

OOSP_INDIVIDUAL_ASSIGNMENT

ABEL GETACHEW
BDU-1601018
SECTION-A

Netrunner OS



Submission: Last-Submission Date: 24/08/2017E.C

What is operating system?

An **operating system (OS)** is **system software** that manages **computer hardware** and **software** resources, and provides common **services** for **computer programs**.

Time-sharing operating systems **schedule tasks** for efficient use of the system and may also include accounting software for cost allocation of **processor time**, **mass storage**, peripherals, and other resources.

For hardware functions such as **input and output** and **memory allocation**, the operating system acts as an intermediary between programs and the computer hardware,^{[1][2]} although the application code is usually executed directly by the hardware and frequently makes **system calls** to an OS function or is **interrupted** by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to **web servers** and **supercomputers**.

It is a System software that manages all the resources of the computing device.

- Acts as an interface between the software and different parts of the computer or the computer hardware.
- Manages the overall resources and operations of the computer.
- Controls and monitors the execution of all other programs that reside in the computer, which also includes application programs and other system software of the computer.
- Examples of Operating Systems are Windows, Linux, macOS, Android, iOS, etc.

Generally, An operating system is difficult to define, but has been called "the **layer of software** that manages a computer's resources for its users and their **applications**". Operating systems include the software that is always running, called a **kernel**—but can include other software as well. The two other types of programs that can run on a computer are **system programs**—which are associated with the operating system, but may not be part of the kernel—and applications—all other software.

There are three main purposes that an operating system fulfills:

- Operating systems allocate resources between different applications, deciding when they will receive **central processing unit (CPU)** time or space in **memory**. On modern personal computers, users often want to run several applications at once. In order to ensure that one program cannot monopolize the computer's limited hardware resources, the operating system gives each application a share of the resource, either in time (CPU) or space (memory). The operating system also must isolate applications from each other to protect them from errors and security vulnerabilities in another application's code, but enable communications between different applications.

- Operating systems provide an interface that abstracts the details of accessing [hardware](#) details (such as physical memory) to make things easier for programmers. [Virtualization](#) also enables the operating system to mask limited hardware resources; for example, [virtual memory](#) can provide a program with the illusion of nearly unlimited memory that exceeds the computer's actual memory.
- Operating systems provide common services, such as an interface for accessing network and disk devices. This enables an application to be run on different hardware without needing to be rewritten. Which services to include in an operating system varies greatly, and this functionality makes up the great majority of code for most operating systems.

What is virtualization?

Virtualization is a technology that allows you to create virtual, simulated environments from a single, physical machine. Through this process, IT professionals can make use out of their previous investments and optimize a physical machine's full capacity by distributing resources that are traditionally bound to hardware across many different environments.

Used for decades, virtualization is a powerful technology within IT infrastructure that can be used to increase efficiency, retain flexibility, and improve scalability. Because multiple operating systems can share the same physical hardware, virtualization can improve resource use, reduce costs associated with physical maintenance, and boost security through isolated systems.

Whether you're a virtualization administrator running test environments on your workstation or a large organization running a multitude of virtual machines (VMs) across your hybrid cloud platform, virtualization plays a key role in modern IT infrastructure and workloads.

Why is Virtualization Useful?

The techniques and features that Oracle VirtualBox provides are useful in the following scenarios:

Running multiple operating systems simultaneously. Oracle VirtualBox enables you to run more than one OS at a time. This way, you can run software written for one OS on another, such as Windows software on Linux or a Mac, without having to reboot to use it. Since you can configure what kinds of virtual hardware should be presented to each such OS, you can install an old OS such as DOS or OS/2 even if your real computer's hardware is no longer supported by that OS.

Easier software installations. Software vendors can use virtual machines to ship entire software configurations. For example, installing a complete mail server solution on a real machine can be a tedious task. With Oracle VirtualBox, such a complex setup, often called an appliance, can be packed into a virtual machine. Installing and running a mail server becomes as easy as importing such an appliance into Oracle VirtualBox.

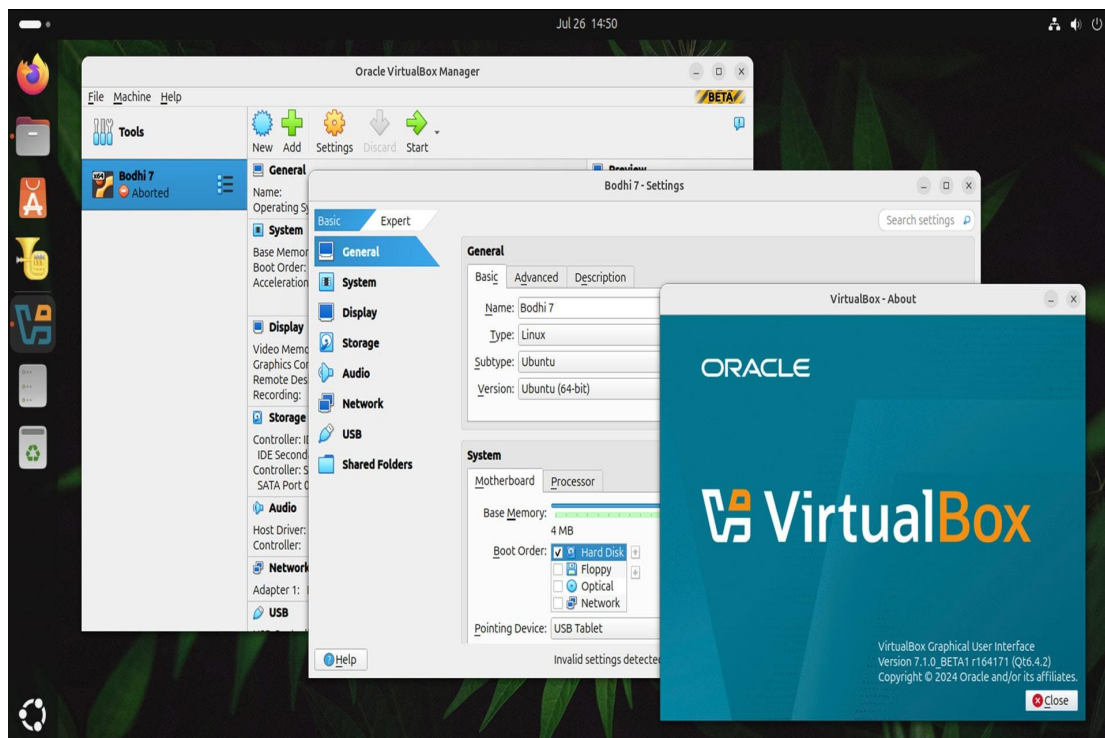
Testing and disaster recovery. Once installed, a virtual machine and its virtual hard disks can be considered a container that can be arbitrarily frozen, woken up, copied, backed up, and transported between hosts.

Using virtual machines enables you to build and test a multinode networked service, for example. Issues with networking, operating system, and software configuration can be investigated easily.

In addition to that, with the use of another Oracle VirtualBox feature called snapshots, one can save a particular state of a virtual machine and revert back to that state, if necessary. This way, one can freely experiment with a computing environment. If something goes wrong, such as problems after installing software or infecting the guest with a virus, you can easily switch back to a previous snapshot and avoid the need of frequent backups and restores.

Any number of snapshots can be created, allowing you to travel back and forward in virtual machine time. You can delete snapshots while a VM is running to reclaim disk space.

Infrastructure consolidation. Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads. A lot of hardware resources as well as electricity is thereby wasted. So, instead of running many such physical computers that are only partially used, one can pack many virtual machines onto a few powerful hosts and balance the loads between them.



Netrunner os

The Netrunner Team is happy to announce the release of Netrunner 21.01 “XOXO” – 64bit ISO.



