SMART TASK MANAGER FOR USER DEFINED PROCESSES

Aayush Srivastava¹, Akshat Singhal², Apoorva Jain³

¹Student, Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, Pune, Maharashtra, India

²Student, Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, Pune, Maharashtra, India

³Student Department of Electronics and Telecommunication Engineering, Bharati Vidyapeeth College of Engineering, Pune, Maharashtra, India

Abstract

Managing and defining multiple tasks over different physical and digital workspaces are done by using various strategies. In this paper we put light on how computer based task management tools can support task management on a desktop or a smart phone and how it reacts in a fatal situation to the processes out of limit. The key concept in any of the operating system which allows the particular system to support multitasking system, etc. is Task Scheduling. It's the core which refers to the basic way; different processes are allowed to share the common CPU. Scheduler and dispatcher are the software which helps to carry out this project.

Keywords: Task Functionality; HWOS; Memory; Creating the Application; Scheduling; GEMPACK.Etc.

1. INTRODUCTION

Task Manager is a software which display all the running programs, processes and service on the system. Not only this task manager is also used to analyze your computers performance, shut down non responsive tasks, view network status if you are connected to a network and it allows user to keep a track of how their network is functioning. When more than one users are connected to the system the task manager can be used to see who all are connected to the system, which operations are being performed by them and it can also send messages to the various nodes connected to the system.

Task Manager can be used to:

- Setting process priorities.
- Adjust process compatibility.
- Abort programs.
- Restart or Shutdown system.

Task automation provides you the basic ability to schedule repetitive document management tasks using task scheduler. For example- The task manager can be used in document locator to computerize importing of document by observing a "Watch Folder" that generally receives scanned files from a Multi functioning printer (MFP).

2. PROBLEM DEFINITION

In recent years there has been a large increase in the demand of High Throughput Computing (HTC) owed to the new systematic challenges of handling files and tasksconcurrently. Various ventures have tried to come up with different interfaces that are convenient for users.But these are designed for a very specific field and can not be applied on a larger scale to address various research fields collectively. To overcome this problem, this task manager is developed, which provides a very light management system for universal-purpose and is capable of working with a large number of different computational resources. The proposestructure here is able to complete thousands of different processes or tasks using numerous algorithms while collecting the results in a simple manner. In this system, the limit is set for the process such as hundreds of processes or even thousands of them and if the limit goes above it the scheduling algorithms comes into play and according to the algorithms the processes are managed and also given the priority in that particular fatal situation.

3. LITERATURE SURVEY

The suggested system in this particular article, has a stute capabilities and creative task management tools which support task management and have been acknowledged in recent times. Some spotlight features include task centric organization of the document and resources, and the use of smart personal assistant for completion of the task. This research provides us with an acumen of task management tools and its various possibilties to provide high througput computing by deploying a support platformfor our tasks.

Task centric organization of available resources or enabling activity is a crucial application formanaging a task. Certain systems with task centric organization require manual input. e.g. - Towel is a task management application that couples a user's to-do-list with a project execution assistant that can complete a learnt set of task capabilities on behalf of the user.

The tasks on each processor are scheduled for execution by an instance of RTOS (Real time Operating system). To improve performance, a Hardware task manager (HWTM) block that simply educes the Worst Case Execution Time (WCET) of the RTOS.

Basically, Round Robin Scheduling algorithm has some problems so, for developing a task manager a basic integrated approach could be set for the Round Robin and Priority scheduling to improve the performance of the CPU.

DIRAC i.e. Distributed Infrastructure with remote agent control can also be a main factor, developed basically to facilitate user analysis tasks spread across both grid and non-grid computing resources and forlarge scale simulation. DIRAC, which has been recently used, involves more than 40 sites and 90 TB of data.

Due to a hike in the demand of HTC (High throughout computing) because of the new scientific challenges, General workload manager (GWM), a universal purpose very light management system, which is capable of working with different computing resources has also been pruposed.

4. BASIC DIAGRAM

The task manager basically deals with the various tasks having priorities High to Low. Here, the HWOS which is the Real time kernel selects the tasks depending upon the priority and assigns it to the CPU.

TheCPU (Central Processing Unit) or MPU (Micro processing Unit) time needs to be managed efficiently to achieve high levels efficiency. Real time kernels are the software that are used for this purpose. Most kernels are written using C language incorporated with assembly language so that they can easily adapt to the various CPU architectures that they are deployed upon.



