

# Construction on Embedded Real-time Operating system of Computer

Huang Xiuqing<sup>1,a</sup>

<sup>1</sup>College of information science and engineering, Qilu Normal University,

Ji'nan, Shandong, 250000, China

<sup>a</sup>15053188785@139.com

**Keywords:** embedded, real-time system; TSLF algorithm, memory manager

**Abstract.** This paper constructs the overall structure of embedded OS with higher reliability, it can enable many system function modules to operate in the same CPU, and it can mutually integrate with each other in function and supports flexible configuration and arrangement of system. This paper puts forward new data structure by adding algorithm of secondary bitmap, it adds access protection mechanism for heap corruption, which completes one kind of new memory manager ERMM in embedded real-time system, the cutting strategy under memory allocation and it also puts forward improvement strategy of precise cutting. The test result indicates that ERMM dynamic memory allocation manager has better real-timeliness, which can effectively enhance memory performance of embedded real-time system.

## Introduction

With the gradual emergence of new technology, complication of real-time software has greatly increased, real-time system must support online updating while updating can unavoidably introducing into risk possibility of error, so that it makes it difficult for universal comprehensive test of system to guarantee reliability<sup>[1,2,3]</sup> of system. Meanwhile, equipment outside system tends to be intelligent equipment of opening, multi-function and controllable. According to dynamic custom-made of demand, expanding drive procedure, equipment safety and easy manageability of embedded real-time system become increasingly important. Due to characteristics of real-time and predictability on memory management in embedded real-time operating system, it usually adopts quick and confirmed memory management model, the usual public characteristics are as follows: it dose not use visual storage technology, static memory management mechanism and relatively simple memory protection model etc<sup>[4, 5]</sup>.

This paper overall structure of embedded OS with higher reliability, it can enable many systems to operate in the same CPU and mutually integrate in function, as well as support flexible configuration and arrangement of system. Through detailed analysis on TSLF algorithm, this paper puts forward one kind of new dynamic allocation manager embedded real-time.

## The overall structure of embedded real-time operating system with high reliability

Embedded OS with high reliability is based on ReWorks to realize, it mainly implements environment by establishing partition(it is similar to process, but it is operating system, includes task scheduling, memory management etc), which makes plenty of partitions operate in the same CPU at the same time and do not mutually disturb with each other. Embedded OS with high-reliability is mainly composed of 2 parts: the core operating system at the bottom and partition operating system. CoreOS constructs environment to implement environment, provides support for operation of partition operating system, including resources distribution of partition, partition loading, partition guidance, partition reconstruction, partition dispatch, time management, support false passage of partition communication etc. Partition OS realizes the basic partition and it also realizes Arim653 library in the partition, including process management, time management, process communication of partition (buffer, blackboard, Semaphore, Event), Port communication among partition(sampling and queuing).

By integrating CoreOS and Partition OS, this paper designs embedded OS with high-reliability according to standard of ARINC653, this system is mainly divided into 3 layers: module support layer: including software support package of CPU/Board, each kind of bus, bottom support of equipment, including network, pci bus, port, hard disc. System software layer, includes Core OS and Partition OS, they are the key point of this project, utility software layer: it is mainly focus on application development of Arinc653. The system frame diagram of high-reliability OS is indicated by diagram 1.

It supports sharing surrounding equipment of partition safety, supports PCI, USB, 1533, StarFabric bus etc, provides typical drive of equipment such as DOC, Flash, Ethernet, audio and video equipment, AD/DA etc, realizes relevant module support layer of hardware.

