International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 1 | ISSN : 2456-3307

Performance Evaluation of Embedded System Using Scheduling Algorithms

M. Sreenath[1], Dr. P. A. Vijaya[2]
[1]Research Scholar, Department of E.C.E., BNMIT, Bangalore under VTU, Belagavi &Asst. Professor, Department of E.C.E., P.I.T.S., Ongole, Andhra Pradesh, India
[2]Professor & Head, Department of ECE., BNMIT, Bangalore, Bengaluru, Karnataka, India

ABSTRACT

The main aim is to share or allocate the processor time to all the tasks using scheduling algorithm in a system. The List and McNaughton's scheduling algorithms, minimize the maximum completion time of all tasks (Cmax). An ordered list of processes will be made and assign priorities to them by using List scheduling algorithm. McNaughton's scheduling algorithm only for independent tasks has to be scheduled on identical processors for the sake of schedule length minimization. Hu's algorithm is planned to schedule the tasks with in-tree precedence constraints, consider unit length tasks. In Hu's algorithm all processors should be of same type. These scheduling algorithms mainly for enhance the performance levels of the system by reducing the time delay. Thus, preferably use these algorithms in the areas of manufacturing & production, Transportation & distribution and Information-processing. List, and McNaughton's scheduling algorithms are used to reduce the difficulty in allocating the tasks to the processors. TORSCHE scheduling toolbox for MATLAB is used to schedule the tasks on processors. User can solve, scheduling problems with the help of MATLAB routines and functions by considering appropriate configurations like resources, task parameters and optimization criterion.

Keywords: Scheduling algorithms, TORSCHE, MATLAB, Cmax. Tardy tasks.

I. INTRODUCTION

Gives a major importance to the embedded technology in integrating the different functions associated with it. Need to consider various functions in a closed loop scheme manner. This changes greatly effects on the parameters like manpower, effectively utilize the time and operates efficiently without human interference. An embedded system is a combination of software and hardware to perform a dedicated task. Allocation of CPU time to the processors and assign time slots to processes and threads take care by Scheduling algorithm. Scheduler is a machine that organizes or maintains schedules. Scheduling algorithms are mainly used for reducing the level of resource starvation problem [1] and creates a good path for utilizing the resources by the processor. TORSCHE scheduling toolbox [2] is a open source toolbox, mainly used for implementing the new scheduling algorithms and effective utilization of processors. MATLAB environment supports TORSCHE toolbox by writing the programs in MATLAB object oriented programming language.

The load balancing between resources is important to get effective utilization when many jobs are running with varying characteristics. Balancing the workload by distribution technique among multiple resources. Aim of the Load balance technique for minimizing the utilization of resource use, increase level of throughput, reduces the time of response, and overcome overload effect of any single resource. The system speed is reduced when the work load is not properly balanced and idle processors [3] are not involved for task execution. To reduce the work load on any single processor and to make use of all the available parallel processors [3]. Three types of scheduling algorithms (i.e., List, Hu's and McNaughton's) are used in this implementation. By making use of these three scheduling algorithms Cmax is minimized and tardy tasks are also reduced. The main objective is by making use of these three scheduling algorithms, delay in the process of manufacturing is reduced and thus

increasing the number of products manufactured in certain time period. This can be achieved by utilizing the resources in the best way possible. Hence, three types of scheduling algorithms are used in the process of manufacturing to obtain an optimal schedule [4] and thus enhance the performance [5] of the system.

The scheduling algorithms are real time in nature for uniprocessor systems, which can be categorized into two classes: off-line and on-line. On-line algorithms are partitioned into either static or dynamic-priority based algorithms [6]. For dynamic-priority based algorithms, normally have two subsets; namely, planning based and best effort scheduling algorithms. Multiprocessor scheduling algorithms [7] is another class of real-time scheduling algorithms. Also describes the techniques to deal with aperiodic and sporadic tasks, precedence constraints, and priority inversion. Most of the practical formulations of the scheduling problem are NP-Complete [8]. These instances contain combinations of dependency, timing and resource constraints. Two most widely used approaches are *list scheduling* [9] and *force-directed scheduling*. List scheduling maintains an ordered list of operations that can potentially be scheduled at a control step with no violation of data dependency. Considering one control step at a time, operations are selected from this ordered list one by one according to some priority function and scheduled at the control step under consideration.

