Enhancing Control Loop Periodicity for Low Computing Power UAV

Nguyen Pham Dang Khoa', Doo-Hyun Kim[2]

[1]EB Inc, Ho Chi Minh City, Vietnam
[2]Department of Multimedia Engineering, Konkuk University, Seoul, Korea
[1]khoanpd@hotmail.com, [2]doohyun@konkuk.ac.kr

Abstract. Due to the limitation of the computing power, a lightweight RTOS with sophisticated scheduling is crucial for the development of more advanced software for small-scale UAVs. This paper presents a coupling of EDF (Earliest Deadline First) and SRP (Stack Resource Protocol) algorithms to enhance a lightweight RTOS, i.e., µC/OS-II for the control software, called OFP (Operational Flight Program), for a small-scale UAV. Experimental results performed over HILS(Hardware-In-the-Loop Simulation) with previously verified OFP show this approach is effective to support higher periodicity and lower jitter.

Keywords: Small-scale UAV, OFP, RTOS, Periodicity, EDF scheduler, Stack Resource Protocol

1. MOTIVATION and APPROACH

The µC/OS-II(Micrum, 2010; Labrosse, 1998) is known as a appropriate OS for a control system, providing steady resource management system and many APIs for design of real time systems, and is effective for limited hardware resource. However, the current scheduling policy available in µC/OS-II does not support dynamic priority scheduling, but supports only the fixed-priority scheduling, in which the highest priority task will be chosen to run first. It is known in the research field that the EDF scheduler (Buttazzo, 2005; Buttazzo, 1997; Jensen et al., 2003), which is one of traditional dynamic scheduling algorithm(Liu, 1973), is expected to be more efficient in managing resources for multiple periodic tasks, and to be more effective in guaranteeing the timeliness for each task and keeping the jitter as small as possible. This paper proposes the EDF-SRP, i.e., the usage of EDF coupled with SRP (Stack Resource Policy)(Baker et al., 1991) as a resource access protocol to avoid deadlock among tasks during accessing competing resources. The given OS features, such as task switching are fully utilized. EDF and SRP are implemented as plug-ins built on the top of the µC/OS-II API. All tasks are invoked using the plug-in's API, and then the plug-in itself uses the µC/OS-II API.

The priority number assigned to each task in µC/OS-II is used to make the Preemptive EDF. The first task in the EDF ready queue is selected as a candidate and set to ready in the µC/OS-II ready list. At the time when the µC/OS-II scheduler is running, if its priority number is the highest one set in the µC/OS-II ready list, it will preempt the current running task. On the other hand, the Preemptive EDF-SRP scheduler requires a resource access protocol to solve the problems, such as Priority Inversion and Deadlock, when tasks access shared resources. The µC/OS-II supports the mutual exclusion by mutex. The mutex in µC/OS-II is a special binary semaphore integrated with a protocol almost the as same as the Priority Inheritance Protocol. This can omit the Priority Inversion problem, but it cannot prevent the Deadlock phenomenon. Therefore, the SRP, which can be applied to a dynamic scheduling algorithm like EDF, has been applied instead, as it is a better protocol (Sha et al., 1991; Baker et al., 1991).

As shown in Fig. 1., the schedule decision is placed in the Timer ISR (Interrupt Service Routine), instead of creating one more task (Balbastr et al., 2001). The schedule decision is performed to select the next task following the EDF-SRP schedule policy stated above, post the event flag to that task to read it in the µC/OS-II ready list, extract it from the EDF ready queue, update itself for the next active time, next absolute deadline, and re-insert it into the EDF queue in the correct position, for each OS tick generated. The task can preempt the running task when its preemption level is higher than the system ceiling and its priority number is highest in the 1.1C/OS-II ready list.

2. EXPERIMENTS and RESULTS

The motivation of our experiment is to show the efficiency of the EDF-SRP implementation compared to the original scheduling of the µC/OS-II. The results will be evaluated using two criteria in this experiment. First, we have tested the periodicity of each task, to prove that the EDF scheduler can guarantee the timeliness for each task. Second, a high delay jitter of start and end time in many control applications can cause instability or a jerky behavior of the controlled system; hence, it must be kept as low as possible. Thus, jitter is evaluated. Response jitter means the gap between the ideal and actual response time of a task. These jitters affect the periodicity accuracy.

