KYGnus Nas

Technical Document

KYGnus

KooshaYeganehGnuLinuxSoftware

Table of Contents

   - Python
   - Flask
   - clamAV
   - RootkitHunter
   - Chkrootkit
   - Tiger
   - LMD (Linux Malware Detect)
   - Nmap
  - CommandLine
    - Bash
    - clamAV
    - RootkitHunter
    - Chkrootkit
    - LMD (Linux Malware Detect)
    - Nmap
    - Monitoring (Netdata)

- Desktops
  - Desktop
    - Gnome
    - KDe
    - Xfce
    - Tiling Window Manager (i3)

- Optimization
    - Desktop
        - Tiling Window Manager (i3)
        - XFCE
        - KDE Plasma

- Project personnel information

- Resources

Introduction

The Nas project is a part of the big project of converting old electronic systems into systems and practical tools, the first phase of which started with the conversion of old PCs into NAS.

The idea of this project started after talking with friends and colleagues. Each of them had an old PC at home that had an old operating system on it or could no longer be used after changing the operating system (all operating systems were Windows).

In the first place, this issue started with the suggestion of installing Linux Lite, and with this move, many of the Henna systems returned to the cycle of daily use.

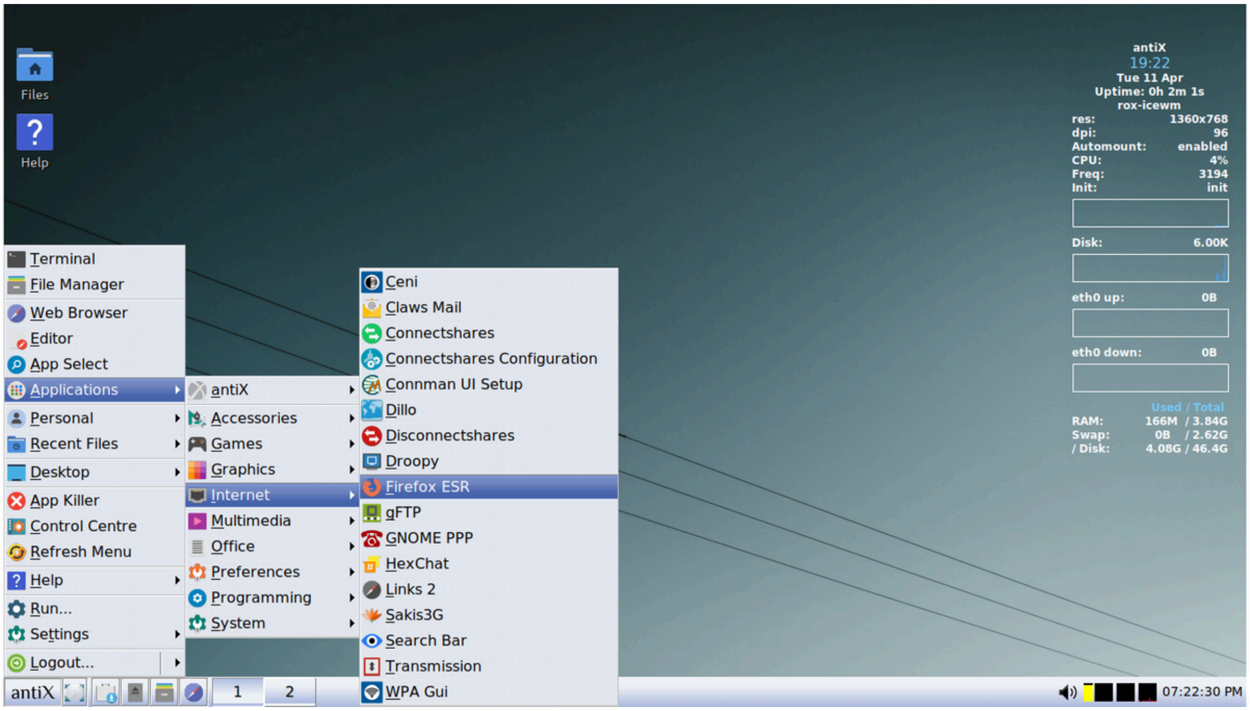
In this NAS project, there are positive points compared to other NAS that can be purchased on the market.

- **Here, the user can install any conventional Linux on his system and maneuver on operating systems as well.**

- **In personal computers, it is possible to upgrade CPU RAM, but this is not possible in devices.**

Lite Linux

1 - Antix



```
antiX
19:22
Tue 11 Apr
Uptime: 0h 2m 1s
rox-icewm
res:           1360x768
dpi:                 96
Automount:      enabled
CPU:                 4%
Freq:              3194
Init:              init

Disk:             6.00K

eth0 up:            0B

eth0 down:          0B

              Used / Total
RAM:        166M  / 3.84G
Swap:        0B   / 2.62G
/ Disk:     4.08G / 46.4G
```

antiX is a fast, lightweight and easy-to-install Linux live CD distribution based on Debian's "Stable" branch for x86 compatible systems. antiX offers users the "antiX Magic" in an environment suitable for old computers. The goal of antiX is to provide a light, but fully functional and flexible free operating system for both newcomers and experienced users of Linux. It should run on most computers, ranging from 256 MB old PIII systems with pre-configured swap to the latest powerful boxes.. antiX can also be used as a fast-booting rescue CD, or run "live" on a USB stick, with or without persistent file storage.

Minimum System Requirements :

- **Processor**: The minimum recommended processor is an i486 or Pentium processor. However, antiX Linux can run on a wide range of processors, including newer ones. For a smoother experience, a modern processor (1 GHz or more) is beneficial.

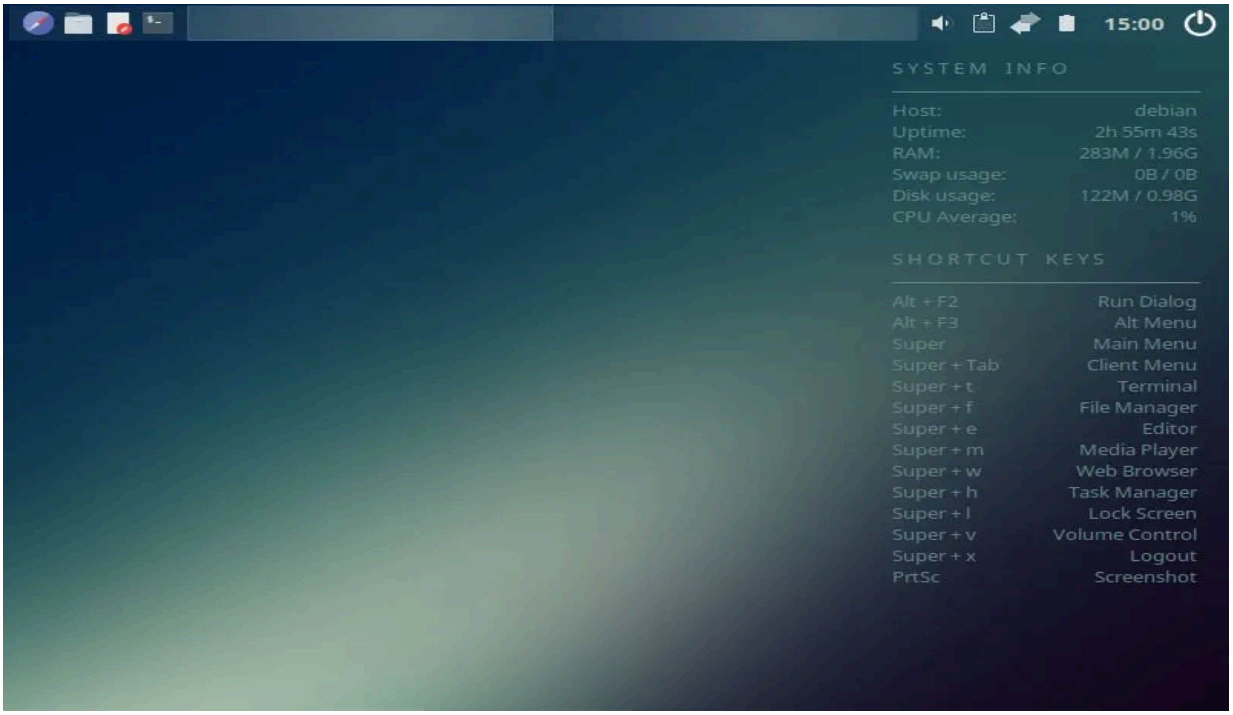
- **RAM**: Minimum RAM requirement is around 256 MB for the base system. However, for a more functional and responsive system, it's recommended to have at least 512 MB to 1 GB of RAM.

- **Storage**: antiX Linux can run on systems with very limited disk space. A minimum of 5 GB of free disk space is recommended for a basic installation. If you plan to install additional software and store data, more space will be needed.

- **Graphics**: antiX Linux supports a variety of graphics cards, including both open-source and proprietary drivers. It's designed to work well with older or less powerful graphics hardware.

- **Display**: A display with a resolution of at least 800x600 is recommended, but higher resolutions are supported.

BunsenLabs Linux



```
SYSTEM INFO

Host:                  debian
Uptime:            2h 55m 43s
RAM:              283M / 1.96G
Swap usage:            0B / 0B
Disk usage:      122M / 0.98G
CPU Average:                1%

SHORTCUT KEYS

Alt + F2          Run Dialog
Alt + F3            Alt Menu
Super              Main Menu
Super + Tab      Client Menu
Super + t           Terminal
Super + f       File Manager
Super + e             Editor
Super + m       Media Player
Super + w        Web Browser
Super + h       Task Manager
Super + l        Lock Screen
Super + v     Volume Control
Super + x             Logout
PrtSc             Screenshot
```

Crunchbang (or #!) was a very popular Debian-derived distro specifically designed to use as few system resources as possible. While it was discontinued in 2013, the community fondly remembered its lightning speed and responded with two Crunchbang-based distros to continue its legacy.

However, one of those successors, Crunchbang++, has now been discontinued. BunsenLabs is still active, though, and its current release (Beryllium) is based on Debian 11 (Bullseye). It features a gorgeously configurable Openbox window manager. You can install extra software from the Debian repositories too.

The distro ships with an assortment of themes and wallpapers, and includes a number of everyday desktop apps to provide a very usable out-of-the-box experience.

BunsenLabs is available for both 32-bit and 64-bit machines, and the developers recommend running the distro on a machine with more than 2 GB of RAM. You can test drive BunsenLabs in "live" mode before installing.

Minimum System Requirements :

- **Processor**: BunsenLabs Linux can run on a wide range of processors, including both 32-bit and 64-bit architectures. For a smooth experience, a modern processor (1 GHz or more) is recommended.

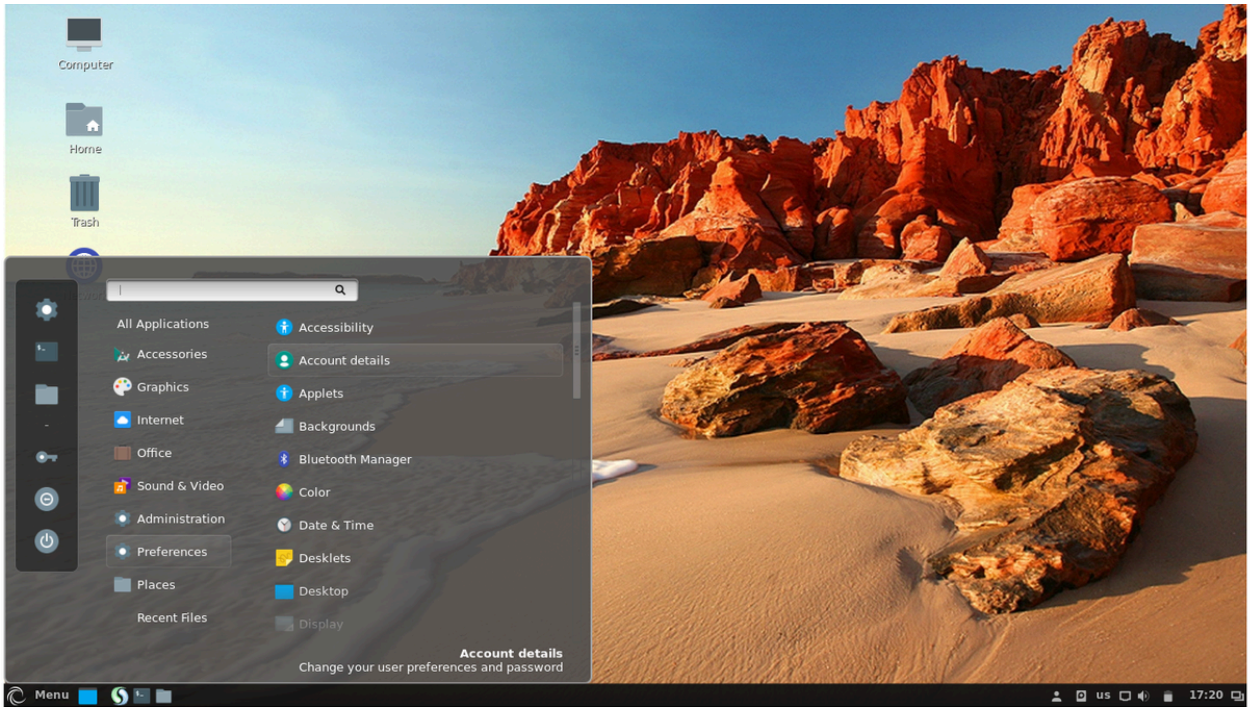
- **RAM**: Minimum 512 MB of RAM is required for running BunsenLabs, but it is advisable to have at least 1 GB or more for a better experience.

- **Storage**: At least 10-15 GB of free disk space is recommended for the base installation and some additional software.

- **Graphics**: BunsenLabs Linux can run on most graphics cards, and it supports a variety of open-source and proprietary graphics drivers. It's based on Openbox, which is lightweight and can work well on older or less powerful graphics hardware.

- **Display**: A display with a resolution of at least 800x600 is recommended, but higher resolutions are supported

Porteus



Porteus is a Slackware-based distro that is designed to be completely portable and run on removable media such as a USB stick or CD, but can just as easily be installed to a hard disk. The distro is incredibly fast as it's small enough to run entirely from system RAM.

The unique selling point of Porteus is that it exists in a compressed state and creates the file system on-the-fly. Besides the preinstalled apps, all additional software for the distro comes in the form of modules, making the OS very small and compact.

Porteus is available for 32-bit and 64-bit machines. If you're running the 64-bit version of the OS, you can also get 32-bit applications to run by installing the relevant libraries from Porteus' software repositories. The distro provides users with the choice of KDE, MATE, Openbox, LXQt, Cinnamon, Xfce and LXDE desktop environments when downloading the ISO image.

Unfortunately the option to build your own custom ISO has been removed since we previously looked at Porteus, but the pre-built images offer a decent selection of software and drivers, as well as an excellent selection of tutorials to help you get started.