This version is based upon the current **Debian Stable 10.7 (“buster”)** and comes with better **modern hardware support** thanks to Linux **Kernel 5.9.15** from Debian Buster Backports.

Netrunner 21.01 ships with all the latest security updates provided by Debian and a new beautiful wallpaper showing of the new Codename of this release.

With the activated Debian Buster Backports repository we provide updated firmware for wifi and ethernet chips aswell as improved printer drivers to allow more modern hardware support.

Firefox-ESR and **Thunderbird** were updated to the latest stable LTS (long term supported) versions, which get regular security updates provided by Debian security. Netrunner maintains its georgous look and feel from the previous version based upon Breeze Window decoration and the red colors cursor.

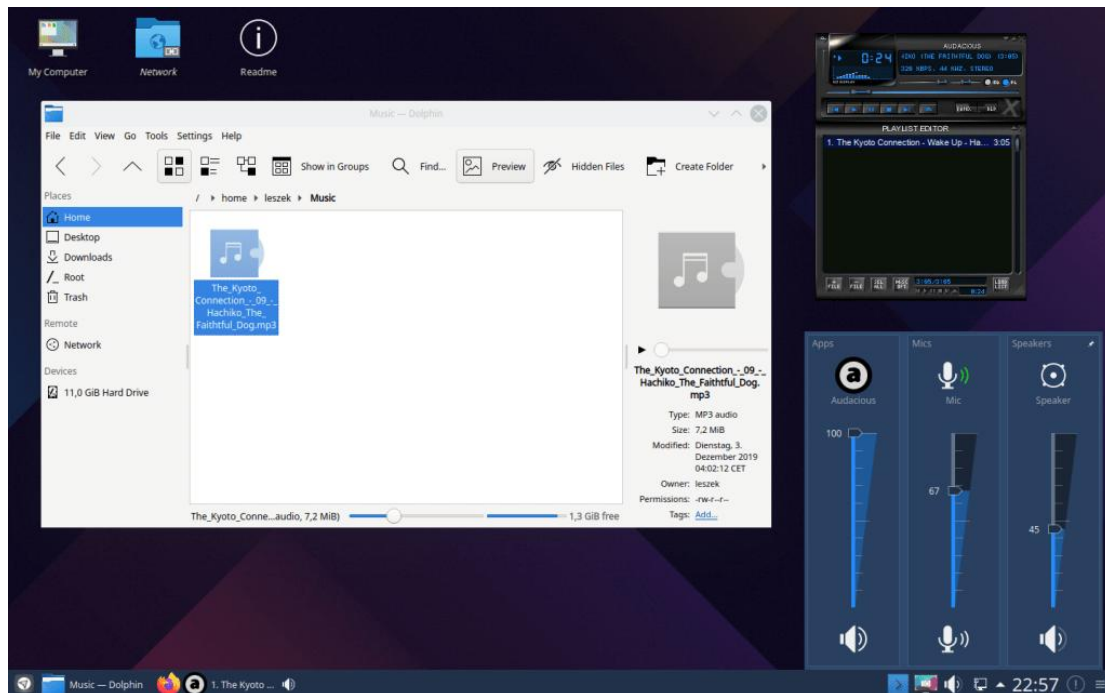
Applications

Netrunner ships with a variety of tools and applications that can be used day to day (Note: If you prefer a minimal install, please take a look at “Netrunner Core” which will be updated shortly to the same 21.01 base)

LibreOffice our Office suite offers everything you might need for writing documents, calculating spreadsheets or creating impressive presentations. Especially presentations and their OpenGL transitions have been updated in **Version 6.1.5**

Editing pictures with **Gimp** (GNU Image Manipulation Program) gives you a powerful tool for tweaking & manipulating images. Draw masterpieces with your

mouse or pen by using **Krita**. Both are extendable via plugins to allow even a wider variety of creative work to shine.



Designing logos, icons and scalable graphics can be done with **Inkscape** which is known to be the favorite vector based image editor of choice for many creators of great Icon packs.



Besides still images videos are more and more popular and so we provide one of the best free and open source video editors available on the market called **Kdenlive**. It provides you with familiar video editing concepts like timeline, audio and video tracks and combines this with a large variety of audio visual effects that help you creating wonderful video clips.

Creativity needs relaxation. This is why we ship with **GMusicbrowser** and **Yarock** to manage your music data and listen to songs. Or you can watch a video with **SMplayer**. With a bunch of games Netrunner allows you to break out of reality for a while and enjoy puzzles or a nice game of chess. If you want to get serious with games the largest Linux gaming platform **Steam** is at your service.

Connecting to the world via **Skype** or **Pidgin** is possible as well as having fun developing small applications with **Kate** or enjoy typing commands into a sliding down terminal like **Yakuake**.

Motivation:

As a software engineering student, I believe that theoretical knowledge must be complemented by practical skills. Installing and configuring an operating system inside a virtual environment allows me to gain real-world, hands-on experience without the risk of damaging actual hardware. This project gives me the opportunity to explore how operating systems function, how filesystems are structured, and how virtualization technologies like Oracle VirtualBox simulate a complete machine. It also prepares me for future system-level tasks, such as system programming, server setup, or working with embedded devices.

Objectives:

The main objectives of this project are to gain practical experience in installing and configuring an operating system (Netrunner OS) within a virtual environment using Oracle VirtualBox. Through this process, I aim to improve my understanding of Linux-based operating systems, learn how to troubleshoot installation issues, and explore different types of filesystems. Additionally, the project helps me enhance my technical documentation skills by carefully recording each step and any challenges faced during the setup.

Requirements

1. Hardware Requirements (your actual or minimum needed)

Hardware Requirements:

Processor: 2 cpu processors

RAM: 4 GB

Hard Disk: 25 GB

Video: 128Mb

Virtualization Support: Enabled in BIOS (Intel VT-x or AMD-V)

2. Software Requirements

Software Requirements:

Oracle VirtualBox (Version 6.x or higher)

Netrunner OS ISO image (latest version from netrunner.com)

Host OS: Windows / Linux / macOS (whichever OS you're running VirtualBox on)

1. Download the Installer

Visit the [official Oracle VirtualBox download page](https://www.oracle.com/uk/technetwork/topics/virt-xp/downloads/index.html) and select the appropriate version for your operating system.

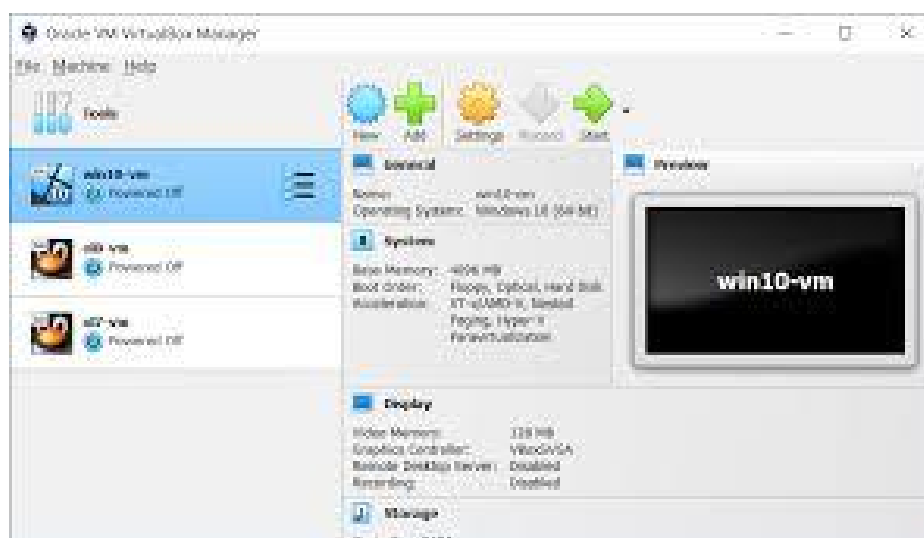
2. Launch the Installer

After downloading, double-click the installer file to begin the installation process.



