

TreeOS 编程规范

2017-07

前言:

编程规范可使程序更加简明、整齐干净、易读、易理解、易移植，从而提高移植性、兼容性、易维护等，提高编程效率。

当然，若规矩太多，难以记得全，反而对程序员是个负担。因此把规范尽量精简，抓住关键之处即可。

编程规范要求贯穿始终。TreeOS 是开源软件，也要求对其修改或扩展的软件也必须遵守相同的编程规范。

以下是 TreeOS ComLib 软件构件库的编程规范。

一 基本原则

- 所有的代码必须采用 ANSI C 标准
- 严格遵守编程规范
- 尽可能使用简单的语法
- 避免使用复杂语句
- 不使用晦涩难懂的表述
- 不要使用 GOTO 语句
- 修改代码，需随时更新相关文档

二、 源代码文件

我们把具有一定功能的源代码文件（如某种设备的驱动程序）称为一个模块，它包括两个文件：执行文件（.c 文件）和头文件（.h 文件）。

执行文件的布局:

- 文件头
- 包含文件(#include)
- 变量，包括全局变量和局部变量
- 表单
- 函数，包括全局函数和局部函数

头文件布局:

- 文件头
- 包含文件(#include)
- Global macros (全局宏定义)
包括：条件编译选择、属性（常量定义）、IO 管脚定义、特殊语句。
- 全局变量外部声明
- 函数声明(前一部分为局部函数原型，后一部分为全局函数原型)

注意：头文件必须保证避免在包含文件中重复定义一个数值。

三、 文件头

文件头是放在源代码文件头部的一个注释块，其内容包括版权、公司信息、程序员、文

件的描述信息、软件版本及修改历史记录等。

注意每次修改软件，应该做相应的版本修改记录。

例如：

```
/****** (C) COPYRIGHT 2012 北京光轮电子科技有限公司  
*****
```

本软件的著作权归北京光轮电子科技有限公司所有。

本软件所提供的所有程序仅供购买者或本人作为学习之用，其目的是为了您能更好地学习和理解 TreeOS 操作系统。若您自行把本软件引用到您的产品之中，运行后出现由本软件直接或间接引起的任何不良后果，本软件著作权所有者将不会作出任何赔偿，亦不承担任何法律责任。

本软件及著作权、免责声明的最终解释权归本软件作者所有。

详情请登录官网：<http://www.treeos.com>。

您若有疑问或建议，请发邮件到 support@treeos.com 与我们联系。

最新技术进展，请关注微信公众号“光轮电子”，或淘宝直营店“TreeOS”。

```
*****
```

文件名称：TreeOS_main.h

适用 TreeOS：本软件适用于 TreeOS 实时操作系统。

配置：缺省配置适用 TreeOS Kepler11 开发板。

版本：隶属 ComLib A2 构件库, 2014

功能：

*TreeOS_main.c 的配置文件

修改记录：

1、完成日期：2014-10-1；作者：TreeOS 老林

```
*****/
```

四、文件包含

TreeOS 要求每个模块的 C 文件必须配备一个同名头文件。

模块 C 文件可以只列出需要包含的头文件，也可以用一个名为 TreeOS_inc.h 的头文件把所有文件包含进去，这样虽然省事（所有模块都只需包含 TreeOS_inc.h 即可），但是会使编译时间加长，而且随着 TreeOS ComLib 的不断扩大，这个 TreeOS_inc.h 也需要根据需求配置。另外，模块若仅包含所需的头文件，可以直观地看到该模块与其它哪些模块出现了耦合。

例如：

```
#include<reg52.h> //M/
```

```
#include "TreeOS_main.h"
```

```
#include "TreeOS_mcu.h"
```

```
#include "TreeOS_int.h"

#include "TreeOS_hc574.h"
#include "TreeOS_hd61202.h"
#include "TreeOS_lcd_disp.h"

#include "TreeOS_I2C.h"
#include "TreeOS_24Cxx.h"
#include "TreeOS_pcf8563.h"

#include "TreeOS_RS232.h"
#include "TreeOS_beep.h"
#include "TreeOS_keyboard.h"
```

五、函数声明

模块中所有定义的函数都在头文件中进行声明。分为两部分，前一部分为局部函数原型，后一部分为全局函数原型。

需要调用其它模块中定义的函数，必须包含该模块的头文件。

六、注释

注释写在语句后面，一般使用//符号。太长了可以分多行写。之所以用//，一是书写简单，二是有时方便把一段程序用/**/注释掉（比使用 #if 0 和 #endif 更直观）。

