CSE 190G Spring 2016: Project

The goal of this project is to develop an energy-efficient EDF(Earliest Deadline First) scheduler, which handles multiple workloads controlling sensors running on Raspberry PI 2 (RPi2). During the course of the project, you will develop a program which characterizes workloads using performance counters (part 1), and an energy-efficient user-level earliest deadline first scheduler program which manages the sensors (part 2). For more detailed information see the instructions document at the class website.

Project Part 1 Assignment

During the first part of the project, you will setup the development environment for Raspberry PI 2 (RPi2), and then measure performance of sample code running on RPi2.

Complete the following steps:

1. Cross-compile/install the provided Linux kernel and the sample kernel module
   - Linux kernel source code is available via git: http://seelab.ucsd.edu/~shepherd/cse190g_rpikernel.git
   - sample kernel code is in "sample_module" directory
2. Implement a kernel module to access the performance measurement unit (PMU) using the provided skeleton code
   - Skeleton code is in "pmuon" directory.
   - Kernel module should monitor L1 accesses, L1 misses, L2 accesses, and L2 misses.
3. Study how the provided workload execution time changes over various frequency settings and data sizes and access patterns. Modify the provided skeleton code to capture performance counter readings for the workload (.../memmeasurement/memmeasurement.c) across various program array sizes and CPU frequencies.
4. Write a report that provides detailed performance analysis using the implemented kernel module (pmuon) and the user-space program (memmeasurement)
   - Analyze the program performance over different workload sizes and CPU frequencies.
   - Provide at least the following plots for min and max frequency of operation, and for array sizes from 1K to 4MB:
     i. execution time vs. array sizes
     ii. L1 and L2 cache miss number/rate vs. array sizes
     iii. execution time vs. cycle count
     iv. execution time vs. L1 and L2 cache miss numbers/rates
     v. CPU energy vs. CPU frequency for array sizes
   - Provide a plot of execution time vs. CPU freq. for array sizes of 8KB, 128KB, and 1MB

Submission:

1. Demo on April 27 (in cse 3219) your Raspberry Pi 2 that uses a single core with "Medium" overclocking setting running correctly compiled sample kernel module, "hello.ko". Demonstrate successful execution of "check_kernel", provided on the class website.
2. Submit these three files into TED (Due: 23:59:59 PST, May 4)
   a. A kernel module source code for the event counter selection, "pmuon.c"
   b. A user-space program which measures the performance counters, "memmeasurement.c"
   c. A report on the performance analysis including the discussion of required plots (PDF)