**Task A**

The program starts by setting an Input Limit. This is done because the syscall will only take in 1 character if nothing was set. The program will then prompt the user to input their Name and ID. After each input the program will store the inputs into a different register so that they won’t be lost on new syscalls. When it is time to output the data, the registers are then loaded into their respective arguments before being called by a syscall.

Input Stage

C:\Users\USER\AppData\Local\Microsoft\Windows\INetCache\Content.Word\javaw_2018-01-08_03-38-20.png

Output Stage

C:\Users\USER\AppData\Local\Microsoft\Windows\INetCache\Content.Word\javaw_2018-01-08_03-38-33.png

Code

#CSF Assignment Task A

#17098640,Irfan Hanafi

.text

#Initialize Name

li $a1, 33 #Input Limit

#Input name

la $a0, NameInput #Input Prompt

li $v0, 4

syscall

li $v0, 8

syscall

la $t0, ($a0) #Stores Name into $t0

#Iniialize ID

li $a1, 9 #Input Limit

#Input ID

la $a0, IDInput #Input Prompt

li $v0, 4

syscall

li $v0, 8

syscall

la $t1, ($a0) #Stores ID into $t1

#Output

la $a0, NewLine #New Line

li $v0, 4

syscall

la $a0, ($t0) #Output Name

li $v0, 4

syscall

la $a0, ($t1) #Output ID

li $v0, 4

syscall

#Exit

li $v0, 10

syscall

.data

NameInput: .asciiz "Enter Your Name: "

IDInput: .asciiz "Enter Your Student ID: "

NewLine: .asciiz "\n"

**Task B**

The code establishes a counter in the beginning and proceeds to run the calculation given. A “loop” symbol was written before the output and the counter is decremented. The code will branch to “loop” as long as the counter is greater than zero.

Input Stage

C:\Users\USER\AppData\Local\Microsoft\Windows\INetCache\Content.Word\javaw_2018-01-12_07-59-34.png

Output Stage

C:\Users\USER\AppData\Local\Microsoft\Windows\INetCache\Content.Word\javaw_2018-01-12_08-00-05.png

Code

#CSF Assignment Task B

#17098640,Irfan Hanafi

.text

#Input ID

la $a0, IDInput #Input Prompt

li $v0, 4

syscall

li $v0, 5

syscall

la $t1, ($v0) #Stores Input in $t1

#Initialize Counter

li $t0, 10

#Intialize 2

li $t2, 2

#Divide By 2

div $s0, $t1,$t2

loop:

#Output

la $a0, NewLine #New Line

li $v0, 4

syscall

#output division

la $a0, ($s0)

li $v0, 1

syscall

#Count Up

addi $t0, $t0, -1

bgtz $t0, loop

#Exit

li $v0, 10

syscall

.data

IDInput: .asciiz "Enter the last two digits of your Student ID: "

NewLine: .asciiz "\n"

**Task C**

The program takes in a 3 digit number and and stores it into 2 different registers($t1, $t7). #t1 is used to do the calculation given while $t7 is used as a counter for one of the loops. To achieve the benchmark of 30 seconds. A loop with 2 nested loops was made. The first 2 loops have a fixed number of repetitions as using the input as the amount of repetitions doesn’t reach the targeted duration of 30 seconds. Using the input as the amount of repetitions, 1 nested loop will have the program complete the processes in under 30 seconds but with 2 nested loops, the program takes a considerably longer time than 30 seconds. Thus, the first loop and the first nested loop was given a fixed number of repetitions. The 2nd nested loop was given the input so that the amount of time can be changed from 30 seconds or lower depending on the size of input number. A system timer was also implemented using syscall 30. The syscall was initiated both after the input and after the main loop. The time taken at the start and end were put in different registers. The End time subtracts with the Start time to produce the amount of time taken for the program to run. Because the duration is recorded in milliseconds, the duration should be divided by 1000 to get the amount in seconds.

Input Stage

C:\Users\USER\AppData\Local\Microsoft\Windows\INetCache\Content.Word\javaw_2018-01-12_14-08-25.png

Output Stage

C:\Users\USER\AppData\Local\Microsoft\Windows\INetCache\Content.Word\javaw_2018-01-12_14-17-37.png

Code

#17098640, Irfan Hanafi

#Task C CSF Assignment

.text

#Input any 3 digit number

la $a0, InputPrompt #Input Prompt

li $v0, 4

syscall

la $v0, 5

syscall

la $t1, ($v0) #stores input into $t1

#Time

li $v0, 30 #Start Time

syscall

#Loads the Start Time into $t2 and $t3

la $t2, ($a0)

Counter:

li $t0, 5 #Makes Input Result into Counter

li $t6, 5

la $t7, ($t1)

Loop1:

blez $t0, EndLoop1 #Branch if <= 0

Loop2:

blez $t6, EndLoop2

Loop3:

blez $t7, EndLoop3

#Calculation

add $s0, $t1, $t1

#Count

subi $t7, $t7, 1

j Loop3

EndLoop3:

la $t7 ($t1)

#Calculation

add $s0, $t1, $t1

#Count

subi $t6, $t6, 1

j Loop2

EndLoop2:

la $t6, ($t1)

#Calculation

add $s0, $t1, $t1

#Count

subi $t0, $t0, 1

j Loop1

EndLoop1:

#Time

li $v0, 30 #End Time

syscall

#Loads the End Time into $s3 and $s4

la $t4, ($a0)

#Program Time

sub $s2, $t4, $t2

li $s4, 1000

div $s3, $s2, $s4

#Test Output

la $a0, NewLine #New Line

li $v0, 4

syscall

la $a0, ($s0) #Calculation

li $v0, 1

syscall

la $a0, NewLine #New Line

li $v0, 4

syscall

la $a0, ($s3) #Time

li $v0, 1

syscall

la $a0, Seconds

la $v0, 4

syscall

#Exit

li $v0, 10

syscall

.data

Seconds: .asciiz "seconds"

InputPrompt: .asciiz "Enter a Three Digit Number: "

NewLine: .asciiz "\n"

Colon: .asciiz ":"

