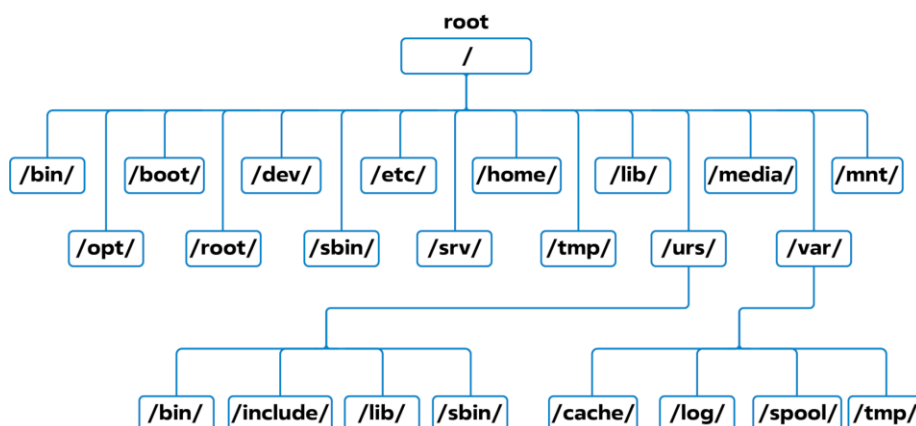


1. Логическая структура файловой системы

В операционной системе Astra Linux используется единое файловое пространство. Под этим понимается логическая структура файловой системы в виде одного перевернутого дерева с корнем (root) вверху.



Корень дерева (root) обозначается как /. Ветки дерева могут находиться на разных устройствах: как на локальных и съёмных устройствах, так и на сетевых. Процедура построения единого дерева из веток, расположенных на разных устройствах, называется монтированием.

Путь до любого файла представляет собой цепочку каталогов, которые разделяются символом /. В системе могут использоваться абсолютные и относительные пути:

- абсолютный путь начинается с / (например: */home/user/Документы/Первое полугодие/Отчёт.odt*);
- относительный путь начинается с текущего каталога (имя текущего каталога – ., родительского – .., например: *./Первое полугодие/Отчёт.odt* или *../Изображения/Процессы/Рабочая схема.odg*).

Также можно использовать путь относительно домашней папки текущего пользователя, т. е. указать путь в файловой системе: не от корня, а от папки текущего пользователя. Чтобы задать путь подобным образом, он должен начинаться с «~/». Например: *~/Документы/Приказ.odt*.

Имя файла может содержать любые символы, кроме / и \0. Не рекомендуется использовать в именах файлов специальные управляющие символы и специальные символы командного интерпретатора. Например, рекомендуется пробел заменять нижним подчеркиванием. В имени файла можно использовать специальные невидимые ASCII символы, но на практике это делать настоятельно не рекомендуется.

ПРИМЕЧАНИЕ

Символ \0 (null) — символ конца строки (символ с кодом 0 в таблице кодировки символов ASCII, *map ascii*).

Имена файлов могут иметь префиксы и суффиксы, которые «подсказывают» назначение файла. Рекомендуется придерживаться принятых соглашений в отношении использования общепринятых суффиксов и префиксов.

Примеры:

- суффикс `.tar` имеют архивы, созданные утилитой `tar`.
- суффикс `.gz` имеют файлы, сжатые утилитой `gzip`.

Для файла `libm-2.24.so` — префикс `lib` (библиотека функций), суффиксы: `2.24` — версия и `so` — `shared object` (разделяемая библиотека).

Неиспользование общепринятых суффиксов для файлов может приводить к проблемам при работе с утилитами, предназначенными для обработки таких файлов (например, утилиты сжатия предполагают, что сжатые файлы имеют вполне определенные суффиксы).

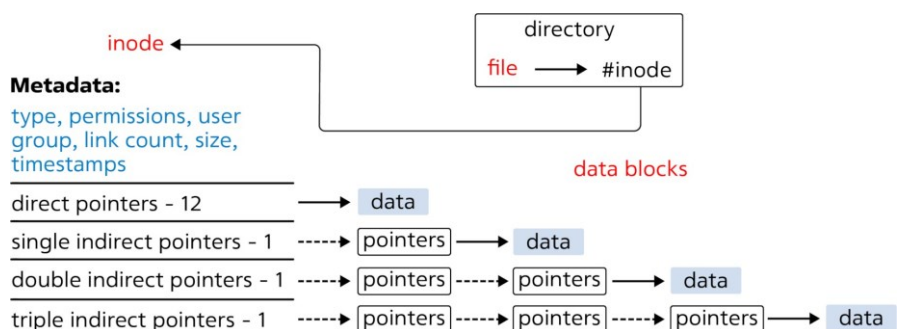
Максимальная длина имени файла ограничена значением в 255 символов, а длина пути не должна превышать 4096 символов.