Minimum System Requirements :

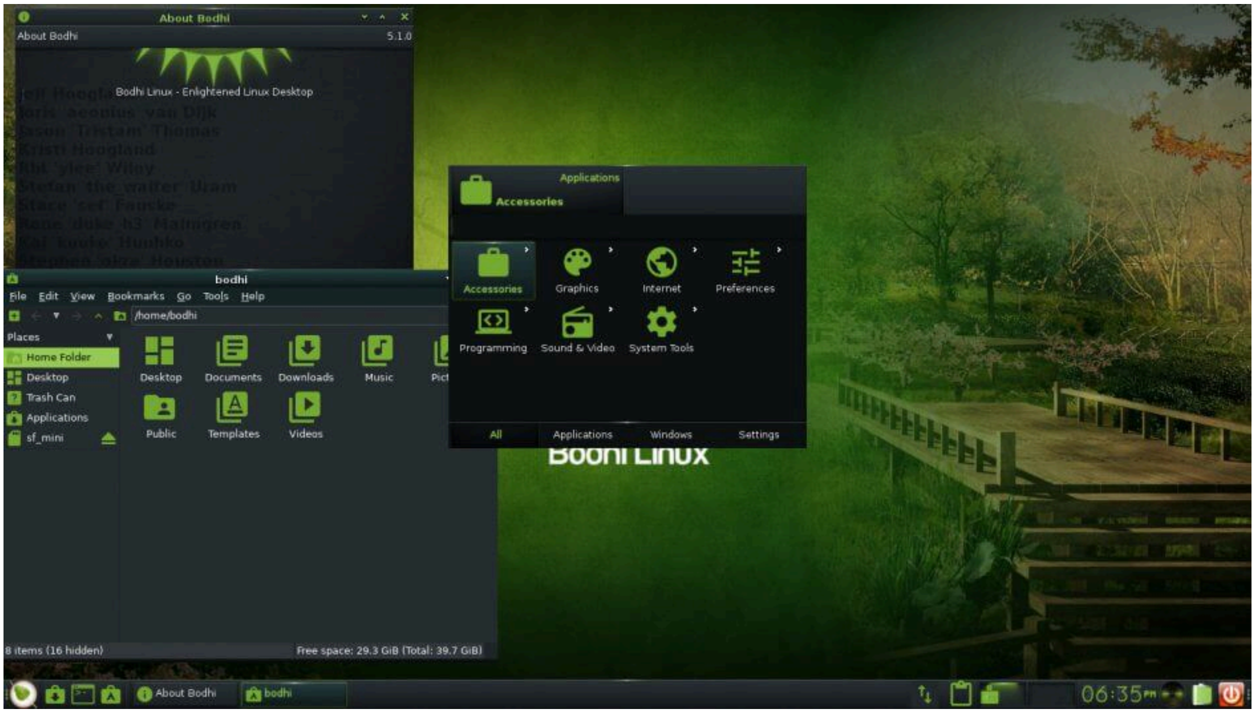
Processor: Porteus Linux is compatible with both 32-bit (x86) and 64-bit (x86_64) processors. It supports a wide range of processors, including Intel and AMD. The specific processor speed is not critical, but a modern processor (1 GHz or more) is recommended for a better experience.

RAM: Porteus can run on systems with as little as 256 MB of RAM. However, for a more functional and responsive system, it's recommended to have at least 512 MB to 1 GB of RAM.

Storage: Porteus is designed to run from portable storage media like USB drives, SD cards, or optical discs. The space required depends on the edition and how much additional software you install. A typical installation might require 300 MB to 700 MB of space.

Graphics: Porteus supports a variety of graphics cards and drivers, both open-source and proprietary. It's designed to work with a broad range of graphics hardware.
Display: A display with a resolution of at least 800x600 pixels is recommended.

bodhi Linux



Bodhi Linux is an elegant and lightweight Debian/Ubuntu-based distribution featuring Moksha, an Enlightenment-17-based desktop environment. The project takes a decidedly minimalist approach by offering modularity, high levels of customisation, and choice of themes. Bodhi releases come in several editions, including Standard (64-bit) and Legacy (32-bit) which are minimalist, only including a web browser, terminal, file manger, text editor and photo GUI applications, while the AppPack edition includes more applications and tools preinstalled. Additional software can be added with Bodhi's web-based AppCenter, Synaptic, and APT.

Minimum System Requirements :

Processor: A modern processor (such as Intel or AMD) capable of running a Linux kernel is sufficient. The specific processor speed and type are not typically critical, but a dual-core processor or better is advisable for a smoother experience.

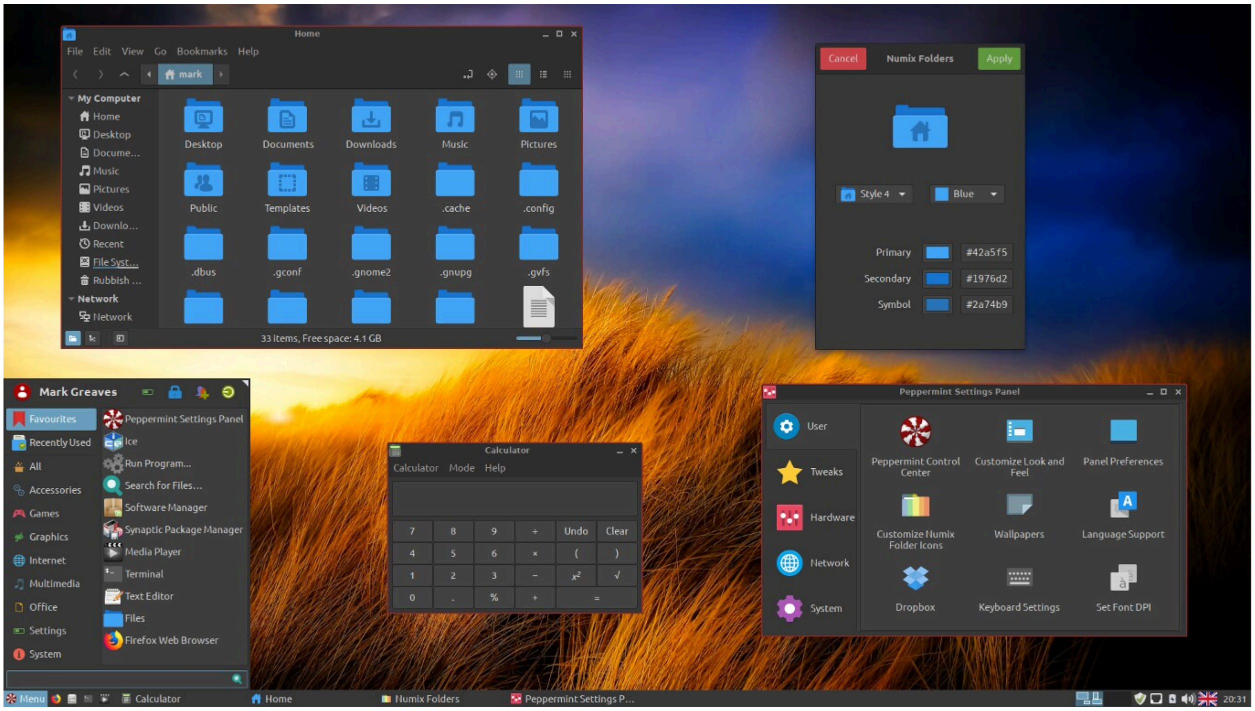
RAM: A minimum of 512 MB of RAM is required for running Bodhi Linux. However, for better performance, especially if you plan to run multiple applications simultaneously or use the Moksha Desktop Environment, it's recommended to have at least 1 GB to 2 GB of RAM.

Storage: Bodhi Linux requires a minimum of 5 GB of free disk space for installation. However, for a more comfortable experience and to accommodate additional software and data, it's advisable to have at least 20 GB of free space.

Graphics: Bodhi Linux supports a variety of graphics cards, including both open-source and proprietary drivers. It is designed to work well with most graphics hardware.

Display: A display with a resolution of at least 800x600 pixels is recommended.

Peppermint OS



33 items, Free space: 4.1 GB

Style 4 Blue

Primary #42a5f5

Secondary #1976d2

Symbol #2a74b9

Peppermint OS is a Debian- and Devuan-based (previously a Lubuntu-based) Linux distribution that aims to be fast and easy on system resources. By employing its Site Specific Browser, Peppermint integrates seamlessly with cloud and web-based applications. The distribution's other features include straight forward updates and easy step-by-step installation using the Calamares installer. The distribution once employed a hybrid LXDE/Xfce desktop environment, mixing LXDE's lxsession with Xfce's panel and application menu. Starting in 2022, Peppermint OS shifted to using the Xfce desktop, dropping the LXDE components.

Minimum System Requirements :

Processor: A modern processor (e.g., Intel x86 or AMD64) is recommended. The specific processor speed and type are not typically critical, but a dual-core processor or better is advisable for a smoother experience.

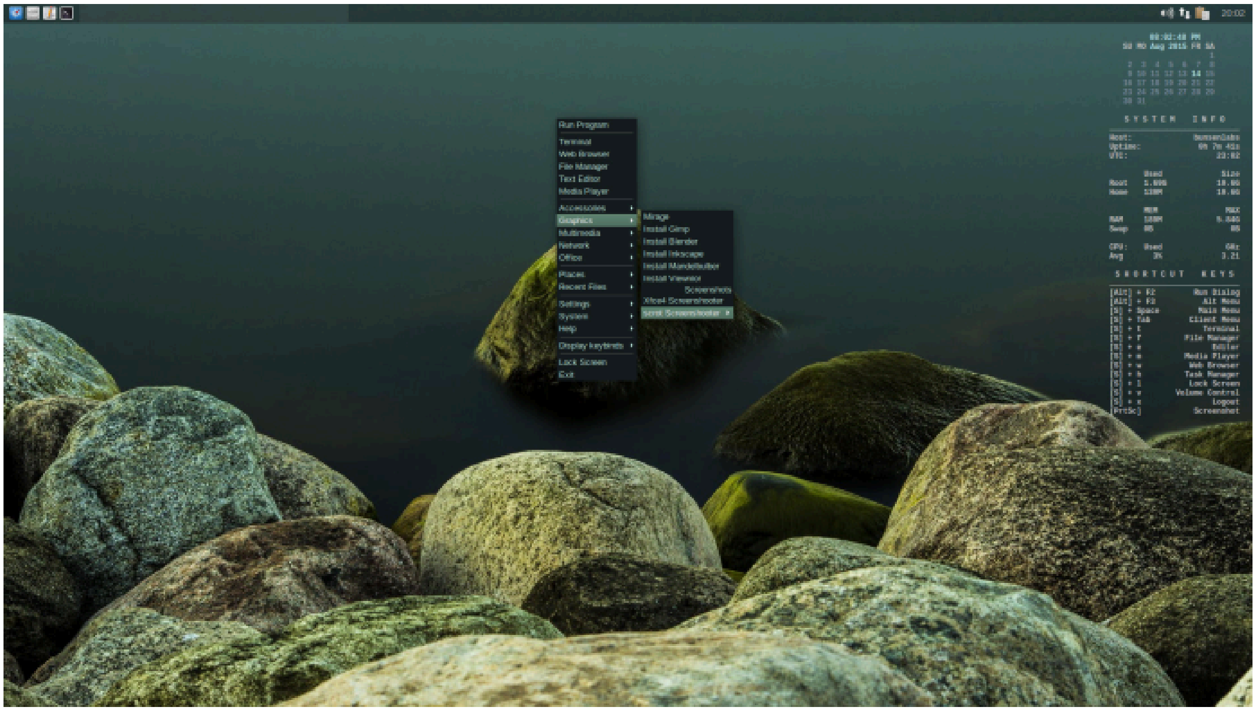
RAM: A minimum of 1 GB of RAM is recommended for running Peppermint OS. However, it's advisable to have at least 2 GB of RAM for a more responsive and enjoyable experience, especially when using multiple applications or web browsing.

Storage: A minimum of 20 GB of free disk space is recommended for the installation of Peppermint OS. However, this requirement can vary based on your specific needs and how much additional software and data you plan to store.

Graphics: Peppermint OS supports a wide range of graphics cards, including both open-source and proprietary drivers. It's designed to work well with most graphics hardware.

Display: A display with a resolution of at least 1024x768 pixels is recommended.

ChurchBang ++



```
        08:02:48 PM
 SU MO Aug 2015 FR SA
                    1
  2  3  4  5  6  7  8
  9 10 11 12 13 14 15
 16 17 18 19 20 21 22
 23 24 25 26 27 28 29
 30 31

    S Y S T E M   I N F O
Host:                BunsenLabs
Uptime:                0h 7m 41s
UTC:                       22:02

        Used                Size
Root    1.69G              18.6G
Home    128M               18.6G

        MEM                  MAX
RAM     188M               5.84G
Swap    0B                    0B

CPU:    Used                 GHz
Avg       3%                3.21

  S H O R T C U T   K E Y S
[Alt] + F2            Run Dialog
[Alt] + F3              Alt Menu
[S] + Space            Main Menu
[S] + Tab            Client Menu
[S] + t                 Terminal
[S] + f             File Manager
[S] + e                   Editor
[S] + m             Media Player
[S] + w              Web Browser
[S] + h             Task Manager
[S] + l              Lock Screen
[S] + v           Volume Control
[S] + x                   Logout
[PrtSc]               Screenshot
```

The classic minimal crunchbang feel, now with debian 12 bookworm.

Minimum System Requirements :

Processor: A modern processor, such as Intel or AMD, is recommended. The specific processor speed and type are not typically critical, but a dual-core processor or better is advisable for smoother performance.

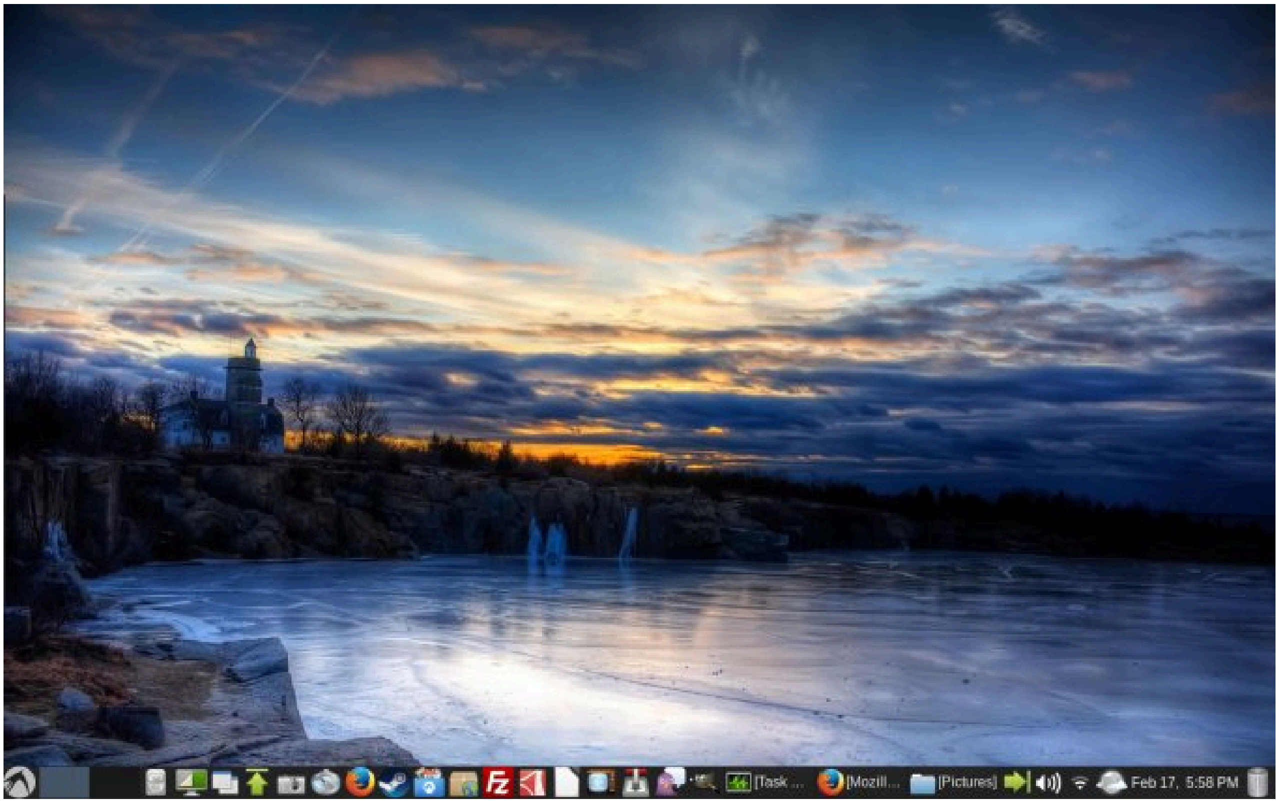
RAM: A minimum of 512 MB to 1 GB of RAM is recommended for running CrunchBang++. However, for a better experience, especially if you plan to run multiple applications or use modern web browsers, having 2 GB or more of RAM is highly beneficial.

