Real-time operating system to control multiple industrial plants through process scheduling

M. G. Borja Borja
School of Mechatronics
Faculty of Mechanical Engineering
Universidad Nacional de Ingeniería
Lima, Perú
mborja33@hotmail.com

***Abstract* — This paper presents a proposal to implement a real-time operating system for control of industrial plants by applying scheduling process control in real time. The scheduling process plant control is performed using as input data the duration of the process and the sampling time of control systems. The round robin algorithm does not allow the execution of tasks in real time to control industrial plants since the essential condition is that the process must be run each time sampling and round robin exactly running in a queue order process. To resolve this we use a schedule previously planned using the sampling time and duration of process in relative time of slice and during the planning phase of the ready queue processes for execution on the processor the real time process is entered into the queue as a high priority process. It is introduced a real-time process with higher priority than other processes round robin algorithm to execute the process until it finishes its execution. The main objective of this work is to solve real-time processes run exactly each sampling time is reached because without delays are placed in the run queue of ready processes and executed to complete the entire process.**

Keywords* — *RTOS, real-time, operating system, multitasking, control industrial plants.

I. INTRODUCTION

The microcontrollers are a small computer in a single chip with several modules like digital ports (I/O), timers, analog-digital converters (ADC), pulse width modulation (PWM) and arithmetic and logical units (ALU); among others. These devices have enough computational power and interfaces to implement control systems of industrial plants.

In order to develop applications, a compiler is used in a personal computer. Then, the compiled program is loaded into the ROM memory of the microcontroller with a programmer.

Usually the programming language used is C++ and the result is a single program, which through the execution of sequential statements it can manage all the different states of the plant that are detected with the corresponding modules.

To control multiple plants, the operating system must execute each controller program in a multitasking fashion employing a Round Robin algorithm. Also, it must follow the sample time of each plant based on a programmed schedule.

II. INDUSTRIAL PLANT S CONTROL

On the industry [5] there are many processes that must be controlled, usually involving physical magnitudes such as pressure, temperature, humidity, pH, and velocity; among others.

The control system shown in fig. 1 allows keeping those magnitudes on a range of desired values for a specific industrial process. To achieve this, the physical magnitude is acquired in the form of an electrical signal by means of an ADC. This signal is then transformed into a digital value which is used as feedback to control the output of the plant.

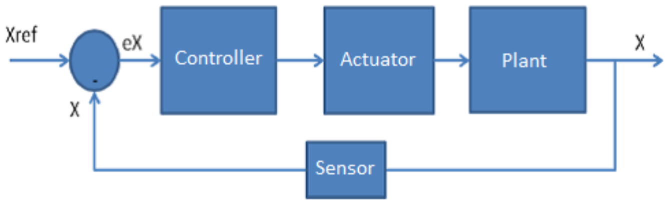


Fig. 1. Feedback control system

The control techniques [2] usually employed are the classic PID controllers (Proportional Integral Differential) or a mix of its sub-components like: P controller (Proportional), PI controller (Proportional Integral) or PD controller (Proportional Differential).

The controller implements the selected control technique using the error magnitude and then the actuator transforms the output signal into a physical magnitude that allows it to manipulate the output of the plant.

The digital control is applied on the plant on a fixed time span which depends on the system dynamics. Therefore, while some systems may require a sample time in the order of milliseconds, other plants work better with a sample time in the order of minutes.

The most complex control technique to program of the classic control theory is the PID controller. With a maximum

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2018.1.1.496
ISBN: 978-0-9993443-1-6
ISSN: 2414-6390

of one thousand machine instructions, this program is executed in a few microseconds.

Currently to control industrial plants a microcontroller is used for each controlled magnitude, or in the best scenario, it is used for cascade control of multiple magnitudes of multiple plants.

The first case raises the cost of the systems because of the oversized hardware. The second case forces all the plants to work at the same time. If one plant consumes all the resources of the microcontroller, it would leave the other plants unattended. This could cause catastrophic results.

III. Computer operating system

A. Definition

An industrial computer has a central processing unit (CPU), program memory and input/output devices like: digital ports, analog inputs and PWM outputs; among others. A software layer, on top of the hardware layer, that manages all hardware components and dialogs with the user through a user-friendly interface is known as operating system.

B. Multiprogramming

Multiprogramming [1] means that a single computer must execute multiple tasks (programs) at the same time while protecting its memory and controlling the access of the programs to the input/output devices. To further enhance the use of resources, operating systems with multiprogramming allows that more than two programs compete for the system resources at the same time.

