Operating System Concepts in Embedded Computing

Wayne Wolf, Chang Hong Lin, Ahmed Abdalla
Department of Electrical Engineering
Princeton University
{wolf,chlin,aabdalla}@princeton.edu

Abstract

Operating systems play an increasing role in embedded computing systems, thanks to advanced applications. The real-time operating systems used in many embedded systems provide real-time scheduling, make efficient use of memory, and often operate at low power levels. This paper describes our approach to teaching operating system concepts in an embedded computing course, including some important aspects of embedded operating systems as well as lectures and labs we developed using Windows CE as an example embedded operating system.

1 Introduction

Embedded computing systems must often perform multiple complex tasks that require the mediation of an operating system. Operating systems are complex objects that programmers and system designers rely on to perform many functions. When designing embedded systems, the operating system plays a somewhat different role and system designers expect their operating systems to have particular attributes that may not be relevant to general-purpose programmers. We are interested in the function of the operating system, its internal mechanisms, the resulting performance, and power consumption of the operating system.

Programmers on general-purpose platforms tend to view the operating system as a set of services for application developers. Embedded system designers tend to view their operating systems as components with particular properties. Because embedded computing systems increasingly rely on real-time operating systems (RTOSs), students interested in embedded computing should have a basic understanding of embedded operating system concepts. Because the requirements and design of RTOSs is substantially different from that of general-purpose operating systems, traditional operating systems classes typically do not cover some material that is of great importance to embedded system designers.

This paper describes our approach to teaching operating system concepts in an embedded computing course. This material is embodied in ELE 464, Embedded Computing, at Princeton University. We are also using this experience to revise the text *Computers as Components*[1]. After briefly reviewing some high-performance embedded computing platforms, we will go on to describe the characteristics of embedded operating systems. We will then describe our experimental setup to measure power consumption of embedded systems.

2 High-Performance Embedded Systems

A full discussion of platforms for high-performance embedded computing is beyond the scope of this paper, but we think it is important to keep in mind that advanced applications impel embedded system designers to the use of advanced operating systems. High-volume markets such as communications, automotive, and multimedia all require large amounts of computation that is often provided by parallel processors. The multiple tasks that run on the platform are generally mediated by a real-time operating system.

Multiprocessor Systems-on-Chips[2] provides a more detailed survey of modern systems-on-chips for embedded computing. We briefly review some examples.

- The Philips Nexperia platform for high-definition television.C This chip includes two processors: a Trimedia 5-issue VLIW CPU and a MIPS RISC CPU. A real-time operating system schedules tasks on the processors, manages interprocess communication, *etc*.
- The TI OMAP family of processors is designed for mobile multimedia, such as image-enabled cell phones. An OMAP chip includes a TI DSP and an ARM RISC CPU that communicate via shared memory. Each processor runs an operating system that manages resources, interprocess communication, *etc*.
- The ST Microelectronics Nomadik system-on-chip is also designed for mobile multimedia. It uses an ARM as a host processor. It also includes processors for audio and video operations; these processors use the MMDSP+ architecture as well as special-purpose units.
- The Freescale MPC5200 is a system-on-chip for automotive applications. It uses a PowerPC 603e core. It also includes a CAN interface, Ethernet. and on-board floating point.

3 Scheduling and Performance

Traditional operating systems courses do not talk about many of the topics of importance to embedded computing systems. Similarly, general-purpose operating systems are not often used for embedded computing. Even if students have a background in operating systems (which most do not), they will need to larn new concepts that apply directly to embedded computing.

Scheduling is an important aspect of real-time operating systems, but scheduling embedded system tasks is a very different problem than scheduling tasks in a general-purpose system. embedded computing systems must generally perform to meet deadlines. Unlike general-purpose operating systems, real-time operating systems must arrange computations in time to ensure that the applications meet their deadlines. A number of theoretical results are available to help explain real-time scheduling and to serve as the basis of real-time operating systems. This theory helps explain and rectify important effects such as priority inversion.

System performance goes beyond scheduling theory, however. The way in which interrupts are handled is critical to both the responsiveness and schedulability of the system. A basic understanding of the sequence of steps in which requests are handled helps students understand the performance limits of embedded operating systems. Cache effects are also very important in high-performance embedded systems. Those effects can be modeled by analysis and tools.

Many of the same mechanisms that affect performance also influence power consumption. Power and energy are very important aspects of embedded system design and the execution of the operating system can consume a significant fraction of total system energy.

Memory footprint is another important characteristic of embedded operating systems. The amount of memory required for the operating system helps to determine the system hardware cost. General-purpose operating systems tend to be very large. In the embedded world, operating systems range from a few hundred bytes to large systems that rival general-purpose operating systems.

4 Windows CE

Grounding the discussion and experiments in real operating systems helps students apprehend theoretical concepts and the assumptions underlying theory. We have built a series of lectures and labs around the Windows CE operating system. Windows CE is widely used. It is also well-documented so that its internals can be explained; the book by Boling[3], for example, provides an excellent introduction to the operating system. All the materials described in this section can be obtained on the Web at http://www.princeton.edu/~wolf/embedded-book/chapter-aids/WinCE/wince-curriculum.htm.

We developed a set of lectures on the operation of Windows CE. The first lecture summarizes the principles of operation of Windows CE: memory organization, the scheduler, process structure, etc. The second lecture concentrates on files and processes, including the file system and the details of process structure.

We also developed a lab that allows students to measure power consumption of a running system. This lab shows them how to perform a measurement that is commonly used in industry to evaluate power consumption. It also lets them observe that power consumption changes significantly as system activity changes.

