ПОНЯТИЕ КАТАЛОГА: ИЕРАРХИЧЕСКИЕ СИСТЕМЫ КАТАЛОГОВ, ОПЕРАЦИИ С

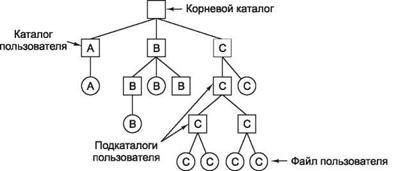
КАТАЛОГАМИ

Обычно в файловой системе для упорядочения файлов имеются **каталоги** или **папки**, которые сами по себе являются файлами.

**Иерархические системы каталогов**

Поэтому нужен способ, позволяющий сгруппировать родственные файлы. К примеру, у профессора может быть коллекция файлов, составляющая книгу, которую он написал для одного учебного курса, вторая коллекция файлов, в которой содержатся студенческие программы, сданные при изучении другого курса, третья группа файлов, в которых со- держится создаваемая им инновационная система компиляции, четвертая группа файлов, в которых содержатся предложения по грантам, а также другие файлы, используемые для электронной почты, протоколов совещаний, неоконченных статей, игр и т. д.

Для организации всего этого нужна иерархия (то есть дерево каталогов). Такой подход позволяет иметь столько каталогов, сколько необходимо для группировки файлов есте- ственным образом. Кроме того, если общий файловый сервер совместно используется несколькими пользователями, как это бывает во многих сетях организаций, каждый пользователь может иметь корневой каталог для собственной иерархии.



**Операции с каталогами**

Допустимые системные вызовы для управления каталогами имеют большее количество вариантов от системы к системе, чем системные вызовы, управляющие файлами. Рас- смотрим примеры, дающие представление об этих системных вызовах и характере их работы (взяты из системы UNIX).

⎝ *Create* (Создать каталог). Каталог создается пустым, за исключением точки и двой- ной точки, которые система помещает в него автоматически (или в некоторых случаях при помощи программы mkdir).

⎝ *Delete* (Удалить каталог). Удалить можно только пустой каталог. Каталог, содержа- щий только точку и двойную точку, рассматривается как пустой, поскольку они не могут быть удалены.

⎝ *Opendir* (Открыть каталог). Каталоги могут быть прочитаны. К примеру, для вы- вода имен всех файлов, содержащихся в каталоге, программа ls открывает каталог

4.3. Реализация файловой системы **319** для чтения имен всех содержащихся в нем файлов. Перед тем как каталог может

быть прочитан, он должен быть открыт по аналогии с открытием и чтением файла.

* ⎝  *Closedir* (Закрыть каталог). Когда каталог прочитан, он должен быть закрыт, чтобы освободить пространство во внутренних таблицах системы.
* ⎝  *Readdir* (Прочитать каталог). Этот вызов возвращает следующую запись из от- крытого каталога. Раньше каталоги можно было читать с помощью обычного системного вызова *read*, но недостаток такого подхода заключался в том, что про- граммист вынужден был работать с внутренней структурой каталогов, о которой он должен был знать заранее. В отличие от этого, *readdir* всегда возвращает одну запись в стандартном формате независимо от того, какая из возможных структур каталогов используется.
* ⎝  *Rename* (Переименовать каталог). Во многих отношениях каталоги подобны файлам и могут быть переименованы точно так же, как и файлы.
* ⎝  *Link* (Привязать). Привязка представляет собой технологию, позволяющую файлу появляться более чем в одном каталоге. В этом системном вызове указываются существующий файл и новое имя файла в некотором существующем каталоге и создается привязка существующего файла к указанному каталогу с указанным новым именем. Таким образом, один и тот же файл может появиться в нескольких каталогах, возможно, под разными именами. Подобная привязка, увеличивающая показания файлового счетчика i-узла (предназначенного для отслеживания коли- чества записей каталогов, в которых фигурирует файл), иногда называется **жесткой связью**, или **жесткой ссылкой** (hard link).
* ⎝  *Unlink* (Отвязать). Удалить запись каталога. Если отвязываемый файл присутству- ет только в одном каталоге (что чаще всего и бывает), то этот вызов удалит его из файловой системы. Если он фигурирует в нескольких каталогах, то он будет уда- лен из каталога, который указан в имени файла. Все остальные записи останутся. Фактически системным вызовом для удаления файлов в UNIX (как ранее уже было рассмотрено) является *unlink*.