School of Economics ang Management, Beihang University

Introduction to C Programming

Jichang Zhao

jichang@buaa.edu.cn

Summary

编译器与编程

- 计算机语言
- 编译器
  - 高级语言-汇编语言-机器语言
  - 常见的C编译器
- 预编译
  - #define
  - typedef
- 编译与编程
  - Compile / Programming

算法

- 算法=程序+数据结构
- 算法的基本特征
  - 有穷性
  - 输入输出
  - 确定
  - 有效
- 成竹在胸
  - 先有算法，后有代码

标识符

- 保留字
- 标准标识符
- 程序员创建的标识符
  - 标识符的首字母必须是字母或下划线
  - 只有字母、数字或下划线可以跟在下划线之后，不充许有空格

基本的数据类型

- 字符
  - 大部分编译器中有符号，读取文件时要注意这一问题
- 整形
- 浮点
  - limits.h

下面哪个数据类型不属于基本数据类型

A．整型　　B．浮点型

C．双精度型　　D．字符串型

无符号整型数据所占内存空间大于整型数据所占内存空间。

内存分配

- 内存的管理
  - 地址空间
- 变量存储空间的分配
  - 栈
  - 堆
  - 局部
  - 全局
  - 静态
- 不同类型占用的空间不一样
  - sizeof
- 内存“对齐”
  - 结构体中尤其注意这一点

类型转换

- 显式
  - (int)4.2
  - (float)2
- 隐式
  - float m=4;
- 计算差异
  - 2/3
  - 2.0/3
  - 2/3.0

标准输入输出

- printf
- scanf
- 控制串
  - %d, %f, %s, %c, %lu, %lf, %p
- scanf需要地址
  - 一般需要&,但并不是都如此
  - printf(“%c\n”,”A”);
  - printf(“%c\n”,’\a’);
  - 关于幻影换行符：在连续读取数字、字符时容易发生
  - 当scanf和gets均被用到时也需要注意
  - DEMO: sget.c

赋值

- 变量 = 操作数
- 操作数可以是
  - 变量
  - 常量
  - 表达式
- x+3=5;
- x=3==5;
- x=5+3;
- x=func(5,3);

选择

- 程序运行的路径出现分叉
- 带来的问题
  - 测试
- 关系运算
  - 与： &&
  - 或：||
  - 非： ！
- 真：非0；假：0
  - 短路求值
  - ！（d>=10 || d==0) == (d<10 && d!=0)

选择

- int x=9;
  - if(x=10)
  - if(x==10)
- FILE *rf=fopen(“1.txt”,”r”);
  - if(rf=NULL)
  - if(rf==NULL)
- 浮点数是否相等
  - 引入精度
- 字符串是否相等
  - char *m1,*m2;
  - …
  - m1==m2

if-else链

- else与最近的if的配对
  - 避免重叠判断或”悬空“

```
struct DATE larger(struct DATE d1,struct DATE d2)
{
   if(d1.year>d2.year)
      return d1;
   else if(d1.year==d2.year)
   {
      if(d1.month>d2.month)
      {
         return d1;
      }


      else if(d1.month==d2.month)
      {
         if(d1.day>d2.day)
            return d1;
      }
   }
   else
      return d2;
}
```

switch语句

- 只能针对整形表达式
- break
- default
- 多重标签

在 switch 语句当中，如果忘记使用 break 命令，则从符合条件的 case 语句开始，所有的 case 语句都会被执行。

循环

- 重复自身的代码段
  - 条件控制
- 循环中要对判断条件做出改变
  - “死”循环的使用
- 三类循环
  - for
  - while
  - do while

循环

- break
- continue
- 多重循环

请用 while 循环重写下列表达式. (5%)

```
int i, s =0;
for(i=0;i<50; i++)
{     s = s + i*2;
}
```

函数

- 模块化的重要性
  - 分工协作
  - 代码重用
- 函数原型
  - 参数声明
  - 形参
- 函数实现
  - 实参
- 函数调用
  - 返回值

关于函数原型

- 编译器的调用前遇到原型：检查并隐式的转换类型
- 编译器调用后遇到原型：执行默认的参数提升
  - 可能无法产生期望的结果
  - 写函数原型是个好习惯
    - 不要写在main函数里

递归

- 函数调用自身
- 需要一个终止条件
- 需要实参发生变化
- 可以写成循环

变量

- 作用域
- 有效或可知的程序段落
- 局部作用域
  - 函数内部声明
  - 一个变量名称能够在多个函数中被声明和使用
- 全局作用域
  - 不宜滥用
  - 会破坏函数间的独立性和隔离性
  - 一般只用于符号常量和函数原型
  - 变量的作用域不会影响或者限制变量的数据类型

