Transactions of the Korean Nuclear Society Spring Meeting
Chuncheon, Korea, May 25-26 2006

Development of Real Time Operating System for Safety Grade PLC (POSAFE-Q) for Nuclear Power Plants

Han Seong Son,[a] Sung Jae Hwang,[b] Young Joon Lee,[c] Chang Hwoi Kim,[c] Dong Young Lee[c]
a ENESYS, 337-2 Jangdae-dong, Yuseong-gu, Daejeon, Korea, 305-308, hsson@enesys.co.kr
b POSCON, Korea Techno Complex, Anam-dong, Seongbuk-gu, Seoul, Korea, 136-713, trustsky@poscon.co.kr
c Korea Atomic Energy Research Institute, 150 Deokjin-dong, Yuseong-gu, Daejeon, Korea, 305-353

1. Introduction

POSAFE-Q is a newly developed programmable logic controller (PLC) in order to apply to digital safety system of nuclear power plants (NPP) according to Nuclear Power Plant safety requirements. POSAFE-Q hardware and software development process, including design, review, verification and validation, and configuration control and quality assurance, satisfies the requirements imposed by 10CFR50, Appendix B.

This article introduces a real time operating system pCOS, which is the core of POSAFE-Q. Section 2 describes the structure of pCOS. Section 3 describes a few important features of pCOS, which are necessary to the application for the digital safety system of NPP.

2. pCOS Structure

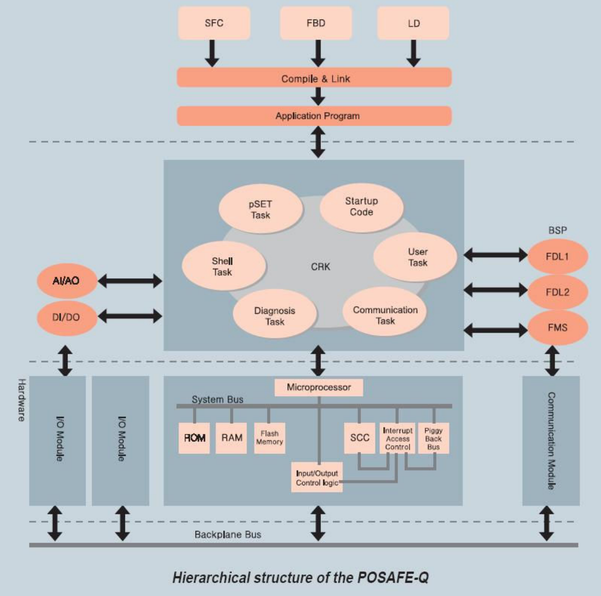

Figure 1. Conceptual Structure of POSAFE-Q

Figure 1 shows the overall conceptual structure of POSAFE-Q. As shown in Figure 1, the operating system pCOS performs two kinds of works. One is the execution of application programs. POSAFE-Q is connected by RS232C with pSET, which is the engineering tool for POSAFE-Q. Application programs created by pSET are downloaded to safety PLC and are executed by the operating system of the processor module, pCOS.

The other is the interface with various kinds of hardware and board support package (BSP) software. The RTOS has a direct interface to certain components of the target-system hardware, either by direct access or by functions provided by pCOS. This includes the access to LEDs on the processor module's front-plate, flash memory programming services, and watchdog services.

pCOS provides the interface of the system task software to an external service unit and its BSP. The communication BSP software interface provides a unified interface for sending and receiving messages via communication channels.

For certain types of input/output module, an I/O driver BSP is provided, which serves for transferring input/output signals between pCOS and the specific I/O module. For communication modules, BSPs are also provided so that application programs may indirectly access the dual-port memory on the communication modules through the communication tasks.

Figure 1 also shows the composition of pCOS. pCOS is composed of CRK (Kernel) and Startup code, Shell task, Diagnosis task, pSET tasks (consist of LoaderRxrdy task and Loader_service task), Communication tasks, and User_task task which executes application programs. pCOS deals with inputs from, or outputs to pSET through LoeaderRxrdy task and Loader_service task. In addition, pCOS communicates with hardware according to action of kernel, system tasks and application tasks.

