

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

BETRIEBSSYSTEME

Blatt 06



Students:

Julian Polzer jp390, David Janzen dj57

Tutor:

GRUPPE 7

1 Aufgabe

HIER FEHLT NOCH DIE SYMBOLTABELLE

st(x)=(var,int,128)

st(y)=(var,int,129)

st(z)=(const,int,2)

ReTi Code	Erklärung
1. LOADI ACC 2	ACC = 2
2. STOREIN DS ACC 0	An Anfang des Datensegments speichern
3. LOADI ACC 3	ACC = 3
4. STOREIN DS ACC 1	An Stelle 1 des Datensegments speichern
5. LOADI ACC 15	ACC = 15
6. STOREIN DS ACC 2	An Stelle 3 des Datensegments speichern
7. LOADIN DS ACC 0	z Laden (Stelle 0 von DS)
8. MOVE ACC IN2	z in IN2 speichern
9. LOADIN DS ACC 1	y in den ACC laden (Stelle 1 DS)
10. MUL ACC IN2	$y * z$
11. MOVE ACC IN1	ACC in IN1 speichern
12. LOADIN DS ACC 2	x laden (DS stelle 2)
13. SUB ACC IN1	$x - (y * z)$ rechnen
14. JUMP ≤ 5	JUMP if $x - (y * z) \leq 0$
15. LOADIN DS ACC 2	x laden (Stelle 2 im DS)
16. SUBI ACC 3	$x - 3$
17. STOREIN DS ACC 2	neuen x-Wert abspeichern
18. JUMP -6	zurück zu zeile 12 (schleife Neu beginnen)
19. JUMP 0	Programmende

andere Version, weil es nur einen Hauptspeicher gibt:

ReTi Code	Erklärung
1. LOADI ACC 2	ACC = 2
2. STORE ACC 0	An Anfang des Datenspeichers speichern
3. LOADI ACC 3	ACC = 3
4. STORE ACC 1	An Stelle 1 des Datenspeichers speichern
5. LOADI ACC 15	ACC = 15
6. STORE ACC 2	An Stelle 3 des Datenspeichers speichern
7. LOAD ACC 0	z Laden (Stelle 0 von DS)
8. MOVE ACC IN2	z in IN2 speichern
9. LOAD ACC 1	y in den ACC laden (Stelle 1 DS)
10. MUL ACC IN2	$y * z$
11. MOVE ACC IN1	ACC in IN1 speichern
12. LOAD ACC 2	x laden (DS stelle 2)
13. SUB ACC IN1	$x - (y * z)$ rechnen
14. JUMP ≤ 5	JUMP if $x - (y * z) \leq 0$
15. LOAD ACC 2	x laden (Stelle 2 im DS)
16. SUBI ACC 3	$x - 3$
17. STORE ACC 2	neuen x-Wert abspeichern
18. JUMP -6	zurück zu zeile 12 (schleife Neu beginnen)
19. JUMP 0	Programmende

2 Aufgabe

3 Aufgabe

a) benutzte Speicherzellen + b) Belegung

genutzte Speicherzellen und Belegung
1. (Vielleicht der inklude) 2. $M(8) = \text{st}(p1)$
3. $M(9) = \text{st}(p2)$
4. $M(10) = \text{st}(a)$
5. $M(15-16) = p2$
6. $M(10) = M(15)$
7. $M(15) = 7$
8. $M(16) = 4$
Marke 1
9. $M(8) = M(33-34)$
10. $M(34) = M(10) = 7$
11. $M(9) = M(8) = M(33-34)$
12. $M(12) = M(15-16)$
Marke 2
$M(10) = 43$
Marke 3
$M(33-34) = \text{free}$

c) ist Free erlaubt ?

Ja weil es genau auf den gleichen Speicherplatz zugreift, den p1 davor allocated hat.