When this option is used, `df` will also display the type of each filesystem (e.g., ext4, NTFS).
- **--i, --inodes**: Instead of displaying disk space usage, this option shows inode usage. Inodes represent the file or directory metadata and can be useful for diagnosing issues related to file system performance.
- **--x FILESYSTEM, --exclude-type=FILESYSTEM**: This excludes filesystems with a specific type from being listed. You can specify multiple types by separating them with commas (e.g., `ext4,nfs`).
- **--X FILESYSTEM, --include-type=FILESYSTEM**: This includes only the specified filesystem types in the list.
- **--sync**: Before printing information, it forces a file system check to ensure that all data is synchronized on disk.
- **--total**: Adds a line at the bottom of the output that shows the total across all listed filesystems.
- **--l, --local**: Lists only local filesystems, excluding networked or remote filesystems.
- **--a, --all**: By default, `df` lists all mounted filesystems. Using this option forces it to include even those that are not mounted but are available on the system.
- **--T**: Similar to `-T`, this again prints the type of each filesystem.
- **--P**: This uses the POSIX standard format for displaying the output, which can be more consistent across different systems.

Example Usage

To display disk space usage in a human-readable format with types, you would use:

```
``bash
df -hT
``
```

This command will provide you with a detailed and easily understandable overview of your disk space usage along with the type of each filesystem.



3. Mshell script example1b without comments for commands from LLM for declarative run (example 1b)

```
#!/bin/mshell
```

```
# Example 1: Find the last 5 lines of messages log (instead of grep on syslog)
# echo "Example 1: Showing the last 5 lines of messages log"
# ollama1exec "No any comments. Run the Ubuntu Linux command: tail -n 5 /var/log/messages "
```

```
# Or alternatively, check a different log file
echo ""
echo "Example 1b: Showing the last 5 lines of auth.log"
ollama1exec "No any comments. Run the Ubuntu Linux command: tail -n 5 /var/log/auth.log "
```

```
# Example 2: Check system information
echo ""
echo "Example 2: Checking system information"
ollama1exec "No any comments. Run the Ubuntu Linux command: hostnamectl "
```

```
# Example 3: Check disk usage
echo ""
echo "Example 3: Checking disk usage"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: df -h "  
  
# Example 4: Show only your own processes instead of all processes  
echo ""  
echo "Example 4: Listing your own processes"  
ollama1exec "No any comments. Run the Ubuntu Linux command: ps -u $USER  
"  
  
# Example 5: Check system uptime  
echo ""  
echo "Example 5: Checking system uptime"  
ollama1exec "No any comments. Run the Ubuntu Linux command: uptime "  
  
# Example 6: Check CPU model (more specific than full lscpu output)  
echo ""  
echo "Example 6: Checking CPU model"  
ollama1exec "No any comments. Run the Ubuntu Linux command: cat  
/proc/cpuinfo "  
  
# Example 7: Check memory information  
echo ""  
echo "Example 7: Checking memory information"  
ollama1exec "No any comments. Run the Ubuntu Linux command: free -h "  
  
# Example 8: Check network interface name (more concise than full ip addr)  
echo ""  
echo "Example 8: Showing network interface names"  
ollama1exec "No any comments. Run the Ubuntu Linux command: ls  
/sys/class/net "  
  
# Example 9: Check system temperature  
echo ""  
echo "Example 9: Checking system temperature"  
ollama1exec "No any comments. Run the Ubuntu Linux command: sensors "  
  
# Example 10: Show kernel information  
echo ""  
echo "Example 10: Showing kernel information"  
ollama1exec "No any comments. Run the Ubuntu Linux command: uname -a "
```

4. Results of script execution for example1b

```
/home/igor/mshell/mscripts > ./test-nocomments.ms  
  
Loading .mshellrc...
```

Loaded environment: PATH=/usr/bin:/bin:/usr/sbin:/usr/local/bin:.\$PATH

Loaded alias: ll='ls -l'

Loaded alias: home='cd ~'

Loaded environment: EDITOR=nano

Lua state initialized successfully

Range function defined successfully

Loading history...

Example 1b: Showing the last 5 lines of auth.log

```
``bash
```

```
tail -n 5 /var/log/auth.log
```

```
````
```

**Executing command block via mshell:**

```
2025-05-11T06:14:18.140353-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/ls -lt
/var/crash
```

```
2025-05-11T06:14:18.143137-05:00 asus sudo: pam_unix(sudo:session):
session opened for user root(uid=0) by igor(uid=1001)
```

```
2025-05-11T06:14:18.151342-05:00 asus sudo: pam_unix(sudo:session):
session closed for user root
```

```
2025-05-11T06:15:01.733426-05:00 asus CRON[794525]:
```

```
pam_unix(cron:session): session opened for user root(uid=0) by
root(uid=0)
```

**2025-05-11T06:15:01.741700-05:00 asus CRON[794525]:**

**pam\_unix(cron:session): session closed for user root**

Example 2: Checking system information

```
```bash
```

```
hostnamectl
```

```
```
```

**Executing command block via mshell:**

**Static hostname: asus**

**Icon name: computer-convertible**

**Chassis: convertible**

**Machine ID: 74078ee5bb564302b4883bd2ba582acc**

**Boot ID: 33e3e0ccd06b425a834c8e18848aab39**

**Operating System: Ubuntu 24.04.2 LTS**

**Kernel: Linux 6.8.0-59-generic**

**Architecture: x86-64**

**Hardware Vendor: ASUSTeK COMPUTER INC.**

**Hardware Model: VivoBook\_ASUSLaptop TP412FA**

**Firmware Version: TP412FA.302**

**Firmware Date: Mon 2019-05-06**

**Firmware Age: 6y 5d**

Example 3: Checking disk usage

```
```bash
```

```
df -h
```

```
```
```

Executing command block via mshell:

| Filesystem     | Size | Used | Avail | Use% | Mounted on                |
|----------------|------|------|-------|------|---------------------------|
| tmpfs          | 2.0G | 197M | 1.8G  | 10%  | /run                      |
| /dev/nvme0n1p5 | 284G | 192G | 77G   | 72%  | /                         |
| tmpfs          | 9.7G | 0    | 9.7G  | 0%   | /dev/shm                  |
| tmpfs          | 5.0M | 8.0K | 5.0M  | 1%   | /run/lock                 |
| efivarfs       | 192K | 63K  | 125K  | 34%  | /sys/firmware/efi/efivars |
| /dev/nvme0n1p1 | 256M | 32M  | 225M  | 13%  | /boot/efi                 |
| tmpfs          | 2.0G | 168K | 2.0G  | 1%   | /run/user/1001            |
| /dev/sdb2      | 1.9T | 7.1G | 1.9T  | 1%   | /media/igor/Backup Plus   |
| /dev/sda2      | 4.6T | 2.3T | 2.3T  | 50%  | /media/igor/twodr         |

Example 4: Listing your own processes

```
```bash
```

```
ps -u igor
```

```
```
```

Executing command block via mshell:

| PID  | TTY | TIME     | CMD     |
|------|-----|----------|---------|
| 3309 | ?   | 00:00:02 | systemd |

|        |                          |
|--------|--------------------------|
| 3312 ? | 00:00:00 (sd-pam)        |
| 3324 ? | 00:00:03 pipewire        |
| 3325 ? | 00:00:00 pipewire        |
| 3328 ? | 00:00:02 wireplumber     |
| 3329 ? | 00:00:01 pipewire-pulse  |
| 3330 ? | 00:00:00 gnome-keyring-d |
| 3345 ? | 00:00:02 dbus-daemon     |
| 3367 ? | 00:00:00 gnome-session-b |
| 3378 ? | 00:00:00 xdg-document-po |
| 3392 ? | 00:00:00 xdg-permission- |
| 3511 ? | 00:00:00 at-spi-bus-laun |
| 3518 ? | 00:00:00 dbus-daemon     |
| 3528 ? | 00:00:00 gcr-ssh-agent   |
| 3534 ? | 00:00:00 gnome-session-c |
| 3549 ? | 00:00:00 gvfsd           |
| 3558 ? | 00:00:00 gvfsd-fuse      |
| 3562 ? | 00:00:00 gnome-session-b |
| 3595 ? | 00:11:47 gnome-shell     |
| 3636 ? | 00:00:31 mutter-x11-fram |
| 3639 ? | 00:00:07 at-spi2-registr |
| 3668 ? | 00:00:00 gnome-shell-cal |
| 3675 ? | 00:00:00 evolution-sourc |
| 3688 ? | 00:00:00 gjs             |
| 3722 ? | 00:02:38 ibus-daemon     |

|        |          |                 |
|--------|----------|-----------------|
| 3723 ? | 00:00:00 | gsd-a11y-settin |
| 3726 ? | 00:00:02 | gsd-color       |
| 3728 ? | 00:00:00 | gsd-datetime    |
| 3732 ? | 00:00:31 | gsd-housekeepin |
| 3736 ? | 00:00:00 | gsd-keyboard    |
| 3738 ? | 00:00:01 | gsd-media-keys  |
| 3741 ? | 00:00:06 | gsd-power       |
| 3742 ? | 00:00:00 | gsd-print-notif |
| 3743 ? | 00:00:00 | gsd-rfkill      |
| 3745 ? | 00:00:00 | gsd-screensaver |
| 3749 ? | 00:00:03 | gsd-sharing     |
| 3752 ? | 00:00:00 | gsd-smartcard   |
| 3753 ? | 00:00:00 | gsd-disk-utilit |
| 3756 ? | 00:00:00 | indicator-messa |
| 3758 ? | 00:00:00 | gsd-sound       |
| 3763 ? | 00:00:00 | gsd-wacom       |
| 3774 ? | 00:00:09 | gsd-xsettings   |
| 3786 ? | 00:01:03 | gvfs-udisks2-vo |
| 3789 ? | 00:00:01 | evolution-alarm |
| 3850 ? | 00:00:00 | goa-daemon      |
| 3895 ? | 00:00:00 | ibus-dconf      |
| 3901 ? | 00:00:22 | ibus-extension- |
| 3907 ? | 00:00:00 | ibus-x11        |
| 3913 ? | 00:00:00 | ibus-portal     |

|            |          |                 |
|------------|----------|-----------------|
| 3916 ?     | 00:00:00 | evolution-calen |
| 3934 ?     | 00:00:00 | goa-identity-se |
| 3943 ?     | 00:00:00 | gsd-printer     |
| 3978 ?     | 00:00:00 | evolution-addre |
| 4012 ?     | 00:00:00 | gvfs-goa-volume |
| 4017 ?     | 00:00:00 | gvfs-mtp-volume |
| 4022 ?     | 00:00:17 | gvfs-afc-volume |
| 4028 ?     | 00:00:00 | gvfs-gphoto2-vo |
| 4039 ?     | 00:00:00 | gjs             |
| 4062 ?     | 00:00:54 | ibus-engine-sim |
| 4071 ?     | 00:00:00 | snapd-desktop-i |
| 4073 ?     | 00:00:02 | tracker-miner-f |
| 4140 ?     | 00:00:07 | snapd-desktop-i |
| 4200 ?     | 00:00:00 | gvfsd-trash     |
| 4238 ?     | 00:00:09 | xdg-desktop-por |
| 4243 ?     | 00:00:17 | xdg-desktop-por |
| 4265 ?     | 00:00:08 | xdg-desktop-por |
| 4278 ?     | 00:00:16 | gjs             |
| 4295 ?     | 00:00:00 | gvfsd-metadata  |
| 4378 ?     | 00:05:26 | gnome-terminal- |
| 4397 ?     | 00:00:00 | gnome-terminal  |
| 4398 ?     | 00:00:00 | gnome-terminal. |
| 4403 pts/1 | 00:00:00 | mshell          |
| 4404 pts/1 | 00:00:00 | sh              |

```
4411 ? 00:00:10 update-notifier
5151 ? 00:00:00 dconf-service
9131 ? 00:00:00 ssh-agent
765987 ? 00:00:00 gnome-terminal
765988 ? 00:00:00 gnome-terminal.
765993 pts/0 00:00:00 mshell
778859 ? 00:00:00 gvfsd-network
778870 ? 00:00:00 gvfsd-smb-brows
778881 ? 00:00:00 gvfsd-dnssd
779867 ? 00:00:00 gnome-terminal
779868 ? 00:00:00 gnome-terminal.
779873 pts/3 00:00:00 mshell
789930 ? 00:00:00 gnome-terminal
789931 ? 00:00:00 gnome-terminal.
789936 pts/2 00:00:00 mshell
789940 pts/2 00:02:04 gedit
794489 ? 00:00:00 gnome-terminal
794490 ? 00:00:00 gnome-terminal.
794495 pts/5 00:00:00 mshell
794539 pts/5 00:00:00 test-nocomments
794570 pts/5 00:00:00 sh
794571 pts/5 00:00:00 ps
```

Example 5: Checking system uptime

```
```bash
```

```
uptime
```

```
```
```

**Executing command block via mshell:**

**06:17:22 up 3 days, 7:55, 1 user, load average: 0.23, 0.27, 0.27**

Example 6: Checking CPU model

```
```bash
```

```
cat /proc/cpuinfo
```

```
```
```

**Executing command block via mshell:**

**processor : 0**

**vendor\_id : GenuineIntel**

**cpu family : 6**

**model : 142**

**model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz**

**stepping : 11**

**microcode : 0xf6**

**cpu MHz : 900.134**

**cache size : 6144 KB**

**physical id : 0**

**siblings : 8**

core id : 0  
cpu cores : 4  
apicid : 0  
initial apicid : 0  
fpu : yes  
fpu\_exception : yes  
cpuid level : 22  
wp : yes  
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge  
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe  
syscall nx pdpe1gb rdtscp lm constant\_tsc art arch\_perfmon pebs bts  
rep\_good nopl xtopology nonstop\_tsc cpuid aperfmperf pni pclmulqdq  
dtes64 monitor ds\_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid  
sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx  
f16c rdrand lahf\_lm abm 3dnowprefetch cpuid\_fault epb ssbd ibrs ibpb  
stibp tpr\_shadow flexpriority ept vpid ept\_ad fsgsbase tsc\_adjust bmi1  
avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel\_pt  
xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp\_notify  
hwp\_act\_window hwp\_epp vnmi md\_clear flush\_l1d arch\_capabilities  
vmx flags: vnmi preemption\_timer invvpid ept\_x\_only ept\_ad ept\_1gb  
flexpriority tsc\_offset vtptr mtf vapic ept vpid unrestricted\_guest ple pml  
ept\_mode\_based\_exec  
bugs : spectre\_v1 spectre\_v2 spec\_store\_bypass mds swapgs  
itlb\_multihit srbds mmio\_stale\_data retbleed gds

**bogomips** : 3600.00  
**clflush size** : 64  
**cache\_alignment** : 64  
**address sizes** : 39 bits physical, 48 bits virtual

**power management:**

**processor** : 1  
**vendor\_id** : GenuineIntel  
**cpu family** : 6  
**model** : 142  
**model name** : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz  
**stepping** : 11  
**microcode** : 0xf6  
**cpu MHz** : 900.028  
**cache size** : 6144 KB  
**physical id** : 0  
**siblings** : 8  
**core id** : 1  
**cpu cores** : 4  
**apicid** : 2  
**initial apicid** : 2  
**fpu** : yes  
**fpu\_exception** : yes  
**cpuid level** : 22

wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge  
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe  
syscall nx pdpe1gb rdtscp lm constant\_tsc art arch\_perfmon pebs bts  
rep\_good nopl xtopology nonstop\_tsc cpuid aperfmperf pni pclmulqdq  
dtes64 monitor ds\_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid  
sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx  
f16c rdrand lahf\_lm abm 3dnowprefetch cpuid\_fault epb ssbd ibrs ibpb  
stibp tpr\_shadow flexpriority ept vpid ept\_ad fsgsbase tsc\_adjust bmi1  
avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel\_pt  
xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp\_notify  
hwp\_act\_window hwp\_epp vnmi md\_clear flush\_l1d arch\_capabilities  
vmx flags: vnmi preemption\_timer invvpid ept\_x\_only ept\_ad ept\_1gb  
flexpriority tsc\_offset vtptr mtf vapic ept vpid unrestricted\_guest ple pml  
ept\_mode\_based\_exec

bugs : spectre\_v1 spectre\_v2 spec\_store\_bypass mds swapgs  
itlb\_multihit srbds mmio\_stale\_data retbleed gds

bogomips : 3600.00

clflush size : 64

cache\_alignment : 64

address sizes : 39 bits physical, 48 bits virtual

power management:

processor : 2

vendor\_id : GenuineIntel  
cpu family : 6  
model : 142  
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz  
stepping : 11  
microcode : 0xf6  
cpu MHz : 900.022  
cache size : 6144 KB  
physical id : 0  
siblings : 8  
core id : 2  
cpu cores : 4  
apicid : 4  
initial apicid : 4  
fpu : yes  
fpu\_exception : yes  
cpuid level : 22  
wp : yes  
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge  
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe  
syscall nx pdpe1gb rdtscp lm constant\_tsc art arch\_perfmon pebs bts  
rep\_good nopl xtopology nonstop\_tsc cpuid aperfmperf pni pclmulqdq  
dtes64 monitor ds\_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid  
sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx

f16c rdrand lahf\_lm abm 3dnowprefetch cpuid\_fault epb ssbd ibrs ibpb  
stibp tpr\_shadow flexpriority ept vpid ept\_ad fsgsbase tsc\_adjust bmi1  
avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel\_pt  
xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp\_notify  
hwp\_act\_window hwp\_epp vnmi md\_clear flush\_l1d arch\_capabilities  
vmx flags: vnmi preemption\_timer invvpid ept\_x\_only ept\_ad ept\_1gb  
flexpriority tsc\_offset vtptr mtf vpic ept vpid unrestricted\_guest ple pml  
ept\_mode\_based\_exec

bugs : spectre\_v1 spectre\_v2 spec\_store\_bypass mds swapgs  
itlb\_multihit srbds mmio\_stale\_data retbleed gds

bogomips : 3600.00

clflush size : 64

cache\_alignment : 64

address sizes : 39 bits physical, 48 bits virtual

power management:

processor : 3

vendor\_id : GenuineIntel

cpu family : 6

model : 142

model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz

stepping : 11

microcode : 0xf6

cpu MHz : 400.000

cache size : 6144 KB  
physical id : 0  
siblings : 8  
core id : 3  
cpu cores : 4  
apicid : 6  
initial apicid : 6  
fpu : yes  
fpu\_exception : yes  
cpuid level : 22  
wp : yes  
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge  
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe  
syscall nx pdpe1gb rdtscp lm constant\_tsc art arch\_perfmon pebs bts  
rep\_good nopl xtopology nonstop\_tsc cpuid aperfmperf pni pclmulqdq  
dtes64 monitor ds\_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid  
sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx  
f16c rdrand lahf\_lm abm 3dnowprefetch cpuid\_fault epb ssbd ibrs ibpb  
stibp tpr\_shadow flexpriority ept vpid ept\_ad fsgsbase tsc\_adjust bmi1  
avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel\_pt  
xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp\_notify  
hwp\_act\_window hwp\_epp vnmi md\_clear flush\_l1d arch\_capabilities

vmx flags: vnmi preemption\_timer invvpid ept\_x\_only ept\_ad ept\_1gb  
flexpriority tsc\_offset vtpr mtf vapic ept vpid unrestricted\_guest ple pml  
ept\_mode\_based\_exec  
bugs : spectre\_v1 spectre\_v2 spec\_store\_bypass mds swapgs  
itlb\_multihit srbds mmio\_stale\_data retbleed gds  
bogomips : 3600.00  
clflush size : 64  
cache\_alignment : 64  
address sizes : 39 bits physical, 48 bits virtual  
power management:  
  
processor : 4  
vendor\_id : GenuineIntel  
cpu family : 6  
model : 142  
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz  
stepping : 11  
microcode : 0xf6  
cpu MHz : 900.115  
cache size : 6144 KB  
physical id : 0  
siblings : 8  
core id : 0  
cpu cores : 4

apicid : 1  
initial apicid : 1  
fpu : yes  
fpu\_exception : yes  
cpuid level : 22  
wp : yes  
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge  
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe  
syscall nx pdpe1gb rdtscp lm constant\_tsc art arch\_perfmon pebs bts  
rep\_good nopl xtopology nonstop\_tsc cpuid aperfmperf pni pclmulqdq  
dtes64 monitor ds\_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid  
sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx  
f16c rdrand lahf\_lm abm 3dnowprefetch cpuid\_fault epb ssbd ibrs ibpb  
stibp tpr\_shadow flexpriority ept vpid ept\_ad fsgsbase tsc\_adjust bmi1  
avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel\_pt  
xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp\_notify  
hwp\_act\_window hwp\_epp vnmi md\_clear flush\_l1d arch\_capabilities  
vmx flags: vnmi preemption\_timer invvpid ept\_x\_only ept\_ad ept\_1gb  
flexpriority tsc\_offset vtptr mtf vapic ept vpid unrestricted\_guest ple pml  
ept\_mode\_based\_exec  
bugs : spectre\_v1 spectre\_v2 spec\_store\_bypass mds swapgs  
itlb\_multihit srbds mmio\_stale\_data retbleed gds  
bogomips : 3600.00  
clflush size : 64

cache\_alignment : 64

address sizes : 39 bits physical, 48 bits virtual

power management:

processor : 5

vendor\_id : GenuineIntel

cpu family : 6

model : 142

model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz

stepping : 11

microcode : 0xf6

cpu MHz : 900.369

cache size : 6144 KB

physical id : 0

siblings : 8

core id : 1

cpu cores : 4

apicid : 3

initial apicid : 3

fpu : yes

fpu\_exception : yes

cpuid level : 22

wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge  
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe  
syscall nx pdpe1gb rdtscp lm constant\_tsc art arch\_perfmon pebs bts  
rep\_good nopl xtopology nonstop\_tsc cpuid aperfmperf pni pclmulqdq  
dtes64 monitor ds\_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid  
sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx  
f16c rdrand lahf\_lm abm 3dnowprefetch cpuid\_fault epb ssbd ibrs ibpb  
stibp tpr\_shadow flexpriority ept vpid ept\_ad fsgsbase tsc\_adjust bmi1  
avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel\_pt  
xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp\_notify  
hwp\_act\_window hwp\_epp vnmi md\_clear flush\_l1d arch\_capabilities  
vmx flags: vnmi preemption\_timer invvpid ept\_x\_only ept\_ad ept\_1gb  
flexpriority tsc\_offset vtptr mtf vapic ept vpid unrestricted\_guest ple pml  
ept\_mode\_based\_exec

bugs : spectre\_v1 spectre\_v2 spec\_store\_bypass mds swapgs  
itlb\_multihit srbds mmio\_stale\_data retbleed gds

bogomips : 3600.00

clflush size : 64

cache\_alignment : 64

address sizes : 39 bits physical, 48 bits virtual

power management:

processor : 6

vendor\_id : GenuineIntel

cpu family : 6  
model : 142  
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz  
stepping : 11  
microcode : 0xf6  
cpu MHz : 900.010  
cache size : 6144 KB  
physical id : 0  
siblings : 8  
core id : 2  
cpu cores : 4  
apicid : 5  
initial apicid : 5  
fpu : yes  
fpu\_exception : yes  
cpuid level : 22  
wp : yes  
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge  
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe  
syscall nx pdpe1gb rdtscp lm constant\_tsc art arch\_perfmon pebs bts  
rep\_good nopl xtopology nonstop\_tsc cpuid aperfmperf pni pclmulqdq  
dtes64 monitor ds\_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid  
sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx  
f16c rdrand lahf\_lm abm 3dnowprefetch cpuid\_fault epb ssbd ibrs ibpb

stibp tpr\_shadow flexpriority ept vpid ept\_ad fsgsbase tsc\_adjust bmi1  
avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel\_pt  
xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp\_notify  
hwp\_act\_window hwp\_epp vnmi md\_clear flush\_l1d arch\_capabilities  
vmx flags: vnmi preemption\_timer invvpid ept\_x\_only ept\_ad ept\_1gb  
flexpriority tsc\_offset vtptr mtf vapic ept vpid unrestricted\_guest ple pml  
ept\_mode\_based\_exec  
bugs : spectre\_v1 spectre\_v2 spec\_store\_bypass mds swapgs  
itlb\_multihit srbds mmio\_stale\_data retbleed gds  
bogomips : 3600.00  
clflush size : 64  
cache\_alignment : 64  
address sizes : 39 bits physical, 48 bits virtual  
power management:  
processor : 7  
vendor\_id : GenuineIntel  
cpu family : 6  
model : 142  
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz  
stepping : 11  
microcode : 0xf6  
cpu MHz : 899.989  
cache size : 6144 KB

physical id : 0  
siblings : 8  
core id : 3  
cpu cores : 4  
apicid : 7  
initial apicid : 7  
fpu : yes  
fpu\_exception : yes  
cpuid level : 22  
wp : yes  
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge  
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe  
syscall nx pdpe1gb rdtscp lm constant\_tsc art arch\_perfmon pebs bts  
rep\_good nopl xtopology nonstop\_tsc cpuid aperfmperf pni pclmulqdq  
dtes64 monitor ds\_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid  
sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx  
f16c rdrand lahf\_lm abm 3dnowprefetch cpuid\_fault epb ssbd ibrs ibpb  
stibp tpr\_shadow flexpriority ept vpid ept\_ad fsgsbase tsc\_adjust bmi1  
avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel\_pt  
xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp\_notify  
hwp\_act\_window hwp\_epp vnmi md\_clear flush\_l1d arch\_capabilities  
vmx flags: vnmi preemption\_timer invvpid ept\_x\_only ept\_ad ept\_1gb  
flexpriority tsc\_offset vtptr mtf vapic ept vpid unrestricted\_guest ple pml  
ept\_mode\_based\_exec

