

gentoo

2013

Содержание

1. Введение	3
1.1. Gentoo	3
2. Установка. Обновление. Профили	8
2.1. Системные требования	8
2.2. Gentoo	8
2.2.1. Gentoo. Установка	8
2.2.2. Gentoo. Обновление	21
2.3. Управление профилями пользователей	21
3. Работа с командной строкой	22
3.1. Поиск файлов	22
4. Менеджер пакетов	23
4.1. утилита rs-update	23
4.2. создание RAID	23
4.3. Установка и удаление программ	23
4.4. Разрешение зависимостей	23
4.5. Поиск зависимостей	23
4.6. Написание собственных ebuild файлов	23
5. Настройка	24
5.1. Настройка сети	24
5.2. Настройка авто запуска	24
5.3. Персональные настройки	24

5.3.1.	Настройка /etc/inputrc	24
5.3.2.	Настройка bash	24
5.3.3.	Настройка gitconfig	25
5.3.4.	Настройка Xorg.conf	26
5.3.5.	Настройка mplayer2	26

6. Мониторинг и производительность 28

6.1.	Мониторинг	28
6.2.	Оптимизация eіx	28
6.3.	SSD	28
6.4.	Распределенная компиляция	28

7. Приложение 29

Список литературы 30

1. Введение

Данный документ распространяется под лицензией CC-BY¹.

1.1. Gentoo

Gentoo Linux (произносится: дженту, сленг - генту, гента) - достаточно популярный дистрибутив Linux с мощной и гибкой технологией Portage, которая совмещает в себе возможности конфигурирования, настройки, а также автоматизированную систему управления пакетами. Последняя создавалась под влиянием системы управления пакетами в FreeBSD. Отличительной особенностью Gentoo является наличие оптимизации под конкретное аппаратное обеспечение.

Дистрибутив ведёт своё начало с разработки Дэниелом Роббинсом (англ.) дистрибутива Epoch Linux в 1999 году. Благодаря системе Portage, позволяющей сделать из Gentoo практически всё, что угодно (от сервера до рабочей станции), этот дистрибутив можно отнести к так называемым метадистрибутивам.

¹Creative Commons - <http://creativecommons.org/licenses/by/3>

Gentoo (по-русски читается генту, название вида пингвинов укрепилося в русском языке задолго до появления дистрибутива) - это английское название вида пингвинов *Pygoscelis papua*. Согласно Книге рекордов Гиннеса 98, этому виду пингвинов принадлежит рекорд скорости плавания (36 км/ч).

Gentoo появился на основе разработок собственного дистрибутива Даниэла Роббинса под названием Enosh Linux. Основной целью дистрибутива являлась собственная сборка (подобная Linux From Scratch), оптимизированная под конкретное аппаратное обеспечение и содержащее минимальный, необходимый пользователю, набор программ.

Эксперименты над сборками GCC показали, что производительность получаемого пакета увеличивалась (в зависимости от аппаратного обеспечения) от 10 до 200 % по сравнению с бинарными сборками gcc, поставляемыми другими дистрибутивами. Нарботки, увеличивающие производительность, были включены в официальный выпуск GCC 2.95, благодаря чему другие дистрибутивы также получили дополнительный прирост в производительности. Начиная с этого момента, Enosh начал приобретать репутацию быстрого дистрибутива, после чего было принято решение сменить название на Gentoo Linux.

31 марта 2002 года вышла первая версия дистри-

бутива. В 2004 году была основана некоммерческая организация Gentoo Foundation, в ведение которой Роббинс передал все права на исходный код и торговые марки, тем самым отстранившись от разработки дистрибутива. На данный момент разработка управляется группой Board of Trustees из пяти человек (ответственных за управление фондом) и советом Gentoo Council (технические вопросы), состоящим из семи членов, выбираемых на срок в один год. В начале 2009 года обновился дизайн сайта. На главной странице отображаются новости, обсуждения и статьи разработчиков, а также извещения о критических уязвимостях (Gentoo Linux Security Advisory). В октябре 2009 года Gentoo исполнилось десять лет, в связи с чем подготовлена большая коллекция скриншотов и хронология дистрибутива.

