Institut für Informatik
und Praktische Mathematik
der Universität Kiel

Kiel, 17.06.2002

Klausur Organisation und Architektur von Rechnern
SS 2002

Aufgabe 1 (4 Punkte)

Schreiben sie ein Assembler-Programm, das das folgende Pseudo-Scheme-Programm für Zahlen in vorzeichenbehafteter 16-Bit(!) Darstellung implementiert:

```
letrec
  f = lambda x y (* (+ x y) (- x y))
in (f 3 4)
```

Dabei ist zu beachten, daß

- die Argumentwerte für die Parameter `x` und `y` (als 16-Bit Werte!) über den Laufzeitstack übergeben werden;
- nach der Beendigung eines Aufrufes der Funktion `f` anstelle der beiden Argumente nur noch das Resultat der Berechnung (als 32-Bit Wert!) auf dem Laufzeitstack liegt;
- sämtliche Register nach Beendigung eine Aufrufes von `f` die gleichen Werte haben wie zu Beginn des Aufrufes.

Hinweis: Eine Überprüfung auf Resultatüberläufe ist nicht erforderlich.

Aufgabe 2 (8 Punkte)

Schreiben Sie ein Assembler-Programm, das folgendes Pseudo-Scheme-Programm implementiert:

```
letrec
  f = lambda u v w letrec
                     g = lambda n if ( eq n 0)
                                  ( + u ( + v w))
                                  ( g ( - n 1))
                   in (g u)
in (f 10 2 3)
```

Halten Sie die aus der Vorlesung bekannten Aufrufkonventionen für verschachtelte Funktionen ein, d.h.

- Argumente und Resultate werden ausschliesslich über den Laufzeitstack übergeben; die Stackeinträge sind entsprechend der Schachtelung der Funktionsdefinitionen ordnungsgemäß zu verketten;
- Die aufgerufene Funktion entfernt Ihre Argumentwerte vom Stack und ersetzt sie durch das Resultat der Berechnung;
- sämtliche Register haben nach Beendigung eines Funktionsaufrufes die gleichen Werte, wie zu dessen Beginn;
- Die aufrufende Funktion stellt sicher, daß das Register `A6` auf das Environment-Frame zeigt, das die Belegungen der relativ freien Variablen der aufgerufenen Funktion enthält.

Aufgabe 3 (8 Punkte)

Es sind zwei Assemblerprogramme zu entwickeln, die eine als pre-order linearisierte Zeichenkette auf einem Quell-Stack vorliegende binäre Baumstruktur auf einen Ziel-Stack traversiert und dort in rechts-links transponierter pre-order linearisierter Form aufbaut. Die Blattknoten des Baumes seien durch nicht-negative ganze Zahlen gegeben, und für die inneren Knoten wird das Symbol Δ benutzt.

Der Traversiervorgang überführt dann beispielsweise die Zeichenkette

$$\Delta\ 1\ \Delta\ \Delta\ 2\ 3\ 4 \qquad | \quad \text{Quell-Stack}$$

in

$$\Delta\ \Delta\ 4\ \Delta\ 3\ 2\ 1 \qquad | \quad \text{Ziel-Stack}$$

Unter Verwendung von 32-Bit Darstellungen für die Zahlenwerte sowie des Wertes `-1` als 32-Bit Zweierkomplement für die inneren Knoten ergibt sich für die Zeichenkette auf dem Quell-Stack die folgende hexadezimal codierte Byte-Folge:

```
                $ff $ff $ff $ff $00 $00 $00 $01 $ff $ff $ff $ff
$ff $ff $ff $ff $00 $00 $00 $02 $00 $00 $00 $03 $00 $00 $00 $04 | Stack-Bottom
```

Schreiben Sie im Assembler

a) eine Funktion `TRANS`, die diesen Traversiervorgang, wie in der Vorlesung beschrieben, unter Verwendung eines weiteren Hilfs-Stack für die Zwischenlagerung der inneren Knotensymbole Δ realisiert.

   Die Funktion `TRANS` verwendet als Argumente Zeiger auf die Spitzen des Quell- und des Ziel-Stacks, und sie gibt als Resultate die veränderten Stack-Top-Zeiger zurück.

   Hinweis: Wie in der Datei `aufgabe3.s` bereits vorgegeben, können die drei Stacks als globale Objekte fester Größe verwendet werden. Tests auf Stack-Überlauf oder Stack-Unterlauf dürfen unterbleiben.

b) eine rekursive Funktion `TRANS_REC`, die das Traversieren ohne Benutzung eines Hilfs-Stack für die inneren Knotensymbole Δ realisiert.

Hinweis: Für alle drei Aufgaben finden sich im Verzeichnis `klausur` Musterdateien. Verwenden Sie diese und ergänzen Sie die fehlenden Teile. Wenn Sie Ihre Lösung abgeben wollen, führen Sie bitte im Verzeichnis `klausur` den Befehl `make abgabe` aus. Dies führt dazu, daß im Raum 502 ein Ausdruck erstellt wird. Nachdem Sie diesen Ausdruck **unterschrieben** haben, können Sie die Klausur bei einem der Betreuer in Raum 502 abgeben.

Viel Erfolg !!