Systemy operacyjne na urządzenia mobilne

Piotr Jastrzębski
Piotr Laskowski
Maciej Szarliński
Tomasz Turski

Plan prezentacji

Odrobina historii

SIBO II (1986 rok – 0,614 MHz, 16 KB RAM, EPOC16)

Odrobina historii

Apple Newton (1993 r.) - 20MHz, 640KB RAM, NewtonOs

Palm m505 (2001 r.)

33 MHz, 8 MB RAM, PalmOs 4.0

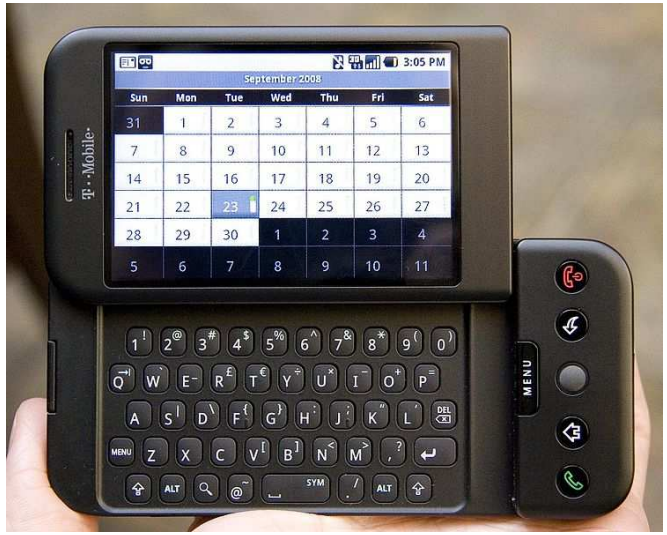
Odrobina historii

BlackBerry 8350

(2005 r.)

312 MHz, 16 MB RAM

BlackBerry OS



T-Mobile G1 (2008r.)

528 Mhz, 192 RAM

Android

iPhone (2007 r.)

412 Mhz, 128 RAM
iPhone OS

Oczekiwania

- jako urządzenia przenośnego:
  - interaktywność
  - funkcjonalność telefonu
  - wymiary
  - waga
  - zasilanie
  - niezawodność
  - bezpieczeństwo
- jako w pełni funkcjonalnego komputera:
  - wielozadaniowość
  - wielowątkowość
  - zarządzanie zasobami
  - rozszerzalność
  - obsługa różnych urządzeń
  - zewnętrzne aplikacje
  - sprawna obsługa multimediów

Sprzęt - procesory

Możliwe ograniczenia:

- ograniczona programowalność
- brak wsparcia sprzętowego (MMU, obsługa przerwań)

Oczekiwania:

- niski pobór energii
- moc obliczeniowa
- dobra przepustowość operacji wejścia/wyjścia
- optymalizacja przetwarzania strumieni danych

Sprzęt - dostępne architektury

- **ARM** (Advanced RISC Machine) – najpopularniejszy. Na jego bazie powstał Xscale. Nacisk położony na zwiększenie efektywności potokowości.
- **MIPS** – (Microprocessor without Interlocked Piped Stages) architektura RISC firmy MIPS Technologies. Dwie wersje: 32- i 64-bitowa. Głownie w Windows CE. 1/3 produkcji mikroprocesorów typu RISC.
- **Crusoe i Efficeon** – Procesory firmy Transmeta, technologia LongRun, kompatybilne z x86.

Pamięć

- S(D)RAM lub dla systemu operacyjnego
- FLASH na kod systemu i aplikacji
- pamięci masowe
- brak pamięci wirtualnej
- wspólna pamięć składowania i uruchamiania
- uruchamianie programów "in place"

