

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

А.В. Переспелов, Р.А. Переспелов

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Дистрибутив операционной системы Gentoo

Лабораторный практикум

РГГМУ
Санкт-Петербург
2018

УДК 004.415(076)
ББК 32.972.11
П27

Рецензент: Тимченко В.И., к.т.н., доцент СПбГУТ им. М.А. Бонч-Бруевича.

П27 **Переспелов А.В., Переспелов Р.А.** Операционные системы: Дистрибутив операционной системы Gentoo. Лабораторный практикум. – СПб.: РГГМУ, 2018. – 68 с.

В практикум включены 10 лабораторных работ, которые выполняются с помощью дистрибутива Gentoo — одного из семейства дистрибутивов GNU/Linux. Каждая работа содержит теоретическую часть, со сведениями, необходимыми для понимания изучаемого материала, и инструкцию по выполнению заданий. Подробное описание работ позволяет использовать практикум для самостоятельного изучения базовых концепций Linux.

В методическом пособии рассмотрена система управления пакетами Portage, совмещающая в себе возможности настройки и автоматизированного управления пакетами системы с учетом конкретного аппаратного обеспечения и нужд пользователя.

Изучение дистрибутива хорошо подходит для получения практических навыков работы с открытыми операционными системами.

Практикум предназначен для студентов, обучающихся по специальности «Информационная безопасность телекоммуникационных систем».

Perespelov A.V., Perespelov R.A. Operational Systems: The Gentoo operating system distribution. The laboratory course. – St. Petersburg: RSHU Publishers, 2018. – 68 pp.

The laboratory course includes 10 labs that are executed with the Gentoo – one of the distributive family of GNU/Linux operating system. Each work contains a theoretical part with the information necessary for an understanding of the studied material, instructions for completing tasks. Detailed description of the work allows the use of a workshop for independent study of fundamental concepts of Linux.

The policy manual considered the Portage package management system, which combines configuration, combines the customization and automated package management system taking into account the specific hardware and the needs of the user.

The study of the distribution is well suited for practical skills with open operating systems.

The workshop is designed for students majoring in «Information security of telecommunication systems».

УДК 004.415(076)
ББК 32.972.11

© Переспелов А.В., Переспелов Р.А., 2018
© Российский государственный гидрометеорологический университет (РГГМУ), 2018

Предисловие

Отличительная особенность дистрибутива Gentoo состоит в том, что пакеты его системы собираются и компилируются на пользовательской машине из исходных текстов, находящихся в репозитории. Ориентация на использование текстового формата дает разработчику возможность задания уровней обеспечения безопасности, которые активны даже при наличии только текстового окружения. При этом возможна оптимизация системы под конкретное аппаратное обеспечение и нужды пользователя.

Оптимизация достигается посредством сборки программ из исходных текстов с использованием так называемых USE-флагов оптимизации и подключением–отключением необходимых модулей. Для удобства все опции могут быть внесены в конфигурационные файлы как для системы в целом (/etc/make.conf), так и для конкретных программ (/etc/portage/package.use). В случае необходимости изменения флагов, они будут учтены при следующем обновлении, и все программы, где используются эти флаги, включая все зависимости, автоматически будут пересобраны.

Дистрибутив хорошо подходит для серверов, маршрутизаторов, автоматизации производства, т.к. потребляет на 25 % меньше ресурсов, чем другие дистрибутивы Linux.

Задача разработки защищенных информационных систем требует знания и понимания принципов работы составляющих, в том числе выбранной, сконфигурированной или разработанной операционной системы. Особенно важно знать этапы разработки информационно защищенных систем специалистам в области информационных технологий.

Лабораторный практикум предназначен в первую очередь для закрепления знаний, полученных в теоретической части дисциплины, приобретения навыков конфигурирования и разработки операционных систем.

Каждая работа содержит набор заданий, которые подобраны так, чтобы за период прохождения практикума были получены практические навыки работы с операционной системой Linux.

В начале каждой работы содержатся теоретические сведения и пояснения по ее выполнению.

Порядок выполнения и оформления лабораторных работ

Отчет по лабораторной работе должен быть выполнен в электронном виде (формат .doc) с соблюдением ГОСТ, иметь титульный лист.

В верхней части титульного листа указывается название университета и кафедры. Затем название и номер лабораторной работы, номер группы и фамилия выполнившего ее студента.

В каждой лабораторной работе необходимо выполнить следующие этапы:

- сформулировать цель работы;
- выполнить задания;
- прокомментировать результаты и ответить на вопросы;
- сделать выводы по лабораторной работе.

Работа 1. Правила установки

Цель работы: проверить аппаратные характеристики системы на соответствие предъявляемым требованиям, активировать источник установки ОС.

Освоить: порядок загрузки с установочного компакт-диска, переключение с первоначальной консоли в другие, перемещение по файловой системе, абсолютные, относительные пути.

Теоретическая часть

Сборка ОС выполняется из исходных текстов (есть поддержка бинарных пакетов), настройка выполняется с помощью обычных текстовых файлов.

В процессе выполнения заданий необходимо выбрать объем самостоятельной компиляции, способ установки, службу журналирования системы ОС. Рассматривается установка ОС с установочного диска как последовательность из 10 шагов (заданий). Значительная часть этапов установки является общей для всех поддерживаемых архитектур. Однако для примера рассмотрена установка на архитектуру x-86.

Аппаратные требования, необходимые для установки Gentoo

	Минимальный CD	Live DVD
Центральный процессор	i486 или новее	i686 или новее
Память	256 МБ	512 МБ
Дисковое пространство	2,5 ГБ	2,5 ГБ
Пространство подкачки	не менее 256 МБ	не менее 256 МБ

Минимальный установочный компакт-диск — это загрузочный диск, содержащий самодостаточную среду Gentoo. Он позволяет загружать Linux прямо с компакт-диска. При запуске определяются устройства и загружаются соответствующие драйверы. Этот диск сопровождается разработчиками и позволяет установить ОС при наличии активного Интернет-соединения. Минимальный установочный диск называется `install-x86-minimal-<release>.iso` и занимает около 140 МБ. Установочный диск можно загрузить с одного из зеркал. Он расположен в каталоге `releases/x86/autobuilds/current-iso/`.

1.1. Загрузка с установочного компакт-диска

На установочном CD существует несколько ядер. По умолчанию загружается `gentoo`. Другие ядра нужны для специфической аппаратной части, и варианты `-nofb` с отключенным фреймбуфером.

Доступные варианты ядра

Ядро	Описание
<code>gentoo</code>	ядро 2.6 с многопроцессорной поддержкой
<code>gentoo-nofb</code>	аналогично <code>gentoo</code> , но без поддержки кадрового буфера
<code>memtest86</code>	для выявления ошибок локальной оперативной памяти

Можно указать параметры ядра. Они отражают необязательные настройки, которые можно включить / выключить [1]. Загрузите систему с компакт-диска, выбрав ядро и указав дополнительные параметры. В качестве примера показана загрузка ядра `gentoo` без дополнительных параметров:

```
boot: gentoo
```

Появится заставка с полосой индикатора загрузки. Установка должна происходить на машину с латинской раскладкой клавиатуры. По окончании загрузки будет выполнен автоматический вход в рабочий Gentoo Linux с правами администратора (root). В текущей консоли должно появиться приглашение root («#»). Возможно переключаться в другие консоли, нажимая Alt-F2, Alt-F3 и Alt-F4. Вернуться в первоначальную консоль можно нажатием Alt-F1.

Контрольные вопросы

- 1) Проверьте, соответствуют ли аппаратные характеристики системы требованиям по установке ОС?
- 2) Каков порядок загрузки с установочного компакт-диска?
- 3) Как переключиться с первоначальной в другие консоли?
- 4) Как можно перемещаться по файловой системе?

Работа 2. Настройка сети

Цель работы: выполнить настройку сети.

Освоить: способы настройка сети, загрузку сетевых модулей ядра, определить прокси-серверы, использование DHCP, подготовку беспроводного доступа, понятия IP-адрес, широковещательный адрес, маска сети.

Теоретическая часть

2.1. Автоматическое подключение к сети

Если система подсоединена к сети Ethernet, в которой есть сервер DHCP, вероятно, что сетевое подключение уже автоматически настроено. Тогда можно воспользоваться многими сетевыми командами, находящимися на установочном компакт-диске, например ssh, scp, ping, irssi, wget и links.

Если сеть уже настроена, команда ifconfig должна показывать сетевые интерфейсы кроме lo, например, eth0:

```
# ifconfig
(...)
eth0  Link encap:Ethernet HWaddr 00:50:BA:8F:61:7A
      inet addr:192.168.0.2 Bcast:192.168.0.255 Mask:255.255.255.0
```

```
inet6 addr: fe80::50:ba8f:617a/10 Scope:Link
UPBROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1498792 errors:0 dropped:0 overruns:0 frame:0
TX packets:1284980 errors:0 dropped:0 overruns:0 carrier:0
collisions:1984 txqueuelen:100
RX bytes:485691215 (463.1 Mb) TX bytes:123951388 (118.2 Mb)
```

```
Interrupt:11 Base address:0xe800
```

Название интерфейса в системе может отличаться от eth0. Новые установочные диски могут отображать обычные сетевые интерфейсы как eno0, ens1 или enp5s0.

Посмотрите в выводе `ifconfig` интерфейс, у которого IP-адрес относится к вашей локальной сети. Предположим, что интерфейс называется eth0.

2.2. Указание прокси-серверов

Если компьютер подключен к интернету через прокси-сервер, при установке может потребоваться ввести сведения о нем. Чтобы задать прокси-сервер нужно определить переменные, содержащие необходимые сведения.

В большинстве случаев в определении переменных достаточно указать имя прокси-сервера. Предположим, что прокси-сервер называется `proxu.gentoo.org`, а его порт — 8080:

```
(если прокси-сервер фильтрует трафик HTTP)
# export http_proxu="http://proxu.gentoo.org:8080"
(если прокси-сервер фильтрует трафик FTP)
# export ftp_proxu="ftp://proxu.gentoo.org:8080"
(если прокси-сервер фильтрует трафик RSYNC)
# export RSYNC_PROXY="proxu.gentoo.org:8080"
```

Если прокси-сервер запрашивает имя пользователя и пароль, для значения переменной следует использовать такой формат:

```
http://имя_пользователя:пароль@proxu.gentoo.org:8080
```

2.3. Проверка сети

Полезно проверить отклик DNS-сервера поставщика услуг интернета (адрес сервера находится в файле `/etc/resolv.conf`), а также

произвольного веб-сайта, чтобы убедиться, что пакеты выходят в интернет, разрешение имен DNS работает.

```
# ping — с 3 www.gentoo.org
```

Если сеть доступна, то достаточно освоить теоретическую часть работы, если нет, то нужно выполнить следующий раздел.

2.4. Автоматизированная настройка сети

Простейший способ настройки сети, если она не произошла автоматически — запуск сценария `net-setup`:

```
# net-setup eth0
```

`net-setup` в диалоговом режиме задаст несколько вопросов о вашей сетевой среде. В результате его работы должно появиться работоспособное подключение к сети. Проверьте сетевое подключение, как это описано выше.

2.5. Ручная настройка сети

При загрузке установочный диск выявляет все установленные устройства и загружает подходящие модули ядра (драйверы) для поддержки оборудования. В некоторых случаях эта операция не приводит к автозагрузке нужных модулей ядра.

Если `net-setup` или `pppoe-setup` не удалось загрузить нужный модуль, возможно, сетевая плата сразу не обнаружена. При этом может потребоваться ручная загрузка необходимых модулей ядра. Чтобы выяснить, какие модули ядра для поддержки сети существуют, используйте `ls`:

```
# ls /lib/modules/`uname -r`/kernel/drivers/net
```

Если драйвер для сетевой платы найден, то для загрузки модуля ядра используйте `modprobe` (для примера загрузим модуль `pcnet32`):

```
# modprobe pcnet32
```

Чтобы убедиться, что сетевая плата теперь обнаружена, используйте `ifconfig`. Если сетевая плата обнаружена, результат выглядит подобным образом (`eth0` здесь только в качестве примера):

```
# ifconfig eth0.
```

```
eth0    Link encap:Ethernet HWaddr FE:FD:00:00:00:00
        BROADCAST NOARP MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0

        RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
```

Если получено сообщение об ошибке, сетевая плата не обнаружена:

```
# ifconfig eth0
```

```
eth0: error fetching interface information: Device not found
```

Список имен существующих сетевых интерфейсов в системе можно получить через файловую систему /sys:

```
# ls /sys/class/net
```

```
dummy0 eth0 lo sit0 tap0 wlan0
```

В приведенном примере было найдено 6 интерфейсов. Вероятно, eth0 является (проводным) адаптером Ethernet, в то время как wlan0 это беспроводной адаптер.

