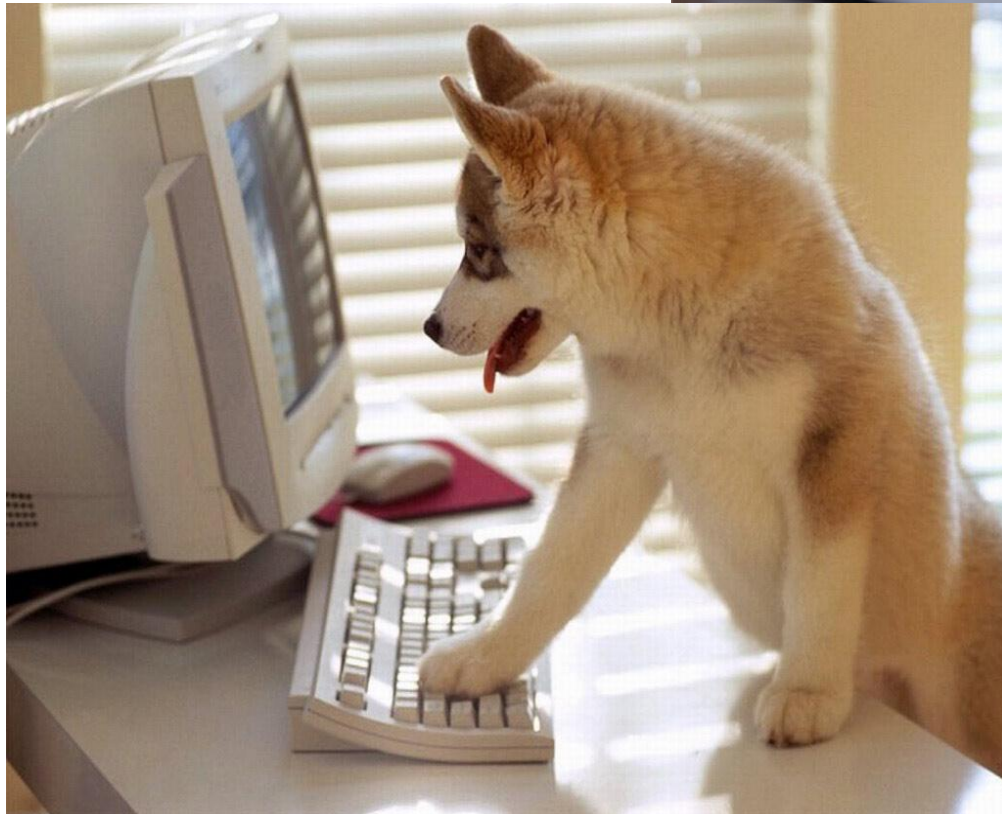
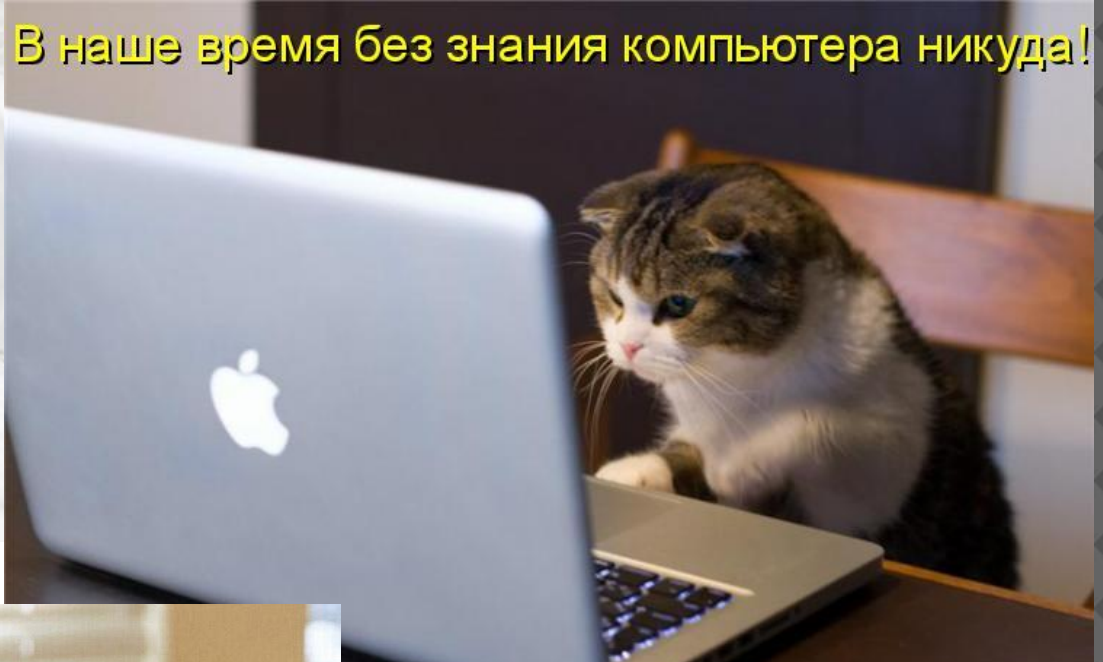


