Содержание

7 Руководство администратора

Общие принципы работы VeOS

Процессы и файлы

VeOS является многопользовательской интегрированной системой, т.е. она разработана в расчете на одновременную работу нескольких пользователей. Пользователь может либо сам работать в системе, выполняя некоторую последовательность команд, либо от его имени могут выполняться прикладные процессы. Пользователь взаимодействует с системой через командный интерпретатор, например, bash. Командный интерпретатор представляет собой прикладную программу, которая принимает от пользователя команды или набор команд и переводит их в системные вызовы к ядру системы. Интерпретатор позволяет пользователю работать с файлами, передвигаться по дереву файловой системы, запускать прикладные процессы. Все командные интерпретаторы UNIX имеют развитый командный язык и позволяют писать сложные программы, упрощающие процесс администрирования системы и работы с ней.

Процессы функционирования ОС

Все программы, которые загружены в память сервера (т.е. выполняются) в текущий момент времени, называются процессами. Процессы можно разделить на два основных класса: системные процессы и пользовательские процессы. Системные процессы — программы, решающие внутренние задачи ОС, например, организацию виртуальной памяти на диске или предоставляющие пользователям те или иные сервисы (процессы-службы). Пользовательские процессы — процессы, запускаемые пользователем из командного интерпретатора для решения задач пользователя или управления системными процессами. VeOS изначально разрабатывался как многозадачная система. Он использует технологии, опробованные и отработанные другими реализациями UNIX - предшественниками VeOS. Фоновый режим работы процесса — режим, когда программа может работать без взаимодействия с пользователем. В случае необходимости интерактивной работы с пользователем (в общем случае) процесс будет «остановлен» ядром, и работа его продолжается только после переведения его в «нормальный» режим работы.

Файловая система ОС

В VeOS использована файловая система Linux, которая является единым деревом. Корень этого дерева — каталог, называемый root (рут) и обозначаемый /. Части дерева файловой системы могут физически располагаться в разных разделах разных дисков или вообще на других компьютерах — для пользователя это прозрачно. Процесс присоединения файловой системы раздела к дереву называется монтированием, удаление — размонтированием. Например, файловая система CD-ROM в дистрибутиве монтируется по умолчанию в каталог /media/cdrom (путь в дистрибутиве обозначается с использованием /, а не \, как в Windows). Текущий каталог обозначается ".", родительский каталог (уровнем выше) - "..", например, для перехода в каталог верхнего уровня:

```
> cd ..
```

Структура каталогов

Корневой каталог /:

- /bin — командные оболочки (shell), основные утилиты;
- /boot — содержит ядро системы;
- /dev — псевдофайлы устройств, позволяющие работать с устройствами напрямую. Файлы в /dev создаются сервисом udev
- /etc — общесистемные конфигурационные файлы для большинства программ в системе;
- /etc/rc?.d, /etc/init.d, /etc/rc.boot, /etc/rc.d — каталоги, где расположены командные файлы, выполняемые при запуске системы или при смене её режима работы;
- /etc/passwd — база данных пользователей, в которой содержится информация об имени пользователя, его настоящем имени, личном каталоге, его зашифрованный пароль и другие данные;
- /etc/shadow — теневая база данных пользователей. При этом информация из файла /etc/passwd перемещается в /etc/shadow, который недоступен для чтения всем, кроме пользователя root. В случае использования альтернативной схемы управления теневыми паролями (ТСВ), все теневые пароли для каждого пользователя располагаются в каталоге /etc/tcb/имя пользователя/shadow;
- /home — домашние каталоги пользователей;
- /lib — содержит файлы динамических библиотек, необходимых для работы большей части приложений, и подгружаемые модули ядра;
- /lost+found — восстановленные файлы;
- /media — подключаемые носители (каталоги для монтирования файловых систем сменных устройств);
- /mnt — точки временного монтирования;
- /opt — вспомогательные пакеты;
- /opt/vasexperts - родительский каталог продуктов компании ВАСЭкспертс
- /proc — виртуальная файловая система, хранящаяся в памяти компьютера при загруженной ОС. В данном каталоге расположены самые свежие сведения обо всех процессах, запущенных на компьютере.
- /root — домашний каталог администратора системы;
- /run — файлы состояния приложений;
- /sbin — набор программ для административной работы с системой (системные утилиты);
- /selinux — виртуальная файловая система SELinux;
- /srv — виртуальные данные сервисных служб;
- /sys — файловая система, содержащая информацию о текущем состоянии системы;
- /tmp — временные файлы.
- /usr — пользовательские двоичные файлы и данные, используемые только для чтения (программы и библиотеки);
- /var — файлы для хранения изменяющихся данных (рабочие файлы программ, очереди, журналы).

