Betriebssysteme Blatt 6

Baran Güner, bg160 Tobias Hangel, th151

2. Dezember 2022

Aufgabe 1

$st(z) = (const, int, 2)$
$st(x) = (var, int, 128)$
$st(y) = (var, int, 129)$

PC	Befehl	Kommentar
1	ADDI 128 3	x wird auf 3 gesetzt.
2	ADDI 129 15	y wird auf 15 gesetzt.
3	SUBI SP 1	
4	LOAD ACC 128	
5	STOREIN SP ACC 1	x wird auf die erste Adresse gelegt.
6	SUBI SP 1	
7	LOADI ACC 129	
8	MULI ACC 2	y wird mit z multipliziert.
8	STOREIN SP ACC 1	Das Ergebnis wird auf die zweite Adresse gelegt.
9	LOADIN SP ACC 2	x wird in ACC geladen.
10	LOADIN SP IN2 1	y wird in IN2 geladen.
11	SUB ACC IN2	x - y wird in ACC geladen.
12	$JUMP_{\geq}3$	Ergebnis 1 wenn $x \geq y$ wahr.
13	LOADI ACC 0	Ergebnis o wenn $x \geq y$ falsch.
14	JUMP 2	
15	LOADI ACC 1	Ergebnis 1 wenn $x \geq y$ wahr.
16	STOREIN SP ACC 2	Ergebnis in zweitoberste Stack-Zelle.
17	ADDI SP 1	Stack um eine Zelle verkürzen.
18	LOADIN SP ACC 1	Wahrheitswert in ACC laden.
19	ADDI SP 1	Stack um eine Zelle verkürzen.
20	$JUMP_{=}$ 5 + 2	af überspringen wenn falsch.
21	LOAD ACC 128	x in ACC.
22	SUBI ACC 3	x = x-3
23	STORE ACC 128	x in 128 speichern.
24	JUMP -22	Zurück zur Auswertung.

Aufgabe 2

$l_1 = s_2 * s_3$, also $l_1 = 6$.
$l_2 = s_3$, also $l_2 = 3$.
l_3 wird nicht verwendet.

Die gesuchte Adresse setzt sich also bei den gegebenen Parametern folgendermaßen zusammen:
$M(a + val(e_1) * 6 + val(e_2) * 3 + val(e_3))$.
Da das Ermitteln der Werte von e_1, e_2 und e_3 mit $code^{aa}(e_i)$ bestimmt werden kann werden diese Werte in den ersten 5 Befehlen mit den bereits berechneten l_i multipliziert und abgespeichert(Mit Ausnahme von e_3, da hier kein $l_3$multipliziert werden muss).
Anschließend werden die berechneten Werte miteinander addiert und in IN1 gespeichert.

PC	Befehl	Kommentar
1	LOADI ACC $code^{aa}(e_1)$	e_1 wird in den ACC geladen.
2	MULI ACC 6	e_1 wird mit l_1(also 6) multipliziert.
2	STORE ACC 0	$l_1 * e_1$ wird in M(0) gespeichert
3	LOADI ACC $code^{aa}(e_2)$	e_2 wird in den ACC geladen.
2	MULI ACC 3	e_2 wird mit l_2(also 3) multipliziert.
4	STORE ACC 0	$l_2 * e_2$ wird in M(1) gespeichert
5	LOADI ACC $code^{aa}(e_3)$	e_3 wird in den ACC geladen.
6	ADD ACC 0	Im ACC befindet sich $e_1 * 6 + e_3$.
7	ADD ACC 1	Im ACC befindet sich $e_1 * 6 + e_2 * 3 + e_3$, die komplette Adresse.
8	STOREIN IN1 ACC	Lädt die Adresse vom ACC in IN1