Design and Implementation of a Linux-based Smartphone

Dong-Yun Shin and Sung-Soo Lim

Abstract—**As the features of mobile phones are being enhanced and the performance is being greatly improved, the generic operating systems are being incrementally used in mobile phones. However, since the mobile phones have different requirements in both features and performance compared with desktop or server systems, the operating systems for generic systems could not be used without customization and modification. This case study paper describes the design and implementation of a Linux-based smartphone hardware and software. The enhancements and additional implementations in both kernel and middleware layers are explained. As an experiment, the power consumption of the implemented mobile phone under a number of user scenarios has been measured.**

Index Terms—**Smartphone, Linux, dynamic power management.**

I. INTRODUCTION

The portion of high-performance, multi-functional smartphones are being dramatically increased in wireless phone market [1]. As the smartphone occupies more portion in the market, the software architecture has been advanced accommodating desktop software layers based on generic operating systems.

When using the generic software architecture for wireless handheld devices, each component included in the software architecture needs to be modified and customized to be well adapted to battery-powered systems. The most important components that need to be carefully customized in the software architecture for smartphones include power management part and the wireless communication support layer.

In this paper, we describe the design and implementation of a smartphone prototype built for various research and development purposes. We explain the hardware design which contains the most of the key features to be used in next generation data centric wireless services and the software architecture that is fully based on Linux kernel and open source software. The software architecture contains the key components including kernel-level power management module and the communication support middleware for CDMA services.

Our prototype design of smartphone platform has been verified using a number of key applications in smartphones including voice call and multimedia decoding. As an experiment for such verification, we measured the power consumption for a number of scenarios when the key applications are running. The power consumption results are shown comparing the cases where the power minimization feature is enabled with the cases where power management feature is not used.

The rest of the paper is organized as follows: Section 2 describes the hardware design of our smartphone prototype and Section 3 explains the software architecture of our platform. Section 4 shows the implementation of the smartphone prototype and experimental results for power consumption measurements. Section 5 concludes the paper.

Manuscript received October 20, 2005; revised December 3, 2005.
School of Computer Science Kookmin University Seoul, Korea
E-mail : elsdy@kookmin.ac.kr

II. SMARTPHONE HARDWARE ARCHITECTURE

Figure 1 shows the structure of our smartphone hardware. The smartphone hardware consists of two different processors, an application processor (PXA255 from Intel) [4] and a co-processor (MQ1188 from nVidia) [5] for multimedia acceleration processing. The reasons why the additional multimedia co-processor is used are 1) to reduce the traffic in the system bus by separating the data traffic between LCD and framebuffer and 2) to utilize the hardware acceleration feature of the additional multimedia co-processor for multimedia performance improvement. The hardware organization used in our smartphone prototype is widely used in high-end multimedia smartphones whose performance and diverse applications are key features.

The CDMA module is connected to the main processor through UART interface. The CDMA module performs modem operations controlled by AT commands from the main processor. This hardware organization has been widely used for high-end multimedia communication devices in the commercial market.

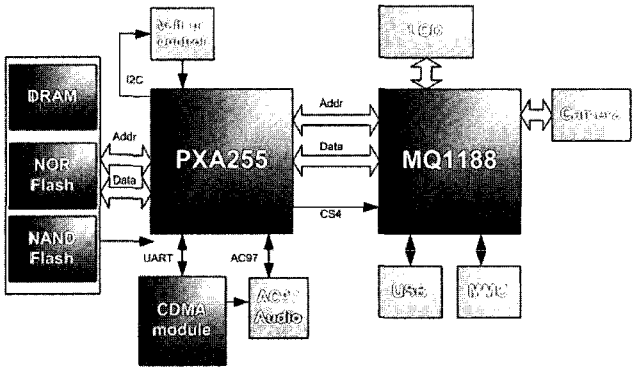

Fig. 1. The structure of smartphone hardware.

III. LINUX-BASED SMARTPHONE SOFTWARE ARCHITECTURE

Our Linux-based software architecture implements all the key features required in the battery-powered communication devices including power management features and communication support.

