

Архитектура микропроцессора. Регистры процессора

Что это

Batch-файлы — это простые программы для операционной системы Windows, написанные на языке пакетной обработки команд (batch). Они представляют собой текстовые файлы с расширением `.bat` или `.cmd`, содержащие последовательность команд, которые выполняются последовательно при запуске файла.

Основные возможности

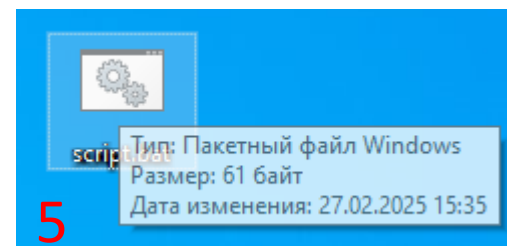
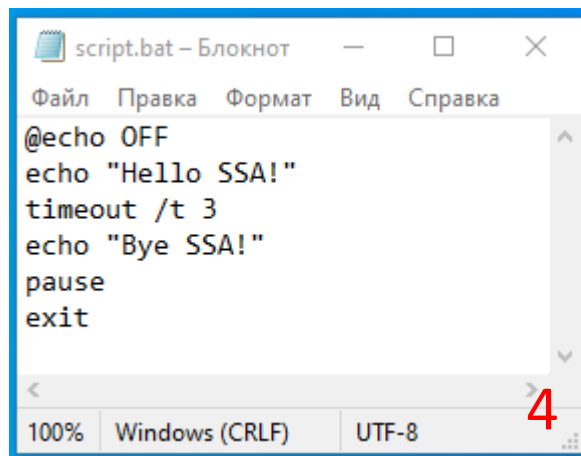
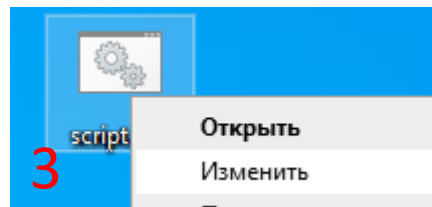
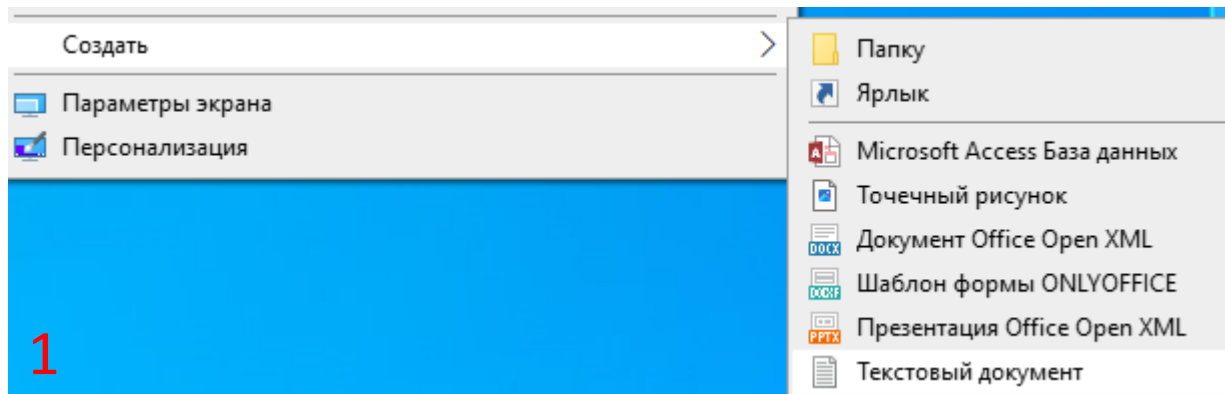
- Автоматизация рутинных задач
- Простота написания
- Совместимость
- Запуск с параметрами

Создание bat файла

Первый и самый простой способ создать bat файл — использовать стандартную программу «Блокнот», присутствующую во всех актуальных версиях Windows.

Создание bat файла

1. Запустить Блокнот
2. Ввести в блокнот код bat файла
3. В меню блокнота выбрать «Файл» — «Сохранить как», выбрать место сохранение файла, указать имя файла с расширением .bat и, обязательно, в поле «Тип файла» установить «Все файлы».
4. Нажать кнопку «Сохранить».



Создание bat файла

Существуют и другие способы сделать bat файл, но все они сводятся к написанию команд по одной команде в строке в текстовый файл в любом текстовом редакторе (без форматирования), который затем сохраняется с расширением .bat

Запуск программы

Для запуска программы

```
start "" путь_к_программе
```

Если путь содержит пробелы, нужны " "

```
start "" "C:\Program Files\program.exe"
```

Можно указать параметры

```
start "" c:\windows\notepad.exe file.txt
```




*script.bat – Блокнот



Файл Правка Формат Вид Справка

```
@echo OFF  
start "" notepad.exe  
start "" notepad.exe "C:\Users\221\Desktop\script.bat"
```



Стр 3, стлб 1

100%

Windows (CRLF)

UTF-8



Запуск bat файла

Запуск другого bat файла из текущего файла

```
call путь_к_файлу.bat параметры
```

script.bat – Блокнот

Файл Правка Формат Вид Справка

```
@echo OFF  
echo "Hello!"  
timeout /t 3
```

<

100% Windows (CRLF) UTF-8

*1.bat – Блокнот

Файл Правка Формат Вид Справка

```
call "C:\Users\221\Desktop\scrip
```

< >

100% Windows (CRLF) UTF-8

Запуск bat файла

Переданные параметры при запуске можно прочесть внутри другого bat файла, например, мы вызываем файл с параметрами:

```
call file2.bat параметр1 параметр2 параметр3
```

