Gentoo Linux Installation Report

1. Introduction

I'm writing about how I installed Gentoo Linux step by step using the LiveGUI ISO. I used OpenRC, the ext4 filesystem, and genkernel to compile the kernel. Doing this manually helped me understand Linux deeply, including partitioning, package management, kernel setup, and bootloader configuration.

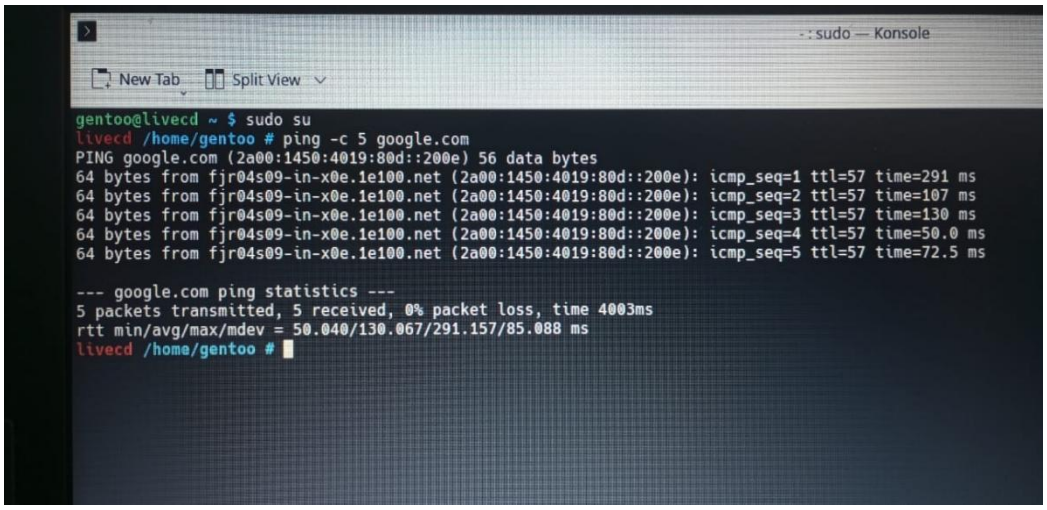
2. System Specifications

- **Laptop Model**: *Lenovo Ideapad 130-15IKB*
- **Processor**: *(Intel Core i3-7020U (2 cores, 4 threads) @ 2.30 GHz)*
- **RAM**: *12 GB*
- **Storage**: *232 GB*

3. Installation Steps

Step 1: Boot the LiveGUI Environment

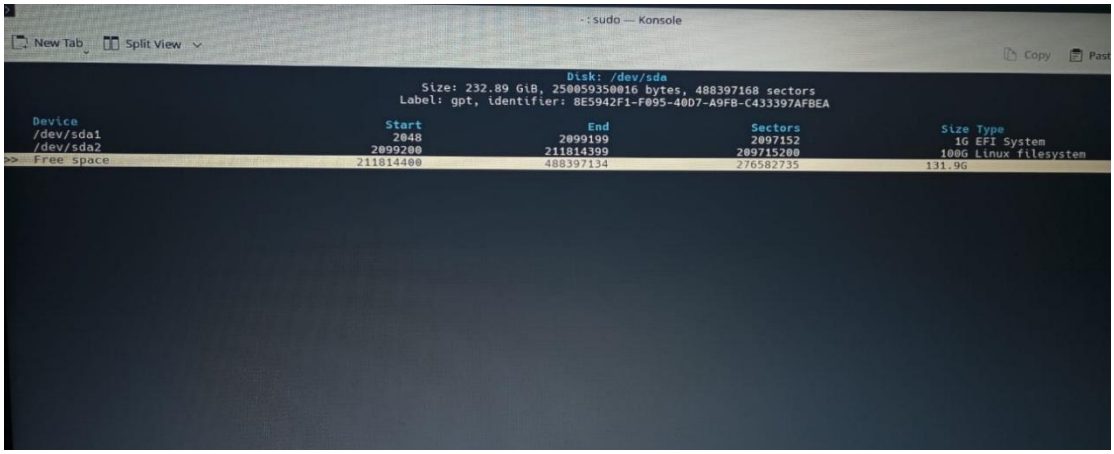
- Started with making the USB bootable through rufus and then booted into Gentoo LiveGUI environment through it.
- Verified the internet connection using "**ping google.com**".



Step 2: Disk Partitioning

Since I was installing Gentoo on bare metal, I did not create a swap partition. Instead, I configured only:

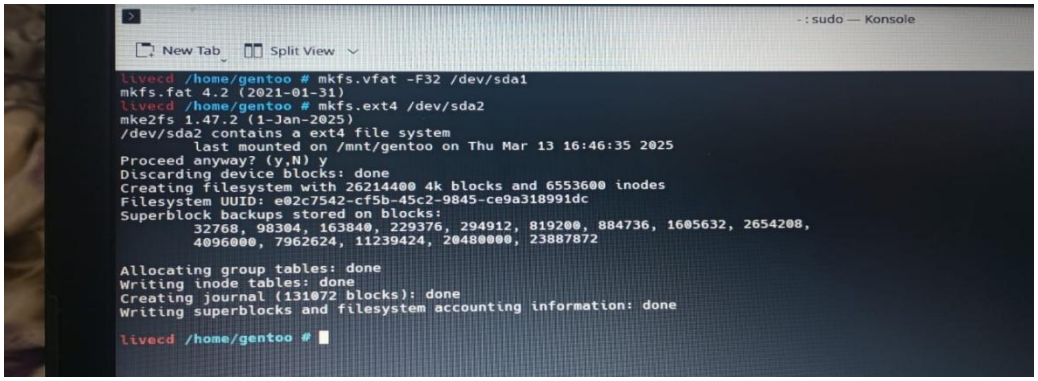
- EFI Partition (1GB, EFI System) → FAT32
- Root Partition (100G, Linux Filesystem) →ext4

