CS 301 Lab 9

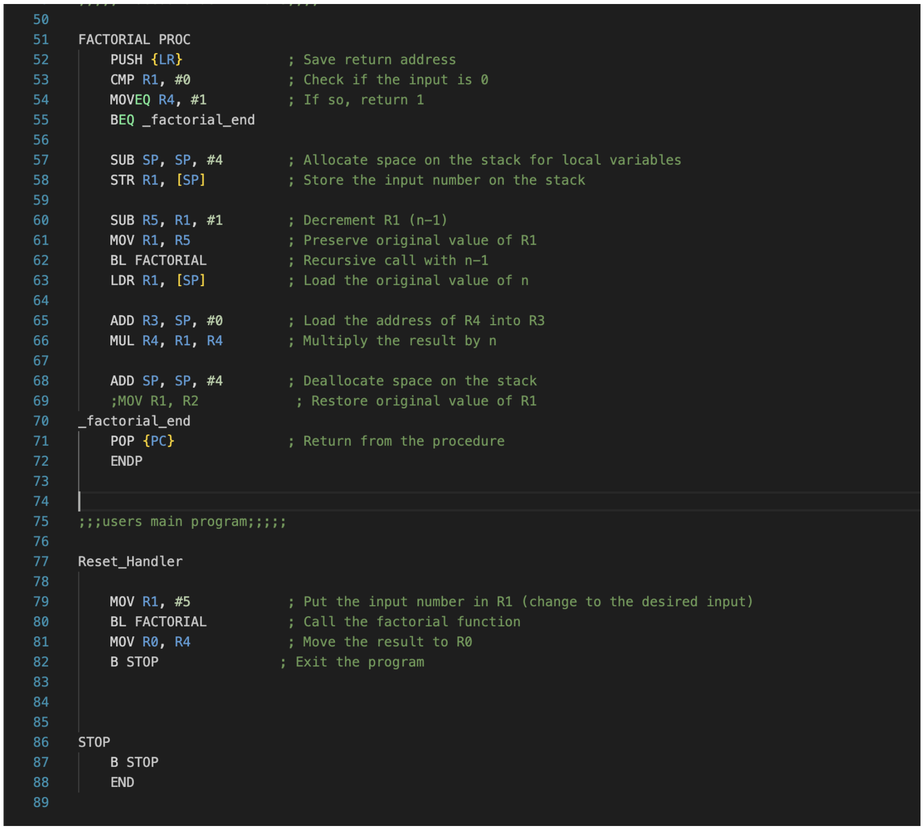
Factorial.s

Student ID: 200482797

Owen Monus

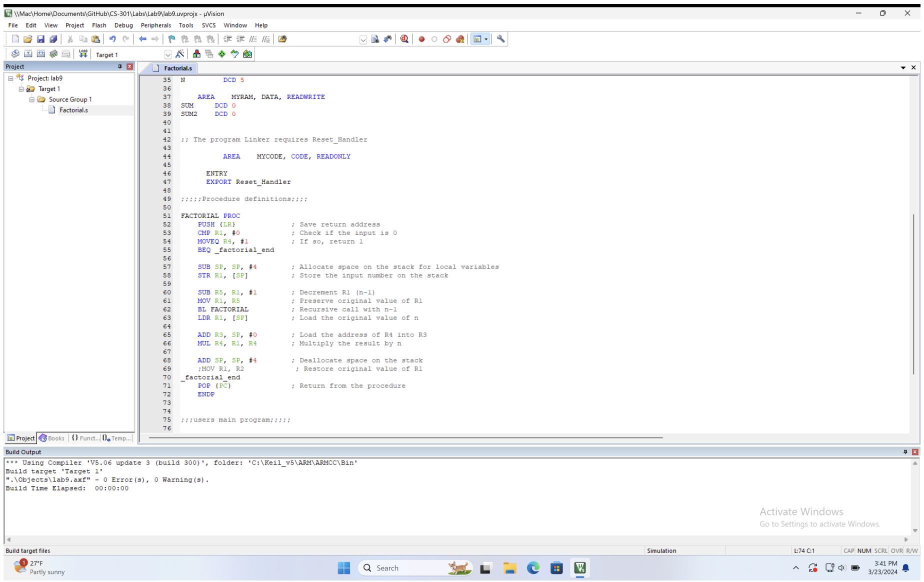
March 23rd, 2024

Source code



```
FACTORIAL PROC
    PUSH {LR}             ; Save return address
    CMP R1, #0            ; Check if the input is 0
    MOVEQ R4, #1          ; If so, return 1
    BEQ _factorial_end

    SUB SP, SP, #4        ; Allocate space on the stack for local variables
    STR R1, [SP]          ; Store the input number on the stack

    SUB R5, R1, #1        ; Decrement R1 (n-1)
    MOV R1, R5            ; Preserve original value of R1
    BL FACTORIAL          ; Recursive call with n-1
    LDR R1, [SP]          ; Load the original value of n

    ADD R3, SP, #0        ; Load the address of R4 into R3
    MUL R4, R1, R4        ; Multiply the result by n

    ADD SP, SP, #4        ; Deallocate space on the stack
    ;MOV R1, R2            ; Restore original value of R1
_factorial_end
    POP {PC}              ; Return from the procedure
    ENDP


;;;users main program;;;;;

Reset_Handler

    MOV R1, #5            ; Put the input number in R1 (change to the desired input)
    BL FACTORIAL          ; Call the factorial function
    MOV R0, R4            ; Move the result to R0
    B STOP               ; Exit the program



STOP
    B STOP
    END
```

