

# Операционные системы

## Автор В.А.Серков

Подсистема управления  
файлами

## **Подсистема управления файлами**

(или файловая система)- это часть операционной системы, назначение которой состоит в том, чтобы обеспечить пользователю удобный интерфейс при работе с данными, хранящимися на диске, и обеспечить совместное использование файлов несколькими пользователями и процессами.

# В широком смысле понятие "файловая система" включает:

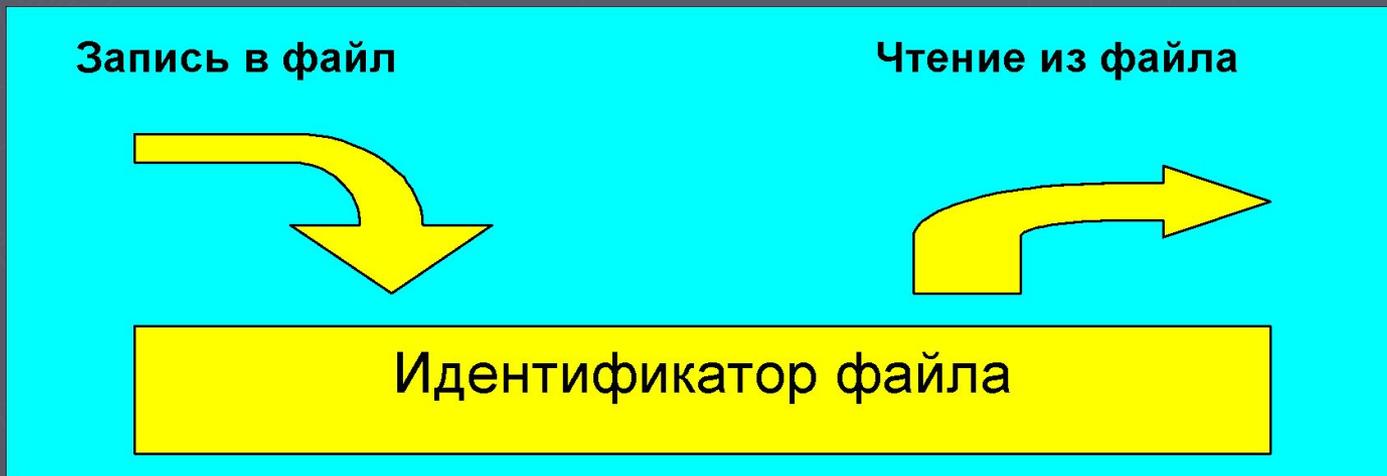
Совокупность всех файлов на диске.

Наборы структур данных, используемых для управления файлами, такие, например, как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске.

Комплекс системных программных средств, реализующих управление файлами (создание, уничтожение, чтение, запись, именованное, поиск и другие операции над файлами).

Файл представляет собой поименованное место на диске (ленте), куда можно записать или откуда можно прочитать информацию.

Файлы идентифицируются именами. Пользователи дают файлам символьные имена, при этом учитываются ограничения ОС как на используемые символы, так и на длину имени.



# Типы файлов

**Обычные файлы** в свою очередь подразделяются на текстовые и двоичные. Текстовые файлы состоят из строк символов, представленных в ASCII-коде. Это могут быть документы, исходные тексты программ и т.п. Текстовые файлы можно прочитать на экране и распечатать на принтере. Двоичные файлы не используют ASCII-коды, они часто имеют сложную внутреннюю структуру, например, объектный код программы или архивный файл. Все операционные системы должны уметь распознавать хотя бы один тип файлов - их собственные исполняемые файлы.

# Типы файлов

**Специальные файлы** - это файлы, ассоциированные с устройствами ввода-вывода, которые позволяют пользователю выполнять операции ввода-вывода, используя обычные команды записи в файл или чтения из файла.

Эти команды обрабатываются вначале программами файловой системы, а затем на некотором этапе выполнения запроса преобразуются ОС в команды управления соответствующим устройством.

Специальные файлы, так же как и устройства ввода-вывода, делятся на блок-ориентированные и байт-ориентированные.

# Типы файлов

**Каталог** - это, с одной стороны, группа файлов, объединенных пользователем исходя из некоторых соображений (например, файлы, содержащие программы игр, или файлы, составляющие один программный пакет), а с другой стороны - это файл, содержащий системную информацию о группе файлов, его составляющих.

# Каталог

В каталоге содержится список файлов, входящих в него, и устанавливается соответствие между файлами и их характеристиками (атрибутами).

В разных файловых системах могут использоваться в качестве атрибутов разные характеристики, например:

1. информация о разрешенном доступе,
2. пароль для доступа к файлу,
3. владелец файла,
4. создатель файла,
5. признак "только для чтения",
6. признак "скрытый файл",
7. признак "системный файл",
8. признак "архивный файл",
9. признак "двоичный/символьный",
10. признак "временный" (удалить после завершения процесса),
11. признак блокировки,
12. длина записи,
13. указатель на ключевое поле в записи,
14. длина ключа,
15. времена создания, последнего доступа и последнего изменения,
16. текущий размер файла,
17. максимальный размер файла.

Каталоги могут непосредственно содержать значения характеристик файлов, как это сделано в файловой системе MS-DOS (а), или ссылаться на таблицы, содержащие эти характеристики, как это реализовано в ОС UNIX (б).

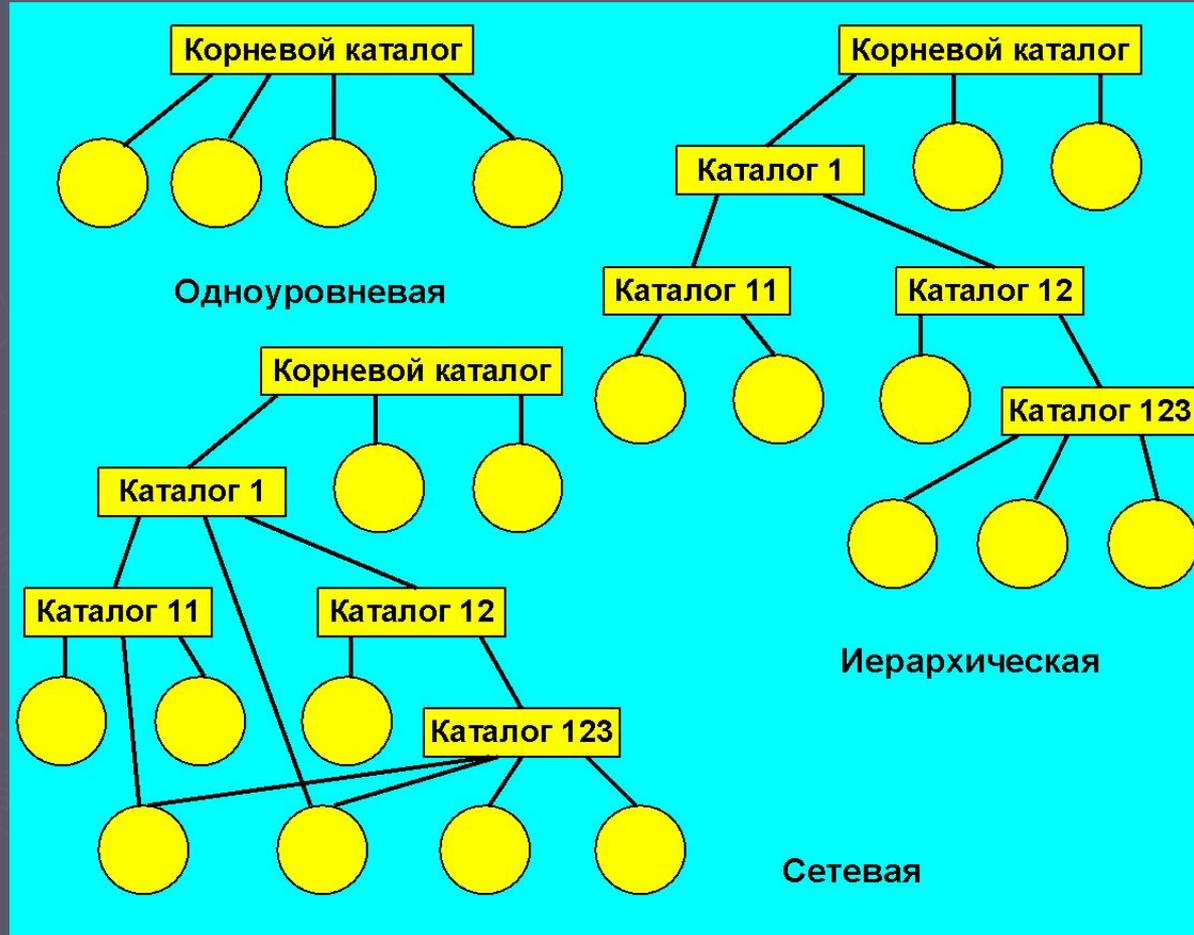
Смещение	0-7	8-10	11	12-21	22-23	24-25	26-27	28-31
	Имя файла	Расширение	Атрибут	Не используется	Время	Дата	Начальный блок	Размер файла
Размер	8 байт	3 байт	1 байт	10 байт	2 байт	2 байт	2 байт	4 байт

а

2	14
Номер индексного дескриптора	Имя файла

б

# Логическая организация файловых систем



# Логическая организация файла

Программист имеет дело с логической организацией файла, представляя файл в виде определенным образом организованных логических записей.

