

Лекция 1. Для начинающих работать в UNIX

Эта книга рассчитана на тех, кто уже представляет себе, что такое операционная система. Более того, мы полагаем, что читатель знаком с какой-нибудь системой, например с Windows. Сейчас, когда российская пресса полна публикаций о борьбе с пиратством и установке бесплатной Linux в учреждениях образования и в корпорациях вместо пиратской Windows, было бы естественно ожидать, что количество опытных и знакомых с UNIX людей уже достаточно велико. На самом деле и сейчас находится немало людей, которые только начинают думать о переходе на системы UNIX.

Именно для тех, кто еще не знаком с этим семейством систем, написана первая глава. Если вы уже имели дело с Linux, FreeBSD, QNX, Solaris или с любой другой системой UNIX, перелистните несколько страниц — или прочтите главу по диагонали, если любите чувствовать, что читаете давно знакомые мысли и формулировки, живущие в вашей голове уже не первый год, месяц или день.

Открытые системы и их преимущества

Возможно, вам доводилось бывать в пивных ресторанах, где процесс пивоварения можно наблюдать из обеденного зала, глядя сквозь прозрачную стену на медные трубы и чаны с будущим пивом. Конечно, никто при этом не дает вам гарантии, что вы пьете свежее пиво, сваренное именно в этих чанах, но по крайней мере вы вольны верить созданной иллюзии. В маленьком ресторанчике в Хейдельберге, где варят самое плотное пиво в мире, эта иллюзия в самом деле воплощена в реальности: сам смотришь, как пиво из чана переливается в бочки, канистры и затем — в пивные кружки посетителей.

Первое, самое очевидное преимущество открытых систем (то есть систем, исходный код которых доступен публике) состоит в том же самом: каждый может взять исходный код продукта, откомпилировать его самостоятельно и получить операционную систему — такую как ему хотелось. В любой момент вы можете просмотреть этот исходный код и убедиться, что в нем нет вредоносных «закладок», которые на официальном языке называются «недокументированными возможностями». Если вы заподозрили, что система работает с ошибками, вы сами можете обнаружить источник ошибки в исходном коде.

Второе преимущество открытой системы в том, что она бесплатна. Нам неизвестны продукты с открытым кодом, которые бы продавались за деньги. С этим связано и третье преимущество: так как производитель

системы должен на чем-то зарабатывать деньги, то он обычно предлагает удобную и качественную поддержку. Качество поддержки в этом случае может быть выше, чем у тех систем, которые сами по себе продаются за деньги, так как разработчики, открывающие код, живут только поддержкой, и снижение ее качества – верное самоубийство для них. Кстати, качество кода – это еще один из плюсов открытых систем, потому что разработчику хочется признания коллег, а не насмешек, и код часто оказывается эффективным, понятным и хорошо откомментированным.

Четвертое преимущество открытых систем – широкое тестирование и быстрота обновлений. Вокруг каждого открытого продукта образуется не только группа разработчиков, но и сообщество пользователей, которые после выхода свежей версии быстро обнаруживают ошибки и рассказывают о них разработчикам, а те быстро исправляют ошибки и выпускают обновления.

Есть и пятое преимущество – сертифицировать открытый код проще, ибо не требуется никаких согласований для передачи его в сертифицирующую организацию.

Если вы еще не стали убежденным сторонником открытого кода к этому абзацу, попытаемся впечатлить вас тем, что подавляющее большинство открытых систем не подвержены заражению существующими компьютерными вирусами. Это не значит, что открытость системы спасает ее от вторжения, это лишь подчеркивает тот факт, что пока количество вирусов для открытых программ невелико, а безопасность этих программ разработчики поддерживают на высоком уровне.

Вполне естественно, что небольшие группы – разработчики программного обеспечения, получающие основной доход от продажи плодов своего труда, могут относиться негативно к идее открытого кода: что же тогда будет приносить им доход, если открывать код всем желающим? Одному из нас пришлось наблюдать постепенное «просветление» такой группы разработчиков, которой на переговорах с цифрами в руках показали, что количество пользователей открытых программ растет лавинообразно и доход от технической поддержки быстрорастущей массы пользователей с лихвой покрывает отсутствие выручки за продажу самого программного обеспечения.

К 2008 году во всем мире сформировалось две новых бизнес-модели, связанных с информационными технологиями, – бизнес технической поддержки открытых решений (типа поддержки Solaris компанией Sun Microsystems) и предоставление платного доступа к удаленным службам (типа Google Documents или Google Maps). В следующих главах мы расскажем о работе с системой Solaris – одной из ярких иллюстраций успешности, новизны и динамики современного рынка IT.

Краткая история семейства UNIX

Общая история

Первая версия многозадачной многопользовательской системы UNIX была разработана в 1969 году сотрудниками подразделения Bell Labs фирмы AT&T Деннисом Ритчи (Dennis Ritchie) и Кеном Томпсоном (Ken Thompson). Она была написана на ассемблере для компьютера PDP-7. К лету 1972 года система была переписана на язык высокого уровня C, что позволяло перенести ее на любую аппаратную платформу. Создатели UNIX впервые сделали операционную систему, полностью написанную на языке высокого уровня.

Первым компьютером, которым управляла новая версия системы, стал компьютер PDP-11 компании Digital Equipment Corporation (DEC). Несколько десятилетий спустя, в 90-е годы XX века, DEC была переименована в Digital, затем поглощена компанией Compaq, которая, в свою очередь, слилась с Hewlett Packard.

В начале 70-х годов XX века вдохновленные новыми идеями UNIX выпускники Университета в Беркли (University of California – Berkeley) начали работу над собственной версией UNIX.

В 1977 году вышла первая версия Berkeley Software Distribution (BSD) – ныне широко известной версии UNIX. Так как эта система была основана на коде AT&T, для ее использования была необходима лицензия от AT&T. Разработчики BSD, подстегиваемые ростом цен на лицензии, полностью переписали исходный код системы к 1991 году.

После этого разработки UNIX в мире пошли в двух направлениях: одни команды разработчиков взяли за основу версию Беркли, другие – версию System V Release 4 (часто это название сокращают до SVR4 и обычно произносят как «систэм файв релиз четыре»), которая была разработана AT&T.

Этот краткий экскурс в историю UNIX необходим нам для того, чтобы понимать, отчего все диалекты UNIX имеют много схожего, но некоторые отличия между разными вариантами систем иногда удивляют или раздражают нас.

Новейшая история

Solaris относится к ветви System V. Это означает, что в этой системе приняты стандартные для указанной ветви структура каталогов с файлами конфигурации, порядок загрузки ОС и системные утилиты, а также их ключи.

С другой стороны, в Solaris используются некоторые весьма специфические именно для этой системы соглашения, которые не имеют отношения ни к System V, ни к BSD, поэтому такие случаи будут оговорены особо, чтобы не дезориентировать тех читателей, которые уже имеют опыт работы с другими UNIX'ами.

В настоящее время широкое распространение получили несколько систем UNIX: Solaris, FreeBSD, Linux, QNX и ряд других, включая основанную на UNIX систему Mac OS X от Apple. Все системы Linux основаны на ядре, которое разрабатывается большой командой разработчиков под общим руководством Линуса Торвальдса (Linus Torvalds). Различные дистрибутивы Linux используют ядро, которое собрано из одних и тех же исходных кодов, доступных на сайте linux.org. Таким образом, отличия Red Hat Linux от Slackware Linux или Debian Linux связаны скорее с комплектованием дистрибутива и работой различных утилит, а ядро одной и той же версии в любой системе Linux функционирует одинаково.

В то же время системы UNIX, отличные от Linux, такие, как Solaris, FreeBSD и другие, имеют схожую архитектуру, очень похожие системные команды и их ключи, но могут незначительно отличаться друг от друга. Рассматриваемые ниже текстовый и графические интерфейсы присутствуют во многих системах UNIX, так как все эти системы совместимы по исходным кодам и отвечают стандарту POSIX. Поэтому, если ваша работа в UNIX не связана с администрированием системы и ограничивается, скажем, набором текстов или редактированием фотографий, не исключено, что перехода от FreeBSD к Solaris вы можете и не заметить.

Новейшая история Solaris

Говоря о Solaris, надо четко обозначить тот факт, что есть несколько дистрибутивов Solaris: Solaris, Solaris Express, OpenSolaris (выпускаемые и поддерживаемые Sun Microsystems) и основанные на открытом коде Solaris системы Nexenta, Belenix, Schillix, marTux и Milax, которые с Sun не связаны. Мы не будем касаться последних пяти в этой книге; если вам интересно больше узнать об этих системах, следуйте ссылкам с сайта opensolaris.org.

