# Installation von Gentoo

**Achtung, älterer Beitrag (nach Update, falls benötigt entfernen)**

## Download und Konfiguration der Festplatte

Zuerst partitonieren und formatieren wir unsere virtuelle Festplatte /dev/vdaX. Danach hängen wir sie in der Live-CD ein.

```
mount /dev/vda1 /mnt/gentoo/
```

Um Zeit und Datum zu überprüfen, führen Sie date aus: Wenn Zeit oder Datum falsch angezeigt wird, passen Sie die Einstellung mit der date MMDDhhmmYYYY Syntax (Month, Day, hour, minute und Year) an. Zu diesem Zeitpunkt sollten Sie UTC-Zeit verwenden. Sie werden später die Gelegenheit dazu haben Ihre lokale Zeitzone anzugeben. Um die Zeiteinstellung auf den 29. März 2005, 16:21 Uhr zu setzen, führen Sie zum Beispiel folgenden Befehl aus:

```
date 102522392010
```

Nun laden wir uns die aktuell Stage3 herunter. Hierzu wechseln wir in unser zuvor eingehängtes Laufwerk.

```
cd /mnt/gentoo
wget -c
http://mirrors.kernel.org/gentoo/releases/amd64/autobuilds/current-stage3/st
age3-amd64-20101021.tar.bz2
wget -c
http://mirrors.kernel.org/gentoo/releases/amd64/autobuilds/current-stage3/st
age3-amd64-20101021.tar.bz2.DIGESTS
wget -c
http://mirrors.kernel.org/gentoo/releases/amd64/autobuilds/current-stage3/st
age3-amd64-20101021.tar.bz2.CONTENTS
```

Jetzt überprüfen wir noch unseren Download.

```
md5sum -c stage3-amd64-20101021.tar.bz2.DIGESTS
```

Ist alles OK, können wir das Archiv entpacken.

```
tar xvjpf stage3-*.tar.bz2
```

# Installation von Portage

Sie müssen einen Snapshot des Portage-Tree installieren, eine Sammlung von Dateien, die Portage informieren, welche Porgramme installiert werden können, welche Profile es gibt, etc. Wechseln Sie zu dem Mountpoint, an den Sie Ihre Dateisystem gemountet haben (meistens /mnt/gentoo):

```
wget -c http://gentoo.inode.at/snapshots/portage-latest.tar.bz2 && wget -c
http://gentoo.inode.at/snapshots/portage-latest.tar.bz2.md5sum && wget -c
http://gentoo.inode.at/snapshots/portage-latest.tar.bz2.gpgsig
```

Jetzt überprüfen wir noch unseren Download.

```
md5sum -c portage-latest.tar.bz2.md5sum
```

Ist alles OK, können wir das Archiv entpacken.

```
tar xvjf /mnt/gentoo/portage-latest.tar.bz2 -C /mnt/gentoo/usr
```

# Die make.conf

Um Gentoo zu optimieren, können Sie einige Variablen setzen, die Einfluss auf das Verhalten von Portage nehmen. Alle diese Variablen können als Umgebungsvariablen (mit export) gesetzt werden, dies ist aber keine dauerhafte Lösung. Um Ihre Einstellungen zu behalten, bietet Portage Ihnen /etc/make.conf, eine Konfigurationsdatei für Portage. Diese Datei werden wir nun bearbeiten.

```
nano /mnt/gentoo/etc/make.conf
```

```
#PORTAGE_ELOG_CLASSES="log warn error info"
#PORTAGE_ELOG_SYSTEM="echo save"
PORTAGE_NICENESS="10"


CFLAGS="-march=native -O2 -pipe"
CXXFLAGS="${CFLAGS}"
CHOST="x86_64-pc-linux-gnu"

USE="ssse3 bonjour zeroconf avahi symlink -X"

FEATURES="ccache"
#FEATURES="ccache -pid-sandbox" # For CT's, LXC
CCACHE_SIZE="2G"
MAKEOPTS="-j3"

INPUT_DEVICES="evdev"

LINGUAS="de"
ACCEPT_LICENSE="*"
```