Fig. 2.1 Basic Block Diagram for task manager

A DPR system is made self-sufficient by HwOS. In previous works, these services were mostly an extension of the Linux OS running on top of the DPR system. A part of the HwOS acts as RAM to the CPU. This RAM is used to observe HwOS functions and give changes.

5. DESIGN AND IMPLEMENTATION

5.1 Task Functionality

When the system loads, all the processes running on the desktop are loaded. Timer object is created to call all the processes by LoadAllProcesses()with an interval of 1 second. In this method, we use GetProcesses where in a

Hash table all the running process details are stored. If the compared process ID of both the Hash table and the list view match then, the updation of its details is done (i.e. Processor time, number of threats).

The ended process's IDwill be removing that item from the list who's process has ended. Generally weobtain the processes ID which are aborted through searching the items which are there in the list View but absent in the List of the current process.Through this the process details present in the list get updated.Without any flicker it controls every second.

We will get the user selected task's path via the framework of new process details form to create a new task.

Process.Start (task name, arguments) will then create that process.

We will now be calling Process.GetProcess by Id (selected process ID, machine name).Kill(). to give the functionality to end the selected processes.

5.2 Priority

Through the context control menu we can set the priority of any running process. The correct menu item will first get a check mark based on its pirorty by the context menu.We can an change the priority of the context menu poping in the task manager by clicking on appropriate menu item. On the status bar, we would display threads and the total count process in each panel

5.3 Algorithms

In this project, we can use quite a few scheduling algorithms to control the fatal situation basically to set the priority of the processes running in the designed task manager and then also prioritize the tasks according to the user's preference. The basic scheduling algorithms which can be used to design task manger to handle many processes are briefed below -:

5.3.1 First Come, First-Served Scheduling (FCFS)

FCFS is a first come first serve based scheduling algorithim, the process are executed in order in which they arrive into the ready queue. This is one of the first and the simplest scheduling algorithm. Process are dispatched based on their arrival time on the ready queue. It's a non premitive scheduling algorithim which means that the process in execution is not interepted by the scheduler it runs its complete duration. The FCFS algorithim are simple but comes with a disadvantage that long jobs make short jobs wait and unimportant jobs make important jobs wait.

Table 1.1 FCFS				
Process	Burst Time			
P(a)	15			
P(b)	5			
P(c)	3			

Suppose that the process arrive in the order P1, P2, P3.

The Gantt chart for the schedule is:



Waiting time for P(a)=0; P(b)=15; P(c)=20Average waiting time: (0+15+20)/3=11.66So, the average here is **11.66ms**.

5.3.2 Priority Scheduling

In the scheduling algorithm, each process is given a rank or priority. The priorities can be in any order, with either the minimum no. denoting the higest priority, or the lowest no. resembling the highest priority. Process with same priorities can be executed on the basis of the conventional FCFS algorithim. Processes are assigned priorities on various parameters ranging from importance, to memory usage.

 Table 1.2 Priority Scheduling

Process	Burst Time	Priority
P(a)	12	1
P(b)	6	3
P(c)	2	2
P(d)	3	4

P(a)	P(c)	P(b)	P(d)	
0		1214	20 23	

Priority Gantt chart

Wait time of each process is following: P(a)=0; P(b)=14; P(c)=12; P9d)=20

Average Wait time: (0+12+14+20) / 4 = 11.5 So, the average here is 11.5ms

5.3.3 Round Robin Scheduling

Round Robin scheduling algorithm is based on the principle that every player gets a turn. Based on this, the CPUmanages the execution of the process by giving each process in the ready queue their chance. RR scheduling is the other name for this algorithim. The algorithim is very efficient in time sharing system. The process get their turn on FCFS basis. Each process is gets their turn to executed based on their or arrival time on the ready queue but they are executed for a predefined unit of time. As soon as the time is up the process is switched.

Process are executed in one time slice or quanta which is the small period of time which is predefined. The process are

kept in a ready queue which is a circular queue where new processes are added at the tail of the queue.

Round Robin algorithm are used to fairly devide the execution time between the processes, no processes are prioritezed, the response time issignificantly reduced, the algorithm uses premptive scheduling.

The only interesting issue with round robin scheme is the length of the quantum but if the length is varied too much the CPU effeciency reduces.

E.g. - Time Quantum = 4

Table 1.3 Round Robin Scheduling

Process	Burst Time
P(a)	20
P(b)	4
P(c)	5

The Gantt chart for the schedule is - :

P(a)	P(b)	P(c)	P(a)	P(c)	P(a)	P(a)	P(a)
0 4	4 81	216	17	21	25 2	29	

Average waiting time= [12+4+12]/3=9.33 sec.