The OFP(Operational Flight Program) scheduler (Peng, 2010; Nodir, 2010; Cai, 2005), was tested to see the advantage of using EDF compared to the Fixed-Priority, the original scheduling of the µC/OS-II in this experiment, as a practical autonomous control application for a small-scale UAV. The start time and end time of every execution of the Main Control Loop (MCL) were logged. In all periodicity evaluation tests, the X axis represents the number of task executions, while the Y axis represents for the period at each point of execution. In all jitter evaluation tests, the X axis also represents the number of task executions, while the Y axis represents for jitter value corresponding to each point of the task.

As depicted in Figure 2, the average execution time of MCL task is 26.743 OS ticks (2.6743 milliseconds). Sometimes, however, the execution time exceeds 12 milliseconds, exactly 13 times. It is assumed that the MCL task has complex and long execution logic, thus resources are easy to be preempted by a higher-priority task or ISR by an external signal interrupting the task. Thus, the periodicity of that task is so unstable. As shown in Figure 2 (b) and Figure 2 (c), although the average period is near 20 milliseconds, entire period values drift greatly, even some periods are over 30 milliseconds. Consequently, accumulations of jitters also frequently occur to cause instability of synchronization with simulation data that are transferred every 10 milliseconds.

Conversely, the results in Figure 3 show better performance of periodicity than in Figure 2. Even though a few executions violate the start time of execution, these faults are soon adjusted. It is assumed that the early instability is due to the need of sufficient time to calculate accurate scheduling. The average execution time is 29.363 ticks, i.e., 2.9363 milliseconds. While the experiment on pure OS shows unsteady variation of execution times, most task operations on the EDF-SRP scheduler are finished within 30 milliseconds. It never exceeded 70 milliseconds, which is defined as the task deadline. According to Figure 3(b) and Figure 3(c), the average periodicity and jitter are much steadier than for pure RC/OS-II, shown in Figure 2. The response time is almost 20 milliseconds and the maximum value is about 30 milliseconds, but it occurred only once. Most response time period values are distributed from 15 milliseconds to 25 milliseconds. Even though there are some inaccuracies around early operations, those are compensated within the following early executions, thus, the error of periodicity is completely restored. This trimming is due to the strong management by the preemptive EDF-SRP algorithm.

3. Conclusions

Real time computing environment is essential to meet sufficient control performance of autonomous system. Unfortunately, designing and implementing a real time system is not easy; sophisticated task scheduling issues exist. These issues become serious in small-scale embedded systems that do not have plentiful resources. We ported a lightweight RTOS, i.e., µC/OS-II, on a small embedded computer to overcome those resource management problems. It consisted of a low-performance microprocessor and small memory, and then improved the kernel with EDF and SRP with an efficient data structure to manage ready tasks. We applied the OFP previously developed for a single rotary-wing aircraft[15] as the target application Thus, the improved µC/OS-II could demonstrate better performance than the original system in guaranteeing periodicity. The conceptual change of task design in which the designer can define the deadline of each task made the real time application design easy.

The enhancement was shown in experimental evaluations based on the HILS environment. The deadline of the main control loop task was designed as 7 milliseconds for 3,000 testing executions. The task response time was properly triggered every 20 msec cycle, and jitter was also minimized so that the variation of execution times was very steady and never violated the deadline. Consequently, it is experimentally proved that the EDF-based task scheduling control and SRP-based efficient resource management can guarantee strict real time requirements, such as task periodicity within a limited hardware environment.

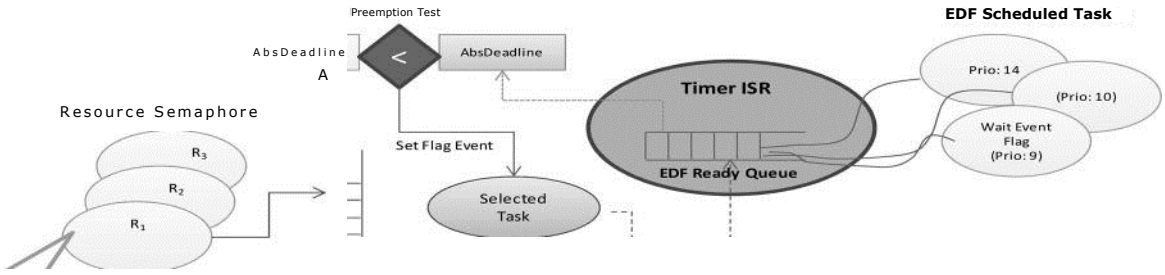
Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(grant number: 2012006817).