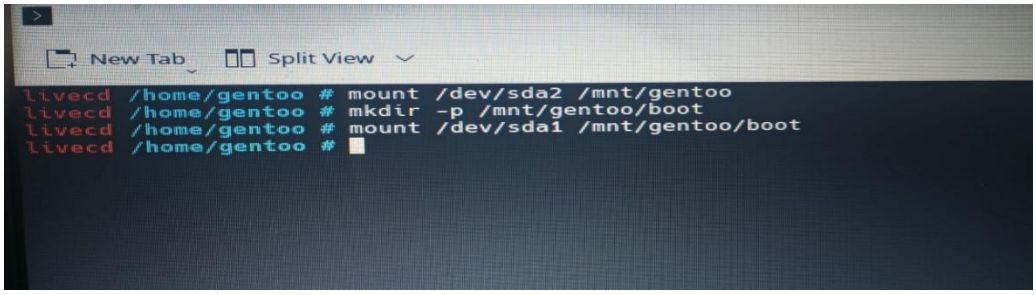


Step 3: Formatting and Mounting Partitions

Formatted and mounted partitions through commands:

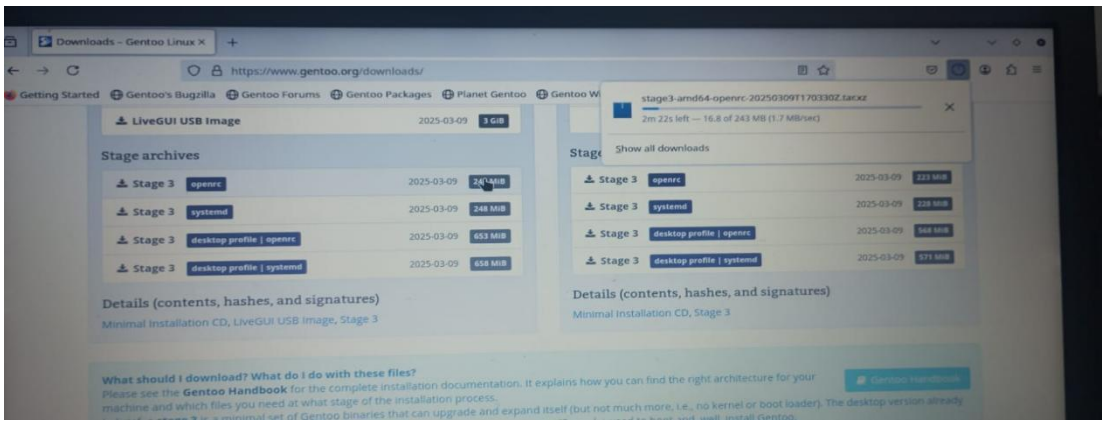
- **mkfs.vfat -F32 /dev/sda1 (EFI)**
- **mkfs.ext4 /dev/sda2 (Root)**
- **Mounted /dev/sda2 to /mnt/gentoo**
- **Created and mounted /boot under /mnt/gentoo**

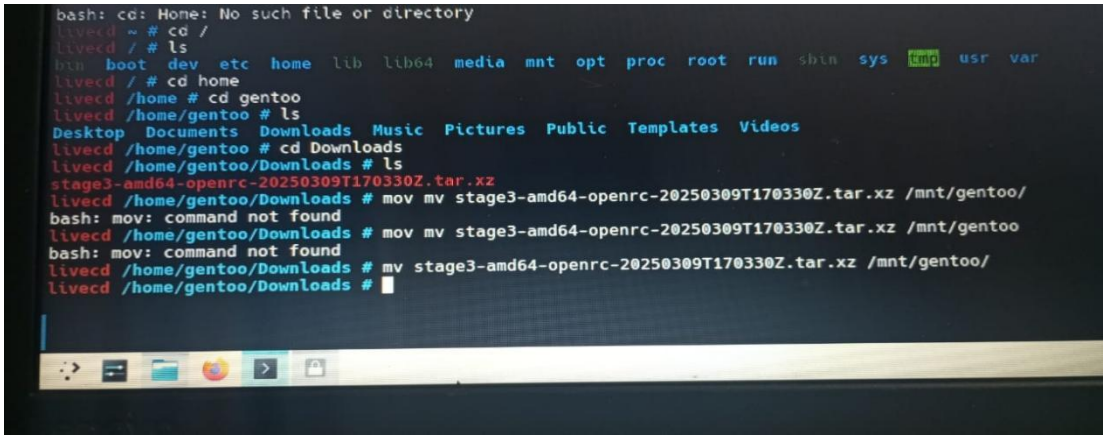




Step 4: Downloading & Extracting Stage 3

I manually downloaded the OpenRC Stage 3 tarball from the official Gentoo website. And moved it to the required directory, and there extract it.





Step 5: Entering the Chroot Environment

Copied the system's DNS settings to ensure internet access inside the chroot.