В наше время без знания компьютера никуда!



ИНТЕРФЕЙС

- (от англ. interface) – граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил взаимодействия (управления, контроля и т. д.) между элементами системы.

ИНТЕРФЕЙС

- — это «проводник» между человеком и программой, операционной системой, техническим устройством или способ взаимодействия приложений между собой

- Чтобы обеспечить взаимодействие пользователя с операционной системой и с прикладными программами необходим интерфейс: система передачи команд пользователя операционной системе и ответов системы обратно пользователю.
- То есть «диалог» пользователя с компьютером на специальном языке.
- На сегодня известны две принципиальные возможности организации интерфейса: графический интерфейс и командная строка.

ИНТЕРФЕЙС КОМАНДНОЙ СТРОКИ

- —текстовый интерфейс между человеком и компьютером, в котором инструкции компьютеру даются текстовыми строками - командами (введенными с клавиатуры или составляющими командного файла - скрипта)

НЕМНОГО ИСТОРИИ

- В 1968-69 гг. Кен Томсон и Деннис Ричи представили первый выпуск ОС Юникс, по сути, явившейся прототипом современных операционных систем и связанных с ним понятий, таких, как процессы и файлы.
- В Юникс содержался логичный и лаконичный язык работы с процессами и файлами, реализованный в пользовательском интерфейсе командной строки

РАЗНООБРАЗИЕ КОМАНДНЫХ ОБОЛОЧЕК

- Чем больше вы будете использовать командную строку, тем больше различных программ вам встретятся. Несмотря на огромное разнообразие таких программ, принципы их взаимодействия с пользователем практически не отличаются. Например, в BASH программа ls выводит на экран список файлов и директорий, в Windows - Dir.
- Достаточно набрать требуемую команду и нажать Enter.

КОМАНДНЫЕ ОБОЛОЧКИ LINUX

- (разнообразны, в отличие от Windows)
- Наиболее известны оболочки sh и bash.
Менее - C shell (csh, tcsh), zsh и tcsh.
- На этом список существующих оболочек не заканчивается.

КОМАНДНЫЙ ФАЙЛ (СКРИПТ)

- **Пакетными или командными файлами называются файлы, содержанием которых являются команды.**

СКРИПТЫ ДЛЯ LINUX

- Не официально часто называется «шел», от английского shell – оболочка.
- Bourne shell, исполняемый файл: sh.
- Bourne again shell, исполняемый файл: bash. Название можно перевести, как «Возрождённый шел Борна». Скорее всего самая популярная оболочка на сегодняшний день. Де-факто стандарт для Linux.
- Z shell, исполняемый файл: zsh.

РАЗНИЦА В СИНТАКСИСЕ СКРИПТОВ

<code>#!/bin/sh</code>	<code>#!/bin/csh</code>
<pre>#!/bin/sh if [\$days -gt 365] then echo</pre>	<pre>if (\$days > 365) then echo This is over a year. endif</pre>

<code>#!/bin/sh</code>	<code>#!/bin/csh</code>
<pre>#!/bin/sh i=2 j=1 while [\$j -le 10]; do echo '2 **' \$j = \$i i=`expr \$i '*' 2` j=`expr \$j + 1` done</pre>	<pre>set i = 2 set j = 1 while (\$j <= 10) echo '2 **' \$j = \$i @ i *= 2 @ j++ end</pre>

ПРИМЕРЫ ВСТРОЕННЫХ КОМАНД, КОТОРЫЕ ИСПОЛЬЗУЮТ ДЛЯ СОЗДАНИЯ СВОИХ СКРИПТОВ LINUX

- break выход из цикла for, while или until
- continue выполнение следующей итерации цикла for, while или until
- echo вывод аргументов, разделенных пробелами, на стандартное устройство вывода
- exit выход из оболочки
- kill посылает сигнал завершения процессу
- pwd выводит текущий рабочий каталог

КАК УСТРОЕНЫ BASH-СКРИПТЫ .

