

# КАТАЛОГ

# ФАЙЛЫ И ФАЙЛОВЫЕ СТРУКТУРЫ.

Чтобы прочитать содержимое файла, необходимо знать его местоположение на дисковом устройстве. Каждый файл занимает на диске определенную группу секторов. Следовательно, местоположение файла можно задавать, указывая номера секторов и дорожек, занятых файлом. Однако такой способ указания местоположения файла очень неудобен, так как в этом случае пользователю необходимо знать номера всех секторов диска, которые отведены под файл. Для повышения эффективности обмена данными несколько подряд расположенных секторов объединяются в *кластер*, и обмен осуществляют сразу всей группой секторов. Такая схема организации обмена существенно увеличивает скорость исполнения операций обмена данными с жесткими дисками. Чтобы не задавать три отдельных числа (номер рабочей поверхности, номер дорожки и номер сектора) в качестве адреса сектора, с которого начинается кластер, для всех кластеров диска введена единая, сплошная нумерация. Для определения кластера, в котором начинается файл, достаточно указывать только одно число - порядковый номер кластера на диске.

# ФАЙЛЫ И ФАЙЛОВЫЕ СТРУКТУРЫ.

**Каталогом** называется таблица файловой системы диска, которая содержит список всех записанных на этот диск файлов. Для каждого файла в этой таблице указываются значения всех его атрибутов, а также номер первого выделенного файлу кластера. Сточки зрения своего назначения каталог можно сравнить с оглавлением в книге, в котором для каждой главы указан начальный номер страницы, или с описью документов, хранящихся в шкафу. Как в книге для определения положения той или иной главы можно по названию главы в содержании книги определить, на какой странице она начинается, так и операционная система по названию файла находит в каталоге кластер, в котором он начинается.

Аналогия между каталогом и оглавлением в книге только частичная из-за того, что кластеры выделяются файлу на диске не сплошным массивом, а в разброс, в то время как в книге все страницы главы размещаются подряд. Представьте себе, одна из глав книги занимает страницы 5, 15, 16, 17, 31, 123, 124 вместо того, чтобы занимать страницы 5, 6, 7, 8, 9, 10, 11 подряд. Такое не сплошное выделение кластеров файлам организовано для того, чтобы оптимизировать использование свободного пространства диска при многочисленных уничтожениях и записях файлов.

Для того чтобы все-таки знать, какие именно кластеры и в каком порядке выделены для хранения файла, в файловой системе предусмотрена таблица размещения файлов (FAT). Каталог содержит только номер начального кластера файла. А таблица FAT - номера всех остальных занятых файлом кластеров. В подавляющем большинстве случаев пользователю не приходится работать с таблицей FAT, так как она заполняется при записи файла и анализируется при его считывании автоматически.

# ФАЙЛЫ И ФАЙЛОВЫЕ СТРУКТУРЫ.

Для кластеров существует линейная адресация: все кластеры пронумерованы от 1 до  $2^n$  (здесь  $n$  - разрядность FAT). Для 16-разрядной FAT количество кластеров на диске составляет  $2^{16} = 65536$ . Не трудно вычислить, что для дисков емкостью 1 Гбайт кластер составляет 32 Кбайта.

Размер современных жестких дисков, как правило, превышает 1 Гбайт. При записи информации на такие диски значительная часть дискового пространства может тратиться впустую, поскольку, например, в случае 16-разрядной FAT файлы размером 31 Кбайт и менее 1 Кбайта занимают каждый одинаковое пространство на диске - 32 Кбайта. Неиспользованное пространство кластера называется «кластерным выступом». Потери на кластерные выступы тем больше, чем большее количество малых файлов записано на диске.

Самый естественный путь для повышения эффективности использования кластеров - это уменьшение их размеров. В настоящее время используется файловая система FAT32, в которой используется 232 кластера.

Рассмотренная выше простая структура каталога, в котором все файлы образуют один общий список, может обеспечить удовлетворительную работу операционной системы только в случае небольших объемов диска и ограничивает общее число файлов, которые могут быть записаны на диск. Так, на гибких дисках объемом 1,44 Мбайт корневой каталог может содержать сведения не более чем о 224 файлах. А когда объем диска становится достаточно большим и, следовательно, на диске могут быть записаны сотни и тысячи файлов, простая структура каталога приводит к существенному замедлению процесса поиска файла на диске или переполнению каталога.