Логическая запись - это наименьший элемент данных, которым может оперировать программист при обмене с внешним устройством.

Даже если физический обмен с устройством осуществляется большими единицами, операционная система обеспечивает программисту доступ к отдельной логической записи.

# Способы логической организации файлов

Запись 1	Запись 2	Запись 3	...	Запись N
----------	----------	----------	-----	----------

Последовательность логических записей фиксированной длины

L1	Запись 1	L2	Запись 2	...	LN	Запись N
----	----------	----	----------	-----	----	----------

Последовательность логических записей переменной длины

Индексная область	2	Запись 1	3	Запись 2	1	Запись 3	4	Запись 4
-------------------	---	----------	---	----------	---	----------	---	----------

Таблица индексов

Индекс	1	2	3	4
Адрес	241	201	221	261

Индексная логическая организация

# Примеры

Байт 0
Байт 1
Байт 2
Байт 3
Байт 4
Байт 5
Байт 6
Байт 7
...
Байт N
EOF

Записи  
фиксированной  
длины

Запись 0	EOLN
Запись 1	EOLN
Запись 2	EOLN
Запись 3	EOLN
Запись 4	EOLN
Запись 5	EOLN
Запись 6	EOLN
Запись 7	EOLN
...	
Запись N	EOLN
EOF	

Записи  
переменной  
длины

# Пример

Индексная обл.

Андреев	371
Белов	281
<b>Захаров</b>	<b>201</b>
Суворов	401
Харламов	331
Юдашкин	451
Ягодин	241

Основная область

<b>201</b>	<b>Захаров</b>	<b>Данные</b>
241	Ягодин	Данные
281	Белов	Данные
331	Харламов	Данные
371	Андреев	Данные
401	Суворов	Данные
451	Юдашкин	Данные

# Физическая организация и адрес файла

Физическая организация файла описывает правила расположения файла на устройстве внешней памяти, в частности на диске. Файл состоит из физических записей – блоков (кластеров). Блок - наименьшая единица данных, которой внешнее устройство обменивается с оперативной памятью.

№ сект.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
№ блока	0		1		2		3		4		5		6		7	

№ сект.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
№ блока	0				1				2				3			

№ сект.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
№ блока	0								1							

# Непрерывное размещение

В этом случае содержимое файла разбивается на фрагменты, размеры которых соответствуют размерам блока.

Фрагменты размещаются в соседних блоках дискового пространства.

Если в процессе «жизни» размеры файла увеличиваются и нет возможности занять соседние блоки, ФС размещает файл в другой области дискового пространства.

Главный недостаток – часто возникающая фрагментация дискового пространства.

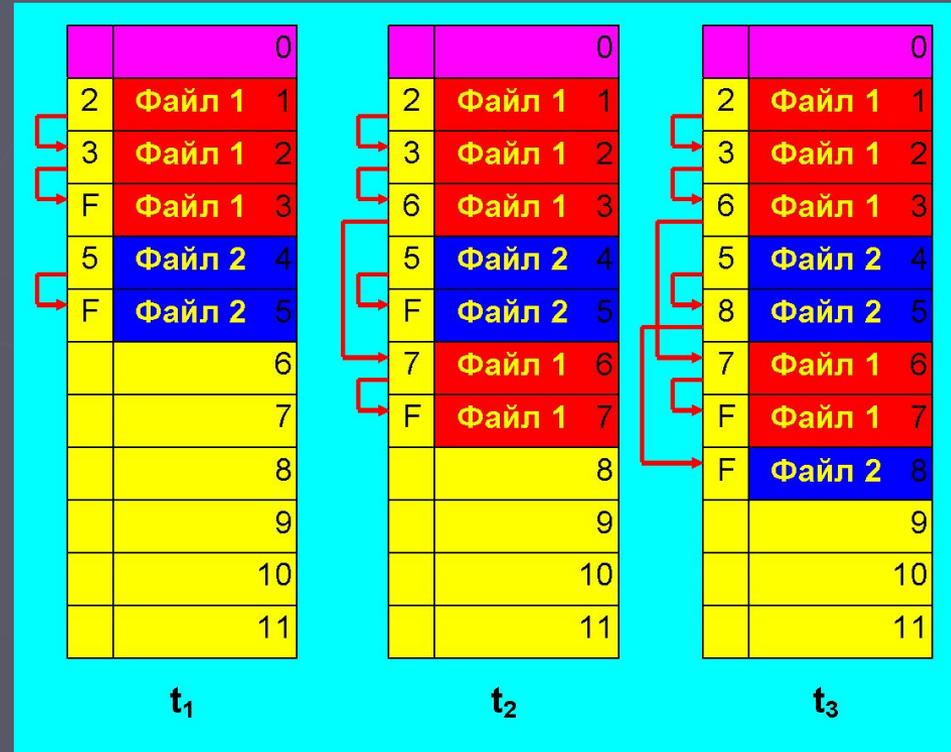


Имя файла	Номер блока	Кол-во блоков

# Связанный список блоков

В начале каждого блока содержится указатель на следующий блок. В этом случае адрес файла также может быть задан одним числом - номером первого блока.

В отличие от предыдущего способа, каждый блок может быть присоединен в цепочку какого-либо файла, следовательно, фрагментация отсутствует. Файл может изменяться во время своего существования, наращивая число блоков.



Имя файла	Номер блока

# Связанный список индексов

С каждым блоком связывается некоторый элемент - индекс. Индексы располагаются в отдельной области диска (в MS-DOS это таблица FAT).

Если некоторый блок распределен некоторому файлу, то индекс этого блока содержит номер следующего блока данного файла.

Номер блока	32	33	34	35	36	37	38	39	40	41
Индекс	0	34	36	37	FF	41	40	0	FF	FF

Индексная обл.	
	32
Файл 1	33
Файл 1	34
Файл 2	35
Файл 1	36
Файл 2	37
Файл 3	38
	39
Файл 3	40
Файл 2	41

Область данных

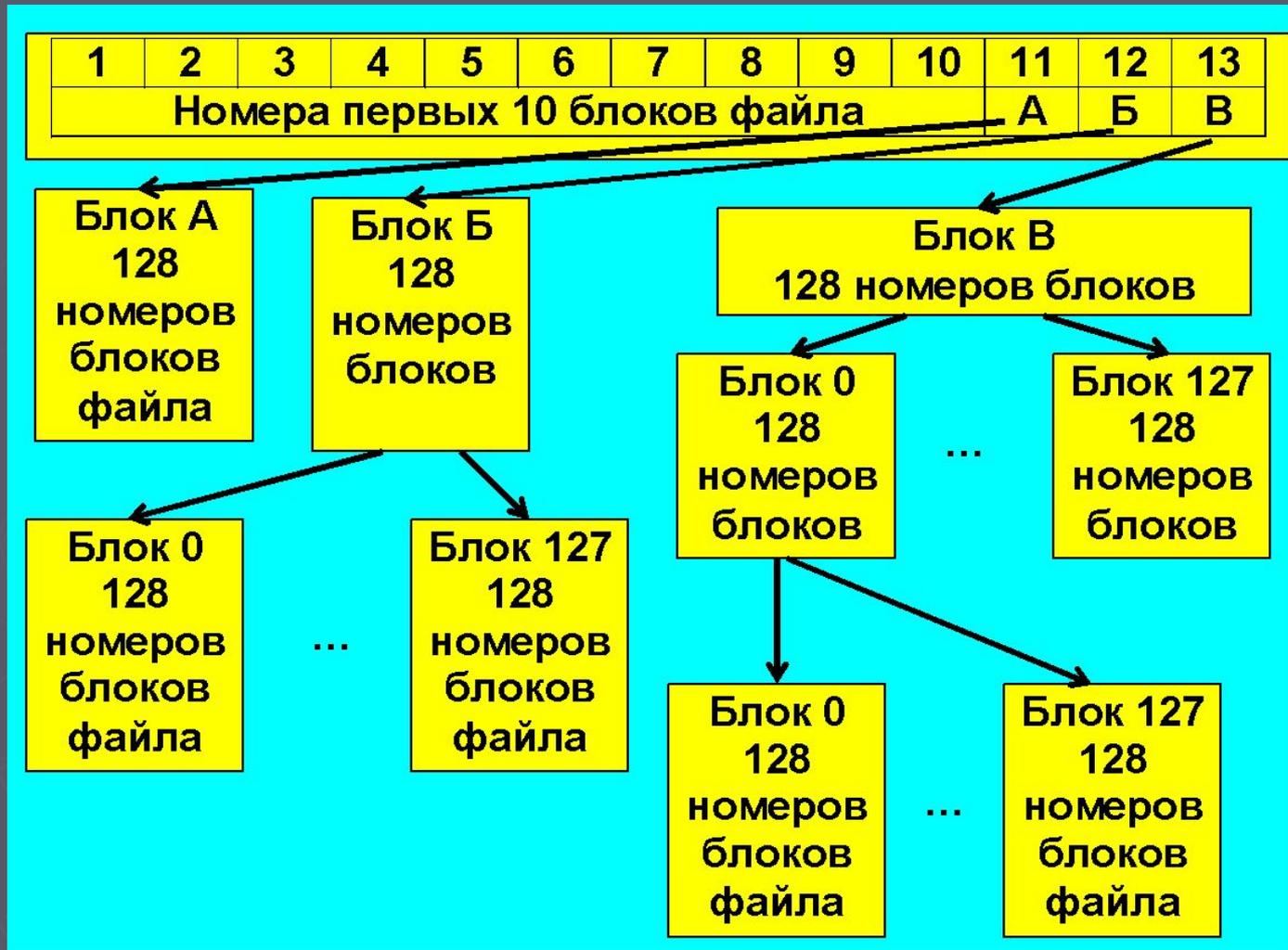
Имя файла	Индекс первого блока

# Перечень номеров блоков

Имя файла	Адреса блоков				
	1	2	3	...	13
Файл 1	33	34	36		
Файл 2	35	37	41		
Файл 3	38	40			
Файл 4	42	43			

