Advances in Computer Science Research, volume 89
2019 International Conference on Wireless Communication, Network and Multimedia Engineering (WCNME 2019)

Research on Embedded Operating System Manage Function to Improve Memory Consumption Issues

Zijian Zhou
Xiamen University Malaysia, Sepang, Selangor, Malaysia

***Abstract*—This paper meet the needs of embedded real-time applications, two new dynamic memory management schemes for embedded systems are proposed. Finally, physics link list is added to make the system run more smoothly. The algorithm of this paper separates the application of large memory from the application of small memory. For small memory allocation, a fixed-size memory allocation method is adopted to reduce the generation of memory fragments, for large block memory allocation, using the actual size of the request to allocate memory to effectively improve memory utilization. This paper mainly realizes how to quickly realize memory allocation, release and recovery, and how to improve memory utilization, and puts forward a new idea and implementation.**

Keywords—memory; management; location; storage

I. Introduction

With the rapid development of computer science and technology and the enhancement of the trend of intelligent electronic products, embedded systems are becoming more and more widely used. In some important fields such as industrial control, medical care, and finance, the security and reliability of embedded systems are particularly important. The research of memory management technology is of great significance to ensure the security and reliability of data storage in embedded real-time systems. The basic task of this memory management is to enable the system to allocate and recycle dynamic memory quickly and efficiently, and to ensure its security and stability.

II. Background

Embedded system is widely used in the field of industrial control because of its low power consumption, high reliability, high cost performance, high efficiency, strong practicality, small in occu- pation and rich network function, not to mention its application value in military and aerospace fields. As a part of embedded system, it is crucial whether real-time embedded operation system can operate normally. At present, there are many companies in the world devoted to the development of embedded real-time systems, including industrial automation, network communication, medical equipment, nuclear monitoring equipment, and intelligent household appliances. The normal operation of the system under the condition of limited resources raises higher requirements for the operating system.

III. ISSUES

The memory resource of embedded system is relatively tight, so the main research direction of embedded real time operating system's memory management is to improve the real time of memory allocation and the rate of memory fragmentation and memory leak.

A general embedded system supports static allocation, and its program size can be determined when compiled and connected. However, only static allocation makes the system lose flexibility, and such an allocation will inevitably lead to great waste, because the memory allocation must be most configured in the worst case.

The mechanism of dynamic memory allocation is flexible, which brings great convenience to program implementation. In some applications, dynamic memory allocation is necessary. But the memory leak caused by it is also a big problem that puzzles programmers.

In embedded real-time systems, we use typical value memory management mechanism, TSFL is a common algorithm which shows below (see figure 1), obviously, this algorithm is very common, but there are many problems as we mentioned above.

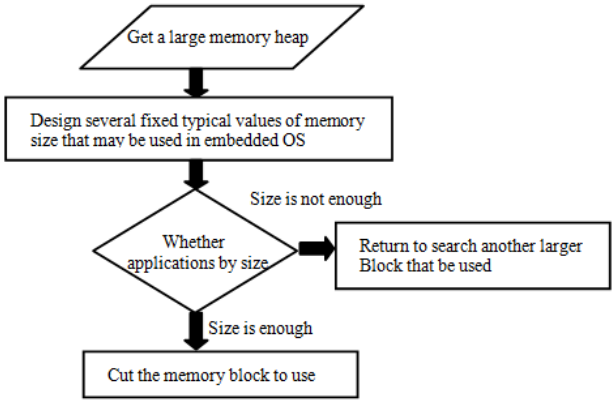

FIGURE I. FLOWCHART OF GENERAL ALGORITHM LIKE TSFL

IV. Recommendation

A. *Effective Management Algorithm of μC/OS II*

μC/OS II (Micro-Controller Operating System Two) is a preemptive, real-time multitasking kernel with high portability, especially suitable for microprocessors and controllers, and is suitable for many real-time operating systems (RTOS).

In C/OS-II, the operating system manages the continuous large memory blocks by partitioning, each partition contains an

integer number of memory blocks of the same size (see figure 2),In this way, the μC/OS II improves the malloc and free functions so that they can allocate and release fixed size memory blocks. By the same reason, the execution time of these kind of function is also fixed.

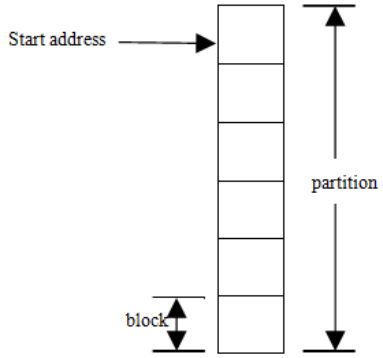

FIGURE II. MEMORY PARTITION OF SAME SIZE

There are multiple memory partitions in a system (see figure 3), so users can get different sizes of memory blocks from different memory partitions. However, when a specific memory block release when they are used, it must be relocated to the memory partition it used to belong to. Obviously, with such a memory management algorithm, memory fragmentation problems can be easily solved.

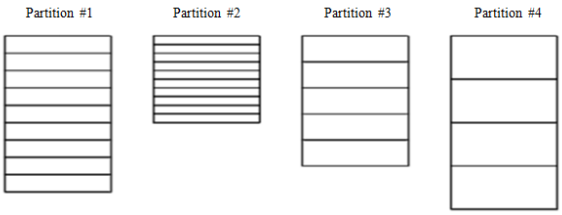

FIGURE III. DIFFERENT SIZE OF MEMORY PARTITION WAIT TO BE USED

B. *RTEMS Memory Management*

RTEMS (Real Time Executive Multiprocessor Systems) is a microkernel preemptive real-time embedded operating system. It is an open source real-time embedded operating system with high real-time performance, strong stability and transplantable portability. At present, it has a wide range of applications in both military and civilian applications.

