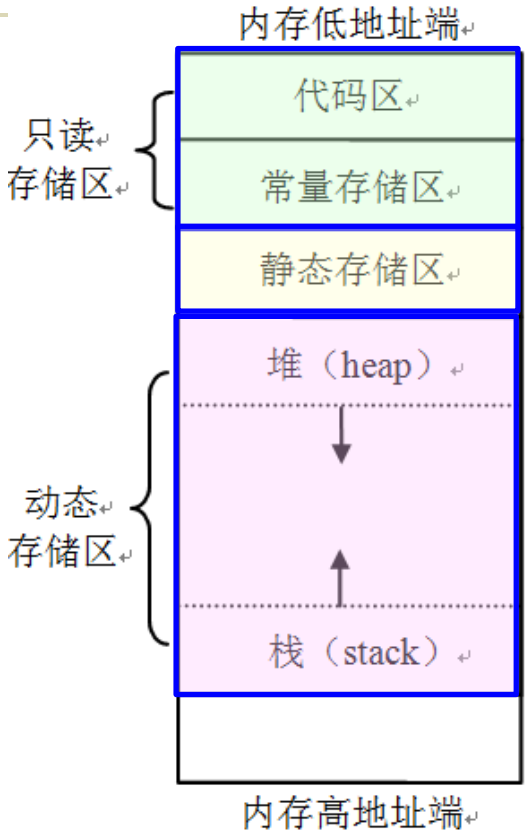
第11章 动态数据结构的C语言实现

内存映像

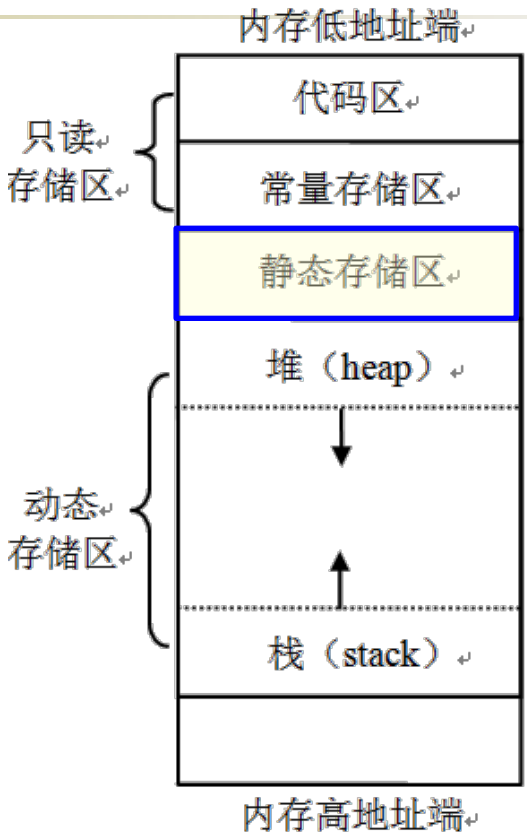
C程序的内存映像

* C程序中变量的内存分配方式



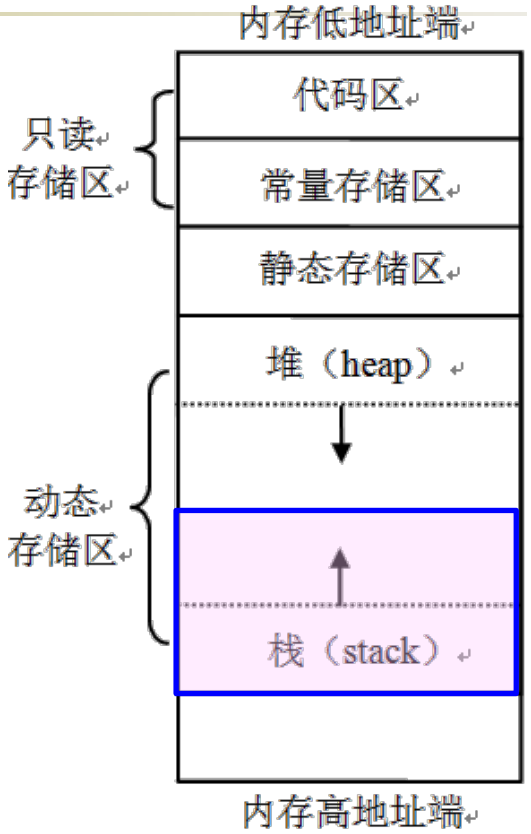
C程序的内存映像

* C程序中变量的内存分配方式
  * 从静态存储区分配
    - 全局变量和静态变量
  *



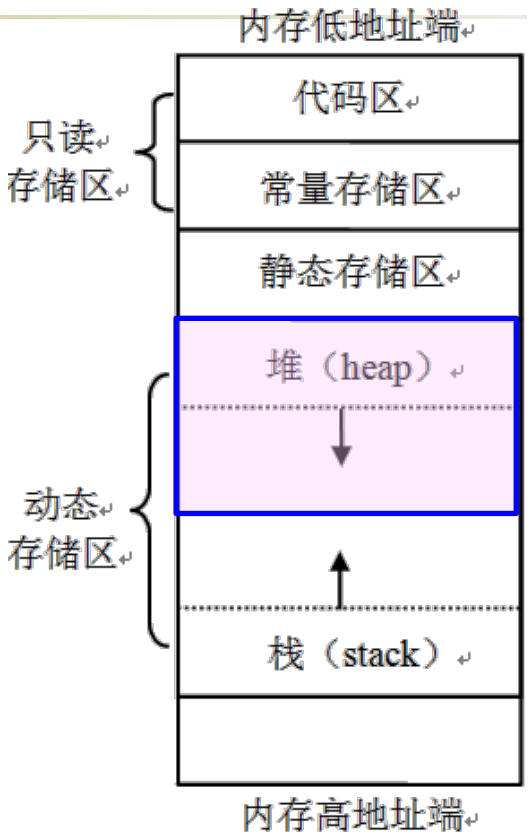
C程序的内存映像

* C程序中变量的内存分配方式
  * 从静态存储区分配
    - 全局变量和静态变量
  * 在栈上分配
    - 存放函数参数值，局部变量值等
    - 在执行函数调用时，系统在栈上为函数内的局部变量及形参分配内存，函数执行结束时，自动释放这些内存



C程序的内存映像

* C程序中变量的内存分配方式
  * 从静态存储区分配
    - 全局变量和静态变量
  * 在栈上分配
    - 存放函数参数值，局部变量值等
    - 在执行函数调用时，系统在栈上为函数内的局部变量及形参分配内存，函数执行结束时，自动释放这些内存
  * 从堆上分配
    - 在程序运行期间，用动态内存分配函数来申请的内存都是从堆上分配的，动态内存的生存期由程序员自己来决定