Storage: CrunchBang++ requires a minimum of 10 GB of free disk space for installation. However, it's advisable to have at least 20 GB of free space for a more comfortable experience and to accommodate additional software and data.

Graphics: CrunchBang++ supports a variety of graphics cards, including both open-source and proprietary drivers. It's designed to work well with most graphics hardware.

Display: A display with a resolution of at least 800x600 pixels is recommended. CrunchBang++ can also adapt to higher resolutions.

LXLE



Based on the **Ubuntu/Lubuntu LTS** release, **LXLE** is a [lightweight Linux distribution]() that is resource-friendly and ideal for old PCs or systems with low system specifications. In fact, **LXLE** features prominently among the [best Linux distributions]() for old machines.

LXLE ships with an optimized **LXDE** desktop environment, which is a lightweight and minimal desktop environment that is easy on system resources while providing a neat, elegant, and intuitive UI for a smooth experience.

Minimum System Requirements :

Processor: A modern processor (e.g., Intel or AMD) capable of running a Linux kernel is recommended. The specific processor speed and type are not typically critical, but a dual-core processor or better is advisable for smoother performance.

RAM: A minimum of 512 MB to 1 GB of RAM is recommended for running LXLE. However, for a more functional and responsive system, it's highly recommended to have at least 2 GB of RAM.

Storage: LXLE requires a minimum of 8 GB of free disk space for installation. However, it's advisable to have at least 20 GB of free space for a more comfortable experience and to accommodate additional software and data.

Graphics: LXLE supports a variety of graphics cards, including both open-source and proprietary drivers. It's designed to work well with most graphics hardware.

Display: A display with a resolution of at least 1024x768 pixels is recommended.

Sparky Linux

SparkyLinux is a lightweight, fast and simple Linux distribution designed for both old and new computers featuring customized Enlightenment and LXDE desktops. It has been built on the "testing" branch of Debian GNU/Linux.

Minimum System Requirements :

Processor: A modern processor, such as Intel or AMD, capable of running a Linux kernel is recommended. The specific processor speed and type are not typically critical, but a dual-core processor or better is advisable for a smoother experience.

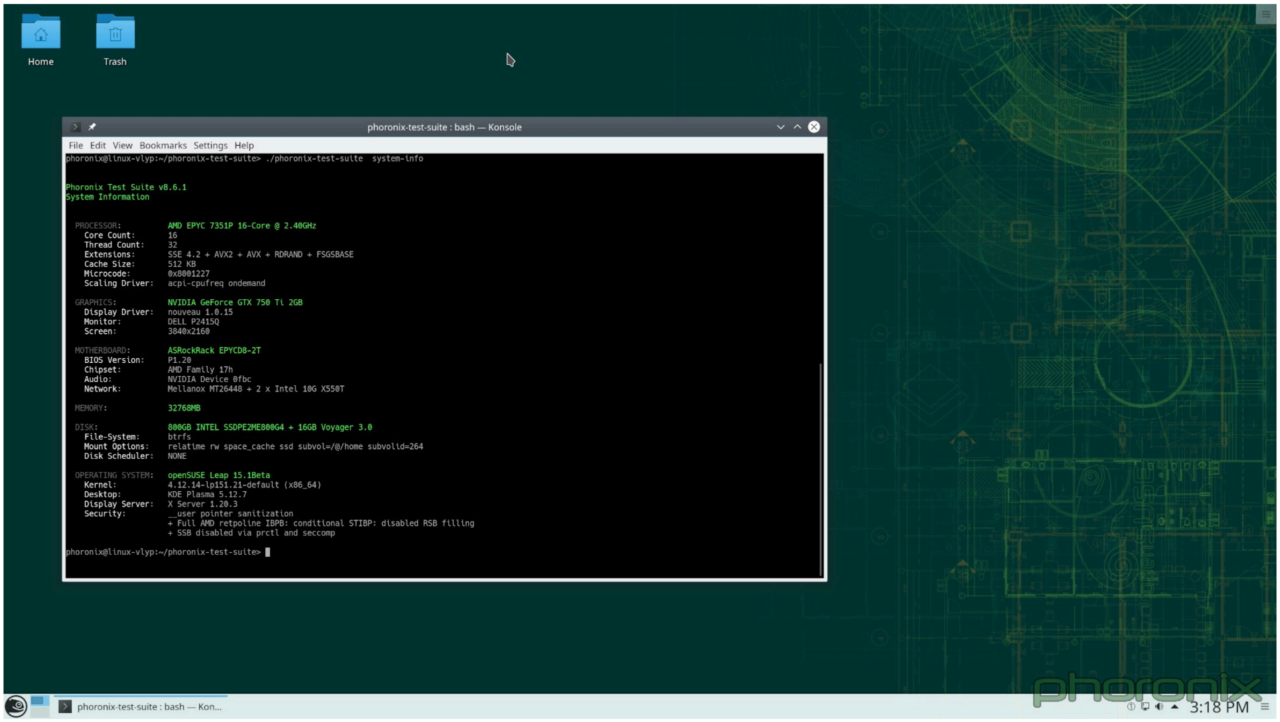
RAM: A minimum of 512 MB to 1 GB of RAM is recommended for running SparkyLinux. However, for better performance, especially if you plan to run a desktop environment, it's advisable to have at least 2 GB of RAM.

Storage: SparkyLinux requires a minimum of 10 GB of free disk space for installation. However, it's advisable to have at least 20 GB of free space for a more comfortable experience and to accommodate additional software and data.

Graphics: SparkyLinux supports a variety of graphics cards, including both open-source and proprietary drivers. It's designed to work well with most graphics hardware.

Display: A display with a resolution of at least 1024x768 pixels is recommended.

openSUSE Leap (Tested)


Home
Trash
phoronix-test-suite : bash — Konsole
File Edit View Bookmarks Settings Help
phoronix-test-suite : bash — Kon...
phoronix
3:18 PM

```
phoronix@linux-vlyp:~/phoronix-test-suite> ./phoronix-test-suite  system-info


Phoronix Test Suite v8.6.1
System Information


  PROCESSOR:          AMD EPYC 7351P 16-Core @ 2.40GHz
    Core Count:       16
    Thread Count:     32
    Extensions:       SSE 4.2 + AVX2 + AVX + RDRAND + FSGSBASE
    Cache Size:       512 KB
    Microcode:        0x8001227
    Scaling Driver:   acpi-cpufreq ondemand

  GRAPHICS:           NVIDIA GeForce GTX 750 Ti 2GB
    Display Driver:   nouveau 1.0.15
    Monitor:          DELL P2415Q
    Screen:           3840x2160

  MOTHERBOARD:        ASRockRack EPYCD8-2T
    BIOS Version:     P1.20
    Chipset:          AMD Family 17h
    Audio:            NVIDIA Device 0fbc
    Network:          Mellanox MT26448 + 2 x Intel 10G X550T

  MEMORY:             32768MB

  DISK:               800GB INTEL SSDPE2ME800G4 + 16GB Voyager 3.0
    File-System:      btrfs
    Mount Options:    relatime rw space_cache ssd subvol=/@/home subvolid=264
    Disk Scheduler:   NONE

  OPERATING SYSTEM:   openSUSE Leap 15.1Beta
    Kernel:           4.12.14-lp151.21-default (x86_64)
    Desktop:          KDE Plasma 5.12.7
    Display Server:   X Server 1.20.3
    Security:         __user pointer sanitization
                      + Full AMD retpoline IBPB: conditional STIBP: disabled RSB filling
                      + SSB disabled via prctl and seccomp

phoronix@linux-vlyp:~/phoronix-test-suite> █
```

The makers' choice for sysadmins, developers and desktop users.

openSUSE Leap is a robust and reliable Linux distribution, built from the same source code used to build SUSE Linux Enterprise (SLE) making it binary compatible with one of the most trusted operating systems to run mission critical workloads.

Minimum System Requirements :

- **Processor** : Pentium* 4 1.6 GHz or higher **processor** (Pentium 4 2.4 GHz or higher or any AMD64 or Intel64 processor recommended)

- **Main memory:** 1 GB physical RAM (at least 1.5 GB when using online repos, 2 GB recommended)
- **Hard disk:** 10 GB available disk space for a minimal install, 16 GB available for a graphical desktop (40 GB or more recommended)
- **Sound and graphics cards:** supports most modern sound and graphics cards, 800 x 600 display resolution (1024 x 768 or higher recommended)
- Booting from DVD drive or USB-Stick for installation, or support for booting over network (you need to setup PXE by yourself, look also at PXE boot installation) or an existing installation of openSUSE, more information at Installation without CD