```
bugs : spectre_v1 spectre_v2 spec_store_bypass mds swapgs
itlb_multihit srbds mmio_stale_data retbleed gds

bogomips : 3600.00

clflush size : 64

cache_alignment : 64

address sizes : 39 bits physical, 48 bits virtual

power management:
```

Example 7: Checking memory information

```
```bash
free -h
...

```

Executing command block via mshell:

| | total | used | free | shared | buff/cache | available |
|-------|-------|-------|-------|--------|------------|-----------|
| Mem: | 19Gi | 3.2Gi | 5.9Gi | 510Mi | 10Gi | 16Gi |
| Swap: | 24Gi | 512Ki | 24Gi | | | |

Example 8: Showing network interface names

```
```bash
ls /sys/class/net
...

```

**Executing command block via mshell:**

**docker0 lo wlo1**

Example 9: Checking system temperature

```
```bash
```

```
sensors
```

```
```
```

**Executing command block via mshell:**

**coretemp-isa-0000**

**Adapter: ISA adapter**

**Package id 0: +35.0°C (high = +100.0°C, crit = +100.0°C)**

**Core 0: +34.0°C (high = +100.0°C, crit = +100.0°C)**

**Core 1: +35.0°C (high = +100.0°C, crit = +100.0°C)**

**Core 2: +34.0°C (high = +100.0°C, crit = +100.0°C)**

**Core 3: +35.0°C (high = +100.0°C, crit = +100.0°C)**

**asus-isa-0000**

**Adapter: ISA adapter**

**cpu\_fan: 0 RPM**

**nvme-pci-0100**

**Adapter: PCI adapter**

**Composite: +25.9°C (low = -0.1°C, high = +76.8°C)**

(crit = +79.8°C)

**acpitz-acpi-0**

**Adapter: ACPI interface**

**temp1: +36.0°C**

**iwlwifi\_1-virtual-0**

**Adapter: Virtual device**

**temp1: +34.0°C**

**pch\_cannonlake-virtual-0**

**Adapter: Virtual device**

**temp1: +33.0°C**

**BAT0-acpi-0**

**Adapter: ACPI interface**

**in0: 11.85 V**

**curr1: 0.00 A**

Example 10: Showing kernel information

```
```bash
```

```
uname -a
```

```
```
```

Executing command block via mshell:

```
Linux asus 6.8.0-59-generic #61-Ubuntu SMP PREEMPT_DYNAMIC Fri Apr 11 23:16:11 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
```

```
/home/igor/mshell/mscripts >
```

## 5. Mshell script example1c with comments from LLM (example1c)

```
#!/bin/mshell
```

```
Example 1: Find the last 5 lines of messages log (instead of grep on syslog)
```

```
echo "Example 1: Showing the last 5 lines of messages log"
```

```
ollama2exec "No any comments. Run the Ubuntu Linux command only: tail -n 5
/var/log/messages "
```

```
Or alternatively, check a different log file
```

```
echo ""
```

```
echo "Example 1b: Showing the last 5 lines of auth.log"
```

```
ollama1 "No markdown. Write your comment what command is doing: tail -n 5
/var/log/auth.log "
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: tail -n 5
/var/log/auth.log "
```

```
Example 2: Check system information
```

```
echo ""
```

```
echo "Example 2: Checking system information"
```

ollama1 "No markdown. Write your comment what command is doing:

```
hostnamectl "
```

ollama1exec "No any comments. Run the Ubuntu Linux command: hostnamectl "

# Example 3: Check disk usage

```
echo ""
```

```
echo "Example 3: Checking disk usage"
```

ollama1 "No markdown. Write your comment what command is doing: df -h "

ollama1exec "No any comments. Run the Ubuntu Linux command: df -h "

# Example 4: Show only your own processes instead of all processes

```
echo ""
```

```
echo "Example 4: Listing your own processes"
```

ollama1 "No markdown. Write your comment what command is doing: ps -u  
\$USER "

ollama1exec "No any comments. Run the Ubuntu Linux command: ps -u \$USER  
"

# Example 5: Check system uptime

```
echo ""
```

```
echo "Example 5: Checking system uptime"
```

ollama1 "No markdown. Write your comment what command is doing: uptime "

ollama1exec "No any comments. Run the Ubuntu Linux command: uptime "

```
Example 6: Check CPU model (more specific than full lscpu output)

echo ""

echo "Example 6: Checking CPU model"

ollama1 "No markdown. Write your comment what command is doing: cat
/proc/cpuinfo "

ollama1exec "No any comments. Run the Ubuntu Linux command: cat
/proc/cpuinfo "

Example 7: Check memory information

echo ""

echo "Example 7: Checking memory information"

ollama1 "No markdown. Write your comment what command is doing: free -h "
ollama1exec "No any comments. Run the Ubuntu Linux command: free -h "

Example 8: Check network interface name (more concise than full ip addr)

echo ""

echo "Example 8: Showing network interface names"

ollama1 "No markdown. Write your comment what command is doing: ls
/sys/class/net "

ollama1exec "No any comments. Run the Ubuntu Linux command: ls
/sys/class/net "

Example 9: Check system temperature
```

```
echo ""
```

```
echo "Example 9: Checking system temperature"
```

```
ollama1 "No markdown. Write your comment what command is doing: sensors "
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: sensors "
```

```
Example 10: Show kernel information
```

```
echo ""
```

```
echo "Example 10: Showing kernel information"
```

```
ollama1 "No markdown. Write your comment what command is doing: uname -a"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: uname -a "
```

## 6. Result of script execution for example1c

```
/home/igor/mshell/mscripts > ./test.ms
```

```
Loading .mshellrc...
```

```
Loaded environment: PATH=/usr/bin:/bin:/usr/sbin:/usr/local/bin:.$PATH
```

```
Loaded alias: ll='ls -l'
```

```
Loaded alias: home='cd ~'
```

```
Loaded environment: EDITOR=nano
```

```
Lua state initialized successfully
```

```
Range function defined successfully
```

```
Loading history...
```

```
Example 1b: Showing the last 5 lines of auth.log
```

```
The command `tail -n 5 /var/log/auth.log` is used to display the last five lines of the
`/var/log/auth.log` file. This file typically contains authentication and authorization logs
```

on a Unix-like operating system, such as Linux. The `-n 5` option specifies that only the last five lines should be displayed.

```
```bash
```

```
tail -n 5 /var/log/auth.log
```

```
```
```

**Executing command block via mshell:**

```
2025-05-11T08:08:21.545017-05:00 asus sudo: pam_unix(sudo:session): session closed for user root
```

```
2025-05-11T08:08:21.556418-05:00 asus sudo: igor : TTY=pts/5 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/apt install lm-sensors
```

```
2025-05-11T08:08:21.557302-05:00 asus sudo: pam_unix(sudo:session): session opened for user root(uid=0) by igor(uid=1001)
```

```
2025-05-11T08:08:22.488900-05:00 asus sudo: pam_unix(sudo:session): session closed for user root
```

```
2025-05-11T08:08:40.789443-05:00 asus pkexec[796012]: igor: The value for the SHELL variable was not found in the /etc/shells file [USER=root] [TTY=unknown] [CWD=/home/igor] [COMMAND=/usr/lib/update-notifier/package-system-locked]
```

**Example 2: Checking system information**

The `hostnamectl` command is used to display or change the system's host name and related settings. When you run `hostnamectl`, it provides information about the current host name, including whether it's a static, transient, or pretty (user-friendly) name, as well as other related details such as the operating system version and kernel release.

If you run `hostnamectl` without any arguments, it will display the current settings. If you provide arguments to change these settings, `hostnamectl` will update them accordingly.

Here are a few examples of what you can do with `hostnamectl`:

1. **Display Current Hostname:**

```
``sh
hostnamectl
``
```

2. **Set a New Static Hostname:**

```
``sh
sudo hostnamectl set-hostname my-new-hostname
``
```

3. **Set a New Pretty Hostname:**

```
``sh
sudo hostnamectl set-pretty-hostname MyNewPrettyHostname
``
```

4. **Change the Icon Name (e.g., for use in desktop environments):**

```
``sh
sudo hostnamectl set-icon-name com.example.myicon
``
```

5. **Set a New Chassis Type:**

```
``sh
sudo hostnamectl set-chassis vm
``
```

Overall, `hostnamectl` is a useful tool for managing and querying host name-related information on a Linux system.

```
```bash
```

```
hostnamectl
```

```
```
```

**Executing command block via mshell:**

**Static hostname: asus**

**Icon name: computer-convertible**

**Chassis: convertible**

**Machine ID: 74078ee5bb564302b4883bd2ba582acc**

**Boot ID: 33e3e0ccd06b425a834c8e18848aab39**

**Operating System: Ubuntu 24.04.2 LTS**

**Kernel: Linux 6.8.0-59-generic**

**Architecture: x86-64**

**Hardware Vendor: ASUSTeK COMPUTER INC.**

**Hardware Model: VivoBook\_ASUSLaptop TP412FA**

**Firmware Version: TP412FA.302**

**Firmware Date: Mon 2019-05-06**

**Firmware Age: 6y 6d**

Example 3: Checking disk usage

The `df -h` command is used to display disk space usage in a human-readable format.

When you run this command, it shows you the amount of disk space available and used on each mounted filesystem on your system.

Here's what each part of the command does:

- **`df`:** This stands for "disk free," and it is the command used to report file system disk space usage.
- **`-h`:** The `-h` option stands for "human-readable." It makes the output more user-friendly by converting large numbers into megabytes (M), gigabytes (G), terabytes (T), etc.

When you execute `df -h`, the output typically includes columns such as:

- **Filesystem:** The device or partition on which the filesystem is mounted.
- **Size:** The total size of the filesystem.
- **Used:** The amount of space currently used on the filesystem.
- **Avail:** The available space left on the filesystem.
- **Use%:** The percentage of the filesystem that is currently used.
- **Mounted on:** The mount point where the filesystem is located.

This information helps you understand how your disk space is being utilized and identify any filesystems that are running out of space.

```
``bash
```

```
df -h
```

```
````
```

Executing command block via mshell:

Filesystem Size Used Avail Use% Mounted on

```
tmpfs      2.0G 197M 1.8G 10% /run
/dev/nvme0n1p5 284G 192G 77G 72% /
tmpfs      9.7G  0 9.7G  0% /dev/shm
tmpfs      5.0M 8.0K 5.0M  1% /run/lock
efivarfs   192K 63K 125K 34% /sys/firmware/efi/efivars
/dev/nvme0n1p1 256M 32M 225M 13% /boot/efi
tmpfs      2.0G 168K 2.0G  1% /run/user/1001
/dev/sdb2   1.9T 7.1G 1.9T  1% /media/igor/Backup Plus
/dev/sda2   4.6T 2.3T 2.3T 50% /media/igor/twodr
```

Example 4: Listing your own processes

The ``ps -u igor`` command is used to display a snapshot of the current processes running on the system for the user named "igor." Here's a breakdown of what each part of the command does:

- `ps``: This stands for "process status," and it is a command used to report a snapshot of the current processes.

- `-u igor``: The `-u`` option specifies that you want to list processes owned by a specific user. In this case, the user is "igor."

When you run ``ps -u igor``, the output will show you details about all the processes currently running for the user "igor." This includes information such as:

- `PID (Process ID)`: A unique identifier for each process.

- **USER:** The username of the owner of the process.
- **%CPU:** The percentage of CPU time used by the process relative to the total available CPU time.
- **%MEM:** The percentage of physical memory used by the process relative to the total available physical memory.
- **VSZ (Virtual Memory Size):** The total virtual memory used by the process, including both resident and non-resident memory.
- **RSS (Resident Set Size):** The amount of physical memory that the process is actually using.
- **TTY:** The terminal associated with the process.
- **STAT:** The current state of the process (e.g., running, sleeping).
- **START:** The time when the process started.
- **TIME:** The total CPU time used by the process.
- **COMMAND:** The command that was executed to start the process.

This information can be useful for monitoring and managing system resources, identifying resource-intensive processes, or troubleshooting issues related to specific users.

```
``bash  
ps -u igor  
...
```

Executing command block via mshell:

| PID | TTY | TIME | CMD |
|------------|------------|-------------|------------|
|------------|------------|-------------|------------|

| | |
|--------|--------------------------|
| 3309 ? | 00:00:02 systemd |
| 3312 ? | 00:00:00 (sd-pam) |
| 3324 ? | 00:00:03 pipewire |
| 3325 ? | 00:00:00 pipewire |
| 3328 ? | 00:00:02 wireplumber |
| 3329 ? | 00:00:01 pipewire-pulse |
| 3330 ? | 00:00:00 gnome-keyring-d |
| 3345 ? | 00:00:02 dbus-daemon |
| 3367 ? | 00:00:00 gnome-session-b |
| 3378 ? | 00:00:00 xdg-document-po |
| 3392 ? | 00:00:00 xdg-permission- |
| 3511 ? | 00:00:00 at-spi-bus-laun |
| 3518 ? | 00:00:00 dbus-daemon |
| 3528 ? | 00:00:00 gcr-ssh-agent |
| 3534 ? | 00:00:00 gnome-session-c |
| 3549 ? | 00:00:00 gvfsd |
| 3558 ? | 00:00:00 gvfsd-fuse |
| 3562 ? | 00:00:00 gnome-session-b |
| 3595 ? | 00:12:06 gnome-shell |
| 3636 ? | 00:00:33 mutter-x11-fram |
| 3639 ? | 00:00:08 at-spi2-registr |
| 3668 ? | 00:00:00 gnome-shell-cal |
| 3675 ? | 00:00:00 evolution-sourc |
| 3688 ? | 00:00:00 gjs |

| | | |
|--------|----------|-----------------|
| 3722 ? | 00:02:39 | ibus-daemon |
| 3723 ? | 00:00:00 | gsd-a11y-settin |
| 3726 ? | 00:00:02 | gsd-color |
| 3728 ? | 00:00:00 | gsd-datetime |
| 3732 ? | 00:00:32 | gsd-housekeepin |
| 3736 ? | 00:00:00 | gsd-keyboard |
| 3738 ? | 00:00:01 | gsd-media-keys |
| 3741 ? | 00:00:06 | gsd-power |
| 3742 ? | 00:00:00 | gsd-print-notif |
| 3743 ? | 00:00:00 | gsd-rfkill |
| 3745 ? | 00:00:00 | gsd-screensaver |
| 3749 ? | 00:00:03 | gsd-sharing |
| 3752 ? | 00:00:00 | gsd-smartcard |
| 3753 ? | 00:00:00 | gsd-disk-utilit |
| 3756 ? | 00:00:00 | indicator-messa |
| 3758 ? | 00:00:00 | gsd-sound |
| 3763 ? | 00:00:00 | gsd-wacom |
| 3774 ? | 00:00:09 | gsd-xsettings |
| 3786 ? | 00:01:05 | gvfs-udisks2-vo |
| 3789 ? | 00:00:01 | evolution-alarm |
| 3850 ? | 00:00:00 | goa-daemon |
| 3895 ? | 00:00:00 | ibus-dconf |
| 3901 ? | 00:00:22 | ibus-extension- |
| 3907 ? | 00:00:00 | ibus-x11 |

| | | |
|------------|----------|-------------------|
| 3913 ? | 00:00:00 | ibus-portal |
| 3916 ? | 00:00:00 | evolution-calen |
| 3934 ? | 00:00:00 | goa-identity-se |
| 3943 ? | 00:00:00 | gsd-printer |
| 3978 ? | 00:00:00 | evolution-addre |
| 4012 ? | 00:00:00 | gvfs-goa-volume |
| 4017 ? | 00:00:00 | gvfs-mtp-volume |
| 4022 ? | 00:00:18 | gvfs-afc-volume |
| 4028 ? | 00:00:00 | gvfs-gphoto2-vo |
| 4039 ? | 00:00:00 | gjs |
| 4062 ? | 00:00:54 | ibus-engine-sim |
| 4071 ? | 00:00:00 | snapped-desktop-i |
| 4073 ? | 00:00:02 | tracker-miner-f |
| 4140 ? | 00:00:07 | snapped-desktop-i |
| 4200 ? | 00:00:00 | gvfsd-trash |
| 4238 ? | 00:00:09 | xdg-desktop-por |
| 4243 ? | 00:00:17 | xdg-desktop-por |
| 4265 ? | 00:00:09 | xdg-desktop-por |
| 4278 ? | 00:00:17 | gjs |
| 4295 ? | 00:00:00 | gvfsd-metadata |
| 4378 ? | 00:05:34 | gnome-terminal- |
| 4397 ? | 00:00:00 | gnome-terminal |
| 4398 ? | 00:00:00 | gnome-terminal. |
| 4403 pts/1 | 00:00:00 | mshell |

4404 pts/1 00:00:00 sh
4411 ? 00:00:11 update-notifier
5151 ? 00:00:00 dconf-service
9131 ? 00:00:00 ssh-agent
765987 ? 00:00:00 gnome-terminal
765988 ? 00:00:00 gnome-terminal.
765993 pts/0 00:00:00 mshell
778859 ? 00:00:00 gvfsd-network
778870 ? 00:00:00 gvfsd-smb-brows
778881 ? 00:00:00 gvfsd-dnssd
779867 ? 00:00:00 gnome-terminal
779868 ? 00:00:00 gnome-terminal.
779873 pts/3 00:00:00 mshell
789930 ? 00:00:00 gnome-terminal
789931 ? 00:00:00 gnome-terminal.
789936 pts/2 00:00:00 mshell
789940 pts/2 00:02:12 gedit
796024 ? 00:00:00 gnome-terminal
796025 ? 00:00:00 gnome-terminal.
796030 pts/4 00:00:00 mshell
796037 pts/4 00:00:00 test.ms
796094 pts/4 00:00:00 sh
796095 pts/4 00:00:00 ps

Example 5: Checking system uptime

The ``uptime`` command displays the current time, how long the system has been running, and statistics about the number of active users and load averages.

