Computer Engineering 571: Embedded Operating Systems

Programming Assignment 3: Dynamic Variable Frequency Scheduling Report

Skye Russ (132722795)

Introduction:

Dynamic frequency scheduling uses varying CPU frequencies to achieve lower total power for a set of tasks. It requires all tasks to be known before the scheduler is called to then create a static schedule for periodic tasks. Some considerations that need to be made are that all tasks must still meet their deadlines and the slowdown of CPU frequency must decrease the total power used by any given task instance.

This idea is then combined with the Rate Monotonic (highest priority task is the one with the least computation remaining) and Earliest Deadline First (highest priority task is the one with the soonest deadline) scheduling algorithms to create energy efficient versions of each algorithm.

Implementation:

The provided script “dvfs.c” implements Rate Monotonic (RM) and Earliest Deadline First (EDF) scheduling with additional functionality to find the most energy efficient CPU frequency for the task set. At execution time, a file is provided that includes the number of tasks, number of time steps to schedule, energy consumed at given CPU frequencies (1888Mhz, 918Mhz, 648Mhz, 384Mhz, Idle) in milliwatts, and then each task is described by name, period, and the worst-case execution time at the four CPU frequencies. Once this file is parsed, tasks are stored in a Task Description struct for easy use. The user passes which scheduling algorithm desired as well as if the schedule should be optimized for energy efficiency.

Each scheduling algorithm is defined in its own function with one function defined for energy efficiency. The energy efficient function calls a recursive helper function that iterates through all frequencies for all tasks that provide a decrease in power consumed when compared to maximum CPU frequency. For the example input (detailed in Figure 1), 432 combinations of CPU frequencies across the five tasks were tested. The lowest energy is stored and then the script prints the results of the tasks at the given frequency.

The definition of the RM function and EDF function are identical except for the priority definition algorithm which is used by the scheduler. The function first computes the total CPU utilization for the task set, if this utilization is above 1, the function immediately returns -1. The current active task is stored as an index value on an array of remaining execution time for each task. The function then iterates through timesteps starting at 0 and ending at the specified limit. The loop first checks for any tasks that have missed their deadline or are going to be restarted by taking the modulus of the period for each task compared to 0 (if this value is true, the scheduler also needs to be called so a flag is set because a new task has arrived). If any tasks missed their deadline, the function immediately returns -1 indicating a failed schedule. If no tasks have missed their deadline, the function checks if the current active task has no remaining computation which would also set the scheduling flag to true. If the scheduler is called, it then compares the available tasks (indicated by those tasks having remaining computation required) using either Rate Monotonic or Earliest Deadline First priorities. If no task is selected, the current active task is set to -1 indicating the CPU is idle. After the task is selected, the loop decrements that task's remaining computation by 1 and continues. Once any task finishes its computation or the scheduler is called, the function prints the start time, task name, CPU frequency, end time and power consumed (in Joules) for the given task period.

There is an additional flag for the function that suppresses print statements which allows for efficient repetitive calls to the scheduling functions (which are used by the energy efficient option.

Results:

Below is an example task definition as well as the outputs of the "dvfs.o" script called using RM, Energy Efficient RM, EDF, and Energy Efficient EDF.

Given the following task set, the schedules for Rate Monotonic (maximum CPU frequency), Earliest Deadline First (maximum CPU frequency) and Energy Efficient Rate Monotonic (minimum CPU frequency) and Energy Efficient Earliest Deadline First (minimum CPU frequency) were computed by the script. Below each schedule, the script outputted the total cost and the percentage of the time that CPU spent idle.

Figure 1: Task Set and CPU Frequency Costs (mW)

5	1000	625	447	307	212	84
w1	520	53	66	89	141	
w2	220	40	50	67	114	
w3	500	104	134	184	313	
w4	200	57	74	103	175	
w5	300	35	45	62	104	

Figure 2: Rate Monotonic

START	NAME	FREQ	END	POWER
0	w5	1188	34	21.8750J
35	w2	1188	74	25.0000J
75	w1	1188	127	33.1250J
128	w4	1188	184	35.6250J
185	w3	1188	199	9.3750J
200	w4	1188	256	35.6250J
257	w2	1188	296	25.0000J
297	w3	1188	299	1.8750J
300	w5	1188	334	21.8750J
335	w3	1188	420	53.7500J
421	w4	1188	477	35.6250J
478	w2	1188	517	25.0000J
518	w3	1188	519	1.2500J
520	w1	1188	572	33.1250J
573	w3	1188	599	16.8750J
600	w5	1188	634	21.8750J
635	w4	1188	691	35.6250J
692	w2	1188	731	25.0000J
732	w3	1188	806	46.8750J
807	w4	1188	863	4.7880J
864	IDLE	IDLE	879	10.0000J
880	w2	1188	919	25.0000J
920	w5	1188	954	2.9400J
955	IDLE	IDLE	1000	3.7800J

