

OS für kleine Endgeräte:

Symbian OS

Sven Walter

19.07.2004

1. Einleitung

Symbian ist ein Software Unternehmen, das ein offenes Betriebssystem – Symbian OS – für datenfähige Mobiltelefone entwickelt. Es wurde im Juni 1998 von Ericsson, Nokia, Motorola, und Psion gegründet. Die heutigen Eigentümer sind: Ericsson, Panasonic, Nokia, Psion, Samsung, Siemens und Sony Ericsson. Zu den Lizenznehmern gehören alle führenden Mobiltelefonhersteller der Welt.

Symbian OS startete als Betriebssystem für PDAs der Psion Serie (unter anderem 5mx, Revo und netBook). Das erste Mobiltelefon mit Symbian OS war das Ericsson R380 Smartphone. Im ersten Quartal 2004 waren 18 Symbian OS Mobiltelefone von fünf Lizenznehmern erhältlich, und weitere 30 von insgesamt neun Lizenznehmern in der Entwicklung. Weltweit wurden in diesem Zeitraum ca. 2,4 Millionen Mobiltelefone mit Symbian OS ausgeliefert.

Das Design eines Mobiltelefons, anders als zum Beispiel das eines PC, stellt gewisse Anforderungen an ein geeignetes Betriebssystem. Es sollte möglichst klein sein und wenig Speicher in Anspruch nehmen, eine effiziente Energieverwaltung haben, und Echtzeit-Unterstützung für Kommunikations- und Telefonprotokolle bieten.

Die Haupteigenschaften von Symbian OS sind:

- Integrierte multimode Mobiltelefonie - es verbindet die Fähigkeiten der EDV mit der Mobiltelefonie
- Offene Anwendungsumgebung – es ermöglicht den Einsatz von Mobiltelefonen als Plattform für Anwendungen und Dienste in einer Vielzahl von Sprachen und Inhalten.
- Offene Standards und Kompatibilität – Symbian OS stellt mit seiner flexiblen und modularen Implementierung einen Kern von „application programming interfaces“ (APIs) bereit.
- Multitasking - Symbian OS basiert auf einer Micro-Kernel-Architektur und beinhaltet volles Multitasking und Multithreading. Alle Systemdienste wie Telefonie, Netzwerk Middleware und Anwendungssysteme laufen als eigene Prozesse.
- Flexibles Benutzer-Schnittstellen Design – mit Hilfe eines flexiblen „graphical user interface“ (GUI) Designs wird die Portierung von Anwendungen für Dritte erleichtert.
- Robustheit – Symbian OS unterstützt direkten Zugriff auf Benutzerdaten. Es gewährleistet Datenintegrität, selbst unter unsicheren Kommunikationsbedingungen und Mangel an Speicher und Strom.

2. Hardware Anforderungen

Prozessor

Symbian OS benötigt einen 32bit, Little-Endian Mikroprozessor der ARM-Architektur mit einer MMU und Cache. Der Befehlssatz muss ARM V4 oder später sein. Außerdem muss die CPU in verschiedenen normalen und privilegierten Modi arbeiten, und Interrupts und Exceptions verarbeiten.

Die MMU muss physischen auf virtuellen Speicher abbilden können, einen Translation Look-aside Buffers (TLB) haben und sowohl kleine, 4 KB, also auch große, 1MB, Seiten unterstützen.

Alle auf Symbian OS basierenden Produkte laufen zufrieden stellend bei einer Taktrate von 18 MHz. Smartphones sollten aber mit 50 MHz oder mehr laufen, Communicators, wie zum Beispiel der gleichnamige von Nokia, sogar mit mehr als 100 MHz.

Speicher

Typisch für ein System sind 8-16 Mbyte RAM und eine ähnlich große Menge ROM, mit einem 16 oder 32 Bit breitem Speicherbus

3. Hardware Architektur

Die Hardware von Mobiltelefonen kann in drei logische Schichten aufgeteilt werden: den CPU Kern, den System-on-Chip (SoC) und die Hauptplatine. Dieses Modell ist in Abbildung 1 dargestellt.

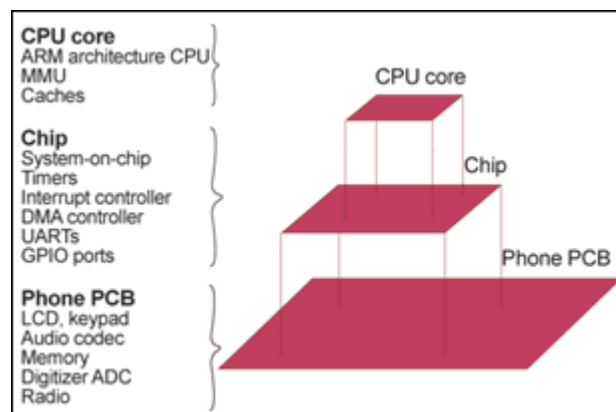


Abbildung 1 aus Creating Symbian OS phones

4. Symbian OS Architektur

Symbian OS verfügt über verschiedene Subsysteme für verschiedene Aufgaben. Zu diesen Systemen gehört das Basis-Subsystem, das den Rahmen für alle anderen Elemente darstellt. Es beinhaltet den Kernel, die Benutzer-Bibliothek, die verschiedenen Gerätetreiber und den File-Server.