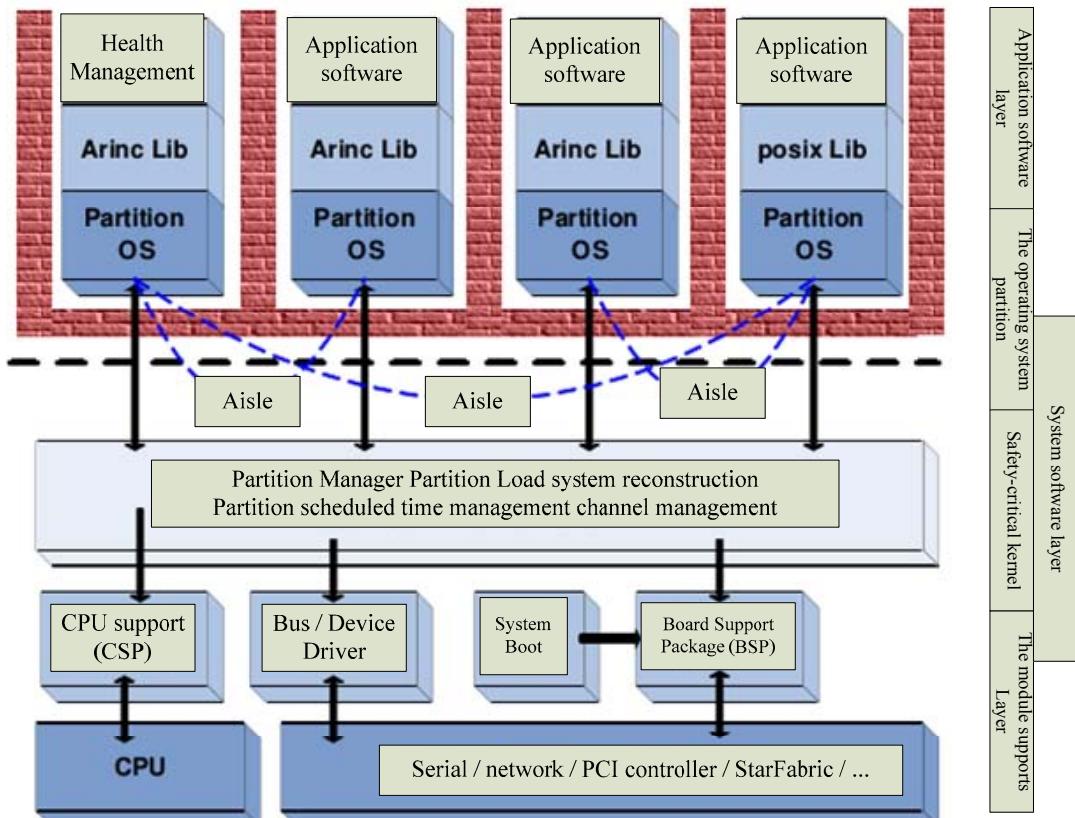


Diagram 1 System frame of high-reliability embedded OS

### Time domain protection technology of high-reliability embedded OS

Many partitions share with CPU and they have no influence among each other, in Arinc653 it defines concept of 2-grade scheduling, which means each partition uses overall scheduling based on time window to carry out partition on the bottom core of each partition, it uses task scheduling method of superior grade in partition, which guarantees certainty of time and space in partition,

Window ID	1.1	4.1	2.1	3.1	4.2	1.2	4.3	2.2	3.2	4.4	5.1
Partition	P1	P4	P2	P3	P4	P1	P4	P3	P2	P4	P5
offset	0.000	0.020	0.030	0.040	0.070	0.100	0.120	0.130	0.160	0.170	0.180
duration	0.020	0.010	0.010	0.030	0.010	0.020	0.010	0.030	0.010	0.010	0.020

