5761's Guide to Installing Gentoo Linux (may or may not brick your device)

Preface

This guide showcases the steps that I (5761) took to install Gentoo Linux on my machine. The same steps should be viable to install Gentoo on your own machine, assuming it is an amd64 machine with UEFI support. If you find that you can't install Gentoo the same way as I did, please refer to the Gentoo handbook for more information.

Setup for Windows Dual Boot (Optional)

The following steps are for installing Gentoo on a machine that already has windows installed. It is meant to allow both windows and Gentoo to be used on the same machine via a dualboot setup without each OS interfering with each other. You can skip these steps if you're only planning to install Gentoo on the machine.

Disabling BitLocker

BitLocker is a feature in windows that automatically encrypts the data drives used in windows. This feature should be turned off when using a dualboot setup as windows might refuse to unencrypt the system when a change is made outside of windows such as while in Gentoo leading to windows not being usable. This feature can be turned off in window's settings and while it is possible to use bitlocker while also using Gentoo, it is recommended to turn it off as the configuration needed to keep bitlocker on is more complicated.

Disabling Fast Startup

Fast startup is a feature that allows windows to start up faster after a shutdown by saving an image of kernel memory in disk so that it can be loaded quickly on power on, possibly skipping boot up entirely. This feature needs to be turned off as the fast startup image on disk might interrupt the boot process of gentoo and bypass your preferred boot loader. The option for this can be found in window's control panel under the "change what power buttons do" setting.

Partitioning the Disk

If your machine has windows already installed, windows most likely have already allocated all disk space on the machine for itself. Because of that, you must repartition your disks to allocate enough space to be able to install Gentoo. You can do this either by formatting and unallocating a partition not containing the windows system or by shrinking the windows partition to allow

space for Gentoo. Both of these operations can be done using windows disk manager which can be launched by the popup menu that shows up when pressing windows + x.

Preparing for an Installation of Gentoo

Before installing Gentoo, we must first prepare the necessary environment for it. This includes getting a live environment that can run tools needed to install Gentoo and also preparing the partitions for a Gentoo install.

Obtaining a LiveCD

A LiveCD is a bootable iso image that contains the necessary tools to configure the machine and to build Gentoo. The Gentoo website provides a list of LiveCD isos that can be used to build Gentoo. This guide assumes that you're running an amd64 machine and as such should get the amd64 LiveCD. This guide will also use the non GUI version of the LiveCD so you can get the real Gentoo experience.

Creating a Bootable USB

After obtaining the LiveCD iso, we will need a way to burn the image into a bootable device such as a USB stick or CD. A USB stick will be used in this guide as they are more common to come across. Make sure that the USB that you're using is not being used to store any data and that the USB supports the GPT partitioning scheme. The LiveCD iso is burned to the USB stick from an already operational system. If using Windows, you can use rufus software to burn the LiveCD into the USB stick.

Booting Up From the USB

Plug the USB stick into the target machine that you want to install Gentoo on. Turn on the machine and wait for it to boot into the USB stick. If the machine already has an OS such as Windows, you should boot to the USB manually by using the system UEFI or BIOS interface that can usually be accessed by the alt, del, esc, or F12 button (every system is configured differently, you should try out different buttons until you find the one that opens up the BIOS or UEFI menu). Once in the menu, find the options to change the boot settings so you can boot to the USB instead of the machine's OS. You should also disable the Secure Boot setting that's present in the menu as the LiveCD image cannot be booted using Secure Boot. When exiting the menu, don't forget to save the changes that you've made so you don't boot back into the machine's OS. Once you've booted into the the USB stick, choose the first entry in the boot menu to enter the installation environment.

Enabling Networking

An internet connection is required to install Gentoo. If your machine is hooked up to an ethernet cable that has dhcp, it is possible that networking may work right away. You can check this by using the ping command on a known host to see if the machine can reach it. Machines that use WiFi should use nmtui to setup a connection via a text UI that's easy to use. Machines that require a static IP address should consult the Gentoo handbook. Before moving on the next steps, ensure that your machine can ping a known host.