3. Welcome Screen

Click "Next" on the welcome screen to proceed. brb.nci.nih.gov+1www.slideshare.net+1



4. Choose Installation Location

Select the destination folder for the installation or proceed with the default location.



5. Select Components

Choose the components you wish to install. It's recommended to keep the default selections unless you have specific needs.



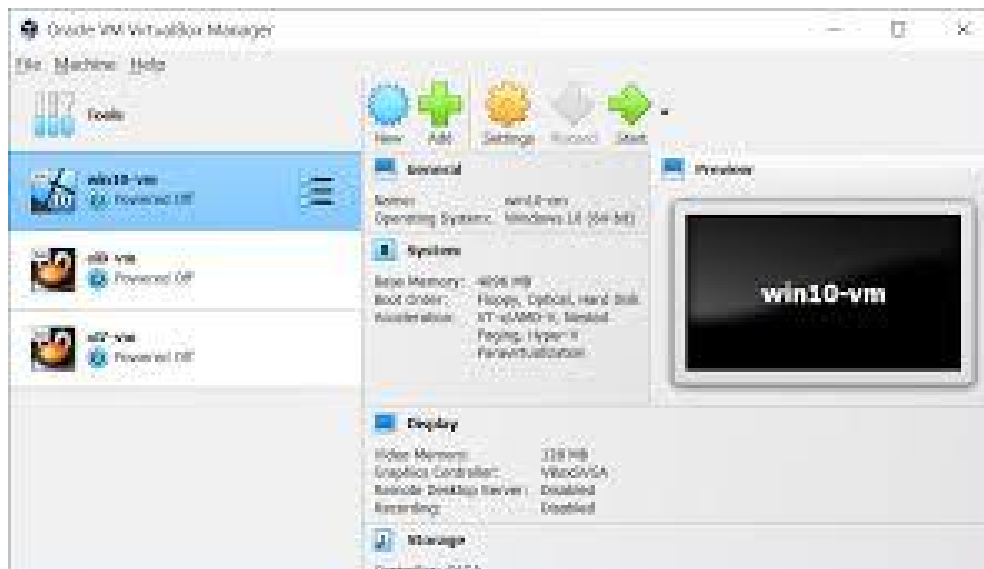
6. Network Interfaces Warning

A warning about network interfaces may appear. Click "Yes" to allow the installation of network features.



7. Begin Installation

Click "Install" to begin the installation process.



8. Installation Progress

Wait for the installation to complete. This may take several minutes.



9. Installation Complete

Once the installation is complete, click "Finish" to exit the setup wizard. [VirtualBox+7dbapostmortem.com+7VirtualBox+7](https://www.dbapostmortem.com/7VirtualBox+7)



Now we setup virtual box goto netrunner set up

Installing Netrunner 20.01 on Oracle VM VirtualBox

Step 1: Download Netrunner 20.01 ISO

Visit the [official Netrunner download page](#) and download the **Netrunner Core 20.01 64-bit ISO**.

Step 2: Create a New Virtual Machine in VirtualBox

Open **Oracle VM VirtualBox**.

Click on the **"New"** button to start the creation of a new virtual machine.

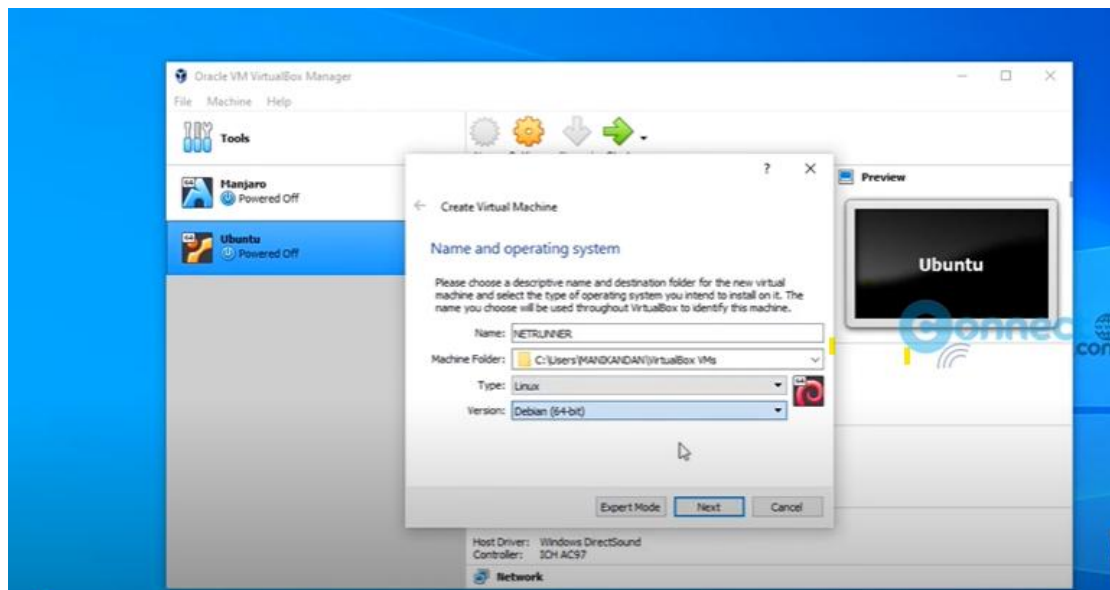
In the **"Name and Operating System"** window:

Name: Enter a descriptive name for your virtual machine (e.g., "Netrunner 20.01").

Type: Select **Linux**.

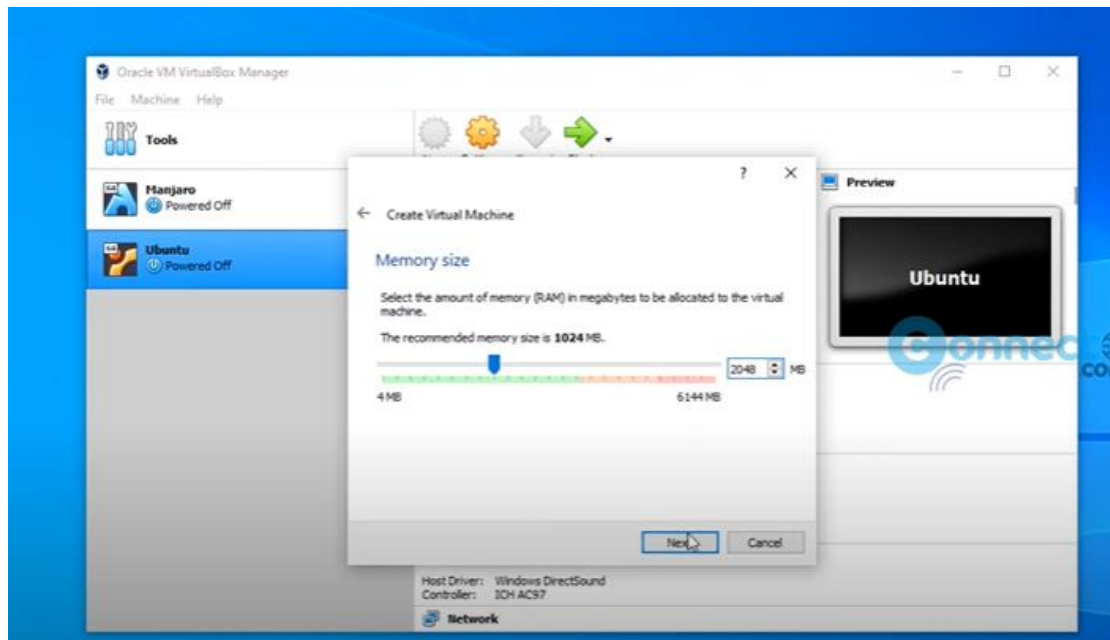
Version: Choose **Debian (64-bit)**.