	32
Файл 1	33
Файл 1	34
Файл 2	35
Файл 1	36
Файл 2	37
Файл 3	38
	39
Файл 3	40
Файл 2	41
Файл 4	42
Файл 4	43

# Перечень номеров блоков (UFS)



# Файловая система NTFS (New Technology File System)

# Структура диска

Файлы в NTFS записываются в виде последовательности блоков (кластеров). Кластеры в файловой системе NTFS могут принимать размеры от 512 байт до 64 Кб. Стандартным значением длины кластера является размер 4 Кб.

Главной структурой NTFS является файл MFT (Master File Table – главная файловая таблица).

Файл MFT содержит записи о каждом файле системы. Размер записи файла составляет 1 Кб. Если при описании файла одной записи не хватает, то используются другие записи. Первые 16 файлов, записи о которых располагаются в начале MFT, являются системными.



# Структура записи MFT для файла

Каждая запись состоит из заголовка, за которым следует заголовок атрибута и его значение.

Каждый заголовок содержит: контрольную сумму, порядковый номер файла, увеличивающийся, когда запись используется для другого файла, счетчик обращений к файлу, количество байт действительно используемых в записи и другие поля.

За заголовком записи располагается заголовок первого атрибута, а далее значение этого атрибута. Затем идет заголовок второго атрибута и т.д. Если атрибут достаточно велик, то он помещается в отдельном файле (нерезидентный атрибут).



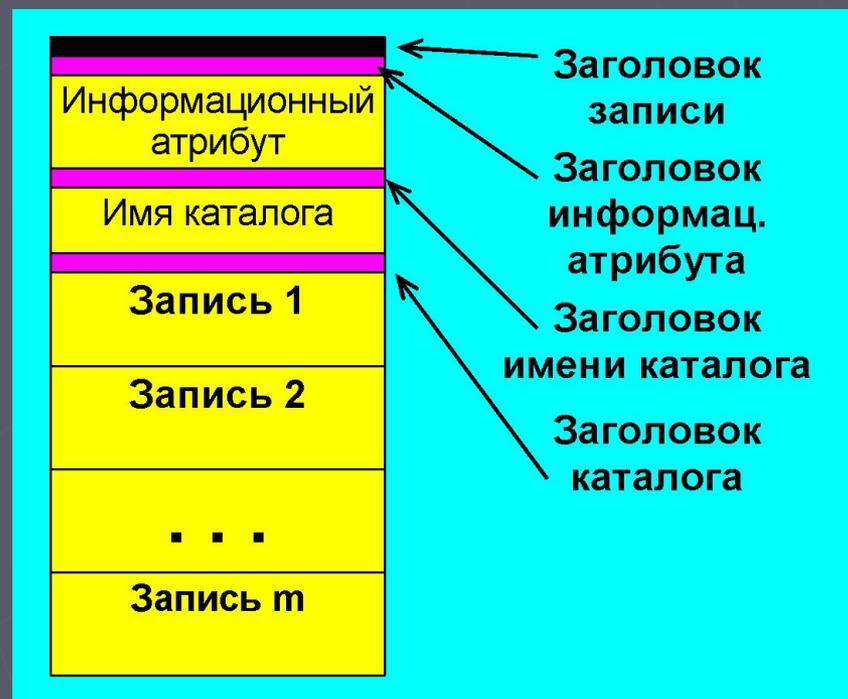
Атрибут	Описание
Стандартная информация (информационный атрибут)	Сведения о владельце, информация о защите, счетчик жестких связей, битовые атрибуты (только для чтения, архивный и т.д.).
Имя файла	Имя файла в кодировке Unicode.
Описатель защиты	Теперь этот атрибут устарел. Теперь используется атрибут \$EXTENDEDSECURE.
Список атрибутов	Расположение дополнительных записей MFT. Используется, если атрибуты не помещаются в записи.
Идентификатор объекта	64-разрядный идентификатор файла, уникальный для данного тома.
Точка повторной обработки	Используется для создания иерархических хранилищ. Наличие этого атрибута предлагает процедуру, анализирующей имя файла выполнить дополнительные действия.
Название тома	Используется в \$VOLUME.
Информация о томе	Версия тома (используется в \$VOLUME).
Корневой индекс	Используется для каталогов.
Размещение индекса	Для очень больших каталогов, которые реализуются не в виде обычных списков, а в виде бинарных деревьев (B-дерев).
Битовый массив	Используется для очень больших каталогов.
Поток данных утилиты регистрации	Управляет регистрацией в файле \$LOGFILE.
Данные	Поток данных файла. Следом за заголовком этого атрибута следует список кластеров, где располагаются данные, либо сами данные, если их объем не превышает несколько сот байтов.

# Каталоги в NTFS

Каталог в системе NTFS представляет собой специфический файл, хранящий ссылки на другие файлы и каталоги, создавая иерархическое строение файловой системы диска.

Как и в случае с обычным файлом, если каталог не слишком велик, то он помещается в записи MFT. Сами записи каталога содержат длину имени файла, некоторые другие его параметры, а самое главное содержат номер (индекс) записи MFT для данного файла, в которой содержится уже полная информация о файле.

Для больших каталогов используется совсем другой формат хранения. Они строятся в виде бинарных деревьев, что обеспечивает быстрый поиск в алфавитном порядке и ускоряет добавление нового файла.



# Права доступа к файлу

- ▶ создание файла
- ▶ уничтожение файла,
- ▶ открытие файла
- ▶ закрытие файла
- ▶ чтение файла
- ▶ запись в файл
- ▶ дополнение файла
- ▶ поиск в файле
- ▶ получение атрибутов файла
- ▶ установление новых значений атрибутов
- ▶ переименование
- ▶ выполнение файла
- ▶ чтение каталога

# Избирательный доступ

Пользователи	Файлы			
	alpha.txt	beta.doc	sigma.exe	omega.xls
Victor	read	read write	-	read write
Kseniy	read append	-	Run	-
Ivan	read write	read write	Run	read write

# Мандатный доступ

При мандатном доступе пользователи разделены на группы и операционная система наделяет пользователя определенными полномочиями по отношению к файлам в зависимости от того, к какой группе он относится.

# Общая модель файловой системы

Функционирование любой файловой системы можно представить многоуровневой моделью, в которой каждый уровень предоставляет некоторый интерфейс (набор функций) вышележащему уровню, а сам, в свою очередь, для выполнения своей работы использует интерфейс (обращается с набором запросов) нижележащего уровня.

# Символьный уровень

Задачей символьного уровня является определение по символьному имени файла его уникального имени.

В файловых системах, в которых каждый файл может иметь только одно символьное имя (например, MS-DOS), этот уровень отсутствует, так как символьное имя, присвоенное файлу пользователем, является одновременно уникальным и может быть использовано операционной системой.

В других файловых системах, в которых один и тот же файл может иметь несколько символьных имен, на данном уровне просматривается цепочка каталогов для определения уникального имени файла.



# Базовый уровень

На базовом уровне по уникальному имени файла определяются его характеристики: права доступа, адрес, размер и другие.

Характеристики файла могут входить в состав каталога или храниться в отдельных таблицах. При открытии файла его характеристики перемещаются с диска в оперативную память, чтобы уменьшить среднее время доступа к файлу.

В некоторых файловых системах (например, HPFS) при открытии файла вместе с его характеристиками в оперативную память перемещаются несколько первых блоков файла, содержащих данные.



# Проверка прав доступа

При проверке прав доступа сравниваются полномочия пользователя или процесса, выдавших запрос, со списком разрешенных видов доступа к данному файлу.

Если запрашиваемый вид доступа разрешен, то выполнение запроса продолжается, если нет, то выдается сообщение о нарушении прав доступа.



# Логический уровень

На логическом уровне определяются координаты запрашиваемой логической записи в файле, то есть требуется определить, на каком расстоянии (в байтах) от начала файла находится требуемая логическая запись.

Например, если файл организован как последовательность логических записей фиксированной длины  $l$ , то  $n$ -ая логическая запись имеет смещение  $l((n-1))$  байт.