- Любой bash-скрипт должен начинаться со строки: `#!/bin/bash`
- в этой строке после `#!` указывается путь к bash-интерпретатору, поэтому если он у вас установлен в другом месте(где, вы можете узнать набрав `whereis bash`) поменяйте её на ваш путь.
- Комментарии начинаются с символа `#` (кроме первой строки).
- В bash переменные не имеют
- Команды оболочки отделяются знаком перевода строки, комментарии выделяют знаком решётки.

ПРИМЕРЫ КОМАНД

- **cd** каталог- переход в указанный каталог.
- **Mkdir** имякаталога - создание каталога
- **File** определяет тип файла
- **cat** имя_файла - выводит содержимое файла на терминал
- Создание файла **touch** имя_файла или >имя_файла
- Копирование файла в другой файл или каталог **cp** файл-источник файл-или-каталог-приемник
- **rm** имя_файла - удаляет файл
- **rmdir** имя_каталога - удаляет каталог
- **rm -rf** имя_каталога- удаляет каталог рекурсивно и молча.

СКРИПТ В WINDOWS

- Пакетными или командными файлами называются файлы, содержанием которых являются команды.
- В Windows - это «батники» - из-за расширения
- создать текстовый файл в "блокноте"
- Изменить расширение на bat (сохранить как, тип - все файлы, имя.bat)

- В батнике, как правило. начинаются с команды @ echo off -отключение вывод команд на экран
- ECHO позволяет вывести текст
- PAUSE -приостанавливает выполнение до нажатия пользователем любой клавиши.
- REM Это комментарий
- chcp 1251 >nul
- REM chcp 1251 >nul кодировка делает читаемой кириллицу
- REM Это комментарий

Hostname

%computername%

rem Имя компьютера

Ipconfig

Rem IP-адрес

%username%

Rem Имя пользователя

Date=01.01.2020

Rem задаем дату

%Date%

Rem значение текущей даты

Echo %date:~6%

Rem вывели подстроку с 6-го символа, т.е.год

Echo %date:~0,2%

rem вывели 2 цифры, начиная с нуля

Set d

Rem Создание переменной d

Set d= %date:~0,2%

Set m= %date:~3,2%

Set y= %date:~6,4%

Rem создали переменные d m y и присвоили им значение дня, месяца, года

КОМАНДЫ РАБОТЫ С ФАЙЛАМИ И КАТАЛОГАМИ

dir

REM отображает содержимое папки

Cd

Rem переход в папку формат cd <путь

Rem в «родительскую» папку cd ..

Rem в корневую cd \ .

Rem Команда cd без параметров - текущий каталог

md

Rem создание папки md

md c:\1\2\3\4

Rem создание дерева папок на диске C, а одного уровня:

md 1 2 B3

rd

Rem удаляет папку rd имя папки

REM ключ /S - удаляет рекурсивно, ключ /Q- без подтверждения

Copy

Copy a.txt C:\b.txt

Rem копирует один или несколько файлов

**copy c:\alpha\1.txt+c:\alpha\2.txt+c:\alpha\3.txt
c:\beta\result.txt**

Rem слияние файлов, в итоге получается текстовый файл

Copy *.txt C:\text.txt

Rem слияние текстовых файлов в текущем каталоге по маске файлов, в итоге получается текстовый файл на C

move

Rem перемещение одного или нескольких файлов

Del

Rem удаление файла del <имя файла>

Rem ключ/Q- убирает запрос подтверждения

Rem /S - удаляющий файлы из подкаталогов

Rem /F удаляющий файлы с атрибутом «Только чтение»

Copy nul file.txt

Rem создание пустого файла

Примеры создания (перезаписи) непустых файлов:

echo hello > t.txt

rem перенаправляется слово с экрана в файл

Tasklist > t.txt

rem в файл записывается перечень выполняемых процессов

> символ перенаправление вывода

Стандартное устройство вывода – экран

>> - записывается в конец файла

- Создать переменную можно с помощью команды `set`.
- Например, `set a=%date%`.
- Создается переменная `a` , ей присваивается значение текущей даты
- `%a%` - обращение к значению переменной
- Если нужно ввести значение с консоли, используют ключ `p`.
- `Set /p a=Enteredvalue:`
- `>` символ перенаправление

