8. Übung Programmierung

Dominic Deckert

31. Mai 2017

Previously on ...

- AM_0
  - Speicher-Strukturen
  - Befehlssatz
- C_0

1 a)

```
#include <stdio.h>
int main(){
int x, y, a;
scanf("%i", &y);
scanf("%i", &y);
x = 0
```

```
1   READ 2
2   READ 3
3   LIT 0
4   STORE 1
```

1 a)

while(x<a)	5	LOAD 1
	6	LOAD 2
	7	LT
	8	JC 18

1 a)

x = x+1;	9	LOAD 1
	10	LIT 1
	11	ADD
	12	STORE 1
y=y*y;	13	LOAD 2
	14	LOAD 2
	15	MULT
	16	STORE 2

1 a)

	17 JMP 5
printf(“%d”, y);	18 WRITE 3
return 0	

1 b)

BZ	DK	HS	In	Out
1	ε	[]	0:1	ε
2		[1/0]	1	
3		[1/0, 2/1]	ε	
4	0			

1 b)

BZ	DK	HS	In	Out
5	1:0			
6	0:1:0			
7	1:0			
8	0			
5				

1 b)

BZ	DK	HS	In	Out
6	0:0			
7	0			
9	ε			
10	ε	[1/0, 2/1]	ε	1

C_1

C-Dialekt mit:

- void-Funktionen
- Unterprogramm-Aufrufen
- Pointer (u.A. zur Wertrückgabe)

AM_1

- Befehlszähler
- Datenkeller
- **Laufzeitkeller** - dynamischer *random access*-Speicher
- **Referenz-Zeiger** REF
- Input-/ Output-Band

Unterprogramm: lokaler Speicherbereich durch **Aktivierungsblock**
Erzeugung: CALL, Löschung: RETURN

Übersetzung

Symboltabelle: verwaltet lokale Adressen der Variablen & “Adressen” der Programme
Baumstrukturierte Adressen: Adresse der Struktureinheit . Subadresse
Struktureinheit: Programm / while / if / atomarer Befehl

trans

Genaue Übersetzung siehe Extrablatt

Zu beachten: - Erstellung mehrerer (lokaler) Symboltabellen
- getrennte Übersetzung der Unterprogramme
- Parameter mit PUSH übergeben, Anzahl lokaler Variablen mit INIT festlegen
- mit RETURN lokale Variablen löschen

Ergebnis

$$INIT0; CALL2; JMP0;$$

$$\begin{aligned}
1 :& INIT1; LIT2; STORE(I,1);\\
& LOAD(I,-2); LIT1; GT; JMC1.2.1;\\
1.2.2.1.1 :& LOAD(I,-2); LOAD(I,1);\\
& MOD; LIT0; ADD; NE; JMC1.2.2.1.2;\\
& LOAD(I,1); LIT1; ADD; STORE(I,1); JMP1.2.2.1.1
\end{aligned}$$

Ergebnis

$$
\begin{aligned}
1.2.2.1.2 : & \mathit{LOAD}(I,-2); \mathit{LOAD}(I,1); \mathit{DIV};\\
& \mathit{STORE}(I,-2); \mathit{WRITE}(I,1); \mathit{LOAD}(I,-2);\\
& \mathit{PUSH}; \mathit{CALL}1;\\
1.2.1 : & \mathit{RET}1;\\
\\
2 : & \mathit{INIT}1; \mathit{READ}(I,1); \mathit{LOAD}(I,1);\\
& \mathit{PUSH}; \mathit{CALL}1; \mathit{RET}0;
\end{aligned}
$$

3 a)

(1.3)	**1.3.3**	LOAD (lokal, 1), LOADI(-2), LT, JC **1.3.1**
	1.3.2	LOAD (lokal, 2), LIT 2, MULT, STORE (lokal, 2), LOAD (lokal, 2), PUSH, LOAD (lokal, -2), PUSH, CALL **2** JMP **1.3.3**
(1.4)	**1.3.1**	LOAD (lokal, 2), STOREI (-3)

3 b)

BZ	DK	LK	REF	In	Out
14	ε	0:0:1	3	4	ε
15		4:0:1		ε	
16	1				
17	ε	4:0:1:1			
3		4:0:1:1:18:3	6		
4					

3 b)

BZ	DK	LK	REF	In	Out
5	4				
6	2:4				
7	1				
8	ε				
9	4				
10	2:4				

3 b)

BZ	DK	LK	REF	In	Out
11	2				
12	ε	2:0:1:1:18:3			
18		2:0:1	3		
19					2
0	ε	2:0:1	3	ε	2