Lab 5 Report

Huy Quang Lai
132000359

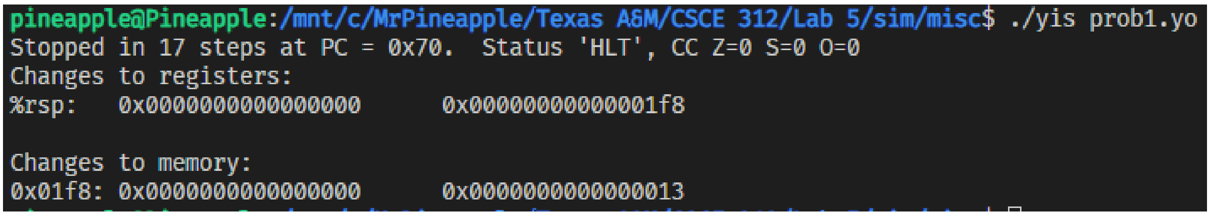
Texas A&M University

8 November 2022

An Aggie does not lie, cheat or steal.
Nor does an Aggie tolerate those who do.

Problem 1

```
irmovq $0x200, %rsp
call main

main:
    pushq %rbx
    pushq %rcx
    irmovq $0x0, %rbx
    irmovq $0x0, %rcx
    pushq %rdx
    rrmovq %rcx, %rdx
    subq %rbx, %rdx
    popq %rdx
    jle else
    iaddq $0x5, %rbx
    jmp end
else:
    irmovq $0x0, %rbx
    iaddq $0x1, %rcx
    jmp end
end:
    popq %rcx
    popq %rbx
    halt
```

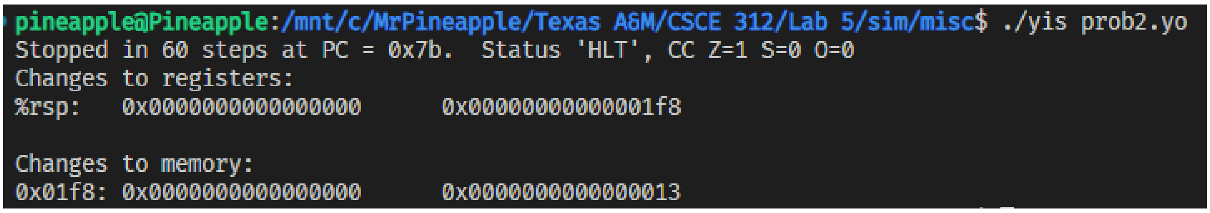


```
pineapple@Pineapple:/mnt/c/MrPineapple/Texas A&M/CSCE 312/Lab 5/sim/misc$ ./yis prob1.yo
Stopped in 17 steps at PC = 0x70.  Status 'HLT', CC Z=0 S=0 O=0
Changes to registers:
%rsp:   0x0000000000000000      0x00000000000001f8

Changes to memory:
0x01f8: 0x0000000000000000      0x0000000000000013
```

Figure 1: Problem 1 Output

Problem 2

```
# Problem 2
irmovq $0x200, %rsp    # Set up stack pointer
call main              # Execute main program

main:
    pushq %rbx           # j
    pushq %rcx           # k
    pushq %rdx           # i
    pushq %rsi           # comp
    irmovq $0x0, %rbx
    irmovq $0x0, %rcx
    irmovq $0x0, %rdx
    jmp loop
loop:
    rrmovq %rdx, %rsi
    addq %rsi, %rsi
    rrmovq %rsi, %rbx
    irmovq $0x1, %rcx
    addq %rbx, %rcx
    irmovq $0x5, %rsi
    iaddq $0x1, %rdx
    subq %rdx, %rsi
    jg loop
end:
    popq %rsi
    popq %rdx
    popq %rcx
    popq %rbx
    halt
```



```
pineapple@Pineapple:/mnt/c/MrPineapple/Texas A&M/CSCE 312/Lab 5/sim/misc$ ./yis prob2.yo
Stopped in 60 steps at PC = 0x7b.  Status 'HLT', CC Z=1 S=0 O=0
Changes to registers:
%rsp:   0x0000000000000000      0x00000000000001f8

Changes to memory:
0x01f8: 0x0000000000000000      0x0000000000000013
```

