Впервые опубликовано на developerWorks 02.09.2005

Учебник для экзамена LPI 201: Оборудование

Администрирование, средний уровень (LPIC-2) тема 204

Конфигурирование RAID

Что такое RAID?

RAID (Redundant Array of Inexpensive Disks) представляет из себя механизм объединения нескольких разделов или отдельных жестких дисков в большие или более надежные виртуальные диски. Изначально было определено большое число различных типов (уровней) RAID, но прижилось только три: RAID-0 (объединение дисков), RAID-1 (зеркалирование), and RAID-5 (объединение с сохранением контрольных сумм). RAID-4 также изредка используется; он достаточно близок к RAID-5, но контрольные суммы размещаются на специально выделенном устройстве, а не распределяются по дискам.

В этом руководстве обсуждается "new-style" RAID для Linux (он входит в ядра версии 2.4 и 2.6, для более ранних версий существуют backport'ы). "Old-style" RAID, изначально использовавшийся в версиях 2.0 и 2.2, содержит ошибки, и его использование не рекомендуется. По своей сути "new-style" представляет из себя 0.90 RAID layer, разработанный Ingo Molnar и др.

Использование RAID массива

Работу с RAID массивами можно разделить на две части. Простейшей задачей можно считать монтирование RAID. Как только виртуальное устройство RAID сконфигурировано, для команды `mount` оно выглядит как обычное блочное устройство. RAID массив после его создания носит название вида /dev/mdN и может быть смонтирован следующим образом:

```
% mount /dev/md0 /home
```

Также вы можете включить монтирование виртуального RAID раздела в /etc/fstab (как правило, это наилучшее решение). Драйвер устройства считывает суперблоки сырых разделов диска для сборки сконфигурированного раздела RAID.

Более сложной задачей (или более многоступенчатой) является создание RAID устройства из соответствующих сырых разделов. Вы можете создать раздел RAID при помощи программы `mkraid` в сочетании с конфигурационным файлом /etc/raidtab.

Так же вы можете воспользоваться новой программой `mdadm`, при помощи которой, вы можете манипулировать RAID-устройствами без необходимости править конфигурационные файлы. В большинстве дистрибутивов `mdadm` вытесняет `raidtools` (который включает `mkraid`), но в этом руководстве обсуждается именно

mkraid, для того, чтобы соответствовать требованиям, предъявляемым экзаменом LPI. Используемые подходы в обоих случаях сходны, но вам будет необходимо справиться со справочным руководством `man` по `mdadm` для изучения опций командной строки.

Формат /etc/raidtab

В файле /etc/raidtab используются следующие поля для описания компонентов RAID. Этот список не является исчерпывающим.

- `raiddev`: Раздел виртуального диска, предоставляемого RAID (/dev/md?). Это устройство, с которым могут работать `mkfs` и `fsck`, оно может быть подмонтировано как обычный дисковый раздел.
- `raid-disk`: Раздел используемый при создании RAID. Он должен иметь тип раздела 0xFD, установленный при помощи fdisk или подобной программы.
- `spare-disk`: Эти диски (как правило, это один диск) обычно остаются не задействованными. В случае если один из дисков, входящих в raid, выходит из строя, spare диск начинает выступать в качестве его замены.

Конфигурирование RAID-0

RAID-0 или "disk striping" дает большую производительность при операциях ввода/вывода ценой уменьшения общей надежности (выход из строя одного диска из raid-массива может привести к утрате всего устройства RAID). В качестве примера ниже приведен /etc/raidtab для создания устройства RAID-0:

```
raiddev /dev/md0
    raid-level      0
    nr-raid-disks   2
    nr-spare-disks  0
    chunk-size      32
    persistent-superblock   1
    device          /dev/sda2
    raid-disk       0
    device          /dev/sdb2
    raid-disk       1
```

Здесь определено виртуальное устройство RAID-0, имеющее название /dev/md0. Первые 32 KB устройства /dev/md0 выделяются на /dev/sda2, следующие 32 KB на /dev/sdb2, третьи на /dev/sda2 и т.д.