Mounted essential virtual filesystems:

- **/proc** for process information
- **/sys** for system-related data
- **/dev** for device management

Changed root into the new Gentoo environment using the chroot command, for me to configure the system as if it were fully installed.

```
livecd /mnt/gentoo # cp --dereference /etc/resolv.conf /mnt/gentoo/etc/
livecd /mnt/gentoo # mount -t proc /proc /mnt/gentoo/proc
livecd /mnt/gentoo # mount --rbind /sys /mnt/gentoo/sys
livecd /mnt/gentoo # mount --make-rslave /mnt/gentoo/sys
livecd /mnt/gentoo # mount --rbind /dev /mnt/gentoo/dev
livecd /mnt/gentoo # mount --make-rslave /mnt/gentoo/dev
livecd /mnt/gentoo #
```

```
livecd /mnt/gentoo # chroot /mnt/gentoo /bin/bash
livecd / # source /etc/profile
livecd / # export PS1="(chroot) $PS1"
(chroot) livecd / #
```

Step 6: Syncing Portage

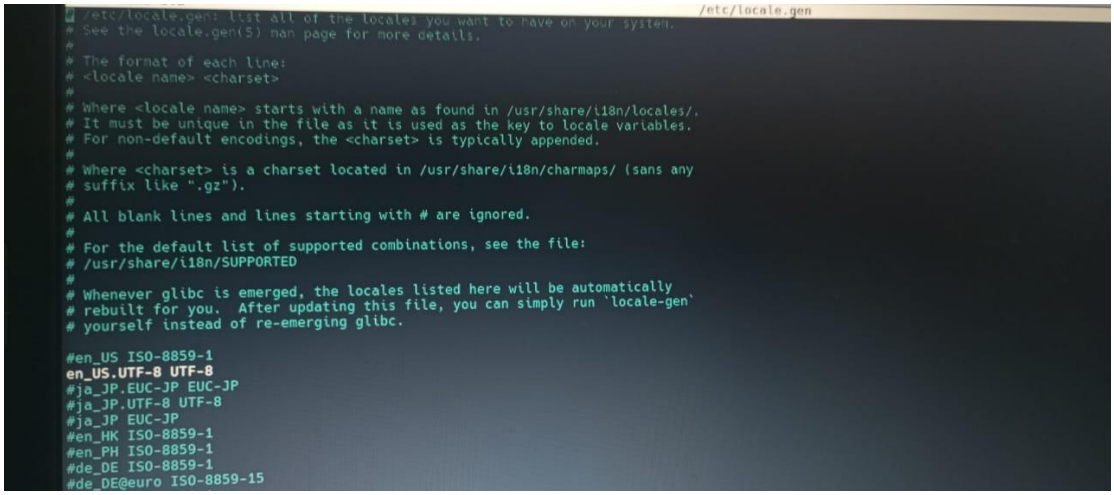
- Ran **emerge-webrsync** to sync the Portage tree.
- Updated the system using **emerge --sync && emerge -uDN @world**.

```
(chroot) livecd / # emerge-webrsync
 * PGP verification method: gemato
 * Fetching most recent snapshot ...
 * Latest snapshot date: 20250312
 *
 * Approximate snapshot timestamp: 1741826700
 *        Current local timestamp: 1741894800
 *
 * The current local timestamp is newer than the timestamp
 * of the latest snapshot. In order to force sync, use the
 * --revert option or remove the timestamp file located at
 * '/var/db/repos/gentoo/metadata/timestamp.x'.
(chroot) livecd / # emerge --sync && emerge -uDN @world
>>> Syncing repository 'gentoo' into '/var/db/repos/gentoo'...
 * Using keys from /usr/share/openpgp-keys/gentoo-release.asc
 * Refreshing keys via WKD ...
>>> Starting rsync with rsync://[2a00:1828:a00d:ffff::6]/gentoo-portage...
>>> Checking server timestamp ...
Welcome to turnstone.gentoo.org / rsync.gentoo.org

Server Address : 89.238.71.6, 2a00:1828:a00d:ffff::6
Contact Name   : mirror-admin@gentoo.org
Hardware       : 16 x Intel(R) Xeon(R) CPU          E5530 , 24152MiB RAM
Sponsor        : Manitu GmbH, St. Wendel, Germany
```

Step 7: Configuring Timezone & Locale

- Set system timezone of karachi thorugh
  echo "UTC" > /etc/timezone
  emerge --config sys-libs/timezone-data.

- Configured locale by uncommenting **en_US.UTF-8 UTF-8** in **/etc/locale.gen**
- And then by running
  locale-gen
  eselect locale set en_US.utf8



Step 8: Installing and Compiling Kernel (Genkernel)

I installed the kernel sources and compiled the kernel using Genkernel with the following steps:

1. I Installed a specific kernel version:
   emerge -v =sys-kernel/gentoo-sources-6.12.16

2. Accepted the necessary firmware license:
   echo "sys-kernel/linux-firmware linux-fw-redistributable" | sudo tee -a /etc/portage/package.license

