

Application of Multithreading Technology in Data Acquisition of Cement Enterprises

Wenkang Zuo^{1, a}, Xiaohong Wang^{2, b} and Shaohong Jing^{2, c}

¹School of Control Science and Engineering, University of Jinan, Jinan 250000, China;

Keywords: cement, energy saving, data acquisition, C#, multithreading.

Abstract. Cement is the basic raw material for China's national economic construction, and cement enterprises are the major energy consumption, but also the focus and difficulty of energy saving and consumption reduction in the process of China's industrial process, So the data acquisition in the cement enterprise energy management should be particularly important. In view of the problems of existing single threaded data acquisition in the presence of low CPU utilization rate, a multithreading technology in data acquisition based on C# is introduced in this paper. This paper introduces the concept, advantages of multithreading and its significance briefly at start, then describes the application of multithreading technology in data acquisition according to SerialPort class and Thread class of .NET VS2012 programming environment.

Introduction

Cement is the basic raw materials of national economic construction of China, occupies a very important position in the development of our national economy and has become the foundation of China's economic construction. Besides, cement is the important support of China's industrialization construction and still one of the main building materials of social development in the future. The cement industry is basic industry and traditional industry of China, but it is also China's high energy consumption and emission industry, which has become the key points and difficulties of energy saving and consumption reduction in the process industry of China. Cement enterprises have always been a major energy consumption, and a considerable part of the energy consumption of cement enterprises is due to monitoring the use of energy in a timely manner and using energy unreasonable at present. Taking electric energy as an example, a lot of electric power equipment within abnormal operation, unnecessary empty operation and unreasonable production scheduling, which leads to wasting a lot of electricity every year^[1]. Therefore, it is necessary to reflect the energy consumption in each production link accurately, rapidly and timely, so data acquisition plays an important role in cement enterprises. But the existing data acquisition of cement enterprises is single thread, which exists some problems, such as low CPU utilization rate, lousy independence between tasks and poor system reliability. In order to solve these problems, multithreading technology is an ideal choice.

Process and Thread. Every program that is running in the system is a process, containing one or more threads. Processes may be the dynamic execution of the entire program or part of the program.

Thread is a set of instructions or a special section of a program, which can be executed independently in the program.

Process. The process is an instance of the application, and it is a dynamic execution of the application as well. Each process is composed of a private virtual address space, code, data, and other resources. Resources that created by the running process will be destroyed with the process termination, meanwhile, the using resources will be released on or off.

Thread. A thread is an execution unit within a process, and a new thread generated in the process is called a free thread. After the process is generated by the system, in fact, the main implementation process is started, which provides the starting point of the program to windows system in the form of address, such as main function. The main thread is terminated, and the process terminates.

Each process has one main thread at least, which does not need to be created by the user, it is created by the system automatically. The user creates other thread in the application according to the

application according to the need, and the multiple threads can run in parallel in the same process^[2]. All threads that in same process is in the virtual address space of this process, they use the virtual space, global variables and system resources jointly , so the communication between threads are very convenient.

Multithreading. Multithreading is a technique to implement multiple threads from software or hardware. Multithreading is used to complete a number of tasks, not to improve the efficiency of the operation, by improving the utilization efficiency of resource to improve the efficiency of the system. Threads are implemented when system needs to complete a number of tasks at the same time.

Advantages of Multithreading. Using multithreading can transfer the task that take a long time in the process into the background to deal with, release a number of precious resources such as memory footprint, and stop the task at any time. Threads share the memory and resources of their processes by default, and threads which in the same process have many common. Because of thread sharing process resources, there is no need to allocate system resources for this thread when you create a thread. It can share the resources of the process directly, and the relative speed to be faster. When switching between different threads, the time cost is less than that of the process. The multithreading technology makes the program response faster, because the user interface can be in a state of activity while it carries out other work.

Meaning of multithreading. The application of multithreading technology can make the best use of system resources. The main advantage of multithreading technology is to make full use of CPU free time, respond to user requests as far as possible, improve the overall efficiency of the process to a great extent, enhance the flexibility of the application at the same time^[3]. What's more, because all threads of the same process share the same memory, there is no need for a special data transfer mechanism, and there is no need to establish a shared memory area or a shared file, so that it is more easy to solve the questions such as coordination operation and operation of different tasks, data interaction, resource allocation and so on.