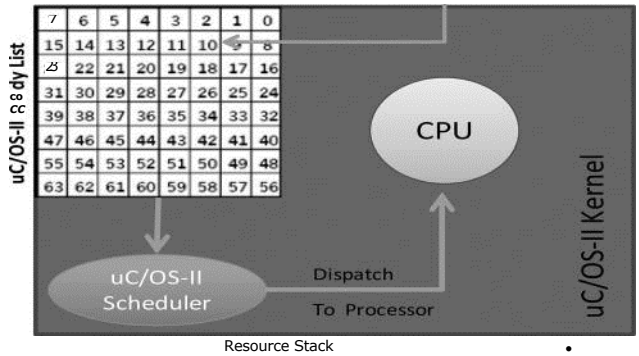
References

1. Micrium, RTOS and tools, http://micrium.com/ **(2010)**
2. J. Labrosse, MicroC/OS-II. R & D Books **(1998)**
3. G. C. Buttazzo, Rate monotonic vs. edf: Judgment day, *Real-Time Systems*, 29(1):5-26 **(2005)**
4. G. C. Buttazzo, Hard Real-Time Computing Systems, Kluwer Academic Publishers **(1997)**
5. L. Sha, R. Rajkumar, J. P. Lehoczky, Priority inheritance protocols: An approach to real-time synchronization, IEEE Trans. Computer, 39(9):1175-1185 **(1990)**
6. T. P. Baker, Stack-based scheduling for realtime processes, *Real-Time Syst.*, 3(1):67-99 **(1991)**
7. G. Cai, K. Peng, B. M. Chen, T. H. Lee, Design and assembling of a UAV helicopter system, in Proceedings of the 5th International Conference on Control and Automation, Budapest, Hungary, pp. 697-702 **(2005)**
8. K. Peng, G. Cai, B. M. Chen, M. Dong, K. Y. Lum, T. H. Lee, Design and implementation of an autonomous flight control law for a UAV helicopter, Automatica, 45(10):2333-2338 **(2009)**
9. S.G. Kim, S.H. Song, C.H. Chang, D.H. Kim, S. Heu, J.G. Kim, Design and Implementation of an Operational Flight Program for an Unmanned Helicopter FCC Based on the TMO Scheme, in Proceedings of the 7th IFIP WG 10., SEUS'09 **(2009)**
10. P.G. Jansen, S.J. Mullender, P.J.M. Havinga, H. Scholten, Lightweight EDF Scheduling with Deadline Inheritance, University of Twente **(2003)**
11. P. Balbastr, Ismael Ripoll, Integrated Dynamic Priority, Scheduler for RTLinux, in Third real-time linux workshop **(2001)**
12. N. Kodirov, D.H. Kim, J.Y. Kim, C.J. Moon, Experiment Effectiveness Analysis of EDF-eCos for Real-Time computing in Small Unmanned Helicopter OFP, in International Conference on Intelligent Unmanned Systems, (ICIUS 2010), Bali, Indonesia **(2010)**
13. C.L. Liu, J. Layland, Scheduling algorithms for multiprogramming in a hard real-time environment, *Journal of the ACM* 20(1): 46-61 **(1973)**
14. S.P. Kim, A. Budiyono, J.H. Lee, D.H. Kim, K.J. Yoon, Control system design and testing for a small-scale autonomous helicopter, Aircraft Engineering and Aerospace Technology: An Int'l Journal 82(6):353-359

(2002)



– Locking Task ID
– Preemption Ceiling (the Absolute Deadline of task at time it locking resource)



Resource Stack

Become ready in uCos-II ready list

- Update EDF Pa ra ms
- Extract from EDF queue
- Re-insert to proper position in EDF queue