2.6. Использование DHCP

DHCP (Dynamic Host Configuration Protocol — протокол динамической настройки хоста) дает возможность автоматически получить параметры сетевого подключения (IP-адрес, маску сети, широковещательный адрес, шлюз, сервера имен). Все это работает, только когда в сети есть сервер DHCP. Чтобы сетевой интерфейс получал эти сведения автоматически, используйте dhcpcd:

```
# dhcpcd eth0
```

Если имя хоста и домена, назначаются сервером DHCP, в этом случае используется:

```
# dhcpcd -HD eth0
```

2.7. Сетевая терминология

Рассмотрим, что такое шлюз, зачем служит маска сети, как формируется широковещательный адрес, зачем нужны серверы имен. Это потребуется для удовлетворительной настройки сети.

В сети узлы (хосты, компьютеры) различаются по IP-адресу (адресу протокола интернета). Такой адрес — это сочетание четырех чисел от 0 до 255. В действительности, IP-адрес состоит из 32 бит (единиц и нулей). Рассмотрим пример:

IP-адрес (числа):	192.168.0.2			
IP-адрес (биты):	11000000	10101000	00000000	00000010
	<hr/>	<hr/>	<hr/>	<hr/>
	192	168	0	2

Такой IP-адрес уникален для узла в рамках всех доступных сетей (т. е. каждый узел, с которым можно связаться, должен иметь уникальный IP-адрес). Чтобы различать узлы, находящиеся внутри и извне сети, IP-адрес подразделяется на две части: часть сети и часть узла. Это разделение записывается с помощью маски сети — набора единиц, за которым следует набор нулей. Часть IP-адреса, которая попадает на единицы — сетевая, оставшаяся часть — узловая. Как обычно, маска сети может записываться в виде IP-адреса.

Пример разделения сети / узла:

IP-адрес:	192	168	0	2
	11000000	10101000	00000000	00000010
Маска сети:	11111111	11111111	11111111	00000000
	255	255	255	0
	+-----+			+
	Сеть			Узел

Адрес 192.168.0.14 — все еще входит в состав сети из нашего примера, а 192.168.1.2 — уже нет.

Широковещательный адрес — это IP-адрес с такой же сетевой частью, как у вашей сети, но у которого узловая часть состоит только из единиц. Каждый узел вашей сети слушает этот IP-адрес. Он действительно предназначен для широковещательной рассылки пакетов.

IP-адрес:	192	168	0	2
	11000000	10101000	00000000	00000010
Широковещательный адрес:	11000000	10101000	00000000	11111111
	192	168	0	255
	+-----+-----+			
	Сеть			Узел

Для нормальной работы в интернете нужно знать, через какой узел происходит подключение к интернету. Этот узел называется шлюзом. Так как это обычный узел, ему присвоен обычный IP-адрес (например, 192.168.0.1).

Каждому узлу присваивается свой собственный IP-адрес. Чтобы связываться с узлом по имени (вместо IP-адреса), нужна служба, которая переводит имя (как, например, dev.gentoo.org) в IP-адрес (например, 64.5.62.82). Такая служба называется службой имен. Чтобы пользоваться ей, нужно указать необходимые серверы имен в `/etc/resolv.conf`.

Иногда шлюз сам является сервером имен. В остальных случаях необходимо указывать серверы имен, предоставляемые поставщиком интернета.

Далее потребуется следующая информация:

Параметр настройки сети	Пример
Ваш IP-адрес	192.168.0.2
Маска сети	255.255.255.0
Широковещательный адрес	192.168.0.255
Шлюз	192.168.0.1
Сервер(ы) имен	195.130.130.5, 195.130.130.133

2.8. Использование `ifconfig` и `route`

Настройка сети состоит из трех шагов.

1. Сначала назначаем IP-адрес с помощью `ifconfig`.
2. Затем настроим маршрутизацию к шлюзу, пользуясь `route`.
3. И в завершение помещаем IP-адреса серверов имен в `/etc/resolv.conf`.

Для назначения IP-адреса потребуется IP-адрес, широковещательный адрес и маска сети. Выполните следующую команду, заменив:

`{IP_ADDR}` на свой IP-адрес,
`{BROADCAST}` на свой широковещательный адрес,
`{NETMASK}` на свою маску сети.

Использование `ifconfig`:

```
# ifconfig eth0 {IP_ADDR} broadcast {BROADCAST} netmask  
{NETMASK} up.
```

Теперь настройте маршрутизацию с помощью `route`.

Подставьте IP-адрес своего шлюза вместо `{GATEWAY}`:

Использование `route`:

```
# route add default gw {GATEWAY}.
```

Затем откройте `/etc/resolv.conf` в своем любимом редакторе
(в примере используется `nano`).

Создание `/etc/resolv.conf`:

```
# nano - w /etc/resolv.conf
```

Заполните данные серверов имен по следующему образцу. Замените `{NAMESERVER1}` и `{NAMESERVER2}` на соответствующие адреса серверов имен:

```
nameserver {NAMESERVER1}  
nameserver {NAMESERVER2}
```

Проверьте свою сеть с помощью команды `ping`.

Контрольные вопросы

- 1) Как производится загрузка сетевых модулей ядра?
- 2) Что такое прокси-сервер?
- 3) Как используется DHCP?
- 4) Как происходит подготовка беспроводного доступа в сеть?
- 5) Что такое IP-адрес?
- 6) Что такое широковещательный адрес?
- 7) Что такое маска сети?

Работа 3. Подготовка дисков

Цель работы: рассмотреть дисковую архитектуру системы, разработать схему разбиения диска, создать разделы.

Освоить: порядок схемы разбиения дисков, способы удаления, создания разделов, файловых систем на созданных разделах, подключение к системе созданных разделов.

Теоретическая часть

3.1. Блочные устройства

Рассмотрим дисковую архитектуру операционной системы, включая файловые системы, разделы дисков и блочные устройства. Типичным блочным устройством является то, которое представлено первым жестким диском SCSI в операционной системе, а именно:

```
/dev/sda
```

Устройства SCSI и Serial ATA обозначаются как:

```
/dev/sd*
```

Даже IDE-устройства в новом фреймворке libata обозначаются как:

```
/dev/sd*
```

Если используется старый фреймворк, первое IDE-устройство будет:

```
/dev/hda
```

Блочные устройства предоставляют абстрактный интерфейс к диску. Пользовательские программы могут использовать эти блочные устройства для взаимодействия с диском, абстрагируясь от типа диска (IDE, SCSI). Программа просто адресует дисковое пространство как набор непрерывных 512-байтных блоков с произвольным доступом.

3.2. Разделы дисков

Для размещения ОС диск можно использовать целиком, на практике так почти никогда не делают. Вместо этого большое физическое

блочное устройство разбивают на меньшие блочные устройства, более удобные для обращения. В архитектуре x86 они называются разделами. На данный момент существуют две стандартные технологии разбиения на разделы MBR и GPT.

MBR (Master Boot Record) использует 32-битные идентификаторы для начального сектора и длины разделов и поддерживают три типа разделов:

- первичные (primary),
- расширенные (extended),
- логические (logical).

Первичные разделы хранят информацию о себе в самом MBR — очень маленьком кусочке пространства (обычно 512 байт) в самом начале диска. Из-за такого маленького объема поддерживаются только четыре первичных раздела: от `/dev/sda1` до `/dev/sda4`. Для поддержки большего числа разделов один из первичных разделов может быть помечен как расширенный раздел. Данный раздел может затем содержать логические разделы.

Каждый раздел имеет ограничение на размер в 2 Тб (из-за 32-битных идентификаторов). MBR не имеет резервного MBR, т. е. если приложение перезапишет MBR, то вся информация о разделах будет потеряна.

GPT (GUID Partition table) использует 64-битные идентификаторы для разделов. Часть пространства, в котором сохраняется информация о разделах больше, чем 512 байт в случае MBR. Размер раздела имеет ограничение, но гораздо большего объема (почти 8 Зб — Зеттабайт).

Если между операционной системой и оборудованием связь осуществляется через UEFI (а не BIOS), GPT является практически обязательной, так как в данном случае возникнут проблемы совместимости с MBR. GPT имеет преимущество — в конце диска имеется копия GPT, которая может использоваться для восстановления повреждения основной GPT в начале диска. В GPT хранятся контрольные суммы по алгоритму CRC32 для обнаружения ошибок в заголовке и таблицах разделов.

3.3. Выбор GPT или MBR

Существует несколько ограничений. Использование GPT при наличии BIOS в компьютере работает, но при этом нельзя иметь второй системой Microsoft Windows. Причина в том, что Microsoft

Windows попытается загрузиться в режиме EFI, если увидит метку раздела GPT.

Включение флага загрузки в разделе MBR:

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

```
WARNING: GPT (GUID Partition Table) detected on '/dev/sda'!  
The util fdisk doesn't support GPT. Use GNU Parted.
```

```
Command (m for help): a
```

```
Partition number (1-4): 1
```

```
Command (m for help): w
```

3.4. Разработка схемы разбиения диска

Схема разбиения по умолчанию

Раздел	Файловая система	Размер	Описание
/dev/sda1	(начальный загрузчик)	2 МБ	загрузочный раздел BIOS
/dev/sda2	ext2	128 МБ	загрузочный раздел
/dev/sda3	(swap)	512 МБ или больше	раздел подкачки
/dev/sda4	ext4	оставшаяся часть диска	корневой раздел

Рекомендуется использовать `fdisk` для создания разделов.

Обе программы являются инструментами для разбивки дисков: `fdisk` известен и стабилен, `parted` новее и поддерживает разделы более чем 2 Тб.

3.5. Выбор параметров

Количество разделов очень сильно зависит от назначения системы. Например, если в системе много пользователей, нужно выделить разделы:

1. `/home` — для повышения безопасности и упрощения резервного копирования;

2. `/var` — если устанавливаете Gentoo в роли почтового сервера, т. к. вся почта хранится там;

3. `/opt` — для игровых серверов, так как большинство программ для их работы устанавливается туда;

4. `/usr` понадобится, потому что там хранится большинство приложений, а также из-за того, что дерево Portage, не считая размещенных в нем архивов с исходными кодами, занимает около 500 МБ.

Наличие отдельных разделов или томов имеет следующие плюсы:

- для каждого раздела или тома можно выбрать наиболее подходящую файловую систему;
- свободное место во всей системе не кончится внезапно из-за того, что сбойная программа постоянно записывает файлы в раздел или том;
- необходимая проверка файловых систем будет занимать меньше времени, т. к. проверка разных разделов может выполняться параллельно;
- можно повысить безопасность системы, монтируя часть разделов в режиме `read-only` (только для чтения), `nosuid` (игнорируется бит `setuid`), `noexec` (игнорируется бит исполнения) и т. д.

У множества разделов также есть недостатки. Если они не сконфигурированы правильно, будет система с большим количеством свободного места на одном разделе и отсутствием его на другой. Также существует лимит в 15 разделов для SCSI и SATA, если не используется метки GPT.

3.6. Пространство подкачки

Целью пространства подкачки является предоставление дискового пространства ядру, когда оперативная память (ОЗУ) активно используется. Пространство подкачки позволяет ядру переносить страницы памяти, которые не будут использоваться в ближайшее время, на диск (что называется `своп` или `page-out`), освобождая память. Если эта память неожиданно понадобится, выбранные страницы должны быть помещены обратно в память (`page-in`), что займет некоторое время (так как диски — это очень медленные устройства, если сравнивать их с оперативной памятью).

Если не нужно запускать приложения, требовательные к памяти, либо в системе много памяти, то не требуется много пространства подкачки. Однако раздел подкачки также используется для сохранения всей памяти в случае гибернации. Если вы планируется использовать гибернацию, будет нужно большее пространство подкачки, минимум равное размеру оперативной памяти.

3.7. Загрузочный раздел BIOS

Загрузочный раздел BIOS обычно маленький (от 1 до 2 Мб). В нем начальные загрузчики типа GRUB2 могут сохранять дополнительные данные, если они не помещаются в выделенное пространство (в случае MBR это несколько сот байт), и их нельзя разместить ни в каком другом месте. Такие разделы не всегда необходимы. Но они занимают мало места и, принимая во внимание сложность документирования множества вариантов, когда они нужны, а когда нет, рекомендуется создать такой раздел в любом случае.

Загрузочный раздел BIOS необходим при использовании GPT с GRUB2, либо при использовании MBR с GRUB2, если первый раздел начинается ранее, чем 1 мегабайт от начала диска.

3.8. Использование parted для создания разделов

Примерная компоновка разделов, которая будет использоваться, показана в таблице:

Раздел	Описание
/dev/sda1	Загрузочный раздел BIOS
/dev/sda2	Загрузочный раздел
/dev/sda3	Swap раздел
/dev/sda4	Корневой раздел

Компоновка разделов выбирается самостоятельно.

3.8.1. Просмотр текущей компоновки разделов

Приложение parted предлагает интерфейс для разбивки дисков. Запустите parted с загрузочного диска (в примере мы используем /dev/sda).

```
# parted -a optimal /dev/sda
```

```
GNU Parted 2.3
```

```
Using /dev/sda
```

```
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

Выравнивание означает, что разделы будут начинаться на известных границах на диске, что приведет к тому, что операции с диском на уровне операционной системы (получение страниц с диска)

будут использовать самое маленькое число внутренних операций с диском. Не выровненные разделы могут потребовать от диска загрузить две страницы вместо одной — даже если операционная система потребовала всего одну страницу.

Все опции, поддерживаемые parted, можно найти с помощью команды help.

3.8.2. Установка метки GPT

Большинство дисков на x86/amd64 используют метку msdos. Используя parted, можно разместить на диске метку GPT с помощью команды:

```
mklabel gpt:
```

Изменение типа раздела удалит все разделы с вашего диска. Все данные на диске будут потеряны.

Установка метки GPT:

```
(parted) mklabel gpt
```

Если нужно, чтоб диск имел MBR, используется команда:

```
mklabel msdos
```

3.8.3. Удаление разделов

Сначала нужно удалить все существующие разделы с диска. Нужно ввести команду print для просмотра текущих разделов и rm <номер>, где <номер> — это раздел, который нужно удалить. Например команды:

```
(parted) rm 2
```

Повторить те же действия для всех остальных разделов, которые не нужны. Удостоверьтесь, что не допустили никаких ошибок — parted запускает изменения сразу (в отличие от fdisk, который накапливает их, позволяя пользователю отменить свои изменения, прежде чем записать их или выйти из fdisk).

3.8.4. Создание разделов

Нужно установить следующие настройки parted.

1) Тип раздела для использования. Обычно это `primary`. Если использована метка раздела `msdos`, то нельзя создать более четырех первичных разделов. Если нужно больше четырех первичных разделов, создайте расширенный (`extended`) раздел, и создайте в нем логические разделы (`logical`).

2) Начальное положение раздела (которое может быть выражено в Мб, Гб).

3) Конечное положение раздела (которое может быть выражено в Мб, Гб).

Настраиваем `parted` так, чтоб использовать единицу измерения мегабайт. В тексте будем использовать МБ, как это гораздо более распространено. Но на самом деле «стандартной» формой записи являются мебибайты, которые обозначаются как МиБ:

```
(parted) unit mib
```

Далее создадим раздел размером в 2 Мб, который будет использоваться начальным загрузчиком GRUB. Используем для этого команду `mkpart`, и настроим `parted` так, чтобы он создал раздел от 1 Мб до 3 Мб (он получится размером в 2 Мб).

```
(parted) mkpart primary 1 3
(parted) name 1 grub
(parted) set 1 bios_grub on
(parted) print
```

```
Model: Virtio Block Device (virtblk)
Disk /dev/sda: 20480MiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

```
Number Start End Size File system Name Flags
```

```
1 1.00MiB 3.00MiB 2.00MiB grub bios_grub
```

Нужно повторить данные шаги для загрузочного раздела (128 Мб), раздела подкачки (в примере 512 Мб) и корневого раздела, который занимает оставшуюся часть диска (для этого его конечное положение обозначено как `-1`, что означает «конец диска минус один мегабайт»).

```
(parted) mkpart primary 3 131
(parted) name 2 boot
(parted) mkpart primary 131 643
(parted) name 3 swap
(parted) mkpart primary 643 -1
(parted) name 4 rootfs
```

Конечный результат будет выглядеть примерно так:

```
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/sda: 20480MiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

Number	Start	End	Size	File system	Name	Flags
1	1.00MiB	3.00MiB	2.00MiB		grub bios_grub	
2	3.00MiB	131MiB	128MiB		boot	
3	131MiB	643MiB	512MiB		swap	
4	643MiB	20479MiB	19836MiB		rootfs	

3.8.5. Текущие параметры разделов

```
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/sda: 20480MiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

Number	Start	End	Size	File system	Name	Flags
1	1.00MiB	3.00MiB	2.00MiB		grub bios_grub	
2	3.00MiB	131MiB	128MiB		boot	
3	131MiB	643MiB	512MiB		swap	
4	643MiB	20479MiB	19836MiB		rootfs	

Для выхода из parted используется команда quit.

3.9. Использование fdisk для создания разделов

fdisk — это утилита для создания разделов на дисках. Запуск fdisk производится с указанием параметра идентификатор диска (в примере используется /dev/sda):

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

После запуска fdisk выдаст приглашение:

```
Command (m for help):
```

Нажав «р», выведем текущую схему разбиения диска:

```
Command (m for help): p
```

```
Disk /dev/sda: 240 heads, 63 sectors, 2184 cylinders  
Units = cylinders of 15120 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	14	105808+	83	Linux
/dev/sda2		15	49	264600	82	Linux swap
/dev/sda3		50	70	158760	83	Linux
/dev/sda4		71	2184	15981840	5	Extended
/dev/sda5		71	209	1050808+	83	Linux
/dev/sda6		210	348	1050808+	83	Linux
/dev/sda7		349	626	2101648+	83	Linux
/dev/sda8		627	904	2101648+	83	Linux
/dev/sda9		905	2184	9676768+	83	Linux

```
Command (m for help):
```

В данном случае на диске есть семь разделов Linux (для которых в графе «System» указано «Linux») и один раздел подкачки (в списке показан как «Linux swap»).

3.9.1. Удаление всех разделов

Для удаления раздела вводите «d». Например, для удаления существующего:

```
/dev/sda1: Command (m for help): d
```

```
Partition number (1-4): 1.
```

Удаление раздела будет запланировано. Он больше не будет отображаться при вводе «р», но фактически не удалится до тех пор, пока не сохраните внесенные изменения. Если произошла ошибка и

нужно прервать разбиение без сохранения изменений, введите «q» и нажмите Enter — тогда раздел сохранится в неизменном виде.

Если предполагается удалить в своей системе все разделы, нужно повторять ввод «r» для вывода информации об оставшихся разделах, затем «d» и номера удаляемого раздела до тех пор, пока разделы не кончатся. В итоге получим пустую таблицу разделов. Например:

```
Disk /dev/sda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes
```

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

Command (m for help):

После очистки таблицы разделов, хранящейся в оперативной памяти, нужно создавать разделы. Примерная схема разбиения, описана выше.

3.9.2. Создание загрузочного раздела BIOS

Введите «n» для создания нового раздела, затем «r», чтобы выбрать первичный раздел, и «1» для указания первого первичного раздела. На появившийся вопрос о первом секторе, проверьте, что он начинается с 2048 (это необходимо для начального загрузчика), и нажмите ввод. На вопрос о последнем секторе введите «+2M», чтобы создать раздел размером 2 МБ.

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 1

First sector (64-10486533532, default 64): 2048

Last sector, +sectors +size {M,K,G} (4096-10486533532, default 10486533532): +2M

Пометим раздел для целей EFI:

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): ef

Changed system type of partition 1 to ef (EFI (FAT-12/16/32))

3.9.3. Создание загрузочного раздела

Нажмите «n» для создания нового раздела, затем «р» для выбора первичного раздела, затем «2» для выбора второго первичного раздела. Когда утилита попросит ввести первый сектор, примите значение по умолчанию, нажав Enter. Когда утилита предложит ввести последний сектор, введите «+128M» для создания раздела размером 128 Мб.

Command (m for help): n
Command action
e extended
p primary partition (1-4)

p
Partition number (1-4): 2
First sector (5198-10486533532, default 5198): (Hit enter)

Last sector, +sectors +size {M,K,G} (4096-10486533532, default 10486533532): +128M.

Теперь после ввода «р» увидим следующий текст:

Command (m for help): p

Disk /dev/sda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		1	3	5198+	ef	EFI(FAT-12/16/32)
/dev/sda2		3	14	105808+	83	Linux

Необходимо сделать этот раздел загружаемым. Введите «a» для включения признака загрузки с раздела, затем нажмите «2». Если снова ввести «р», можно увидеть, что в столбце «boot» появился знак «*».

3.9.4. Создание раздела подкачки

Введите «n» (создание нового раздела), затем «p» для выбора первичного раздела. Потом нажмите «3», чтобы создать третий первичный раздел, в нашем случае:

```
/dev/sda3
```

На вопрос о первом секторе просто нажмите Enter. На вопрос о последнем — ответьте «+512M», чтобы создать раздел размером 512 МБ.

Сделав это, введите «t» для указания типа раздела, «3» — для выбора только что созданного раздела, и «82», чтобы установить тип раздела «Linux Swap».

3.9.5. Создание корневого раздела

Введите «n» (создание нового раздела), затем «p» (первичный раздел). После этого нажмите «4» для создания четвертого первичного раздела, в нашем случае:

```
/dev/sda4
```

На вопрос о первом секторе нажмите ввод. На вопрос о последнем разделе также нажмите Enter, чтобы раздел занял все оставшееся свободное место на диске. По завершении этих шагов, при вводе «p» должна выводиться подобная таблица разделов:

```
Command (m for help): p
```

```
Disk /dev/sda: 30.0 GB, 30005821440 bytes  
240 heads, 63 sectors/track, 3876 cylinders  
Units = cylinders of 15120 * 512 = 7741440 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		1	3	5198+	ef	EFI(FAT-12/16/32)
/dev/sda2	*	3	14	105808+	83	Linux
/dev/sda3		15	81	506520	82	Linux swap
/dev/sda4		82	3876	28690200	83	Linux

3.9.6. Сохранение схемы разбиения

Для сохранения схемы разбиения и выхода из fdisk, введите «w»:

```
Command (m for help): w
```

3.9.7. Создание файловых систем

Ядро Linux поддерживает различные файловые системы.

ext2 — это старая файловая система Linux. В ней нет журналирования, что означает, что проверки файловой системы ext2 при загрузке системы могут быть достаточно долгими. В настоящее время есть большой выбор более новых журналируемых файловых систем, которые можно проверить очень быстро, поэтому их предпочитают перед их нежурналируемыми аналогами. Журналируемые файловые системы позволяют избежать долгого ожидания при загрузке системы, когда файловая система находится в ошибочном состоянии.

ext3 — это журналируемая версия файловой системы ext2. Она имеет возможность журналирования метаданных для быстрого восстановления, а также дополнительные режимы журналирования, такие как полное журналирование данных и журналирование упорядоченных данных. Она использует индекс NTree, который дает высокую производительность почти во всех ситуациях. ext3 — это надежная файловая система.

ext4 — это файловая система, созданная как ветка ext3. Она имеет новые возможности, улучшения скорости, а также не имеет лимита на размеры, с не очень большими изменениями формата на диске. Она может работать с дисками до 1 Эб, и максимальным размером файла 16 Тб. Вместо классической карты выделения блоков, как в ext2/3, в ext4 используются экстенты, которые улучшают производительность на больших файлах и уменьшают фрагментацию. Кроме того, ext4 также имеет более сложные алгоритмы выделения блока (отложенное выделение и многоблоковое выделение), что дает драйверу файловой системы больше возможностей для оптимизации распределения данных на диске. ext4 — это рекомендованная всеплатформенная файловая система.

Если устанавливаете Gentoo на очень маленький раздел (менее 8 Гб), тогда нужно указать ext2, ext3 либо ext4 (если такие файловые системы есть) зарезервировать достаточно инодов, при создании файловой системы.

JFS — это высокопроизводительная журналируемая файловая система от IBM. JFS — маленькая, быстрая и надежная файловая система, основанная на B-tree, с хорошей производительностью в различных условиях.

ReiserFS — это основанная на B-tree журналируемая файловая система с хорошей производительностью, особенно, когда происходит работа со множеством маленьких файлов за счет большей

нагрузки на ЦП. ReiserFS хуже поддерживается, чем другие файловые системы.

XFS — это файловая система с журналированием метаданных, которая имеет удобный набор возможностей и оптимизирована для масштабирования. XFS имеет меньше возможностей по восстановлению после различных проблем с аппаратной частью системы.

3.9.8. Размещение файловой системы в разделе

Для создания в разделе или томе файловой системы каждого типа существуют специальные средства:

Файловая система	Команда создания
ext2	mkfs.ext2
ext3	mkfs.ext3
ext4	mkfs.ext4
reiserfs	mkreiserfs
xfs	mkfs.xfs
jfs	mkfs.jfs

Например, чтобы у загрузочного раздела (/dev/sda2 в наших примерах) была файловая система ext2, а у корневого раздела (/dev/sda4 в наших примерах) — ext4, требуется выполнить:

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

Создайте файловые системы на созданных разделах (логических томах).

3.9.9. Подключение раздела подкачки

mkswap — команда, используемая для инициализации разделов подкачки:

```
# mkswap /dev/sda3
```

Для подключения раздела подкачки воспользуйтесь swapon:

```
# swapon /dev/sda3
```

Далее командами, приведенными выше, нужно создать и подключить раздел подкачки в своей системе.

3.9.10. Монтирование разделов

После создания разделов и размещения файловых систем, нужно смонтировать (подключить к системе) эти разделы. Используется команда `mount`. Для каждого раздела нужно создать каталоги монтирования. Например, смонтируем корневой и загрузочный разделы:

```
# mount /dev/sda4 /mnt/gentoo
# mkdir /mnt/gentoo/boot
# mount /dev/sda2 /mnt/gentoo/boot
```

Еще нам потребуется смонтировать файловую систему `proc` (виртуальный интерфейс ядра) в каталог `/proc`. Но сначала надо поместить в разделы нужные файлы.

