

ZORIN OS



A Case Study Project

Fatima Farooq 2023006109

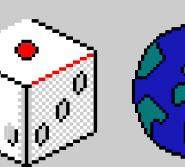
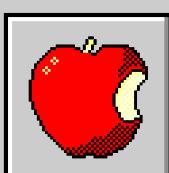
Alya Alzaabi - 2022005560

Rabaa Alshbib 2022005690

roodh alblooshi 2022005691

Lana Zanneh 2022005620

Dema Ammar Al-sos 2021004885



11:11PM

Topics Covered

Start

Introduction

Process Implementation

Thread Implementation

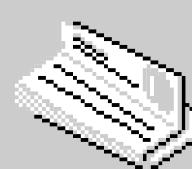
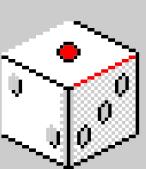
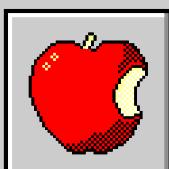
System Calls

Agenda

A vertical scroll bar is positioned to the right of the 'Topics Covered' window. A large button labeled 'Start' is located to the right of the scroll bar, with a mouse cursor pointing at it.

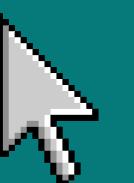
INTRODUCTION TO ZORIN OS?

- Zorin is built on stable Ubuntu Long Term Support Base (LTS) which was launched in 2009 by the Zorin Group in Dublin, Ireland.
- Zorin is a user-friendly Linux distribution which is designed to serve as a convincing alternative to Windows and macOS, allowing the options of both free and paid.
- Its primary goal is to make advanced computing accessible to users by combining the power and security of Linux that provides an interface which is similar for use to other operating systems.
- Our case study features the latest Zorin OS 17.3 release version in the form of core.

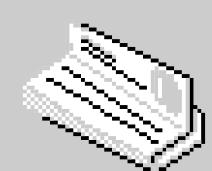
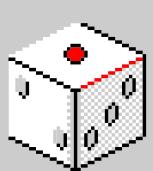
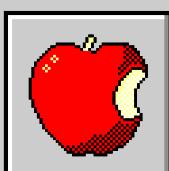


[Back to Agenda Page](#)

WHY ZORIN OS?



- As we are primarily Windows and macOS users, we wanted to use such an operating system that provides a familiar and intuitive interface for such transitioning users..
- Zorin OS even allows the users to customize the environment to resemble Windows, macOS and even older versions of the Windows, allowing the transition to be smoother.
- Another feature is that Zorin OS comes with pre-installed essential productivity, multimedia and internet applications so users can start working immediately.
- The OS also supports additional apps through the Ubuntu repositories, Flathub and Snap Store.

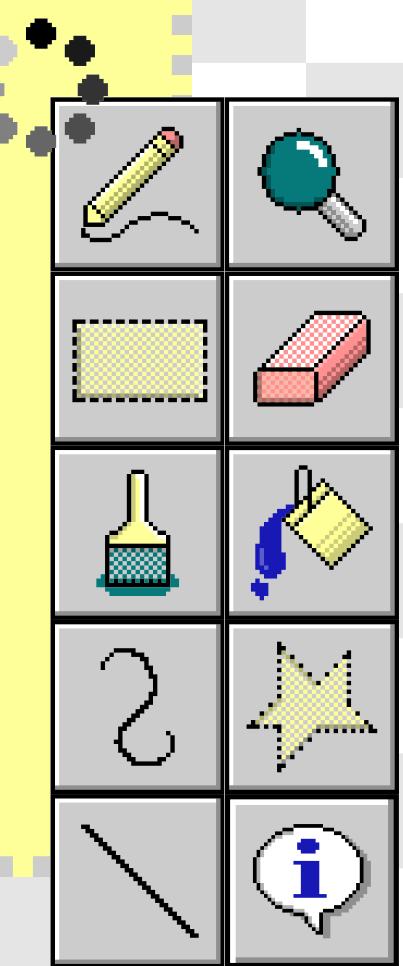
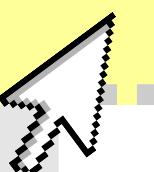


[Back to Agenda Page](#)

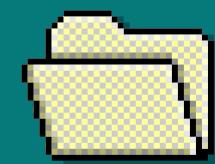


Steps for Installing Zorin OS

[Back to Agenda Page](#)



1. Set Up a Virtual Machine



V VirtualBox

Home Download Documentation Community Search:

Download VirtualBox

The VirtualBox Extension Pack is available for personal and educational use on this page under the PUEL license. The VirtualBox Extension Pack is also available under commercial or enterprise terms. By downloading, you agree to the terms and conditions of the respective license.

VirtualBox Platform | Click on Windows hosts, since I use Windows 11

VirtualBox 7.1.6 platform packages

- [Windows hosts](#)
- [macOS / Intel hosts](#)
- [macOS / Apple Silicon hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)
- [Solaris 11 IPS hosts](#)

Platform packages are released under the terms of the [GPL version 3](#)

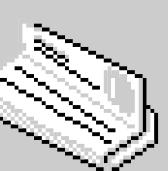
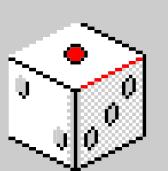
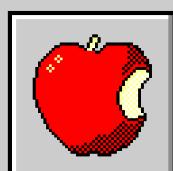
VirtualBox Extension Pack
VirtualBox 7.1.6 Extension Pack

This VirtualBox Extension Pack Personal Use and Educational License governs your access to and use of the VirtualBox Extension Pack. It does not apply to the VirtualBox base package and/or its source code, which are licensed under version 3 of the GNU General Public License "GPL").

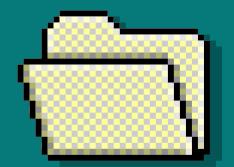
See our [FAQ](#) for answers to common questions.

VirtualBox Extension Pack Personal Use and Educational License (PUEL)

[PUEL License FAQ](#) [PUEL License Text](#) [Accept and download](#)



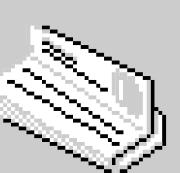
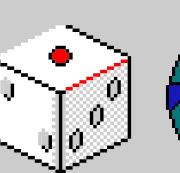
[Back to Agenda Page](#)



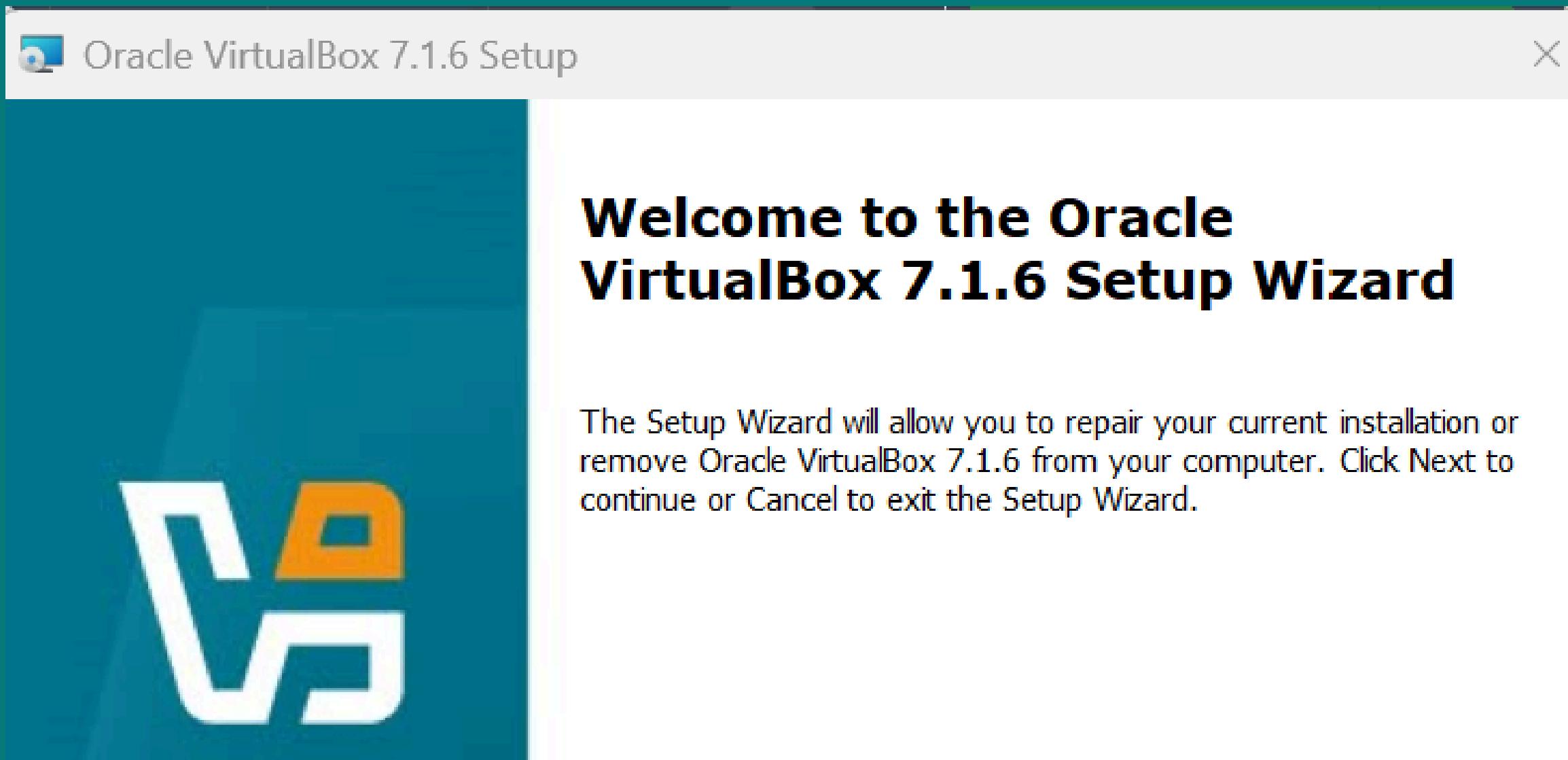
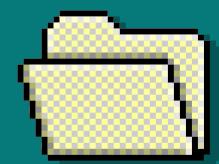
Downloads	os	05/04/2025 7:33 PM	PNG File	254 KB
Documents	Zorin-OS-17.3-Core-64-bit.iso	05/04/2025 1:32 AM	Windows.IsoFile	3,571,712 KB
Pictures	Assignment 2			2 KB
Music	VirtualBox-7.1.6-167084-Win			4 KB
Videos	Oracle_VirtualBox_Extension_Pack-7.1.6	05/04/2025 12:15 AM	VirtualBox Extensio...	22,435 KB

Locate the file in your Downloads folder

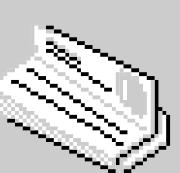
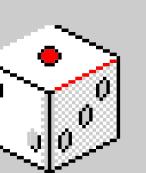
Click on the file



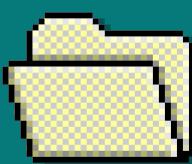
[Back to Agenda Page](#)



Set up the Oracle VirtualBox



[Back to Agenda Page](#)



Oracle VirtualBox 7.1.6 License Agreement

End-User License Agreement

Please read the following license agreement carefully.

COPYING file for VirtualBox versions 7.0 and later versions that include this file

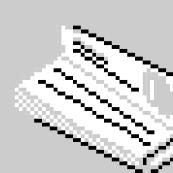
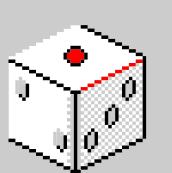
Preliminary notes:

1) The majority of the code in the VirtualBox base package is licensed under the GNU General Public License, version 3 (GPL). VirtualBox contains many components developed by Oracle and various third parties. The license for each component is located in the licensing

I accept the terms in the License Agreement
 I do not accept the terms in the License Agreement

Version 7.1.6 < Back Next > Cancel

Accept the License Agreement



[Back to Agenda Page](#)

Oracle VirtualBox 7.1.6 Setup

Custom Setup

Select the way you want features to be installed.

Click on the icons in the tree below to change the way features will be installed.

VirtualBox Application

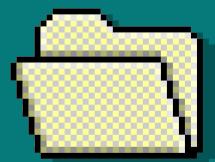
- VirtualBox USB Support
- VirtualBox Networking
 - VirtualBox Bridged Network
 - VirtualBox Host-Only Network
- VirtualBox Python Support

Oracle VirtualBox 7.1.6 application.

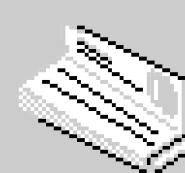
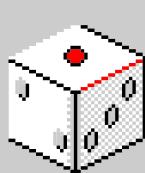
This feature requires 226MB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 936KB on your hard drive.

Location: C:\Program Files\Oracle\VirtualBox\

Browse



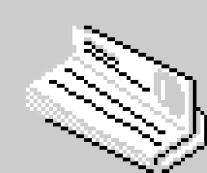
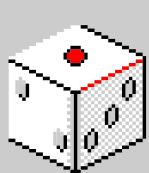
Keep Default Settings



[Back to Agenda Page](#)



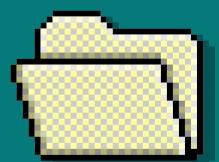
Proceed with Installation



[Back to Agenda Page](#)



Custom Setup

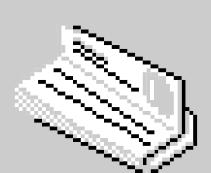
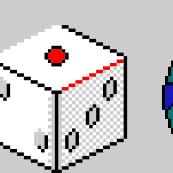


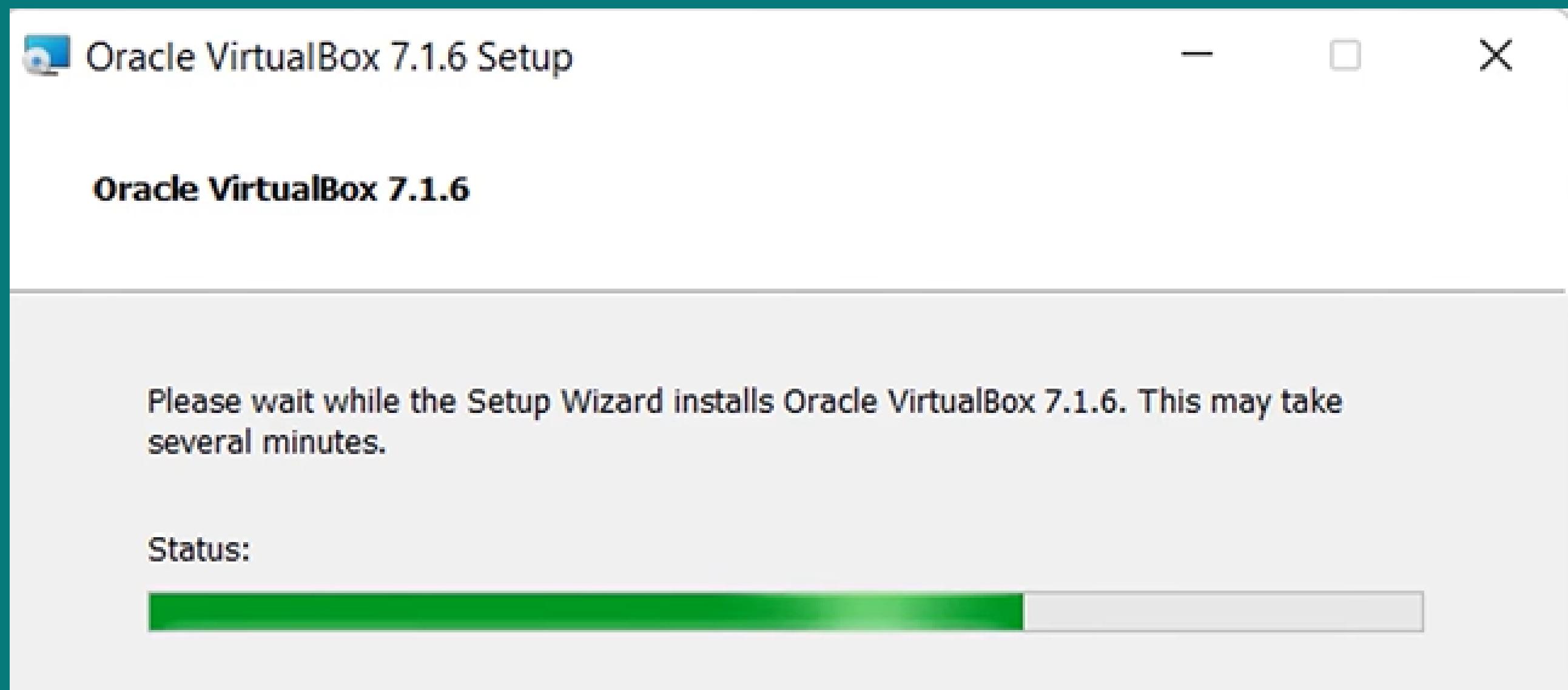
Select the way you want features to be installed.

Please choose from the options below:

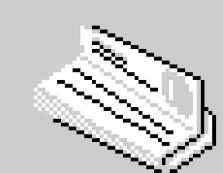
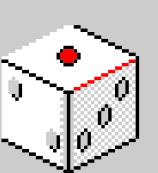
- Create start menu entries
- Create a shortcut on the desktop
- Create a shortcut in the Quick Launch Bar
- Register file associations

Choose based on your preference

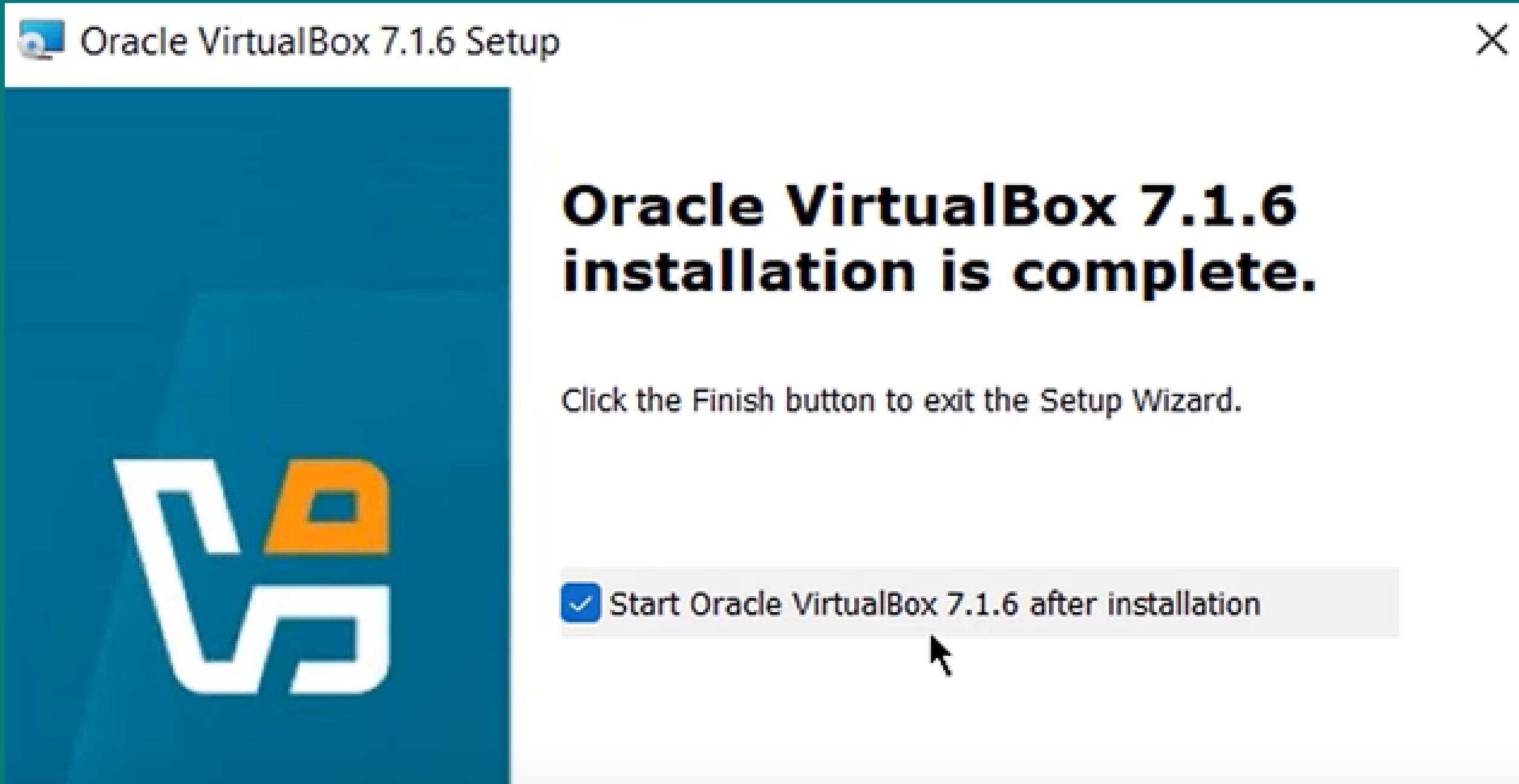




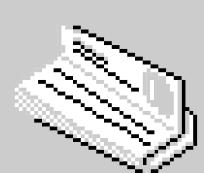
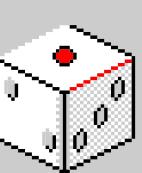
Wait for it to install



[Back to Agenda Page](#)

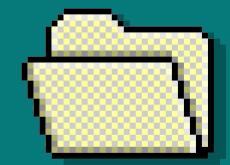


Once it is finished, run it



[Back to Agenda Page](#)

2. Install Zorin OS (Core Version)



zorin.com/os/download/

**Zorin OS 17.3
Core**

For basic use.

Download

Free

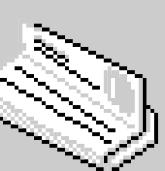
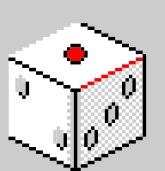
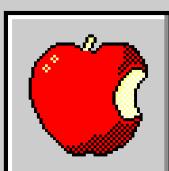
**Zorin OS 17.3
Education**

With educational software for schools and students.

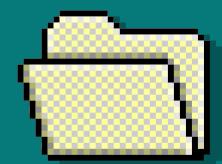
Download

Free

[Learn more >](#)



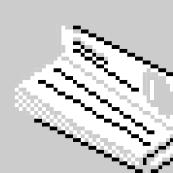
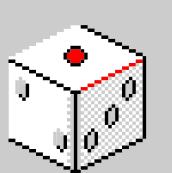
[Back to Agenda Page](#)



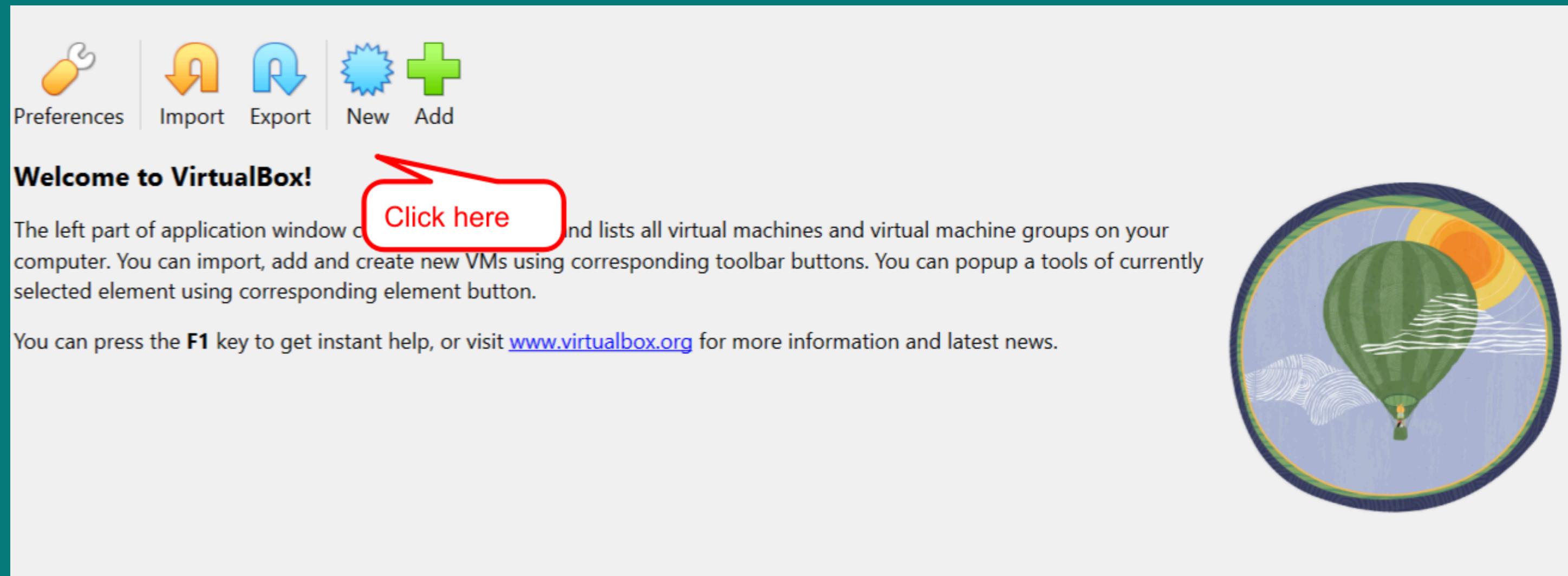
Downloads	os	05/04/2025 7:33 PM	PNG File	254 KB
Documents	Zorin-OS-17.3-Core-64-bit.iso	05/04/2025 1:32 AM	Windows.IsoFile	3,571,712 KB
Pictures	Assignment 2		Microsoft Word Do...	492 KB
Music	VirtualBox-7.1.6-167084-Win		Application	120,134 KB
Videos	Oracle_VirtualBox_Extension_Pack-7.1.6	05/04/2025 12:15 AM	VirtualBox Extensio...	22,435 KB

Downloaded OS file

Locate the .iso file in your
downloads folder



[Back to Agenda Page](#)



The screenshot shows the VirtualBox application window. At the top, there's a toolbar with five icons: Preferences (gear), Import (arrow pointing down), Export (arrow pointing up), New (blue starburst), and Add (green plus). Below the toolbar, the text "Welcome to VirtualBox!" is displayed. A red callout bubble points from the text "Click here" to the "New" button in the toolbar. The main area contains descriptive text about the application and its features, followed by a link to the official website. To the right of the text is a circular logo featuring a hot air balloon over water.

