2024 Fall CPSC 240 Assignment 5

Execution Speed

Goals

Show how to measure the run time required by a block of assembly code.

In the process we will learn about cpu frequency, conversion of tics to nanoseconds, and how to read the clock inside an X86 processor.

Background information

From mathematics we know that ex can be computed by formula

$$\sum_{k=0}^{n} \frac{x^k}{k!}$$

which is x^0/0! + x^1/1! + x^2/2! + x^3/3! + …… + x^n/n!

Given a fixed x, if you computed that sum for an infinity of terms, then you will have the true value of e raised to x.

This project

Make an assembly function that receives two parameter, x and n. The function computes an approximation of e^x by computing the sum of (n+1) terms of the formula shown above.

Notice that we count terms from 0. Therefore, the term x^n/n! is truly the $(n+1)^{st}$ term.

//Inserted comment. If you did only that much of this software specification, then you have completed a substantial software product. It is worth partial credit. In a lecture I talked about the value of partial credit.

Sample execution for partial credit

Welcome to Taylor Series by Jane Eyere.

This software will compute any power of e that you may need.

Please enter a float number value for x: 17.55

Please enter the number of terms to include in theTaylor sum: 8

Thank you for waiting

The computed approximation of e^x is 209.71058313 with precision of 8 digits right of the point.

Thank you for using the Eyere exponential calculator.

Please return another day. The computed value will be set to the caller functions

The driver received this number 209.71058313 and will keep it.

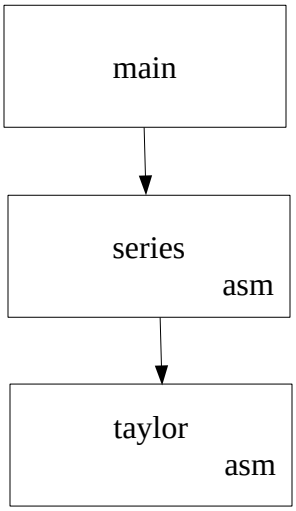
Good-bye

Footnotes. Replace Jane's name with your own name.

The numeric calculation is not correct. Its purpose is to show you the format of output.

The major exercise here is the block that computes e to the x power. That block involves a loop running from 0 to (n-1), which is n terms.

Calling diagram



Footnote. The prototype for taylor may be like this

double taylor(double x, unsigned long int n)

The series function is the person in charge. It obtains the x number from the user and it obtains the n integer from the user. The function series passes those two value to taylor, which computes the series number and sends it back to the function series.

Function taylor is cohesive. It has one solitary duty, which is to compute a single number and send it back. It does not input anything. It does not output anything. It does not compute two things. It computes one thing. It is highly cohesive. In fact, it is so good that a future programmer could replace it with a taylor function written in C++ very easily and nothing in the rest of the program would be affected. Great modularity!

Function taylor is separated out from commanding function series in order to make taylor into a library function. Taylor become reusable in many future programs.

Don't forget that library function are licensed by LGPL3.0 and not GPL3.0 The LGPL makes the software re-usable in future programs without disruption of the licensing already in place in that other program.

An entire application program, like Assignment 5, is of course licensed by GPL3.0 The one library function within Assignment 6 carries LGPL.

Sample execution with time measurement

Welcome to Taylor Series by Jane Eyere.

This software will compute any power of e that you may need.

Please enter a float number value for x: 17.55

Please enter the number of terms to include in the Taylor sum: 8

The time on the clock is now 125632771 tics and taylor has been called

Thank you for waiting

The time on the clock when Taylor terminated was 127301831 tics

The execution time was 1669060 tics.

The computed approximation of e^x is 209.71058313 with precision of 8 digits right of the point.

Thank you for using the Eyere exponential calculator.

Please return another day. The computed value will be set to the caller functions

The driver received this number 209.71058313 and will keep it.

Good-bye

Footnotes: The full version of assignment 5 expands to include measuring runtime on the cpu's clock.

One part was deleted from this specification, namely, convert the elapsed tics number to nanoseconds. That omission is intentional here. That is a math exercise. If you are uncomfortable practicing that conversion with the me, then I am sure the SI teachers will lead you through calculation and explain how it works.

Previously you learned how to extract a true random number from the cpu and here you learn how to extract the time from the cpu.

Conclusion

According to my count there remain six week of this semester counting the Thanksgiving relaxation week as one of those six weeks. Therefore, we'll split the difference, and give 3 weeks to Assignment 5 and 3 weeks to Assignment 6.

Due date November 26, 2024 at midnight. That is midway between Nov 26 and Nov 27.

The files in this program: main.cpp, series.asm, taylor.asm, and r.sh

Make your source files have a professional like appearance. The program will be worthy of a place in your portfolio. Keep building your portfolio until you obtain a career position you have always dreamed of.

On or before the due time, gather the four files named above and attach them as genuine attachments to a genuine email message. No links to cloud storage will be considered for more that one second. One second is the estimated time to delete an email containing a link to the clouds. If you don't know what this paragraph is saying then ask either of your Si instructors. Your points depend on getting this right. Don't come back later saying that I didn't know I could not send a link (URL) to my cloud.

No pdfs. No links to clouds.

In this entire semester do not ever send a pdf file for any reason. Pdfs are all deleted as soon as they are received and the owner is not informed that his submission was deleted without being read.

Pdf are great for storage of software engineering reports or design documents. Use pdfs in those other classes. But here they are deleted immediately.

Official Autumn reprieve: Nov 26-through Dec 1, 2024 copied from the official CSUF academic calendar.

Send to cpscgrader@proton.me

When you read this we are probably close to the winter break. Enjoy the feasts but stay healthy. Have a great break.

See you on December 2 or 3, 2024.

Peace from your professor.