

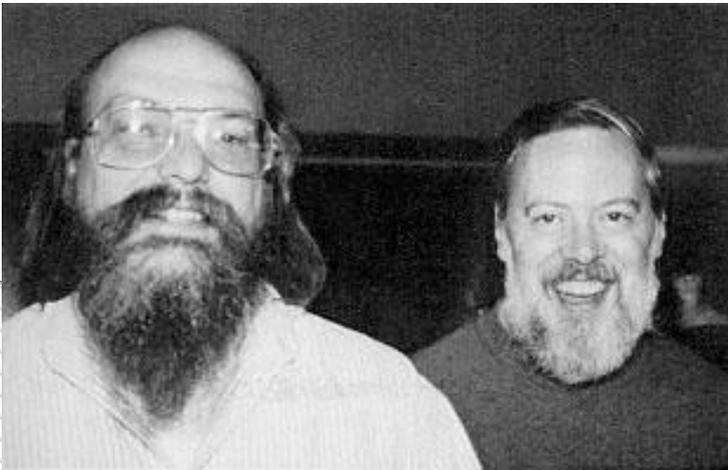
Операционная система: **UNIX**



Исторические сведения

- ▣ Первоначально UNIX была разработана в конце 1960-х годов сотрудниками Bell Labs, в первую очередь Кеном Томпсоном, Деннисом Ритчи и Дугласом Макилроем.

В 1969 году Кен Томпсон, стремясь реализовать идеи, которые были положены в основу MULTICS, но на более скромном аппаратном обеспечении (DEC PDP-7), написал первую версию новой операционной системы, а Брайан Керниган придумал для неё название — UNICS (UNIplexed Information and Computing System, рус. Примитивная информационная и вычислительная служба) — в противовес MULTICS (MULTIplexed Information and Computing Service). Позже это название сократилось до UNIX.



- В 1983 году Ричард Столлман объявил о создании проекта GNU — попытки создания свободной UNIX-подобной операционной системы с нуля, без использования оригинального исходного кода. Большая часть программного обеспечения, разработанного в рамках данного проекта, — такого как GNU toolchain, Glibc (стандартная библиотека языка Си) и Coreutils — играет ключевую роль в других свободных операционных системах. Однако, работы по созданию замены для ядра UNIX, необходимые для полного выполнения задач GNU, продвигались крайне медленно. В настоящее время GNU Hurd — попытка создать современное ядро на основе микроядерной архитектуры Mach — всё ещё далека от завершения.

Общие сведения.

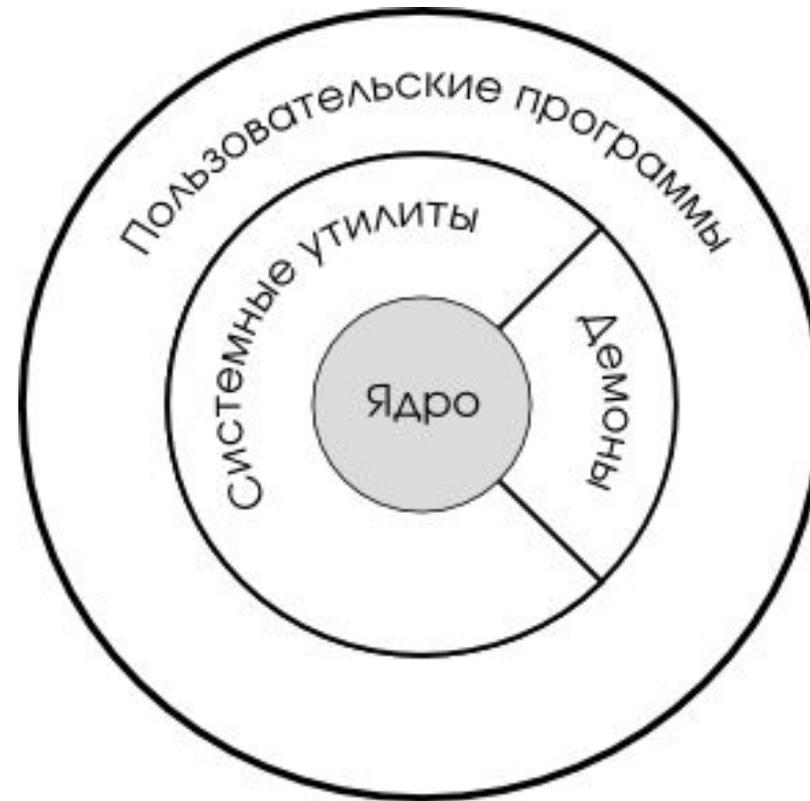
Особенности UNIX, отличающие данное семейство от других ОС:

- Файловая система древовидная, чувствительная к регистру символов в именах, очень слабые ограничения на длину имён и пути.
- Нет поддержки структурированных файлов ядром ОС, на уровне системных вызовов файл есть поток байтов.
- Командная строка находится в адресном пространстве запускаемого процесса, а не извлекается системным вызовом из процесса интерпретатора команд (как это происходит, например, в RSX-11).
- Понятие «переменных окружения».
- Запуск процессов вызовом `fork()`, то есть возможность клонирования текущего процесса со всем состоянием.
- Понятия `stdin/stdout/stderr`.
- Ввод/вывод только через дескрипторы файлов.
- Традиционно крайне слабая поддержка асинхронного ввода/вывода, по сравнению с VMS и Windows NT.

- Интерпретатор команд есть обыкновенное приложение, общающееся с ядром обыкновенными системными вызовами (в RSX-11 и VMS интерпретатор команд выполнялся как специальное приложение, специальным образом размещенное в памяти, пользующееся специальными системными вызовами, поддерживались также системные вызовы, дающие возможность приложению обращаться к своему родительскому интерпретатору команд).
- Команда командной строки есть не более чем имя файла программы, не требуется специальная регистрация и специальная разработка программ как команд (что являлось обычной практикой в RSX-11, RT-11).
- Не принят подход с программой, задающей пользователю вопросы о режимах своей работы, вместо этого используются параметры командной строки (в VMS, RSX-11, RT-11 программы работали также с командной строкой, но при её отсутствии выдавали запрос на ввод параметров).
- Пространство имён устройств на диске в каталоге /dev, поддающееся управлению администратором, в отличие от подхода Windows, где это пространство имен размещается в памяти ядра, и администрирование этого пространства (например, задание прав доступа) крайне затруднено из-за отсутствия его постоянного хранения на дисках (строится каждый раз при загрузке).
- Широкое использование текстовых файлов для хранения настроек, в отличие от двоичной базы данных настроек, как, например, в Windows.
- Широкое использование утилит обработки текста для выполнения повседневных задач под управлением скриптов.
- «Раскрутка» ОС после загрузки ядра путём исполнения скриптов стандартным интерпретатором команд.
- Широкое использование именованных каналов (pipe).
- Все процессы, кроме init, равны между собой, не бывает «специальных процессов».

- Адресное пространство делится на глобальное для всех процессов ядро и на локальную для процесса части, нет «групповой» части адресного пространства, как в VMS и Windows NT, как и возможности загрузки туда кода и его исполнения там.
- Использование двух уровней привилегий процессора вместо четырёх в VMS.
- Отказ от использования оверлеев в пользу деления программы на несколько программ поменьше, общающихся через именованные каналы или временные файлы.
- Отсутствие APC и аналогов, то есть произвольных (а не жестко перечисленных в стандартном множестве) сигналов, не доставляемых до явного пожелания процесса их получить (Windows, VMS).
- Концепция сигнала уникальна для UNIX, и крайне сложна в переносе на другие ОС, такие как Windows.

Архитектура операционной системы UNIX



Демон – это процесс, выполняющий определенную функцию в системе, который запускается при старте системы и не связан ни с одним пользовательским терминалом. Демоны предоставляют пользователям определенные сервисы, примерами которых могут служить системный журнал, веб-сервер и т.п..

ядро

Введение в администрирование UNIX
Лекция 2: Архитектура операционной системы UNIX

Ядро UNIX

Системные библиотеки

Пространство пользователя

Ядро

Интерфейс системных вызовов

Файловая подсистема

Символьные

Блочные

Драйверы устройств

Управление процессами

Планировщик

Межпроцессный обмен

Управление памятью

Аппаратно-зависимая часть

Аппаратное обеспечение

Общие инструкции по установке

- Для первой и наиболее распространенной цели вам нужны три вещи: Сам **PHP, веб-сервер и веб-браузер**. Вероятно, вы уже имеете веб-браузер и в зависимости от настроек вашей операционной системы, вы так же можете иметь и веб-сервер (например Apache в Linux и MacOS X; IIS в Windows). Так же вы можете арендовать веб-сервер или некоторое дисковое пространство на веб-сервере. В этом случае вам не нужно устанавливать дополнительное программное обеспечение, только писать PHP скрипты, загружать их на сервер, и просматривать результат их работы в браузере.