Preferences Import Export

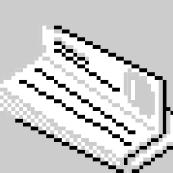
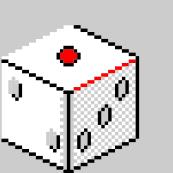
New Add

Welcome to VirtualBox!

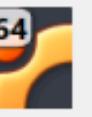
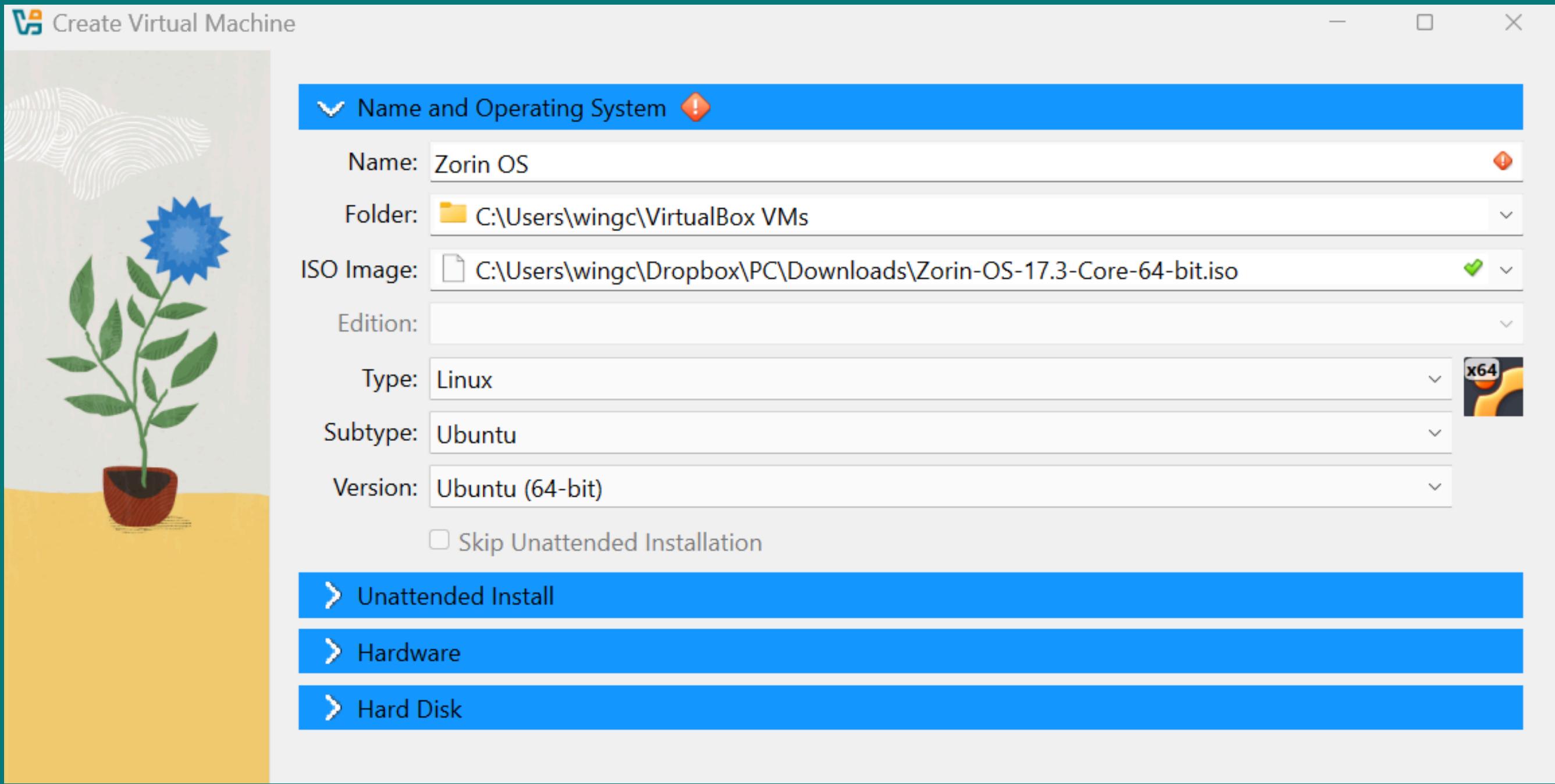
The left part of application window c Click here and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the F1 key to get instant help, or visit www.virtualbox.org for more information and latest news.

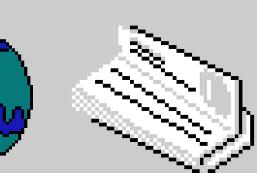
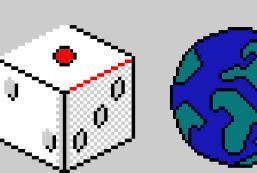
Create a new Virtual Machine (for Zorin OS) by clicking on new



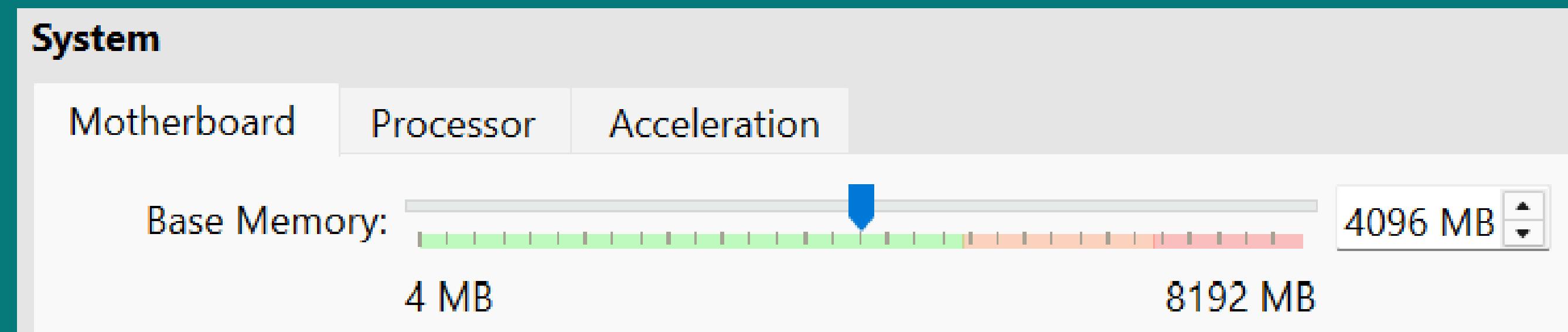
[Back to Agenda Page](#)



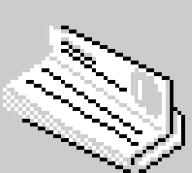
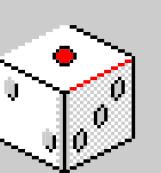
Will automatically detect it as Ubuntu (64-bit)



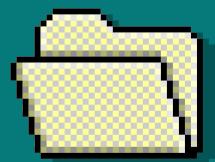
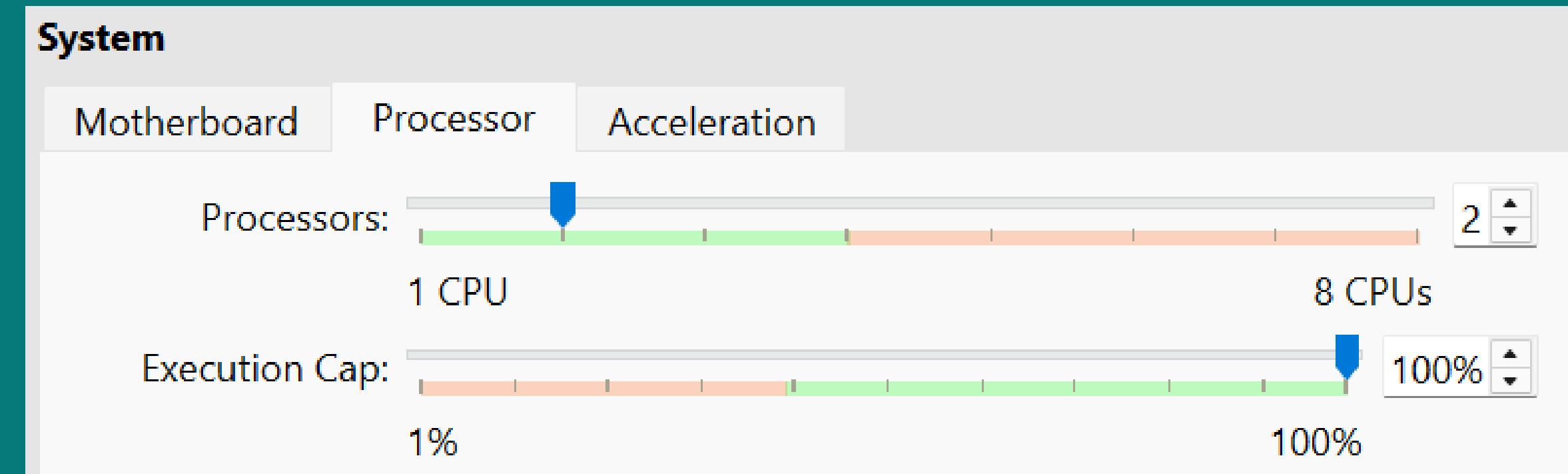
[Back to Agenda Page](#)



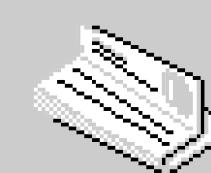
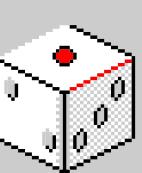
Set the Base Memory to 4GB, since my Laptop has only 8GB RAM, so ensures smooth performance for both VM and laptop.



[Back to Agenda Page](#)



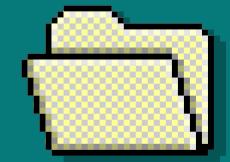
Set the processors to two, so it balances the performance without slowing down the laptop



[Back to Agenda Page](#)

Information

Type (Format): Normal (VDI)



Virtual size: 25.00 GB

Actual size: 14.49 GB

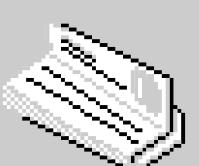
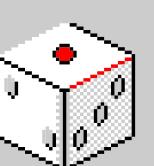
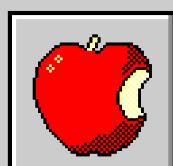
Storage details: Dynamically allocate...

Location: C:\Users\wingc\Virtu...

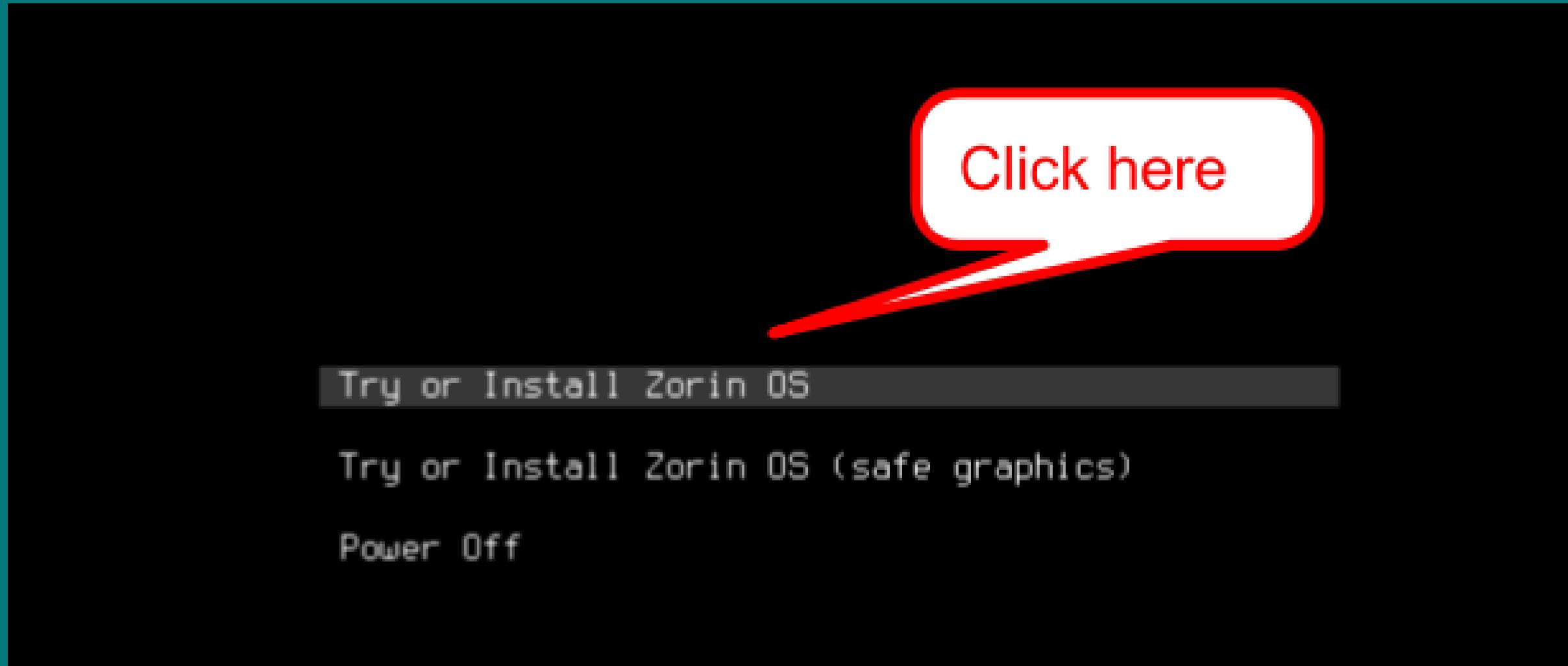
Attached to: Zorin OS

Encryption key: --

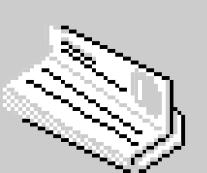
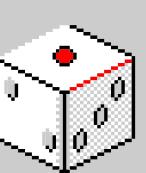
Zorin requires minimum of 20GB, for
flexibility, choose 25GB.



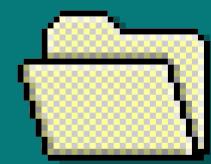
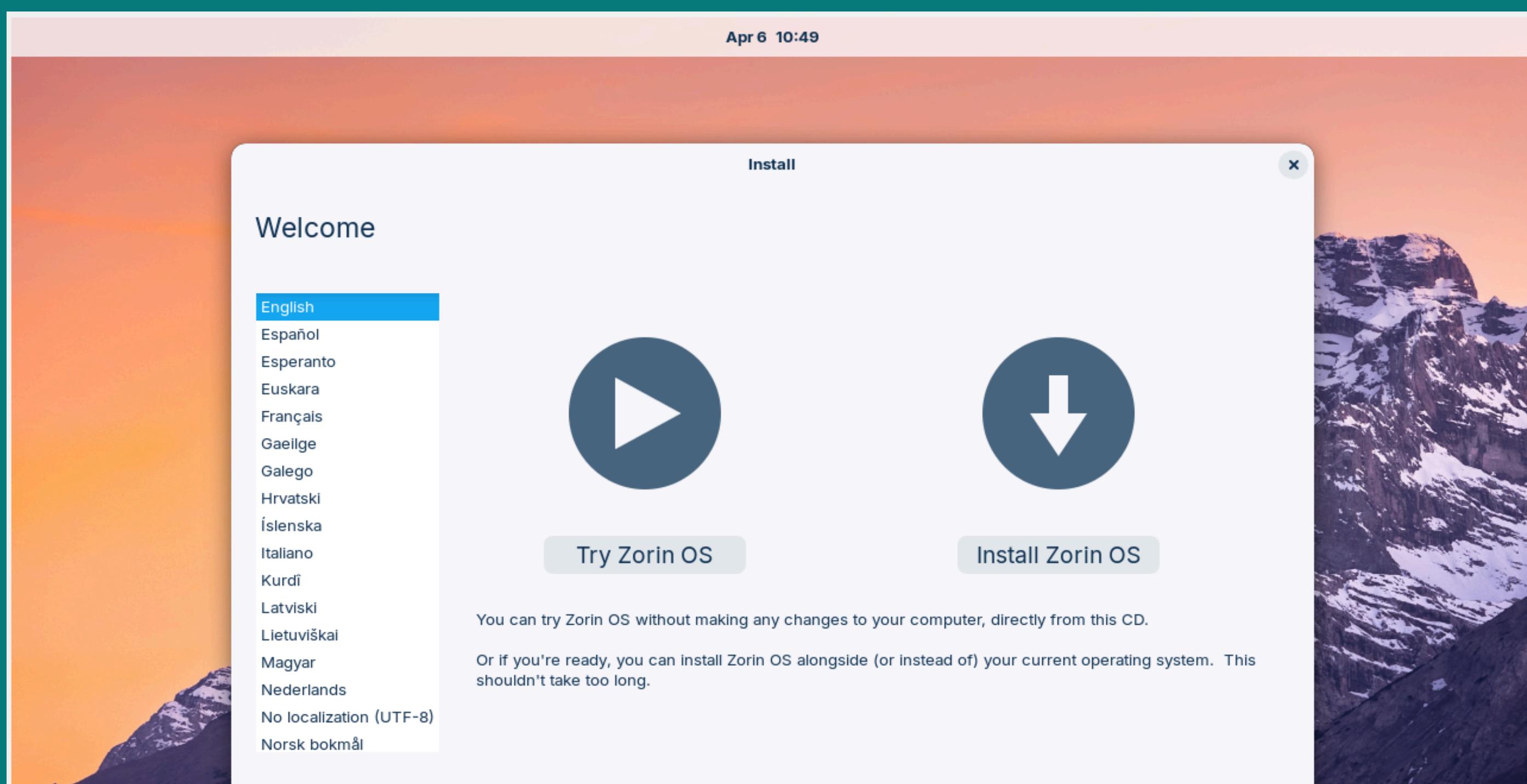
[Back to Agenda Page](#)



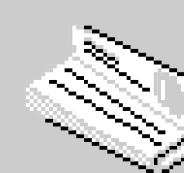
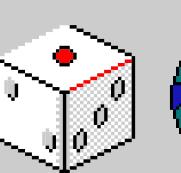
After you have finished your hardware specifications, you are led to this menu



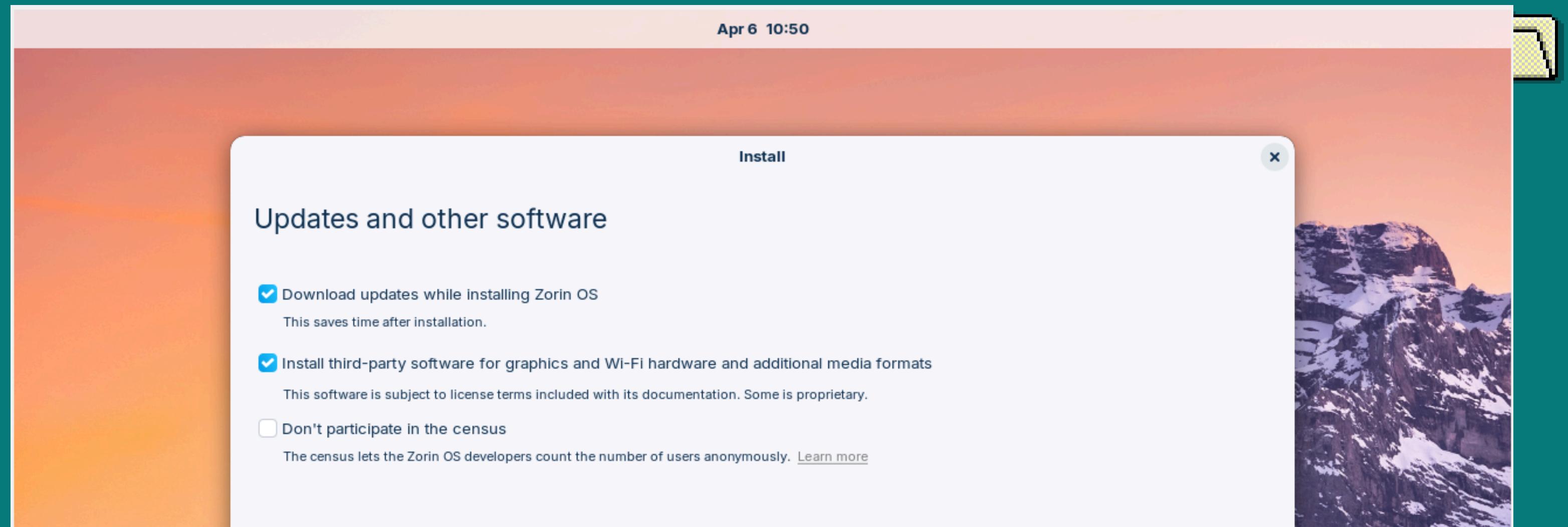
[Back to Agenda Page](#)



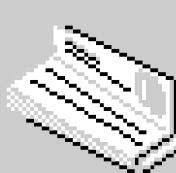
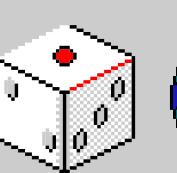
Install Zorin OS