当然，也不排斥注释使用/*..*/，但要求同一个模组中(即对应的.c 和.h 文件)，使用同一种注释风格。

七、缩进格式

记住函数的“{}”直接靠近左栏边沿，语句则相对左栏边沿缩进 4 个空格。

对于 if、for、while、switch 等语句，其语句块大括号“{”与“}”不缩进，且单独放一行，并成对处在同一列。这样便于区分众多的{}。大括号{}内的语句缩进 4 格，把{}露出来。

例如：

```
if((t - t0) > 500) //500ms 计时
{
    t0 = t;
    for (i = 0; i < 5; i++)
    {
        disp_chinese(hao + i * 32, 0);
    } //for
} //if
```

对于 case 语句，缩进 4 个空格，把 case 露出来。

例如：

```
case UP:
    if (item)
```

```

        {
            .....
        }
        break;
case DOWN:
    if (item < 5)
    {
        .....
    }
    break;
.....

```

八、命名规则

● 宏定义

宏定义用大写表示，用下划线分隔单词。特殊情况允许使用 1 个小写字母，且只能放在最前面，主要是为了避免与其它引进程序产生冲突。

宏定义一般以所在的文件的文件名的一部分作为前缀（开头）。

例如：

```
#define UART_BUFFER_SIZE_SEND 20 //M/ 发送缓冲区字节数。
```

注意：`#define` 和宏名之间空一个空格就可以了。

● 变量名与函数名

命名要尽量表达清楚，尽量让人一眼能猜出其意义。有两种命名方式：一种是词的首字母大写，以此来分开不同词；二是所有词都用小写字母，中间用下划线“`_`”分开。两种方法都可用，第二种方法虽然多了“`_`”，但更有利于阅读。

变量名与函数名一般以所在的文件的文件名的一部分作为前缀（开头）。

例如：

```
ui8 UartSendPrt = 0;
```

```
void lcd_init(void)。
```

九、数据类型

为了通用于各种处理器的 IDE 以及书写方便提高效率，TreeOS 用 `typedef` 定义了所有数据类型，这些定义放在一个专门的头文件 `TreeOS_typedef.h`。

有些平台 `char` 等价于 `signed char`，另一些则等价于 `unsigned char`，为此，要在代码中明确的使用 `signed char` 或 `unsigned char`。

不能直接使用 `int` 类型，要使用 `short` 和 `long`。

例如：

```
typedef signed long i32;
```

```
typedef signed short i16;
```

```
typedef signed char i8;
```

```
typedef code signed long i32c; //code data,read only
```

```
typedef code signed short i16c; //code data,read only
```

```
typedef code signed char i8c; //code data,read only
```

```

typedef unsigned long   ui32;
typedef unsigned short  ui16;
typedef unsigned char   ui8;

typedef code unsigned long   ui32c; //code data,read only
typedef code unsigned short  ui16c; //code data,read only
typedef code unsigned char   ui8c;  //code data,read only

```

十、语句与表达式

- 每行只放一个语句;

例如:

```

    if (a < b)
        return 1;
    else
        return 0;

```

- 代码块之间加空行分隔;
 - 每个逗号或分号后面要有一个空格符分隔;
- 例如: `ui8 send_data_24Cxx(ui16 SubAdr, ui8 ByteCnt ui8 *p)`
`for (i = 0; i < 100; i++)`

注意: 函数名后的括号不可以有空格符;
 括号()与括号内的字符间不要有空格。

- 一元操作符与操作数之间不可以包含空格;
 例如: `!a, ~b, i++, i--, *p, &x, (ui16)y` 等。
- 二元和多元操作符与操作数之间至少要有个空格;
 例如: `a + b; a = b; a | b; a > b; a >= b;` 等。
- 部分关键字后面要跟随一个空格符
 例如: `for (i = 0; i < 100; i++)`, `if (a > b)`, `else { ... }`, `switch (a)`, `return (a)`等。

十一、结构体与共用体

结构类型用大写字母表示(允许带1个小写字母),规则同宏定义。

例如:

```

typedef struct    //定义时分 hh:mm
{
    ui8 hour;      //0~23
    ui8 min;       //0~59
} tHHMM;

```

十二、功能配置

TreeOS 是一个可配置、可剪裁的构件化操作系统。对于具体应用,需要对功能进行配置(在.h文件中修改),甚至对一些软件进行修改(在.c文件中修改)。所有这些需要修改的地方,都带有“//M/”(M即Modify的意思),提醒用户此处可能需要修改。

例如:

```

#define UART_BUFFER_SIZE_SEND 20 //M/ 发送缓冲区字节数。
//M/在该处提醒修改串口发送缓冲区的大小。太小了不合适,太大了浪费内存。

```