Контрольные вопросы

- 1) Что такое разделы диска, и какие типы у них бывают?
- 2) Сравните программы разбиения дисков `parted` и `fdisk`?
- 3) Зачем нужен раздел подкачки, как его создать?
- 4) Нужен ли загрузочный раздел BIOS, как его создать?
- 5) Прокомментируйте текущие параметры разделов.
- 6) Как сохранить схему разбиения?
- 7) Какие файловые системы поддерживает ядро ОС. Их особенности?
- 8) Как разместить файловые системы в разделах?
- 9) Как подключить раздел подкачки?
- 10) Что такое монтирование разделов, как это происходит?

Работа 4.

Конфигурация установочных файлов

Цель работы: установить текущую дату и время, выбрать источник загрузки архива стадии, распаковать архив стадии, подготовить систему к установке.

Освоить: загрузку архива стадии, проверить целостность архива с помощью вычисления контрольной суммы, настроить параметры компиляции.

Теоретическая часть

4.1. Установка текущей даты и времени

Для проверки времени, нужно ввести команду:

```
# date
```

Получим данные о текущем времени, например:

```
Fri Mar 29 16:21:18 UTC 2005
```

Настройка даты и времени выполняется командой:

```
date ММДДчммГГГГГ  
(формат: Месяц, День, часы, минуты и Год)
```

На этом этапе следует указывать время UTC.

Позднее можно установить свой часовой пояс. Для установки часов на 29 марта, 16:21, 2005 год, выполните:

```
# date 032916212005
```

4.2. Выбор источника установки системы

Установка архива stage3 на систему. Нужно использовать команду `uname -m`, чтобы определить, какой файл стадии нужно скопировать, так как она дает информацию об архитектуре системы.

4.2.1. Загрузка архива стадии

Перейдите в точку подключения файловой системы (обычно — `/mnt/gentoo`):

```
# cd /mnt/gentoo
```

Для загрузки файла стадии можно использовать программу `links`, далее открыть список зеркал Gentoo и выбрать ближайшее из них:

```
links http://www.gentoo.org/main/en/mirrors.xml
```

и нажать Enter.

Когда links нет, в вашем распоряжении должен оказаться lynx. Если требуется использовать прокси-сервер, экспортируйте переменные http_proxy и ftp_proxy:

```
# export http_proxy="http://proxy.server.com:port"  
# export ftp_proxy=http://proxy.server.com:port
```

Предположим, установлена links. Выберите ближайшее зеркало. Перейдите в каталог:

```
releases/x86/autobuilds/
```

Там можно увидеть все существующие файлы стадий для вашей архитектуры (они могут располагаться в подкаталогах с названиями разновидностей архитектуры). Выберите один из них и нажмите «D» для загрузки. По окончании — нажмите «Q», чтобы выйти из браузера.

Большинство пользователей PC должны использовать архив третьей стадии:

```
stage3-i686-<release>.tar.bz2
```

Все современные PC считаются i686. Если используется старая машина, можно проверить список i686-совместимых процессоров. Старые процессоры, типа Pentium, K5, K6 или VIA C3, требуют общую третью стадию x86. Процессоры более старые, чем i486, не поддерживаются.

```
# links http://www.gentoo.org/main/ru/mirrors.xml  
(если в links нужна поддержка прокси-сервера)
```

```
# links -http-proxy proxy.server.com:8080  
http://www.gentoo.org/main/ru/mirrors.xml
```

Если нужно проверить целостность загруженного архива стадии, используется утилита openssl и результат сравнивается с контрольными суммами, взятыми с зеркала. Файлы с расширением .DIGEST дают несколько контрольных сумм, рассчитанных по различным алгоритмам. Рекомендуемыми являются SHA512 и Whirlpool.

```
## Вычисляем контрольную сумму SHA512  
# openssl dgst -r -sha512 stage3-i686-<release>.tar.bz2
```

или

```
# sha512sum stage3-i686-<release>.tar.bz2
## Вычисляем контрольную сумму Whirlpool
# openssl dgst -r -whirlpool stage3-i686-<release>.tar.bz2
```

Далее сравнивается вывод данных команд со значением, записанным в файлах `.DIGEST(.asc)`, которые также можно найти на зеркалах. Значения должны совпадать, иначе скопированный файл может быть испорчен. Можно проверить криптографическую сигнатуру файла `.DIGESTS.asc` используя `gpg`, чтобы быть уверенным, что контрольные суммы не менялись:

```
# gpg --verify stage3-i686-<release>.tar.bz2.DIGESTS.asc
```

4.2.2. Распаковка архива стадии

Распакуйте загруженный архив стадии в своей системе. Обычно для этого используется программа `tar`:

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

Убедитесь, что используются такие же параметры командной строки (`xvjpf`). Значения параметров: `x` — извлечение, `v` — подробные сообщения, чтобы видеть, что происходит во время распаковки (необязательный параметр), `j` — декомпрессия `bzip2`, `p` — сохранение прав доступа, и, наконец, `f` — указывает на то, что мы распаковываем файл, а не то, что подается на стандартный ввод.

4.3. Настройка параметров компиляции

Для оптимизации ОС можно создать ряд переменных, которые повлияют на поведение Portage. Их можно создавать как переменные среды (с помощью `export`), но тогда они будут лишь временными. Для хранения настроек предназначен конфигурационный файл Portage:

```
/etc/portage/make.conf
```

Отредактируем его. Список и описание всех допустимых переменных находятся в файле:

```
/mnt/gentoo/usr/share/portage/config/make.conf.example
```

Для успешной установки Gentoo, достаточно установить значения только тех переменных, которые описаны ниже. Запустите редактор nano, для изменения параметров оптимизации:

```
# nano -w /mnt/gentoo/etc/portage/make.conf
```

У файла make.conf.example обычная структура: строки комментария начинаются со знака «#», а в остальных строках выражениями вида «ПЕРЕМЕННАЯ=значение» определяются переменные. У файла make.conf такой же формат.

4.3.1. Переменные CFLAGS и CXXFLAGS

Переменные CFLAGS и CXXFLAGS определяют параметры оптимизации компилятора gcc для языков C и C++, соответственно. Общие значения следует устанавливать здесь, максимальная производительность достигается в том случае, когда для каждой компилируемой программы устанавливаются свои собственные оптимальные параметры. Причина в том, что все программы различны.

В файле make.conf следует указывать параметры оптимизации, которые повысят скорость системы в целом. Сюда не помещают экспериментальные значения. Прочитайте файл make.conf.example: в нем есть примеры и полезные сведения.

Первый параметр — флаг «-march=» или флаг «-mtune=», где указывается название целевой архитектуры. Возможные варианты описаны в комментариях в файле make.conf.example. Часто используется значение native, которое сообщает компилятору, чтобы он выбрал архитектуру текущей системы.

Следующий — флаг «-O» (заглавная латинская «O»), определяющий класс оптимизации в gcc. Допустимые значения:

s — оптимизация по размеру;

0 — ноль, без оптимизации;

1, 2 или даже 3 — все большая оптимизация по скорости (в каждый класс входят все флаги предыдущего, и некоторые дополнительные). -O2 является рекомендуемым значением по умолчанию.

-O3 при использовании для всей системы, иногда вызывает проблемы.

Другой популярный флаг оптимизации — «-pipe» (использование для связи между различными проходами компилятора каналов вместо временных файлов). Этот флаг не влияет на получаемый код, но использует больше памяти. На системах с малым объемом

памяти gcc может аварийно завершиться. В этом случае не используйте данный флаг.

Использование «-fomit-frame-pointer» (не хранить в регистре указатель стекового кадра для функций, которым он не нужен) может всерьез помешать отладке программ.

При определении переменных CFLAGS и CXXFLAGS нужно объединять несколько флагов оптимизации. Значения по умолчанию, содержащиеся в третьей стадии, должны быть приемлемы. Следующие значения приведены для примера:

```
CFLAGS=»-O2 -march=i686 -pipe»
```

```
# указывайте одинаковые значения обеих переменных
```

```
CXXFLAGS=»${CFLAGS}»
```

4.3.2. Переменная MAKEOPTS

С помощью MAKEOPTS определяется, сколько параллельных процессов компиляции можно запускать при установке пакета. Хорошим вариантом является значение, равное количеству процессоров (или ядер) в системе плюс один.

```
MAKEOPTS=»-j2»
```

4.3.3. Подготовка к установке системы

Измените:

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

в соответствии с пожеланиями, и сохраните изменения (пользователям nano нужно нажать комбинацию клавиш Ctrl + X.

Теперь все готово к установке базовой системы Gentoo.

Контрольные вопросы

- 1) Как установить текущую дату и время?
- 2) Как выбрать источник загрузки архива стадии?
- 3) Как проверить целостность архива стадии?
- 4) Как настроить параметры компиляции?

Работа 5. Установка базовой системы

Цель работы: выбрать зеркала загрузки исходного кода установки, выбор сервера удаленной синхронизации, копирование сведений о DNS, монтирование файловых систем, выбор часового пояса.

Освоить: порядок и способы выбора зеркал, выбор зеркала с ротацией, изменение корневого каталога для перехода в базовую операционную среду, сохранение работоспособности сети при переходе, установка снимка дерева Portage, обновление дерева Portage, выбор профиля ОС, настройка переменных USE (пользователя), конфигурация локальных настроек.

Теоретическая часть

5.1. Выбор зеркала

Для ускорения загрузки исходного кода рекомендуется выбрать зеркало. Portage ищет переменную GENTOO_MIRRORS в файле `make.conf` и использует зеркала, перечисленные в ней.

Утилита `mirrorselect` позволяет выбрать зеркала. Нужно перейти к зеркалам, и нажать клавишу пробел для выбора одного или более зеркал:

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

Вторая важная настройка — установка значения переменной SYNC в файле `make.conf`. Эта переменная указывает на сервер `rsync` (сервер удаленной синхронизации), выбранный для обновления дерева Portage (коллекции файлов `ebuild` — сборочных сценариев, содержащих все данные, нужные Portage для скачивания и установки программ). Адрес сервера в SYNC можно вписать вручную, но `mirrorselect` позволяет упростить это действие:

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

После выполнения `mirrorselect`, рекомендуется проверить все значения в файле `/mnt/gentoo/etc/portage/make.conf`.

Рекомендуется выбрать зеркало с ротацией, такое как `rsync.us.gentoo.org`, а не единственное зеркало. Это помогает распределить

нагрузку, и дает надежное решение в случае, если одно из зеркал выйдет из строя.

5.2. Копирование сведений о DNS

Перед тем, как перейти в новую среду, необходимо: скопировать информацию о DNS (системе доменных имен) в файл `/etc/resolv.conf`. Это нужно, чтобы при переходе сохранить работоспособность сети. В файле `/etc/resolv.conf` содержатся адреса серверов имен, используемых в сети.

Копирование информации о DNS:

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

Параметр «-L» нужен, чтобы случайно не скопировать символическую ссылку.

5.3. Монтирование файловых систем

Нужно изменить корень файловой системы Linux на новое местоположение. Для правильной работы нового окружения нужно, чтобы там существовали некоторые файловые системы. Для этого файловая система `/proc` монтируется в `/mnt/gentoo/proc`, чтобы после изменения корневого каталога устанавливаемая система смогла обращаться к информации, предоставляемой ядром. Затем нужно создать привязку монтирования файловых систем `/dev` и `/sys`.

```
# mount -t proc proc /mnt/gentoo/proc
# mount --rbind /sys /mnt/gentoo/sys
# mount --rbind /dev /mnt/gentoo/dev
```

5.4. Переход в новую среду

Разделы подготовлены. Базовая операционная среда установлена. Для входа в нее нужно изменить корневой каталог. Таким образом, мы перейдем из текущей установочной среды (с компакт-диска или другого установочного носителя) в устанавливаемую систему (находящуюся в недавно размеченных разделах).

Изменение выполняется в три этапа.

1. Сначала, с помощью `chroot` изменяется корневой каталог с / (находящийся на установочном носителе) на `/mnt/gentoo` (находящийся на созданных дисковых разделах).

2. Далее перезагрузим некоторые настройки, как записано в /etc/profile в память, используя source.

3. Изменим вид приглашения, чтобы он напоминал, что мы находимся в новой среде.

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

Теперь находимся в своей среде Gentoo Linux. Осталось выполнить еще несколько разделов.

5.5. Настройка Portage

5.5.1. Установка снимка дерева Portage

Нужно установить снимок дерева Portage — набор файлов, содержащих сведения для Portage о программном обеспечении, доступном для установки, имеющихся профилях и т. д.

Рекомендуется использовать emerge-websync. Данная команда скопирует самый новый образ Portage (который Gentoo выпускает каждый день) с одного из зеркал, и установит его на систему:

```
# emerge-websync
```

В процессе работы emerge-websync может сообщить, что отсутствует каталог /usr/portage. Программа создаст этот каталог сама. Начиная с этого момента, Portage может отмечать, что рекомендует произвести некоторые обновления. Это происходит из-за того, что у некоторых системных пакетов, которые установили из файла stage3, могут появиться более новые версии, и теперь Portage узнала о них, так как установлен новый снимок дерева Portage. Можно игнорировать такие сообщения и обновить систему после того, как закончится установка Gentoo.

