Introduction of Deterministic OS for SPLC in Advanced Nuclear I&C System

C. W. Son*, D. H. Kim, G. S. Son
Korea Atomic Energy Research Institute
989-111 Daedeok-daero, Yuseong-gu, Daejeon, 305-353, Republic of KOREA
**Corresponding author: scw@kaeri.re.kr*

1. Introduction

The real-time operation system of the priority driven focusing on the function of managing the limited resource effectively was used for the operation system of PLC used for NPPs safety system.[1] and It, however, has disadvantage in terms that it generates the deadlock and the starvation state during the competing process of the resource preemptive since the OS like this controls the software module by manipulating the operating intervals.

Also, existing PLCs either had no output logic with regard to devices' redundant structure or it was set in a fixed way, and as a result it was extremely inefficient to use them for redundant systems such as that of a nuclear power plant and their use was limited.[2][4]

This project has developed a real-time operating system with Redundancy handling logic in order to improve the problems mentioned.

In Chapter 2 of this paper describes the structure and key features being developed SpCOS. That is the scheduling method, redundancy handling function of supervisory. Finally, chapter 3 concludes.

2. SpCOS

2.1 Structure of SpCOS

The SpCOS consists of SRK(Safety Real-Time Kernel), IHS(Interrupt Handling System) and system tasks. SpCOS is an interface between user applications and hardware modules. Figure 1. shows structure of SpCOS

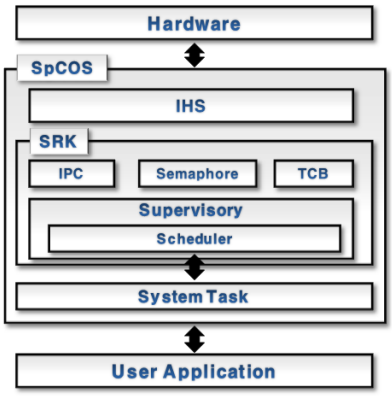

Figure 1. Structure of SpCOS

2.2 Scheduling Method

SpCOS use sequential non-preemptive scheduling method to solved a uncertainty of priority driven scheduling method. Each software module, uses the resources of a fixed amount of time every cycle and will run continuously. It also schedules the software module so that the value of the software module that is processed previously can be connected to the input of the next processed software module.[2] Figure 2. shows scheduling method of SpCOS

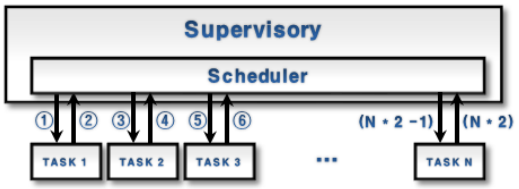

Figure 2. Scheduling Method of SpCOS

The following (Figure 3.) is the example of scheduling each software module in the way of scheduling through Figure 2.

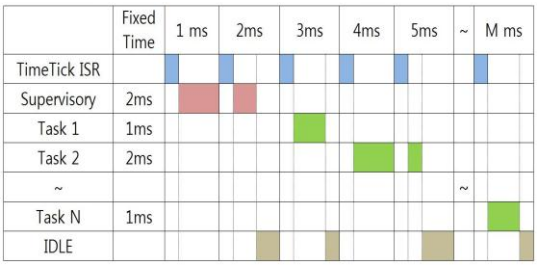

Figure 3. Scheduling Flow Chart

Here is the T of software module using the resource below 1ms in Figure 3.

- T = TimeTickISR + TaskFunction + IDLE

and, The next is about the T of software that excesses 1ms.

- T = TimeTickISR * FixedTime(ms) + TaskFunction + IDLE

Each software module holds time in the status of IDLE till the next Time Tick Interrupt is generated after processing the present function. IDLE time is changeable depending on the delay time.

2.3 Redundancy Handling

SpCOS supports the composition of I/O and the processor module from single to triple at most in order to be applied efficiently while changing the redundancy structure of NPPs safety system. Supervisory schedules the appropriate software module as well as confirms the redundancy information of the hardware module.

3. Conclusions

A controller used in Nuclear Power Plants safety system has been developed differently from industrial PLC by its design, production and test.[1] Therefore, the operation system that controls the NPPs safety PLC should be developed properly so that it can work well on the handling logic of the safety system.[3]

SpCOS has secured the deterministic for the safety PLC control by possessing the sequential scheduling, time management, redundancy handling logic and system diagnosis function considering the connectivity of all the software module by Supervisory.

REFERENCES

[1] G. S. Son, D. H. Kim and C. W. Son, Development of the High Reliable Safety PLC for the Nuclear Power Plants, The Transactions of the Korean Institute of Electrical Engineers, Vol. 62, No. 1, pp.109~119, 2013
[2] C. W. Son, D. H. Kim and G. S. Son, Design of Deterministic OS for SPLC, Transaction of Korea Nuclear Society Autumn Meeting, Vol.2, p.999, 2012
[3] D. H. Yun, “Detailed Design Report for SPLC (ANICS-SPLC-RR101)”, PoscoICT, 2012
[4] JEAN J. LABROSSE, “MicroC/OS-II The Real Time Kernel”, 2002

ACKNOWLEDGMENT
This work was supported by the Nuclear Technology Development Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Knowledge Economy
(No. 2010161010001G)