В случае установки сервера и PHP самостоятельно, у вас есть две варианта установки PHP. Для многих серверов PHP может быть установлен как модуль сервера. Это возможно для таких серверов как Apache, Microsoft Internet Information Server, Netscape и iPlanet. Если PHP не поддерживает интерфейс для вашего сервера, вы всегда можете использовать его как обработчик CGI или FastCGI. Это означает, что вы должны настроить ваш сервер так, чтобы он исполнял все PHP файлы, как CGI-скрипты.

- Если вы так же собираетесь использовать PHP в командной строке (для генерации изображений, обработки текстов и т.д.), то вам понадобится PHP CLI. Подробнее об этом можно прочитать в разделе Использование PHP в командной строке. В этом случае вам не понадобятся ни сервер ни браузер.

Установка ОС

- Есть несколько способов установки PHP на Unix платформы. Один из них - процесс конфигурирования и компиляции исходников, другой - установка из пакетов. Этот документ сфокусирован на установке из исходных кодов. Много Unix-подобных систем имеют различные системы установки программ из пакетов. Они могут помочь при установке стандартных конфигураций, но если вам необходимы другие варианты (такие как безопасный сервер или другой драйвер базы данных), вам может потребоваться скомпилировать PHP и/или ваш веб-сервер. Если вы незнакомы с компилированием собственного программного обеспечения, то, может быть, стоит поискать готовый пакет PHP с нужными вам возможностями, собранный кем-нибудь еще.

□ **Для компиляции PHP из исходных кодов вам потребуется следующее:**

- Базовые знания Unix (способность обращаться с "make" и компилятором C)
- Компилятор ANSI C
- Веб-сервер
- Любые компоненты, специфичные для отдельных модулей PHP (такие как библиотеки GD, PDF и т.д.)

При сборке непосредственно из исходников Git или после ручных изменений вам может также понадобиться:

- autconf: 2.13+ (для PHP < 5.4.0), 2.59+ (для PHP >= 5.4.0)
- automake: 1.4+
- libtool: 1.4.x+ (кроме версии 1.4.2)
- re2c: Версия 0.13.4 или новее
- flex: Версия 2.5.4 (для PHP <= 5.2)
- bison: Версия 1.28 (предпочтительно), 1.35 или 1.75

- Общая (или начальная) конфигурация РНР задается параметрами скрипта **configure**. Вы можете просмотреть список допустимых параметров вместе с их кратким пояснением при помощи команды **./configure --help**. Различные опции документированы в данном руководстве отдельно, список основных параметров можно просмотреть в приложении Основные параметры конфигурации, тогда как параметры, специфичные для различных расширений, описаны на страницах документации, посвященных этим расширениям.
- Когда РНР сконфигурирован, все готово к сборке модулей и/или исполняемых файлов. Об этом должна позаботиться команда **make**.

досистемный загрузчик

Как правило, сразу после включения питания программа ПЗУ BIOS проводит тестирование оборудования, затем запускается досистемный загрузчик.

Задача этого этапа – определить (возможно, с помощью пользователя), с какого устройства будет идти загрузка, загрузить оттуда специальную программу-загрузчик и запустить её. Например, выяснить, что устройство для загрузки – жесткий диск, считать самый первый сектор этого диска и передать управление программе, которая находится в считанной области.

загрузчик первого уровня

Загрузчик первого уровня занимает обычно не более одного сектора в самом начале диска – в его загрузочном секторе (Master Boot Record).

Ядро операционной системы имеет довольно сложную структуру – а значит, и непростой способ загрузки; оно может быть довольно большим, и может располагаться неизвестно где на диске, подчиняясь законам файловой системы (например, состоять из нескольких частей, разбросанных по диску). Учесть все это первичный загрузчик не в состоянии, поэтому его задача: определить, где на диске находится загрузчик второго уровня, загрузить его в память и передать ему управление.

загрузчик второго уровня



загрузчик второго уровня

Вторичный загрузчик – сложная программа с интерфейсом пользователя, который даёт возможность выбирать операционную систему или параметры загрузки ядра. Загрузчик должен иметь доступ к образу ядра, поэтому в него включается поддержка файловых систем.