Successful build



\\Mac\Home\Documents\GitHub\CS-301\Labs\Lab9\lab9.uvprojx - µVision

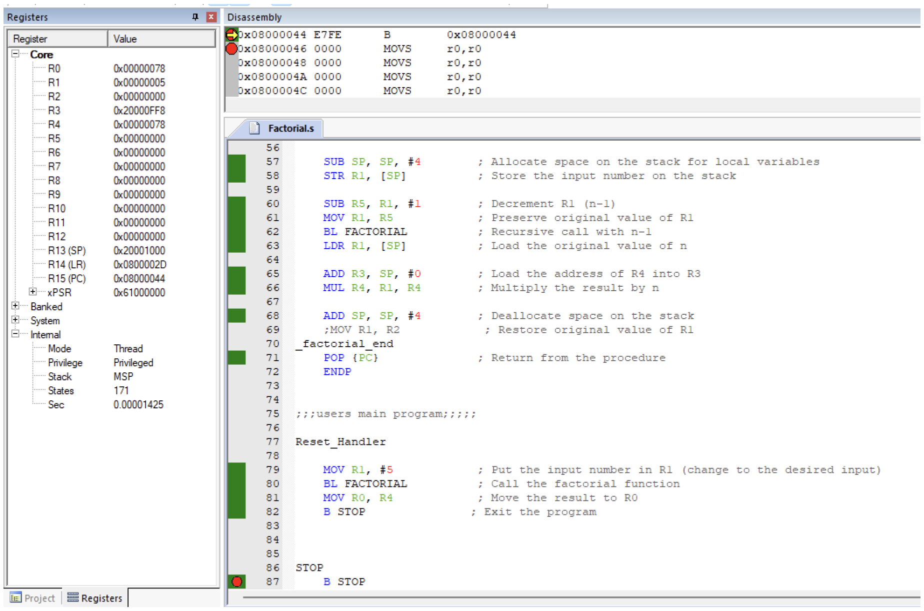
Factorial.s

```
N          DCD 5

    AREA    MYRAM, DATA, READWRITE
SUM     DCD 0
SUM2    DCD 0


;; The program Linker requires Reset_Handler

          AREA    MYCODE, CODE, READONLY

       ENTRY
       EXPORT Reset_Handler

;;;;;Procedure definitions;;;;

FACTORIAL PROC
    PUSH {LR}               ; Save return address
    CMP R1, #0              ; Check if the input is 0
    MOVEQ R4, #1            ; If so, return 1
    BEQ _factorial_end

    SUB SP, SP, #4          ; Allocate space on the stack for local variables
    STR R1, [SP]            ; Store the input number on the stack

    SUB R5, R1, #1          ; Decrement R1 (n-1)
    MOV R1, R5              ; Preserve original value of R1
    BL FACTORIAL            ; Recursive call with n-1
    LDR R1, [SP]            ; Load the original value of n

    ADD R3, SP, #0          ; Load the address of R4 into R3
    MUL R4, R1, R4          ; Multiply the result by n

    ADD SP, SP, #4          ; Deallocate space on the stack
    ;MOV R1, R2              ; Restore original value of R1
_factorial_end
    POP {PC}                ; Return from the procedure
    ENDP


;;;users main program;;;;;

```

Build Output

```
*** Using Compiler 'V5.06 update 3 (build 300)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
".\Objects\lab9.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

Registers post run



Registers

Register	Value
Core	
R0	0x00000078
R1	0x00000005
R2	0x00000000
R3	0x20000FF8
R4	0x00000078
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20001000
R14 (LR)	0x0800002D
R15 (PC)	0x08000044
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	171
Sec	0.00001425

Disassembly

```
0x08000044 E7FE      B         0x08000044
0x08000046 0000      MOVS      r0,r0
0x08000048 0000      MOVS      r0,r0
0x0800004A 0000      MOVS      r0,r0
0x0800004C 0000      MOVS      r0,r0
```

Factorial.s

```

    SUB SP, SP, #4          ; Allocate space on the stack for local variables
    STR R1, [SP]            ; Store the input number on the stack

    SUB R5, R1, #1          ; Decrement R1 (n-1)
    MOV R1, R5              ; Preserve original value of R1
    BL FACTORIAL            ; Recursive call with n-1
    LDR R1, [SP]            ; Load the original value of n

    ADD R3, SP, #0          ; Load the address of R4 into R3
    MUL R4, R1, R4          ; Multiply the result by n

    ADD SP, SP, #4          ; Deallocate space on the stack
    ;MOV R1, R2              ; Restore original value of R1
_factorial_end
    POP {PC}                ; Return from the procedure
    ENDP


;;;users main program;;;;;

Reset_Handler

    MOV R1, #5              ; Put the input number in R1 (change to the desired input)
    BL FACTORIAL            ; Call the factorial function
    MOV R0, R4              ; Move the result to R0
    B STOP                 ; Exit the program



STOP
    B STOP
```