Для создания устройства выполните следующую команду:

```
% sudo mkraid /dev/md0
```

При использовании `mdadm` вместо файла /etc/raidtab используются опции.

Конфигурирование RAID-1

RAID-1 или "disk mirroring" просто дублирует данные на обоих блочных устройствах. RAID-1 великолепно справляется с задачами защиты от аппаратных сбоев, но

заметно снижает производительность. RAID-1 в целом стоит дороже, так как половина вашего дискового пространства резервируется. Например:

```
raiddev /dev/md0
    raid-level      1
    nr-raid-disks   2
    nr-spare-disks  1
    persistent-superblock 1
    device          /dev/sdb6
    raid-disk       0
    device          /dev/sdc5
    raid-disk       1
    device          /dev/sdd5
    spare-disk      0
```

Данные, записываемые на /dev/md0, будут сохранены и на /dev/sdb6 и на /dev/sdc5. Устройство /dev/sdd5 сконфигурировано как hot spare. В случае сбоя на устройстве /dev/sdb6 или /dev/sdc5, данные будут перенесены на /dev/sdd5, и оно будет переведено во включенное состояние для замены сбойного устройства.

Конфигурирование RAID-5

RAID-5 требует, по крайней мере, трех устройств и использует коррекцию ошибок для получения преимуществ, предоставляемых распределенными дисками, вместе с устойчивостью к сбою одного из устройств. Положительным моментом является необходимость использования только одного дополнительного устройства для обеспечения надежности. Отрицательным моментом является большая сложность RAID-5; при возникновении сбоя в одном из устройств, он переходит в режим degraded mode, который существенно снижает пропускную способность операций ввода/вывода, пока не будет завершена процедура подключения резервного spare-диска и перекачивание на него данных.

```
raiddev /dev/md0
    raid-level        5
    nr-raid-disks     7
    nr-spare-disks    0
    persistent-superblock 1
    parity-algorithm  left-symmetric
    chunk-size        32
    device            /dev/sda3
    raid-disk         0
    device            /dev/sdb1
    raid-disk         1
    device            /dev/sdc1
    raid-disk         2
    device            /dev/sdd1
    raid-disk         3
    device            /dev/sde1
    raid-disk         4
    device            /dev/sdf1
    raid-disk         5
    device            /dev/sdg1
    raid-disk         6
```

Использование mke2fs или mke3fs

Если вы форматируете виртуальные устройства RAID-5 при помощи e2fs или e3fs, вы должны обращать внимание на опцию stride. Опция `-R stride=nn` позволяет mke2fs размещать данные файловой системы ext2 так, что они лучше воспринимаются RAID-устройством.

Если chunk size установлен в 32 KB, это означает, что эти 32 KB последовательных данных будут размещаться на одном диске. Если файловая система ext2 имеет размер блока в 4 KB, то на восемь блоков файловой системы будет приходиться один array chunk. Мы можем указать эту информацию файловой системе, запустив команду:

```
% mke2fs -b 4096 -R stride=8 /dev/md0
```

Производительность RAID-5 существенно увеличивается при создании файловой системы с правильной информацией о stride.

Поддержка в ядре, обслуживание сбоев

Включение в ядре опции persistent-superblock дает возможность ядру стартовать RAID автоматически при загрузке системы. New-style RAID использует persistent superblock и поддерживается в ядрах 2.4 и 2.6. Для устаревших ядер версий 2.0 и 2.2 доступны соответствующие патчи.

При выходе из строя устройства происходит следующее:

- **RAID-0**: Все данные теряются;
- **RAID-1/RAID-5**: Сбойное устройство отключается, а spare-диск (если он имеется) включается, и данные переносятся на него.

Документ "The Software-RAID HOWTO" из Linux HOWTO project описывает переключение между устройствами при сбоях или обновлении устройств, включая hot-swap диски, и описывает случаи, когда необходима перезагрузка. Обычно, SCSI (или Firewire) устройства поддерживают горячую замену, а IDE устройства – нет.