Solaris Express — это дистрибутив, предназначенный для ознакомления разработчиков и других потенциальных пользователей Solaris с новыми технологиями, которые проходят тестирование перед их включением в основной дистрибутив Solaris. Можно использовать Solaris Express как для тестирования стабильности самой системы, так и для освоения новых технологий, пока еще не включенных в основную версию Solaris (сейчас ее новейшая версия — Solaris 10 update 5, или, что то же самое, Solaris 10 05/08). Система Solaris Express доступна для скачивания с сайта

opensolaris.org, установочный DVD-диск с ней можно заказать по почте (в некоторых странах, включая Россию, бесплатно, подробности на сайте developers.sun.ru). Solaris Express выпускался в двух реализациях – Solaris Express Community Edition, SXCE и Solaris Express Developer Edition, SXDE. Первый выходил раз в две недели, второй – раз в три месяца (разница была в том, что в SXDE вносили только существенные улучшения, а в SXCE включали все сделанные улучшения к моменту выпуска. С мая 2008 года компания Sun перестала распространять Solaris Express, и теперь раз в полгода будет выходить очередная версия OpenSolaris. Дистрибутивы SXDE и OpenSolaris доступны только для платформы x86, в 2008 году предполагается выпустить OpenSolaris для SPARC.

Платную техническую поддержку и полный спектр обновлений Sun Microsystems предоставляет для Solaris и OpenSolaris, а ограниченную поддержку – для Solaris Express Developer Edition. Также важно заметить, что любая версия Solaris бесплатна для любого использования (коммерческого или некоммерческого).

Какую систему UNIX выбрать?

Каждый из нас хотя бы раз в жизни задавал себе этот вопрос. Если вы еще не делали это, попробуйте прямо сейчас!

Разумеется, ответ на него зависит от задачи, которую вы собираетесь решать. Если вам только что подарили Mac Book Pro, то весьма вероятно, что ваше решение на сегодня – Mac OS X, потому что именно она там уже установлена и прекрасно подходит для очень многих задач.

Если требуется спроектировать вычислительный кластер из нескольких тысяч узлов, вы можете выбрать Solaris (и проверить, действительно ли Sun Grid N1 такое удобное средство управления распределенной вычислительной сетью, как говорят рекламные проспекты).

Если ваша задача – обеспечивать хостинг двух десятков веб-сайтов своих друзей, с ней легко справится практически любая система UNIX, хотя выбор Solaris в этом случае может дополнить вашу затею высочайшей масштабируемостью (если количество друзей резко вырастет и сайты себе захотят сделать все ваши знакомые из vkontakte.ru).

В сущности, любая система UNIX на достаточно мощном оборудовании может решить любую современную задачу – от поддержки почтового сервера организации до файлового сервера большого университета; прочие сетевые службы, равно как и управление технологическими процессами, также проблемы не представят.

Фактически, сейчас при выборе операционной системы следует исходить из следующего:

1. какое оборудование требуется для решения задачи;

2. каковы специфические условия задачи (например, требуется частое резервное копирование больших объемов данных, или работа в режиме реального времени, или поддержка нестандартных внешних устройств, или совместимость со строго определенным программным обеспечением – скажем, SAP R/3);
3. есть ли необходимость и возможность получать квалифицированную техническую поддержку при установке и эксплуатации системы;
4. есть ли персонал (или возможность его обучения), способный управлять системой.

Иногда необходимое оборудование сразу определяет выбор системы. Например, ОС Solaris является единственной эффективной системой для управления серверами на платформе SPARC, которые наилучшим образом подходят для решения ряда задач, а иногда уже просто интегрированы в решение более высокого уровня (систему автоматизации расчетов, оборудование телекоммуникационной станции, систему управления производством и т. п.) Некоторые из них можно заставить работать и под Linux, но в большинстве случаев этого просто не требуется.

Как правило, Solaris имеет хорошие шансы везде, где необходима масштабируемость и распространение операционной системы на различные аппаратные платформы или требуется унификация систем, установленных на разном оборудовании – от лэптопов до мощных серверов, так как Solaris есть и для SPARC, и для x86, и работает на оборудовании любого масштаба.

С другой стороны, системный администратор или менеджер может выбрать систему раз в жизни, в самом начале профессиональной карьеры, и затем следовать этому выбору в силу привычки, а также наработанных и теперь ставших родными приемов работы. Такой выбор нередко обусловливается самыми иррациональными причинами – например тем, какую систему уже установил более опытный приятель. Ибо приятелю предстоит отвечать первый месяц или год на все вопросы новоявленного администратора UNIX.

Как ни странно, выбор «по примеру приятеля» может оказаться самым верным. Хорошая техническая поддержка и разумный совет – это главное подспорье для новичка в деле администрирования систем. Если у вас есть друг или коллега, готовые делиться с вами знаниями днем и ночью, стоит поинтересоваться их мнением, какую систему устанавливать. Даже если их совет окажется неоптимальным с точки зрения соответствия системы вашим задачам, вы первое время будете избавлены от мук одиночества в часы затруднений и ночных бдений у монитора.

Если проблема технической поддержки для вас – не главное, начинать делать выбор придется «от Адама». Прежде всего следует определить, какую функцию будет выполнять тот компьютер, на который надо уста-

новить новую систему. После определения функции станут ясны требования к оборудованию, к программному обеспечению. Обычно на этом этапе контуры будущей системы уже вырисовываются.

Традиционно в России в конце XX – начале XXI века коммуникационные сетевые серверы работали под FreeBSD, серверы приложений (web-серверы и т. п.) – под Linux, а везде, где требовалась большая мощность (скажем, в крупных серверах баз данных), стремились устанавливать Solaris. Сейчас ситуация изменилась, и Solaris готов работать во всех перечисленных областях, конкурируя и с FreeBSD, и с Linux, занимая также и нишу рабочих мест обычных офисных пользователей. От того, как близко и умело вы познакомитесь с Solaris, зависит, будет ли задействован в полной мере его потенциал для решения всех этих задач.

Свое мнение о том, насколько эта книга помогла вам освоиться с Solaris, вы можете высказать на форумах developers.sun.ru – чтобы ее следующая редакция оказалась еще более информативной.

Быстрый взгляд внутрь UNIX

В любой системе UNIX есть *ядро, системные библиотеки, системные службы и служебные утилиты*. Solaris в этом смысле подобен другим системам UNIX. Solaris – это многоплатформенная ОС, работающая с архитектурами SPARC (это – 64-разрядная архитектура) и x86 (32- и 64-разрядные). Эта система разработана компанией Sun Microsystems; история системы началась в 1992 году, когда Sun объявила о создании принципиально новой версии системы Sun OS, которая получила имя Solaris.

Исполняемый код в Solaris работает в одном из двух пространств – пространстве ядра или пространстве пользователя. В пространстве ядра задействовано только ядро и часть кода функций системных библиотек. Код, функционирующий в пространстве пользователя, не имеет доступа к памяти, отнесенной к пространству ядра, чем гарантируется стабильность работы системы: никакое пользовательское приложение не может ее нарушить.

Ядро – это набор системных таблиц и подпрограмм работы с ними. В ядро также входят драйверы устройств: небольшие программы, обеспечивающие системе интерфейс для общения с аппаратурой компьютера – дисками, видеоподсистемой, сетевыми интерфейсами и т. п. Ядро состоит из статической части, которая загружается при старте системы и модулей. Модули могут динамически загружаться при старте системы или во время работы, при необходимости поддержки той или иной функции. В частности, подсистема поддержки NFS и драйверы внешних устройств оформлены в виде модулей.

Основная часть ядра Solaris состоит из двух компонент – *genunix* и *unix*. Это статическая часть кода ядра. Независимый от платформы файл *genunix* содержит общие для всех ядер Solaris компоненты, а файл *unix* – это зависимый от платформы файл. Специфичный для платформы файл называется `/platform/`uname -i`/kernel/unix`. Конструкция «обратные апострофы» (`` ``) обозначает, что часть названия (а именно название аппаратной платформы), можно узнать по команде `uname -i` (в ранних версиях Solaris рекомендовалось использовать `uname -m`).