Setting Up Partitions

A partition on the disk is needed to serve as the root filesystem of the Gentoo OS. You can allocate this partition by using the cfdisk tool which is a more user friendly version of the fdisk command. It is recommended to allocate at least 16 GB on the root partition (I'm not an expert on this but I have run out of space when allocating less than that). Gentoo also needs an EFI partition to place the kernel and bootloader to boot from. Your machine might already have this partition, but if not you must make one.

Writing Filesystems

Each partition that you've set up requires a filesystem to actually store and retrieve files. We can make use of the various mkfs commands to do that. This guide will use xfs as the filesystem on the root partition but you can use other filesystems such as ext4 or exfat. The EFI partition MUST use fat32 or it will not be recognized as a bootable partition by the system. You can check what filesystem each partition is using with the df command. This is useful to check if an existing EFI partition is already using fat32.

Mounting the Partitions

The root partition and the efi partition must be mounted on directories to continue the installation. Create a directory at /mnt/gentoo that will be the mount point of the root partition and /mnt/gentoo/efi that will be the mount point of the EFI partition. After mounting the two partitions, change the current directory to /mnt/gentoo to enter the root partition.

Installing a Stage File

After mounting the partitions, a stage file can be installed on the root partition via /mnt/gentoo. This stage file will contain the RUNNING environment of the future Gentoo system. Stage files have different profiles that can be seen on the Gentoo website. Each profile caters to a certain intended installation feature. For this guide, we will use the openrc desktop stage file. Inside the /mnt/gentoo directory, install the stage file from the gentoo website using the links command that will open a text based browser which you can navigate to get to the stage file downloads page. Download the stagefile you want (amd64 with desktop and no systemd). After you finish installing the file, unzip it to set up the running environment.

Changing Roots

Once the running environment has been setup, we will change the root of our installation to the running environment. This is so that any changes we make to the environment is done on the running environment and will persist after the installation is finished. We change roots by using the arch-chroot command that is provided in the LiveCD.

Gentoo Kernel Installation

Configuring Portage

The Gentoo package manager Portage is set up to compile packages from source by default. This allows for better customization of packages at the cost of much manual configuration. You can configure the flags that portage uses to compile packages in the /etc/portage/make.conf file. You can set up individual compiler options to create more optimized executables, but most importantly you can set USE flags that determine features that compiled packages use. This can be used to setup your system's features that you want your compiled packages to know and make use of. This variable can be updated throughout the installation, but personally I use the sign-modules, dist-kernel, secureboot, and dracut USE flags. These flags alter the kernel compilation process to create secure boot compatible kernels. You can also set individual USE flags for individual packages by adding files into the /etc/portage/package.use/ directory. The file names do not matter but each file contains one or more lines, each line specifying a package and a space separated list of USE flags to use when compiling that package. There are also other directories such as package.accept_license and package.accept_keywords to set the accept_license and accept_keywords flags for individual packages. You can also add these variables in the global make.conf file.

Installing the Repository

The Portage repository can be pulled using the emerge-webrsync command. This will pull information about all the packages available in Portage so that they can be downloaded and compiled in your system. This step is necessary in order to use Portage properly.

Machine Specific Drivers

In this step we supply portage with information about our machine so that it can build optimized packages and also kernels. We do this by writing to /etc/portage/package.use/00cpu-flags and /etc/portage/package.use/00video_cards. CPU flags can be obtained using the cpuid2cpuflags tool available on portage while video card flags have to be set manually. Refer to the Gentoo handbook for a list of available options.

