

Systemy operacyjne dla urządzeń mobilnych

Aleksander Gajewski
Adam Morawski
Karol Ruszczyk

MIMUW

13 stycznia 2009

Plan prezentacji

- 1 Wstęp
 - Trochę historii
 - Użytkownik
 - Hardware
 - Architektura ARM
- 2 Przegląd mobilnych systemów operacyjnych
 - Android
 - Palm OS, Palm webOS
 - iPhone OS
 - Symbian OS
 - Windows Mobile
- 3 Podsumowanie

Początki

Pierwszy mikroprocesor

- Intel 4004:
 - 1971, Ted Hoff
 - 4 bity, do kalkulatora

Pierwsze mikrokontrolery

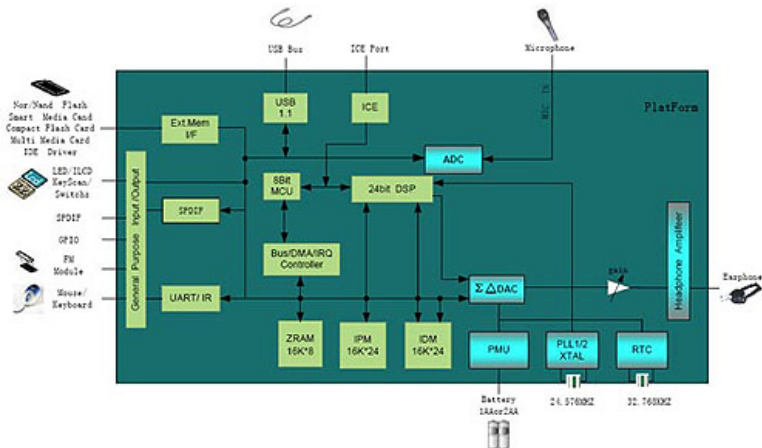
- TI TMS1000 (1971)
- Intel 8048

Rozwój mikrokontrolerów

Mikrokontroler - system mikroprocesorowy zrealizowany w postaci pojedynczego układu scalonego, zawierającego jednostkę centralną (CPU), pamięć RAM, na ogół, pamięć programu oraz rozbudowane układy wejścia-wyjścia.

- Proste, zorientowane na operacje we/wy i sterowanie innymi urządzeniami: Intel 8051, Atmel AVR
- Specjalizowane: prosty rdzeń i sprzętowy układ do konkretnych większych obliczeń
- Skomplikowane, zorientowane na obliczenia wewnętrzne, wydajne: rdzeń ARM lub MIPS
- Application Processors

Mikrokontroler specjalizowany - ATJ2085



Plan prezentacji

- 1 Wstęp
 - Trochę historii
 - Użytkownik
 - Hardware
 - Architektura ARM
- 2 Przegląd mobilnych systemów operacyjnych
 - Android
 - Palm OS, Palm webOS
 - iPhone OS
 - Symbian OS
 - Windows Mobile
- 3 Podsumowanie

Użytkownik

Oczekiwania:

- Małe wymiary
- Szybkie działanie
- Długie działanie po naładowaniu
- Wygodne użytkowanie
- Wodotryski
- Niezawodność
- Niska cena

Plan prezentacji

- 1 Wstęp
 - Trochę historii
 - Użytkownik
 - **Hardware**
 - Architektura ARM
- 2 Przegląd mobilnych systemów operacyjnych
 - Android
 - Palm OS, Palm webOS
 - iPhone OS
 - Symbian OS
 - Windows Mobile
- 3 Podsumowanie

Application processor

Obecny trend w mobilnych urządzeniach multimedialnych, to application processor - układ zawierający:

- Szybki rdzeń, zwykle z rodziny ARM
- Niewielką pamięć cache, brak wbudowanej pamięci RAM
- Niewielką pamięć ROM programu, w której zwykle znajduje się bootloader ładujący system z zewnętrznej kości flash
- Procesor sygnałowy, często sprzętowe kodeki AV
- Układ dźwiękowy, udostępniający wejścia i wyjścia analogowe

oraz mogący zawierać:

- Układ graficzny, często rozbudowany
- Nietypowe i wymagające interfejsy, np. do modułu kamery
- Układy Bluetooth czy Wifi, wymagające do działania tylko części radiowej
- Układ ładowania baterii

