Канд.техн.наук С.Г. Семенцов, инж. Н.Р. Герман, инж. П.В. Саранцев

LINUX-ПОДОБНЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ В СИСТЕМАХ, ВАЖНЫХ ДЛЯ БЕЗОПАСНОСТИ АЭС

Проблема создания специализированной операционной системы (ОС) для КЭ СУЗ на базе платформ с открытыми исходными текстами является не только актуальной, но и существенно новой как для НПП ВНИИЭМ, так и для отрасли в целом, поскольку такая задача никогда не выполнялась ранее для оборудования АЭС, к которому предъявляются повышенные требования (в том числе: обеспечение отказоустойчивости, резервирования, быстрого восстановления после сбоев и отказов). В значительной мере всем этим требованиям удовлетворяют клоны операционной системы Unix, получившие название Unix-подобных ОС. Рассмотрим архитектуру и основные характеристики Unix-подобных операционных систем.

UNIX-подобные операционные системы

Операционная система Unix изначально создавалась как компактная многозадачная система для программных разработок. Практически все компоненты системы написаны на языке С и вследствие этого ОС проста для понимания и легко переносима на различные аппаратные платформы. Важнейшей особенностью этой операционной системы является компактность кода - всего около 12 млн. строк. После появления в конце 70-х годов миниЭВМ ОС Unix стала практически стандартом для этих машин, поскольку, обладая всей функциональностью коммерческих ОС (RSX, RT, VMS и др.) могла легко переноситься на различные платформы без существенных трудозатрат. С появлением в середине 80-х годов первых ПК возникли и версии ОС Unix для платформы Intel.

В отличие от классических операционных систем для персональных компьютеров (DOS, Windows, OS/2, NetWare), Unix является многопользовательской многозадачной операционной системой с разделением времени. Одна и та же вычислительная система под управлением ОС Unix может использоваться как сервер приложений, коммуникационный сервер, сервер печати или файлов и в то же время обслуживать запросы большого количества пользователей. В этом смысле Unix очень близка к таким известным ОС как

MVS, VMS, OS/400, т. е. серверных операционных систем. Несмотря на многообразие версий Unix, архитектура и основные интерфейсы системы достаточно жестко стандартизованы. Наличие многопользовательского режима работы является важнейшей отличительной чертой этой ОС. Данный режим позволяет значительно повысить уровень безопасности системы и организовывать работу пользователей через терминальный интерфейс, чрезвычайно экономичный и надежный. Крайне важным аспектом использования ОС Unix является то, что практически с самого начала своего существования она использовалась для организации сетевых вычислений. Все наиболее распространенные сетевые протоколы (TCP/IP, NFS) возникли и были отработаны в среде ОС Unix, и в последних версиях клонов этой ОС поддержка сетевых протоколов, а также таких интерфейсов как, например, USB реализована на уровне ядра.

До недавнего времени одним из немногих недостатков Unix-подобных ОС была высокая стоимость решений на их платформе. Это обстоятельство, безусловно, являлось сдерживающим фактором для широкого использования Unix-подобных ОС в системах управления различного назначения. Однако ситуация в корне изменилась в начале 90-х годов, когда появились и начали бурно развиваться бесплатные клоны ОС Unix, в частности операционная система Linux.

Операционная система Linux и принципы ее работы

ОС Linux - свободно распространяемая операционная система, создававшаяся при помощи большого числа программистов по всей сети Интернет. Linux обладает всеми свойствами современной Unix-подобной системы, включая многозадачность, развитую подсистему управления памятью и сетевую подсистему. Ядро Linux, поставляемое вместе с распространяемыми прикладными и системными программами образует полнофункциональную универсальную операционную систему. Большую часть базовых системных компонент Linux унаследовал от проекта GNU System V, целью которого является создание свободной микроядерной операционной системы. На сегодняшний день существует множество различных поставок Linux, дистрибутивов, которые можно разделить на дистрибутивы общего назначения и специализированные. К специализированным дистрибутивам относятся, в частности, и системы управления на базе Linux, такие как, например, LinuxRouter - поставка Linux для создания сетевого маршрутизатора на базе архитектуры PC/AT и др. В отличие от ядра, дистрибу-