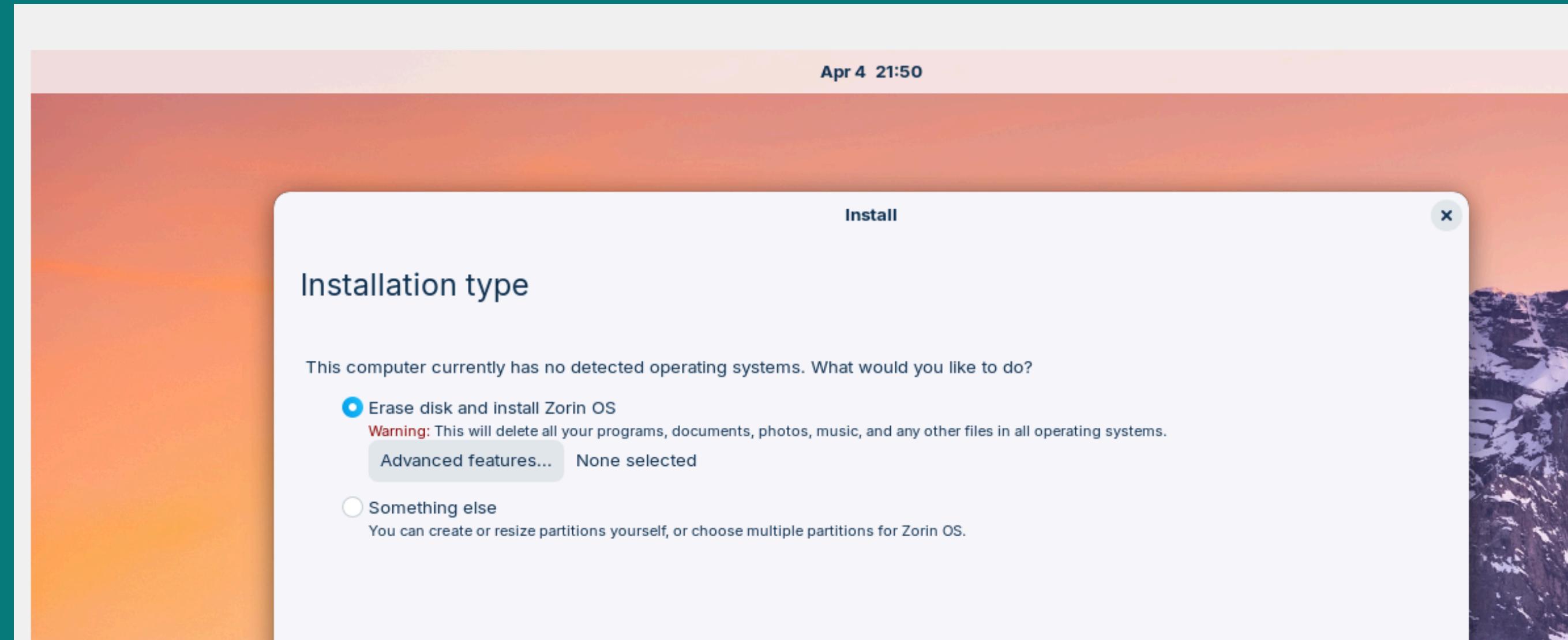
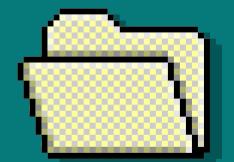
[Back to Agenda Page](#)



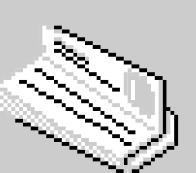
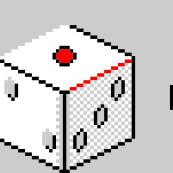
Choose based on your preference



[Back to Agenda Page](#)

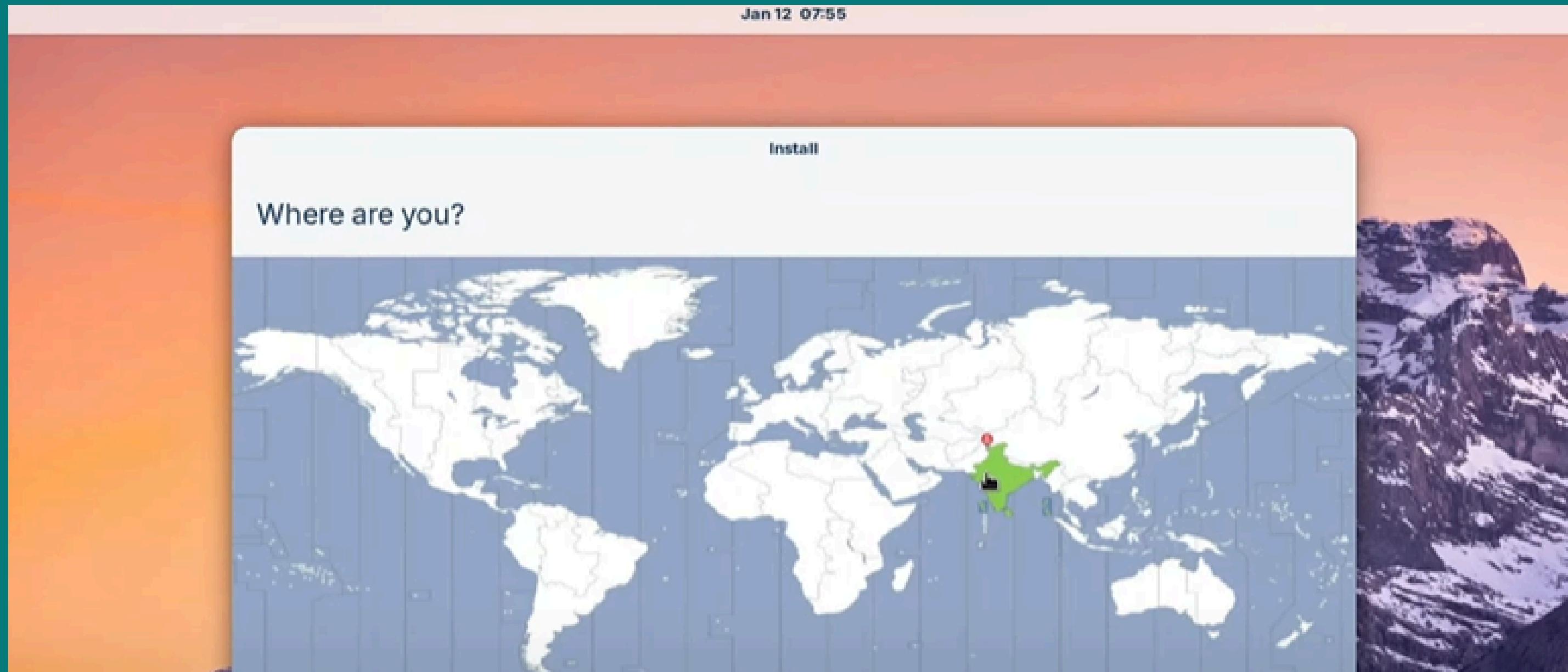
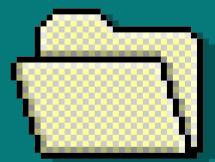


Choose Erase Disk because we do not have an existing OS on VM

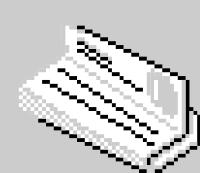
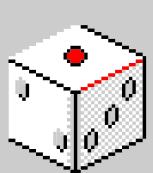


[Back to Agenda Page](#)

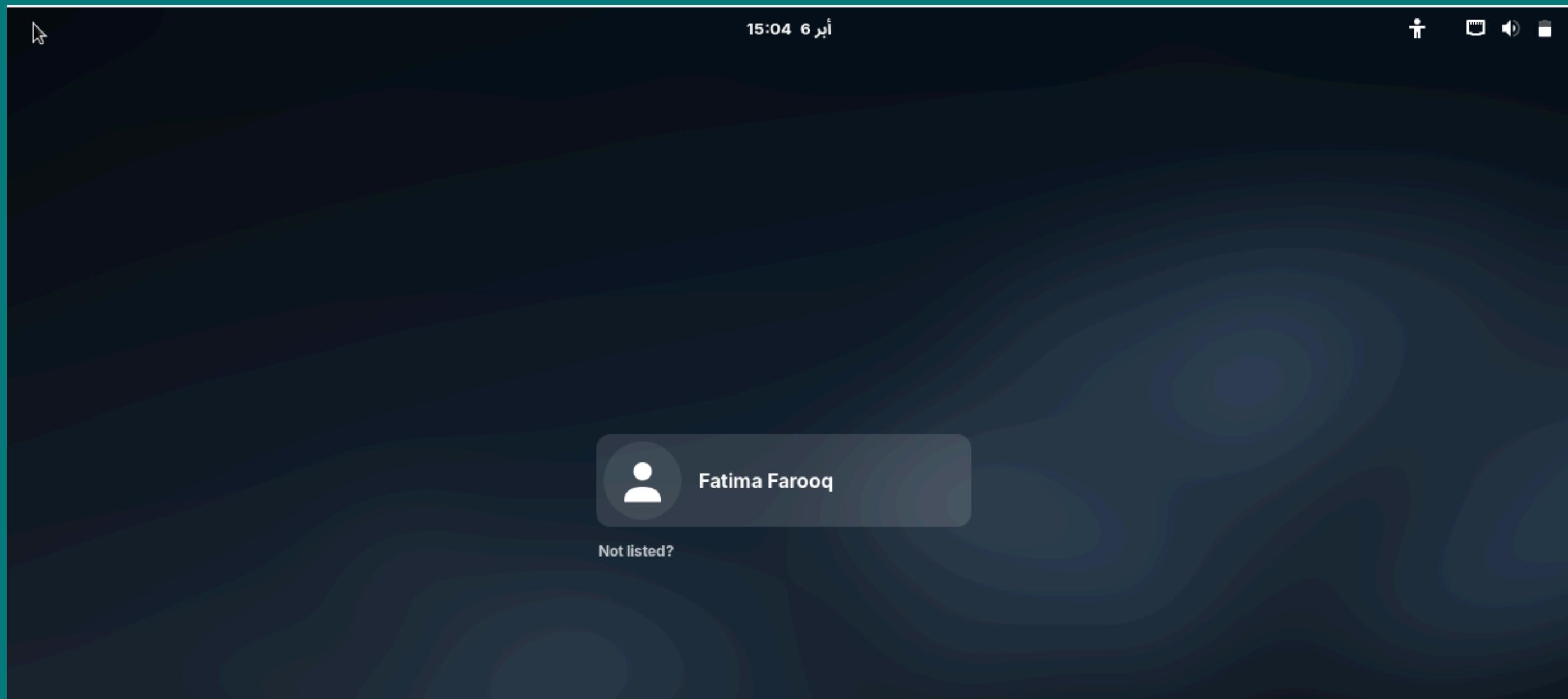
Jan 12 07:55



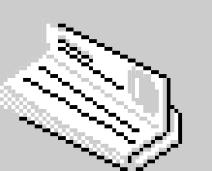
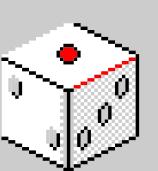
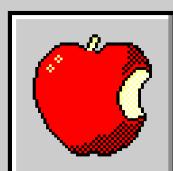
Choose your location



[Back to Agenda Page](#)

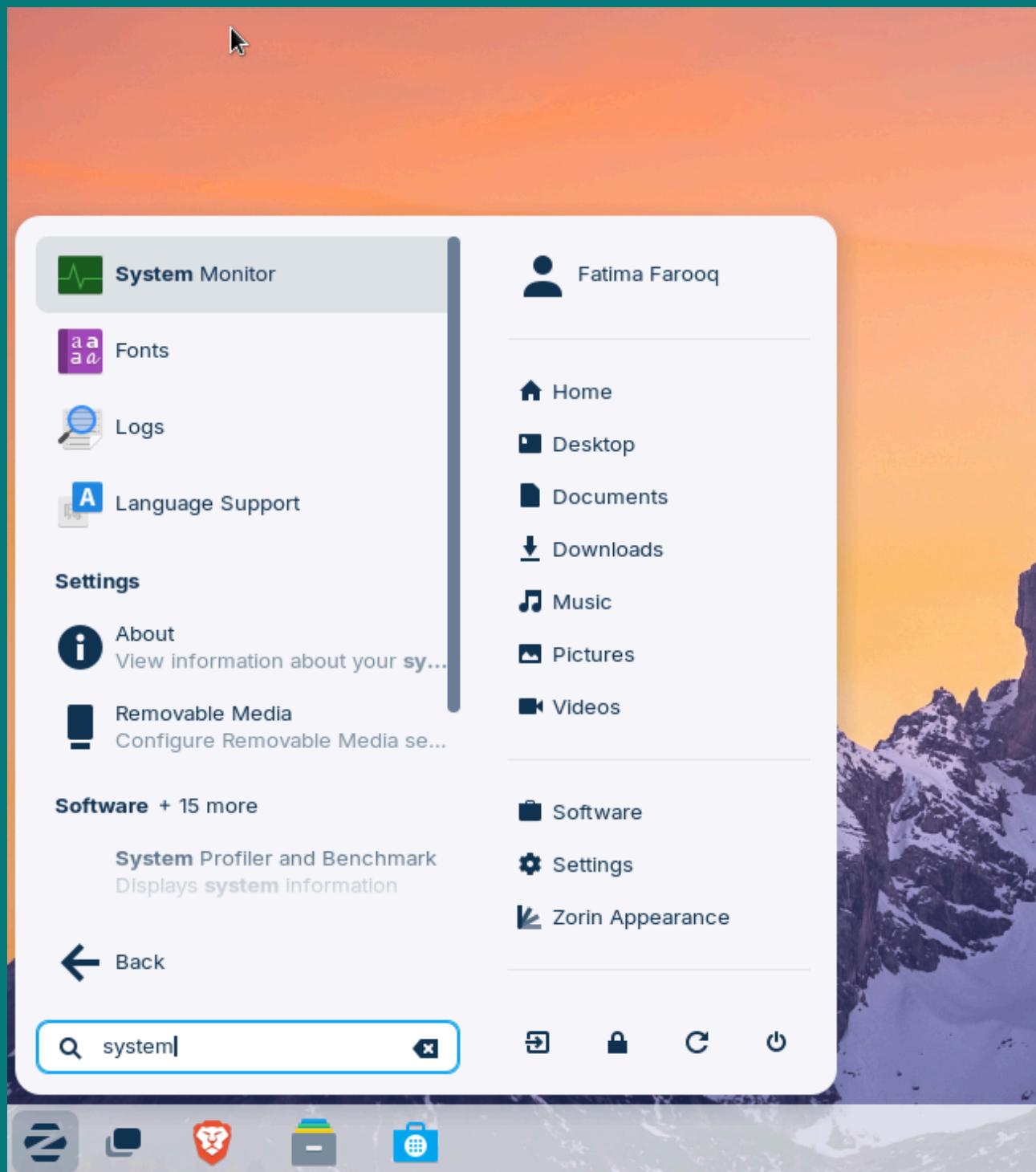
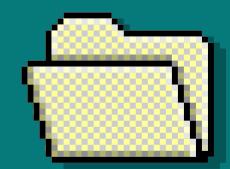


Finally, login

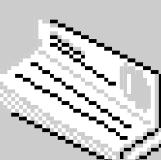
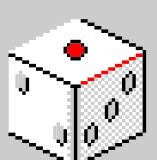


[Back to Agenda Page](#)

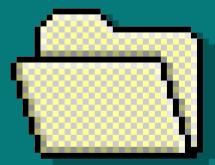
3. Install System Monitoring Tools



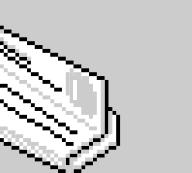
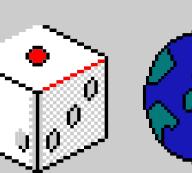
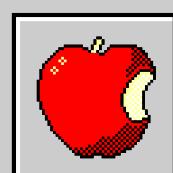
Already installed by default



Process Name	User	% CPU	ID	Memory	Disk read total	Disk write
at-spi2-registryd	julius	0.00	1241	655.4 kB	N/A	
at-spi-bus-launcher	julius	0.00	1102	786.4 kB	N/A	
dbus-daemon	julius	0.00	1012	2.1 MB	364.5 kB	
dbus-daemon	julius	0.00	1111	524.3 kB	N/A	
dconf-service	julius	0.00	1206	655.4 kB	77.8 kB	8.1 kB
evolution-addressbook-factory	julius	0.00	1213	10.2 MB	4.7 MB	36.1 kB
evolution-alarm-notify	julius	0.00	1344	15.9 MB	1.1 MB	
evolution-calendar-factory	julius	0.00	1197	10.4 MB	942.1 kB	
evolution-source-registry	julius	0.00	1151	9.7 MB	16.9 MB	
gdm-wayland-session	julius	0.00	1010	524.3 kB	4.1 kB	
gjs	julius	0.00	1240	5.2 MB	N/A	
gjs	julius	0.00	1450	5.3 MB	N/A	
gjs	julius	0.00	4594	14.0 MB	N/A	
gnome-calculator-search-provid	julius	0.00	4685	5.8 MB	N/A	
gnome-calendar	julius	0.00	4315	14.3 MB	1.2 MB	
gnome-control-center-search-p	julius	0.00	4679	5.9 MB	N/A	
gnome-keyring-daemon	julius	0.00	1005	1.1 MB	348.2 kB	24.1 kB
gnome-session-binary	julius	0.00	1014	2.0 MB	2.6 MB	
gnome-session-binary	julius	0.00	1079	3.0 MB	647.2 kB	4.1 kB

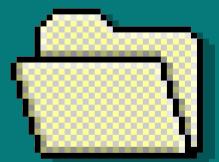


Opens the current running processes

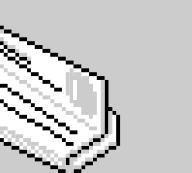
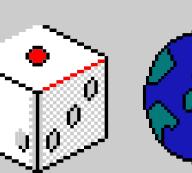
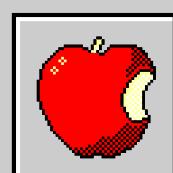


[Back to Agenda Page](#)

Process Name	User	% CPU	ID	Memory	Disk read total	Disk write
at-spi2-registryd	julius	0.00	1241	655.4 kB	N/A	
at-spi-bus-launcher	julius	0.00	1102	786.4 kB	N/A	
dbus-daemon	julius	0.00	1012	2.1 MB	364.5 kB	
dbus-daemon	julius	0.00	1111	524.3 kB	N/A	
dconf-service	julius	0.00	1206	655.4 kB	77.8 kB	8.1 kB
evolution-addressbook-factory	julius	0.00	1213	10.2 MB	4.7 MB	36.1 kB
evolution-alarm-notify	julius	0.00	1344	15.9 MB	1.1 MB	
evolution-calendar-factory	julius	0.00	1197	10.4 MB	942.1 kB	
evolution-source-registry	julius	0.00	1151	9.7 MB	16.9 MB	
gdm-wayland-session	julius	0.00	1010	524.3 kB	4.1 kB	
gjs	julius	0.00	1240	5.2 MB	N/A	
gjs	julius	0.00	1450	5.3 MB	N/A	
gjs	julius	0.00	4594	14.0 MB	N/A	
gnome-calculator-search-provid	julius	0.00	4685	5.8 MB	N/A	
gnome-calendar	julius	0.00	4315	14.3 MB	1.2 MB	
gnome-control-center-search-p	julius	0.00	4679	5.9 MB	N/A	
gnome-keyring-daemon	julius	0.00	1005	1.1 MB	348.2 kB	24.1 kB
gnome-session-binary	julius	0.00	1014	2.0 MB	2.6 MB	
gnome-session-binary	julius	0.00	1079	3.0 MB	647.2 kB	4.1 kB



Opens the current running processes



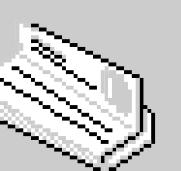
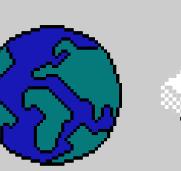
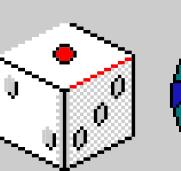
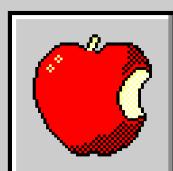
[Back to Agenda Page](#)

OVERVIEW OF THE RUNNING PROCESSES



System and Background Services

- dbus-daemon: this process aids the inter-process communication (IPC) framework for messaging and coordination utilized by many other processes.
- NetworkManager: it allows the managing of networking interfaces and ensuring connectivity, while making sure that the system has an established connection and can adjust to changes caused by network availability.
- systemd: This is the system and service manager of the Linux Operating Systems.
- cupsd: This process is responsible for handling printing services.
- Audio Server (pulseaudio or pipewire): This handles the stream mixing of audio, its routing and volume control for both the sounds of the system and the user applications.
- avahi-daemon: This provides a service discovery on a local network.
- sshd: This daemon Secure Shell (SSH) enables remote access.



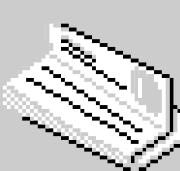
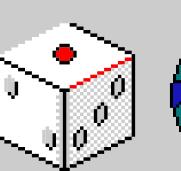
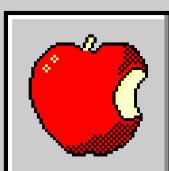
[Back to Agenda Page](#)

OVERVIEW OF THE RUNNING PROCESSES



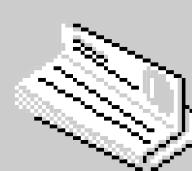
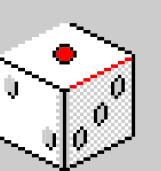
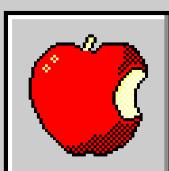
GNOME/Desktop Components

- **gnome-shell:** This is the core process of the GNOME desktop environment that takes care of the graphical interface, window management and the overall desktop composition. Because it is largely multi-threaded, it can separate UI rendering, input entry and other desktop functions.
- **gnome-session:** This process manages and deals with the GNOME session.
- **gnome-settings-daemon:** This process allows the implementation of different settings for the GNOME desktop.
- **gnome-shell-calendar-server:** This allows us to manage the calendar events in the GNOME shell.
- **gnome-software:** This is an application for Software Management.
- **GNOME Settings Daemons (gsd-*):** These incorporate a list of dedicated daemons which handle particular roles.
goa-daemon (GNOME Online Accounts Daemon): This makes the integration and synchronization of online accounts like Google, Facebook with GNOME applications which are usually running in user space.



[Back to Agenda Page](#)

OVERVIEW OF THE RUNNING PROCESSES



User Applications

- Evolution: This includes email, calendar, contacts and tasks which acts as a frontal implementation for management for personal information.
- evolution-data-server: For purposes of Evolution, it runs in the background to save and manage the data such as email, calendar and contacts.
- evolution-calendar-factory / evolution-addressbook-factory: By these processes, we can run and handle the required parts for calendars and contacts from the Evolution.

[Back to Agenda Page](#)

Resource Utilization

CPU



Most processes represent a near-zero CPU usage, which means that they are idle or in a waiting state, which is supportive that is because, I was not running any programs such as web browsers, or media players on the laptop at that moment. Processes like gnome-shell and gnome-settings-daemon represent a significant amount of CPU usage, however it is a continuous and low activity, which means that they are functioning to perform tasks that needs computational power.

Memory Footprint

There is a detectable change in the memory allocation. Lightweight Daemons require a minimum amount of memory such as a few megabytes. Nonetheless, applications that involve user interactions like web browsers or multimedia tools consume a larger memory. Through this distribution, we can highlight the efficiency of memory management so that important processes are placed an emphasis on response while user related requests and applications may have more varied resource demands. We see that gnome-shell uses a significant amount of the memory which is assertive of the fact that the graphical user interface along with other system functionalities needs to be maintained.