Установка нового оборудования

Оборудование

Linux, особенно последние версии, обладает удивительной стабильностью работы и широким спектром совместимости с различными устройствами. В общем и целом, есть два уровня поддержки аппаратуры, которыми следует озаботиться. Первый уровень обеспечивает базовую поддержку на уровне системы, обычно это означает загрузку модуля ядра, соответствующего вашему устройству.

Второй уровень имеет отношение к некоторым устройствам, в той или иной степени требующим поддержки со стороны подсистемы X11R6: обычно это или XFree86, или X.Org (в прежние времена, использовались и коммерческие подсистемы X11, но в данном руководстве они не обсуждаются).

Поддержка основных категорий hot-swappable устройств, например, работающих через PCMCIA или USB интерфейсы, обсуждается далее в соответствующих разделах.

X11

К сведению: изначально X.Org является преемником проекта XFree86 (технически, это просто другая его ветвь). Не смотря на то, что XFree86 официально не свернут, практически все производители переключились на X.Org по лицензионным причинам. К счастью, за исключением небольших изменений в названиях конфигурационных файлов, основная часть кода в обеих ветвях одинакова, некоторые новые возможности с большей вероятностью будут поддерживаться только в X.Org.

X11R6 представляет собой (сетевое) графическое представление приложений на пользовательской рабочей станции. Вопреки интуитивному пониманию, "X сервер" -- это физическая машина, с которой непосредственно взаимодействует пользователь через клавиатуру, мышь, видео карту, дисплей и т.д. "X клиент" -- это машина, которая предоставляет процессорное время, дисковое пространство и другие ресурсы, не имеющие непосредственного отношения к уровню представления для запуска приложения. Во многих и даже в большинстве Linux систем, X сервер и X клиент сосуществуют на одной и той же машине, и эффективный локальный канал обмена информацией используется для взаимодействия с пользовательскими устройствами ввода/вывода.

X сервер, такой как X.Org, необходим для обеспечения поддержки устройств ввода/вывода, при помощи которых пользователь будет взаимодействовать с приложением. В подавляющем большинстве случаев сложности встречаются при настройке видео-карт и мониторов. К счастью, эти сложности остаются в прошлом с выходом последних версий X.Org/XFree86, где успешно реализована процедура автоматического детектирования оборудования. Технически X сервер также необходим для поддержки устройств ввода -- клавиатуры и мыши -- но обычно их подключение происходит безболезненно, поскольку они имеют стандартные интерфейсы. Все остальное: доступ к дисковым устройствам, принтерам, таким специальным устройствам, как сканеры и др. обслуживается клиентскими X приложениями и, в конечном счете, ядром Linux.

Поддержка устройств ядром системы

Практически все, что вам следует знать о поддержке устройств ядром Linux, ограничивается поиском, сборкой и загрузкой правильных модулей ядра. Все это исчерпывающе изложено в руководстве по теме 201, на большинство вопросов читатели смогут найти ответы там.

Для работы с модулями ядра системный администратор должен иметь представление о таких командах, как lsmod, insmod и modprobe и, в меньшей степени, о rmmod. Команды lsmod, insmod и rmmod -- это команды низкого уровня

для получения списка загруженных модулей, их загрузки и выгрузки в работающее ядро Linux. Команда modprobe проводит эти действия на более высоком уровне, проверяя взаимные зависимости и осуществляя необходимые вызовы insmod и rmmod в зависимости от необходимости.

Осмотр оборудования

Отдельные утилиты могут оказаться полезными для получения информации об имеющемся оборудовании. Команда lspci выдает детальную информацию о найденных PCI устройствах (во многих случаях включая даже те, которые работают через PCMCIA или USB шины). Соответственно, setpci может конфигурировать устройства на PCI шине. Команда lspnp выводит список BIOS device node и ресурсов для plug-and-play устройств. Команда lsusb подобным образом просматривает USB (для модификации конфигураций используется setpnp).

