

# Rechnerarchitektur

SS 2020

## Übungszettel 3

9. 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:
- Welche Bedeutung haben `.data` und `.text`?
  - Was ist der Unterschied zwischen `add`, `addu`, `addi`, `addui`, `addf` und `addd`?
  - Was sind Pointer und wozu braucht man sie?
  - Was macht der Befehl `jal`?
10. Schreiben Sie ein DLX-Unterprogramm `potenz`, das die Potenz  $c = a^b$  berechnet.  $a$  wird in R1 übergeben,  $b$  in R2.  $c$  soll in R3 zurückgegeben werden.
11. Schreiben Sie ein DLX-Programm, das den Binomialkoeffizienten  $b = \binom{n}{k} = \frac{n}{1} \frac{(n-1)}{2} \dots \frac{(n-k+1)}{k}$  berechnet. Halten Sie sich am besten an folgendes Gerüst und folgenden Algorithmus:

```

        .data
n:      .word   10
k:      .word   3
b:      .word   0

        .text
main:   ; Programm folgt hier

                                b = 1;
                                i = 1;
                                while (i <= k) {
                                    b = b * n / i;
                                    n = n - 1;
                                    i = i + 1;
                                }

```

12. 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; }

```