3. Installed Genkernel:
   emerge sys-kernel/genkernel

4. Created a symbolic link for the kernel sources:
   ln -sf /usr/src/linux-6.12.16-gentoo /usr/src/linux

5. Compiled the kernel using Genkernel:
   genkernel all

```
-:sudo — Konsole
New Tab   Split View
(chroot) livecd / # emerge -v =sys-kernel/gentoo-sources-6.12.16

 * IMPORTANT: 15 news items need reading for repository 'gentoo'.
 * Use eselect news read to view new items.


These are the packages that would be merged, in order:

Calculating dependencies... done!
Dependency resolution took 0.99 s (backtrack: 0/20).

[ebuild  N     ] dev-libs/elfutils-0.191-r2::gentoo  USE="bzip2 nls utils -debuginfod -lzma -stati
,092 KiB
[ebuild  N     ] virtual/libelf-3-r1:0/1::gentoo  ABI_X86="(64) -32 (-x32)" 0 KiB
[ebuild  N     ] app-arch/cpio-2.15::gentoo  USE="nls" 1,613 KiB
[ebuild  N     ] app-alternatives/cpio-0::gentoo  USE="gnu -libarchive (-split-usr)" 0 KiB
[ebuild  N     ] sys-kernel/gentoo-sources-6.12.16:6.12.16::gentoo  USE="-build -experimental -sym

Total: 5 packages (5 new), Size of downloads: 156,324 KiB
```

j…

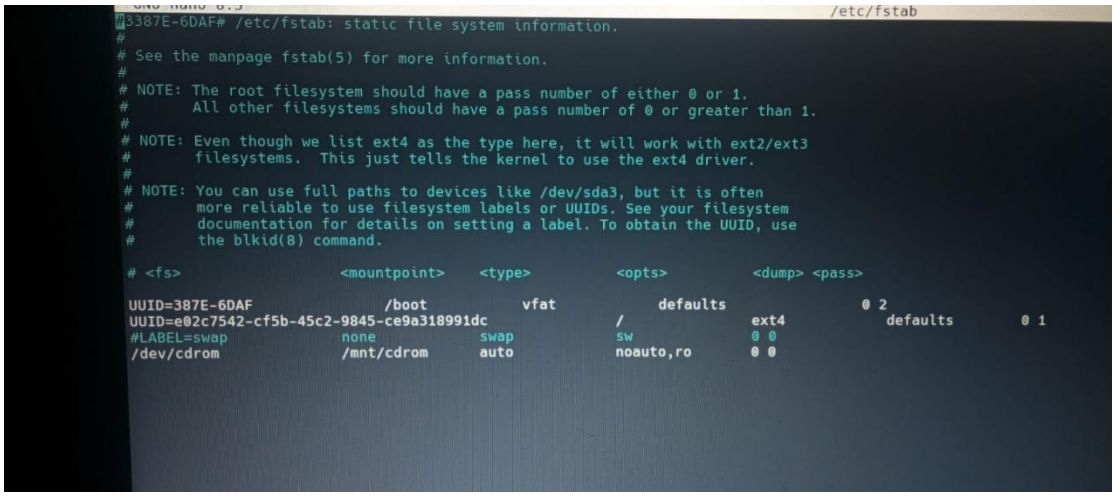
```
- sys-kernel/linux-firmware-20250211::gentoo (masked by: || ( ) linux-fw-redistributable license(s)), ~amd64 keyword(s)
- sys-kernel/linux-firmware-20250211::gentoo (masked by: || ( ) linux-fw-redistributable license(s))
- sys-kernel/linux-firmware-20250109-r1::gentoo (masked by: || ( ) linux-fw-redistributable license(s))
- sys-kernel/linux-firmware-20241210-r1::gentoo (masked by: || ( ) linux-fw-redistributable license(s))

(dependency required by "sys-kernel/genkernel-4.3.16-r2::gentoo[firmware]" [ebuild])
(dependency required by "sys-kernel/genkernel" [argument])
For more information, see the MASKED PACKAGES section in the emerge
man page or refer to the Gentoo Handbook.

(chroot) livecd / # ^C
(chroot) livecd / # echo "sys-kernel/linux-firmware linux-fw-redistributable" | sudo tee -a /etc/portage/package.license
sys-kernel/linux-firmware linux-fw-redistributable
(chroot) livecd / #
```

```
 * IMPORTANT: 19 news items need reading for repository 'gentoo'.
 * Use eselect news read to view new items.

(chroot) livecd / # ln -sf /usr/src/linux-6.12.16-gentoo /usr/src/linux
(chroot) livecd / # genkernel all
* Gentoo Linux Genkernel; Version 4.3.16
* Using genkernel configuration from '/etc/genkernel.conf' ...
* Running with options: all

* Working with Linux kernel 6.12.16-gentoo for x86_64
* Using kernel config file '/usr/share/genkernel/arch/x86_64/generated-config' ...
*
* Note: The version above is subject to change (depends on config and status of kernel sources).

* kernel: >> Initializing ...
*         >> Running 'make mrproper' ...
*         >> Running 'make oldconfig' ...
*         >> Re-running 'make oldconfig' due to changed kernel options ...
*         >> Kernel version has changed (probably due to config change) since genkernel start:
*            We are now building Linux kernel 6.12.16-gentoo-x86_64 for x86_64 ...
*         >> Compiling 6.12.16-gentoo-x86_64 bzImage ...
*         >> Compiling 6.12.16-gentoo-x86_64 modules ...
```

```
(chroot) livecd / # ln -sf /usr/src/linux-6.12.16-gentoo /usr/src/linux
(chroot) livecd / # genkernel all
* Gentoo Linux Genkernel; Version 4.3.16
* Using genkernel configuration from '/etc/genkernel.conf' ...
* Running with options: all

* Working with Linux kernel 6.12.16-gentoo for x86_64
* Using kernel config file '/usr/share/genkernel/arch/x86_64/generated-config' ...
*
* Note: The version above is subject to change (depends on config and status of kernel sources).

* kernel: >> Initializing ...
*         >> Running 'make mrproper' ...
*         >> Running 'make oldconfig' ...
*         >> Re-running 'make oldconfig' due to changed kernel options ...
*         >> Kernel version has changed (probably due to config change) since genkernel start:
*            We are now building Linux kernel 6.12.16-gentoo-x86_64 for x86_64 ...
*         >> Compiling 6.12.16-gentoo-x86_64 bzImage ...
*         >> Compiling 6.12.16-gentoo-x86_64 modules ...
*         >> Installing 6.12.16-gentoo-x86_64 modules (and stripping) ...
*         >> Generating module dependency data ...
*         >> Compiling out-of-tree module(s) ...
*         >> Saving config of successful build to '/etc/kernels/kernel-config-6.12.16-gentoo-x86_

* initramfs: >> Initializing ...
*         >> Appending devices cpio data ...
*         >> Appending base_layout cpio data ...
*         >> Appending util-linux cpio data ...
*         >> Appending eudev cpio data ...
*         >> Appending auxiliary cpio data ...
*         >> Appending busybox cpio data ...
```