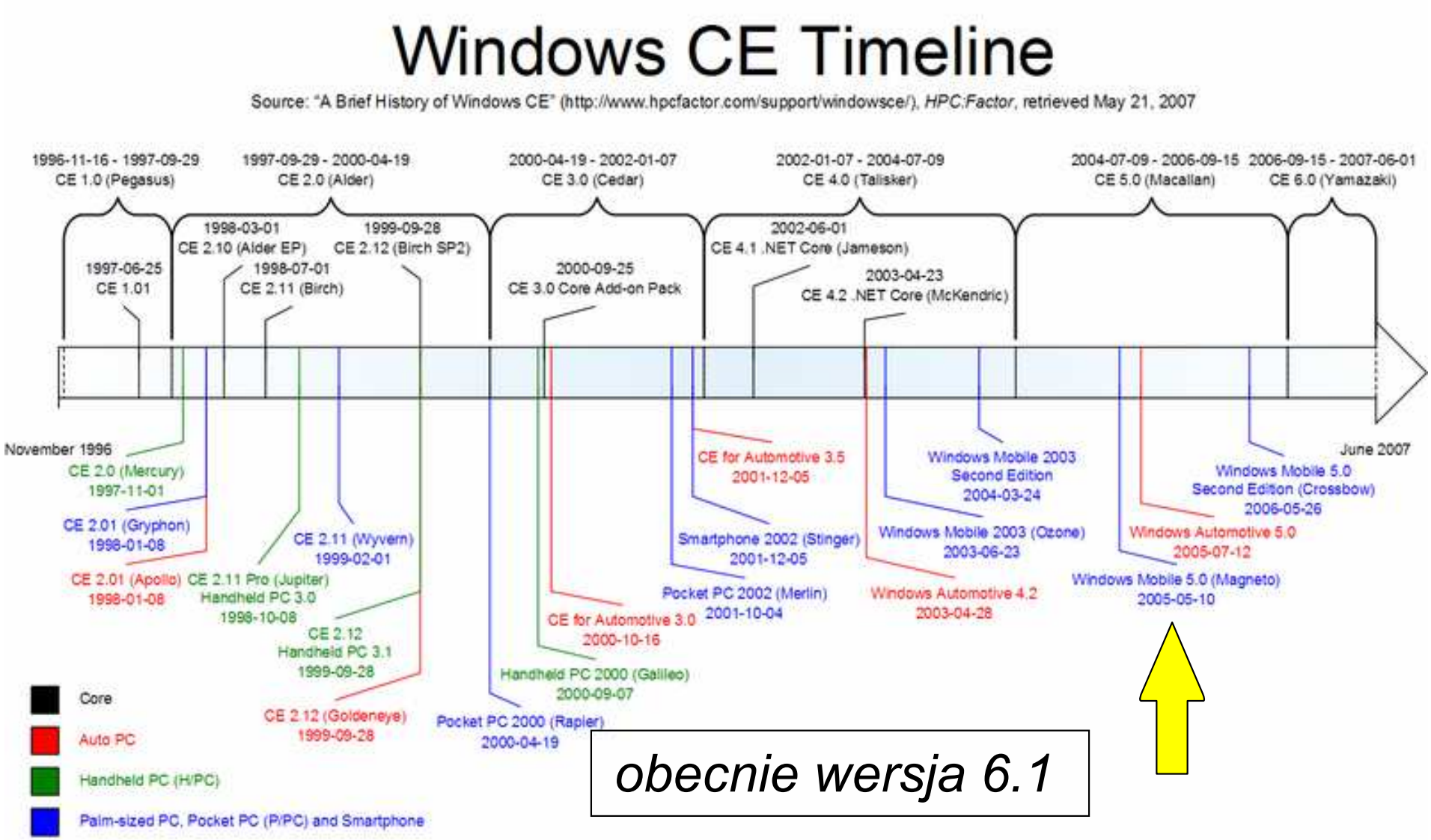
Funkcjonalność telefonu

- różnorodność sprzętu
- mnogość form komunikacji
- software real-time

Microsoft Mobile

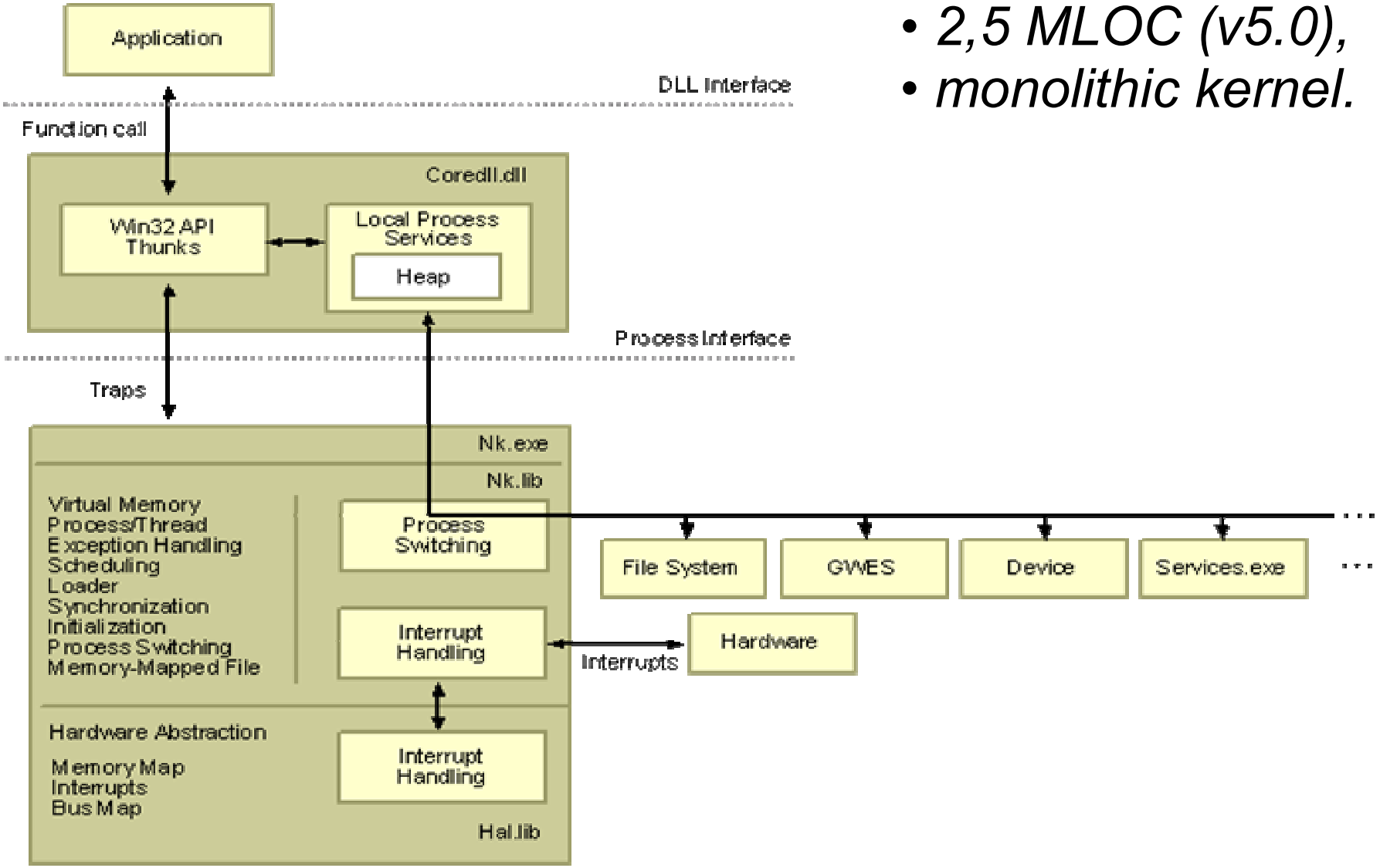
1. **Windows** dla urządzeń mobilnych.
2. Architektura jądra.
3. Procesy i wątki.
4. Syscalls.
5. Zarządzanie pamięcią i energią.

Windows dla urządzeń mobilnych



obecnie wersja 6.1

Architektura jądra



- *2,5 MLOC (v5.0),*
- *monolithic kernel.*

Procesy i wątki

- do 32 procesów (*Windows Mobile 6.0*)

```
BOOL CreateProcess(
LPCTSTR lpApplicationName,
LPTSTR lpCommandLine,
LPSECURITY_ATTRIBUTES lpProcessAttributes,
LPSECURITY_ATTRIBUTES lpThreadAttributes,
BOOL bInheritHandles,
DWORD dwCreationFlags,
LPVOID lpEnvironment,
LPCTSTR lpCurrentDirectory,
LPSTARTUPINFO lpStartupInfo,
LPPROCESS_INFORMATION lpProcessInformation );
```