Для определения координат логической записи в файле с индексно-последовательной организацией выполняется чтение таблицы индексов (ключей), в которой непосредственно указывается адрес логической записи.



# Физический уровень

На физическом уровне файловая система определяет номер физического блока, который содержит требуемую логическую запись, и смещение логической записи в физическом блоке. Для решения этой задачи используются результаты работы логического уровня - смещение логической записи в файле, адрес файла на внешнем устройстве, а также сведения о физической организации файла, включая размер блока.



## Исходные данные:

$V$  - размер блока

$N$  - номер первого блока файла

$S$  - смещение логической записи в файле

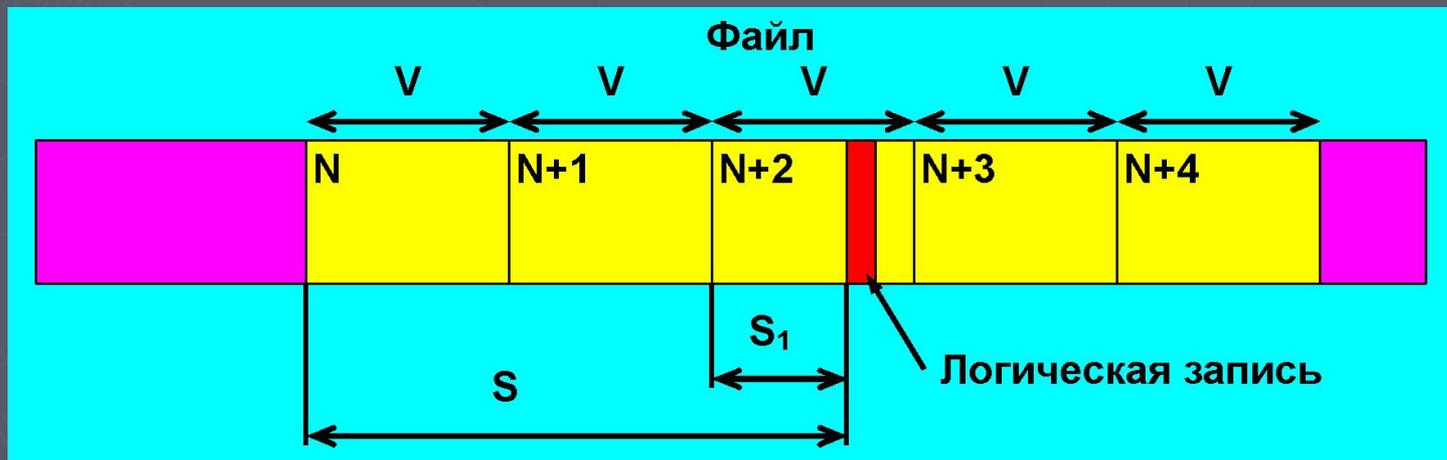
## Требуется определить на физическом уровне:

$n$  - номер блока, содержащего требуемую логическую запись

$s$  - смещение логической записи в пределах блока

$n = N + [S/V]$ , где  $[S/V]$  - целая часть числа  $S/V$

$s = R [S/V]$  - дробная часть числа  $S/V$



Подчеркнем, что задача физического уровня решается независимо от того, как был логически организован файл.

После определения номера физического блока, файловая система обращается к системе ввода-вывода для выполнения операции обмена с внешним устройством. В ответ на этот запрос в буфер файловой системы будет передан нужный блок, в котором на основании полученного при работе физического уровня смещения выбирается требуемая логическая запись.

Отображаемые в  
память файлы

Некоторые ОС, начиная с MULTICS, обеспечивают отображение файлов в адресное пространство выполняемого процесса. Отмеченное можно рассматривать как появление дополнительного сегмента процесса.



В системе появляются два новых системных вызова: MAP (отобразить) и UNMAP (отменить отображение). Первый вызов передает операционной системе в качестве параметров имя файла и виртуальный адрес, и операционная система отображает указанный файл в виртуальное адресное пространство по указанному адресу.

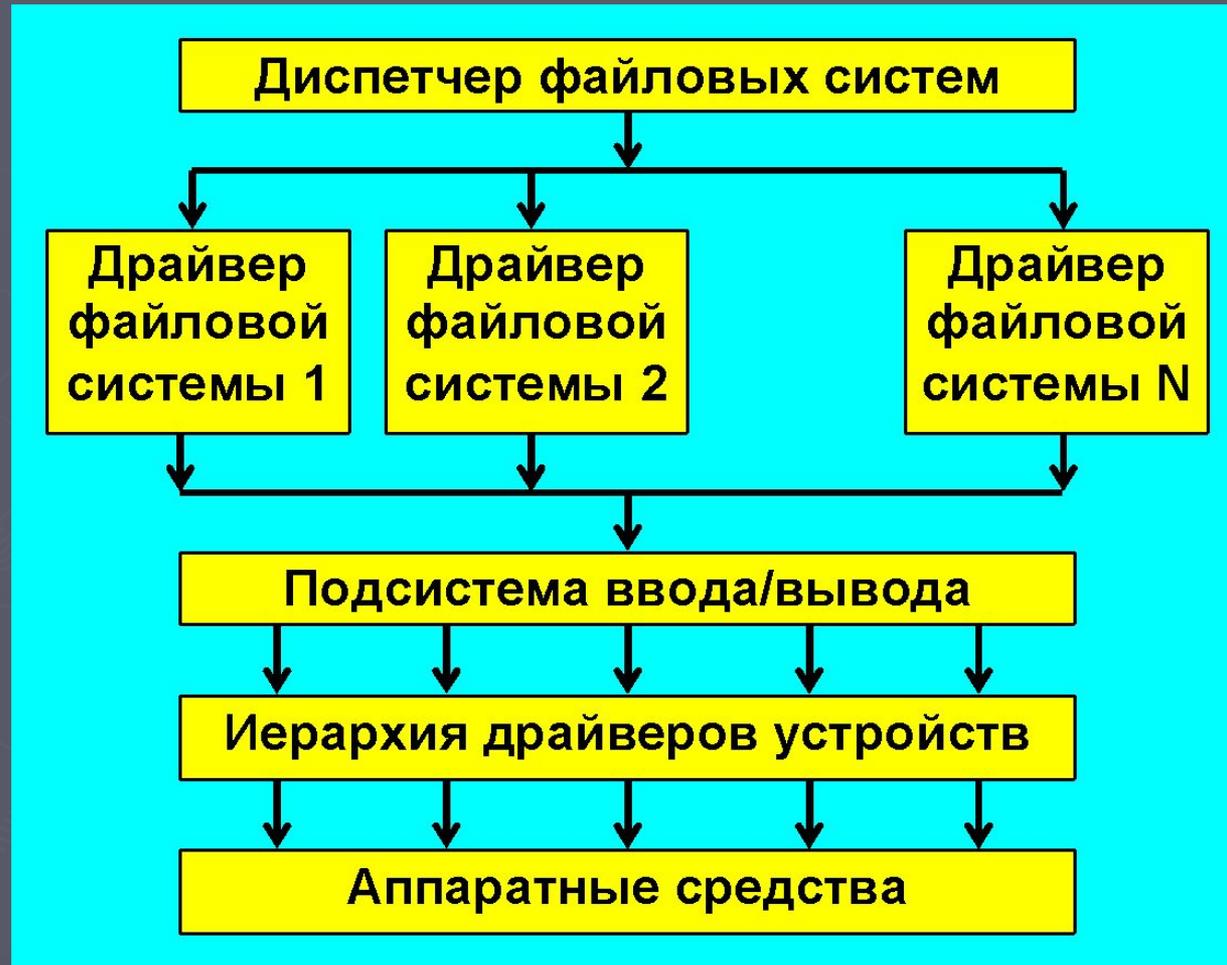


# Проблемы

1. Системе сложно определить размеры выходного файла.
2. Поддержание согласованности данных файла для двух и более процессов. Допустим, один процесс отображает файл для записи, а другой процесс открывает его для обычного файлового доступа. Если первый процесс изменяет страницу, то это изменение не будет отражено в файле на диске до тех пор, пока страница не будет вытеснена на диск.
4. Файл может быть больше, чем сегмент, и даже больше, чем все виртуальное адресное пространство.