Application processor

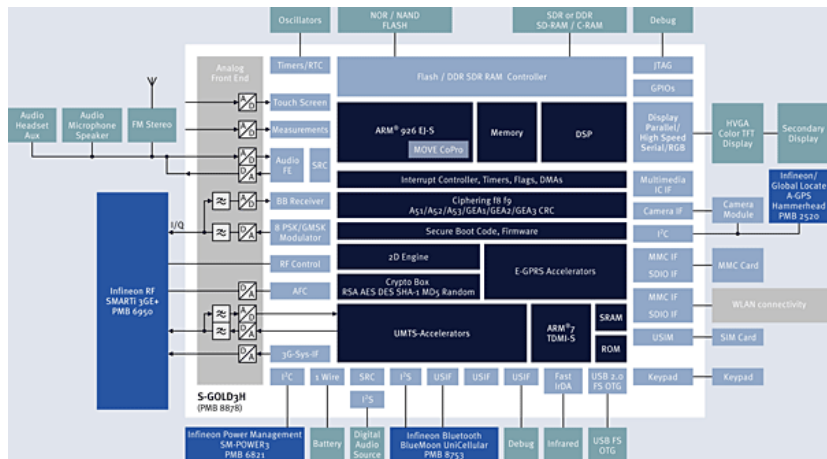
Poza układem znajdują się:

- Względnie duża pamięć RAM
- Pamięć flash (właściwe oprogramowanie, dane użytkownika)
- Ewentualny zewnętrzny, rozbudowany, układ graficzny

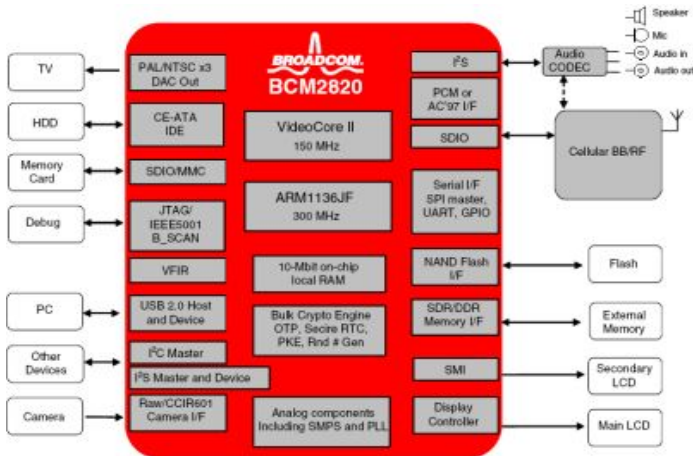
Przykłady:

- Marvell (d. Intel) XScale
- Texas Instruments OMAP
- Nvidia GoForce 6100

Application processor - Infineon



Application processor - Broadcom



Hardware - Specyfika podzespołów

- Rdzeń - mikroprocesor
 - Obecnie najczęściej ARM lub MIPS - RISC
- Pamięć RAM
 - Mniejsza i wolniejsza, niż w PC
 - Oszczędność pamięci wymagana przez koszt i oszczędność energii
 - Często typowym trybem "wyłączenia" urządzenia jest wyłączenie peryferiów i podtrzymanie pamięci RAM
- Pamięć ROM
 - Obecnie prawie zawsze FLASH
 - Oprogramowanie, czasem część dostępna dla użytkownika
- Pozostałe komponenty
 - Zwykle bardziej wyspecjalizowane i o mniejszych możliwościach niż ich odpowiedniki w PC.

Hardware - Specyfika urządzenia

- Brak tak uniwersalnej architektury, jak w przypadku komputerów PC
- Wszystko (szybkość, pamięć, używane interfejsy, zasilanie) jest dostosowane do danego urządzenia, projektowane bez dużego zapasu.
- System operacyjny dostosowany do danego urządzenia, bez komponentów, które "mogą się przydać"

Plan prezentacji

- 1 Wstęp
 - Trochę historii
 - Użytkownik
 - Hardware
 - Architektura ARM
- 2 Przegląd mobilnych systemów operacyjnych
 - Android
 - Palm OS, Palm webOS
 - iPhone OS
 - Symbian OS
 - Windows Mobile
- 3 Podsumowanie

Architektura ARM

- IP Core
- Zasięg: ok. 75% systemów wbudowanych, w 2007 - 98% sprzedanych telefonów komórkowych, przenośne konsole do gier, PDA, odtwarzacze multimedialne
- 32-bitowy RISC

Oznaczenia rozszerzeń architektury:

- T - zestaw instrukcji Thumb - mniejsze i wolniejsze
- E - zestaw instrukcji DSP
- J - Jazelle



Architektura ARM - ARMv4

- StrongARM
 - Opracowany przez DEC w 1995, przejęty przez Intel
 - Zmodyfikowana część instrukcji oryginalnego ARM, rozdzielona pamięć instrukcji i danych, MMU
 - Znacznie zwiększona wydajność
 - Zastosowanie: PDA (m.in. nowsze Apple Newton, część wczesnych PocketPC), terminale set-top box
- ARM7TDMI
 - Zestaw instrukcji THUMB, uł
 - Optymalizowane pod kątem zużycia energii
 - Bardzo rozpowszechnione: jako główne rdzenie wielu niedrogich mikrokontrolerów, SoCach m.in. starszych iPodów, GameBoy Advance, przez pewien czas wszystkich telefonów Nokia
 - Procesory pomocnicze w bardziej wydajnych procesorach aplikacji

Architektura ARM - ARMv5

- ARM9E

- Pierwszy ARM oryginalnie z architekturą harwardzką
- Rozszerzone instrukcje DSP, możliwe rozszerzenia: MMU, Jazelle
- Zastosowanie: nowsze telefony SonyEricsson, Siemens, niektóre Nokie N-series, TI OMAP 16xx, 17xx

- Intel/Marvell XScale

- Następca StrongARM, istnieją różne specjalizacje (Application Processor, I/O Processor, Network Processor, Control Plane Processor, Consumer Electronics Processor)
- Application Processors
 - Zastosowanie: większość współczesnych palmtopów, nawigacje GPS
 - Istniejące w różnych wersjach rozszerzenia: zintegrowana w jednej obudowie duża pamięć RAM i flash, SpeedStep, MMX

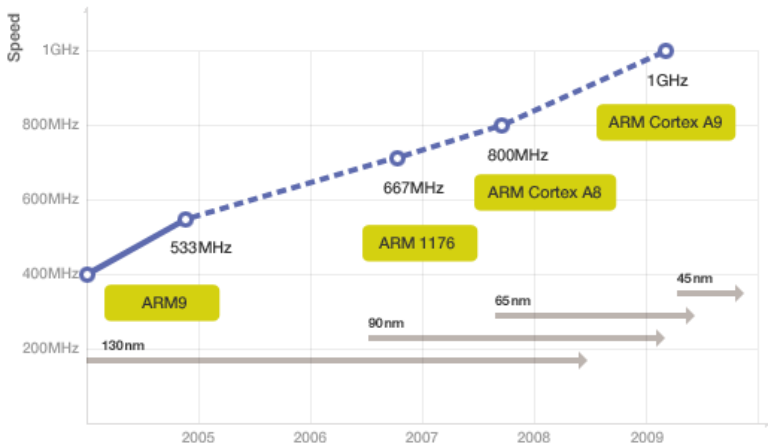
Architektura ARM - ARMv6

- ARM11
 - Możliwe rozszerzenia: MMU, SIMD, Jazelle, VFP (koprocessor), SMP
 - Zastosowanie: szybsze Nokie N-Series, iPhone/iPod touch, Nvidia Tegra APX 2500

Architektura ARM - ARMv7

- Cortex
 - Możliwe rozszerzenia: TrustZone, 2 poziomowy cache, VFP, NEON (Advanced SIMD)
 - Zastosowania: Palm Pre, mikrokontrolery (Cortex-M3)

Architektura ARM



Plan prezentacji

- 1 Wstęp
 - Trochę historii
 - Użytkownik
 - Hardware
 - Architektura ARM
- 2 Przegląd mobilnych systemów operacyjnych
 - **Android**
 - Palm OS, Palm webOS
 - iPhone OS
 - Symbian OS
 - Windows Mobile
- 3 Podsumowanie

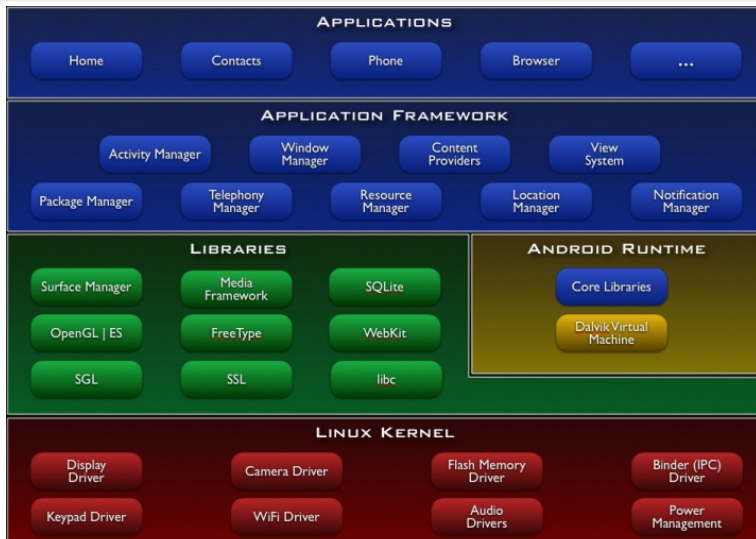
Open Handset Alliance



Android - kernel

Android opiera się o kernel w wersji 2.6 - to jest wykorzystuje usługi systemowe takie jak bezpieczeństwo systemu, zarządzanie pamięcią, zarządzanie procesami, stos sieci, model sterowników.

Android - struktura



Android - charakterystyka

- framework umożliwiający reużywalność i podmienianie komponentów
- zoptymalizowany Dalvik virtual machine
- zintegrowana przeglądarka oparta o otwarty silnik webkit
- zoptymalizowana grafika oparta o biblioteki 2d i 3D (OpenGL ES 1.0)
- baza danych SQLite

Android - charakterystyka cd.

- wsparcie dla dźwięku, filmów obrazów (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- wsparcie dla GSM
- Bluetooth, EDGE, 3G, i WiFi
- Camera, GPS, kompas, akcelerometr
- bogate IDE zawierające emulator, narzędzia do debugowania i profilowania pamięci i wydajności

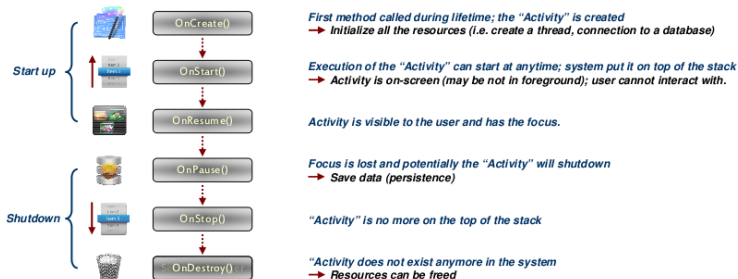
Android - komponenty - przykłady

- bogaty zestaw widoków (listy, gridy, text box'y, buttony, zintegrowana web browser)
- Content Providers
- Resource Manager (zasoby typu non-code jak napisy, obrazy, pliki)
- Notification Manager
- Activity Manager

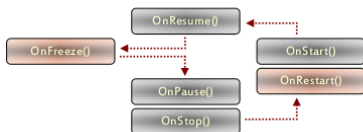
Android - biblioteki

- System C library - pochodna implementacji libc BSD, zoptymalizowana pod urządzenia mobilne
- Media Libraries
- Surface Manager - zarządzanie podsystemem wyświetlacza
- LibWebCore - silnik przeglądarki
- SGL - silnik 2D
- OpenGL ES 1.0 API, wsparcie akceleracji sprzętowej
- SQLite - wydajny i lekki relacyjny silnik baz danych

Android - activity

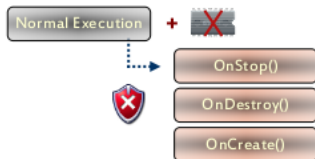


Android - activity - cykl życia



On Freeze(): All the "Activity" to save UI state
Save the value belonging to the UI (i.e. field in a form)

On Restart(): Signal the "Activity" will be restarted



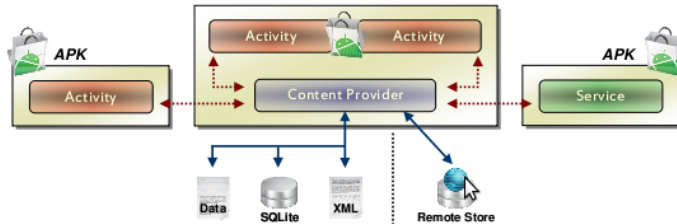
The system can:

- Ask the user to finish it
- Directly kill the process it belongs to

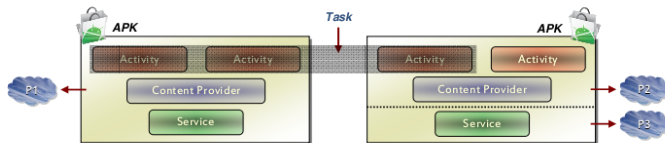
Android - żądania



Android - content provider



Android - procesy i zadania



Plan prezentacji

- 1 Wstęp
 - Trochę historii
 - Użytkownik
 - Hardware
 - Architektura ARM
- 2 Przegląd mobilnych systemów operacyjnych
 - Android
 - Palm OS, Palm webOS
 - iPhone OS
 - Symbian OS
 - Windows Mobile
- 3 Podsumowanie

Palm



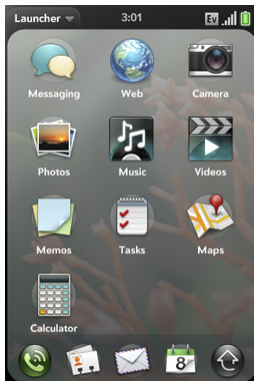
PalmOS - cechy 'obecnej' wersji

- proste, jednozadaniowe środowisko
- obsługa wyświetlacza 480x320
- system rozpoznawania pisma ręcznego Graffiti 2
- synchronizacja HotSync
- możliwość odtwarzania i rejestrowania dźwięku
- dostęp do sieci TCP/IP
- obsługa portu szeregowego, USB, podczerwieni, Bluetooth oraz Wi-Fi
- rozszerzenie o karty pamięci sd, mmc
- ustalony standard formatu dla podstawowych aplikacji: calendar, address, task, notes (standardowa synchronizacja)

PalmOS - rozwój aplikacji

- aplikacje kodowane w C/C++
- dwa środowiska developerskie:
- komercyjne CodeWarrior Development Studio
- otwarte prc-tools, oparte o gcc.
- inne środowiska takie jak pocketC, freepascal, basic
- java (już) niewspierana

Palm webOS - xhtml, css, javascript



Plan prezentacji

- 1 Wstęp
 - Trochę historii
 - Użytkownik
 - Hardware
 - Architektura ARM
- 2 Przegląd mobilnych systemów operacyjnych
 - Android
 - Palm OS, Palm webOS
 - **iPhone OS**
 - Symbian OS
 - Windows Mobile
- 3 Podsumowanie

iPhone OS



iPhone - warstwy abstrakcyjne

Core OS - To najniższa warstwa w której celem jest interakcja między sprzętem a oprogramowaniem, w jego skład wchodzi jądro Darwin.

iPhone - warstwy abstrakcyjne

Core Services - To rdzenny zestaw podstawowych bibliotek do zarządzania pracą aplikacji i wątków, obsługa sieci, obsługa bazy danych (SQLite) i inne których działanie nie jest bezpośrednio widoczne przez użytkownika.

iPhone - warstwy abstrakcyjne

Media - Ta warstwa zawierająca obsługę obrazu i dźwięku, również odtwarzaniem wideo i obsługę formatów plików graficznych, w skład warstwy wchodzi znane biblioteki takie jak OpenGL, OpenAL czy Core Animation.

iPhone - warstwy abstrakcyjne

Cocoa Touch - Jest to biblioteka interfejsu użytkownika wykorzystaniem ekranu dotykowego, co różni go od tradycyjnego Cocoa z systemu Mac OS X. W jego skład wchodzi również obsługa przyspieszeniomierza.