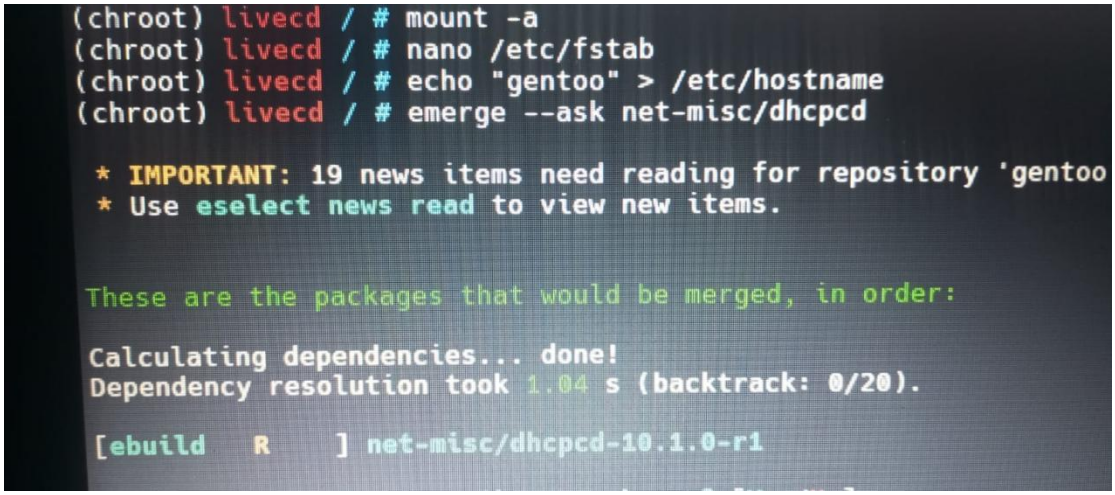
Step 9: Configuring fstab

Edited /etc/fstab to include only EFI and root partitions through **nano /etc/fstab.**



Step 10: Setting Up Network Configuration

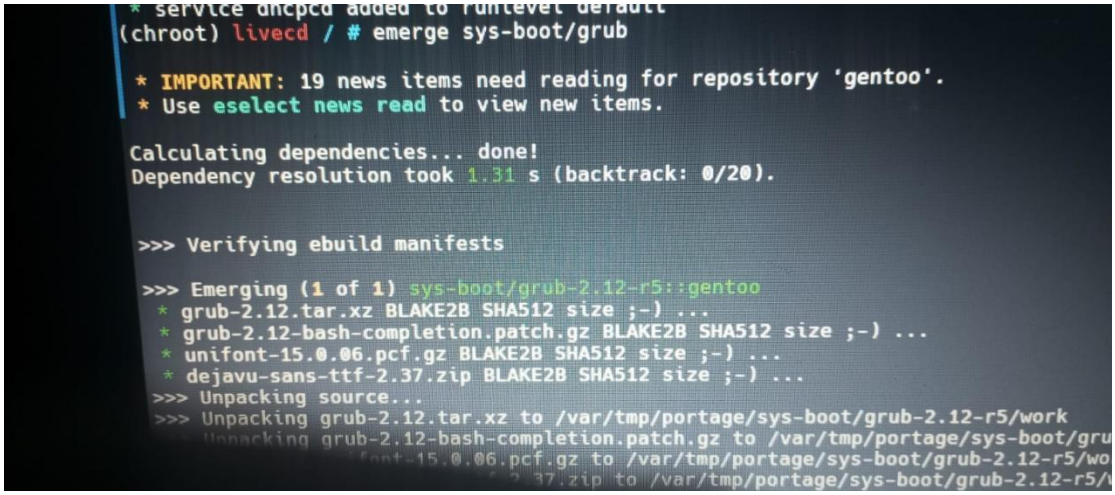
- Set hostname **echo "gentoo" > /etc/hostname**.
- Installed and enabled dhcpcd for networking.

