

一. 字符串函数 (续)

I. strchr

1. prototype: `char * strchr(const char * s, int c);`
2. 作用：在s中寻找字符c首次出现的位置，找到了返回指针，找不到返回NULL

II. strrchr

1. prototype: `char * strrchr(const char * s, int c);`
2. 作用：在s中寻找字符c最后出现的位置，找到了返回指针，找不到返回NULL

III. strpbrk

1. prototype: `char * strpbrk(const char * s1, const char * s2);`
2. 作用：在s1中找一找有没有s2的**字符**，找到了返回**s1的首位置的指针**。

IV. strstr

1. prototype: `char * strstr(const char * s1, const char * s2);`
2. 作用：在s1中找一找有没有s2**字符串**，找到了返回**s2第一次出现的指针**。

二. 把字符串转化为数字

1. 头文件相关：stdlib.h
2. atoi(), atol(), atof()三个函数只需要一个字符串，然后就能把相应的字符串转化为int, long, 和**double**类型。注意传入不含数字的字符串是**未定义行为**。
3. 更为智能的函数：strtoul(), strtoul(), strtod()将对应字符串分别转化为long, unsigned long和double类型。这几个的使用比较复杂，以strtoul为例
 - ①prototype: `long strtoul(const * restrict nptr, char ** restrict endptr, int base);`
 - ②使用举例：`value = strtoul(number, &end, 10);`
 - ③endptr：这是一个指针的地址，这个指针指向数字结束字符,从上面可见一斑，我们是传入一个指针加&，这样做可以大大地提升灵活性：使用转换说明可以将end转换为后面的字符串，比如输入123abc，打印的就是abc；而*end是一个字符（正常是空字符）。
 - ④base: 就是储存进数组的进制，最多可以是36进制（10+26）

三. 命令行参数

1. 定义：命令行中跟在可执行文件（或批处理文件等）后面的附加项叫做命令行参数。
2. 读取：通过main函数的参数来读取——标准形式：`int main(int argc, char *argv[]){.....}`

3. argc的含义：命令行共有多少个字符串（参数计数，argument count）
4. argv的含义：这时一个指针数组，里面存放的是参数值（argument value），其中，argv[0]一般是命令本身。
5. 技巧：可执行文件在命令行内只读取单词形式的字符串，想要传入含有多个单词的字符串，就要使用双引号把相关的“多个单词读取成一个”。e.g. "I am hungry." 被视为是一个参数。
6. 在IDE（不包含Macintosh）中使用命令行比较麻烦，建议直接在命令行内执行编译器。