2. Структура файла

Каждый файл состоит из следующих компонентов:

- имя файла;
- индексный дескриптор (`inode`);
- блоки данных (0 или более).

В `inode` хранятся атрибуты (метаданные) файла и ссылки на блоки данных.



ПРИМЕЧАНИЕ

Для просмотра содержимого `inode` следует воспользоваться командой `stat имя_файла`.

Имя файла и соответствующий ему номер индексного дескриптора хранятся в каталоге.

ПРИМЕЧАНИЕ

- ☐ С помощью команды `ls -li имя_файла` можно просмотреть номер индексного дескриптора и имя файла.
- ☐ С помощью команды `df -li` можно просмотреть количество занятых и свободных индексных дескрипторов в файловой системе.

3. Типы файлов

В системе используются файлы следующих типов:

- Регулярные файлы — обычные файлы, которые могут содержать текст, исполняемые инструкции для программ, изображения или другую информацию.
- Каталог — сопоставляет имя файла с номером inode.
- Символическая ссылка — в области данных содержится имя файла на который указывает ссылка.
- Файл устройств (блочных и символьных) — в inode которого хранятся номер драйвера (major number) и номер устройства (minor number).
- Локальные (UNIX) сокеты (socket) и именованные каналы (pipe).

ПРИМЕЧАНИЕ

Права доступа, установленные на символическую ссылку, не имеют значения. Права доступа определяются правами файла, на который указывает ссылка.

Символьные устройства — вид файла устройства, обеспечивающий интерфейс доступа к устройству, реальному или виртуальному, с возможностью посимвольного обмена информацией.

Блочные устройства — вид файла устройств, обеспечивающий интерфейс доступа к устройству, реальному или виртуальному, в виде файла в файловой системе. С блочным устройством обеспечивается обмен данными блоками данных.

Также в ОС используется так называемая жесткая ссылка. Она не является специальным файлом, это ещё одно имя файла. В inode хранится количество жестких ссылок (link count).

4. Стандарт иерархии ФС (FHS)

Распределение данных происходит, исходя из статичности данных (static, variable) и возможности использования удаленных хостов (shareable, unshareable).

Каталоги первого уровня логической структуры файловой системы ОС и их назначение приведены в таблице ниже.

Каталог	Назначение
/	Корневой каталог.
/bin	Основные утилиты пользователя (ссылка на /usr/bin).
/boot	Загрузочные файлы.
/dev	Файлы устройств.
/etc	Общесистемные конфигурационные файлы.
/home	Домашние каталоги пользователей.
/lib	Основные библиотеки, программное обеспечение статическая часть (ссылка на /usr/lib).
/media	Точки монтирования для сменных носителей.
/mnt	Для временного монтирования файловых системы.
/opt	Дополнительное программное обеспечение.
/proc	Псевдо ФС — информация об ядре, процессах.
/root	Домашний каталог пользователя root.
/run	Данные, необходимые для работы демонов.
/sbin	Основные команды для администрирования (ссылка на /usr/sbin).
/srv	Данные для служб.
/sys	Псевдо ФС.
/tmp	Временные файлы.

В Debian принята структура файловой системы, называемая «merged-/usr», которая не включает каталоги /bin, /sbin, /lib или необязательные варианты, такие как /lib64. В новом макете устаревшие каталоги заменены символическими ссылками на соответствующие местоположения /usr/bin, /usr/sbin, /usr/lib и /usr/lib64.

В каталоге /usr хранятся данные программ пользователей (подробнее приведено в таблице ниже).