When you run ``uptime``, the output typically includes:

1. **Current Time:** The exact time when the command was executed.
2. **System Uptime:** How long the system has been up, including days, hours, minutes, and seconds.
3. **Number of Users:** The current number of users logged into the system.
4. **Load Averages:** Three load averages:
 - 1-minute average: The average number of processes that are in a runnable state or waiting for resources over the past minute.
 - 5-minute average: Over the past five minutes.
 - 15-minute average: Over the past fifteen minutes.

For example, an output might look like this:

```
...
```

```
20:34:56 up 2 days, 3:45, 2 users, load average: 0.12, 0.08, 0.05
```

```
...
```

This indicates that the system has been running for 2 days and 3 hours and 45 minutes, with 2 users currently logged in. The load averages suggest that there are relatively few processes competing for resources.

The `uptime` command is a quick way to get an overview of the system's current state and resource usage.

```
``bash
```

```
uptime
```

```
...
```

Executing command block via mshell:

```
08:22:23 up 3 days, 10:00, 1 user, load average: 0.05, 0.07, 0.07
```

Example 6: Checking CPU model

The `cat /proc/cpuinfo` command displays detailed information about the CPU(s) in the system.

When you run `cat /proc/cpuinfo`, it reads and outputs the contents of the `/proc/cpuinfo` file. This file provides a wealth of information about each CPU core on your system, including:

- **Model name:** The model or type of CPU.
- **Processor:** A numerical identifier for the CPU core.
- **Core id:** The ID number of the core within its package.
- **Socket id:** The socket or physical location where the CPU is installed.
- **CPU MHz:** The current speed of the CPU in megahertz.
- **Caches:** Information about various cache levels (L1, L2, L3).
- **Flags:** Various flags indicating features supported by the CPU.

This information is crucial for system administrators and developers to understand the hardware capabilities of their machines and optimize performance accordingly.

```
```bash
```

```
cat /proc/cpuinfo
```

```
```
```

Executing command block via mshell:

```
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 142
model name    : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
stepping      : 11
microcode     : 0xf6
cpu MHz       : 900.002
cache size    : 6144 KB
physical id   : 0
siblings      : 8
core id       : 0
cpu cores     : 4
apicid        : 0
initial apicid : 0
fpu           : yes
fpu_exception : yes
```

cpuid level : 22
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe
popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch
cpuid_fault epb ssbd ibrs ibpb stibp tpr_shadow flexpriority ept vpid ept_ad
fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap
clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp
hwp_notify hwp_act_window hwp_epp vnmi md_clear flush_l1d arch_capabilities
vmx flags : vnmi preemption_timer invvpid ept_x_only ept_ad ept_1gb
flexpriority tsc_offset vtptr mtf vapic ept vpid unrestricted_guest ple pml
ept_mode_based_exec
bugs : spectre_v1 spectre_v2 spec_store_bypass mds swapgs
itlb_multihit srbds mmio_stale_data retbleed gds
bogomips : 3600.00
clflush size : 64
cache_alignment : 64
address sizes : 39 bits physical, 48 bits virtual
power management:
processor : 1
vendor_id : GenuineIntel

cpu family : 6
model : 142
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
stepping : 11
microcode : 0xf6
cpu MHz : 900.214
cache size : 6144 KB
physical id : 0
siblings : 8
core id : 1
cpu cores : 4
apicid : 2
initial apicid : 2
fpu : yes
fpu_exception : yes
cpuid level : 22
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe
popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch
cpuid_fault epb ssbd ibrs ibpb stibp tpr_shadow flexpriority ept vpid ept_ad

fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap
clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp
hwp_notify hwp_act_window hwp_epp vnmi md_clear flush_l1d arch_capabilities
vmx flags : vnmi preemption_timer invvpid ept_x_only ept_ad ept_1gb
flexpriority tsc_offset vtptr mtf vapic ept vpid unrestricted_guest ple pml
ept_mode_based_exec
bugs : spectre_v1 spectre_v2 spec_store_bypass mds swapgs
itlb_multihit srbds mmio_stale_data retbleed gds
bogomips : 3600.00
clflush size : 64
cache_alignment : 64
address sizes : 39 bits physical, 48 bits virtual
power management:

processor : 2
vendor_id : GenuineIntel
cpu family : 6
model : 142
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
stepping : 11
microcode : 0xf6
cpu MHz : 900.129
cache size : 6144 KB
physical id : 0

siblings : 8
core id : 2
cpu cores : 4
apicid : 4
initial apicid : 4
fpu : yes
fpu_exception : yes
cpuid level : 22
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe
popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch
cpuid_fault epb ssbd ibrs ibpb stibp tpr_shadow flexpriority ept vpid ept_ad
fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap
clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp
hwp_notify hwp_act_window hwp_epp vnmi md_clear flush_l1d arch_capabilities
vmx flags : vnmi preemption_timer invvpid ept_x_only ept_ad ept_1gb
flexpriority tsc_offset vtptr mtf vapic ept vpid unrestricted_guest ple pml
ept_mode_based_exec
bugs : spectre_v1 spectre_v2 spec_store_bypass mds swapgs
itlb_multihit srbds mmio_stale_data retbleed gds

bogomips : 3600.00
clflush size : 64
cache_alignment : 64
address sizes : 39 bits physical, 48 bits virtual

power management:

processor : 3
vendor_id : GenuineIntel
cpu family : 6
model : 142
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
stepping : 11
microcode : 0xf6
cpu MHz : 400.000
cache size : 6144 KB
physical id : 0
siblings: 8
core id : 3
cpu cores : 4
apicid : 6
initial apicid : 6
fpu : yes
fpu_exception : yes
cpuid level : 22

wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe
popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch
cpuid_fault epb ssbd ibrs ibpb stibp tpr_shadow flexpriority ept vpid ept_ad
fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap
clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp
hwp_notify hwp_act_window hwp_epp vnmi md_clear flush_l1d arch_capabilities

vmx flags : vnmi preemption_timer invvpid ept_x_only ept_ad ept_1gb
flexpriority tsc_offset vtptr mtf vapic ept vpid unrestricted_guest ple pml
ept_mode_based_exec

bugs : spectre_v1 spectre_v2 spec_store_bypass mds swapgs
itlb_multihit srbds mmio_stale_data retbleed gds

bogomips : 3600.00

clflush size : 64

cache_alignment : 64

address sizes : 39 bits physical, 48 bits virtual

power management:

processor : 4

vendor_id : GenuineIntel

cpu family : 6
model : 142
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
stepping : 11
microcode : 0xf6
cpu MHz : 400.000
cache size : 6144 KB
physical id : 0
siblings : 8
core id : 0
cpu cores : 4
apicid : 1
initial apicid : 1
fpu : yes
fpu_exception : yes
cpuid level : 22
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe
popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch
cpuid_fault epb ssbd ibrs ibpb stibp tpr_shadow flexpriority ept vpid ept_ad

fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap
clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp
hwp_notify hwp_act_window hwp_epp vnmi md_clear flush_l1d arch_capabilities
vmx flags : vnmi preemption_timer invvpid ept_x_only ept_ad ept_1gb
flexpriority tsc_offset vtptr mtf vapic ept vpid unrestricted_guest ple pml
ept_mode_based_exec
bugs : spectre_v1 spectre_v2 spec_store_bypass mds swapgs
itlb_multihit srbds mmio_stale_data retbleed gds
bogomips : 3600.00
clflush size : 64
cache_alignment : 64
address sizes : 39 bits physical, 48 bits virtual
power management:

processor : 5
vendor_id : GenuineIntel
cpu family : 6
model : 142
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
stepping : 11
microcode : 0xf6
cpu MHz : 900.007
cache size : 6144 KB
physical id : 0

siblings: 8

core id : 1

cpu cores : 4

apicid : 3

initial apicid : 3

fpu : yes

fpu_exception : yes

cpuid level : 22

wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca

cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx

pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl

xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl

vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe

popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch

cpuid_fault epb ssbd ibrs ibpb stibp tpr_shadow flexpriority ept vpid ept_ad

fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap

clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp

hwp_notify hwp_act_window hwp_epp vnmi md_clear flush_l1d arch_capabilities

vmx flags : vnmi preemption_timer invvpid ept_x_only ept_ad ept_1gb

flexpriority tsc_offset vtptr mtf vapic ept vpid unrestricted_guest ple pml

ept_mode_based_exec

bugs : spectre_v1 spectre_v2 spec_store_bypass mds swapgs

itlb_multihit srbds mmio_stale_data retbleed gds

bogomips : 3600.00
clflush size : 64
cache_alignment : 64
address sizes : 39 bits physical, 48 bits virtual

power management:

processor : 6
vendor_id : GenuineIntel
cpu family : 6
model : 142
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
stepping : 11
microcode : 0xf6
cpu MHz : 899.982
cache size : 6144 KB
physical id : 0
siblings : 8
core id : 2
cpu cores : 4
apicid : 5
initial apicid : 5
fpu : yes
fpu_exception : yes
cpuid level : 22

wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe
popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch
cpuid_fault epb ssbd ibrs ibpb stibp tpr_shadow flexpriority ept vpid ept_ad
fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap
clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp
hwp_notify hwp_act_window hwp_epp vnmi md_clear flush_l1d arch_capabilities

vmx flags : vnmi preemption_timer invvpid ept_x_only ept_ad ept_1gb
flexpriority tsc_offset vtptr mtf vapic ept vpid unrestricted_guest ple pml
ept_mode_based_exec

bugs : spectre_v1 spectre_v2 spec_store_bypass mds swapgs
itlb_multihit srbds mmio_stale_data retbleed gds

bogomips : 3600.00

clflush size : 64

cache_alignment : 64

address sizes : 39 bits physical, 48 bits virtual

power management:

processor : 7

vendor_id : GenuineIntel

cpu family : 6
model : 142
model name : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
stepping : 11
microcode : 0xf6
cpu MHz : 900.013
cache size : 6144 KB
physical id : 0
siblings: 8
core id : 3
cpu cores : 4
apicid : 7
initial apicid : 7
fpu : yes
fpu_exception : yes
cpuid level : 22
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe
popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch
cpuid_fault epb ssbd ibrs ibpb stibp tpr_shadow flexpriority ept vpid ept_ad

fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap
clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp
hwp_notify hwp_act_window hwp_epp vnmi md_clear flush_l1d arch_capabilities
vmx flags : vnmi preemption_timer invvpid ept_x_only ept_ad ept_1gb
flexpriority tsc_offset vtptr mtf vapic ept vpid unrestricted_guest ple pml
ept_mode_based_exec
bugs : spectre_v1 spectre_v2 spec_store_bypass mds swapgs
itlb_multihit srbds mmio_stale_data retbleed gds
bogomips : 3600.00
clflush size : 64
cache_alignment : 64
address sizes : 39 bits physical, 48 bits virtual
power management:

Example 7: Checking memory information

The `free -h` command displays information about the amount of physical memory (RAM) and swap space used or available on the system in a human-readable format.

When you run `free -h`, the output typically includes:

- **total:** The total amount of memory or swap space.
- **used:** The amount of memory or swap space currently being used.
- **free:** The amount of memory or swap space that is not being used.

- **shared:** The amount of memory used for shared memory segments.
- **buff/cache:** The amount of memory used by the kernel as a buffer and cache. This includes memory used for caching file system data, directory structures, and other system-related data.
- **available:** The amount of memory that is available to start new applications, without swapping.

The `-h` option makes the output more readable by converting large numbers into megabytes (M), gigabytes (G), etc.

For example, an output might look like this:

```
...  
  
      total    used    free   shared buff/cache   available  
Mem:   7.7G    1.8G    2.3G    204M    3.6G    5.0G  
Swap:  2.0G     0B    2.0G  
...
```

This indicates that the system has a total of 7.7 gigabytes of memory, with 1.8 gigabytes currently in use, 2.3 gigabytes free, and 3.6 gigabytes available for applications without swapping to disk. The swap space also shows a total of 2.0 gigabytes, but none is being used at the moment.

The `free -h` command is useful for monitoring memory usage and identifying whether there is enough memory available for current system operations or if additional physical memory may be required.

```
``bash  
free -h
```

...

Executing command block via mshell:

| | total | used | free | shared | buff/cache | available |
|-------|-------|-------|-------|--------|------------|-----------|
| Mem: | 19Gi | 3.2Gi | 5.2Gi | 511Mi | 11Gi | 16Gi |
| Swap: | 24Gi | 512Ki | 24Gi | | | |

Example 8: Showing network interface names

The ``ls /sys/class/net`` command lists all the network interfaces present in the system.

When you run ``ls /sys/class/net``, it displays the names of all network interfaces available on your system. These interfaces are typically represented as directories under ``/sys/class/net``.

Each directory represents a specific network interface, and its name corresponds to the interface's identifier (e.g., `eth0`, `wlan0`, `lo`). Inside each directory, you can find various files and subdirectories that provide information about the interface's status, configuration, statistics, etc.

For example, running ``ls /sys/class/net`` might produce an output like this:

...

```
eth0 lo wlan0
```

...

This indicates that the system has three network interfaces: ``eth0``, ``lo``, and ``wlan0``.

The ``/sys/class/net`` directory provides a view of the network hardware as seen by the Linux kernel, making it useful for developers working with networking code or administrators who need to programmatically access network interface information.

```
```bash
```

```
ls /sys/class/net
```

```
```
```

Executing command block via mshell:

```
docker0 lo wlo1
```

Example 9: Checking system temperature

The `sensors` command displays a real-time reading of hardware sensors on your system, such as temperature, fan speeds, and voltage levels.

When you run `sensors`, it typically outputs data from various hardware monitoring chips that are connected to the system. The output includes information about different types of sensors, such as:

- **Temperature:** Current temperature readings for CPU, motherboard, hard drives, etc.
- **Fan Speed:** RPM values for fans.
- **Voltage:** Power supply voltages.

The exact output depends on the hardware and the sensor drivers that are installed on your system. Some common sensors you might see include:

```
```
```

```
Core 0: +45.2°C (high = +100.0°C, crit = +108.0°C)
```

```
Core 1: +44.9°C (high = +100.0°C, crit = +108.0°C)
```

CPUID 3c0000f1: +52.6°C (high = +100.0°C, crit = +108.0°C)

Northbridge: +97.4°C (crit = +130.0°C)

Southbridge: +71.3°C (crit = +100.0°C)

...

This information is crucial for monitoring the health and performance of your hardware, as well as detecting any potential issues that might require attention.

The `sensors` command is typically part of the `lm-sensors` package, which provides a framework for reading data from various hardware sensors on Linux systems. If you don't have this package installed, you can usually install it using your system's package manager (e.g., `sudo apt-get install lm-sensors` on Debian-based systems or `sudo yum install lm_sensors` on Red Hat-based systems).

```
``bash
```

```
sensors
```

```
...
```

### Executing command block via mshell:

**coretemp-isa-0000**

**Adapter: ISA adapter**

**Package id 0: +35.0°C (high = +100.0°C, crit = +100.0°C)**

**Core 0: +35.0°C (high = +100.0°C, crit = +100.0°C)**

**Core 1: +34.0°C (high = +100.0°C, crit = +100.0°C)**

**Core 2: +34.0°C (high = +100.0°C, crit = +100.0°C)**

**Core 3: +34.0°C (high = +100.0°C, crit = +100.0°C)**

**asus-isa-0000**

**Adapter: ISA adapter**

**cpu\_fan: 0 RPM**

**nvme-pci-0100**

**Adapter: PCI adapter**

**Composite: +25.9°C (low = -0.1°C, high = +76.8°C)  
(crit = +79.8°C)**

**acpitz-acpi-0**

**Adapter: ACPI interface**

**temp1: +36.0°C**

**iwlwifi\_1-virtual-0**

**Adapter: Virtual device**

**temp1: +34.0°C**

**pch\_cannonlake-virtual-0**

**Adapter: Virtual device**

**temp1: +32.0°C**

**BAT0-acpi-0**

**Adapter: ACPI interface**

**in0: 11.85 V**

curr1: 0.00 A

### Example 10: Showing kernel information

The `uname -a` command displays a variety of system-related information.

When you run `uname -a`, it outputs the following details:

- **Kernel Name:** The name of the operating system kernel.
- **Network Node Hostname:** The hostname of the machine on the network.
- **Kernel Version:** The version number of the kernel.
- **Machine Hardware Name:** The hardware architecture of the system (e.g., `x86_64`).
- **Processor Type:** The type of processor or CPU.
- **Hardware Platform:** The platform or machine model.
- **Operating System:** The name of the operating system.

For example, an output might look like this:

...

```
Linux myhost 5.10.0-29-generic #32-Ubuntu SMP Thu Jul 8 04:20:17 UTC 2021 x86_64
x86_64 x86_64 GNU/Linux
```

...

This indicates that the system is running Linux, with a specific kernel version, on an `x86_64` architecture processor.

The `uname -a` command provides essential information about the system's hardware and operating environment, making it useful for debugging and administrative tasks.

