TO: Professor Robert Antonetti

FROM: Arjun Aravind

DATE: February 26th, 2025

SUBJECT: Program 5 Observations

The purpose of Program 5 is to expand on the implementation of Program 4, a basic calculator program. Program 5 ensures that the input operands given by the user can be accepted in the format of decimal values, specifically to 2 decimal places. It utilizes the same calc.s and menu.s files as Program 4 and only modifies addOp.s and subOp.s. (Figures 1-4)

A screen shot of a computer

AI-generated content may be incorrect.

Figure 1 (calc.s)

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2 (menu.s)

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 3 (addOp.s)

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 4 (subOp.s)

Paired with each of these assembly files, there are 4 object files created to link the files together and ensure that the calc program can access the information inside of the other three programs. When creating the object files for calc.s and menu.s, the command “as -g -o \_\_\_\_.o \_\_\_\_\_.s” is used, where the blanks are filled in by calc and menu. This does not work for addOp.s or subOp.s, as these files utilize floating points and must be compiled uniquely through the command “as -mfpu=neon -march=armv7-a -o \_\_\_\_\_.o \_\_\_\_\_.s” which tells the assembler to use the ARMv7-A architecture and the NEON floating-point unit for operations.

Lastly, an executable file of the calc program is created to run the whole program. This is done by utilizing the command “gcc -g -o calc calc.o menu.o addOp.o subOp.o”, where each of the object files is called to link. Below is the final list of files generated to ensure proper connection links between the various program files and appropriate execution. (Figure 5)



Figure 5

The next step is to run the executable file based on the given test case example. This can be done by running the command “./calc” which calls upon the calc executable program and runs it. It is tested using numbers that each have 2 decimal values. (Figure 6)

A screenshot of a computer

AI-generated content may be incorrect.

Figure 6

Next, we analyze the information held in each register as the program executes by accessing the object dump file through the command “objdump calc -d”. This file shows all of the links and calls to functions and other files that occur throughout the execution of the calc program. (Figure 7)

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 7

*(Unfortunately, not all of the information is viewable on one screen, so the entire object dump is attached to the Index section)*

Lastly, to test the proper execution of the program in between steps, it is important to open the debugger window to insert an appropriate breakpoint and validate the steps that occur before it. The debugger window can be opened through the command “gdb calc”. From here, the command “b 30” is entered to create a breakpoint at line 30, and the command “r” is entered to run the program until line 30. This stops the program after allowing the user to select an operation in the menu and displays line 30, which calls the prompt\_choice to be loaded into the address right before it is printed. (Figure 8) This shows that the code runs smoothly until it reaches the breakpoint.

A computer screen shot of a black screen

AI-generated content may be incorrect.

Figure 8

**Index:**

calc: file format elf32-littlearm

Disassembly of section .init:

000102cc <\_init>:

102cc: e92d4008 push {r3, lr}

102d0: eb000023 bl 10364 <call\_weak\_fn>

102d4: e8bd8008 pop {r3, pc}

Disassembly of section .plt:

000102d8 <printf@plt-0x14>:

102d8: e52de004 push {lr} ; (str lr, [sp, #-4]!)

102dc: e59fe004 ldr lr, [pc, #4] ; 102e8 <\_init+0x1c>

102e0: e08fe00e add lr, pc, lr

102e4: e5bef008 ldr pc, [lr, #8]!

102e8: 00010474 .word 0x00010474

000102ec <printf@plt>:

102ec: e28fc600 add ip, pc, #0, 12

102f0: e28cca10 add ip, ip, #16, 20 ; 0x10000

102f4: e5bcf474 ldr pc, [ip, #1140]! ; 0x474

000102f8 <\_\_libc\_start\_main@plt>:

102f8: e28fc600 add ip, pc, #0, 12

102fc: e28cca10 add ip, ip, #16, 20 ; 0x10000

10300: e5bcf46c ldr pc, [ip, #1132]! ; 0x46c

00010304 <scanf@plt>:

10304: e28fc600 add ip, pc, #0, 12

10308: e28cca10 add ip, ip, #16, 20 ; 0x10000

1030c: e5bcf464 ldr pc, [ip, #1124]! ; 0x464

00010310 <\_\_gmon\_start\_\_@plt>:

10310: e28fc600 add ip, pc, #0, 12

10314: e28cca10 add ip, ip, #16, 20 ; 0x10000

10318: e5bcf45c ldr pc, [ip, #1116]! ; 0x45c

0001031c <abort@plt>:

1031c: e28fc600 add ip, pc, #0, 12

10320: e28cca10 add ip, ip, #16, 20 ; 0x10000

10324: e5bcf454 ldr pc, [ip, #1108]! ; 0x454

Disassembly of section .text:

00010328 <\_start>:

10328: e3a0b000 mov fp, #0

1032c: e3a0e000 mov lr, #0

10330: e49d1004 pop {r1} ; (ldr r1, [sp], #4)

10334: e1a0200d mov r2, sp

10338: e52d2004 push {r2} ; (str r2, [sp, #-4]!)