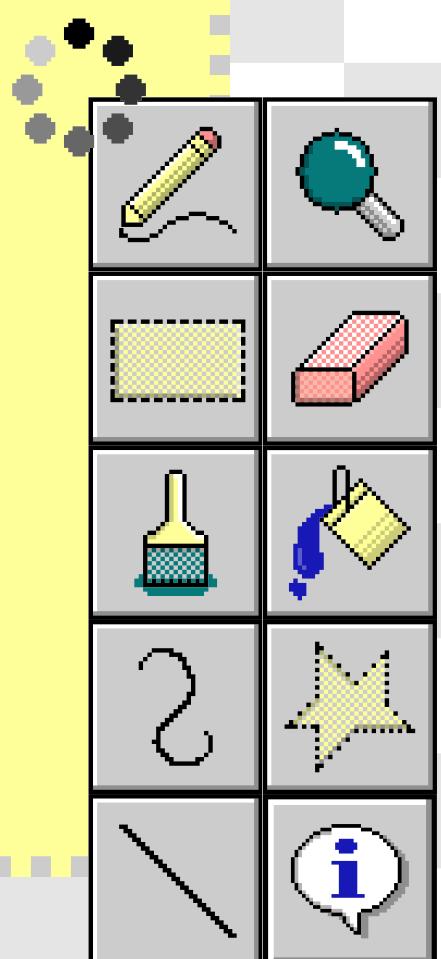
Disk I/O

Because of the several number of processes running in the background, these tend to be updated at regular cycles by the monitoring tools, because some processes may not show active disk usage. There can be temporary spikes or inactivity in the disk I/O which can signal which stage the process is at, reading or writing or even loading while performing tasks like file transfers or software updates

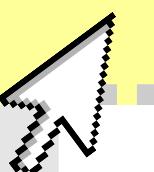




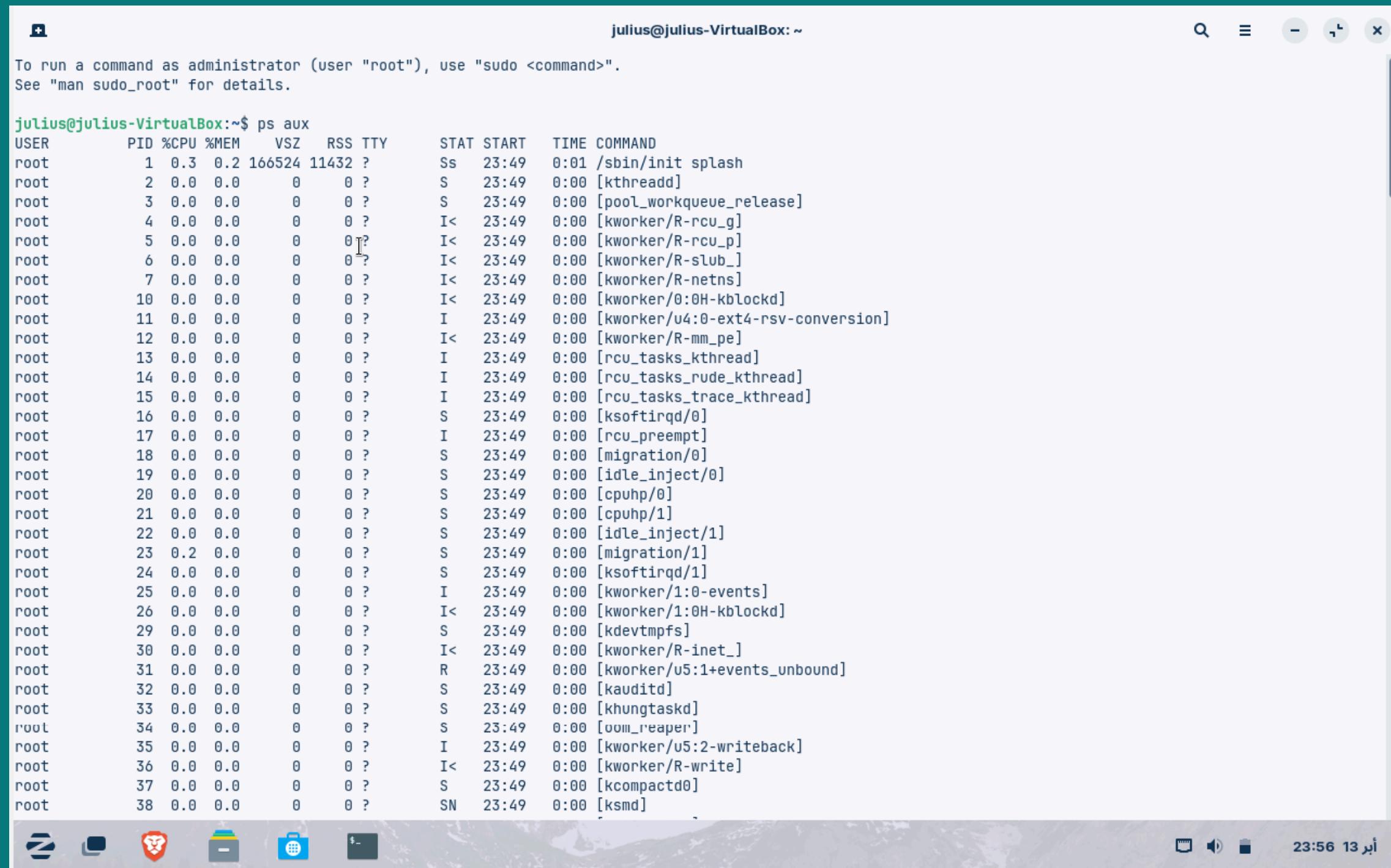
Terminal Implementation of Processes



[Back to Agenda Page](#)



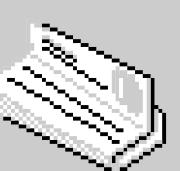
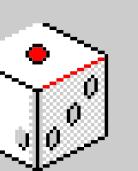
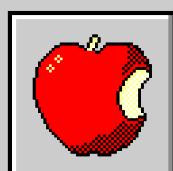
Process Status (ps): ps aux



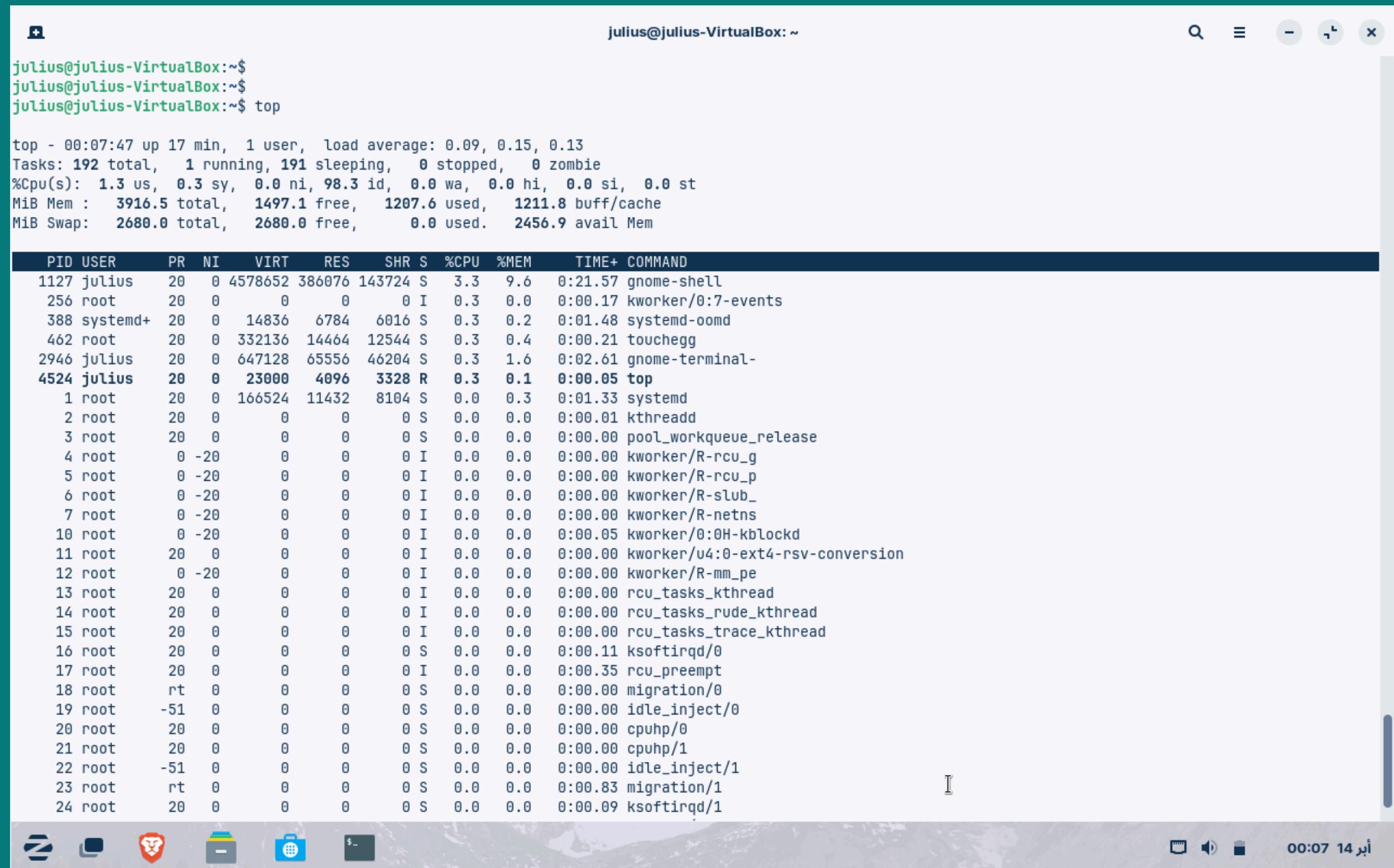
```
julius@julius-VirtualBox:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY STAT START  TIME COMMAND
root         1  0.3  0.2 166524 11432 ?
root         2  0.0  0.0     0    0 ?
root         3  0.0  0.0     0    0 ?
root         4  0.0  0.0     0    0 ?
root         5  0.0  0.0     0    0 ?
root         6  0.0  0.0     0    0 ?
root         7  0.0  0.0     0    0 ?
root        10  0.0  0.0     0    0 ?
root        11  0.0  0.0     0    0 ?
root        12  0.0  0.0     0    0 ?
root        13  0.0  0.0     0    0 ?
root        14  0.0  0.0     0    0 ?
root        15  0.0  0.0     0    0 ?
root        16  0.0  0.0     0    0 ?
root        17  0.0  0.0     0    0 ?
root        18  0.0  0.0     0    0 ?
root        19  0.0  0.0     0    0 ?
root        20  0.0  0.0     0    0 ?
root        21  0.0  0.0     0    0 ?
root        22  0.0  0.0     0    0 ?
root        23  0.2  0.0     0    0 ?
root        24  0.0  0.0     0    0 ?
root        25  0.0  0.0     0    0 ?
root        26  0.0  0.0     0    0 ?
root        29  0.0  0.0     0    0 ?
root        30  0.0  0.0     0    0 ?
root        31  0.0  0.0     0    0 ?
root        32  0.0  0.0     0    0 ?
root        33  0.0  0.0     0    0 ?
root        34  0.0  0.0     0    0 ?
root        35  0.0  0.0     0    0 ?
root        36  0.0  0.0     0    0 ?
root        37  0.0  0.0     0    0 ?
root        38  0.0  0.0     0    0 ?
```

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

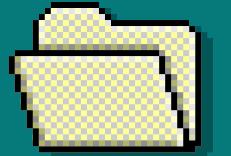
Shows all running processes in a detailed, user-friendly format, including background ones. Filter with grep: ps aux | grep gnome



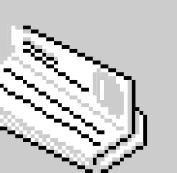
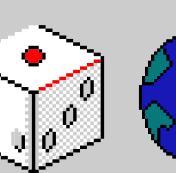
Real-Time Process Viewer: top



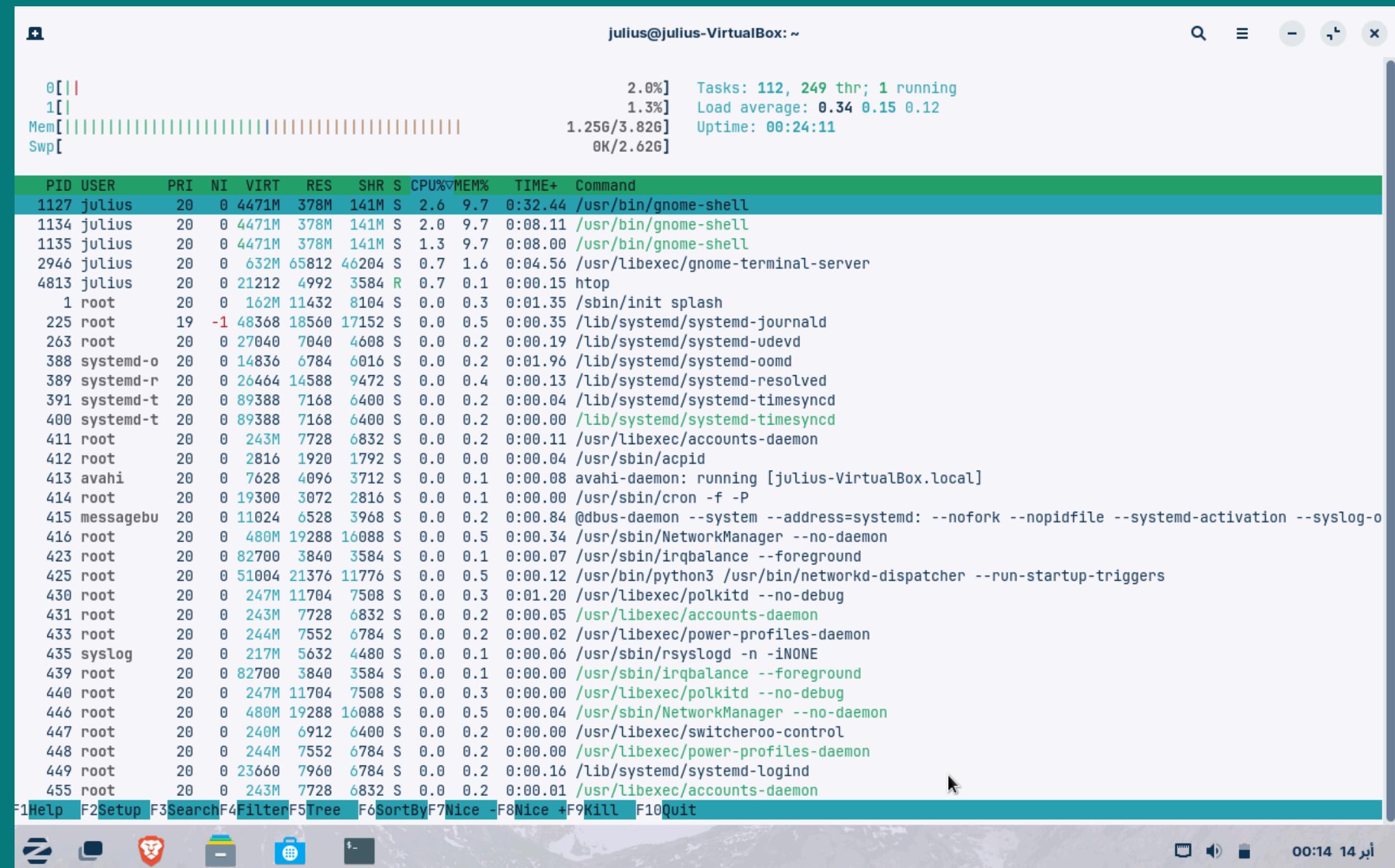
```
julius@julius-VirtualBox:~$  
julius@julius-VirtualBox:~$  
julius@julius-VirtualBox:~$ top  
  
top - 00:07:47 up 17 min,  1 user,  load average: 0.09, 0.15, 0.13  
Tasks: 192 total,   1 running, 191 sleeping,   0 stopped,   0 zombie  
%Cpu(s):  1.3 us,  0.3 sy,  0.0 ni, 98.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st  
MiB Mem : 3916.5 total, 1497.1 free, 1207.6 used, 1211.8 buff/cache  
MiB Swap: 2680.0 total, 2680.0 free,     0.0 used. 2456.9 avail Mem  
  
 PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND  
1127 julius    20   0 4578652 386076 143724 S  3.3  9.6 0:21.57 gnome-shell  
 256 root      20   0      0      0      0 I  0.3  0.0 0:00.17 kworker/0:7-events  
 388 systemd+  20   0 14836   6784   6016 S  0.3  0.2 0:01.48 systemd-oomd  
 462 root      20   0 332136 14464 12544 S  0.3  0.4 0:00.21 touchegg  
2946 julius    20   0 647128 65556 46204 S  0.3  1.6 0:02.61 gnome-terminal-  
4524 julius    20   0 23000   4096   3328 R  0.3  0.1 0:00.05 top  
  1 root      20   0 166524 11432   8104 S  0.0  0.3 0:01.33 systemd  
  2 root      20   0      0      0      0 S  0.0  0.0 0:00.01 kthreadd  
  3 root      20   0      0      0      0 S  0.0  0.0 0:00.00 pool_workqueue_release  
  4 root      0 -20   0      0      0 I  0.0  0.0 0:00.00 kworker/R-rCU_g  
  5 root      0 -20   0      0      0 I  0.0  0.0 0:00.00 kworker/R-rCU_p  
  6 root      0 -20   0      0      0 I  0.0  0.0 0:00.00 kworker/R-slub_  
  7 root      0 -20   0      0      0 I  0.0  0.0 0:00.00 kworker/R-netns  
 10 root      0 -20   0      0      0 I  0.0  0.0 0:00.05 kworker/0:0H-kblockd  
 11 root      20   0      0      0 I  0.0  0.0 0:00.00 kworker/u4:0-ext4-rsv-conversion  
 12 root      0 -20   0      0      0 I  0.0  0.0 0:00.00 kworker/R-mm_pe  
 13 root      20   0      0      0 I  0.0  0.0 0:00.00 rcu_tasks_kthread  
 14 root      20   0      0      0 I  0.0  0.0 0:00.00 rcu_tasks_rude_kthread  
 15 root      20   0      0      0 I  0.0  0.0 0:00.00 rcu_tasks_trace_kthread  
 16 root      20   0      0      0 S  0.0  0.0 0:00.11 ksoftirqd/0  
 17 root      20   0      0      0 I  0.0  0.0 0:00.35 rcu_preempt  
 18 root      rt  0      0      0 S  0.0  0.0 0:00.00 migration/0  
 19 root     -51  0      0      0 S  0.0  0.0 0:00.00 idle_inject/0  
 20 root      20   0      0      0 S  0.0  0.0 0:00.00 cpuhp/0  
 21 root      20   0      0      0 S  0.0  0.0 0:00.00 cpuhp/1  
 22 root     -51  0      0      0 S  0.0  0.0 0:00.00 idle_inject/1  
 23 root      rt  0      0      0 S  0.0  0.0 0:00.83 migration/1  
 24 root      20   0      0      0 S  0.0  0.0 0:00.09 ksoftirqd/1
```



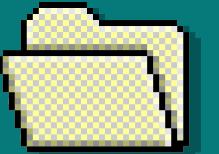
Shows live CPU and memory usage for processes. It refreshes every few seconds.



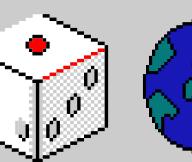
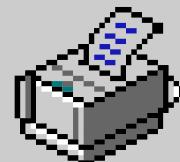
Improved top Interface:htop



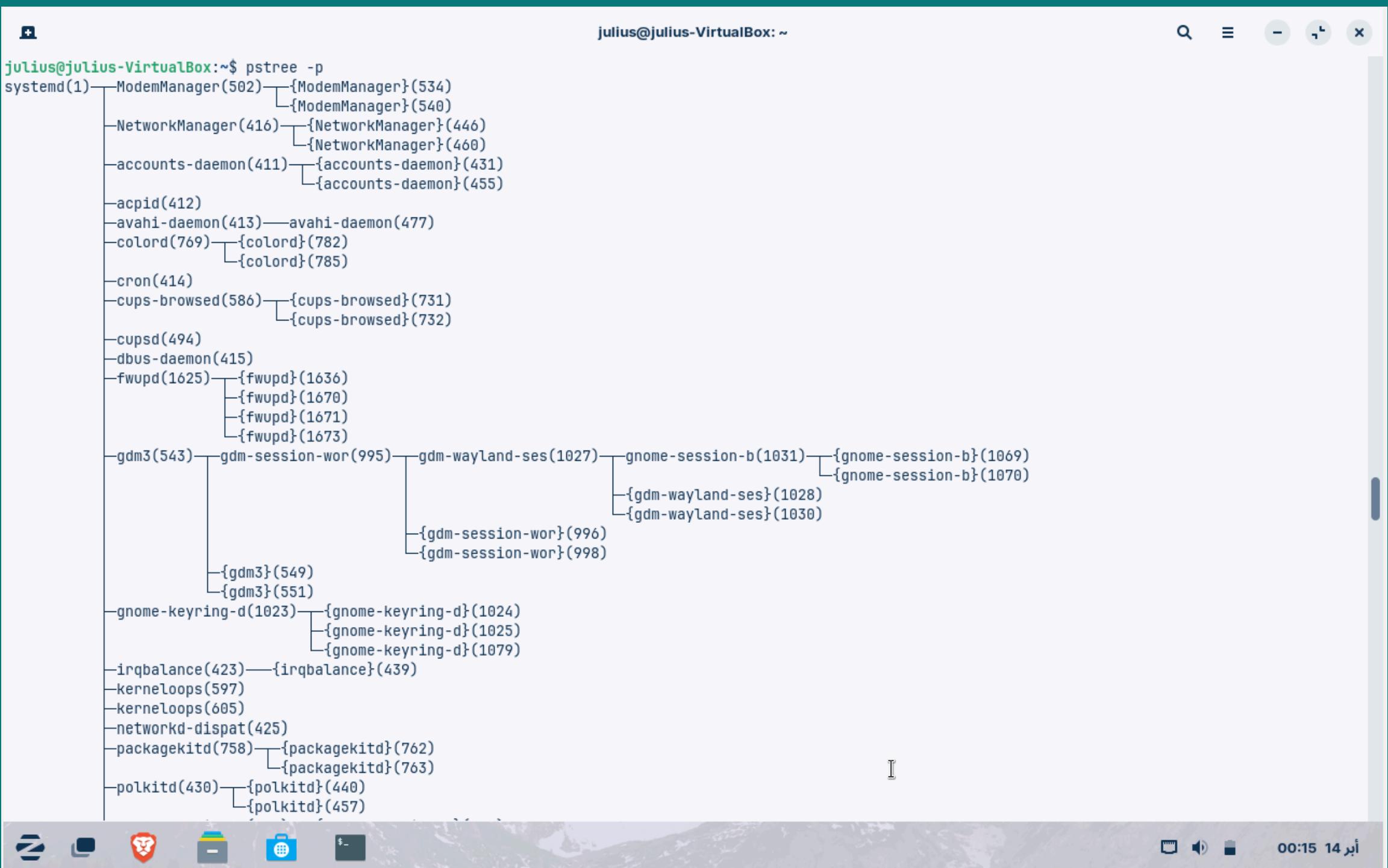
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	%MEM%	TIME+	Command
1127	julius	20	0	4471M	378M	141M	S	2.6	9.7	0:32.44	/usr/bin/gnome-shell
1134	julius	20	0	4471M	378M	141M	S	2.0	9.7	0:08.11	/usr/bin/gnome-shell
1135	julius	20	0	4471M	378M	141M	S	1.3	9.7	0:08.00	/usr/bin/gnome-shell
2946	julius	20	0	632M	65812	46204	S	0.7	1.6	0:04.56	/usr/libexec/gnome-terminal-server
4813	julius	20	0	21212	4992	3584	R	0.7	0.1	0:00.15	htop
1	root	20	0	162M	11432	8104	S	0.0	0.3	0:01.35	/sbin/init splash
225	root	19	-1	48368	18560	17152	S	0.0	0.5	0:00.35	/lib/systemd/systemd-journald
263	root	20	0	27040	7040	4608	S	0.0	0.2	0:00.19	/lib/systemd/systemd-udevd
388	systemd-o	20	0	14836	6784	6016	S	0.0	0.2	0:01.96	/lib/systemd/systemd-oomd
389	systemd-r	20	0	26464	14588	9472	S	0.0	0.4	0:00.13	/lib/systemd/systemd-resolved
391	systemd-t	20	0	89388	7168	6400	S	0.0	0.2	0:00.04	/lib/systemd/systemd-timesyncd
400	systemd-t	20	0	89388	7168	6400	S	0.0	0.2	0:00.00	/lib/systemd/systemd-timesyncd
411	root	20	0	243M	7728	6832	S	0.0	0.2	0:00.11	/usr/libexec/accounts-daemon
412	root	20	0	2816	1920	1792	S	0.0	0.0	0:00.04	/usr/sbin/acpid
413	avahi	20	0	7628	4096	3712	S	0.0	0.1	0:00.08	avahi-daemon: running [julius-VirtualBox.local]
414	root	20	0	19300	3072	2816	S	0.0	0.1	0:00.00	/usr/sbin/cron -f -P
415	messagebus	20	0	11024	6528	3968	S	0.0	0.2	0:00.84	@dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-o
416	root	20	0	480M	19288	16088	S	0.0	0.5	0:00.34	/usr/sbin/NetworkManager --no-daemon
423	root	20	0	82700	3840	3584	S	0.0	0.1	0:00.07	/usr/sbin/irqbalance --foreground
425	root	20	0	51004	21376	11776	S	0.0	0.5	0:00.12	/usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
430	root	20	0	247M	11704	7508	S	0.0	0.3	0:01.20	/usr/libexec/polkitd --no-debug
431	root	20	0	243M	7728	6832	S	0.0	0.2	0:00.05	/usr/libexec/accounts-daemon
433	root	20	0	244M	7552	6784	S	0.0	0.2	0:00.02	/usr/libexec/power-profiles-daemon
435	syslog	20	0	217M	5632	4480	S	0.0	0.1	0:00.06	/usr/sbin/rsyslogd -n -iNONE
439	root	20	0	82700	3840	3584	S	0.0	0.1	0:00.00	/usr/sbin/irqbalance --foreground
440	root	20	0	247M	11704	7508	S	0.0	0.3	0:00.00	/usr/libexec/polkitd --no-debug
446	root	20	0	480M	19288	16088	S	0.0	0.5	0:00.04	/usr/sbin/NetworkManager --no-daemon
447	root	20	0	240M	6912	6400	S	0.0	0.2	0:00.00	/usr/libexec/switcheroo-control
448	root	20	0	244M	7552	6784	S	0.0	0.2	0:00.00	/usr/libexec/power-profiles-daemon
449	root	20	0	23660	7960	6784	S	0.0	0.2	0:00.16	/lib/systemd/systemd-logind
455	root	20	0	243M	7728	6832	S	0.0	0.2	0:00.01	/usr/libexec/accounts-daemon



Provides a colorful, interactive process list. It even allows you to scroll, search (/), and sort by CPU/memory.