C# Support for Multithreading. .NET Framework supports free thread, so we can use free thread in all .NET languages, including C# and other language. Any language under the framework of .NET, when writing multithreaded applications, we can use the object and method system provided, rather than Win32API, so it can greatly reduce the difficulty of development and reduce the occurrence of errors^[4].

The .NET runtime monitors all the threads created by the .NET code. It also monitors all unmanaged threads that can execute managed code. When the unmanaged code executes in a managed thread, the runtime checks the security transport layer protocol, determines whether there is a managed Thread object. If you find a managed thread, the runtime uses the thread, otherwise the runtime will create one and use it. No matter where the thread comes from, we all want to get its object representation.

Multithreading Data Acquisition. Multithreading data acquisition includes many functional modules, the most critical part of which is the Serial Communication and management of multithreading.

Serial Communication. When programming in Windows, there are three main methods of serial communication: MSComm communication controls, Windows API, the third party to provide a number of serial communication classes^[5].

Using MSComm control can make programming simple, however, the controllability and flexibility of programming will downgrade due to a large number of packages, so it is necessary to do a lot of complex processing in the multithreading serial programming. Programmers are not only familiar with the thread of the programming method, but also skilled use of a large number of API functions when using Windows API to write serial procedures. It is efficient to programme with some serial communication classes provided by third party, and program controllability is strong. SerialPort class is the most widely used in the C# language, which is based on multithreading and used to control the serial port file resources. Besides, the functionality of SerialPort class can be packaged in a Stream object, and it also can be visited through the property of BaseStream.

By comparing the three kinds of methods of communication, you can find the best choice of programming in Windows system is SerialPort class provided by third party. Serial communication programming steps are as follows.

Initialization of serial port. Firstly, the information needed to initialize the serial port is obtained by reading the database or reading the configuration file, Including serial number, serial port configuration parameters and so on. Secondly, to create an object of the SerialPort class and assign the properties of the object, such as the BaudRate property is used to get or set the baud rate, the Parity property is used to get or set the parity of checking protocol, the DataBits property is used to get or set the length of standard data bit for each byte and the StopBits property is used to get or set the standard stop number for each byte. Related code is as follows:

```
private SerialPort serialPort;           //Declare object type
private void InitializeSerialPort()
{
    serialPort = new SerialPort();       //Create objects
    serialPort.BaudRate = 9600;          //BaudRate
    serialPort.Parity = Parity.Even;     //Parity
    serialPort.DataBits= 8;              //DataBits
    serialPort.StopBits = StopBits.One;  //StopBits
}
```

Connection of serial port. The PortName property of SerialPort class is to get or set the port, the role of the Open method is to open a new serial port connection. Related code is as follows:

```
private void ConnectPort(object sender, EventArgs e)
{
    serialport.PortName ="COM5"; //Get serial port number
    serialPort.Open();           //Open the serial port
}
```

Packet sending. The generated message must be converted into an array of bytes before the message is sent, then to write the message in the form of byte into the serial port output buffer by Write method of SerialPort class. Related code is as follows:

```
private void ReportSend(object sender, EventArgs e)
{
    byte[] tosend = HexStringToByte(ReportCode);
    serialPort.Write(tosend, 0, tosend.Length);
}
```

The function HexStringToByte is custom and used to convert a message from a string to an array of bytes.

Packet reception. DataReceived event will be triggered when there is data to the serial port of the read cache, which is the method for processing data received of SerialPort object. In this event, the Read method of SerialPort class can be used to read the returned data from the SerialPort input buffer.

Creation and Management of Multithreading. The creation and management of multithreading is achieved through the use of System.Threading.Thread class. The basic task of the thread is achieved by Thread class, and multithreading can be used to complete many things. Generally, we create a Thread object that operates the new thread to create a new thread. To accomplish many different tasks, threads must be managed reasonably and effectively, common management methods such as the management of thread life cycle, thread revocation and multithreading scheduling.

