Institut für Informatik
und Praktische Mathematik
der Universität Kiel

Kiel, 29.05.2002

Übungen zu Organisation und Architektur von Rechnern
SS 2002
Serie 5

Aufgabe 14 (10 Punkte)

Schreiben Sie eine rekursive Fakultätsfunktion, die folgende Aufrufskonventionen für Funktionsaufrufe verwendet:

- Alle Argumente der Funktion werden vor dem eigentlichen Funktionsaufruf auf den Laufzeit-Stack gelegt.
- Nach dem Funktionsaufruf sind alle Argumente vom Laufzeit-Stack entfernt worden. Stattdessen befindet sich das Resultat der Funktion als oberster Eintrag auf dem Laufzeit-Stack.
- Alle Register haben nach dem Funktionsaufruf den gleichen Wert wie vor dem Funktionsaufruf. **Hinweis:** Lagern Sie die Werte der in der Funktion benötigten Register auf dem Laufzeit-Stack zwischen.

Entsprechend obiger Konventionen läßt sich damit der initiale Aufruf der Fakultätsfunktion folgendermaßen spezifizieren:

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

START:          MOVE.L  #6,-(SP)        push the argument on the stack
                JSR     FAC             call factorial
                MOVE.L  (SP)+,D0        pop the result from the stack
                BREAK

*******************************************************************************
*
* function:   (SP) FAC( (SP))
*
* description:
*   computes the factorial of the number that was top of the runtime stack
*   right before calling the function. The result is returned via the very
*   same stack location.
*
* registers affected: none
*

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

Aufgabe 15 (10 Punkte)

Implementieren Sie eine leicht modifizierte Version der in der Vorlesung vorgestellten "Roller-Coaster"-Funktion:

```
letrec
  f = lambda u v n
             letrec
               g = lambda w z n
                   (if (EQ n 0)
                       (+ w z)
                       (if (GT u w)
                           (g (- 1 u) z (- 1 n))
                           (f v (+ 1 w) (- 1 n))))
             in (g v u (- 1 n))
in (f 1 2 20)
```

Realisieren Sie zwei entsprechende Funktionen `F` und `G` in MC68000 Assembler. Übergeben Sie sämtliche Parameter und Resultate über den Laufzeitstack gemäß der in Aufgabe 14 erläuterten Aufrufkonvention. Erweitern Sie die Aufrufkonvention dahingehend, daß Zugriffe auf relativ freie Variablen (hier `u` und `v` in `G`) direkt im Original-Frame der übergeordneten Funktion (hier `F`) stattfinden können. Verwenden Sie dabei die in der Vorlesung u.a. im Zusammenhang mit der SECDH-2 Maschine erläuterte Technik der Verkettung von Environments.

Kommentieren Sie **ausführlich**, wie Sie diesen Verkettungsmechanismus im allgemeinen organisieren und erläutern Sie speziell an dem Roller-Coaster Beispiel, wie sich der Laufzeitstack im Laufe der Berechnung entwickelt.

Abgabe: Di., 10.06.2002 in den Übungen.