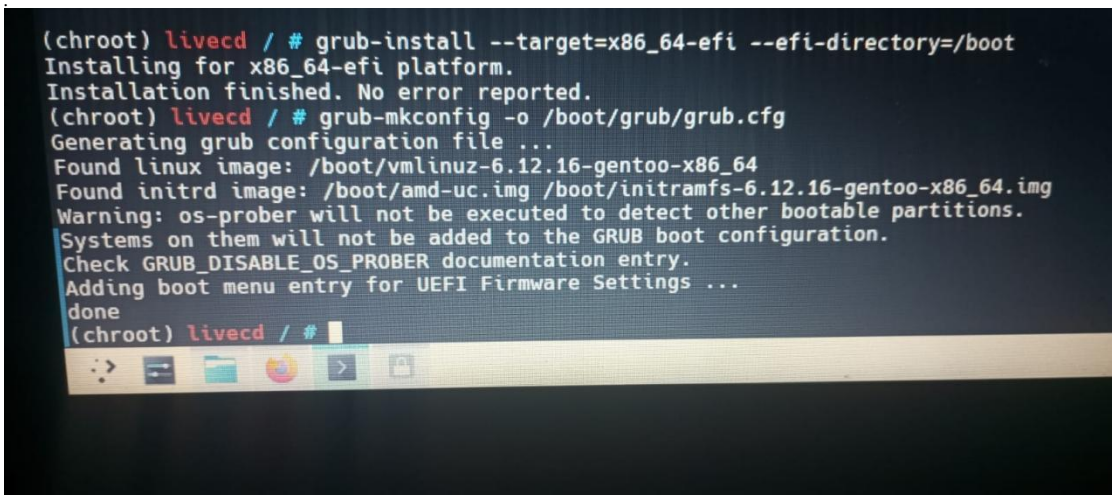


Step 11: Installing and Configuring GRUB

- Installed **sys-boot/grub.**

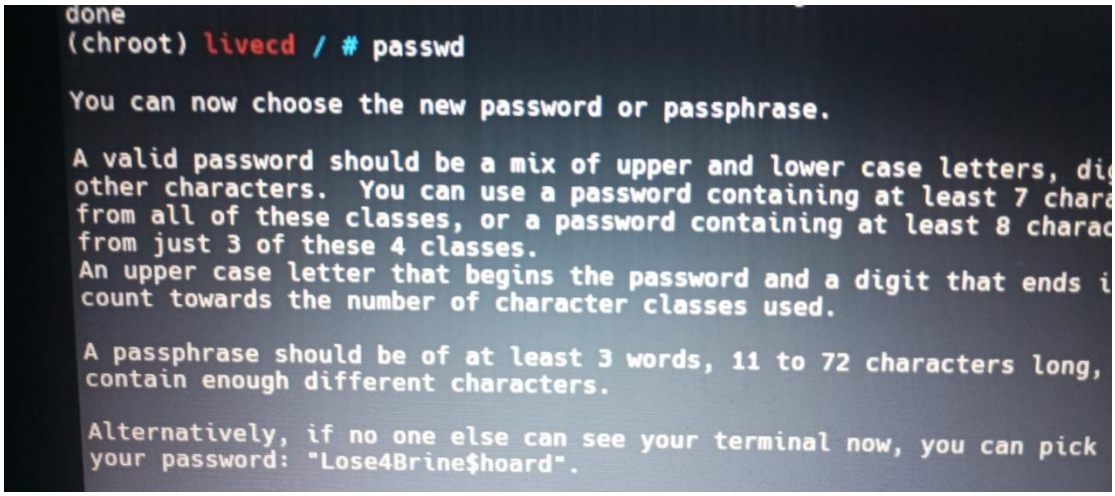
- Installed GRUB to the disk:
  grub-install --target=x86_64-efi --efi-directory=/boot
  grub-mkconfig -o /boot/grub/grub.cfg



