



单片机中环形队列取余运算的改进

梁凤涛

(南京苏美达智能技术有限公司 研发部,南京 210000)

摘要: 针对环形队列入队、出队时取余运算耗时问题,提出一种使用按位与运算的方法,大大缩短了入队、出队耗时;针对批量写入/读出数据时,使用单字节方式效率低的缺点,提出了一种对队列一次性读/写多字节的实现方式。实际测试结果表明,本文提出的方法耗时更少、执行效率更高。

关键词: 环形队列;MCU;嵌入式系统

中图分类号: TP311

文献标识码: A

Improvement of Ring Queue Remainder Calculation in MCU

Liang Fengtao

(R & D Department, Nanjing SUMEC Intelligence Technology Co., Ltd., Nanjing 210000, China)

Abstract: In order to solve the problem of time consuming in the process of queue entry and queue exit, this paper proposes a method of bit-sum operation, which greatly reduces the time consuming in queue entry and queue exit. In view of the low efficiency of using single byte mode when writing and reading data in batch, this paper proposes a way to read and write multiple bytes in a queue at one time. The actual test shows that the proposed method is less time-consuming and more efficient.

Keywords: ring queue; MCU; embedded system

0 引言

环形队列是一种很重要的数据结构,在软件开发中有广泛应用。因为在硬件底层只需执行数据入队操作,在业务层只需判断队列长度并进行数据读取或者执行数据出队后再进行数据处理即可,因而可获得良好的软件分层效果,广泛应用于嵌入式系统中。需要注意的是,环形队列使用的内存分为动态内存和静态内存。受资源所限,嵌入式系统一般采取静态内存实现环形队列。相较于动态内存,静态内存分配的环形队列可以得到更高的访问效率和内存利用率,有效避免了内存碎片产生。本文所述环形队列的内存分配方式为静态内存分配,在此基础上本文提出了按位与运算访问队列及一次性读写多字节的实现方式。

1 常规环形队列实现

实现环形队列需要如下操作:入队(将数据压入队列)、出队(将数据从队列中取出)、队列状态判断(队列空、队列满、队列非空非满)。其中所涉及到的变量有:写索引 wIndex、读索引 rIndex、数组 buf、数组大小 bufSize。其中,数组及数组大小在开始时给定,入队时更新写索引,出队时更新读索引。

队列状态判断的实质就是计算当前队列内的有效数据长度。一般有两种方法可以实现:①根据读索引 rIndex 与写索引 wIndex 关系判定。若 wIndex 等于 rIndex,则队列为空;若 (wIndex+1) 等于 rIndex,则队列为满,此时需要注意的是当 (wIndex+1) 等于 bufSize 时,需要减去 bufSize 或者对 bufSize 取余后再与 rIndex 比较。其余情况则为非空非满。在这种方法中,队列的最后一个字节始终是不可用的。②引入新变量队列实时长度 len。当执行入队操作时 len 累加,出队时 len 递减,根据 len 与 bufSize 的关系可以判断队列的当前状态。两种方法没有优劣之分,只是实现方式不同。本文队列状态判断采用第一种方法。

索引更新:索引在更新时执行累加操作。当索引累加后等于 bufSize 时,索引重新设置为 0。

入队流程:入队时,首先判断队列是否为满。若队列为满,则返回操作失败;若队列不满,则将数据压入队列后更新写索引 wIndex。

出队流程:出队时,首先判断队列是否为空。若队列为空,则返回操作失败;若队列不空,则将数据从队列取出后更新读索引 rIndex。

2 常规环形队列的改进

常规环形队列的实现适用于单个字节写入的情况,对

应用情况常见于嵌入式单字节通信接收中断。随着技术的发展,MCU资源也日益丰富,且有时出于通信速度考虑,经常使用DMA通信方式,一次性批量接收长数据。此时,常规环形队列实现的数据存储方式无法满足需求。为了改善这一问题,本文提出一种改进方法,可以对队列实现一次性读写多字节,进而提高队列的读写效率。

在改进之前,先需对队列数据访问的效率进行研究。在实际应用中,设备间的通信都会使用通信协议对数据格式进行约定。通信协议一般分为4部分:帧头、帧长度、校验值、数据正文。一般帧头和帧尾都是固定1字节或多字节的数据,如0x55、0xAA。更多帧格式相关内容此处不再讨论。