### make.conf für Citrix Xengäste

Um einen aktuellen Spiegelserver zu benutzen suchen wir uns einen aus:

```
mirrorselect -i -o >> /mnt/gentoo/etc/make.conf
mirrorselect -i -r -o >> /mnt/gentoo/etc/make.conf
```

Kopieren der DNS-Informationen

```
cp -L /etc/resolv.conf /mnt/gentoo/etc/
```

## Betreten der Chroot

```
mount -t proc none /mnt/gentoo/proc
mount -o bind /dev /mnt/gentoo/dev
```

Nachdem nun alle Partitionen initialisiert und das Basissystem installiert ist, können Sie nun Ihre neue Installationsumgebung betreten, indem Sie in die Umgebung chrooten. Das bedeutet, dass Sie von der aktuellen Installationsumgebung (die Installations-CD oder ein anderes Installationsmedium) in Ihr zu installierendes System (nämlich das der initialisierten Partitionen) wechseln. Das chrooting erfolgt in 3 Schritten. Als erstes werden wir das root von / (auf dem Installationsmedium) nach /mnt/gentoo (auf Ihren Partitionen) mittels chroot wechseln. Sie müssen nun eine neue Umgebung anlegen, indem Sie env-update benutzen, was hauptsächlich neue Umgebungsvariablen anlegt. Zu guter Letzt laden Sie diese Variablen durch den Befehl source in den Speicher.

```
chroot /mnt/gentoo /bin/bash
env-update
source /etc/profile
export PS1="(chroot) $PS1"
```

## Portage konfigurieren

Sie sollten nun Ihren Portage-Tree auf den aktuellsten Stand bringen. `emerge –sync` macht dies für uns. Zunächst ist eine kleine Defintion notwendig. Ein Profil ist existenziell für ein Gentoo-System. Es definiert nicht nur Standardwerte für USE, CFLAGS und andere wichtige Variablen, es beschränkt das System auch auf eine definierte Masse an Paketversionen. Dies alles wird von Gentoo-Entwicklern betreut. Bisher hatten Benutzer mit Profilen wenig zu tun. Jedoch kann es bestimmte Umstände geben in denen ein Profilwechsel möglicherweise notwendig ist. Sie können das aktuell verwendete Profil mit dem folgenden Befehl anzeigen:

```
eselect profile list
```

Unseren Fall bleiben wir bei dem Standardprofil.

# Setzen der Zeitzone und der Sprache

```
cp /usr/share/zoneinfo/Europe/Vienna /etc/localtime
nano -w /etc/conf.d/hwclock (sollte auf UTC stehen)
nano /etc/timezone
```

und dies eintragen:

```
Europe/Vienna
```

Jetzt kann man schon mal mit „date“ die Zone überprüfen. Es wird empfohlen die Zeit mit Hilfe eines Zeitservers abzugleichen. Jetzt noch die Datei /etc/env.d/02locale und die gewünschte Codierung eintragen.

```
nano /etc/env.d/02locale
```

und

```
LANG="de_AT.utf8"
LC_COLLATE="C"
```

und

```
nano /etc/locale.gen
```

Folgende Dinge eintragen:

```
#en_US ISO-8859-1
en_US UTF-8
#de_DE ISO-8859-1
de_DE UTF-8
de_AT UTF-8
```

eintragen. Damit die Änderungen auch wirksam werden muss man die locale generieren und die ENV neu einlesen.

```
locale-gen
env-update && source /etc/profile && export PS1="(chroot) $PS1"
locale -a
eselect locale list
eselect locale set <your number>
```

neu einlesen.

So das System ist jetzt auf Deutsch localisiert. Das heist aber noch lange nicht das die Programm auch auf Deutsch sind. Hierzu muss man natürlich die Deutschen Sprachpakete nachinstallieren. Möchte man auch das Tastaturenlayout auf Deutsche Tastatur umstellen, geht man wie folgt vor. Das von der Konsole verwendete Tastaturlayout wird in /etc/conf.d/keymaps durch die Variable KEYMAP gesetzt. Gültige Werte können /usr/share/keymaps/{arch}/ entnommen werden. i386 hat weitere

Unterteilungen nach dem Layout (qwerty/, azerty/, usw.). Einige Sprachen haben mehrere Optionen, daher müssen Sie möglicherweise etwas experimentieren, um entscheiden zu können, welche davon Ihren Bedürfnissen am ehesten entspricht.

```
nano /etc/conf.d/keymaps
```

