

Implementation of a Frequency Scaling Algorithm for the iPAQ H3975

After speaking with Dr. Mueller, we would like to put forward several changes to our original proposal. Originally, we proposed to develop a device driver under Windows CE for supporting frequency scaling. We have determined that this approach would require tools that are inaccessible by our group (and not free!). The tool in particular is Platform Builder. This is a development tool used to write device drivers for Windows CE, it is similar to DDK for Microsoft's non-embedded operating systems.

Dr. Mueller spoke with our group about simulating a scheduler on top of the OS. We propose to simulate an EDF scheduler modified to support DFS. This simulation will be similar to the schedulers of our homework in most aspects. The main difference is that we will simulate the execution of each task. The tasks running on the scheduler will be dummy threads. We propose to use a feedback driven EDF scheduling algorithm to incorporate frequency scaling. The idea behind the algorithm is to utilize available slack time to scale the frequency and hence voltage. The scheduling algorithm is described in the paper, "Energy-Conserving Feedback Scheduling for Embedded Systems with Real-Time Constraints" by A. Dudani, F. Mueller, and Y Zhu.

Issues Solved:

- Prakash: Research and choose available DFS algorithms
- Mark: Research device driver development
- All: Research tools needed for advancement of project
- Mark: Interface to Windows API for battery status.
- Ben: Implement round robin scheduler for thread execution
 - This was performed so that we have a framework for scheduling and executing threads.

Current/Future Issues:

- Mark: Research into scaling processor voltage via assembly programming
 - He will be looking into generating separate .asm file and compile it using the included assembler armasm.exe then attempting to link this object code into the scheduler application
 - Week 1 - **<Major Milestone>** - *major impact on project completion*
 - Successfully link assembly code to scale frequency

- Prakash: Implementation of DFS scheduling algorithm
 - Week 1 - 2
 - Port existing EDF framework, if possible from HW 2, for EDF/DFS scheduler
 - Week 3 (All)
 - Verify/Modify scheduler
- Ben: Completion of thread support
 - Week 1
 - Debug problem with Sleep() function
 - Begin working with Prakash on scheduler implementation
- All:
 - Experiments
 - More Experiments :)

Major Issues:

At the present, we have run into several major hurdles. The API of Windows CE is very limited in terms of support for power management. As a result, we have begun looking into accessing the voltage/frequency scaling mechanisms, of the X-scale processor, via assembly code. However, we have not yet determined whether these registers are protected. This is one area on which we are now heavily concentrating our efforts. Another issue is the difference between the Pocket PC and Pocket PC 2002 SDK's. We have written a simple thread scheduler that does not behave as expected on the new SDK or the device. When using the Sleep() command, there is no change in system time if the specified amount of time is less than ~1 sec. We are currently tracing the source of the problem so that we may have fine timing granularity (finer than 1 sec anyway).