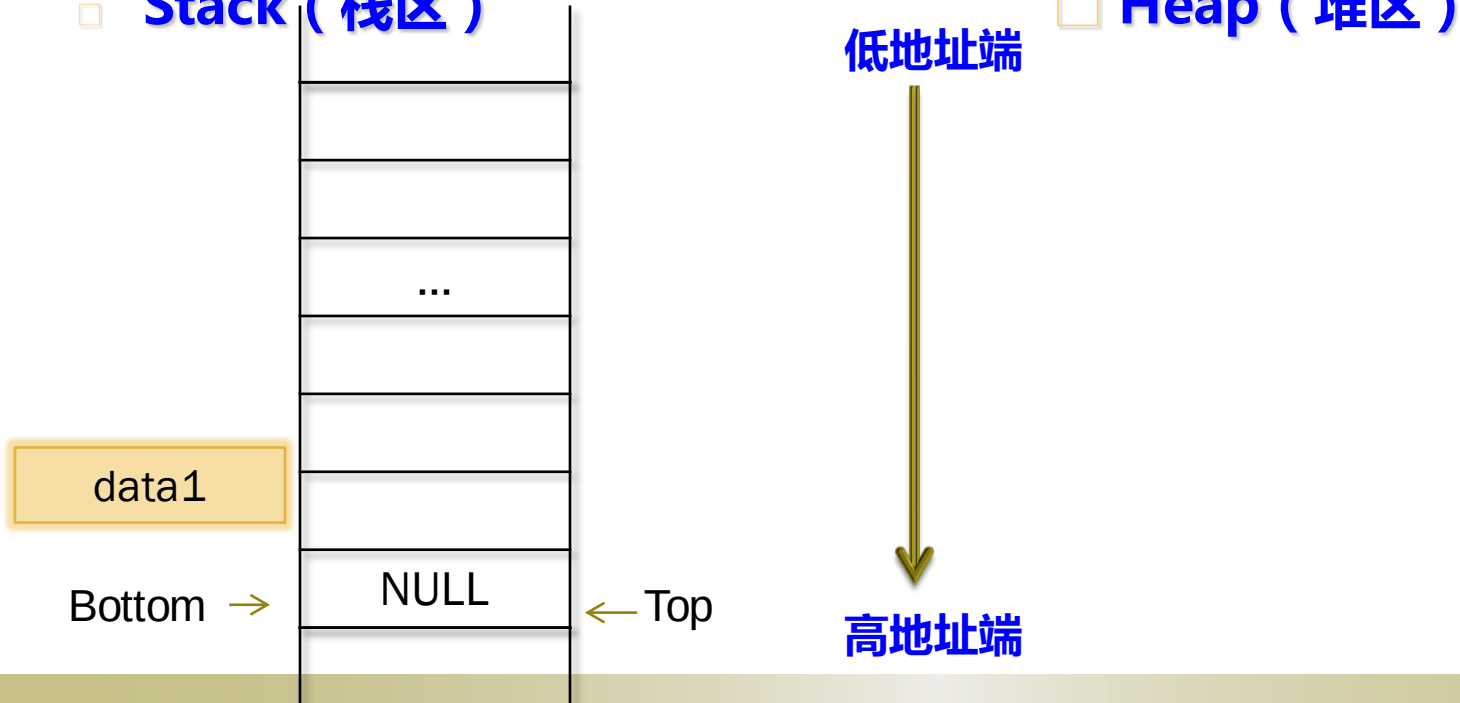


C程序的内存映像

* Stack and Heap Based Memory

Stack（栈区）

Heap（堆区）

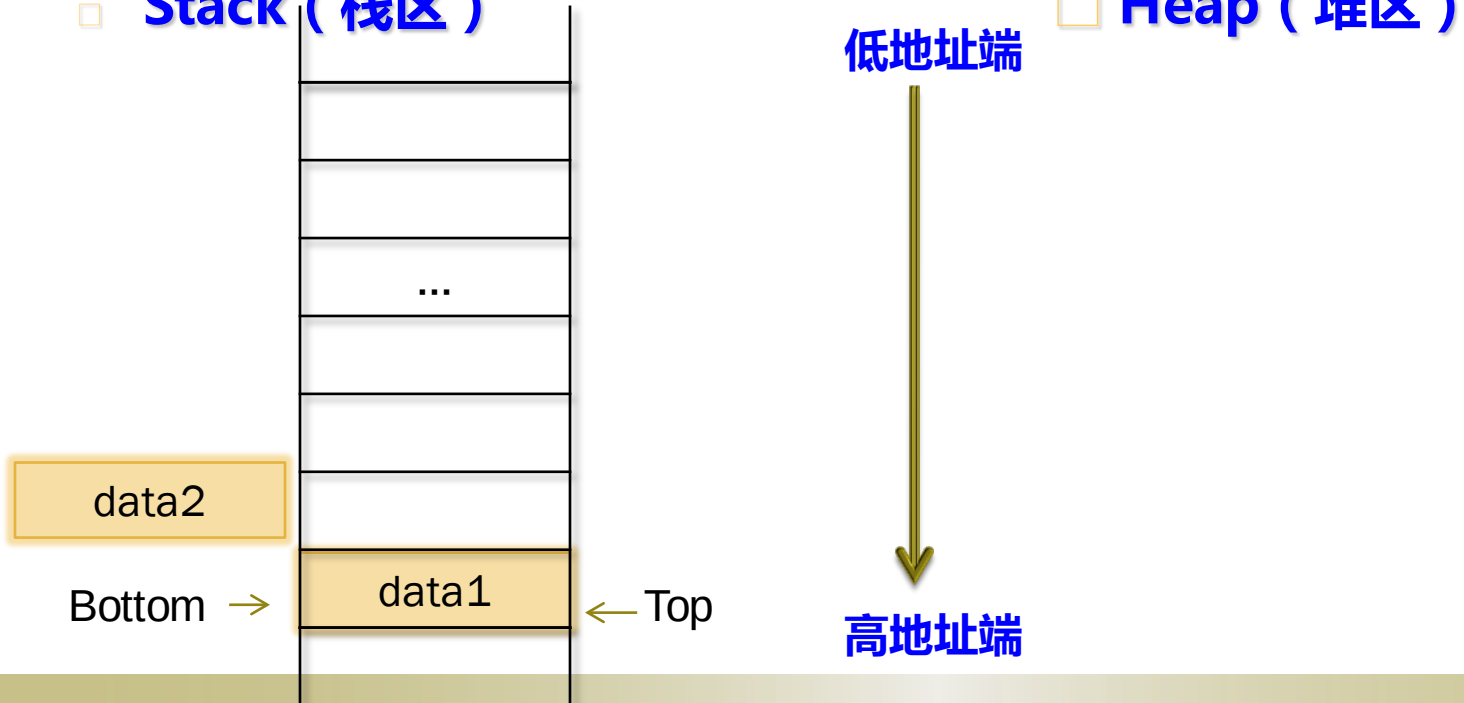


C程序的内存映像

* Stack and Heap Based Memory

Stack（栈区）

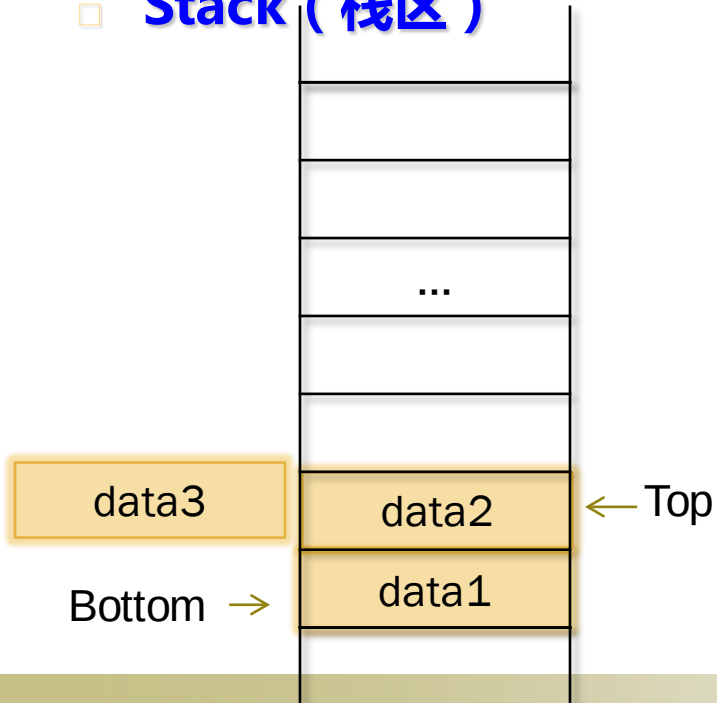