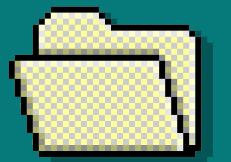
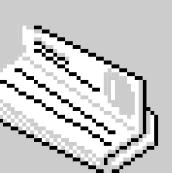
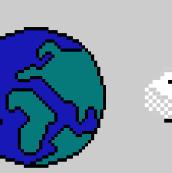
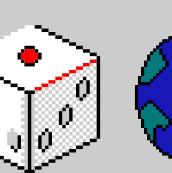
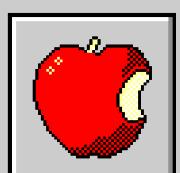


Process Hierarchy: pstree -p



```
julius@julius-VirtualBox:~$ pstree -p
systemd(1)─ ModemManager(502) ──{ModemManager}(534)
              └──{ModemManager}(540)
              └──NetworkManager(416) ──{NetworkManager}(446)
              └──{NetworkManager}(460)
              └──accounts-daemon(411) ──{accounts-daemon}(431)
              └──{accounts-daemon}(455)
              └──acpid(412)
              └──avahi-daemon(413) ──avahi-daemon(477)
              └──colord(769) ──{colord}(782)
              └──{colord}(785)
              └──cron(414)
              └──cups-browsed(586) ──{cups-browsed}(731)
              └──{cups-browsed}(732)
              └──cupsd(494)
              └──dbus-daemon(415)
              └──fwupd(1625) ──{fwupd}(1636)
              └──{fwupd}(1670)
              └──{fwupd}(1671)
              └──{fwupd}(1673)
              └──gdm3(543) ──gdm-session-wor(995) ──gdm-wayland-ses(1027) ──gnome-session-b(1031) ──{gnome-session-b}(1069)
              └──{gnome-session-b}(1070)
              └──{gdm-wayland-ses}(1028)
              └──{gdm-wayland-ses}(1030)
              └──{gdm-session-wor}(996)
              └──{gdm-session-wor}(998)
              └──{gdm3}(549)
              └──{gdm3}(551)
              └──gnome-keyring-d(1023) ──{gnome-keyring-d}(1024)
              └──{gnome-keyring-d}(1025)
              └──{gnome-keyring-d}(1079)
              └──irqbalance(423) ──{irqbalance}(439)
              └──kerneloops(597)
              └──kerneloops(605)
              └──networkd-dispat(425)
              └──packagekitd(758) ──{packagekitd}(762)
              └──{packagekitd}(763)
              └──polkitd(430) ──{polkitd}(440)
              └──{polkitd}(457)
```

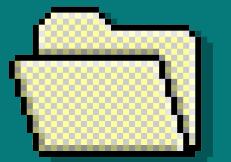
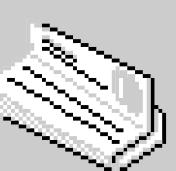
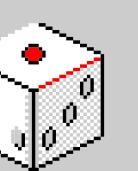
Shows a tree view of all processes



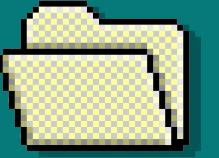
Viewing_Daemons:systemctl list-units --type=service}

```
julius@julius-VirtualBox: ~
command 'systemctl' from deb systemd (249.11-0ubuntu3.12)
command 'systemctl' from deb systemctl (1.4.4181-1.1)
Try: sudo apt install <deb name>
julius@julius-VirtualBox:~$ systemctl list-units --type=service
 _UNIT           LOAD   ACTIVE   SUB   DESCRIPTION
accounts-daemon.service loaded active running Accounts Service
acpid.service      loaded active running ACPI event daemon
alsa-restore.service loaded active exited Save/Restore Sound Card State
apparmor.service    loaded active exited Load AppArmor profiles
avahi-daemon.service loaded active running Avahi mDNS/DNS-SD Stack
colord.service     loaded active running Manage, Install and Generate Color Profiles
console-setup.service loaded active exited Set console font and keymap
cron.service       loaded active running Regular background program processing daemon
cups-browsed.service loaded active running Make remote CUPS printers available locally
cups.service        loaded active running CUPS Scheduler
dbus.service        loaded active running D-Bus System Message Bus
fwupd.service       loaded active running Firmware update daemon
gdm.service         loaded active running GNOME Display Manager
irqbalance.service loaded active running irqbalance daemon
kerneloops.service loaded active running Tool to automatically collect and submit kernel crash signatures
keyboard-setup.service loaded active exited Set the console keyboard layout
kmod-static-nodes.service loaded active exited Create List of Static Device Nodes
ModemManager.service loaded active running Modem Manager
networkd-dispatcher.service loaded active running Dispatcher daemon for systemd-networkd
NetworkManager-wait-online.service loaded active exited Network Manager Wait Online
NetworkManager.service loaded active running Network Manager
openvpn.service     loaded active exited OpenVPN service
packagekit.service  loaded active running PackageKit Daemon
plymouth-quit-wait.service loaded active exited Hold until boot process finishes up
plymouth-read-write.service loaded active exited Tell Plymouth To Write Out Runtime Data
plymouth-start.service loaded active exited Show Plymouth Boot Screen
polkit.service      loaded active running Authorization Manager
power-profiles-daemon.service loaded active running Power Profiles daemon
rsyslog.service     loaded active running System Logging Service
rtkit-daemon.service loaded active running RealtimeKit Scheduling Policy Service
setvtrgb.service   loaded active exited Set console scheme
snapd.apparmor.service loaded active exited Load AppArmor profiles managed internally by snapd
snapd.seeded.service loaded active exited Wait until snapd is fully seeded
switcheroo-control.service loaded active running Switcheroo Control Proxy service
```

Lists all the system services and their statuses



Specific Daemon: systemctl status NetworkManager.service

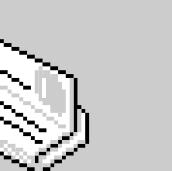
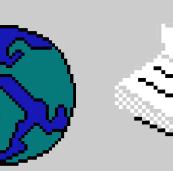
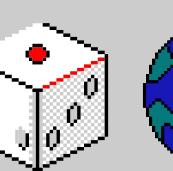
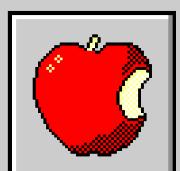


```
julius@julius-VirtualBox: ~
systemd-fsck@dev-disk-by\x2duuid-60D3\x2d41D3.service loaded active exited  File System Check on /dev/disk/by-uuid/60D3-41D3
systemd-journal-flush.service loaded active exited  Flush Journal to Persistent Storage
systemd-journald.service loaded active running Journal Service
systemd-logind.service loaded active running User Login Management
systemd-modules-load.service loaded active exited Load Kernel Modules
systemd-oomd.service loaded active running Userspace Out-Of-Memory (OOM) Killer
systemd-random-seed.service loaded active exited Load/Save Random Seed
systemd-remount-fs.service loaded active exited Remount Root and Kernel File Systems
lines 7-44

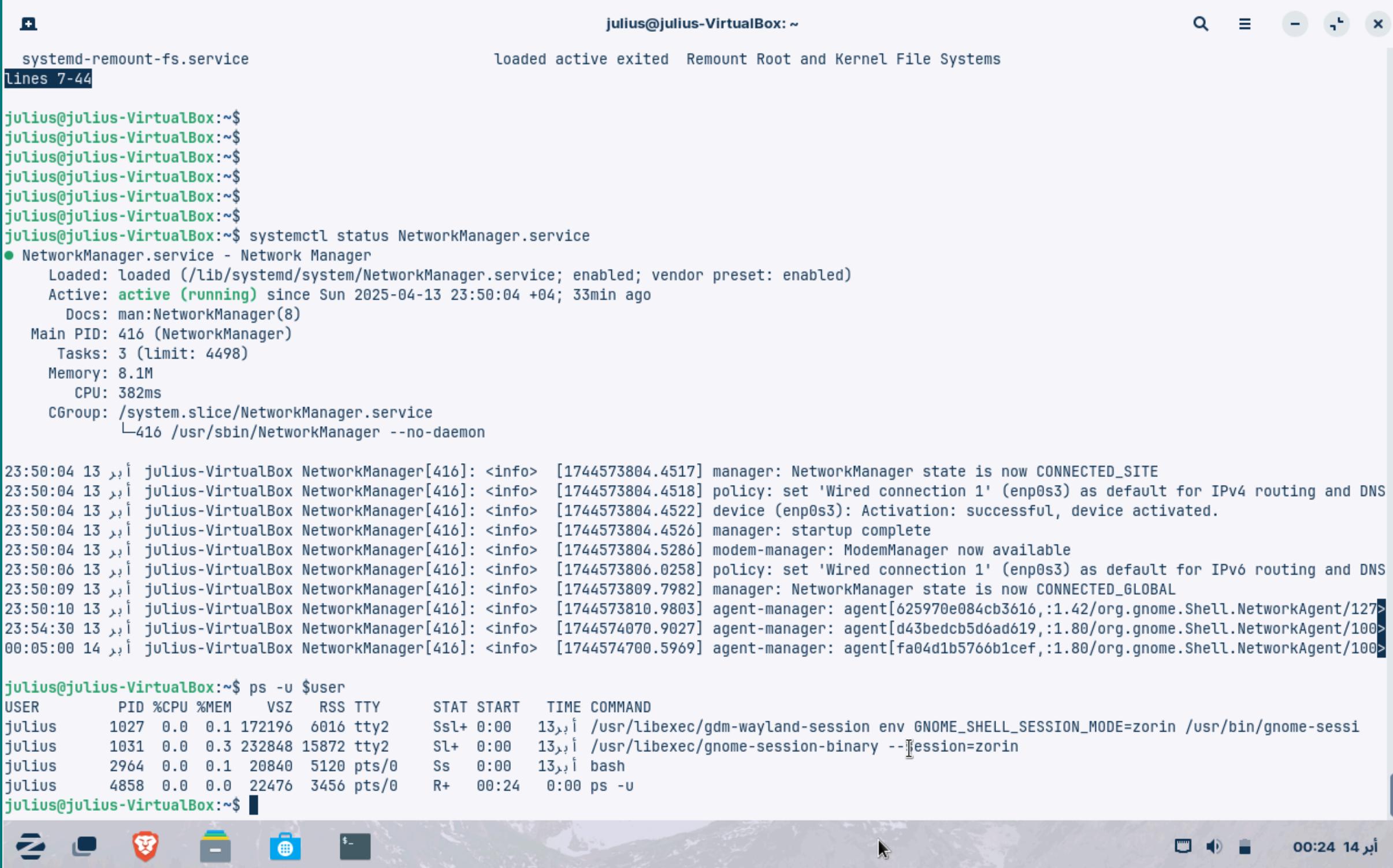
julius@julius-VirtualBox: ~$ 
julius@julius-VirtualBox: ~$ systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-04-13 23:50:04 +04; 33min ago
     Docs: man:NetworkManager(8)
     Main PID: 416 (NetworkManager)
       Tasks: 3 (limit: 4498)
      Memory: 8.1M
        CPU: 382ms
       CGroup: /system.slice/NetworkManager.service
               └─416 /usr/sbin/NetworkManager --no-daemon

23:50:04 13 آبری julius-VirtualBox NetworkManager[416]: <info> [1744573804.4517] manager: NetworkManager state is now CONNECTED_SITE
23:50:04 13 آبری julius-VirtualBox NetworkManager[416]: <info> [1744573804.4518] policy: set 'Wired connection 1' (enp0s3) as default for IPv4 routing and DNS
23:50:04 13 آبری julius-VirtualBox NetworkManager[416]: <info> [1744573804.4522] device (enp0s3): Activation: successful, device activated.
23:50:04 13 آبری julius-VirtualBox NetworkManager[416]: <info> [1744573804.4526] manager: startup complete
23:50:04 13 آبری julius-VirtualBox NetworkManager[416]: <info> [1744573804.5286] modem-manager: ModemManager now available
23:50:06 13 آبری julius-VirtualBox NetworkManager[416]: <info> [1744573806.0258] policy: set 'Wired connection 1' (enp0s3) as default for IPv6 routing and DNS
23:50:09 13 آبری julius-VirtualBox NetworkManager[416]: <info> [1744573809.7982] manager: NetworkManager state is now CONNECTED_GLOBAL
23:50:10 13 آبری julius-VirtualBox NetworkManager[416]: <info> [1744573810.9803] agent-manager: agent[6259f0e084cb3616,:1.42/org.gnome.Shell.NetworkAgent/127>
23:54:30 13 آبری julius-VirtualBox NetworkManager[416]: <info> [1744574070.9027] agent-manager: agent[d43b6dc5d6ad619,:1.80/org.gnome.Shell.NetworkAgent/100>
00:05:00 14 آبری julius-VirtualBox NetworkManager[416]: <info> [1744574700.5969] agent-manager: agent[fa04d1b5766b1cef,:1.80/org.gnome.Shell.NetworkAgent/100>
lines 1-21/21 (END)
```

Checks background daemons not shown in GUI



Checks Processes for Current User: ps -u \$USER

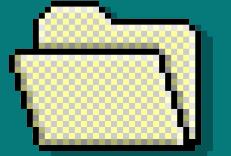
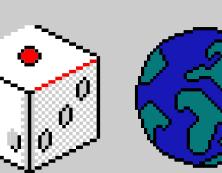
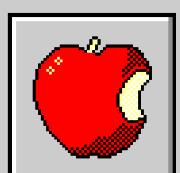


The screenshot shows a terminal window titled "julius@julius-VirtualBox:~". It displays the output of several commands:

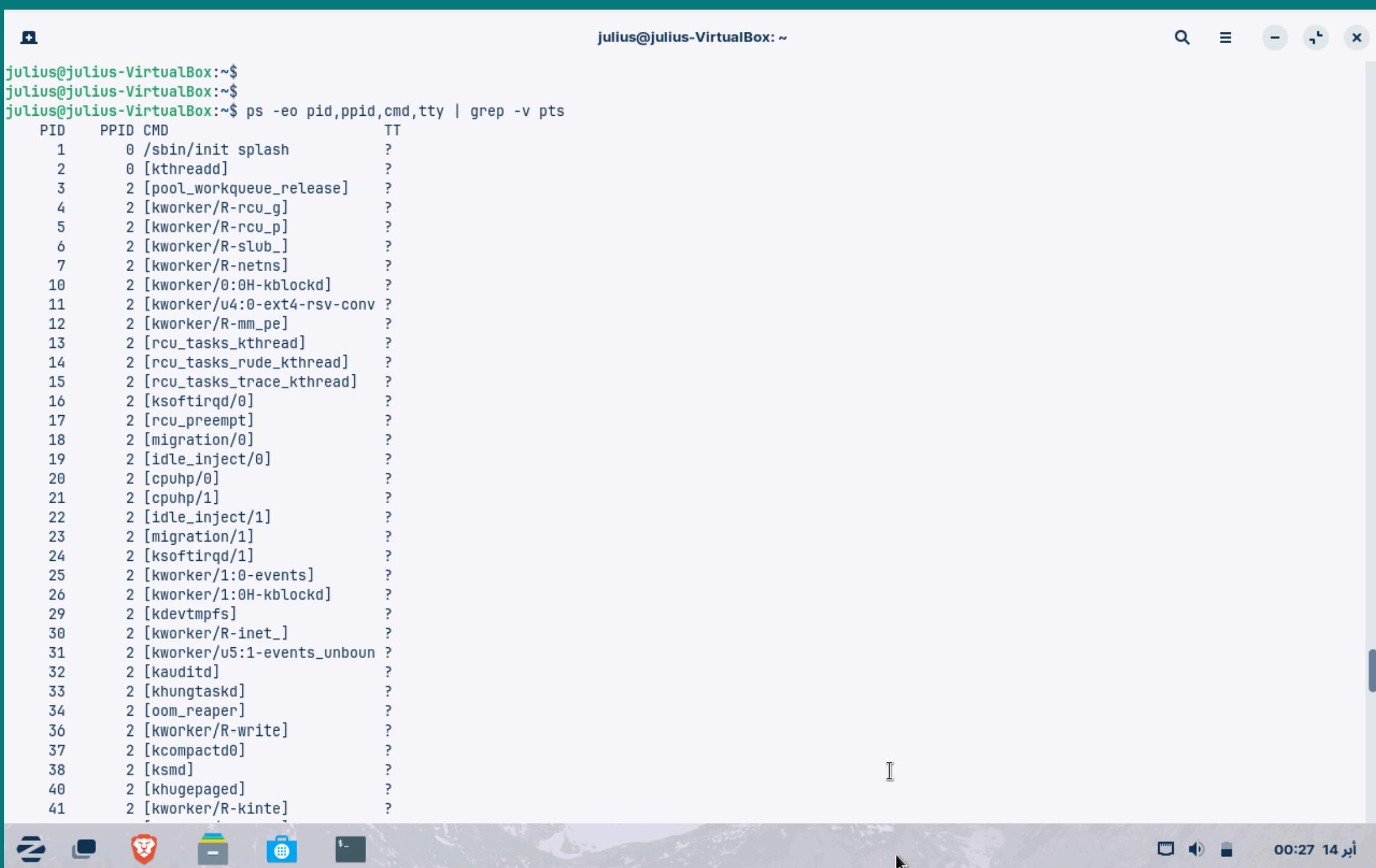
- `systemctl status NetworkManager.service`: Shows the Network Manager service is active (running) since Sun 2025-04-13 23:50:04 +04; 33min ago. It provides details like Main PID, Tasks, Memory, CPU usage, and CGroup.
- `journalctl`: Shows log entries from the NetworkManager service, including activation of device (enp0s3), startup complete, and policy changes for IPv4 and IPv6 routing and DNS.
- `ps -u $USER`: Shows a list of processes running under the current user "julius". The output includes columns for USER, PID, %CPU, %MEM, VSZ, RSS, TTY, STAT, START, TIME, and COMMAND. The processes listed are:

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
julius	1027	0.0	0.1	172196	6016	tty2	Ssl+	0:00	13:24	/usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=zorin /usr/bin/gnome-sessi
julius	1031	0.0	0.3	232848	15872	tty2	Sl+	0:00	13:24	/usr/libexec/gnome-session-binary --session=zorin
julius	2964	0.0	0.1	20840	5120	pts/0	Ss	0:00	13:24	bash
julius	4858	0.0	0.0	22476	3456	pts/0	R+	00:24	0:00	ps -u

Shows only the processes running under your current user julius

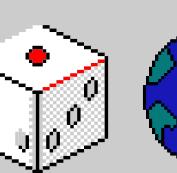


View Active Daemons:`ps -eo pid,ppid,cmd,tty | grep -v pts`



```
julius@julius-VirtualBox:~$  
julius@julius-VirtualBox:~$  
julius@julius-VirtualBox:~$ ps -eo pid,ppid,cmd,tty | grep -v pts  
PID  PPID CMD          TTY  
 1    0 /sbin/init splash      ?  
 2    0 [kthreadd]           ?  
 3    2 [pool_workqueue_release] ?  
 4    2 [kworker/R-rcu_g]       ?  
 5    2 [kworker/R-rcu_p]       ?  
 6    2 [kworker/R-slub_]      ?  
 7    2 [kworker/R-netns]      ?  
10   2 [kworker/0:0H-kblockd]  ?  
11   2 [kworker/u4:0-ext4-rsv-conv] ?  
12   2 [kworker/R-mm_pe]      ?  
13   2 [rcu_tasks_kthread]    ?  
14   2 [rcu_tasks_rude_kthread] ?  
15   2 [rcu_tasks_trace_kthread] ?  
16   2 [ksoftirqd/0]          ?  
17   2 [rcu_preempt]          ?  
18   2 [migration/0]          ?  
19   2 [idle_inject/0]         ?  
20   2 [cpuhp/0]              ?  
21   2 [cpuhp/1]              ?  
22   2 [idle_inject/1]         ?  
23   2 [migration/1]          ?  
24   2 [ksoftirqd/1]          ?  
25   2 [kworker/1:0-events]   ?  
26   2 [kworker/1:0H-kblockd]  ?  
29   2 [kdevtmpfs]            ?  
30   2 [kworker/R-inet_]      ?  
31   2 [kworker/u5:1-events_unbound] ?  
32   2 [kauditfd]             ?  
33   2 [khungtaskd]           ?  
34   2 [oom_reaper]            ?  
36   2 [kworker/R-write]      ?  
37   2 [kcompactd0]            ?  
38   2 [ksmd]                  ?  
40   2 [khugepaged]           ?  
41   2 [kworker/R-kint]        ?
```

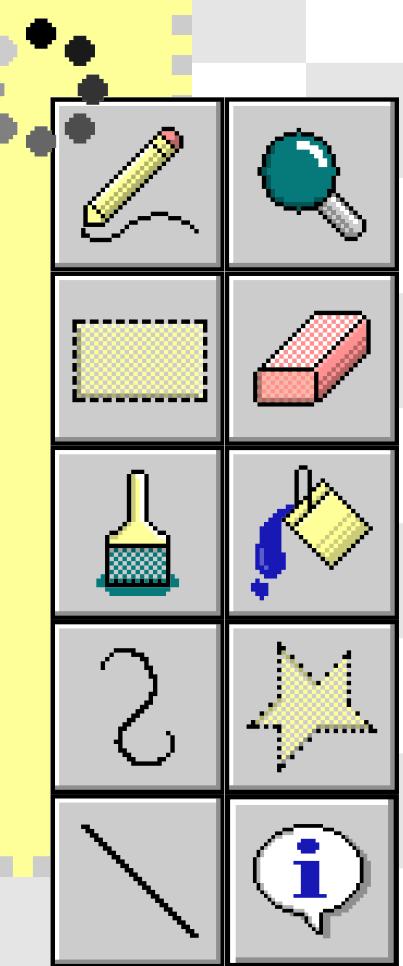
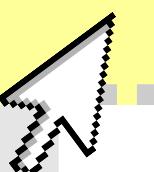
Filters out user terminal processes and shows only the Daemons



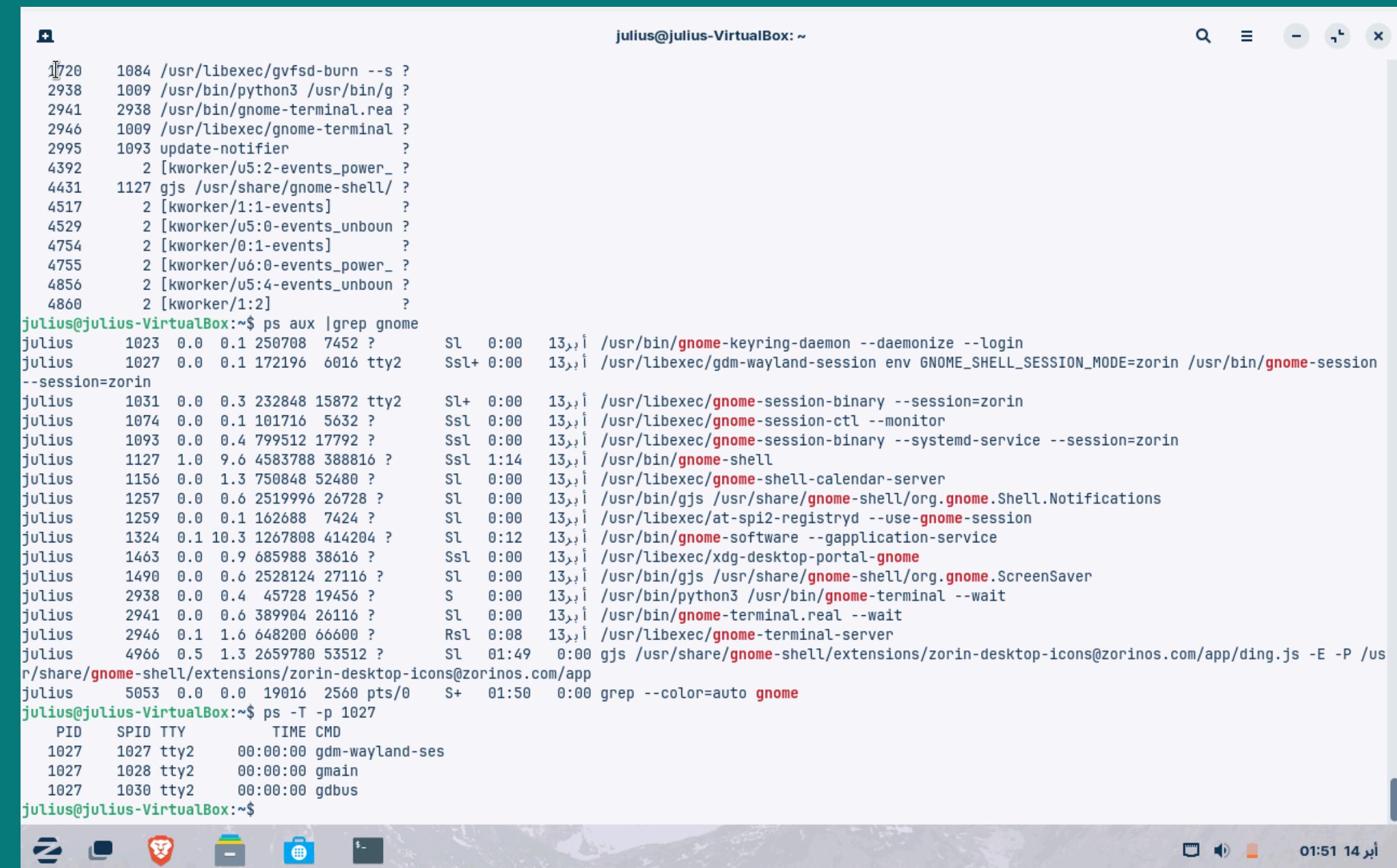


Terminal Implementation of Threads

[Back to Agenda Page](#)



Find the PID (Process ID):`ps aux | grep gnome` then find the Thread using `ps -T -p <PID>`



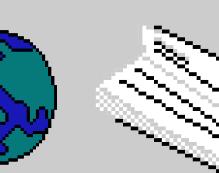
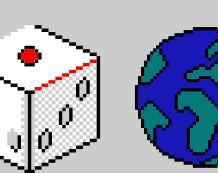
The screenshot shows a terminal window titled "julius@julius-VirtualBox: ~". It displays the output of several command-line operations:

- `ps aux` command output showing many processes, including several with "gnome" in their names.
- `ps aux | grep gnome` command output, which filters the processes listed above, showing more details for those related to the GNOME desktop environment.
- `ps -T -p 1027` command output, which lists the threads (LWPs) for the process with PID 1027.

