变量的命名规则的目的:增加代码的可读性

变量命名规则

普通变量的命名规则

变量的命名	**1.变量名 = 作用域前缀 + 类型前缀 + 一个或者多个单词。为了便于接界定，每一个单词的首字母必须大写.** **对于一些用于循环的变量可以使用i j k 等一些简单字符。其他必须写明含义**
作用域	作用域前缀表明一个变量的可见范围。作用域可以有如下几种。 前缀 说明 无 局部变量 m 类成员变量(member) sm_ 类的静态成员变量 s_ 静态成员变量 g_ 全局变量 sg_ 静态全局变量 gg_ 进程间共享的数据段的全局变量 除非不得已，否者尽可能少用全局变量
类型前缀	类型前缀标明一个变量的类型，可以有如下几种 n 整形 e 枚举 c 字符型变量(char) b 布尔型变量(bool) f 浮点型变量 p 指针型变量和迭代子 pfn 函数指针变量和函数对象指针 Callpfh 回调函数指针 g 数组 i 类的实列(instance) 类型前缀可以组合使用,例如: “gc”表示字符数组， “ppn”表示指向整形的指针的指针等等
推荐的组成形式	变量的名称应当使用 "名词" 或者 “形容词 + 名词" 例如：nCode , m_nState nMaxWidth

常量名称

常量名 类型前缀 + 全大写字母 组成，单词之间通过下划线来定义如: cDELIMITER nMAX_BUFFLEN

枚举 结构体 宏 联合体命名规则

枚举

1.必须全部大写

2.枚举名加小写前缀"enum"

typedef enum_KFILE_OPEN_MODE

{

enumOPEN_READONLY = 0,

enumOPEN_READWRITE = 1,

enumCREATE_ALWAY = 3,

}

宏名

宏名加小写前缀"def"_加类型名

#define def_nMAXNUMBER 100

结构体

typedef struct tagKPOINT

{

int x,

int y,

}

联合体

1.必须全部大写

2.枚举名加小写前缀"uni"

typedef union_VARIANT

{

char unicVal;

int uninVal

float uniftval

}VARIANT

函数

函数的注释

1.具体的工能(实现具体的功能，需要注意的东西)

2.输入参数

3.输出参数

4.返回值(返回值说明：如果你不需要这个返回，必须定义为void)

注明:必须写函数内部必须写注释

函数的命名规则

普通函数命名	**函数的名称由一个或者多个单词组成。为了便于界定，首字母需要大写 推荐写法 动词 + 名词**
回调函数命名	函数的名称由一个或者多个单词组成。为了便于界定，首字母需要大写 推荐写法 Call + 动词 + 名词
模块接口函数	需要依据模块的具体名称 + API +动词+名词 GUI_API_GetName()

注明 : 以后函数名称需要在前面加上 模块_ 例如: 普通函数: GetName() 模块接口函数:GUI_API_GetName() 回调函数 GUI_API_CallGetName()

文件

文件名的命名规则:

暂时参考开源代码Linux内核的文件命名和组织结构

文件的注释说明：

1：文件的功能说明

2：版权问题(版权时间+作者)