РАБОТА С ПЕРЕМЕННЫМИ И СТРОКАМИ

Символы в строке нумеруются, начиная с 0

Если нужно извлечь n символов, начиная с m ,
из строковой переменной str , шаблон:

`%str:~m,n%`

Если нужно взять n последних символов

`%str:~-n%`

Примеры :

```
Set str=Black
```

```
Echo %str:~0,1%
```

(получится B)

```
Set str=Black
```

```
Echo %str:~0,2%
```

(получится Bl)

Set str=Black

Echo %str:~0,2%

(получится Bl)

Set str=Black

Echo %str:~-3%

(получится ack)

Set str=Black

Echo %str:~-1%

(вывод кроме первого символа, получится lack)

Set x=Black

Set y=Cat

Echo %x%%y%

Создание арифметической переменной - ключ /a

Set /a x=5

ДЕЙСТВИЯ С АРИФМЕТИЧЕСКИМИ ПЕРЕМЕННЫМИ

Set /a x+=1

Rem увеличили на 1

Set /a x*=2

Set /a x+=%x%

Rem увеличили в 2 раза

Set /a x=15

Set /a y=4

Set /a xy=%x%/y%

«БАТНИК» С ПАРАМЕТРАМИ

- параметры вызова bat-файла %<цифра 0-9>
- Всего может быть 10 параметров - одновременно существующих независимых переменных. Переменная %0 будет содержать имя .bat-файла и, если вы указали, путь к нему.
- Пример запуска файла abc.bat со следующими параметрами:
- abc.bat a bc def
- Если параметр имеет пробелы, то берется в кавычки

РАБОТА С МЕТКАМИ

- Общий синтаксис:
- `goto <имя метки, указывающую переход>`

- `:<имя_метки>`

Проверка наличия файла:

`IF EXIST < имя или шаблон файла > < команда >`

Условие считается выполненным при обнаружении файла.

проверка появления файла (Утилита `sleep` выполняет ожидание указанное время

`Exist` - проверка существования , `goto` переход по метке, :
имя_метки - обращение к метке) :

`:test`

`if exist c:\1.txt goto go`

`sleep 10`

`goto test`

`:go`

`Notepad`

Команда for в виде числового цикла

Команда for позволяет организовать выполнение повторяющихся однотипных действий.

Можно использовать ее для того, чтобы вывести на экран числа от одного до десяти:

```
for /l %%i in (1,1,10) do echo %%i
```

Переменная `i` называется счетчиком цикла.

В силу своеобразия синтаксиса команды `for`, имя счетчика цикла должно состоять из одной буквы. Причем, если мы пишем командный файл, то перед именем счетчика цикла надо поставить сдвоенный знак процента, если же мы просто набираем команду в командной строке, то одиночный.

После слова `in` указан диапазон изменения счетчика цикла.

Здесь это тройка чисел: начальное значение счетчика, шаг счета, предельное значение счетчика.

При выполнении команды командный процессор сначала присвоит переменной *i* значение 1, а потом на каждом шаге цикла будет увеличивать его на 1, пока оно не превысит 10. Очевидно, таких шагов получится десять. Если бы в качестве шага счета мы указали число 2, то цикл выполнялся бы пять раз. На каждом шаге цикла выполняется тело цикла, написанное после слова `do`.

```
for /l %%i in (1,1,10) do echo %%i
```

В рассмотренном примере это команда `echo`, которая выводит на экран текущее значение счетчика цикла.

```
FOR /f %%a IN ("C:\1.txt") DO echo %%a  
REM выведет все строки файла 1.txt
```

ПОИСК ПОДСТРОКИ

Find

Пример:

```
Find /i file.txt "new"
```

Ключи

Без учета регистра /i

Вывод номера строки /n

Общее количество строк /c

Вывод строк, не содержащих шаблон /v

ПОИСК ФАЙЛОВ

Where

Where [ключ] [где] [шаблон]

Пример:

Where /r c:\ *.txt

Поиск текстовых файлов на диске с рекурсивно.

Ключи

/r -рекурсивный поиск

/t - вывод размера, даты. времени

/f - имя в кавычках