Multiprogramming has been traditionally used to increase the usefulness of the resources of a computer, allowing multiple users to use the computer simultaneously.

C. Process manager

One of the key functions of an operating system is managing and planning the execution of the different process in the processor of the computer

The planning of the processor is the definition through algorithms of the execution time of each process in the computer.

In operating systems, this planning is done at three levels: high, medium and low.

In the high level, the execution of processes is done through batch processing for programs of low priority and intensive use of resources.

In the medium level, processes suspended due to the lack of resources are planned when those resources become available again.

In the low level, processes ready to be executed are planned to be executed in the CPU.

A Round Robin algorithm is used to implement the multiprogramming technique. This consists in assigning to each process a time span for its execution in the processor, this time span is known as quantum. Usually the length of the quantum is in the range of ten to one hundred microseconds. Each time a quantum ends, the state and context of the processor is saved and the processor is handed to another process.

The processes are arranged in a circular queue following the FIFO rule (First Input, First output) and when one process leaves the processor, it is put at the end of the queue. Other results can be achieved through the use of other algorithms such as: priority rule, biggest size rule and smallest time of execution rule; among others.

A process control block (PCB) is used in the operating system to store all the information needed to execute each process correctly.

The PCB is a structure that contains information about the processor state, the program counter register, CPU registers, CPU planning information, pointers to the queue and memory administration information; among others.

In order to perform the process swap in the processor through the use of the Round Robin algorithm, it is mandatory to interrupt the execution of the process and hand over the control to the operating system, so it can assign the processor to another process.

This interruption can happen as part of the process or through the use of a hardware timer. In both cases the process is interrupted and the control is passed on to the operating system.

IV. Real-time process scheduling

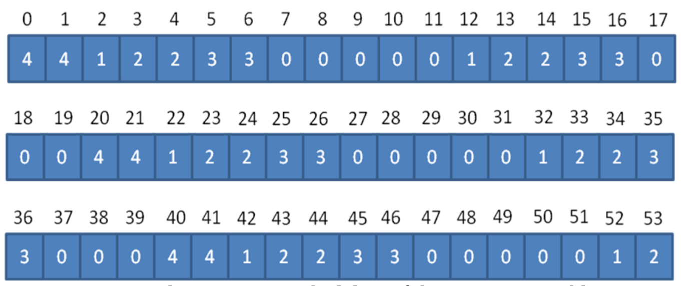
The real-time processes that must be executed on the microcontroller are control systems (see item II) which are small programs that must be executed each sample time.

The process scheduling consists in creating a temporal schedule of processes execution according to the sample time and length of each process, as shown in table I.

TABLE I. PROCESSES TABLE

Nº	**Processes parameters**	
	Sample time	***Length***
1	10	1
2	10	2
3	10	3
4	20	2

The sample time and length of the process are measured in quantums because the operating system can only execute processes in a time span multiple of quantum.



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
4	4	1	2	2	3	3	0	0	0	0	0	1	2	2	3	3	0

18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
0	0	4	4	1	2	2	3	3	0	0	0	0	0	1	2	2	3

36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
3	0	0	0	4	4	1	2	2	3	3	0	0	0	0	0	1	2

Fig. 2. Real-time process scheduling of the processes in table I

On fig. 2 is shown the execution schedule in real-time of the processes described in table I, where each slot represents a quantum and it shows which process is going to be executed in each quantum. A zero in a quantum slot means no process is going to be executed in that quantum.

In order to be possible to schedule all the processes, it is mandatory that:

- All sample times must be multiples of the longer sample times.
- The total length of execution of all processes must be minor or equal to the shortest sample time.
- The schedule is periodic and it repeats itself every least common multiple of all the sample times. On fig. 2, the process schedule has a periodicity of twenty quantums.

V. PROPOSED REAL-TIME MULTITASKING OPERATING SYSTEM

The proposed operating system focuses on executing real-time control processes for industrial plants must be executed each sampling time and this depending on the dynamics of the plant can be milliseconds, seconds, minutes, etc. It is not run quickly.

Other solutions such as RTLinux systems are used to control high level because they have no resources as analog inputs, PWM outputs and other required for low-level control.

Some proposals [6] focus on speed of execution of tasks in embedded systems is a different problem discussed in this paper.

There are also proposals for real tine operating systems [7] solve the problem of ensuring the execution of the same program on different interactions at the same time.

Taking into account the characteristics of the industrial control processes to be executed in the processor, according to the condition that the processes must be executed in an independent fashion and using microcontrollers with a read-only memory (ROM) to reduce the cost of hardware; a memory organization [5] is proposed for the operating system as shown in fig. 3. On fig. 4 it is shown an example of the distribution of the operating system and the processes inside of the program memory of the microcontroller.