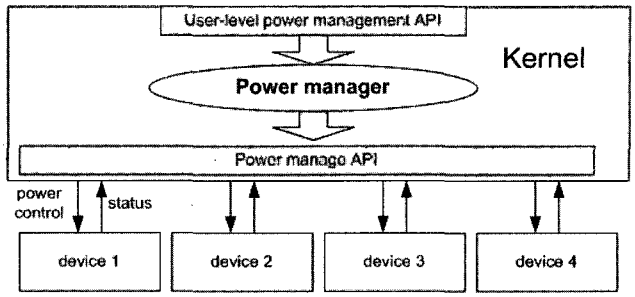

Fig. 2. Kernel-level power management module.

Figure 2 shows the power management module implemented in the Linux kernel running on our smartphone hardware. As shown in the Figure 2, two different APIs are implemented around the power management core component: the user-level power manamgent APIs are used by the applications to get the status of the devices whose power consumption modes can be controlled and the device driver-level power management APIs are used by device drivers so that such devices can be controlled by the core power manager. The power manager in the kernel determines the appropriate power consumption mode of each device based on the usage status of each device and the current system load.

The power manager implemented in our smartphone platform software contains both dynamic power management (DPM) of devices and dynamic CPU voltage/frequency scaling features (DVS) [2] [6] [7] [8]. The DPM controls the power consumption modes of the devices attached to the system: most of the devices used in handheld devices have various power consumption modes such as run mode and sleep mode. The role of the power manager in the kernel is to determine which power consumption mode is appropriate for each device. The current implementation of the power manager is basically an infrastructure in that all the APIs and data structures to be used by power management algorithms are implemented.

The DVS is also implemented in the way that appropriate CPU frequency and voltage pair among the discrete voltage/frequency scaling steps of PXA255 ranging from 98 Mhz to 400 Mhz of CPU frequencies along with 1.0 V through 1.3 V of CPU operation voltages could be selected by the power management core.

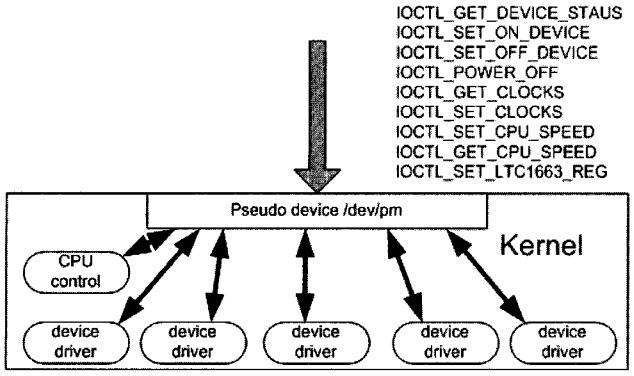

Fig. 3. Pseudo device for power management control.

Figure 3 shows the implementation details of the power management part in our Linux kernel. A pseudo device called /dev/pm is implemented thorugh which all the power management control commands are transfered to both power management core in the kernel and device drivers. Both the DPM and DVS can be controlled from the user-level through the IOCTL commands implemented in the pseudo device driver.

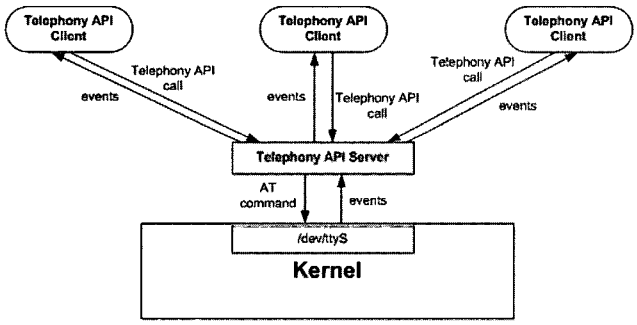

Fig. 4. elephony API.

