解决问题之前一定要出一个框架，不要盲目地上来就写程序，这是一个非常不好的习惯，要画个NS图、流程图这些

画流程图特别方便的，Windows下的visio，Linux下的DIA

**关键字**

选择：if-else switch-case

循环：while do-while for if-goto

辅助控制：continue break

**详解**

if-else

语法格式：

if(exp)

cmd1;

else

cmd2;

注意，else只与离它最近的if相匹配，即使你的else的缩进与上上个if对齐，除非你用了{ }

if后面的语句如果超过一句了，那就加上{ }

switch-case

语法格式：

switch(exp)

{

case 常量或常量表达式:

…;

break;

case 常量或常量表达式:

…;

break;

……

default:

…;（这里一定要有，用于报错，非常重要）

break;（这里的break倒是可有可无，因为程序已经执行完了）

}

default那里的…可以用，\_exit(0);，表示什么都不敢做了，直接结束当前进程，不会做任何的清理工作，因为这时候你不敢做清理了，IO也不敢刷新，钩子函数也不敢调用，你什么都不敢做了，因为你的程序已经出现意外了。除了exit，你还可以用某一个sig，选择一个信号，选择什么功能的信号呢，能够杀死当前进程的信号，最好在杀死当前进程的同时，能够产生一个core dump（没听清，不知道是不是叫这个）文件，这个文件实际上就是你当前出错的一个现场，如果你再有gdb等工具的话，其实可以把出错现场放到gdb当中进行调试，这个出错现场对于我们程序员来说是一个非常宝贵的资源之一

while

语法格式：

while(exp)

loop;

do-while

语法格式：

do

{

loop;

}while(exp);

for

语法格式：

for(exp1; exp2; exp3)

loop;

if-goto（慎用）

goto语句实现的是无条件跳转，会破坏结构化，模块化的编程，而且它不能跨函数跳转

嵌入式开发当中，很多地方会用到死循环，典型的

while(1);

for(:,:,:);

杀掉死循环，用ctrl+c，实际上是发送了一个信号，叫signal interrupt，把当前进程终止