In order to manage the processes, a processes vector, whose elements are PCB structures, is used. The implemented PCB structure stores information such as: the process identifier, process program start address, process program current address, process state, process work register, process status register, process time of execution in quantums, process priority and process name.

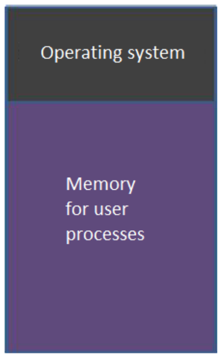


Fig. 3. Memory organization

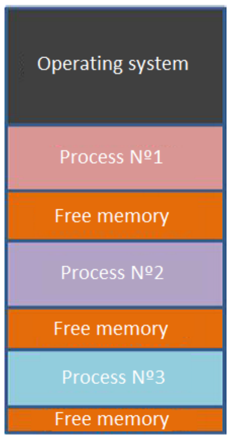


Fig. 4. Distribution of the operating system and processes inside of the program memory

A process queue in a vector containing the processes identifiers is used to manage the executions of the processes. Those processes are organized according to their priorities. If all have the same priority, the queue is organized following the FIFO rule.

To swap the execution of the processes a hardware timer is used to interrupt the process and hand over the control to the operating system.

Real-time processes have a high priority and are executed on the operating systems along with low priority processes. If a real-time processes takes longer than a quantum, it is interrupted and then the operating systems generates a new queue, putting the real-time process on the top again because of its priority. This means if a real-time process starts its execution, it is going to be on top of the queue until it is finished.

In order to implement the execution of the schedule of real-time processes, a schedule vector is created on the operating system. Then, when the execution of real-time processes begins, a clock (counter of quantums) is initialized.

If a real-time process is programmed to be executed in a quantum that matches the value of the clock, that process is loaded into the queue and executed in a multitasking fashion. This goes on until the end of the schedule is reached. At this time, the clock is restored to zero and the schedule is executed again from the beginning.

An operating system for a PIC microcontroller has been developed and it is based con multitasking with the capacity of process scheduling.

RESULTS

Using the theoretical key concepts of operating systems reviewed on item III, a multitasking operating system has been developed. Based on the scheduling considerations shown on item IV, the execution of real-time processes fulfilling the expectations mentioned on item II became possible. This lead to the real-time multitasking operating system shown on item V.

The real-time operating system proposed and implemented in a microcontroller satisfies exactly with the requirements of process control industrial plants where it is very important to comply with the execution exactly at each sampling and time with a single microcontroller can control many industrial plants independently.

The main difference with existing real time operating systems that runs in expensive personal computers or on proprietary architectures, this proposal runs on small microcontrollers in low level industrial control systems.

CONCLUSIONS

- The developed operating system allows controlling multiple industrial plants, achieving the execution of each controller in the specified sample time use small microcontroller.
- It is possible to schedule processes with sample time that are multiple of the longer sample times.
- The sum of the length of execution of all the processes must be minor or equal to the shortest sample time.
- The amount of controlled plants and the smallest possible sample time depends on the velocity of the microcontroller or industrial microcomputer used.
- The real application of this operating system is very easy as only need a small program that implements round robin and complemented by the administration of the ready queue processes according to the schedule planned in advance and should be distributed in memory the operating system and control programs as shown in figure No. 4.

REFERENCES

[1] A. S. Tanenbaum, “Sistemas operativos modernos,” Pretice Hall. Mexico, 2009.

[2] K. Ogata, “Ingeniería de control moderna”. Pretice Hall. Mexico, 2010.

[3] J. Famerl, “A Real-Time Operating System for PICmicro™ Microcontroller”, MicroChip, 1997.

[4] Foster, Caxton, “Real Time Programming - Neglected Topics,” Reading, Massachusetts, Addison-Wesley Publishing Company, 1981.

[5] Borja M. Sistema operativo multitarea para control independiente de múltiples plantas industriales aplicando microcontroladores de bajo costo.

[6] C.Yaashuwanth. Dr.R.Ramesh. A New Scheduling Algorithmsfor Real Time Tasks. International Journal of Computer Science and Information Security, Vol. 6, No.2, 2009

[7] Samarth Shah. Real-Time Operating Systems: The Next Stage in Embedded Systems. Advance in Electronic and Electric Engineering.ISSN 2231-1297, Volume 3, Number 5 (2013), pp. 543-552.