

Оглавление

1	Общая архитектура системы.....	1
1.1	База стандартов Linux	1
1.2	Архитектура системы	2
1.3	Основные подсистемы ядра	2

1 Общая архитектура системы

1.1 База стандартов Linux

(**Linux Standards Base**, www.linuxbase.org) представляет собой проект, ставящий целью стандартизацию самой операционной системы Linux для того, чтобы приложения, написанные для одного дистрибутива, соответствующего стандарту LSB, точно также компилировались и выполнялись в любом другом LSB-совместимом дистрибутиве.

База стандартов Linux содержит общие стандарты компонентов операционной системы, включая библиотеки, форматы пакетов и дистрибутивов, команд и утилит.

Например, стандарт LSB определяет стандартную структуру файловой системы. База стандартов Linux также включает архитектурно-зависимые стандарты, необходимые для сертификации на соответствие LSB.

Желающие протестировать и сертифицировать дистрибутив на соответствие базе стандартов Linux могут приобрести необходимые для этого утилиты и пройти сертификацию, обратившись к организации LSB (опять-таки, -за определенную плату).

До последнего времени задача соответствия стандартам имела достаточно низкий приоритет для разработчиков ядра, поскольку в первую очередь они стремились расширить функциональные возможности и надежность работы Linux. Вследствие этого, большинство версий ядра не согласуются ни с одним из наборов стандартов.

1.2 Архитектура системы

Хотя ядро Linux является монолитным по своей природе, последние усовершенствования, внесенные для обеспечения масштабируемости ядра, включают в себя возможности модульности, подобные тем, которые поддерживаются операционными системами с микроядром.

Linux часто называют UNIX-подобной операционной системой, поскольку в ней существуют многие службы, характерные для таких UNIX-систем, как UNIX System V от компании AT&T или BSD, разработанной в Беркли.

Ядро Linux состоит из шести основных подсистем:

- **управления процессами,**
- **взаимодействия между процессами,**
- **управления памятью,**
- **управления файловой системой,**
- **управления операциями ввода/вывода и**
- **сетевой подсистемы.**

Все шесть подсистем контролируют доступ к системным ресурсам.

Процессы в Linux могут выполняться в режиме ядра либо пользовательском режиме.

Пользовательские процессы выполняются в пользовательском режиме, поэтому доступ к службам ядра они получают через интерфейс системных вызовов. Когда от пользовательского процесса приходит разрешенный системный вызов (в пользовательском режиме), ядро обрабатывает системный вызов в режиме ядра от имени процесса. Если запрос некорректен (например, процесс пытается осуществить запись в файл, который не был открыт), ядро возвратит ошибку.

1.3 Основные подсистемы ядра

Диспетчер процессов (**process manager**) является главной подсистемой Linux, отвечающей за создание процессов, обеспечение доступа к процессору

(процессорам) системы и удалению процессов из системы по завершению их работы.

Подсистема взаимодействия процессов (interprocess communication, IPC) ядра позволяет процессам взаимодействовать друг с другом. Эта подсистема работает совместно с диспетчером процессов, обеспечивая совместный доступ к информации и передачу сообщений с помощью самых разнообразных методов.

Подсистема управления памятью обеспечивает процессам доступ к памяти. Linux выделяет каждому процессу виртуальное адресное пространство, которое делится на пользовательское адресное пространство и пространство ядра.

Включение пространства ядра в контекст каждого процесса уменьшает затраты на переключение между режимами ядра и пользователя, поскольку ядро может получить доступ к своим данным из виртуального адресного пространства любого процесса.

Пользователи получают доступ к файлам и папкам, перемещаясь по дереву каталогов. Корень дерева каталогов называется корневым (root) каталогом.

Из корневого каталога пользователи могут перейти к любой доступной файловой системе.

Пользовательские процессы запрашивают данные файловой системы через интерфейс системных вызовов.

Когда системе нужен доступ к файлу или папке дерева каталогов, взаимодействие осуществляется через интерфейс виртуальной файловой системы (virtual file system, VFS), обеспечивающий единый способ доступа ко всем файлам и каталогам, размещенным в неоднородных файловых системах (например, ext2 и NFS). Виртуальная файловая система передает запросы конкретной файловой системе, которая отвечает за схему размещения и место хранения данных.

Основываясь на модели UNIX, операционная система Linux взаимодействует с устройствами, как с файлами, то есть использует те же механизмы доступа к данным, что и при работе с файлами.

Когда пользовательские процессы обмениваются данными с устройством, ядро передает запросы интерфейсу виртуальной файловой системы, которая перенаправляет их интерфейсу ввода/вывода. Интерфейс ввода/вывода передает запросы далее, драйверам устройств, выполняющим операции ввода/вывода для системного оборудования.

В Linux существует **сетевая подсистема**, позволяющая процессам обмениваться данными с компьютерами по сети. Для отправки и приема пакетов сетевая подсистема использует сетевое оборудование системы, получая доступ к нему через интерфейс ввода/вывода. Эта подсистема позволяет приложениям и ядру инспектировать и модифицировать пакеты, проходящие по сетевым уровням системы с помощью интерфейса фильтрации пакетов. Данный интерфейс позволяет реализовать брандмауэры, маршрутизаторы и другие сетевые средства.