Tu Lam

CS 472 / Justin Goins

April 11th, 2021

Homework #1

(Amicable Pair)

Question 1-4 were on about making sure the code that was written is compile and run for every test cases.

5. *Review the x86_64bit.asm file. How many lines of assembly instructions were utilized inside the* ***check_amicable*** *function?*

Answer: In total of the x86_64bit.asm file, there is a total of 56 lines in the code. But due to labels or directives we won't be counting those as assembly language. So, there is a total of 16 labels/directives. So, subtracting we get 56 – 16 = **40 lines** of assembly codes.

6. *Use the time utility (included as part of BASH) to determine the amount of CPU time spent in user mode to execute the x86_64bit.exe file. Round your answer to the nearest millisecond.*

Answer: When running the x86_64bit.exe file using the time utility, the time that was given is was 7.964 seconds (0:07.964 seconds).

Question 7 is a simple instruction to create an .asm file to run faster

8. *Review the x86_64bit_s.asm file. How many lines of assembly instructions were utilized inside the* ***check_amicable*** *function? How much more efficient (in terms of assembly instructions) is the optimized code?*

Answer: There is a total of 34 lines of code and 15 lines of directives/labels. So, the total assembly instruction is 34 – 15 = **19 lines** of assembly instructions.

9. *Use the same process as step 6 to measure the user mode CPU time of x86_64bit_s.exe (Which version of code executes faster? By how much?)*

Answer: This time with this file, the time came out to be 7.893 seconds (0:07.893 seconds). To compare the performance rate, this file run faster by 7.964 – 7.893 = **0.071 seconds'** differences.

Question 10 ask us to create another set of .asm and .exe file to see if it run a better performance to compare with the other files we created.

11. *As before, use the time utility (from BASH) to measure the user-mode CPU time of x86_32bit_3.exe and x86_64bit_3.exe (generated in step 10). Which version executes faster, and by how much?*

Answer: The file with x86_32bit_3.exe have a CPU time of **7.498 seconds** (0:07.498 seconds) while the x86_64bit_3.exe have a CPU time of **7.523 seconds** (0:07.523 seconds). In comparison to both files, the version that execute faster is the **x86_32bit_3.exe** file.

12. *x86 assembly commonly uses eight32-bit registers. They are: eax, ebx, ecx, edx, esi, edi, ebp, and esp. Which of these registers are used in the x86_32bit_3.asm file that was generated in step 10?*

Answer: In the x86_32bit_3.asm file, the file contains the registers of **edi**, **esi**, **ebx**, **esp**, **ecx**, **eax**, **and edx**.

Question 13 ask users to run and generate a MIPS assembly code.

14. *How many lines of MIPS assembly instructions were utilized to implement the* ***check_amicable*** *function? How does this compare to the number of instructions used to implement the same function in the x86_32bit_3.asm file (the 32-bit x86 code)?*

Answer: The MIPS file gave us a total of 67 lines of code – 44 directives/labels give us **20 lines of instruction codes.** Comparing to x86_32bit_3.asm file have 48 instruction codes. This means that MIPS uses less line of instructions than the x86 32bit .asm file.

15. *As noted in the textbook, MIPS has 32 32-bit registers (see page one of the MIPS reference card). Which of these registers are used in the assembly code generated in step 13?*

Answer: In the MIPS file, the registers that were spot seeing in the code is register **$29, $31, $3, $2, $0, $6, $4, $7, and $5.**

Question 16 ask use to implement a faster version with another C file

17. *Once you have ensured that your code is working correctly, compile an optimized 64-bit version (using the -O3 flag as demonstrated in step 10). Be sure to use values of 1982313333 and 1892277387 for num_a and num_b. Use the BASH time utility to measure the user-mode execution time and compare your results to the values from step 10. How much faster is version 2, compared to version 1?*

Answer: When creating the x86_64bit_v2.exe, the CPU time was given is **0.001 seconds** (0:00.001 seconds). Comparing it to the version 1 with 7.523 seconds (0:07.523 seconds, version 2 is faster by **7.522 seconds** from the first version.