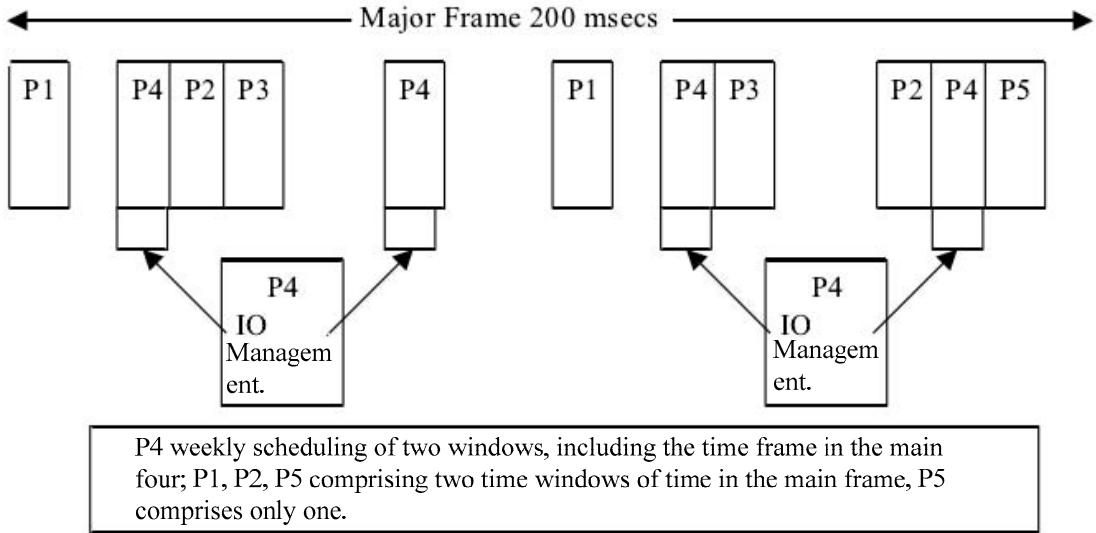


Diagram 2 Time scheduling of partition

meanwhile, it solves coincidence and certainty of time among partition. In order to activate rolling, core module OS and fixed cycle to maintain one major time frame, and realizes timing and repeated operation of this module. The major time frame distributes one or many partition windows for each partition, and it also confirms deviation and pre-set operating time of partition window in the major time frame.

Such as the time scheduling diagram of partition time indicated by diagram 2, of which, P1 is healthy monitoring partition, P2 is data output partition, P3 is data process partition, P4 is IO partition, P 5 is data output partition.

Because partition is not to completely occupy CPU, it needs to do some special treatment on partition time, it is mainly time synchronization. Its concept is that all the partition time in one module keeps synchronization and keeps accordance with core operating system at the bottom. On realization method, firstly partition time is obtained from core operating system at the bottom, secondly is that it needs to make time supplementary when other partitions are transferred to original partitions after implementation on other partitions, it also needs to do corresponding treatment.

#### Design on memory allocation manager

TLSF is one kind of 2-grade separated-adaptation algorithm, using way of combining with Bitmaps and segregated list to manage [6, 7] memory pool. TLSF algorithm has better performance in aspect of memory failed rate and real-time, but it has some shortages in safety as well as memory allocation and release. As for shortages of TLSF algorithm, this paper puts forward one kind of new dynamic memory allocation manager ERMM.

FL1, SL1 and MBS is 3 major parameters<sup>[8,9]</sup> of this algorithm, TLSF uses way of combining with bitmap and segregated list to manage memory pool, the idle memory block uses SL1 to make organization. TLSF defines 2 index values: REAL\_FLI and MAX\_SLI, FL1 cuts idle memory field into memory range of REAL\_FLI, memory range size corresponds to index value i is  $[2^i, 2^{i+1})$ , on this basis, SL1 cuts memory range  $[2^i, 2^{i+1})$  divided by FL1 into MAX\_FLI index values. When n=27, MAX\_SLI(number of SL1)=8, data structure of TLSF is indicated by diagram 3. Of which, fl is FL1 value, sl is SLL1. FL\_Bitmap and SL\_Bitmap describe index list.

Model expected output is indicated by  $y_1(k)$ : the real output of model of using  $y(k)$ , q indicates the number of network training data. Make partial derivative for connection weight  $\alpha_i^j$ , in the formula,  $\beta$  indicates learning rate.

$$E = \sum_{k=1}^q [y_1(k) - y(k)]/2 \quad (1)$$

$$a_i^n(k+1) = a_i^n(k) - \beta \frac{\partial E}{\partial a_i^n} = a_i^n(k) + \beta[y_1(k) - y(k)]\bar{u}_t y_t \quad (2)$$

ERMM divides 2 memory partitions MP1 and MP2 in the system, MP1 is the small memory allocation range, MP2 is big memory allocation range, each memory allocation range has its own memory control module, and it records corresponding information of memory partition. MP1 partition is divided into memory block with the same size and response to allocation request of small memory. Memory space of MP2 makes division according to data structure of TSLF algorithm after improvement, and makes link by SL1 and responses to allocation request of big memory. The overall time complication of ERMM is O(1). The data structure after improvement organizes idle memory block by SL1, FL1 cuts memory partition MP2 into 24 ranges, size of the 0 range is 24\*[0,28-1), the first range is 23\*[28-1,28+1), other partition range is 22\*[2start,2start+1), 21\*[2start+1,2start+2)...[2end-1,2end). The SL1 makes linear division on the basis of FL1; the cut number reduces one by one. When start=9、end=30, partition state of MP2 is indicated by diagram 4.