Setting Proper Timezone and Locale Settings

Many system processes require correct timezone and locale settings to run properly. To configure the timezone, find the timezone that you want in the /usr/share/zoneinfo directory. You might have to traverse into a subdirectory to get to a specific timezone in a region. After finding the timezone that you want, create a symbolic link to /etc/localtime. To configure the locale, edit the file at /etc/locale.gen and uncomment the locales that you want to use. If you want to add locales not present in the file, refer to the Gentoo handbook. After changing the locale, run the locale-gen command, select the locale using eselect, then reload the environment.

Installing Firmware

Sometimes, certain hardware requires firmware to function. We can obtain linux firmware from the linux-firmware package and extra sound firmware from the sof-firmware package. If you are using an Intel machine, you should also get the intel-microcode package.

Configuring an EFI Stub Kernel

An EFI Stub will generate an efi executable for the kernel so that it can be booted to directly from UEFI. Creating an EFI stub also makes it easier to chainload the kernel from other bootloaders. We can configure the kernel installation to generate an efi stub using the installkernel package. The installkernel package will automate the process of generating the kernel, initramfs, and other images required and put them at the appropriate location. When using the installkernel package, make sure to apply the efi-stub USE flag to generate an efi stub. You should also set the dracut USE flag as a global flag to allow automated generation of an initramfs image.

Generating Secureboot Keys

Secureboot keys will be used to boot our kernel from secureboot. The USE flags modules-sign and secureboot will be used to tell the kernel installer to sign all modules and the kernel using our supplied keys. The keys can be specified as global variables MODULES_SIGN_KEY, MODULES_SIGN_CERT, SECUREBOOT_SIGN_KEY, SECUREBOOT_SIGN_CERT. All of those global variables will contain a path to the key and certificate file located somewhere in the system. Use openssh to generate an x509 key and certificate on the same file using the PEM format.

Installing the Kernel

There are multiple ways to install the kernel in Gentoo. This guide will focus on automatically compiling the kernel from source. This means that the kernel is still compiled on the machine but the used configuration is automatically determined. We do this by emerging the gentoo-kernel package which will automatically install the kernel, and generate the efistub, initramfs, and other modules at the efi/EFI/Gentoo directory.

Post Kernel Installation

After installing the kernel, there are still more installation procedures to go through to completely prepare Gentoo for usage on boot.

Creating an fstab File

The fstab file is an important file in Linux used to configure mount points at boot time. This file helps to store the configuration of the filesystem tree by storing mount points so that they are mounted to the same locations every boot. The fstab file is located at /etc/fstab and stores information in a table-like form with each column separated by a tab. For the partition identifier, I use PARTUUID as they are bound to the partition itself and will not change, giving me peace of mind that the filesystem tree won't just break one day. Details of the fstab file can be found on the Gentoo handbook.

Configuring Network Management

The network management provided by the LiveCD is not automatically available in the running environment. We need to install our own network management system. For this, I use the network-manager package which provides the nmtui tool that we used in the beginning to connect to Wifi. If you do not need wifi support and can make use of dhcp, use the dhcpcd package instead which provides automatic configuration. For other setups please refer to the Gentoo handbook. As a note when using network-manager, you should add the tkip USE flags as some old routers need the flag to be found by network-manager.

Adding a System Logger

Emerge the sysklogd package for an out of the box fully operational system logger. Use rc-update to add sysklogd to the default runlevel, allowing it to run from boot.

Adding a Cron Daemon

A cron daemon allows you to schedule certain operations at appointed times. This is useful for system maintenance and upkeep. You can add a cron daemon by emerging the dcron package. Once installed, run the crontab command with /etc/crontab as a parameter.

Synchronizing Time

Synchronizing time automatically is just a generally good convenience for users. It is easy to implement by just installing the chrony package and adding it to the default runlevel in openrc so that it starts on boot.

Installing Filesystem Tools

Filesystem tools allow you to manage the filesystems that your partitions use. It is handy to have these tools around for whenever you need to manage your partition's filesystems. Gentoo has several filesystem tools available, but for this install we will only get the xfs tools and vfat tools as they are the only filesystems present in our system.

