

Appendix 5.1.2

Department of Computing Final Year Project Proposal

Name: Daniel Young

Size: double

Course: BSc (Hons) Software Engineering
Brown

Discussed with (lecturer): Rosemary

Problem Context

Project To Be Built

Development Environment for Scripts complete with Interpreter, Form Creator and “Compiler”.

The Windows environment doesn't provide a Scripting tool for users to create simple applications or scripts. The concept of allowing small businesses and users to create simple “programs” died out with the days of DOS. For my project I intend to build a complete scripting suite for the Windows platform.

Benefits include: Being able to create simple applications, such as a “System Information” tool, “Text File Search” tool, or something like an “Application Menu”.

Past Experiences

My past experience with string manipulation includes creating a very simple scripting language using Visual Basic. The interpreter was used in conjunction with Internet Information Services in order to parse and interpret any *.exp file just as Microsoft's “asp.dll” parses and interprets any *.asp file.

The syntax of this language was basic, and allowed use of

1. Variables (Typeless)
2. Iteration (While)
3. If Statements (including Nested)

Initial Testing

For this project I have already made a few prototype test applications that:

1. Parse a mathematical formula e.g.
 $\#add(10, \#div(\#mul(10,10), \#sub(100,90))) = 20$
2. Create Form Buttons dynamically.
3. Appended data onto the end of a standard executable to see if it ran flawlessly, which it did. This opens up the opportunity of allowing users to in effect “Compile” their scripts.

The Problem

I intend to make a development environment for a very simple Windows based Scripting language that allows a computer user with very little, if any programming knowledge to create scripts to perform various actions. The development environment must be simple to use, and include all aspects required to fully create a piece of software.

The development environment will be able to create Forms like one can do in C++ Builder and Visual Basic, for example place Buttons and Text Boxes on the Form to allow interaction with the user of the script. It will also contain a debug window with a list of all current variable contents.

Example Functions:

#abs(value) - returns the absolute value of an integer, (-10 becomes 10, 10 stays 10)
#asc(char) - returns the ASCII value of the first or only character in a string.
#date() - returns the current date.
#time() - returns the current time.
#equal(value, value) - equates to null if not equal, otherwise 1.

Specific Objectives

- Investigate and find information concerning Interpretation and Compilation.
- Investigate Windows API functionality to convert complex functions into simple one line instructions.
- Decide on a Syntax and create a full complement of functions and commands for the language, ranging from simple arithmetic to more complex string manipulation.
- Create and fully debug an interpreter for the language, made as efficient as possible using a combination of assembly and C++.
- Create and fully debug a development environment for the scripts, to allow:
 - a. Create, Open, Save and Close script files.
 - b. Create and Modify Forms, for allowing the interaction of scripts with users.
 - c. Run the script files, be it Step Through or Normal.
 - d. Pause the script.
 - e. View contents of variables used at any time.
 - f. “Compile” the script, by appending it onto a base executable containing the Interpreter.
- Create an optional scheduler to run scripts on a set date at a set time.
- Create help files.

The Approach

Tools

C++ Builder to create the project

Techniques

Reverse Polish to possibly interpret accurate mathematical expressions.

Resources

- C++ Builder to create the interpreter and development environment.
- Windows Help File Maker for Help Files.

- Microsoft Project 2003 for Project Planning and Time Management.
- Internet for Reference and views.
- Possible surveys.

Suggested MiniPaper Title

Interpretation versus Compilation,
their benefits and the situations where they are best used.

References

Mini Paper – JPEG/GIF Compression

- http://www.pagetutor.com/about_gifs/
- <http://www.webdevelopersnotes.com/graphics/jpgs.php3>
- Article : <http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html>
- Article : <http://www.faqs.org/faqs/jpeg-faq/part1/>
- Article : http://netghost.narod.ru/gff/graphics/book/ch09_04.htm
- <http://www.worldstart.com/guides/imagefile.htm>

Mini Paper – INTERPRETER/COMPILATION

- <http://www.webopedia.com/TERM/c/compiler.html>
- <http://www.cs.utah.edu/outreach/contest/1997/takehome/node2.html>
- <http://www.everything2.com/index.pl?node=source%20code>