Философия Gentoo проистекает от дней основания дистрибутива и остаётся в неизменном виде и по сей день. Вот как вкратце излагает концепцию проекта его основатель Дэниэл Роббинс:

Каждому пользователю приходится выполнять определённую работу. Цель Gentoo — разработка инструментов и систем, позволяющих пользователю заниматься своим делом как можно эффективнее и в своё удовольствие, так, как он сочтёт нужным. Наши инструменты должны приносить радость и помогать

пользователю оценить по достоинству все богатство Linux и сообщества свободного программного обеспечения, а также гибкость свободных программ. Такое возможно только тогда, когда инструменты создаются, чтобы отражать и проводить волю пользователя, оставляя для него открытыми все возможности с самого начала (с исходного кода). Когда инструмент заставляет пользователя действовать определённым образом, инструмент работает против него, а не на него. Все мы сталкивались с ситуациями, когда инструменты стремились навязать нам свою собственную волю. Такой подход — полная противоположность, несовместимая с философией Gentoo. Иными словами, философия Gentoo — создание лучших инструментов. Когда инструмент в совершенстве выполняет свою задачу, вы можете даже не замечать его присутствия, потому что он не перечит вам, не проявляет себя, и не заставляет вас возиться с ним, когда вам совершенно не до этого. Инструмент служит пользователю, а не пользователь — инструменту. Будущая задача Gentoo — продолжать борьбу за создание инструментов, близких к идеалу. Инструментов, удовлетворяющих нужды множества различных пользователей (каждого — со своими разнообразными целями) с простотой, идущей рука об руку с непревзойдённой мощностью. Разве вы не любите

пользоваться инструментами, которые отлично подходят для ваших нужд? Разве это не великолепное ощущение? Наша цель — передать это чувство как можно большему числу людей.

— Дэниел Роббинс

2. Установка. Обновление. Профили

2.1. Системные требования

2.2. Gentoo

2.2.1. Gentoo. Установка

Для установки необходимо скачать установочный диск с минимальным набором программ: `install-amd64-minimal-...iso` (`install-amd64-minimal-20110811`).

После того, как установили настройки в биосе, о том что необходимо произвести загрузку с `cdrom`, грузимся. После загрузки нам предлагают выбрать какое ядро грузить, пишем `gentoo`. Далее идет процесс загрузки ядра и в результате получаем консоль с правами доступа `root`.

Подготавливаем HDD

Для установки потребуется жесткий диск размером не менее 6,5 гб. Разбить диск необходимо следующим образом:

mount point	size	filesystem	part
swap	2 * RAM	swap	/dev/sda
/boot	200 Mb	ext2	/dev/sda
/	минимум 6 Гб	ext4	/dev/sda

Что бы произвести разметку диска, необходимо воспользоваться программой fdisk:

```
# fdisk /dev/sda
```

После того как разделы были созданы, необходимо их отформатировать, с указанием файловой системы:

```
# mkfs.ext2 /dev/sda2
# mkfs.ext4 /dev/sda4
```

На ssd отключаем журналирование:

```
# tune2fs -o journal_data_writeback /dev/sdaX
```

затем в fstab прописывается опция монтирования data=writeback

О режимах:

Режим data=writeback - файловая система не производит какого либо журналирования данных. При

неожиданных перезагрузках системы это может вызвать потерю данных в обновляемых файлах. Данный режим обеспечивает самую высокую производительность.

Режим `data=ordered` - файловая система журналирует только метаданные (данные и методанные группируются в один модуль - транзакцию). Этот режим, хотя без гарантии, защищает данные при неожиданной перезагрузке, в отличие от предыдущего. Тем не менее полного журналирования не происходит. Производительность уступает `data=writeback`, но она гораздо быстрее полного журналирования.

Режим `data=journal` - обеспечивает полное журналирование метаданных и самих данных. Данные сначала пишутся в журнал и потом только переносятся на постоянное место. При аварийных ситуациях журнал можно перечитать - приведя данные в непротиворечивое состояние. Данный режим самый медленный, но в отдельных случаях он показывает хорошие результаты. Он имеет преимущества при одновременных операциях ввода/вывода данных (при записи и одновременном чтении, скорость чтения в тестах была выше на порядок чем при других режимах).