Key applications of smartphone include communication applications such as voice call, messaging, and data service applications. Though more and more data centric applications are being increasingly used in mobile phones, the voice call and messaging applications are the most important applications. Therefore, the software layer to seamlessly support the communication applications should be prepared.

We designed and implemented a communication application support software layer called telephony API (TAPI) providing all the necessary APIs to be used by voice call, messaging, and data service applications. Since the CDMA module is controlled by AT command set, the TAPI implemented in our platform traslates the API calls from communication applications using TAPI to the appropriate AT commands to be interpreted by the CDMA module.

Our TAPI is based on a client-server model in that all the communication applications talk to TAPI server and the TAPI server is the only channel to control the CDMA module and transmit/receive the CDMA data. TAPI server provides the transparency for both transmission and reception of CDMA data. In addition, the TAPI server manages all the exceptional scenarios in CDMA communications such as messaging call arrival during voice communication. Figure 4 shows the structure of our TAPI implementation.

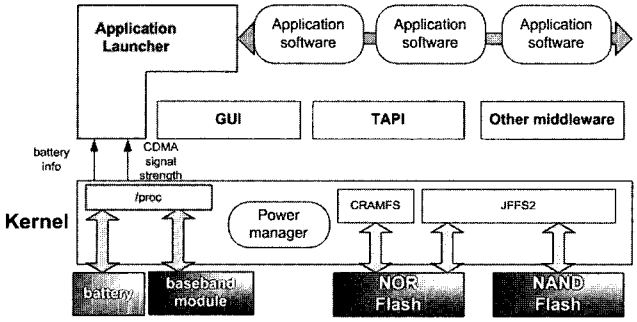

Fig. 5. Linux-based smartphone software layers.

Figure 5 shows the overall software layers for our Linux-based smartphone platform. The application launcher is an application manager performing both window management role and the overall application control. The roles of the application launcher include collecting the various system information such as battery status and CDMA signal strength and displaying the information at user screen. We have used the combination of CRAMFS and JFFS2 [3] flash memory file systems in order to get the best performance in file transactions in flash memory

IV. Implementation and Experimental Results

As a prototype of the Linux smartphone platform, we implemented the hardware based on the design described

in this paper and ported the Linux-based smartphone software layers on the hardware; Figure 6 shows the hardware. In order to verify the operations of the hardware and smartphone software, we have executed a number of key smartphone applications including CDMA voice call, data service call, and multimedia decoding. In addition, to test whether the power management feature of the smartphone software is in operation, the power consumptions of the smartphone hardware are measured when the power minimization technique is applied. The measurement results of the power consumptions for various scenarios are shown in Figure 7 and Figure 8.

Figure 7 shows the measurement results for power consumption when the CDMA voice call communitaion is in operation. Figure 7 (a) is the power consumption results when the smartphone is operating with the maximum CPU frequency and voltages and all the devices running while Figure 7 (b) is the power consumption when the dynamic power management is applied so that the CPU voltage/frequency is minimized with all the unnecessary devices put to sleep modes. As shown in the figures, the CPU power consumption of Figure 7 (b) is drastically reduced.

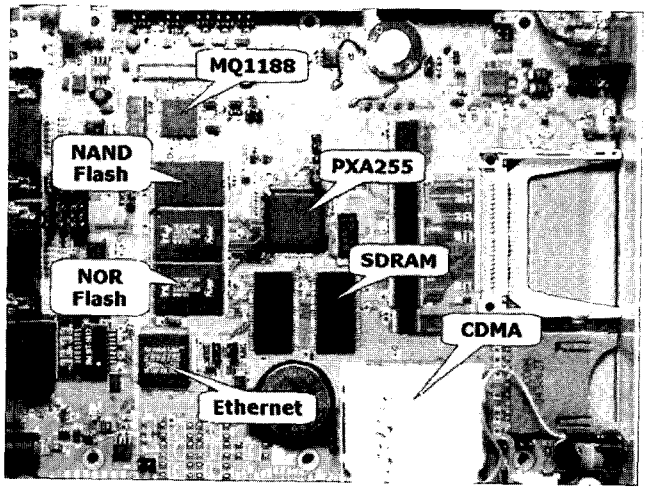

Fig. 6. Smartphone hardware prototype.