Heap（堆区）



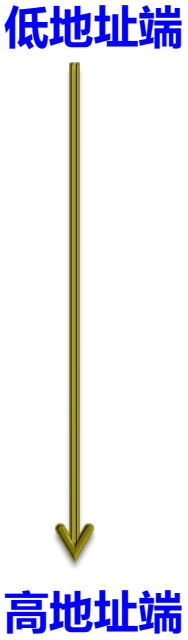
C程序的内存映像

* Stack and Heap Based Memory

Stack（栈区）



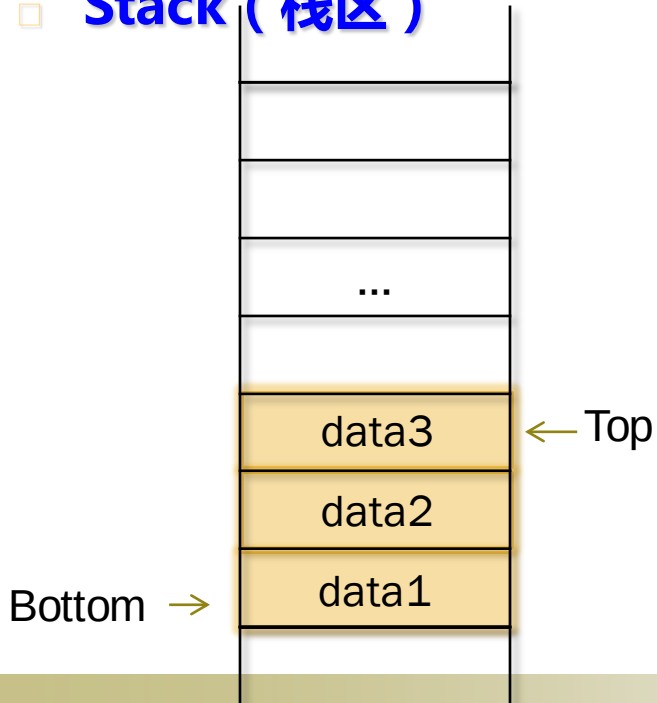
Heap（堆区）



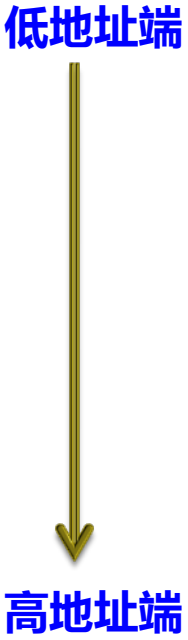
C程序的内存映像

* Stack and Heap Based Memory

Stack（栈区）



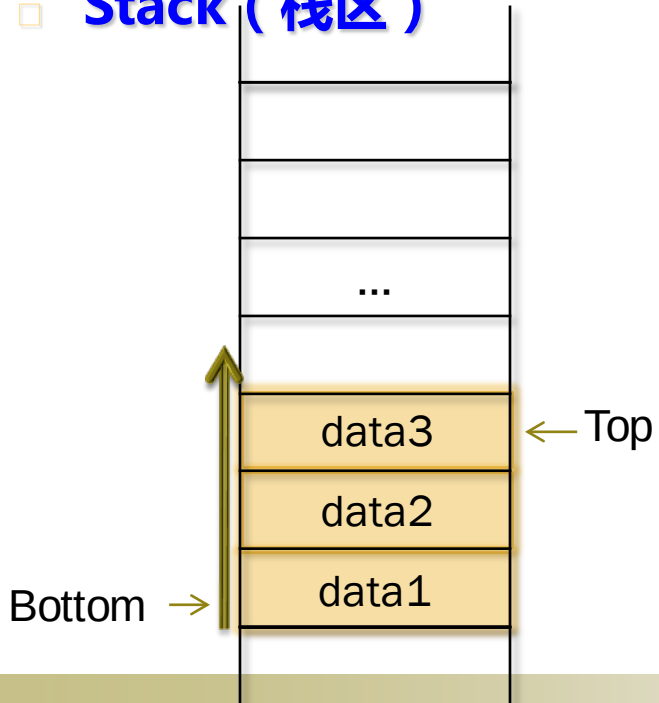
Heap（堆区）



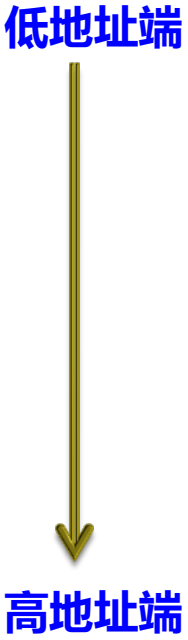
C程序的内存映像

* Stack and Heap Based Memory

Stack（栈区）



Heap（堆区）

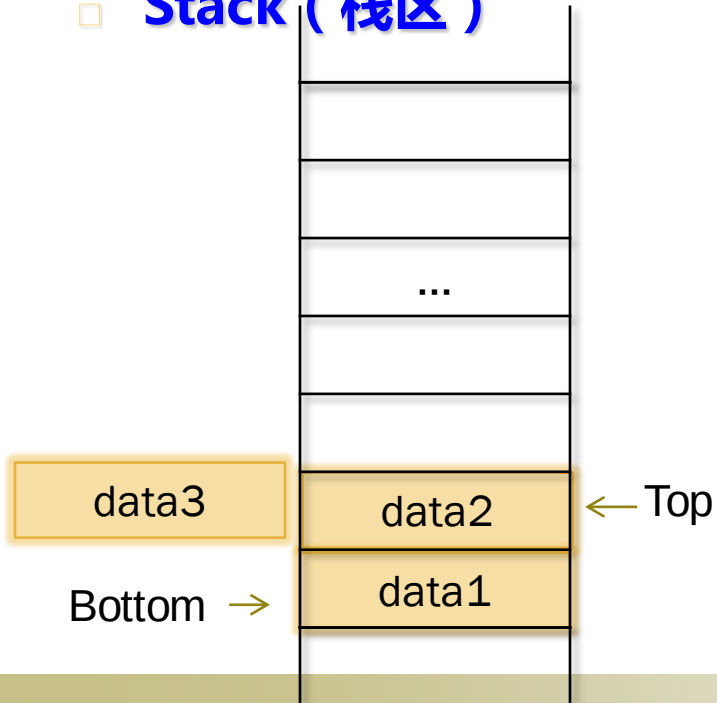


C程序的内存映像

* Stack and Heap Based Memory

Stack（栈区）

Heap（堆区）

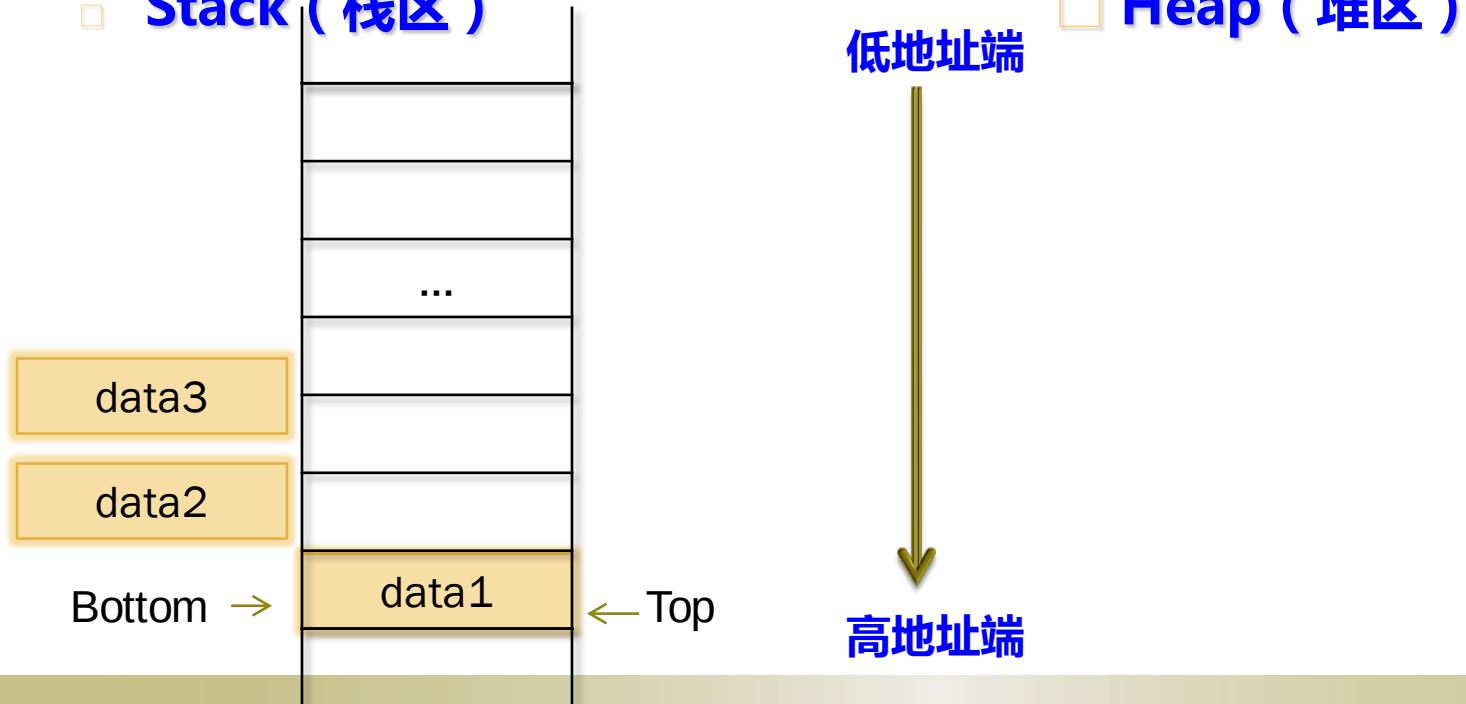


C程序的内存映像

* Stack and Heap Based Memory

Stack（栈区）

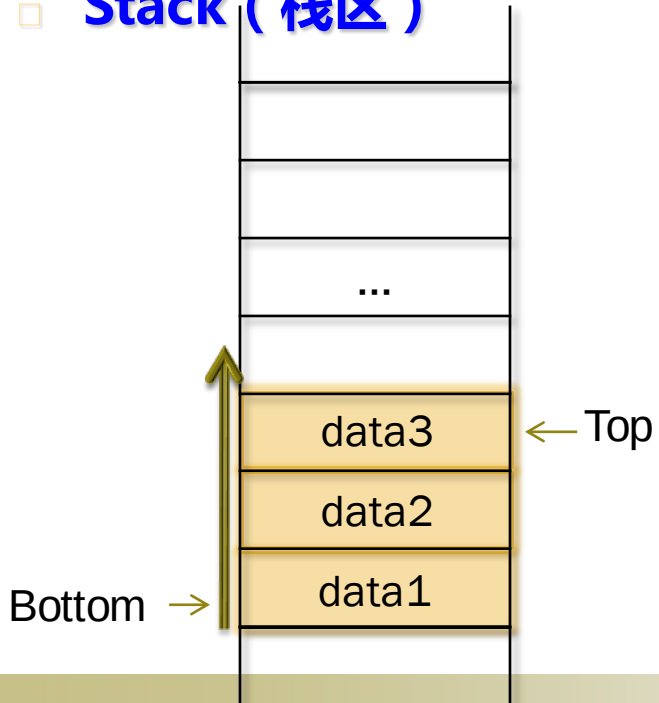
Heap（堆区）



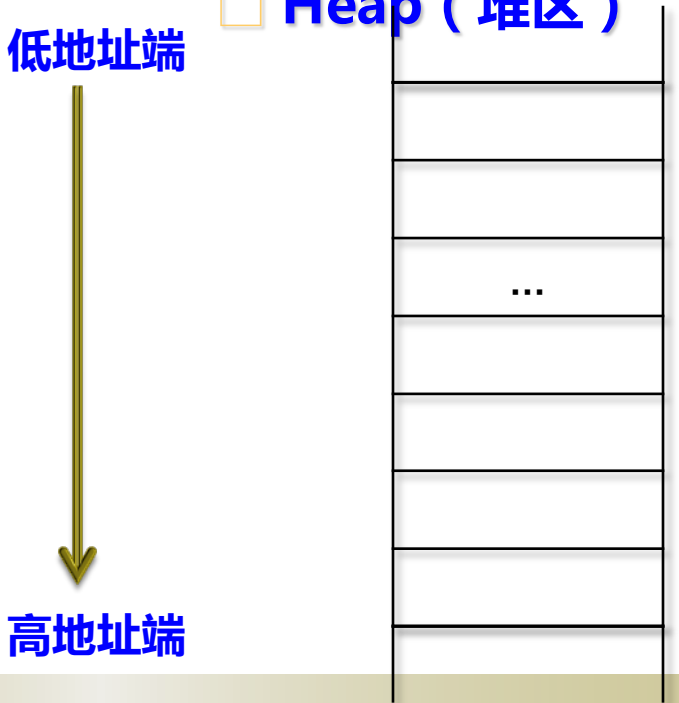
C程序的内存映像

* Stack and Heap Based Memory

Stack（栈区）



Heap（堆区）

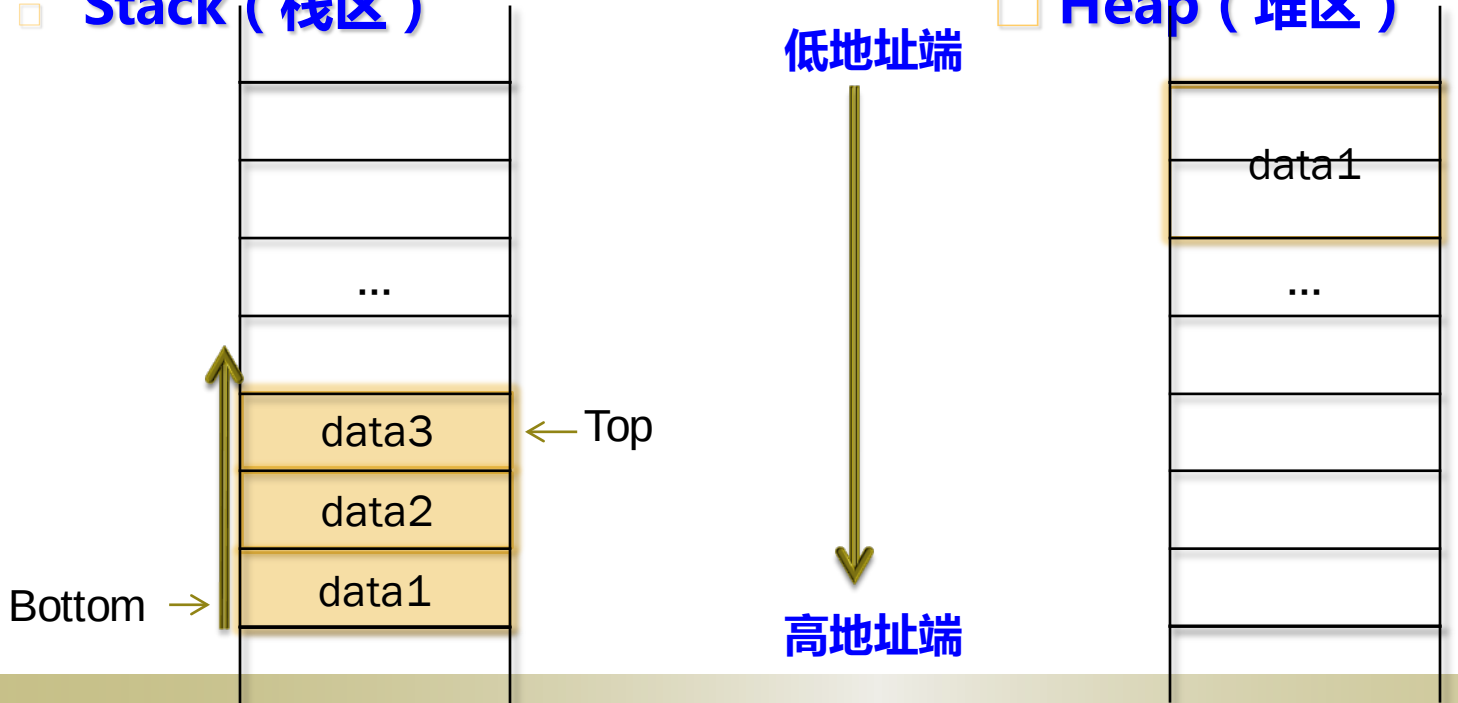


C程序的内存映像

* Stack and Heap Based Memory

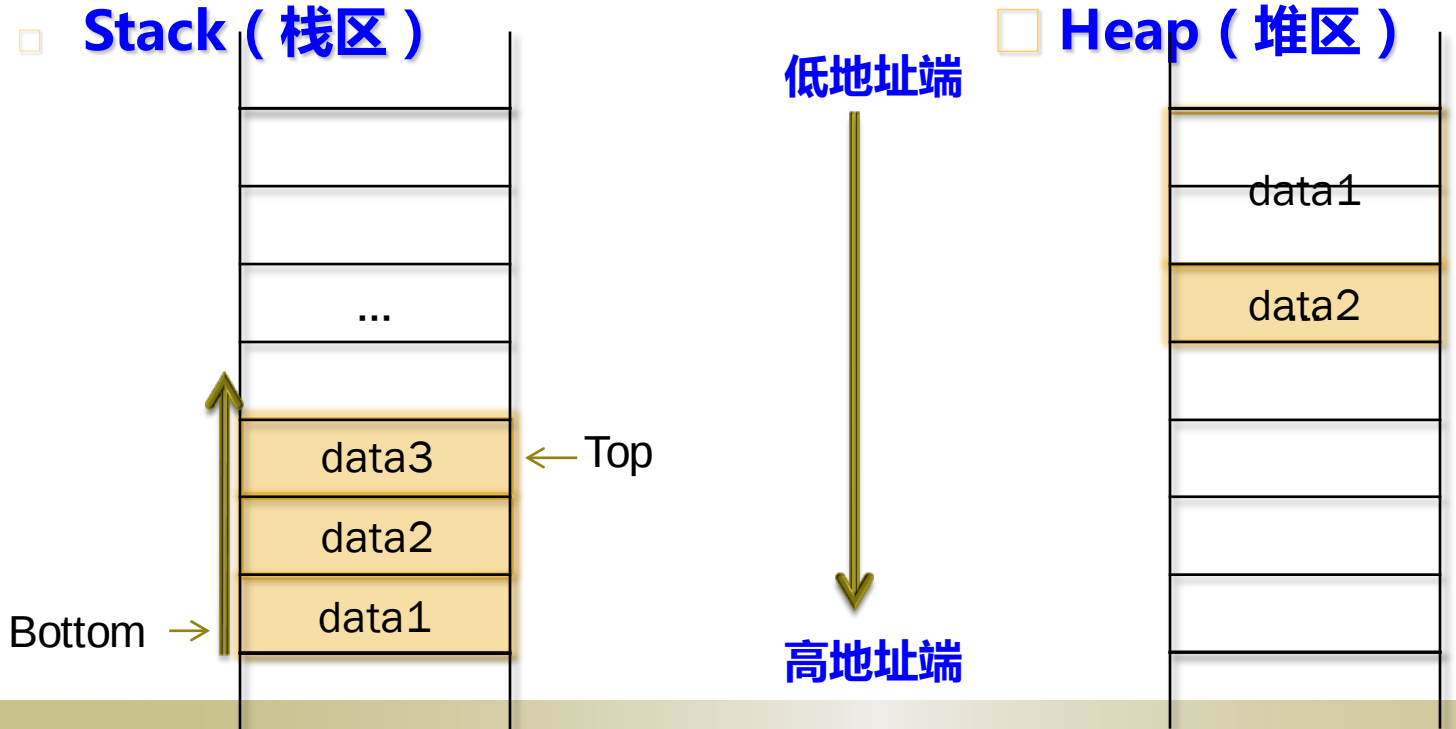
Stack（栈区）

Heap（堆区）



C程序的内存映像

* Stack and Heap Based Memory

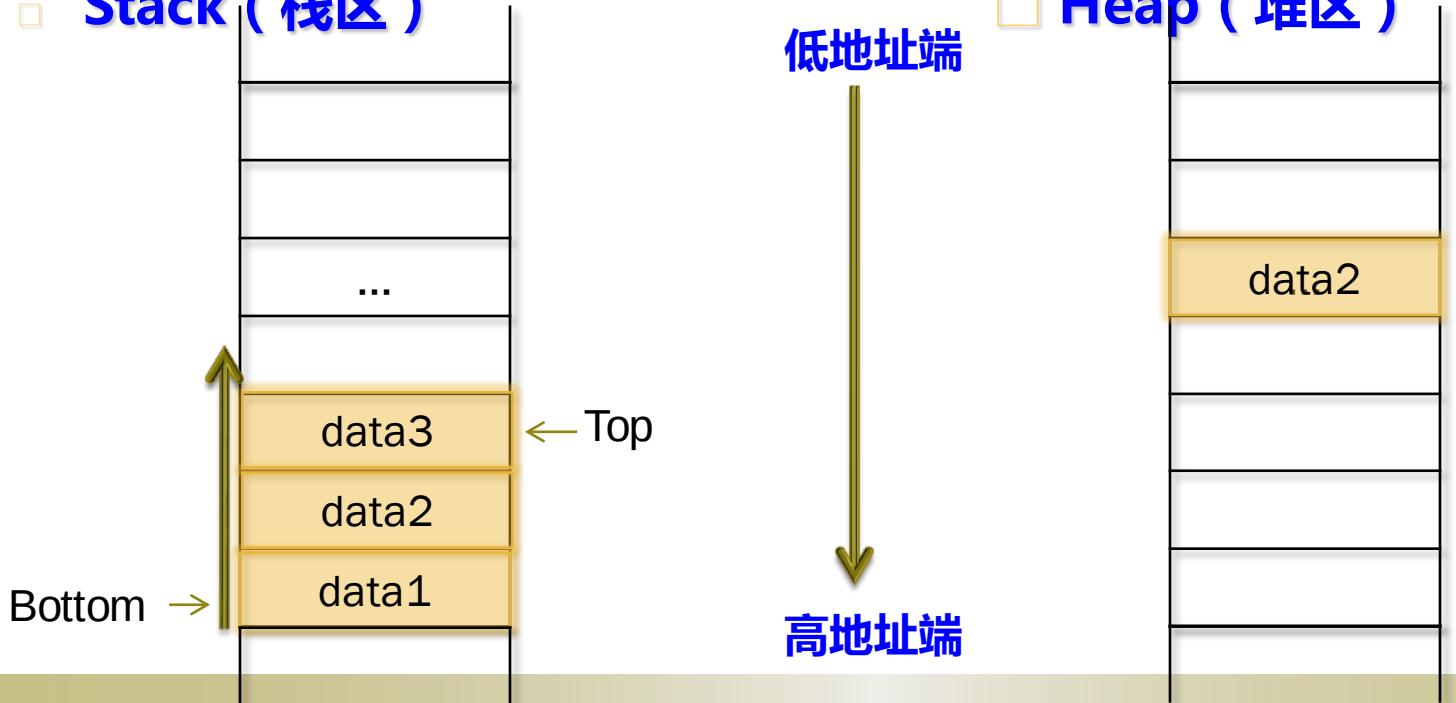


C程序的内存映像

* Stack and Heap Based Memory

Stack（栈区）

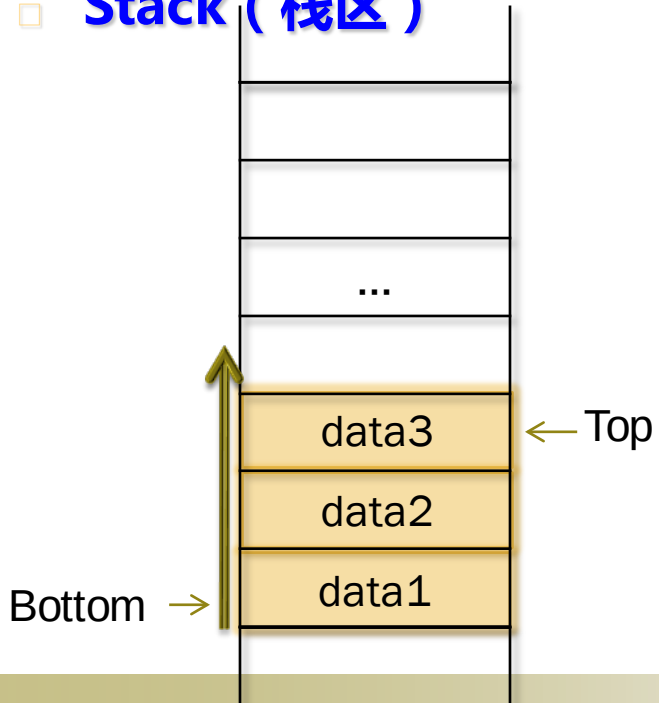
Heap（堆区）



C程序的内存映像

* Stack and Heap Based Memory

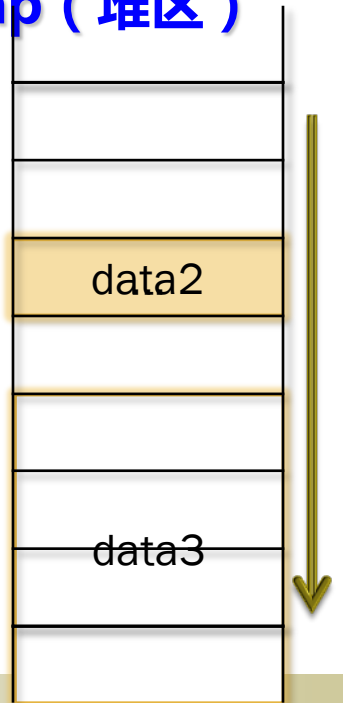
Stack（栈区）



低地址端

高地址端

Heap（堆区）



C程序的内存映像

- **Stack-based memory:**
  - 生存期由函数决定
- **特点:**
  - 由编译系统自动分配释放，无需程序员管理
  - 生长方向向下
  - 分配效率高
  - 无碎片问题

- **Heap- based memory:**
  - 生存期由程序员决定
- **特点:**
  - 程序员不释放，会造成内存泄漏（memory leakage）
  - 生长方向是向上的
  - 分配效率低
  - 频繁申请/释放易造成内存碎片（heap fragmentation）