Und folgendes eintragen:

```
KEYMAP="de"
KEYMAP="de-latin1"
KEYMAP="de-latin1-nodeadkeys"
```

| **A C H T U N G** |
| --- |
| Das Rootpasswort mit **passwd** nicht vergessen zu setzen!!! |

# Bauen des Kernels mit Genkernel und wichtigen Paketen

Zuerst erstellt man sich die richtigen Useflags:

```
nano /etc/portage/package.use

edia-libs/gd jpeg png
net-dns/avahi mdnsresponder-compat autoipd howl-compat dbus python
net-misc/ntp caps -zeroconf
media-libs/gd truetype fontconfig
www-client/elinks ftp
net-nds/openldap minimal
app-portage/layman cvs git mercurial subversion
#app-admin/webmin apache2 mysql
net-analyzer/nagios-plugins ipv6 nagios-ssh ssl
net-analyzer/nagios-nrpe command-args
net-fs/nfs-utils -nfsv3
```

**Webmin freischalten:**

```
echo "app-admin/webmin" >> /etc/portage/package.unmask
nano /etc/portage/package.keywords

app-admin/webmin **
>=dev-perl/Authen-Libwrap-0.220.0 ~amd64
```

Am einfachsten kommt man über das Tool genkernel zu einem lauffähigen Kernel für das neu zu installierende System. Es konfiguriert und kompiliert die Kernel-Quellen. Da es nicht per Default installiert ist, müssen wir es an dieser Stelle einspielen.

```
emerge -av sys-kernel/genkernel app-portage/portage-utils gentoo-sources
sys-boot/grub ccache net-misc/dhcpcd
app-portage/gentoolkit app-portage/portage-utils eix avahi htop nmap net-
misc/ntp acpid layman pciutils nagios-plugins
```

```
vixie-cron app-admin/syslog-ng elogv lsof mlocate app-misc/screen sys-
auth/nss-mdns games-misc/cowsay
```

Um nun wirklich den einfachsten Weg zu wählen, machen wir uns zu Nutze, dass der Kernel der LiveDVD auf dem gerade zu installierenden System funktioniert. Ließ sich der Rechner erfolgreich von der Boot-DVD starten, muss die Konfiguration des Kernels auf der DVD für das entsprechende System passen. Entsprechend können wir die Konfiguration des aktuell über laufenden Kernels aus /proc/config.gz unbesehen übernehmen und so die Tücken einer vollständigen Kernel-Konfiguration erst einmal umgehen:

```
zcat /proc/config.gz > /usr/src/linux/.config
```

Mit dieser Basiskonfiguration ist genkernel in der Lage, den Kernel automatisiert zu übersetzen und zu installieren:

```
genkernel --dmraid --bootloader=grub --oldconfig --menuconfig all
```

## Bauen von Genkernel mit Virtio für KVM

Hierfür muss man zuerst mal die Datei

```
nano /etc/genkernel.conf
```

und folgendes am Ende eintragen.

```
MODULES_KVM="virtio virtio_balloon virtio_ring virtio_pci virtio_blk
virtio_net"
```

Danach muss man vor dem Kernel kompilieren noch folgende Dinge als Modul konfigurieren:

```
genkernel --bootloader=grub --oldconfig --menuconfig all

Device Drivers --> Block devices --> Virtio block driver
Device Drivers --> Network device support --> Virtio network driver

cat .config | grep VIRTIO
CONFIG_VIRTIO_BLK=m
CONFIG_VIRTIO_NET=m
# CONFIG_VIRTIO_CONSOLE is not set
# CONFIG_HW_RANDOM_VIRTIO is not set
CONFIG_VIRTIO=m
CONFIG_VIRTIO_RING=m
CONFIG_VIRTIO_PCI=m
CONFIG_VIRTIO_BALLOON=m
```

Nach dem Kompilieren muss man in der **nano grub/menu.lst** noch folgende Kerneloption anhängen:

```
doload=virtio_pci,virtio_blk
```

