CSE3666 — Lab 1

Mike Medved

January 24th, 2022

1 Deliverables

```
# CSE 3666 Lab 1

.globl main
.text

main:
        # use system call 5 to read integers
        addi    a7, x0, 5
        ecall
        addi    s1, a0, 0       # a in s1

        # using pseudoinstructions
        li      a7, 5
        ecall
        mv      s2, a0          # b in s2

        # compute GCD(a, b) and print it
        bne s1, s2, loop  # enter the loop
        beq s1, s2, exit  # a == b, GCD is a

loop:
        ble s1, s2, lte   # if a < b, enter the "lte" routine
        j gt              # if a > b, enter the "gt" routine

gt:
        sub s1, s1, s2    # a = a - b
        beq s1, s2, exit  # if a == b, exit the loop
        bne s1, s2, loop  # if a != b, restart the loop

lte:
        sub s2, s2, s1    # b = b - a
        beq s1, s2, exit  # if a == b, exit the loop
        bne s1, s2, loop  # if a != b, restart the loop

exit:
        li, a7, 1       # set a7, the service number, to 1 (PrintInt)
        add a0, s1, x0  # load determined GCD into argument register a0
        ecall           # execute the syscall to print to stdout

        # sys call to exit
        addi a7, x0, 10
        ecall
```

2 Run Examples

```
11
121
11
-- program is finished running (0) --

24
60
12
-- program is finished running (0) --

192
270
6
-- program is finished running (0) --

14
97
1
-- program is finished running (0) --

2
2
2
-- program is finished running (0) --
```

3 Limitations

This algorithm works for all integers $a > 0$, $b > 0$, however due to it's nature of subtracting the larger of a, b it cannot work for negative integers and will incur an infinite loop. Additionally, for the algorithm to correctly compute the gcd of a, b both $(a, b) > 0$, or $a = b$ must be true. In order to allow for the gcd of negative numbers to be computed, the absolute value of values in registers s_1, s_2 can be computed and reassigned prior to entering the loop.