Запуск bat файла

В file2.bat можно прочесть эти параметры и использовать в качестве путей, параметров для запуска других программ таким образом:

```
echo %1
```

```
echo %2
```

```
echo %3
```

```
pause
```

1.bat – Блокнот

Файл Правка Формат Вид Справка

```
call "C:\Users\221\Desktop\script.bat" SSA
```

script.bat – Блокнот

Файл Правка Формат Вид Справка

```
@echo OFF  
echo "Hello!"  
echo %1  
timeout /t 3
```

C:\Windows\system32\cmd.exe

```
C:\Users\221\Desktop>call "C:\Users\221\Desktop\script.bat" SSA  
"Hello!"  
SSA  
Время ожидания 1_сек., нажмите любую клавишу для продолжения ...
```

Остановка

По умолчанию, окно команд закрывается сразу после выполнения всех команд. Если нужно прочитать информацию внутри окна, используйте команду `pause` — она остановит выполнение команд

Задержка

Иногда, перед выполнением очередной команды нужно подождать некоторое время (например, до полного запуска первой программы). Для этого можно использовать команду:

```
timeout /t время_в_секундах
```


script.bat – Блокнот

Файл Правка Формат Вид Справка

```
@echo OFF  
echo "Hello!"  
timeout /t 3  
pause
```

C:\Windows\system32\cmd.exe

"Hello!"

Время ожидания 0 сек., нажмите любую клавишу для продолжения ...
Для продолжения нажмите любую клавишу . . . _

Свернутый/развернутый запуск

При желании, вы можете запустить программу в свернутом виде или развернутом видео с помощью параметров MIN и MAX перед указанием самой программы, например:

```
start "" /MIN c:\windows\notepad.exe
```

ЕCHO

Команда используется для отображения сообщений на экране. Она полезна для информирования пользователя о ходе выполнения сценария.

`echo` Это тестовый текст.

Дополнительные опции:

`@echo off` — подавляет вывод всех последующих команд на экран.

`echo.` — выводит пустую строку.

CD

Позволяет перейти в нужную директорию.

```
cd C:\Windows
```

COPY

Копирует один или несколько файлов из одного места в другое.

```
copy file.txt C:\DestinationFolder
```

Параметры:

/Y — подавление запроса подтверждения перезаписи существующего файла.

/V — проверка правильности копирования.

DEL (или ERASE)

Удаляет файл или группу файлов.

```
del file.txt
```

Параметры:

/Q — тихий режим удаления (без запросов подтверждения).

/S — удаление файлов из всех подкаталогов.

MOVE

Перемещает файл или группу файлов из одного места в другое.

```
move file.txt C:\NewLocation
```

MKDIR (или MD)

Создает новую директорию.

```
mkdir NewDirectory
```


FOR

Выполняет команду для каждого элемента в заданном наборе. Это особенно полезно для работы с циклами.

```
for %%f in (*.txt) do echo %%f
```

Этот цикл пройдет через все файлы с расширением .txt в текущей директории и выведет их имена.

%%f — переменная цикла.

IF

Проверяет условия и выполняет соответствующие действия.

```
if exist file.txt (
```

```
    echo Файл существует.
```

```
) else (
```

```
    echo Файл не найден.
```

```
)
```

IF

Условия:

`exist` — проверяет существование файла.

`==`, `!=`, `<`, `>` — сравнения строковых значений.

SET

Устанавливает значение переменной.

```
set myvar=Hello World
```

```
echo %myvar%
```

Использование переменных:

%myvar% — обращение к значению переменной.

XCOPY

Расширенный вариант команды `copy`. Поддерживает рекурсивное копирование и другие дополнительные функции.

```
xcopy C:\SourceFolder C:\DestinationFolder /E /H /I
```

`/E` — копирует все подпапки, включая пустые.

`/H` — копирует скрытые и системные файлы.

`/I` — предполагает, что назначение является директорией, если оно не существует.

REM

Добавляет комментарии в сценарий. Эти строки игнорируются интерпретатором.

```
rem Это комментарий.
```

Где все это пригодится

Автоматическое резервное копирование файлов

```
@echo off
:: Резервное копирование файлов
echo Началось резервное копирование...
robocopy C:\ImportantFiles D:\Backup /MIR /R:1 /W:1
echo Резервное копирование успешно завершено!
pause
```

Где все это пригодится

Преимущества:

- Возможность запуска одним кликом мыши.
- Легкость добавления уведомлений и паузы для контроля процесса.
- Упрощенное планирование регулярных задач через Планировщик заданий Windows.

Создание интерактивных интерфейсов

```
1  @echo off
2  :: Интерфейс выбора
3  echo Выберите одно из действий:
4  echo 1. Запустить приложение
5  echo 2. Открыть файл
6  echo 3. Выход
7
8  set /p choice="Ваш выбор: "
9
10 if "%choice%"=="1" (
11     start app.exe
12 ) else if "%choice%"=="2" (
13     start document.txt
14 ) else if "%choice%"=="3" (
15     exit /b 0
16 ) else (
17     echo Неправильный ввод.
18 )
```

Задание

Создать batch файл, приветствующий пользователя по имени (имя можно передавать в параметрах).

Создать batch файл, создающий папку с произвольным именем, а затем внутри этой папки файл 1.txt

Создать batch файл, запускающий браузер chrome