

# Rechnerarchitektur

SS 2017

## Übungszettel 3

8. Lesen Sie die Einführung in die DLX-Programmierung und starten Sie **openDLX** (im Ordner **apps/dlx\_apps** sind einige Testprogramme) und beantworten Sie folgende Fragen:

- (a) Welche Bedeutung haben **.data** und **.text**?
- (b) Was ist der Unterschied zwischen **add**, **addu**, **addi**, **addui**, **addf** und **addd**?
- (c) Was sind Pointer und wozu braucht man sie?
- (d) Was macht der Befehl **jal**?

9. Schreiben Sie ein DLX-Programm, das die abgerundete Wurzel  $b$  einer natürlichen Zahl  $a$  mittels Intervallhalbierung berechnet, entsprechend folgendem Java-Codeausschnitt:

```
int a = 81, b;  
int u = 0, o = a, m = o/2;  
while (m != u)  
{ if (m*m > a) o = m; else u = m;  
  m = (o+u)/2;  
}  
b = m;
```

10. Schreiben Sie ein DLX-Programm, das das ggT (den größten gemeinsamen Teiler)  $c$  zweier positiver Zahlen  $a, b$  mit dem euklidischen Algorithmus berechnet. Dabei wird der Divisionsrest  $r$  von  $a/b$  (d.h.  $a = nb + r, r < b$ ) berechnet und dann auf gleiche Weise mit  $b$  und  $r$  fortgefahren, bis der Rest  $r = 0$  ist. Halten Sie sich an folgenden Java-Code:

```
int a = 15, b = 6, c;  
while (b > 0)  
{  
  while (a >= b) { a = a - b; } // a = Divisionsrest  
  int tmp = a; a = b; b = tmp; // a und b vertauschen  
}  
c = a;
```

11. Schreiben Sie ein DLX-Programm, das den Bubble-Sort-Algorithmus implementiert. Versuchen Sie, Pointer zu verwenden, um die Performance zu optimieren.

```
int len = 10;  
int arr[] = new int[len];  
for (int i = len - 1; i > 0; i--)  
  for (int j = 0; j < i; j++)  
    if (arr[j] > arr[j+1])  
      { int tmp = arr[j]; arr[j] = arr[j+1]; arr[j+1] = tmp; }
```