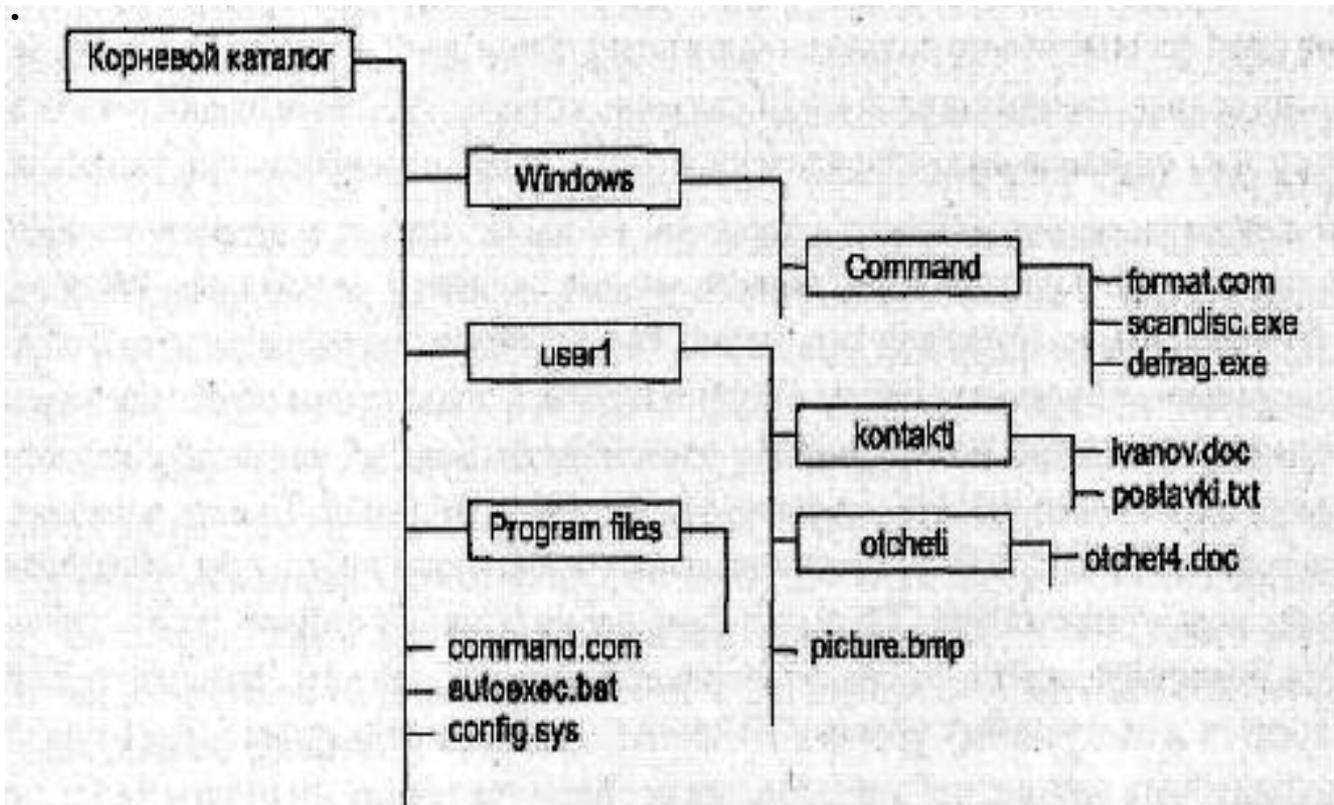
# ФАЙЛЫ И ФАЙЛОВЫЕ СТРУКТУРЫ.

Каталог в операционных системах имеет более сложную структуру. Произвольные группы файлов каталога могут объединяться и образовывать *подкаталоги*. В некоторых операционных системах подкаталоги называются *папками*. Фактически подкаталоги, как и корневой каталог, являются таблицами, размещаемыми на диске и содержащими информацию об отнесенных к подкаталогу файлах. В отличие от корневого каталога положение подкаталогов на диске не привязано к системной области. Поэтому размеры подкаталогов могут быть достаточно произвольными, что позволяет снять ограничение на количество указываемых в подкаталоге файлов.

Подкаталоги создаются пользователями по своему усмотрению. Каждый подкаталог имеет собственное имя (обычно без расширения), которое подбирается по тем же правилам, что и имя файла.

