Thực hành Kiến trúc máy tính

Week 10

Vũ Ngọc Đức – 20225816

Assignment 1

1.

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

# 0 : North (up)

# 90: East (right)

# 180: South (down)

# 270: West (left)

.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, 90 # Marsbot rotates 90\* and start running

jal ROTATE

jal GO

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,15000

syscall

addi $a0, $zero, 180

jal ROTATE

jal GO

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep1:

addi $a0, $zero, 150

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep2:

addi $a0, $zero, 270

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep3:

addi $a0, $zero, 30

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

end\_main:

jal UNTRACK # keep old track

addi $a0, $zero, 90

jal ROTATE

jal GO

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,3000

syscall

jal STOP

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

ROTATE:

li $at, HEADING # change HEADING port

sw $a0, 0($at) # to rotate robot

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

Ảnh có chứa ảnh chụp màn hình, hàng, biểu đồ

Mô tả được tạo tự động

2.

.eqv HEADING 0xffff8010

.eqv MOVING 0xffff8050

.eqv LEAVETRACK 0xffff8020

.eqv WHEREX 0xffff8030

.eqv WHEREY 0xffff8040

.text

main:

addi $a0, $zero, 90

jal ROTATE

jal GO

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,15000

syscall

addi $a0, $zero, 180

jal ROTATE

jal GO

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep1:

addi $a0, $zero, 90

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep2:

addi $a0, $zero, 180

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep3:

addi $a0, $zero, 270

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep4:

addi $a0, $zero, 0

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

end\_main:

jal UNTRACK # keep old track

addi $a0, $zero, 90

jal ROTATE

jal GO

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,3000

syscall

jal STOP

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

ROTATE:

li $at, HEADING # change HEADING port

sw $a0, 0($at) # to rotate robot

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

Ảnh có chứa ảnh chụp màn hình, biểu đồ, hàng, văn bản

Mô tả được tạo tự động

3.

.eqv HEADING 0xffff8010

.eqv MOVING 0xffff8050

.eqv LEAVETRACK 0xffff8020

.eqv WHEREX 0xffff8030

.eqv WHEREY 0xffff8040

.text

main:

addi $a0, $zero, 90

jal ROTATE

jal GO

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,15000

syscall

addi $a0, $zero, 180

jal ROTATE

jal GO

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep1:

addi $a0, $zero, 162

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep2:

addi $a0, $zero, 306

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep3:

addi $a0, $zero, 90

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep4:

addi $a0, $zero, 234

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

sleep5:

addi $a0, $zero, 18

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,7000

syscall

end\_main:

jal UNTRACK # keep old track

addi $a0, $zero, 90

jal ROTATE

jal GO

addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,3000

syscall

jal STOP

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

ROTATE:

li $at, HEADING # change HEADING port

sw $a0, 0($at) # to rotate robot

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

Ảnh có chứa ảnh chụp màn hình, văn bản, biểu đồ

Mô tả được tạo tự động

Assignment 2

.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 # chua ky tu can in ra man hinh

li $s1, DISPLAY\_READY

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

Check:

Check\_E:

beq $t3, 1, Check\_X

beq $t0, 101, Co

Check\_X:

beq $t3, 2, Check\_I

beq $t0, 120, Co

Check\_I:

beq $t3, 3, Check\_T

beq $t0, 105, Co

Check\_T:

beq $t3, 4, Print

beq $t0, 116, Co

Reset:

addi $t3, $zero, 0

Print:

ChuHoa:

bgt $t0, 90, ChuThuong

blt $t0, 65, ChuThuong

addi $t0, $t0, 32

j ShowKey

ChuThuong:

bgt $t0, 122, ChuSo

blt $t0, 97, ChuSo

addi $t0, $t0, -32

j ShowKey

ChuSo:

bgt $t0, 57, Other

blt $t0, 48, Other

addi $t0, $t0, 0

j ShowKey

Other:

addi $t0, $zero, 42

ShowKey:

sw $t0, 0($s0) # show key

nop

beq $t3, 4, Exit

j Loop

Co:

addi $t3, $t3, 1

j Print

Exit:

li $v0, 10

syscall

Ảnh có chứa văn bản, ảnh chụp màn hình, số, màn hình

Mô tả được tạo tự động

Assignment 3

.eqv HEADING 0xffff8010

.eqv MOVING 0xffff8050

.eqv LEAVETRACK 0xffff8020

.eqv WHEREX 0xffff8030

.eqv WHEREY 0xffff8040

.eqv KEY\_CODE 0xFFFF0004

.eqv KEY\_READY 0xFFFF0000

.eqv DISPLAY\_CODE 0xFFFF000C

.eqv DISPLAY\_READY 0xFFFF0008

.text

li $k0, KEY\_CODE

li $k1, KEY\_READY

li $s0, DISPLAY\_CODE

li $s1, DISPLAY\_READY

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

Print:

beq $t0, 65, sleepA

beq $t0, 97, sleepA

beq $t0, 87, sleepW

beq $t0, 119, sleepW

beq $t0, 68, sleepD

beq $t0, 100, sleepD

beq $t0, 83, sleepS

beq $t0, 115, sleepS

beq $t0, 32, Nghiem

beq $t0, 67, Ditiep

beq $t0, 99, Ditiep

ShowKey:

sw $t0, 0($s0) # show key

nop

j Loop

sleepW:

addi $a0, $zero, 0

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

j ShowKey

sleepS:

addi $a0, $zero, 180

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

j ShowKey

sleepD:

addi $a0, $zero, 90

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

j ShowKey

sleepA:

addi $a0, $zero, 270

jal ROTATE

jal GO

jal UNTRACK # keep old track

jal TRACK # and draw new track line

j ShowKey

Nghiem:

jal STOP

j ShowKey

Ditiep:

jal GO

j ShowKey

GO:

li $at, MOVING # change MOVING port

addi $k0, $zero,1 # to logic 1,

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

jr $ra

ROTATE:

li $at, HEADING # change HEADING port

sw $a0, 0($at) # to rotate robot

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