

Betriebssysteme im Wintersemester 2008/2009

Übungsblatt 2

Abgabetermin: 22.10.2008, 13:30 Uhr

Achtung: Zu jeder Aufgabe des Typs H kann eine Datei abgegeben werden, deren Name eindeutig die bearbeitete Aufgabe kennzeichnet (z.B. *Aufgabe-5.txt*). Erlaubt sind Abgaben in den Formaten .pdf, .jpg, .txt, .png und .gif. Stellen Sie die Dateien in ein extra Verzeichnis (mit beliebigem Namen) und packen Sie dieses zu einem ZIP-Archiv. Geben Sie dieses per UniWorx ab.

Aufgabe 5: (H/T) Unterprogramme

(8+8 Pkt.)

Achtung: Nur die Teilaufgaben a. bis f. zählen als Hausaufgabe und sind schriftlich zu lösen. Teilaufgabe g. ist keine Hausaufgabe, sondern eine Tutoraufgabe.

- a. (H) Nennen Sie einige Nachteile, die sich ergeben, wenn ein Programmierer ausschließlich offene Unterprogramme (siehe Skript) verwendet.
- b. (H) Offensichtlich ist es sehr ineffizient, auf geschlossene Unterprogramme (Prozeduren) ganz zu verzichten. Bei besonders kleinen Unterprogrammen ist es aber wiederum ungünstig, diese als geschlossene Unterprogramme zu implementieren. Warum?
- c. (H) Welche Arten der Parameterübergabe gibt es?
- d. (H) Wie werden Sprünge innerhalb eines Programms technisch realisiert?
- e. (H) Was ist der wesentliche Unterschied zwischen Sprüngen, die mit den Befehlen `JMP` und `CALL` eingeleitet werden?
- f. (H) Welche zwei wesentlichen Möglichkeiten gibt es, den `RET`-Befehl zu implementieren und woraus ergeben sie sich?
- g. (T) Nehmen Sie an, Sie verfügen über ein System mit einem Hauptspeicher der Größe 1 KB (= 1024 Bytes). Weil jede Speicherzelle 1 Byte breit ist, stehen Ihnen 1024 Speicherzellen (Adressen 0 bis 1023) zur Verfügung. Außerdem stehen ein Kellerspeicher (Stack) mit 16 Speicherplätzen (0 bis 15, Keller wird beginnend mit Adresse 0 gefüllt und wächst nach oben) und eine CPU mit diesen sechs Registern zur Verfügung:
 - PC (Program Counter): enthält die Adresse der Speicherzelle des nächsten auszuführenden Befehls (also einen Wert zwischen 0 und 1023)
 - RA (Return Address): zur Sicherung der Rücksprungsadresse bei einem Unterprogrammaufruf
 - SP (Stack Pointer): enthält die Nummer der obersten belegten Kellerzelle (also einen Wert zwischen 0 und 15)

- R1, R2 und R3: beliebige Register (z.B. zur Speicherung von Berechnungsergebnissen); alle drei Register sind Caller-saved

Geben Sie unter diesen Annahmen das Schema eines beispielhaften Unterprogrammaufrufs an. Beachten Sie dabei:

- Dokumentieren Sie auch jede Veränderung der CPU-Register-Inhalte.
- Das Hauptprogramm hat bei seinen bisherigen Berechnungen ins Register R1 den Wert 23 und ins Register R2 den Wert 42 abgelegt. Garantieren Sie, dass diese Daten dem Hauptprogramm auch nach dem Unterprogrammaufruf noch zur Verfügung stehen – selbst wenn das Unterprogramm die beiden Register für lokale Berechnungen überschreibt.
- Der CALL-Befehl des Hauptprogramms belegt die Speicherzelle mit der Adresse 410.
- Das Unterprogramm beginnt an der Adresse 770.
- Das Unterprogramm erwartet zwei Aufrufparameter (Argumente) und liefert einen Rückgabewert (Ergebnis).

Aufgabe 6: (T) Offene vs. geschlossene Unterprogramme (8 Pkt.)

