1. calloc 和 realloc
   1. calloc 和malloc 都是在堆区分配内存
   2. 与malloc不同的是，calloc会将空间初始化为0
   3. calloc（个数，大小）
   4. realloc 重新分配内存
      1. 如果重新分配的内存比原来大，那么不会初始化新空间为0
      2. 先看后续空间，如果足够，那么直接扩展
      3. 如果后续空闲空间不足，那么申请足够大的空间，将原有数据拷贝到新空间下，释放掉原有空间，将新空间的首地址返回
      4. 如果重新分配的内存比原来小，那么释放后序空间，只有权限操作申请空间
2. sscanf的使用

|  |  |
| --- | --- |
| %\*s或%\*d | 跳过数据 |
| %[width]s | 读指定宽度的数据 |
| %[a-z] | 匹配a到z中任意字符(尽可能多的匹配) |
| %[aBc] | 匹配a、B、c中一员，贪婪性 |
| %[^a] | 匹配非a的任意字符，贪婪性 |
| %[^a-z] | 表示读取除a-z以外的所有字符 |

* 1. 练习
     1. 将 ip 分别截取到 num1 到 num4中
     2. 将 123456#zhangtao@abcde 中截取中间的zhangtao有效数据

1. 查找子串
   1. 实现mystrstr 自己查找子串功能
2. 指针的易错点
   1. 越界
   2. 指针叠加会不断改变指针指向
   3. 返回局部变量地址
   4. 同一块内存释放多次（不可以释放野指针）
3. const使用场景
   1. const使用 修饰形参 防止误操作
4. 二级指针做函数参数的输入输出特性
   1. 二级指针做函数参数的输入特性
      1. 创建在堆区
      2. 创建在栈区
   2. 二级指针做函数参数的输出特性
      1. 被调函数分配内存，主调函数使用
5. 二级指针练习-文件操作
   1. 读取配置文件信息 ，并且将信息存放到 数组中
   2. 注意： 释放堆区，关闭文件
6. 位运算
   1. 按位取反 ~ 0变1 1 变0
   2. 按位与 & 全1为1 一0为0
   3. 按位或 | 全0为0 一1为1
   4. 按位异或 ^ 相同为0 不同为1
7. 位移运算
   1. 左移运算 << X 乘以2 ^ X
   2. 右移运算 >> X 除以 2 ^X
      1. 有些机器用0填充高位
      2. 有些机器用1填充高位
      3. 如果是无符号，都是用0填充