

Операционные системы

Огородников Юрий Юрьевич
yogorodnikov@gmail.com

6 сентября 2024 г.

Windows. Углубление.

Переменные окружения

- Свойства компьютера → Advanced System Settings → Advanced → Environment Variables

Windows. Углубление.

Переменные окружения

- Свойства компьютера → Advanced System Settings → Advanced → Environment Variables
- переменные TEMP и TMP

Windows. Углубление.

Переменные окружения

- Свойства компьютера → Advanced System Settings → Advanced → Environment Variables
- переменные TEMP и TMP
- переменная WINDIR

Windows. Углубление.

Переменные окружения

- Свойства компьютера → Advanced System Settings → Advanced → Environment Variables
- переменные TEMP и TMP
- переменная WINDIR
- можно добавить свою переменную и присвоить ей значение

Windows. Углубление.

Переменные окружения

- Свойства компьютера → Advanced System Settings → Advanced → Environment Variables
- переменные TEMP и TMP
- переменная WINDIR
- можно добавить свою переменную и присвоить ей значение
- переменные PATH и PATHEXT

Windows. Углубление

Команды в консоли:

- 1 внутренние: команды самой консоли, для них не существует никакого исполняемого файла (echo, dir, cd, type)

Windows. Углубление

Команды в консоли:

- 1 внутренние: команды самой консоли, для них не существует никакого исполняемого файла (echo, dir, cd, type)
- 2 внешние: исполняемый файл есть (find, sort, calc, notepad)

Windows. Углубление

Команды в консоли:

- ❶ внутренние: команды самой консоли, для них не существует никакого исполняемого файла (echo, dir, cd, type)

Как система определяет, что именно надо сделать при корректном запуске команды в консоли:

- ❶ Если набранное является внутренней командой, то она и выполняется.
- ❷ Система проверяет текущую директорию, есть ли в ней запрошенный файл. Если система знает, как файл запускать, то выполняется запуск. Иначе же система пытается подставлять к набранному нами имени команды расширения из PATHEXT. Как только существующий файл найден, программа запускается, а поиск прекращается.
- ❸ Если ранее ничего найти не смогли, то последовательно начинаем перебирать все директории из PATH, применяя действия, описанные в предыдущем шаге (вначале поиск по имени файла, потом с подстановкой расширений).

Windows. Углубление

- 1 Что произойдёт, если написать в консоли `hello.cmd`?
- 2 Что произойдёт, если написать в консоли `hello`?
- 3 Создаем файл `hello.exe`, после чего пишем `hello`. Что произойдёт?
- 4 Создаем файл `hello.cmd.exe`, после чего пишем `hello.cmd`

Windows. Углубление

- 1 Что произойдёт, если написать в консоли `hello.cmd`?
- 2 Что произойдёт, если написать в консоли `hello`?
- 3 Создаем файл `hello.exe`, после чего пишем `hello`. Что произойдёт?
- 4 Создаем файл `hello.cmd.exe`, после чего пишем `hello.cmd`
- 5 Создаем файл `secret.txt.cmd`, выводящий на экран «Hello, secret!», и пишем `secret.txt`

Windows. Углубление

- ❶ Что произойдёт, если написать в консоли `hello.cmd`?
- ❷ Что произойдёт, если написать в консоли `hello`?
- ❸ Создаем файл `hello.exe`, после чего пишем `hello`. Что произойдёт?
- ❹ Создаем файл `hello.cmd.exe`, после чего пишем `hello.cmd`
- ❺ Создаем файл `secret.txt.cmd`, выводящий на экран «Hello, secret!», и пишем `secret.txt`
- ❻ Создаем `secret.txt` с содержимым «секретный текстовый файл» и пишем `secret.txt`. Что произойдёт?

Windows. Углубление

- 1 Что произойдёт, если написать в консоли `hello.cmd`?
- 2 Что произойдёт, если написать в консоли `hello`?
- 3 Создаем файл `hello.exe`, после чего пишем `hello`. Что произойдёт?
- 4 Создаем файл `hello.cmd.exe`, после чего пишем `hello.cmd`
- 5 Создаем файл `secret.txt.cmd`, выводящий на экран «Hello, secret!», и пишем `secret.txt`
- 6 Создаем `secret.txt` с содержимым «секретный текстовый файл» и пишем `secret.txt`. Что произойдёт?
- 7 Создаем файл с именем `echo`, внутри которого `cmd`-скрипт, выводящий «Privet!». Что произойдёт, если написать сейчас в консоли `echo`?