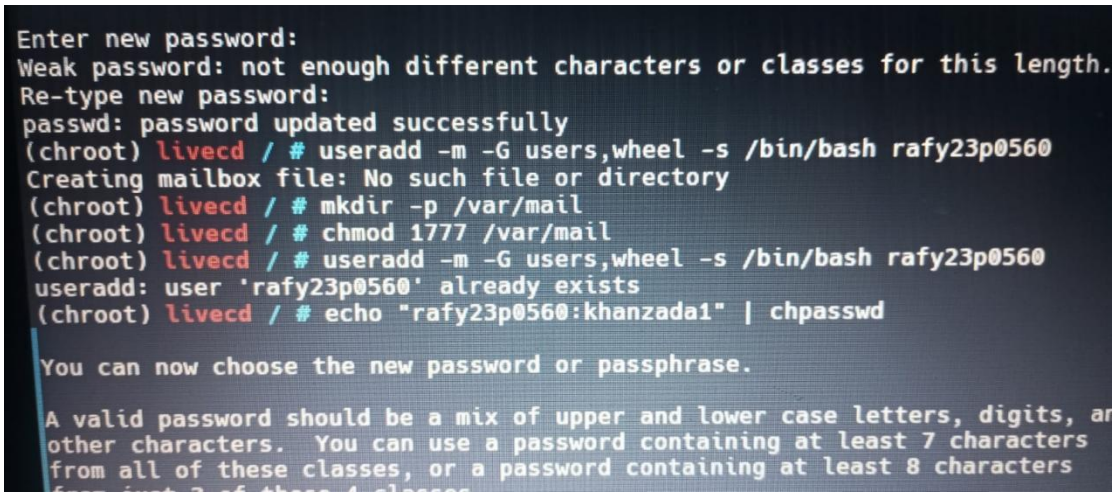


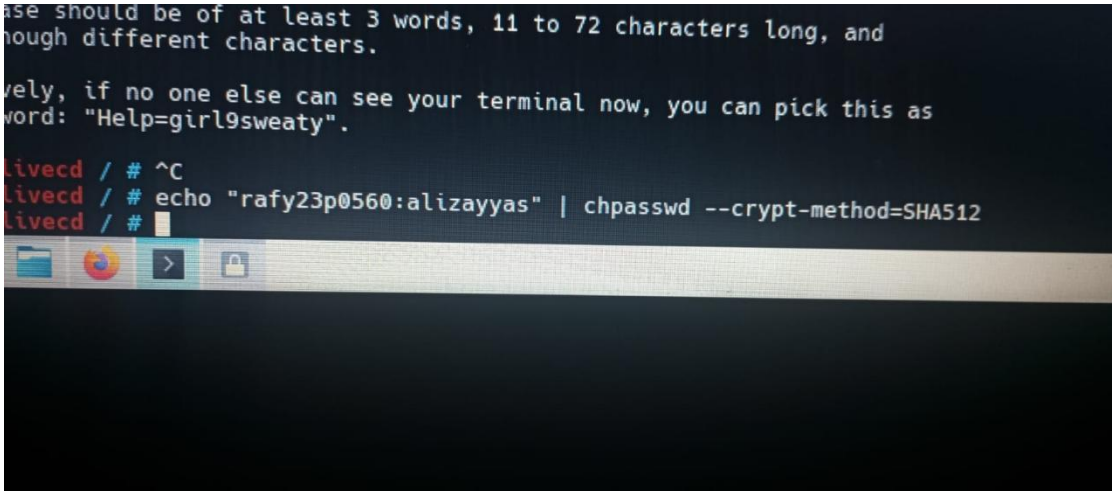
Step 12: Creating User and Finalizing Installation

Set root password (passwd).

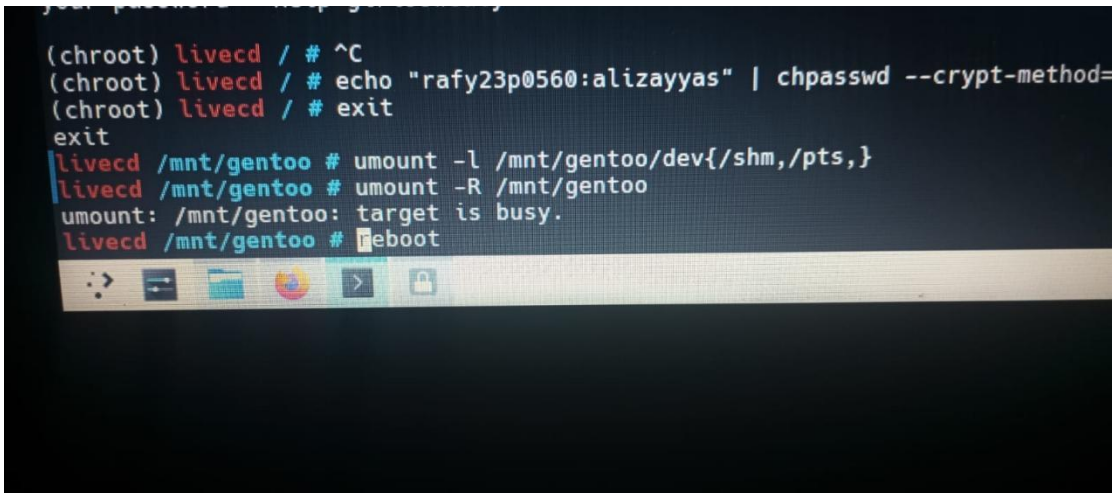


Created a new user with sudo privileges.