Figure 2: Problem 2

Problem 3

Part 1

```
.file "lab5_prob3_1.c"
.text
.section .rodata
.LC0:
.string "Hello, world"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi_startproc
endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $16, %rsp
movl %edi, -4(%rbp)
movq %rsi, -16(%rbp)
leaq .LC0(%rip), %rdi
call puts@PLT
movl $0, %eax
leave
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE0:
.size main, .-main
.ident "GCC: (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0"
.section .note.GNU-stack,"",@progbits
.section .note.gnu.property,"a"
.align 8
.long  1f - 0f
.long  4f - 1f
.long  5
0:
.string  "GNU"
1:
.align 8
.long  0xc0000002
```

```
.long  3f - 2f
2:
.long  0x3
3:
.align 8
4:
```

Part 2

```
.file "lab5_prob3_2.c"
.text
.section .rodata
.LC0:
.string "The value of  i is %d\n"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi_startproc
endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $32, %rsp
movl %edi, -20(%rbp)
movq %rsi, -32(%rbp)
movl $1, -4(%rbp)
addl $1, -4(%rbp)
movl -4(%rbp), %eax
movl %eax, %esi
leaq .LC0(%rip), %rdi
movl $0, %eax
call printf@PLT
movl $0, %eax
leave
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE0:
.size main, .-main
.ident "GCC: (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0"
```

```
.section .note.GNU-stack,"",@progbits
.section .note.gnu.property,"a"
.align 8
.long  1f - 0f
.long  4f - 1f
.long  5
0:
.string  "GNU"
1:
.align 8
.long  0xc0000002
.long  3f - 2f
2:
.long  0x3
3:
.align 8
4:
```

Since the string in the second block of code is longer, more memory needs to be allocated for it. Additionally, `i` also needs to be stored. This explains the difference in `line 18`

The other differences can be explained by the fact that the second code block is performing arithmetic on `i`. Additionally, formatting the string also requires code. This is done on `line 21` through `line 25`. This is not present in the first block of code.

Problem 4

```
.file "lab5_prob4.c"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi_startproc
endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $16, %rsp
movl %edi, -4(%rbp)
movq %rsi, -16(%rbp)
movl $0, %eax
call print_hello
movl $0, %eax
leave
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE0:
.size main, .-main
.section .rodata
.LC0:
.string "Hello, world"
.text
.globl print_hello
.type print_hello, @function
print_hello:
.LFB1:
.cfi_startproc
endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
leaq .LC0(%rip), %rdi
call puts@PLT
nop
```

```
popq %rbp
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE1:
.size print_hello, .-print_hello
.ident "GCC: (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0"
.section .note.GNU-stack,"",@progbits
.section .note.gnu.property,"a"
.align 8
.long  1f - 0f
.long  4f - 1f
.long  5
0:
.string  "GNU"
1:
.align 8
.long  0xc0000002
.long  3f - 2f
2:
.long  0x3
3:
.align 8
4:
```

The function `print_hello` needs to be compiled. As such an additional label is created for it. Furthermore, since this function needs to be called, `line 21` sets up `%eax` as to return 0 as the end of the function.