1033c: e52d0004 push {r0} ; (str r0, [sp, #-4]!)

10340: e59fc010 ldr ip, [pc, #16] ; 10358 <\_start+0x30>

10344: e52dc004 push {ip} ; (str ip, [sp, #-4]!)

10348: e59f000c ldr r0, [pc, #12] ; 1035c <\_start+0x34>

1034c: e59f300c ldr r3, [pc, #12] ; 10360 <\_start+0x38>

10350: ebffffe8 bl 102f8 <\_\_libc\_start\_main@plt>

10354: ebfffff0 bl 1031c <abort@plt>

10358: 0001064c .word 0x0001064c

1035c: 00010450 .word 0x00010450

10360: 000105e8 .word 0x000105e8

00010364 <call\_weak\_fn>:

10364: e59f3014 ldr r3, [pc, #20] ; 10380 <call\_weak\_fn+0x1c>

10368: e59f2014 ldr r2, [pc, #20] ; 10384 <call\_weak\_fn+0x20>

1036c: e08f3003 add r3, pc, r3

10370: e7932002 ldr r2, [r3, r2]

10374: e3520000 cmp r2, #0

10378: 012fff1e bxeq lr

1037c: eaffffe3 b 10310 <\_\_gmon\_start\_\_@plt>

10380: 000103e8 .word 0x000103e8

10384: 00000020 .word 0x00000020

00010388 <deregister\_tm\_clones>:

10388: e59f301c ldr r3, [pc, #28] ; 103ac <deregister\_tm\_clones+0x24>

1038c: e59f001c ldr r0, [pc, #28] ; 103b0 <deregister\_tm\_clones+0x28>

10390: e0603003 rsb r3, r0, r3

10394: e3530006 cmp r3, #6

10398: 912fff1e bxls lr

1039c: e59f3010 ldr r3, [pc, #16] ; 103b4 <deregister\_tm\_clones+0x2c>

103a0: e3530000 cmp r3, #0

103a4: 012fff1e bxeq lr

103a8: e12fff13 bx r3

103ac: 000208bb .word 0x000208bb

103b0: 000208b8 .word 0x000208b8

103b4: 00000000 .word 0x00000000

000103b8 <register\_tm\_clones>:

103b8: e59f1024 ldr r1, [pc, #36] ; 103e4 <register\_tm\_clones+0x2c>

103bc: e59f0024 ldr r0, [pc, #36] ; 103e8 <register\_tm\_clones+0x30>

103c0: e0601001 rsb r1, r0, r1

103c4: e1a01141 asr r1, r1, #2

103c8: e0811fa1 add r1, r1, r1, lsr #31

103cc: e1b010c1 asrs r1, r1, #1

103d0: 012fff1e bxeq lr

103d4: e59f3010 ldr r3, [pc, #16] ; 103ec <register\_tm\_clones+0x34>

103d8: e3530000 cmp r3, #0

103dc: 012fff1e bxeq lr

103e0: e12fff13 bx r3

103e4: 000208b8 .word 0x000208b8

103e8: 000208b8 .word 0x000208b8

103ec: 00000000 .word 0x00000000

000103f0 <\_\_do\_global\_dtors\_aux>:

103f0: e92d4010 push {r4, lr}

103f4: e59f4018 ldr r4, [pc, #24] ; 10414 <\_\_do\_global\_dtors\_aux+0x24>

103f8: e5d43000 ldrb r3, [r4]

103fc: e3530000 cmp r3, #0

10400: 18bd8010 popne {r4, pc}

10404: ebffffdf bl 10388 <deregister\_tm\_clones>

10408: e3a03001 mov r3, #1

1040c: e5c43000 strb r3, [r4]

10410: e8bd8010 pop {r4, pc}

10414: 000208b5 .word 0x000208b5

00010418 <frame\_dummy>:

10418: e92d4008 push {r3, lr}

1041c: e59f0024 ldr r0, [pc, #36] ; 10448 <frame\_dummy+0x30>

10420: e5903000 ldr r3, [r0]

10424: e3530000 cmp r3, #0

10428: 1a000001 bne 10434 <frame\_dummy+0x1c>

1042c: e8bd4008 pop {r3, lr}

10430: eaffffe0 b 103b8 <register\_tm\_clones>

10434: e59f3010 ldr r3, [pc, #16] ; 1044c <frame\_dummy+0x34>

10438: e3530000 cmp r3, #0

1043c: 0afffffa beq 1042c <frame\_dummy+0x14>

10440: e12fff33 blx r3

10444: eafffff8 b 1042c <frame\_dummy+0x14>

10448: 00020670 .word 0x00020670

1044c: 00000000 .word 0x00000000

00010450 <main>:

10450: e52de004 push {lr} ; (str lr, [sp, #-4]!)