Настройка сервера X11 (первая часть)

Исходно X.Org (или XFree86) поставляется с набором видео драйверов и драйверов для других периферийных устройств, вам просто нужно подобрать правильные. В конечном счете, вся конфигурация X сервера располагается в достаточно детализированном, а местами несколько напоминающем шифровку, файле /etc/X11/xorg.conf (или xfree86.conf). Ряд стандартных утилит может быть использован для упрощения процесса конфигурирования, но текстовый редактор сработает в любом случае. Некоторые фронтэнды включаются непосредственно в X.Org, так для графического конфигурирования включен xorgcfg (допускаю, что вам он покажется не слишком работоспособным) и xorgconfig для конфигурирования в текстовом режиме. Многие дистрибутивы Linux снабжены более дружественными фронтендами.

Команда SuperProbe может оказаться полезной для определения модели вашей видео-карты. Вы можете так же обратиться к базе данных /usr/X11R6/lib/X11/Cards для получения детальной информации о поддерживаемых видео-картах.

Настройка сервера X11 (вторая часть)

Внутри конфигурационного файла /etc/X11/xorg.conf вы должны создать серию блоков "Section" ... "EndSection", каждая из которых определяет ряд деталей и опций, относящихся к конкретному устройству. Имена этих разделов следующие: * Files: Пути поиска файлов

```
* ServerFlags:    Флаги сервера
 * Module:         Загрузка динамических модулей
 * InputDevice:    Описание устройств ввода
 * Device:         Описание графического устройства
 * VideoAdaptor:   Xv описание видео-адаптера
 * Monitor:        Описание монитора
 * Modes:          Описание видео мод
 * Screen:         Конфигурация экрана
   * ServerLayout:   Общая конфигурация
 * DRI:            Конфигурация относящаяся к DRI
 * Vendor:         Специфичная для данного производителя конфигурация
```

Настройка сервера X11 (часть третья)

Среди всех секций, Screen выступает в качестве мастер-конфигурации системы отображения. Например, вы можете определить несколько разделов Monitor, но выбран будет только один, указанный в:

```
Section "Screen"
        Identifier    "Default Screen"
        Device        "My Video Card"
        Monitor       "Current Monitor"
        DefaultDepth  24
        SubSection "Display:
              Depth       24
              Modes       "1280x1024" "1024x768" "800x600"
        EndSubSection
        # more subsections and directives
    Endsection
```

Раздел, носящий название `ServerLayout`, в действительности является главной ("master") конфигурацией -- он ссылается и к используемой секции `Screen`, и к различным секциям `InputDevice`. В случае, если у вас возникли проблемы, скорее всего, они решаются правильным выбором `Device` или `Monitor`. К счастью, DPMS-мониторы нынче, как правило, избавляют нас от болезненной настройки опций `Modeline` (в прежние суровые времена, вам было необходимо отыскать крайне специфичные для вашего монитора частотно/временные характеристики, нынче же, DPMS делает эту работу за вас).

Конфигурирование PCMCIA устройств

PCMCIA

PCMCIA утройства иногда также называются устройствами PC-Card. Это тонкие, размером с кредитную карту, модули изначально проектировавшиеся с расчетом на hot-swap и легкость транспортировки. Наибольшее распространение получили они в ноутбуках. Однако некоторые десктопные и серверные конфигурации также имеют PCMCIA интерфейсы, зачастую установленные на внешних креплениях и подсоединенные через различные шины (специальные PCI или ISA карты, USB-трансляторы и т.д.). Различное оборудование может быть реализовано в размерах PCMCIA карт, включая Wireless и Ethernet адаптеры, микродрайвы, flash-диски, модемы, SCSI-адаптеры и многое другое специфическое оборудование.

Технически, PCMCIA интерфейс представляет из себя переходник к расположенным ниже уровнем ISA или PCI шинам. В большинстве случаев, передача происходит прозрачным образом -- одни и те же модули ядра или программы, которые взаимодействуют с ISA или PCI устройством, используются для обслуживания

протокола обмена предоставляемого PCMCIA. Единственное настоящее различие PCMCIA устройств заключается в распознавании события при установке устройства и определении типа карты, для которого следует загрузить драйвер.