iPhone - w Polsce

Jailbreak - polega na zdobyciu dostępu do systemu plików iPhone, umożliwiając uruchamianie nieautoryzowanych przez Apple aplikacji, dogrywanie/podmienianie plików oraz aktywację omijającą iTunes

Plan prezentacji

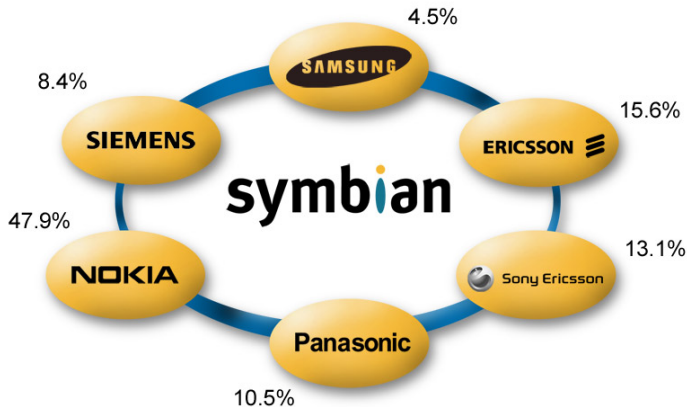
- 1 Wstęp
 - Trochę historii
 - Użytkownik
 - Hardware
 - Architektura ARM
- 2 Przegląd mobilnych systemów operacyjnych
 - Android
 - Palm OS, Palm webOS
 - iPhone OS
 - Symbian OS
 - Windows Mobile
- 3 Podsumowanie

Symbian OS

Informacje/założenia:

- 32-bitowy, little-endian
- zwarty wielozadaniowy OS z bardzo małymi zależnościami
- wsparcie dla urządzeń peryferyjnych jest zintegrowane z kernelem jako rozszerzenia jądra, napisane jako oddzielne DLL.
- kernel pracuje w trybie uprzywilejowanym i wystawia swoje usługi aplikacjom za pomocą bibliotek użytkownika
- Symbian dostarcza również zbiór bibliotek obsługujących:
 - sieć (TCP/IP, PPP, FTP)
 - komunikację bezprzewodową (Bluetooth, IrDA)

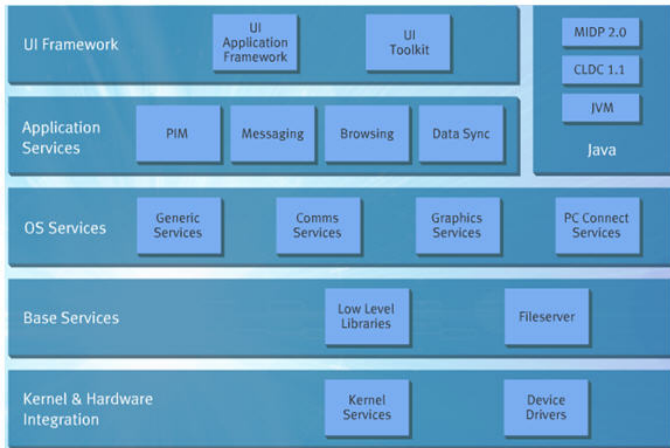
Symbian OS - Właściciele



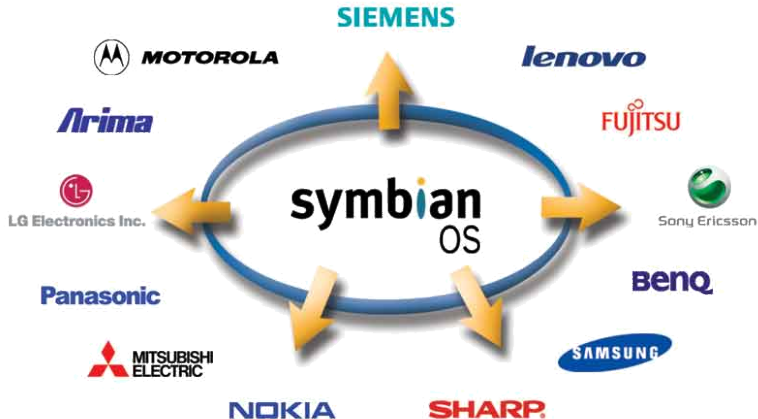
Symbian OS

- Symbian OS jest dostępny z kilkoma różnymi interfejsami użytkownika, najbardziej popularne to:
 - S60
 - UIQ
- Interfejs użytkownika jest modyfikowany i dostosowywany przez wytwórców konkretnych urządzeń.

