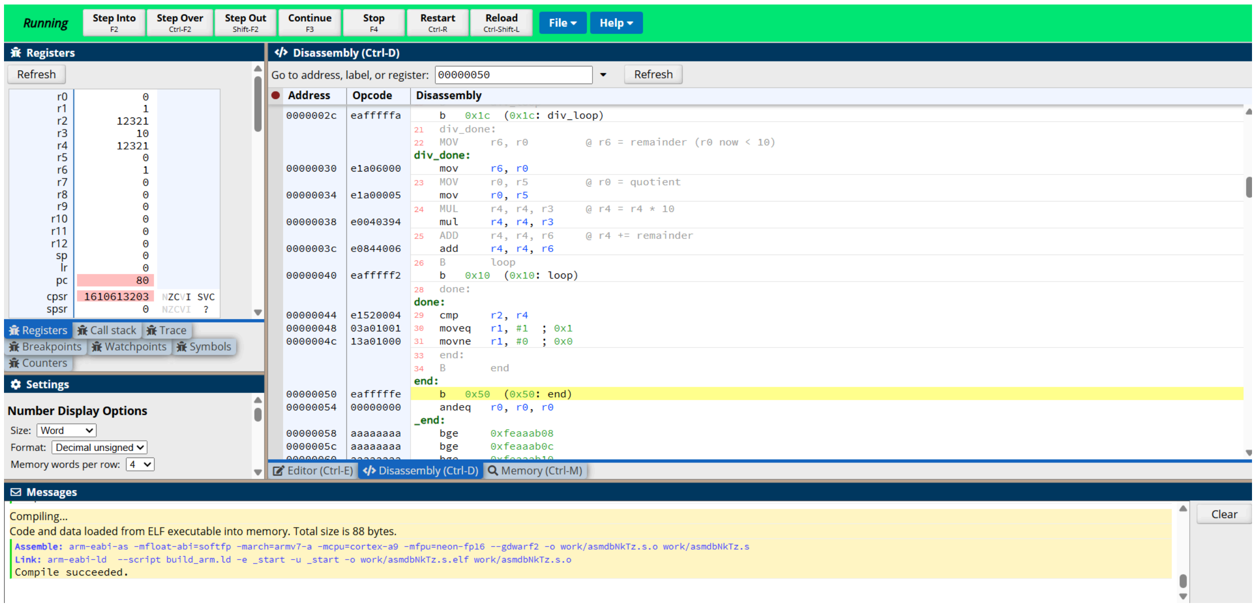
Siddarth Pai

RTOS Implementation of Human Vitals Monitor - Assignment 1

1.

```
.global _start

_start:
   MOV     r0, #12321      @ Number
   MOV     r2, r0        @ Copy
   MOV     r4, #0        @ Reversed number
   MOV     r3, #10        @ Divisor

loop:
   CMP     r0, #0        @ for division by 10 to continue number shldnt be equal to 0 ==
while (num>0)
   BEQ     done
   MOV     r5, #0        @ quotient = 0

div_loop:
   CMP     r0, r3
   BLT     div_done
   SUB     r0, r0, r3     @ r0-=10
   ADD     r5, r5, #1     @ quotient+=1
   B       div_loop

div_done:
   MOV     r6, r0        @ r6 = remainder (r0 now < 10)
   MOV     r0, r5        @ r0 = quotient
   MUL     r4, r4, r3     @ r4 = r4 * 10
   ADD     r4, r4, r6     @ r4 += remainder
   B       loop

done:
   CMP     r2, r4
   MOVEQ   r1, #1
   MOVNE   r1, #0

end:
   B       end
```

Siddarth Pai



```
r0            0
r1            1
r2        12321
r3           10
r4        12321
r5            0
r6            1
r7            0
r8            0
r9            0
r10           0
r11           0
r12           0
sp            0
lr            0
pc           80
cpsr 1610613203  NZCVI SVC
spsr          0  NZCVI  ?
```