Tested Hardware

```
localhost.localdomain
        description: Desktop Computer
        product: EP41T-UD3L
        vendor: Gigabyte Technology Co., Ltd.
        width: 64 bits
        capabilities: smbios-2.4 dmi-2.4 smp vsyscall32
        configuration: boot=normal chassis=desktop
uuid=00000000-0000-0000-0000-6cf0490bbfe1
  *-core
        description: Motherboard
        product: EP41T-UD3L
        vendor: Gigabyte Technology Co., Ltd.
        physical id: 0
        *-firmware
        description: BIOS
        vendor: Award Software International, Inc.
        physical id: 0
        version: F1
        date: 11/06/2009
        size: 128KiB
        capacity: 1MiB
        capabilities: pci pnp apm upgrade shadowing cdboot bootselect edd int13floppy360
int13floppy1200 int13floppy720 int13floppy2880 int5printscreen int9keyboard int14serial
int17printer int10video acpi usb ls120boot zipboot biosbootspecification
        *-cpu
        description: CPU
        product: Pentium(R) Dual-Core  CPU      E5400  @ 2.70GHz
        vendor: Intel Corp.
        physical id: 4
        bus info: cpu@0
        version: 6.23.10
        slot: Socket 775
        size: 2699MHz
        capacity: 4GHz
        width: 64 bits
        clock: 200MHz
        capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ht tm pbe syscall nx x86-64 constant_tsc arch_perfmon pebs bts rep_good nopl cpuid aperfmperf pni dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm xsave lahf_lm pti tpr_shadow vnmi flexpriority vpid dtherm cpufreq
        configuration: microcode=2571
        *-cache:0
        description: L1 cache
        physical id: a
        slot: Internal Cache
        size: 64KiB
        capacity: 64KiB
        capabilities: synchronous internal write-back
        configuration: level=1
        *-cache:1
        description: L2 cache
        physical id: b
        slot: External Cache
        size: 2MiB
        capacity: 2MiB
        capabilities: synchronous internal write-back
```

```
configuration: level=2
*-memory
description: System Memory
physical id: 1a
slot: System board or motherboard
size: 2GiB
*-bank:0
description: DIMM 400 MHz (2.5 ns)
physical id: 0
slot: A0
size: 2GiB
width: 2244 bits
clock: 400MHz (2.5ns)
*-bank:1
description: DIMM [empty]
physical id: 1
slot: A1
*-bank:2
description: DIMM [empty]
physical id: 2
slot: A2
*-bank:3
description: DIMM [empty]
physical id: 3
slot: A3
*-pci
description: Host bridge
product: 4 Series Chipset DRAM Controller
vendor: Intel Corporation
physical id: 100
```

```
        bus info: pci@0000:00:00.0
        version: 03
        width: 32 bits
        clock: 33MHz
        *-pci:0
        description: PCI bridge
        product: 4 Series Chipset PCI Express Root Port
        vendor: Intel Corporation
        physical id: 1
        bus info: pci@0000:00:01.0
        version: 03
        width: 32 bits
        clock: 33MHz
        capabilities: pci pm msi pciexpress normal_decode bus_master cap_list
        configuration: driver=pcieport
        resources: irq:24 ioport:c000(size=4096) memory:f8000000-fbffffff
ioport:e0000000(size=268435456)
        *-display
                description: VGA compatible controller
                product: G96C [GeForce 9500 GT]
                vendor: NVIDIA Corporation
                physical id: 0
                bus info: pci@0000:01:00.0
                logical name: /dev/fb0
                version: a1
                width: 64 bits
                clock: 33MHz
                capabilities: pm msi pciexpress vga_controller bus_master cap_list rom fb
                configuration: depth=32 driver=nouveau latency=0 resolution=1024,768
            resources: irq:25 memory:fa000000-faffffff memory:e0000000-efffffff memory:f8000000-f9ffffff ioport:cf00(size=128) memory:c0000-dffff
        *-multimedia
        description: Audio device
        product: NM10/ICH7 Family High Definition Audio Controller
        vendor: Intel Corporation
        physical id: 1b
        bus info: pci@0000:00:1b.0
        logical name: card0
        logical name: /dev/snd/controlC0
        logical name: /dev/snd/hwC0D2
        logical name: /dev/snd/pcmC0D0c
        logical name: /dev/snd/pcmC0D0p
        logical name: /dev/snd/pcmC0D1c
        logical name: /dev/snd/pcmC0D1p
        logical name: /dev/snd/pcmC0D2c
        version: 01
        width: 64 bits
        clock: 33MHz
        capabilities: pm msi pciexpress bus_master cap_list
        configuration: driver=snd_hda_intel latency=0
        resources: irq:27 memory:fdff8000-fdffbfff
        *-input:0
                product: HDA Intel Line
                physical id: 0
                logical name: input10
                logical name: /dev/input/event7
        *-input:1
                product: HDA Intel Line Out Front
                physical id: 1
                logical name: input11
                logical name: /dev/input/event8
        *-input:2
                product: HDA Intel Line Out Surround
                physical id: 2
                logical name: input12
                logical name: /dev/input/event9
        *-input:3
                product: HDA Intel Line Out CLFE
                physical id: 3
                logical name: input13
                logical name: /dev/input/event10
        *-input:4
                product: HDA Intel Line Out Side
                physical id: 4
                logical name: input14
                logical name: /dev/input/event11
        *-input:5
                product: HDA Digital PCBeep
                physical id: 5
                logical name: input7
                logical name: /dev/input/event4
                capabilities: pci
        *-input:6
                product: HDA Intel Rear Mic
                physical id: 6
                logical name: input8
                logical name: /dev/input/event5
        *-input:7
                product: HDA Intel Front Mic
                physical id: 7
                logical name: input9
                logical name: /dev/input/event6
        *-pci:1
        description: PCI bridge
        product: NM10/ICH7 Family PCI Express Port 1
        vendor: Intel Corporation
        physical id: 1c
        bus info: pci@0000:00:1c.0
        version: 01
        width: 32 bits
        clock: 33MHz
        capabilities: pci pciexpress msi pm normal_decode cap_list
        configuration: driver=pcieport
        resources: irq:16 ioport:1000(size=4096) memory:7ff00000-800fffff
    ioport:80100000(size=2097152)
        *-pci:2
        description: PCI bridge
        product: NM10/ICH7 Family PCI Express Port 4
        vendor: Intel Corporation
        physical id: 1c.3
        bus info: pci@0000:00:1c.3
        version: 01
        width: 32 bits
        clock: 33MHz
        capabilities: pci pciexpress msi pm normal_decode bus_master cap_list
        configuration: driver=pcieport
        resources: irq:19 ioport:b000(size=4096) memory:fdc00000-fdcfffff
    ioport:fdd00000(size=1048576)
        *-network
```

```
                description: Ethernet interface
                product: RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller
                vendor: Realtek Semiconductor Co., Ltd.
                physical id: 0
                bus info: pci@0000:03:00.0
                logical name: eth0
                version: 03
                serial: 6c:f0:49:0b:bf:e1
                size: 100Mbit/s
                capacity: 1Gbit/s
                width: 64 bits
                clock: 33MHz
                capabilities: pm msi pciexpress msix vpd bus_master cap_list rom ethernet
physical tp mii 10bt 10bt-fd 100bt 100bt-fd 1000bt 1000bt-fd autonegotiation
                configuration: autonegotiation=on broadcast=yes driver=r8169
driverversion=5.14.21-150500.53-default duplex=full firmware=rtl_nic/rtl8168d-2.fw
ip=192.168.1.9 latency=0 link=yes multicast=yes port=twisted pair speed=100Mbit/s
                resources: irq:19 ioport:be00(size=256) memory:fddff000-fddfffff
memory:fddf8000-fddfbfff memory:fdc00000-fdc1ffff
        *-usb:0
        description: USB controller
        product: NM10/ICH7 Family USB UHCI Controller #1
        vendor: Intel Corporation
        physical id: 1d
        bus info: pci@0000:00:1d.0
        version: 01
        width: 32 bits
        clock: 33MHz
        capabilities: uhci bus_master
        configuration: driver=uhci_hcd latency=0
```

```
resources: irq:23 ioport:ff00(size=32)
*-usbhost
        product: UHCI Host Controller
        vendor: Linux 5.14.21-150500.53-default uhci_hcd
        physical id: 1
        bus info: usb@2
        logical name: usb2
        version: 5.14
        capabilities: usb-1.10
        configuration: driver=hub slots=2 speed=12Mbit/s
*-usb:1
description: USB controller
product: NM10/ICH7 Family USB UHCI Controller #2
vendor: Intel Corporation
physical id: 1d.1
bus info: pci@0000:00:1d.1
version: 01
width: 32 bits
clock: 33MHz
capabilities: uhci bus_master
configuration: driver=uhci_hcd latency=0
resources: irq:19 ioport:fe00(size=32)
*-usbhost
        product: UHCI Host Controller
        vendor: Linux 5.14.21-150500.53-default uhci_hcd
        physical id: 1
        bus info: usb@3
        logical name: usb3
        version: 5.14
        capabilities: usb-1.10
```

```
    configuration: driver=hub slots=2 speed=12Mbit/s
*-usb:2
description: USB controller
product: NM10/ICH7 Family USB UHCI Controller #3
vendor: Intel Corporation
physical id: 1d.2
bus info: pci@0000:00:1d.2
version: 01
width: 32 bits
clock: 33MHz
capabilities: uhci bus_master
configuration: driver=uhci_hcd latency=0
resources: irq:18 ioport:fd00(size=32)
*-usbhost
    product: UHCI Host Controller
    vendor: Linux 5.14.21-150500.53-default uhci_hcd
    physical id: 1
    bus info: usb@4
    logical name: usb4
    version: 5.14
    capabilities: usb-1.10
    configuration: driver=hub slots=2 speed=12Mbit/s
*-usb:3
description: USB controller
product: NM10/ICH7 Family USB UHCI Controller #4
vendor: Intel Corporation
physical id: 1d.3
bus info: pci@0000:00:1d.3
version: 01
width: 32 bits
```

```
clock: 33MHz
capabilities: uhci bus_master
configuration: driver=uhci_hcd latency=0
resources: irq:16 ioport:fc00(size=32)
*-usbhost
        product: UHCI Host Controller
        vendor: Linux 5.14.21-150500.53-default uhci_hcd
        physical id: 1
        bus info: usb@5
        logical name: usb5
        version: 5.14
        capabilities: usb-1.10
        configuration: driver=hub slots=2 speed=12Mbit/s
*-usb:4
description: USB controller
product: NM10/ICH7 Family USB2 EHCI Controller
vendor: Intel Corporation
physical id: 1d.7
bus info: pci@0000:00:1d.7
version: 01
width: 32 bits
clock: 33MHz
capabilities: pm ehci bus_master cap_list
configuration: driver=ehci-pci latency=0
resources: irq:23 memory:fdfff000-fdfff3ff
*-usbhost
        product: EHCI Host Controller
        vendor: Linux 5.14.21-150500.53-default ehci_hcd
        physical id: 1
        bus info: usb@1
```

```
    logical name: usb1
    version: 5.14
    capabilities: usb-2.00
    configuration: driver=hub slots=8 speed=480Mbit/s
*-pci:3
description: PCI bridge
product: 82801 PCI Bridge
vendor: Intel Corporation
physical id: 1e
bus info: pci@0000:00:1e.0
version: e1
width: 32 bits
clock: 33MHz
capabilities: pci subtractive_decode bus_master cap_list
resources: ioport:d000(size=4096) memory:fde00000-fdefffff
*-communication
    description: Modem
    product: SmartLink SmartPCI563 56K Modem
    vendor: ULi Electronics Inc.
    physical id: 0
    bus info: pci@0000:04:00.0
    version: 00
    width: 32 bits
    clock: 33MHz
    capabilities: pm generic bus_master cap_list
    configuration: driver=serial latency=32
    resources: irq:20 memory:fdeff000-fdefffff ioport:de00(size=256)
*-isa
description: ISA bridge
product: 82801GB/GR (ICH7 Family) LPC Interface Bridge
```

```
vendor: Intel Corporation
physical id: 1f
bus info: pci@0000:00:1f.0
version: 01
width: 32 bits
clock: 33MHz
capabilities: isa bus_master cap_list
configuration: driver=lpc_ich latency=0
resources: irq:0
*-pnp00:00
        product: PnP device PNP0c02
        physical id: 0
        capabilities: pnp
        configuration: driver=system
*-pnp00:01
        product: PnP device PNP0b00
        physical id: 1
        capabilities: pnp
        configuration: driver=rtc_cmos
*-pnp00:02
        product: PnP device PNP0700
        physical id: 2
        capabilities: pnp
*-pnp00:03
        product: PnP device PNP0501
        physical id: 3
        capabilities: pnp
        configuration: driver=serial
*-pnp00:04
        product: PnP device PNP0400
        physical id: 4
        capabilities: pnp
        configuration: driver=parport_pc
*-pnp00:05
        product: PnP device PNP0c02
        physical id: 5
        capabilities: pnp
        configuration: driver=system
*-pnp00:06
        product: PnP device PNP0c02
        physical id: 6
        capabilities: pnp
        configuration: driver=system
*-pnp00:07
        product: PnP device PNP0c01
        physical id: 7
        capabilities: pnp
        configuration: driver=system
*-ide
description: IDE interface
product: NM10/ICH7 Family SATA Controller [IDE mode]
vendor: Intel Corporation
physical id: 1f.2
bus info: pci@0000:00:1f.2
logical name: scsi0
version: 01
width: 32 bits
clock: 66MHz
capabilities: ide pm isa_compat_mode bus_master cap_list emulated
configuration: driver=ata_piix latency=0
```

```
        resources: irq:19 ioport:1f0(size=8) ioport:3f6 ioport:170(size=8) ioport:376
ioport:f900(size=16)
        *-disk:0
                description: ATA Disk
                product: STM3250318AS
                physical id: 0.0.0
                bus info: scsi@0:0.0.0
                logical name: /dev/sda
                version: CC37
                serial: 9VM5QZH1
                size: 232GiB (250GB)
                capabilities: gpt-1.00 partitioned partitioned:gpt
                configuration: ansiversion=5 guid=8c89d2e0-413e-4560-91c4-275dda6785ea
logicalsectorsize=512 sectorsize=512
        *-volume:0
                description: BIOS Boot partition
                vendor: EFI
                physical id: 1
                bus info: scsi@0:0.0.0,1
                logical name: /dev/sda1
                serial: bbe15e05-682c-42aa-beaf-81d9791daad1
                capacity: 8191KiB
                capabilities: nofs
        *-volume:1
                description: LVM Physical Volume
                vendor: Linux
                physical id: 2
                bus info: scsi@0:0.0.0,2
                logical name: /dev/sda2
                serial: A0WggT-H9xN-bImV-fIfs-lTHM-bqdR-iupU93
                size: 232GiB
                capabilities: multi lvm2
        *-disk:1
                description: ATA Disk
                product: ST3250318AS
                physical id: 0.1.0
                bus info: scsi@0:0.1.0
                logical name: /dev/sdb
                version: CC37
                serial: 9VM4L0HM
                size: 232GiB (250GB)
                capabilities: partitioned partitioned:dos
                configuration: ansiversion=5 logicalsectorsize=512 sectorsize=512
signature=33028e5e
        *-volume
                description: Linux filesystem partition
                physical id: 1
                bus info: scsi@0:0.1.0,1
                logical name: /dev/sdb1
                capacity: 232GiB
                capabilities: primary
        *-serial
        description: SMBus
        product: NM10/ICH7 Family SMBus Controller
        vendor: Intel Corporation
        physical id: 1f.3
        bus info: pci@0000:00:1f.3
        version: 01
        width: 32 bits
        clock: 33MHz
```

```
      configuration: driver=i801_smbus latency=0
      resources: irq:19 ioport:500(size=32)
*-input:0
      product: Power Button
      physical id: 1
      logical name: input3
      logical name: /dev/input/event0
      capabilities: platform
*-input:1
      product: Power Button
      physical id: 2
      logical name: input4
      logical name: /dev/input/event1
      capabilities: platform
*-input:2
      product: PC Speaker
      physical id: 3
      logical name: input6
      logical name: /dev/input/event3
      capabilities: isa
```

Run

Deploy with Nginx on Debian

Deploying a Flask application with Nginx on a Debian-based system involves several steps, including installing necessary software, configuring Flask, setting up Gunicorn as the application server, configuring Nginx to proxy requests to Gunicorn, and finally, securing the application with SSL. Here's a step-by-step guide to achieve this:

Step 1: Update and Upgrade
Ensure your System is up to date by running the following commands:

```
sudo apt update
sudo apt upgrade
```

Step 2: Install Necessary Software

```
sudo apt install python3-pip python3-venv nginx
```

Step 3: Set Up a Python Virtual Environment and Install Flask and Gunicorn

Create a directory for your Flask application and set up a virtual environment:

```
mkdir /path/to/your/app
cd /path/to/your/app
python3 -m venv venv
source venv/bin/activate
pip install flask gunicorn
```

Step 4: Configure Gunicorn

Create a Gunicorn service file, for example

`cd /etc/systemd/system/yourapp_gunicorn.service`

```
[Unit]
Description=gunicorn daemon for your Flask application
After=network.target

[Service]
User=username  # Replace with the username under which your app will run
Group=groupname  # Replace with the group name
WorkingDirectory=/path/to/your/app
ExecStart=/path/to/your/app/venv/bin/gunicorn -b localhost:8000 -w 4 yourapp:app

[Install]
WantedBy=multi-user.target
```

Replace `username`, `groupname`, `/path/to/your/app`, and `yourapp` with appropriate values.

Step 5: Start and Enable Gunicorn Service

```
sudo systemctl start yourapp_gunicorn
sudo systemctl enable yourapp_gunicorn
```

Step 6: Configure Nginx

Create a Nginx server block configuration file, for example

```
cd /etc/nginx/sites-available/yourapp
```

```
server {
     listen 80;
     server_name your_domain.com;  # Replace with your domain name

     location / {
     proxy_pass http://localhost:8000;
     proxy_set_header Host $host;
     proxy_set_header X-Real-IP $remote_addr;
     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
     }
}
```

Step 7: Enable the Nginx Server Block and Restart Nginx

```
sudo ln -s /etc/nginx/sites-available/yourapp
/etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

Step 8: Configure SSL (Optional)

If you want to secure your application with SSL, you can obtain an SSL certificate from a trusted provider and configure Nginx accordingly.

Remember to replace placeholders such as `yourapp`, `your_domain.com`, `username`, `groupname`, and paths with your actual application and server details.

Finally, you should be able to access your Flask application by visiting your domain name in a web browser.

Deploy with Apache on Debian

Deploying a Flask application with Apache on a Debian-based system involves several steps, including installing necessary software, configuring Flask, setting up Apache, configuring a VirtualHost for Apache, and optionally securing the application with SSL. Here's a step-by-step guide to achieve this:

Step 1: Update and Upgrade

Ensure your system is up to date by running the following commands:

```
sudo apt update
sudo apt upgrade
```

Step 2: Install Necessary Software

```
sudo apt install python3-pip python3-venv apache2
libapache2-mod-wsgi-py3
```

Step 3: Set Up a Python Virtual Environment and Install Flask

Create a directory for your Flask application and set up a virtual environment:

```
mkdir /path/to/your/app
```

```
cd /path/to/your/app
python3 -m venv venv && source venv/bin/activate && pip install flask
```

Step 4: Configure Apache

Create a Flask WSGI script. In your app directory, create a file named `wsgi.py`:

```
from yourapp import app

if __name__ == "__main__":
     app.run()
```

Replace `yourapp` with the actual name of your Flask application.

Step 5: Configure a VirtualHost for Apache

Create a VirtualHost configuration file for your Flask application. Create a file named `yourapp.conf` in the `/etc/apache2/sites-available/` directory:

apache

```
<VirtualHost *:80>
     ServerName your_domain.com  # Replace with your domain name
     ServerAdmin webmaster@localhost
     WSGIDaemonProcess yourapp user=your_username group=your_group
threads=5 home=/path/to/your/app
     WSGIScriptAlias / /path/to/your/app/wsgi.py

```

```
    <Directory /path/to/your/app>
    WSGIProcessGroup yourapp
    WSGIApplicationGroup %{GLOBAL}
    Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Replace `your_domain.com`, `your_username`, `your_group`, and `/path/to/your/app` with appropriate values.

Step 6: Enable the VirtualHost and Restart Apache

```
sudo a2ensite yourapp
```

```
sudo systemctl restart apache2
```

Deploy with Nginx on openSUSE (recomended)

This configuration Tested on openSUSE Leap 15.5

Note : These settings were made for the system I tested on, you can change these settings according to your taste

```
 cd /home/$USER/kygnus_nas
```

Directory file list:

- **app**
- opennas-1.0.0-1.noarch.rpm
- README.md
- venv
- install
- openNAS.service
- requirements.txt
- openNAS
- script

```
sudo vi /etc/nginx/conf.d/myflaskapp.conf
```

```
server {
     listen 80;
     server_name your_domain_or_IP;

     location / {
     proxy_pass http://127.0.0.1:5005;
     proxy_set_header Host $host;
     proxy_set_header X-Real-IP $remote_addr;
     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
     }
}
```

```
sudo mkdir /etc/nginx/sites-available
```

```
sudo mkdir /etc/nginx/sites-enabled
```

```
sudo ln -s /etc/nginx/sites-available/myflaskapp
/etc/nginx/sites-enabled/
```

Change Directory to Application Directory :

```
cd /w/kygnus_nas/app
```

```
gunicorn -w 4 -b 127.0.0.1:5005 main:app
```

Remove

```
sudo iptables -F && sudo rm
/etc/systemd/system/opennas.service && sudo rm -rf
/var/log/opennas && sudo rm -rf /etc/opennas  && rm -rf
/opt/opennas && echo "opannas Removed [ Successfully ]"
```

CommanLine

```
cd /tmp && wget
https://gitlab.com/KooshaYeganeh/opennas/archive/refs/heads/main.zip
&& unzip main.zip && mv opennas-main opennas && mv opennas /opt && cd
/opt/opennas
```

Create softlink of main File

```
cd /usr/bin && sudo ln -s /opt/opennas/opennas opennas && echo
"softlink Created"
```

Then just type opennas to run application

Install with .rpm File

change Directory to kygnus-nas and then install .rom File

```
sudo rpm -ivh opennas-1.0.0-1.noarch.rpm
```

Software information

python

Python is a high level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2.