Wenn sich in einem Programm eine Folge von Befehlen während der Ausführung (u.U. mehrfach) wiederholt, können diese Befehle zu einem Block zusammengefasst und in ein entsprechendes Unterprogramm ausgelagert werden. Dazu haben Sie in der Vorlesung die Konzepte von geschlossenen und offenen Unterprogrammen kennengelernt.

Das folgende Programm nutzt beide Arten: Geschlossene Unterprogramme liegen in Form definierter Funktionen vor. Jede Anwendung einer binären arithmetischen Operation (z.B. Addition, Multiplikation, ...) ist dagegen als offenes Unterprogramm zu betrachten, sofern sie nicht in eine Funktion ausgelagert ist, die **allein** diese Operation auf den ihr übergebenen Argumenten durchführt und das Ergebnis zurückgibt.

Das Shell-Skript berechnet den Binomialkoeffizienten der ganzen Zahlen n und k . Dabei werden n und k mit den ersten beiden Parametern belegt, mit denen das Skript aufgerufen wird. Die Formel für die Berechnung des Binomialkoeffizienten lautet:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

```

#!/bin/sh
2 if test $# = 2 ; then
    echo "Argumente_OK."
4 else
    echo "Aufruf_mit_zwei_Parametern:_n_k"
6 fi

8 fakultaet(){
    declare -i zahl=$arg
    declare -i fakultaet=1
10 while [ $zahl -gt 1 ]; do
12     fakultaet=$fakultaet*$zahl
        zahl=$zahl-1
14 done
    erg=$fakultaet
16 }

18 arg=$1
    fakultaet
20 nfak=$erg
    echo "n!:_$nfak"
22
    arg=$2
24 fakultaet
    kfak=$erg
26 echo "k!:_$kfak"

28 arg=$1-$2
    fakultaet
30 nminkfak=$erg
    echo "(n-k)!:_$nminkfak"
32
    declare -i nenner=$kfak*$nminkfak
34 declare -i ergebnis=`expr $nfak / $nenner`
    echo Ergebnis: $ergebnis

```

- a. Schreiben Sie das Programm so um, dass es **ausschließlich offene Unterprogramme** enthält. D.h. im gesamten Skript dürfen keine Funktionsdefinitionen (bzw. Aufrufe extern definierter Funktionen) vorkommen.
- b. Schreiben Sie das Programm so um, dass es **ausschließlich geschlossene Unterprogramme** enthält. Das bedeutet (gemäß obiger Festlegung), dass z.B. das Symbol + nur ein einziges Mal im gesamten Skript vorkommen darf – und zwar im Rumpf der Funktion, die die binäre Addition definiert (z.B. add()). Weder im Hauptprogramm, noch in anderen Unterprogrammen dürfen arithmetische Symbole verwendet werden, sondern zur Ausführung einer arithmetischen Operation muss ein entsprechender Funktionsaufruf erfolgen.

Aufgabe 7: (T) Multiprogramming

(4 Pkt.)

- a. Was versteht man unter Multiprogramming?
- b. Was ist der Hauptvorteil von Multiprogramming?
- c. Was sind die wesentlichen Vor- und Nachteile eines Multiprozessor-Systems?

Aufgabe 8: (K) Multiple Choice: Betriebssysteme

(2.5 Pkt.)

Sind die folgenden Aussagen wahr oder falsch?

- a. In der logischen Hierarchie der Rechnerfunktionen ist das Betriebssystem in die Schicht der Anwendungsprogramme einzuordnen.
- b. Das Betriebssystem kann als eine erweiterte Maschine aufgefasst werden, die aus Nutzersicht eine Komplexitätsreduktion bewirkt.
- c. Das Betriebssystem dient als Schnittstelle zwischen Hardware und Anwendungsprogrammen.
- d. Ein Betriebssystem verwaltet Speicher, E/A-Geräte, Dateien und Prozesse und muss zu diesem Zweck geeignete Kontrollstrukturen (Tabellen) bereitstellen.
- e. Bei einem verteilten Betriebssystem (Network Operating System) weiß der Benutzer um die Existenz mehrerer Rechner und kann sich gezielt auf fremden Rechnern anmelden (etwa um Daten zu kopieren).