Weiterhin gibt es Subsysteme für Telefonie, Messaging, Multimedia und einiges andere mehr. Auf diese soll hier aber nicht weiter eingegangen werden. Im Folgenden soll nur auf das Basis-Subsystem eingegangen werden, das den Kern des Betriebssystems darstellt.

Informationen zu den anderen Subsystemen sind in [1] erhältlich. Abbildung 2 zeigt einen allgemeinen Überblick über die Architektur von Symbian OS v8.0.

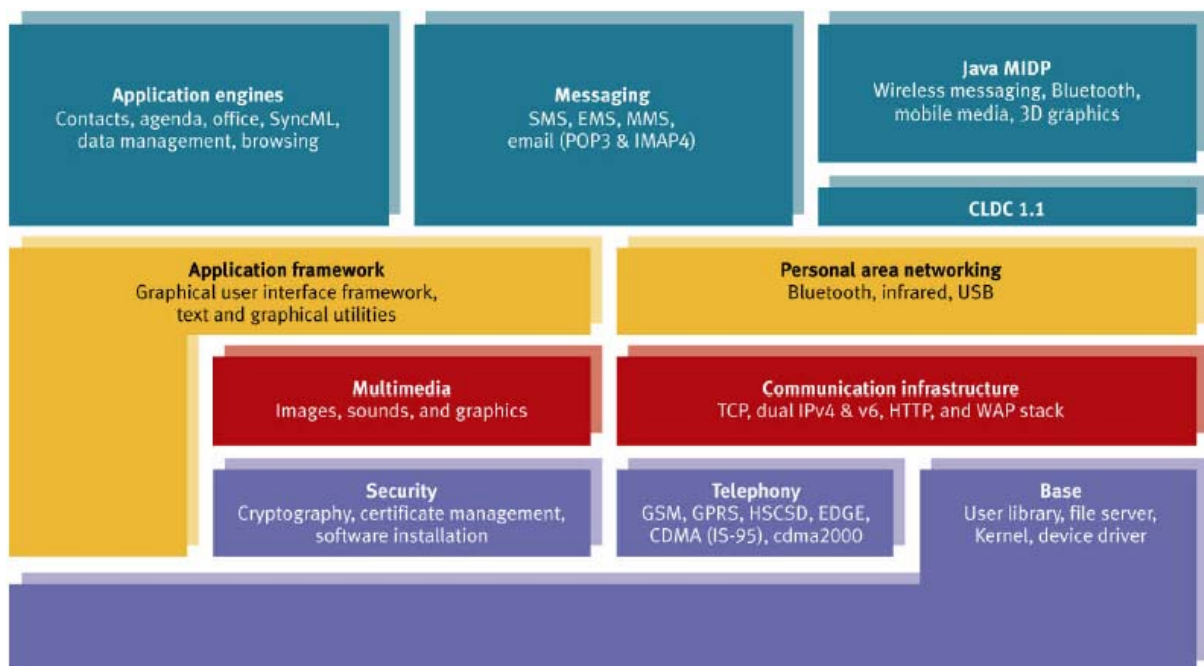


Abbildung 2 aus Symbian OS v8.0 product description

5. Kernel

Symbian OS hat einen kompakten, multitasking- und multithreading- fähigen Kernel, der sowohl monolithische als auch Mikrokern-Eigenschaften vereint.

An einen Mikrokern sind folgende Eigenschaften angelehnt:

- Ein Message-Passing Framework, das speziell für die Unterstützung von Servern auf der User-Seite entwickelt wurde.
- Netzwerk- und Telefonstack sind in Servern auf der User-Seite realisiert.
- Das Dateisystem ist in einem Server auf der User-Seite realisiert.

Für ein monolithisches System spricht:

- Gerätetreiber laufen auf der Kernel-Seite. Sie sind zwar nicht in den Kernel-Code eingebettet, können aber zur Laufzeit geladen werden, da sie in Bibliotheken implementiert sind.
- Der Scheduler und die Scheduling-Richtlinien sind in den Kernel implementiert.
- Die Speicherverwaltung ist in den Kernel implementiert.

Die MMU ist so konfiguriert, dass auf Hardware Register nur im Privilegierten Modus zugegriffen werden kann. Der Kernel läuft immer im privilegierten Modus und hat damit Zugriff auf alle Register. Anwendungen greifen auf Kernel-Dienste über eine API zu, die von der Benutzer-Bibliothek zur Verfügung gestellt wird. Da alle Anwendungen im nicht privilegierten Modus laufen, müssen Operationen, die auf Hardware zugreifen, eine Anfrage an den Kernel Server stellen, die einen Context-Switch zum Kernel Prozess einschließt.