Первым делом ядро занимается определением параметров вычислительной подсистемы компьютера: выясняет тип и быстродействие центрального процессора, объем оперативной памяти, объем и структуру кэш-памяти; делает предположение об архитектуре компьютера в целом и многое другое.

На следующем шаге ядро определяет состав и архитектуру всего аппаратного наполнения компьютера: тип и параметры шин передачи данных и устройств управления ими (контроллеров), список внешних устройств, доступных по шинам, настройки этих устройств – диапазон портов ввода-вывода, адрес ПЗУ, занимаемое аппаратное прерывание, номер канала прямого доступа к памяти и т.п.. Ядро на основании переданного ему параметра выбирает корневой раздел – файловую систему, содержащую будущий каталог / и его подкаталоги (для системной начальной загрузки важны каталоги /etc, /bin, и /sbin). Корневой раздел монтируется в качестве /. После этого ядро запускает свой первый процесс – init (по умолчанию, /sbin/init).

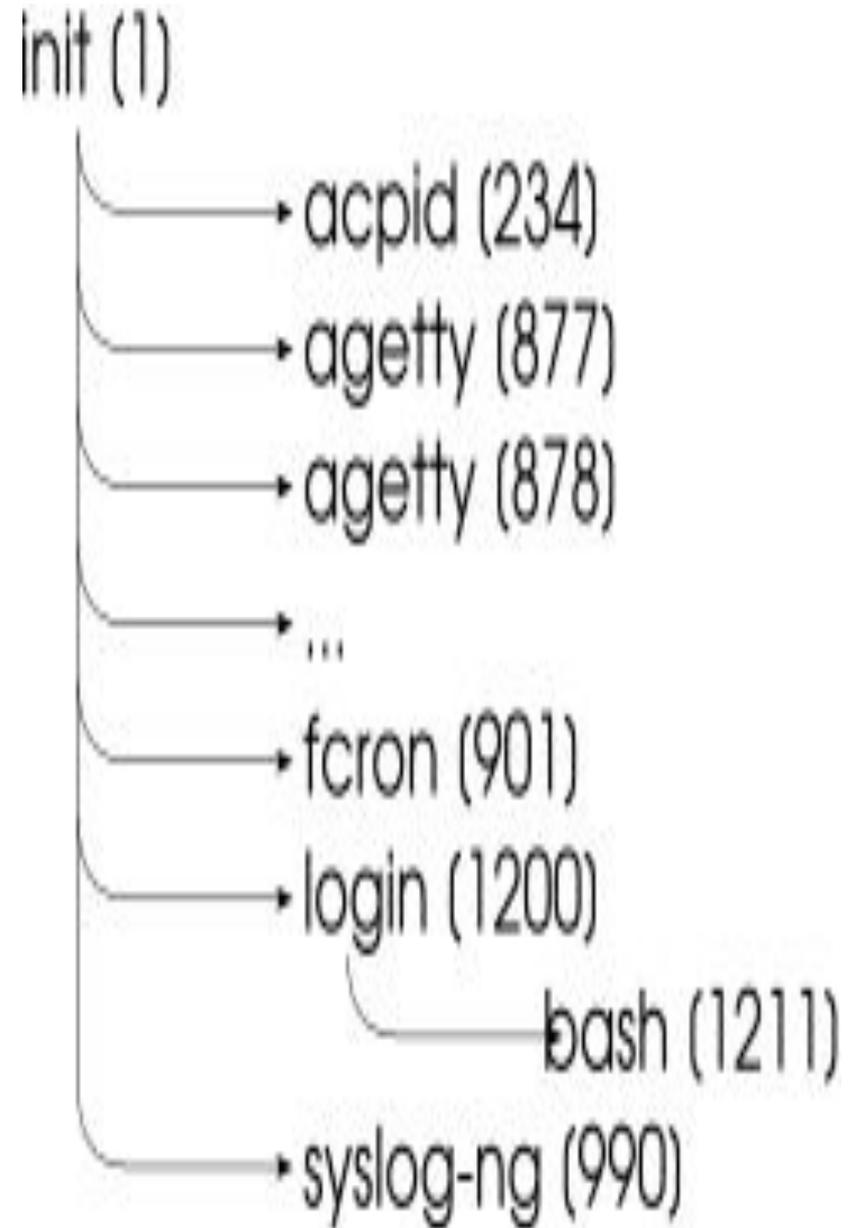
▣ процесс **init**

С этого момента операционная система обеспечивает полноценную функциональность всем исполняющимся процессам. В UNIX первым запускаемым процессом является init, о котором сказано в следующем разделе.

