PDA Operating Systems

Aarne Klemetti
EVTEK, Media Technology

Aarne.Klemetti@evtek.fi

Abstract

This paper describes the operating systems – OS – for personal digital assistants – PDAs. Three systems, namely Windows CE, Palm OS, and Symbian OS are discussed. At first the general features of PDA operating systems are introduced. Then the differences of the three systems are discussed from the functionality, usability, and scalability of each system. After that some future predictions are given and analyzed.

1 INTRODUCTION

This paper presents three of the most prominent operating systems (OSs) for personal digital assistants (PDAs). The OSs discussed here are:

1. Windows CE, which is developed by Microsoft, Inc. and it is intended to be a general purpose operating system for wireless and personal systems.

2. Palm OS, which was originally developed for the handheld devices produced by Palm Computing, Inc. It is now adopted by several important manufacturers of the industry.

3. Symbian OS, which is an open operating system developed by a consortium of the major developers for portable and mobile devices.

The purpose of this paper is to make a distinction between the systems and their goals and especially to analyze the functionality and the future of each PDA OS.

2 BACKGROUND

Small devices come in many shapes and sizes, each addressing different target markets and having different requirements. The market segment of the scope of this paper is that of the wireless information device (WID). This spans the spectrum between traditional PDAs (with handheld computer functionality and occasional communication) and mobile phones (with voice communication and enough data capability to support an address book and basic messaging).

It is important to look at the WID market on its own. It has some specific needs that make it unlike most other markets such as PCs or fixed domestic appliances. Scaling down a PC operating system or bolting communication on to a small and basic OS results in too many fundamental compromises. Symbian believes that the WID market has five key characteristics at its core that make it unique, and require a specifically designed operating system:

- devices are both small and mobile;
- the target is a mass market of consumer, enterprise and professional users;
- devices are occasionally connected: they can be used both when connected to the wireless phone network or locally to other devices, and also when not connected to any network;
- manufacturers need to differentiate their products in order to innovate and compete in a fast-evolving market;
- and last but not least, the platform has to be open for third-party application development.

To address each of these characteristics is the only way to create good products where technology doesn't limit functionality and creating good products is the way to grow the WID market. (www.devx.com/wireless/articles/PDA/PDAIntro.asp(20.11.2001))

3 OVERVIEW OF THE PDA OPERATING SYSTEMS

For the PDA OS there are several viewpoints that need to be addressed. Those include:

- fast bootstrap processes
- immediate response to switching from one application to another
- efficient back-up/synchronization operations

The requirements are best met by using real-time operating systems (RTOS), which provide the means for fast and efficient responses needed for running the PDAs.

In general, real-time operating systems are said to require:

- Multitasking
- Process threads that can be prioritized
- A sufficient number of interrupt levels

Real-time operating systems are often required in small embedded operating systems that are packaged as part of microdevices. Some kernels can be considered to meet the requirements of a real-time operating system. However, since other components, such

as device drivers, are also usually needed for a particular solution, a real-time operating system is usually larger than just the kernel. (www.whatis.com (19.11.2001))

It is important to distinguish between a real-time system and a real-time operating system (RTOS). The real-time system consists of all elements—the hardware, operating system, and applications—that are needed to meet the system requirements. The RTOS is just one element of the complete real-time system and must provide sufficient functionality to enable the overall real-time system to meet its requirements.

3.1 Windows CE

Although the Windows CE Operating System is the leader among handheld personal computers (H/PCs), it holds only about 10% of the PDA market. Those numbers won't stay that low.

The latest version of Windows CE, now known as Windows CE for the PocketPC (P/PC), is coming back with a vengeance. Microsoft has partnerships with some key companies in the industry (Casio, HP, and Compaq) – all manufactures of PocketPC devices.