5.4 Memory

CLR memory details like heap size are displayed in this part of the task manager. Improper garbage collection slows down the speed of some of the applications. Then we can determine whether the processes or any application is running slow because of insufficient CLR memory and then this helps us in making decision whether the garbage collector should be called explicitly in our application mode or not.



Fig. 3.1 Memory Usage

When there is shortage of memory i.e. RAM then the Windows at the cost of severe speed reductionuses hard disk space (called "paging"). A GEMPACK program starving for raw would be quite slow. A lot of memory is consumed by the OS and the background processes. To investigate how much memory is required to run one simulation and to meet background needs use task manager. Memory option in the task manager shows the usage of the memory and the available memory and also the cached memory.

P	2				Manag	er
File	Opt	ions	View	_		
Proc	~	Alw	ays on top	Jsers	Details	Service
	~	Min	imize on use			
News	•	Hid	e when minimized			8%
Nam		Sho	w full account name	\vdash		CPU
Ap		Sho	w history for all processes			
ь А			Fig 3 2 Options in Task Ma	nage	r	00/

Fig. 3.2 Options in Task Manager

6. SOFTWARE REQUIREMENT

SPECIFICATION

This Software Requirement Specification also known as SRS will provide a description for a task management system. In this SRS section, the document purpose, the scope of the task management product will be defined.

6.1 Document Purpose

The software requirement specification will describe the processes and the functions of the task management system. A possible upgrade for the product made is described in the project which may include the point's rewards system redeemable for basic upgrades.

6.2 Product Scope

The task management system provides an interface for both managers and workers to track their daily tasks. The manager or any other user will have a traditional worker interface. The interface will allow the manager to enter new tasks, editing them and also deleting them.

And, this system will also provide a history of previous tasks assigned or completed.

6.3 Operating Environment

The task management system designed will run on Windows 8 or above platform or can also run on Mac X system.

As shown in the figure, the task manager should be connected with the application on the desktop and also connected with the SQL database.



Fig. 5.1 Task Manager Environment

7. CONCLUSION

The conclusion of this project is to handle the excess tasks in the task management system by setting the limit for the tasks to enter in the system, we've used the algorithms such as that of Round Robin Scheduling, Priority Scheduling and the First Come, First- Serve Scheduling to manage the excessive tasks in the task management system so as to manage the fatality situation which could occur if the limit would not have been defined.

ACKNOWLEDGEMENT

All the authors are thankful to the Project Faculty members, and the management for their support in providing the hardware as well as the software for the need.

REFERENCES

[1]. Christina Yum, "Towards Ubiquitous Task Management", 'School of Information Technologies, The University of Sydney', Australia, 2 November'09.

[2]. Pavel G. Zaykov, "Hardware Task Status Manager for an RTOS with FIFO Communication", "Delft University of Technology", CE Lab, Mekelweg 4, 2628 CD Delft, The Netherlands.

[3]. Ishwari Singh Rajput, Deepa Gupta, "Department of Computer Science and Engineering", Amity School of Engineering and Technology, Amity University, NOIDA, UP, India.

[4]. Stokes-Rees, "DIRAC: A Scalable Lightweight Architecture for High Throughput Computing", "Department of Particle Physics", University of Oxford, OX1 3RH, UK.

[5]. T. Ferrari and L. Gaido, "Resources and Services of the EGEE Production Infrastructure", "Journal of Grid computing", Vol. 9, Page Number 119-133, 2011.

[6]. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", Sixth Edition.

[7]. G. Indalecio, F. Gomez-Folgar, A.J. Garcia-Loureiro. "General Workload Manager: A task manager as a service", 2015 IEEE International Conference on Communication Workshop (ICCW), 2015.

[8]. Noon, Abbas; Kalakech, Ali and Kadry, Seifedine. "A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average", International Journal of Computer Science Issues (IJCSI), 2011.

BIOGRAPHIES



Aayush Srivastava, Student, Department of Computer, Bharati Vidyapeeth College of Engineering, Pune, Maharashtra, India



Akshat Singhal, Student, Department of Computer , Bharati Vidyapeeth College of Engineering, Pune, Maharashtra, India



Apoorva Jain, Student, Department of Electronics and Telecommunication , Bharati Vidyapeeth College of Engineering, Pune, Maharashtra, India