The management of thread life cycle. There are two ways to call the Thread.Sleep method. The first way by calling the Thread.Sleep with a value of 0, the second way to achieve through the use of Thread.Suspend method. The difference between the two methods: the first way can call the Thread.Suspend method in the current execution of a thread or on another thread; once a thread is suspended in the second way, only with the Thread.Resume method can it be restarted by another

thread. It should be noted that the Thread.Sleep method is a static method that cannot be called with an instance of the Thread object. In addition to the currently executing thread, the Thread.Sleep is not allowed in any other thread.

Thread revocation. If you need to undo a thread, you can call the Thread.Abort method to complete. The thread does not immediately stop running until the thread reaches the security point of the document when the Thread.Abort method is called. In order to ensure that the thread is stopped, Thread.Join can be used which is a synchronous method, and it will not return until the thread has been stopped.

Multithreading scheduling. The Priority property of the Thread class is used to observe and set the priority of the thread, the value of which is Highest, AboveNormal, Normal, BelowNormal, Lowest. The priority of the threads are created by default to Normal during run time, and the threads are created to maintain their original priority during a run outside. Processor can select the thread to be executed according to the priority of the thread if the time slice of a given thread ends When the it changes between multiple threads^[6]. Combined with the serial communication programming, the code of creation and management of multithreading as follows:

```
public static void Main(string[] args)
{
    //Create a ThreadStart class object, and call the method
    ThreadStart worker1 = new ThreadStart(InitializeSerialPort);
    ThreadStart worker2 = new ThreadStart(ConnectPort);
    ThreadStart worker3 = new ThreadStart(ReportSend);
    //Create threads
    Thread t1 = new Thread(worker1);
    Thread t2 = new Thread(worker2);
    Thread t3 = new Thread(worker3);
    //Multithreading scheduling
    t1.Priority = ThreadPriority.Highest;
    t2.Priority = ThreadPriority.Normal;
    t3.Priority = ThreadPriority.Lowest;
}
```

At the same time, in order to ensure that the members of an object has always maintained a legal state When using multithreading, the safety of thread is also very important. So only a thread is allowed to enter the key part of the code by using the means of synchronization to avoid unpredictable States in data acquisition.

Summary

In view of the existing problems of cement enterprises about single threaded data acquisition, multithreading technology is proposed and elaborated which base on C# language in this paper. C# is a new, simple, efficient, type safe and object-oriented programming language introduced in the.NET platform, which inherits the powerful features of the C/C++ language, and enables developers to develop application program efficiently and quickly. Multithreading technology is a better choice for concurrent execution of application program, which can make full use of CPU reasonably, solve the problem of switching between tasks, improve the independence of tasks, enhance the stability and reliability of data acquisition, and provide a strong basis for energy management and consumption reduction in cement enterprises.

Acknowledgments

This work was supported by China-EU SMEs Energy-Saving and Emission-reduction Research Collaboration Fund Project under Grant SQ2013ZOC600003, Science and Technology Major Project

of Shandong Province under Grant 2015ZDXX0101F01, Independent Innovation and Achievement Transformation Project of Shandong Province under Grant 2014CGZH0601.

References

- [1] Runhua Wang. Argumentation of Management in electric power requirement, Government agency energy-saving, Avoid the rush time[J]. Distributed power generation and integration technology, 2008, 1-8.
- [2] Ayose Falcon,Alex Ramirez,Mateo Valero.A low-complexity,high-performance fetch unit for simultaneous multithreading processors[C]. Spain:Proceedings 10th International Symposium on HPCA-10,2004:244-253.
- [3] Yanlin Zhang.Implementation of Multithread Technique Based on VB.NET[J]. Application of Computer System, 2009, (2): 136-138.
- [4] Min WANG. C# Threading Handbook[M]. Beijing: Tsinghua University Press, 2003: 13-19.
- [5] Dakui Jiang. Serial Communication to Achieve Real-time Data Acquisition[J]. 2007, 17(4): 15-16.
- [6] Jingren Zhou, John Cieslewicz, Kenneth A Ross, et al. Improving database performance on simultaneous mutithreading processors[C]. Trondheim, Norway: Proceedings of the 31st VLDB Conference, 2005: 49-60.