

附录：C 语言编程规范

一、排版

1. 程序块要采用缩进风格编写,缩进的 TAB 键一个.
2. 相对独立的程序块之间、变量说明之后必须加空行.
3. 较长的语句(>80 字符)要分成多行书写,长表达式要在低优先级操作符处划分新行,操作符放在新行之首,划分出的新行要进行适当的缩进,使排版整齐,语句可读.
4. 循环、判断等语句中若有较长的表达式或语句,则要进行适应的划分,长表达式要在低优先级操作符处划分新行,操作符放在新行之首.
5. 若函数或过程中的参数较长,则要进行适当的划分.
6. 不允许把多个短语句写在一行中,即一行只写一条语句.
7. if、while、for、default、do 等语句自占一行.
8. 对齐只使用 TAB 键,不使用空格键.
9. 函数或过程的开始、结构的定义及循环、判断等语句中的代码都要采用缩进风格,case 语句下的情况处理语句也要遵从语句缩进要求.
10. 程序块的分界符(如 C/C++ 语言的大括号 '{' 和 '}')应各独占一行并且位于同一列,同时与引用它们的语句左对齐.在函数体的开始、类的定义、结构的定义、枚举的定义以及 if、for、do、while、switch、case 语句中的程序都要采用如上的缩进方式.
11. 在两个以上的关键字、变量、常量进行对等操作时,它们之间的操作符之前、之后或者前后要加空格;进行非对等操作时,如果是关系密切的操作符(如 ->),后不应加空格.
12. 程序结构清晰,简单易懂,单个函数的程序行数不得超过 100 行.

二、注释

1. 一般情况下,源程序有效注释量必须在 20%以上.
2. 说明性文件(如头文件.h 文件、.inc 文件、.def 文件、编译说明文件.cfg 等)头部应进行注释,注释必须列出:版权说明、版本号、生成日期、作者、内容、

功能、与其它文件的关系、修改日志等,头文件的注释中还应有函数功能简要说明.

3. 源文件头部应进行注释,列出:版权说明、版本号、生成日期、作者、模块目的/功能、主要函数及其功能、修改日志等.
4. 函数头部应进行注释,列出:函数的目的/功能、输入参数、输出参数、返回值、调用关系(函数、表)等.
5. 边写代码边注释,修改代码同时修改相应的注释,以保证注释与代码的一致性.不再有用的注释要删除.
6. 注释的内容要清楚、明了,含义准确.
7. 避免在注释中使用缩写,特别是非常用缩写.
8. 注释应与其描述的代码相近,对代码的注释应放在其上方或右方(对单条语句的注释)相邻位置,不可放在下面,如放于上方则需与其上面的代码用空行隔开.
9. 对于所有有物理含义的变量、常量,如果其命名不是充分自注释的,在声明时都必须加以注释,说明其物理含义.变量、常量、宏的注释应放在其上方相邻位置或右方.
10. 数据结构声明(包括数组、结构、类、枚举等),如果其命名不是充分自注释的,必须加以注释.对数据结构的注释应放在其上方相邻位置,不可放在下面;对结构中的每个域的注释放在此域的右方.
11. 全局变量要有较详细的注释,包括对其功能、取值范围、哪些函数或过程存取它以及存取时注意事项等的说明.
12. 注释与所描述内容进行同样的缩排.
13. 将注释与其上面的代码用空行隔开.
14. 对变量的定义和分支语句(条件分支、循环语句等)必须编写注释.
15. 对于 switch 语句下的 case 语句,如果因为特殊情况需要处理完一个 case 后进入下一个 case 处理,必须在该 case 语句处理完、下一个 case 语句前加上明确的注释.

三、 标识符命名

1. 标识符的命名要清晰、明了,有明确含义,同时使用完整的单词或大家基本可以理解的缩写,避免使人产生误解.
2. 命名中若使用特殊约定或缩写,则要有注释说明.
3. 自己特有的命名风格,要自始至终保持一致,不可来回变化.
4. 对于变量命名,禁止取单个字符(如 i、j、k...),建议除了要有具体含义外,还能表明其变量类型、数据类型等,但 i、j、k 作局部循环变量是允许的.
5. 命名规范必须与所使用的系统风格保持一致,并在同一项目中统一,比如采用 UNIX 的全小写加下划线的风格或大小写混排的方式,不要使用大小写与下划线混排的方式.

四、 可读性

1. 注意运算符的优先级,并用括号明确表达式的操作顺序,避免使用默认优先级.
2. 避免使用不易理解的数字,用有意义的标识来替代.涉及物理状态或者含有物理意义的常量,不应直接使用数字,必须用有意义的枚举或宏来代替.

五、 变量

1. 去掉没必要的公共变量.
2. 仔细定义并明确公共变量的含义、作用、取值范围及公共变量间的关系.
3. 明确公共变量与操作此公共变量的函数或过程的关系,如访问、修改及创建等.
4. 当向公共变量传递数据时,要十分小心,防止赋与不合理的值或越界等现象发生.
5. 防止局部变量与公共变量同名.
6. 严禁使用未经初始化的变量作为右值.

六、 函数、过程

1. 对所调用函数的错误返回码要仔细、全面地处理.
2. 明确函数功能,精确(而不是近似)地实现函数设计.

七、 质量保证

1. 只引用属于自己的存储空间.
2. 防止引用已经释放的内存空间.
3. 过程/函数中分配的内存,在过程/函数退出之前要释放.
4. 过程/函数中申请的(为打开文件而使用的)文件句柄,在过程/函数退出之前要关闭.
5. 防止内存操作越界.
6. 认真处理程序所能遇到的各种出错情况.
7. 系统运行之初,要初始化有关变量及运行环境,防止未经初始化的变量被引用.
8. 有可能的话,if 语句尽量加上 else 分支,对没有 else 分支的语句要小心对待;switch 语句必须有 default 分支.
9. 禁止 GOTO 语句.