In force-directed scheduling the goal is to create a balanced distribution of operations among control steps. Using the mobility of each operation to define possible intervals of execution, potential *demand* for each control step is determined. The operation-to-control step assignment, which will contribute towards the most homogeneous distribution, is accepted at each step. This approach has been incorporated into high-level synthesis systems as well. Another type of scheduling method is referred to as path-based scheduling. This method has been proposed for scheduling control flow graphs [10]. Paths of execution within the control/data flow are handled individually. Each such possible path is scheduled independently in an optimal fashion [11]. Then the final schedule is constructed by imposing the resource constraints and overlapping the path's schedules accordingly. Other popular techniques for scheduling with control flow are trace scheduling from microcode compaction and percolation scheduling. While most of the above mentioned scheduling heuristics produce a scheduling decision for one operation at a time, sequence of algorithms generate an assignment between multiple operations and control steps. The particular set of operations to be scheduled at each step constitutes a chain of data dependency. This may be also called a path spanning through the DFG.

II. TABLE MANUFACTURING

The manufacture of table will be started with collecting and arranging all parts of the table. It generally consists of 4 legs, 1 top and 1 drawer. Consider it may be movable or fixed table.

Initially take all 4 legs for manufacture and each leg will take 5 processing time. Finally it will take processing time of 20 for manufacturing the 4 legs. In the table 1, processing time 5 represents time taken for each leg. Similarly take 10 for Top, Assemble 1, Drawer and Assemble 2. Now, overall processing time for manufacture the table is about 60. The table is a basic need of house appliances. Apply LIST scheduling algorithms on application of table manufacture for arranging the order and reconfigure the execution of all tasks one by one. The execution time of the tasks on available processor should be reduced after applying LIST scheduling algorithm. Basically LIST used for reolving the issue of CMax problem.

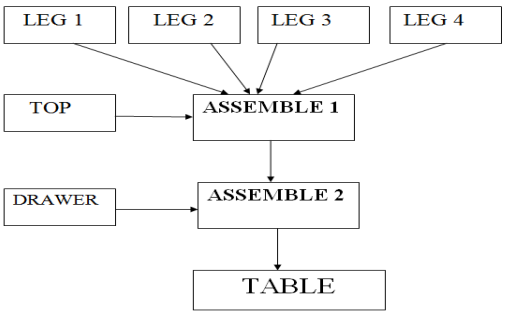

Table 1. Details of name of task, processing time, release time.

S.NO	NAME OF THE TASK	PROCESSING TIME	RELEASE TIME
1	LEG1	5	0
2	LEG2	5	0
3	LEG3	5	0
4	LEG4	5	0
5	TOP	10	0
6	ASSEMBLE1	10	0
7	DRAWER	10	12

8	ASSEMBLE2	10	0

III. RESULTS

By using List, Mcnaughton & Hu's scheduling algorithms the tasks of "table manufacturing" are scheduled in a way that more number of products can be manufactured.

1. RESULTS OF LIST SCHEDULING ALGORITHM:

An Application of TABLE MANUFACTURING given with set of eight tasks with names, processing time and release time is shown in Table 1. Longest Processing Time first (LPT) [12], intended to solve PjjCmax is a strategy for LS algorithm. The tasks are arranged in non-increasing order of processing time pj before the application of List Scheduling algorithm Shortest Processing Time first (SPT) [13], intended to solve P||Cmax problem, is a strategy for LS algorithm. The tasks are arranged in non-decreasing order of processing time pj before the application of List Scheduling algorithm. The schedule obtained by using four strategies of LS algorithm shown in below Figure 2 and Figure 3. Earliest Completion Time first (ECT) [14], intended to solve P||_Cj problem, is a strategy for LS algorithm in which the tasks are arranged in non-decreasing order of completion time Cj in each iteration of List Scheduling algorithm. Earliest Starting Time first (EST)[15], intended to solve P||Cj problem, is a strategy for LS algorithm. The tasks are arranged in non-decreasing order of release time rj before the application of List Scheduling algorithm.