```
```bash
```

```
uname -a
```

```
```
```

**Executing command block via mshell:**

```
Linux asus 6.8.0-59-generic #61-Ubuntu SMP PREEMPT_DYNAMIC Fri Apr 11
```

```
23:16:11 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
```

```
/home/igor/mshell/mscripts >
```

## **7. Mshell script example2a without comments for commands from LLM for declarative run (example2a)**

```
#!/bin/mshell
```

```
system_error_checker.ms - Commands to check for errors in Ubuntu
```

```
Example 1: Show recent errors in kernel log
```

```
echo "Checking recent kernel errors (dmesg):"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: '\sudo dmesg --
```

```
level=err,warn\' "
```

```
Example 2: Check systemd journal errors
```

```
echo ""
```

```
echo "Checking systemd journal errors:"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: sudo journalctl -xn
10 "
```

```
Example 3: Check authentication failures
```

```
echo ""
```

```
echo "Checking authentication failures:"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: sudo grep
'authentication failure' /var/log/auth.log "
```

```
Example 4: Check for failed services
```

```
echo ""
```

```
echo "Checking for failed systemd services:"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: sudo systemctl --
failed "
```

```
Example 5: Check for disk errors
```

```
echo ""
```

```
echo "Checking for disk errors:"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: sudo grep -i error
/var/log/kern.log "
```

```
Example 6: Check for SSH login failures
```

```
echo ""
```

```
echo "Checking for SSH login failures:"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: sudo grep 'Failed password' /var/log/auth.log "
```

```
Example 7: Check application crashes
```

```
echo ""
```

```
echo "Checking for application crashes:"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: sudo ls -lt /var/crash "
```

```
Example 8: Check boot logs for errors
```

```
echo ""
```

```
echo "Checking boot logs for errors:"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: sudo dmesg | grep -i error "
```

```
Example 9: Check apt package manager errors
```

```
echo ""
```

```
echo "Checking apt package manager errors:"
```

```
ollama1exec "No any comments. Run the Ubuntu Linux command: sudo grep -i error /var/log/apt/term.log "
```

```
Example 10: Check for hardware failures
```

```
echo ""
```

```
echo "Checking hardware diagnostics:"
```

ollama1exec "No any comments. Run the Ubuntu Linux command: sudo lshw -short "

## 8. Result of script execution for example2a

```
/home/igor/mshell/mscripts > test1-nocomments.ms
```

```
Loading .mshellrc...
```

```
Loaded environment: PATH=/usr/bin:/bin:/usr/sbin:/usr/local/bin:.$PATH
```

```
Loaded alias: ll='ls -l'
```

```
Loaded alias: home='cd ~'
```

```
Loaded environment: EDITOR=nano
```

```
Lua state initialized successfully
```

```
Range function defined successfully
```

```
Loading history...
```

```
Checking recent kernel errors (dmesg):
```

```
DEBUG: Variable assignment value = 'err,warn\''
```

Certainly! The `dmesg` command in Linux is used to view or control the kernel ring buffer, which contains messages from the kernel and drivers. When you run `sudo dmesg --level=err,warn`, it filters the output to show only error (err) and warning (warn) messages.

Here is how you can run this command:

```
``sh
```

```
sudo dmesg --level=err,warn
```

```
...
```

This will display all kernel messages that are classified as errors or warnings. If you want to see these messages in real-time, you can use the `-w` option along with `tail -f`:

```
``sh
```

```
sudo dmesg --level=err,warn | tail -f
```

```
...
```

This will continuously monitor and display any new error or warning messages as they appear in the kernel ring buffer.

Executing command block via mshell:

**DEBUG: Variable assignment value = 'err,warn'**

**[sudo] password for igor:**

**[ 0.194384] x86/cpu: SGX disabled by BIOS.**

**[ 0.204376] MDS CPU bug present and SMT on, data leak possible. See <https://www.kernel.org/doc/html/latest/admin-guide/hw-vuln/mds.html> for more details.**

**[ 0.204384] MMIO Stale Data CPU bug present and SMT on, data leak possible. See [https://www.kernel.org/doc/html/latest/admin-guide/hw-vuln/processor\\_mmio\\_stale\\_data.html](https://www.kernel.org/doc/html/latest/admin-guide/hw-vuln/processor_mmio_stale_data.html) for more details.**

**[ 0.783212] hpet\_acpi\_add: no address or irqs in \_CRS**

**[ 0.800576] i8042: PNP: PS/2 appears to have AUX port disabled, if this is incorrect please boot with i8042.nopnp**

**[ 0.806259] device-mapper: core: CONFIG\_IMA\_DISABLE\_HTABLE is disabled. Duplicate IMA measurements will not be recorded in the IMA log.**

**[ 0.806377] platform eisa.0: EISA: Cannot allocate resource for mainboard**

**[ 0.806379] platform eisa.0: Cannot allocate resource for EISA slot 1**

**[ 0.806381] platform eisa.0: Cannot allocate resource for EISA slot 2**

**[ 0.806383] platform eisa.0: Cannot allocate resource for EISA slot 3**

```
[0.806385] platform eisa.0: Cannot allocate resource for EISA slot 4
[0.806386] platform eisa.0: Cannot allocate resource for EISA slot 5
[0.806388] platform eisa.0: Cannot allocate resource for EISA slot 6
[0.806390] platform eisa.0: Cannot allocate resource for EISA slot 7
[0.806391] platform eisa.0: Cannot allocate resource for EISA slot 8
[0.873930] ENERGY_PERF_BIAS: Set to 'normal', was 'performance'
[0.912473] blacklist: Duplicate blacklisted hash
bin:47ff1b63b140b6fc04ed79131331e651da5b2e2f170f5daef4153dc2fbc532b1
[0.912589] blacklist: Duplicate blacklisted hash
bin:992d359aa7a5f789d268b94c11b9485a6b1ce64362b0edb4441ccc187c39647b
[0.912665] blacklist: Duplicate blacklisted hash
bin:c452ab846073df5ace25cca64d6b7a09d906308a1a65eb5240e3c4ebcaa9cc0c
[0.912710] blacklist: Duplicate blacklisted hash
bin:e051b788ecbaeda53046c70e6af6058f95222c046157b8c4c1b9c2cfc65f46e5
[1.333477] i2c i2c-0: Failed to register i2c client XXXX0000:00 at 0x15 (-16)
[1.334887] i2c i2c-0: Failed to register i2c client XXXX0000:02 at 0x15 (-16)
[1.341597] usb: port power management may be unreliable
[1.535986] i2c_hid_acpi i2c-ELAN2097:00: i2c_hid_get_input: IRQ triggered but
there's no data
[1.634231] i2c_hid_acpi i2c-ELAN1300:00: device returned incorrect report (2 vs
14 expected)
[1.634238] hid-multitouch 0018:04F3:310C.0001: failed to fetch feature 14
[2.865566] sd 16:0:0:0: [sda] Optimal transfer size 33553920 bytes not a
multiple of preferred minimum block size (4096 bytes)
```

[ 2.877791] sd 17:0:0:0: [sdb] Optimal transfer size 33553920 bytes not a multiple of preferred minimum block size (4096 bytes)

[ 6.492015] iwlwifi 0000:00:14.3: WRT: Overriding region id 0

[ 6.492026] iwlwifi 0000:00:14.3: WRT: Overriding region id 1

[ 6.492032] iwlwifi 0000:00:14.3: WRT: Overriding region id 2

[ 6.492036] iwlwifi 0000:00:14.3: WRT: Overriding region id 3

[ 6.492040] iwlwifi 0000:00:14.3: WRT: Overriding region id 4

[ 6.492044] iwlwifi 0000:00:14.3: WRT: Overriding region id 6

[ 6.492048] iwlwifi 0000:00:14.3: WRT: Overriding region id 8

[ 6.492052] iwlwifi 0000:00:14.3: WRT: Overriding region id 9

[ 6.492056] iwlwifi 0000:00:14.3: WRT: Overriding region id 10

[ 6.492061] iwlwifi 0000:00:14.3: WRT: Overriding region id 11

[ 6.492065] iwlwifi 0000:00:14.3: WRT: Overriding region id 15

[ 6.492069] iwlwifi 0000:00:14.3: WRT: Overriding region id 16

[ 6.492073] iwlwifi 0000:00:14.3: WRT: Overriding region id 18

[ 6.492077] iwlwifi 0000:00:14.3: WRT: Overriding region id 19

[ 6.492082] iwlwifi 0000:00:14.3: WRT: Overriding region id 20

[ 6.492087] iwlwifi 0000:00:14.3: WRT: Overriding region id 21

[ 6.492091] iwlwifi 0000:00:14.3: WRT: Overriding region id 28

[ 6.567882] asus\_wmi: fan\_curve\_get\_factory\_default (0x00110024) failed: -61

[ 6.570043] asus\_wmi: fan\_curve\_get\_factory\_default (0x00110025) failed: -61

[ 6.572161] asus\_wmi: fan\_curve\_get\_factory\_default (0x00110032) failed: -61

[ 6.662788] Bluetooth: hci0: HCI LE Coded PHY feature bit is set, but its usage is not supported.

```
[7.554461] thermal thermal_zone5: failed to read out thermal zone (-61)
[8.096083] thermal thermal_zone5: failed to read out thermal zone (-61)
[10.280790] block nvme0n1: No UUID available providing old NGUID
[13.372694] kauditd_printk_skb: 256 callbacks suppressed
[416.030339] workqueue: i915_hpd_poll_init_work [i915] hogged CPU for
>13333us 4 times, consider switching to WQ_UNBOUND
[471.970064] workqueue: output_poll_execute hogged CPU for >13333us 4
times, consider switching to WQ_UNBOUND
[532.250197] workqueue: i915_hpd_poll_init_work [i915] hogged CPU for
>13333us 8 times, consider switching to WQ_UNBOUND
[590.760979] workqueue: output_poll_execute hogged CPU for >13333us 8
times, consider switching to WQ_UNBOUND
[772.390018] workqueue: i915_hpd_poll_init_work [i915] hogged CPU for
>13333us 16 times, consider switching to WQ_UNBOUND
[831.916253] workqueue: output_poll_execute hogged CPU for >13333us 16
times, consider switching to WQ_UNBOUND
[1484.665405] thermal thermal_zone5: failed to read out thermal zone (-61)
[1485.307409] workqueue: i915_hpd_poll_init_work [i915] hogged CPU for
>13333us 32 times, consider switching to WQ_UNBOUND
[1512.426692] exFAT-fs (sdb2): Volume was not properly unmounted. Some data
may be corrupt. Please run fsck.
[1513.622094] exFAT-fs (sda2): Volume was not properly unmounted. Some data
may be corrupt. Please run fsck.
```

**[ 1757.842845] warning: `ThreadPoolForeg' uses wireless extensions which will stop working for Wi-Fi 7 hardware; use nl80211**

**[ 5915.245843] kauditd\_printk\_skb: 13 callbacks suppressed**

**[68826.273477] i915 0000:00:02.0: [drm] \*ERROR\* CPU pipe A FIFO underrun**

**[178714.591682] kauditd\_printk\_skb: 25 callbacks suppressed**

Checking systemd journal errors:

Certainly! The `journalctl` command is used to query the system log, which includes various types of log messages generated by different parts of the system. When you run `sudo journalctl -xn 10`, it displays the 10 most recent entries from the system log, including their priorities and timestamps.

Here is how you can run this command:

```
``sh
```

```
sudo journalctl -xn 10
```

```
````
```

This will output the 10 most recent messages from the system log, showing their priority levels (e.g., ERR, WARN, INFO) along with timestamps. This is useful for quickly checking the most recent events that occurred on your system.

Executing command block via mshell:

May 11 08:50:03 asus nmbd[2717]: This response was from IP 192.168.1.28, repo>

May 11 08:55:01 asus CRON[796216]: pam_unix(cron:session): session opened for u>

```
May 11 08:55:01 asus CRON[796217]: (root) CMD (command -v debian-sa1 > /dev/nul>
```

```
May 11 08:55:01 asus CRON[796216]: pam_unix(cron:session): session closed for u>
```

```
May 11 08:55:32 asus x11vnc[2082]: 11/05/2025 08:55:32 active keyboard: waiting>
```

```
May 11 08:57:23 asus sudo[796240]: igor : TTY=pts/4 ; PWD=/home/igor/mshell>
```

```
May 11 08:57:23 asus sudo[796240]: pam_unix(sudo:session): session opened for u>
```

```
May 11 08:57:23 asus sudo[796240]: pam_unix(sudo:session): session closed for u>
```

```
May 11 08:58:19 asus sudo[796252]: igor : TTY=pts/4 ; PWD=/home/igor/mshell>
```

```
May 11 08:58:19 asus sudo[796252]: pam_unix(sudo:session): session opened for u>
```

Checking authentication failures:

Certainly! The `grep` command is used to search text using patterns. When you run `sudo grep 'authentication failure' /var/log/auth.log`, it searches for lines containing the phrase "authentication failure" in the `/var/log/auth.log` file, which typically contains authentication-related messages.

Here is how you can run this command:

```
``sh
```

```
sudo grep 'authentication failure' /var/log/auth.log
```

...

This will display all lines from the `/var/log/auth.log` file that contain the phrase "authentication failure". This can be helpful for monitoring and troubleshooting login attempts on your system.

Executing command block via mshell:

```
2025-05-11T03:49:10.724222-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep  
'authentication failure' /var/log/auth.log
```

```
2025-05-11T05:09:52.840641-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep  
'authentication failure' /var/log/auth.log
```

```
2025-05-11T06:09:08.117927-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep  
'authentication failure' /var/log/auth.log
```

```
2025-05-11T09:08:00.036450-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep  
'authentication failure' /var/log/auth.log
```

Checking for failed systemd services:

Certainly! The `systemctl --failed` command is used to list all services that are currently in a failed state on your systemd-based system.

Here is how you can run this command:

```
``sh
```

```
sudo systemctl --failed
```

```
...
```

This will output information about any services that have recently failed, including their names, load status, active status, and the reason for failure. This is useful for diagnosing issues with specific services that are not functioning as expected.

Executing command block via mshell:

```
UNIT LOAD ACTIVE SUB DESCRIPTION
```

```
0 loaded units listed.
```

Checking for disk errors:

Certainly! The `grep` command is used to search text using patterns. When you run `sudo grep -i error /var/log/kern.log`, it searches for lines containing the word "error" in a case-insensitive manner (i.e., it matches "error", "Error", "ERROR", etc.) in the `/var/log/kern.log` file, which typically contains kernel-related messages.

Here is how you can run this command:

```
``sh
```

```
sudo grep -i error /var/log/kern.log
```

```
...
```

This will display all lines from the `/var/log/kern.log` file that contain the word "error" (case-insensitive). This is useful for identifying and troubleshooting kernel-related issues on your system.

Executing command block via mshell:

```
2025-05-11T00:47:46.749348-05:00 asus kernel: gedit[779884]: segfault at f1 ip
00007d142b55fdaf sp 00007ffc6d5bd7b0 error 4 in libgtk-
3.so.0.2409.32[7d142b283000+3ae000] likely on CPU 7 (core 3, socket 0)
```

Checking for SSH login failures:

Certainly! The `grep` command is used to search text using patterns. When you run `sudo grep 'Failed password' /var/log/auth.log`, it searches for lines containing the phrase "Failed password" in the `/var/log/auth.log` file, which typically contains authentication-related messages.

Here is how you can run this command:

```
``sh
sudo grep 'Failed password' /var/log/auth.log
````
```

This will display all lines from the `/var/log/auth.log` file that contain the phrase "Failed password". This is useful for monitoring and troubleshooting login attempts on your system, specifically to identify failed authentication attempts.

### Executing command block via mshell:

```
2025-05-11T03:51:34.355729-05:00 asus sudo: igor : TTY=pts/4 ;
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep 'Failed
password' /var/log/auth.log
```

2025-05-11T05:10:28.271485-05:00 asus sudo: igor : TTY=pts/4 ;

PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep 'Failed password' /var/log/auth.log

2025-05-11T06:11:59.932240-05:00 asus sudo: igor : TTY=pts/4 ;

PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep 'Failed password' /var/log/auth.log

2025-05-11T09:10:35.534679-05:00 asus sudo: igor : TTY=pts/4 ;

PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep 'Failed password' /var/log/auth.log

Checking for application crashes:

Certainly! The `ls` command is used to list directory contents, and when you run `sudo ls -lt /var/crash`, it lists the files in the `/var/crash` directory in reverse chronological order (newest first).

Here is how you can run this command:

```
``sh
```

```
sudo ls -lt /var/crash
```

```
````
```

This will output a list of all files and directories in the `/var/crash` directory, sorted by their modification times. Files with more recent modification times will appear at the top of the list. This is useful for examining crash reports and logs related to system crashes.

Executing command block via mshell:

total 9784

```
-rw-r----- 1 igor whoopsie 10016677 May  7 19:22 _usr_bin_gedit.1001.crash
```

Checking hardware diagnostics:

Certainly! The `lshw` (List Hardware) command provides a detailed description of the hardware present on your computer. When you run `sudo lshw -short`, it displays a brief summary of the system's hardware configuration.

Here is how you can run this command:

```
``sh
```

```
sudo lshw -short
```

```
...
```

This will output a concise list of the system's hardware components, including information about the CPU, memory, storage devices, and other peripherals. This is useful for getting an overview of your system's hardware setup without a lot of detail.

Executing command block via mshell:

| H/W path | Device | Class | Description |
|----------|--------|-------|----------------------------|
| ===== | | | |
| | system | | VivoBook_ASUSLaptop TP412F |
| /0 | bus | | TP412FA |
| /0/0 | memory | | 64KiB BIOS |
| /0/3a | memory | | 20GiB System Memory |
| /0/3a/0 | memory | | 4GiB SODIMM DDR4 Synchrono |
| /0/3a/1 | memory | | 16GiB SODIMM DDR4 Synchron |

| | | | |
|-----------------------|-----------|---------------|----------------------------|
| /0/45 | | memory | 256KiB L1 cache |
| /0/46 | | memory | 1MiB L2 cache |
| /0/47 | | memory | 6MiB L3 cache |
| /0/48 | | processor | Intel(R) Core(TM) i5-8265U |
| /0/100 | | bridge | Coffee Lake HOST and DRAM |
| /0/100/2 | /dev/fb0 | display | WhiskeyLake-U GT2 [UHD Gra |
| /0/100/4 | | generic | Xeon E3-1200 v5/E3-1500 v5 |
| /0/100/8 | | generic | Xeon E3-1200 v5/v6 / E3-15 |
| /0/100/12 | | generic | Cannon Point-LP Thermal Co |
| /0/100/13 | | communication | Cannon Point-LP Integrated |
| /0/100/14 | | bus | Cannon Point-LP USB 3.1 xH |
| /0/100/14/0 | usb1 | bus | xHCI Host Controller |
| /0/100/14/0/1 | scsi16 | storage | One Touch HDD |
| /0/100/14/0/1/0.0.0 | /dev/sda | disk | 5TB One Touch HDD |
| /0/100/14/0/1/0.0.0/1 | /dev/sda1 | volume | 199MiB Windows FAT volume |
| /0/100/14/0/1/0.0.0/2 | /dev/sda2 | volume | 4657GiB data partition |
| /0/100/14/0/2 | scsi17 | storage | BUP Slim |
| /0/100/14/0/2/0.0.0 | /dev/sdb | disk | 2TB BUP Slim |
| /0/100/14/0/2/0.0.0/1 | /dev/sdb1 | volume | 199MiB Windows FAT volume |
| /0/100/14/0/2/0.0.0/2 | /dev/sdb2 | volume | 1862GiB data partition |
| /0/100/14/0/3 | | input | USB Receiver |
| /0/100/14/0/3/0 | input27 | input | Logitech M325 |
| /0/100/14/0/4 | | multimedia | USB2.0 VGA UVC WebCam |
| /0/100/14/0/a | | communication | Bluetooth 9460/9560 Jeffer |

| | | | |
|-----------------|----------------|---------------|----------------------------|
| /0/100/14/1 | usb2 | bus | xHCI Host Controller |
| /0/100/14.2 | | memory | RAM memory |
| /0/100/14.3 | wlo1 | network | Cannon Point-LP CNVi [Wire |
| /0/100/15 | | bus | Cannon Point-LP Serial IO |
| /0/100/15.1 | | bus | Cannon Point-LP Serial IO |
| /0/100/16 | | communication | Cannon Point-LP MEI Contro |
| /0/100/17 | | storage | 82801 Mobile SATA Controll |
| /0/100/1d | | bridge | Cannon Point-LP PCI Expres |
| /0/100/1d/0 | /dev/nvme0 | storage | INTEL SSDPEKNW512G8 |
| /0/100/1d/0/0 | hwmon2 | disk | NVMe disk |
| /0/100/1d/0/2 | /dev/ng0n1 | disk | NVMe disk |
| /0/100/1d/0/1 | /dev/nvme0n1 | disk | 512GB NVMe disk |
| /0/100/1d/0/1/1 | /dev/nvme0n1p1 | volume | 259MiB Windows FAT volume |
| /0/100/1d/0/1/2 | /dev/nvme0n1p2 | volume | 15MiB reserved partition |
| /0/100/1d/0/1/3 | /dev/nvme0n1p3 | volume | 162GiB Windows NTFS volume |
| /0/100/1d/0/1/4 | /dev/nvme0n1p4 | volume | 849MiB Windows NTFS volume |
| /0/100/1d/0/1/5 | /dev/nvme0n1p5 | volume | 288GiB EXT4 volume |
| /0/100/1d/0/1/6 | /dev/nvme0n1p6 | volume | 24GiB Linux swap volume |
| /0/100/1e | | communication | Cannon Point-LP Serial IO |
| /0/100/1e.3 | | bus | Cannon Point-LP Serial IO |
| /0/100/1f | | bridge | Cannon Point-LP LPC Contro |
| /0/100/1f/0 | | system | PnP device PNP0c02 |
| /0/100/1f/1 | | system | PnP device PNP0c02 |
| /0/100/1f/2 | | generic | PnP device INT3f0d |

| | | | |
|---------------|---------|------------|----------------------------|
| /0/100/1f/3 | | system | PnP device PNP0c02 |
| /0/100/1f/4 | | input | PnP device PNP0303 |
| /0/100/1f/5 | | system | PnP device PNP0c02 |
| /0/100/1f/6 | | system | PnP device PNP0c02 |
| /0/100/1f/7 | | system | PnP device PNP0c02 |
| /0/100/1f/8 | | system | PnP device PNP0c02 |
| /0/100/1f.3 | card0 | multimedia | Cannon Point-LP High Defin |
| /0/100/1f.3/0 | input33 | input | HDA Intel PCH Headphone |
| /0/100/1f.3/1 | input34 | input | HDA Intel PCH HDMI/DP,pcm= |
| /0/100/1f.3/2 | input35 | input | HDA Intel PCH HDMI/DP,pcm= |
| /0/100/1f.3/3 | input36 | input | HDA Intel PCH HDMI/DP,pcm= |
| /0/100/1f.4 | | bus | Cannon Point-LP SMBus Cont |
| /0/100/1f.5 | | bus | Cannon Point-LP SPI Contro |
| /1 | | power | To Be Filled By O.E.M. |
| /2 | input0 | input | Lid Switch |
| /3 | input1 | input | Power Button |
| /4 | input11 | input | ELAN1300:00 04F3:310C Mous |
| /5 | input12 | input | ELAN1300:00 04F3:310C Touc |
| /6 | input13 | input | ELAN2097:00 04F3:241F |
| /7 | input14 | input | ELAN2097:00 04F3:241F UNKN |
| /8 | input15 | input | ELAN2097:00 04F3:241F UNKN |
| /9 | input16 | input | ELAN2097:00 04F3:241F Styl |
| /a | input2 | input | Power Button |
| /b | input28 | input | Intel HID events |

```
/c      input29      input      Intel HID 5 button array
/d      input3       input      AT Translated Set 2 keyboa
/e      input30      input      Intel HID switches
/f      input31      input      Asus WMI hotkeys
/10     input32      input      Video Bus
```

```
/home/igor/mshell/mscripts >
```