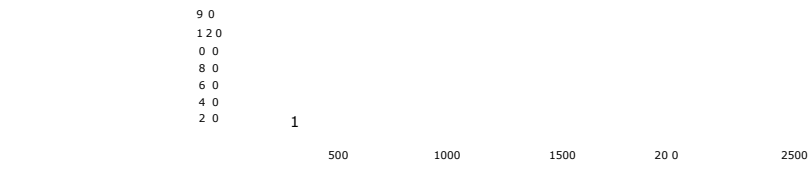
36 35 34 33 32
44 43 42 41 40
52 51 50 49 48
60 59 58 57 56

uC/OS-I I Scheduler

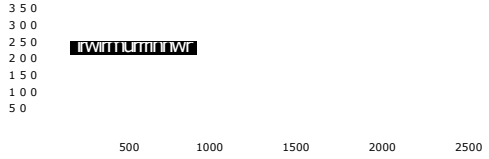
Dispatch

I o Processor

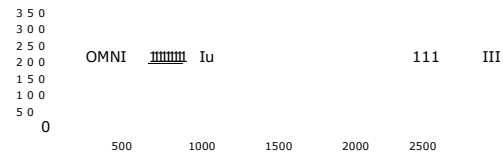
Fig. 1. Preemptive EDF-SRP scheduler



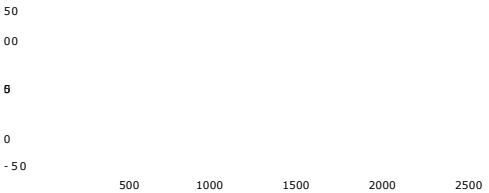
(a) Execution Time (average : 26.743 icks)

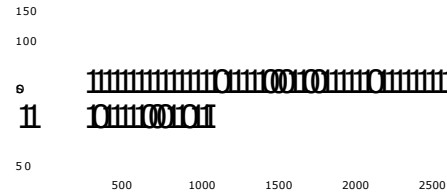


(b) response time period (average : 201.851 ticks)



(c) end time period(average : 201.929 ticks)





(d) response time jitter (e) end time jitter accumulation

Fig. 2. Periods and jitters of OFP executions on pure tC/OS-II

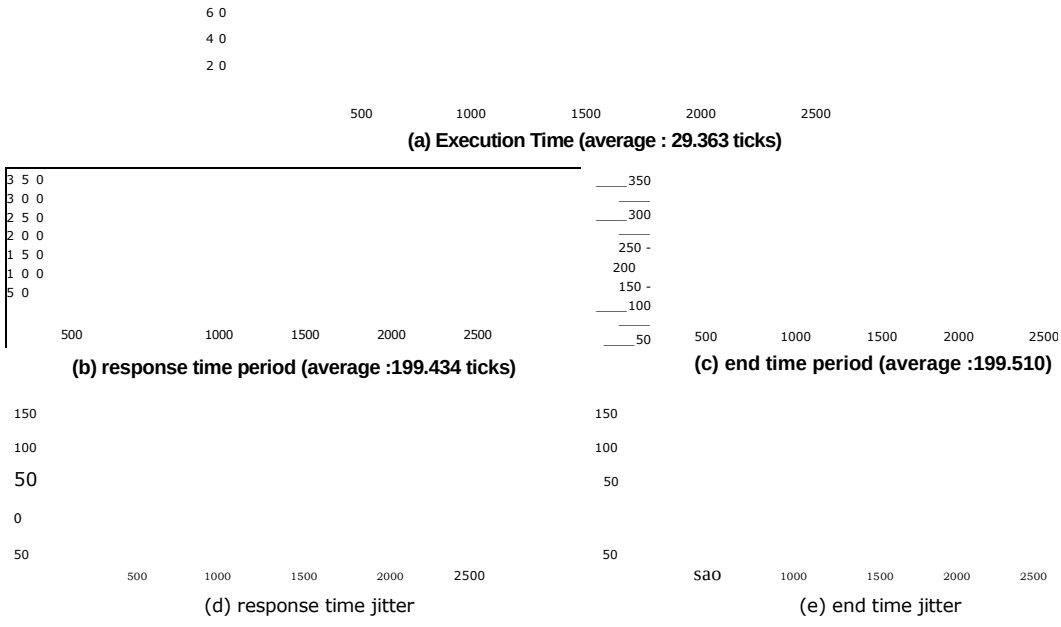


Fig. 3. Periods and jitters of OFP executions on EDF-SRP