В настоящее время оборудование PCMCIA скрывается в тени USB и/или Firewire устройств. Несмотря на то, что PCMCIA обладает более удобным конструктивом (обычно скрывающий карту в корпусе), USB более универсален для широкого круга машин. В результате многие устройства, прежде выпускавшиеся в стандарте PCMCIA, нынче оформляются в виде устройств в стиле USB "dongle", а старые доступны только на распродажах.

Определение PCMCIA устройства (часть первая)

В современных ядрах -- 2.4 и выше -- поддержка PCMCIA доступна в виде модуля ядра. Современные дистрибутивы такую поддержку включают, но если вы собираете свое собственное ядро, включите опции `CONFIG_HOTPLUG`, `CONFIG_PCMCIA` и `CONFIG_CARDBUS`. Ранее такая поддержка обеспечивалась пакетом `pcmcia-cs`.

Модули `pcmcia_core` и `pcmcia` отвечают за поддержку PCMCIA устройств. Модуль `yenta_socket` так же загружается для поддержки интерфейса CardBus (PCI-over-PCMCIA):

```
% lsmod | egrep '(yenta)|(pcmcia)'
    pcmcia                  21380  3 atmel_cs
    yenta_socket            19584  1
    pcmcia_core             53568  3 atmel_cs,pcmcia,yenta_socket
```

При установке карты в PCMCIA слот демон `cardmgr` обращается к базе данных /etc/pcmcia/config и загружает тот драйвер, который нужен.

Определение PCMCIA устройства (часть вторая)

Теперь взглянем на процесс идентификации PCMCIA устройства в действии. Я вставляю карту в Linux лэптоп со слотом PCMCIA, с включенной поддержкой указанных ранее модулей ядра. Я могу воспользоваться программой `cardctl` для просмотра информации об имеющемся оборудовании:

```
% cardctl ident
    Socket 0:
      product info: "Belkin", "11Mbps-Wireless-Notebook-Network-
   Adapter"
      manfid: 0x01bf, 0x3302
      function: 6 (network)
```

Эта информация предоставляется модулем ядра `pcmcia_core`, запросившего ее у физической карты. Как только идентификация проведена, `cardmgr` сканирует базу данных для того, чтобы найти соответствующий драйвер. Выглядит это примерно так:

```
% grep -C 1 '0x01bf,0x3302' /etc/pcmcia/config
    card "Belkin FSD6020 v2"
      manfid 0x01bf,0x3302
      bind "atmel_cs"
```

В этом случае нам нужен модуль ядра `atmel_cs` для поддержки wireless интерфейса, предоставляемого этой картой. Вы можете увидеть это, заглянув в /var/lib/pcmcia/stab или /var/run/stab, в зависимости от вашей системы:

```
% cat /var/run/stab
     Socket 0: Belkin FSD6020 v2
     0       network atmel_cs        0       eth2
```

Получение отладочной информации о PCMCIA устройстве

В приведенном выше примере выполнявшиеся этапы были незаметны. Карта была опознана, драйвера загружены, и все было подключено. Это идеальная ситуация. Если же что-то не в порядке, вы можете обнаружить, что драйвер, который надо загрузить, не найден.

Если вы уверены, что ваше PCMCIA может использовать имеющийся драйвер (например, он совместим с другим чипсетом), вы можете запустить insmod вручную для загрузки подходящего модуля. Или же, если вы постоянно используете эту карту, вы можете отредактировать /etc/pcmcia/config для того, что бы добавить поддержку этой карты, указав необходимый драйвер. Однако в случае, если ваше предположение относительно модуля не оправдалось, вам следует убедиться, что карта действительно совместима с какой-то другой известной PCMCIA картой.

Настройка загрузки PCMCIA может быть сделана через установочный скрипт /etc/pcmcia/, поименованный согласно функциональной категории. Например, когда 802.11b карта, подобная той, что загружалась в предыдущем примере, запускается скрипт /etc/pcmcia/wireless. Вы можете настроить эти скрипты, если устройство требует специальных настроек.