Click **"Next"**.



Step 3: Allocate Memory (RAM)

Allocate at least **1,200 MB** of RAM to the virtual machine but I set 2048.



Click **"Next"**.

Step 4: Create a Virtual Hard Disk

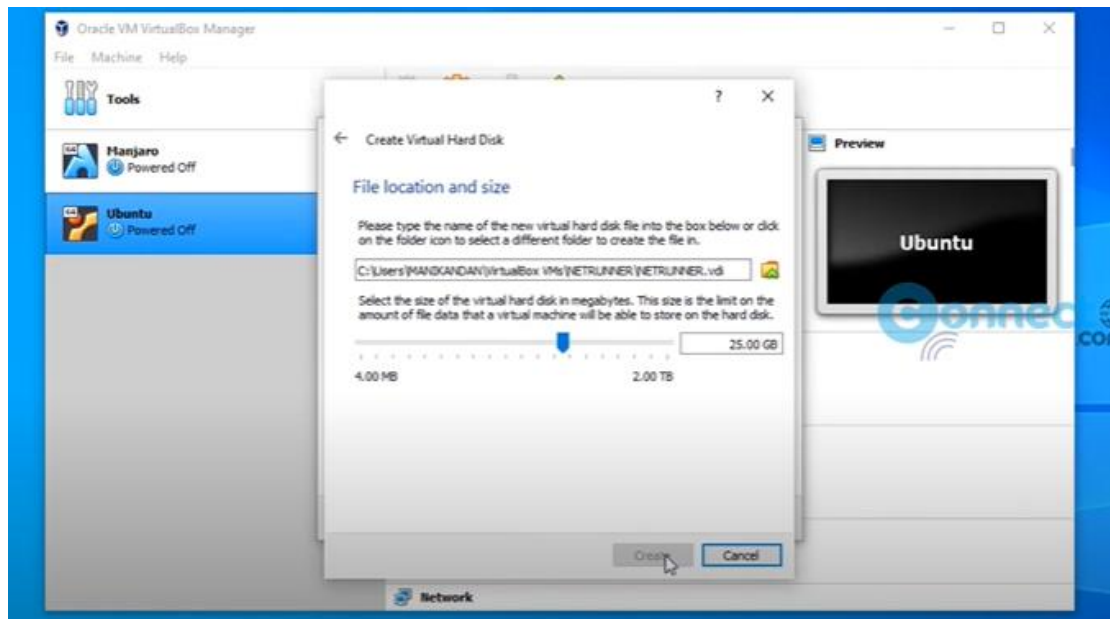
Select **"Create a virtual hard disk now"** and click **"Create"**.

In the **"Hard Disk File Type"** window, choose **VDI (VirtualBox Disk Image)** and click **"Next"**.

In the **"Storage on Physical Hard Disk"** window, select **Dynamically allocated** and click **"Next"**.

In the **"File Location and Size"** window:

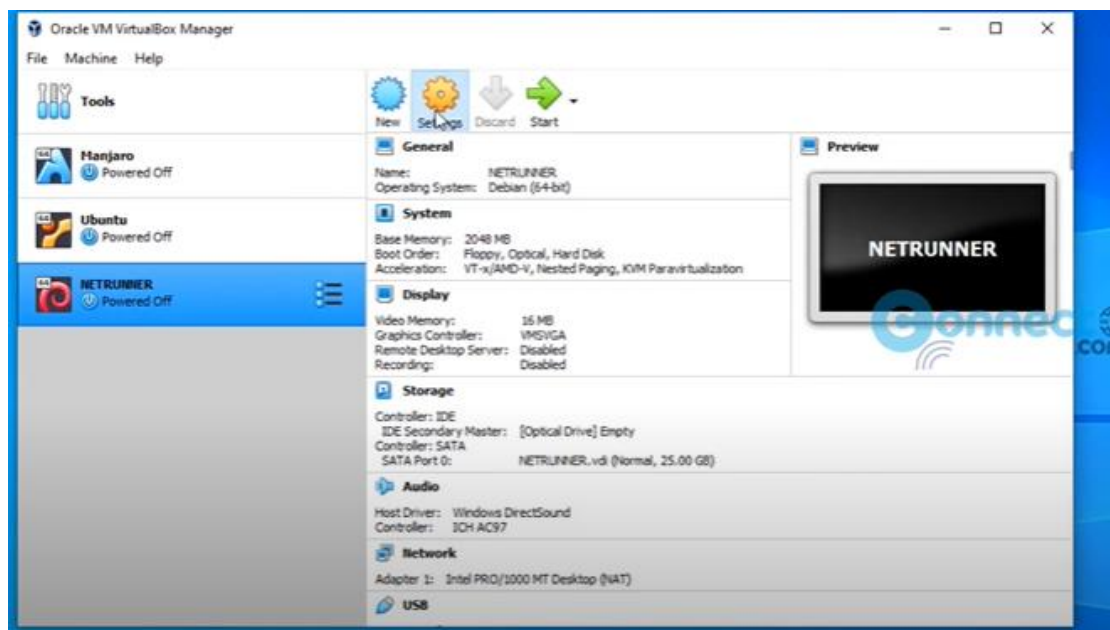
Set the size to at 25Gb in my choose.



Click **"Create"**.

Step 5: Mount the Netrunner ISO

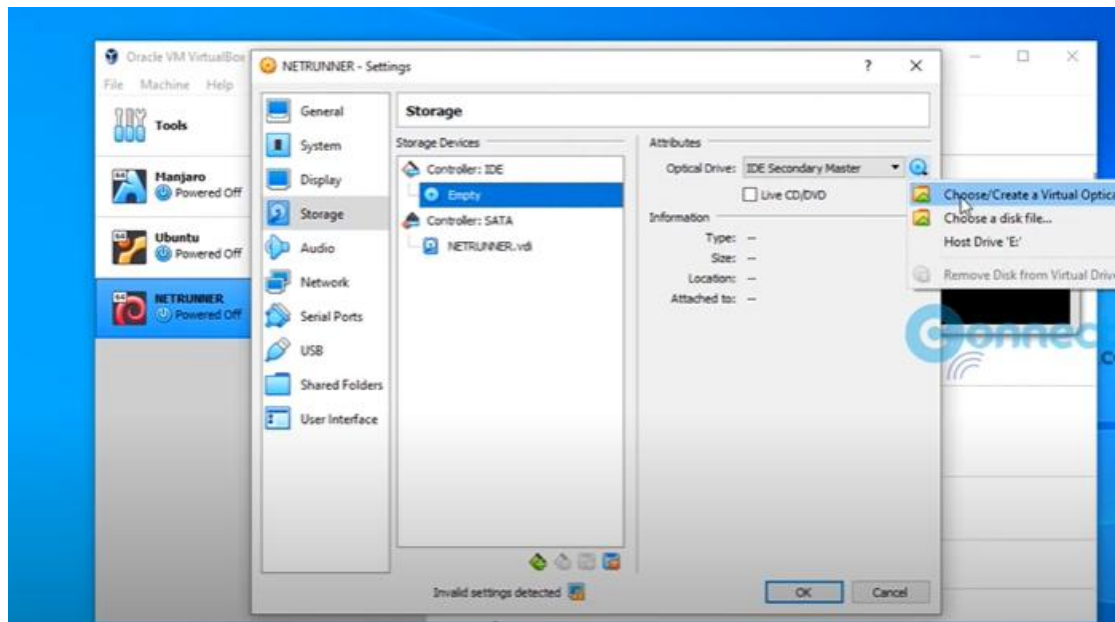
With your new VM selected, click on **"Settings"**.