Каталог /usr:

- /usr/bin — дополнительные программы для всех учетных записей;
- /usr/sbin — команды, используемые при администрировании системы и не предназначенные для размещения в файловой системе root;

- /usr/local — место, где рекомендуется размещать файлы, установленные без использования пакетных менеджеров, внутренняя организация каталогов практически такая же, как и корневого каталога;
- /usr/man — каталог, где хранятся файлы справочного руководства man;
- /usr/share — каталог для размещения общедоступных файлов большей части приложений.

Каталог /var:

- /var/log — место, где хранятся файлы аудита работы системы и приложений;
- /var/spool — каталог для хранения файлов, находящихся в очереди на обработку для того или иного процесса (очереди печати, непрочитанные или не отправленные письма, задачи cron т.д.).

Организация файловой структуры

Система домашних каталогов пользователей помогает организовывать безопасную работу пользователей в многопользовательской системе. Вне своего домашнего каталога пользователь обладает минимальными правами (обычно чтение и выполнение файлов) и не может нанести ущерб системе, например, удалив или изменив файл. Кроме файлов, созданных пользователем, в его домашнем каталоге обычно содержатся персональные конфигурационные файлы некоторых программ. Маршрут (путь) — это последовательность имён каталогов, представляющая собой путь в файловой системе к данному файлу, где каждое следующее имя отделяется от предыдущего наклонной чертой (слешем). Если название маршрута начинается со слеша, то путь в искомый файл начинается от корневого каталога всего дерева системы. В обратном случае, если название маршрута начинается непосредственно с имени файла, то путь к искомому файлу должен начаться от текущего каталога (рабочего каталога). Имя файла может содержать любые символы за исключением косой черты (/). Однако следует избегать применения в именах файлов большинства знаков препинания и непечатаемых символов. При выборе имен файлов рекомендуется ограничиться следующими символами:

- строчные и ПРОПИСНЫЕ буквы.
- символ подчеркивания (_);
- точка (.).

Имена файлов и каталогов - регистрозависимы, т.е. команды touch test.txt и touch TEST.TXT создадут два разных файла с соответствующими именами

Для удобства работы точку можно использовать для отделения имени файла от расширения файла. Данная возможность может быть необходима пользователям или некоторым программам, но не имеет значение для shell.

Имена дисков и разделов

Все физические устройства вашего компьютера отображаются в каталог /dev файловой системы дистрибутива (об этом — ниже). Диски (в том числе IDE/SATA/SCSI/SAS жёсткие диски, USB-диски) имеют имена:

- /dev/sda — первый диск;
- /dev/sdb — второй диск;

и т.д. Диски обозначаются /dev/sdX, где X — a, b, c, d, e, … в зависимости от порядкового номера диска на шине. Раздел диска обозначается числом после его имени. Например, /dev/sdb4 — четвертый раздел второго диска.

Разделы, необходимые для работы VeOS

Для работы ОС на жестком диске (дисках) должны быть созданы, по крайней мере, два раздела: корневой (то есть тот, который будет содержать каталог /) и раздел подкачки (swap). Размер последнего, как правило, составляет от однократной до двукратной величины оперативной памяти компьютера. Если на диске много свободного места, то можно создать отдельные разделы для каталогов /usr, /home, /var.

Работа с наиболее часто используемыми компонентами

Виртуальная консоль

VeOS предоставляет доступ к виртуальным консолям, с которых можно осуществлять одновременно несколько сеансов работы в системе (сессий). Переключение между первыми шести виртуальными консолями производится нажатием комбинации клавиш Alt+F1 — Alt+F6 (Ctrl+Alt+F1 — Ctrl+Alt+F6).

Командные оболочки

Для управления VeOS используются командные интерпретаторы (shell). Зайдя в систему, Вы увидите приглашение — строку, содержащую символ «$» (далее этот символ будет обозначать командную строку). Программа ожидает ваших команд. Роль командного интерпретатора — передавать ваши команды операционной системе. При помощи командных интерпретаторов можно писать небольшие программы — сценарии (скрипты). В VeOS доступны следующие командные оболочки:

- bash — Bourne Again Shell, самая распространенная оболочка
- ksh — Korn Shell, хорошо известная оболочка в UNIX™ системах.