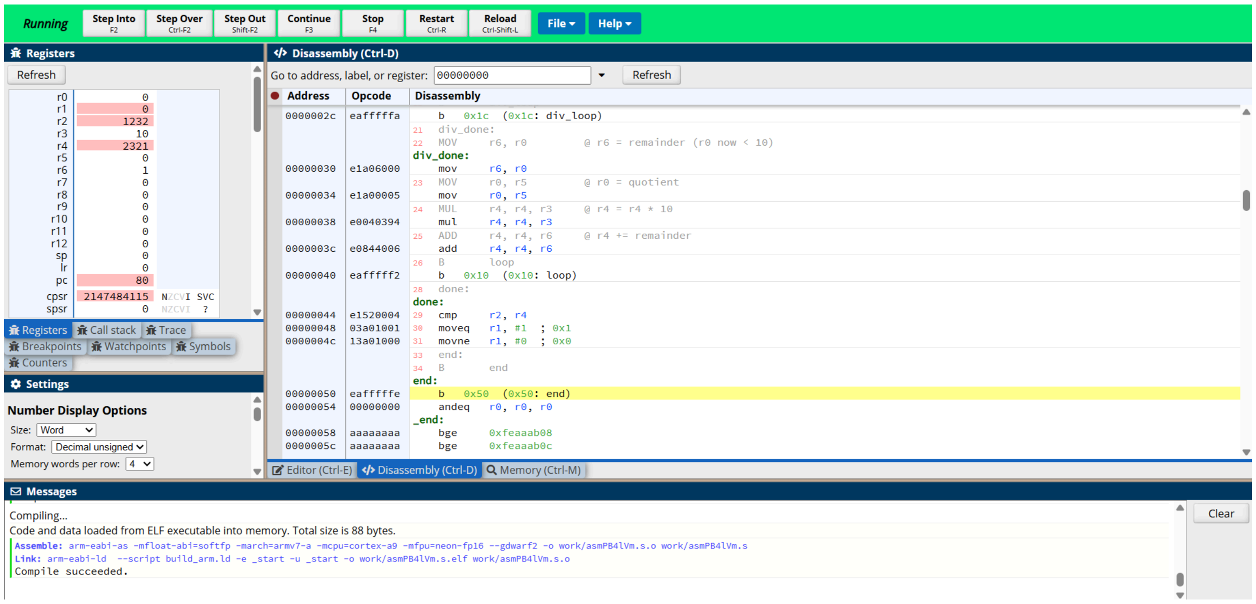
Number Display Options

Size: Word
Format: Decimal unsigned
Memory words per row: 4

Go to address, label, or register: 00000050

```
Address   Opcode    Disassembly
0000002c  eaffffa       b    0x1c  (0x1c: div_loop)
                    div_done:
                    MOV      r6, r0         @ r6 = remainder (r0 now < 10)
                    div_done:
00000030  e1a06000      mov      r6, r0
                    MOV      r0, r5         @ r0 = quotient
00000034  e1a00005      mov      r0, r5
                    MUL      r4, r4, r3     @ r4 = r4 * 10
00000038  e0040394      mul      r4, r4, r3
                    ADD      r4, r4, r6     @ r4 += remainder
0000003c  e0844006      add      r4, r4, r6
                    B        loop
00000040  eafffff2      b    0x10  (0x10: loop)
                    done:
                    done:
00000044  e1520004  cmp      r2, r4
00000048  03a01001  moveq    r1, #1   ; 0x1
0000004c  13a01000  movne    r1, #0   ; 0x0
                    end:
                    B        end
                    end:
00000050  eafffffe      b    0x50  (0x50: end)
00000054  00000000      andeq    r0, r0, r0
                    _end:
00000058  aaaaaaaa      bge      0xfeaaab08
0000005c  aaaaaaaa      bge      0xfeaaab0c
```

```
Compiling...
Code and data loaded from ELF executable into memory. Total size is 88 bytes.
Assemble: arm-eabi-as -mfloat-abi=softfp -march=armv7-a -mcpu=cortex-a9 -mfpu=neon-fp16 --gdwarf2 -o work/asmdbNkTz.s.o work/asmdbNkTz.s
Link: arm-eabi-ld  --script build_arm.ld -e _start -u _start -o work/asmdbNkTz.s.elf work/asmdbNkTz.s.o
Compile succeeded.
```



```
r0            0
r1            0
r2         1232
r3           10
r4         2321
r5            0
r6            1
r7            0
r8            0
r9            0
r10           0
r11           0
r12           0
sp            0
lr            0
pc           80
cpsr 2147484115  NZCVI SVC
spsr          0  NZCVI  ?
```

