

基于事件驱动的实验室管理软件数据处理方案

施咏月 鲍 瀛

(南京大学医学院附属鼓楼医院 南京 210008)

〔摘要〕 提出一种基于事件驱动的实验室信息管理系统数据处理方案, 阐述基本原理、具体流程以及实际数据解析情况等方面, 指出该方案有助于实现系统数据高效获取与解析, 提高检验技师工作效率。

〔关键词〕 实验室管理软件; 通信程序; 事件驱动; 多线程; 数据处理

〔中图分类号〕 R-058 〔文献标识码〕 A 〔DOI〕 10.3969/j.issn.1673-6036.2021.09.013

A Data Processing Scheme of Laboratory Management Software Based on Event Driven SHI Yongyue, BAO Ying, Nanjing Drum Tower Hospital, The Affiliated Hospital of Nanjing University Medical School, Nanjing 210008, China

〔Abstract〕 A data processing scheme of Laboratory Information Management System (LIMS) based on event-driven is proposed. The paper expounds the basic principles, the specific process, the actual data analysis situation, etc., and points out that this scheme is helpful to realize the efficient acquisition and analysis of system data and improve the working efficiency of medical laboratory technicians.

〔Keywords〕 laboratory management software; communication program; event driven; multi-thread; data processing

1 引言

随着信息化发展, 实验室信息管理系统 (Laboratory Information Management System, LIMS) 在医院检验科室普遍使用, 大幅提高检验技师工作效率和质量, 实现检验结果可追溯^[1]。LIMS 将检验仪器与计算机连接起来, 直接收集仪器检验结果, 通过硬件传输提交到计算机 LIMS 终端中, 同时将数据保存在数据库, 便于打印标准格式纸制报告和向平台提交电子数据报告, 在后期查询统计中 LIMS 数据有助于统计分析。由于仪器厂商生产工艺和标

准不同、缺乏通用仪器与计算机通信标准协议等原因, 现有仪器与计算机接口协议种类多样, 如卫生信息交换标准 (Health Level 7, HL7) 协议^[2], 美国材料实验协会 (American Society of Testing Materials, ASTM) 协议^[3]等, 导致 LIMS 解析仪器结果数据程序无法固定。因此需在 LIMS 客户端外, 针对每台仪器另行开发单独通讯解析程序, 负责仪器与 LIMS 间的数据及指令交互, 见图 1。对此本文提出基于事件驱动的数据交互和解析方法。

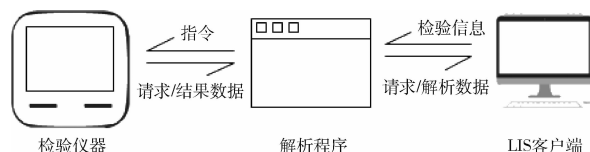


图 1 仪器、解析程序和 LIMS 客户端交互流程

〔修回日期〕 2021-04-24

〔作者简介〕 施咏月, 硕士, 助理工程师, 发表论文 2 篇; 通讯作者: 鲍瀛, 主管护师。

2 基于事件驱动的数据交互和解析方法原理

2.1 现有数据交互和解析存在的问题及相应措施

2.1.1 现有问题 现阶段仪器与计算机通信方式主要是串口和网口通信。部分自带上位机控制软件的仪器可通过将数据写在本地数据库或文件中的方式由 LIMS 解析程序主动读取。此种方式较简单,由解析程序定时读取即可。而通过串口或网口通信方式进行数据交互受硬件传输或数据格式等限制,导致数据交互和解析流程更加复杂。通常仪器检测标本后自动通过串口或网口发送检验结果数据,如果 1 次交互能够发送完成 1 个完整结果数据包,解析程序收到数据包后即可直接解析。但在实际过程中经常出现 1 个结果数据被分成多个数据包进行发送的情况,此时解析程序需等待确定接收到完整数据结果后才能开始解析。因此确定可开始解析数据的时机是影响程序稳定性和效率的关键因素。数据不完整便开始解析可能导致结果数据缺失甚至解析错误,数据接收完整后未开始解析则会影响解析程序效率。综上所述解析程序通信方式确定后,如何保证复杂数据交互解析的稳定性和效率是解析程序的核心工作。准确定位解析数据的最优时机,便可以提供可靠检验数据结果并缩短仪器检验出标本结果到结果显示在 LIMS 的时间,进而提高检验技师工作效率。

