**Thực hành kiến trúc máy tính tuần 37**

**Full name: Lê Quang Khải**

**Student ID: 20225638**

Assignment 1

.eqv HEADING 0xffff8010 # Integer: An angle between 0 and 359

.eqv MOVING 0xffff8050 # Boolean: whether or not to move

.eqv LEAVETRACK 0xffff8020 # Boolean (0 or non-0): # whether or not to leave a track

.eqv WHEREX 0xffff8030 # Integer: Current x-location of MarsBot

.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot

.text

main:

    addi $a0, $zero, 120 # Marsbot rotates 90\* and start running

    jal ROTATE

    jal GO

sleep1:

    addi $v0,$zero, 32 # Keep running by sleeping in 5000 ms

    li $a0, 5000

    syscall

    jal TRACK # and draw new track line

go\_150:

    addi $a0, $zero, 150 # Marsbot rotates 150\*

    jal ROTATE

sleep2:

    addi $v0,$zero, 32 # Keep running by sleeping in 3000 ms

    li $a0, 5000

    syscall

    jal UNTRACK # keep old track

    jal TRACK # and draw new track line

goLEFT:

    addi $a0, $zero, 270 # Marsbot rotates 270\* đi ngang sang trái

    jal ROTATE

sleep3:

    addi $v0,$zero, 32 # Keep running by sleeping in 5000 ms

    li $a0, 5000

    syscall

    jal UNTRACK # keep old track

    jal TRACK # and draw new track line

goUP:

    addi $a0, $zero, 30 # Marsbot rotates 30\*

    jal ROTATE

sleep4:

    addi $v0,$zero,32 # Keep running by sleeping in 5000 ms

    li $a0,5000

    syscall

    jal UNTRACK

    jal STOP

end\_main:

    li $v0, 10

    syscall

GO:

    li $at, MOVING # change MOVING port

    addi $k0, $zero, 1 # to logic 1,

    sb $k0, 0($at) # to start running

    jr $ra

STOP:

    li $at, MOVING # change MOVING port to 0

    sb $zero, 0($at) # to stop

    jr $ra

TRACK:

    li $at, LEAVETRACK # change LEAVETRACK port

    addi $k0, $zero, 1 # to logic 1

    sb $k0, 0($at) # to start tracking

    jr $ra

UNTRACK:

    li $at, LEAVETRACK # change LEAVETRACK port to 0

    sb $zero, 0($at) # to stop drawing tail

    jr $ra

ROTATE:

    li $at, HEADING # change HEADING port

    sw $a0, 0($at) # to rotate robot

    jr $ra

**Kết quả:**

A screenshot of a computer

Description automatically generated

Assignment 2:

.eqv HEADING 0xffff8010 # Integer: An angle between 0 and 359

.eqv MOVING 0xffff8050 # Boolean: whether or not to move

.eqv LEAVETRACK 0xffff8020 # Boolean (0 or non-0): # whether or not to leave a track

.eqv WHEREX 0xffff8030 # Integer: Current x-location of MarsBot

.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot

.text

main:

    addi $a0, $zero, 120 # Marsbot rotates 90\* and start running

    jal ROTATE

    jal GO

sleep1:

    addi $v0, $zero, 32 # Keep running by sleeping in 5000 ms

    li $a0, 8000

    syscall

    jal TRACK # and draw new track line

goDOWN:

    addi $a0, $zero, 180 # Marsbot rotates 150\*

    jal ROTATE

sleep2:

    addi $v0,$zero, 32 # Keep running by sleeping in 3000 ms

    li $a0, 5000

    syscall

    jal UNTRACK # keep old track

    jal TRACK # and draw new track line

goLEFT:

    addi $a0, $zero, 270 # Marsbot rotates 270\* đi ngang sang trái

    jal ROTATE

sleep3:

    addi $v0,$zero, 32 # Keep running by sleeping in 5000 ms

    li $a0, 5000

    syscall

    jal UNTRACK # keep old track

    jal TRACK # and draw new track line

goUP:

    addi $a0, $zero, 0 # Marsbot rotates 30\*

    jal ROTATE

sleep4:

    addi $v0 ,$zero, 32 # Keep running by sleeping in 5000 ms

    li $a0,5000

    syscall

    jal UNTRACK

    jal TRACK

goRIGHT:

    addi $a0, $zero, 90 # Marsbot rotates 30\*

    jal ROTATE

sleep5:

    addi $v0,$zero, 32 # Keep running by sleeping in 5000 ms

    li $a0,5000

    syscall

    jal UNTRACK

    jal TRACK

    jal STOP

end\_main:

    li $v0, 10

    syscall

GO:

    li $at, MOVING # change MOVING port

    addi $k0, $zero, 1 # to logic 1,

    sb $k0, 0($at) # to start running

    jr $ra

STOP:

    li $at, MOVING # change MOVING port to 0

    sb $zero, 0($at) # to stop

    jr $ra

TRACK:

    li $at, LEAVETRACK # change LEAVETRACK port

    addi $k0, $zero, 1 # to logic 1

    sb $k0, 0($at) # to start tracking

    jr $ra

UNTRACK:

    li $at, LEAVETRACK # change LEAVETRACK port to 0

    sb $zero, 0($at) # to stop drawing tail

    jr $ra

ROTATE:

    li $at, HEADING # change HEADING port

    sw $a0, 0($at) # to rotate robot

    jr $ra

**Kết quả:**

**A black and blue square

Description automatically generated**

Assignment 3:

.eqv HEADING 0xffff8010 # Integer: An angle between 0 and 359

.eqv MOVING 0xffff8050 # Boolean: whether or not to move

.eqv LEAVETRACK 0xffff8020 # Boolean (0 or non-0): # whether or not to leave a track

.eqv WHEREX 0xffff8030 # Integer: Current x-location of MarsBot

.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot

.text

main:

    addi $a0, $zero, 120 # Marsbot rotates 90\* and start running

    jal ROTATE

    jal GO

sleep1:

    addi $v0, $zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 5000

    syscall

    jal TRACK # and draw new track line

    addi $a0, $zero, 162

    jal ROTATE

sleep2:

    addi $v0,$zero, 32 # Keep running by sleeping in 3000 ms

    li $a0, 2500

    syscall

    jal UNTRACK # keep old track

    jal TRACK # and draw new track line

    addi $a0, $zero, 90 # goRIGHT

    jal ROTATE

sleep3:

    addi $v0,$zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 2500

    syscall

    jal UNTRACK # keep old track

    jal TRACK # and draw new track line

    addi $a0, $zero, 234

    jal ROTATE

sleep4:

    addi $v0 ,$zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 2500

    syscall

    jal UNTRACK

    jal TRACK

    addi $a0, $zero, 162 # Marsbot rotates 30\*

    jal ROTATE

sleep5:

    addi $v0 ,$zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 2500

    syscall

    jal UNTRACK

    jal TRACK

    addi $a0, $zero, 306 # Marsbot rotates 30\*

    jal ROTATE

sleep6:

    addi $v0 ,$zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 2500

    syscall

    jal UNTRACK

    jal TRACK

    addi $a0, $zero, 234

    jal ROTATE

sleep7:

    addi $v0 ,$zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 2500

    syscall

    jal UNTRACK

    jal TRACK

    addi $a0, $zero, 18

    jal ROTATE

sleep8:

    addi $v0 ,$zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 2500

    syscall

    jal UNTRACK

    jal TRACK

    addi $a0, $zero, 306

    jal ROTATE

sleep9:

    addi $v0 ,$zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 2500

    syscall

    jal UNTRACK

    jal TRACK

    addi $a0, $zero, 90

    jal ROTATE

sleep10:

        addi $v0 ,$zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 2500

    syscall

    jal UNTRACK

    jal TRACK

    addi $a0, $zero, 18

    jal ROTATE

stop:

    addi $v0,$zero, 32 # Keep running by sleeping in 2500 ms

    li $a0, 2500

    syscall

    jal UNTRACK

    jal TRACK

    jal STOP

end\_main:

    li $v0, 10

    syscall

GO:

    li $at, MOVING # change MOVING port

    addi $k0, $zero, 1 # to logic 1,

    sb $k0, 0($at) # to start running

    jr $ra

STOP:

    li $at, MOVING # change MOVING port to 0

    sb $zero, 0($at) # to stop

    jr $ra

TRACK:

    li $at, LEAVETRACK # change LEAVETRACK port

    addi $k0, $zero, 1 # to logic 1

    sb $k0, 0($at) # to start tracking

    jr $ra

UNTRACK:

    li $at, LEAVETRACK # change LEAVETRACK port to 0

    sb $zero, 0($at) # to stop drawing tail

    jr $ra

ROTATE:

    li $at, HEADING # change HEADING port

    sw $a0, 0($at) # to rotate robot

    jr $ra

**Kết quả:**

**A white star with a black square

Description automatically generated**

Assignment 4:

.eqv KEY\_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte

.eqv KEY\_READY 0xFFFF0000 # =1 if has a new keycode ? # Auto clear after lw

.eqv DISPLAY\_CODE 0xFFFF000C # ASCII code to show, 1 byte

.eqv DISPLAY\_READY 0xFFFF0008 # =1 if the display has already to do # Auto clear after sw