由于使用的是环形队列,考虑到数据粘包的可能情况下,并不能事先确定帧头/帧尾出现在队列的何处,因此在数据处理时,首先要遍历队列,找到帧头在队列中的索引值。遍历时,索引有可能会跨过队尾,因此要注意索引的值不应超过(bufSize-1)。进一步的,找到帧头并不意味着帧格式正确,需要进一步判断当前帧长度是否合规,校验值是否正确,这都涉及到队列的数据访问。常规的队列访问方式是使用取模运算,即索引index对数组大小bufSize取余,才不会造成数组访问越界。取模运算实际上是除法运算,相较于加减、位运算等运算符,效率还是比较低的,比较浪费资源。为了解决这个问题,本文设计了一种规则来避免取模运算:将队列数组大小设置为2的N次幂,并引入新变量速算数fastNum= $2^N - 1$ 。在访问队列时,使用按位与运算,即索引index与速算数fastNum进行按位与运算。此种访问方式可有效减少队列数据访问时间,提升队列数据访问效率。

结束了队列数据访问效率研究后,我们开始改进常规环形队列,使其可以一次性读写多个字节。一次性写多字节时也需要注意两个变量:队列剩余可用空间和写索引wIndex到队尾容纳的字节数(跨队尾时需要用到)。一次性读多字节时需要注意两个变量:队列当前数据长度和读索引rIndex到队尾容纳的字节数(跨队尾时需要用到)。在多字节读写时,可以使用C库中的memcpy内存复制函数直接复制内存,进而加快数据读/写速度。

改进入队流程如下(假设待写入的数据长度为wLen):
①首先判断队列剩余空间是否足够。②若剩余空间不足,则返回入队操作失败。③若剩余空间足够,则计算当前写索引wIndex到队尾的长度len1。④若len1不小于待入队的长度wLen,则直接从当前写索引位置开始写入长度为wLen的数据;若len1小于待入队长度wLen,则先从当前写索引位置开始写入长度为len1的数据,再从队首位置开始

写入长度为(wLen-len1)的数据。⑤更新写索引。

改进出队流程如下(假设待取出的数据长度为rLen):
①首先判断队列当前有效数据长度是否足够。②若当前有效数据长度不足,则返回出队操作失败。③若当前有效数据长度足够,则计算当前读索引rIndex到队尾的长度len1。④若len1不小于待出队的长度rLen,则直接从当前读索引位置取出长度为rLen的数据;若len1小于待出队长度rLen,则先从当前写索引位置取出长度为len1的数据,再从队首位置取出长度为(rLen-len1)的数据。⑤更新读索引。

3 运行时间对比

为更好地评估运行所需时间,循环执行多次后比较运行时间。测试MCU为兆易创新GD32F330R8T6,主频为84 MHz,结果如表1和表2所列。

表1 与运算与取模运算时间对比
(测试循环30 000次)

参数	与运算	取模运算
耗时/ms	54	73

表2 改进队列与原始队列读写时间对比
(数据操作长度为1 000 000字节)

参数	改进后队列	原始队列
入队耗时/ms	253	825
出队耗时/ms	240	717

4 结 语

本文分析了传统队列的不足,设计了队列的大小规则,提出了与运算方式访问队列,并在此基础上提出了批量读/写多字节的方法。根据实测的运行时间对比可知,改进后的队列能有效提高队列的数据读/写效率,数据访问效率提升50%。一次性读写多字节时,效率提升超过200%。

参考文献

- [1] 严蔚敏. 数据结构(C语言版)[M]. 北京:清华大学出版社, 2007.
- [2] Kenneth A Reek. Pointers on C[M]. 徐波,译. 北京:人民邮电出版社,2017.
- [3] 李精华. 微机原理与单片机接口技术[M]. 北京:电子工业出版社,2018.
- [4] 张淑清. 嵌入式单片机STM32原理及应用[M]. 北京:机械工业出版社,2019.

通信作者:梁凤涛, lft900223@163.com。

(责任编辑:薛士然 收稿日期:2023-05-26)