# Современные архитектуры файловых систем

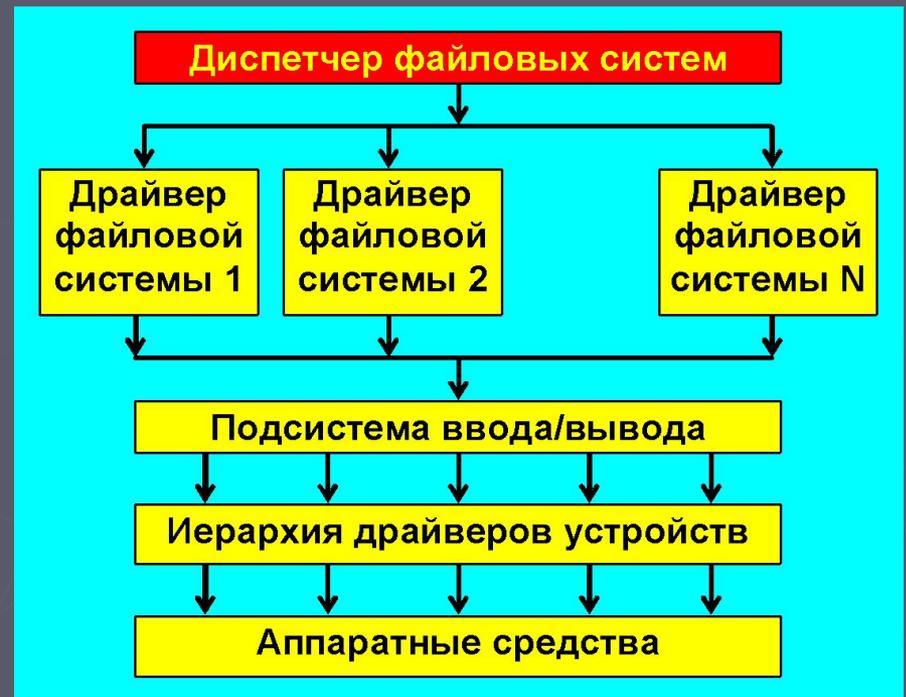
# Архитектура современной файловой системы



**Диспетчер файловых систем** является обеспечивает интерфейс между запросами приложения и конкретной файловой системой, к которой обращается это приложение.

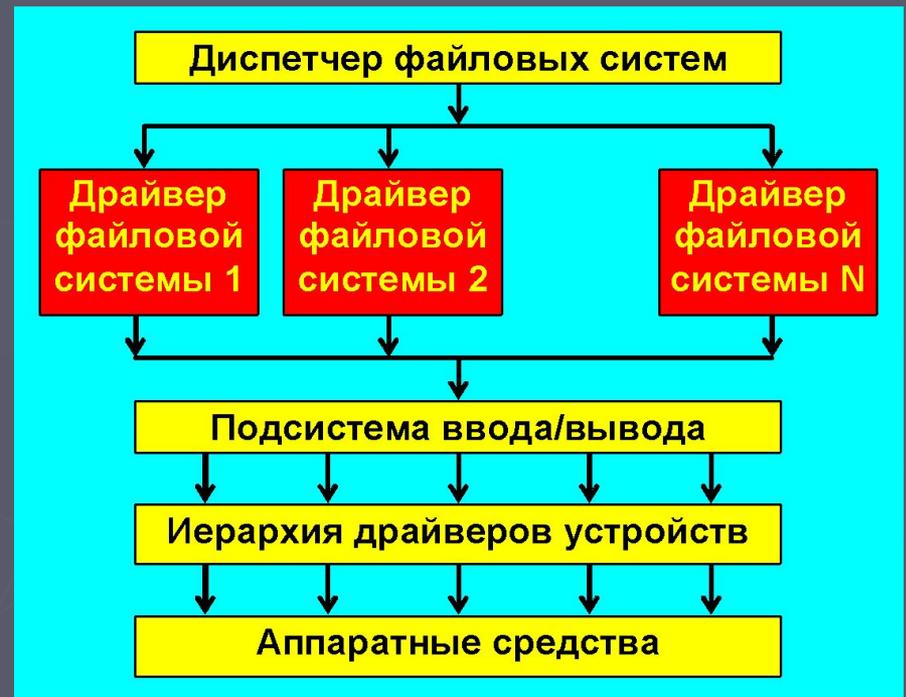
Он преобразует запросы в формат, воспринимаемый следующим уровнем - уровнем файловых систем и является единственным модулем, который может обращаться к драйверу файловой системы.

Приложение не может обращаться к драйверам ФС напрямую.



**Драйвер файловой системы** может быть написан в виде реентерабельного кода, что позволяет сразу нескольким приложениям выполнять операции с файлами.

Каждый драйвер файловой системы в процессе собственной инициализации регистрируется у диспетчера, передавая ему таблицу точек входа, которые будут использоваться при последующих обращениях к файловой системе.

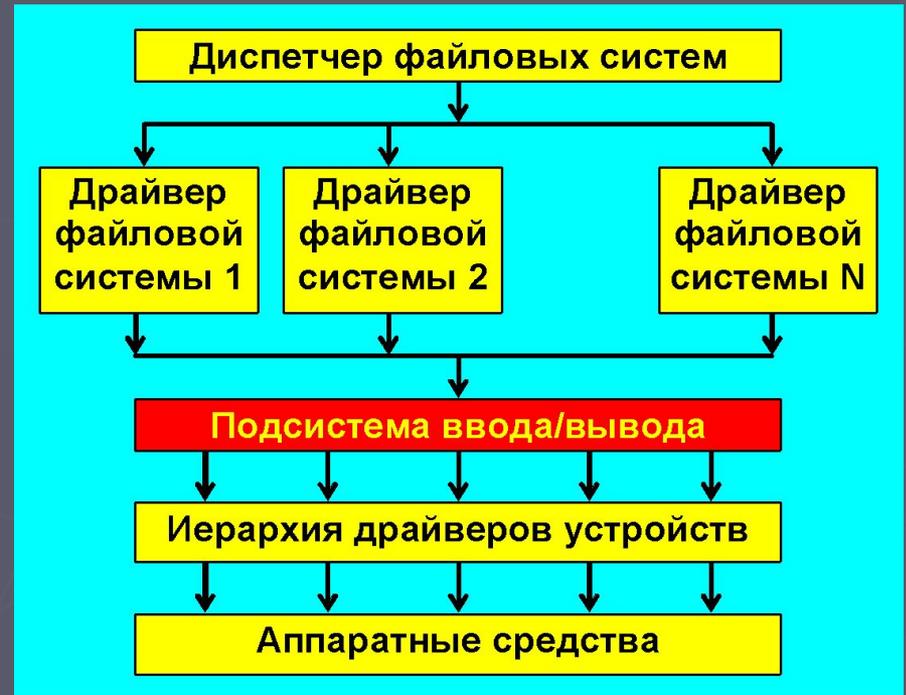


## Подсистема ввода вывода

отвечает за загрузку, инициализацию и управление всеми модулями низших уровней файловой системы.

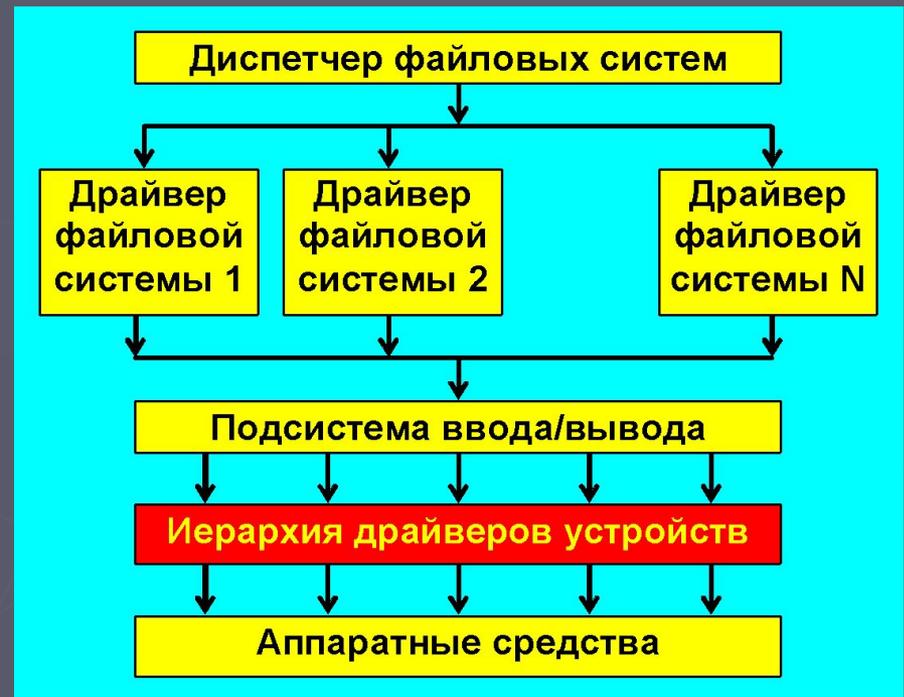
Эти модули представляют собой драйверы портов, которые непосредственно занимаются работой с аппаратными средствами.

Кроме этого подсистема ввода-вывода обеспечивает некоторый сервис драйверам файловой системы, что позволяет им осуществлять запросы к конкретным устройствам, должна постоянно присутствовать в памяти.



## Драйверы устройств

определенного типа (драйверы жестких дисков или накопителей на лентах), драйверы, поддерживаемые поставщиками (такие драйверы перехватывают запросы к блочным устройствам и могут частично изменить поведение существующего драйвера этого устройства, например, зашифровать данные), драйверы портов, которые управляют конкретными адаптерами.



