

CPSC 240: Computer Organization and Assembly Language

Assignment 04, Fall Semester 2024

CWID: 885024539 Name: Riley Blacklock

Programming:

1. Download the "CPSC-240 Assignment04.docx" document.
2. Design "multiple.asm" program to implement an if-else structure in assembly language, and use assembly language to realize the function of the following C++ instructions. **NOTE: variable sizes and program functions should be equivalent to C/C++ instructions.**

```
unsigned short num = 65535;           // use dw to declare 16-bit variable
unsigned short mul_3 = 0, other = 0;   // use dw to declare 16-bit variable
if(num % 3 == 0 && num % 5 != 0) {
    mul_3++;
} else {
    other++;
}
```
3. Assemble the "multiple.asm" file and link the "multiple.o" file to get the "multiple" executable file.
4. Run the "multiple" file with the GDB debugger to display the memory of **num**, as well as the simulation results of **mul_3** and **other**.
5. Insert source code (multiple.asm) and simulation results (GDB window) of the memory (num, mul_3, and other) in the document. Write an analysis to verify simulation results.
6. Save the file in pdf format and submit the pdf file to Canvas before the deadline.

[Insert multiple.asm source code here]

```
; multiple.asm;
; unsigned short num = 65535;
; unsigned short mul_3 = 0, other = 0;
; if(num % 3 == 0 && num % 5 != 0) { mul_3++; } else { other++; };
```

```
section .data
SYS_exit equ 60
EXIT_SUCCESS equ 0
num      dw 65535
mul_3    dw 0
other    dw 0
```

```
section .text
    global _start
_start:
```

```

mov ax, [num]
mov bx, 3
xor dx, dx
div bx
cmp dx, 0
jne increment_other
mov ax, [num]
mov bx, 5
xor dx, dx
div bx
cmp dx, 0
je increment_other
mov ax, [mul_3]
inc ax
mov [mul_3], ax
jmp exit

```

increment_other:

```

mov ax, [other]
inc ax
mov [other], ax

```

exit:

```

mov rax, SYS_exit
mov rdi, EXIT_SUCCESS
syscall

```

[Insert multiple simulation result here]

```

(gdb) x/uh &num      num = 65535
0x402000: 65535
(gdb) x/uh &mul_3     mul_3 = 0
0x402002: 0
(gdb) x/uh &other     other = 1
0x402004: 1
(gdb) l

```

[Insert the simulation result verification here]

```

3  int main(int argc, char *argv[]) {
4      unsigned short num = 65535;
5      unsigned short mul_3 = 0;
6      unsigned short other = 0;
7
8      if ((num % 3 == 0) && (num % 5 != 0)) {
9          mul_3++;
10     } else {
11         other++;
12     }
13     printf("num: %d, mul_3: %d, other: %d\n", num, mul_3, other);
14     return 0;
15 }

```

num = 65535, mul_3: 0, other: 1

num = 65535

mul_3 = 0

other = 1