Внимание: для использования любого другого режима, кроме `data=ordered` для корневой файловой

системы, вам придется задать в параметрах загрузки ядра (boot/grub/menu.lst) следующую строку:

```
rootflags=data=writeback
```

Создадим swap и запустим его:

```
# mkswap /dev/sda1  
# swapon /dev/sda1
```

Устанавливаем разделы

```
# mount /dev/sda3 /mnt/gentoo  
# mkdir /mnt/gentoo/boot  
# mount /dev/sda2 /mnt/gentoo/boot
```

Далее необходимо скачать последние пакеты (stage3-i686-xxx.tar.bz2, portage-latest.tar.bz2)

Настройка сети

откроем для редактирования след. файл:

```
# vim /etc/conf.d/net
```

допишем настройки:

```
# config_eth0=("192.168.0.2 netmask 255.255.255.0 br  
## config_eth0=("dhcp")  
# routes_eth0=("default via 192.168.0.1")
```

добавим настройку сети в загрузку:

```
# rc-update add net.eth0 default
* net.eth0 added to runlevel default
# cd /etc/init.d/
# ln -s net.lo net.eth1
# rc-update add net.eth1 default
* net.eth1 added to runlevel default
```

Скачиваем пакеты. В случае, если нету интернета, а установить приложение очень хочется, то можно скачать пакет руками на другой машине и положить архив в каталог:

```
/usr/portage/distfiles
```

Распаковываем:

```
# cd /mnt/gentoo
# tar xvjpf /mnt/gentoo/tmp/stage3-*.tar.bz2
```

Затем:

```
# cd /mnt/gentoo
# tar xvjf /mnt/gentoo/tmp/portage-latest.tar.bz2 -C
```

Отредактируем файл:

```
# vim /mnt/gentoo/etc/make.conf
```

исправив строчку с параметрами компиляции на:

```
LINGUAS="ru"
```

```
ACCEPT_LICENSE="*"
```

```
INPUT_DEVICES="evdev keyboard mouse virtualbox"
```

```
VIDEO_CARDS="nvidia"
```

```
PORTDIR_OVERLAY="/usr/local/portage"
```

```
#source /var/lib/layman/make.conf
```

```
USE="sse nls sse2 sse3 ssse3 sse4_1 sse4_2 avx  
qt4 python theora python2_7 python3_3  
gmp posix usb png ogg udev threads xvid webm  
x264 ogg flac opengl policykit consolekit X  
xorg vdpau unicode utf8 doc iconv icu m17n-lib "
```

```
CFLAGS="-march=native -mtune=corei7 -Ofast -flto  
-pipe -msse4.1 -msse4.2 -mpopcnt -fomit-frame-pointe
```

```
MAKEOPTS="-j8"
```

```
EMERGE_DEFAULT_OPTS="--jobs=4"
```

Вход в новую систему

Выбираем зеркало.

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

затем копируем dns:

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

монтируем /proc и /dev:

```
# mount -t proc none /mnt/gentoo/proc
```

```
# mount -o bind /dev /mnt/gentoo/dev
```

переносим оставшиеся настройки:

```
# chroot /mnt/gentoo /bin/bash
```

```
# env-update
```

```
>> Regenerating /etc/ld.so.cache...
```

```
# source /etc/profile
```

```
# export PS1="(chroot) $PS1"
```

Конфигурируем Portage

```
# emerge --sync --quiet
```

Укажем в переменную USE программы, которые мы хотим тоже скомпилировать. список всех программ доступен для просмотра так:

```
# less /usr/portage/profiles/use.desc
```

откроем для редактирования файл /etc/make.conf
и уточним переменную USE:

```
# vim /etc/make.conf
```

```
USE="sse sse2 sse3 ssse3 sse4.1 sse4.2 qt4 gmp posix
```

Осталось только локализовать:

```
# vim /etc/locale.gen
```

в файл /etc/locale.gen записываем:

```
en_US.UTF-8 UTF-8
```

```
ru_RU.UTF-8 UTF-8
```

в файл /etc/environment записываем:

```
LANG="en_US.UTF-8"
```

в файл /etc/conf.d/consolefont записываем:

```
consolefont="LatArCyrHeb-16"
```

в файл /etc/conf.d/keymaps записываем:


```
keymap="us"  
windowkeys="yes"  
extended_keymaps=""  
dumpkeys_charset=""  
fix_euro="no"
```

запускаем генерацию:

```
# locale-gen
```

Конфигурируем ядро.