В файле конфигурации ядра `/etc/system` можно установить значение переменной `moddir`, которая содержит список каталогов, разделенных пробелом. Предполагается, что в этих каталогах находятся загружаемые модули ядра. Минимальное значение по умолчанию

```
/kernel /usr/kernel
```

но это значение может быть различным для различных платформ. Наиболее частым значением для большинства платформ является

```
/platform/<platform-name>/kernel /platform/<hardware-class-name>/  
kernel /kernel /usr/kernel
```

Конкретное значение `platform-name` сообщает команда

```
uname -i
```

Значение `hardware-class-name` сообщает

```
uname -p
```

Файл конфигурации ядра в Solaris – `/etc/system` (для получения детальной информации о нем следует ознакомиться со страницей руководства `system(4)`), это можно сделать командой

```
man -s 4 system
```

В системах, работающих на процессорах SPARC, файл *unix* называется `/platform/`uname -i`/kernel/sparcv9/unix`. Когда загрузчик поместит оба этих файла (*genunix* и *unix*) в память, они становятся одним целым – ядром. Затем это ядро загружает модули.

Подробнее о процессе загрузки рассказывает лекция 11.

Системные службы – это серверные приложения или их группы, отвечающие за обработку запросов от других (клиентских) программ.

В системах UNIX такое серверное приложение обычно называют демонами (или, если произносить это вернее, даймонами – daemons).

Служебные утилиты – это программы, которые нужны для выполнения разных базовых работ в системе: копирования файлов, управления процессами, назначения прав доступа и т. п.

В разных системах UNIX используются разные файловые системы, причем часто система поддерживает несколько разных файловых систем. Например, Linux умеет работать с ext2, ext3 (это ее родные файловые системы), UFS, HPFS, NTFS, FAT и другими. Solaris, как и некоторые другие ОС UNIX, использует файловую систему типа UFS. В Solaris 10 впервые в мире была реализована поддержка новой транзакционной 128-разрядной файловой системы ZFS. О системе ZFS более подробно рассказано в лекции 8.

UNIX – это многопользовательская многопроцессная система, т. е. в ней может одновременно быть запущено несколько процессов от имени разных пользователей. Число процессов, которые можно одновременно запустить, ограничивается размерами таблицы процессов в ядре и другими настройками ядра.

Каждый процесс в UNIX работает в своем собственном адресном пространстве, поэтому сбой в работе одного процесса никак не влияет на работу других. Подсистема виртуальной памяти, являющаяся частью ядра, запрещает процессам обращаться к чужим адресным пространствам.

Одной из функций ядра является планирование процессов, т. е. передача управления от одного процесса к другому. Для этого в ядре есть отдельная подпрограмма, называемая *планировщиком процессов* или *планировщиком задач*. Процессы получают управление от планировщика задач в соответствии со своим приоритетом. Планировщик задач через определенное количество микросекунд решает, следует ли передать управление следующему в очереди процессу.

Ядро представляет собой отдельный процесс, выполняющийся с наивысшим приоритетом. В большинстве систем UNIX, включая Solaris, реализована возможность выполнения несколько параллельных подпроцессов внутри процесса. Эти подпроцессы называют потоками (threads). Более подробно этот аспект процессов рассмотрен в лекции 9.

Каждый пользователь в UNIX имеет свою собственную учетную запись пользователя (account). Она содержит имя пользователя, пароль, идентификатор пользователя (UID), идентификатор главной группы пользователя (GID), описание пользователя, его домашний каталог и путь к командному процессору, который следует запустить при интерактивном входе пользователя в систему.

Пользователь, работающий в UNIX, имеет уникальное имя пользователя и уникальный идентификатор. Идентификатор пользователя (User

ID, UID) – это целое число от 0 до 2147483647. Обычные пользователи в Solaris имеют идентификатор в диапазоне от 100 до 60000, а пользователи с идентификаторами со значениями выше 60000 имеют различные ограничения при работе, поэтому реальным пользователям эти идентификаторы, как правило, не назначают.

Пользователю не надо знать свой идентификатор, потому что он используется только системой; для входа в систему пользователь указывает свое имя (username). Имя пользователя может состоять из латинских букв и цифр; длина имени должна быть от 1 до 8 символов. Первый символ должен быть буквой, и по крайней мере один символ должен быть буквой нижнего регистра. Имя пользователя назначает системный администратор при создании новой учетной записи. Имя пользователя в руководствах часто называют «логин» (login) или «username».

Пользователи объединены в группы. Каждая группа имеет свое имя и уникальный идентификатор (Group ID, GID). В группе может быть сколько угодно пользователей, и каждый из них может быть участником любого количества групп. Однако у каждого пользователя есть главная группа – она указывается в свойствах любого файла, который создает пользователь. Идентификатор группы имеет значение от 100 до 60000, если только это не специальная группа. Для специальных (предопределенных) групп зарезервирован диапазон от 0 до 99.

Пользователей объединяют в группы для того, чтобы было удобнее администрировать систему. Например, если право редактировать содержимое web-сайта надо предоставить нескольким ответственным, то достаточно создать специальную группу, например `webedit`, и включить их всех в эту группу.

Предположим, что эти ответственные – пользователи `ivan`, `lena` и `kira`. Тогда в файл `/etc/group` следует добавить новую группу:

```
webedit::101:ivan, lena, kira
```

Концепция прав доступа в UNIX требует объединять пользователей в группы всегда, когда нужно предоставить одинаковые права доступа к файлам или каталогам группе людей. Разумеется, в Solaris 10 есть и графический интерфейс для создания и модификации состава групп – редактировать `/etc/group` в текстовом редакторе не обязательно.

Домашний каталог и путь к командному процессору играют роль только при интерактивном входе пользователя в систему. В UNIX каждый пользователь может работать с системой как непосредственно (набирая команды ОС на клавиатуре), так и обращаясь через сеть к тем или иным службам, запущенным на компьютере под UNIX.

При входе в систему пользователь набирает имя в ответ на приглашение `login:`. Затем, в ответ на `password` следует набрать пароль. При вводе пароль никак не отображается, в том числе и звездочками.

Пароль может назначаться системным администратором при добавлении пользователя в систему. Подробнее о параметрах пароля (частота изменения, ограничение длины и т. п.) в Solaris рассказано в лекции 5.

Пользователь или системный администратор имеют возможность изменить пароль с помощью команды `passwd имя_пользователя` (или с помощью графического интерфейса). Системный администратор может изменить пароль любого пользователя, все остальные — только свой собственный. Команда `passwd` без параметров меняет пароль того пользователя, который ее запустил. При изменении своего пароля потребуется набрать старый пароль (чтобы подтвердить, что вы — действительно тот, за кого себя выдаете) и затем — новый пароль. Новый пароль потребуется набрать дважды, чтобы система была уверена в вашем выборе.

Пароль может содержать любые символы. Допустимая длина пароля в Solaris составляет от 6 до 8 символов. Регистр в паролях, как и повсюду в UNIX, имеет значение: символы разных регистров считаются различными.

Пароль не должен легко отгадываться. Идеальный пароль в системах UNIX состоит из восьми символов, среди которых есть латинские буквы разных регистров, цифры и знаки препинания. Не следует набирать пароль латинскими буквами, соответствующими русскому слову на клавиатуре — эти комбинации давно попали в словари взломщиков, так же как комбинации последовательных клавиш типа `qazwsx`.

Как на военном заводе нельзя одолжить чужой пропуск на денек, пользователям нельзя делиться с коллегами своим паролем. Не следует также писать пароль на бумажке и приклеивать ее к столу, на стену или монитор.

Если под одним именем и паролем работает несколько человек, нельзя выяснить, кто конкретно выполнил в системе действие от этого имени.

Список пользователей системы хранится в файлах `/etc/passwd` (в нем нет зашифрованных паролей, и его можно читать всем) и `/etc/shadow` (здесь есть зашифрованные пароли и дополнительная информация о параметрах пароля, читать можно только пользователю `root`). Список групп хранится в `/etc/group`. Структура файлов `/etc/passwd`, `/etc/shadow` и `/etc/group`, а также иные возможные источники информации о пользователях и группах в системе рассмотрены в лекции 5.

Режимы работы систем UNIX

Любая система UNIX, и Solaris в том числе, может работать в нескольких режимах. Это как минимум однопользовательский и многопользовательский режимы. В однопользовательском режиме систему загружают только в экстренном случае – для какой-то серьезной модификации или ремонта. В этом режиме загружается только ядро и командный процессор – сетевые службы не задействуются. Многопользовательский режим – это обычный режим работы системы, который включается после загрузки по умолчанию. В ряде систем (например, в Linux) существует большее количество режимов работы (часто – семь, с номерами от 0 до 6, в Solaris к ним еще добавляется режим s). Среди этих режимов выделяют режим выключения системы, однопользовательский режим и несколько многопользовательских, которые отличаются друг от друга тем, какие именно сетевые службы запускаются при старте системы.

Для обычной работы система загружается в многопользовательском режиме. В нем пользователи могут одновременно входить в систему локально или через сеть, посылать программам, работающим в системе, запросы различного характера по сети. В этом режиме множество пользователей одновременно могут работать в системе. Их количество ограничивается размером таблиц ядра. Ограничение числа одновременно работающих пользователей связано не с числом одновременно запущенных командных процессоров или сеансов связи, а с количеством запущенных процессов и потребляемых ими ресурсов. Поэтому в каждый момент времени максимальное число пользователей, имеющих возможность работать с системой, индивидуально.

Для загрузки системы в однопользовательском режиме следует при загрузке дать команду

```
boot -s
```

или

```
b -s
```

программе-загрузчику.

Для перехода из многопользовательского режима в однопользовательский дайте команду

```
init s
```

или

```
init S
```

Тот же эффект даст выполнение `shutdown` без параметров.

Для того чтобы перевести систему из однопользовательского режима в многопользовательский, нужно перезагрузить компьютер или выйти из командного процессора однопользовательского режима (`exit` или `Ctrl+D`, работает в большинстве случаев). Также можно запустить программу `init` с параметром — названием режима работы, например

```
init 3
```

Существует несколько режимов работы (`runlevels`) операционной системы. Режим с номером 1 соответствует однопользовательскому режиму, а 3 — многопользовательскому. Режим номер 0 — это остановка (на этот уровень систему переводит команда `shutdown`).

О режимах работы системы можно узнать больше из руководства по системе:

```
man init
```

В Solaris 10 введено новое понятие «этап» (`milestone`), которое чем-то похоже на «режим работы». Этап — это определенное состояние системы, в котором запущены характерные для него службы. Этап, по сути, — это определенный набор служб. Например, для того чтобы оставить работать только те службы, которые характерны для однопользовательского режима работы, следует дать команду

```
svcadm milestone -d milestone/single-user:default
```

Этап, указываемый команде `svcadm milestone`, может быть ключевым словом «`single-user`», «`multi-user`» или «`multi-user-server`». Кроме этого, есть специальные этапы «`all`» (запуск всех служб, которые должны быть запущены при старте системы) and «`none`» (временный запрет всех служб, кроме `svc:/system/svc/restarter:default`). При выдаче этой команды в указанной форме этап становится этапом по умолчанию и сохраняется таковым при перезагрузке системы.

Для перевода системы в другой режим работы следует использовать вызов `init` напрямую, — переход к другому этапу не вызывает перехода к новому режиму работы. Из соображений совместимости в Solaris 10 режимы работы контролируются процессом `init`, как и в других системах семейства System V.

Дополнительную информацию об режимах работы, этапах и связанном с этапами механизме SMF можно получить из лекции 11.

Быстрый взгляд на экран в UNIX

Теперь мы уже знаем достаточно, чтобы не удивляться сообщениям системы при работе. Кстати, а какие у нас есть средства общения с системой? Как мы будем разрабатывать в ней программы, как выходить в Интернет, читать почту, набирать тексты, записывать диски, смотреть фильмы, слушать музыку, наконец, настраивать саму систему?

В любой системе UNIX для всех этих задач предлагается воспользоваться либо графическим интерфейсом (его еще называют оконным — каждая програма общается с человеком в одном или нескольких окнах), либо интерфейсом командной строки (это когда для выполнения действия надо написать системе команду — например, напечатать команду `ls` и нажать клавишу `Enter`, чтобы посмотреть список файлов в каталоге).

Графический интерфейс

Для каждодневной работы в системе вы, скорее всего, предпочитаете пользоваться графическим интерфейсом. Solaris по умолчанию (если не настроен иначе) после загрузки предлагает ввести имя и пароль, причем делает это уже в графическом режиме. Пароль традиционно не отображается при вводе ни звездочками, ни как-либо иначе. Это делается для того, чтобы злоумышленник или случайная видеочка камера наблюдения не подглядели из-за плеча, сколько символов в вашем пароле.

После ввода корректных имени и пароля вы окажетесь в графической среде GNOME, которая является средой по умолчанию в Solaris 10 и Solaris Express. Вообще говоря, при вводе имени и пароля можно выбрать традиционную для Solaris и HP-UX среду CDE (Common Desktop Environment), воспользовавшись кнопкой Options в окне ввода. Если же вы последовательный сторонник KDE, то знайте, что она не поставляется в составе дистрибутива Solaris, но ее можно установить самостоятельно, скачав готовый пакет KDE для Solaris с сайта blastwave.org. О пакетах и их установке рассказывается в лекции 14.

Менеджер окон

В любой графической среде существует программа, которая называется менеджером окон. Она используется для управления окнами других программ. Именно она позволяет свертывать окна, изменять их размеры, перемещать по экрану, переключаться между ними, развертывать их на весь

экран. Она же отображает окна в привычном для нас виде – с заголовком окна и элементами управления для свертывания и перемещения окон.

Экран в каждый момент времени показывает содержимое рабочего стола, который в Solaris носит название «рабочего пространства» (workspace). Рабочих пространств может быть несколько. С точки зрения менеджера окон, рабочее пространство – это множество окон. Менеджер окон позволяет переключаться между разными рабочими пространствами. По умолчанию их четыре, обычно используют от четырех до шести рабочих пространств. С помощью настроек менеджера окон число рабочих пространств можно изменить.

Графическая среда GNOME

Работу графической среды обеспечивает несколько компонент, о которых рассказано подробнее в лекции 24, а пока достаточно сказать, что при использовании GNOME на экране вы увидите рабочий стол (Desktop), внизу экрана расположена панель задач, в правой части которой может отображаться служебная информация вспомогательных программ (так называемых *апплетов* панели задач). Например, там может быть указан текущий язык ввода, там же располагается кнопка регулятора громкости звука и т. п. В левой части панели отображается кнопка вызова календаря, на которую выведены системные дата и время (рис. 1.1).



Рис. 1.1. Вид рабочего стола в Solaris

Для доступа к меню вызова различных приложений надо нажать кнопку с логотипом Java (чашка ароматного кофе) в левом нижнем углу* экрана. Если вы еще не изменяли настройки вашей графической среды после установки системы, то, нажав эту кнопку, вы сможете вызвать ряд приложений, включая офисные приложения, графический редактор, игры, справку по графической среде и т. п. Приложения для настройки системы тоже будут доступны, но при их запуске потребуются вводиться пароль администратора (пользователя root), если вы вошли в систему не от имени root.

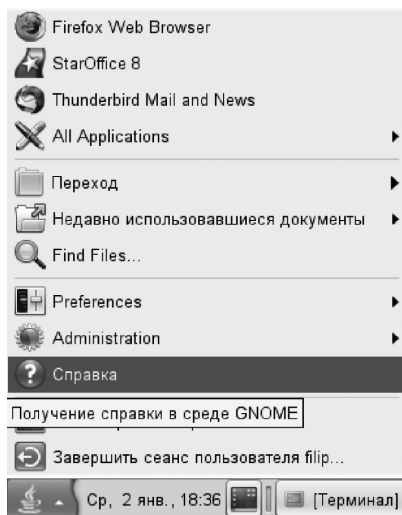


Рис. 1.2. Меню запуска приложений в GNOME

Работа в графической среде в Solaris не имеет заметных отличий от работы в аналогичной среде в Linux, FreeBSD или других системах UNIX, и весьма похожа на работу в графической среде Windows. Таким образом, так как у вас есть опыт работы в какой-нибудь из этих систем, с работой в Solaris вы легко освоитесь (или уже сделали это; вероятно, у вас это заняло времени не больше, чем чтение этого абзаца).

Графическая среда CDE

Управляющая панель (front panel) менеджера окон доступна в любом рабочем пространстве внизу экрана. Кроме переключения между пространствами управляющая панель менеджера окон (рис. 1.3) позволяет

* В OpenSolaris 2008.05 меню вызова приложений расположено вверху экрана — так же как в Ubuntu Linux.

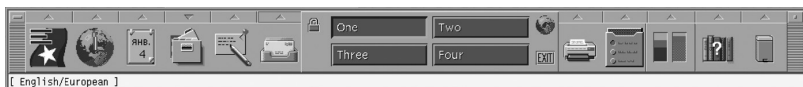


Рис. 1.3. Управляющая панель менеджера окон в CDE

вызывать программы, кликая мышью по соответствующим иконкам. С подобной системой знакомы те из вас, кто пользовался ярлыками в панели задач Windows. За каждой иконкой, кроме конкретной программы, скрывается целый набор ярлыков, служащих для вызова других, тематически связанных с ней программ (рис. 1.4). Для доступа к ним следует нажать стрелку над иконкой (на рис. 1.3 нажата стрелка над иконкой «приложение», четвертой слева).