Navigate to the **"Storage"** tab.

Under **"Controller: IDE"**, click on the empty disk icon.

On the right side, click on the disk icon next to **"Optical Drive"** and select **"Choose a disk file..."**.



Browse to the location of the downloaded Netrunner 20.01 ISO file and select it.

Click **"OK"** to close the settings.

Step 6: Start the Virtual Machine

With the VM selected, click **"Start"**.

The Netrunner live session will boot up.

Once the desktop loads, click on the **"Install Netrunner"** icon on the desktop to begin the installation process.

Step 7: Begin Installation

Select your language and click **"Continue"**.

Choose your location and keyboard layout, then click **"Continue"**.

Configure the network settings as prompted, then click **"Continue"**.

Set up your user account and password, then click **"Continue"**.

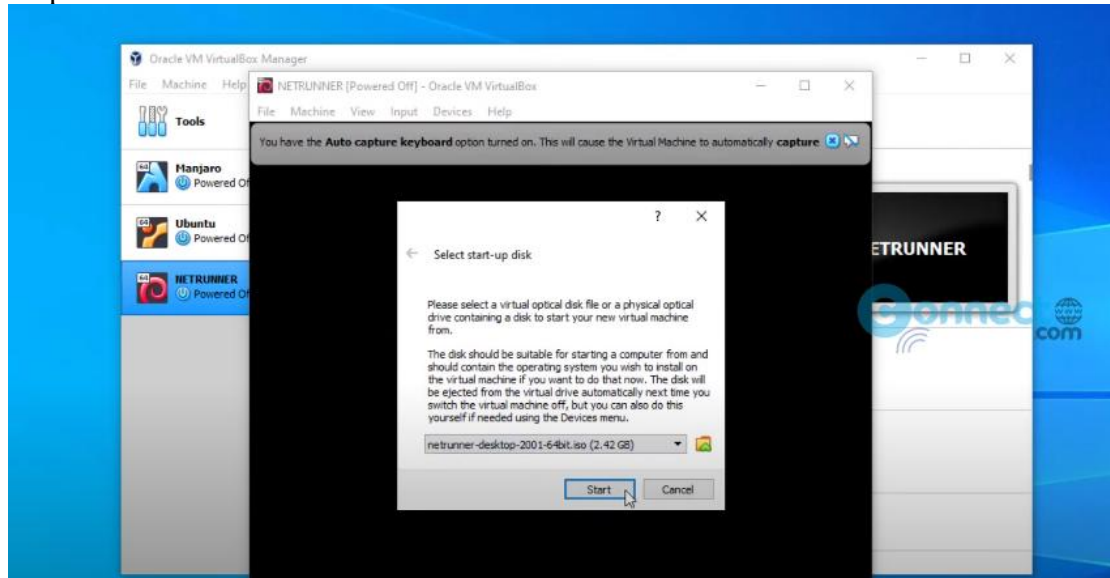
Choose the installation disk (the virtual hard disk you created earlier) and click **"Install Now"**.

Confirm the partition changes and click **"Continue"**.

Step 8: Installation Progress

The installation will proceed. This may take some time. Once completed, you will be prompted to restart the system.

Step 9: Reboot and Remove Installation Media



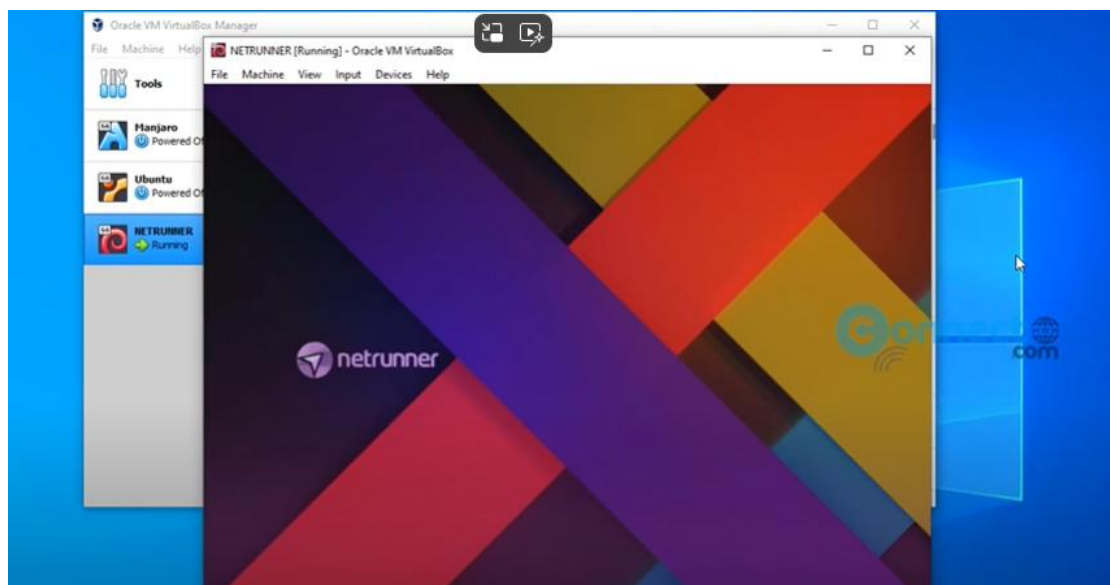
Before rebooting, go back to the **"Storage"** settings of your VM.

Under **"Controller: IDE"**, click on the Netrunner ISO and click the **minus (-)** button to remove it.

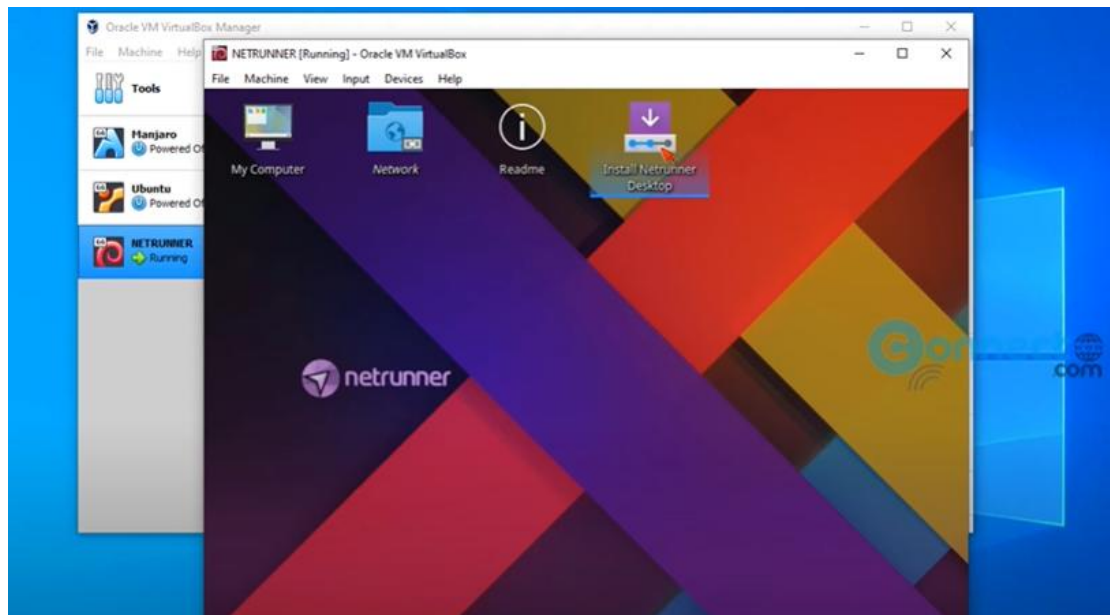
Click **"OK"** to save the settings.

Start the VM again.

my Netrunner 20.01 installation is complete.