Symbian OS - Architektura



Symbian OS - Licencje



Symbian OS - Pierwszy program

Instalacja:

- Carbide.c++ (bazuje na Eclipse)
- ActivePerl 5.1.8
- Symbian SDK

Symbian OS - Pierwszy program

Prezentacja tworzenia prostego programu.

Symbian OS - Wnioski z punktu widzenia programisty

- Bardzo przyjazne IDE
- Prosta instalacja
- Działający emulator pod Windows
- Przymus trzymania się dość przestarzałych paradygmatów programowania (stworzonych na początku lat 90)

Plan prezentacji

- 1 Wstęp
 - Trochę historii
 - Użytkownik
 - Hardware
 - Architektura ARM
- 2 Przegląd mobilnych systemów operacyjnych
 - Android
 - Palm OS, Palm webOS
 - iPhone OS
 - Symbian OS
 - Windows Mobile
- 3 Podsumowanie

Windows Mobile

Informacje/założenia:

- Microsoft look-and-feel
- Platforma rozrywki
- Od wersji 6 - stabilność
- Dostępny zarówno dla urządzeń z dotykowymi ekranami jak i bez (odpowiednio wersje Professional i Standard)
- Gotowy do użycia zaraz po wyjęciu z pudełka.

Windows Mobile

Pisząc aplikację na Windows Mobile możemy używać:

- natywnego kodu (wysoka wydajność, bezpośredni dostęp do sprzętu) : Visual C++, Windows Mobile Native API
- kodu zarządzanego (w większości przypadków gdy chcemy szybko zakodować jakąś aplikację) : C# .NET Compact Framework
- kodu po stronie serwera (wbudowana przeglądarka umożliwia używanie aplikacji webowych)

Windows Mobile - .NET Compact Framework

Czym jest .NET Compact Framework?

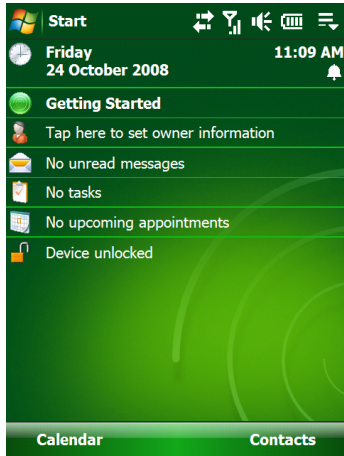
- .NET Compact Framework umożliwia używanie z .NET Framework w urządzeniach zaopatrzonych w Windows Mobile.
- .NET Compact Framework jest niezależnym od sprzętu środowiskiem do uruchamiania programów na urządzeniach ze ściśle ograniczonymi zasobami.
- Sprowadza się to do tego, że .NET Compact Framework jest podzbiorem biblioteki klas .NET Framework.

Symbian vs Windows Mobile

Zalety Windows Mobile:

- standaryzacja
- więcej 'profesjonalnych' aplikacji
- widok 'Today'

Symbian vs Windows Mobile



Symbian vs Windows Mobile

Zalety Symbian OS:

- stabilność
- większa ilość możliwych do doinstalowania aplikacji
- urządzenie z Symbian OS to nadal telefon

Symbian vs Windows Mobile - Podsumowanie

*"Jeżeli chcesz telefon z 'bajerem' wybierz Symbian OS,
a jak chcesz 'bajer' z telefonem to Windows Mobile"*

Podsumowanie

- Obecnie na rynku dominują systemy:
 - Symbian OS
 - Windows Mobile
 - Palm OS
- Od około 2 lat powstał silny trend sugerujący wykorzystanie systemów z rodziny Unix w urządzeniach mobilnych. Jego efektem są:
 - iPhone OS
 - Android
 - nowe wydania Palm OS