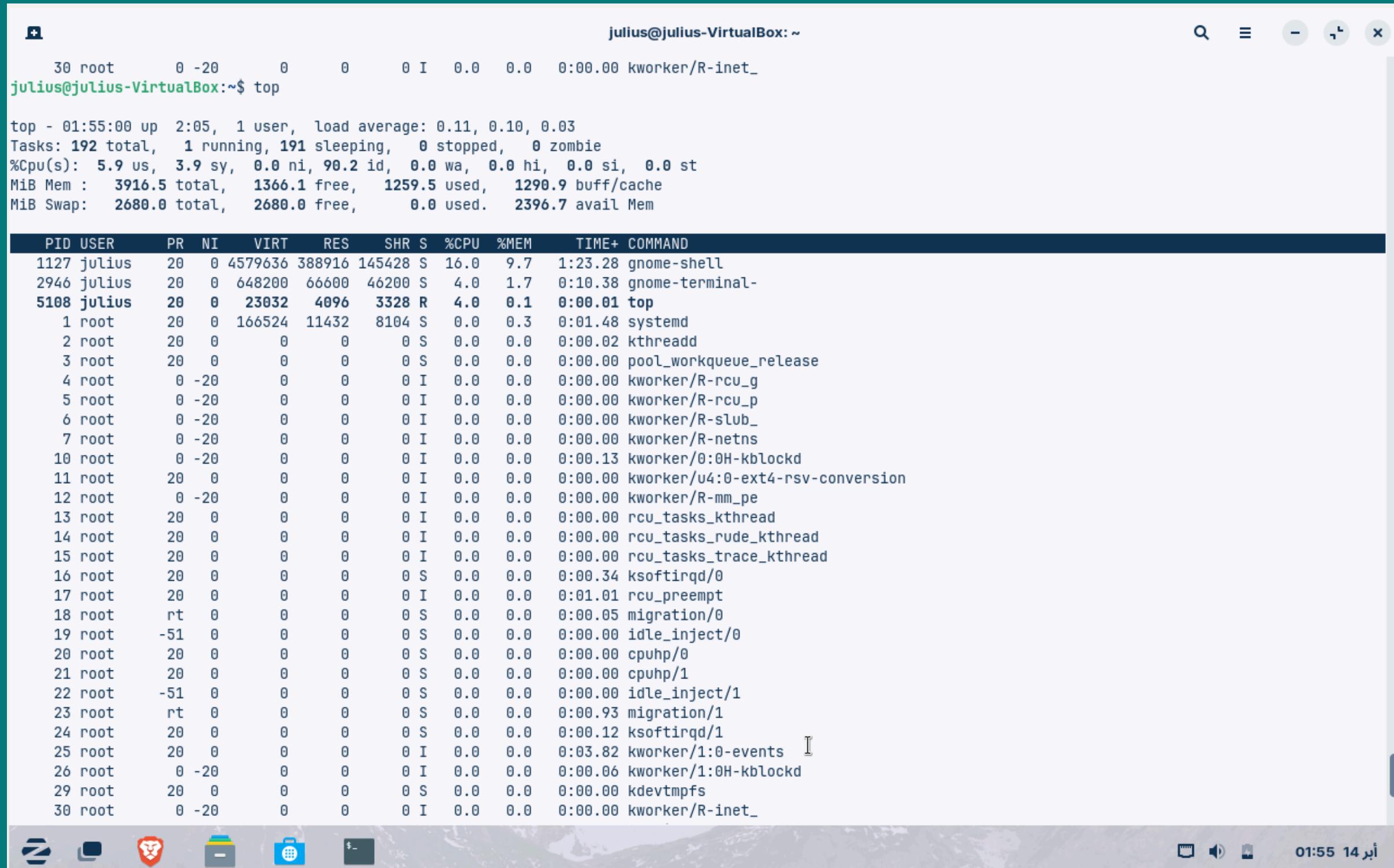
The terminal window has a light gray background and a dark gray border. The command history at the bottom shows the user's session. The system tray at the bottom right indicates the date and time as 01:51 14 Jun.

```
julius@julius-VirtualBox:~$ ps aux
julius 1084 0.0 0.1 250708 7452 ? Sl 0:00 13,?i /usr/bin/gnome-keyring-daemon --daemonize --login
julius 1023 0.0 0.1 250708 7452 ? Ssl+ 0:00 13,?i /usr/libexec/gdm-wayland-session env GNOME_SESSION_MODE=zorin /usr/bin/gnome-session
--session=zorin
julius 1031 0.0 0.3 232848 15872 tty2 Sl+ 0:00 13,?i /usr/libexec/gnome-session-binary --session=zorin
julius 1074 0.0 0.1 101716 5632 ? Ssl 0:00 13,?i /usr/libexec/gnome-session-ctl --monitor
julius 1093 0.0 0.4 799512 17792 ? Ssl 0:00 13,?i /usr/libexec/gnome-session-binary --systemd-service --session=zorin
julius 1127 1.0 9.6 4583788 388816 ? Ssl 1:14 13,?i /usr/bin/gnome-shell
julius 1156 0.0 1.3 750848 52480 ? Sl 0:00 13,?i /usr/libexec/gnome-shell-calendar-server
julius 1257 0.0 0.6 2519996 26728 ? Sl 0:00 13,?i /usr/bin/gjs /usr/share/gnome-shell/org.gnome.Shell.Notifications
julius 1259 0.0 0.1 162688 7424 ? Sl 0:00 13,?i /usr/libexec/at-spi2-registryd --use-gnome-session
julius 1324 0.1 10.3 1267808 414204 ? Sl 0:12 13,?i /usr/bin/gnome-software --gapplication-service
julius 1463 0.0 0.9 685988 38616 ? Ssl 0:00 13,?i /usr/libexec/xdg-desktop-portal-gnome
julius 1490 0.0 0.6 2528124 27116 ? Sl 0:00 13,?i /usr/bin/gjs /usr/share/gnome-shell/org.gnome.ScreenSaver
julius 2938 0.0 0.4 45728 19456 ? S 0:00 13,?i /usr/bin/python3 /usr/bin/gnome-terminal --wait
julius 2941 0.0 0.6 389904 26116 ? Sl 0:00 13,?i /usr/bin/gnome-terminal.real --wait
julius 2946 0.1 1.6 648200 66600 ? Rsl 0:08 13,?i /usr/libexec/gnome-terminal-server
julius 4966 0.5 1.3 2659780 53512 ? Sl 0:1:49 0:00 gjs /usr/share/gnome-shell/extensions/zorin-desktop-icons@zorinos.com/app/ding.js -E -P /us
r/share/gnome-shell/extensions/zorin-desktop-icons@zorinos.com/app
julius 5053 0.0 0.0 19016 2560 pts/0 S+ 0:1:50 0:00 grep --color=auto gnome
julius@julius-VirtualBox:~$ ps -T -p 1027
 PID  SPID TTY      TIME CMD
 1027  1027 tty2    00:00:00 gdm-wayland-ses
 1027  1028 tty2    00:00:00 gmain
 1027  1030 tty2    00:00:00 gibus
julius@julius-VirtualBox:~$
```

This will list all threads known as LWPs (Light Weight Processes) under that process



Using top to View Threads, start with top then press H

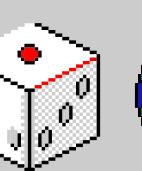
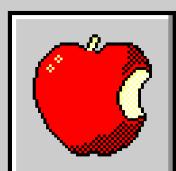


```
julius@julius-VirtualBox: ~
30 root      0 -20      0      0      0 I    0.0    0.0  0:00.00 kworker/R-inet_
julius@julius-VirtualBox:~$ top

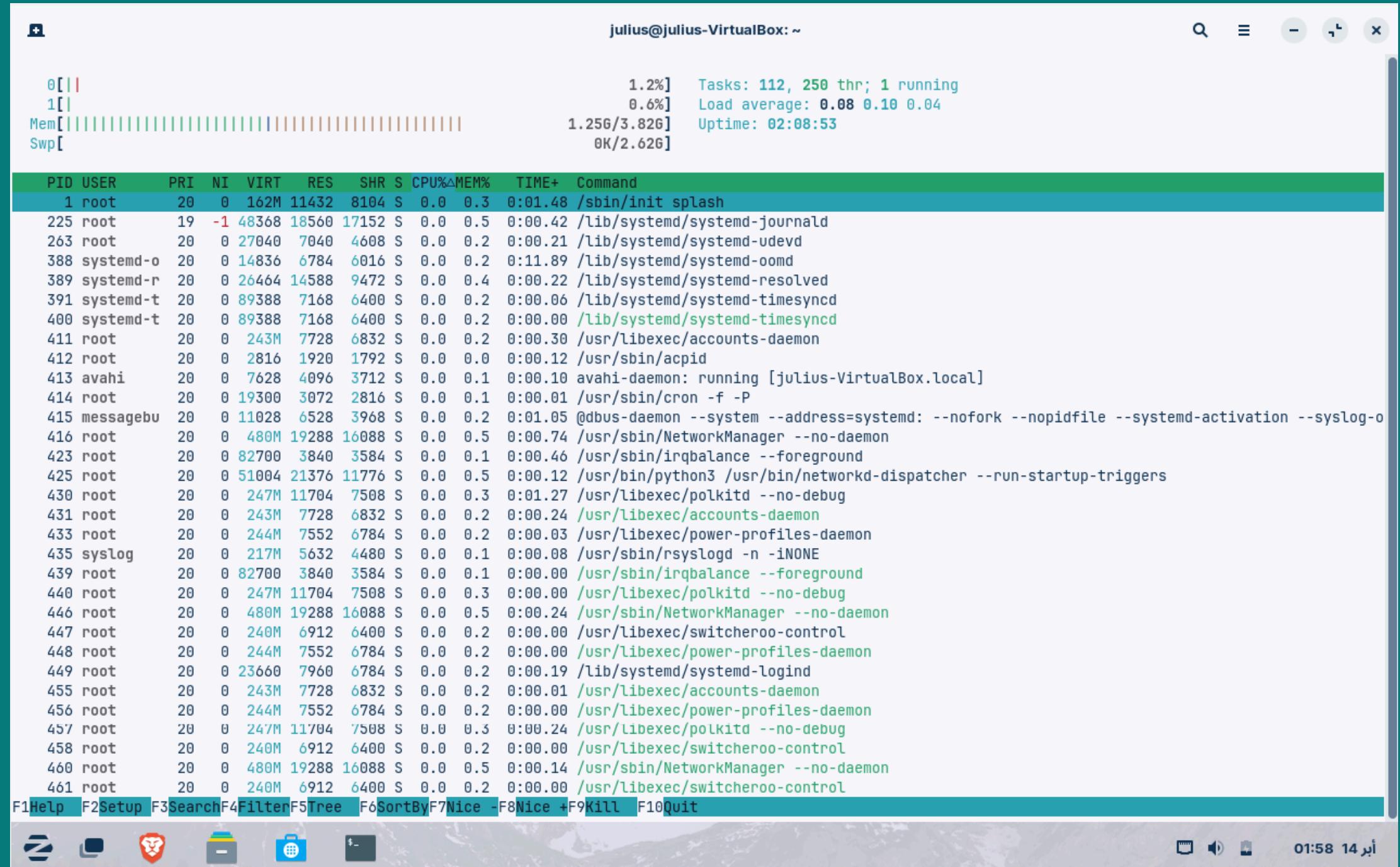
top - 01:55:00 up 2:05, 1 user, load average: 0.11, 0.10, 0.03
Tasks: 192 total, 1 running, 191 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.9 us, 3.9 sy, 0.0 ni, 90.2 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3916.5 total, 1366.1 free, 1259.5 used, 1290.9 buff/cache
MiB Swap: 2680.0 total, 2680.0 free, 0.0 used. 2396.7 avail Mem

PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
1127 julius    20   0 4579636 388916 145428 S 16.0  9.7  1:23.28 gnome-shell
2946 julius    20   0 648200 66600 46200 S  4.0  1.7  0:10.38 gnome-terminal-
5108 julius    20   0 23032  4096  3328 R  4.0  0.1  0:00.01 top
 1 root       20   0 166524 11432  8104 S  0.0  0.3  0:01.48 systemd
 2 root       20   0      0      0      0 S  0.0  0.0  0:00.02 kthreadd
 3 root       20   0      0      0      0 S  0.0  0.0  0:00.00 pool_workqueue_release
 4 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-rcu_g
 5 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-rcu_p
 6 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-slub_
 7 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-netns
10 root      0 -20      0      0      0 I  0.0  0.0  0:00.13 kworker/0:H-kblockd
11 root      20   0      0      0      0 I  0.0  0.0  0:00.00 kworker/u4:0-ext4-rsv-conversion
12 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-mm_pe
13 root      20   0      0      0      0 I  0.0  0.0  0:00.00 rcu_tasks_kthread
14 root      20   0      0      0      0 I  0.0  0.0  0:00.00 rcu_tasks_rude_kthread
15 root      20   0      0      0      0 I  0.0  0.0  0:00.00 rcu_tasks_trace_kthread
16 root      20   0      0      0      0 S  0.0  0.0  0:00.34 ksoftirqd/0
17 root      20   0      0      0      0 I  0.0  0.0  0:01.01 rcu_preempt
18 root      rt   0      0      0      0 S  0.0  0.0  0:00.05 migration/0
19 root     -51   0      0      0      0 S  0.0  0.0  0:00.00 idle_inject/0
20 root      20   0      0      0      0 S  0.0  0.0  0:00.00 cpuhp/0
21 root      20   0      0      0      0 S  0.0  0.0  0:00.00 cpuhp/1
22 root     -51   0      0      0      0 S  0.0  0.0  0:00.00 idle_inject/1
23 root      rt   0      0      0      0 S  0.0  0.0  0:00.93 migration/1
24 root      20   0      0      0      0 S  0.0  0.0  0:00.12 ksoftirqd/1
25 root      20   0      0      0      0 I  0.0  0.0  0:03.82 kworker/1:0-events  II
26 root      0 -20      0      0      0 I  0.0  0.0  0:00.06 kworker/1:0:H-kblockd
29 root      20   0      0      0      0 S  0.0  0.0  0:00.00 kdevtmpfs
30 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker/R-inet_
```

By pressing H we can change from showing processes to threads, so each thread will have its own LWP ID.



Use of htop for Thread view. Begin with htop then press H

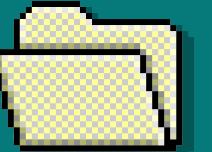
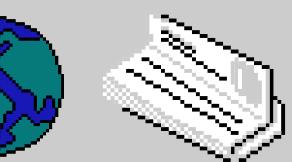
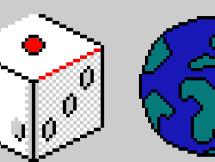
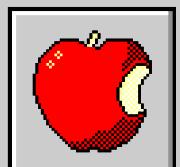


The screenshot shows the htop command running in a terminal window. The window title is "julius@julius-VirtualBox: ~". At the top, there's a memory usage bar with sections for CPU, Mem, and Swap, followed by system statistics: 1.2% CPU usage, 112 tasks, 1 running, 0.08 0.10 0.04 load average, 1.25G/3.82G memory, and 0K/2.62G swap. Below this is a table of processes:

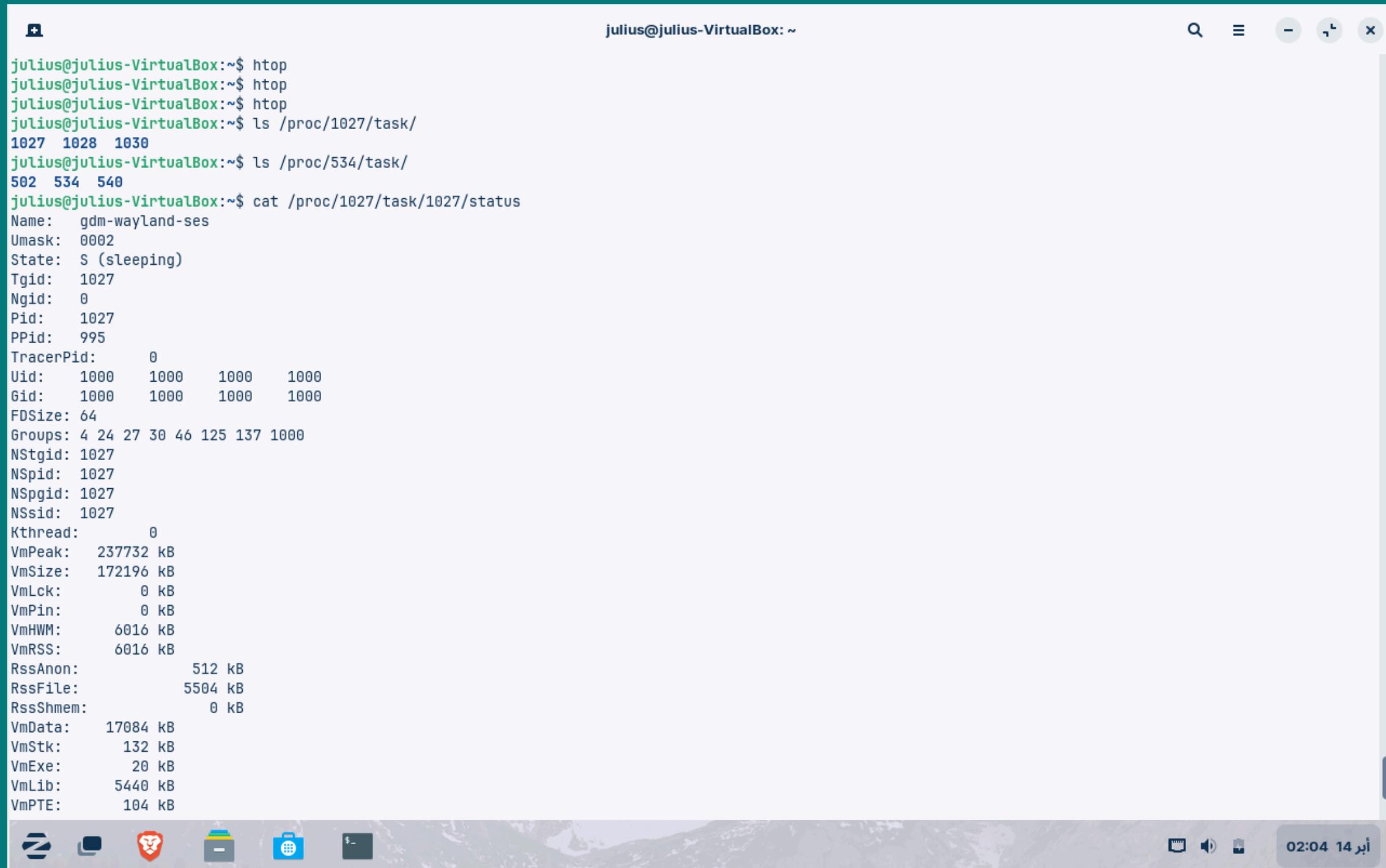
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU% MEM%	TIME+	Command
1	root	20	0	162M	11432	8104	S	0.0 0.3	0:01.48	/sbin/init splash
225	root	19	-1	48368	18560	17152	S	0.0 0.5	0:00.42	/lib/systemd/systemd-journald
263	root	20	0	27040	7040	4608	S	0.0 0.2	0:00.21	/lib/systemd/systemd-udevd
388	systemd-o	20	0	14836	6784	6016	S	0.0 0.2	0:11.89	/lib/systemd/systemd-oomd
389	systemd-r	20	0	26464	14588	9472	S	0.0 0.4	0:00.22	/lib/systemd/systemd-resolved
391	systemd-t	20	0	89388	7168	6400	S	0.0 0.2	0:00.06	/lib/systemd/systemd-timesyncd
400	systemd-t	20	0	89388	7168	6400	S	0.0 0.2	0:00.00	/lib/systemd/systemd-timesyncd
411	root	20	0	243M	7728	6832	S	0.0 0.2	0:00.30	/usr/libexec/accounts-daemon
412	root	20	0	2816	1920	1792	S	0.0 0.0	0:00.12	/usr/sbin/acpid
413	avahi	20	0	7628	4096	3712	S	0.0 0.1	0:00.10	avahi-daemon: running [julius-VirtualBox.local]
414	root	20	0	19300	3072	2816	S	0.0 0.1	0:00.01	/usr/sbin/cron -f -P
415	messagebu	20	0	11028	6528	3968	S	0.0 0.2	0:01.05	@dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-o
416	root	20	0	480M	19288	16088	S	0.0 0.5	0:00.74	/usr/sbin/NetworkManager --no-daemon
423	root	20	0	82700	3840	3584	S	0.0 0.1	0:00.46	/usr/sbin/irqbalance --foreground
425	root	20	0	51004	21376	11776	S	0.0 0.5	0:00.12	/usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
430	root	20	0	247M	11704	7508	S	0.0 0.3	0:01.27	/usr/libexec/polkitd --no-debug
431	root	20	0	243M	7728	6832	S	0.0 0.2	0:00.24	/usr/libexec/accounts-daemon
433	root	20	0	244M	7552	6784	S	0.0 0.2	0:00.03	/usr/libexec/power-profiles-daemon
435	syslog	20	0	217M	5632	4480	S	0.0 0.1	0:00.08	/usr/sbin/rsyslogd -n -iNONE
439	root	20	0	82700	3840	3584	S	0.0 0.1	0:00.00	/usr/sbin/irqbalance --foreground
440	root	20	0	247M	11704	7508	S	0.0 0.3	0:00.00	/usr/libexec/polkitd --no-debug
446	root	20	0	480M	19288	16088	S	0.0 0.5	0:00.24	/usr/sbin/NetworkManager --no-daemon
447	root	20	0	240M	6912	6400	S	0.0 0.2	0:00.00	/usr/libexec/switcheroo-control
448	root	20	0	244M	7552	6784	S	0.0 0.2	0:00.00	/usr/libexec/power-profiles-daemon
449	root	20	0	23660	7960	6784	S	0.0 0.2	0:00.19	/lib/systemd/systemd-logind
455	root	20	0	243M	7728	6832	S	0.0 0.2	0:00.01	/usr/libexec/accounts-daemon
456	root	20	0	244M	7552	6784	S	0.0 0.2	0:00.00	/usr/libexec/power-profiles-daemon
457	root	20	0	247M	11704	7508	S	0.0 0.3	0:00.24	/usr/libexec/polkitd --no-debug
458	root	20	0	240M	6912	6400	S	0.0 0.2	0:00.00	/usr/libexec/switcheroo-control
460	root	20	0	480M	19288	16088	S	0.0 0.5	0:00.14	/usr/sbin/NetworkManager --no-daemon
461	root	20	0	240M	6912	6400	S	0.0 0.2	0:00.00	/usr/libexec/switcheroo-control

At the bottom, there's a status bar with keyboard shortcuts: F1 Help, F2 Setup, F3 Search, F4 Filter, F5 Tree, F6 SortBy, F7 Nice -F8 Nice +F9 Kill, F10 Quit. The system tray shows icons for network, battery, and volume, with the time 01:58 and date 14.