5.5.2. Обновление дерева Portage

Можно обновить дерево Portage до самой последней версии. Команда emerge --sync будет использовать протокол rsync для обновления дерева Portage (которое загрузили ранее с помощью emerge-websync) до самого нового состояния с помощью команды:

```
# emerge --sync.
```

Для ускорения процесса можно добавить параметр `--quiet`:

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

Если компьютер подключен к Интернету через межсетевой экран, блокирующий прохождение `rsync`-пакетов, можно игнорировать данный шаг, так как у вас уже достаточно новое дерево Portage.

Если выдано предупреждение, что имеется новая версия Portage и ее нужно обновить, можно выполнить обновление командой `emerge --oneshot portage`.

5.5.3. *Новости Portage*

Когда дерево Portage на новой системе синхронизировано, Portage может предупредить следующим сообщением:

- * IMPORTANT: 2 news items need reading for repository 'gentoo'.
- * Use `eselect news` to read news items.

Новости Portage были созданы чтобы обеспечить возможность передачи критических сообщений пользователям через дерево `rsync`. Чтобы работать с ними, нужно использовать `eselect news`. С помощью подкоманды `read` можно прочитать все новости. С помощью `list` можно увидеть общий обзор всех новостей, а с помощью `urge` можно удалить их, как только вы их читаете.

```
# eselect news list  
# eselect news read
```

5.6. Выбор нужного профиля

Профиль — конструктивный элемент любой ОС. В нем указываются не только значения по умолчанию для `USE`, `CFLAGS` и других переменных, он также привязывает систему к определенному диапазону допустимых версий пакетов. Диапазоны поддерживаются разработчиками ОС.

Ранее такой профиль не менялся пользователями. Однако, могут быть ситуации, когда пользователь может решить, что необходимо изменить профиль.

Узнать, какой профиль используется в системе в данный момент, можно командой:

```
# eselect profile list
```

Available profile symlink targets:

- [1] default/linux/x86/13.0 *
- [2] default/linux/x86/13.0/desktop
- [3] default/linux/x86/13.0/desktop/gnome
- [4] default/linux/x86/13.0/desktop/kde

Для некоторых архитектур предусмотрены подпрофили desktop. Запустив `eselect profile list`, можно увидеть список всех существующих профилей. После просмотра профилей, которые существуют для вашей архитектуры, можно выбрать какой-нибудь другой профиль:

```
# eselect profile set 2
```

Подпрофиль `developer` создан специально для целей разработки Gentoo Linux. Он не предназначен для настройки общих окружений для разработки.

5.6.1. Настройка переменной USE

USE («использовать») — одна из самых главных переменных, имеющих в распоряжении пользователей. Она позволяет при компиляции программ включать или отключать поддержку определенных необязательных функций. Например, некоторые программы можно компилировать с поддержкой `gtk` или `qt` на выбор. Другие можно собирать, включая или отключая поддержку SSL. Некоторые программы можно компилировать с поддержкой кадрового буфера (`svgalib`) вместо поддержки X11 (X-сервера).

В большинстве дистрибутивов пакеты собраны с поддержкой практически всех возможных вариантов. Это увеличивает размер программ и время запуска, приводит к чрезмерным зависимостям. Можно самостоятельно определять, с какими возможностями следует компилировать пакет. Здесь играет роль переменная USE.

В переменной USE указываются ключевые слова, которые отражаются на параметрах компиляции. Например, параметр `ssl` включает компиляцию с поддержкой `ssl` всех программ, которые способны его поддерживать. `-X` отключает поддержку X-сервера (обратите внимание на предшествующий знак «минус»). Параметры `gnome gtk -kde -qt4` обеспечивают компиляцию программ с поддержкой Gnome и `gtk`, но без поддержки KDE и `qt`, делая систему оптимальной для GNOME.

Настройки USE по умолчанию хранятся в файлах `make.defaults` профиля. Файлы `make.defaults` находятся в каталоге, на который указывает `/etc/make.profile`, а также во всех родительских каталогах. Значение USE по умолчанию — это сумма всех значений USE во всех файлах `make.defaults`. Все, что вносится в `/etc/portage/make.conf`, рассчитывается относительно этих значений. Когда добавляется что-либо к значению USE, это добавляется в список по умолчанию. Когда удаляете что-либо (указывая ключевое слово с предшествующим знаком минус), оно удаляется из списка по умолчанию. Ничего не меняйте в `/etc/make.profile`. Все, что там находится, перезаписывается при обновлении Portage!

Полный список возможных USE-флагов находится в вашей системе в файле `/usr/portage/profiles/use.desc`. Например:

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

Для прокрутки пользуйтесь клавишами стрелок, для выхода нажимайте «q».

Ниже приведены значения USE для системы, базирующейся на KDE, с включением поддержки DVD, ALSA и записи CD:

```
# nano -w /etc/portage/make.conf  
USE="gtk -gnome qt4 kde dvd alsa cdr"
```

5.6.2. Часовой пояс

Необходимо выбрать часовой пояс (time zone), чтобы система знала, свое местонахождение. Найдите нужный пояс в `/usr/share/zoneinfo`, затем запишите его в файл `/etc/timezone`:

```
# ls /usr/share/zoneinfo
```

Допустим, хотите использовать `Europe/Brussels`:

```
# echo "Europe/Brussels" > /etc/timezone
```

Избегайте использования часовых поясов `/usr/share/zoneinfo/Etc/GMT*`, т. к. их названия не соответствуют ожидаемым поясам. Например, `GMT-8` фактически является поясом `GMT+8`.

Переконфигурируйте пакет `timezone-data`, который обновит файл `/etc/localtime`, основываясь на `/etc/timezone`. Файл `/etc/localtime`

используется библиотекой C для получения информации о часовом поясе:

```
# emerge --config sys-libs/timezone-data
```

5.6.3. Конфигурация локальных настроек

Возможно использовать только одну или две локальные настройки на системе. Можно определить локальные настройки, которые нужны в `/etc/locale.gen`:

```
# nano -w /etc/locale.gen
```

Следующие локальные настройки являются примером, когда надо получить как Английский (Соединенные Штаты), так и Немецкий (Германия) языки с соответствующими форматами символов (вроде UTF-8).

```
en_US ISO-8859-1
en_US.UTF-8 UTF-8
de_DE ISO-8859-1
de_DE@euro ISO-8859-15
```

Можно выбрать желаемые локальные настройки из списка, запустив команду `locale --a`.

Рекомендуется использовать хотя бы одну локальную настройку UTF-8, так как некоторые приложения могут ее требовать.

Следующий шаг — запуск `locale-gen`. Эта программа сгенерирует все локальные настройки, которые определили в файле `/etc/locale.gen`:

```
# locale-gen
```

Можно проверить, что выбранные локальные настройки создались, запустив `locale --a`. Как только данная команда будет выполнена, появится возможность установить общесистемные настройки локальных настроек. С помощью `eselect locale list`, отображаются существующие цели:

```
# eselect locale list
Available targets for the LANG variable:
[1] C
```

- [2] POSIX
- [3] en_US
- [4] en_US.iso88591
- [5] en_US.utf8
- [6] de_DE
- [7] de_DE.iso88591
- [8] de_DE.iso885915
- [9] de_DE.utf8

[] (free form)

Нужную локальную настройку можно установить с помощью `eselect locale set <значение>`:

```
# eselect locale set 9
```

Можно выполнить все это вручную, создав или исправив файл `/etc/env.d/02locale`:

```
LANG="de_DE.UTF-8"  
LC_COLLATE="C"
```

Не забудьте перезагрузить окружение:

```
# env-update && source /etc/profile
```

Контрольные вопросы

- 1) Как выбрать зеркала загрузки исходного кода установки?
- 2) Как выбрать сервер удаленной синхронизации?
- 3) Зачем нужно копировать сведения о DNS?
- 4) Как выполняется монтирование файловых систем?
- 5) Как производится выбор часового пояса?
- 6) Каков порядок изменения корневого каталога при переходе в базовую операционную среду?
- 7) Как и зачем производится обновление дерева Portage?
- 8) Что такое профиль ОС?
- 9) Зачем производится настройка переменных USE (пользователя)?
- 10) Что такое локальные настройки?

Работа 6. Настройка ядра

Цель работы: установить исходные коды ядра, познакомиться с ручной настройкой ядра, выполнить автоматическую установку ядра, назначить автоматически запускаемые модули ядра.

Освоить: выбор из нескольких типов ядра необходимый, определение состав аппаратуры компьютера, компиляцию и установку ядра, сконфигурировать модули ядра.

Теоретическая часть

6.1. Установка исходных кодов ядра

Ядро ОС это слой между пользовательскими программами и оборудованием. С помощью команды `emerge` выберите и установите ядро:

```
# emerge gentoo-sources
```

Просмотрев содержимое каталога `/usr/src`, можно увидеть символическую ссылку `linux`, которая указывает на каталог с исходными кодами ядра. В данном случае, установленный исходный код ядра указывает на `gentoo-sources-3.12.20`. Текущая версия может отличаться.

```
# ls -l /usr/src/linux
```

```
lrwxrwxrwx 1 root root 12 Oct 13 11:04 /usr/src/linux -> linux-3.12.20
```

Перейдем к настройке и компиляции ядра. Можно использовать сценарий `genkernel`, который сформирует стандартное ядро, аналогичное используемому на установочном компакт-диске.

6.2. Ручная настройка

Ручная настройка ядра считается одним из самых сложных вопросов для пользователей Linux. Запуская ручную настройку ядра, следует знать состав аппаратуры своего компьютера. Большую часть информации можно собрать, установив пакет `pciutils` (`emerge pciutils`), в который входит `lspci`.

`lspci` можно использовать, находясь в среде с временным корневым каталогом. Можно не обращать внимания на предупреждения, касающиеся `pcilib` (например: `pcilib: cannot open /sys/bus/pci/devices [pcilib: не открыть /sys/bus/pci/devices]`), которые выводит на экран `lspci`.

Можно запустить `lspci` и вне временной среды установки. Результат тот же. Можно запустить `lsmod` для просмотра модулей ядра, используемых установочным компакт-диском (это даст хорошую подсказку, какие параметры включать). Войдите в каталог с исходными кодами ядра и введите команду `make menuconfig`. Этой командой вызовете меню настройки, использующее `ncurses`.

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

Появятся несколько разделов настройки. Перечислим пункты, включение которых обязательно.

6.2.1. Разделы настройки ядра

Убедитесь, что каждый драйвер, необходимый для загрузки системы (например, SCSI-контроллера), собран внутри ядра, а не как модуль, иначе система вообще не сможет загрузиться.

Выберите нужное семейство процессоров:

Processor type and features -->
(измените в соответствии со своей системой)
(Athlon/Duron/K7) Processor family

Выберите `Maintain a devtmpfs file system to mount at /dev`, чтобы критические файлы устройств уже существовали во время процесса загрузки.

Device Drivers -->
Generic Driver Options -->
[*] `Maintain a devtmpfs filesystem to mount at /dev`
[] `Automount devtmpfs at /dev, after the kernel mounted the rootfs`

Войдите в `File Systems` и выберите поддержку всех нужных файловых систем. Не компилируйте поддержку файловой системы, которую имеет ваш корневой раздел как модуль, иначе система Gentoo не сможет смонтировать ваш раздел. Также выберите `Virtual memory` и `/proc file system`.

File systems -->
(выберите одну или более опций в зависимости от вашей системы)
<*> `Second extended fs support`

- <*> Ext3 journalling file system support
- <*> The Extended 4 (ext4) filesystem
- <*> Reiserfs support
- <*> JFS filesystem support
- <*> XFS filesystem support

...

Pseudo Filesystems -->

- [*] /proc file system support

- [*] Virtual memory file system support (former shm fs)

(Включите поддержку меток GPT, если вы ее использовали)

-*- Enable the block layer -->

...

Partition Types -->

- [*] Advanced partition selection

...

- [*] EFI GUID Partition support

Если используется PPPoE или коммутируемого соединения для соединения с Интернетом, потребуется включить в ядре следующие функции:

Device Drivers -->

Networking device Support -->

- <*> PPP (point-to-point protocol) support

- <*> PPP support for async serial ports

- <*> PPP support for sync tty ports

Два параметра сжатия данных не являются обязательными, но и не повредят. То же относится и к параметру PPP over Ethernet, который нужен только для ppp, для поддержки PPPoE в режиме ядра.

Включите в ядре поддержку своей сетевой платы. Если у вас процессор Intel с поддержкой HyperThreading или многопроцессорная система, следует включить «симметричную многопроцессорную обработку».

Processor type and features -->

- [*] Symmetric multi-processing support

В многоядерных системах каждое ядро считается отдельным процессором.

Если у вас более 4 Гб ОЗУ, вам нужно включить опцию «High Memory Support (64 G)»

Если вы пользуетесь устройствами ввода на шине USB (например, мышью или клавиатурой), не забудьте включить их поддержку.