Windows. Углубление

- 1 Что произойдёт, если написать в консоли `hello.cmd`?
- 2 Что произойдёт, если написать в консоли `hello`?
- 3 Создаем файл `hello.exe`, после чего пишем `hello`. Что произойдёт?
- 4 Создаем файл `hello.cmd.exe`, после чего пишем `hello.cmd`
- 5 Создаем файл `secret.txt.cmd`, выводящий на экран «Hello, secret!», и пишем `secret.txt`
- 6 Создаем `secret.txt` с содержимым «секретный текстовый файл» и пишем `secret.txt`. Что произойдёт?
- 7 Создаем файл с именем `echo`, внутри которого `cmd`-скрипт, выводящий «Privet!». Что произойдёт, если написать сейчас в консоли `echo`?
- 8 Переименуем `echo` в `echo.cmd`: `ren echo echo.cmd`. Что сейчас произойдёт, если написать в консоли `echo`?

Windows. Углубление

- 1 Что произойдёт, если написать в консоли `hello.cmd`?
- 2 Что произойдёт, если написать в консоли `hello`?
- 3 Создаем файл `hello.exe`, после чего пишем `hello`. Что произойдёт?
- 4 Создаем файл `hello.cmd.exe`, после чего пишем `hello.cmd`
- 5 Создаем файл `secret.txt.cmd`, выводящий на экран «Hello, secret!», и пишем `secret.txt`
- 6 Создаем `secret.txt` с содержимым «секретный текстовый файл» и пишем `secret.txt`. Что произойдёт?
- 7 Создаем файл с именем `echo`, внутри которого `cmd`-скрипт, выводящий «Privet!». Что произойдёт, если написать сейчас в консоли `echo`?
- 8 Переименуем `echo` в `echo.cmd`: `ren echo echo.cmd`. Что сейчас произойдёт, если написать в консоли `echo`?
- 9 Как запустить наш файл с `echo` вместо внутренней команды, если вдруг очень надо?

Windows. Углубление

- ❶ Что произойдёт, если написать в консоли `hello.cmd`?
- ❷ Что произойдёт, если написать в консоли `hello`?
- ❸ Создаем файл `hello.exe`, после чего пишем `hello`. Что произойдёт?
- ❹ Создаем файл `hello.cmd.exe`, после чего пишем `hello.cmd`
- ❺ Создаем файл `secret.txt.cmd`, выводящий на экран «Hello, secret!», и пишем `secret.txt`
- ❻ Создаем `secret.txt` с содержимым «секретный текстовый файл» и пишем `secret.txt`. Что произойдёт?
- ❼ Создаем файл с именем `echo`, внутри которого `cmd`-скрипт, выводящий «Privet!». Что произойдёт, если написать сейчас в консоли `echo`?
- ❽ Переименоуем `echo` в `echo.cmd`: `ren echo echo.cmd`. Что сейчас произойдёт, если написать в консоли `echo`?
- ❾ Как запустить наш файл с `echo` вместо внутренней команды, если вдруг очень надо?
- ❿ Что происходит, если написать в консоли `calc`?

Windows. Углубление

- ❶ Что произойдёт, если написать в консоли `hello.cmd`?
- ❷ Что произойдёт, если написать в консоли `hello`?
- ❸ Создаем файл `hello.exe`, после чего пишем `hello`. Что произойдёт?
- ❹ Создаем файл `hello.cmd.exe`, после чего пишем `hello.cmd`
- ❺ Создаем файл `secret.txt.cmd`, выводящий на экран «Hello, secret!», и пишем `secret.txt`
- ❻ Создаем `secret.txt` с содержимым «секретный текстовый файл» и пишем `secret.txt`. Что произойдёт?
- ❼ Создаем файл с именем `echo`, внутри которого `cmd`-скрипт, выводящий «Privet!». Что произойдёт, если написать сейчас в консоли `echo`?
- ❽ Переименуем `echo` в `echo.cmd`: `ren echo echo.cmd`. Что сейчас произойдёт, если написать в консоли `echo`?
- ❾ Как запустить наш файл с `echo` вместо внутренней команды, если вдруг очень надо?
- ❿ Что происходит, если написать в консоли `calc`?
- ⓫ Как сделать так, чтобы при набирании в консоли `calc` калькулятор не открылся?