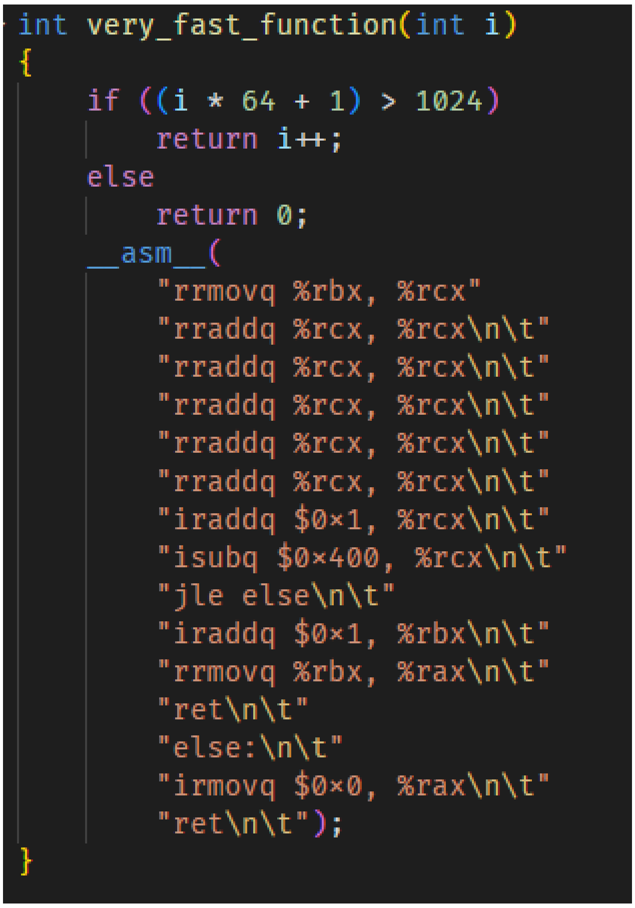
Problem 5

```
.file "lab5_prob5_main.c"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi_startproc
endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $16, %rsp
movl %edi, -4(%rbp)
movq %rsi, -16(%rbp)
movl $0, %eax
call print_hello@PLT
movl $0, %eax
leave
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE0:
.size main, .-main
.ident "GCC: (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0"
.section .note.GNU-stack,"",@progbits
.section .note.gnu.property,"a"
.align 8
.long  1f - 0f
.long  4f - 1f
.long  5
0:
.string  "GNU"
1:
.align 8
.long  0xc0000002
.long  3f - 2f
2:
.long  0x3
3:
.align 8
4:
```

```
.file "lab5_prob5_print.c"
.text
.section .rodata
.LC0:
.string "Hello, world"
.text
.globl print_hello
.type print_hello, @function
print_hello:
.LFB0:
.cfi_startproc
endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
leaq .LC0(%rip), %rdi
call puts@PLT
nop
popq %rbp
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE0:
.size print_hello, .-print_hello
.ident "GCC: (Ubuntu 9.4.0-1ubuntu1˜20.04.1) 9.4.0"
.section .note.GNU-stack,"",@progbits
.section .note.gnu.property,"a"
.align 8
.long  1f - 0f
.long  4f - 1f
.long  5
0:
.string  "GNU"
1:
.align 8
.long  0xc0000002
.long  3f - 2f
2:
.long  0x3
3:
.align 8
4:
```

Since `lab5_prob5_print` needs to be included in `lab5_prob5_main`. Because of this `line 18` adds a `@PLT` to the `call`.

Problem 6



```
int very_fast_function(int i)
{
    if ((i * 64 + 1) > 1024)
        return i++;
    else
        return 0;
    __asm__(
        "rrmovq %rbx, %rcx"
        "rraddq %rcx, %rcx\n\t"
        "rraddq %rcx, %rcx\n\t"
        "rraddq %rcx, %rcx\n\t"
        "rraddq %rcx, %rcx\n\t"
        "rraddq %rcx, %rcx\n\t"
        "iraddq $0x1, %rcx\n\t"
        "isubq $0x400, %rcx\n\t"
        "jle else\n\t"
        "iraddq $0x1, %rbx\n\t"
        "rrmovq %rbx, %rax\n\t"
        "ret\n\t"
        "else:\n\t"
        "irmovq $0x0, %rax\n\t"
        "ret\n\t");
}
```

Figure 3: Question 6