

Klausur IA12.0/13.0/14.0 451 Assembler am 22.08.2003

Dauer : 120 Minuten

keine externen Hilfsmittel

Aufgabe 1 :

(15 Punkte)

Wie berechnet sich jeweils auf der Basis des 8086-Prozessors die 20-bittige physikalische Adresse der folgenden Speicheradressierungen ?

MOV AX,[BP+4]

MOV AX,[BP-2]

MOV AX,[BX]

MOV AX,[BX+SI]

MOV AX,[42]

Aufgabe 2 :

(45 Punkte)

Übersetze auf der Basis des 8086-Prozessors möglichst bedeutungstreu die Prozedur CBVADDSUB :

```
Type tpStruktur = ^tStruktur;
   tStruktur   = Record
               A : Integer;
               B : Integer
               End;

Procedure CBVADDSUB(X:tpStruktur;Y,Z:Integer);
Var L1,L2 : Integer;
Begin
  L1 := Y+Z;
  L2 := Y-Z;
  X^.A := L1;
  X^.B := L2
End;
```

Aufgabe 3 :

(15 Punkte)

Welche - im Vergleich zum Assembler-Programm der vorherigen Aufgabe - circa vier Befehle (an welcher Stelle) sind zusätzlich notwendig und welche circa zwei Befehle sind überflüssig zur möglichst bedeutungstreuen Übersetzung der Prozedur CBRADDSUB auf der Basis des 8086-Prozessors ?

```
Type tpStruktur = ^tStruktur;
      tStruktur  = Record
                A : Integer;
                B : Integer
                End;

Procedure CBRADDSUB(Var X:tpStruktur;Y,Z:Integer);
Var L1,L2 : Integer;
Begin
  L1 := Y+Z;
  L2 := Y-Z;
  X^.A := L1;
  X^.B := L2
End;
```

Aufgabe 4 :

(15 Punkte)

Vervollständige auf der Basis des 8086-Prozessors und des Befehls REP MOVSB das nachfolgende Programm zum Kopieren des Speicherbereichs BENNO in den Speicherbereich HAUPT :

```
DSEG  SEGMENT
BENNO DB "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
HAUPT DB 26 DUP (?)
      DB 13,10,"$"
DSEG  ENDS

CSEG  SEGMENT
      ASSUME CS:CSEG

MAIN  PROC
; <<< Anfang der Ergänzung um circa neun Befehle >>>

; <<< Ende der Ergänzung um circa neun Befehle >>>
      MOV  AH,9
      MOV  DX,OFFSET HAUPT
      INT  21h
      MOV  AX,4C00h
      INT  21h
MAIN  ENDP

CSEG  ENDS

SSEG  SEGMENT STACK
      DW 128 DUP (?)
SSEG  ENDS

      END MAIN
```

Aufgabe 5 :

(15 Punkte)

Welcher Text würde im Programm der vorherigen Aufgabe für folgende um die Zeile `ORG BENNO+13` ergänzte Datendefinition ausgegeben werden ?

```
DSEG  SEGMENT
BENNO DB "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
      ORG BENNO+13
HAUPT DB 26 DUP (?)
      DB 13,10,"$"
DSEG  ENDS
```

Welche Änderungen am Programmcode werden für die ursprüngliche Textausgabe `ABCDEFGHIJKLMNOPQRSTUVWXYZ` erforderlich ?

Aufgabe 6 :

(15 Punkte)

Vervollständige auf Basis des 8086-Prozessors und des Befehls REPNE SCASB das nachfolgende Programm zur Suche und Ausgabe der Position des Buchstabens X in der Zeichenkette ALPHABET :

```
DSEG      SEGMENT
ALPHABET DB "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
DSEG      ENDS
CSEG      SEGMENT
          ASSUME CS:CSEG
MAIN      PROC
; <<< Anfang der Ergänzung um circa acht Befehle >>>

; <<< Ende der Ergänzung um circa acht Befehle >>>
MISSERFOLG:
          MOV     AX,CS
          MOV     DS,AX
          MOV     DX,OFFSET SCHADE
          MOV     AH,9
          INT     21h
          JMP     ENDE
ERFOLG:
; <<< Anfang der Ergänzung um circa zwei Befehle >>>

; <<< Ende der Ergänzung um circa zwei Befehle >>>
          MOV     BX,10
          MOV     CX,0
DIVI:     MOV     DX,0
          DIV     BX
          PUSH    DX
          INC     CX
          CMP     AX,0
          JNZ     DIVI
POPI:     POP     DX
          ADD     DL,'0'
          MOV     AH,2
          INT     21h
          LOOP    POPI
ENDE:     MOV     AX,4C00h
          INT     21h
MAIN      ENDP
SCHADE   DB "Das gesuchte Zeichen wurde nicht gefunden :-( $"
CSEG      ENDS
SSEG      SEGMENT STACK
          DW     128 DUP (?)
SSEG      ENDS
          END     MAIN
```

**Folgende Tipps mögen bei der Bearbeitung der
Aufgaben 2 & 3 hilfreich sein :**

- Gemäß den Konventionen von Borland TurboPascal soll die Parameterübergabe (sowohl für Wert- als auch Referenzparameter) über den Stack und die Ablage lokaler Variablen auf dem Stack erfolgen.
- Bei der Übergabe einer Variablen per Referenzaufruf wird die Adresse der Variablen in der Reihenfolge Segment / Offset - Anteil auf den Stack gepusht.
- Eine Pointer-Variable enthält die Adresse eines Wertes in der Reihenfolge Offset / Segment - Anteil.
- Bei der Übergabe einer Pointer-Variablen per Wertaufruf wird der Inhalt in der Reihenfolge Segment / Offset - Anteil auf den Stack gepusht.
- Bei der Übergabe einer Pointer-Variablen per Referenzaufruf wird quasi die Adresse der Adresse eines Wertes übergeben.

Das PTL-Team wünscht viel Erfolg

Anlage : Befehlssatz des 8088 & Stringverarbeitung