Python consistently ranks as one of the most popular programming languages.

Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

clamAV

Clam AntiVirus (ClamAV) is a free and open source command line interface antivirus software program. It is used to detect trojans and malicious softwares including viruses. It can scan files quickly and can scan over one million viruses and trojans. One of its main uses is to scan emails on mail gateways. ClamAV is supported by the following Linux Operating Systems Ubuntu (16.04, 18.04), Debian (7,8), CentOS (6,7). In this blog we will discuss how to install and use ClamAV in Ubuntu.

Installing ClamAV

In order to install ClamAV on your machine, first of all run the following command to update your system

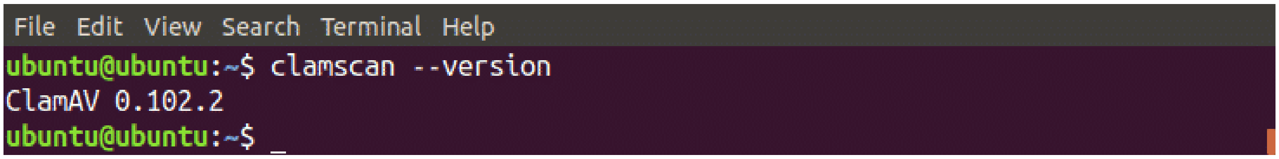
```
sudo apt-get update
```

After updating your machine, now run the following command to install ClamAV

```
sudo apt-get install clamav clamav-daemon
```

Now ClamAV has been installed on your machine. Run the following command to check whether it has been installed or not

```
clamscan --version
```



```
ubuntu@ubuntu:~$ clamscan --version
ClamAV 0.102.2
ubuntu@ubuntu:~$ _
```

If the above command gives the version of ClamAV then it has been installed successfully.

Updating the ClamAV Signature Database

So far you have installed ClamAV on your machine, now you need to update the ClamAV signature database. To install ClamAV signature database, follow the given steps

- Stop freshclam service
- Update the signature database (Two methods)
  - Update by running the command in the terminal
  - Update by downloading daily.cvd file
- Start freshclam service

First step is to stop the clamav-freshclam service by running the following command in the terminal window

```
sudo systemctl stop clamav-freshclam
```

In the Second Step, now we have to update the signature database manually. There are two ways to do so. First Method involves to run the following command in the terminal

```
sudo freshclam
```

This command will install the signature database in your machine. If this command does not work, then goto the following link to download signature database file

https://database.clamav.net/daily.cvd

Now create a directory named “clamav”, if does not exist, in a specific location by running the following command

```
sudo mkdir /var/lib/clamav
```

And move the downloaded file in this location by running the following command

```
cp daily.cvd /var/lib/clamav/daily.cvd
```

Now the **third step** is to start the clamav-freshclam service by running the following command.

```
sudo systemctl start clamav-freshclam
```

RootKitHunter

Rkhunter, short for Rootkit Hunter, is an open-source security tool that scans Linux and Unix systems for rootkits, backdoors, and other possible security threats. It is an essential addition to any security-conscious user's toolkit. This article will provide a step-by-step guide on how to install and use Rkhunter on Ubuntu 22.04 and Ubuntu 20.04 LTS Linux systems.

To follow this guide, ensure that you have the following:

- A system running Ubuntu 22.04 or Ubuntu 20.04.
- A user with sudo privileges.

1.Update Your System

Before installing any new software, it is crucial to update your system. Run the following commands to update your package list and upgrade the installed packages:

```
sudo apt update  && sudo apt upgrade
```

2.Install Rkhunter

Rkhunter packages is available in the official Ubuntu repositories. You can quickly Install it using the following command:

```
sudo apt install rkhunter
```

3.Update Rkhunter Data Files

To get the latest malware definitions and improve the accuracy of Rkhunter scans, update the data files using the following command:

```
sudo rkhunter --update
```

4.Configure Rkhunter

To configure Rkhunter, edit its configuration file located at /etc/rkhunter.conf. You can use any text editor, such as nano or vim, to edit the file:

```
sudo vi /etc/rkhunter.conf
```

Here are some recommended configurations:

Enable automatic updates by uncommenting and setting

UPDATE_MIRRORS to 1

Configure the download mirrors by uncommenting and setting

MIRRORS_MODE to 0

Enable email notifications by uncommenting and setting

MAIL-ON-WARNING with your email address

```
UPDATE_MIRRORS=1
MIRRORS_MODE=0
MAIL-ON-WARNING="youremail@example.com"
```

5. Run a System Scan

To run an initial system scan, execute the following command:

```
sudo rkhunter --check --skip-keypress
```

This command will run Rkhunter with a check option, scanning your system for potential threats. The `--skip-keypress` flag avoids the need for pressing a key after every test.

Once the scan is complete, you can view the results in the log file at /var/log/rkhunter.log.

Chkrootkit

chkrootkit: Main program which checks operating system binaries for rootkit modifications to learn if the code was adulterated.

ifpromisc.c: checks if the interface is in promiscuous mode. If a network interface is in promiscuous mode, it can be used by an attacker or malicious software to capture the network traffic to later analyze it.

chklastlog.c: checks for lastlog deletions. Lastlog is a command which shows information on last logins. An attacker or rootkit may modify the file to avoid detection if the sysadmin checks this command to learn information on logins.

chkwtmp.c: checks for wtmp deletions. Similarly, to the previous script, chkwtmp checks the file wtmp, which contains information on users' logins to try to detect modifications on it in case a rootkit modified the entries to prevent detection of intrusions.

check_wtmpx.c: This script is the same as the above but Solaris systems.

chkproc.c: checks for signs of trojans within LKM (Loadable Kernel Modules).

chkdirs.c: has the same function as the above, checks for trojans within kernel modules.

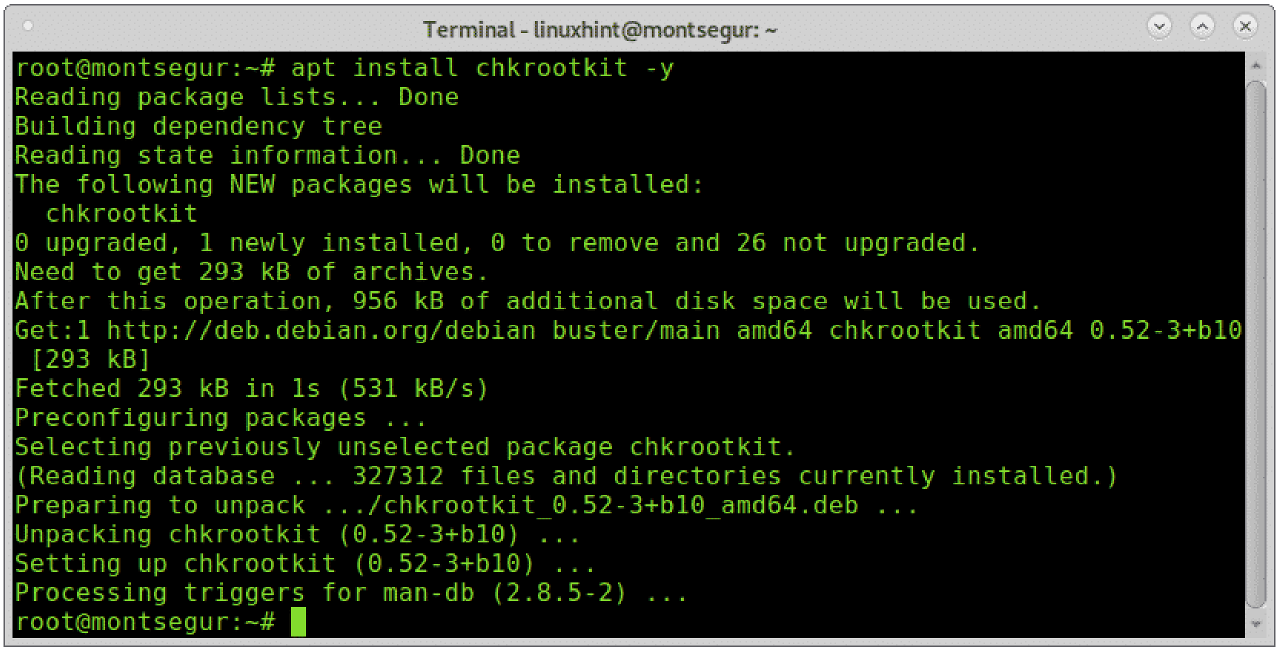
strings.c: quick and dirty strings replacement aiming to hide the nature of the rootkit.

chkutmp.c: this is similar to chkwtmp but checks the utmp file instead.

All the scripts mentioned above are executed when we run chkrootkit.

To begin installing chkrootkit on Debian and based Linux distributions run:

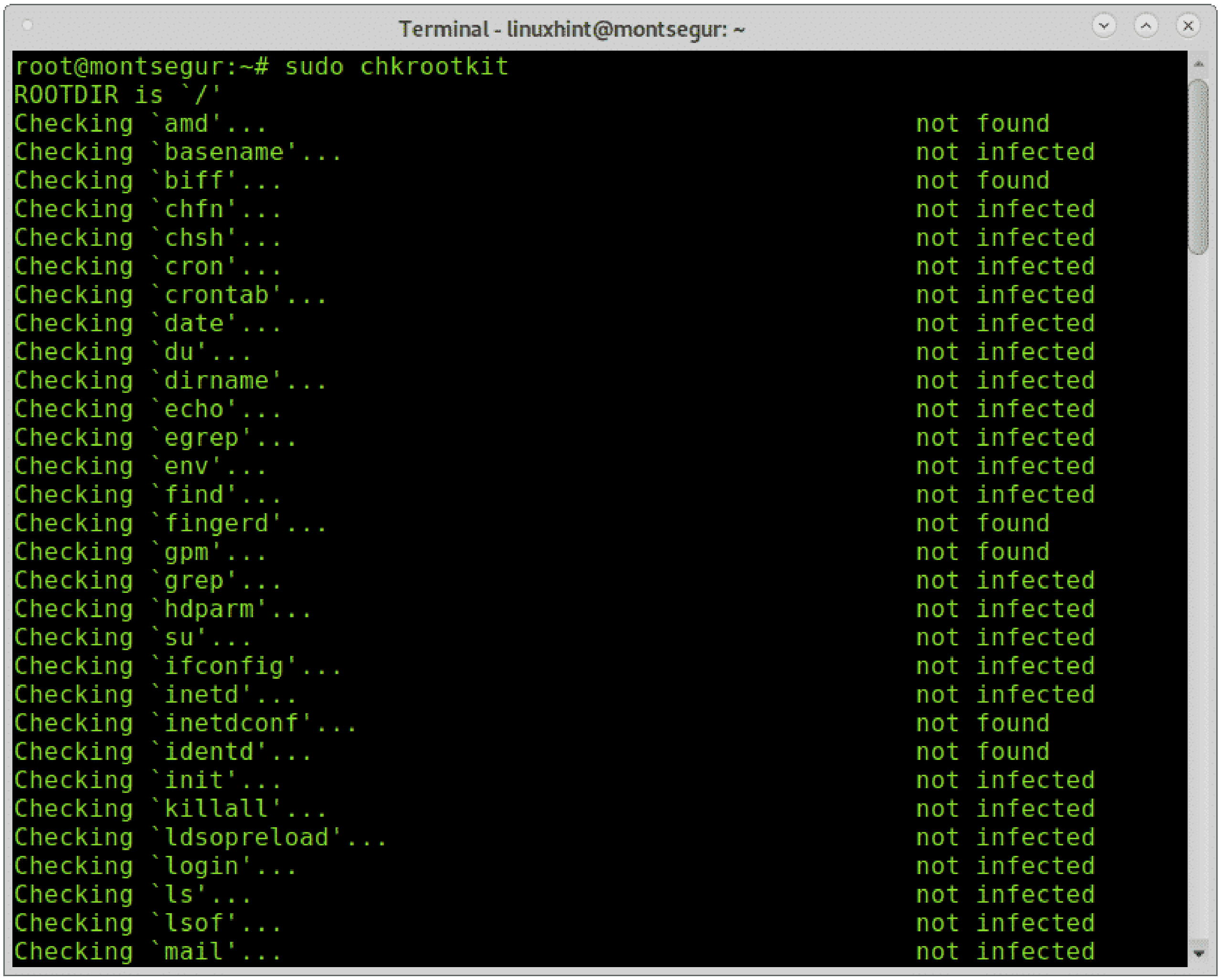
```
apt install chkrootkit -y
```



```
root@montsegur:~# apt install chkrootkit -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  chkrootkit
0 upgraded, 1 newly installed, 0 to remove and 26 not upgraded.
Need to get 293 kB of archives.
After this operation, 956 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian buster/main amd64 chkrootkit amd64 0.52-3+b10
 [293 kB]
Fetched 293 kB in 1s (531 kB/s)
Preconfiguring packages ...
Selecting previously unselected package chkrootkit.
(Reading database ... 327312 files and directories currently installed.)
Preparing to unpack .../chkrootkit_0.52-3+b10_amd64.deb ...
Unpacking chkrootkit (0.52-3+b10) ...
Setting up chkrootkit (0.52-3+b10) ...
Processing triggers for man-db (2.8.5-2) ...
root@montsegur:~#
```

Once installed to run it execute:

```
sudo chkrootkit
```



```
root@montsegur:~# sudo chkrootkit
ROOTDIR is `/'
Checking `amd'...                                           not found
Checking `basename'...                                      not infected
Checking `biff'...                                          not found
Checking `chfn'...                                          not infected
Checking `chsh'...                                          not infected
Checking `cron'...                                          not infected
Checking `crontab'...                                       not infected
Checking `date'...                                          not infected
Checking `du'...                                            not infected
Checking `dirname'...                                       not infected
Checking `echo'...                                          not infected
Checking `egrep'...                                         not infected
Checking `env'...                                           not infected
Checking `find'...                                          not infected
Checking `fingerd'...                                       not found
Checking `gpm'...                                           not found
Checking `grep'...                                          not infected
Checking `hdparm'...                                        not infected
Checking `su'...                                            not infected
Checking `ifconfig'...                                      not infected
Checking `inetd'...                                         not infected
Checking `inetdconf'...                                     not found
Checking `identd'...                                        not found
Checking `init'...                                          not infected
Checking `killall'...                                       not infected
Checking `ldsopreload'...                                   not infected
Checking `login'...                                         not infected
Checking `ls'...                                            not infected
Checking `lsof'...                                          not infected
Checking `mail'...                                          not infected
```

You can also export the results to a file using the following syntax:

```
sudo chkrootkit > results
```

