Графические Системы. Часть II

Лекция № 10 (ПЗ № 8)

Программирование графического пользовательского интерфейса средствами X-WINDOW. ИПВУ. Tcl/TK

Программирование в TCL

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL. Работа с файлами

Рассмотрим несколько примеров работы с файлами в **Tcl**. По ходу рассмотрения изучим основные встроенные команды, предназначенные для работы с файлами.

Tcl работает с файловой системой через систему буферов. Этот механизм представляет файл как поток символов, который начинается с началом файла и заканчивается в конце файла. Таким образом, работа с файлом выглядит аналогично простому выводу данных на экран командой **puts**, только добавляется новый аргумент. Данные из файла читаются командой **gets**, и так же могут считываться с клавиатуры.

Перед началом работы с файлом программа должна открыть файл для каких-нибудь действий (чтение, запись, или оба) командой **open**. Как только файл откроется, программа может выполнить **gets** или **puts** для чтения или записи информации.

Когда программа закончит работать с файлом, следует его закрыть командой **close**. Число одновременно открытых программой файлов ограничено операционной системой. **Поэтому если не закрывать файлы, то программа может перестать работать.**

Итак, основные команды Tcl для работы с файлами и их короткое описание:

`open` ***имяФайла ?доступ?***

Открывает файл и возвращает указатель, который используется для доступа командами **gets**, **puts**, **close** и другими. ***имяФайла*** - имя файла, который будет открыт. Желательно, с указанием полного пути к нему. ***доступ*** - режим работы с файлом.

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL. Работа с файлами

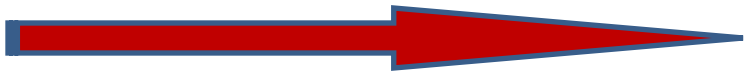
- ✓ **r** - Режим чтения. Файл должен существовать.
- ✓ **r+** - Режим чтения и записи. Файл должен существовать.
- ✓ **w** - Режим записи. Создаёт файл если он не существует, или обнуляет существующий.
- ✓ **w+** - Режим записи и чтения. Создаёт файл в случае необходимости, или обнуляет существующий.
- ✓ **a** - Режим добавления, открывает файл для записи. Файл должен существовать. Текущая позиция устанавливается на конец файла.
- ✓ **a+** - Режим добавления, открывает на запись. Если файл не существует, то создаётся. Если существует - позиция устанавливается на конец файла.

```
close файл
```

Файл - указатель на файл. Команда закрывает файл открытый **open**.

```
gets файл ?имяПерем?
```

Считывает строку из ***файла***, и удаляет символ новой строки.
Если задан аргумент ***имяПерем***, **gets** возвращает количество прочитанных символов (или -1 если достигнут конец файла), и помещает прочитанную строку в ***имяПерем***.
Если ***имяПерем*** не задано, то **gets** возвращает прочитанную строку. Пустая строка может быть возвращена в случае если прочитана пустая строка из файла, или достигнут конец файла.



Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL. Работа с файлами

`puts ?-nonewline? ?файл? строка`

Записывает ***строку*** в ***файл***. Если файл не задан, то выводится на экран.

`read ?-nonewline? файл`

Читает все оставшиеся символы из ***файла***, и возвращает как строку. Если задана опция **-nonewline**, то если последний символ новой строки - он будет удалён.

`read файл колСимв`

Читает не более ***колСимв*** символов из файла и возвращает как строку.

`seek файл позиция ?отсчет?`

Изменяет текущую позицию работы с ***файлом***. ***Позиция*** - количество символов, на которую будет сдвинута текущая позиция. С помощью аргумента ***отсчет*** можно поменять точку отсчета позиции. Значение ***отсчет*** может быть одним из:

- **start** - *позиция* измеряется от начала файла.
- **current** - *позиция* измеряется от текущей позиции в файле.
- **end** - *позиция* измеряется от конца файла.