9. Mshell script example2b with comments from LLM (example2b)

```
#!/bin/mshell

# system_error_checker.ms - Commands to check for errors in Ubuntu

# Example 1: Show recent errors in kernel log
echo "Checking recent kernel errors (dmesg):"
ollama1 "No markdown. Write your comment what command is doing: \'sudo
dmesg --level=err,warn\' "
ollama2exec "No any comments. Run the Ubuntu Linux command: sudo dmesg -
-level=err,warn "

# Example 2: Check systemd journal errors
echo ""
echo "Checking systemd journal errors:"
ollama1 "No markdown. Write your comment what command is doing: sudo
journalctl -xn 10 "
ollama2exec "No any comments. Run the Ubuntu Linux command: sudo
journalctl -xn 10 "
```

```
# Example 3: Check authentication failures
```

```
echo ""
```

```
echo "Checking authentication failures:"
```

```
ollama1 "No markdown. Write your comment what command is doing: sudo grep  
'authentication failure' /var/log/auth.log "
```

```
ollama2exec "No any comments. Run the Ubuntu Linux command: sudo grep  
'authentication failure' /var/log/auth.log "
```

```
# Example 4: Check for failed services
```

```
echo ""
```

```
echo "Checking for failed systemd services:"
```

```
ollama1 "No markdown. Write your comment what command is doing: sudo  
systemctl --failed "
```

```
ollama2exec "No any comments. Run the Ubuntu Linux command: sudo  
systemctl --failed "
```

```
# Example 5: Check for kernel and disk errors
```

```
echo ""
```

```
echo "Checking for kernel and disk errors:"
```

```
ollama1 "No markdown. Write your comment what command is doing: sudo grep  
-i error /var/log/kern.log "
```

```
ollama2exec "No any comments. Run the Ubuntu Linux command: sudo grep -i  
error /var/log/kern.log "
```

```
# Example 6: Check for SSH login failures
```

```
echo ""
```

```
echo "Checking for SSH login failures:"
```

```
ollama1 "No markdown. Write your comment what command is doing: sudo grep
```

```
'Failed password' /var/log/auth.log "
```

```
ollama2exec "No any comments. Run the Ubuntu Linux command: sudo grep
```

```
'Failed password' /var/log/auth.log "
```

```
# Example 7: Check application crashes
```

```
echo ""
```

```
echo "Checking for application crashes:"
```

```
ollama1 "No markdown. Write your comment what command is doing: sudo ls -lt
```

```
/var/crash "
```

```
ollama2exec "No any comments. Run the Ubuntu Linux command: sudo ls -lt
```

```
/var/crash "
```

```
# Example 8: Check boot logs for errors
```

```
# echo ""
```

```
# echo "Checking boot logs for errors:"
```

```
# ollama1 "No markdown. Write your comment what command is doing: sudo
```

```
dmesg | grep -i error "
```

```
# ollama2exec "No any comments. Run the Ubuntu Linux command: sudo
```

```
dmesg | grep -i error "
```

```
# Example 9: Check apt package manager errors

# echo ""

# echo "Checking apt package manager errors:"

# ollama1 "No markdown. Write your comment what command is doing: sudo
grep -i error /var/log/apt/term.log "

# ollama2exec "No any comments. Run the Ubuntu Linux command: sudo grep -i
error /var/log/apt/term.log "

# Example 10: Check for hardware failures

echo ""

echo "Checking hardware diagnostics:"

ollama1 "No markdown. Write your comment what command is doing: sudo lshw
-short "

ollama2exec "No any comments. Run the Ubuntu Linux command: sudo lshw -
short "
```

10. Result of script execution for example2b

```
/home/igor/mshell/mscripts > ./test1.ms

Loading .mshellrc...

Loaded environment: PATH=/usr/bin:/bin:/usr/sbin:/usr/local/bin:.$PATH

Loaded alias: ll='ls -l'

Loaded alias: home='cd ~'

Loaded environment: EDITOR=nano
```

Lua state initialized successfully

Range function defined successfully

Loading history...

Checking recent kernel errors (dmesg):

```
DEBUG: Variable assignment value = 'err,warn' ""
```

The command ``sudo dmesg --level=err,warn`` uses the ``dmesg`` utility to display or control the kernel ring buffer, which logs a wide range of messages generated by the kernel and device drivers. Here's what each part of the command does:

- ``sudo``: This stands for "superuser do" and allows you to run the following command with administrative privileges.

- ``dmesg``: This is the command used to display or control the kernel ring buffer.

- ``--level=err,warn``: This option specifies the levels of messages that should be displayed. In this case, it filters the output to show only error (``err``) and warning (``warn``) messages. Other messages will not be shown.

When you run this command, it will display all the error and warning messages recorded by the kernel since the last boot. This can be useful for diagnosing problems or understanding the health of your system's hardware and drivers.

```
DEBUG: Variable assignment value = 'err,warn ""
```

```
...
```

```
[ 0.000000] Initializing cgroup subsys cpu
```

```
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] LVM version 2 (01/00)
[ 0.000000] loaded
[ 0.000000] platform b3c10294-d5f7-41a7-a4db-e77d8e9d44ad on
0000:00:1f.6
[ 0.000000] arch x86_64
[ 0.000000] features 0x25a9fb<sse4,ssbd,fma3,tsc,abm,pni,adx>
[ 0.000000] flags 0x000001:0 (no msr)
[ 0.000000] vendor_intel_idle=1
[ 0.000000] vendor_x86_max_usable_addr=7fffffffffffffff
[ 0.000000] smpboot-cpu 0x4f: Intel(R) Core i9-11900H CPU @ 2.30GHz,
order: 5, acting as physical ID: 1, online.
[ 0.000000] smpboot-cpu 1x: Not Found
[ 0.000000] smpboot-cpu 2x: Not Found
[ 0.000000] smpboot-cpu 3x: Not Found
[ 0.000000] Total of 1 processors activated (in CPUinit).
[ 0.011144] Per-CPU System RAM initialization
[ 0.011150] zswap: loaded using pool-sizepool-size pool-size.
[ 0.011164] pcieport 0000:00:1c.4: Signaling PM suspend event to legacy irqs
without core irqs!
[ 0.029155] audit: initializing netlink subsys (disabled)
[ 0.030158] SELinux: enabled
[ 14.135456] systemd-udevd[533]: starting version 2.276
[ 20.156876] systemd-logind[534]: created logind socket /run/systemd/logind
```

```
[ 20.157059] systemd-udevd[535]: started.  
[ 164.141763] Console: switching to colour false on colour auto  
...
```

Executing command block via mshell:

DEBUG: Variable assignment value = 'err,warn'

[sudo] password for igor:

[0.194384] x86/cpu: SGX disabled by BIOS.

[0.204376] MDS CPU bug present and SMT on, data leak possible. See <https://www.kernel.org/doc/html/latest/admin-guide/hw-vuln/mds.html> for more details.

[0.204384] MMIO Stale Data CPU bug present and SMT on, data leak possible. See https://www.kernel.org/doc/html/latest/admin-guide/hw-vuln/processor_mmio_stale_data.html for more details.

[0.783212] hpet_acpi_add: no address or irqs in _CRS

[0.800576] i8042: PNP: PS/2 appears to have AUX port disabled, if this is incorrect please boot with i8042.nopnp

[0.806259] device-mapper: core: CONFIG_IMA_DISABLE_HTABLE is disabled. Duplicate IMA measurements will not be recorded in the IMA log.

[0.806377] platform eisa.0: EISA: Cannot allocate resource for mainboard

[0.806379] platform eisa.0: Cannot allocate resource for EISA slot 1

[0.806381] platform eisa.0: Cannot allocate resource for EISA slot 2

[0.806383] platform eisa.0: Cannot allocate resource for EISA slot 3

[0.806385] platform eisa.0: Cannot allocate resource for EISA slot 4

[0.806386] platform eisa.0: Cannot allocate resource for EISA slot 5

[0.806388] platform eisa.0: Cannot allocate resource for EISA slot 6

[0.806390] platform eisa.0: Cannot allocate resource for EISA slot 7

[0.806391] platform eisa.0: Cannot allocate resource for EISA slot 8

[0.873930] ENERGY_PERF_BIAS: Set to 'normal', was 'performance'

[0.912473] blacklist: Duplicate blacklisted hash
bin:47ff1b63b140b6fc04ed79131331e651da5b2e2f170f5daef4153dc2fbc532b1

[0.912589] blacklist: Duplicate blacklisted hash
bin:992d359aa7a5f789d268b94c11b9485a6b1ce64362b0edb4441ccc187c39647b

[0.912665] blacklist: Duplicate blacklisted hash
bin:c452ab846073df5ace25cca64d6b7a09d906308a1a65eb5240e3c4ebcaa9cc0c

[0.912710] blacklist: Duplicate blacklisted hash
bin:e051b788ecbaeda53046c70e6af6058f95222c046157b8c4c1b9c2cfc65f46e5

[1.333477] i2c i2c-0: Failed to register i2c client XXXX0000:00 at 0x15 (-16)

[1.334887] i2c i2c-0: Failed to register i2c client XXXX0000:02 at 0x15 (-16)

[1.341597] usb: port power management may be unreliable

[1.535986] i2c_hid_acpi i2c-ELAN2097:00: i2c_hid_get_input: IRQ triggered but
there's no data

[1.634231] i2c_hid_acpi i2c-ELAN1300:00: device returned incorrect report (2 vs
14 expected)

[1.634238] hid-multitouch 0018:04F3:310C.0001: failed to fetch feature 14

[2.865566] sd 16:0:0:0: [sda] Optimal transfer size 33553920 bytes not a
multiple of preferred minimum block size (4096 bytes)

[2.877791] sd 17:0:0:0: [sdb] Optimal transfer size 33553920 bytes not a
multiple of preferred minimum block size (4096 bytes)

[6.492015] iwlfwifi 0000:00:14.3: WRT: Overriding region id 0
[6.492026] iwlfwifi 0000:00:14.3: WRT: Overriding region id 1
[6.492032] iwlfwifi 0000:00:14.3: WRT: Overriding region id 2
[6.492036] iwlfwifi 0000:00:14.3: WRT: Overriding region id 3
[6.492040] iwlfwifi 0000:00:14.3: WRT: Overriding region id 4
[6.492044] iwlfwifi 0000:00:14.3: WRT: Overriding region id 6
[6.492048] iwlfwifi 0000:00:14.3: WRT: Overriding region id 8
[6.492052] iwlfwifi 0000:00:14.3: WRT: Overriding region id 9
[6.492056] iwlfwifi 0000:00:14.3: WRT: Overriding region id 10
[6.492061] iwlfwifi 0000:00:14.3: WRT: Overriding region id 11
[6.492065] iwlfwifi 0000:00:14.3: WRT: Overriding region id 15
[6.492069] iwlfwifi 0000:00:14.3: WRT: Overriding region id 16
[6.492073] iwlfwifi 0000:00:14.3: WRT: Overriding region id 18
[6.492077] iwlfwifi 0000:00:14.3: WRT: Overriding region id 19
[6.492082] iwlfwifi 0000:00:14.3: WRT: Overriding region id 20
[6.492087] iwlfwifi 0000:00:14.3: WRT: Overriding region id 21
[6.492091] iwlfwifi 0000:00:14.3: WRT: Overriding region id 28
[6.567882] asus_wmi: fan_curve_get_factory_default (0x00110024) failed: -61
[6.570043] asus_wmi: fan_curve_get_factory_default (0x00110025) failed: -61
[6.572161] asus_wmi: fan_curve_get_factory_default (0x00110032) failed: -61
[6.662788] Bluetooth: hci0: HCI LE Coded PHY feature bit is set, but its usage is not supported.
[7.554461] thermal thermal_zone5: failed to read out thermal zone (-61)
[8.096083] thermal thermal_zone5: failed to read out thermal zone (-61)

[10.280790] block nvme0n1: No UUID available providing old NGUID

[13.372694] kauditd_printk_skb: 256 callbacks suppressed

[416.030339] workqueue: i915_hpd_poll_init_work [i915] hogged CPU for >13333us 4 times, consider switching to WQ_UNBOUND

[471.970064] workqueue: output_poll_execute hogged CPU for >13333us 4 times, consider switching to WQ_UNBOUND

[532.250197] workqueue: i915_hpd_poll_init_work [i915] hogged CPU for >13333us 8 times, consider switching to WQ_UNBOUND

[590.760979] workqueue: output_poll_execute hogged CPU for >13333us 8 times, consider switching to WQ_UNBOUND

[772.390018] workqueue: i915_hpd_poll_init_work [i915] hogged CPU for >13333us 16 times, consider switching to WQ_UNBOUND

[831.916253] workqueue: output_poll_execute hogged CPU for >13333us 16 times, consider switching to WQ_UNBOUND

[1484.665405] thermal thermal_zone5: failed to read out thermal zone (-61)

[1485.307409] workqueue: i915_hpd_poll_init_work [i915] hogged CPU for >13333us 32 times, consider switching to WQ_UNBOUND

[1512.426692] exFAT-fs (sdb2): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.

[1513.622094] exFAT-fs (sda2): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.

[1757.842845] warning: `ThreadPoolForeg' uses wireless extensions which will stop working for Wi-Fi 7 hardware; use nl80211

[5915.245843] kauditd_printk_skb: 13 callbacks suppressed

[68826.273477] i915 0000:00:02.0: [drm] *ERROR* CPU pipe A FIFO underrun

[178714.591682] kauditd_printk_skb: 25 callbacks suppressed

Checking systemd journal errors:

The command `sudo journalctl -xn 10` uses the `journalctl` utility to display entries from the system log (the systemd journal). Here's a breakdown of what each part of the command does:

- `sudo`: This stands for "superuser do" and allows you to run the following command with administrative privileges.

- `journalctl`: This is the command used to query and manage the systemd journal, which logs all system messages, including kernel messages and service log entries.

- `-x`: This option tells `journalctl` to augment the log entries with contextual information that can help in understanding the context of the message. This typically includes suggestions for the severity level of the message based on its content.

- `-n 10`: This option limits the output to the last 10 log entries. If you omit this, `journalctl` will show all available log entries.

When you run this command, it will display the last 10 log entries from the system journal, enhancing them with contextual information that can help in interpreting their significance. This is useful for quickly checking recent events or diagnosing issues based on system logs.

To run the `sudo journalctl -xn 10` command on an Ubuntu Linux system, you need to have appropriate permissions and ensure that the `systemd-journald` service is running. Here's how you can do it:

1. **Open a Terminal**: You can open a terminal by pressing `Ctrl + Alt + T`.
2. **Run the Command**:

```
``bash
sudo journalctl -xn 10
``
```

This command will display the last 10 entries from the system journal, showing the most recent messages along with their priorities and timestamps.

Explanation of the Command:

- `sudo`: This runs the command with superuser privileges.
- `journalctl`: This is the command-line interface for querying the `systemd` journal.
- `-x`: This option adds an extended explanation to each log entry, which can help in understanding why certain events occurred.
- `-n 10`: This limits the output to the last 10 entries.

Example Output:

...

-- Logs begin at Mon 2023-04-01 10:00:00 UTC, end at Tue 2023-04-04 15:30:00 UTC. --

Apr 04 15:28:45 myhost systemd[1]: Starting APT daily upgrade task...

-- Subject: A start job is running

-- Defined-By: systemd

-- Support: <https://help.ubuntu.com/systemd.html>

--

A start job is running for APT daily upgrade task (3min 47s / no limit)

Apr 04 15:29:42 myhost systemd[1]: Finished APT daily upgrade task.

-- Subject: A stop job is running

-- Defined-By: systemd

-- Support: <https://help.ubuntu.com/systemd.html>

--

A stop job is running for APT daily upgrade task (3min 57s / no limit)

...

This output shows the last 10 log entries, including their timestamps and descriptions.

If you encounter any issues or need further assistance, feel free to ask!

Executing command block via mshell:

May 11 10:01:28 asus x11vnc[2082]: 11/05/2025 10:01:28 Client
68.190.125.86 gone

May 11 10:01:28 asus x11vnc[2082]: 11/05/2025 10:01:28 Statistics
e>

May 11 10:01:28 asus x11vnc[2082]: 11/05/2025 10:01:28 TOTALS :
>

May 11 10:01:28 asus x11vnc[2082]: 11/05/2025 10:01:28 Statistics
e>

May 11 10:01:28 asus x11vnc[2082]: 11/05/2025 10:01:28 TOTALS :
>

May 11 10:05:01 asus CRON[804329]: pam_unix(cron:session): session
opened for u>

May 11 10:05:01 asus CRON[804330]: (root) CMD (command -v debian-sa1
> /dev/nul>

May 11 10:05:01 asus CRON[804329]: pam_unix(cron:session): session
closed for u>

May 11 10:06:40 asus sudo[804336]: igor : TTY=pts/4 ;
PWD=/home/igor/mshell>

May 11 10:06:40 asus sudo[804336]: pam_unix(sudo:session): session
opened for u>

Checking authentication failures:

The command `sudo grep 'authentication failure' /var/log/auth.log` uses the `grep` utility to search for specific text within a file. Here's a breakdown of what each part of the command does:

- `sudo`: This stands for "superuser do" and allows you to run the following command with administrative privileges.

- `grep`: This is the command used to search for patterns in files. It scans through one or more files, printing lines that contain a match to the specified pattern.

- `'authentication failure'`: This is the pattern that `grep` will search for within the file. In this case, it looks for any line that contains the exact phrase "authentication failure".

- `/var/log/auth.log`: This is the file that `grep` will search through. The `/var/log/auth.log` file typically contains authentication-related messages from the system.

When you run this command, it will display all lines from the `/var/log/auth.log` file that contain the phrase "authentication failure". This can be useful for monitoring and investigating failed login attempts or other authentication issues on a system.

To run the ``sudo grep 'authentication failure' /var/log/auth.log`` command on an Ubuntu Linux system, you need to have appropriate permissions and ensure that the ``/var/log/auth.log`` file exists. Here\u2019s how you can do it:

1. **Open a Terminal**: You can open a terminal by pressing ``Ctrl + Alt + T``.

2. **Run the Command**:

```
``bash
sudo grep 'authentication failure' /var/log/auth.log
``
```

This command will search for lines containing the phrase "authentication failure" in the ``/var/log/auth.log`` file and display them.

Explanation of the Command:

- ``sudo``: This runs the command with superuser privileges.
- ``grep``: This is a command-line utility for searching plain-text data sets for lines that match a regular expression.
- ``'authentication failure'``: This is the search pattern you are looking for in the log file.
- ``/var/log/auth.log``: This is the path to the authentication log file.

Example Output:

```
``
```

```
Apr 04 15:28:45 myhost sshd[1234]: Failed password for user foo from
192.168.1.1 port 22 ssh2
```

```
Apr 04 15:30:00 myhost sshd[5678]: Failed password for root from 192.168.1.2
port 22 ssh2
```

```
...
```

This output shows lines from the authentication log that contain the phrase "authentication failure", indicating failed login attempts.

If you encounter any issues or need further assistance, feel free to ask!

Executing command block via mshell:

```
[sudo] password for igor:
```

```
2025-05-11T03:49:10.724222-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep
```

```
'authentication failure' /var/log/auth.log
```

```
2025-05-11T05:09:52.840641-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep
```

```
'authentication failure' /var/log/auth.log
```

```
2025-05-11T06:09:08.117927-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep
```

```
'authentication failure' /var/log/auth.log
```

```
2025-05-11T09:08:00.036450-05:00 asus sudo: igor : TTY=pts/4 ;  
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep  
'authentication failure' /var/log/auth.log
```

```
2025-05-11T10:25:18.423580-05:00 asus sudo: igor : TTY=pts/4 ;  
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep  
'authentication failure' /var/log/auth.log
```

Checking for failed systemd services:

The command `sudo systemctl --failed` uses the `systemctl` utility to list all services that have failed since the last boot. Here's a breakdown of what each part of the command does:

- `sudo`: This stands for "superuser do" and allows you to run the following command with administrative privileges.

- `systemctl`: This is the command used to control the systemd system and service manager, which manages all services on most modern Linux distributions.

- `--failed`: This option tells `systemctl` to list only those units (services) that have failed. A unit is considered failed if it has entered a failed state, which typically indicates an issue with the service.

When you run this command, it will display a list of all services and units that have failed since the last boot. This can be useful for quickly identifying and troubleshooting problems with specific services on your system.

To run the `sudo systemctl --failed` command on an Ubuntu Linux system, you need to have appropriate permissions and ensure that the `systemd` service is running. Here is how you can do it:

1. **Open a Terminal**: You can open a terminal by pressing `Ctrl + Alt + T`.

2. **Run the Command**:

```
``bash
sudo systemctl --failed
...`
```

This command will list all services that are currently in a failed state.

Explanation of the Command:

- `sudo`: This runs the command with superuser privileges.

- `systemctl`: This is the command-line interface for controlling the `systemd` system and service manager.

- `--failed`: This option filters the list to show only services that are in a failed state.

Example Output:

...

| UNIT | LOAD | ACTIVE | SUBSTATE | DESCRIPTION |
|-----------------|------|--------|---------------|-------------------------------------|
| docker.service | | loaded | failed failed | Docker Application Container Engine |
| apache2.service | | loaded | failed failed | The Apache HTTP Server |

...

This output shows services that are currently failing. Each line includes the unit name, load status, active state, substate, and a brief description.

If you encounter any issues or need further assistance, feel free to ask!

Executing command block via mshell:

```
UNIT LOAD ACTIVE SUB DESCRIPTION
```

0 loaded units listed.

Checking for kernel and disk errors:

The command `sudo grep -i error /var/log/kern.log` uses the `grep` utility to search for specific text within a file, with case insensitivity. Here's a breakdown of what each part of the command does:

- ``sudo``: This stands for "superuser do" and allows you to run the following command with administrative privileges.

- ``grep``: This is the command used to search for patterns in files. It scans through one or more files, printing lines that contain a match to the specified pattern.

- ``-i``: This option tells ``grep`` to ignore case distinctions when matching the pattern. In other words, it will match "error", "Error", "ERROR", etc.

- ``error``: This is the pattern that ``grep`` will search for within the file. It looks for any line that contains the word "error" regardless of its case.

- ``/var/log/kern.log``: This is the file that ``grep`` will search through. The ``/var/log/kern.log`` file typically contains kernel-related messages and error logs from the system's kernel.

When you run this command, it will display all lines from the ``/var/log/kern.log`` file that contain the word "error" in any case. This can be useful for monitoring kernel events or diagnosing issues related to the system's kernel.

To run the ``sudo grep -i error /var/log/kern.log`` command on an Ubuntu Linux system, you need to have appropriate permissions and ensure that the ``/var/log/kern.log`` file exists. Here's how you can do it:

1. ****Open a Terminal****: You can open a terminal by pressing ``Ctrl + Alt + T``.
2. ****Run the Command****:

```
``bash  
  
sudo grep -i error /var/log/kern.log  
  
...
```

This command will search for lines containing the word "error" (case-insensitive) in the ``/var/log/kern.log`` file and display them.

Explanation of the Command:

- ``sudo``: This runs the command with superuser privileges.
- ``grep``: This is a command-line utility for searching plain-text data sets for lines that match a regular expression.
- ``-i``: This option makes the search case-insensitive, so it will match "error", "Error", "ERROR", etc.
- ``/var/log/kern.log``: This is the path to the kernel log file.

Example Output:

```
...  
  
Apr 04 15:28:45 myhost kernel: [ 0.000000] Initializing cgroup subsys cpuset  
Apr 04 15:29:00 myhost kernel: [ 0.000000] Initializing cgroup subsys cpu  
Apr 04 15:30:15 myhost kernel: [ 0.000000] ACPI: bus type PCI registered  
  
...
```

This output shows lines from the kernel log that contain the word "error". Note that in this specific example, no actual error messages are present because the sample log entries do not contain the word "error".

If you encounter any issues or need further assistance, feel free to ask!

Executing command block via mshell:

```
2025-05-11T00:47:46.749348-05:00 asus kernel: gedit[779884]: segfault at f1
ip 00007d142b55fdaf sp 00007ffc6d5bd7b0 error 4 in libgtk-
3.so.0.2409.32[7d142b283000+3ae000] likely on CPU 7 (core 3, socket 0)
```

Checking for SSH login failures:

The command ``sudo grep 'Failed password' /var/log/auth.log`` uses the ``grep`` utility to search for specific text within a file. Here's a breakdown of what each part of the command does:

- ``sudo``: This stands for "superuser do" and allows you to run the following command with administrative privileges.
- ``grep``: This is the command used to search for patterns in files. It scans through one or more files, printing lines that contain a match to the specified pattern.
- ``Failed password``: This is the pattern that ``grep`` will search for within the file. It looks for any line that contains the exact phrase "Failed password".

- `/var/log/auth.log`: This is the file that `grep` will search through. The `/var/log/auth.log` file typically contains authentication-related messages from the system.

When you run this command, it will display all lines from the `/var/log/auth.log` file that contain the phrase "Failed password". This can be useful for monitoring and investigating failed login attempts or other authentication issues on a system. To run the `sudo grep 'Failed password' /var/log/auth.log` command on an Ubuntu Linux system, you need to have appropriate permissions and ensure that the `/var/log/auth.log` file exists. Here's how you can do it:

1. **Open a Terminal**: You can open a terminal by pressing `Ctrl + Alt + T`.
2. **Run the Command**:

```
``bash
sudo grep 'Failed password' /var/log/auth.log
``
```

This command will search for lines containing the phrase "Failed password" in the `/var/log/auth.log` file and display them.

Explanation of the Command:

- `sudo`: This runs the command with superuser privileges.
- `grep`: This is a command-line utility for searching plain-text data sets for lines that match a regular expression.
- `'Failed password'`: This is the search pattern you are looking for in the log file.
- `/var/log/auth.log`: This is the path to the authentication log file.

Example Output:

...

```
Apr 04 15:28:45 myhost sshd[1234]: Failed password for user foo from  
192.168.1.1 port 22 ssh2
```

```
Apr 04 15:30:00 myhost sshd[5678]: Failed password for root from 192.168.1.2  
port 22 ssh2
```

...

This output shows lines from the authentication log that contain the phrase "Failed password", indicating failed login attempts.

If you encounter any issues or need further assistance, feel free to ask!

Executing command block via mshell:

```
2025-05-11T03:51:34.355729-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep
```

```
'Failed password' /var/log/auth.log
```

```
2025-05-11T05:10:28.271485-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep
```

```
'Failed password' /var/log/auth.log
```

```
2025-05-11T06:11:59.932240-05:00 asus sudo: igor : TTY=pts/4 ;
```

```
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep
```

```
'Failed password' /var/log/auth.log
```

```
2025-05-11T09:10:35.534679-05:00 asus sudo: igor : TTY=pts/4 ;  
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep  
'Failed password' /var/log/auth.log  
2025-05-11T10:44:30.648153-05:00 asus sudo: igor : TTY=pts/4 ;  
PWD=/home/igor/mshell/mscripts ; USER=root ; COMMAND=/usr/bin/grep  
'Failed password' /var/log/auth.log
```

Checking for application crashes:

The command `sudo ls -lt /var/crash` uses the `ls` utility to list files and directories in a specified directory, sorted by time. Here's a breakdown of what each part of the command does:

- `sudo`: This stands for "superuser do" and allows you to run the following command with administrative privileges.
- `ls`: This is the command used to list files and directories in the current directory or a specified directory.
- `-l`: This option tells `ls` to use a long listing format, which provides detailed information about each file or directory, including permissions, number of links, owner, group, size, time of last modification, and the name of the file or directory.
- `-t`: This option sorts the list by modification time, with the newest files or directories first.
- `/var/crash`: This is the directory that `ls` will list. The `/var/crash` directory typically contains crash dumps from the system, which are used for debugging purposes when a system crashes.

When you run this command, it will display a long listing of all files and directories in the `/var/crash` directory, sorted by the time they were last modified, with the most recently modified files first. This can be useful for identifying recent crash dumps or other important files in that directory.

To run the `sudo ls -lt /var/crash` command on an Ubuntu Linux system, you need to have appropriate permissions and ensure that the `/var/crash` directory exists. Here's how you can do it:

1. **Open a Terminal**: You can open a terminal by pressing `Ctrl + Alt + T`.
2. **Run the Command**:

```
``bash
sudo ls -lt /var/crash
...

```

This command will list the contents of the `/var/crash` directory in long format, sorted by modification time from newest to oldest.

Explanation of the Command:

- `sudo`: This runs the command with superuser privileges.
- `ls`: This is a command-line utility for listing directory contents.
- `-l`: This option provides a long listing format, which includes details such as permissions, number of links, owner, group, size, and time of last modification.
- `-t`: This option sorts the files by modification time, with the newest first.
- `/var/crash`: This is the path to the directory you want to list.

Example Output:

```
...  
drwx----- 2 root root    4096 Apr  1 15:30 user-1  
drwx----- 2 root root    4096 Mar 31 14:20 user-2  
-rw----- 1 root root   8192 Feb 28 12:10 ubuntu-core-20.04.tar.gz  
...
```

This output shows the contents of the ``/var/crash`` directory, including subdirectories and files, sorted by their last modification time.

If you encounter any issues or need further assistance, feel free to ask!

Executing command block via mshell:

```
total 9784
```

```
-rw-r----- 1 igor whoopsie 10016677 May  7 19:22 _usr_bin_gedit.1001.crash
```

Checking hardware diagnostics:

The command ``sudo lshw -short`` uses the ``lshw`` utility to list hardware information about a system, but with reduced detail. Here's a breakdown of what each part of the command does:

- ``sudo``: This stands for "superuser do" and allows you to run the following command with administrative privileges.

- ``lshw``: This is the command used to list Hardware Information (hardware inventory). It provides detailed information about the hardware components on a system, including CPU, memory, storage devices, network interfaces, and more.

- `-short`: This option tells `lshw` to provide a short output. Instead of displaying all available details for each hardware component, it limits the output to essential information.

When you run this command, it will display a concise list of hardware components on your system, including their type, description, and some basic configuration details. This can be useful for quickly getting an overview of your system's hardware without overwhelming yourself with detailed information.

To run the `sudo lshw -short` command on an Ubuntu Linux system, you need to have appropriate permissions and ensure that the `lshw` utility is installed.

Here's how you can do it:

1. **Open a Terminal**: You can open a terminal by pressing `Ctrl + Alt + T`.
2. **Run the Command**:

```
``bash
sudo lshw -short
...`
```

This command will display a concise overview of the hardware configuration of your system.

Explanation of the Command:

- `sudo`: This runs the command with superuser privileges.

- `lshw`: This is a command-line utility for displaying information about the hardware configuration of the computer.

- ``-short``: This option provides a brief summary of the hardware, focusing on key components without detailed specifications.

Example Output:

...

*-core

description: Motherboard

product: H110M-S2C

vendor: ASRock

physical id: 0

version: 3.0

serial: to_be_filled_by_oem

slot: To Be Filled By O.E.M.

*-cpu

description: CPU

product: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz

vendor: Intel Corp.

physical id: 4

bus info: cpu@0

version: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz

size: 1796MHz

capacity: 3100MHz

width: 64 bits

clock: 100MHz

*-cache:0

description: L1 cache

physical id: 0

size: 32KiB

capacity: 32KiB

clock: 1GHz

*-cache:1

description: L2 cache

physical id: 1

size: 256KiB

capacity: 256KiB

clock: 1GHz

*-cache:2

description: L3 cache

physical id: 2

size: 4MiB

capacity: 4MiB

clock: 1GHz

*-memory

description: System memory

physical id: 0

size: 8GiB

*-bank:0

description: SODIMM DDR3 Synchronous 1600 MHz (0.6 ns)

product: KHX1600C9D4/8G

vendor: Kingston Technology

physical id: 0

serial: 514568E7

slot: DIMM0

size: 4GiB

width: 64 bits

clock: 1600MHz

*-bank:1

description: SODIMM DDR3 Synchronous 1600 MHz (0.6 ns)

product: KHX1600C9D4/8G

vendor: Kingston Technology

physical id: 1

serial: 514568E7

slot: DIMM1

size: 4GiB

width: 64 bits

clock: 1600MHz

*-pci

description: Host bridge

product: Intel(R) Haswell-E DRAM Controller

vendor: Intel Corp.

physical id: 0

bus info: pci@0000:00:00.0

```
version: 05
width: 32 bits
clock: 33MHz
...

```

This output provides a high-level summary of the hardware components in your system, including CPU, memory, and other key peripherals.

If you encounter any issues or need further assistance, feel free to ask!

Executing command block via mshell:

| H/W path | Device | Class | Description |
|----------|----------|-----------|----------------------------|
| | | system | VivoBook_ASUSLaptop TP412F |
| /0 | | bus | TP412FA |
| /0/0 | | memory | 64KiB BIOS |
| /0/3a | | memory | 20GiB System Memory |
| /0/3a/0 | | memory | 4GiB SODIMM DDR4 Synchrono |
| /0/3a/1 | | memory | 16GiB SODIMM DDR4 Synchron |
| /0/45 | | memory | 256KiB L1 cache |
| /0/46 | | memory | 1MiB L2 cache |
| /0/47 | | memory | 6MiB L3 cache |
| /0/48 | | processor | Intel(R) Core(TM) i5-8265U |
| /0/100 | | bridge | Coffee Lake HOST and DRAM |
| /0/100/2 | /dev/fb0 | display | WhiskeyLake-U GT2 [UHD Gra |
| /0/100/4 | | generic | Xeon E3-1200 v5/E3-1500 v5 |

| | | | |
|-----------------------|-----------|---------------|------------------------------|
| /0/100/8 | | generic | Xeon E3-1200 v5/v6 / E3-15 |
| /0/100/12 | | generic | Cannon Point-LP Thermal Co |
| /0/100/13 | | communication | Cannon Point-LP Integrated |
| /0/100/14 | | bus | Cannon Point-LP USB 3.1 xH |
| /0/100/14/0 | usb1 | bus | xHCI Host Controller |
| /0/100/14/0/1 | scsi16 | storage | One Touch HDD |
| /0/100/14/0/1/0.0.0 | /dev/sda | disk | 5TB One Touch HDD |
| /0/100/14/0/1/0.0.0/1 | /dev/sda1 | volume | 199MiB Windows FAT
volume |
| /0/100/14/0/1/0.0.0/2 | /dev/sda2 | volume | 4657GiB data partition |
| /0/100/14/0/2 | scsi17 | storage | BUP Slim |
| /0/100/14/0/2/0.0.0 | /dev/sdb | disk | 2TB BUP Slim |
| /0/100/14/0/2/0.0.0/1 | /dev/sdb1 | volume | 199MiB Windows FAT
volume |
| /0/100/14/0/2/0.0.0/2 | /dev/sdb2 | volume | 1862GiB data partition |
| /0/100/14/0/3 | | input | USB Receiver |
| /0/100/14/0/3/0 | input27 | input | Logitech M325 |
| /0/100/14/0/4 | | multimedia | USB2.0 VGA UVC WebCam |
| /0/100/14/0/a | | communication | Bluetooth 9460/9560 Jeffer |
| /0/100/14/1 | usb2 | bus | xHCI Host Controller |
| /0/100/14.2 | | memory | RAM memory |
| /0/100/14.3 | wlo1 | network | Cannon Point-LP CNVi [Wire |
| /0/100/15 | | bus | Cannon Point-LP Serial IO |
| /0/100/15.1 | | bus | Cannon Point-LP Serial IO |

| | | | |
|-----------------|----------------|---------------|-------------------------------|
| /0/100/16 | | communication | Cannon Point-LP MEI Contro |
| /0/100/17 | | storage | 82801 Mobile SATA Controll |
| /0/100/1d | | bridge | Cannon Point-LP PCI Expres |
| /0/100/1d/0 | /dev/nvme0 | storage | INTEL SSDPEKNW512G8 |
| /0/100/1d/0/0 | hwmon2 | disk | NVMe disk |
| /0/100/1d/0/2 | /dev/ng0n1 | disk | NVMe disk |
| /0/100/1d/0/1 | /dev/nvme0n1 | disk | 512GB NVMe disk |
| /0/100/1d/0/1/1 | /dev/nvme0n1p1 | volume | 259MiB Windows FAT
volume |
| /0/100/1d/0/1/2 | /dev/nvme0n1p2 | volume | 15MiB reserved partition |
| /0/100/1d/0/1/3 | /dev/nvme0n1p3 | volume | 162GiB Windows NTFS
volume |
| /0/100/1d/0/1/4 | /dev/nvme0n1p4 | volume | 849MiB Windows NTFS
volume |
| /0/100/1d/0/1/5 | /dev/nvme0n1p5 | volume | 288GiB EXT4 volume |
| /0/100/1d/0/1/6 | /dev/nvme0n1p6 | volume | 24GiB Linux swap volume |
| /0/100/1e | | communication | Cannon Point-LP Serial IO |
| /0/100/1e.3 | | bus | Cannon Point-LP Serial IO |
| /0/100/1f | | bridge | Cannon Point-LP LPC Contro |
| /0/100/1f/0 | | system | PnP device PNP0c02 |
| /0/100/1f/1 | | system | PnP device PNP0c02 |
| /0/100/1f/2 | | generic | PnP device INT3f0d |
| /0/100/1f/3 | | system | PnP device PNP0c02 |
| /0/100/1f/4 | | input | PnP device PNP0303 |

| | | | |
|---------------|---------|------------|----------------------------|
| /0/100/1f/5 | | system | PnP device PNP0c02 |
| /0/100/1f/6 | | system | PnP device PNP0c02 |
| /0/100/1f/7 | | system | PnP device PNP0c02 |
| /0/100/1f/8 | | system | PnP device PNP0c02 |
| /0/100/1f.3 | card0 | multimedia | Cannon Point-LP High Defin |
| /0/100/1f.3/0 | input33 | input | HDA Intel PCH Headphone |
| /0/100/1f.3/1 | input34 | input | HDA Intel PCH HDMI/DP,pcm= |
| /0/100/1f.3/2 | input35 | input | HDA Intel PCH HDMI/DP,pcm= |
| /0/100/1f.3/3 | input36 | input | HDA Intel PCH HDMI/DP,pcm= |
| /0/100/1f.4 | | bus | Cannon Point-LP SMBus Cont |
| /0/100/1f.5 | | bus | Cannon Point-LP SPI Contro |
| /1 | | power | To Be Filled By O.E.M. |
| /2 | input0 | input | Lid Switch |
| /3 | input1 | input | Power Button |
| /4 | input11 | input | ELAN1300:00 04F3:310C Mous |
| /5 | input12 | input | ELAN1300:00 04F3:310C Touc |
| /6 | input13 | input | ELAN2097:00 04F3:241F |
| /7 | input14 | input | ELAN2097:00 04F3:241F UNKN |
| /8 | input15 | input | ELAN2097:00 04F3:241F UNKN |
| /9 | input16 | input | ELAN2097:00 04F3:241F Styl |
| /a | input2 | input | Power Button |
| /b | input28 | input | Intel HID events |
| /c | input29 | input | Intel HID 5 button array |
| /d | input3 | input | AT Translated Set 2 keyboa |

```
/e          input30    input    Intel HID switches
/f          input31    input    Asus WMI hotkeys
/10        input32    input    Video Bus
```

```
/home/igor/mshell/mscripts >
```

11. Example 3 of executing a single commands to determine the basic parameters of computer.

```
/home/igor/mshell/mscripts > ollama2 "What linux ubuntu command inxi is doing?"
```

`inxi` is a powerful and versatile Linux terminal command used to gather detailed information about your system's hardware, software, and environment.