# Файловые менеджеры

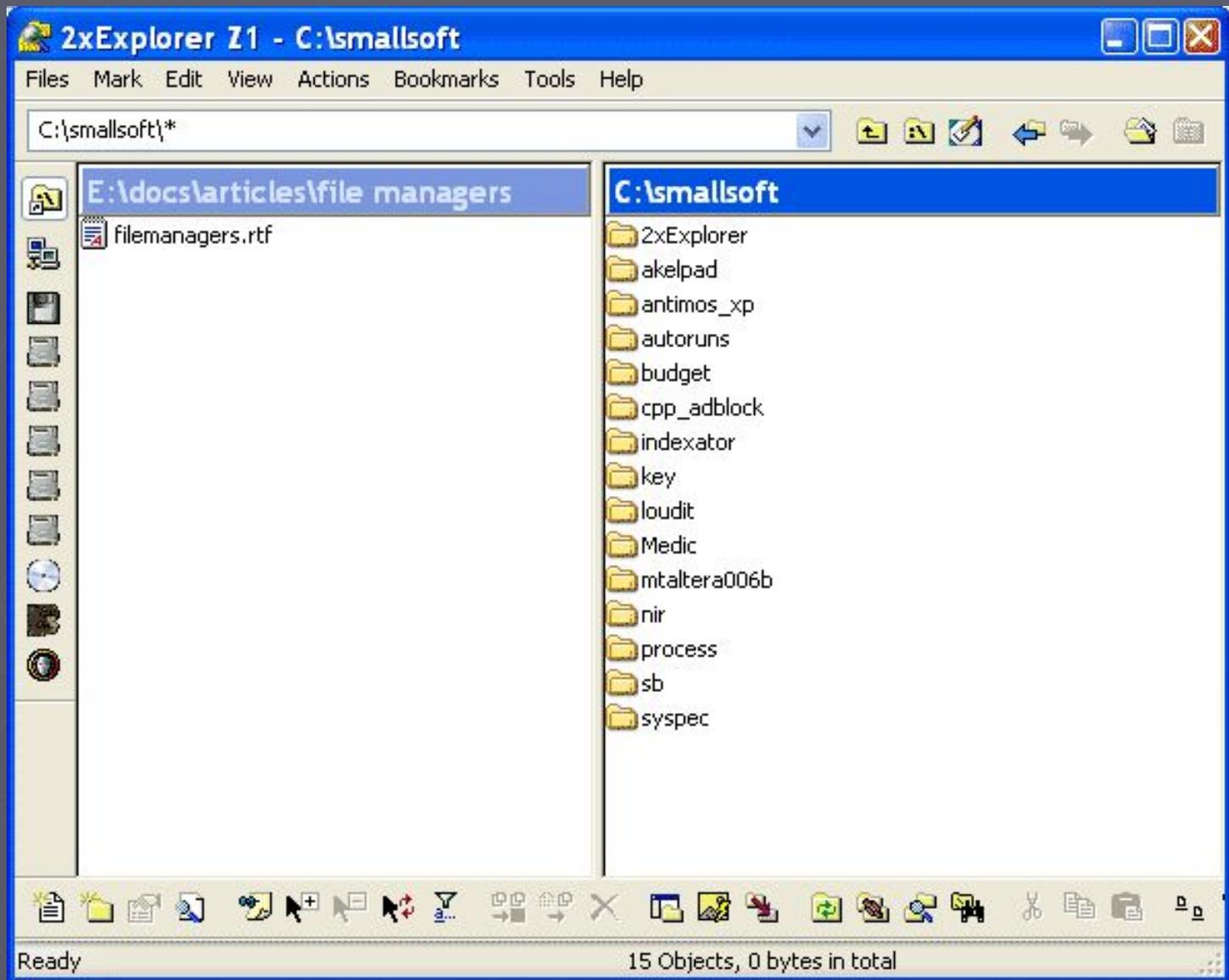
# 2xExplorer 1.4.1.12

2xExplorer во многом повторяет концепцию стандартного инструмента Windows (проводника), при этом имея массу дополнительных полезных функций.

2xExplorer, имея поддержку множества горячих клавиш, позволяет осуществлять управление только с клавиатуры. Для часто используемых папок можно назначать собственные клавиатурные сочетания. Разработчики уделили пристальное внимание просмотру и изменению файлов, включив в дистрибутив неплохой редактор.

Небольшой размер дистрибутива и бесплатный статус программы. Оба эти фактора делают 2xExplorer максимально доступным.

Разработка 2xExplorer прекращена несколько лет назад, перспектив улучшения продукта нет.

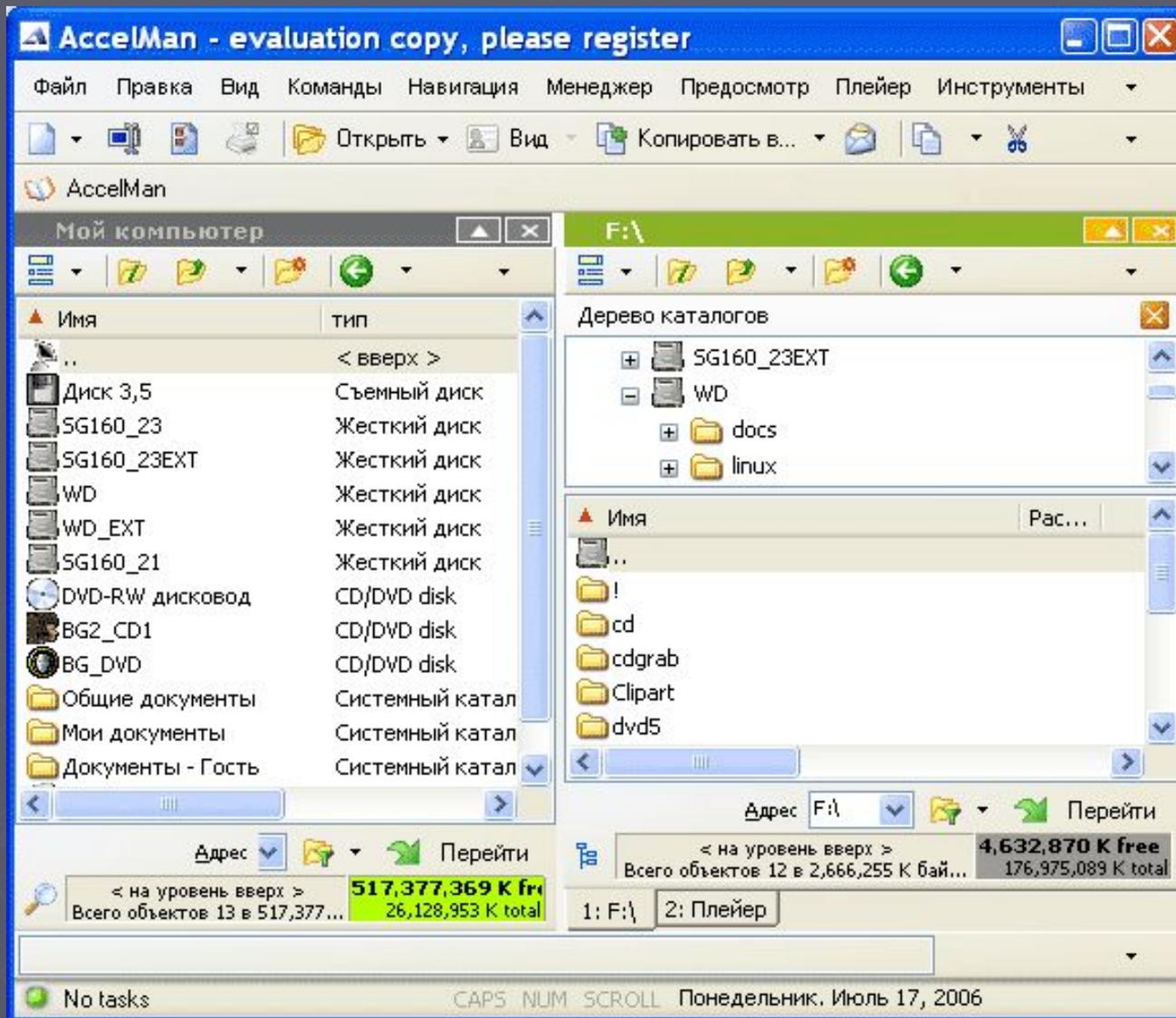


# AccelMan 3.0.0.3250

AccelMan объединяет в себе функции файлового менеджера и программы для просмотра множества различных типов документов.

Файловый менеджер содержит полноценный медиа-проигрыватель, текстовый редактор с возможностью подсветки синтаксиса, а также с поддержкой документов, использующих сложное форматирование (RTF, DOC).

AccelMan понимает несколько типов архивов, в число которых входит TAR, GZIP, ARJ и RAR, а также имеет встроенный ZIP-архиватор. В файловый менеджер встроен инструмент конвертирования графических документов с поддержкой форматов JPEG, TIFF, GIF, PNG и BMP.