Группировка и включение файлов в подкаталог могут производиться по любым критериям. Например, в отдельный подкаталог с названием WINDOWS (рис. 3.3) целесообразно собрать все файлы, имеющие отношение к операционной системе. Точно так же целесообразно сгруппировать в отдельный подкаталог все файлы, необходимые для работы какого-либо текстового редактора или игровой программы. Если на машине по очереди работают несколько пользователей, то имеет смысл организовать отдельные подкаталоги для каждого пользователя. Например, назвать подкаталоги именами: user1, user2, user3,... (user – пользователь), сгруппировав в подкаталоге user1 файлы первого пользователя, в подкаталоге user2 – второго и т.д. Кроме снятия количественных ограничений, связанных с использованием одного каталога, это создает определенную упорядоченность при хранении информации на дисках.

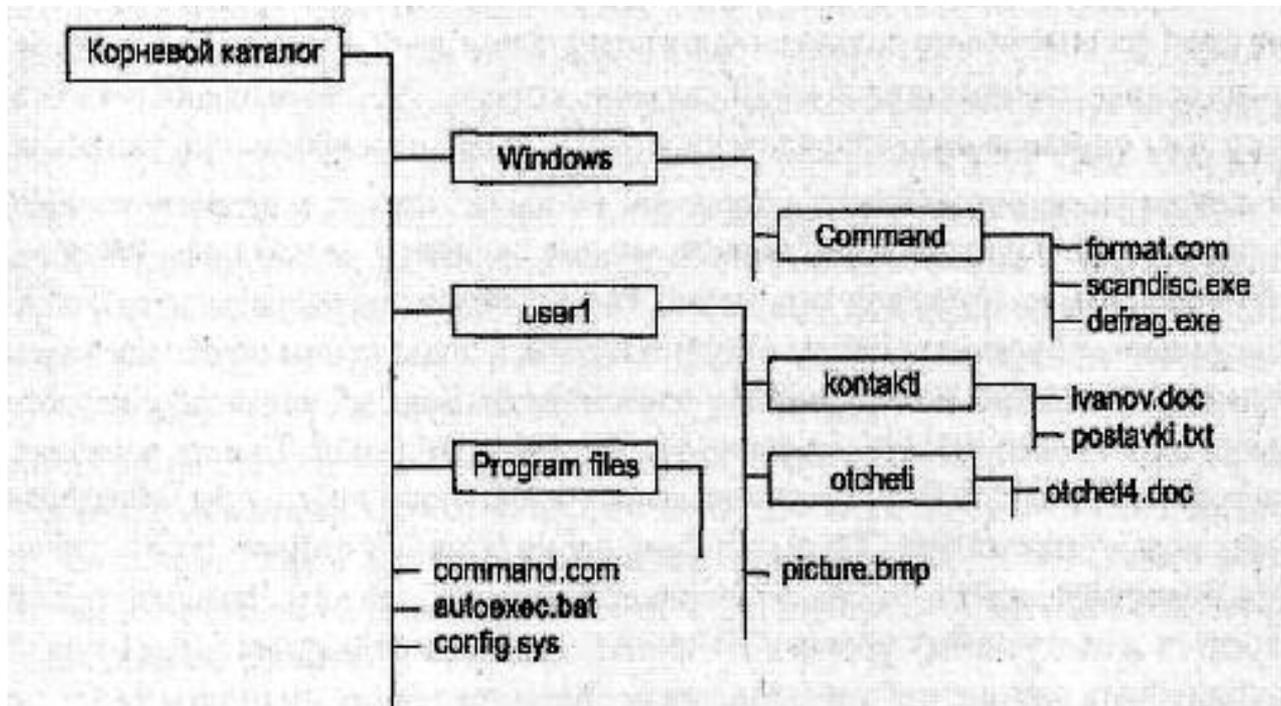
Все подкаталоги, находящиеся в корневом каталоге, относят к первому уровню. На рисунке