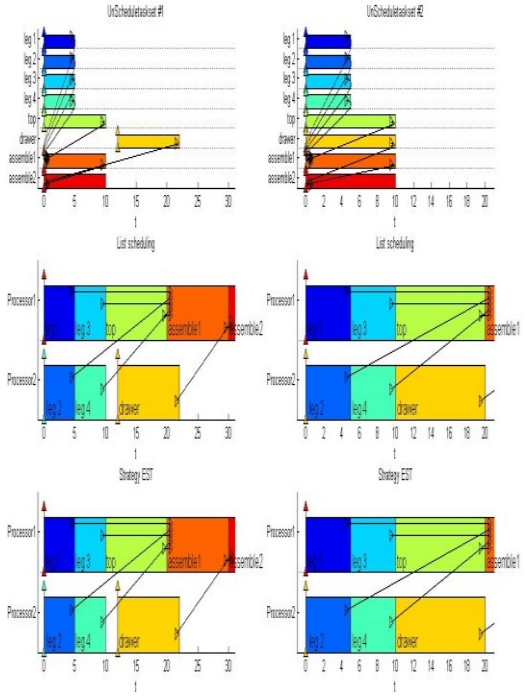

Figure 2. Results of LS algorithm with EST strategy

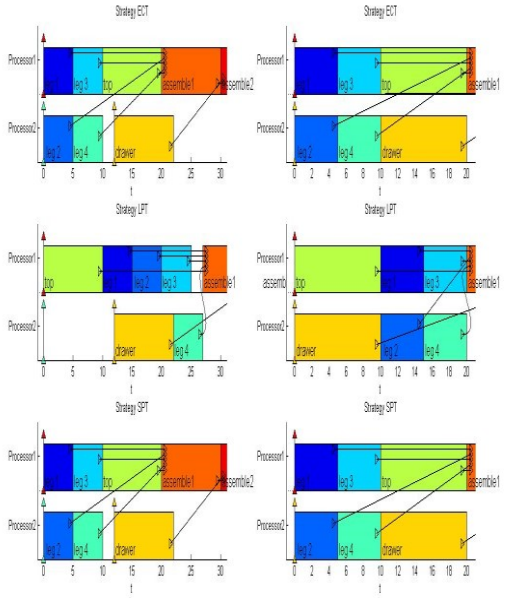

Figure 3. Results of LS algorithm with ECT,LPT and SPT strategies.

2. RESULTS OF MCNAUGHTON'S SCHEDU - LING ALGORITHM

MCNAUGHTON'S ALGORITHM [16] is used to equal time sharing of processors in order to minimize schedule length. An Application of TABLE MANUFACTURING given with set of eight tasks with names, processing time and release time is shown in Table 1. The schedule obtained by using MCNAUGHTON'S algorithm shown in below Figure 4.

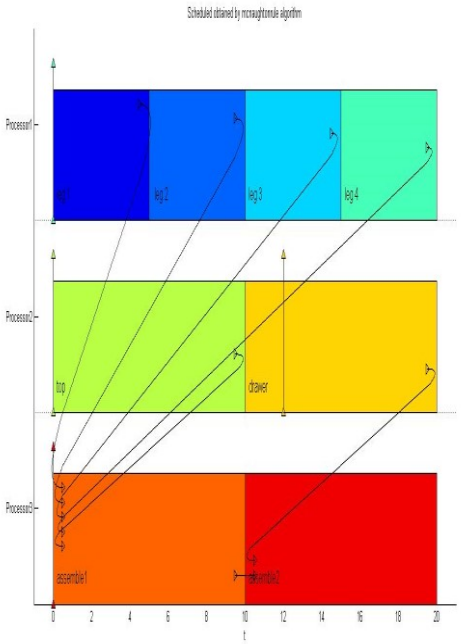

Figure 4. Results of MCNAUGHTON'S algorithm .

Above figure shows that total processing time of all tasks is 60 and then applying MCNAUGHTON'S algorithm[15] by sharing equal processesing times to all processors.

3. RESULTS OF HU'S SCHEDULING ALGORITHM:

Table 2. Details of name of task, processing time.

S.NO	NAME OF THE TASK	PROSESSING TIME
1.	LEG1	1
2.	LEG2	1
3.	LEG3	1
4.	LEG4	1
5.	LEG5	1
6.	LEG6	1
7.	LEG7	1
8.	LEG8	1
9.	LEG9	1
10.	LEG10	1
11.	LEG11	1
12.	LEG12	1

An application of TABLE MANUFACTURING given with set of twelve tasks with names and having same processing time is shown in Table 2. The schedule obtained by using HU'S [17] algorithm shown in below Figure 5. Hu's algorithm is intended to schedule unit length tasks with in-tree precedence constraints.

