

Riešenie 5. zadania

UPRATOVACÍ ROBOT - MIPSIM

Zadanie:

Napíšte program, ktorý bude simulovať pohyb upratovacieho robota po miestnosti reprezentovanej ako pole podľa obrázka. Ľavé horné políčko má súradnice (riadok, stĺpec) = (1,1) a pravé spodné políčko má súradnice (5,5). Robot môže začínať na ľubovoľnom políčku, okrem prekážok a môže vykonávať kroky o 1 políčko smerom na niektorú svetovú stranu. Políčko, ktoré robot navštívi, je následne považované za upratané (vrátane štartovacieho). Políčko s prekážkou sa vždy počíta ako neupratané. Na niektorých políčkach už robot bol (viď. obrázok), avšak môže na nich opätovne vstúpiť aj viac krát. Od adresy 305 bude v pamäti údajov pred spustením programu uložená postupnosť čísel (každé číslo na samostatnej adrese) reprezentujúca trasu, ktorú robot vykoná, a to nasledovným spôsobom. Pohyb smerom:

- hore = 1,
- vpravo = 2,
- dole = 3,
- vľavo = 4.

Hodnota 0 reprezentuje ukončenie postupnosti. Môžete predpokladať, že robot sa nikdy nepokúsi vstúpiť na políčko s prekážkou, ani sa nepokúsi opustiť miestnosť. Zistíte, či po vykonaní zadanej postupnosti krokov robot upratol aspoň zadanú časť miestnosti (viď. konkrétne úlohy). Ak áno, na adresu 302 uložte hodnotu 1. Ak nie, na adresu 302 uložte hodnotu 0. Aktuálnu (a teda aj štartovaciu) riadkovú a stĺpcovú súradnicu robota uchovávať na adresách 300 resp. 301.

Úloha 54

	1	2	3	4	5
1	✓	✓	✓	✓	✓
2	✓	■		■	
3	✓				
4		■		■	
5					

■ - prekážka, ✓ - už upratané

cieľ: 14 políček

Riešenie

Pamäť programu

Label	Inštrukcia	Komentár
	LW \$25, 00A0(\$0)	načítame začiatočnú riadkovú súradnicu z PÚ z adresy a0h do registra R25
	LW \$26, 00B0(\$0)	načítame začiatočnú stĺpcovú súradnicu z PÚ z adresy b0h do registra R26
zac	LW \$22, 0000(\$20)	do registra R22 načítame prvok postupnosti z PÚ z adresy, na ktorú ukazuje ukazovateľ v registri R20
	ADDI \$20, \$20, 0004	zvážšime ukazovateľ v registri R20 o 4, aby ukazoval na ďalší prvok postupnosti v poradí
	...	
	BEQ \$22, \$1, pohyb1	ak je načítaný prvok postupnosti v reg. R22 rovný 1 (konštantu 1 máme uloženú v reg. R1) skoč na podprogram pre vykonanie pohybu hore ktorý sa nachádza na labeli „pohyb1“
	BEQ \$22, \$2, pohyb2	ak je načítaný prvok postupnosti v reg. R22 rovný 2 (konštantu 2 máme uloženú v reg. R2) skoč na podprogram pre vykonanie pohybu vpravo ktorý sa nachádza na labeli „pohyb2“
	BEQ \$22, \$3, pohyb3	ak je načítaný prvok postupnosti v reg. R22 rovný 3 (konštantu 3 máme uloženú v reg. R3) skoč na podprogram pre vykonanie pohybu dole ktorý sa nachádza na labeli „pohyb3“
	BEQ \$22, \$4, pohyb4	ak je načítaný prvok postupnosti v reg. R22 rovný 4 (konštantu 4 máme uloženú v reg. R4) skoč na podprogram pre vykonanie pohybu vľavo ktorý sa nachádza na labeli „pohyb4“
	...	
	SW \$25, 00A0(\$0)	uložíme výslednú riadkovú adresu z R25 do PÚ na adresu a0h
	SW \$26, 00B0(\$0)	uložíme výslednú stĺpcovú adresu z R26 do PÚ na adresu b0h
	BEQ \$0, \$0, kon	a skočíme niekam na koniec programu
	...	
pohyb1	SUBI \$25, \$25, 0001	pohyb hore – znížime riadkovú súradnicu o 1
	BEQ \$0, \$0, najdibod	a skočíme na podprogram na hľadanie uprataného bodu na labeli “najdibod”
	...	
pohyb2	ADDI \$26, \$26, 0001	pohyb vpravo – zvýšime stĺpcovú súradnicu o 1
	BEQ \$0, \$0, najdibod	a skočíme na podprogram na hľadanie uprataného bodu na labeli “najdibod”