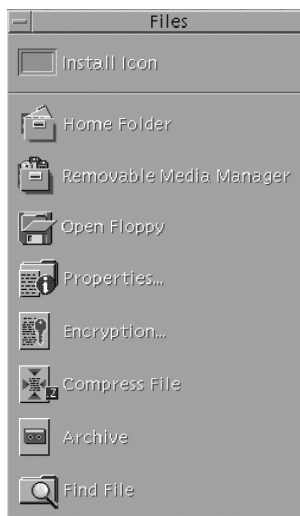




Рис. 1.4. Связанные с иконкой «приложения» ярлыки

Более подробно о настройке среды CDE рассказано в лекции 24.

Экран – рабочее пространство пользователя

На рабочем пространстве располагаются окна запущенных в данный момент программ. Окно может быть свернутым (тогда оно изображается иконкой в рабочем пространстве); по умолчанию размещение иконок свернутых окон начинается в левом верхнем углу и ведется сверху вниз слева направо. Также окно может занимать весь экран или любую его часть.

Что можно сделать с окном

Верхняя строка окна (рис. 1.5) содержит три элемента управления: черточку в левом верхнем углу окна (первый элемент), точку  (второй) и выпуклый квадрат  (третий элемент) в правом верхнем углу окна.

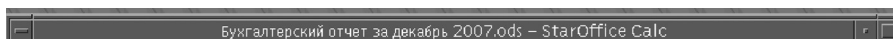



Рис. 1.5. Верхняя строка окна в CDE

При нажатии на первый открывается меню, которое позволяет восстановить первоначальный размер окна — *restore* (если окно развертывалось на полный экран до того), переместить окно — *move*, изменить его размер — *resize*, свернуть — *minimize*, развернуть на полный экран — *maximize*, поместить окно на задний план — *lower* (за окна других программ) или закрыть — *close*. С помощью этого же меню можно указать, что данное окно должно присутствовать во всех (*occur all workspaces*) или нескольких (*occur workspace*) рабочих пространствах. При переключении из одного пространства в другое окно будет «переходить» вслед за вниманием пользователя из одного пространства в другое.

Второй элемент управления служит для свертывания окна (эквивалентно выбору *minimize* в меню первого элемента управления, а третий —  для развертывания во весь экран (эквивалентно *maximize*). У развернутого окна третий элемент управления выглядит иначе (впалый квадрат) и служит для восстановления первоначального размера окна (*restore*).

Интерфейс командной строки

Понятие терминала

Терминал — это экран и клавиатура, с помощью которых осуществляется связь с компьютером.

Терминалы бывают графические и текстовые. Графические терминалы могут работать в графическом режиме.

Терминал предназначен исключительно для ввода информации и ее отображения на экране. Терминалы бывают физическими (еще их называют реальными), виртуальными и псевдо-терминалами (т. е. программами, которые «притворяются» терминалами).

Физический терминал — это устройство, состоящее из экрана и клавиатуры, обычно подключенное к компьютеру через последовательный интерфейс, например с помощью кабеля или модема. Если вы соединитесь с UNIX-машиной с помощью эмулятора терминала через модем

своего компьютера, позвонив на другой модем, подключенный непосредственно к UNIX-машине, то с точки зрения UNIX вы работаете на физическом терминале. В этом случае связка «модем-телефонная сеть-модем» рассматривается как единый кабель.

В старых версиях Solaris неофициально поддерживались виртуальные терминалы. Начиная с версии 9 эта поддержка отсутствует. Основа виртуального терминала — это монитор и клавиатура, непосредственно подключенные к компьютеру. Однако, хотя монитор всего один, виртуальных терминалов может быть несколько. Вы можете по очереди работать с разными виртуальными терминалами, нажимая комбинацию клавиш Alt+Fx, где x — номер терминала. С монитором и клавиатурой связаны несколько независимых виртуальных терминалов. При работе в графическом режиме надо использовать комбинацию клавиш Ctrl+Alt+Fn для переключения между виртуальными терминалами.

Виртуальные терминалы, как правило, текстовые, хотя один из них может быть графическим.

Виртуальные терминалы используются в компьютерах на платформе i386, потому что терминалы, подключаемые к компьютерам на других платформах, обычно постоянно работают в графическом режиме, а в нем вместо нескольких виртуальных терминалов проще открыть несколько окон эмуляторов терминала. Виртуальные терминалы широко применяются в Linux и FreeBSD.

При работе с графическим терминалом для получения нескольких терминалов достаточно просто открыть несколько окон и в каждом из них запустить терминальную программу. Так, в Solaris можно запустить несколько приложений, обозначенных в меню CDE как «console». Программа, которая запускается при выборе приложения «console», называется dtconsole. Выбрав в меню «This host», вы запустите терминальную программу dtterm.

Консоль — это терминал, который применяется для загрузки операционной системы.

Проще говоря, если системный администратор при загрузке использует (или имеет возможность это делать) некий терминал для наблюдения за процессом загрузки и (возможно) вмешательства в него, то этот терминал называется консолью. Для UNIX-систем консоль — это либо непосредственно подключенный к компьютеру монитор и клавиатура, либо подключенный к первому последовательному порту терминал. В приложении к виртуальным терминалам консолью является первый виртуальный терминал (Alt+F1).

Псевдотерминал — это программа-эмулятор терминала, которая соединяется с UNIX-компьютером через сеть или запускается в графическом режиме (например, dtterm в CDE и gnome-terminal в GNOME).

Любой терминал (физический, виртуальный или псевдотерминал) может быть графическим. Эмулятор графического терминала базируется на программе X-Server'a. Подробнее об X-server'ах и системе X-Window — в лекции 24.

Текстовые терминалы могут отображать только текст и псевдографику. Мы будем рассматривать многие команды Solaris в предположении, что мы работаем с текстовым терминалом. В то же время, когда в Solaris доступна и полезна программа с графическим интерфейсом, которая часто нужна системному администратору в работе, мы будем рассматривать и такие случаи.

Существует большое количество разных текстовых терминалов. Наиболее известны терминалы ANSI и VT-100. Разные терминалы отличаются друг от друга управляющими последовательностями. Например, чтобы передать терминалу команду «поднять курсор на одну строку вверх» терминалу VT-100 нужно передать последовательность <Esc>[A. Описания терминалов (и соответствующих управляющих последовательностей) содержатся в файле /etc/termcap и базе данных в подкаталогах /usr/share/lib/terminfo.

Переменная среды окружения TERM должна иметь значение типа терминала, на котором вы работаете. Возможно, ее придется переопределить, если в начале вашей работы система неверно определила тип вашего терминала. Фактический тип вашего терминала может отличаться от того, что принят в системе по умолчанию. Переопределение выполняется путем присваивания переменной нужного значения:

```
TERM='VT100'; export TERM
```

Вывести на экран значение переменной TERM можно командой

```
echo $TERM
```

Если переменная TERM определена неверно, терминал будет вести себя непредсказуемо. Например, при нажатии Enter не произойдет перевода строки или при нажатии стрелки вверх на экране появится [[A, а курсор вверх не сдвинется. Некоторые терминалы практически совместимы между собой, например ANSI и VT-100. Однако некоторые управляющие последовательности могут не совпадать. Поэтому лучше, чтобы в TERM был записан тип именно того терминала, на котором вы работаете.

Управляющие комбинации клавиш

Текстовый терминал воспринимает и передает активной программе коды следующих управляющих комбинаций клавиш, которые любая программа должна интерпретировать стандартным образом (см. таблицу 1.1).

Таблица 1.1. Управляющие комбинации клавиш

Ctrl+C	Прерывание выполнения программы (termination)
Ctrl+D	Конец ввода текста
Ctrl+S	Остановка вывода на экран
Ctrl+Q	Продолжение вывода на экран
Ctrl+U	Стирание введенной строки
Ctrl+Z	Остановка выполнения программы (suspend)
Backspace	Удаление последнего введенного символа

Некоторые терминалы не воспринимают клавишу Backspace стандартным образом, вместо нее на таких терминалах можно использовать клавишу Delete.

В старых системах клавиша Delete могла применяться вместо Ctrl+C для прерывания работы программы.

Команда Ctrl+D (конец ввода текста) может использоваться в командном интерпретаторе для выхода из него. При работе в текстовом режиме выход из командного интерпретатора, который был запущен для пользователя при входе в систему (такой экземпляр интерпретатора называется login shell), означает автоматическое завершение сеанса работы с системой (logout).

В некоторых системах командный интерпретатор по умолчанию настраивается так, чтобы такого завершения работы не происходило. В этом случае выйти из системы пользователь может, только дав команду logout.