Каталог	Назначение
---------	------------

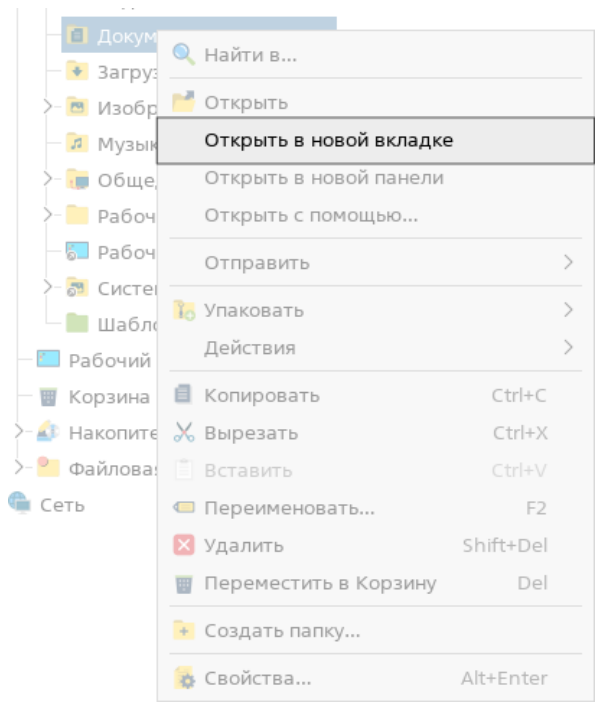
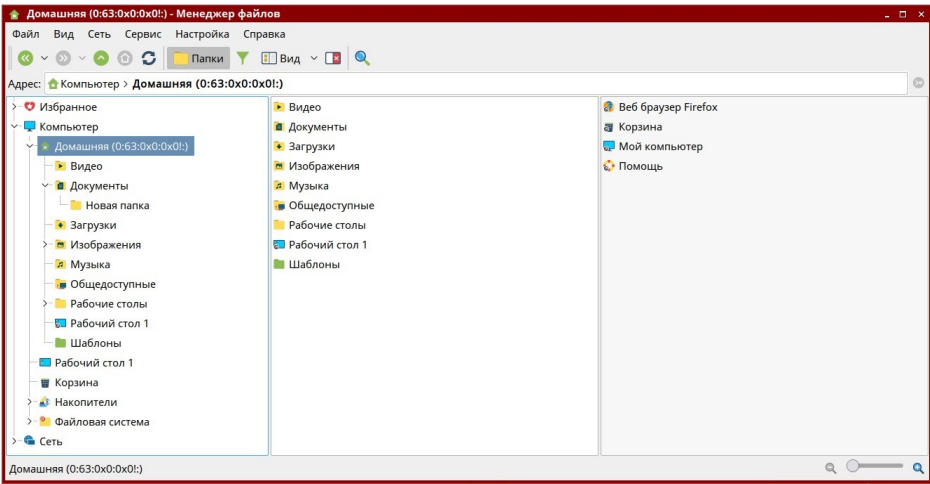
<code>/usr/bin</code>	Дополнительные программы для всех пользователей
<code>/usr/include</code>	Заголовочные файлы
<code>/usr/lib</code>	Основные и дополнительные библиотеки, программное обеспечение (статическая часть)
<code>/usr/local</code>	Третичная иерархия программного обеспечения
<code>/usr/sbin</code>	Дополнительные системные программы администратора
<code>/usr/share</code>	Архитектурно-независимые данные
<code>/usr/src</code>	Исходные коды программного обеспечения

В каталоге `/var` хранятся изменяемые файлы (подробнее приведено в таблице ниже).

Каталог	Назначение
<code>/var/lib</code>	Данные, изменяемые программами в процессе работы
<code>/var/lock</code>	Файлы блокировки, которые означают, что определенный ресурс, файл или устройство занят и не может быть использован другим процессом
<code>/var/log</code>	Журналы программ, установленных в операционной системе
<code>/var/mail</code>	Почтовые ящики пользователей
<code>/var/spool</code>	Очереди ожидающие обработки (очереди печати, непрочитанные или не отправленные письма, задачи cron и т.п.)

5. Менеджер файлов

Менеджер файлов позволяет осуществлять базовые операции с файлами и каталогами: создавать, копировать, переименовать файлы и каталоги. Для файловых операций менеджер файлов предоставляет удобный 2-х панельный интерфейс, а также имеет возможность использовать вкладки. Чтобы отобразить вторую панель работы с файлами и папками, необходимо нажать кнопку  на панели инструментов. Чтобы открыть каталог в новой вкладке, нужно вызвать контекстное меню каталога и выбрать пункт



Администратор системы для возможности осуществления операций с любыми файлами должен вызывать **Менеджер файлов** через команду `sudo fly-fm`.