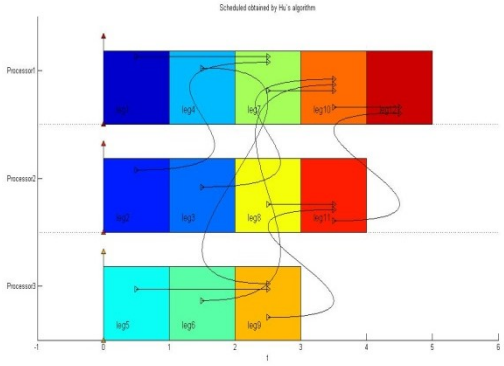

Figure 5. Results of Hu's algorithm

IV. CONCLUSION AND FUTURE ENHANCEMENT

In manufacturing industries, scheduling algorithms plays a major role to release the products into a market within time. By applying List, Hu's and McNaughton's scheduling algorithms for allocating or share the tasks to available active processors. This procedure enhances the performance of the system by reducing the execution time.

In future, novel scheduling algorithms can also be implemented by amalgamate or modify the algorithms as per requirements. Finally, execute the tasks faster and enhance the performance of the system by allocating the tasks to processors within a stipulated time period.

V. REFERENCES

[1]. Brucker, P. (1981). Minimizing maximum lateness in a two-machine unit-time job shop. Computing, 27(4), 367-370. Springer

[2]. P. Sucha, M. Kutil, M.sojika and Z. Hanzlek, "TORSCHE Scheduling toolbox for MATLAB", in 2006 IEEE International Conference on Control Applications, pp. 1181-1186.

[3]. Liu, C. L., & Layland, J. W. (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM (JACM), 20(1), 46-61.

[4]. Ahmed, M. S., Mohamed, A., Khatib, T., Shareef, H., Homod, R. Z., & Ali, J. A. (2017). Real time optimal schedule controller for home energy management system using new binary backtracking search algorithm. Energy and Buildings, 138, 215-227.Elsevier

[5]. Gyung-Leen Park, "Performance evaluation of a list scheduling algorithm in distributed memory multiprocessor systems", Future Generation Computer Systems,(20):249–256, 2004. ACM

[6]. J. C. Palencia and M. G. Harbour, "Schedulability analysis for tasks with static and dynamic offsets.," in Proceedings of the 19th Real-Time Systems Symposium, pp. 26–37, IEEE Computer Society Press, December 1998.

[7]. H. Kasahara, S. Narita, "Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing", IEEE Transactions on computers, vol.33, no.11, pp. 1023-1029, November 1984.

[8]. Kousalya, G., Balakrishnan, P., & Raj, C. P. (2017). Workflow Scheduling Algorithms and Approaches. In Automated Workflow Scheduling in Self-Adaptive Clouds (pp. 65-83). Springer International Publishing.

[9]. Zhou, N., Qi, D., Wang, X., Zheng, Z., & Lin, W. (2017). A list scheduling algorithm for heterogeneous systems based on a critical node cost table and pessimistic cost table. Concurrency

and Computation: Practice and Experience, 29(5). WILEY online library.
[10]. S. O. Memik and F. Fallah, "Accelerated SAT-based Scheduling of Control/Data Flow Graphs.," in ICCD, pp. 395–400, IEEE Computer Society, 2002.
[11]. Cacchiani, V., & Salazar-González, J. J. (2016). Optimal solutions to a real-world integrated airline scheduling problem. Transportation Science, 51(1), 250-268.
[12]. Lee, C. Y. (1991). Parallel machines scheduling with non-simultaneous machine available time. Discrete Applied Mathematics, 30(1), 53-61.Elsevier
[13]. Xu, D., Wan, L., Liu, A., & Yang, D. L. (2015). Single machine total completion time scheduling problem with workload-dependent maintenance duration. Omega, 52, 101-106. Elsevier
[14]. Liu, Y., Li, W., Li, K., Qi, H., Tao, X., & Chen, S. (2016, August). Scheduling Dependent Coflows with Guaranteed Job Completion Time. In Trustcom/BigDataSE/I SPA, 2016 IEEE (pp. 2109-2115). IEEE.
[15]. Zhao, W., Ramamritham, K., & Stankovic, J. A. (1987). Preemptive scheduling under time and resource constraints. IEEE Transactions on computers, 100(8), 949-960.
[16]. Hong, K. S., & Leung, J. T. (1988, December). On-line scheduling of real-time tasks. In Real-Time Systems Symposium, 1988., Proceedings. (pp. 244-250). IEEE.
[17]. McHugh, J. A. (1984). Hu's precedence tree scheduling algorithm: A simple proof. Naval Research Logistics (NRL), 31(3), 409-411. Wiley online library.