When you run `inxi`, it outputs various details such as:

1. **CPU**: Information about the CPU including model, frequency, number of cores, and more.
2. **Memory (RAM)**: Details on installed RAM, total memory, available memory, etc.
3. **GPU**: Information about your graphics card(s), including model, driver version, and memory usage.
4. **Storage**: Details on hard drives, SSDs, and other storage devices connected to your system.
5. **Display**: Information about connected displays, resolution, refresh rate, etc.
6. **Kernel & OS**: The current kernel version, distribution name, and version, among other OS-related details.

``inxi`` can provide information with varying levels of detail by using different options. For example:

- ``inxi -Fx``: Provides a full report with extended hardware and software information.
- ``inxi -C``: Focuses specifically on CPU information.
- ``inxi -G``: Provides GPU-specific details.

Here is a sample output of running ``inxi``:

```
``plaintext
```

```
CPU: Quad core Intel i7-9700K @ 3.6GHz, average load: 0.15
```

```
GPU: NVIDIA GeForce RTX 2080 Ti, driver: nvidia, v: 418.87.02
```

```
Memory: RAM: total: 31.7Gi used: 3.2Gi free: 28.5Gi swap: total: 8Gi used: 0B  
free: 8Gi
```

```
Disk: NVMe Samsung SSD 960 EVO Plus, model: MZVLB1T4NSDB, size:  
931.5Gi
```

```
CPU Flags: lm nx pae sse sse2 sse3 ssse3 sse4_1 sse4_2 avx avx2 fma aes  
xsaves pdpe1gb rdtscp tsc_adjust bmi1 bmi2 erms invpcid mpx rdseed  
adx smep bmi1 bmi2 erms invpcid mpx rdseed adx smep bmi1 bmi2 erms  
invpcid mpx rdseed adx smep
```

```
````
```

This command is particularly useful for system administrators, developers, and users who need detailed information about their Linux system's hardware configuration.

```
/home/igor/mshell/mscripts > ollama2exec "No any comments. Run linux
ubuntu command: inxi "
```

Sure! `inxi` is a great tool for getting detailed system information on Linux. Here's how you can run it:

```
```bash  
inxi -Fxz  
```
```

This command will provide detailed system information, including CPU, GPU, memory, and disk usage, formatted in an easy-to-read way.

If you don't have `inxi` installed, you can install it using the following commands:

For Debian-based systems (like Ubuntu):

```
```bash  
sudo apt update  
sudo apt install inxi  
```
```

For Fedora:

```
```bash  
sudo dnf install inxi  
```
```

For Arch Linux:

```
```bash  
sudo pacman -S inxi  
```
```

Once installed, you can run `inxi -Fxz` to get your system information.

#### Executing command block via mshell:

#### System:

Kernel: 6.8.0-59-generic arch: x86\_64 bits: 64 compiler: gcc v: 13.3.0

Desktop: GNOME v: 46.0 Distro: Ubuntu 24.04.2 LTS (Noble Numbat)

#### Machine:

Type: Convertible System: ASUSTeK product: VivoBook\_ASUSLaptop  
TP412FA

v: 1.0 serial: <superuser required>

Mobo: ASUSTeK model: TP412FA v: 1.0 serial: <superuser required>

UEFI: American Megatrends v: TP412FA.302 date: 05/06/2019

#### Battery:

ID-1: BAT0 charge: 29.5 Wh (97.4%) condition: 30.3/41.9 Wh (72.4%)

volts: 11.8 min: 11.8 model: ASUSTeK TP412FA status: not charging

Device-1: hidpp\_battery\_0 model: Logitech Wireless Mouse M325

charge: 50% (should be ignored) status: N/A

#### CPU:

Info: quad core model: Intel Core i5-8265U bits: 64 type: MT MCP

arch: Whiskey Lake rev: B cache: L1: 256 KiB L2: 1024 KiB L3: 6 MiB

Speed (MHz): avg: 1012 high: 1100 min/max: 400/3900 cores: 1: 1100 2:

1100

3: 1100 4: 1100 5: 1100 6: 400 7: 1100 8: 1100 bogomips: 28800

Flags: avx avx2 ht lm nx pae sse sse2 sse3 sse4\_1 sse4\_2 ssse3 vmx

#### Graphics:

Device-1: Intel WhiskeyLake-U GT2 [UHD Graphics 620] vendor: ASUSTeK  
driver: i915 v: kernel arch: Gen-9.5 bus-ID: 00:02.0

Device-2: Chicony USB2.0 VGA UVC WebCam driver: uvcvideo type: USB  
bus-ID: 1-4:5

Display: x11 server: X.Org v: 21.1.11 with: Xwayland v: 23.2.6 driver: X:  
loaded: modesetting unloaded: fbdev,vesa dri: iris gpu: i915  
resolution: 1920x1080~60Hz

API: EGL v: 1.5 drivers: iris,swrast platforms:  
active: gbm,x11,surfaceless,device inactive: wayland

API: OpenGL v: 4.6 compat-v: 4.5 vendor: intel mesa  
v: 24.2.8-1ubuntu1~24.04.1 glx-v: 1.4 direct-render: yes renderer: Mesa  
Intel UHD Graphics 620 (WHL GT2)

#### Audio:

Device-1: Intel Cannon Point-LP High Definition Audio vendor: ASUSTeK  
driver: snd\_hda\_intel v: kernel bus-ID: 00:1f.3

API: ALSA v: k6.8.0-59-generic status: kernel-api

Server-1: PipeWire v: 1.0.5 status: active

Server-2: PulseAudio v: 16.1 status: off (using pipewire-pulse)

#### Network:

Device-1: Intel Cannon Point-LP CNVi [Wireless-AC] driver: iwlwifi v:

kernel

bus-ID: 00:14.3

IF: wlo1 state: up mac: <filter>

IF-ID-1: docker0 state: down mac: <filter>

#### Bluetooth:

Device-1: Intel Bluetooth 9460/9560 Jefferson Peak (JfP) driver: btusb

v: 0.8 type: USB bus-ID: 1-10:6

Report: hciconfig ID: hci0 rfk-id: 0 state: down

bt-service: enabled,running rfk-block: hardware: no software: yes

address: <filter>

#### RAID:

Hardware-1: Intel 82801 Mobile SATA Controller [RAID mode] driver: ahci

v: 3.0 bus-ID: 00:17.0

#### Drives:

Local Storage: total: 6.83 TiB used: 2.46 TiB (36.0%)

ID-1: /dev/nvme0n1 vendor: Intel model: SSDPEKNW512G8 size: 476.94

#### GiB

temp: 33.9 C

ID-2: /dev/sda vendor: Seagate model: One Touch HDD size: 4.55 TiB

type: USB

ID-3: /dev/sdb vendor: Seagate model: BUP Slim size: 1.82 TiB type: USB

#### Partition:

ID-1: / size: 283.11 GiB used: 191.72 GiB (67.7%) fs: ext4

dev: /dev/nvme0n1p5

ID-2: /boot/efi size: 256 MiB used: 31.8 MiB (12.4%) fs: vfat

dev: /dev/nvme0n1p1

### Swap:

ID-1: swap-1 type: partition size: 24.41 GiB used: 13.2 MiB (0.1%)

dev: /dev/nvme0n1p6

### Sensors:

System Temperatures: cpu: 84.0 C pch: 77.0 C mobo: N/A

Fan Speeds (rpm): cpu: 0

### Info:

Memory: total: 20 GiB available: 19.25 GiB used: 8.47 GiB (44.0%)

Processes: 340 Uptime: 3d 13h 16m Init: systemd target: graphical (5)

Packages: 2971 Compilers: clang: 18.1.3 gcc: 13.3.0 Shell: sh inxi: 3.3.34

/home/igor/mshell/mscripts >

## **12 . Organization of loop in mshell scripts for the implementation the process of diagnostic automation and other routine tasks (Example4).**

The easiest way to get information about the configuration and state of a computer, node or container is to use “for” and “while” loops in mshell.

### Example4 mshell script:

```
#!/bin/mshell
```

```
Enumerate through a fixed list of items
```

```
print "Initializing sudo..."
```

```
ollama1exec "No any comments. Write the Ubuntu Linux command sudo ls "
```

```
print "Changing directory to mshell scripts..."
```

```
ollama1exec "No any comments. Write the Ubuntu Linux command cd
```

```
/home/igor/mshell/mscripts "
```

```
for command in inxi df tree free iostat vmstat
do
 ollama1exec "No any comments. Write the Ubuntu Linux command $command"
done
ollama1exec "Analyze the results of worked commands and mark performance of
computer to use" | tee /home/igor/mshell/mscripts/for/performanceassessment.txt
```

**Result of Example4 mshell script (analyze will go to performanceassessment.txt file):**

```
/home/igor/mshell/mscripts/for > ./test1.ms
```

```
Loading .mshellrc...
```

```
Loaded environment: PATH=/usr/bin:/bin:/usr/sbin:/usr/local/bin:.$PATH
```

```
Loaded alias: ll='ls -l'
```

```
Loaded alias: home='cd ~'
```

```
Loaded environment: EDITOR=nano
```

```
Lua state initialized successfully
```

```
Range function defined successfully
```

```
Loading history...
```

```
"Initializing sudo..."
```

```
The command you're looking for is:
```

```
```bash
```

```
sudo ls /
```

```
```
```

This command lists all files and directories in the root directory (`/`) with superuser (root) privileges. However, running `ls /` without arguments as `sudo ls` by itself does not

make sense because `sudo` alone doesn't modify the command being run; it only elevates your permissions. The `/` is necessary to specify the root directory as the target for the `ls` command.

If you want to list all files and directories in the current directory with superuser privileges, you would use:

```
```bash
```

```
sudo ls -la
```

```
```
```

This will show a detailed listing of all files and directories in the current directory, including hidden ones, with extended information such as permissions, ownership, size, and timestamps. Remember that using `sudo` can affect your system's security, so it should be used carefully.

### Executing command block via mshell:

```
[sudo] password for igor:
```

```
bin etc lib.usr-is-merged opt sbin.usr-is-merged
```

```
usr bin.usr-is-merged home libx32 proc snap var
```

```
boot lib lost+found root srv cdrom lib32 media
```

```
run sys dev lib64 mnt sbin tmp
```

```
"Changing directory to mshell scripts..."
```

```
```bash
```

```
cd /home/igor/mshell/mscripts
```

```
```
```

Executing command block via mshell:

```
Set command = inxi (string)
```

```
```bash
```

inxi

```

Executing command block via mshell:

CPU: quad core Intel Core i5-8265U (-MT MCP-) speed/min/max: 474/400/3900 MHz

Kernel: 6.8.0-59-generic x86\_64 Up: 2h 23m Mem: 2.79/19.25 GiB (14.5%)

Storage: 6.83 TiB (36.0% used) Procs: 343 Shell: sh inxi: 3.3.34

Set command = df (string)

```bash

df

```

Executing command block via mshell:

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
tmpfs	2018768	58864	1959904	3%	/run
/dev/nvme0n1p5	296867408	202256060	79458300	72%	/tmpfs
tmpfs	10093836	0	10093836	0%	/dev/shm
tmpfs	5120	8	5112	1%	/run/lock
efivarfs	192	63	125	34%	/sys/firmware/efi/efivars
/dev/nvme0n1p1	262144	32512	229632	13%	/boot/efi
tmpfs	2018764	160	2018604	1%	/run/user/1001
/dev/sdb2	1953247232	7352320	1945894912	1%	/media/igor/Backup Plus
/dev/sda2	4883466240	2434794240	2448672000	50%	/media/igor/twodr

Set command = tree (string)

```bash

tree

'''

Executing command block via mshell:

.

```
|— a
|   |— 1.ms
|   |— 2.ms
|   |— 3a.ms
|   |— 3.ms
|   |— 4.ms
|   |— 5.ms
|   |— 6.ms
|   |— 7.ms
|   |— comb.ms
|   |— for.ms
|   |— if.ms
|   |— test.ms
|   |— while.ms
|— b
|   |— 1a.ms
|   |— 1.ms
|   |— 2a.ms
|   |— 2.ms
```

- | |— 3a.ms
- | |— 3.ms
- | |— 4a.ms
- | |— 4.ms
- | |— 5a.ms
- | |— 5.ms
- | |— new1.ms
- | |— new222.ms
- | |— new2.ms
- | |— new.ms
- | |— printeng.ms
- | |— printnoquotes.ms
- | |— printrus.ms
- | |— test1.ms
- |— common
- | |— analyze_real_disk.ms
- | |— calculate.ms
- | |— mel1.ms
- | |— myscript.ms
- | |— test.ms
- | |— tt.ms
- | |— ttt.ms
- |— complex

- | |— 1.ms
- | |— 2.ms
- | |— 3else.ms
- | |— 3l.ms
- | |— 3.ms
- | |— 4.ms
- | |— 5.ms
- | |— complex_for.ms
- | |— complex_if_else.ms
- | |— complex_math_analysis.ms
- | |— complex_while.ms
- | |— math_loops.ms
- | |— mixed_statements.ms
- | |— probelscript.ms
- | |— vector_calculations.ms
- |— conditional
- | |— test_conditional.ms
- | |— test_files.ms
- | |— test_increment.ms
- | |— test_sequence.ms
- | |— test_strings1.ms
- | |— test_strings.ms
- | |— test_vars.ms

- |— condo
- | |— a1.ms
- | |— a.ms
- | |— forcheck.ms
- | |— if1.ms
- | |— if2.ms
- | |— if3compile.ms
- | |— ifcondo.ms
- | |— ifelsecondo.ms
- | |— ifless.ms
- | |— if.ms
- | |— pipered
- | | |— test_modes.ms
- | | |— test_pipe1.ms
- | | |— test_pipe2.ms
- | | |— test_pipe3.ms
- | | |— test_redirect1.ms
- | | |— test_redirect2.ms
- | | |— test_redirect3.ms
- | | |— test_redirect4.ms
- | |— pipered1
- | | |— test_modes.ms
- | | |— test_pipe1.ms

- | | | — test_pipe2.ms
- | | | — test_pipe3.ms
- | | | — test_redirect1.ms
- | | | — test_redirect2.ms
- | | | — test_redirect3.ms
- | | | — test_redirect4.ms
- | | — test_control.ms
- | | — test_eval.ms
- | | — test_loops123.ms
- | | — test_loops1.ms
- | | — test_loops_for.ms
- | | — test_loops.ms
- | | — test_vars1.ms
- | | — test_vars2.ms
- | | — whilecondo.ms
- | — for
- | | — colors.txt
- | | — for1.ms
- | | — for2.ms
- | | — for3.ms
- | | — for4.ms
- | | — for5.ms
- | | — test1.ms

- | | — test2.ms
- | | — test3.ms
- | | — test.ms
- | — math
- | | — a1.ms
- | | — aaasss
- | | — a.ms
- | | — b.ms
- | | — test_math1.ms
- | | — test_math2.ms
- | | — test_math3.ms
- | | — test_math_advanced2.ms
- | | — test_math_advanced3.ms
- | | — test_math_advanced.ms
- | | — test_math_combinatorics.ms
- | | — test_math_complex.ms
- | | — test_math_extended.ms
- | | — test_math.ms
- | | — test_math_trig.ms
- | — pro
- | | — a.ms
- | | — data
- | | | — sample.txt

```
| |— README.md
| |— src
| | |— main1.py
| | |— main.py
| | |— test_main.py
| |— text_analyzer
|— struct
| |— a.ms
| |— for.ms
| |— ifelse.ms
| |— iftest.ms
| |— ollamainb.ms
| |— while.ms
|— test1
| |— a.ms
| |— go_project1.ms
| |— go_project.ms
| |— master_test.ms
| |— matrix_project.ms
| |— system_analysis.ms
| |— test_commands.ms
| |— test_control.ms
| |— test_environment.ms
```

- | |— test_multiline.ms
- | |— test_ollama_act.ms
- | |— test_ollama_exec_all.ms
- | |— test_ollama_exec.ms
- | |— test_ollama_interactive.ms
- | |— test_ollama.ms
- | |— test_ollama_output.ms
- | |— test_vars.ms
- |— test1.ms
- |— test1-nocomments.ms
- |— test3
- | |— 3ollama.ms
- | |— for.ms
- | |— forollama.ms
- | |— if.ms
- | |— ifollama.ms
- | |— while.ms
- |— test4
- | |— a0.ms
- | |— a10.ms
- | |— a11.ms
- | |— a12.ms
- | |— a13a.ms

| — a13b.ms

| — a13c.ms

| — a13.ms

| — a14.ms

| — a15.ms

| — a16.ms

| — a17.ms

| — a18.ms

| — a19.ms

| — a1.ms

| — a20.ms

| — a21.ms

| — a22.ms

| — a23.ms

| — a24.ms

| — a25.ms

| — a26.ms

| — a27.ms

| — a28.ms

| — a29.ms

| — a2.ms

| — a30.ms

| — a31.ms

| — a32.ms

| — a33.ms

| — a34.ms

| — a35.ms

| — a36.ms

| — a37.ms

| — a38.ms

| — a39.ms

| — a3.ms

| — a40.ms

| — a4.ms

| — a5.ms

| — a6.ms

| — a7.ms

| — a8.ms

| — a9.ms

| — bashcall1.ms

| — bashcall.ms

| — bashloop1.sh

| — bashloop.sh

| — data.txt

| — dir_test.ms

| — enu.ms

- | |— f1.ms
- | |— f2.ms
- | |— f3.ms
- | |— f4.ms
- | |— f5.ms
- | |— file_test1.ms
- | |— file_test.ms
- | |— for1.ms
- | |— for2.ms
- | |— for3.ms
- | |— for4.ms
- | |— functest.ms
- | |— ifelse.ms
- | |— if.ms
- | |— multineest.ms
- | |— nested_replace.ms
- | |— nested_test.ms
- | |— numeric_test.ms
- | |— output1.txt
- | |— string_test.ms
- | |— test.txt
- | |— updated.txt
- | |— while.ms

- |— test5
 - | |— ol10.ms
 - | |— ol11.ms
 - | |— ol12.ms
 - | |— ol13.ms
 - | |— ol14.ms
 - | |— ol1.ms
 - | |— ol2.ms
 - | |— ol3.ms
 - | |— ol4.ms
 - | |— ol5.ms
 - | |— ol6.ms
 - | |— ol7.ms
 - | |— ol8.ms
 - | |— ol9_fixed.ms
 - | |— ol9.ms
 - | |— poem.txt
- |— testcomp
 - | |— env_variables.ms
 - | |— filesystem_test.ms
 - | |— master.ms
 - | |— math_advanced_test.ms
 - | |— math_test.ms

- | └─ navigation.ms
- | └─ network_test.ms
- | └─ pipe_redirect_test.ms
- | └─ variable_test.ms
- └─ test.ms
- └─ test-nocomments.ms
- └─ testo
- | └─ co.ms
- | └─ ifwhilefor.ms
- | └─ t1.ms
- | └─ test_all.ms
- | └─ test_ls.ms
- | └─ testo.ms
- └─ testoperators
- | └─ if_ollama_test.ms
- | └─ if_test1.ms
- | └─ if_test2.ms
- └─ while
- └─ e.ms
- └─ f.ms
- └─ g.ms
- └─ h1.ms
- └─ h2.ms

```
|— h3.ms
|— h.ms
|— test.ms
|— while1.ms
|— while2.ms
└— while3.ms
```