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL. Работа с файлами

`tell` ***файл***

Возвращает номер текущей позиции в *файле*.

`flush` ***файл***

Записывает все данные *файла*, хранящиеся в буфере.

`eof` ***файл***

Возвращает 1 если достигнут конец файла, иначе 0.

При работе с файлами следует иметь в виду:

Все данные в Tcl хранятся как строки символов. Поэтому читая двоичный файл результат получится непредсказуем.

Есть несколько стандартных указателей на файлы, которые создаются автоматически при запуске программы. Их можно использовать так же, как и обычные указатели:

- ✓ **stdin - стандартный поток ввода (клавиатура)**
- ✓ **stdout - стандартный поток вывода (экран)**
- ✓ **strerr - стандартный поток ошибок (экран)**

Количество открытых файлов ограничено, поэтому помните о том, что надо закрывать неиспользуемые файлы.

Чтобы определить конец файла используйте команду eof перед чтением очередной строки.

Работа с файлом осуществляется через буфер. Содержимое буфера хранится в памяти до того момента, пока процессор освободится, после чего данные будут записаны на диск. Можно заставить компьютер записать данные на диск командой flush. Когда используется команда close или программа завершается, данные так же скидываются на диск. Однако при сбое содержимое буфера может не записаться на диск, и данные будут потеряны.

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL. Работа с файлами. Примеры

Пример 1. Построчное копирование файлов

Представим данное задание в виде псевдокода:

```
while  another line is read into Line from input file
       write Line to output file
endwhile
```

```
set InFile [open input.txt r]
set OutFile [open output.txt w]
while {-1 != [gets $InFile
Line]} {
puts $OutFile $Line
}
close $InFile
close $OutFile
```

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL. Работа с файлами. Примеры

Пример 2. Заполнение **listbox'a** с именем **.lb1** значениями цветов из подготовленного заранее файла **rgb1.txt**

```
set rgb [open rgb1.txt r]
while {[eof $rgb] == 0} {
set colorstr [gets $rgb]
if {$colorstr != ""} {
.lb1 insert 0 $colorstr
}
}
.lb1 delete 0
close $rgb
```

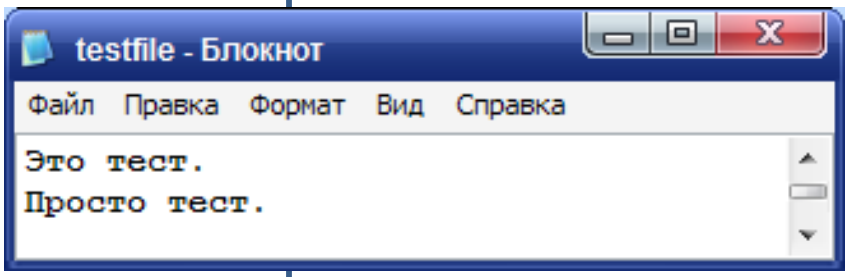
Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL. Работа с файлами. Примеры

Пример 3. Работа с файлом с ***доступом*** **w+**. Открытие в режиме **w+** позволяет изменять записанные данные, но очищает файл перед использованием.

```
set fileid [open "testfile" w+]

seek $fileid 0 start

puts $fileid "Это тест.\nПросто тест."

seek $fileid 0 start

set chars [gets $fileid line1];
set line2 [gets $fileid];

puts "$chars символов в строке
\"$line1\""
puts "Вторая строка в файле: \"$line2\""

seek $fileid 0 start

set buffer [read $fileid];
puts "\nВ файле содержится
текст:\n$buffer"
close $fileid
```

В результате в home-директории появляется файл testfile:



Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL. Работа с файлами. Примеры

Пример 3. Работа с файлом с ***доступом*** **w+**. Открытие в режиме **w+** позволяет изменять записанные данные, но очищает файл перед использованием. Что получается при отладке этого примера в интерактивном режиме интерпретатора: