动态数组

程 淼

E-mail: mew_cheng@outlook.com

指针数组

- 一维字符数组可存储一个字符串
- 二维字符数组可存储多个字符串
  - char name[N][MAX_LEN];

- 用二维数组存储多个字符串时，需按最长的字符串的长度来定义这个二维数组的列数
- 二维数组的每一行存储一个字符串

指针数组

- 用二维数组对多个字符串排序

实参数组名	形参数组名	字符串排序前									
name[0]	str[0]	A	m	e	r	l	c	a	\0	\0	\0
name[1]	str[1]	E	n	g	l	a	n	d	\0	\0	\0
name[2]	str[2]	A	u	s	t	r	a	l	i	a	\0
name[3]	str[3]	S	w	e	d	e	n	\0	\0	\0	\0
name[4]	str[4]	F	i	n	l	a	n	d	\0	\0	\0

指针数组

- 指向数组的指针是一个指针变量，指针变量中保存的是一个数组的首地址
- 指针数组是一个数组，只不过是指针作为数组的元素，形成了指针数组

- 由若干基类型相同的指针所构成的数组，称为指针数组

指针数组

- 因为指针数组的元素是一个指针，在使用指针数组之前必须对数组元素进行初始化
- 指针变量未初始化时，其值是不确定的
  - 指向的存储单元是不确定的

指针数组

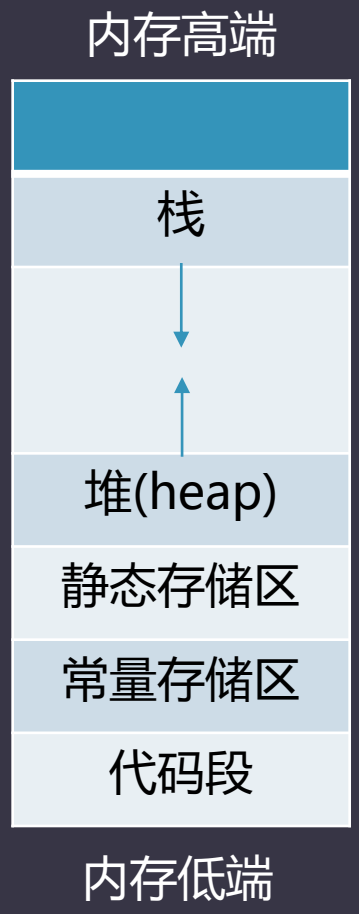
- 使用指针数组表示广义的函数参数列表
- int main(int argc, char * argv[])
- 示例 11.5

动态数组

动态数组

- 编译后的C程序获得并使用4块在逻辑上不同且用于不同目的的内存存储区

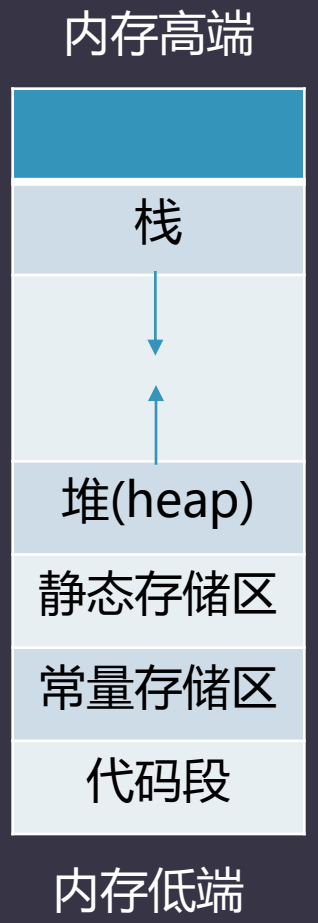
- 从内存的低端开始，第一块内存为只读存储区，存放程序的机器代码和字符串常量等只读数据，相邻的一块内存是静态存储区
  - 用于存放程序中的全局变量和静态变量等
  - 堆和栈



动态数组

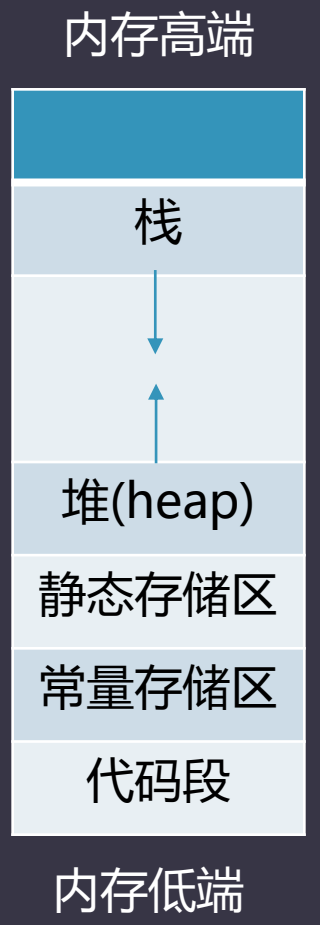
- 栈用于保存函数调用时的返回地址、函数的形参、局部变量及CPU的当前状态等程序的运行信息

- 堆是一个自由存储区，程序可利用C的动态内存分配函数来使用它



动态数组

- C语言程序中变量的内存分配方式：
  - 从静态存储区分配
  - 在 栈上分配
  - 从堆上分配



动态数组

- 在C语言中，指针之所以重要
  - 指针为函数提供修改变量值的手段；
  - 指针为C的动态内存分配系统提供支持；
  - 指针为动态数据结构提供支持；
  - 指针可以改善某些子程序的效率

动态数组

- 把指针与动态内存分配函数联用，使定义动态数组称为可能
- 动态内存分配是指在程序运行时为变量分配内存的一种方法
- 全局变量是编译时分配的
- 非静态的局部变量使用栈空间

两者在程序运行时既不能添加，也不能减少

动态数组

- C的动态内存分配函数从堆上分配内存
- 使用这些函数时只要在程序开头将头文件<stdlib.h>包含到源程序中即可

动态数组

- 函数malloc( )
  - 用于分配若干字节的内存空间，返回一个指向该内存首地址的指针
  - 若系统不提供足够的内存单元，将返回空指针NULL

- 函数malloc( )的原型为：
  - void * malloc(unsigned int size);

动态数组

- 函数calloc( )
  - 用于给若干同一类型的数据项分配连续的存储空间并赋值为0

- 函数原型为：
  - void * calloc(unsigned int num, unsigned int size);

动态数组

- free( )函数
  - 释放系统动态申请的由指针p指向的存储空间
  - 原型为:
    - void free(void * p);

动态数组

- realloc( )
  - 用于改变原来分配的存储空间的大小
- 函数原型为：
  - void * realloc(void * p, unsigned int size);