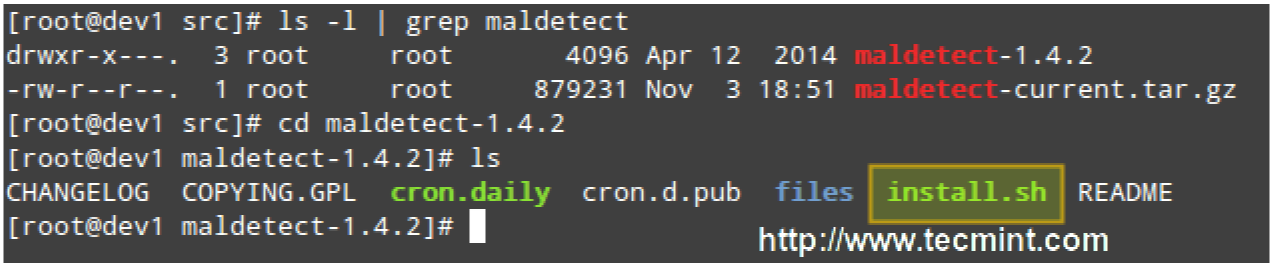
Linux Malware Detect (LMD)

LMD is not available from online repositories but is distributed as a tarball from the project's web site. The tarball containing the source code of the latest version is always available at the following link, where it can be downloaded with wget command:

```
wget http://www.rfxn.com/downloads/maldetect-current.tar.gz
```

Then we need to unpack the tarball and enter the directory where its contents were extracted. Since the current version is 1.6.4, the directory is maldetect-1.6.4. There we will find the installation script, install.sh.

```
tar xvf maldetect-current.tar.gz && cd maldetect-1.6.*
```



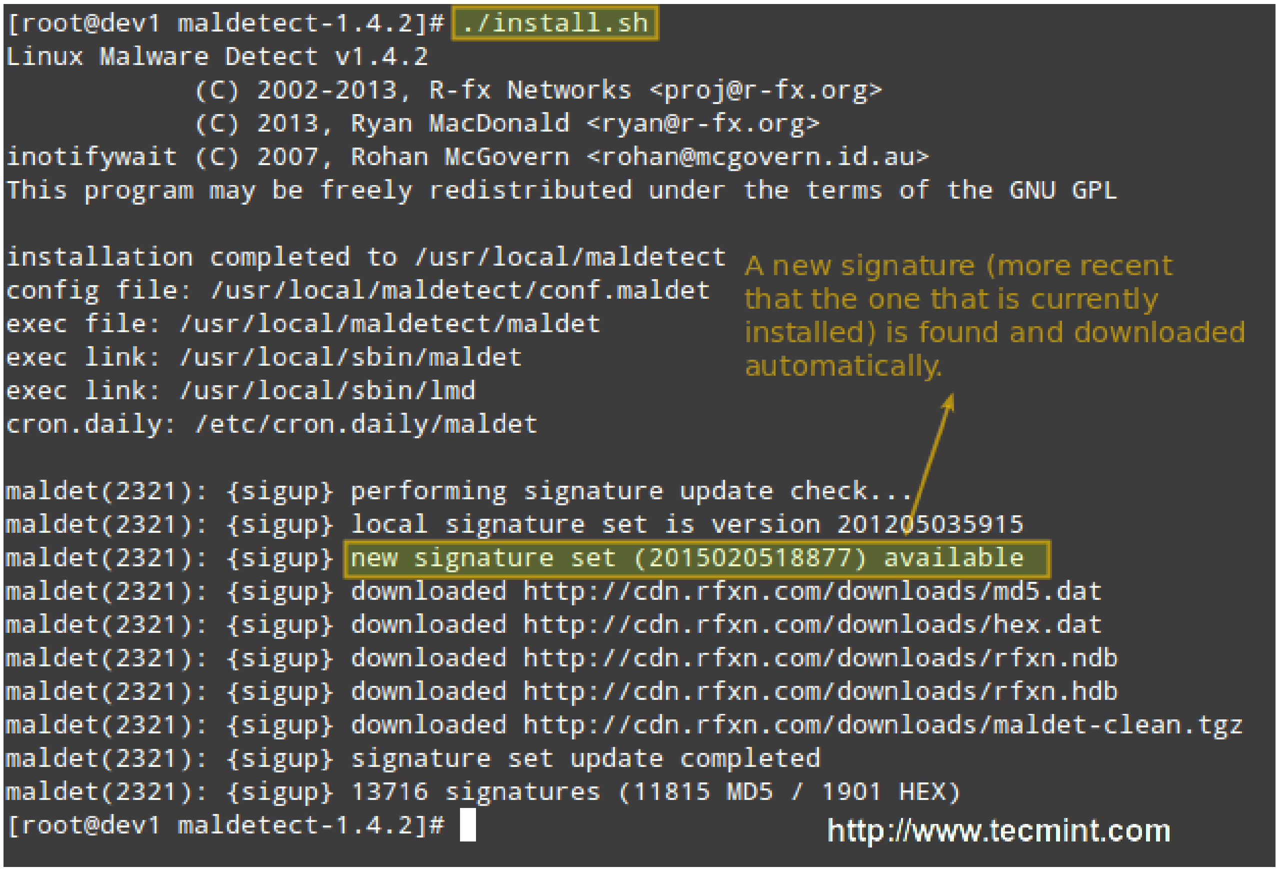
```
[root@dev1 src]# ls -l | grep maldetect
drwxr-x---.  3 root     root       4096 Apr 12  2014 maldetect-1.4.2
-rw-r--r--.  1 root     root     879231 Nov  3 18:51 maldetect-current.tar.gz
[root@dev1 src]# cd maldetect-1.4.2
[root@dev1 maldetect-1.4.2]# ls
CHANGELOG  COPYING.GPL  cron.daily  cron.d.pub  files  install.sh  README
[root@dev1 maldetect-1.4.2]#
```

If we inspect the installation script, which is only 75 lines long (including comments), we will see that it not only installs the tool but also performs a pre-check to see if the default installation directory (/usr/local/maldetect) exists. If not, the script creates the installation directory before proceeding.

Finally, after the installation is completed, a daily execution via cron is scheduled by placing the cron.daily script (refer to the image above) in /etc/cron.daily. This helper script will, among other things, clear old temporary data, check for new LMD releases, and scan the default Apache and web control panels (i.e., CPanel, DirectAdmin, to name a few) default data directories.

That being said, run the installation script as usual:

```
sudo ./install.sh
```



```
[root@dev1 maldetect-1.4.2]# ./install.sh
Linux Malware Detect v1.4.2
            (C) 2002-2013, R-fx Networks <proj@r-fx.org>
            (C) 2013, Ryan MacDonald <ryan@r-fx.org>
inotifywait (C) 2007, Rohan McGovern <rohan@mcgovern.id.au>
This program may be freely redistributed under the terms of the GNU GPL

installation completed to /usr/local/maldetect
config file: /usr/local/maldetect/conf.maldet
exec file: /usr/local/maldetect/maldet
exec link: /usr/local/sbin/maldet
exec link: /usr/local/sbin/lmd
cron.daily: /etc/cron.daily/maldet

maldet(2321): {sigup} performing signature update check...
maldet(2321): {sigup} local signature set is version 201205035915
maldet(2321): {sigup} new signature set (2015020518877) available
maldet(2321): {sigup} downloaded http://cdn.rfxn.com/downloads/md5.dat
maldet(2321): {sigup} downloaded http://cdn.rfxn.com/downloads/hex.dat
maldet(2321): {sigup} downloaded http://cdn.rfxn.com/downloads/rfxn.ndb
maldet(2321): {sigup} downloaded http://cdn.rfxn.com/downloads/rfxn.hdb
maldet(2321): {sigup} downloaded http://cdn.rfxn.com/downloads/maldet-clean.tgz
maldet(2321): {sigup} signature set update completed
maldet(2321): {sigup} 13716 signatures (11815 MD5 / 1901 HEX)
[root@dev1 maldetect-1.4.2]#
```

A new signature (more recent that the one that is currently installed) is found and downloaded automatically.

Configuring Linux Malware Detect

The configuration of LMD is handled through /usr/local/maldetect/conf.maldet and all options are well commented to make configuration a rather easy task. In case you get stuck, you can also refer to /maldetect-1.6.4/README for further instructions.

In the configuration file you will find the following sections, enclosed inside square brackets:

1. EMAIL ALERTS
2. QUARANTINE OPTIONS
3. SCAN OPTIONS
4. STATISTICAL ANALYSIS
5. MONITORING OPTIONS

Each of these sections contains several variables that indicate how LMD will behave and what features are available.

1. Set email_alert=1 if you want to receive email notifications of malware inspection results. For the sake of brevity, we will only relay mail to local system users, but you can explore other options such as sending mail alerts to the outside as well.
2. Set email_subj=”Your subject here” and email_addr=username@localhost if you have previously set email_alert=1.
3. With quar_hits, the default quarantine action for malware hits (0 = alert only, 1 = move to quarantine & alert) you will tell LMD what to do when malware is detected.
4. quar_clean will let you decide whether you want to clean string-based malware injections. Keep in mind that a string signature is, by definition, “a contiguous byte sequence that potentially can match many variants of a malware family”.

5. quar_susp, the default suspend action for users with hits, will allow you to disable an account whose owned files have been identified as hits.
6. clamav_scan=1 will tell LMD to attempt to detect the presence of ClamAV binary and use as default scanner engine. This yields an up to four times faster scan performance and superior hex analysis. This option only uses ClamAV as the scanner engine, and LMD signatures are still the basis for detecting threats.

Summing up, the lines with these variables should look as follows in /usr/local/maldetect/conf.maldet:

```
email_alert=1
email_addr=gacanepa@localhost
email_subj="Malware alerts for $HOSTNAME - $(date +%Y-%m-%d)"
quar_hits=1
quar_clean=1
quar_susp=1
clam_av=1
```

Nmap

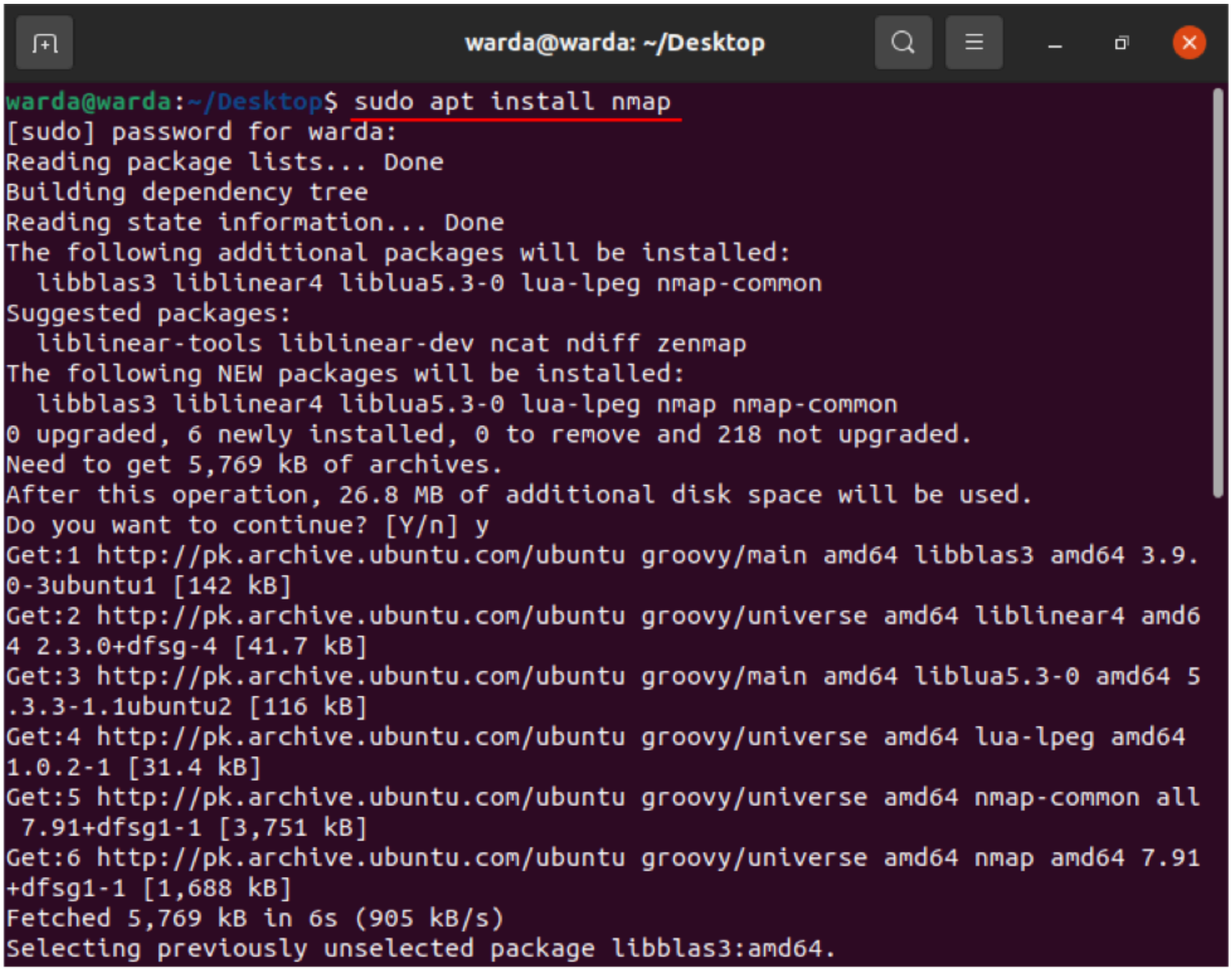
The Network Mapper, also known as “Nmap,” is a versatile, flexible, and famous tool used to manage and secure networks. It helps administrators to map their networks and security scans.

Through command-line prompt Nmap, the tool provides multiple features such as exploring networks, finding open ports, ping sweeps, OS detection, and many more.