Bootloader Installation

The bootloader that we will be using for this install is refind. Refind is easy to install with support for shim allowing for secureboot to be used with certificates other than those already present in the motherboard. This allows our source compiled Gentoo kernel to be loaded with secureboot.

Installing Shim

Shim can be installed from the shim package. By default, shim comes with the mokutils package which we will use to add our own kernel certificates to MOK, the key manager used by shim to verify efi executables. We will not only add the kernel certificate, but also add the refind certificate.

Installing Refind

Refind can be installed from the refind package. After obtaining both shim and refind, you can execute refind-install to install refind in the system's efi partition. Make sure that the efi partition is still mounted at this point to prevent any problems from using refind-install. After completing the install, refind will also generate a refind_linux.conf file at /boot. This file should be present at the directory which the Gentoo kernel efi stub lies (that is /efi/EFI/Gentoo). You can either move the file there or write a new one at that location to start specifying configurations. The /boot directory can be erased.

Configuring Refind and Shim

After installing both refind and shim, you can now add the kernel keys and also the refind key to the MOK using mokutils. The refind key should already be in the /efi/EFI/refind/keys directory while we need to move the kernel keys manually. Before that we must convert the kernel keys that were generated as a pem file to a der file using openssl first. Then we move the new der file to the /efi/EFI/refind/keys directory. Finally we can use mokutils to add the keys to MOK after a reboot. When rebooting, you can also enable secureboot back on as at this point, the system should be able to boot with secureboot.

Finalization

Installing Sway

Sway is the window manager that we will use to provide a desktop experience to users. It is a tiling window manager which means windows will automatically be layed out in tiles. Sway can be installed from the gui-wm/sway package. When installing sway, you should enable the wallpapers USE flags to obtain a default wallpaper for sway to use. Aside from Sway as the window manager, you should also install wmenu and foot packages which are the default application launcher and terminal emulator that Sway tries to use. After installing Sway, you must edit the default bashrc script and add Sway initialization code so that new users start sway after login.

Installing Applications

At this point, you can pretty much install any application you want. While doing this it is important to manage your USE flags, be it global or package specific. This is to ensure that each package interacts with each other just the way you expect it. Personally, I installed firefox as a web browser, vscode as a text editor, and wine for windows emulation.

Adding Users

After installing applications, you can add users so that you don't use the root user every time you use Gentoo. This is done using the adduser command. You can also specify the groups that the user belongs to that gives users certain privileges. A list of important groups can be found on the Gentoo handbook. The most important group to keep in mind is the wheel group, as this group has the privilege to change privilege levels and are usually the group used for admins of the system. After creating a user, you can also create a password for the user using the passwd command.

Other Stuff

Feature	Type	Proof
Gentoo Installation	Mandatory	Sorry but please look at the video, I'm sorry this is so ass
GUI	Mandatory	
Audio-Visual	Mandatory	
Internet Connection	Mandatory	

Package Management	Mandatory	
Unprivileged User	Mandatory	
Graphical Text Editor	Mandatory	
Graphical Web Browser	Mandatory	
Wallpaper	Mandatory	
Open Source Text Editor and Web Browser	Bonus	
Tiling Window Manager	Bonus	
Wayland	Bonus	
Doom on Terminal	Bonus	
Physical Hardware	Bonus	
Secure Boot	Bonus	
Secure Boot on Gentoo	Bonus	

https://drive.google.com/file/d/1uvb_CXWFFQlz-TpJJysfl3GBlnhPxunR/view?usp=drive_link
https://drive.google.com/file/d/1Z_PkDz2c5PcOX3h-fe4rNjdv6ghShmbD/view?usp=drive_link
https://drive.google.com/file/d/12X_xaNlJMAs7cERg-1kALRkVIeCYgPw7/view?usp=drive_link