The power measurement simply measures current going into the processor, typically using a measurement resistor and measuring voltage. Some development boards provide taps that allow CPU current to be measured independently of the current drawn by other components.

We first tried to use the Advantech WinCE development board for our power measurements. This board is designed for application development and provides easy access to the major system components. However, we found that poor documentation and a poor development environment made this a poor choice for classroom ues.

Our second attempt at measuring power consumption was based on an iPAQ PDA. Figure 1 shows the interior of a PDA that we disassembled for our experiments. The PDA was a much better platform for this type of experiment. Although it is a production unit, it is fairly easy to disas-

FIGURE 1. The interior of an iPAQ PDA.

semble the device to the extent necessary to measure power consumption. We found it relatively easy to take apart the PDA, locate the power trace, and install the measurement resistor. The PDA is also considerably cheaper than the Advantech board.

Figure 2 shows the results of power measurements on both the iPAQ and Advantech board. We measured power consumption on a number of system tasks. The measurements show that power consumption can vary significantly based on system activity.

5 Conclusions

Operating systems play an increasing role in embedded computing systems. Embedded operating systems are judged on their ability to schedule real-time tasks, memory footprint, and power consumption, among other attributes. We have developed material that helps to explain basic operating system concepts for students of embedded computing systems. This material includes both lectures and labs that are available on the World Wide Web.

Acknowledgments

This work was made possible by a generous grant from Microsoft Corporation.

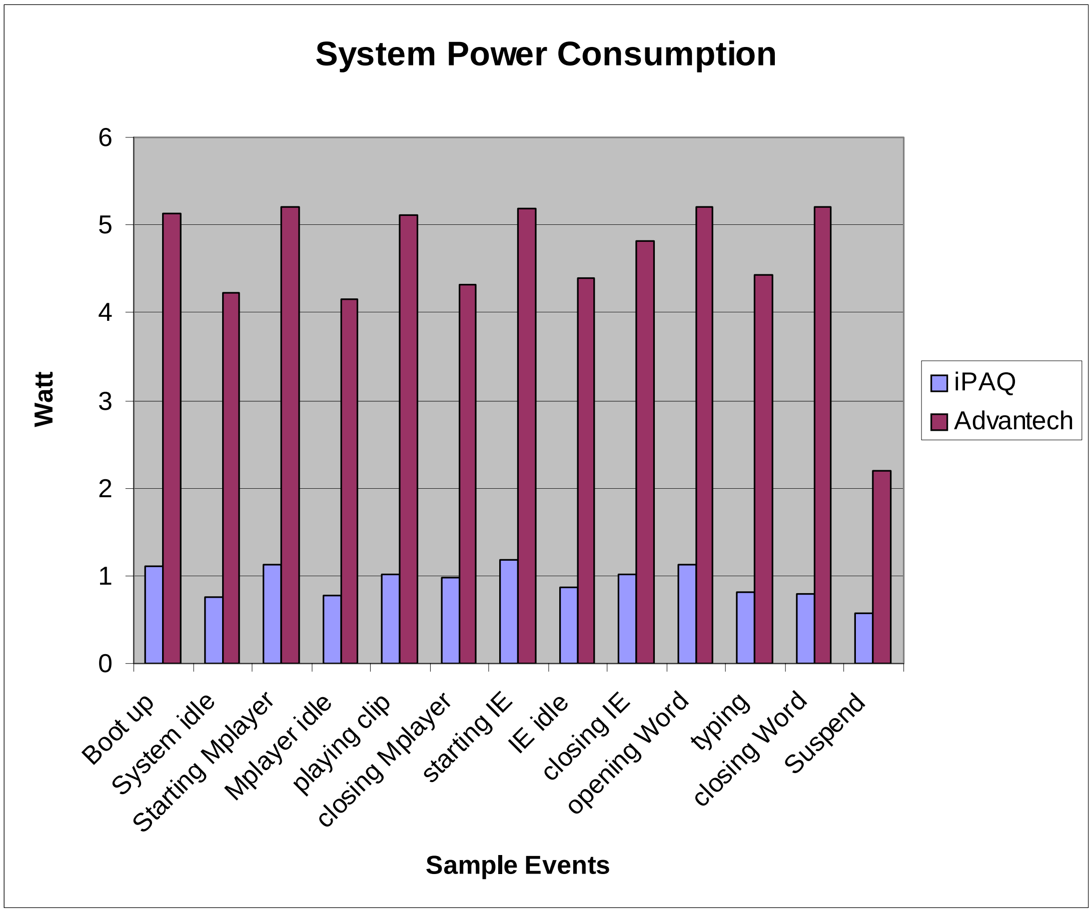

FIGURE 2. Measurements of system power consumption.

References

[1] Wayne Wolf, *Computers as Components: Principles of Embedded System Design*, Morgan Kaufman, 2000.

[2] Ahmed Amine Jerraya and Wayne Wolf, eds., *Multiprocessor Systems-on-Chips*, Morgan Kaufman, 2004.

[3] Douglas Boling, *Programming Microsoft Windows CE .NET*, third edition, Microsoft Press, 2003.

Biography

Wayne Wolf is Professor of Electrical Engineering at Princeton University. Before joining Princeton, he was with AT&T Bell Laboratories, Murray Hill NJ. He received all three degrees in electrical engineering from Stanford University. He is a Fellow of the IEEE and ACM and a member of ASEE and SPIE. He received the 2003 ASEE/HP Frederick E. Terman Award.

Chang Hong Lin is a graduate student in the Department of Electrical Engineering at Princeton University. His research interests include VLSI systems, embedded computing, and distributed video.

Ahmed Abdalla is a graduate student in the Department of Electrical Engineering at Princeton University. His research interests include digital audio and sensor networks.