	...	
pohyb3	ADDI \$25, \$25, 0001	
	BEQ \$0, \$0, najdibod	a skočíme na podprogram na hľadanie uprataného bodu na labeli "najdibod"
	...	
pohyb4	SUBI \$26, \$26, 0001	
	BEQ \$0, \$0, najdibod	a skočíme na podprogram na hľadanie uprataného bodu na labeli "najdibod"
	...	
kon	NOP	ukončíme program
	...	
najdibod	LW \$14, 0000(\$10)	načítame hodnotu uloženú na adrese 0000 na ktoru ukazuje register R10 a vložíme ju do registra R14
	LW \$15, 0004(\$10)	načítame hodnotu uloženú na adrese 0004 na ktoru ukazuje register R10 a vložíme ju do registra R15
	BEQ \$14, \$25, najdibo2	ak sa rovnajú riadkové suradnice skoč na podprogram pre skontrolovanie stĺpcovej suradnice na labeli "najdibo2"
	BEQ \$14, \$6, upracbod	ak sa riadková suradnica uložená v registri R25 rovná konstante 10 uloženej v registri R6 skoč na podprogram na labeli "upracbod"
	...	
	ADDI \$10, \$10, 0008	zvys hodnotu R10 o 8
	BNEQ \$14, \$25, najdibod	skoč na ďalšiu iteráciu cyklu
	...	
najdibo2	BEQ \$15, \$26, reset	ak sa rovnajú stĺpcové suradnice skoč na podprogram reset registra R10 na labeli "reset"
	BNEQ \$15, \$26, najdibod	ak nie skoč na ďalšiu iteráciu cyklu
	...	
upracbod	SW \$25, 0000(\$10)	vlož na adresu uloženú v R10 hodnotu uloženú v R1
	SW \$26, 0004(\$10)	vlož na adresu uloženú v R10 hodnotu uloženú v R1
	ADDI \$10, \$10, 0008	zvys hodnotu R10 o 8
	SW \$6, 0000(\$10)	vlož na adresu uloženú v R10 hodnotu uloženú v R6
reset	ADDI \$16, \$16, 0001	zvys hodnotu R16 o 1
	ADDI \$10, \$7, 0000	nastav hodnotu v R10 na takú aká je v R7
	BEQ \$16, \$8, upratane	ak je uprataných 14 bodov skoc na label "upratane"

Emma Macháčová, id 103037
Utorok 18:00

	BEQ \$10, \$7, zac	ak nie skoč na začiatok
	...	
upratane	SW \$1, 0000(\$9)	ulož na adresu uloženú v R9 hodnotu uloženú v R1
	BEQ \$10, \$7, zac	skoč na začiatok

Stav registrov na začiatku programu

P0	00000000	R16	00000007
R1	00000001	R17	00000000
R2	00000002	R18	00000000
R3	00000003	R19	00000000
R4	00000004	R20	00000000
R5	00000005	R21	00000000
R6	00000010	R22	00000000
R7	00000200	R23	00000000
R8	00000014	R24	00000000
R9	000000e0	R25	00000000
R10	00000200	R26	00000000
R11	00000000	R27	00000000
R12	00000000	R28	00000000
R13	00000000	R29	00000000
R14	00000000	R30	00000000
R15	00000000	R31	00000000

- R1 - konštanta 1
- R2- konštanta 2
- R3 - konštanta 3
- R4 - konštanta 4
- R5 - konštanta 5
- R6- konštanta 10
- R7 - adresa prvého uloženého bodu
- R8 - koľko bodov má byť upratane
- R9 - adresa miesta v pamäti kde sa ukladá výsledok programu
- R14 - sem sa načíta riadková súradnica bodu
- R15 - sem sa načíta stĺpcová súradnica bodu
- R16 - aktuálny počet uprataných bodov
- R20 - ukazovateľ na prvok postupnosti príkazov
- R22- načítaný prvok postupnosti príkazov
- R25 - riadková súradnica robota
- R26 - stĺpcová súradnica robota

Pamäť údajov

prvky postupnosti:

address	data memory			
00000000	00000001	00000001	00000001	00000002
00000010	00000002	00000002	00000002	00000003
00000020	00000003	00000003	00000003	00000004
00000030	00000004	00000004	00000004	00000004

začiatočné súradnice

address	data memory			
000000a0	00000004	00000000	00000000	00000000
000000b0	00000001	00000000	00000000	00000000

výsledok programu

address	data memory			
000000e0	00000000	00000000	00000000	00000000

uložene body

address	data memory			
00000210	00000002	00000001	00000000	00000000
00000210	00000002	00000001	00000000	00000000
00000220	00000001	00000001	00000000	00000000
00000230	00000001	00000002	00000000	00000000
00000240	00000001	00000003	00000000	00000000
00000250	00000001	00000004	00000000	00000000
00000260	00000001	00000005	00000000	00000000
00000270	00000010	00000010	00000000	00000000