24 directories, 278 files

Set command = free (string)

```
```bash
```

```
free
```

```
```
```

Executing command block via mshell:

| | total | used | free | shared | buff/cache | available |
|--------------|-----------------|----------------|-----------------|---------------|-----------------|-----------------|
| Mem: | 20187676 | 2877132 | 5921568 | 368316 | 12126756 | 17310544 |
| Swap: | 25597948 | 0 | 25597948 | | | |

Set command = iostat (string)

```
```bash
```

```
iostat
```

```
```
```

Executing command block via mshell:

Linux 6.8.0-59-generic (asus) 05/11/2025 _x86_64_ (8 CPU)

avg-cpu: %user %nice %system %iowait %steal %idle

1.41 0.00 1.48 0.58 0.00 96.54

| Device | tps | kB_read/s | kB_wrtn/s | kB_dscd/s | kB_read | kB_wrtn | kB_dscd |
|--------|-----|-----------|-----------|-----------|---------|---------|---------|
|--------|-----|-----------|-----------|-----------|---------|---------|---------|

| | | | | | | | |
|-------|------|------|------|------|----|---|---|
| loop0 | 0.00 | 0.00 | 0.00 | 0.00 | 17 | 0 | 0 |
|-------|------|------|------|------|----|---|---|

| | | | | | | | |
|-------|------|------|------|------|-----|---|---|
| loop1 | 0.01 | 0.05 | 0.00 | 0.00 | 473 | 0 | 0 |
|-------|------|------|------|------|-----|---|---|

| | | | | | | | |
|--------|------|------|------|------|------|---|---|
| loop10 | 0.06 | 0.16 | 0.00 | 0.00 | 1385 | 0 | 0 |
|--------|------|------|------|------|------|---|---|

| | | | | | | | |
|---------|------|------|------|------|------|---|---|
| loop100 | 0.18 | 0.42 | 0.00 | 0.00 | 3632 | 0 | 0 |
|---------|------|------|------|------|------|---|---|

| | | | | | | | |
|---------|------|------|------|------|------|---|---|
| loop101 | 0.19 | 0.42 | 0.00 | 0.00 | 3669 | 0 | 0 |
|---------|------|------|------|------|------|---|---|

| | | | | | | | |
|---------|------|------|------|------|------|---|---|
| loop102 | 0.19 | 0.42 | 0.00 | 0.00 | 3669 | 0 | 0 |
|---------|------|------|------|------|------|---|---|

| | | | | | | | |
|---------|------|------|------|------|------|---|---|
| loop103 | 0.18 | 0.42 | 0.00 | 0.00 | 3632 | 0 | 0 |
|---------|------|------|------|------|------|---|---|

| | | | | | | | |
|---------|------|------|------|------|-----|---|---|
| loop104 | 0.02 | 0.04 | 0.00 | 0.00 | 365 | 0 | 0 |
|---------|------|------|------|------|-----|---|---|

| | | | | | | | |
|--------|------|------|------|------|------|---|---|
| loop11 | 0.07 | 0.26 | 0.00 | 0.00 | 2282 | 0 | 0 |
|--------|------|------|------|------|------|---|---|

| | | | | | | | |
|--------|------|------|------|------|-------|---|---|
| loop12 | 0.12 | 1.31 | 0.00 | 0.00 | 11387 | 0 | 0 |
|--------|------|------|------|------|-------|---|---|

| | | | | | | | |
|--------|------|------|------|------|------|---|---|
| loop13 | 0.05 | 0.23 | 0.00 | 0.00 | 1996 | 0 | 0 |
|--------|------|------|------|------|------|---|---|

| | | | | | | | |
|--------|------|------|------|------|------|---|---|
| loop14 | 0.06 | 0.23 | 0.00 | 0.00 | 2025 | 0 | 0 |
|--------|------|------|------|------|------|---|---|

| | | | | | | | |
|--------|------|------|------|------|------|---|---|
| loop15 | 0.01 | 0.14 | 0.00 | 0.00 | 1203 | 0 | 0 |
|--------|------|------|------|------|------|---|---|

| | | | | | | | |
|--------|------|------|------|------|------|---|---|
| loop16 | 0.03 | 0.47 | 0.00 | 0.00 | 4061 | 0 | 0 |
|--------|------|------|------|------|------|---|---|

| | | | | | | | |
|--------|------|------|------|------|-----|---|---|
| loop17 | 0.00 | 0.02 | 0.00 | 0.00 | 138 | 0 | 0 |
|--------|------|------|------|------|-----|---|---|

| | | | | | | | |
|--------|------|------|------|------|-----|---|---|
| loop18 | 0.00 | 0.02 | 0.00 | 0.00 | 139 | 0 | 0 |
|--------|------|------|------|------|-----|---|---|

| | | | | | | | |
|--------|------|------|------|------|-----|---|---|
| loop19 | 0.01 | 0.04 | 0.00 | 0.00 | 359 | 0 | 0 |
|--------|------|------|------|------|-----|---|---|

| | | | | | | | |
|-------|------|------|------|------|-----|---|---|
| loop2 | 0.01 | 0.05 | 0.00 | 0.00 | 476 | 0 | 0 |
|-------|------|------|------|------|-----|---|---|

| | | | | | | | |
|--------|------|------|------|------|------|---|---|
| loop20 | 0.01 | 0.12 | 0.00 | 0.00 | 1082 | 0 | 0 |
|--------|------|------|------|------|------|---|---|

| | | | | | | | |
|--------|------|------|------|------|-------|---|---|
| loop21 | 0.01 | 0.04 | 0.00 | 0.00 | 359 | 0 | 0 |
| loop22 | 0.01 | 0.13 | 0.00 | 0.00 | 1142 | 0 | 0 |
| loop23 | 0.01 | 0.14 | 0.00 | 0.00 | 1174 | 0 | 0 |
| loop24 | 0.17 | 0.49 | 0.00 | 0.00 | 4236 | 0 | 0 |
| loop25 | 0.17 | 0.49 | 0.00 | 0.00 | 4228 | 0 | 0 |
| loop26 | 0.18 | 0.46 | 0.00 | 0.00 | 3944 | 0 | 0 |
| loop27 | 0.18 | 0.45 | 0.00 | 0.00 | 3936 | 0 | 0 |
| loop28 | 0.09 | 0.29 | 0.00 | 0.00 | 2519 | 0 | 0 |
| loop29 | 0.09 | 0.29 | 0.00 | 0.00 | 2526 | 0 | 0 |
| loop3 | 0.01 | 0.13 | 0.00 | 0.00 | 1140 | 0 | 0 |
| loop30 | 0.08 | 0.31 | 0.00 | 0.00 | 2653 | 0 | 0 |
| loop31 | 0.10 | 0.34 | 0.00 | 0.00 | 2976 | 0 | 0 |
| loop32 | 0.08 | 0.31 | 0.00 | 0.00 | 2722 | 0 | 0 |
| loop33 | 0.40 | 6.48 | 0.00 | 0.00 | 56137 | 0 | 0 |
| loop34 | 0.10 | 0.34 | 0.00 | 0.00 | 2984 | 0 | 0 |
| loop35 | 0.09 | 0.34 | 0.00 | 0.00 | 2928 | 0 | 0 |
| loop36 | 0.06 | 0.23 | 0.00 | 0.00 | 2011 | 0 | 0 |
| loop37 | 0.06 | 0.23 | 0.00 | 0.00 | 2027 | 0 | 0 |
| loop38 | 0.00 | 0.04 | 0.00 | 0.00 | 346 | 0 | 0 |
| loop39 | 0.01 | 0.04 | 0.00 | 0.00 | 347 | 0 | 0 |
| loop4 | 0.01 | 0.13 | 0.00 | 0.00 | 1162 | 0 | 0 |
| loop40 | 0.00 | 0.01 | 0.00 | 0.00 | 49 | 0 | 0 |
| loop41 | 0.00 | 0.01 | 0.00 | 0.00 | 51 | 0 | 0 |

| | | | | | | | |
|--------|------|------|------|------|-------|---|---|
| loop42 | 0.61 | 1.29 | 0.00 | 0.00 | 11145 | 0 | 0 |
| loop43 | 0.75 | 1.84 | 0.00 | 0.00 | 15920 | 0 | 0 |
| loop44 | 0.00 | 0.01 | 0.00 | 0.00 | 76 | 0 | 0 |
| loop45 | 0.01 | 0.01 | 0.00 | 0.00 | 81 | 0 | 0 |
| loop46 | 0.00 | 0.00 | 0.00 | 0.00 | 32 | 0 | 0 |
| loop47 | 0.01 | 0.04 | 0.00 | 0.00 | 346 | 0 | 0 |
| loop48 | 1.15 | 2.78 | 0.00 | 0.00 | 24071 | 0 | 0 |
| loop49 | 0.43 | 0.91 | 0.00 | 0.00 | 7908 | 0 | 0 |
| loop5 | 0.06 | 0.25 | 0.00 | 0.00 | 2147 | 0 | 0 |
| loop50 | 0.39 | 0.83 | 0.00 | 0.00 | 7187 | 0 | 0 |
| loop51 | 0.65 | 1.63 | 0.00 | 0.00 | 14134 | 0 | 0 |
| loop52 | 0.65 | 1.63 | 0.00 | 0.00 | 14104 | 0 | 0 |
| loop53 | 0.67 | 1.69 | 0.00 | 0.00 | 14598 | 0 | 0 |
| loop54 | 0.62 | 1.54 | 0.00 | 0.00 | 13352 | 0 | 0 |
| loop55 | 0.82 | 2.03 | 0.00 | 0.00 | 17550 | 0 | 0 |
| loop56 | 0.62 | 1.53 | 0.00 | 0.00 | 13291 | 0 | 0 |
| loop57 | 0.58 | 1.43 | 0.00 | 0.00 | 12410 | 0 | 0 |
| loop58 | 0.17 | 0.39 | 0.00 | 0.00 | 3386 | 0 | 0 |
| loop59 | 0.14 | 0.33 | 0.00 | 0.00 | 2822 | 0 | 0 |
| loop6 | 0.06 | 0.25 | 0.00 | 0.00 | 2196 | 0 | 0 |
| loop60 | 0.00 | 0.01 | 0.00 | 0.00 | 63 | 0 | 0 |
| loop61 | 0.00 | 0.01 | 0.00 | 0.00 | 67 | 0 | 0 |
| loop62 | 0.02 | 0.03 | 0.00 | 0.00 | 302 | 0 | 0 |

| | | | | | | | |
|--------|------|------|------|------|-------|---|---|
| loop63 | 0.02 | 0.03 | 0.00 | 0.00 | 303 | 0 | 0 |
| loop64 | 0.00 | 0.00 | 0.00 | 0.00 | 16 | 0 | 0 |
| loop65 | 0.00 | 0.00 | 0.00 | 0.00 | 20 | 0 | 0 |
| loop66 | 0.12 | 0.24 | 0.00 | 0.00 | 2071 | 0 | 0 |
| loop67 | 0.15 | 0.29 | 0.00 | 0.00 | 2479 | 0 | 0 |
| loop68 | 0.15 | 0.29 | 0.00 | 0.00 | 2472 | 0 | 0 |
| loop69 | 0.04 | 0.09 | 0.00 | 0.00 | 815 | 0 | 0 |
| loop7 | 0.05 | 0.14 | 0.00 | 0.00 | 1227 | 0 | 0 |
| loop70 | 0.04 | 0.11 | 0.00 | 0.00 | 920 | 0 | 0 |
| loop71 | 0.16 | 0.31 | 0.00 | 0.00 | 2682 | 0 | 0 |
| loop72 | 0.00 | 0.00 | 0.00 | 0.00 | 30 | 0 | 0 |
| loop73 | 0.00 | 0.00 | 0.00 | 0.00 | 26 | 0 | 0 |
| loop74 | 0.12 | 0.27 | 0.00 | 0.00 | 2354 | 0 | 0 |
| loop75 | 0.12 | 0.27 | 0.00 | 0.00 | 2347 | 0 | 0 |
| loop76 | 0.00 | 0.00 | 0.00 | 0.00 | 37 | 0 | 0 |
| loop77 | 0.01 | 0.01 | 0.00 | 0.00 | 119 | 0 | 0 |
| loop78 | 0.00 | 0.00 | 0.00 | 0.00 | 41 | 0 | 0 |
| loop79 | 0.00 | 0.00 | 0.00 | 0.00 | 30 | 0 | 0 |
| loop8 | 0.05 | 0.14 | 0.00 | 0.00 | 1238 | 0 | 0 |
| loop80 | 0.00 | 0.00 | 0.00 | 0.00 | 42 | 0 | 0 |
| loop81 | 0.08 | 3.17 | 0.00 | 0.00 | 27484 | 0 | 0 |
| loop82 | 0.00 | 0.00 | 0.00 | 0.00 | 11 | 0 | 0 |
| loop83 | 0.00 | 0.03 | 0.00 | 0.00 | 285 | 0 | 0 |

| | | | | | | | |
|---------|-------|--------|--------|------|---------|---------|---|
| loop84 | 0.17 | 0.38 | 0.00 | 0.00 | 3334 | 0 | 0 |
| loop85 | 0.17 | 0.39 | 0.00 | 0.00 | 3365 | 0 | 0 |
| loop86 | 0.01 | 0.01 | 0.00 | 0.00 | 116 | 0 | 0 |
| loop87 | 0.01 | 0.02 | 0.00 | 0.00 | 170 | 0 | 0 |
| loop88 | 0.65 | 1.35 | 0.00 | 0.00 | 11661 | 0 | 0 |
| loop89 | 0.01 | 0.03 | 0.00 | 0.00 | 233 | 0 | 0 |
| loop9 | 0.06 | 0.16 | 0.00 | 0.00 | 1367 | 0 | 0 |
| loop90 | 0.01 | 0.03 | 0.00 | 0.00 | 235 | 0 | 0 |
| loop91 | 0.01 | 0.03 | 0.00 | 0.00 | 235 | 0 | 0 |
| loop92 | 0.01 | 0.03 | 0.00 | 0.00 | 236 | 0 | 0 |
| loop93 | 0.01 | 0.03 | 0.00 | 0.00 | 242 | 0 | 0 |
| loop94 | 0.01 | 0.03 | 0.00 | 0.00 | 248 | 0 | 0 |
| loop95 | 0.65 | 1.35 | 0.00 | 0.00 | 11665 | 0 | 0 |
| loop96 | 0.01 | 0.03 | 0.00 | 0.00 | 252 | 0 | 0 |
| loop97 | 0.18 | 0.40 | 0.00 | 0.00 | 3458 | 0 | 0 |
| loop98 | 0.01 | 0.03 | 0.00 | 0.00 | 238 | 0 | 0 |
| loop99 | 0.18 | 0.40 | 0.00 | 0.00 | 3460 | 0 | 0 |
| nvme0n1 | 9.70 | 338.58 | 159.50 | 0.00 | 2932780 | 1381549 | 0 |
| sda | 10.61 | 323.61 | 0.00 | 0.00 | 2803089 | 1 | 0 |
| sdb | 0.50 | 2.70 | 0.00 | 0.00 | 23391 | 1 | 0 |

Set command = vmstat (string)

```
```bash
```

```
vmstat
```

...

## Executing command block via mshell:

```
procs -----memory----- --swap-- -----io---- -system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st gu
2 0 0 5919028 3072736 9054188 0 0 710 159 1396 5 1 1 97 1 0 0
```

Analyzing the results of commands like `df`, `tree`, `free`, `iostat`, and `vmstat` can provide valuable insights into the performance and health of your Ubuntu Linux system. Here is a breakdown of what each command reveals and how you can interpret it:

### ### 1. **df**

- **Command**: `df`

- **Purpose**: Reports disk space usage.

- **Output Example**:

...

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	20G	15G	5G	75%	/

...

- **Interpretation**:

- **Size**: Total disk space.

- **Used**: Space that has been used.

- **Avail**: Free space available.

- **Use%**: Percentage of disk space used.

- **Mounted on**: Directory where the filesystem is mounted.

### **Performance Considerations**:

- If `Use%` is close to 100%, it indicates that the disk might be running out of space. You should consider cleaning up unnecessary files or increasing disk space.

### ### 2. **tree**

- **Command**: `tree``

- **Purpose**: Creates a directory tree visualization.

- **Output Example**:

...

.

|— Desktop

|— Documents

|— Downloads

|— Pictures

4 directories, 0 files

...

**Interpretation**:

- The output shows the hierarchical structure of your filesystem, including directories and subdirectories.

**Performance Considerations**:

- `tree`` does not directly impact performance but helps in understanding the directory structure. Performance issues may arise if you have many small files that take time to traverse.

### ### 3. **free**

- **Command**: `free``

- **Purpose**: Reports memory usage.

- **Output Example**:

...

total	used	free	shared	buff/cache	available
-------	------	------	--------	------------	-----------

```
Mem: 7961 2403 1874 159 3684 4721
```

```
Swap: 2047 0 2047
```

```
...
```

#### **\*\*Interpretation\*\*:**

- **\*\*total\*\***: Total physical memory available.
- **\*\*used\*\***: Memory that is currently in use by processes.
- **\*\*free\*\***: Physical memory that is not used at all.
- **\*\*shared\*\***: Shared memory used by multiple processes.
- **\*\*buff/cache\*\***: Memory used as buffer and cache for disk I/O operations.
- **\*\*available\*\***: Estimated amount of memory available to applications.

#### **\*\*Performance Considerations\*\*:**

- If `used` is close to `total`, it might indicate that your system is running low on RAM. Monitor and optimize processes or consider adding more RAM if necessary.

#### **### 4. \*\*iostat\*\***

- **\*\*Command\*\***: `iostat`
- **\*\*Purpose\*\***: Reports CPU statistics and I/O device usage.
- **\*\*Output Example\*\***:

```
...
```

```
avg-cpu: %user %nice %system %iowait %steal %idle
```

```
0.12 0.00 0.34 0.23 0.00 99.11
```

```
Device: rrqm/s wrqm/s r/s w/s kB/s kB/s avgrq-sz avgqu-sz await
r_await w_await svctm %util
```

```
 sda 0.00 0.00 0.23 4.67 1.25 23.89 50.68 0.02 25.67
24.78 37.50 0.25 0.02
```

```
...
```

**Interpretation**:

- **avg-cpu**: CPU usage statistics.
- **Device**: I/O device statistics.

**Performance Considerations**:

- High `await` and `iowait` values indicate that the system is spending a lot of time waiting for disk I/O operations. This can be due to slow disks or heavy I/O loads. Optimize applications or consider upgrading storage.

### 5. **vmstat**

- **Command**: `vmstat`
- **Purpose**: Reports virtual memory statistics.
- **Output Example**:

...

```
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
```

```
r b swpd free buff cache si so bi bo in cs us sy id wa st
```

```
1 0 0 2564832 311792 4544172 0 0 0 0 20 15 0 0 100 0 0
```

...

**Interpretation**:

- **procs**: Number of processes in various states.
- **memory**: Memory usage statistics.
- **swap**: Swap space usage.
- **io**: I/O activity statistics.
- **system**: System-related statistics.

- **cpu**: CPU usage statistics.

### **Performance Considerations**:

- If `us`` and `sy`` (user and system) are high, it indicates that your system is heavily loaded with processes. Monitor and optimize processes or consider scaling up resources if necessary.

### ### Summary

- **Disk Space (`df``)**: Monitor disk usage to avoid running out of space.

- **Memory Usage (`free``)**: Ensure you have enough free memory for applications.

- **I/O Device Usage (`iostat``)**: Identify slow disks or heavy I/O loads and optimize accordingly.

- **System-Wide Performance (`vmstat``)**: Check CPU and memory usage to ensure the system is not overwhelmed.

By regularly monitoring these metrics, you can make informed decisions about optimizing your Ubuntu Linux system for better performance.

## 13. Conclusion.

- Basic description of mshell functionality (Linux compatible OSes like Ubuntu, Debian, etc., Raspberry PI OS) can be found here:

<https://www.appservgrid.com/paw92/index.php/2025/03/30/mshell-new-linux-shell-for-ai-and-mathematics/>

- The basic principles of how the Linux distributed LLM evaluation framework works can be found in the diagram here:

<https://www.appservgrid.com/paw92/index.php/2025/03/31/llm-evaluation-framework-for-local-use-may-aug-2024/>

- The mshell script + LLM standalone on-premise technology (or Linux LLM distributed evaluation framework) is quite suitable for diagnostics, monitoring and troubleshooting of different non-critical environments.

Adding info: [igor.lukyanov@appservgrid.com](mailto:igor.lukyanov@appservgrid.com) Art2Dec Soft Lab.

05.11.2025

---

In order to see the whole picture, you need to make assessments, not analysis. "War of Meanings" Igor.