After my netrunner is installed to create account I goto install netrunner dektop as you see in image upper corner



1. Netrunner Desktop Installer: Initial Setup

If you're seeing a **desktop installer** (a welcome screen that runs right after logging in), this is part of Netrunner's first-time setup. This will guide you through some essential configurations for the desktop.

Language Settings: The installer will ask you to confirm your **preferred language**. Select the appropriate one (e.g., English, German, etc.) and click **Next**.

Keyboard Layout: Next, you'll be prompted to choose your **keyboard layout**. You can choose the default (likely US) or a layout specific to your region.

Time Zone: The installer will ask you to select your **time zone**. It should detect it automatically, but you can adjust it manually if needed.

2. Connect to the Internet

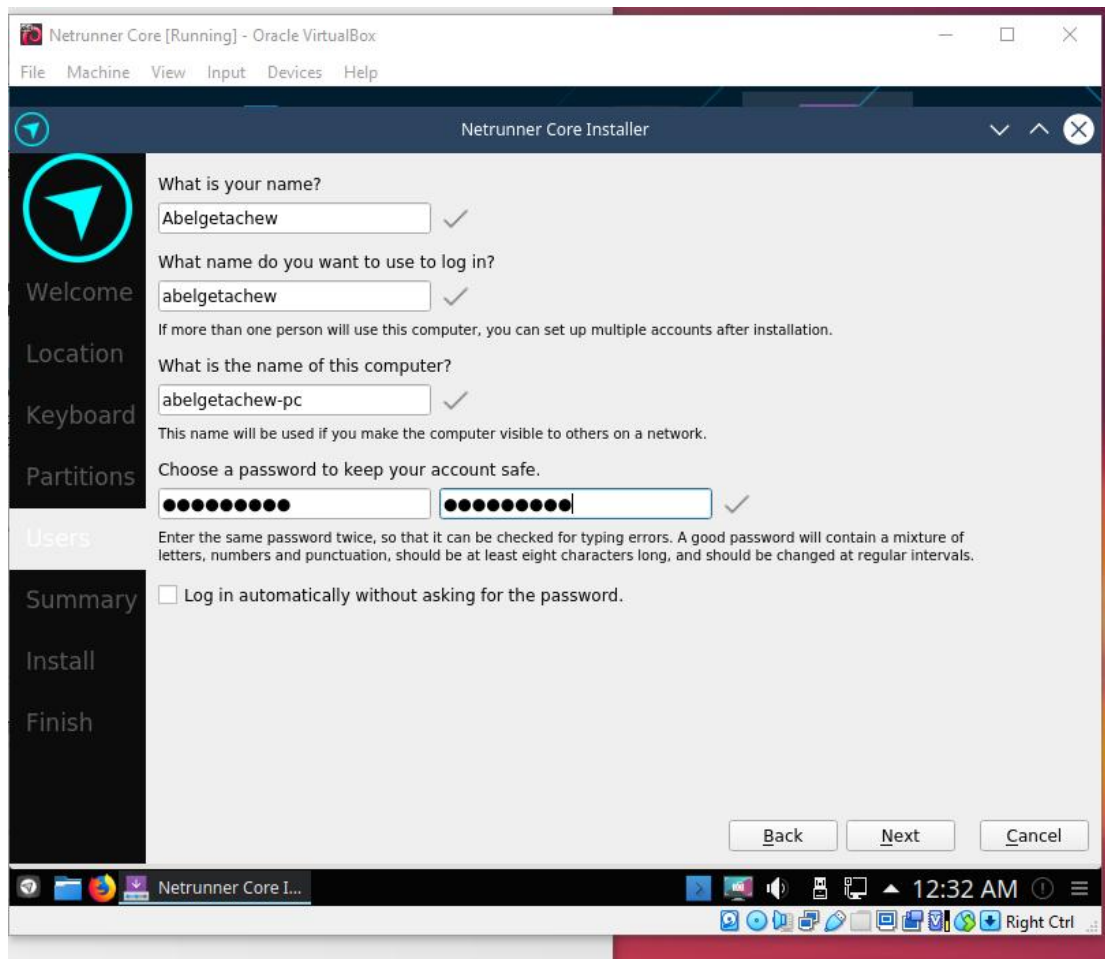
The **Network Setup** window might appear. If it does, ensure that you're connected to the internet (via Wi-Fi or Ethernet).

If you're using a Wi-Fi network, select your network and enter the **Wi-Fi password**.

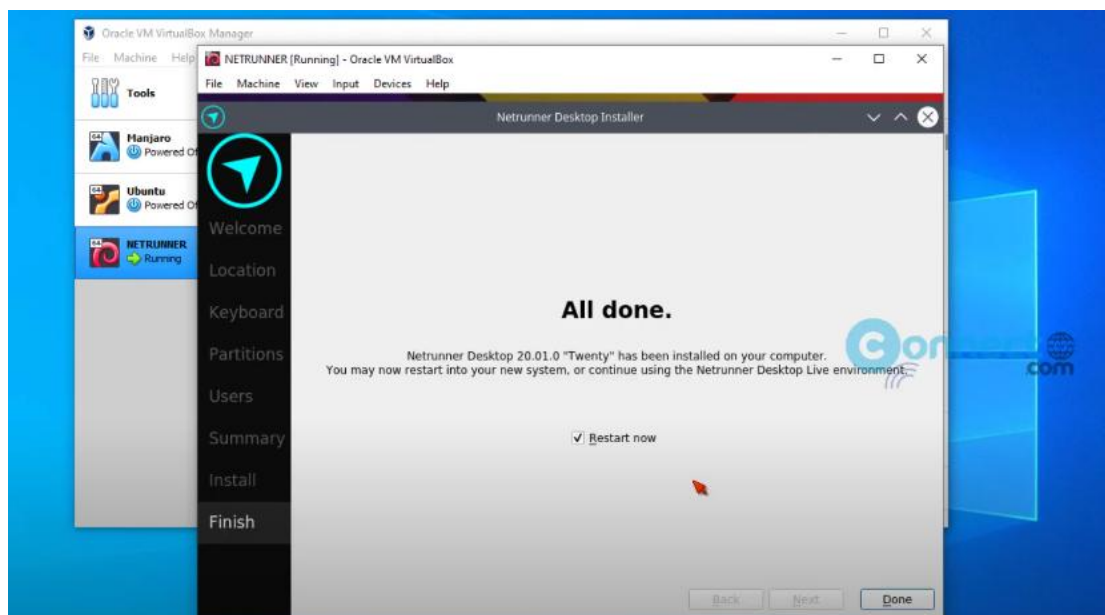
If you're using Ethernet, it should automatically configure, but you can check by clicking the **network icon** in the taskbar to see the connection status.