□ **Процесс `init`** является обычным процессом операционной системы, однако он имеет некоторые особенности: его PID всегда равен 1, и процесс этот работает всё время, пока работает система.

В UNIX-системах `init` играет две важные роли:

1. производит инициализацию системы – как правило, для работы запущенного ядра не достаточно, нужно примонтировать все файловые системы, загрузить дополнительные драйверы устройств, запустить демоны и т. п.;
2. является родительским для всех процессов в системе – это является гарантией того, что в UNIX любой процесс имеет своего родителя.



- Исторически, различные версии UNIX наследовались от двух систем: оригинальной UNIX компании AT&T (вплоть до версии System V) и BSD UNIX, созданной в университете Беркли. В них применялись различные принципы загрузки системы, так что современные версии UNIX по этому критерию можно разделить на:
- наследники System V – так называемая UNIX System Group (USG-системы): AIX, Solaris, UnixWare, Linux (дистрибутивы RedHat, Mandriva, ALT Linux);
- наследники BSD: семейство BSD, Linux (Slackware);
- смешанные схемы: Linux (Debian, Gentoo).

Различия загрузки UNIX-подобных систем

Системы, наследующие System V

- Основным признаком этих систем является наличие уровня выполнения (run level) – одного из возможных режимов работы системы. Каждый уровень исполнения имеет свой номер – часть этих номеров стандартизована. В любой момент времени система может находиться в одном из них – изменение режима работы производится с помощью перезапуска init с указанным номером.

- **Уровень 0**

остановка системы (halt) – работа системы должна быть прекращена;

- **Уровень 1**

однопользовательский режим работы – система инициализирует минимум служб и даёт единственному пользователю (как правило, суперпользователю) без проведения аутентификации командную строку. Как правило, этот режим используется для восстановления системы;

- **Уровень 2**

многопользовательский режим – пользователи могут работать на разных терминалах, вход в систему с процессом аутентификации;

- **Уровень 3**

многопользовательский сетевой режим – в отличие от предыдущего уровня, осуществляется настройка сети и запускаются различные сетевые службы;

- **Уровень 4**

не имеет стандартного толкования и практически не используется;

- **Уровень 5**

запуск графической подсистемы – по сравнению с уровнем 3 производится также старт графической подсистемы X11, и вход в систему осуществляется уже в графическом режиме;

- **Уровень 6**

перезагрузка системы – при включении этого режима останавливаются все запущенные программы и производится перезагрузка.

Системы, наследующие BSD

- В этих системах традиционно используется линейная схема загрузки. Эта схема устроена намного проще (загрузка таких систем проходит намного быстрее – особенно на медленных машинах), но зато более сложная в администрировании. Инициализация системы осуществляется единым скриптом `/etc/rc`. В этом скрипте прописаны последовательные команды инициализации системы, запуска демонов и т.п.. Следом за ним следует запуск скрипта `/etc/rc.local`, который служит для запуска всех локальных программ и настроек, установленных системным администратором сверх дистрибутива операционной системы.
- При обновлении отдельных программ или изменении их настроек администратору приходится вручную править стартовые скрипты. Эти сложности привели к тому, что в современные BSD-системы внедряются более легкие в администрировании схемы

Конфигурация запуска init

□ Конигурация процесса init находится в файле /etc/inittab:

Default runlevel.

□ **id:3:initdefault:**

System initialization, mount local filesystems, etc.

□ **si::sysinit:/sbin/rc sysinit**

Further system initialization, brings up the boot runlevel.

□ **rc::bootwait:/sbin/rc boot**

l0:0:wait:/sbin/rc shutdown

□ **l1:S1:wait:/sbin/rc single**

□ **l2:2:wait:/sbin/rc nonetwork**

□ **l3:3:wait:/sbin/rc default**

□ **l4:4:wait:/sbin/rc default**

□ **l5:5:wait:/sbin/rc default**

□ **l6:6:wait:/sbin/rc reboot**

TERMINALS

□ **c1:12345:respawn:/sbin/agetty 38400 tty1 linux**

□ **c2:2345:respawn:/sbin/agetty 38400 tty2 linux**

□ **c3:2345:respawn:/sbin/agetty 38400 tty3 linux**

□ **c4:2345:respawn:/sbin/agetty 38400 tty4 linux**

□ **c5:2345:respawn:/sbin/agetty 38400 tty5 linux**

□ **c6:2345:respawn:/sbin/agetty 38400 tty6 linux**

□ **# What to do at the "Three Finger Salute".**

□ **ca:12345:ctrlaltdel:/sbin/shutdown -r now**

Used by /etc/init.d/xdm to control DM startup.