определим время:

```
# cp /usr/share/zoneinfo/Etc/GMT+3 /etc/localtime
```

Выбираем ядро и установим с помощью emerge:

```
# emerge gentoo-sources
```

конфигурируем ядро:

```
# cd /usr/src/linux  
# make menuconfig
```

После компилируем:

```
# make && make modules_install
```

затем копируем в загрузочный раздел получившиеся ядро:

```
# cp arch/x86_64/boot/bzImage /boot/kernel-3.1.4-gen
```

Конфигурирование системы

в файле `/etc/fstab` содержится информация о том, какие разделы и где находятся и параметры их монтирования. Этот файл нужно отредактировать.

`/etc/fstab` использует специальный синтаксис. Каждая строка - это запись, состоящ. из 6 полей, разделенные табом или пробелами:

- `partition` - путь к файлу устройства
- `mount point` - указывает точку монтирования, в которую должен быть установлен раздел
- `filesystem` - файловая система, которую использует раздел
- `mount options` - опции монтирования, разделенные запятыми(подробнее в `man`)
- `dump` - используется дампом для определения сбрасывания или нет(Нужно ставить 0)

- `fsck` - используется для определения и проверки файловой системы, если она была выключена не должным образом. Конечная файловая система должна иметь 1, в то время как остальные должны иметь 2 (или 0, если проверка файловой системы не нужна).

Переходим непосредственно к редактированию:

```
# vim /etc/fstab
```

должно получиться что то такое:

<code>/dev/sda1</code>	<code>none</code>	<code>swap</code>	<code>sw</code>
<code>/dev/sda2</code>	<code>/boot</code>	<code>ext2</code>	<code>defaults, noat</code>
<code>/dev/sda3</code>	<code>/</code>	<code>ext4</code>	<code>defaults, noat</code>
<code>/dev/cdrom</code>	<code>/mnt/cdrom</code>	<code>auto</code>	<code>noauto,user,ro</code>
<code>proc</code>	<code>/proc</code>	<code>proc</code>	<code>nosuid,noexec,</code>

Настройка входа в систему

сменим пароль рута:

```
# passwd
```

Установим вход в систему:

```
# emerge syslog-ng
# rc-update add syslog-ng default
```

```
# emerge vixie-cron
# rc-update add vixie-cron default
# crontab /etc/crontab

## for indexing your files
# emerge mlocate

# emerge dhcpcd

# emerge ppp vim links ...
```

Конфигурируем загрузчик

Теперь, когда наше ядро сконфигурировано и откомпилировано, и необходимые системные файлы конфигурации заполнены правильно, пришло время установить программу, которая запустит наше ядро.

Установим GRUB:

```
# emerge grub
# vim /boot/grub/grub.conf
```

Отредактируем файл, и получим приблизительно след.:

```
default 0
timeout 10
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
```

```
title Gentoo Linux 3.0.3
root (hd0,1)
kernel /boot/kernel-3.0.3-gentoo root=/dev/sda1 root
```

```
title windows XP
rootnoverify (hd0,5)
makeactive
chainloader +1
```

установим grub:

```
# grep -v rootfs /proc/mounts > etc/mtab
# grub-install --no-floppy /dev/sda
```

Теперь все ! ребутаемся.

```
# exit
cdimage~# cd
cdimage~# umount /mnt/gentoo/boot
cdimage~# umount /mnt/gentoo/dev
cdimage~# umount /mnt/gentoo/proc
cdimage~# umount /mnt/gentoo
cdimage~# reboot
```