2.1.2 相应措施 针对上述问题可创建新的线程,持续监控串口或网口接收数据的缓存区数据,监测到缓冲区存在数据即取出处理,如代码段 (1):

```
while (! str.IsEmpty ()) {} (1)
```

但此种方法会占用大部分的中央处理器 (Center Processing Unit, CPU), 较耗费电脑资源, 明显降低电脑运行速度。对此可在监控数据缓存区的基础上添加线程休眠, 如代码段 (2):

```
for (; ! str.IsEmpty (); std:: this_thread:: sleep_ for (std:: chrono:: minutes (5))) {} (2)
```

此方法可实现不持续占用计算机资源, 但在响应不及时等缺点。

2.2 基于事件驱动的数据交互和解析方法

鉴于上述问题, 提出基于事件驱动的数据交互和解析方法。在解析程序初始化时创建 3 个线程, 根据具体功能分别命名为接收线程、取线程和解析线程。3 个线程间通过事件进行相互驱动。具体流程为: 接收线程负责接收仪器通过串口或网口发送的数据, 接收到数据即放到缓冲区中, 同时将其作为 1 个事件驱动激活取线程并即时查找数据缓冲区中完整的标本数据。如果取线程能找到完整标本结果数据, 则作为 1 个事件去驱动激活解析线程, 进行解析数据并保存结果, 从缓冲区删除已解析数据。通过此方法不论标本结果数据分多段发送还是 1 次发送均可以稳定快速地解析, 得出正确样本结果数据。

3 基本知识

3.1 仪器结果数据格式

3.1.1 HL7 协议 医院检验实验室使用仪器的 1 个完整样本数据称作 1 帧数据。通常 1 帧数据包括帧头、正文和帧尾 3 部分。通过查找帧头和帧尾位置可得到完整数据正文进行解析。如部分网口通讯仪器常用 HL7 协议数据, 格式如下:

```
<SB> msg <EB> <CR>
```

其中 <SB> 为帧头, 通常 ASCII 码对应 <VT>, 即 <0x0B>; msg 为正文, 即消息主体内容; <EB> 为帧尾, 通常 ASCII 码对应 <FS>, 即 <0x1C>; <CR> 为 ASCII 码回车符, 即 <0x0D>, 见图 2。

