Ain Shams University
Faculty of Computer and Information Sciences
Computer Science Department

Operating Systems Scheduling Algorithms

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Computer and Information Sciences

Prepared By
Noran Wael Hossam el Din Ahmed
Demonstrator at Computer Science Department

Supervised By
Prof. Dr. Abdel Badeeh Mohamed Salem
Vice Dean for Post-Graduate and Research & Head of Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University

Prof. Dr. Mostafa Mahmoud Syiam
Department of Computer Science
Faculty of Computer and Information Sciences
Ain Shams University

Dr. Shymaa Mohamed Arafat
Department of Computer Science
Faculty of Computer and Information Sciences
Ain Shams University

Cairo - ٢٠٠٧

ACKNOWLEDGEMENTS

This is a great opportunity to express my thanks to all who contributed to the fulfillment of this work. First, all gratitude is due to God, who made this success possible and affordable.

I was proud to work under the supervision of the late Prof. Dr. Mostafa Syiam. I wish to express my sincere thanks for his valuable advice, and encouragement during this work and special thanks to my current supervisor Prof. Dr. Abdel Badeeh Salem.

I was honored to work under the supervision of Dr. Shymaa Arafat. I wish to express my thanks, respect and appreciation to her for her useful assistance and guidance all the time. She gave me all the support I needed throughout this work.

I am also grateful for the guidance given to me by Prof. Dr. Essam Khalifa. My thanks also include the head and staff members of the Computer Science department for their kind help.

Finally, I would like to mention the major role of my family and my husband.

ABSTRACT

Processor task scheduling in real-time systems is a complicated problem that has been the subject of many research articles. In this thesis, we present a centralized algorithm for scheduling real-time tasks in a heterogeneous multiprocessor system. We first use an offline algorithm for scheduling periodic static hard real-time tasks where reliability and fault tolerance measures are considered. Then, in order to schedule dynamic aperiodic firm real-time tasks we suggest two approaches -other than the traditional linear search methods- to be used along with the offline algorithm. In the first approach, we use an online probabilistic algorithm, which starts the search for an adequate gap in a processor's schedule, from the processors having the highest probability of success. And in the second approach, we propose a new online cut-off technique to be used along with the same offline algorithm. This is done by cutting off unsuitable processors, and then choosing from the remaining ones, using heuristics that we will call best-fit and worst-fit following the naming conventions of memory management. When a new online task arrives, we first exclude processors whose largest idle gap is smaller than the execution time of the task. Then, we start the search either from the processor having the largest free gap (worst-fit), or the processor with the least largest gap that can hold the task (best-fit). A simulation was done for both algorithms where a scheduling time improvement and a success ratio improvement were found in the cases of worst-fit and best-fit respectively, when compared to the traditional linear search, through all processors.

Key-Words:

task scheduling algorithm, aperiodic firm real-time tasks, periodic hard real-time tasks, heterogeneous multiprocessor system, fault tolerance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

CHAPTER ١

INTRODUCTION

Ever since operating systems allowed for multitasking, they were forced to provide a methodology for competing tasks to manage computer resources among them. Most resources are mutually exclusive, thus algorithms are needed to allocate resources to tasks by the means of scheduling. The processor(s) are themselves the most important resource that needs to be managed. This thesis is concerned with processor scheduling.

١.١ SCOPE OF THE WORK

The scheduling problem involves many performance metrics and tradeoffs, such as response time versus throughput and fairness versus balancing of resources; the word optimal may have different meanings in different systems. More constraints are even imposed in real-time systems, such as meeting deadlines, and fault-tolerance requirements where the system is supposed to meet its tasks' deadlines even in the presence of failures. A great improvement in real-time applications happened after applying them in multiprocessor systems, but taking into consideration that scheduling in multiprocessor systems is much more complex than that of uniprocessor systems.

١.٢ AIM OF THE WORK

In general, the problem of obtaining an optimal schedule for a set of real-time tasks in a multiprocessor system was proved to be NP-complete [1].