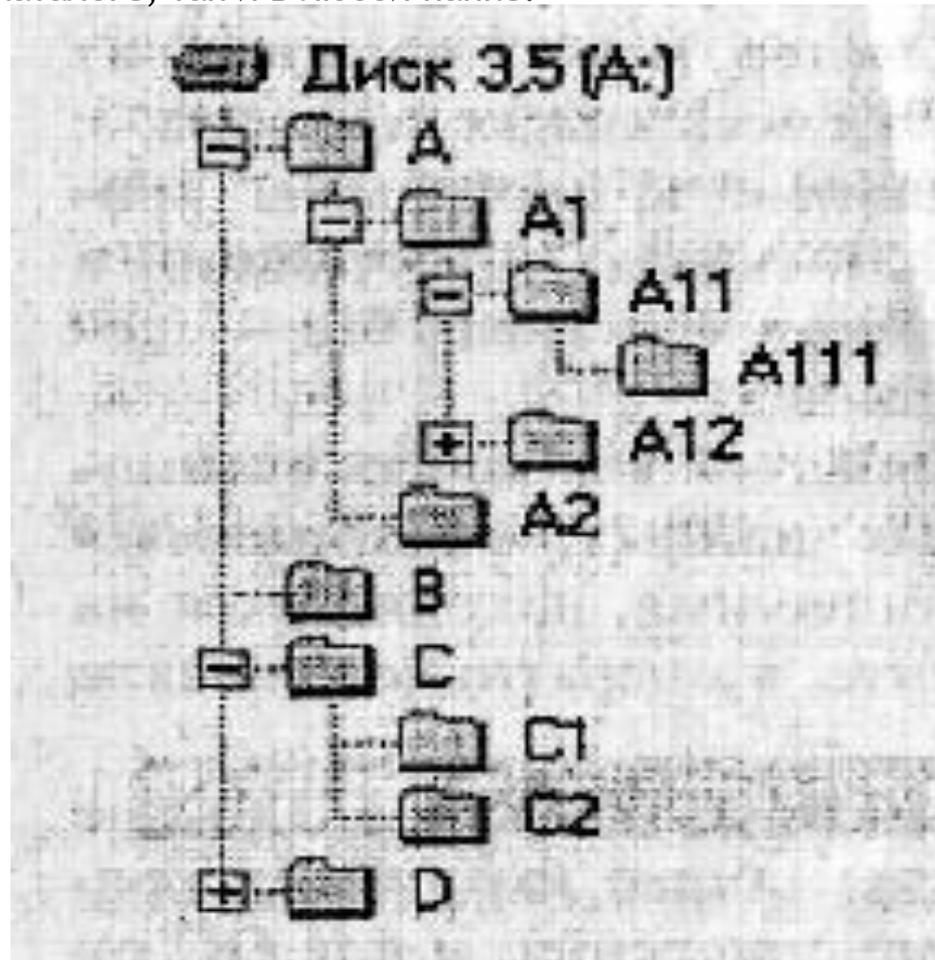
подкаталогами первого уровня являются подкаталоги Windows, user1, Program files. Корневой каталог по отношению к включенным в него подкаталогам первого уровня называют *родительским*, а подкаталоги по отношению к корневому считаются *дочерними* или *вложенными*

- Каждый подкаталог первого уровня в свою очередь устроен точно так же, как и корневой. В подкаталоге первого уровня могут быть организованы подкаталоги второго уровня и т.д. Например, владелец подкаталога user1 может сгруппировать внутри этого подкаталога все подготовленные им отчеты в отдельный подкаталог с названием otcheti, а, скажем, файлы, содержащие информацию о деловых контактах, собрать в подкаталоге kontakti. Подкаталоги первого уровня по отношению к включенным в них подкаталогам второго уровня считаются родительскими. Подкаталоги второго уровня выступают в роли дочерних по отношению к подкаталогам первого уровня.

Каталог по своей структуре напоминает дерево. Корневой каталог можно сопоставить со стволом дерева, подкаталоги играют роль ветвей, а файлы являются листьями этого «дерева». Такая структура каталога называется *древовидной* или *иерархической*.



В операционных системах с графическим интерфейсом каталоги изображаются в виде папок. На рисунке показано дерево папок одного из дисков. Из рис. 3.4 видно, что в корневом каталоге имеется четыре папки: А, В, С и D. При этом внутри папки А находятся папки А1 и А2. В папке А1 находятся папки А11, а в последней - папка А111. В папке А12 находится папка А111, а в последней - папка А1111. В папке А2 находятся папки А21 и А22. В папке С располагаются папки С1 и С2. В папке D находятся папки D1 и D2. Крестик на дереве говорит о том, что внутри соответствующих папок находятся другие папки (внутри папок D и А12 находятся папки, которые не видны). На этом рисунке не видны файлы, которые могут находиться как в корневом каталоге, так и в любой папке.



# ФАЙЛЫ И ФАЙЛОВЫЕ СТРУКТУРЫ.

**Спасибо за внимание.**