□ **xa:once:/etc/X11/startDM.sh**

□ Основными параметрами загрузки, задаваемыми в этом файле, являются:

начальный уровень загрузки (строка с `initdefault`) – номер уровня выполнения, в который переводится система при старте;

скрипты для запуска уровней исполнения – для каждого уровня (0 – 6) задаётся программа с аргументами запуска, которая будет запущена в случае перевода системы на один из уровней выполнения;

настройки виртуальных терминалов – сколько необходимо инициализировать при старте системы, какую программу для этого использовать;

настройка реакции на нажатие `Ctrl-Alt-Del` – какую программу необходимо запустить при этом;

прочие настройки, специфичные для данной версии UNIX.

Системные службы

- ▣ **Системные службы** – это программы, выполняющие в системе определённую функцию и, как правило, стартующие при запуске системы. Каждой системной службе соответствует **стартовый скрипт** – специальная программа, осуществляющая запуск демона или программы, которая и обеспечивает функциональность службы.

Таким образом, можно построить **дерево зависимостей**, и перезапуск одного скрипта будет приводить к перезапуску всех его потомков.

Автоматическая загрузка служб

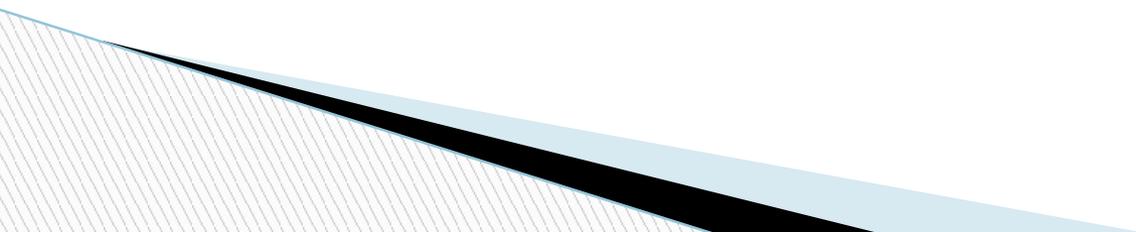
- ▣ Службы используются в UNIX-системах, использующих **System V**-подобную схему загрузки системы. При этом каждому уровню выполнения соответствует набор служб, запускаемых при переключении на этот уровень.

В директории `/etc/rc.d/` можно увидеть директории `rc0.d`, `rc1.d` и т.д. – по одной на каждый уровень выполнения. В этих директориях содержатся ссылки на службы, которые будут запущены или остановлены при переходе на соответствующий уровень выполнения.

Особый интерес представляют имена ссылок на стартовые скрипты служб: например, `/etc/rc.d/rc0.d/K60crond` и `/etc/rc.d/rc3.d/S40crond`, указывающие на **один скрипт `/etc/init.d/crond`** службы системного журнала. Скрипт, начинающийся с «K» соответствует останову службы, а «S» – запуску. Числа, следующие перед именем службы задают порядок запуска скриптов в директории. Например, скрипт `/etc/rc.d/rc3.d/S34syslogd` будет запущен до скрипта `/etc/rc.d/rc3.d/S40crond`, тогда как `/etc/rc.d/rc3.d/K60crond` до `/etc/rc.d/rc3.d/K66syslogd`. Можно заметить, что сумма чисел для одной службы равна 100 – это позволяет упорядочить все скрипты в порядке старта, обратном порядку завершения.

Для установления связи между службами и уровнями выполнения используется утилита **chkconfig**.

