4.1 Grundlagen

- Bereiche aufeinanderfolgender Adressen werden als Segmente bezeichnet.

- Während der Ausführung eines Aktivitätsträgers müssen wenigstens die folgenden drei Informationsklassen im Arbeitspeicher bereit stehen:

  - Ein Befehlssegment, das die auszuführenden Befehle enthält.

  - Ein Datensegment, das die zu bearbeitenden Daten enthält.

  - Ein Stapelsegment (*stack*), das zu organisatorischen Zwecken herangezogen wird, z. B. zur Organisation von Prozedur- und Methodenaufrufen.

- Programme werden häufig aus Gründen der Übersichtlichkeit und der Aufteilbarkeit bei der Implementierung aus mehreren Teilprogrammen (Modulen, Objekten) zusammengesetzt.
  - Wie und wann werden die Speicheradressen in den Befehlen ermittelt, ein Vorgang der als **Binden** bezeichnet wird?
    Drei wesentliche Möglichkeiten:
    - Während des Compilierens: **absolutes Binden**

      Setzt voraus, daß bereits während des Compilierens die endgültige Lage im Arbeitsspeicher bekannt ist.

      Wird im wesentlichen nur bei eingebetteten Systemen (*embedded systems*) verwendet.

    - Während des Ladens in den Arbeitsspeicher:
      statisches Binden

      Setzt voraus, daß Compiler und Assembler Informationen mitliefern, die eine Anpassung an die endgültige Lage im Arbeitsspeicher erlauben (*relocation information*).

    - Während des Programmablaufs:
      dynamisches Binden

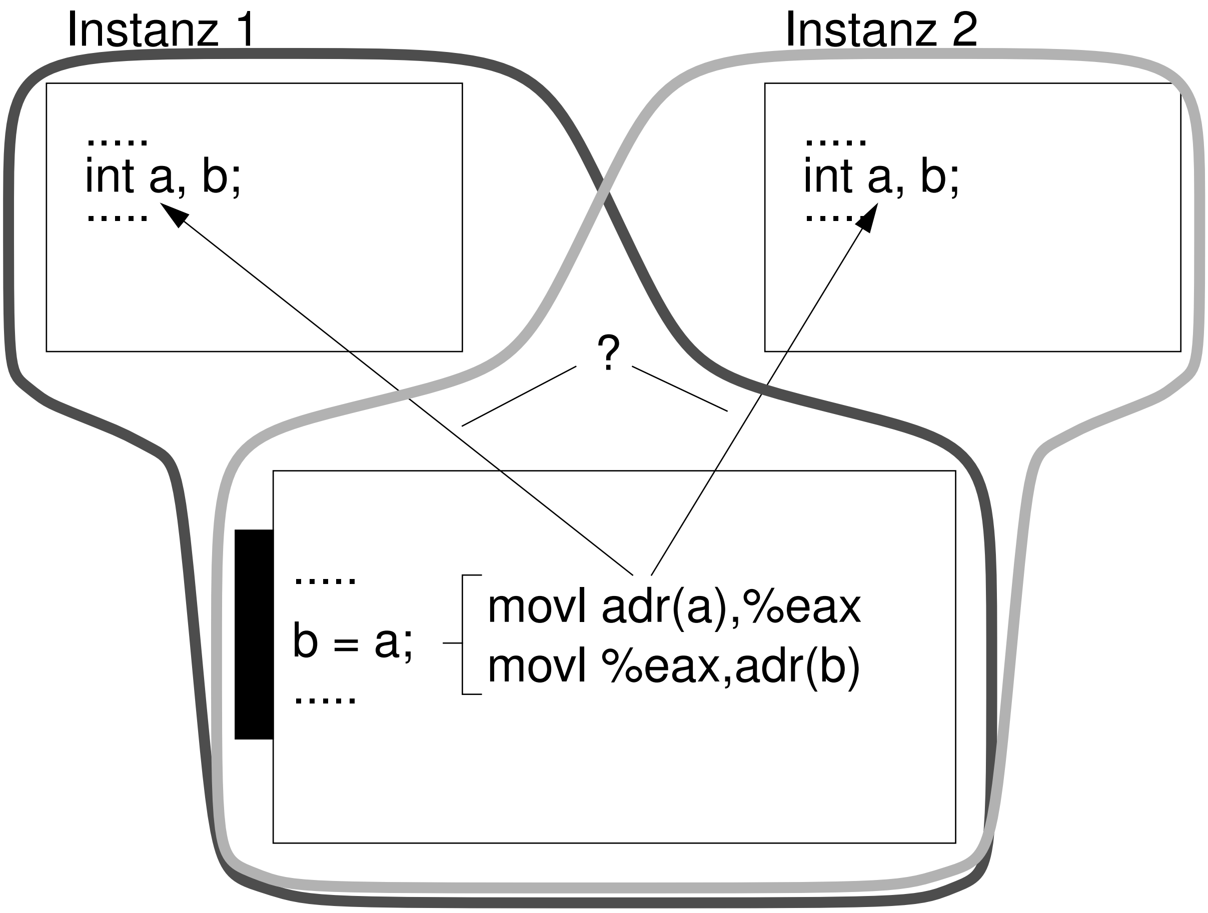
      Erlaubt mehrfache Nutzung von Befehlssegmenten.

      Realisierbar durch Verwendung:

      indizierter Adressierung

      geeigneter, noch zu besprechender Hardwaremechanismen, die auf der Assemblerebene transparent sind.

Bindung Befehlsadressen - Datensegment mittels Indizierung bei objektorientierter Programmierung



Mögliche (wenn auch unbefriedigende) Lösung:

```
                         basis:
  0000 00000000          a:        .long   0
  0004 00000000          b:        .long   0

                         start:
  0008 8D 2D 00000000      leal basis,%ebp
  000e 8B 85 00000000      movl a-basis(%ebp),%eax
  0014 89 85 04000000      movl %eax,b-basis(%ebp)
```