Проверить, какая оболочка используется в данный момент можно, выполнив команду:

```
$ echo $SHELL
```

Оболочкой по умолчанию является Bash — самая распространённая оболочка под Linux, которая ведет историю команд и предоставляет возможность их редактирования.

Командная оболочка Bash

В Bash имеется несколько приемов для работы со строкой команд. Например, можно использовать следующие сочетания:

- Ctrl+A — перейти на начало строки;
- Ctrl+U — удалить текущую строку;
- Ctrl+C — остановить текущую задачу.

Для ввода нескольких команд одной строкой можно использовать разделитель «;». По истории команд можно перемещаться с помощью клавиш ↑ («вверх») и ↓ («вниз»). Чтобы найти конкретную команду в списке набранных, не пролистывая всю историю, можно нажать Ctrl+R и начать вводить символы ранее введенной команды. Дуля просмотра истории команд можно воспользоваться командой histor. Команды, присутствующие в истории, отображаются в списке пронумерованными. Чтобы запустить конкретную команду необходимо набрать:

```
> !номер команды
```

Если ввести:

```
> !!
```

запустится последняя из набранных команд.

В Bash имеется возможность подстановки имен команд из общего списка команд, что существенно облегчает работу с системой. При нажатии клавиши Tab, Bash завершает имя команды, программы или каталога, либо выводит несколько альтернативных вариантов, предлагаю пользователю ввести следующий символ и нажать клавишу Tab для уточнения выбора. Например, чтобы использовать программу декомпрессии gunzip, можно набрать следующую команду:

```
> gu
```

Затем нажать клавишу Tab. Так как в данном случае существует несколько возможных вариантов завершения команды, то необходимо повторно нажать клавишу Tab, чтобы получить список имен, начинающихся с gu. В предложенном примере можно получить следующий список:

```
> gu
guile gunzip gupnp-binding-tool
```

Если набрать: n (gunzip — это единственное имя, третьей буквой которого является «n»), а затем нажать клавишу Tab, то оболочка самостоятельно дополнит имя. Чтобы запустить команду нужно нажать Enter. Программы, вызываемые из командной строки, Bash ищет в каталогах, определяемых в системной переменной $PATH. По умолчанию в этот перечень каталогов не входит текущий каталог, обозначаемый ./ (точка слеш). Поэтому, для запуска программы из текущего каталога, необходимо использовать команду (в примере запускается команда test):

```
> ./test
```

Команда

Простейшая команда состоит из одного «слова» и может быть как встроенной в интерпретатор, так и отдельной командой, находящейся в исполняемом бинарном файле на диске. Текст, что следует за командой называется параметрами (или аргументами) и они вводятся для изменения поведения команды. В большинстве случаев, первое слово считается именем команды, а остальные — её параметрами. Поведением команд можно управлять, например, для изменения формата вывода команды ls можно использовать ключ -l.

```
> ls
> ls -l
```

Такого рода параметры называются ключами или модификаторами выполнения. Ключ принадлежит данной конкретной команде и сам по себе смысла не имеет. Этим он отличается от других параметров (например, имён файлов, чисел), имеющих собственный смысл, не зависящий ни от какой команды. Каждая команда может распознавать некоторый набор ключей и соответственно изменять своё поведение. Один и тот же ключ может определять для разных команд совершенно разные значения. Для формата ключей нет жёсткого стандарта, однако существуют договорённости:

- Если ключ начинается на -, то это простой ключ. За -, как правило, следует один символ, чаще всего буква, обозначающая действие или свойство, которое этот ключ придаёт команде. Так проще отличать ключи от других параметров.
- Если ключ начинается на --, то он называется расширенным ключом. Расширенный формат ключа начинается на два знака --, за которыми следует полное имя обозначаемого этим ключом содержания.

Некоторые ключи имеют и простой, и расширенный формат, а некоторые — только один из форматов. Информацию о ресурсах каждой команды можно получить, используя ключ --help или -h. К примеру, получить подсказку о том, что делает команда touch, можно, набрав в терминале touch --help.

Стыкование команд в системе VeOS

Режим суперпользователя

Управление пользователями

Система инициализации systemd и sysvinit