This post is focusing on how to use the Nmap command in the terminal with detailed examples. Let's check them one by one:

How to install Nmap Command

Before exploring with Nmap commands, the Nmap scanner tool must have installed on your system. So, if it is not downloaded yet, get it by opening up the terminal and executing the following command:

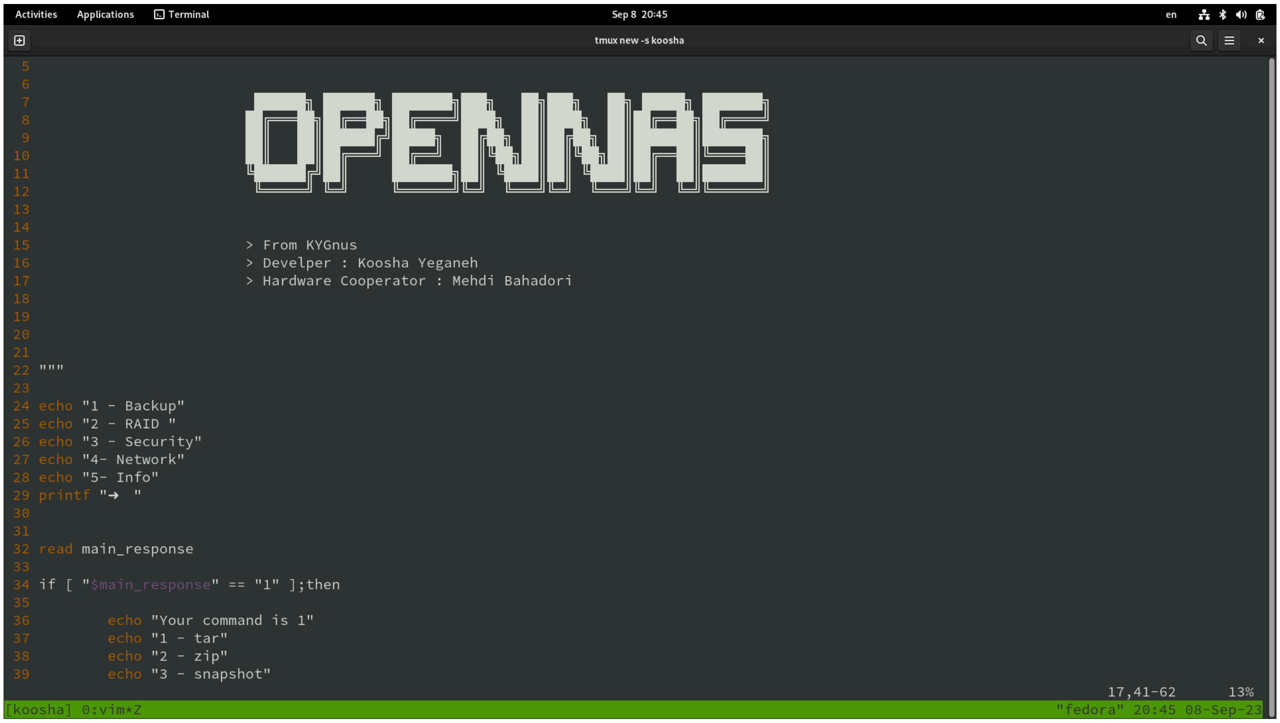


```
warda@warda:~/Desktop$ sudo apt install nmap
[sudo] password for warda:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libblas3 liblinear4 liblua5.3-0 lua-lpeg nmap-common
Suggested packages:
  liblinear-tools liblinear-dev ncat ndiff zenmap
The following NEW packages will be installed:
  libblas3 liblinear4 liblua5.3-0 lua-lpeg nmap nmap-common
0 upgraded, 6 newly installed, 0 to remove and 218 not upgraded.
Need to get 5,769 kB of archives.
After this operation, 26.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://pk.archive.ubuntu.com/ubuntu groovy/main amd64 libblas3 amd64 3.9.
0-3ubuntu1 [142 kB]
Get:2 http://pk.archive.ubuntu.com/ubuntu groovy/universe amd64 liblinear4 amd6
4 2.3.0+dfsg-4 [41.7 kB]
Get:3 http://pk.archive.ubuntu.com/ubuntu groovy/main amd64 liblua5.3-0 amd64 5
.3.3-1.1ubuntu2 [116 kB]
Get:4 http://pk.archive.ubuntu.com/ubuntu groovy/universe amd64 lua-lpeg amd64
1.0.2-1 [31.4 kB]
Get:5 http://pk.archive.ubuntu.com/ubuntu groovy/universe amd64 nmap-common all
 7.91+dfsg1-1 [3,751 kB]
Get:6 http://pk.archive.ubuntu.com/ubuntu groovy/universe amd64 nmap amd64 7.91
+dfsg1-1 [1,688 kB]
Fetched 5,769 kB in 6s (905 kB/s)
Selecting previously unselected package libblas3:amd64.
```

CommandLine Tools

The required software in the command line version is the same as the web version, only Bash has been added here

Bash is the GNU Project's shell—the Bourne Again SHell. This is an sh-compatible shell that incorporates useful features from the Korn shell (ksh) and the C shell (csh). It is intended to conform to the IEEE POSIX P1003.2/ISO 9945.2 Shell and Tools standard. It offers functional improvements over sh for both programming and interactive use. In addition, most sh scripts can be run by Bash without modification.



```
                              > From KYGnus
                              > Develper : Koosha Yeganeh
                              > Hardware Cooperator : Mehdi Bahadori




"""

echo "1 - Backup"
echo "2 - RAID "
echo "3 - Security"
echo "4- Network"
echo "5- Info"
printf "➜  "


read main_response

if [ "$main_response" == "1" ];then

        echo "Your command is 1"
        echo "1 - tar"
        echo "2 - zip"
        echo "3 - snapshot"
                                                                                                                      17,41-62      13%
[koosha] 0:vim*Z                                                                                                "fedora" 20:45 08-Sep-23
```

Netdata (Monitoring) :

Netdata is an open source performance and health monitoring system for Linux and MacOS. Metrics in Netdata are organized in collections called charts. Charts have a purpose and a scope. This makes Netdata extremely useful for learning the underlying technologies. It helps us to understand how things work and what is available. Real - time alarms are also supported in Netdata and these alarms can be setup on any metrics or any combination of them. Netdata is very resource efficient and you can control its resource consumption

openSUSE installer :

```
sudo zypper -n install netdata
```

```
sudo systemctl restart netdata
```

```
sudo systemctl restart n
```

```
sudo vi /etc/nginx/conf.d/netdata.conf
```

```
server {
     listen 80;
     server_name netdata.techviewleo.com;

location / {
     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
     proxy_set_header X-Forwarded-Host $host;
     proxy_set_header X-Forwarded-Server $host;
     proxy_pass http://netdata;
     proxy_http_version 1.1;
     proxy_pass_request_headers on;
     proxy_set_header Connection "keep-alive";
     proxy_store off;
     }
}
```

Access Netdata

By default, Netdata listens to port 19999. We are going to access Netdata from a web browser. Therefore, open a web browser and browse the URL shown.

```
http://server-ip:19999
```

Desktops

Gnome

GNOME (GNU Network Object Model Environment, pronounced gah-NOHM) is a graphical user interface (GUI) and set of computer desktop applications for users of the Linux operating system. It's intended to make a Linux operating system easy to use for non-programmers and generally corresponds to the Windows desktop interface and its most common set of applications. In fact, GNOME allows the user to select one of several desktop appearances. With GNOME, the user interface can, for example, be made to look like Windows or like Mac OS. In addition, GNOME includes a set of the same type of applications found in Microsoft Office: a word processor, a spreadsheet program, a database manager, a presentation developer, a Web browser and an email program.

GNOME is derived from a long-running volunteer effort under the auspices of the Free Software Foundation, the organization founded by Richard Stallman. Stallman and fellow members of the Free Software Foundation believe that software source code should always be public and open to change so that it can continually be improved by others. GNOME is in part an effort to make Linux a viable alternative to Windows so that the desktop operating system market is not controlled by a single vendor. GNU is the Free Software Foundations own operating system and set of applications. Linux, the operating system, was developed by Linus Torvalds who, assisted by contributors, added a kernel to additional operating system components from GNU.

GNOME comes with an object request broker (ORB) supporting the Common Object Request Broker Architecture (CORBA) so that GNOME programs and programs from other operating system platforms in a network will be able to interoperate. GNOME also includes a widget library that programmers can use to develop applications that use the GNOME user interface. In addition to a desktop version, GNOME also comes as a user interface and set of applications for mobile devices.

```
sudo apt install gnome/stable -y
```

Resource usage :

The resource usage of the GNOME desktop environment can vary depending on several factors, including the version of GNOME, your hardware specifications, and the specific applications you have running. However, I can provide you with some general information on the typical resource usage of GNOME.

Memory (RAM) Usage:

- GNOME is known for being a bit more resource-intensive compared to some other lightweight desktop environments. It can use anywhere from 600MB to 1.5GB or more of RAM, depending on the version and the number of running applications.

CPU Usage :

- GNOME typically doesn't consume a significant amount of CPU when idle. CPU usage can increase when running applications or performing tasks, but it should generally be responsive and not overly taxing on modern hardware.

Graphics Requirements:

- GNOME relies on a 3D-accelerated graphics stack for its visual effects. This means that it performs best when you have a compatible graphics card with good driver support. Without proper graphics acceleration, the desktop may fall back to a "GNOME Classic" mode with reduced graphical effects.

Disk Space:

- The disk space requirements for GNOME itself are relatively modest. The GNOME desktop itself doesn't consume much space. However, additional space will be needed for installed applications, user files, and system updates.

Performance Optimization:

- If you find that GNOME is consuming too many resources on your system, you can take several steps to optimize its performance:

  - Disable or uninstall unnecessary GNOME extensions.

  - Use lighter applications where possible.

  - Ensure that you are using the latest graphics drivers for your hardware.

  - Adjust the screen resolution and effects settings.

  - Consider using a lighter desktop environment if resource usage is a significant concern.

It's important to note that the resource usage of GNOME can vary between different versions (e.g., GNOME 3.36, GNOME 40) and distributions that customize it. To get specific resource usage information for your system, you can use system monitoring tools like `top`, `htop`, or the system monitor application that comes with your Linux distribution. These tools will provide real-time information about CPU, memory, and other resource usage.

KDE

KDE stands for K Desktop Environment/Kool Desktop Environment, one of the most popular open-source desktop environments in Linux systems. Linux users are always confused when choosing between KDE and GNOME, that's why it is recommended to check the differences between KDE and GNOME when selecting the right desktop environment for your needs.

However, in recent years, KDE has been able to replace the GNOME desktop environment with its progress. KDE has added modern and efficient features such as support for Wayland and KRunner to its facilities to improve its graphical user interface and provides the possibility of creative and efficient use of the Linux system. Especially KDE Plasma5 which offers the latest innovations and features to create an amazing user experience.

Since KDE is one of the most famous and popular desktop environments for Linux users, in this article we decided to teach how to install the KDE desktop environment in Linux so that you can benefit from its unique features of the KDE desktop environment.

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install kde-plasma-desktop -y
```

```
sudo reboot
```

XFCE

Xfce is a popular lightweight desktop environment for UNIX-like operating systems. It is designed to be fast and light on the utilization of system resources such as memory and CPU. In doing so, Xfce provides optimal performance and is usually recommended for old computers and PCs with low resource specifications.

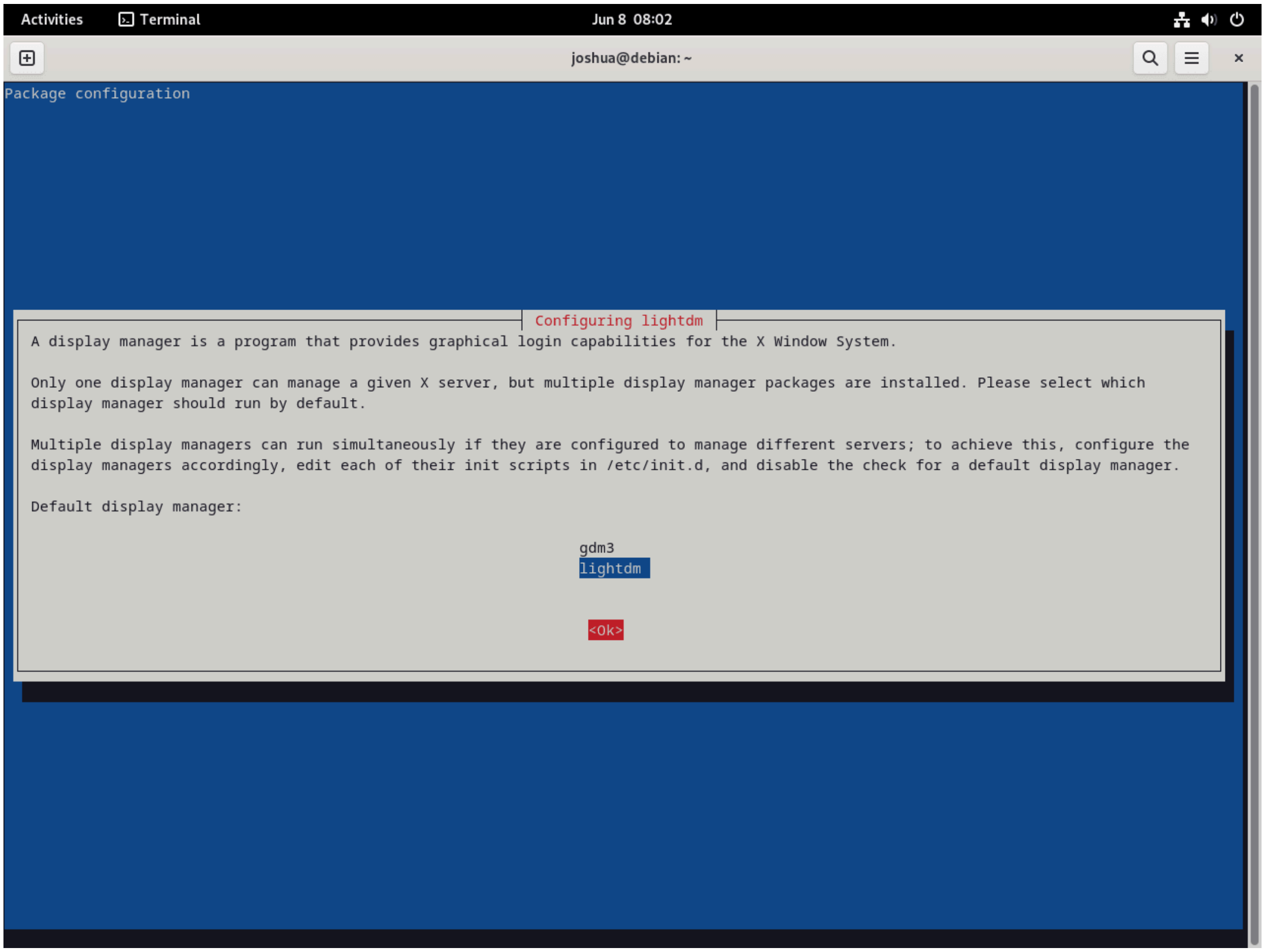
```
sudo apt update && sudo apt upgrade
```

```
sudo apt install task-xfce-desktop
```

The installation of XFCE is generally quicker than its counterparts due to its lightweight nature. You will encounter a configuration screen for the LightDM display manager during this process. It's worth noting that XFCE is designed to run efficiently with

LightDM. If XFCE is going to be your default desktop environment, ensure to select *LightDM* in this step.

Use the TAB key to navigate the interface, select <Ok>, then press the ENTER key to confim your selection.



```
Package configuration

                         Configuring lightdm
A display manager is a program that provides graphical login capabilities for the X Window System.

Only one display manager can manage a given X server, but multiple display manager packages are installed. Please select which
display manager should run by default.

Multiple display managers can run simultaneously if they are configured to manage different servers; to achieve this, configure the
display managers accordingly, edit each of their init scripts in /etc/init.d, and disable the check for a default display manager.

