

## ÜBUNGEN ZU ORGANISATION UND ARCHITEKTUR VON RECHNERN SS 2002

### SERIE 4

#### Aufgabe 12

(10 Punkte)

Machen Sie sich mit dem Simulator `bsvc` sowie dem Assembler `68kasm` für den MC68000 vertraut. Schreiben Sie dazu ein Assembler-Programm, das die Fakultät von 6 berechnet. Realisieren Sie die Fakultät als eine Unterfunktion mit dem Einsprung-Label `FAC`. Das zu übergebende Argument sowie das Resultat der Berechnung sollen jeweils im Register `D0` übergeben werden. Folgender Auszug könnte also als Programmgerüst dienen:

```
                ORG     $0
                DC.L   $8000           Stack pointer value after a reset
                DC.L   START          Program counter value after a reset

START:         MOVE.L #6,D0           argument value
                JSR   FAC             call factorial
                BREAK                now, D0 should hold fac( 6) = $2d0
```

\*\*\*\*\*

\*

\* function: D0 FAC( D0)

\*

\* description:

\* computes the factorial of the number given in D0 and returns  
\* the result in D0 as well.

\*

\* registers affected: D0, ...

\*

```
FAC:           * here the implementation of the
                * factorial function is to be filled in.
```

```
                RTS
```

**Hinweis:** verwenden Sie zunächst die unter den Beispieldateien für den MC68000 befindliche Setup-Datei `simple.setup`.

#### Aufgabe 13

(10 Punkte)

Stellen Sie fest, für welche Argumentwerte Ihre Implementierung korrekte Resultate liefert.

Falls die Berechnung der Fakultät zu einem Resultatsüberlauf führt, sollte Ihre Implementierung den Wert 0 zurückliefern, um den Überlauf erkennbar zu machen.

Sorgen Sie dafür, daß Ihre Fakultätsfunktion auch für das Argument 12 das korrekte Resultat `$1C8CFC00` liefert. Definieren Sie dazu eine Hilfsfunktion `MULU_L`, die zwei Long-Formate in `D0` und `D1` erhält, miteinander multipliziert und das 2 Long-Formate umfassende Resultat in `D2` und `D3` zurückliefert.

**Abgabe:** Di., 3.06.2002 in den Übungen.