Design and Implementation of an EDF Scheduling Class in Linux for Soft Real-Time Applications

Student's Name	**Mohammad Zafar Ziya**	**RTES (PT-2011)**
Academic Supervisor(s)	Jishmi Jos Choondal and N. D. Gangadhar	
Industrial Supervisor(s)		

Mohammad Zafar Ziya

md.zafar.ziya@gmail.com
Ph. No: 0 89713 96071

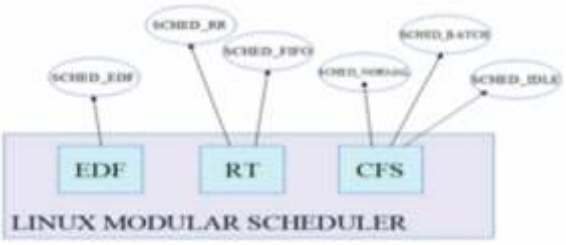
Keywords: EDF (Earliest Deadline First), Process management, Scheduling, Soft Real-Time, Linux

Abstract:

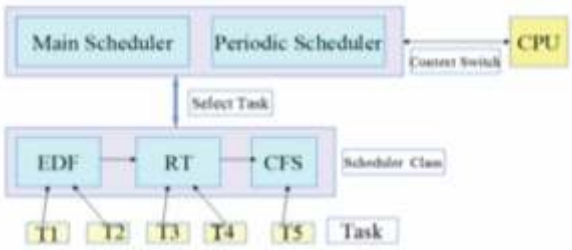
Real-time applications have become very common in general purpose desktop systems. Real-time performance is greatly influenced by the real-time scheduling algorithms employed by the operating system. When compared with other real-time scheduler algorithms, Earliest Deadline First (EDF) has the advantage of higher CPU utilization and handling aperiodic requests while meeting the timing deadlines. In addition, EDF is desirable under system overload conditions. However, industry has not yet adopted it owing to its higher runtime overhead as well as being complex to implement.

This project deals with the development of an EDF scheduling class on Linux 3.x kernel using the Linux modular scheduler architecture. The developed EDF scheduler provides timing facility to the scheduler. Our EDF implementation applies a bandwidth reservation technique for providing the guarantee of execution time of the task using a data taken based on task runtime and deadline. The EDF scheduling class is designed to follow the Global EDF scheduling approach for task scheduling making it usable on X86_64 based Intel multiprocessor systems. This work involves the addition of EDF scheduling class function routine, which are invoked by the main scheduler. The EDF run-queue is implemented using a Red-Black tree data structure and one run-queue for each CPU on multi-processor platform. EDF class task are placed above the RT class tasks in priority hierarchy and below the STOP class tasks.

The performance of the developed EDF scheduler is measured and compared against the RT-FIFO real-time scheduler for varying system load of 1-95% of CPU utilizations as well as varying deadline for 90% system load using multiple runs of a task set. Under varying system load, the EDF scheduler is able to complete all the tasks before the deadline where as the FIFO scheduler missed the deadlines under overload. With 90% loaded system, the EDF shows better slack results than FIFO. It also consumed same or less CPU ticks to complete the tasks. One can employ more efficient data structures for run queue management for improving the performance of the developed scheduler.



Linux scheduler with EDF scheduling class



Main scheduler and scheduler class relationships in Linux

Priority hierarchy of scheduler modules