Device Drivers -->

[*] HID Devices -->

<*> USB Human Interface Device (full HID) support

Если нужна поддержка PCMCIA (для ноутбука), то включите поддержку моста PCMCIA своей системы.

Bus options (PCI etc.) -->

PCCARD (PCMCIA/CardBus) support -->

<*> PCCard (PCMCIA/CardBus) support

(если нужно (обычно нужно), включите поддержку старых 16-битных карт PCMCIA)

<*> 16-bit PCMCIA support

[*]32-bit CardBus support

(выберите наиболее подходящий мост)

*** PC-card bridges ***

<*> CardBus yenta-compatible bridge support (NEW)

<*> Cirrus PD6729 compatible bridge support (NEW)

<*> i82092 compatible bridge support (NEW)

6.3. Компиляция и установка

Выйдите из настройки и запустите процесс компиляции:

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

По завершении компиляции, скопируйте образ ядра в каталог /boot. Это делается командой: # make install. Данная команда копирует образ ядра в каталог /boot вместе с файлом System.map и файлом настройки ядра.

6.3.1. Компиляция *Initramfs*

Если используется особая разбивка на разделы, в которой важные для файловой системы каталоги (вроде /usr или /var) находятся на отдельных разделах, нужно настроить *initramfs* так, чтобы эти разделы могли быть смонтированы до того, как они понадобятся.

Без `initramfs` есть возможность, что система не загрузится так, как необходимо. Программы, которые нужны для монтирования файловых систем будут нуждаться в информации, расположенной на этих файловых системах. `Initramfs` поместит необходимые файлы в архив, который будет использован сразу после загрузки ядра, но перед тем, как будет запущена в работу программа `init`. Скрипты в `initramfs` проверят, что разделы смонтированы правильно, прежде чем система продолжит загружаться.

Чтобы установить `initramfs`, сначала нужно установить программу `genkernel`, и с ее помощью сгенерировать `initramfs`.

```
# emerge genkernel
# genkernel --install initramfs
```

Если нужна поддержка каких-либо функций в `initramfs`, например `lvm` или `raid`, нужно добавить необходимые опции при вызове `genkernel`. Посмотрите `genkernel -help` для более детальной информации.

В примере добавляется поддержка `LVM` и программного `raid` (`mdadm`):

```
# genkernel --lvm --mdadm --install initramfs
```

`Initramfs` будет сохранен в `/boot`. Можно найти этот файл, просто просмотрев список файлов, начинающихся с `initramfs`:

```
# ls /boot/initramfs*
```

6.4. Использование `genkernel`

Если исходные коды ядра установлены, возможно скомпилировать ядро, собрав его автоматически при помощи сценария `genkernel`.

`Genkernel` выполняет настройку ядра так же, как настраивается ядро на установочном компакт-диске. То есть система, ядро которой собрано `genkernel` будет выполнять определение всех устройств при загрузке. `Genkernel` не требует ручной настройки ядра.

Рассмотрим, как использовать `genkernel`. Сначала установите его:

```
# emerge genkernel
```

Необходимо отредактировать файл `/etc/fstab`, чтобы строка, содержащая во второй колонке `/boot` содержала в первой колонке

правильное устройство. Если следовать вышеприведенному примеру разбивки дисков, то скорее всего это устройство будет /dev/sda2 с файловой системой ext2. Запись в данном файле будет выглядеть примерно так:

```
# nano -w /etc/fstab
...
/dev/sda2 /boot ext2 defaults 0 2
```

Теперь скомпилируйте исходный код ядра, запустив `genkernel all`. Компиляция займет значительное время, поскольку `genkernel` собирает ядро, поддерживающее практически любое оборудование.

Если в загрузочном разделе не используется файловая система `ext2` или `ext3`, то придется вручную настроить ядро, запустив команду `genkernel --menuconfig all`, добавив поддержку используемой файловой системы в ядро (т. е. не модулем). При использовании LVM2, нужно добавить параметр командной строки `--lvm`.

Запуск `genkernel`: `# genkernel all`.

В итоге работы `genkernel` будет создано ядро, полный набор модулей и начальный RAM-диск (`initramfs`). Ядро и `initrd` будут использоваться в процессе настройки загрузчика системы, которая описана далее. Необходимо записать имена ядра и начального корневого диска, так как они понадобятся при составлении конфигурационного файла загрузчика. `initrd` запускается сразу при начальной загрузке, выполняя автоматическое определение устройств (точно так же, как при запуске установочного диска), до запуска «настоящей» системы.

Уточнение названия созданного образа ядра и `initrd`:

```
# ls /boot/kernel* /boot/initramfs*
```

6.5. Модули ядра

Конфигурируем модули. Необходимо описать модули, которые будут запускаться автоматически, в `/etc/conf.d/modules`. Можно добавить дополнительные опции к модулям, если это необходимо.

Чтобы увидеть все существующие модули, нужно запустить следующую команду `find`. Необходимо подставить вместо `<версия ядра>` версию ядра, которая только что скомпилирована:

```
#find/lib/modules/<версия ядра>/-type f-iname '*.o'-or-iname '*.ko'|less
```

Например, чтобы автоматически загрузить модуль 3c59x.ko (который является драйвером для специфического семейства сетевых карт 3Com), отредактируйте файл /etc/conf.d/modules и добавьте имя модуля в него.

```
# nano -w /etc/conf.d/modules  
modules_2_6="3c59x"
```

Контрольные вопросы

- 1) Как определить текущую версию ядра?
- 2) Как войти в каталог с исходными кодами ядра?
- 3) Перечислите и объясните, зачем нужны обязательные разделы настройки ядра?
- 4) Как запустить процесс компиляции ядра?
- 5) Как автоматически скомпилировать ядро?
- 6) Как уточнить название созданного образа ядра?
- 7) Где находятся модули, которые будут запускаться автоматически?
- 8) Как увидеть все существующие модули?

Работа 7.

Настройка параметров системы

Цель работы: настроить подключение разделов системы, сетевое подключение, установить пароль администратора.

Освоить: использование символических ссылок системы, определение сетевого интерфейса, выбор раскладки клавиатуры, выбор часового пояса.

Теоретическая часть

Все разделы, используемые системой, должны быть перечислены в файле /etc/fstab. В этом файле указываются точки подключения разделов (mountpoints, местоположение разделов в файловой системе), порядок подключения, а также дополнительные параметры (автоматический или ручной режим подключения, достаточность прав пользователя для подключения).

7.1. Создание файла `/etc/fstab`

В файле `/etc/fstab` используется специальный формат. Каждая строка состоит из шести полей, разделяемых пробелами, знаками табуляции или их сочетанием. Каждое поле имеет свое назначение.

- Первое поле обозначает раздел (`partition`) (путь к файлу устройства).
- Второе поле указывает точку подключения (`mountpoint`), в которую монтируется раздел.
- Третье поле задает тип файловой системы (`filesystem`), используемой в разделе.
- В четвертом поле указываются параметры подключения (`mountoptions`), используемые `mount` при подключении раздела. При указании нескольких параметров подключения их следует разделять запятыми.
- Пятое поле используется `dump` для определения, требуется ли резервное копирование раздела средствами `dump`. Обычно это поле можно просто установить в 0 (ноль).
- Шестое поле используется `fsck` для определения порядка проверки (`check`) файловых систем после некорректного завершения работы системы. Для корневой файловой системы рекомендуется значение 1, а для остальных — 2 (или 0, когда проверка файловой системы не требуется).

Вариант файла `/etc/fstab` по умолчанию, входящий в `Gentoo`, не является работоспособным.

Потребуется создать собственный `/etc/fstab`:

```
# nano -w /etc/fstab
```

В оставшейся части текста используются стандартные блочные файлы устройств `/dev/sd*` в качестве разделов. Можно использовать символические ссылки, находящиеся в `/dev/disk/by-id` или `/dev/disk/by-uuid`. Эти имена не изменятся, в то время как стандартное имя файлов блочных устройств зависит от множества факторов (например, как и в каком порядке диски подсоединены к системе). Если порядок дисков изменяться не будет, можно продолжить с наименованием блочных устройств по умолчанию.

Рассмотрим на примере, как указываются параметры загрузочного раздела `/boot`. В примере разбивки для x86 по умолчанию, `/boot` — это обычно раздел `/dev/sda2` с файловой системой `ext2`. При загрузке требуется его проверка.

Пример строки:

```
/boot в /etc/fstab  
/dev/sda2      /boot  ext2   defaults      0 2
```

Можно не подключать раздел /boot автоматически, чтобы повысить безопасность системы. Для этого нужно заменить defaults на noauto. В таком случае придется подключать раздел вручную каждый раз, когда он потребуется. Далее нужно указать правила, соответствующие схеме разбивки, и добавить правила для дисководов CD-ROM. На основе приведенного примера создайте файл /etc/fstab:

```
/dev/sda2      /boot      ext2   defaults,noatime      0 2  
/dev/sda3      none       swap   sw                    0 0  
/dev/sda4      /          ext4   noatime              0 1  
/dev/cdrom     /mnt/cdrom auto    noauto,user          0 0
```

Параметр auto позволяет mount определять тип файловой системы автоматически (рекомендуется для съемных носителей, которые могут оказаться размечены в одной из множества существующих файловых систем), а user позволяет монтировать компакт-диски обычным пользователям. Чтоб повысить быстродействие, нужно добавить параметр noatime в параметры подключения, что приведет к ускорению за счет отключения регистрации отметки времени доступа к файлам (обычно в ней все равно нет необходимости). Это также рекомендуется сделать при использовании твердотельных жестких дисков (SSD), которые также должны включить опцию монтирования discard (пока ее поддерживают только ext4 и btrfs), что включает команду TRIM. Проверьте отредактированный файл /etc/fstab, сохраните его, и выйдите из редактора, чтобы продолжить настройку.

7.2. Параметры сети

Нужно назвать свой компьютер, имя потом можно изменить. Например, назвать свою систему tux, а домен — homenetwork.

```
# nano -w /etc/conf.d/hostname  
(присвойте переменной hostname имя своего узла)  
hostname=>tux»
```

Если нужно имя домена, его устанавливают в файле /etc/conf.d/net. Домен нужен в том случае, если это необходимо провайдеру

или сетевому администратору, или у вас есть DNS-сервер, но нет DHCP-сервера. По умолчанию файл `/etc/conf.d/net` не существует, поэтому вам может понадобиться создать его.

```
# nano -w /etc/conf.d/net
(присвойте переменной DNSDOMAIN имя своего домена)
dns_domain_lo=»homenetwork»
```

Если выберете не устанавливать имя домена, то можно отредактировать файл `/etc/issue` так, чтобы избавиться от сообщений «This is hostname.(none)» при входе в систему. Нужно удалить строку `.\O` из данного файла. Если есть домен NIS его также необходимо указать его имя:

```
# nano -w /etc/conf.d/net
(укажите название своего домена NIS переменной nis_domain)
nis_domain_lo="my-nisdomain"
```

Более детальная информация, по вопросам конфигурации DNS и NIS, находится в файле `/usr/share/doc/netifrc-*/net.example.bz2`. Их можно прочитать, используя `bzless`.

7.3. Настройка сети

Все параметры настройки сети собраны в файле `/etc/conf.d/net`. В нем используется простой формат. В файле `/usr/share/doc/netifrc-*/net.example.bz2` приведен подробно прокомментированный пример, рассматривающий различные конфигурации. Сначала нужно установить `netifrc`:

```
# emerge --noreplace netifrc
```

По умолчанию используется DHCP. Чтобы DHCP заработал, требуется установить DHCP-клиент. Если настройка сетевого подключения нужна для указания специфических параметров DHCP, или из-за того, что вообще не используется DHCP, нужно открыть `/etc/conf.d/net` в редакторе (`nano`):

```
# nano -w /etc/conf.d/net
```

Чтобы указать свой адрес IP, маску сети и шлюз, потребуется настроить как `config_eth0`, так и `routes_eth0`.

Возможно, что сетевой интерфейс вашей системы называется eth0. Предположим, что интерфейс называется так же, как назывался интерфейс после загрузки с установочного диска. Ручная настройка параметров IP для eth0:

```
config_eth0="192.168.0.2 netmask 255.255.255.0 brd 192.168.0.255"  
routes_eth0="default via 192.168.0.1"
```

Чтобы использовать DHCP, определите config_eth0.
Автоматическое получение адреса IP для eth0:

```
config_eth0=»dhcp»
```

В файле /usr/share/doc/netifrc-*/net.example.bz2 находится список всех возможных опций.

Если имеется несколько сетевых интерфейсов, нужно повторить эти шаги для config_eth1, config_eth2 и т. д. Сохраните параметры и выйдите из редактора, чтобы продолжить настройку.

7.4. Автоматический запуск сетевого подключения

Для запуска сетевых интерфейсов при загрузке необходимо добавить их в уровень запуска по умолчанию.