Number Display Options

Size: Word
Format: Decimal unsigned
Memory words per row: 4

Go to address, label, or register: 00000000

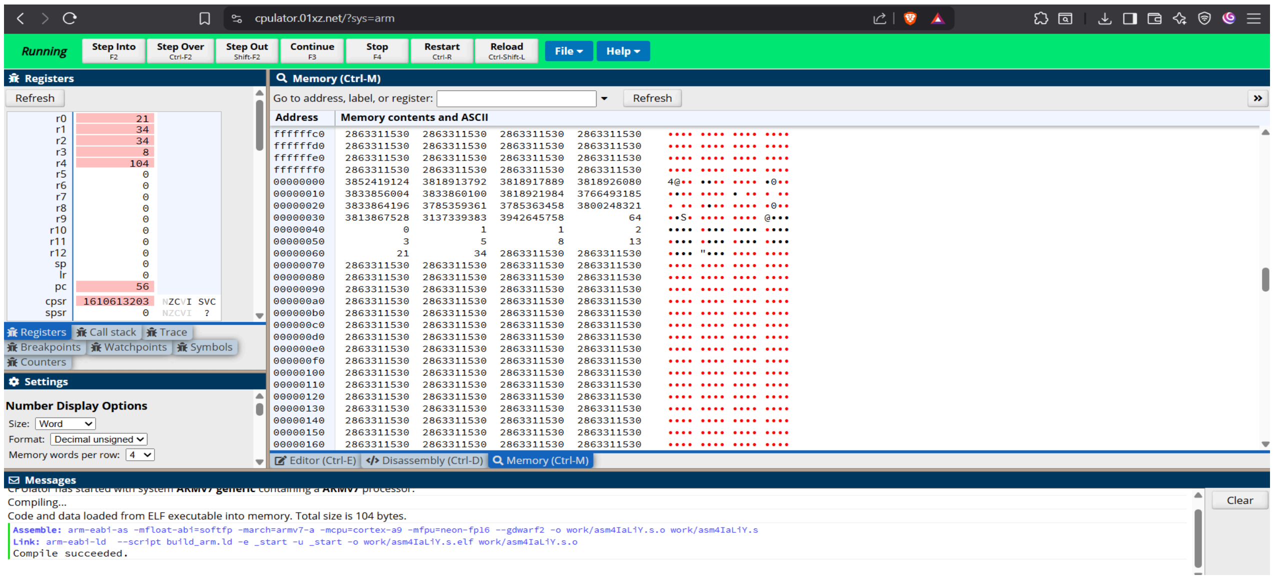
```
Address   Opcode    Disassembly
0000002c  eaffffa       b    0x1c  (0x1c: div_loop)
                    div_done:
                    MOV      r6, r0         @ r6 = remainder (r0 now < 10)
                    div_done:
00000030  e1a06000      mov      r6, r0
                    MOV      r0, r5         @ r0 = quotient
00000034  e1a00005      mov      r0, r5
                    MUL      r4, r4, r3     @ r4 = r4 * 10
00000038  e0040394      mul      r4, r4, r3
                    ADD      r4, r4, r6     @ r4 += remainder
0000003c  e0844006      add      r4, r4, r6
                    B        loop
00000040  eafffff2      b    0x10  (0x10: loop)
                    done:
                    done:
00000044  e1520004  cmp      r2, r4
00000048  03a01001  moveq    r1, #1   ; 0x1
0000004c  13a01000  movne    r1, #0   ; 0x0
                    end:
                    B        end
                    end:
00000050  eafffffe      b    0x50  (0x50: end)
00000054  00000000      andeq    r0, r0, r0
                    _end:
00000058  aaaaaaaa      bge      0xfeaaab08
0000005c  aaaaaaaa      bge      0xfeaaab0c
```

```
Compiling...
Code and data loaded from ELF executable into memory. Total size is 88 bytes.
Assemble: arm-eabi-as -mfloat-abi=softfp -march=armv7-a -mcpu=cortex-a9 -mfpu=neon-fp16 --gdwarf2 -o work/asmPB4lVm.s.o work/asmPB4lVm.s
Link: arm-eabi-ld  --script build_arm.ld -e _start -u _start -o work/asmPB4lVm.s.elf work/asmPB4lVm.s.o
Compile succeeded.
```

Siddarth Pai

2.

```
.data
memory:  .space 40
.text
.global _start
_start:
        LDR r4,=memory
        MOV r0,#0
        MOV r1,#1
        MOV r3,#0
        STR r0, [r4],#4
        STR r1, [r4],#4

loop:
        MOV r2,#0
        ADD r2,r0,r1
        STR r2,[r4],#4
        MOV r0,r1
        MOV r1,r2
        ADD r3,#1
        CMP r3,#8
        BLT loop

end:
        B end
```



```
  r0          21
  r1          34
  r2          34
  r3           8
  r4         104
  r5           0
  r6           0
  r7           0
  r8           0
  r9           0
 r10           0
 r11           0
 r12           0
  sp           0
  lr           0
  pc          56
cpsr  1610613203  NZCVI SVC
spsr           0  NZCVI  ?
```

Number Display Options
Size: Word
Format: Decimal unsigned
Memory words per row: 4

Go to address, label, or register:

```
Address   Memory contents and ASCII
ffffffc0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
ffffffd0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
ffffffe0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
fffffff0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000000  3852419124  3818913792  3818917889  3818926080   4@•• ••••• •••• •0••
00000010  3833856004  3833860100  3818921984  3766493185   •••• •••• • •• • ••
00000020  3833864196  3785359361  3785363458  3800248321   • •• •••• •••• •0••
00000030  3813867528  3137339383  3942645758          64   ••S• •••• •••• @•••
00000040           0           1           1           2   •••• •••• •••• ••••
00000050           3           5           8          13   •••• •••• •••• ••••
00000060          21          34  2863311530  2863311530   •••• "••• •••• ••••
00000070  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000080  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000090  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
000000a0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
000000b0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
000000c0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
000000d0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
000000e0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
000000f0  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000100  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000110  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000120  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000130  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000140  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000150  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
00000160  2863311530  2863311530  2863311530  2863311530   •••• •••• •••• ••••
```

```
CPUlator has started with system ARMv7 generic containing a ARMv7 processor.
Compiling...
Code and data loaded from ELF executable into memory. Total size is 104 bytes.
Assemble: arm-eabi-as -mfloat-abi=softfp -march=armv7-a -mcpu=cortex-a9 -mfpu=neon-fp16 --gdwarf2 -o work/asm4IaLiY.s.o work/asm4IaLiY.s
Link: arm-eabi-ld  --script build_arm.ld -e _start -u _start -o work/asm4IaLiY.s.elf work/asm4IaLiY.s.o
Compile succeeded.
```