# Netzwerk und Hostname konfigurieren

```
cd /etc && echo "127.0.0.1 gnagios.darkwolf.local gnagios localhost" > hosts
&& sed -i -e 's/HOSTNAME.*/HOSTNAME="gnagios"/' conf.d/hostname && hostname
gnagios && hostname -f
```

# OpenRC

In Openrc sollte man in der Konfiguratonsdatei noch folgende Änderungen vornehmen und Datensatze ein-kommentieren.

```
nano /etc/rc.conf
rc_hotplug="*"
rc_sys=""
```

# Konfigurieren der FSTAB

```
# <fs>                  <mountpoint>    <type>          <opts>
<dump/pass>

/dev/sda1               /boot           ext3            noauto          0 0
/dev/sda2               /               ext4            noatime,discard,acl
0 1
/dev/sdb1               /var/tmp/portage     ext4       noatime,auto    0 0
/dev/sdb3               /mnt/daten      jfs             noatime,auto    0 2
/dev/sdb4               /mnt/system     ext4            noatime,auto,acl
0 3
/dev/cdrom3             /media/cdrom      auto            noauto,user     0
0
#/dev/sdc1              /mnt/backup-ztor ext4           noauto,user,noatime
0 0
# glibc 2.2 and above expects tmpfs to be mounted at /dev/shm for
# POSIX shared memory (shm_open, shm_unlink).
# (tmpfs is a dynamically expandable/shrinkable ramdisk, and will
#  use almost no memory if not populated with files)
shm                     /dev/shm        tmpfs           nodev,nosuid,noexec
0 0
#virtualbox USB fix
#none /proc/bus/usb usbfs devgid=1017,devmode=664 0 0

#none /proc/bus/usb usbfs devgid=85,devmode=664 0 0
```

# Grub konfigurieren

```
nano /boot/grub/menu.lst

title Gentoo Linux 2.6.34-gentoo-r12
root (hd0,0)
kernel /boot/kernel-genkernel-x86_64-2.6.34-gentoo-r12 root=/dev/ram0
real_root=/dev/vda1
initrd /boot/initramfs-genkernel-x86_64-2.6.34-gentoo-r12
```

Grub installieren:

```
grep -v rootfs /proc/mounts > /etc/mtab
```

Wegen eines Bugs Grub >2.0 muss folgende Datei bearbeiten:

```
nano /boot/grub/device.map

(hd0) /dev/vda
```

Jetzt kann man mit folgenden Befehl Grub wie gewohnt installieren:

```
grub-install --no-floppy /dev/vda
```

Nun steht dem Reboot ins neue System nichts mehr entgegen.

# Nach dem ersten Start des neuen Gentoosystems

Ist alles richtig hochgefahren muss man folgende Dinge in die „init.d“ eintragen. Vorher kann man noch die aktuellen Einträge mit **rc-status** aufrufen.

```
rc-update add ntp-client default
rc-update add avahi-daemon default
rc-update add avahi-dnsconfd default
rc-update add dbus default
rc-update add udev-postmount default
rc-update add sshd default
rc-update add net.eth0 default
rc-update add vixie-cron default
rc-update add acpid default
rc-update add syslog-ng default
```

Sollte es Probleme mit dem Zeitstempel Timestamp geben, kann dies wie folgt begeben.

```
touch /etc/{conf.d,init.d}/*
```

Crontab konfigurieren, mit dem Befehl

```
crontab -e
```

und dann folgendes eintragen:

```
# Diese Datei wurde von KCron generiert Donnerstag 05 August 2010 15:51.
# Shell variable for cron
SHELL=/bin/bash
# PATH variable for cron
PATH=/usr/local/bin:/usr/local/sbin:/sbin:/usr/sbin:/bin:/usr/bin:/usr/bin/X
11
```

Nun sollte man das System von auf den aktuellen Stand bringen und das Basesystem mit Optimierung neu bauen.

```
emerge -qeauDN world
```

Und das wars auch schon.

## Weitere Informationen

- Gentoo Handbuch AMD64
- Bug in Virtio und Grub >2.0
- Kernelkonfiguration für Virtio

From:
https://deepdoc.at/dokuwiki/ - **DEEPDOC.AT - enjoy your brain**

Permanent link:
**https://deepdoc.at/dokuwiki/doku.php?id=gentoo:installation_von_gentoo**

Last update: **2019/04/14 09:19**