Добавление net.eth0 в уровень запуска default:

```
# cd /etc/init.d  
# ln -s net.lo net.eth0  
# rc-update add net.eth0 default
```

Если имеется несколько сетевых интерфейсов, потребуется создать для них файлы net.*, как только что сделали для net.eth0. Если позже выяснится, что предположение о имени сетевого интерфейса (которое, как мы подразумеваем, eth0), было ошибочным, тогда:

1. обновите файл /etc/conf.d/net и внесите туда правильное имя интерфейса (например, enr3s0 вместо eth0);
2. создайте новую символическую ссылку (например, /etc/init.d/net.enr3s0);
3. удалите старую символическую ссылку (rm /etc/init.d/net.eth0);
4. добавьте новую ссылку на уровень запуска default;
5. удалите старую ссылку, используя rc-update del net.eth0 default.

7.5. Сетевые узлы

Сведения в системе Linux о локальной сети указываются в `/etc/hosts`, и помогают разрешению имен в IP-адреса для узлов, не обрабатываемых сервером имен. Требуется определить вашу систему. Также можно определить другие системы в сети, если не собираетесь устанавливать собственную систему DNS.

Открытие `/etc/hosts`:

```
# nano -w /etc/hosts
```

Указание сведений об узлах сети (определение текущей системы):

```
127.0.0.1 tux.homenetwork tux localhost
(определите другие машины в своей сети — для этого у них должен
быть статический IP-адрес.)
192.168.0.5 jenny.homenetwork jenny
192.168.0.6 benny.homenetwork benny
```

Чтобы продолжить настройку, нужно сохранить файл и выйти из редактора.

7.6. Запуск PCMCIA

Пользователям PCMCIA нужно установить пакет `pcmciautils`:

```
# emerge pcmciautils
```

7.7. Параметры системы

Пароль root

Нужно установить пароль `root` (администратора), набрав:

```
# passwd
```

Gentoo использует файл `/etc/rc.conf` для конфигурации сервисов, а также для запуска и завершения работы системы. Откройте `/etc/rc.conf`, и прочитайте комментарии, находящиеся в этом файле.

```
# nano -w /etc/rc.conf
```

Завершив изменение этих двух файлов, сохраните их и выйдите из редактора.

Для управления раскладками клавиатуры в Gentoo используется `/etc/conf.d/keymaps`. Для настройки своей клавиатуры измените его:

```
# nano -w /etc/conf.d/keymaps
```

Выбрав неверную раскладку, можно получить непредсказуемый результат при попытке ввода с клавиатуры.

По завершении изменения `/etc/conf.d/keymaps` сохраните файл и выйдите из редактора.

Для настройки параметров часов в Gentoo используется `/etc/conf.d/hwclock`:

```
# nano -w /etc/conf.d/hwclock
```

Если аппаратные часы системы настроены не на часовой пояс UTC (Гринвич), в файл необходимо добавить строку `clock=»local»`. В противном случае можно заметить сдвиг часового пояса.

После завершения настройки `/etc/conf.d/hwclock` сохраните файл и выйдите из редактора.

Контрольные вопросы

- 1) Как указать точки подключения разделов?
- 2) Где указывается порядок подключения разделов?
- 3) Как указывается режимы подключения разделов?
- 4) Как определить сетевой интерфейс системы?
- 5) Как установить раскладку клавиатуры?
- 6) Как установить часовой пояс?
- 7) Как установить пароль администратора?

Работа 8.

Установка системных средств

Цель работы: выбрать и установить службу журналирования, установить утилиты для обслуживания файловых систем, установить клиент DHCP.

Освоить: выполнение команд по расписанию, индексацию файлов в системе, удаленный доступ к системе.

Теоретическая часть

8.1. Системные службы журналирования

Некоторые системные средства не включены в архив stage3, поскольку одинаковые возможности можно обеспечить различными пакетами. Их можно выбрать самостоятельно.

Первый инструмент, который необходимо выбрать, должен дать системе возможность журналирования.

Предлагается несколько служб журналирования. Это `sysklogd`, традиционный набор журналирующих программ, `syslog-ng`, расширенная служба журналирования, и `metalog` — служба журналирования с очень гибкими возможностями настройки. В Portage имеются и другие службы журналирования: количество доступных пакетов постоянно растет.

Если планируется использовать `sysklogd` или `syslog-ng`, позднее может потребоваться установка `logrotate`, поскольку в этих службах журналирования не предусмотрен никакой механизм ротации системных журналов. Чтобы установить выбранную службу журналирования, воспользуйтесь `emerge`, а затем добавьте ее в уровень запуска по умолчанию с помощью `rc-update`. В примере показана установка `syslog-ng`. Вместо нее можно подставить другую службу журналирования:

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

8.2. Демон cron

Демон `cron` выполняет команды по расписанию. Это удобно, когда нужно выполнять какие-либо команды регулярно.

Существует несколько вариантов демонов `cron` на выбор, включая `bcron`, `dcron`, `fcron` и `crontab`. Установить один из них можно тем же способом, что и службу журналирования. Однако, `dcron` и `fcron` требуют дополнительную команду для конфигурирования, а именно `crontab /etc/crontab`. Используйте `crontab`:

```
# emerge crontab
# rc-update add crontab default
(только если выбран dcron или fcron) # crontab /etc/crontab.
```

8.3. Индексация файлов

Если нужно индексировать файлы в своей системе (чтобы быстро находить их с помощью locate) нужно установить sys-apps/mlocate:

```
# emerge mlocate
```

8.4. Удаленный доступ

Если нужен доступ к системе удаленно после установки, нужно добавить sshd на уровень запуска по умолчанию:

```
# rc-update add sshd default
```

Если нужен доступ к терминалу по последовательному порту (что возможно в случае удаленных серверов), может понадобиться раскомментировать раздел последовательных консолей в /etc/inittab, если это не было сделано автоматически:

```
# nano -w /etc/inittab
```

Следующая выдержка показывает раскомментированный раздел:

```
# SERIAL CONSOLES
s0:12345:respawn:/sbin/agetty 9600 ttyS0 vt100
s1:12345:respawn:/sbin/agetty 9600 ttyS1 vt100
```

8.5. Утилиты

для обслуживания файловых систем

Для проверки целостности файловых систем, создания дополнительных файловых систем потребуются определенные утилиты, состав которых зависит от используемых файловых систем. Программы для работы с файловыми системами ext2, ext3 или ext4 (e2fsprogs) уже установлены как часть системы.

В таблице перечислены утилиты, которые необходимо устанавливать для обслуживания файловых систем различных типов:

Файловая система	Утилиты	Команда установки
XFS	xfsprogs	emerge xfsprogs
ReiserFS	reiserfsprogs	emerge reiserfsprogs
JFS	jfsutils	emerge jfsutils

8.6. Сетевые средства

8.6.1. Установка клиента DHCP

Если требуется, чтобы ОС автоматически получала IP-адрес для сетевых интерфейсов, необходимо установить dhcpd (или любой другой клиент DHCP). Если не сделать этого сейчас, то после завершения установки вы не сможете подключиться к Интернету.

```
# emerge dhcpd
```

8.6.2. Установка клиента PPPoE

Если для выхода в сеть требуется ppp, его нужно установить:

```
# emerge ppp
```

Контрольные вопросы

- 1) Для чего нужна служба журналирования?
- 2) Как выбрать и установить службу журналирования?
- 3) Какие утилиты используются для обслуживания файловых систем?
- 4) Что такое DHCP?
- 5) Как установить клиент DHCP?
- 6) Как настроить выполнение команд по расписанию?
- 7) Какие утилиты используются для обслуживания файловых систем?
- 8) Для чего нужна индексация файлов в системе?
- 9) Как установить удаленный доступ к системе?

Работа 9.

Настройка начального загрузчика

Цель работы: выбрать и установить начальный загрузчик, установить кадровый буфер, перезагрузить операционную систему.

Освоить: настройки отображения кадрового буфера, конфигурирование начального загрузчика, терминологию GRUB, порядок установки GRUB вручную и автоматически, особенности перезагрузки системы.

Теоретическая часть

Необходимо установить программу, которая будет запускать ядро при старте системы. Такого рода программа называется загрузчиком.

9.1. Кадровый буфер

Если настроили ядро с поддержкой кадрового буфера (или оставили настройки по умолчанию при использовании genkernel), можно включить буфер, добавив параметр `video` в файл конфигурации своего загрузчика.

Нужно знать устройство кадрового буфера. Рекомендуется использовать `uvesafb` в качестве драйвера VESA.

Параметр `video` отвечает за порядок отображения кадрового буфера. В нем указывается название драйвера кадрового буфера, а затем режимы, которые нужно включить. Все режимы приведены в: `/usr/src/linux/Documentation/fb/uvesafb.txt`.

Часто используемые опции:

Переменная	Описание
<code>ywrap</code>	Считаем, что видеоплата может закольцовывать свою память (продолжать выборку с начального адреса, когда достигнут конечный).
<code>mtrr:n</code>	Установка регистров MTRR; допустимые значения n: 0 — отключено 1 — без кэширования 2 — отложенная запись (write-back) 3 — объединенная запись (write-combining) 4 — сквозная запись (write-through).
<code>mode</code>	Установить разрешение, цветность и частоту кадров. Например, <code>1024 × 768-32@85</code> для разрешения <code>1024 × 768</code> , 32-битного цвета и частоты кадров 85 Гц

В результате должно получиться:

```
video=uvesafb:mtrr:3,ywrap,1024x768-32@85
```

Запишите составленное значение, оно понадобится далее.

9.2. Установка GRUB2

GRUB2 находится в пакете `sys-boot/grub`:

```
# emerge sys-boot/grub
```

Программа GRUB2 теперь установлена на системе, но еще не активирована.

9.2.1. Конфигурирование GRUB2

Дополнительно нужно установить файлы GRUB2 в `/boot/grub`. Если предположить, что первый диск (тот, откуда система загружается), называется `/dev/sda`, то следующая команда выполнит это автоматически:

```
# grub2-install /dev/sda
```

Можно сгенерировать конфигурацию GRUB2 на основании конфигурации из файла `/etc/default/grub` и скриптов `/etc/grub.d`.

Обычно GRUB2 автоматически определит ядро для загрузки (ядро самой новой версии, из тех которые находятся в `/boot`), а также то, какая файловая система является корневой.

Для генерации финальной конфигурации GRUB2 выполните команду `grub2-mkconfig`:

```
# grub2-mkconfig -o /boot/grub/grub.cfg
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-3.12.20-gentoo
Found initrd image: /boot/initramfs-genkernel-x86-3.12.20-gentoo
done
```

В выводе команды должно быть указано, что найдено хотя бы одно ядро Linux так как оно нужно для загрузки системы. Если используется `initramfs`, либо использовали `genkernel` для компиляции ядра, также должен определиться правильный образ `initrd`. Если этого не произошло, перейдите в `/boot` и проверьте его содержимое с помощью команды `ls`. Если файлы и в самом деле отсутствуют, вернитесь к инструкциям по конфигурации и установке ядра.

9.2.2. Терминология GRUB

При освоении GRUB нужно обратить внимание на то, как в нем именуются жесткие диски и разделы. Linux-раздел `/dev/sda2`, скорее

всего, в GRUB станет называться (hd0,1). Круглые скобки вокруг hd0,1 —обязательны.

Жесткие диски нумеруются, начиная с нуля, а не с «а»; разделы — с нуля, а не с единицы. Нужно помнить, что в виде hd нумеруются только жесткие диски, но не устройства atapi-ide, такие как приводы компакт-дисков. Та же нумерация используется для устройств SCSI (обычно им присваиваются номера большие, чем устройствам IDE, кроме случаев, когда BIOS настроен на загрузку с устройства SCSI). Когда BIOS настроен на загрузку с другого жесткого диска (например, с первичного ведомого), именно этот жесткий диск и становится hd0.

Например, если есть жесткий диск /dev/sda, и еще два /dev/sdb и /dev/sdc, то /dev/sdb7 будет соответствовать (hd1,6). В GRUB есть механизм автодополнения по клавише tab, облегчающий идентификацию жестких дисков и разделов, а проверку схемы нумерации устройств GRUB.

9.2.3. Альтернатива: использование GRUB Legacy

GRUB был переработан, и теперь существует новая версия, называемая GRUB2. Новый код GRUB2 очень отличается от GRUB, и именно поэтому версия GRUB теперь называется Legacy (наследием). Для установки GRUB Legacy его нужно добавить в систему:

```
# emerge sys-boot/grub:0
```

GRUB уже установлен, но потребуется подправить его файл конфигурации, и поместить GRUB в MBR, чтобы он автоматически загружал ядро. С помощью nano создайте файл /boot/grub/grub.conf:

```
# nano -w /boot/grub/grub.conf
```

Заполните grub.conf своими значениями. Удостоверьтесь, что указываете имя установленного файла образа ядра, и при необходимости имя своего образа начального корневого диска (initrd).

GRUB назначает имена устройствам из BIOS. Если измените настройки BIOS, буквы и цифры в именах устройств также могут измениться. Например, если поменяете порядок загрузки с устройств, может понадобиться также поменять конфигурацию GRUB.

