

Betriebssysteme

Übungsblatt 6

Anne Rossl

Diana Hörth

December 2, 2022

Aufgabe 1

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

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

st(x) = (var int, 128)

PC	Befehl	Kommentar
0	LOADIN SP IN1 2	Lade y in IN1.
1	LOADIN SP IN2 1	Lade x in IN2
2	LOADIN SP ACC 3	Lade z in ACC.
3	MUL ACC IN1	Multipliziere y und z..
4	SUB ACC IN2	Subtrahiere das Ergebnis der vorherigen Multiplikation mit x
5	JUMP _{>} 5	Falls das Ergebnis größer 0 ist bende Schleife.
6	LOAD ACC IN2	Überschreibe ACC mit x.
7	SUBI ACC 3	Ziehe drei von x ab.
8	LOAD IN2 ACC	x aktualisieren.
9	JUMP ₋₇	Wieder hochspringen und Schleifenbedingung überprüfen.
10	SUBI SP 1	SP wir eins nach oben verschoben.
11	STOREIN SP IN2 1	Ergebnis wir drauf gelegt
12	JUMP 0	beenden

Aufgabe 2

Aufgabe 3

a)

- (struct point p2) : Gelesen, weil der Typ eine selbstdefinierte Klasse ist, und nach geschaut werden muss was für Attribute die Klasse hat.
- (p2.x = 7) : Hier wird auf die Speicheradresse geschrieben und zwar eine 7.
- (p2.y = 4) : Hier wird auf die Speicheradresse geschrieben und zwar eine 4.
- (a = (p2.x)) : Auf den Pointer a wird die Referenz von p2.x geschrieben.
- ((*p1).y = *a) : Hier wird gelesen worauf a zeigt.

- `p3 = p1` : `p1` wird in `p3` geschrieben.
- `p1 = p2` : Die Referenz von `p2` wird in `p1` geschrieben.
- `if((*p1).y > 5)` : wird abgelesen, um es mit 5 zu vergleichen.
- `*a = 42` : hier wird auf den Speicher zugegriffen auf die der Pointer zeigt und überschreibt es mit 42.
- `*a = 1` : hier wird auch wie der auf den Speicher zugegriffen auf den `a` zeigt und der Inhalt dort wird mit 1 überschrieben.
- `p1 = (struct point *) malloc(sizeof(struct point))` : hier wird gelesen und geschrieben. Erst wird mit `sizeof()` gelesen um herauszufinden wie groß `struct point` ist, damit sie wissen wie viel Platz gebraucht wird. Später wird noch geschrieben.
- `free(p3)` : hier wird geschrieben und zwar weil hier was der Inhalt gelöscht wird.

b)

c)

Die letzte Zeile ist zulässig. Hier wird die Speicherzelle auf die der Zeiger zeigt freigegeben.