Настройка терминала: команда stty

Проверить, как настроен терминал, можно с помощью программы stty:

```
stty -a
```

Эта команда позволяет узнать все настройки терминала, в том числе скорость, другие физические параметры передачи символов, а также управляющие комбинации:

```
stty -a
speed 9600 baud;
rows = 24; columns = 80; ypixels = 490; xpixels = 570;
csdata ?
eucw 1:0:0:0, scrw 1:0:0:0
intr = ^c; quit = ^\; erase = ^h; kill = ^u;
eof = ^d; eol = <undef>; eol2 = <undef>; swtch = <undef>;
start = ^q; stop = ^s; susp = ^z; dsusp = ^y;
rprnt = ^r; flush = ^o; werase = ^w; lnext = ^v;
-parenb -parodd cs8 -cstopb hupcl cread -clocal -loblk -crtscs -crtsex-
off -parext
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl
-iuclc
ixon -ixany ixoff -imaxbel
isig icanon -xcase echo echoe echok -echonl -noflsh
-tostop echoctl -echoprt echoke -defecho -flusho -pendin iexten
opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel
```

Для настройки правильной реакции на стандартные комбинации (см. таблицу 1.1) клавиш нужно выполнить команды

```
stty susp '^z'
stty intr '^c'
stty erase '^h'
```

и т. д. Список параметров установки (susp, intr, erase и другие) можно увидеть по команде `stty -a` или в руководстве (`man stty`).

Лекция 2. Изменение настроек системы для начинающих

В этой лекции будут перечислены только основные практические способы изменения настроек системы, которые могут понадобиться человеку, начинающему работать с Solaris. Детали управления системой рассматриваются в других лекциях.

Изучение руководства по системе

Невозможно ни изучать, ни управлять UNIX без руководства по системе. Печатное руководство поставляется только с коммерческими системами, да и то не всегда. Кроме того, печатные руководства имеют скверную привычку теряться. Потом они отыскиваются, но только не в тот момент, когда позарез нужны вам.

К счастью, любой диалект UNIX устанавливается вместе с электронной копией документации. Эта документация состоит из нескольких разделов (обычно их семь или восемь, иногда больше). Каждый раздел содержит статьи. Статья посвящена одной команде, файлу, понятию или системному вызову UNIX. Любая статья имеет свой заголовок, в котором есть имя команды (файла, вызова и т. п.) и ее очень короткое описание, например:

```
cp - copy files and directories
```

Руководство вызывается командой `man`. Программа `man` форматирует страницу руководства и показывает ее на экране с помощью программы форматирования `nroff` и программы поэкранной выдачи текста `more` (или ее аналога — `page`, `pg`, `less`). На диске страницы руководства хранятся в неформатированном виде, так как это экономит место на диске. Страницы руководства хранятся в каталогах `/usr/share/man/manN` (`N` — номер раздела), или `/usr/man/manN`, или в каком-нибудь похожем подкаталоге `/usr`. Отформатированные страницы `man` складывает в подкаталоги `catN`. Иногда они расположены там же, где и `manN`, иногда — нет. Программу `man` можно запустить с ключом `-M`, чтобы указать, в каком каталоге искать страницы руководства. Иногда при установке некоторых программ руководства по ним размещаются вне стандартного дерева каталогов `man`, тогда этот ключ может пригодиться.

Ссылки на `man`-страницы в литературе обычно включают номер раздела, в котором расположена статья, например `inetd.conf(5)` означает статью о конфигурационном файле `inetd.conf` демона `inetd` в пятом

разделе руководства. В этой книге мы тоже будем так ссылаться на статьи руководства по системе.

В таблице 2.1 перечислены разделы руководства, к которым обращается `man`. В BSD и System V руководства организованы немного по-разному.

Таблица 2.1.

Раздел	System V	BSD
1	Команды	
2	Системные вызовы	
3	Библиотечные несистемные вызовы (например, из библиотеки Perl)	
4	Драйверы устройств, файлы и протоколы	
5	Разные документы	Файлы конфигурации
6	Игры и демонстрационные программы	Игры и демонстрационные программы
7	Драйверы устройств и сетевые протоколы (другие System V системы)	Тексты, кодировки, выражения
8		Утилиты системного администрирования

Некоторые коммерческие системы UNIX имеют дополнительные страницы руководства, описывающие специфические элементы данной системы. Они обозначаются, как правило, номером раздела с какой-нибудь буквой, например, 1b.

Чтобы получить справку по любой команде UNIX или даже понятию, наберите

```
man то_что_вас_интересует
```

Набирайте `man -k тема` для поиска темы в заголовках статей руководства.

Команда `man -k` предполагает, что существует индексированный список заголовков статей руководства. Этот список создается командой `catman` или каким-нибудь ее аналогом. Синоним `man -k` — это команда `argoros`. Для поиска информации о программах можно использовать еще и команду `whatis`.

Некоторые команды в UNIX – это встроенные команды командного процессора, а не отдельные программы, например `alias` и `jobs`. Поэтому, если вы не можете найти справку по таким командам, попробуйте прочитать `man` по командному процессору, которым вы пользуетесь (скорее всего, `sh` или `bash`).

Более подробно об организации страниц руководства, о поиске информации о системе и ее инструментах рассказано в лекции 10.

Кроме `man`, документация содержится в гипертекстовых файлах формата `info` и файлах `HOWTO`. Файлы `info` нужно читать программой `info`. Можно ее запустить без параметров и затем дать команду `?` для знакомства с системой встроенных команд `info`. Она отличается от `man` своей «гипертекстовостью»: можно переходить по ссылкам как внутри документа, так и между документами.

`HOWTO` – это набор документации, распространенный в Linux. В документах `HOWTO` разъясняются многие действия, которые могут понадобиться при администрировании. `HOWTO` распространяется в формате HTML или в виде текстовых файлов. В отличие от `man`, не существует жесткой структуры `HOWTO`. Это просто набор документов, и на одном сайте он может оказаться более полным, чем на другом.

В Интернете есть несколько сайтов, на которых размещен перевод на русский язык `man` и `HOWTO` в формате HTML. Перевод иногда хорош, иногда дурен. Если вы не в состоянии понять, что написано в переводе документации, остается в сердцах ругать переводчика и обращаться к оригиналу.

Настройки графической среды

Настройка графической среды в GNOME сводится к настройкам параметров экрана и настройкам вида рабочего стола. Настройка рабочего стола тривиальна; единственный тонкий момент – это модификация панели задач внизу экрана. Для того чтобы добавить на нее элемент, надо щелкнуть по свободной части панели правой клавишей мыши и действовать, выбирая пункты контекстного меню (рис. 2.1). Для изменения раз-

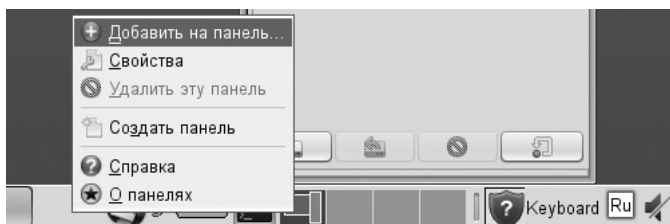


Рис. 2.1. Добавление апплета на панель задач в GNOME

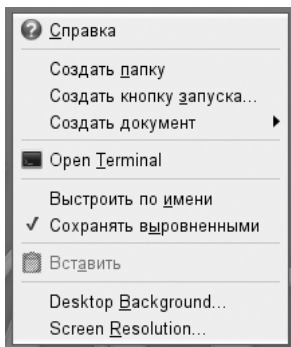


Рис. 2.2. Контекстное меню свойств рабочего стола в GNOME

решения экрана следует щелкнуть правой клавишей мыши по рабочему столу и далее выбрать в контекстном меню «Screen Resolution» (рис. 2.2). Как видно на рисунке, из этого же контекстного меню можно установить фон рабочего стола и запустить эмулятор терминала (`gnome-terminal`).

Настройки сети

Настройки сети в Solaris лучше всего оставить программе — установщику системы. Однако при надобности их модифицировать можно воспользоваться различными способами. Во-первых, можно из командной строки отредактировать файлы `/etc/hosts`, `/etc/resolv.conf` и выполнить настройку интерфейсов командой `ifconfig`. Если вы — обычный пользователь, не имеющий опыта администрирования UNIX, этот путь потребует от вас приобретения новых знаний — о сетевых интерфейсах, адресах, редактировании файлов настроек и т. п. Поэтому, чтобы автоматизировать процесс настройки, можно воспользоваться программой `inetmenu`. В новых версиях Solaris 10 (начиная с Solaris 08/07 или, что то же самое, Solaris 10 update 4) для этого следует использовать службу NWAM. Подробнее о настройках сети рассказано в лекции 15.

