多线程累加求和程序编写

01

多线程累加求和程序编写

运动程序代码和上面例程相同。应用程序负责编译、下载、初始化、启动运动程序，不同之处在于启动2个线程完成累加运算任务，代码如下。

```
int main(int argc, char* argv[])
{
    short rtn;
    short funId;
    TVarInfo sum, begin, end;
    double value;
    TThreadSts thread;
    // 打开运动控制器
    rtn = GT_Open();
    printf("GT_Open()=%d\n", rtn);
    // 复位运动控制器
    rtn = GT_Reset();
    printf("GT_Reset()=%d\n", rtn);
    // 下载运动程序sum.bin
    // 必须保证sum.bin文件位于工程文件夹中
    rtn = GT_Download("sum.bin");
    printf("GT_Download()=%d\n", rtn);
```

01

多线程累加求和程序编写

```
// 获取函数ID
rtn = GT_GetFunId("add", &funId);
printf("GT_GetFunId()=%d\n", rtn);
// 获取全局变量sum的ID
rtn = GT_GetVarId(NULL, "sum", &sum);
printf("GT_GetVarId()=%d\n", rtn);
// 获取局部变量begin的ID
rtn = GT_GetVarId("add", "begin", &begin);
printf("GT_GetVarId()=%d\n", rtn);
// 获取局部变量end的ID
rtn = GT_GetVarId ("add", "end", &end);
printf("GT_GetVarId()=%d\n", rtn);
// 绑定线程，函数，数据页
rtn = GT_Bind(0, funId, 0);
printf("GT_Bind()=%d\n", rtn);
// 绑定线程，函数，数据页
rtn = GT_Bind(1, funId, 1);
printf("GT_Bind()=%d\n", rtn);
value = 0;
```

01

多线程累加求和程序编写

```
// 初始化运动程序的全局变量sum
rtn = GT_SetVarValue(-1, &sum, &value);
printf("GT_SetVarValue()=%d\n", rtn);
value = 1;
// 初始化运动程序的局部变量begin
rtn = GT_SetVarValue(0, &begin, &value);
printf("GT_SetVarValue()=%d\n", rtn);
value = 50;
// 初始化运动程序的局部变量end
rtn = GT_SetVarValue(0, &end, &value);
printf("GT_SetVarValue()=%d\n", rtn);
value = 51;
// 初始化运动程序的局部变量begin
rtn = GT_SetVarValue(1, &begin, &value);
printf("GT_SetVarValue()=%d\n", rtn);
value = 100;
// 初始化运动程序的局部变量end
rtn = GT_SetVarValue(1, &end, &value);
printf("GT_SetVarValue()=%d\n", rtn);
```

01

多线程累加求和程序编写

```
    // 启动线程
    rtn = GT_RunThread(0);
    printf("GT_RunThread()=%d\n", rtn);
    // 启动线程
    rtn = GT_RunThread(1);
    printf("GT_RunThread ()=%d\n", rtn);
    do
    {// 查询线程状态
    rtn = GT_GetThreadSts(0, &thread);
    }while( 1 == thread.run ); // 等待线程运行结束
    do
    {// 查询线程状态
    rtn = GT_GetThreadSts(1, &thread);
    }while( 1 == thread.run ); // 等待线程运行结束
    // 查询全局变量sum的值
    rtn = GT_GetVarValue(-1, &sum, &value);
    printf("sum=%-10.0lf", value);
    getchar();
    return 0;
}
```

谢　谢　观　看