Figure 8 shows the power consumption results when a multimedia decoding application is running. Similarly to Figure 7, Figure 8 (a) is the power consumption results when the system is running at the maximum power consumption status while Figure 8 (b) is the power consumption results when the dynamic power management is applied so that the CPU power is minimized and unnecessary devices are turned off. The CPU power consumption in Figure 8 (b) is reduced significantly and thus the total power consumption is accordingly reduced. Our experimental results show that the total power consumption is reduced up to 21% when we apply dynamic power management techniques. We can note that the total power consumption of the platform is not significantly reduced compared to the power consumption reduction of CPU. This is due to the effect of various current leakage of the hardware which is not at the commercialization stage. The hardware could be shrunk down to handful size with the same organization and then the power leakage could also be minimized.

V. Conclusions

In this paper, the hardware and software architecture of our Linux-based smartphone platform is described. The hardware has a two processor organization with a main application processor and a multimedia co-processor which has been widely used in the market. The software architecture includes necessary modification and enhancement in both Linux kernel and application-level middleware. Our experimental results show that the overall smartphone platform is operational and especially the power management feature implemented in the kernel is effective.

Our objective of designing the smartphone platform described in this paper is to perform diverse research for both system level power/performance optimization for wireless handheld devices and novel middleware implementation for next generation ubiquitous handheld devices.

VI. Acknowledgments

This work was supported by research program 2004 of Kookmin University in Korea.

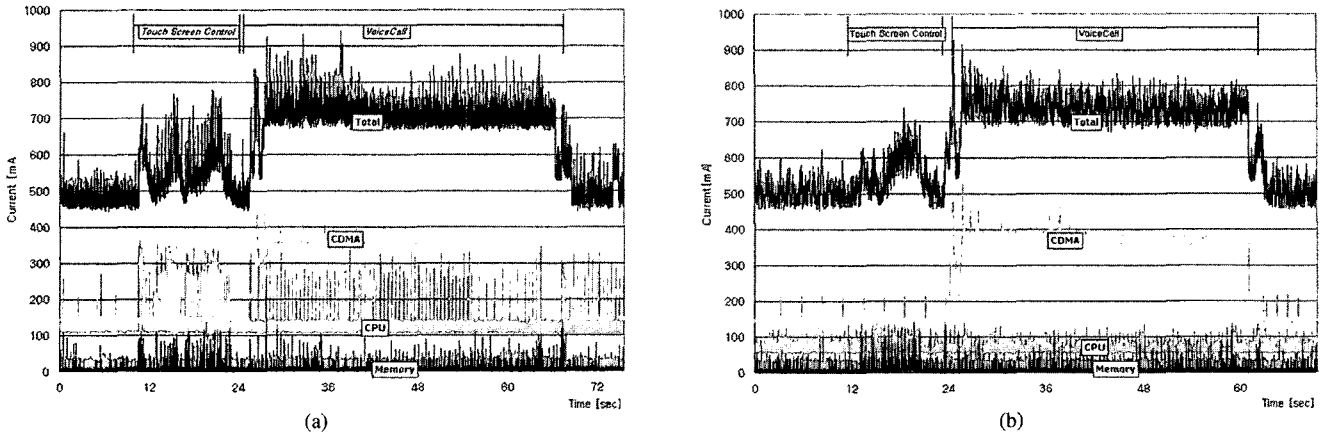

Fig. 7. Power consumption for CDMA voice call application.