Windows. Углубление

Работа с переменными окружения из консоли

- в консоли: набрать set

Windows. Углубление

Работа с переменными окружения из консоли

- в консоли: набрать `set`
- получение значения переменной: `echo %PATH%`

Windows. Углубление

Работа с переменными окружения из консоли

- в консоли: набрать `set`
- получение значения переменной: `echo %PATH%`
- `echo %RANDOM%`

Windows. Углубление

Работа с переменными окружения из консоли

- в консоли: набрать `set`
- получение значения переменной: `echo %PATH%`
- `echo %RANDOM%`
- `set OurVar=ourVal`

Windows. Углубление

Работа с переменными окружения из консоли

- в консоли: набрать `set`
- получение значения переменной: `echo %PATH%`
- `echo %RANDOM%`
- `set OurVar=ourVal`
- `echo %OurVar%`
- `echo %OURVAR%`

Windows. Углубление

Работа с переменными окружения из консоли

- в консоли: набрать `set`
- получение значения переменной: `echo %PATH%`
- `echo %RANDOM%`
- `set OurVar=ourVal`
- `echo %OurVar%`
- `echo %OURVAR%`
- удалить переменную: `set OurVar=`

Windows. Углубление

Обработка переданных аргументов

```
@echo off
```

```
echo %1
```

```
args first second 3 4
```


Windows. Углубление

Обработка переданных аргументов

```
@echo off  
echo %1  
args first second 3 4  
echo %2
```

Windows. Углубление

Обработка переданных аргументов

```
@echo off  
echo %1  
args first second 3 4  
echo %2  
echo %0
```

Windows. Углубление

Обработка переданных аргументов

```
@echo off
```

```
echo %1
```

```
args first second 3 4
```

```
echo %2
```

```
echo %0
```

Что будет, если запустить без аргументов?

Windows. Углубление

Обработка переданных аргументов

```
@echo off
```

```
echo %1
```

```
args first second 3 4
```

```
echo %2
```

```
echo %0
```

Что будет, если запустить без аргументов?

в консоли запускаем: args 4 3 2 1 5 6 7 8 9 10 11 12

Windows. Углубление

Обработка переданных аргументов

```
@echo off
```

```
echo %1
```

```
args first second 3 4
```

```
echo %2
```

```
echo %0
```

Что будет, если запустить без аргументов?

в консоли запускаем: args 4 3 2 1 5 6 7 8 9 10 11 12

команда shift: превращает 2-й аргумент в 1-й, 3-й во 2-й и т.д.

Windows. Углубление

Цель: переписать `hello.cmd` так, чтобы программа здоровалась с тем, кого ей передадут в качестве аргумента

Windows. Углубление

Цель: переписать `hello.cmd` так, чтобы программа здоровалась с тем, кого ей передадут в качестве аргумента

Усложнение: программа должна иметь свою справку: при передаче ключа `/?` или слова `help` в качестве аргумента эта справка должна выводиться, и программа должна завершать свою работу

Windows. Углубление

Цель: переписать `hello.cmd` так, чтобы программа здоровалась с тем, кого ей передадут в качестве аргумента

Усложнение: программа должна иметь свою справку: при передаче ключа `/?` или слова `help` в качестве аргумента эта справка должна выводиться, и программа должна завершать свою работу

```
if %1==help (  
  echo The program says hello to given username  
  echo.  
  echo %0 username  
  exit /b )
```


Windows. Углубление

Цель: переписать `hello.cmd` так, чтобы программа здоровалась с тем, кого ей передадут в качестве аргумента

Усложнение: программа должна иметь свою справку: при передаче ключа `/?` или слова `help` в качестве аргумента эта справка должна выводиться, и программа должна завершать свою работу

```
if %1==help (  
echo The program says hello to given username  
echo.  
echo %0 username  
exit /b )
```