Использование "схем" для различных конфигураций

Если у вас возникает необходимость использовать PCMCIA устройство в различных конфигурациях, вы можете воспользоваться командой `cardctl scheme` для установки (или запроса) конфигурации. Например:

```
% sudo cardctl scheme foo
    checking: eth2
    /sbin/ifdown: interface eth2 not configured
    Changing scheme to 'foo'...
    Ignoring unknown interface eth2=eth2.
    % cardctl scheme
    Current scheme: 'foo'.
```

В этом случае, я в действительности не определяю схему `foo`, но если вы ее измените, то произойдет реконфигурация. Схемы могут быть использованы в настроечных скриптах при помощи анализа переменной `$ADDRESS`:

```
# in /etc/pcmcia/network.opts (called by /etc/pcmcia/network)
     case "$ADDRESS" in
     work,*,*,*)
         # definitions for network in work scheme ...
         ;;
     default,*,*,*)
         # definitions for network in default scheme ...
         ;;
     esac
```

Конечно, вы можете устанавливать схемы в инициализационных скриптах или переключать их событиями (через задания cron, GUI интерфейс и т.д.).

Конфигурирование Universal Serial Bus устройств

USB

Как уже было отмечено в разделе о PCMCIA, USB представляет из себя новую технологию, вытесняющую сейчас PCMCIA. USB поддерживает до 127 устройств с помощью гибкой радиальной топологии состоящей из хабов у оконечных устройств. USB имеет несколько версий с последовательно увеличивающимися скоростями передачи, последняя из них -- 2.0. Эта последняя версия USB теоретически поддерживает скорость обмена до 480 MBsec. USB 1.1 поддерживает меньшие скорости, до 12 MBsec. На практике, по ряду причин некоторые устройства фактически работают на меньших скоростях, чем предусмотренные теоретически -- тем не менее, более быстрый интерфейс более перспективен.

Распознавание USB устройств (часть первая)

С точки зрения администрирования, USB работает подобно PCMCIA. Он обслуживается модулем ядра usbcore. В ядрах 2.4+, предусмотрена более совершенная поддержка, чем в ядрах 2.2. На следующем уровне за usbcore, вступает в действие один из следующих модулей: uhci, uhci_hcp, ohci, ohci_hcp, ehci, ehci_hcp. Какой именно модуль понадобится, зависит от чипсета, использованного в вашем компьютере. Модуль ehci подключается, если они поддерживают высокоскоростную передачу по USB 2.0. Вообще же говоря, если ваш компьютер поддерживает ehci (или ehci_hcp), может потребоваться загрузка и модуля ehci для обеспечения обратной совместимости. Книга Брэда Хардса "The Linux USB sub-system" содержит детальное описание соответствий между различными чипсетами и модулями ядра. При создании ядра, которое будет использоваться на различных машинах, вам следует собрать все USB модули.

Для обеспечения корректной поддержки ядра система hotplug должна обеспечивать загрузку любых драйверов, необходимых для обслуживания подключенного устройства USB. Файл /proc/bus/usb/devices содержит детальную информацию о

доступных в настоящее время USB устройствах (как хабов, так и периферийных устройств).

Распознавание USB устройств (часть вторая)

Обычно шина USB монтируется как динамически генерируемая файловая система, подобная файловой системе /proc/. В зависимости от дистрибутива, /proc/bus/usb/ может монтироваться или в стартовых скриптах, типа /etc/rcS.d/S02mountvirtfs, или же через конфигурацию /etc/fstab. В последнем случае, вы сможете увидеть там строку подобную следующей:

```
# /etc/fstab
     none /proc/bus/usb usbdevfs defaults 0 0
```

Инициализационный же скрипт может выглядеть следующим образом:

```
mount -t usbdevfs none /proc/bus/usb
```

Механизмы распознавания устройств и управление всей подсистемой USB кроется в /etc/hotplug/, в первую очередь, в /etc/hotplug/usb.rc и /etc/hotplug/usb.agent. Установка USB устройства будет проводиться через операцию modprobe для нужного драйвера. Вы можете провести и дальнейшую настройку для данного устройства путем создания скрипта /etc/hotplug/usb/$DRIVER для вашего конкретного устройства.