00010454 <main\_loop>:

10454: eb000025 bl 104f0 <menu>

10458: e59f0078 ldr r0, [pc, #120] ; 104d8 <exit\_program+0x10>

1045c: ebffffa2 bl 102ec <printf@plt>

10460: e59f0074 ldr r0, [pc, #116] ; 104dc <exit\_program+0x14>

10464: e59f1074 ldr r1, [pc, #116] ; 104e0 <exit\_program+0x18>

10468: ebffffa5 bl 10304 <scanf@plt>

1046c: e1a02000 mov r2, r0

10470: e3520000 cmp r2, #0

10474: 0a00000a beq 104a4 <clear\_buffer>

10478: e59f0060 ldr r0, [pc, #96] ; 104e0 <exit\_program+0x18>

1047c: e5900000 ldr r0, [r0]

10480: e3500001 cmp r0, #1

10484: 0a00000b beq 104b8 <add>

10488: e3500002 cmp r0, #2

1048c: 0a00000b beq 104c0 <subtract>

10490: e3500005 cmp r0, #5

10494: 0a00000b beq 104c8 <exit\_program>

10498: e59f0044 ldr r0, [pc, #68] ; 104e4 <exit\_program+0x1c>

1049c: ebffff92 bl 102ec <printf@plt>

104a0: eaffffeb b 10454 <main\_loop>

000104a4 <clear\_buffer>:

104a4: e59f003c ldr r0, [pc, #60] ; 104e8 <exit\_program+0x20>

104a8: ebffff95 bl 10304 <scanf@plt>

104ac: e59f0030 ldr r0, [pc, #48] ; 104e4 <exit\_program+0x1c>

104b0: ebffff8d bl 102ec <printf@plt>

104b4: eaffffe6 b 10454 <main\_loop>

000104b8 <add>:

104b8: eb000012 bl 10508 <addOp>

104bc: eaffffe4 b 10454 <main\_loop>

000104c0 <subtract>:

104c0: eb00002c bl 10578 <subOp>

104c4: eaffffe2 b 10454 <main\_loop>

000104c8 <exit\_program>:

104c8: e59f001c ldr r0, [pc, #28] ; 104ec <exit\_program+0x24>

104cc: ebffff86 bl 102ec <printf@plt>

104d0: e49de004 pop {lr} ; (ldr lr, [sp], #4)

104d4: e12fff1e bx lr

104d8: 00020790 .word 0x00020790

104dc: 000207d0 .word 0x000207d0

104e0: 000208b6 .word 0x000208b6

104e4: 000207b4 .word 0x000207b4

104e8: 000207d3 .word 0x000207d3

104ec: 000207a1 .word 0x000207a1

000104f0 <menu>:

104f0: e52de004 push {lr} ; (str lr, [sp, #-4]!)

104f4: e59f0008 ldr r0, [pc, #8] ; 10504 <menu+0x14>

104f8: ebffff7b bl 102ec <printf@plt>

104fc: e49de004 pop {lr} ; (ldr lr, [sp], #4)

10500: e12fff1e bx lr

10504: 000207e0 .word 0x000207e0

00010508 <addOp>:

10508: e52de004 push {lr} ; (str lr, [sp, #-4]!)

1050c: e24dd018 sub sp, sp, #24

10510: e59f0050 ldr r0, [pc, #80] ; 10568 <addOp+0x60>

10514: ebffff74 bl 102ec <printf@plt>

10518: e59f004c ldr r0, [pc, #76] ; 1056c <addOp+0x64>

1051c: e1a0100d mov r1, sp

10520: ebffff77 bl 10304 <scanf@plt>

10524: e59f0044 ldr r0, [pc, #68] ; 10570 <addOp+0x68>

10528: ebffff6f bl 102ec <printf@plt>

1052c: e59f0038 ldr r0, [pc, #56] ; 1056c <addOp+0x64>

10530: e28d1008 add r1, sp, #8

10534: ebffff72 bl 10304 <scanf@plt>

10538: ed9d0b00 vldr d0, [sp]