Вполне возможно, что ваш компьютер уже настроен (вручную или автоматически) для работы с сетью. Поэтому перед тем как применять знания, полученные из чтения этого раздела, попробуйте запустить программу просмотра Сети (ее еще называют неблагозвучным термином «браузер», а правильное — «web-обозреватель») и посмотреть какой-нибудь веб-сайт, например `yandex.ru`. Если вы увидите осмысленный текст, а не сообщение об ошибке подключения, стало быть, сеть уже настроена.

Помните, что некоторые программы (например Mozilla Firefox и Mozilla Thunderbird) могут кэшировать настройки сети при запуске в своем локальном кэше и вследствие этого не будут корректно работать,

если во время их работы изменить настройки сети. Если вы пользуетесь этими программами, их, возможно, придется закрыть и запустить заново после настройки сети.

Полный перечень сетевых интерфейсов, распознанных системой, и их настроек можно узнать по команде

```
ifconfig -a
```

Среди интерфейсов, которые показывает эта команда, обязательно будет интерфейс `lo0`, который является «внутренним», «локальным» сетевым интерфейсом — через него сетевое приложение, работающее на вашем компьютере, может обратиться к другому сетевому приложению, работающему на том же самом компьютере. Этот локальный интерфейс не требует настройки.

Здесь мы приведем описание настройки сети только с помощью `inetmenu`, как самой простой.

Прежде всего, запустите `inetmenu`. После запуска вы увидите список сетевых интерфейсов и выберите один из них. Локальный интерфейс `lo0` программа `inetmenu` не показывает. Возможно, список будет состоять из одного элемента — если ваш компьютер имеет один физический сетевой интерфейс. Имя интерфейса может быть разным и зависит от типа сетевого адаптера, который у вас установлен. Если интерфейсов несколько (скажем, есть проводной и беспроводной интерфейсы), то в списке будут содержаться и те и другие.

Проводное соединение (Ethernet)

Интерфейс, представляющий собой сетевой адаптер, может называться `qme0`, `bge0` или иначе — в зависимости от модели адаптера. После выбора интерфейса в программе `inetmenu` потребуется выбрать тип настройки: `deconfigure` (следует выбирать, когда требуется отменить предыдущую настройку), `DHCP` (автоматическая настройка), `Static address` (статический, т. е. постоянный адрес) и, возможно, иные варианты, если они предусмотрены файлом конфигурации `inetmenu`. Подробнее о конфигурации `inetmenu` рассказано в лекции 15. Если у вас есть параметры настройки статического адреса, воспользуйтесь ими; если в вашей сети принята динамическая настройка, выберите `DHCP`.

После того как вы сделаете выбор, программа `inetmenu` сама вызовет `ifconfig` с необходимыми параметрами (вы можете потом посмотреть, как именно настроен интерфейс с помощью уже знакомой вам команды `ifconfig -a`), модифицирует файлы `/etc/hosts` и `/etc/resolv.conf`.

Беспроводное соединение (Wi-Fi)

Ваш компьютер может быть оборудован адаптером беспроводной сети стандарта 802.11 (часто сеть этого типа называют Wi-Fi, реже – словом wireless). В этом случае кроме настройки сетевых интерфейсов может потребоваться подключение к одной из присутствующих в эфире беспроводных сетей. При подключении к беспроводной сети следует выбрать название точки доступа, к которой вы собираетесь подключиться. Для обзора всех имеющихся беспроводных сетей и их важных параметров (наличие шифрации и уровень сигнала) надо использовать команду

```
wificonfig scan
```

Может оказаться, что в наличии есть несколько точек доступа с разными именами сетей. Какую выбрать? Если вы точно знаете имя и (необязательно) пароль доступа к одной из сетей, выбирайте именно ее. Если не знаете – пробуйте все имеющиеся беспарольные. Иногда сеть, предоставляющая доступ без пароля, оказывается платной. В этом случае о порядке оплаты следует спросить знающих людей вокруг или персонал того места, где вы находитесь. В цивилизованных странах корректно настроенная платная сеть обычно сообщает вам об условиях подключения, как только вы подключитесь к ней и попытаетесь просмотреть любую страницу в Интернете.

При запуске `inetmenu` для настройки беспроводного доступа следует выбрать адаптер беспроводного доступа из списка сетевых адаптеров (вероятно, он будет называться `ath0`). После этого `inetmenu` покажет список доступных сетей, вы выберете одну из них, и после этого, возможно, придется ввести пароль на доступ к ней. Затем появится такое же меню выбора конфигурации, что и для проводной сети, с возможностью выбрать DHCP и другие варианты настройки. Беспроводные адаптеры в подавляющем большинстве случаев настраиваются с помощью DHCP.

После выбора варианта настройки `inetmenu` подключит вас к выбранной сети с помощью вызова программы `wificonfig` и внесет изменения в файлы `/etc/hosts` и `/etc/resolv.conf`, а также вызовет `ifconfig` для настройки беспроводного интерфейса. Вы можете посмотреть результаты настройки командой `ifconfig -a`, а также посмотреть параметры подключения к беспроводной сети с помощью команды `wificonfig status`.

Если в вашей системе разрешена работа службы NWAM, то настройка беспроводной сети будет выглядеть похоже на настройку с помощью `inetmenu`, но все произойдет автоматически – запускать какие-либо программы не придется. О настройке с помощью NWAM подробнее рассказано в лекции 15.

Администратор системы (привилегированный пользователь)

Администратором, т. е. пользователем, наделенным всеми возможными правами в системе, во всех диалектах UNIX, включая Solaris, является пользователь с именем `root` и идентификатором (UID), равным нулю. Этого пользователя также иногда называют суперпользователем (`superuser`). Он имеет неограниченные права на доступ к любому объекту в системе.

Тот, кому доверен пароль суперпользователя, должен хорошо знать основные процедуры администрирования UNIX и работать в системе так, чтобы не навредить ей. Системный администратор отвечает за безопасность системы, ее стабильную работу, добавление и удаление пользователей, регулярное резервное копирование и т. д. Он должен хранить пароль суперпользователя как зеницу ока. Доверять пароль суперпользователя многим людям не следует: системный администратор всегда должен точно знать, что действие от имени `root` сделал проверенный и легко идентифицируемый человек.

Вход в систему под именем `root` разрешен только с терминалов, непосредственно присоединенных к UNIX-машине. Подключение через сеть от имени `root` запрещено. При надобности выполнить команду от имени `root` через сеть следует подключиться от имени обычного пользователя, а затем выполнить команду «превращения» в привилегированного пользователя (`su`). Пользователь `root` никогда не должен иметь пустой пароль.

Как правильно быть администратором системы

Администратор системы – это, по сути, роль, играть которую может быть поручено одному или нескольким пользователям системы. Принято, чтобы каждый пользователь работал в системе под своим собственным именем, а при необходимости выполнить действие от имени администратора «переходил к исполнению» его роли.

В большинстве систем UNIX для этого использовалась одна из двух схем: передача пароля пользователя `root` строго ограниченному кругу доверенных лиц администратора или применение команды `sudo`, которая позволяла обычному пользователю выполнить определенный набор действий от имени администратора в соответствии с настройками файла `/etc/sudoers`. Суперпользователь в последнем случае должен был заранее сформировать этот файл, указав, кому и что позволено делать.

Оба этих способа не идеальны: первый в силу размазывания ответственности, второй – из-за ограниченности возможностей `sudo`.

В Solaris начиная с версии 8 была введена система управления доступом, основанная на ролях (RBAC – Role-Based Access Control). Она подразумевает создание большего количества ролей, чем всего один суперпользователь. Каждая роль может обладать строго определенным набором привилегий, позволяющих ей выполнить то или иное действие, связанное с безопасностью системы, – скажем, только добавлять пользователей или только подключаться к беспроводной сети. Дарованная роли привилегия может охватывать широкий спектр действий или предполагать разные пути их выполнения. Например, добавление пользователей может происходить в командной строке командой `useradd` или через программу графического интерфейса `smc`.

Более подробно о ролевом управлении доступом рассказано в лекции 21.

Однако для того чтобы быть хорошим администратором системы, недостаточно знать теорию управления доступом, важно еще и соблюдать некоторые правила, которые выработаны поколениями системных администраторов.