.text

    li $k0, KEY\_CODE

    li $k1, KEY\_READY

    li $s0, DISPLAY\_CODE

    li $s1, DISPLAY\_READY

    li $t3, 0

loop:

    nop

WaitForKey:

    lw $t1, 0($k1) # $t1 = [$k1] = KEY\_READY

    beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling

ReadKey:

    lw $t0, 0($k0) # $t0 = [$k0] = KEY\_CODE

WaitForDis:

    lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY\_READY

    beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling

Encrypt:

    jal check\_exit

    jal check\_uppercase

ShowKey:

    sw $t0, 0($s0) # show key

    bge $t3, 4, end\_main # khi $t3 = 4 là đã chạy hết chữ exit

    nop

    j loop

end\_main:

    li $v0, 10

    syscall

check\_uppercase:

    bgt $t0, 'Z', check\_lowercase

    blt $t0, 'A', check\_number

    addi $t0, $t0, 32

    jr $ra

check\_lowercase:

    bgt $t0, 'z', default

    blt $t0, 'a', default

    subi $t0, $t0, 32

    jr $ra

check\_number:

    blt $t0, '0', default

    bgt $t0, '9', default

    jr $ra

default:

    addi $t0, $zero, 32 # $t0 = \*

    jr $ra

check\_exit:

    beq $t0, 'e', check\_e

    beq $t0, 'x', check\_x

    beq $t0, 'i', check\_i

    beq $t0, 't', check\_t

    li $t3, 0

    jr $ra

check\_e:

    bne $t3, 0, return

    addi $t3, $t3, 1

    jr $ra

check\_x:

    bne $t3, 1, return

    addi $t3, $t3, 1

    jr $ra

check\_i:

    bne $t3, 2, return

    addi $t3, $t3, 1

    jr $ra

check\_t:

    bne $t3, 3, return

    addi $t3, $t3, 1

    jr $ra

return:

    li $t3, 0

    jr $ra

**Kết quả:**

A screenshot of a computer

Description automatically generated

Assignment 5:

.eqv KEY\_CODE 0xFFFF0004    # ASCII code from keyboard, 1 byte

.eqv KEY\_READY 0xFFFF0000   # =1 if has a new keycode ? # Auto clear after lw

.eqv DISPLAY\_CODE 0xFFFF000C    # ASCII code to show, 1 byte

.eqv DISPLAY\_READY 0xFFFF0008   # =1 if the display has already to do # Auto clear after sw

.eqv HEADING 0xffff8010 # Integer: An angle between 0 and 359

.eqv MOVING 0xffff8050 # Boolean: whether or not to move

.eqv LEAVETRACK 0xffff8020 # Boolean (0 or non-0): # whether or not to leave a track

.text

    li $k0, KEY\_CODE

    li $k1, KEY\_READY

    li $s0, DISPLAY\_CODE

    li $s1, DISPLAY\_READY

main:

loop:

    nop

WaitForKey:

    lw $t1, 0($k1) # $t1 = [$k1] = KEY\_READY

    beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling

ReadKey:

    lw $t0, 0($k0) # $t0 = [$k0] = KEY\_CODE

WaitForDis:

    lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY\_READY

    beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling

bot\_move:

    beq $t0, 'w', GO\_UP

    beq $t0, 'W', GO\_UP

    beq $t0, 's', GO\_DOWN

    beq $t0, 'S', GO\_DOWN

    beq $t0, 'a', GO\_LEFT

    beq $t0, 'A', GO\_LEFT

    beq $t0, 'd', GO\_RIGHT

    beq $t0, 'D', GO\_RIGHT

    beq $t0, 32, start\_stop # Space key for start/stop

    j start

GO\_UP:

    addi $a0, $zero, 0

    jal UNTRACK

    jal ROTATE

    jal TRACK

    j start

GO\_RIGHT:

    addi $a0, $zero, 90

    jal UNTRACK

    jal ROTATE

    jal TRACK

    j start

GO\_DOWN:

    addi $a0, $zero, 180

    jal UNTRACK

    jal ROTATE

    jal TRACK

    j start

GO\_LEFT:

    addi $a0, $zero, 270

    jal UNTRACK

    jal ROTATE

    jal TRACK

    j start

start\_stop:

    li $at, MOVING

    lb $t3, 0($at)

    beq $t3, 1, STOP

    beq $t3, 0, GO

start:

    j loop

end\_main:

    li $v0, 10

    syscall

GO:

    li $at, MOVING # change MOVING port

    addi $s2, $zero, 1 # to logic 1,

    sb $s2, 0($at) # to start running

    jr $ra

STOP:

    li $at, MOVING # change MOVING port to 0

    sb $zero, 0($at) # to stop

    jr $ra

TRACK:

    li $at, LEAVETRACK # change LEAVETRACK port

    addi $s2, $zero, 1 # to logic 1

    sb $s2, 0($at) # to start tracking

    jr $ra

UNTRACK:

    li $at, LEAVETRACK # change LEAVETRACK port to 0

    sb $zero, 0($at) # to stop drawing tail

    jr $ra

ROTATE:

    li $at, HEADING # change HEADING port

    sw $a0, 0($at) # to rotate robot

    jr $ra

**Kết quả:**