3. Now after we get the interface of users we fill our data

Below image show what I do in my progress



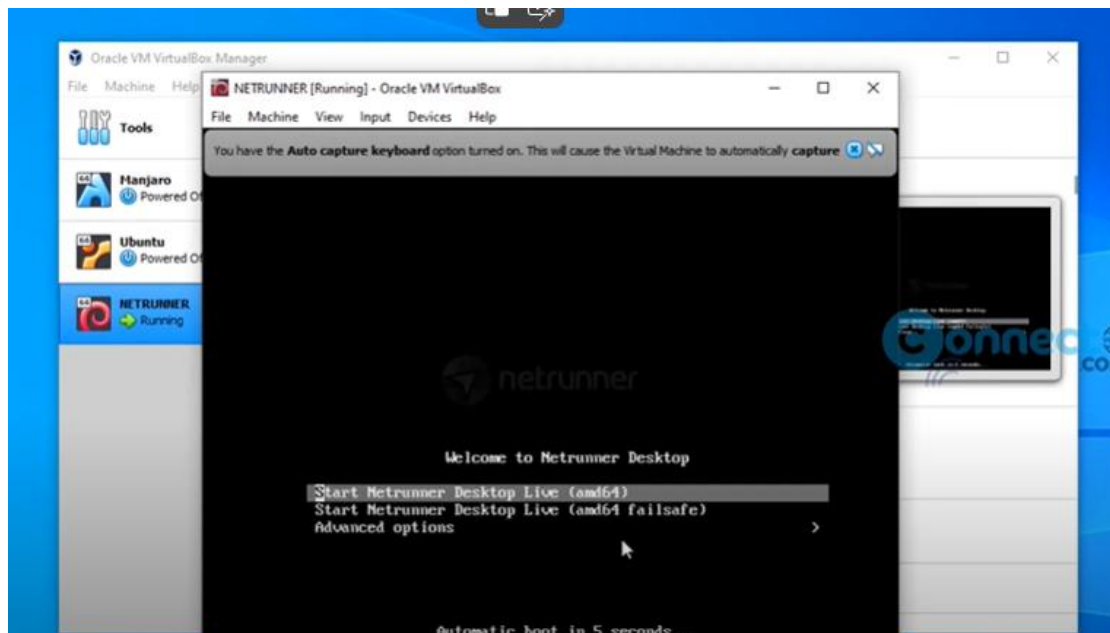
Then we click next and in next choice we click finish and install with fill restart fill checkbox.



now we successfully finished. now open terminal in netrunner it's easy in left right corner we see the the triangle shape menubar and click, then on search bar write "Konsole" we gate.

Problems and solutions I get in installation process:

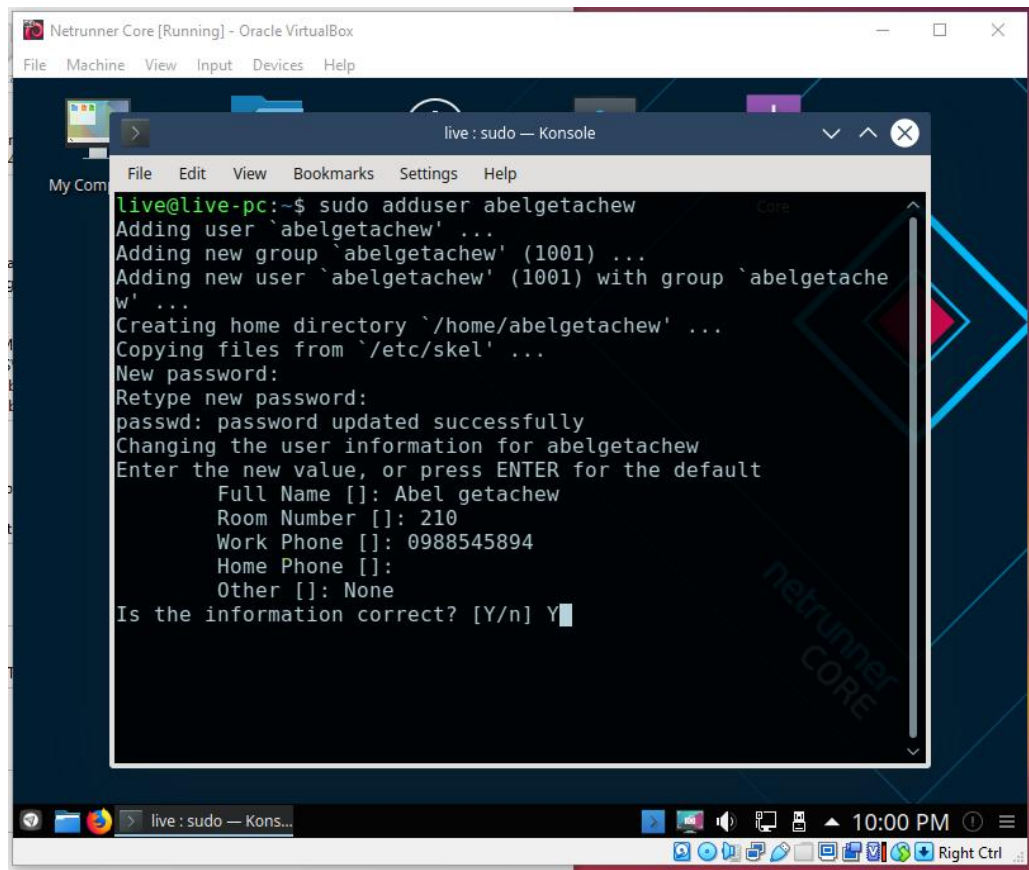
1. First in version difference I try to follow you ube channels for installing but because of that them vm responds different from my vm response in time my computer persforms verry slowly because of I give large Ram first then I fix it in storage drive giving in above requirment as you see. On come the main problem version difference hen follow Distrowatch.com is a site full information about all linux distributions.
2. When I reboot normaly the system aiutomatically responds but I felt in follow steps in reboot again and start with mounting I felt and on in bellow image my mouse dissapers and



But I get resstarting and follow the above commant in imahe "start netrunner desctop live(amd 64)" I continued.

3. when Create account I already open an account but when get in I do't get my name for login restart is must but I do that reset my data account I made confussion netrunner even not exit in already in my computer I do't know but get a way I get in terminal I see with commands I can create account reproccess. Those commands are

Sudo adduser abelgetachew or su -u abelgetachew I choose first and I fill subsequent instructions like you see in below the image finally the account is created when enter Y for command of "is your information is correct."



Filesystem Support:

Netrunner OS, being based on Debian Linux, supports several filesystems including ext4, Btrfs, XFS, and FAT32. During the installation, the default filesystem used was **ext4**. This is because ext4 is known for its balance between performance, reliability, and stability. It supports journaling, large file sizes, and is widely compatible across Linux distributions. Although filesystems like Btrfs offer advanced features such as snapshots and compression, ext4 remains the most commonly used due to its maturity and proven reliability. Below more about ext4

ext4 (fourth extended filesystem) is a [journaling file system](#) for [Linux](#), developed as the successor to [ext3](#).

ext4 was initially a series of [backward-compatible](#) extensions to ext3, many of them originally developed by Cluster File Systems for the [Lustre file system](#) between 2003 and 2006, meant to extend storage limits and add other performance improvements. However, other [Linux kernel](#) developers opposed accepting extensions to ext3 for stability reasons, and proposed to [fork](#) the source code of ext3, rename it as ext4, and perform all the development there, without affecting existing ext3 users. This proposal was accepted, and on 28 June 2006, [Theodore Ts'o](#), the ext3 maintainer, announced the new plan of development for ext4.

A preliminary development version of ext4 was included in version 2.6.19 of the Linux kernel. On 11 October 2008, the patches that mark ext4 as stable code were merged in the Linux 2.6.28 source code repositories, denoting the end of the development phase and recommending ext4 adoption. Kernel 2.6.28, containing the

ext4 filesystem, was finally released on 25 December 2008. On 15 January 2010, [Google](#) announced that it would upgrade its storage infrastructure from [ext2](#) to ext4. On 14 December 2010, Google also announced it would use ext4, instead of [YAFFS](#), on [Android 2.3](#).

Now we see **benefits and drawbacks** i observed while working with Netrunner OS in the virtual environment.

Advantages :

User-friendly interface: Netrunner is KDE-based and beginner-friendly without thinking in my way I love it.

Pre-installed applications: Comes with many tools for daily use.

Stable and secure: Based on Debian (LTS = Long-Term Support).

Ideal for testing Linux: No need to change your host OS.

Easy recovery/reset with snapshots in VirtualBox.

No risk to hardware: Safe environment for experimentation.

Disadvantages :

Performance limitations: Can be slow if not enough RAM/CPU is allocated.