Настройка ОС

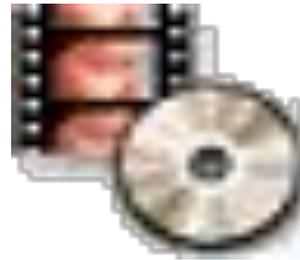


ОС и Мультимедия

- ▣ **VLC media player** - кросс-платформенный медиаплеер и сервер медиапотока.
- ▣ Поддерживается работа с большинством видеоформатов, включая DVD, VCD, MPEG, DivX и потоковые (Nullsoft Streaming Video (.nsv) и Real Media (.rm)). Есть возможность использовать программу как обыкновенный медиаплеер.S

Audacity - отличный аудиоредактор с большими возможностями, особенно при подключении плагинов. Версия для Linux/Unix

Totem 2.21.1 - Небольшой медиаплеер для GNOME, базирующийся на xine или GStreamer. Проигрывание аудиофайлов в Totem может сопровождаться красивыми визуальными эффектами; что касается видео, то программа понимает и DVD, и VCD, и Mpeg4. Поддерживаются плейлисты, полноэкранный режим, субтитры, настройка параметров изображения и многое другое.



Прикладные программы общего назначения

Текстовые редакторы

joe: минималистский консольный редактор

Редактор joe обладает полноценным интерфейсом к консольном режиме. Для открытия файла на редактирования можно запустить команду **joe имя_файла**. Текущая (3-я) версия редактора имеет уже достаточно широкие возможности – это: подсветка синтаксиса, открытие для редактирования нескольких файлов, запуск внутренние команды, использование комбинации клавиш и т.д.. Редактор **joe** имеет ряд встроенных команд. При этом joe может быть запущен в качестве потокового редактора, когда на стандартный вход подаётся последовательность команд по изменению или созданию текста.

Для вычисления простых математических функций и выражений используется встроенный калькулятор, вызываемый по нажатию **ESC-M**.

vi: универсальный текстовый редактор

Редактор vi является одним из самых распространённых текстовых редакторов для различных версий UNIX.

Администратору необходимо овладеть хотя бы основами этого редактора, так как найти его можно практически в любой UNIX-системе.

Как и с другими экранными редакторами, чтобы открыть файл, достаточно набрать **vi имя_файла**.

Основной особенностью редактора vi является наличие **режимов работы**. Основным режимом является режим команд, в котором нажатия на клавиши задают команду редактору. Для реактирования текста удобно применять режим вставки, в котором курсор может перемещаться по редактируемому тексту, вставлять или удалять символы.

В начале работы редактор находится в командном режиме, а для перехода в режим вставки необходимо ввести команду **i (insert)** или **a (append)** – для вставки текста перед и после курсора соответственно. Для возвращения в командный режим используется клавиша **ESC**.

Программы сетевого обмена

sendmail: программа отправления почты

Программа sendmail является частью одноимённого SMTP-сервера и позволяет отправлять письма электронной почты. Однако, многие другие распространённые почтовые серверы или (например, Postfix или ssmtp) программы предоставляют совместимый с sendmail интерфейс.

По умолчанию sendmail принимает на стандартный ввод тело письма вплоть до получения EOF. Таким образом, при работе в командной строке для завершения ввода текста необходимо нажать Ctrl-D.

Для простейшего использования программы достаточно ключа -t. При этом программа sendmail проанализирует заголовок письма и выберет из него имена и адреса отправителя и получателя письма.

wget: консольный загрузчик файлов

wget – универсальная программа загрузки файлов по протоколам HTTP, HTTPS и FTP. При работе с HTML/XML-файлами программа может находить ссылки и переходить по ним, получая полноценную локальную версию сайта.

Команда имеет простой формат:

- ▣ **wget [опции] URL.**
- ▣ **Загруженный файл сохраняется в локальной директории**

Работа В Интернете

- ▣ **Mozilla Firefox** — свободно распространяемый браузер. Второй по популярности браузер в мире. В браузере присутствуют вкладочный интерфейс, проверка орфографии, поиск по мере набора, «живые закладки», менеджер зачек, поисковая система. Новые функции можно добавлять при помощи расширений. Firefox выпускается для Microsoft Windows, BeOS, Mac OS X, Linux и множества других Unix-подобных операционных систем.
- ▣ **Konqueror (Конкерор, от KDE + conqueror)** — веб-браузер и файловый менеджер, являющийся ключевым компонентом графической среды KDE (свободная среда рабочего стола для UNIX-подобных операционных систем). Для вывода HTML использует собственный движок KHTML.
- ▣ **Dillo** — браузер, целью его авторов было создание быстрого браузера — как Lynx, но с поддержкой просмотра графических изображений. Главные изменения коснулись интерфейса - он был полностью переписан с GTK1 на FLTK2, что позволило сократить вдвое требования к памяти, а также реализовать поддержку UTF-8. Dillo доступен для разных Unix-платформ, таких как GNU/Linux, BSD, Mac OS X.и другие...