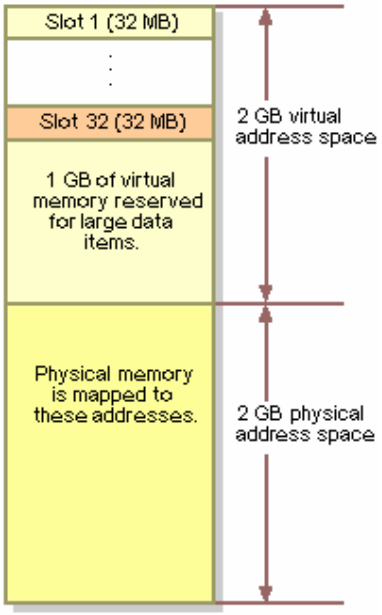

Figure 1. How memory is allocated in the Windows CE address space. (www.microsoft.com/windows/embedded (20.11.2001))

Microsoft Windows CE is an open, scalable, 32-bit operating system that is designed to meet the needs of a broad range of intelligent devices, from enterprise tools such as industrial controllers, communications hubs, and point-of-sale terminals to consumer products such as cameras, telephones, and home entertainment devices. A typical Windows CE-based embedded system is targeted for a specific use, often runs disconnected from other computers, and requires an operating system that has a small footprint and a built-in deterministic response to interrupts.

Real-time performance is essential for the time-critical responses required in high-performance embedded applications such as telecommunications switching equipment, industrial automation and control systems, medical monitoring equipment, and space navigation and guidance systems. Such applications must deliver their responses within specified time parameters in real-time.

Although previous versions of Windows CE offered some RTOS capabilities, a number of significant changes made to the kernel of Windows CE 3.0 have greatly enhanced real-time performance. Some of the changes made for Windows CE 3.0:

- Increased number of thread priority levels from 8 to 256.
- More control over times and scheduling. Applications can control the amount of time provided to each thread and manipulate the scheduler to their advantage. Timer accuracy is now one millisecond for Sleep- and Wait-related APIs.
- Improved method for handling priority inversion.
- Full support for nested interrupts.
- Reduced ISR and interrupt service thread (IST) latencies.
- More granular memory management control.

The kernel is the inner core of the Windows CE operating system and is responsible for scheduling and synchronizing threads, processing exceptions and interrupts, loading applications, and managing virtual memory. In Windows CE 3.0, the kernel has undergone several changes to increase performance and reduce latencies, including:

- Moving all kernel data structures into physical memory (Figure 1), thus largely avoiding translation look-aside buffer (TLB) misses while executing non-preemptible code in the kernel.
- All non-preemptible, but interruptible, portions of the kernel, known as Kcalls, were broken into smaller non-preemptible sections. This introduces some complexity, due to the increased number of sections, but now lets preemption be turned off for shorter periods of time.

3.2 The Palm Operating System

In 1996, a product called the PalmPilot was released by US Robotics. The Palm Pilot ran on an operating system made especially for that device, called the Palm OS.

The Palm Operating System (Palm OS) is the current leader in the PDA market, accounting for 70% of the market share. The Palm Pilot (now known as just Palm), became one of the fastest growing computer platforms in history, reaching the million-sold mark faster than the IBM PC or Apple's Macintosh.

Today, the Palm line has grown to include a variety of models. In addition, a number of other companies such as IBM, Qualcomm, and Symbol Technologies released their own Palm OS PDA models, with Sony's version hitting the market later this year.

Palm OS is the computer operating system that provides a software platform for the PalmPilot series of handheld personal digital assistants made by Palm Computing, now part of 3Com. Palm OS was designed from the beginning to fit into a palm-size device of a specific size and with a specific display size.

Palm OS uses multitasking, but only one task is for applications. The user uses one application at a time, one application program must finish before the next can be selected. This constraint allows the operating system to devote full attention to the application that is open. The space needed by the system for any application that is running is kept in dynamic, reusable random access memory.

The application and its related database are kept in what is called permanent storage, but here the permanent storage is RAM (rather than a hard disk) that cannot be reused as the dynamic RAM can. Palm OS divides an application into runnable code and different types of data elements, such as user interface elements and icons. The data elements can be easily changed without necessarily having to rewrite code.

Palm Computing chose not to include a keyboard in the PalmPilot in order to produce a truly palm-size device. Learning from Apple's Newton, an earlier attempt at a pen-and-notepad interface, the company also chose not to provide full handwriting recognition code. Instead, PalmPilot users learn to use a more quickly recognized but restrictive set of pen strokes. These decisions helped keep Palm OS small in size. Palm OS comes with these applications built-in: Dates, Address Book, To Do List, Memo Pad, Calculator, and Password Protection. New applications can be written and added using several facilities that accelerate development. (www.palmos.com(20.11.2001))

The Palm OS platform consists of five primary components:

- Palm OS software
- Reference hardware design
- HotSync conduit data synchronization technology for one-button synchronization
- Platform component tools including an API that enables developers to write applications
- Software interface capabilities to support hardware add-ons

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.