тивы могут содержать коммерческие компоненты и потому их свободное распространение может быть ограничено. В таком случае авторы дистрибутива делают доступными все свободные компоненты. Обычно под словосочетанием "ОС Linux" понимают дистрибутивы Linux общего назначения. Модульная структура ОС Linux позволяет осуществлять сборку дистрибутива под каждую конкретную задачу, т.е. система может быть оптимизирована по критерию скорости, минимального размера и т.д. В зависимости от области применения системы в ее состав могут быть включены или удалены дополнительные системные пакеты, реализующие ту или иную функцию.

Одной из базовых особенностей Linux и Unix-подобных систем является концепция разделения прав пользователей, когда каждому пользователю системы разрешается определенный набор действий и доступ к определенным файлам и каталогам файловой системы.

Linux также поддерживает виртуальные консоли (virtual consoles), которые позволяют "переключать экраны" на консоли в текстовом режиме. Ядро может само эмулировать команды 387-FPU, так что системы без сопроцессора могут выполнять программы, на него рассчитывающие (т.е. с плавающей точкой). Ядро Linux сразу создано с учетом специального защищенного режима для процессоров Intel 80386 и 80486. В частности Linux использует парадигму описания памяти в защищенном режиме и другие новые свойства процессоров. Ядро Linux поддерживает загрузку только нужных страниц, т. е. с диска в память загружаются те сегменты программы, которые действительно используются. Возможно использование одной страницы, физически один раз загруженной в память, несколькими выполняемыми программами.

Для увеличения объема доступной памяти Linux осуществляет также разбиение диска на страницы: на диске может быть выделено до 256 Мбайт пространства для свопинга (swap space). Когда системе нужно больше физической памяти, то она с помощью свопинга выводит неактивные страницы на диск. Это позволяет выполнять более объемные программы и обслуживать одновременно больше пользователей. Однако свопинг не исключает наращивания физической памяти, поскольку он снижает быстродействие, увеличивает время доступа.

Выполняемые программы используют динамически связываемые библиотеки, т.е. выполняемые программы могут совместно использовать библиотечную программу, представленную одним физическим файлом на диске. Это позволяет выполняемым файлам занимать меньше места на диске, особенно тем, которые много-

кратно используют библиотечные функции. Для обеспечения отладки ядро Linux выдает дампы памяти для "посмертного" анализа. Использование дампа и динамических отладчиков позволяет определить причины краха программы.

Основной файловой системой Linux является технология ext2fs (ext3fs). Официальное ядро содержит поддержку более 20 различных файловых систем, включая FAT (FAT/VFAT/FAT32), ISO9660 (CDROM), HPFS (OS/2), NTFS (Windows NT), SysV (SCO Unix и др.), UFS (BSD и др.).

ОС Linux может служить файл сервером по протоколам NFS, SMB (Netbios over TCP/IP, используемый на различных Windows платформах), AppleShare и IPX (Novell). В ОС Linux также реализован интерфейс оконной системы X. XFree86 поддерживает многочисленные популярные графические адаптеры на платформе Intel и некоторых других. Оконная система X построена на клиент-серверной архитектуре, таким образом место запуска приложения и место его отображения могут быть физически разнесены по сети. В последнюю версию XFree86 - 4.0 входит поддержка OpenGL и аппаратной 3D акселерации.

Основные производители систем управления базами данных уже перенесли свои продукты на Linux: существуют Linux версии IBM DB2, Informix, Oracle, Sybase, Corel/Inprise Interbase и др. Linux может использоваться в качестве платформы для запуска системы управления предприятием SAP R/3.