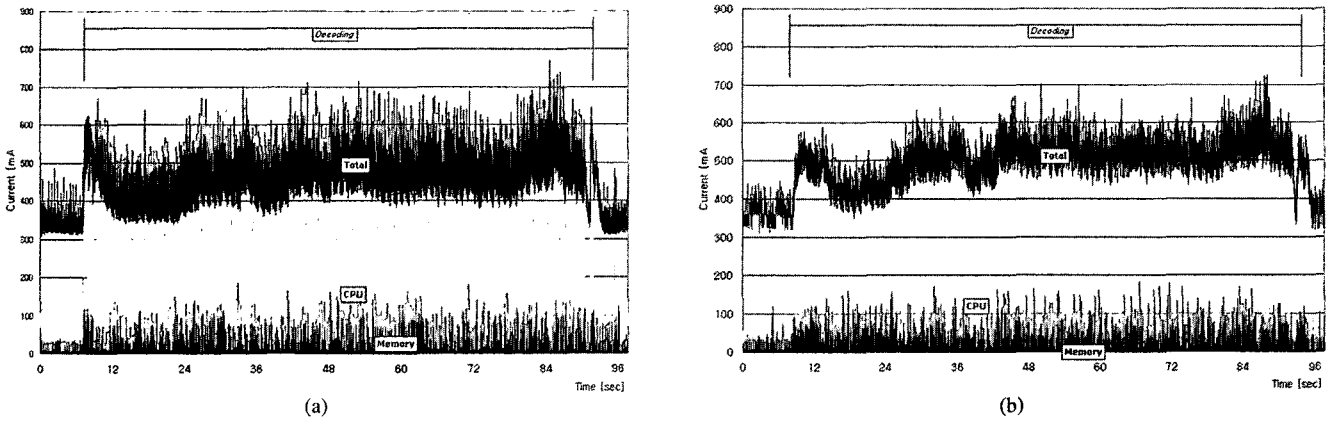

Fig. 8. Power consumption for multimedia decoding application.

REFERENCES

[1] Liviu Iftode, Cristian Borcea, Nishkam Ravi, Porlin Kang, Peng Zhou, "Smart Phone: An Embedded System for Universal Interactions", *10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)*, pp 88-94, May 2004.

[2] K. Choi, R. Soma, and M. Pedram, "Off-chip latency-driven dynamic voltage and frequency scaling for an MPEG decoding," *Proc. of 41st Design Automation Conference*, San Diego, CA, Jun. 2004, pp.544-549.

[3] D. Woodhouse, "JFFS: The Journaling Flash File System," *Ottawa Linux Symposium 2001*, 2001.

[4] Intel, PXA255 Application Processors Developer°Øs Manual, http://www.intel.com/design/pca/applicationsprocessors/manuals/278693.htm.

[5] nVidia, MQ1188 Product Brief, http://www.nvidia.com/page/pg_20031126985559.html.

[6] Tajana Simunic, Luca Benini, Peter Glynn, Giovanni De Micheli, "Dynamic power management for portable systems", *Proceedings of the 6th annual international conference on Mobile computing and networking*, August 2000.

[7] Tajana Simunic, Luca Benini, Andrea Acquaviva, Peter Glynn, Giovanni De Micheli, "Dynamic voltage scaling and power management for portable systems", *Proceedings of the 38th conference on Design automation conference*, Pages: 524 - 529, June 2001.

[8] Kihwan Choi, Ramakrishna Soma, Massoud Pedram, "Power supply, voltage, and frequency management: Dynamic voltage and frequency scaling based on workload decomposition", *Proceedings of the 2004 international symposium on Low power electronics and design*, August 2004.

Dong-Yun Shin He is a researcher on an MS degree in the embedded system laboratory in the school of computer science in the Kookmin University in Korea. His research interests include embedded system, mobile communication device, computer architectures, real time system. He received double BS degrees in computer science and information communication from Kookmin University.

Sung-Soo Lim He is a full-time lecturer in School of Computer Science, Kookmin University, Korea. His research interest includes embedded mobile systems, real-time systems, and computer architecture. He received BS, MS, and PhD degrees in Computer Science and Engineering from Seoul National University. He was the Chief Technical Officer in PalmPalm Technology, Inc. in Korea from 1999 to 2004 and a visiting researcher in real-time system laboratory, University of Illinois at Urbana-Champaign in USA from 2000 to 2001.