1053c: ed9d1b02 vldr d1, [sp, #8]

10540: ee302b01 vadd.f64 d2, d0, d1

10544: ed8d2b04 vstr d2, [sp, #16]

10548: e59f0024 ldr r0, [pc, #36] ; 10574 <addOp+0x6c>

1054c: ed9d0b00 vldr d0, [sp]

10550: ed9d1b02 vldr d1, [sp, #8]

10554: ed9d2b04 vldr d2, [sp, #16]

10558: ebffff63 bl 102ec <printf@plt>

1055c: e28dd018 add sp, sp, #24

10560: e49de004 pop {lr} ; (ldr lr, [sp], #4)

10564: e12fff1e bx lr

10568: 00020827 .word 0x00020827

1056c: 0002084d .word 0x0002084d

10570: 0002083a .word 0x0002083a

10574: 00020851 .word 0x00020851

00010578 <subOp>:

10578: e52de004 push {lr} ; (str lr, [sp, #-4]!)

1057c: e24dd018 sub sp, sp, #24

10580: e59f0050 ldr r0, [pc, #80] ; 105d8 <subOp+0x60>

10584: ebffff58 bl 102ec <printf@plt>

10588: e59f004c ldr r0, [pc, #76] ; 105dc <subOp+0x64>

1058c: e1a0100d mov r1, sp

10590: ebffff5b bl 10304 <scanf@plt>

10594: e59f0044 ldr r0, [pc, #68] ; 105e0 <subOp+0x68>

10598: ebffff53 bl 102ec <printf@plt>

1059c: e59f0038 ldr r0, [pc, #56] ; 105dc <subOp+0x64>

105a0: e28d1008 add r1, sp, #8

105a4: ebffff56 bl 10304 <scanf@plt>

105a8: ed9d0b00 vldr d0, [sp]

105ac: ed9d1b02 vldr d1, [sp, #8]

105b0: ee302b41 vsub.f64 d2, d0, d1

105b4: ed8d2b04 vstr d2, [sp, #16]

105b8: e59f0024 ldr r0, [pc, #36] ; 105e4 <subOp+0x6c>

105bc: ed9d0b00 vldr d0, [sp]

105c0: ed9d1b02 vldr d1, [sp, #8]

105c4: ed9d2b04 vldr d2, [sp, #16]

105c8: ebffff47 bl 102ec <printf@plt>

105cc: e28dd018 add sp, sp, #24

105d0: e49de004 pop {lr} ; (ldr lr, [sp], #4)

105d4: e12fff1e bx lr

105d8: 0002086e .word 0x0002086e

105dc: 00020894 .word 0x00020894

105e0: 00020881 .word 0x00020881

105e4: 00020898 .word 0x00020898

000105e8 <\_\_libc\_csu\_init>:

105e8: e92d43f8 push {r3, r4, r5, r6, r7, r8, r9, lr}

105ec: e1a07000 mov r7, r0

105f0: e59f604c ldr r6, [pc, #76] ; 10644 <\_\_libc\_csu\_init+0x5c>

105f4: e59f504c ldr r5, [pc, #76] ; 10648 <\_\_libc\_csu\_init+0x60>

105f8: e08f6006 add r6, pc, r6

105fc: e08f5005 add r5, pc, r5

10600: e0656006 rsb r6, r5, r6

10604: e1a08001 mov r8, r1

10608: e1a09002 mov r9, r2

1060c: ebffff2e bl 102cc <\_init>

10610: e1b06146 asrs r6, r6, #2

10614: 08bd83f8 popeq {r3, r4, r5, r6, r7, r8, r9, pc}

10618: e2455004 sub r5, r5, #4

1061c: e3a04000 mov r4, #0

10620: e2844001 add r4, r4, #1

10624: e5b53004 ldr r3, [r5, #4]!

10628: e1a00007 mov r0, r7

1062c: e1a01008 mov r1, r8

10630: e1a02009 mov r2, r9

10634: e12fff33 blx r3

10638: e1540006 cmp r4, r6

1063c: 1afffff7 bne 10620 <\_\_libc\_csu\_init+0x38>

10640: e8bd83f8 pop {r3, r4, r5, r6, r7, r8, r9, pc}

10644: 0001006c .word 0x0001006c

10648: 00010064 .word 0x00010064

0001064c <\_\_libc\_csu\_fini>:

1064c: e12fff1e bx lr

Disassembly of section .fini:

00010650 <\_fini>:

10650: e92d4008 push {r3, lr}

10654: e8bd8008 pop {r3, pc}