Ядро операционной системы Linux

Одно из основных достижений Unix-подобных ОС состоит в том, что система обладает свойством высокой мобильности [1]. Смысл этого качества состоит в том, что вся операционная система, включая ее ядро, сравнительно просто переносится на различные аппаратные платформы. Все части системы, не считая ядра, являются полностью машинно-независимыми. Ядро также поддерживает универсальный пул памяти для пользовательских программ и дискового кэша. При этом для кэша может использоваться вся память, и наоборот, кэш уменьшается при работе больших программ.

К основным функциям ядра ОС Linux принято относить следующие [1]:

1. Инициализация системы - функция запуска и раскрутки. Ядро системы обеспечивает средство раскрутки (bootstrap), которое обеспечивает загрузку полного ядра в память компьютера и запускает ядро.

2. Управление процессами и нитями - функция создания, завершения и отслеживания существующих процессов и нитей ("процессов", выполняемых на общей виртуальной памяти). Поскольку ОС Linux является мультипроцессной операционной системой, ядро обеспечивает разделение между запущенными процессами времени процессора (или процессоров в мультипроцессорных системах) и других ресурсов компьютера для создания внешнего ощущения того, что процессы реально выполняются в параллельном режиме.

3. Управление памятью - функция отображения практически неограниченной виртуальной памяти процессов в физическую оперативную память компьютера, которая имеет ограниченные размеры. Соответствующий компонент ядра обеспечивает разделяемое использование одних и тех же областей оперативной памяти несколькими процессами с использованием внешней памяти.

4. Управление файлами - функция, реализующая абстракцию файловой системы, иерархии каталогов и файлов. Файловые системы ОС Linux поддерживают несколько типов файлов. Некоторые файлы могут содержать данные в формате ASCII, другие будут соответствовать внешним устройствам. В файловой системе хранятся объектные файлы, выполняемые файлы и т.д. Файлы обычно хранятся на устройствах внешней памяти; доступ к ним обеспечивается средствами ядра. В мире Linux существует несколько типов организации файловых систем.

5. Коммуникационные средства - функция, обеспечивающая возможности обмена данными между процессами, выполняющимися внутри одного компьютера (IPC - Inter-Process Communications), между процессами, выполняющимися в разных узлах локальной или глобальной сети передачи данных, а также между процессами и драйверами внешних устройств.

6. Программный интерфейс - функция, обеспечивающая доступ к возможностям ядра со стороны пользовательских процессов на основе механизма системных вызовов, оформленных в виде библиотеки функций.

Ядро системы Linux состоит из нескольких основных частей [2]: блок управления процессами, блок управления памятью, драйверы устройств, драйверы файловых систем, блок управления сетью, а также другие небольшие процедуры.

Наиболее важные составляющие ядра (обеспечивающие жизнеспособность системы) - это блок управления памятью и процессами. Блок управления памятью обеспечивает распределение областей памяти и swap-областей между процессами, составляющими

ядра и для кэш-буфера. Блок управления процессами создает новые процессы и обеспечивает многозадачность путем переключения задач.

Устойчивость системы обеспечивается за счет многоуровнего контроля системных процессов с помощью служебных модулей ядра [3]. Модуль распределения памяти контролирует выделение памяти процессам. Если в какой-то момент система испытывает недостаток в физической памяти для запуска всех процессов, ядро пересылает процессы между основной и внешней памятью с тем, чтобы все процессы имели возможность выполняться.

Модуль "планировщик" распределяет между процессами время центрального процессора, обеспечивая, таким образом, многозадачность ОС. Он планирует очередность выполнения процессов до тех пор, пока они добровольно не освободят центральный процессор, дождавшись выделения какого-либо ресурса, или до тех пор, пока ядро системы не выгрузит их после того, как их время выполнения превысит заранее определенный квант времени. Планировщик выбирает на выполнение готовый к запуску процесс с наивысшим приоритетом; выполнение предыдущего процесса (приостановленного) будет продолжено тогда, когда его приоритет будет наивысшим среди приоритетов всех готовых к запуску процессов. Существует несколько форм взаимодействия процессов между собой, от асинхронного обмена сигналами о событиях до синхронного обмена сообщениями.

