Introduction to REXX Programming Language

Clarence Burg
September 9, 2011

1 Introduction to REXX

REXX is a programming language that has much less syntax and extraneous content than other available languages. As such, it is an excellent choice for transferring pseudo-code into a programming implementation, since the pseudo-code can be directly mapped to statements in REXX, without a lot of extra syntax and commands that are unfamiliar to the users.

When learning to program for the first time, a student is forced to master several distinct concepts simultaneously - usage of an unfamiliar programming environment, programming syntax and the actual programming itself. REXX eliminates a lot of the unfamiliar programming syntax as well as the need to master an unfamiliar programming environment. However, a student must learn some initial basics about REXX in order to get started.

2 Installation of REXX

You can download a Windows version of REXX called Regina from my faculty homepage at http://faculty.uca.edu/clarenceb . After downloading the file, go to your Download directory and double click on the file called Regina. This action will launch the Regina installer. Simply click OK and accept the defaults. These actions will install Regina in a directory called C:\Regina .

I suggest that you create a folder called 'REXX' in your 'My Documents' folder to store your REXX programs. I will assume that you have done so.

3 A Simple REXX program

One of the simplest programs is the 'Hello World !' which prints out 'Hello World' when it is run. Once you can write and run this program, you have mastered the programming environment necessary to write and run all REXX programs.

To write the program, you need to open a file editor. I suggest that you use Windows WordPad which can be found by clicking on 'START', 'All Programs', 'Accessories', and finally 'WordPad'. This program is a simple text editor. Type the following program into WordPad:

```
say 'Hello World !'

exit
```

The trick now is to save the program correctly. You need to save the program to the folder 'REXX' in your 'My Documents' folder, and you need to save the file as a text document. To do this, select 'File' and 'Save As'. These actions open the 'Save As' dialogue box. Navigate to the 'REXX' folder. Save the file as 'helloWorld' and save as type 'Text Document'. Click 'Save'.

You may get an error message stating that all formating will be lost. You do not want the formating, as it involves special hidden characters that 'REXX' can not interpret. If you get this error message, click 'OK'

You have written your first REXX program !

To run the program, you will use the DOS command prompt (Disk Operating System). To open a command prompt, click on 'START', 'All Programs', 'Accessories', and finally 'Command Prompt'. This action opens up a black window into which you can type commands. The default location within the file structure is 'C:\Documents and Settings\<User Name>' where <User Name> is the name of the user of the computer.

Type 'dir'. This command shows a directory listing of all the files in the current directory.

To change directory to the directory 'My Documents', type 'cd M' and tab until the command looks like 'cd "My Documents" ' and hit 'enter'. The command 'dir' and 'cd' are examples of DOS commands. DOS or Disk Operating System predates Windows and is one of the ways that people interacted with computers in the early days of personal computing. In the early days, black spaces had special meaning in DOS commands as separators between commands and options within the command. Now, in Windows, a file or a folder can have spaces in the name. Thus, the name of the folder is in double quotes as "My Documents". The use of the tab button scrolls through all files or folders that start with "M" in the command that you used to locate the "My Documents" folder.

Once you are in the "My Documents" directory, type 'dir' to get a directory listing. You should see a directory called 'REXX'. Type 'cd REXX' to change directory to the REXX directory.

Type 'dir' again. This time, you should see three files in the directory. The output of the 'dir' command should look like

```
 Volume in drive C has no label.
 Volume Serial Number is 7C81-CE8D

 Directory of C:\Documents and Settings\Student\My Documents\REXX

09/08/2011  09:04 AM    <DIR>          .
09/08/2011  09:04 AM    <DIR>          ..
09/08/2011  09:04 AM                29 helloWorld.txt
               1 File(s)             29 bytes
               2 Dir(s)   7,289,281,314 bytes free
```

This output shows the two directories which are . and .. and one file which is helloWorld.txt . The directory . is the current directory. If you type 'cd .', you will change directory to the current directory, so you will stay in the same place in the directory structure. (Note: a directory and a folder are the same thing.) The directory .. is the directory one up from the current directory. If you type 'cd ..', you back up one directory to the "My Documents" directory.

If you have changed to a different directory, please 'cd' back to the REXX directory.

Now that you are in the REXX directory, you can run the helloWorld program using the Regina REXX interpreter, by typing

'regina helloWorld.txt'

You should see 'Hello World !' as the output. Please note that you must type in the program file name exactly, including the .txt extension.

Congratulations! You have written and run your first REXX program. You have mastered the REXX programming environment!

4 Counting from 1 to N

In the first program that you wrote in REXX, you learn about the command to print a statement to the screen, which is the 'say' command. In this section, you will learn about the 'pull' command and the 'do loop' structure.

Suppose you want to print all the numbers from 1 to N where N is specified by the user. For instance, if the user of the program wants to output the numbers from 1 to 8, the user will enter 8.

The process of counting from 1 to N involves knowing what N is and looping or iterating through the numbers from 1 to N. Hence, the pseudo-code looks like

```
Ask for N
Get N from the user

Loop over the index i as it goes from 1 to N
    print i
end loop

exit the program
```

In REXX, to ask for N, you can have a print statement that ask the user to enter N. The command 'pull N' get the value typed in my the user and stores the result in N. We will assume that the user enters an integer.

To loop over the index i as it goes from 1 to N, we using a 'do loop'. This loop involves three things - the name of the index, the starting value and the ending value. This loop has the same three components as a summation statement, such as

$$\sum_{i=1}^{N}$$

which states that the index is 'i' which starts at 1 and ends at N. The 'do loop' has the same components and looks like

```
do i=1 to N
```

This command starts the 'do loop'. The command 'end' specifies the end of the do loop. The REXX code looks like

```
say 'Enter an integer to count up to'
pull N

do i=1 to N
    say i
end

exit
```

Remember to use WordPad to write the program. Save it in the REXX folder as 'count.txt' and remember to save it as type 'Text Document'.

Using the Command Prompt, make sure you are in the REXX folder. Once in the REXX folder, type

```
regina count.txt
```

In this program, please note the different usage of the command 'say'. In the command say 'Enter an integer to count up to', the program prints out 'Enter an integer to count up to'; whereas, in the command say i, the program prints out the value of i, rather than i itself. The reason for this difference is the use of the quotation marks. Without the quotation marks, the program will output the value stored in the variable; with the quotation marks, the program will output what is typed between the quotation marks.

5 Summations

Several of the components from the previous code can be used in a code to calculate the summation

$$\sum_{i=1}^{N} i$$

In this summation, the user will specify the value of N, and the loop will add up the summation as i goes from 1 to N. As pseudo-code, the algorithm looks like

```
Ask for N
Get N from the user

Set the summation to 0

Loop over the index i as it goes from 1 to N
    add i to the summation
end loop

print out the answer

exit the program
```

Any variable that is used by a program should be initialized to some value prior to its first use. In this program, we have three different variables, i, N and the summation. The user specifies the value of N, the 'do loop' initializes i to 1, and the summation is set to 0 prior to entering the loop. In any program, it is critical that all variables are initialized to some value prior to their first use.

Once translated to REXX, the pseudo-code looks like

```
say 'Enter an integer to sum up to'
pull N

summation = 0

do i=1 to N
    summation = summation + i
end

say 'The summation from 1 to ' N ' is ' summation

exit
```

The statement 'summation = summation + i' is an assignment statement rather than an equation. This statement assigns the value of 'summation + i' to the variable 'summation'.