Klausur IA12.0/13.0/14.0 451 Assembler am 13.02.2004

Dauer : 120 Minuten **keine externen Hilfsmittel**

Aufgabe 1 :

Entwickle ein vollständiges Assemblerprogramm (8086,EXE) zur Berechnung der Summe S aller Elemente eines Vektors. Die Eingabe des Vektors und die Ausgabe der Summe ist *nicht* gefordert.

$(a_1 , a_2 , ... , a_n) \Rightarrow S = a_1 + a_2 + ... + a_n$

Alle Vektorelemente seien vorzeichenbehaftete ganze Zahlen, und die Anzahl der Vektorelemente sei in der Konstanten N definiert.

Aufgabe 2 :

Übersetze das nachfolgende TurboPascal-Programm *Aufgabe2* in ein äquivalentes (d.h. möglichst bedeutungstreues) Assemblerprogramm (8086, EXE).

Gemäß den Konventionen von Borland TurboPascal soll die Parameterübergabe (sowohl für Wert- als auch Referenzparameter) über den Stack, die Ablage lokaler Variablen auf dem Stack und die Rückgabe eines Integer-Funktionswerts über das AX-Register erfolgen. Bedenke : Referenzparameter bestehen aus einem 16-Bit-Segment- und 16-Bit-Offsetanteil.

```
Program Aufgabe2;

Var Y, Z : Integer;

Function X (Var A : Integer; B : Integer) : Integer;

Var C : Integer;

Begin
  C := A + B;
  A := C;
  X := C
End;

Begin
  Y := 5;
  Z := X (Y, -1)
End.
```

Aufgabe 3 :

Übersetze die Prozedur *X* des TurboPascal-Programms *Aufg3a* (AUFG3A.PAS) in ein äquivalentes und separates 8086-Assembler-Unterprogramm (X.ASM) :

```
Program Aufg3a;

Var Y,Z : Integer;

Procedure X(A:Integer;Var B:Integer);

Var C : Integer;

Begin
  If A = 0 Then
    B := A
  Else Begin
    X(A-1,B);
    B := B+A
  End
End;

Begin
  For Y := 1 To 10 Do Begin
    X(Y,Z);
    Writeln(Y,',',Z)
  End
End.
```

Ziel ist die Einbindung des Objektcodes des 8086-Assembler-Unterprogramms (X.OBJ) in das TurboPascal-Programm *Aufg3b* (AUFG3B.PAS) mittels der Compiler-Direktive {$L X} :

```
Program Aufg3b;

Var Y,Z : Integer;

Procedure X(A:Integer;Var B:Integer);
{$L X}  { X.OBJ, d.h. Objektcode }

Begin
  For Y := 1 To 10 Do Begin
    X(Y,Z);
    Writeln(Y,',',Z)
  End
End.
```

Gemäß den Konventionen von Borland TurboPascal erfolgt die Parameterübergabe (sowohl für Wert- als auch Referenzparameter) über den Stack und die Ablage lokaler Variablen auf dem Stack. Bedenke : Referenzparameter bestehen aus einem 16-Bit-Segment- und 16-Bit-Offsetanteil.

Aufgabe 4 :

Übersetze das nachfolgende TurboPascal-Programm in ein äquivalentes (d.h. möglichst bedeutungstreues) Assemblerprogramm (8086, EXE).

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

Var Struktur : tpStruktur;

Procedure Aufgabe4(X:tpStruktur;Y,Z:Word);
Begin
  X^.A := Y;
  X^.B := Z
End;

Begin
  New(Struktur);
  Aufgabe4(Struktur,8086,42);
  WriteLn(Struktur^.A);
  WriteLn(Struktur^.B);
  Dispose(Struktur)
End.
```

Eine Pointer-Variable enthält die Adresse eines Wertes in der Reihenfolge Offset / Segment - Anteil. Gemäß den Konventionen von Borland TurboPascal soll die Parameterübergabe über den Stack erfolgen. Bei der Übergabe einer Pointer-Variablen per Wertaufruf wird der Inhalt in der Reihenfolge Segment / Offset - Anteil auf den Stack gepusht.

Zur Realisierung der New-Prozedur und der Dispose-Prozedur werden die DOS-Funktionen "Allocate Memory" (Funktionscode 48h) und "Free Memory" (Funktionscode 49h) verwendet. Die Funktion zum Reservieren von Speicherplatz erwartet im BX-Register die Anzahl der benötigten zusammenhängenden 16-Byte Blöcke und liefert als Ergebnis im AX-Register die Segmentadresse der reservierten Blöcke (als Offsetadresse innerhalb des Segments wird Null impliziert). Der Fall "nicht genügend zur Verfügung stehender Speicher" bleibt unberücksichtigt. Die Funktion zum Freigeben von Speicherplatz erwartet im ES-Register die Segmentadresse der freizugebenden Blöcke, also eine Adresse die von der Funktion "Allocate Memory" zurückgeliefert wurde.

Im Modul IOWORD steht die Routine WWORD zur Ausgabe von ganzzahligen vorzeichenlosen Zahlen zur Verfügung. Der Datentransfer erfolgt über das DX-Register.

Das PTL-Team wünscht viel Erfolg

Anlage : Befehlssatz des 8088