Die Kernel Bibliothek stellt die Unterstützung für jegliche Peripherie-Hardware dar, die sich auf dem Chip befindet. Dazu gehören unter anderem Timer, Interrupt Controller und die DMA-Engine. Die Kernel Bibliothek ist auf einen speziellen Chip zugeschnitten. Anwendungen dürfen nicht direkt auf diese Peripherie-Hardware zugreifen. Sie müssen stattdessen auf die Benutzer-Bibliothek zugreifen, deren Funktionen dann über den Kernel die Peripherie steuern.

Peripherie Geräte für die Eingabe werden mit Hilfe von Kernel-Erweiterungen eingebunden. Es können verschiedene Erweiterungen für Tastatur, Keypad, Digitizer und Navigationsknöpfe geschrieben werden. Die entsprechenden Erweiterungen werden beim Booten vom Kernel erkannt und initialisiert.

6. Gerätetreiber

Gerätetreiber stellen eine API für Anwendungen zur Verfügung, welche die Steuerung von Hardware erlaubt, die nicht unbedingt notwendig für das Ausführen des Betriebssystems ist. Gerätetreiber können zu jeder Zeit nachgeladen und wieder entfernt werden. Ein Gerätetreiber besteht aus zwei Komponenten: eine Bibliothek die eine API des Gerätes bereitstellt, auf die Anwendungen dann zugreifen können; und eine Bibliothek auf der Kernel-Seite, die im privilegierten Modus läuft und auf die Hardware zugreift.

Die Bibliothek auf der Kernel-Seite ist häufig in zwei Teile aufgeteilt: einen „Logical Device Driver“ (LDD) und einen „Physical Device Driver“ (PDD). Die Idee dahinter ist, dass manche Arten von Hardware verschiedene Register haben, aber die gleiche Funktion haben. Der Hauptteil ist der LDD, welcher die logischen Funktionen eines Gerätes beinhaltet, wie zum Beispiel „an“ und „aus“, oder „lesen“ und „schreiben“, und nicht hardware-spezifisch ist. Die unterste Ebene die auf die Hardware Register zugreift ist der PDD. Dieses System macht die Portierung auf neue Hardware einfach, da immer nur der PDD angepasst werden muss.

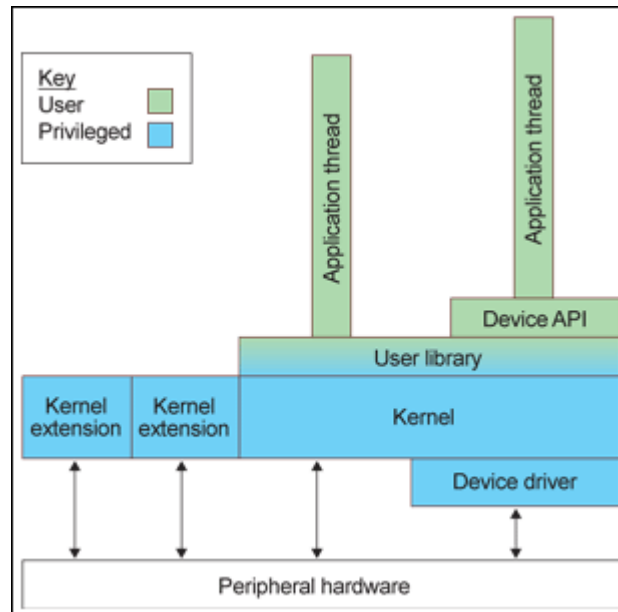


Abbildung 3 aus Creating Symbian OS phones

7. Threading Modell

Nur der Kernel läuft im privilegierten Modus und hat damit Zugriff auf den gesamten Speicher. Alle anderen Anwendungen können nur auf den ihnen zugewiesenen Speicherbereich zugreifen. Die kleinste Einheit dieses Speicherschutzbereichs wird in Symbian OS „Prozess“ genannt, während die kleinste Ausführungs-Einheit „Thread“ genannt wird. Ein Prozess kann aus einem oder mehreren Threads bestehen, im Falle des Kernels sind das der „Kernel Server“ und der „Null“ Thread. Die Prozessorzeit wird auf Threads und nicht auf Prozesse verteilt.

Der Kernel Server ist der Thread mit der höchsten Priorität und bedient Anfragen von Benutzer-Threads. Im gleichen Prozess ist der Null Thread derjenige mit der niedrigsten Priorität. Er ist beim Start für das Laden des Fileserver und das Booten des Kernel zuständig. Sonst ist er für das Aufrufen der Energie-Subsystem-Dienste zuständig, um das System in die verschiedenen Energiesparmodi zu versetzen.

Quellen:

- [1] Sander Siezen, Symbian OS Version 8.0 Product description, Revision 2.1, Februar 2004
- [2] Peter Sanders, Creating Symbian OS phones Revision 1.1. April 2002, auf www.symbian.com
- [3] Symbian Developer Library für Symbian OS Version 7.0, 2002, auf www.symbian.com
- [4] Symbian Limited Q1 2004 results, Symbian press release, 06.05.2004
- [5] www.symbian.com vom 12.07.2004