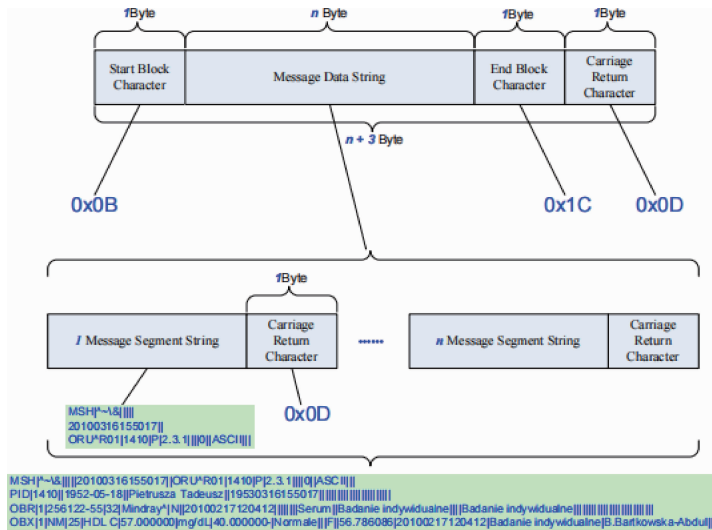


图2 HL7 协议数据格式

3.1.2 ASTM 协议 串口通信常用的 ASTM 协议数据格式如下：

< SB > msg < EB >

其中 < SB > 为帧头，通常 ASCII 码对应 < STX

>，即 < 0x02 >；msg 为正文，即消息主体内容；< EB > 为帧尾，通常 ASCII 码对应 < ETX >，即 < 0x03 >，见图 3。

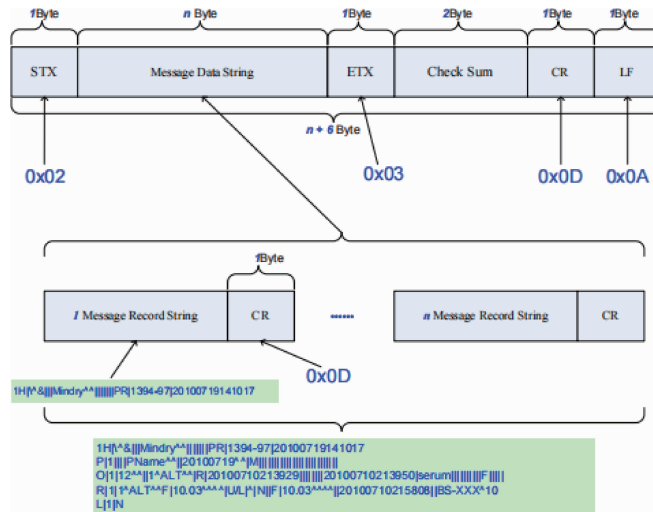


图3 ASTM 协议数据格式

3.2 线程

3.2.1 多线程并行原理 在操作系统中进程可以被看作是 LIMS 解析程序的 1 个运行实例。线程是运行在进程中进行运算调度的最小单位，1 个进程中可以只运行 1 个线程，也可多个线程同时运行^[4]。1 个线程就是进程中的 1 条单向顺序的控制流，进程中每个线程可并行异步执行不同任务。同

一进程中的多个线程可以共享同进程下系统资源，如虚拟地址空间、文件描述符和信号等，本文中存储仪器原始结果数据的缓存区域可以供多个线程共享使用。

3.2.2 多线程并行设计方法 在程序设计中多线程并行设计可以提高执行效率，通过采用多线程可以将程序中负责输入输出处理和解析计算的部分异步执行，解决接口交互阻塞问题^[5]。如解析程

序中创建的 3 个线程中, 接收线程负责从串口或网口获取数据并存入数据缓存区中, 取线程从数据缓存区中查找完整结果数据, 而解析线程只需解析标本结果数据存入 LIMS 中。3 个线程可以并行运行, 相互之间有一定关联又保持相互独立。

3.3 事件机制

3.3.1 使用介绍 以 C++ 语言为例, 其主要使用 5 个应用程序接口 (Application Programming Interface, API) 函数分别表示创建事件 (CreateEvent ())、开启事件 (SetEvent ())、重置事件 (ResetEvent ())、等待时间内事件状态 (WaitForSingleObject ()) 和发生事件序号 (WaitForMultipleObject ())。事件机制即为 1 个进程对已定义的 1 个属性或变量标识的值进行检测, 进而判断后续对应执行操作。

3.3.2 使用描述 以取线程为例, 当接收线程接收到数据并存入数据缓存区后, 通过事件 m_ ReaderEvent 驱动取线程运行, 即调用 m_ ReaderEvent.SetEvent ()。

```

UINT ReaderProc (LPVOID lpParam)
{
HANDLE hArray [2];
DWORD dwReturn;
BOOL fDone = FALSE;
hArray [0] = m_ ReaderEvent;
hArray [1] = m_ ThreadExitEvent;
while (! fDone)
{
dwReturn = WaitForMultipleObjects (2, hArray, FALSE,
WRITE_ CHECK_ TIMEOUT);
switch (dwReturn)
{
case WAIT_ TIMEOUT:
break;
case WAIT_ FAILED:
break;
case WAIT_ OBJECT_ 0:
ReaderDeviceData ();
break;
case WAIT_ OBJECT_ 0 + 1:

```

```

fDone = TRUE;
break;
}
}
return 1;
}

```

当 m_ ReaderEvent 事件被开启后, 取线程中 ReaderData () 则会被调用, 进行数据缓存区读取并查找完整结果数据。如果找到完整结果数据则开启其对应事件驱动解析线程, 开始解析数据。

4 基于事件驱动的数据处理流程

4.1 概述

完整结果数据是解析程序完成解析的基本条件。基于事件驱动的解析方法可以稳定、高效地找到完整结果数据。分析从仪器发送结果到解析程序解析完成的过程, 为提高运行效率, 在程序中创建 3 个线程并行地执行各自功能, 分别定义为接收线程 ReceiveThread、取线程 ReaderThread 和解析线程 AnalysisThread。线程之间通过事件驱动激活, 实现交互。