变量

- 生命周期
  - 存储类，为变量保留的存储区位置的时间长度
  - auto, register, static, extern
  - 局部变量只能是auto, register, static
    - auto是局部变量的默认类型
    - register不能取址
- 全局变量只能是static, extern
  - 全局变量的默认存储类是？
  - 除了静态变量（全局或局部），所有的变量在它们第一次进入作用域时都应初始化

指针

- 为什么需要指针
  - 优点
  - 缺点
- 指针
  - 指针也是变量：存储地址的变量
  - 指向类型要明确
  - 间接寻址
  - 空指针
- 指针的运算
  - 要具备物理含义
  - 往往与数组密切相关
- 指针与函数
  - 参数：“传值”
  - 返回值：返回是否有意义（与存储类、内存属性密切相关）

数组

- 重要的聚合类型
  - 线性结构的优缺点
- 类型和长度要提前确定
  - 内存分配
  - “变长”数组不能初始化
- 数组相关的操作
  - “越界”
- 数组与函数
  - 长度应该作为参数同时传入
- 多维数组
  - 空间分配
    - char s[10][1025]
    - char *s[10];...

字符串

- 为什么需要\0
  - 一种约定：便利字符串的处理
- 字符串常量
  - char s[]=“Test”；
  - char *s=“Test”；
- 字符串与函数
  - 注意末尾有\0
- 相关的库函数要熟悉
  - strcpy
  - strcmp
  - strcat

指针与数组

- 大部分情况下可以互换
  - 主要差异在于空间分配
- 数组名的”特殊性“
  - int a[10];
  - a==&a;
- 二维数组
  - int a[10][10];
  - a[0]与a[1]差多少？
  - a[0]的类型是？
  - 如何按列遍历二维数组？

结构体

- 为什么需要结构
  - 并行数组的缺陷
  - 一种聚合类型
- 结构的内存分配
  - 成员顺序分配
  - 会有两次内存对齐
- 结构的成员访问
  - .运算符
- 结构指针
  - ->运算符
  - 指向第一个成员
- 结构可以复制
  - 如何从函数返回一个数组？（实现了多值的返回）
  - 有缺点，所以需要结构指针

联合

- 为什么需要联合
  - 内存空间的节约
- 联合的内存分配
  - 最大成员
- 伴生变量
  - 记录当前存储的有效成员
- 常与结构体一起使用
  - 商品的例子

文件

- 为什么需要文件
  - 持续化存储数据
- 文件结构体与文件指针
  - FILE
  - FILE *inFile
- 文件操作
  - fopen
  - fclose
  - fprintf
  - fscanf
  - fgetc
  - fseek
  - ftell

练习

- double average(int a[10], int n);
  - 原型里没有必要声明长度，会忽略
- double average(int [],int)
  - 如果需要数组的长度，需要通过参数传递
- int a[10];
- double ave=average(a[],10)
- double ave=average(a,5)
- double ave=average(a,100)

练习

- exit也能使程序中止
  - 不管那个函数调用exit，都能使程序中止
  - 对于return,只有main函数调用进才能终止程序
  - exit更容易定位错误

练习

- 局部变量与形式参数异同
  - 形式参数具有与局部变量一样的auto存储类和块作用域
  - 区别在于函数调用时对形式参数进行自动初始化（通过实参）

练习

- 请用prinf输出”\Hello World\”
- 二维数据x[10][10]，通过循环计算对角线元素的和
- 不占用新空间的情况下，将一个数组倒序
- 下面的代码是为了实现什么功能？

```
void func(int a[], int n)//n为数组长度
{
    int i;
    for(i=n-1;i>=1;i--)
    {
            int j=rand()%(i+1);
            int temp=a[j];
            a[j]=a[i];
            a[i]=temp;
    }
}
```

练习

- 高效地实现从一个字符串删除某特定字符。
  - 不占用额外空间
  - 仅遍历一次字符串
  - DEMO: del.c

练习

- 平方数
  - 四位数
  - 前两位和后两位的和的平方等其自身

练习

- 从键盘输入10个不同的整数并存入一维数组，找到其中的最大值与最小值并交换位置，其他数位置不变。

练习

- 从键盘输入10个数，找出其中不重复的数，若没有，输出NO

练习

- 找出二维数组的鞍点。所谓鞍点，就是该位置上的数在该行最大，在该列最小。若没有鞍点，输出NO。