The system's dynamic memory management algorithm is the first-fit algorithm. The basic idea is to define two linked lists, one is the free linked list used to manage free blocks in memory. Another is the distribution linked list used to manage the allocated memory blocks.(see figure 4)when an allocation request occurs, first search free list, until you find a free block that satisfies the memory request, at the same time, the two linked lists should be updated accordingly . The release process of memory is searched for the distribution linked list first, find the memory block to release, remove from the distribution list. At the same time, the free list need to be updated accordingly.

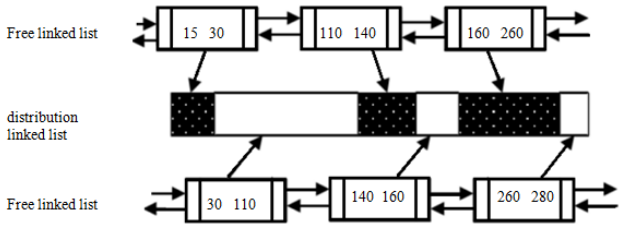

FIGURE IV. TWO KIND OF LINK LIST OF FIRST-FIT ALGORITHM ARE WORKING

C. *Physical Linked List-A New and More Efficient Memory Management Tool*

In the first-fit algorithm, when you allocate space, you only need to search one linked list, but you need to search two linked lists when releasing space. Therefore, the time and memory consumption on the two case is different. In this case, we introduce a new data structure called a physical linked list to reduce memory consumption. The idea of improving the algorithm is to manage allocated memory blocks and free memory blocks with a linked list. (See figure 5) In this way, only one linked list will be searched when the space is allocated or released.

It is used to connect all memory blocks (free blocks and occupied blocks) with adjacent physical locations. The introduction of physical linked lists can save a lot of time of releasing allocated memory blocks and merging adjacent memory blocks. Because It only needs to check the status of the adjacent two memory blocks then can determine whether need to merge or not.

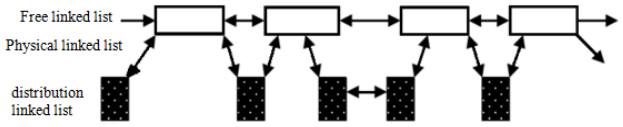

FIGURE V. PHYSICAL LINKED LIST BETWEEN FREE AND ALLOCATED LINKED LIST

Above all, we can provide a more effective memory management method and design a new flowchart for these algorithms (see figure 6).

V. CONCLUSION

Through the above introduction, we can know how the old and new algorithms execute. Compared with the old algorithm, I think the new algorithm is more in line with the development requirements of modern computers. The memory management mechanism must meet the following characteristics: real-time, efficient, reliable. It must be changeable in real time and efficient and reliable. Next we compared the old algorithms separately, based on the above three points.

Memory fragmentation inevitably occurs in memory allocation, but memory fragmentation can seriously affect memory usage efficiency and extend the time required to search and manage Lists, so we need appropriate algorithms to reduce memory fragmentation and improve Memory efficiency.

Compared with the old algorithm, the new management algorithm has a certain improvement in operational efficiency, memory fragmentation which improves the efficiency of memory usage, significantly reduced the memory waste.

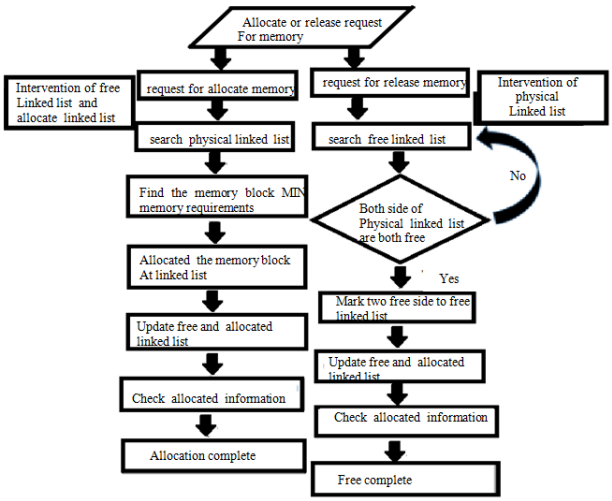

FIGURE VI. FLOWCHART FOR EFFECTIVE MEMORY MANAGEMENT

REFERENCES

[1] Tassadaq Hussain, Amna Haider, Adrian Cristal, Eduard Ayguadé, EMVS: Embedded Multi Vector-core System, Journal of Systems Architecture, Volume 87, June 2018, Pages 12-22.

[2] Yousaf Bin Zikria, Heejung Yu, Muhammad Khalil Afzal, Mubashir Husain Rehmani, Oliver Hahm, Internet of Things (IoT): Operating System, Applications and Protocols Design, and Validation Techniques, Future Generation Computer Systems, Volume 88, November 2018, Pages 699-706.

[3] Josefa Díaz Álvarez, José L. Risco-Martín, J. Manuel Colmenar, Evolutionary design of the memory subsystem, Applied Soft Computing, Volume 62, January 2018, Pages 1088-1101.

[4] Zhonghua Dai. A Fuzzing Test Method for Embedded Device Firmware Based on Taint Analysis, JOURNAL OF SICHUAN UNIVERSITY, Mar, 2016.Vol 48, No. 2, Pages 125-131.