4.2 线程构建与驱动

4.2.1 接收线程 负责从串口或网口中接收仪器结果数据并存储到数据缓存区中。接收线程启动后阻塞在串口或网口的监听事件中, 当串口或网口接收到数据时, 接收线程即继续工作, 将所接收数据添加到数据缓存区并激活取数据的事件, 以驱动取线程进行工作。

4.2.2 取线程 负责从数据缓存区中查找并取出完整标本结果数据。根据数据格式特点, 每个完整数据均有头尾标志。在数据缓存区中, 从头开始找到数据帧头和帧头位置后的第 1 个帧尾, 其间的的数据即是 1 组完整的标本结果数据。从数据缓存区中取出该组结果数据并交给解析线程, 将取出 1 组完整的标本结果数据作为激活解析线程的事件, 驱动解析线程开始工作。重复上述取数据操作, 直到缓存区中没有完整结果数据则取线程停止, 等待下一次被事件驱动激活。

4.2.3 解析线程 每接收到1个完整仪器标本结果数据,即按照具体数据解析协议获取检验时间、样本号、通道号和结果等信息,组装成固定格式保存到LIMS中。

4.2.4 线程及其事件驱动流程 3个线程均通过事件进行驱动激活,从而执行1次工作任务,不会一直占用操作系统资源而造成资源浪费或系统卡顿。接收线程由硬件接收到数据作为接收事件

被驱动,完成1次接收数据操作后,通过取数据事件驱动取线程;取线程通过查找获取到数据并分隔定位到完整仪器标本数据后,通过解析事件驱动解析线程;解析线程将解析出的每个项目结果发送到LIMS中进行保存。3个线程可并行处理各自工作,不会出现获得完整结果数据后程序未开始解析的现象,保证数据处理效率,见图4。

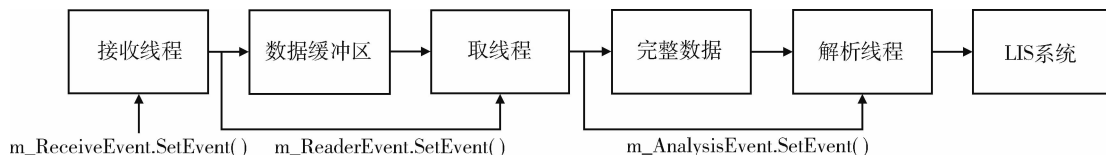


图4 线程及其事件驱动流程

4.3 具体流程

步骤(a):接收线程等待被触发驱动。当串口或网口接收到仪器传输的结果数据,接收线程被激活便开始工作,将接收到的结果数据存入数据缓存区。步骤(b):数据缓存区被添加新数据后,取线程被触发激活。查找获得完整结果数据后便进行步骤(c),所有完整结果都被循环取完,或是没有找到完整结果数据便返回到步骤(a)。步骤(c):得到1个完整标本结果数据,开始解析得到标本检验时间、样本号、通道号和对应结果等信息,见图5。

5 实际数据解析

5.1 结果数据解析展示

以电化学发光免疫分析仪(型号Cobas e411)真实数据为例进行结果数据解析展示。可能出现3种情况:仪器1次传输1个完整结果数据,是理想格式;1个标本数据被仪器分成多段进行传输,即多次接收到数据叠加整合后才是1个完整结果数据;仪器1次传输多个完整结果数据,即需根据数据格式拆分成多个完整结果数据进行解析。使用串口调试工具模拟仪器发送数据,分别实现以上可能出现的传输情况。