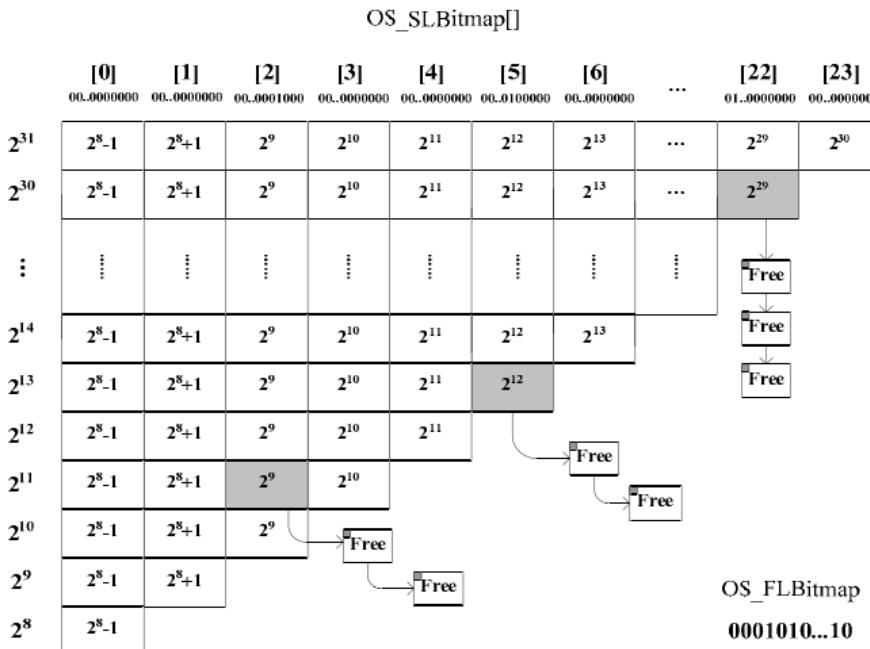


Diagram 4 TSLF data structure diagram after improvement

According to the memory block obtained from index value of FL1 and SL1, clear position of bitmap corresponds to idle memory block. The difference between maximum of allocating memory and maximum of requesting memory is the memory failed rate<sup>[10]</sup> of current system, it is indicated by formula (3):

$$F = \frac{M_{\max} - M_{\text{real}}}{M_{\max}} \quad (3)$$

In the formula, F is failed rate,  $M_{\text{real}}$  is maximum of actual memory allocation,  $M_{\max}$  is maximum of memory allocation under ideal conditions. This formula is applicable to calculate external fragment, memory fragment is the decimal bigger or equal to 0 and less than 1. Define and integrate threshold L, it is determined by experience. When F is less than L, it does not make integration, when F is bigger than L, check memory block waiting for release whether it has adjacent idle memory block, if it has them immediately make integration.

Check SL1 and FL1, insert the released memory block into list head of corresponding lost by this index, and forms one complete 2-way list and corresponding positions complete release of dynamic memory.

## Realization of ERMM in High-reliability and Embedded OS System

The test divides memory of user request into 8 ranges(unit it byte), they are respectively [0,128], [128,256], [256,1024], [1024,4096], [4096,16384], [16384,65536], [65536,262144] and [262144, 1048576], the test program is randomly generated in each range and it will frequently allocate and release memory with random value. Each group tests for 100 times and it takes the average value. It uses the timing function in OS system, measures memory dispatch and release time of memory, compares with real-time function of ERMM and TSLF algorithm.

