

2022 WTR 招新试题-电控部分

哈喽同学！欢迎报名WTR战队。

这是招新试题的电控部分，部分题目可能有些许难度，**允许百度、允许百度、允许百度 (重要的话说三次)**。

欢迎接触过 RTOS 或有多线程开发经验的同学们做一下 RTOS 选做部分。（不会做也没关系，毕竟是选做嘛，进入战队后也会有相关培训的）

一、C 语言

(1) 原码、反码、补码

8位二进制数补码范围是多少？请写出十进制数13，19，-5，-13对应的8位二进制数补码。

(2) 关键字

关键字 inline 和 static 修饰的函数特性和在使用中有什么不同之处？请简要回答。

(3) 内存区域

请简要回答以下问题。

1. 在 C 程序中，什么是堆区，什么是栈区？
2. 一段 C 语言代码如下：
 1. 请指出其中 var1, var2, var3, var4 和 malloc() 分配的内存各存储在哪个区域。
 2. var1~var4 和 *p 中，哪些初始值一定是 0？

```
#include <stdio.h>

int var1;

int func(int var2)
{
    int var3;
    static int var4;
    void *p = malloc(4);
}
```

(4) 多文件编程

有如下 3 个文件，请补全以完成以下要求：

- Add() 为 PrintAdd.c 内部使用的函数，不要将其暴露给 main.c
- 尽量不要出现 warning

提示：

- printf() 需要包含 stdio.h
- 函数定义已经实现，请补充函数声明、头文件包含以及其他必要部分

main.c

```
/**
 * @file main.c
 *
 */

#include "PrintAdd.h"

int main()
{
    PrintAdd(4,5);
    return 0;
}
```

PrintAdd.c

```
/**
 * @file PrintAdd.c
 *
 */

#include "PrintAdd.h"
// TO DO

void PrintAdd(int x, int y)
{
    int result = Add(x, y);

    printf("%d\n", result);
}

static int Add(int x, int y)
{
    return x + y;
}
```

PrintAdd.h

```
/**
 * @file PrintAdd.h
 *
 */
```

```
// TO DO
```

二、算法部分

(1) PID 调参

在使用PID控制律控制电机转速的场景下，你会依据什么原则怎么调整参数使得该系统达到“快、准、稳”的效果？[自动化专业同学必做]

(2) 应用题：数据转换

一种 24 bit 的数模转换芯片可以将模拟电压转换为 **24 bit 的有符号补码整数**，并将其分 3 段传输给单片机。

数模芯片的输出为**大端模式**（例如：原数字为 0x123456，则接收到的3段数据先后为0x12、0x34、0x56）。

由于C语言中没有 int24_t 数据类型，为方便在字节序为**小端模式**的单片机上使用，请将其转换为 **int32_t 的有符号整数**。（例如，接收到 0x12、0x34、0x56，则应存储为 0x56、0x34、0x12、0x00）

代码框架已经写好，请在此基础上完成。

注：家用电脑都是小端模式，所以以下代码可以在你的电脑上编写测试

```
#include <stdio.h>
#include <stdint.h>

int32_t ConvertData(uint8_t raw_data[3])
{
    // TO DO
}

int main()
{
    uint8_t RxGroup[3] = {0x12, 0x34, 0x56}; // 接收到的原始数据
    printf("%d\n", ConvertData(RxGroup)); // 对于 {0x12, 0x34, 0x56}, 正确结果应为
1193046
    return 0;
}
```

选做部分：实时操作系统 (RTOS)

(1) 基础

请简要回答以下问题。

1. 之前是否有多线程开发经验？是否使用过 RTOS？用的是哪款？
2. 对于单核 MCU，同一时刻只能执行一条指令。那么在使用 RTOS 后，多个线程是如何“同时”运行的？
3. 列举出 3 个将运行中的线程阻塞的方法。

(2) 线程安全

请简要回答以下问题。

1. 多个线程同时读写单个 `int8_t` 类型的全局变量时，是否需要考虑线程安全问题？为什么？
2. 多个线程需要读写同一个全局变量时，怎样保证线程安全？
3. 有两个优先级相同的线程循环调用同一个函数（如下），其中一个线程调用 `AccumulativeAdd(1)` 1000000 次，另一个线程调用 `AccumulativeAdd(2)` 1000000次。若两个线程同时启动，当它们都执行完毕后，`AccumulativeAdd` 中的变量 `previous` 是否一定为 3000000？为什么？

```
int AccumulativeAdd(int num)
{
    static int previous = 0;
    previous = previous + num;
    return previous;
}
```