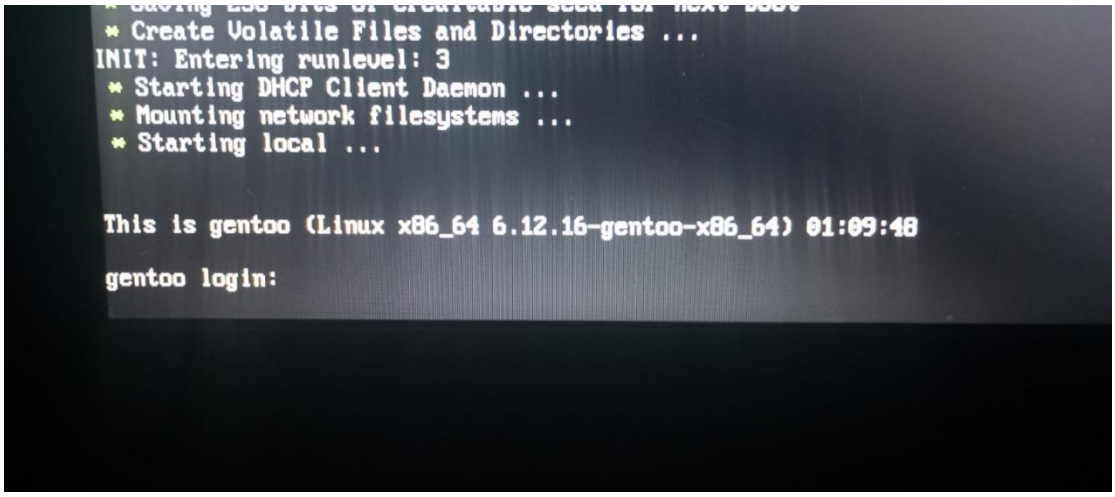


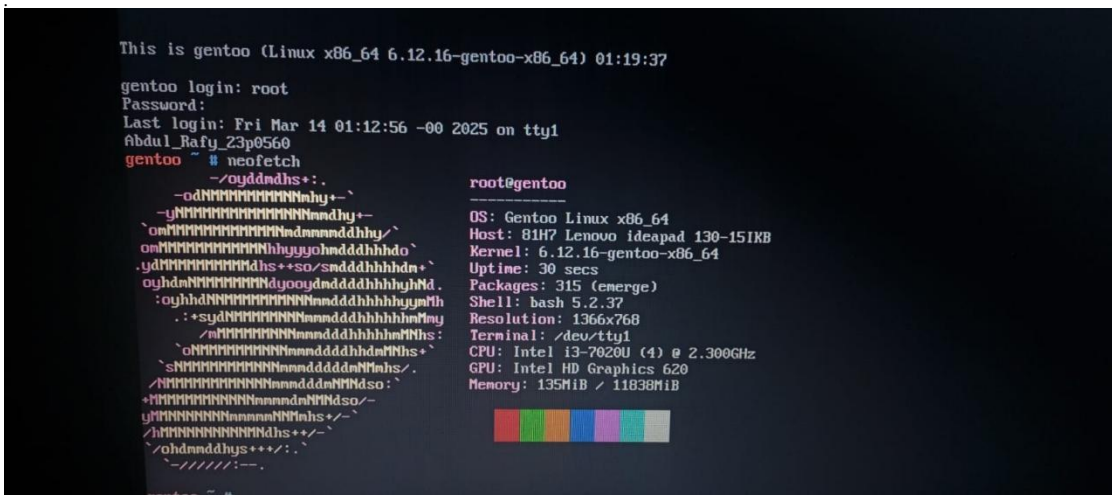


Exited chroot, unmounted partitions



Rebooted into fully installed Gentoo.



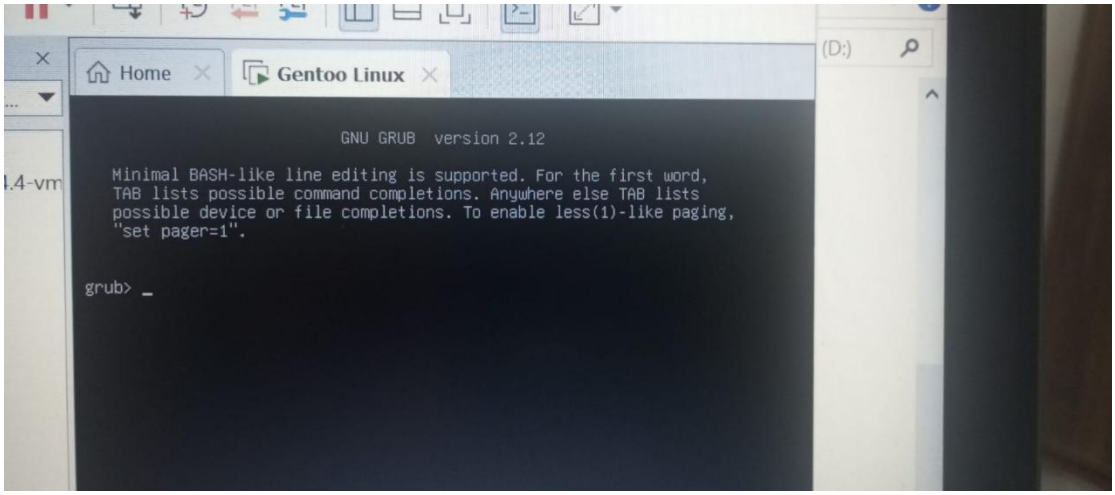


4. **Challenges Faced & Solutions**

I faced a ton of problems during this installation, but listing every single one would take too much time and space. Many were small mistakes, but the major challenges were:

1. First Time Installing Gentoo

- This was only the second OS installation I had ever done in my life.
- To practice, I first tried installing it on VMware multiple times. Each time, I would miss a step and had to start over.
- On my 7th attempt, I finally completed all the steps, but as soon as I rebooted....
- Kernel Panic! It also panicked me!
- After debugging, I found that I had not compiled the kernel properly, so GRUB could not find my kernel.



Its solution: Instead of configuring kernel manually I compile it through gen-kernel and ensured that GRUB was configured properly.

2. WiFi Not Connecting on Bare Metal Installation

- When I finally installed Gentoo on a friend's laptop (bare metal installation), the only real issue I faced was that WiFi was not connecting.

- The system couldn't detect or connect to available networks.

Its solution: I manually connected to WiFi using the command line with the help of ChatGPT. Once configured, everything worked fine.

5. What I Learned from This Installation

Installing Gentoo from scratch taught me a lot about Linux and system setup.like

How Linux Boots – I now understand the full process, from disk partitioning to configuring GRUB and compiling the kernel.

Troubleshooting Skills – I faced issues like kernel panic and WiFi not working, but solving them helped me get better at debugging.

Working Without a GUI – Since I did everything through the terminal, I learned how powerful CLI commands can be.

Patience & Persistence – The process wasn't easy, but completing it gave me a huge sense of achievement.

Gentoo Linux: Advantages and Challenges

Advantages

Highly Customizable – I can control everything, from the kernel to installed packages.

Better Performance – Since software is compiled from source, it runs faster and is optimized for my hardware.

Great Learning Experience – Installing and using Gentoo teaches a lot about Linux internals.

Always Up to Date – Uses a rolling release model, so I will always get the latest software.

Lightweight – No unnecessary software; I will install only what I need.

Challenges

Difficult Installation – Unlike Ubuntu, Gentoo requires manual setup, which can take hours or days.

Slow Software Installation – Since everything is compiled from source, installing software takes longer.

Hard for Beginners – Managing the kernel and system updates can be very tricky.

Not for Casual Users – Needs regular maintenance, making it less suitable for everyday use.

6. Conclusion

Installing gentoo was challenging but reqarding as well. It tought me deep system-level understanding and Now, I have a fully functional, optimized Gentoo system configured exactly to my needs.