цель: в случае запуска без аргументов спрашивать имя пользователя, после чего здороваться с введенным результатом

Windows. Углубление

Цель: переписать hello.cmd так, чтобы программа здоровалась с тем, кого ей передадут в качестве аргумента

Усложнение: программа должна иметь свою справку: при передаче ключа /? или слова help в качестве аргумента эта справка должна выводиться, и программа должна завершать свою работу

```
if %1==help (  
echo The program says hello to given username  
echo.  
echo %0 username  
exit /b )
```

цель: в случае запуска без аргументов спрашивать имя пользователя, после чего здороваться с введенным результатом

```
if %1=="/?" (  
echo The program says hello to given username  
echo.  
echo %0 username  
exit /b )  
if %1=="%" (  
set /p username=Who are you?  
) else (  
set username=%1 )  
echo hello, %username%
```

Windows. Углубление

цель: поздороваться со всеми переданными аргументами

Windows. Углубление

цель: поздороваться со всеми переданными аргументами

```
:while  
if NOT '%1'=='' (  
  echo Hello, %1  
  shift  
  goto:while  
) else (  
  exit /b  
)
```

Windows. Углубление

цель: поздороваться со всеми переданными аргументами

```
:while  
if NOT '%1'=='' (  
  echo Hello, %1  
  shift  
  goto:while  
) else (  
  exit /b  
)
```

команды **setlocal** и **endlocal**

Windows. Углубление.

```
@echo off  
cd musor  
del /f /q *  
cd ..  
echo ta-da
```

Windows. Углубление.

```
@echo off
cd musor
del /f /q *
cd ..
echo ta-da  переменная %ERRORLEVEL%: уровень ошибки последней
выполненной команды
значение 0: отсутствие ошибок при выполнении последней команды
```

Windows. Углубление.

```
@echo off
cd musor
del /f /q *
cd ..
echo ta-da    переменная %ERRORLEVEL%: уровень ошибки последней
               выполненной команды
значение 0: отсутствие ошибок при выполнении последней команды
метка :help, переход к которой goto:help
```


Windows. Углубление.

```
@echo off
cd musor
del /f /q *
cd ..
echo ta-da   переменная %ERRORLEVEL%: уровень ошибки последней
              выполненной команды
значение 0: отсутствие ошибок при выполнении последней команды
метка :help, переход к которой goto:help
Символы &, &&, |, ||
```

Windows. Углубление.

```
@echo off
cd musor
del /f /q *
cd ..
echo ta-da   переменная %ERRORLEVEL%: уровень ошибки последней
              выполненной команды
значение 0: отсутствие ошибок при выполнении последней команды
метка :help, переход к которой goto:help
Символы &, &&, |, ||
цикл for
for %f in (*) do @echo %f
```

Windows. Углубление.

```
@echo off
cd musor
del /f /q *
cd ..
echo ta-da  переменная %ERRORLEVEL%: уровень ошибки последней
выполненной команды
значение 0: отсутствие ошибок при выполнении последней команды
метка :help, переход к которой goto:help
Символы &, &&, |, ||
цикл for
for %f in (*) do @echo %f
for %f in (*.txt) do @echo %f
```

Windows. Углубление.

```
@echo off
cd musor
del /f /q *
cd ..
echo ta-da   переменная %ERRORLEVEL%: уровень ошибки последней
              выполненной команды
значение 0: отсутствие ошибок при выполнении последней команды
метка :help, переход к которой goto:help
Символы &, &&, |, ||
цикл for
for %f in (*) do @echo %f
for %f in (*.txt) do @echo %f
for %f in (*.linux*) do @echo %f
```

Windows. Углубление.

```
@echo off
cd musor
del /f /q *
cd ..
echo ta-da   переменная %ERRORLEVEL%: уровень ошибки последней
              выполненной команды
              значение 0: отсутствие ошибок при выполнении последней команды
метка :help, переход к которой goto:help
Символы &, &&, |, ||
цикл for
for %f in (*) do @echo %f
for %f in (*.txt) do @echo %f
for %f in (*.linux*) do @echo %f
В файле *.cmd:
for %%f in (*.txt) do (
echo %%f
echo ===
type %%f
)
```