С точки зрения возможностей, требуемых администратору, возможности менеджера файлов ограничены. Например, невозможно копировать файлы с сохранением атрибутов (владельца, группы, прав), создавать жесткие и символические ссылки.

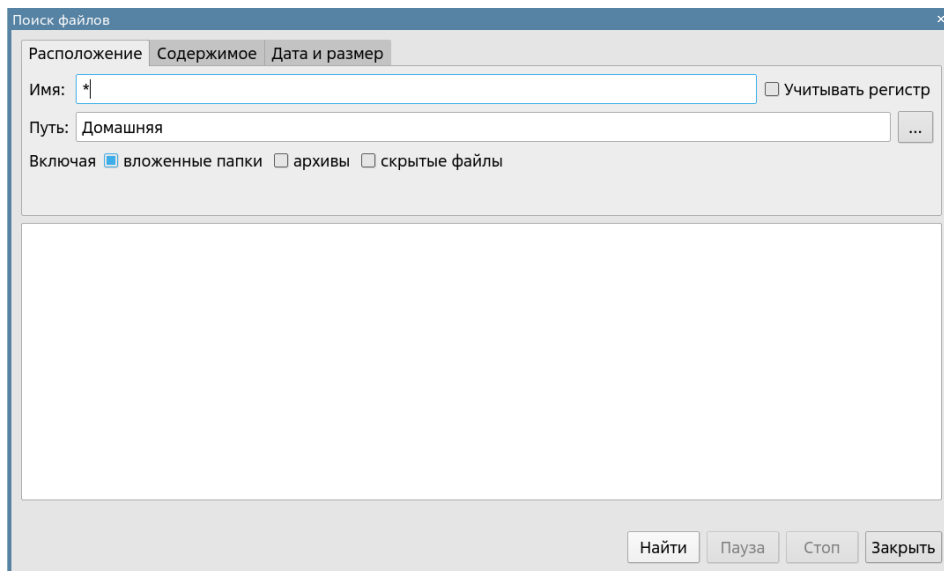
Приложение Ж. Полезные утилиты для работы с файлами в командной строке

Утилиты	Назначение
<code>cd имя_каталога</code>	переход в каталог
<code>pwd</code>	абсолютное имя текущего каталога
<code>ls (-l -a -d -i)</code>	получение информации о файлах/каталогах

<code>dir</code>	получение информации о файлах/каталогах (основные параметры те же, что и у <code>ls</code>);
<code>tree</code>	вывод информации о содержимом каталогов в древовидном формате
<code>touch имя_файла</code>	создание пустого регулярного файла
<code>mkdir имя_каталога (-p)</code>	создание нового каталога
<code>ln -s имя_файла имя_ссылки</code>	создание символической ссылки
<code>ln имя_файла имя_жесткой_ссылки</code>	создание жесткой ссылки;
<code>mknod</code>	создание специального файла устройств
<code>mkfifo</code>	создание именованного канала
<code>cp (-a, -r, -s, -d)</code>	копирование файлов и каталогов
<code>mv</code>	перемещение/переименование файлов и каталогов
<code>rm (-r, -f)</code>	удаление файлов
<code>rmdir имя_каталога</code>	удаление пустого каталога
<code>file</code>	информация о характере содержимого файла
<code>which</code>	поиск программы по имени в каталогах, перечисленных в переменной <code>PATH</code>
<code>find точка_начала_поиска параметры_поиска</code>	поиск файлов по атрибутам файлов

Поиск файлов осуществляется с помощью инструмента менеджера файлов **Поиск файлов**, который может быть вызван несколькими способами:

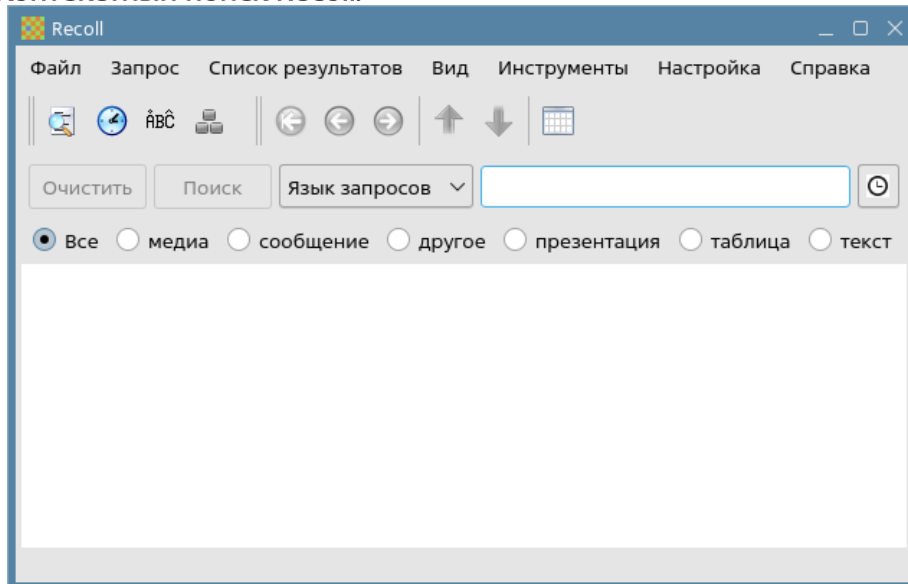
- Кнопка  на панели инструментов.
- Комбинация клавиш **Ctrl+F**.
- На панели главного меню менеджера файлов раскрыть меню **Сервис** и выбрать пункт **Найти**.



Фильтры поиска настраиваются по:

- имени (допускается использование символов подстановки);
- расположению;
- типу файлов;
- содержанию;
- дате изменения и размеру.

Дополнительным средством поиска является программа **Контекстный поиск Recoll**.



При первом запуске Recoll предлагает провести индексирование файлов:

- каталог по умолчанию /home (/media — исключен);
- параметры индексирования настраиваются встроенными инструментами.

Во время индексации не выполняется деструктивное выделение корней, поэтому вся информация попадает в индекс.

Во время запроса по умолчанию вводимые пользователем

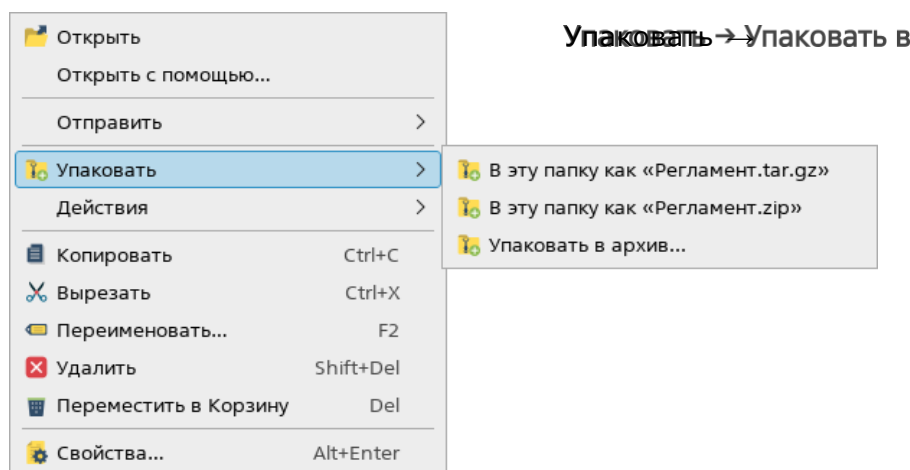
термины сопоставляются, затем сопоставляются с основной базой данных, и запрос расширяется, чтобы включить все производные термины.

Фактически, Recoll представляет собой файловую базу данных с графическим конструктором запросов и фильтров.