Всегда помните о том, что система беззащитна перед человеком, знающим пароль администратора (т. е. пароль пользователя `root`). Поэтому:

1. не работайте в системе от имени `root`, если вам не надо настраивать систему; набор текстов или разработка программ при работе от имени `root` – признак разгильдяйства сисадмина;
2. не работайте от имени `root` будучи в подпитии или с похмелья; это – совершенно серьезно, потому что работая от `root`, легко по неосторожности удалить важные данные или внести необратимые изменения в систему – вы же не повезете родных из гостей домой на своей машине, если уже выпили граммов по триста водочки?
3. всегда делайте резервные копии важной информации;

Лирическое отступление по поводу важной информации:

однажды один из авторов этой книги был уверен, что в его системе все важные данные ежедневно копируются на удаленный сервер, а там усидчивый коллега обеспечивает их ежедневную запись на ленту. В один весьма далекий от прекрасности момент случилось неприятное: по неосторожности было стерто содержимое одного раздела диска. «Что ж, бывает», – подумал автор и занялся восстановлением данных, копируя их с удаленного сервера. Каково же было его удивление, когда часть данных оказалось невозможным восстановить – их не было на сервере. Выяснилось, что один важный каталог не попал в список того, что копируется ежедневно... «Ладно», – подумал автор и обратился к усидчивому администрато-

тору удаленного сервера за лентой, поскольку на нее раз в квартал копировалось вообще все. И тут случилось самое забавное: оказалось, что на ленту уже полгода как ничего не копируется, и об этом усидчивый администратор никому не сказал. А важный каталог был достаточно свежим, чтобы его не оказалось на более старой ленте.

Мораль истории в том, что при организации резервного копирования (1) обязательно убедитесь в том, что вы делаете резервную копию всей важной информации (и каталога /etc, и каталогов настроек пользователей – всяких там .mozilla из домашних каталогов, и вспомогательных скриптов, написанных вами самими и положенных куда-нибудь в /usr/scripts) и (2) надейтесь только на себя, всегда проверяйте, соблюдаются ли установленные в компании правила резервного копирования; следите за тем, что делаете сами и проверяйте то, что делают ваши коллеги и подчиненные – если от них зависит то, за что вы отвечаете перед Богом и людьми.

4. передавайте пароль root только тем коллегам, которым доверяете больше чем себе; и даже в этом случае, как только их доступ к системе с администраторскими полномочиями более не требуется, меняйте пароль root;
5. меняйте пароль root регулярно, делайте его достаточно сложным. Пароль Jrcfyf – плохой, потому что он легко отгадывается подбором по словарю, в котором русские слова написаны нажатием соответствующих клавиш без переключения языка на русский – такие словари созданы уже много лет назад. Пароль должен быть достаточно длинным, но помните о том, что во многих UNIX из соображений совместимости фактически во внимание принимается 8 первых символов пароля. Примером хорошего пароля является K?3085/j – в нем используется случайное сочетание символов латинского алфавита разных регистров, знаков препинания и цифр. Кажущаяся трудность запоминания такого пароля – это иллюзия, на самом деле средний человек способен запомнить любое сочетание восьми символов, просто набрав его несколько раз на клавиатуре;
6. пользуйтесь ролевым управлением доступом в Solaris – это мощное подспорье тем серьезным администраторам, которые понимают вред распространения пароля root среди нескольких человек одновременно;
7. никогда не запускайте скрипты, полученные из Сети, от имени root (так, например, незачем запускать скрипт configure от имени root при установке нового приложения).

Управление пользователями

Управление пользователями сводится к их добавлению, удалению и назначению им определенных прав. В простых случаях все ограничивается только первой задачей. Для добавления пользователя достаточно в командной строке дать команду `useradd имя`, чтобы добавить пользователя с именем «имя». После такого добавления учетная запись пользователя блокируется, потому что пользователю еще не назначен пароль. Пароль можно назначить в командной строке командой `passwd имя`. После этого надо будет дважды набрать устанавливаемый для пользователя пароль, и он будет присвоен пользователю. По факту назначения пароля учетная запись будет разблокирована, и пользователь сможет входить в систему, используя свои имя и пароль.

Для выполнения этих же действий можно использовать программу Solaris Management Console (`smc`), которая имеет оконный интерфейс и вызывается командой `smc &` или `/usr/sadm/bin/smc &` либо запускается посредством двойного щелчка мыши по соответствующей записи в папке `/usr/sadm/bin`. При работе в GNOME можно вызвать `/usr/bin/users-admin` из меню Administration → Users and Groups.

По умолчанию управлять учетными записями пользователей имеет право только администратор (пользователь с именем `root`).

Об управлении пользователями и программе `smc` более подробно рассказано в лекции 5.

Предопределенные пользователи и группы

При установке системы в файлы `/etc/passwd` и `/etc/group` автоматически записываются сведения о предопределенных пользователях и группах. Таким образом гораздо проще управлять правами доступа к системным файлам.

Предопределенные группы и пользователи существуют для того, чтобы от их имени работали системные службы и в то же время доступ к файлам этих служб был ограничен для всех остальных.

В Solaris предопределенными являются пользователи, перечисленные ниже (список корректен для версий Solaris 9 и 10, жирным шрифтом выделены пользователи, которые есть только в Solaris 10):

```
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1:/:
bin:x:2:2:./usr/bin:
sys:x:3:3:/:
adm:x:4:4:Admin:/var/adm:
```



```
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/:
webservd:x:80:80:WebServer Reserved UID:/:
postgres:x:90:90:PostgreSQL Reserved UID:/:/usr/bin/pfksk
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
```

К предопределенным группам относятся следующие группы (список корректен для версий Solaris 9 и 10, жирным шрифтом выделены группы, которые есть только в Solaris 10):

```
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
sysadmin::14:
smmsp::25:smmsp
gdm::50:
webservd::80:
postgres::90:
nobody::60001:
noaccess::60002:
nogroup::65534:
```

Другие настройки

В Solaris существует довольно много файлов настроек, но все они делятся на два типа — *общесистемные* и *пользовательские*. Первые относятся либо к самой операционной системе, либо к установленным в

ней приложениям. Если приложение имеет какие-то общесистемные настройки, то они будут одинаковыми для всех пользователей системы. Пользовательские настройки приложения имеют влияние только на поведение этого приложения при его запуске конкретным пользователем и на других пользователей не распространяются. Общесистемные настройки сосредоточены в каталоге `/etc`, а пользовательские располагаются в домашнем каталоге пользователя в «скрытых» подкаталогах, имена которых начинаются символом «.» (точка).

Где хранятся настройки Solaris

Большинство общесистемных файлов настроек в любой системе UNIX расположены в каталоге `/etc` либо в его подкаталогах. Практически все файлы настроек представляют собой обычные текстовые файлы. Довольно часто их содержание вполне сносно откомментировано. Все строки, которые в файле настроек начинаются со знака `#` (решетка), являются комментариями. Исключение составляют файлы настроек name-сервера (DNS) – там признаком комментария служит символ `;` (точка с запятой). Формат файлов настроек бывает разным и зависит от того, к какой службе или приложению этот файл относится.

Файлы настроек пользовательских приложений, особенно тех, что установлены после установки системы без компиляции из исходных текстов, могут находиться не в `/etc`, а в каталогах типа `/usr/local/etc` или внутри дерева каталогов, куда установлен пакет программ (например, `/usr/local/apache/conf` для web-сервера `apache`).

Файлы настроек можно редактировать в обычном текстовом редакторе. Некоторые файлы сгенерированы системой или какой-нибудь утилитой автоматически на основании других; в начале таких файлов обычно содержится предупреждение, что для реального изменения настроек нужно редактировать не этот файл, а источник, на основании которого создан данный файл.

Что делать, если забыт пароль администратора

Если пользователь забыл свой пароль, то сообщить ему, каким он был, невозможно. Надо просто установить ему новый. Это можно сделать командой

```
passwd user
```

или с помощью программы `smc` (в графическом режиме). Изменить пароль любого пользователя может только `root`. Если же забыт пароль

root, то следует загрузить Solaris с другого носителя, смонтировать раздел, на котором находится файл `/etc/shadow` злополучного экземпляра системы, и удалить поле пароля (между первым и вторым двоеточиями) в строке, соответствующей пользователю `root`. Затем следует перезагрузить систему, войти под именем `root` и установить новый пароль администратора командой `passwd root`.

Смонтировать раздел вы сможете, загрузившись с дистрибутивного DVD-диска Solaris или Solaris Express или указав `cdrom` в качестве источника монтирования (для систем SPARC):

```
boot cdrom -s
```

Современная программа установки Solaris на платформе x86 (проверялось на Solaris Express Developer Edition 1/08) при обнаружении существующего раздела Solaris на диске предлагает его смонтировать как `/a`. Можно воспользоваться этим любезным предложением и после этого заняться редактированием файла `/a/etc/shadow`, как рассказано выше.