Total Energy Consumed: 550.8830J
Total Idle time: 137 seconds (0.1370%)

Figure 3: Energy Efficient Rate Monotonic

START	NAME	FREQ	END	POWER
0	w5	1188	34	15.6450J
35	w2	918	84	31.2500J
85	w1	1188	137	33.1250J
138	w4	1188	194	35.6250J
195	w3	1188	199	3.1250J
200	w4	1188	256	25.4790J
257	w2	918	306	31.2500J
307	w5	1188	341	21.8750J
342	w3	1188	440	44.2530J
441	w2	918	490	31.2500J
491	w4	1188	547	35.6250J
548	w1	1188	600	33.1250J
601	w5	1188	635	21.8750J
636	w4	1188	692	25.4790J
693	w2	918	742	31.2500J
743	w3	1188	846	65.0000J
847	w4	1188	903	35.6250J
904	w5	1188	938	15.6450J
939	w2	918	988	4.2000J
989	IDLE	IDLE	1000	0.9240J

Total Energy Consumed: 541.6250J
Total Idle time: 61 seconds (0.0610%)

Figure 4: Earliest Deadline First

START	NAME	FREQ	END	POWER
0	w4	1188	56	35.6250J
57	w2	1188	96	25.0000J
97	w5	1188	131	21.8750J
132	w3	1188	199	42.5000J
200	w4	1188	256	35.6250J
257	w2	1188	296	25.0000J
297	w3	1188	332	22.5000J
333	w1	1188	385	33.1250J
386	w5	1188	399	8.7500J
400	w4	1188	456	35.6250J
457	w5	1188	477	13.1250J
478	w2	1188	517	25.0000J
518	w3	1188	599	51.2500J
600	w4	1188	656	35.6250J
657	w5	1188	659	1.8750J
660	w2	1188	699	25.0000J
700	w5	1188	731	20.0000J
732	w3	1188	753	13.7500J
754	w1	1188	799	28.7500J
800	w4	1188	856	35.6250J
857	w1	1188	863	0.5880J
864	IDLE	IDLE	879	10.0000J
880	w2	1188	919	25.0000J
920	w5	1188	954	2.9400J
955	IDLE	IDLE	1000	3.7800J

Total Energy Consumed: 577.9330J
Total Idle time: 87 seconds (0.0870%)

Figure 5: Energy Efficient Earliest Deadline First

START	NAME	FREQ	END	POWER
0	w4	1188	56	25.4790J
57	w2	918	106	22.3500J
107	w5	918	151	28.1250J
152	w3	1188	199	30.0000J
200	w4	1188	256	25.4790J
257	w2	918	306	31.2500J
307	w3	1188	362	25.0320J
363	w1	918	428	41.2500J
429	w4	1188	485	25.4790J
486	w5	918	530	20.1150J
531	w2	918	580	31.2500J
581	w3	1188	599	11.8750J
600	w4	1188	656	25.4790J
657	w5	918	659	1.3410J
660	w2	918	709	22.3500J
710	w5	918	751	26.2500J
752	w3	1188	836	53.1250J
837	w4	1188	893	25.4790J
894	w1	918	959	29.5020J
960	w2	918	1000	17.8800J

Total Energy Consumed: 519.0900J
Total Idle time: 0 seconds (0.0000%)

As noted, the highest CPU frequency Earliest Deadline First algorithm uses the most power (577.93 Joules), but has the second highest amount of Idle time. This is followed by Rate Monotonic and Energy Efficient Rate Monotonic (0.061% idle time) with the middle amounts of energy consumed and idle time. The optimized energy efficient Earliest Deadline First algorithm has 0 idle time (100% CPU utilization) and uses the least amount of power (519.09 Joules). It is notable that both energy efficient algorithms for EDF and RM result in less idle time for the CPU as opposed to the maximum CPU frequency versions.

The calculated energy efficient schedules are the most optimal energy efficient schedules because all frequencies that resulted in energy savings were tested for all tasks and the least energy consumed set was displayed.