This paper transplantsμC/OS-III system to X86 processor under VC environment, using development environment of VC++ 6.0 to carry out system operating and debug, test environment is : ) memory: 2G, embedded real-time operating system: OS system, TSLF algorithm: TSLF2.4.6 version, debug environment: development environment of VC++ 6.0.

Allocation of dynamic memory and test result of release time are indicated by table 1, diagram 5 and 6 indicate test result of table 1 by way of bar graph.

Table 1 Comparison table between allocation and release time (us) by way of bar graph.

Operation	Algorithm	Range 1	Range 2	Range 3	Range 4
memory	TLSF	5.856	6.994	9.354	8.454
allocation	ERMM	1.036	5.882	7.256	7.921
Memory	TLSF	4.365	4.886	7.652	7.112
release	ERMM	0.986	3.127	4.365	4.653
Operating	algorithm	Range 5	Range 6	Range 7	Range 8
Memory	TLSF	10.237	12.365	11.564	14.153
allocation	ERMM	8.945	11.323	11.238	14.862
Memory	LSF	9.002	10.386	10.986	12.364
release	ERMM	6.354	10.167	11.237	12.743

From diagram 5 and 6, it can be seen that memory allocation and release time of ERMM are less than TSLF in the overall, it has better real-time. Small memory allocation and release speed of ERMM are quicker than TSLF, this is because small memory allocation and release adopt 2-grade bitmap algorithm to allocate and release memory block of fixed size, and the command cycle is less than TSLF. Memory allocation time of ERMM is always less than TSLF, but memory release time becomes smaller with increase of allocating memory and TSLF difference, even it exceeds TSLF in memory release time of the 7<sup>th</sup> and 8<sup>th</sup> group.