Если кроме Linux нужно пользоваться Windows, то ранее приведенный пример разбивки будет недостаточен. В примере используются все четыре первичных раздела для Linux, в то время как хотя

бы один из них должен быть расширенным, если предполагается установить Windows на логический раздел.

Пример grub.conf:

```
# какой пункт загружать по умолчанию: 0 — первый, 1 — второй
и т. д.
default 0
# время ожидания до начала загрузки (по умолчанию timeout 30)
# симпатичная заставка,
# прокомментируйте, если у вас не установлена графическая видео-
плата
splashimage=(hd0,1)/boot/grub/splash.xpm.gz

title Gentoo Linux 3.12.20
# раздел с файлом образа ядра (или операционной системой)
root (hd0,1)
kernel /boot/vmlinuz-3.12.20-gentoo root=/dev/sda4

title Gentoo Linux 3.12.20 (rescue)
# раздел с файлом образа ядра (или операционной системой)
root (hd0,1)
kernel /boot/vmlinuz-3.12.20-gentoo root=/dev/sda4 init=/bin/bb
# следующие 4 строки нужны только, если вы также используете
# также систему с Windows. В данном случае, мы считаем, что
# Windows находится на /dev/sda6
title Windows XP
rootnoverify (hd0,5)
makeactive

chainloader +1
```

Если при компиляции ядра Linux включен `initramfs` для загрузки, то нужно изменить конфигурацию, указав на этот файл `initramfs` и настроив `initramfs` так, где находится настоящее корневое устройство:

Пример из GRUB для ядра с включенным `initramfs`:

```
title Gentoo Linux 3.12.20
root (hd0,1)
kernel /boot/3.12.20 real_root=/dev/sda4
initrd /boot/initramfs-genkernel-x86-3.12.20-gentoo
```

Если жесткий диск разбит по-другому или другое ядро, нужно внести необходимые изменения. Все пути, следующие за упоминанием устройства GRUB (например, (hd0,1)), приведены относительно точки подключения, а не корня файловой системы. Иначе, (hd0,1)/grub/splash.xpm.gz — на самом деле /boot/grub/splash.xpm.gz, так как (hd0,1) — это /boot.

Если избрана другая схема разбиения диска и не выделен для /boot отдельный раздел, префикс /boot, использованный в примерах выше, обязателен. Если использован рекомендованный план разбиения, префикс /boot не требуется, но все работает благодаря символической ссылке boot. Приведенные примеры должны работать независимо от того, есть отдельный раздел для /boot или нет.

Если надо передать ядру дополнительные параметры, нужно добавить их в конец строки kernel. Один параметр передаем ядру (root=/dev/sda4 или real_root=/dev/sda4). Можно добавлять и другие, например, параметр video для кадрового буфера, обсуждавшийся выше.

Если файл конфигурации начального загрузчика содержит параметр real_root, то нужно использовать параметр real_rootflags для установки опций монтирования корневой файловой системы.

Если используется ядро 2.6.7 или выше, а объем жесткого диска ограничен перемигками из-за того, что BIOS не в состоянии работать с дисками большого размера, потребуется добавить sda=stroke. Замените sda на устройство, которое требует данную опцию.

При использовании genkernel нужно помнить, что ядро применяет такие же загрузочные параметры, как на установочном компакт-диске. Например, если есть устройства SCSI, следует передать ядру параметр doscsi.

Сохраните grub.conf и выйдите из редактора. По-прежнему необходимо записать GRUB в MBR (Master Boot Record), чтобы GRUB автоматически запускался при загрузке системы. Разработчики GRUB рекомендуют использовать grub-install. Однако на случай некорректной работы grub-install есть возможность записать GRUB вручную.

9.2.4. Установка GRUB с помощью grub-install

Для установки GRUB надо выполнить команду grub-install. Однако grub-install не заработает сам по себе, т. к. мы находимся в среде с измененным корневым каталогом. Нужно создать файл /etc/mtab,

перечислив в нем все смонтированные файловые системы. Для этого есть легкий способ: просто скопировать содержимое `/proc/mounts` поверх `/etc/mtab`, исключив строку `rootfs`, если не создавали отдельный загрузочный раздел. Следующая команда подойдет в обоих случаях:

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

При использовании дисков virtio Linux, нужно сообщить grub, где нужно искать диски, так как в противном случае команда `grub-install` не сможет выполниться. Это делается путем добавления описания устройства в файл `device.map`:

```
# echo "(hd0) /dev/vda" >> /boot/grub/device.map
```

Теперь можно установить GRUB, используя `grub-install`:

```
# grub-install --no-floppy /dev/sda
```

9.2.5. Установка GRUB вручную

Для начала настройки, нужно ввести `grub`. Появится приглашение `grub>` — это командная строка `grub`. Теперь потребуется набрать команды, нужные для установки загрузочной записи GRUB на ваш жесткий диск:

```
# grub --no-floppy
```

Если отсутствуют приводы для дискет, к приведенной команде добавьте `--no-floppy`, чтобы `grub` не опрашивал несуществующие дисководы.

В примере GRUB устанавливаем так, чтобы он считывал нужную информацию с загрузочного раздела `/dev/sda2`, а загрузочная запись GRUB находилась в MBR (Master Boot Record) жесткого диска, чтобы первое, что видели после включения компьютера — это приглашение GRUB.

Находясь в GRUB, можно использовать автодополнение по клавише TAB. К примеру, если ввести «`root` (», а затем TAB, появится список устройств (таких как `hd0`). Если ввести «`root (hd0,`» и нажать TAB, появится список для выбора раздела из возможных (такого как `hd0,1`).

Установка GRUB в MBR:

```
grub> root (hd0,1) (указание расположения раздела с /boot)
grub> setup (hd0) (установка GRUB в MBR)
grub> quit (выход из оболочки GRUB)
```

9.3. Перезагрузка системы

Далее нужно выйти из изолированной среды и размонтировать все смонтированные разделы. Затем ввести команду `reboot`:

```
# exit
cdimage ~# cd
cdimage ~# umount -l /mnt/gentoo/dev{/shm,/pts,}
cdimage ~# umount /mnt/gentoo{/boot,/sys,/proc,}
cdimage ~# reboot
```

Контрольные вопросы

- 1) Для чего нужен начальный загрузчик?
- 2) Как выбрать начальный загрузчик?
- 3) Как установить начальный загрузчик?
- 4) Что такое кадровый буфер?
- 5) Как настроить кадровый буфер?
- 6) Как установить GRUB вручную?
- 7) Как установить GRUB автоматически?
- 8) Что нужно сделать перед перезагрузкой системы?

Работа 10.

Управление учетными записями

Цель работы: настроить учетные записи суперпользователя, пользователя для повседневной работы, групп пользователей.

Освоить: порядок настройки учетных записей, правила временного получения привилегий администратора, очистка диска.

Теоретическая часть

10.1. Завершение установки

Работа в учетной записи `root` (администратора) в системе Unix / Linux небезопасна, и ее следует всячески избегать. Рекомендуется

добавить учетную запись пользователя для повседневной работы. Членством пользователя в группах определяется, какие действия он сможет выполнять. В следующей таблице перечислено несколько важных групп, в которые, возможно включать пользователей.

Группа	Описание
audio	возможность доступа к аудиоустройствам;
cdrom	возможность прямого доступа к оптическим накопителям;
floppy	возможность прямого доступа к гибким дискам;
games	возможность играть в игры;
portage	возможность использования emerge –pretend с правами пользователя;
usb	возможность доступа к устройствам USB;
video	возможность доступа к средствам видеозахвата и аппаратному ускорению видео;
wheel	возможность использования команды su

Пример. Создадим учетную запись пользователя по имени john, входящего в группы wheel, users и audio. Сначала войдите в систему как root (только root может создавать учетные записи пользователей), а затем запустите useradd:

Login: root

Password: (ваш пароль root)

```
# useradd -m -G users,wheel,audio -s /bin/bash john
```

```
# passwd john
```

Password: (введите пароль для john)

Re-enter password: (повторно введите пароль для подтверждения).

Для временного получения привилегий root можно использовать команду su. Другой способ — пользоваться пакетом sudo, который при правильной настройке безопасен.

10.2. Очистка диска

По окончании установки и перезагрузке, можно удалить скопированный архив tar со stage3 с жесткого диска. Они были скопированы в каталог /:

```
rm /stage3-*.tar.bz2*
```

Получена работающая ОС с большим выбором возможностей и свойств.

Контрольные вопросы

- 1) Для чего нужны учетные записи?
- 2) Каков порядок настройки учетных записей?
- 3) Что нужно сделать для получения временных привилегий администратора?
- 4) Как производится очистка диска?
- 5) Для чего нужны групп пользователей?
- 6) Каков порядок настройки учетных записей?
- 7) Как настраивается учетная запись пользователя для повседневной работы?

Литература

1. *Девис У.* Операционные системы. Функциональный подход. – М.: Мир, 2014. – 437 с.
2. *Дейтел Х.* Введение в операционные системы UNIX, VAX, CP/M, MVS, VM. – М.: Мир, 2014. – 704 с.
3. *Сборник статей по Gentoo Linux* [Электронный ресурс]: http://www.opennet.ru:/docs/RUS/gentoo_book-linux.html.gz 25.05.2018.
4. *Таненбаум Э., Бос Х.* Современные операционные системы. – СПб.: Питер, 2017. – 1120 с.
5. *Robbins, Daniel.* Gentoo Linux Documentation – Making the distribution, Part1. [Электронный ресурс]: [http://www. Gentoo.org](http://www.Gentoo.org). 30.05.2018.

Содержание

Предисловие	3
Порядок выполнения и оформления лабораторных работ	4
Работа 1. Правила установки	4
Теоретическая часть.	4
1.1. Загрузка с установочного компакт-диска	5
Контрольные вопросы	6
Работа 2. Настройка сети	6
Теоретическая часть.	6
2.1. Автоматическое подключение к сети	6
2.2. Указание прокси-серверов	7
2.3. Проверка сети	7
2.4. Автоматизированная настройка сети	8
2.5. Ручная настройка сети	8
2.6. Использование DHCP	9
2.7. Сетевая терминология	10
2.8. Использование ifconfig и route	11
Контрольные вопросы	12
Работа 3. Подготовка дисков	13
Теоретическая часть.	13
3.1. Блочные устройства	13
3.2. Разделы дисков	13
3.3. Выбор GPT или MBR	14
3.4. Разработка схемы разбиения диска	15
3.5. Выбор параметров	15
3.6. Пространство подкачки	16
3.7. Загрузочный раздел BIOS.	17
3.8. Использование parted для создания разделов	17
3.9. Использование fdisk для создания разделов.	20
Контрольные вопросы	27
Работа 4. Конфигурация установочных файлов	27
Теоретическая часть.	28
4.1. Установка текущей даты и времени	28
4.2. Выбор источника установки системы.	28
4.3. Настройка параметров компиляции	30
Контрольные вопросы	32
Работа 5. Установка базовой системы	33
Теоретическая часть.	33
5.1. Выбор зеркала	33
5.2. Копирование сведений о DNS	34
5.3. Монтирование файловых систем.	34
5.4. Переход в новую среду	34

5.5. Настройка Portage	35
5.6. Выбор нужного профиля	36
Контрольные вопросы	40
Работа 6. Настройка ядра	41
Теоретическая часть	41
6.1. Установка исходных кодов ядра	41
6.2. Ручная настройка	41
6.3. Компиляция и установка	44
6.4. Использование genkernel	45
6.5. Модули ядра	46
Контрольные вопросы	47
Работа 7. Настройка параметров системы	47
Теоретическая часть	47
7.1. Создание файла /etc/fstab	48
7.2. Параметры сети	49
7.3. Настройка сети	50
7.4. Автоматический запуск сетевого подключения	51
7.5. Сетевые узлы	52
7.6. Запуск PCMCIA	52
7.7. Параметры системы	52
Контрольные вопросы	53
Работа 8. Установка системных средств	53
Теоретическая часть	54
8.1. Системные службы журналирования	54
8.2. Демон snoop	54
8.3. Индексация файлов	55
8.4. Удаленный доступ	55
8.5. Утилиты для обслуживания файловых систем	55
8.6. Сетевые средства	56
Контрольные вопросы	56
Работа 9. Настройка начального загрузчика	56
Теоретическая часть	57
9.1. Кадровый буфер	57
9.2. Установка GRUB2	58
9.3. Перезагрузка системы	63
Контрольные вопросы	63
Работа 10. Управление учетными записями	63
Теоретическая часть	63
10.1. Завершение установки	63
10.2. Очистка диска	64
Контрольные вопросы	65
Литература	65

Лабораторный практикум

Анатолий Витальевич Переспелов,
Руслан Анатольевич Переспелов

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Дистрибутив операционной системы Gentoo

*Начальник РИО А.В. Ляхтейнен
Редактор Л.Ю. Кладова
Верстка М.В. Ивановой*

Подписано в печать 05.10.18. Формат 60×90 ¹/₁₆. Гарнитура Times New Roman.
Печать цифровая. Усл. печ. л. 17. Тираж 100 экз. Заказ № 703.
РГГМУ, 195196, Санкт-Петербург, Малоохтинский пр., 98.