Наконец аппаратный контроль отвечает за обработку прерываний и за связь с машиной. Такие устройства, как диски и терминалы, могут прерывать работу центрального процессора во время выполнения процесса. При этом ядро системы после обработки прерывания может возобновить выполнение прерванного процесса. Прерывания обрабатываются не самими процессами, а специальными функциями ядра системы, перечисленными в контексте выполняемого процесса.

На самом нижнем уровне ядро содержит драйверы устройств для каждого типа поддерживаемого оборудования. Существует довольно большой набор различных драйверов, так как постоянно разрабатываются новые типы устройств. Существует довольно много одинаковых устройств, которые различаются только тем, как происходит взаимодействие между самим устройством и драйвером. Такое сходство позволяет использовать классы драйверов, поддерживающих одинаковые операции. В каждом члене такого класса используется однотипный интерфейс для ядра, но различные схемы взаимодействия с устройством. Например все драйверы

жесткого диска представляются для ядра абсолютно одинаково, т.е. у них у всех имеются такие операции как 'инициализация жесткого диска', 'чтение сектора N', 'запись сектора N'.

Файловая структура системы

Понятие файла и файловой системы является одним из наиболее важных для ОС Linux [3]. Все файлы, с которыми могут работать пользователи, располагаются в файловой системе, представляющей собой дерево, промежуточные вершины которого соответствуют каталогам и листья - файлам и пустым каталогам. На самом деле, на каждом логическом диске (разделе физического дискового пакета) располагается отдельная иерархия каталогов и файлов. Для получения общего дерева в динамике используется монтирование отдельных иерархий к фиксированной корневой файловой системе. Файловая система обычно размещается на дисках или других устройствах внешней памяти, имеющих блочную структуру. Кроме блоков, сохраняющих каталоги и файлы, во внешней памяти поддерживается еще несколько служебных областей.

Построение структуры каталогов изначально предполагает ее разбиение на отдельные части, каждая из которых может размещаться на отдельном диске или его разделе. Это используется для облегчения контроля объема диска, создания резервных копий и других обязанностей системного администратора. Основными частями являются файловые системы root, /usr, /var и /home. Структура каталогов разрабатывалась также для работы в сети, где возможно распределение ее некоторых частей посредством какого-либо устройства (например CD-ROM) или сети с использованием NFS. В дистрибутиве, разработанном в НПП ВНИИЭМ (ОС LiNEM), были использованы наработки по использованию системных директорий, а также добавлены новые для успешного функционирования прикладных программ.

Для работы с любой файловой системой в ОС Linux ее сначала необходимо подмонтировать, т.е. поставить в жесткое соответствие с тем или иным каталогом корневой файловой системы. Во время выполнения системного вызова монтирования корневой каталог монтируемой файловой системы совмещается с каталогом - точкой монтирования, в результате чего образуется новая иерархия с полными именами каталогов и файлов. Необходимо отметить, что ОС Linux обеспечивает работу в режиме чтение/запись не со всеми типами файловых систем, список которых приведен в п. 2. В частности для файловой системы NTFS поддерживается только режим чтения.

Для обеспечения необходимого для систем управления уровня безопасности при работе с файловой системой необходимо выполнение ряда требований. Прежде всего, это ограничение прав пользователей на запуск каких-либо программ, кроме заранее определенных, т.е. запуск SUID/SGID программ из пользовательских домашних каталогов. Также необходима установка лимита использования файловой системы для каждого пользователя. В большинстве Linux-подобных систем по умолчанию разрешено неограниченное использование файловой системы. Контроль лимитов каждого пользователя возможен с помощью специального модуля лимитов ресурсов PAM и файла limits.conf. Например лимиты для группы 'users' могут запретить создание core-файлов, ограничить количество процессов значением 50 и установить квоту использования памяти в размере 5Мбайт на пользователя.

Для предотвращения случайного удаления или перезаписи файлов, которые должны быть защищены, можно использовать имунный бит. Он также предотвращает создание кем бы то ни было символьной связи на этот файл, что является одним из методов атаки с целью удаления /etc/passwd или /etc/shadow. Также необходимо четко определить перечень файлов, для которых право на запись имеют все пользователи.