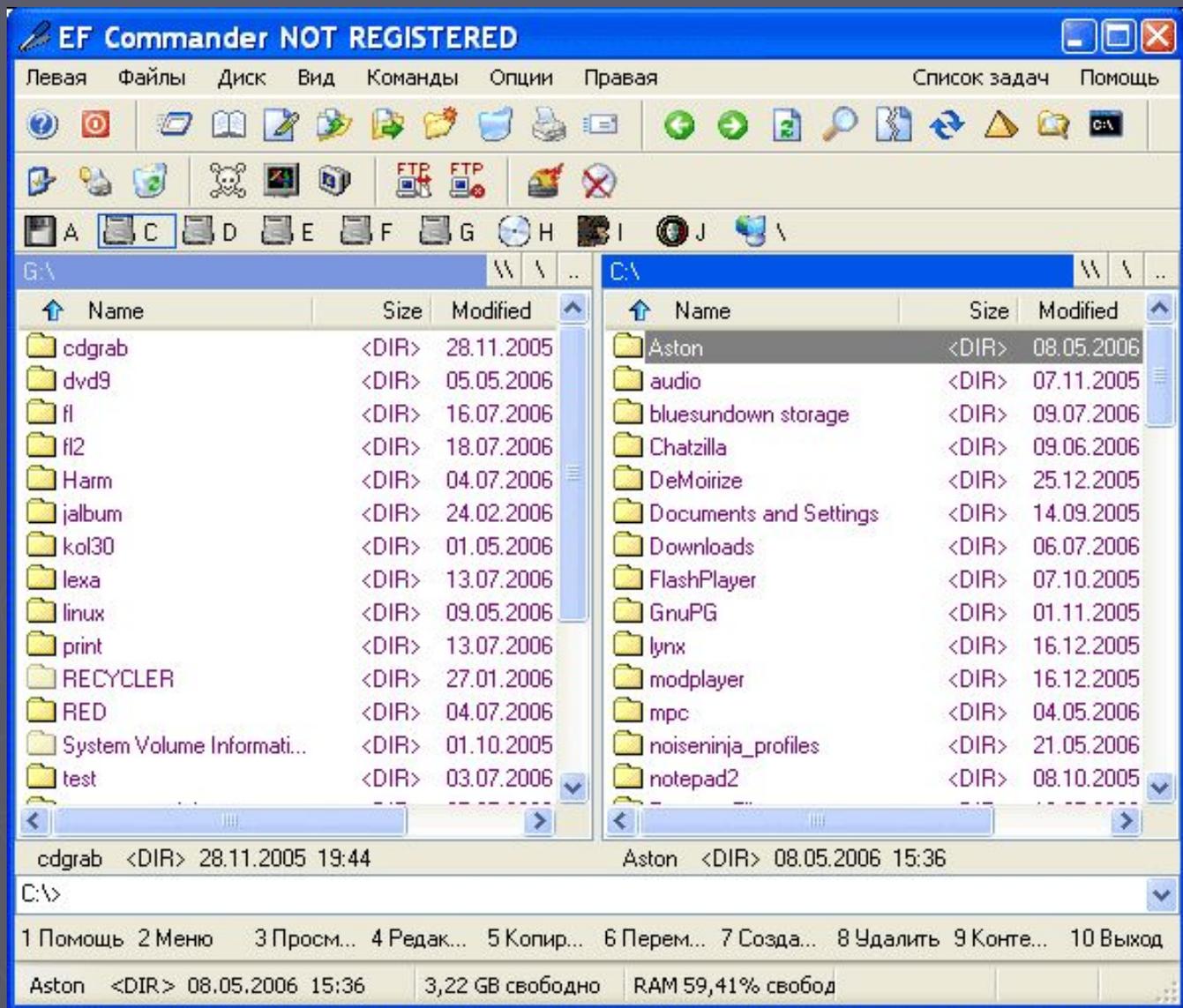


# EF Commander 5.50

EF Commander - один из старейших файловых менеджеров на сегодняшнем рынке.

Изначально он был написан в 1994 году для OS/2 и лишь два года спустя адаптирован в среде Win32.

EF Commander обладает мощными средствами просмотра документов, вплоть до HEX-редактора (данные представлены в виде матрицы двухзначных шестнадцатеричных чисел). С помощью плагинов можно осуществлять поддержку дополнительных типов документов. Файловый менеджер понимает более 20 типов архивов.

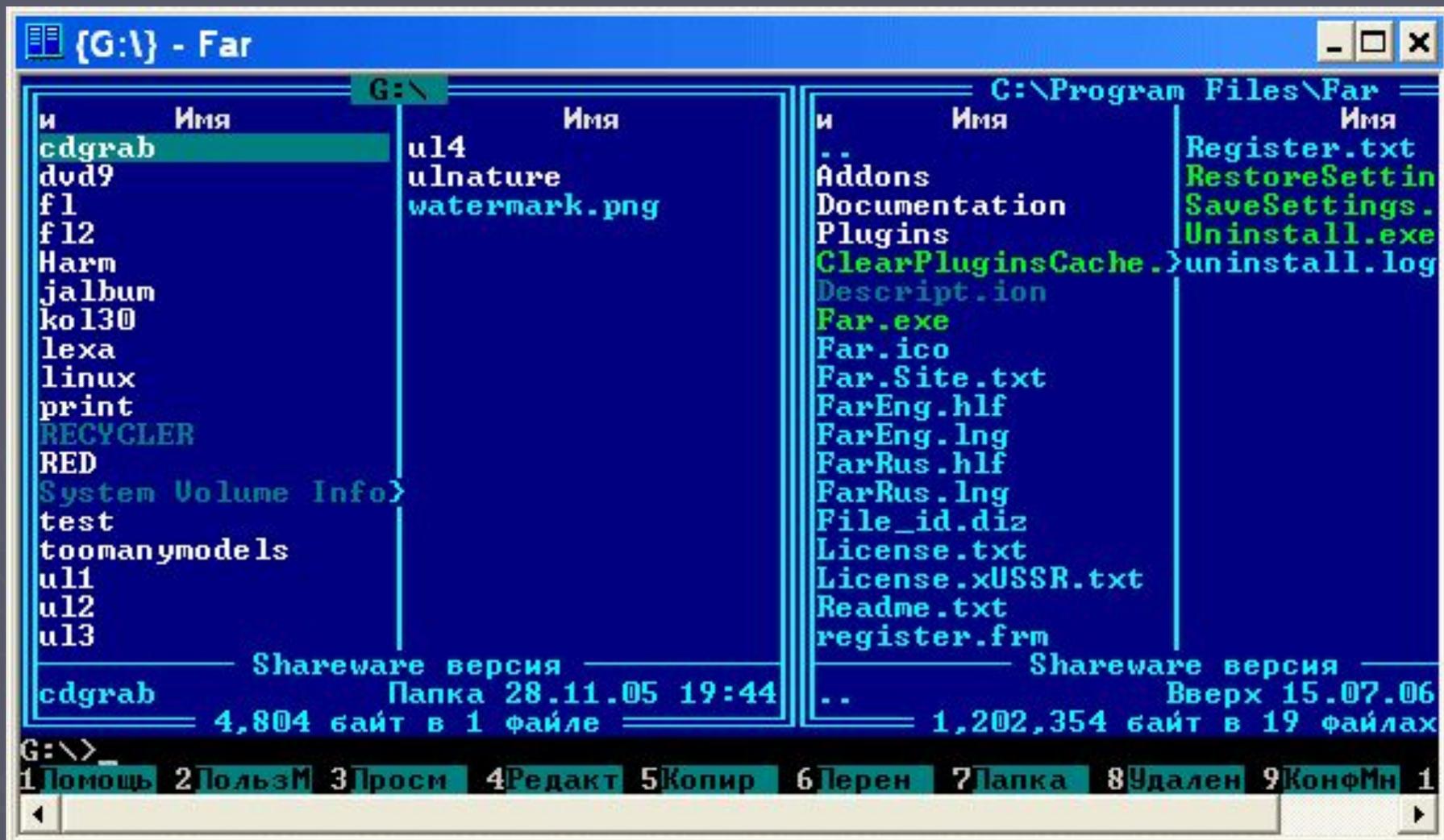


# FAR 1.7

Александр Рошал, автор архиватора RAR, хотел в то время написать его консольную 32-битную версию. Видимо, дело пошло даже лучше, чем ожидалось, и вместо простой оболочки в стиле RAR для DOS, мы увидели настоящий файловый менеджер, несущий старые, проверенные временем традиции управления файлами.

FAR позволяет работать не только с локальными ресурсами, но и с сетевыми дисками, а также имеет FTP-клиент. В дистрибутив FAR включено несколько дополнительных модулей, позволяющих организовывать временную панель, список системных процессов, редактор реестра, а также инструмент, позволяющий изменять регистры букв в названиях групп файлов.

FAR бесплатен для пользователей, проживающих на территории USSR.