Not suitable for older PCs: Netrunner's KDE desktop uses more resources.

Limited hardware access: Some hardware features (like GPU acceleration) may not work in VirtualBox.

Guest Additions issues: Sometimes screen resizing or clipboard sharing doesn't work unless configured manually.

Conclusion:

Working on this project has been a very valuable and eye-opening experience for me. As someone studying software engineering, I've always known that operating systems are a crucial part of computer systems — but getting the chance to actually install and interact with one in a virtual environment helped me understand how everything fits together in practice.

Setting up Netrunner OS in Oracle VirtualBox allowed me to experience what it's like to prepare a machine from scratch — from configuring hardware settings virtually to selecting the right filesystem and dealing with common issues like display resolution or performance limitations. I now have a better understanding of how Linux-based

systems work, and why tools like VirtualBox are important for developers, testers, and learners like me.

One of the biggest takeaways for me was learning how to approach technical problems during installation. It taught me to stay calm, search for solutions, and try different things until something works — which is a skill that goes beyond just operating systems.

Overall, this project didn't just teach me about the steps involved in installing an OS — it made me more confident in my ability to explore new technologies, use virtualization tools, and apply what I've learned in a real-world context. I now feel more prepared for future work in system-level programming and development.

Future Outlook / Recommendations:

After completing this project, I feel encouraged to continue exploring more advanced aspects of operating systems. One of my goals moving forward is to dive deeper into Linux system administration and shell scripting, especially now that I've gained confidence working in a virtual environment. I'm also interested in testing other Linux distributions to compare their performance, desktop environments, and system tools.

In terms of improvements, I would recommend allocating more system resources — such as RAM and CPU cores — to the virtual machine if possible, especially when using heavier desktop environments like KDE. Installing VirtualBox Guest Additions early in the process also makes the experience much smoother, particularly for screen resizing and shared clipboard functions.

For other students working on similar projects, I'd recommend documenting every step, including any problems faced. This not only helps with the final report but also makes it easier to repeat or explain the process later. Overall, I believe this project lays a strong foundation for anyone looking to get into system-level development or working more closely with open-source technologies.

Simple System Call(wait()- parent keep untill child exexcute.)

Now I do wait() system call in my terminal so follow me I choose write scripts in c.

I goto konsole and write: `nano systemcall.c`

Then we write the following c code.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
#include <sys/types.h>
```

```
#include <sys/wait.h> // for wait()

int main() {

    pid_t pid = fork();

    if (pid < 0) {

        perror("Fork failed");

        exit(1);

    }

    if (pid == 0) {

        // Child process

        printf("Child process is running with pid: %d\n", getpid());

        sleep(2);

        printf("Child process exiting.\n");

    } else {

        // Parent process

        wait(NULL); // Parent waits for child to finish

        printf("Parent process (pid: %d) detected that child finished.\n", getpid());

    }

    return 0;

}
```

As I did in my terminal from following image.

```
GNU nano 3.2 systemcall.c

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h> //for wait()

int main(){
    pid_t pid=fork();

    if(pid>0){
        perror("fork failed")
        exit(1);
    }

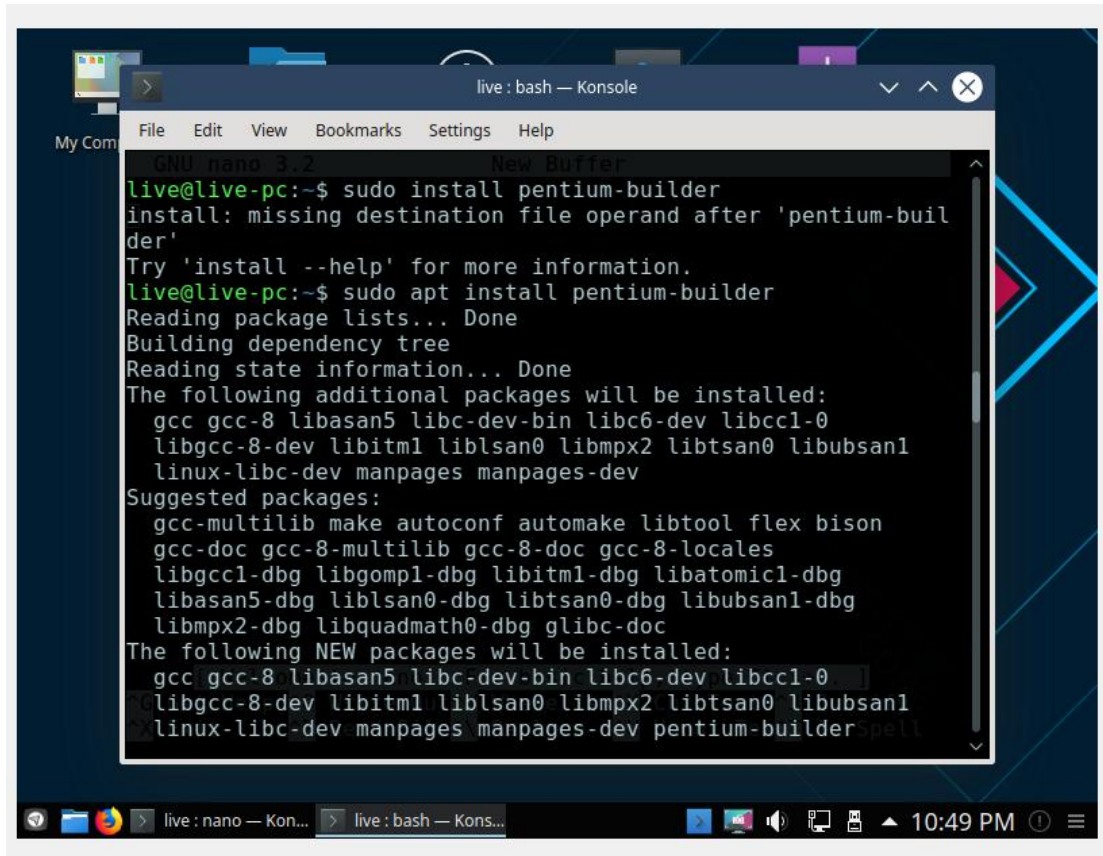
    if(pid==0){
        printf("children process is runningwith pid:$
        sleep(2);
    }

Welcome to nano. For basic help, type Ctrl+G.
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Tex ^T To Spell
```

Now to save we do: ctrl+x and then “enter”

Now we compile but from not we install gcc we do:

sudo apt install gcc/petium-buider



```
GNU nano 3.2 New Buffer
live@live-pc:~$ sudo install pentium-builder
install: missing destination file operand after 'pentium-builder'
Try 'install --help' for more information.
live@live-pc:~$ sudo apt install pentium-builder
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gcc gcc-8 libasan5 libc-dev-bin libc6-dev libcc1-0
  libgcc-8-dev libitm1 liblsan0 libmpx2 libtsan0 libubsan1
  linux-libc-dev manpages manpages-dev
Suggested packages:
  gcc-multilib make autoconf automake libtool flex bison
  gcc-doc gcc-8-multilib gcc-8-doc gcc-8-locales
  libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg
  libasan5-dbg liblsan0-dbg libtsan0-dbg libubsan1-dbg
  libmpx2-dbg libquadmath0-dbg glibc-doc
The following NEW packages will be installed:
  gcc gcc-8 libasan5 libc-dev-bin libc6-dev libcc1-0
  libgcc-8-dev libitm1 liblsan0 libmpx2 libtsan0 libubsan1
  linux-libc-dev manpages manpages-dev pentium-builder
```

And now we compile: `gcc syscall.c -o syscall`

And for run: `./syscall`

Now we get out put Child process is running with pid: 12345

Child process exiting.

Parent process (pid: 12340) detected that child finished.

Thank you

You push me forward!!!!