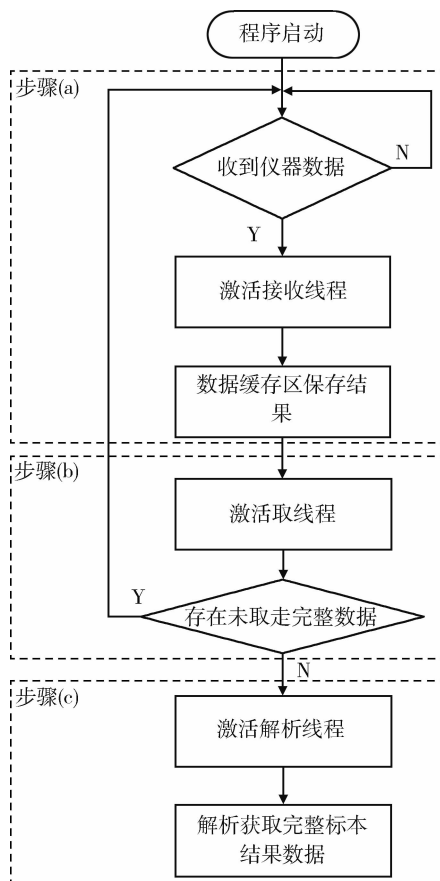


图5 基于事件驱动的数据处理流程

5.2 仪器 1 次传输 1 个完整结果数据的情况

此场景为理想传输和解析情况，由解析程序日志可以得出解析程序接收到数据并存入数据缓存区后立即激活取线程，由于传输的是 1 个完整数据，取线程获取到完整数据结果交给解析线程，解析成功。

5.3 仪器多次才传输完 1 个完整结果数据的情况

由解析程序日志可以看出，解析程序接收到数据并存入数据缓存区后立即激活取线程，由于取线程在数据缓存区中未找到完整结果数据，所以继续等待，接收到新数据后再查找完整数据。当数据缓存区内有完整数据后，取线程即时取到完整数据结果交给解析线程，解析成功。

5.4 仪器 1 次传输多于 1 个结果数据的情况

由解析程序日志可以得出，解析程序接收到数据并存入数据缓存区后立即激活取线程，由于取线程在数据缓存区中只找到 1 号标本完整结果数据，找到后立即交给解析线程并异步解析成功。等待接收到新数据后继续查找完整数据，当数据缓存区内有 2 号标本完整数据后，取线程即时取到其结果交给解析线程，解析成功。

6 结语

本文从提高检验数据解析效率出发，提出一种基于事件驱动的实验室管理系统数据处理方法，通过构建接收线程、取线程和解析线程 3 个线程，以关键节点事件驱动激活各线程工作，3 个线程通过相互关联又彼此独立的工作模式将不同标准实验室仪器数据以高效、可靠的方式进行处理并提供到 LIMS。在节约操作系统计算资源的同时，可减少数据解析处理时延，为 LIMS 提供准确、完整结果数据，提高医疗机构检验技师工作效率。

参考文献

- 1 陈莉月, 杨安兴, 潘文晖. 实验室信息管理系统发展与国内应用概述 [J]. 中国管理信息化, 2020, 23 (23): 194-196.
- 2 张晓, 龙伟, 卢斌. 基于 HL7 的移动式微型临床实验室 Mini Lis 接口设计 [J]. 计算机测量与控制, 2015, 23 (4): 1367-1370.
- 3 陈铁英, 贺嘉嘉, 高育林. 基于 ASTM 的检验仪器双向通信控制的研究与实现 [J]. 医学信息, 2008 (5): 577-580.
- 4 郝文化, 文自勇, 王浩强, 等. Windows 多线程编程技术与实例 [M]. 北京: 中国水利水电出版社, 2005: 124-127.
- 5 贾广雷, 刘培玉, 耿长欣. 多线程技术及其在串口通信中的应用 [J]. 计算机工程, 2003 (1): 247-249.

(上接第 21 页)

- 21 Krabbenhoef N, Skinner K, Schultz M, et al. Chronicles in Preservation: preserving digital news and newspapers [J]. Preservation, Digital Technology&Culture, 2013, 42 (4): 199-203.
- 22 Van Garderen P, Mumma C C. Realizing the Archivemata Vision: delivering a comprehensive and free OAIS implementation [C]. Lisbon: 10th International Conference on Preservation of Digital Objects, 2013.
- 23 Abrams S, Kunze J, Loy D. An Emergent Micro-services

- Approach to Digital Curation Infrastructure [J]. International Journal of Digital Curation, 2010, 5 (1): 172-186.
- 24 肖铮, 林俊伟. 用微服务架构下一代图书馆服务平台——以 FOLIO 为例 [J]. 图书馆杂志, 2018 (11): 63-69.
- 25 Jaramillo D, Nguyen D V, Smart R. Leveraging Microservices Architecture by Using Docker Technology [C]. Norfolk: IEEE SoutheastCon 2016, 2016: 1-5.
- 26 方安, 杨晨柳, 王军辉. WHO 西太平洋地区医学索引收录期刊的数据提交能力分析 [J]. 医学信息学杂志, 2016, 37 (8): 79-83.