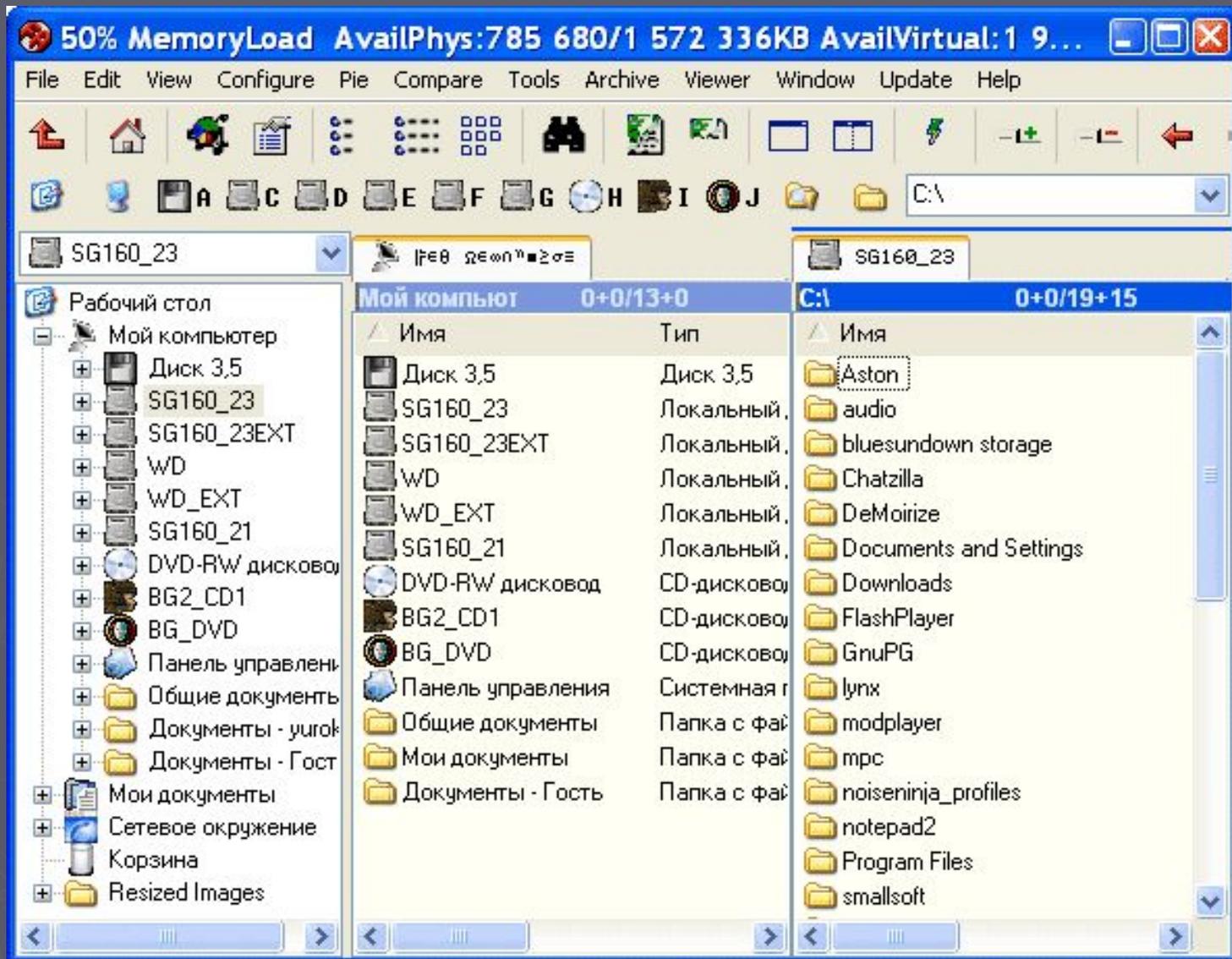


# File Ant 20050830

FileAnt - весьма необычный файловый менеджер для Windows. Во-первых, управление программой с клавиатуры резко отличается от принятых стандартов (F5 - копировать, F7 - создать папку и т.д.). Во-вторых, программа сочетает в себе как дерево каталогов (как Проводник), так и две традиционные панели (Norton Commander).

Просмотр документов осуществляется внутри специальной панели, расположенной над списком файлов и каталогов.

Файловый менеджер содержит множество мелких необычных функций, например, выделение файлов, имеющих определенный процент от общего объема каталога, сравнение файлов с учетом CRC32, поддержку собственных хранителей экрана и другое.



# Frigate 3.33

Важнейшим достоинством Frigate является сохранение простоты освоения при всем богатстве возможностей.

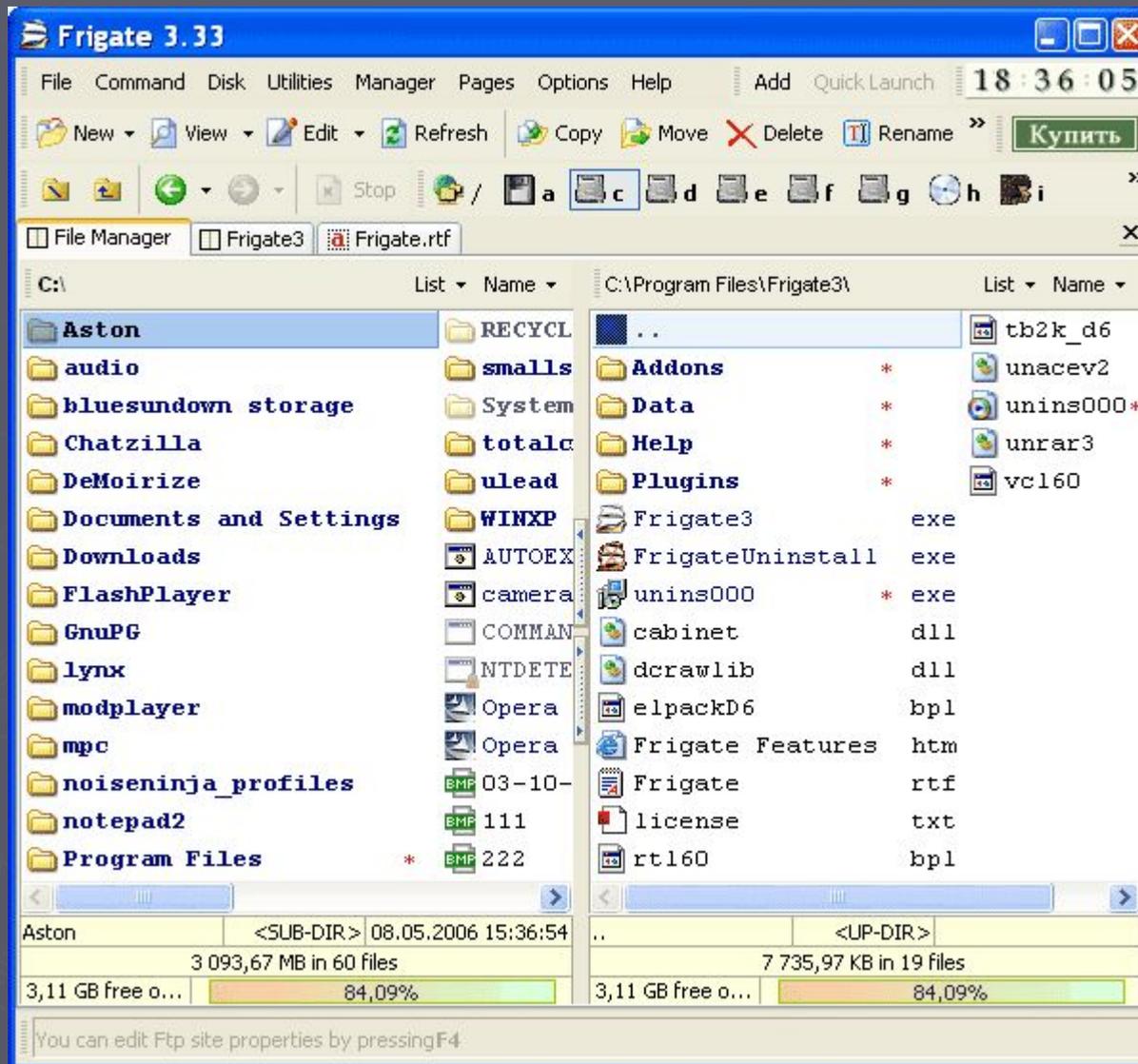
Frigate 3.33 позволяет просматривать огромное количество типов документов. Помимо традиционных архивов, текстовых документов, аудио и видео информации, можно открывать базы данных (DBF) и электронные таблицы (XLS).

Встроена утилита контроля за автоматически запускающимися программами Windows, браузер для просмотра Web-страниц, а также собственная консоль.

Frigate распространяется в трех вариантах.

Облегченный вариант (Lite) можно установить бесплатно и пользоваться без ограничений по времени.

Остальные варианты имеют ознакомительный период 30 дней.



# Total Commander 6.53

Кристиан Гислер, будучи студентом Университета Берна, в 1993 году написал файловый менеджер. Ему не нравились другие программы.

В начале 90-х годов большинство программ подобного класса были рассчитаны на работу в MS DOS. Windows Commander (старое название продукта) фактически стал первой ласточкой, первой программой, использовавшей все преимущества GUI.

Total Commander нельзя назвать самой функциональной, самой красивой или самой быстрой программой данного класса. Причина популярности файлового менеджера кроется в сбалансированности всех составляющих.

Лицензия Total Commander позволяет бесплатно использовать продукт неограниченное количество времени. В этом случае при старте программы отображается окно с предложением купить программу. Вы закрываете его и после этого пользуетесь полнофункциональной версией файлового менеджера. Заплатив за программу, вы сможете работать с Total Commander без назойливого стартового окна.