Shows threads indented under parent process with LWP numbers and different CPU usage per thread



Deep Dive of File System:ls /proc/1234/task/

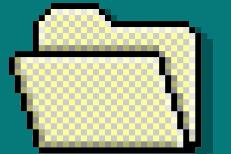
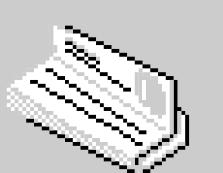
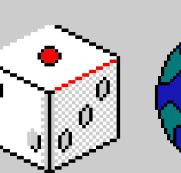
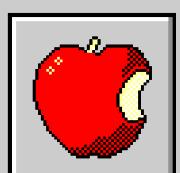


The screenshot shows a terminal window titled "julius@julius-VirtualBox:~". The terminal displays the following command history and output:

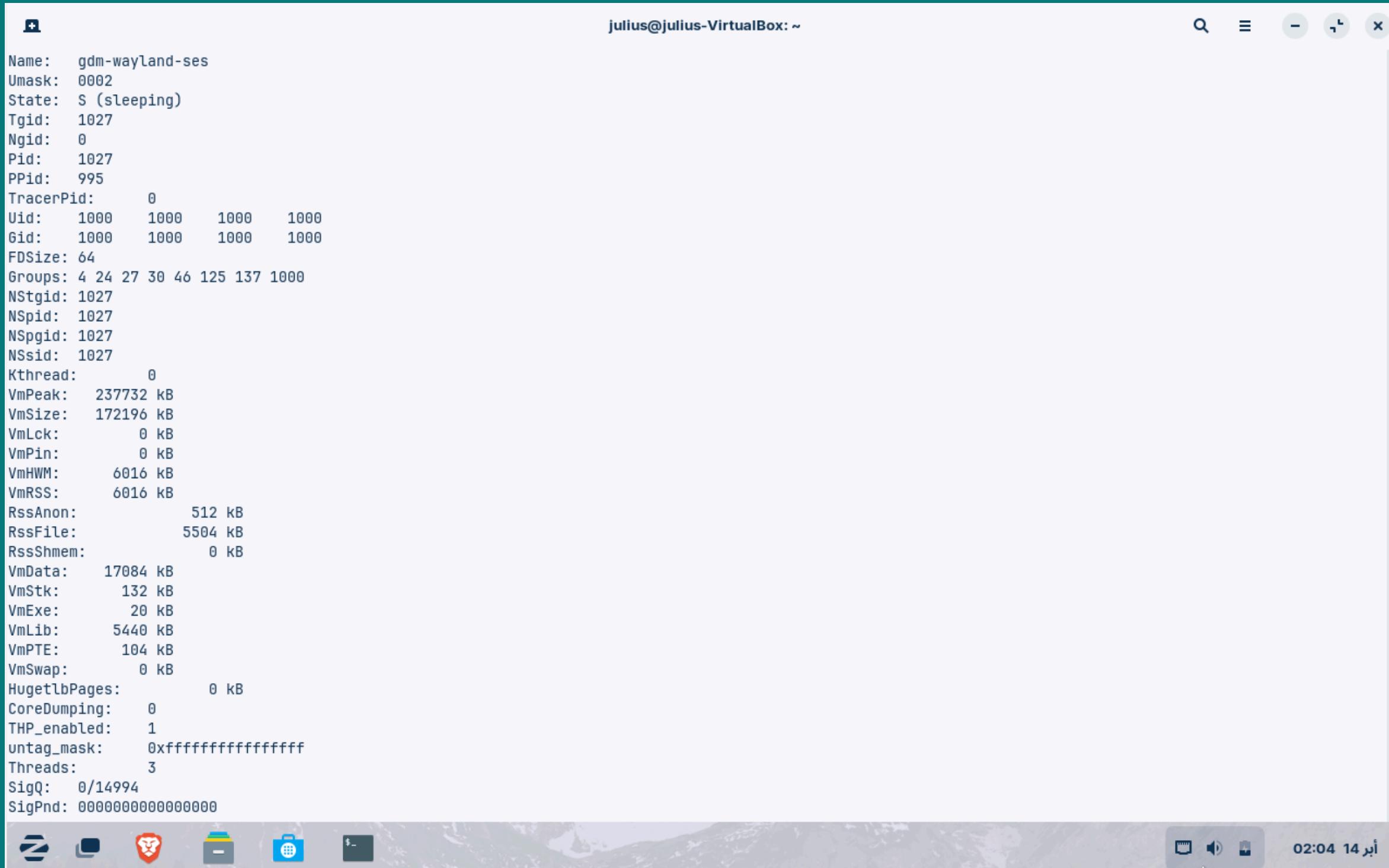
```
julius@julius-VirtualBox:~$ htop
julius@julius-VirtualBox:~$ htop
julius@julius-VirtualBox:~$ htop
julius@julius-VirtualBox:~$ ls /proc/1027/task/
1027 1028 1030
julius@julius-VirtualBox:~$ ls /proc/534/task/
502 534 540
julius@julius-VirtualBox:~$ cat /proc/1027/task/1027/status
Name: gdm-wayland-ses
Umask: 0002
State: S (sleeping)
Tgid: 1027
Ngid: 0
Pid: 1027
PPid: 995
TracerPid: 0
Uid: 1000 1000 1000 1000
Gid: 1000 1000 1000 1000
FDSize: 64
Groups: 4 24 27 30 46 125 137 1000
NSTgid: 1027
NSpid: 1027
NSpgid: 1027
NSSid: 1027
Kthread: 0
VmPeak: 237732 kB
VmSize: 172196 kB
VmLck: 0 kB
VmPin: 0 kB
VmHWM: 6016 kB
VmRSS: 6016 kB
RssAnon: 512 kB
RssFile: 5504 kB
RssShmem: 0 kB
VmData: 17084 kB
VmStk: 132 kB
VmExe: 20 kB
VmLib: 5440 kB
VmPTE: 104 kB
```

The terminal window has a dark theme with light-colored text. It includes standard Linux terminal icons in the bottom bar and a system tray at the bottom right showing the date and time.

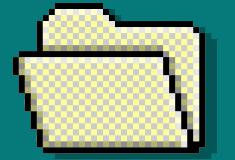
For a particular PID each folder inside /proc/1027/task/ and /proc/534/task/ corresponds to a thread ID (TID) of that process



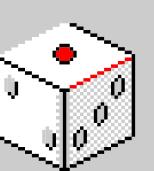
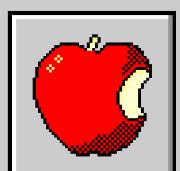
An even Deeper Deep Dive:`cat /proc/1027/task/1027/status`



```
Name: gdm-wayland-ses
Umask: 0002
State: S (sleeping)
Tgid: 1027
Ngid: 0
Pid: 1027
PPid: 995
TracerPid: 0
Uid: 1000 1000 1000 1000
Gid: 1000 1000 1000 1000
FDSize: 64
Groups: 4 24 27 30 46 125 137 1000
NSTgid: 1027
NSpid: 1027
NSpgid: 1027
NSsid: 1027
Kthread: 0
VmPeak: 237732 kB
VmSize: 172196 kB
VmLck: 0 kB
VmPin: 0 kB
VmHWM: 6016 kB
VmRSS: 6016 kB
RssAnon: 512 kB
RssFile: 5504 kB
RssShmem: 0 kB
VmData: 17084 kB
VmStk: 132 kB
VmExe: 20 kB
VmLib: 5440 kB
VmPTE: 104 kB
VmSwap: 0 kB
HugetlbPages: 0 kB
CoreDumping: 0
THP_enabled: 1
untag_mask: 0xfffffffffffffff
Threads: 3
SigQ: 0/14994
SigPnd: 0000000000000000
```

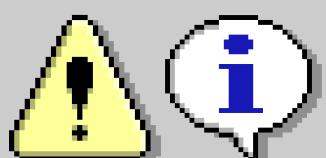


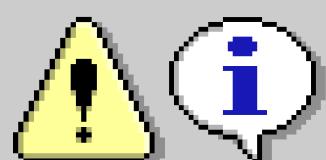
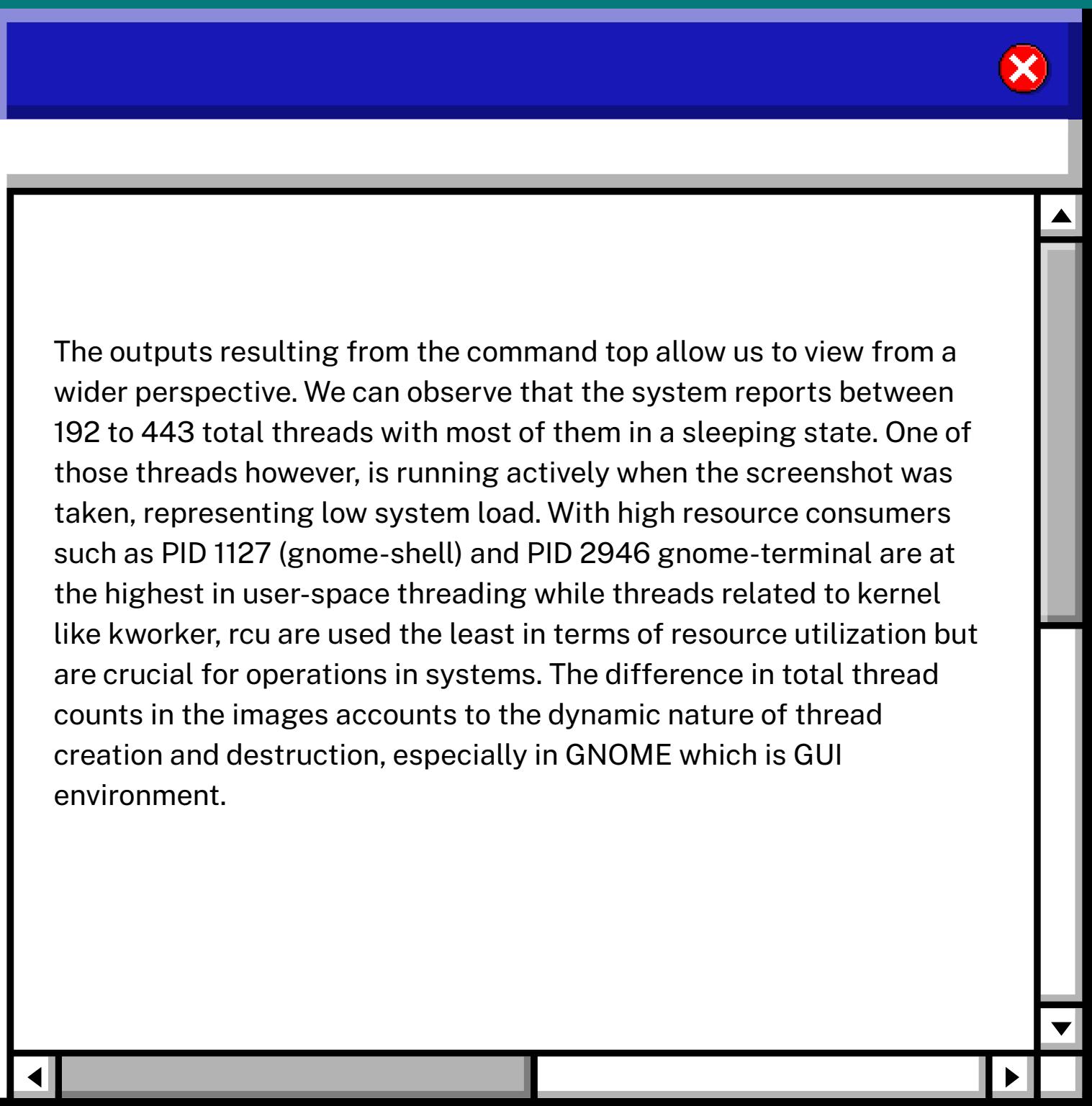
Shows the status of the main thread. Using /proc, we can do manual inspection through Linux displays threads and processes.



Thread Analysis

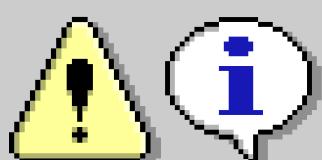
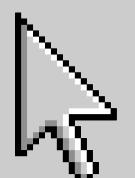
Through these screenshots, we can get a detailed overview of the thread behavior in Zorin OS. The primary tool used here was terminal, more of cmd version of Windows OS. The primary commands used as written above are top, htop, /proc filesystem inspection and ps shows us how the threads are handled in the user and kernel space. The system showcases a typical Linux thread architecture where Light Weight Processes (LWP) or threads share significant resources like memory and PID namespaces simultaneously monitoring different states of running. One important takeaway here is from /proc/1027/status which refers to the process gdm-wayland-ses. Generally, this process was randomly chosen to represent in screenshots and analyze. It is responsible for dealing with the GNOME display manager, and spawns three threads (IDs 1027, 1028 and 1030). As indicated in the image, these threads are in a sleeping state S which means they are idle and hence waiting for events like user input or system calls. Moreover, through these memory metrics, we can see the shared memory usage like VmRSS for resident memory and even RssAnon for thread-private allocations which signifies that threads in a same process share resources but have separate stacks and execution contexts.





Kernel VS User Threads

In the results produced by the command top, the Kernel threads as [kworker] or [rcu] are running with the least amount of overheads with near-zero memory and CPU usage but are critical for tasks such as interrupt handling and memory management. On the other hand, user threads, particularly which come under gnome-shell exhibit higher resource demands. We can observe that threaded applications like GNOME terminal gnome-terminal and systemd services systemd-udevd leverage threading to allow for parallelism. Through Python interpreters /usr/bin/python3 in the process list, we also see the multithreaded scripting, which is common in modern Linux distributions.

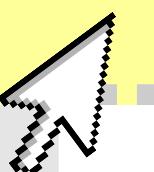




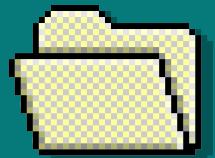
Terminal Implementation of System Calls



[Back to Agenda Page](#)



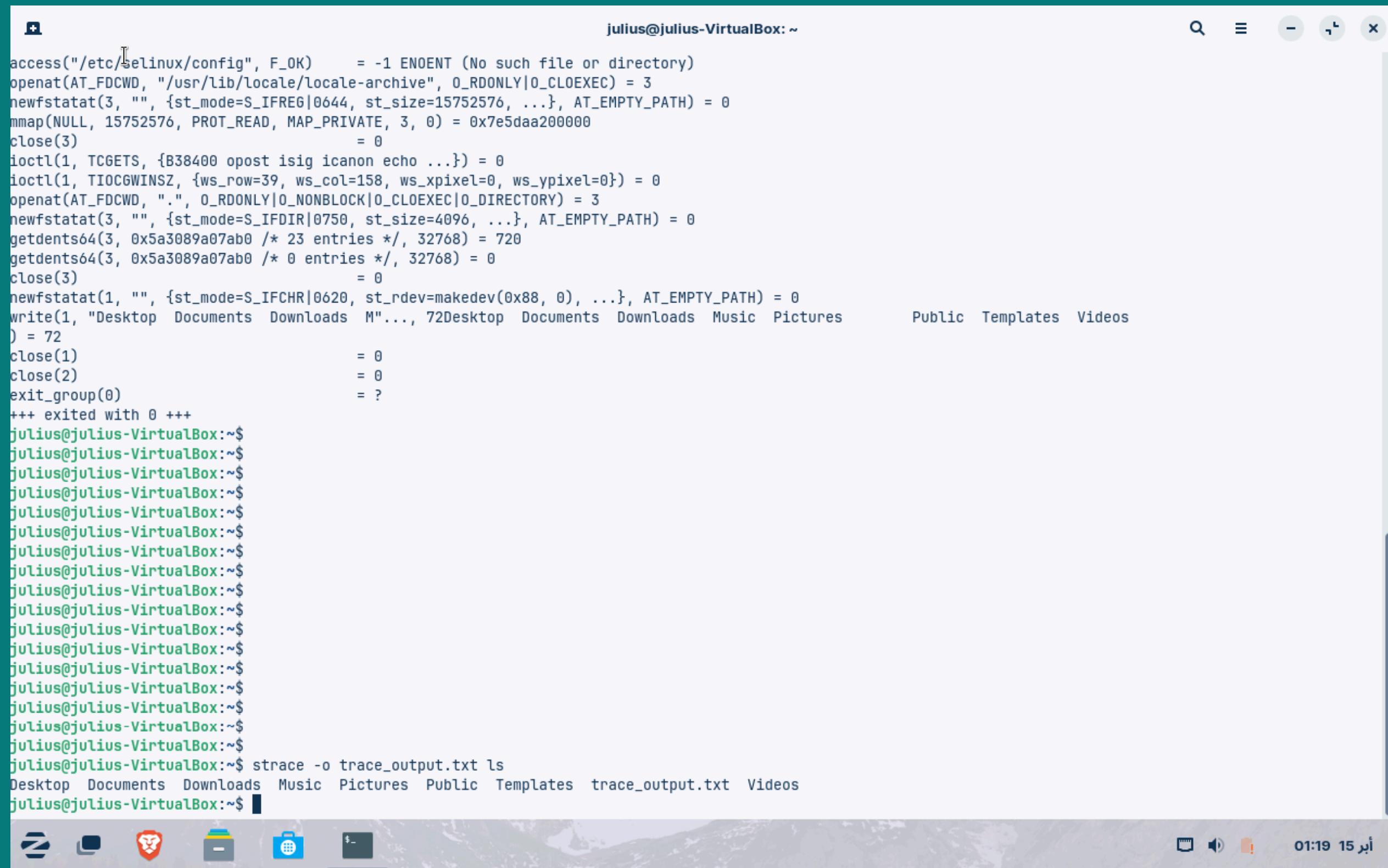
Using strace: strace ls



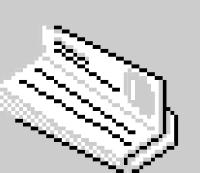
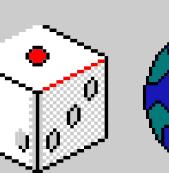
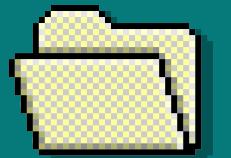
Trace system calls for the ls command using strace ls



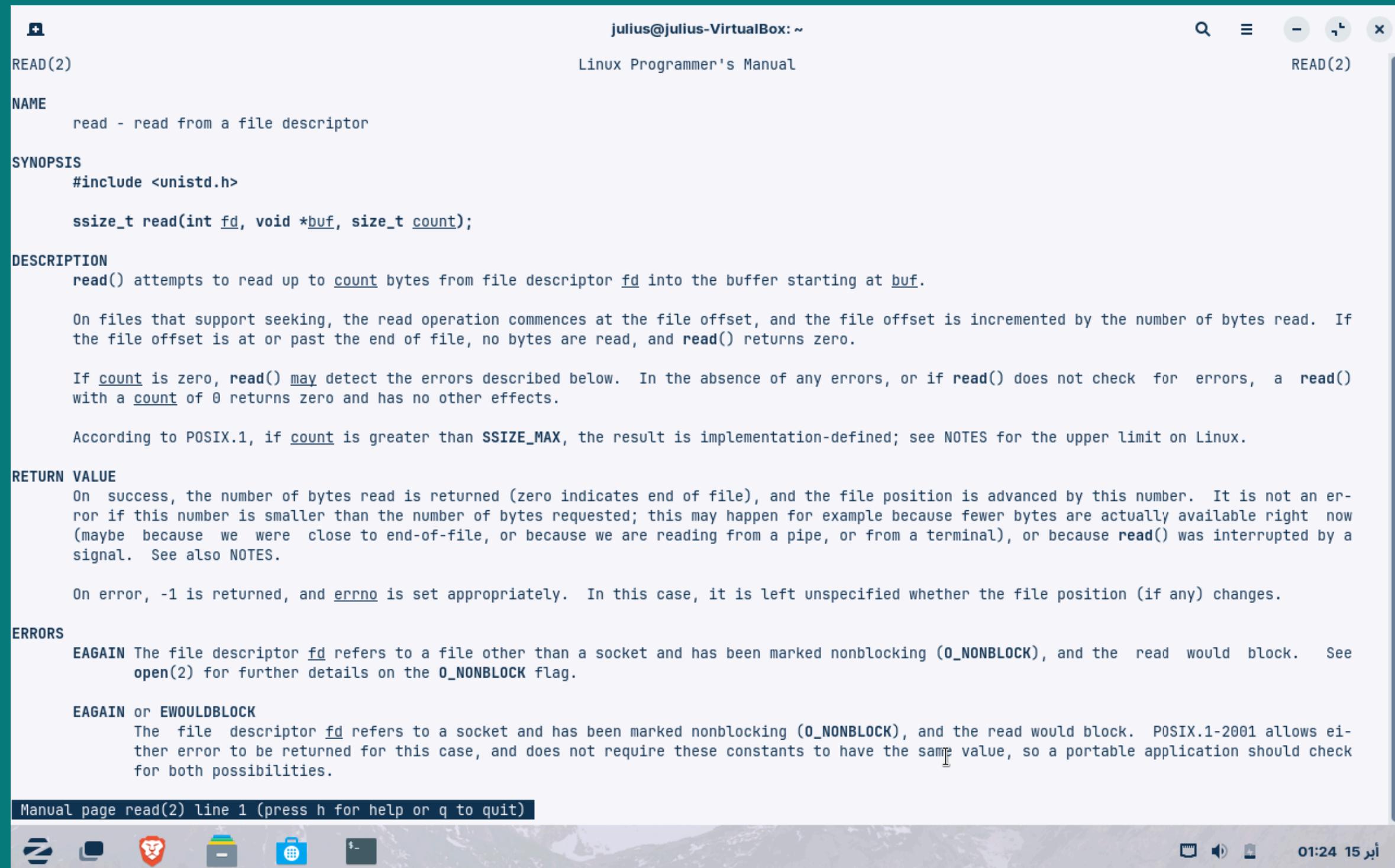
Redirecting_output_to_a_file_using_strace -o trace_output.txt ls



```
julius@julius-VirtualBox: ~
access("/etc/selinux/config", F_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=15752576, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 15752576, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7e5daa200000
close(3)                                = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=39, ws_col=158, ws_xpixel=0, ws_ypixel=0}) = 0
openat(AT_FDCWD, ".", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) = 3
newfstatat(3, "", {st_mode=S_IFDIR|0750, st_size=4096, ...}, AT_EMPTY_PATH) = 0
getdents64(3, 0x5a3089a07ab0 /* 23 entries */, 32768) = 720
getdents64(3, 0x5a3089a07ab0 /* 0 entries */, 32768) = 0
close(3)                                = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
write(1, "Desktop Documents Downloads M...", 72) = 72
write(1, "Desktop Documents Downloads Music Pictures", 26) = 26
write(1, "Public Templates Videos", 16) = 16
exit_group(0)                            = ?
+++ exited with 0 ++++
julius@julius-VirtualBox:~$ 
julius@julius-VirtualBox:~$ strace -o trace_output.txt ls
Desktop Documents Downloads Music Pictures Public Templates trace_output.txt Videos
julius@julius-VirtualBox:~$ 
```



Use of man to see available system calls



READ(2) julius@julius-VirtualBox: ~ Linux Programmer's Manual READ(2)

NAME
read - read from a file descriptor

SYNOPSIS

```
#include <unistd.h>

ssize_t read(int fd, void *buf, size_t count);
```

DESCRIPTION

read() attempts to read up to `count` bytes from file descriptor `fd` into the buffer starting at `buf`.