Default display manager:

                              gdm3
                              lightdm

                               <Ok>
```

```
sudo reboot
```

Performance

Desktop

I3 (Tiling Window Manager)

Written in C language, the i3wm (i3 Windows Manager) is a lightweight, easy-to-configure, and hugely popular tiling windows manager. Unlike the conventional desktop environment, a tiling manager provides just sufficient functionality to arrange windows on your screen in an easy and appealing manner suited for your workflow.

i3 is a minimalist tiling manager that intelligently arranges the windows on your screen in a seamless non-overlapping manner. Other tiling managers include xmonad and wmii.

In this guide, we will explain how to install and use the i3 Windows manager on Linux desktop systems.

Benefits of i3 Windows Manager

Unlike X windows managers such as Fluxbox, KWin, and enlightenment, i3 comes with a bag of goodies that we have listed below for a smooth desktop experience.

1. Resource Friendly

Unlike the fully-featured desktop environments such as GNOME, i3 windows manager is quite minimalistic and is designed for simplicity and efficiency. With low resource utilization, it makes up for a fast tiling Windows manager and leaves your system with plenty of RAM and CPU for other applications.

2. Flexibility

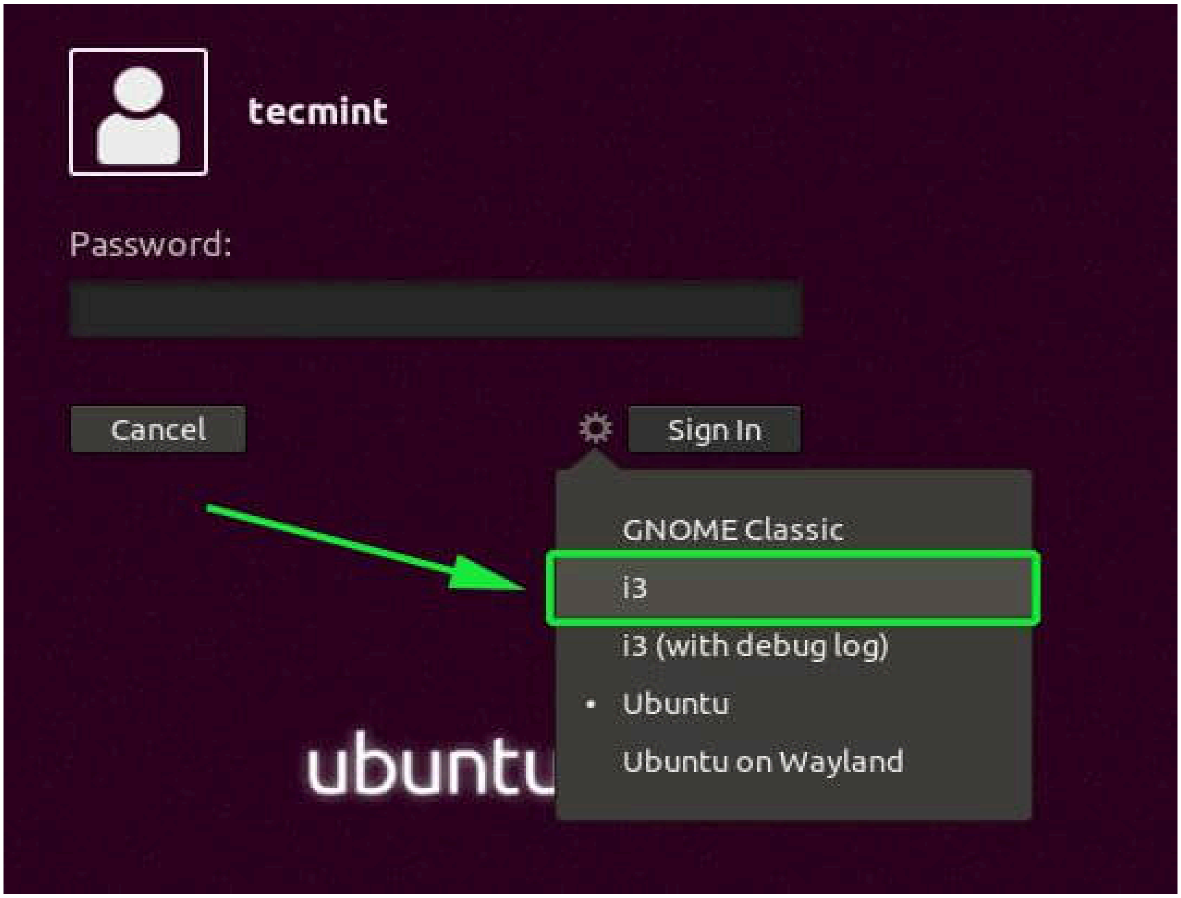
Apart from having the ability to automatically arrange windows in a neat and organized manner, i3 is fully configurable and you can tweak a few settings to match your preferred screen layout. Using external tools, you can enhance the appearance by selecting the background image, adjusting the transparency and window fading effect, and enabling desktop notifications.

3. Easy to Navigate Between Workspaces

The i3 tiling manager provides an easy and quick way to switch between workspaces thanks to a wide array of keyboard shortcuts that you can easily configure. You can seamlessly group Windows to suit your workflow, which enhances your productivity.

```
sudo apt update && sudo apt install i3
```

Once installed, you will need to restart your system and click on the small gear wheel at the login window and select the ‘i3’ option as shown.



XFCE :

1 – Install htop for Monitoring:

```
sudo apt-get install htop
```

2 – Enable Compositor (Xfwm4):

Open "Window Manager Tweaks" from the XFCE settings.

Go to the "Compositor" tab and enable it:

```
xfconf-query -c xfwm4 -p /general/use_compositing -s true
```

3 – Disable Window Effects:

Still in "Window Manager Tweaks," go to the "Accessibility" tab and disable "Enable display compositing."

```
xfconf-query -c xfwm4 -p /general/use_compositing -s false
```

4 – Adjust Swappiness (Reduce Disk Swapping):

Edit the sysctl.conf file:

```
sudo vi /etc/sysctl.conf
```

Add or modify the following line:

```
vm.swappiness = 10
```

Apply changes:

```
sudo sysctl -p
```

5 – Disable Unnecessary Services:

Use systemctl to list and disable services:

```
systemctl list-units --type=service
sudo systemctl stop <service_name>
sudo systemctl disable <service_name>
```

6 – Install Preload for Faster Application Launch:

```
sudo apt-get install preload -y
```

7 – Optimize Startup Applications:

Open "Session and Startup" from XFCE settings.

Disable unnecessary startup applications.

8 – Use Lighter Applications:

Replace resource-intensive applications with lighter alternatives.

9 – Optimize Graphics Drivers:

Ensure you have the appropriate graphics drivers installed.

10 – Tweak Power Management Settings:

Open "Power Manager" from XFCE settings and adjust settings for performance.

11 – Use Zswap for Compressed Swap:

Edit the GRUB configuration:

```
sudo vi /etc/default/grub
```

Add the following to the GRUB_CMDLINE_LINUX_DEFAULT line:

```
zswap.enabled=1 zswap.compressor=lz4
```

Update GRUB and reboot:

```
sudo update-grub
sudo reboot
```

12 – Clean up Temporary Files:

```
sudo apt-get autoclean
```

```
sudo apt-get autoremove
```

13 – Update System and Packages:

```
sudo apt-get update && sudo apt-get upgrade -y
```

KDE Plasma

Optimizing the KDE Plasma desktop environment in Linux involves fine-tuning settings, improving performance, and customizing the user experience. Here are some tips and commands to help optimize KDE Plasma:

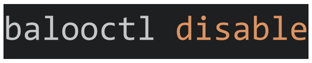
1. Adjust Desktop Effects:

- Open "System Settings" > "Display and Monitor" > "Compositor."

- Disable or tweak desktop effects based on your preference and hardware capabilities.

2. Disable Baloo File Indexing:

- Baloo is the file indexer in KDE. If you don't use desktop search extensively, you can disable it:



```
balooctl disable
```

3. Optimize KWin (Window Manager):

- Open "System Settings" > "Window Management" > "KWin Scripts."
- Disable unnecessary scripts and effects.

4. Adjust Window Management Settings:

- Open "System Settings" > "Window Management."
- Tweak settings such as Window Behavior, Task Switcher, and Window Rules based on your preferences.

5. Reduce Startup Applications:

- Open "System Settings" > "Startup and Shutdown" > "Autostart."
- Disable unnecessary autostart applications.

6. Optimize Fonts:

- Open "System Settings" > "Fonts."
- Adjust font settings and anti-aliasing based on your preference.

7. Optimize Desktop Effects via Command Line:

- Use the following command to enable or disable desktop effects:

```
kwriteconfig5 --file kwinrc --group Compositing --key Enabled false
```

8. Reduce Animation Duration:

- Open "System Settings" > "Workspace" > "Desktop Behavior" > "Desktop Effects."
- Reduce the duration of animations for a snappier feel.

9. Optimize Power Management:

- Open "System Settings" > "Power Management."
- Adjust power management settings based on your preferences and hardware.

10. Clean up Temporary Files:

```
sudo apt-get autoclean
```

```
sudo apt-get autoremove
```

11. Update System and Packages:

```
sudo apt-get update
sudo apt-get upgrade
```

12. Graphics Drivers:

- Ensure that you have the appropriate graphics drivers installed for your hardware. Consider using proprietary drivers for better performance.

13. KDE Neon Specific:

- If you are using KDE Neon, ensure that your system is updated regularly, as KDE Neon provides the latest KDE Plasma updates.

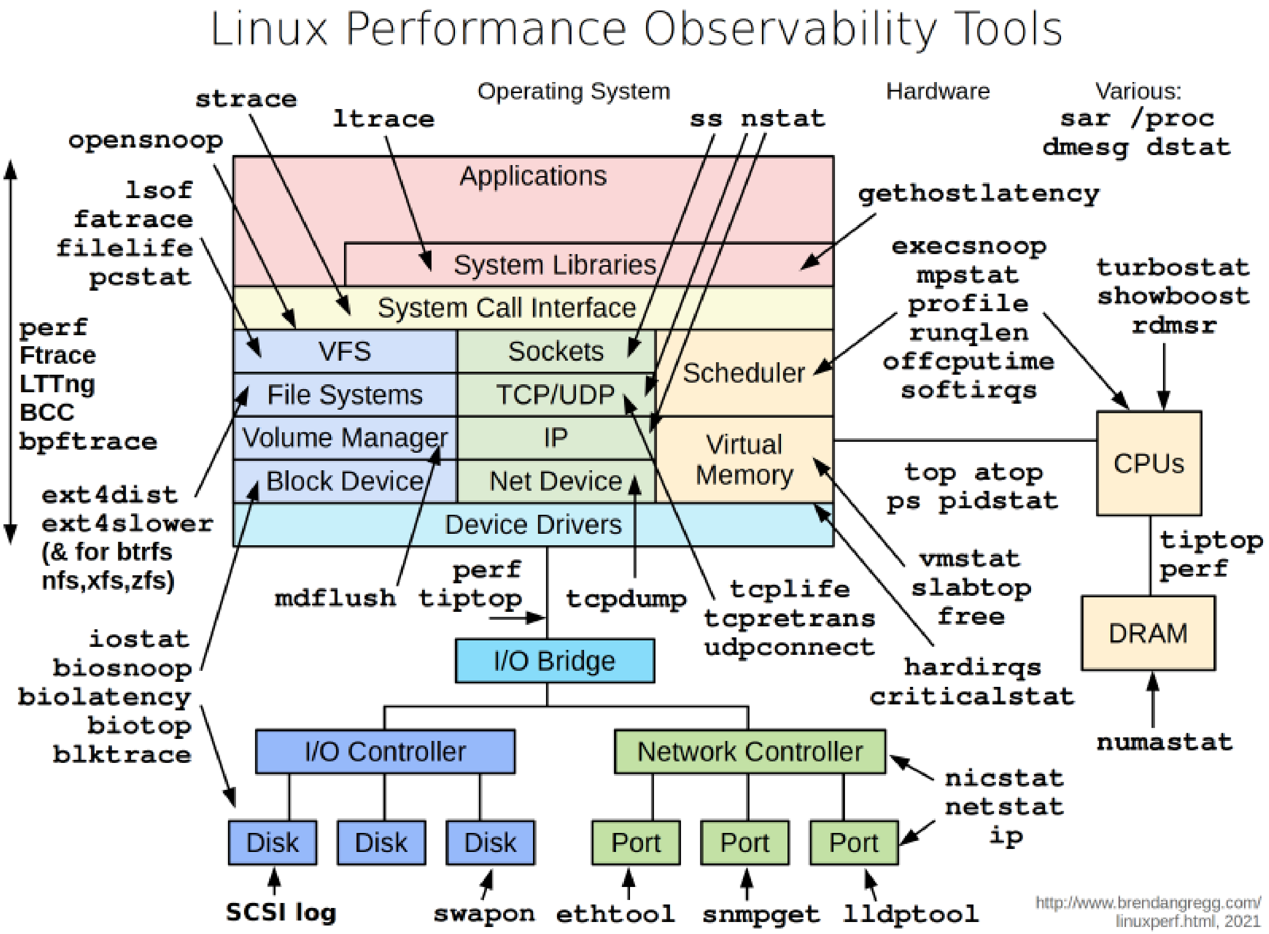
14. Disk Optimization:

- Periodically check your disk space and optimize it if needed. Remove unnecessary files and old kernels to free up space.

15. Use Lightweight Widgets and Themes:

- Choose lightweight widgets and themes to reduce resource usage.

Remember to monitor your system after making changes to ensure that the adjustments have the desired effect. The impact of optimizations may vary based on your specific hardware and use case.


Linux Performance Observability Tools
Operating System
Hardware
Various:
sar /proc
dmesg dstat
strace
ltrace
ss nstat
opensnoop
lsof
fatrace
filelife
pcstat
perf
Ftrace
LTTng
BCC
bpftrace
ext4dist
ext4slower
(& for btrfs
nfs,xfs,zfs)
iostat
biosnoop
biolatency
biotop
blktrace
Applications
System Libraries
System Call Interface
VFS
Sockets
Scheduler
File Systems
TCP/UDP
Volume Manager
IP
Virtual
Memory
Block Device
Net Device
Device Drivers
gethostlatency
execsnoop
mpstat
profile
runqlen
offcputime
softirqs
turbostat
showboost
rdmsr
CPUs
top atop
ps pidstat
tiptop
perf
vmstat
slabtop
free
DRAM
hardirqs
criticalstat
numastat
mdflush
perf
tiptop
tcpdump
tcplife
tcpretrans
udpconnect
I/O Bridge
I/O Controller
Network Controller
nicstat
netstat
ip
Disk
Disk
Disk
Port
Port
Port
SCSI log
swapon
ethtool
snmpget
lldptool
http://www.brendangregg.com/
linuxperf.html, 2021

Resources

https://learnvalley.org/product/lfs426/

https://www.redhat.com/sysadmin/tune-linux-tips

https://www.akitaonrails.com/2017/01/17/optimizing-linux-for-slow-computers

https://github.com/sn99/Optimizing-linux

https://www.site24x7.com/learn/linux/linux-performance-optimization.html

https://wiki.archlinux.org/title/improving_performance

https://www.linux-magazine.com/Issues/2006/65/Orca/(language)/eng-US

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/performance_tuning_guide/index

https://www.linux-magazine.com/Issues/2018/212/Linux-on-Old-Hardware/(offset)/12/(language)/eng-US

https://github.com/sn99/Optimizing-linux

https://github.com/hawshemi/Linux-Optimizer

https://www.ibm.com/docs/en/was/8.5.5?topic=SSEQTP_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/tprf_tunelinux.htm

https://www.linkedin.com/pulse/linux-performance-tuning-reza-bojnordi/

System

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/performance_tuning_guide/chap-red_hat_enterprise_linux-performance_tuning_guide-tuned
http://www.yolinux.com/TUTORIALS/LinuxTutorialOptimization.html
https://vlang.io/
https://www.linkedin.com/advice/0/how-do-you-use-linux-kernel-parameters-modules
https://github.com/sn99/Optimizing-linux
https://tekneed.com/linux-kernel-optimization-managing-the-linux-kernel/
https://www.phoronix.com/
https://freedesktop.org/wiki/Software/systemd/Optimizations/
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_basic_system_settings/optimizing-systemd-to-shorten-the-boot-time_configuring-basic-system-settings
https://www.thegoodpenguin.co.uk/blog/reducing-boot-time-with-systemd/

Contact :

```
Koosha Yeganeh : Devops & Founder
Hamed Fard : Network Administrator
Website : http://kooshayeganeh.github.io
Github : https://github.com/KooshaYeganeh
GitLab : https://gitlab.com/users/KooshaYeganeh
Gmail : kooshakooshadv@gmail.com
Gmail : hamed.hfaz@gmail.com
```