Consequently, existing results are approximation algorithms or heuristics to achieve a near optimal solution in a polynomial time.

In this thesis we present a centralized multiprocessor algorithm for scheduling randomly submitted online aperiodic tasks along with offline periodic tasks in a real-time environment while considering several issues such as reliability and fault-tolerance. In the offline part, time is not very significant. On the contrary, in the online part, time is greatly significant, so non-conventional approaches in scheduling where presented instead of the conventional linear search.

١.٣ ORGANIZATION OF THE THESIS

The remaining of the thesis is organized as follows: in chapter two a background on the topic is given. In chapter three, a more detailed discussion of the topic is presented and the related work is surveyed. In chapters four and five, the proposed algorithm is presented, its complexity is studied, and it is implemented, where the simulation results show that the proposed algorithm is superior to other existing algorithms. Finally, the conclusion of the work and some possible future extensions are presented in chapter six.

CHAPTER TWO

SCIENTIFIC BACKGROUND

CHAPTER ٢

SCIENTIFIC BACKGROUND

٢.١ SCHEDULING OVERVIEW

Over the years, scheduling has been the focus of intensive research, and many different algorithms have been implemented. Today the emphasis in scheduling research is on exploiting multiprocessor systems, particularly for multi-threaded applications and real-time systems.

We will start by defining and clarifying many design issues and performance criteria related to scheduling and present a variety of uniprocessor scheduling algorithms.

Afterwards, we will discuss the two areas that are the focus of contemporary scheduling research. The first area to discuss is multiprocessor scheduling, where the presence of multiple processors complicates the scheduling decision but opens up new opportunities. The other area is real-time scheduling, where requirements go beyond fairness or priority to specifying time limits for the start or finish of given tasks or processes.

The aim of processor scheduling is to assign processes to be executed by the processor or processors over time, in a way that meets the system objectives, such as response time, throughput and processor efficiency. Scheduling affects the performance of the system because it determines which processes will wait and which will progress.

Fundamentally, scheduling is a matter of managing queues to minimize queuing delay and to optimize performance in a queuing environment [٢].

٢.٢ PERFORMANCE CRITERIA

The commonly used performance metrics can be categorized along two dimensions. The first is whether it is user oriented or system oriented criteria. ***User oriented*** criteria relate to the behavior of the system as perceived by the individual user or process. In the ***system oriented*** criteria, the focus is on effective and efficient utilization of the processor.

The second dimension is whether the criterion is performance related or not directly performance related. ***Performance related*** criteria are quantitative and generally can be readily measured. Criteria that are ***not performance related*** are either qualitative in nature or do not lend themselves readily to measurement and analysis.

Table ٢.١ summarizes key scheduling criteria. These are interdependent and it is impossible to optimize all of them simultaneously. For example, providing good response time may require a scheduling algorithm that switches between processes frequently. This increases the overhead of the system, reducing throughput. Thus the design of a scheduling policy involves compromising among competing requirements.

Table 2. 1 Scheduling Criteria

User oriented, performance related	
Response time	• This is the time from submission of the request until the response begins to be received • For an interactive process not a batch process

	Table ٢.١ Continued • The process can begin producing some output to the user while continuing to process the request • It should be minimized • Number of interactive users receiving acceptable response times should be maximized
Turnaround time	• This is the interval of time between the submission of a process and its completion • Includes actual execution time plus the waiting time • Appropriate measure for a batch job
Deadlines	• When specified, scheduling discipline should subordinate other goals to that of maximizing percentage of deadlines met
User oriented, other	
Predictability	• Any given job should run in about the same amount of time and at about the same cost regardless of the load on the system • Wide variations in the response times and turnaround times distract the user
System oriented, performance related	
Throughput	• This is the number of processes completed per unit time • Should be maximized
Processor utilization	• This is the percentage of time that the processor is busy • Significant in expensive shared systems and less important in single-user and real-time systems.