On files that support seeking, the read operation commences at the file offset, and the file offset is incremented by the number of bytes read. If the file offset is at or past the end of file, no bytes are read, and read() returns zero.

If `count` is zero, read() may detect the errors described below. In the absence of any errors, or if read() does not check for errors, a read() with a `count` of 0 returns zero and has no other effects.

According to POSIX.1, if `count` is greater than `SSIZE_MAX`, the result is implementation-defined; see NOTES for the upper limit on Linux.

RETURN VALUE

On success, the number of bytes read is returned (zero indicates end of file), and the file position is advanced by this number. It is not an error if this number is smaller than the number of bytes requested; this may happen for example because fewer bytes are actually available right now (maybe because we were close to end-of-file, or because we are reading from a pipe, or from a terminal), or because read() was interrupted by a signal. See also NOTES.

On error, -1 is returned, and `errno` is set appropriately. In this case, it is left unspecified whether the file position (if any) changes.

ERRORS

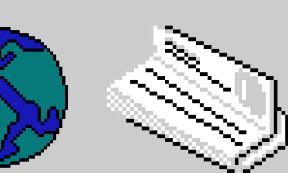
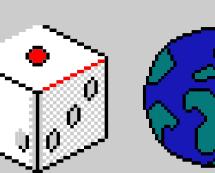
EAGAIN The file descriptor `fd` refers to a file other than a socket and has been marked nonblocking (`O_NONBLOCK`), and the read would block. See `open(2)` for further details on the `O_NONBLOCK` flag.

EAGAIN or EWOULDBLOCK The file descriptor `fd` refers to a socket and has been marked nonblocking (`O_NONBLOCK`), and the read would block. POSIX.1-2001 allows either error to be returned for this case, and does not require these constants to have the same value, so a portable application should check for both possibilities.

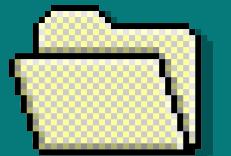
Manual page read(2) line 1 (press h for help or q to quit)



man 2 read shows the manual page for the read system call.



Use of man to see available system calls



```
julius@julius-VirtualBox: ~
INTRO(2)                               Linux Programmer's Manual
                                         INTRO(2)

NAME
    intro - introduction to system calls

DESCRIPTION
    Section 2 of the manual describes the Linux system calls. A system call is an entry point into the Linux kernel. Usually, system calls are not invoked directly: instead, most system calls have corresponding C library wrapper functions which perform the steps required (e.g., trapping to kernel mode) in order to invoke the system call. Thus, making a system call looks the same as invoking a normal library function.

    In many cases, the C library wrapper function does nothing more than:
    * copying arguments and the unique system call number to the registers where the kernel expects them;
    * trapping to kernel mode, at which point the kernel does the real work of the system call;
    * setting errno if the system call returns an error number when the kernel returns the CPU to user mode.

    However, in a few cases, a wrapper function may do rather more than this, for example, performing some preprocessing of the arguments before trapping to kernel mode, or postprocessing of values returned by the system call. Where this is the case, the manual pages in Section 2 generally try to note the details of both the (usually GNU) C library API interface and the raw system call. Most commonly, the main DESCRIPTION will focus on the C library interface, and differences for the system call are covered in the NOTES section.

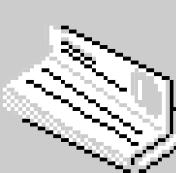
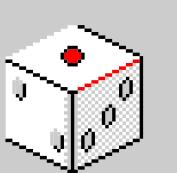
    For a list of the Linux system calls, see syscalls(2).

RETURN VALUE
    On error, most system calls return a negative error number (i.e., the negated value of one of the constants described in errno(3)). The C library wrapper hides this detail from the caller: when a system call returns a negative value, the wrapper copies the absolute value into the errno variable, and returns -1 as the return value of the wrapper.

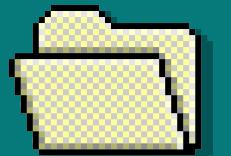
    The value returned by a successful system call depends on the call. Many system calls return 0 on success, but some can return nonzero values from a successful call. The details are described in the individual manual pages.

    In some cases, the programmer must define a feature test macro in order to obtain the declaration of a system call from the header file specified in the man page SYNOPSIS section. (Where required, these feature test macros must be defined before including any header files.) In such cases, the required macro is described in the man page. For further information on feature test macros, see feature_test_macros(7).
```

man 2 intro shows an introduction to section 2 of the manual, which is dedicated to system calls.



Use of man to see available system calls



```
Julius@Julius-VirtualBox: ~
Linux Programmer's Manual
WRITER(2)

NAME
    write - write to a file descriptor

SYNOPSIS
    #include <unistd.h>

    ssize_t write(int fd, const void *buf, size_t count);

DESCRIPTION
    write() writes up to count bytes from the buffer starting at buf to the file referred to by the file descriptor fd.

    The number of bytes written may be less than count if, for example, there is insufficient space on the underlying physical medium, or the RLIMIT_FSIZE resource limit is encountered (see setrlimit(2)), or the call was interrupted by a signal handler after having written less than count bytes. (See also pipe(7).)

    For a seekable file (i.e., one to which lseek(2) may be applied, for example, a regular file) writing takes place at the file offset, and the file offset is incremented by the number of bytes actually written. If the file was open(2)ed with O_APPEND, the file offset is first set to the end of the file before writing. The adjustment of the file offset and the write operation are performed as an atomic step.

    POSIX requires that a read(2) that can be proved to occur after a write() has returned will return the new data. Note that not all filesystems are POSIX conforming.

    According to POSIX.1, if count is greater than SSIZE_MAX, the result is implementation-defined; see NOTES for the upper limit on Linux.

RETURN VALUE
    On success, the number of bytes written is returned. On error, -1 is returned, and errno is set to indicate the cause of the error.

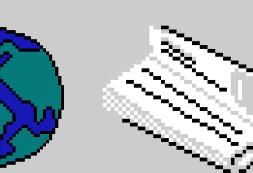
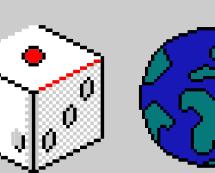
    Note that a successful write() may transfer fewer than count bytes. Such partial writes can occur for various reasons; for example, because there was insufficient space on the disk device to write all of the requested bytes, or because a blocked write() to a socket, pipe, or similar was interrupted by a signal handler after it had transferred some, but before it had transferred all of the requested bytes. In the event of a partial write, the caller can make another write() call to transfer the remaining bytes. The subsequent call will either transfer further bytes or may result in an error (e.g., if the disk is now full).

    If count is zero and fd refers to a regular file, then write() may return a failure status if one of the errors below is detected. If no errors are detected, or error detection is not performed, 0 will be returned without causing any other effect. If count is zero and fd refers to a file other than a regular file, the results are not specified.

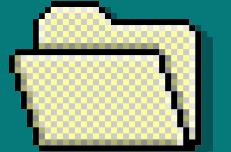
Manual page write(2) line 1 (press h for help or q to quit)
```



man 2 write shows the manual page for the write system call.



Use of man to see available system calls



```
FORK(2)                                julius@julius-VirtualBox: ~
                                         Linux Programmer's Manual
FORK(2)

NAME
    fork - create a child process

SYNOPSIS
    #include <sys/types.h>
    #include <unistd.h>

    pid_t fork(void);

DESCRIPTION
    fork() creates a new process by duplicating the calling process. The new process is referred to as the child process. The calling process is referred to as the parent process.

    The child process and the parent process run in separate memory spaces. At the time of fork() both memory spaces have the same content. Memory writes, file mappings (mmap(2)), and unmappings (munmap(2)) performed by one of the processes do not affect the other.

    The child process is an exact duplicate of the parent process except for the following points:

    * The child has its own unique process ID, and this PID does not match the ID of any existing process group (setpgid(2)) or session.

    * The child's parent process ID is the same as the parent's process ID.

    * The child does not inherit its parent's memory locks (mlock(2), mlockall(2)).

    * Process resource utilizations (getrusage(2)) and CPU time counters (times(2)) are reset to zero in the child.

    * The child's set of pending signals is initially empty (sigpending(2)).

    * The child does not inherit semaphore adjustments from its parent (semop(2)).

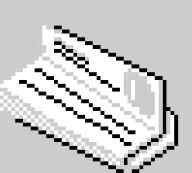
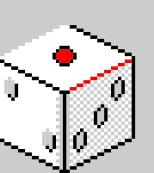
    * The child does not inherit process-associated record locks from its parent (fcntl(2)). (On the other hand, it does inherit fcntl(2) open file
description locks and flock(2) locks from its parent.)

    * The child does not inherit timers from its parent (setitimer(2), alarm(2), timer_create(2)).

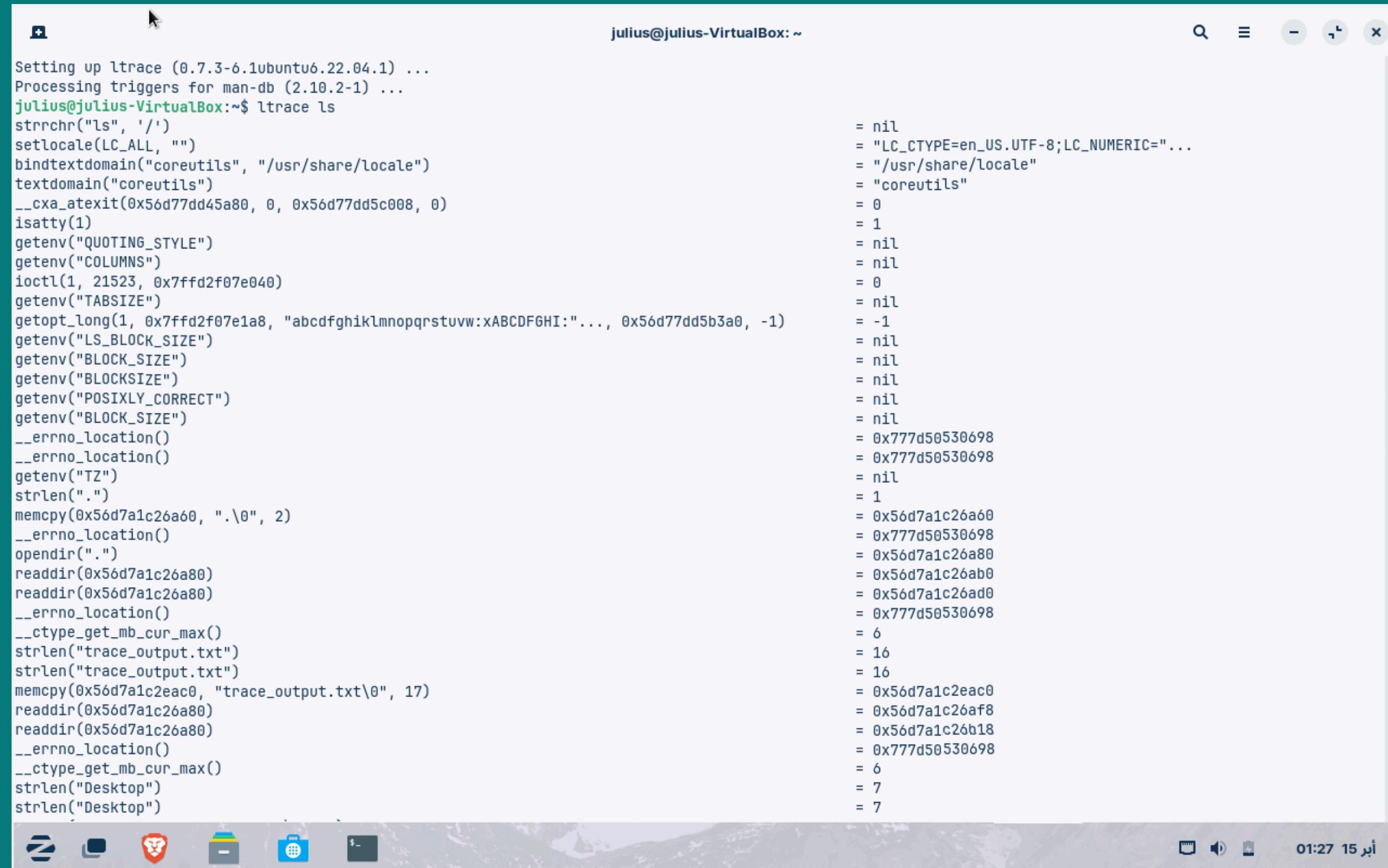
    * The child does not inherit outstanding asynchronous I/O operations from its parent (aio_read(3), aio_write(3)), nor does it inherit any asyn-
```

Manual page fork(2) line 1 (press h for help or q to quit)

man 2 fork shows the manual page for the fork system call.

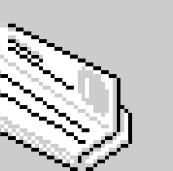
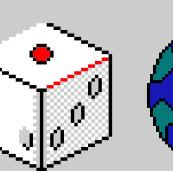


ltrace ls

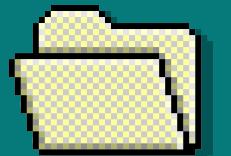


```
Setting up ltrace (0.7.3-6.1ubuntu6.22.04.1) ...
Processing triggers for man-db (2.10.2-1) ...
julius@julius-VirtualBox:~$ ltrace ls
strchr("ls", '/')
setlocale(LC_ALL, "")
bindtextdomain("coreutils", "/usr/share/locale")
textdomain("coreutils")
__cxa_atexit(0x56d77dd45a80, 0, 0x56d77dd5c008, 0)
isatty(1)
getenv("QUOTING_STYLE")
getenv("COLUMNS")
ioctl(1, 21523, 0x7ffd2f07e040)
getenv("TABSIZE")
 getopt_long(1, 0x7ffd2f07e1a8, "abcdefghijklmnopqrstuvwxyz:xABCDEFGHIJKLMNOPQRSTUVWXYZ:"..., 0x56d77dd5b3a0, -1)
getenv("LS_BLOCK_SIZE")
getenv("BLOCK_SIZE")
getenv("BLOCKSIZE")
getenv("POSIXLY_CORRECT")
getenv("BLOCK_SIZE")
__errno_location()
__errno_location()
getenv("TZ")
strlen(".")
memcpy(0x56d7a1c26a60, ".\0", 2)
__errno_location()
opendir(".")
readdir(0x56d7a1c26a80)
readdir(0x56d7a1c26a80)
__errno_location()
__ctype_get_mb_cur_max()
strlen("trace_output.txt")
strlen("trace_output.txt")
memcpy(0x56d7a1c2eac0, "trace_output.txt\0", 17)
readdir(0x56d7a1c26a80)
readdir(0x56d7a1c26a80)
__errno_location()
__ctype_get_mb_cur_max()
strlen("Desktop")
strlen("Desktop")
```

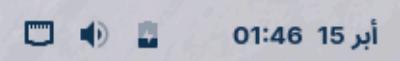
Use of ltrace for library calls (different to system calls)



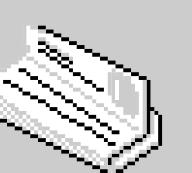
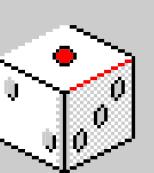
System Calls using _C



```
julius@julius-VirtualBox:~$ sudo apt install build-essential
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu dpkg-dev fakeroot g++ g++-11 gcc gcc-11 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan0 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libfakeroot
  libfile-fcntllock-perl libgcc-11-dev libitm1 liblsan0 libnsl-dev libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev lto-disabled-list make
  rpcsvc-proto
Suggested packages:
  binutils-doc debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-11-multilib
  gcc-11-locales glibc-doc git bzr libstdc++-11-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-11 gcc gcc-11 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan0 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0
  libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-11-dev libitm1 liblsan0 libnsl-dev libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev
  lto-disabled-list make rpcsvc-proto
0 upgraded, 37 newly installed, 0 to remove and 87 not upgraded.
Need to get 51.7 MB of archives.
After this operation, 182 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 binutils-common amd64 2.38-4ubuntu2.8 [223 kB]
Get:2 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libbinutils amd64 2.38-4ubuntu2.8 [661 kB]
Get:3 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libctf-nobfd0 amd64 2.38-4ubuntu2.8 [108 kB]
Get:4 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libctf0 amd64 2.38-4ubuntu2.8 [103 kB]
Get:5 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 binutils-x86_64-linux-gnu amd64 2.38-4ubuntu2.8 [2,324 kB]
Get:6 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 binutils amd64 2.38-4ubuntu2.8 [3,196 B]
Get:7 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc-dev-bin amd64 2.35-0ubuntu3.9 [20.3 kB]
Get:8 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 linux-libc-dev amd64 5.15.0-136.147 [1,320 kB]
Get:9 http://ae.archive.ubuntu.com/ubuntu jammy/main amd64 libcrypt-dev amd64 1:4.4.27-1 [112 kB]
Get:10 http://ae.archive.ubuntu.com/ubuntu jammy/main amd64 rpcsvc-proto amd64 1.4.2-0ubuntu6 [68.5 kB]
Get:11 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtirpc-dev amd64 1.3.2-2ubuntu0.1 [192 kB]
Get:12 http://ae.archive.ubuntu.com/ubuntu jammy/main amd64 libnsl-dev amd64 1.3.0-2build2 [71.3 kB]
Get:13 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc6-dev amd64 2.35-0ubuntu3.9 [2,100 kB]
Get:14 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcc1-0 amd64 12.3.0-1ubuntu1~22.04 [48.3 kB]
Get:15 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libitm1 amd64 12.3.0-1ubuntu1~22.04 [30.2 kB]
Get:16 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libasan0 amd64 11.4.0-1ubuntu1~22.04 [2,282 kB]
Get:17 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 liblsan0 amd64 12.3.0-1ubuntu1~22.04 [1,069 kB]
Get:18 http://ae.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtsan0 amd64 11.4.0-1ubuntu1~22.04 [2,260 kB]
```

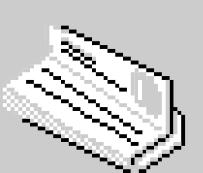
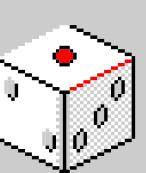
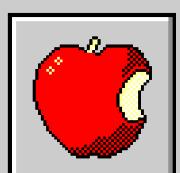


sudo apt install build-essential

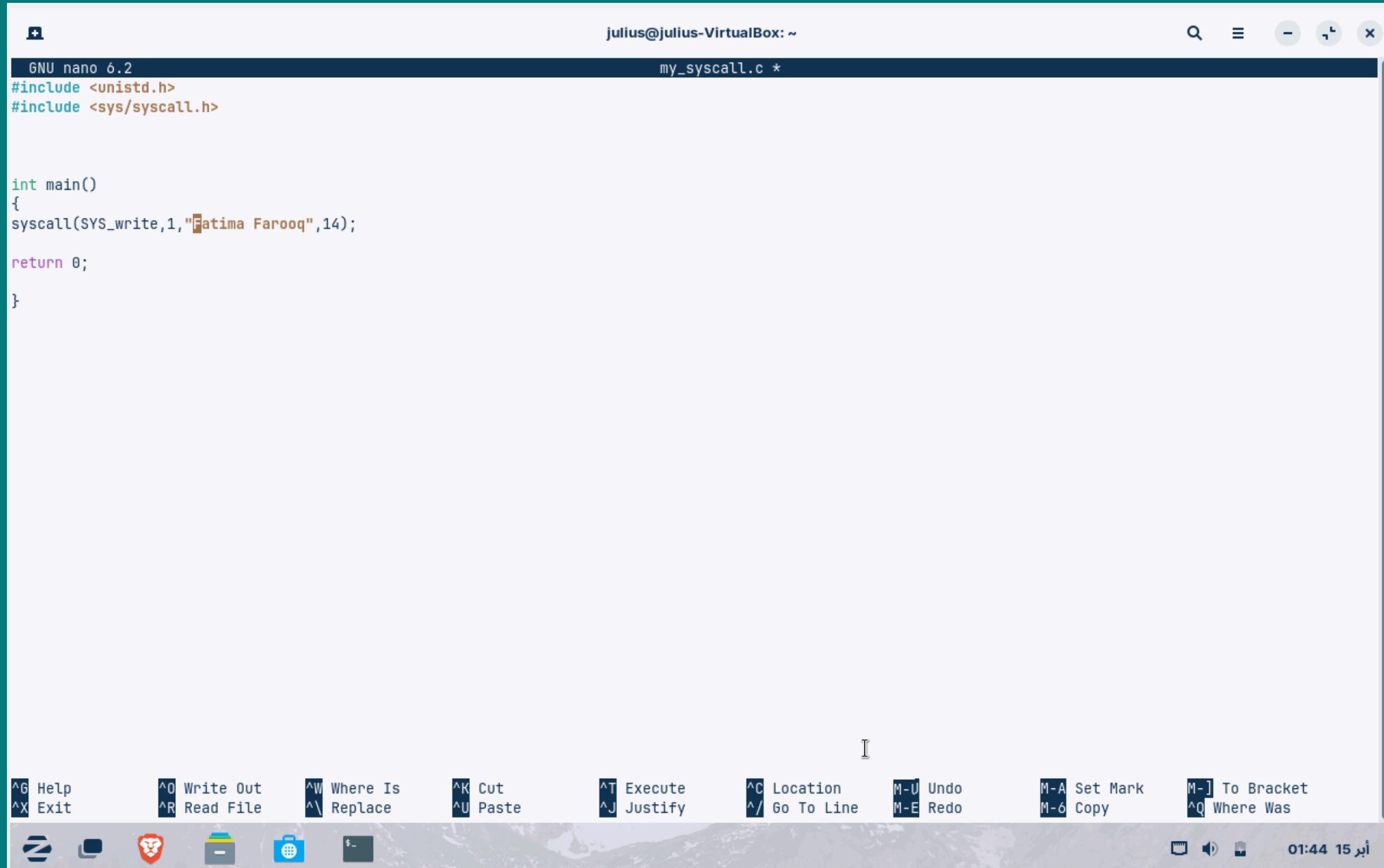


System Calls using C

```
nano my_syscall.c
```



System Calls using_C



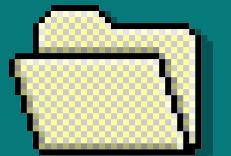
The screenshot shows a terminal window titled "my_syscall.c *". The code in the terminal is:

```
GNU nano 6.2
#include <unistd.h>
#include <sys/syscall.h>

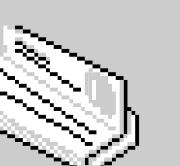
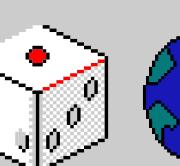
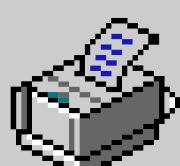
int main()
{
    syscall(SYS_write, 1, "Fatima Farooq", 14);

    return 0;
}
```

The terminal window has a standard Linux-style interface with a menu bar, a toolbar at the bottom, and a status bar at the bottom right.

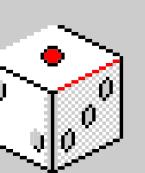
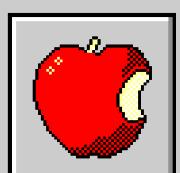


Type a simple C code and include the relevant libraries

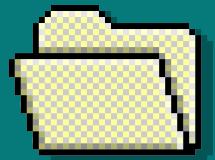


System Calls using C

```
gcc my_syscall.c -o my_syscall
```

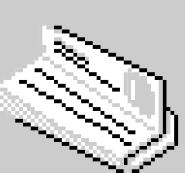
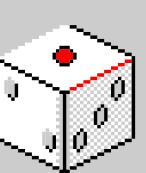
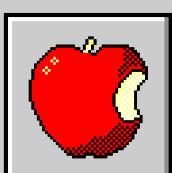


`./my_syscall`



System Calls using C

```
strace ./my_syscall
```



Thank you!