6. Архивация и сжатие данных

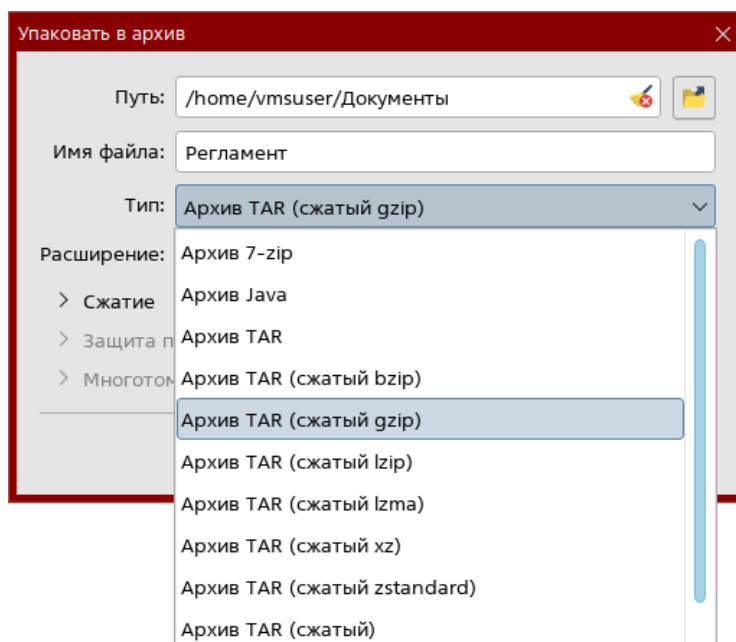
Основным средством пользователя для работы с файлами архивов является Менеджер файлов. Алгоритмы работы с архивами в менеджере не отличаются от работы с обычными файлами.

Создание архива осуществляется через контекстное меню **Упаковать**.



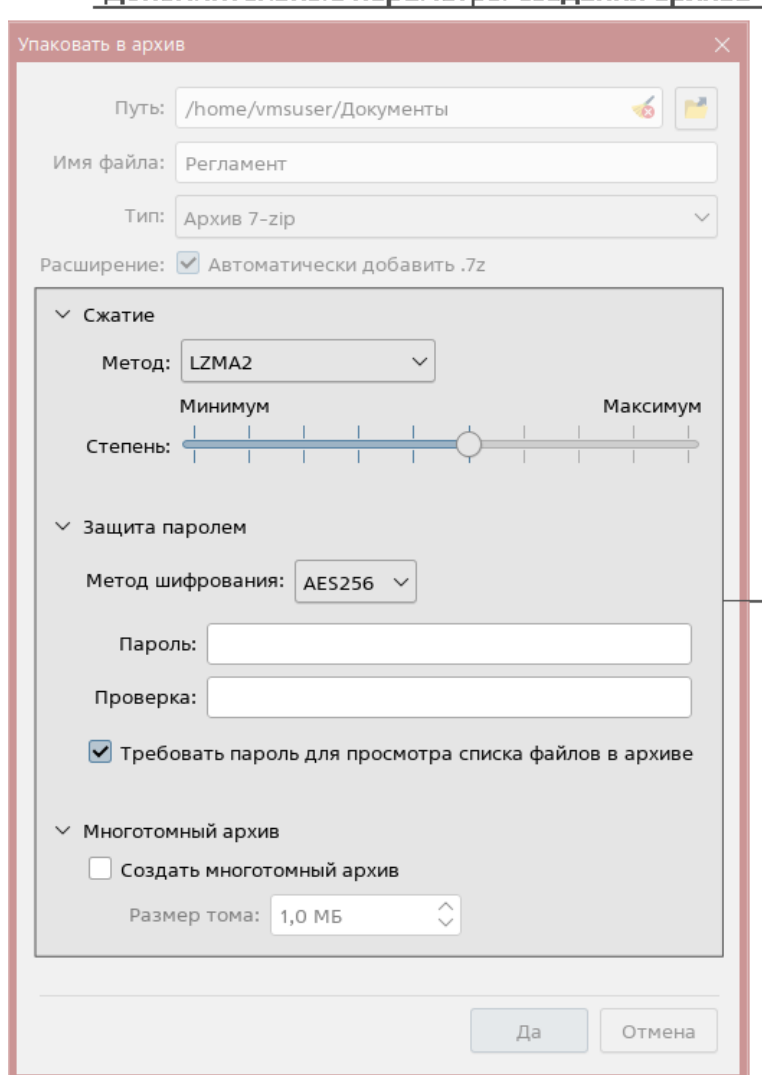
Если в контекстном меню файла выбрать **архив...**, то можно создать архив следующих типов:

- TAR (с различными алгоритмами сжатия);
- 7-ZIP;
- ZIP;
- JAVA.



В зависимости от типа архива поддерживается степень сжатия, шифрование, создание многотомных архивов.

Дополнительные параметры создания архива



Дополнительным средством для работы с архивами является утилита **Работа с архивами Ark**. Утилита представляет собой графическую оболочку для консольных утилит работы с архивами, которые позволяет просматривать содержимое архивов без распаковки, создавать и распаковывать архивы.