После загрузки можно поставить eix, layman. Искать можно так:

```
# emerge --search eix
```

Заходить из под рута плохо, поэтому создадим пользователя:

```
# useradd -m -G audio,cdrom,portage,usb,users,video,  
# passwd user
```

2.2.2. Gentoo. Обновление

```
# emerge --update --deep --newuse world  
# emerge --depclean  
# revdep-rebuild
```

Установим и настроим eix:

```
# emerge eix  
# eix-update
```

Установим layman:

```
# emerge layman
```

Обновим gcc:

```
# emerge -av gcc
```

Пересборка мира:

```
emerge -aveNUD @system @world
```

2.3. Управление профилями пользователей

3. Работа с командной строкой

3.1. Поиск файлов

4. Менеджер пакетов

4.1. утилита rc-update

4.2. создание RAID

4.3. Установка и удаление программ

Установить пакет в бинарном виде:

```
# emerge -avg media-video/vcdimager
```

4.4. Разрешение зависимостей

```
# equery d sys-auth/thinkfinger
```

4.5. Поиск зависимостей

4.6. Написание собственных ebuild файлов

5. Настройка

5.1. Настройка сети

5.2. Настройка авто запуска

5.3. Персональные настройки

5.3.1. Настройка /etc/inputrc

Необходимо дописать для удобного поиска:

```
"\e[A":history-search-backward
```

```
"\e[B":history-search-forward
```

5.3.2. Настройка bash

```
PS1='\[ \e[1;32m\]\u \[ \e[0;31m\] \[ \e[1;34m\]\w  
\[ \e[0;31m\] \[ \e[1;37m\]{\t}\n\$ '
```

```
PATH=$PATH:/opt/bin:/opt/sublime-text2/
```

```
#LD_LIBRARY_PATH
```

```
export ANDROID_NDK=/opt/android-ndk/
```

```
export ANDROID_STANDALONE_TOOLCHAIN=/opt/android-ndk
```

```
alias l="ls -lh"
```

```
alias ll="ls -lah"
```

```
alias m="make"
alias n="ninja"
alias sl="/opt/sublime-text2/sublime_text"
alias mm="make -j8"

alias hddfree="du -h --max-depth=1 | sort -h"

shopt -s histappend
PROMPT_COMMAND='history -a'
shopt -s cdspell

source ~/.git-completion.bash

HISTSIZE=100000
HISTFILESIZE=100000
HISTCONTROL=ignoreboth
HISTIGNORE="&:pwd:ls:ls -l:ls -la:ll:cd:cd \-:cd /:cd"

. /etc/profile
```

5.3.3. Настройка gitconfig

[user]

name = user

email = email@mail.com

[alias]

```
hist = log --pretty=format:@"%h %ad |  
%s%d [%an]\n" --graph --date=short  
st = status
```

[giggle]

```
main-window-view = HistoryView  
history-view-vpane-position = 279
```

5.3.4. Настройка Xorg.conf

```
Section "InputDevice"
```

```
Identifier "Keyboard0"
```

```
Driver "kbd"
```

```
Option "XkbModel" "pc105"
```

```
Option "XkbLayout" "us,ru"
```

```
Option "XkbVariant" ",winkeys"
```

```
Option "XkbOptions" "grp:caps_toggle,grp_led:scroll"
```

```
EndSection
```

5.3.5. Настройка mplayer2

file: /.mplayer

```
fontconfig=yes
```

```
font="Droid Sans Mono"
```

```
double=yes
```

```
cache=500000
```

```
cache-min=95
```

fs=yes

subfont-blur=1

subfont-outline=2

subfont-autoscale=3

subfont-osd-scale=1

subfont-text-scale=4

nolirc=yes

lavdopts=threads=8

osdlevel=3

spuaa=8

vf=pp=default/lowpass5/ac

af=volnorm

volstep=1

6. Мониторинг и производительность

6.1. Мониторинг

6.2. Оптимизация `exit`

6.3. SSD

6.4. Распределенная компиляция

7. Приложение

Список литературы

[1] <http://ru.wikipedia.org/wiki/Gentoo>

[2] <http://www.gentoo.org/>