Графический интерфейс пользователя

В мире ОС Linux предпринималось несколько попыток создания оконных систем, и большинство из них успешно использовалось практически. Однако ни одна из этих систем не выходила за пределы ведомственного использования, что, естественно, резко ограничивало мобильность программ, обладающих графическим интерфейсом. Успеха удалось добиться группе программистов из Массачусетского технологического института, которые создали оконную систему под кратким и предельно скромным названием X [4]. В то же время сегодня именно оконная система X является базовым механизмом организации графических интерфейсов пользователя в большинстве Linux-подобных систем. Как и в классических системах Unix, в ОС Linux, пользовательский интерфейс не встраивается в ядро системы. Вместо этого он представляется программами пользовательского уровня. Этот подход применяется как к текстовым, так и к графическим оболочкам. Такой стандарт делает систему более гибкой, хотя и имеет свои недостатки, например затрудняет изучение системы.

Первоначально используемой с ОС семейства Unix графической оболочкой была система X Window System (сокращенно X). X не

реализует пользовательский интерфейс, а только оконную систему, т.е. средства, с помощью которых может быть реализован графический интерфейс. Три наиболее популярных версии графических интерфейсов на основе X - это Athena, Motif и Open Look. Motif (официальное название этого продукта - OSF/Motif) представляет собой программный пакет, включающий оконный менеджер, набор вспомогательных утилит, а также библиотеку классов, построенных на основе Xt Intrinsics. Motif также является мощным инструментом для разработки графических приложений под Linux. Он существенно расширяет возможности Xt Intrinsics, поскольку в его библиотеке поддерживается большое число классов, позволяющих создавать меню, "нажимаемые" кнопки и т.д. Также к преимуществам Motif можно отнести развитую библиотеку классов языка Си++, возможность применения этих классов при использовании обычного стиля программирования и поддержку визуального программирования с немедленным отображением получающихся графических объектов.

Подключение к системе через сеть

В ОС Linux, как и у любой другой ОС семейства Unix, существует брандмауэр (межсетевой экран) значительно снижающий риск заражения вирусом через программу работающую сервисом в локальной сети. Организация сетевого взаимодействия (платформа ОС Linux) происходит несколько иначе, чем обычное подключение в других системах (Novell Netware \ Windows NT). Существуют отдельные физические последовательные линии для каждого терминала, через которые и происходит подключение. Для каждого пользователя, подключающегося к системе, может существовать любое количество отдельных виртуальных сетевых соединений. Существуют также и другие способы подключения к системе посредством сети. Например telnet и rlogin - основные службы в TCP/IP сетях.

В среде Linux невозможен автоматический запуск файлов Windows – exe, com, pif, dll, поскольку в Linux-подобных ОС признаком исполняемости файла является не определенное расширение, а специальный атрибут, не передающийся через почтовый сервис.

Как дополнительное средство обеспечения безопасности все системы на базе Linux имеют развитую схему фильтрации пакетов (statefull firewall), которая позволяет защититься от распространения вирусов напрямую, в обход почтовых и файловых серверов.

При организации сети для подключения к системе используется отдельная программа-демон, которая отслеживает все попытки со-

единения с компьютером. Если устанавливается попытка соединения, то программа создает новый процесс - создает сама себя для обработки этого соединения, а затем продолжает отслеживание новых соединений. Такой подход позволяет значительно повысить безопасность работы с сетью и ограничить доступ к сети извне.

ЛИТЕРАТУРА

1. Кузнецов С.Д. Операционная система Unix. М.: Мир. 1999.
2. Операционная система Unix (руководство пользователя) // www.null.ru. 1998.
3. Lars Wirzenius. Руководство системного администратора ОС Linux // www.citforum.ru. 2001.
4. Larry Greenfield. Руководство пользователя Linux // www.linuxdoc.ru. 1994.