3. Features of pCOS

pCOS, the POSAFE-Q operating system, is capable of data acquisition from input module, executing application program in successive loop during run mode, data transmission to output module, task scheduling, inter task communication, interrupting, online diagnosis, preventing deadlock and livelock, operating communication tasks, uploading and downloading of application programs, and interfacing with pSET.

pCOS supports 8 application program tasks whose capacity is 1MDW and the user can fix one task capacity within 256KDW flexibly. The kernel CRK provides a scheduling function for both built-in system task and application program task downloaded from pSET. It can also support the features needed to apply to the safety-related software systems such as memory protection, priority inversion prevention, diagnosis function.

For the POSAFE-Q processing the input scanning and application program in multi-tasking, the operating system and/or engineering tools can assure that the input scanning and application program are processed deterministically and completely. If the run of deterministic application program would not be proved transparently, the interrupt by a run of non-deterministic application program is not permitted. In other words, a simple and straight forward task organization is used in an operation mode, as shown in Figure 2. It is assumed that there is only one application task (application task 3). The Diagnosis task has the highest predefined priority. The LoaderRxRdy task and Loader_Service task has the second and the third highest priorities, respectively. The application tasks have the priorities that are lower than that of the Loader_Service task and higher than that of the Shell task. The Shell task has the higher priority than the statistics task has. The application tasks are operated when the 'RUN' command is issued by pSET and then the Loader_Service task processes it.

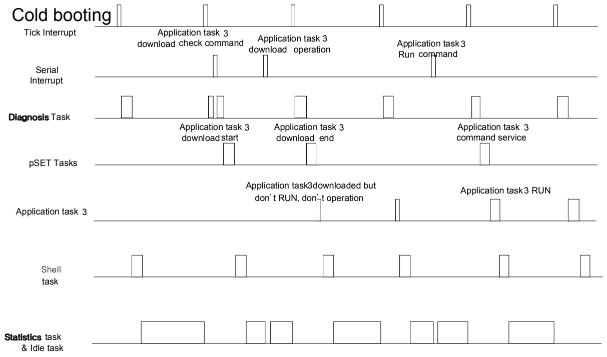
Cold booting
Tick Interrupt
Application task 3 download check command
Application task 3 download operation
Application task 3 Run command
Serial Interrupt
Diagnosis Task
Application task 3 download start
Application task 3 download end
Application task 3 command service
pSET Tasks
Application task3downloaded but don't RUN, don't operation
Application task3 RUN
Application task 3
Shell task
Statistics task & Idle task

Figure 2. Task Organization in a Mode of pCOS

After initialization, control is permanently passed to the Diagnosis task. The Diagnosis task controls the processing of control commands and received data being synchronized with the clock tick. This guarantees that the start of the cyclic tasks can not be delayed inadmissibly by the Diagnosis task. When no more application tasks need to be processed, the Diagnosis task updates various inputs and outputs.

All the tasks except the pSET tasks operate with a predefined, constant cycle time. These cyclic tasks handle all communication via messages, the I/O modules, and the cyclic operation of the application tasks. If a new control message from the pSET has been received, the Diagnosis task allows the Serial ISR to process the control commands. This takes place asynchronously with the task and may last several cycles.

As mentioned above, only three interrupts, clock tick interrupt, context switching, and serial interrupt for the communication with pSET, are allowed in the RTOS. Among them, clock tick and context switching are basic interrupts for multi-tasking and serial interrupts occur only when the application programs are not in operation services. So the interrupts do not affect the deterministic run of the application programs.

The Shell task is also automatically started by the operating system after each reset. It has the lowest priority of all tasks except the statistics task and the idle task, and is only scheduled when the application tasks and other system tasks are not active.

pCOS satisfies the real time performance requirements in a point of view of priority inversion, interrupt delay time, interrupt response time, interrupt recovery time and scheduling overhead time. The resolution of POSAFE-Q response time is 10ms, and its response time (the response time from PLC data input to output) is 50ms as required in safety systems.

4. Conclusions

This article introduces a real time operating system pCOS, which is the key software component of POSAFE-Q. The structure and the features of pCOS described in this article demonstrate the possibility of the application for the digital safety system of NPP.

REFERENCES

[1] USNRC, "Monitoring the Effectiveness of Maintenance at Nuclear Power Plants," Regulatory Guide 1.160, 1995.

[2] NEI, "Industry Guideline for Monitoring the Effectiveness of Maintenance at Nuclear Power Plant," NUMARC 93-01, rev. 3, 2000.

[3] USNRC, "Lessons Learned from Maintenance Rule Baseline Inspections," NUREG-1648, 1999.

[4] H. C. Yang, et. al., "Development of Maintenance Effectiveness and Target Observation System," Proc. of KNS, Spring, 2004.