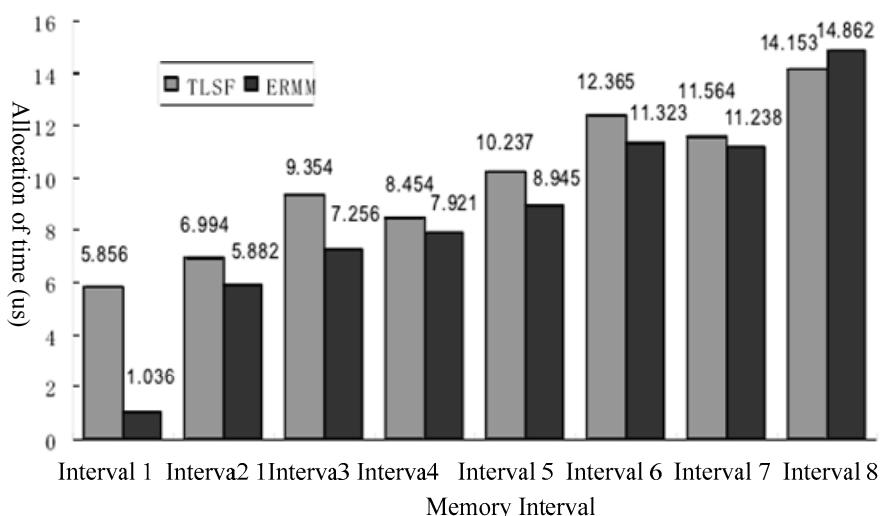


Diagram 5 Comparison diagram of memory allocation time

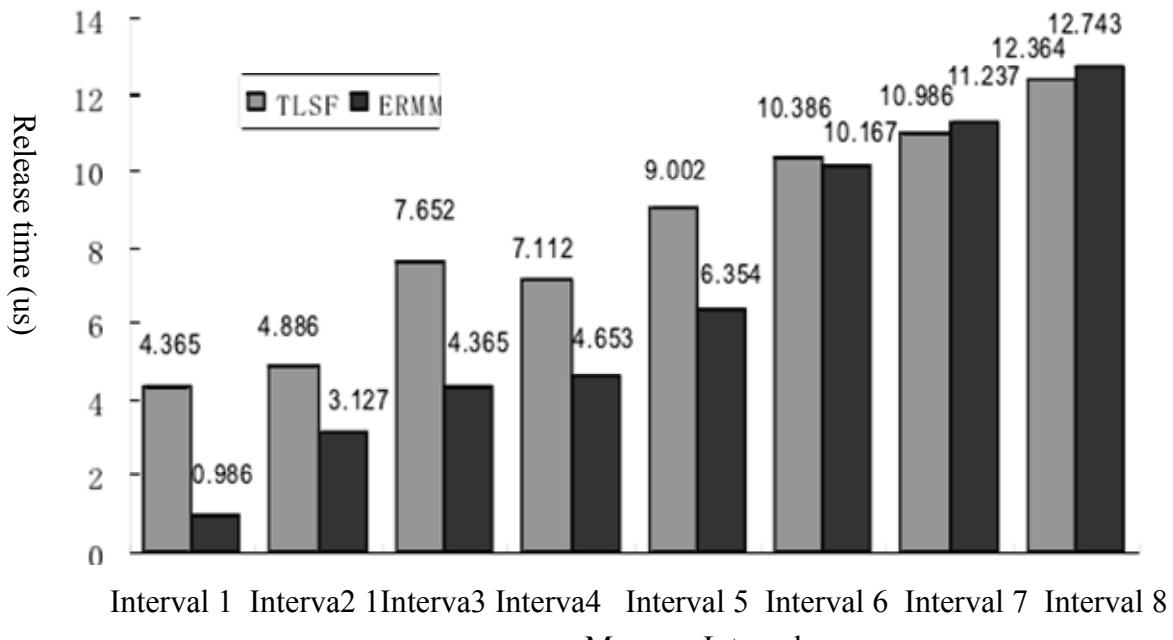


Diagram6 Comparison diagram of memory release time

## Summary

Real-time operating system must provide base guarantee for high reliability of embedded system, support reliable updating of system, fault separation, fault avoidance and fault-tolerant, guarantee timeliness of real-time software and reliable operating, embedded real-time technology with high reliability has become to be the development trend for real-time operating system in the future. The innovation of this paper lies in that it realizes high-reliability system by viral partition, it can enable many system function modules to operate in the same CPU, and mutually integrate with each other in function, and it also supports flexible configuration and arrangement of system. Memory allocation with different memory adopts different allocation strategy, using strategies of adding boundary marking technology, memory allocation adopts precise cut and memory release adopts delay integration. It enhances memory allocation efficiency, reduces memory failed rate as well as memory allocation and release time by these improvements, increases safety and reliability of memory allocation.

## References

- [1] Dong Enqiang .Research on High-reliability Technology under Real-time Environment. Computer Engineering and Science,2006,p71-82.
- [2] You Zhiyu. Research on Vision Control System Based on Embedded Machine. Master Thesis of Xihua University.2009,p59-66.
- [3] Xu Guanghui, Huang Ru. Embedded Development and Application Based on FPGA. Beijing: Publishing House of Electronics Industry,2006.,p184-192.
- [4] Ge Baojian. Research and Realization on Coordinated Design Platform Based on SOPC Software. Master Thesis of Wuhan University Of Technology2008.,p51-62.
- [5]Zhang Fei. Research on Dynamic Memory Management of Real-time Embedded Operating System. Master Thesis of University of Science and Technology of China, 2011,p93-102.
- [6] Lu Xiaohuang, Shuai Jianmei. One Kind of New Memory Manager towards Object Program. Computer Engineering, 2012, 38(9):21-23.

- [7] Yan Guangming. Research and Realization on Separation Technology on Memory Space Field of Embedded System. Harbin Engineering University, 2011, Vol 48(8),p 1414-1420.
- [8] Li Falong, Wu Gang, Chen Zhanglong. Application of Bitmap in Memory Management of Embedded System. Computer Engineering and Design, 2005, 26(4), p1020-1023.
- [9] Yan Mengtian, Ding Zhigang etc. Performance Test on Memory Allocation in Real-time Operating System. Computer Application, 2007, 27(11),p 2838-2840.
- [10] Qian Weiming, Zhang Zhenlv. Analysis and Research on Memory Management Mechanism of VxWorks. Micro-computer Information, 2009,25(12),p115-117.