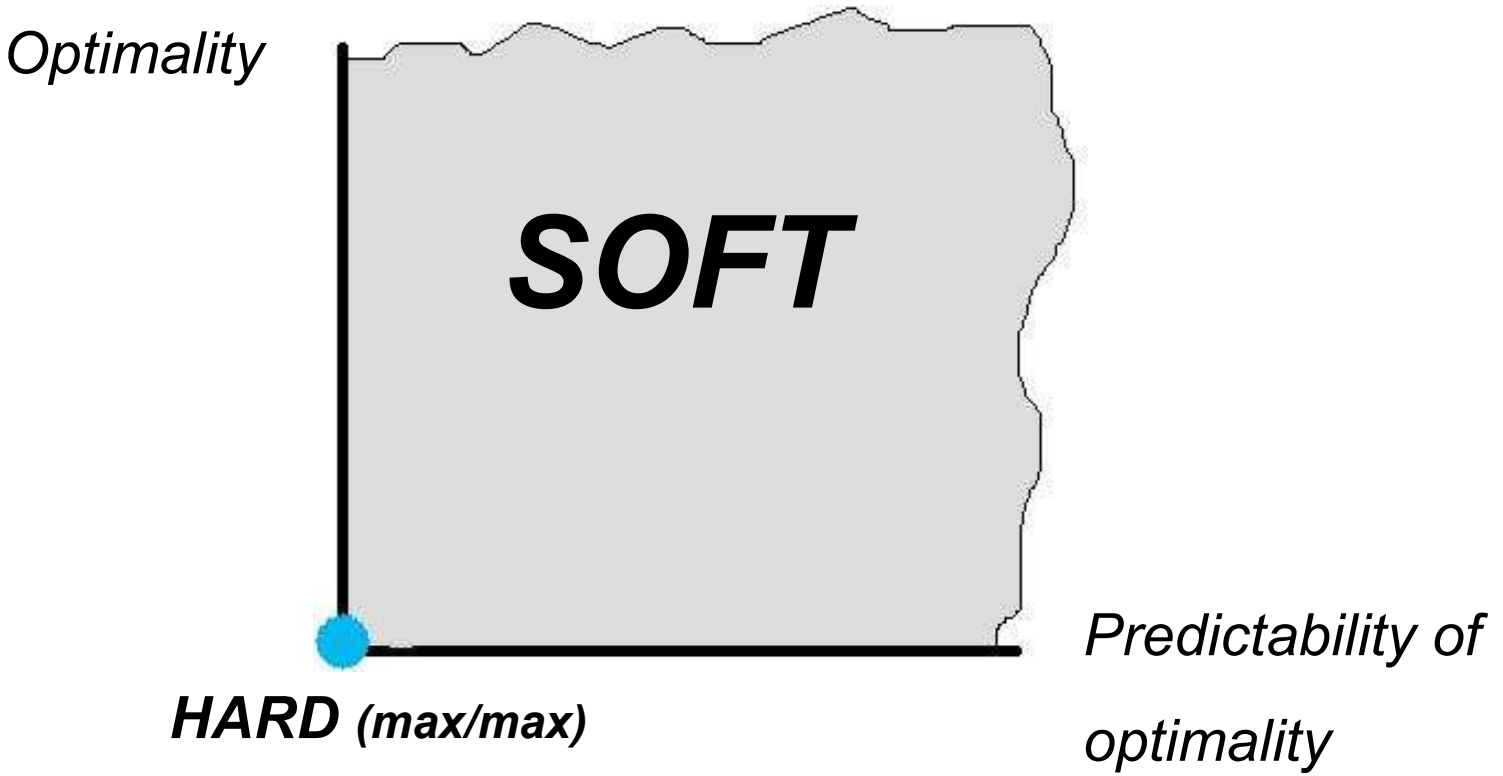
Procesy i wątki

- 256 priorytetów dla wątków,
- maksymalny priorytet to zero,
- wspólne zasoby,
- każdy wątek ma swój stos - `STACK_SIZE_PARAM_IS_A_RESERVATION`,
- 5 stanów - `RUNNING`, `SLEEPING`, `SUSPENDED`, `BLOCKED`, `TERMINATED`.

Procesy i wątki

Hard real-time a soft real-time

- *Hard time* – nieprzekraczalne deadliny (sprzęt medyczny),
- *Soft time* – minimalizacja ilości przekroczonych terminów i sumarycznego opóźnienia (niedeterminizm).



Procesy i wątki

Windows Mobile *chce być hard real-time*

- scheduling wątków o wysokim priorytecie z górnymi granicami,
- obsługa przerwań z górnymi granicami (*interrupt latency*),
- ***"Fine control over scheduler and how it schedules threads"*** ;)
- zapobieganie *priority inversion*,
- zegar systemowy co 1 ms,
- synchronizacja (mutexy, semfaory, …).

Narzuty:

- *Graphics, Windowing, Event Subsystem (GWES)* – wątek może czekać nieograniczony czas na sekcję,
- odwołania do systemu plików (problem j.w.),
- błędy braku strony.

Procesy i wątki

Wbudowane benchmarki

- *ILTimer.exe,*
- *OSBench.exe*.

Procesy i wątki

Więcej o schedulingu

- round-robin
- wątek o większym priorytecie wygania ten o mniejszym, ale:
- *dziedziczenie priorytetu* - przykład

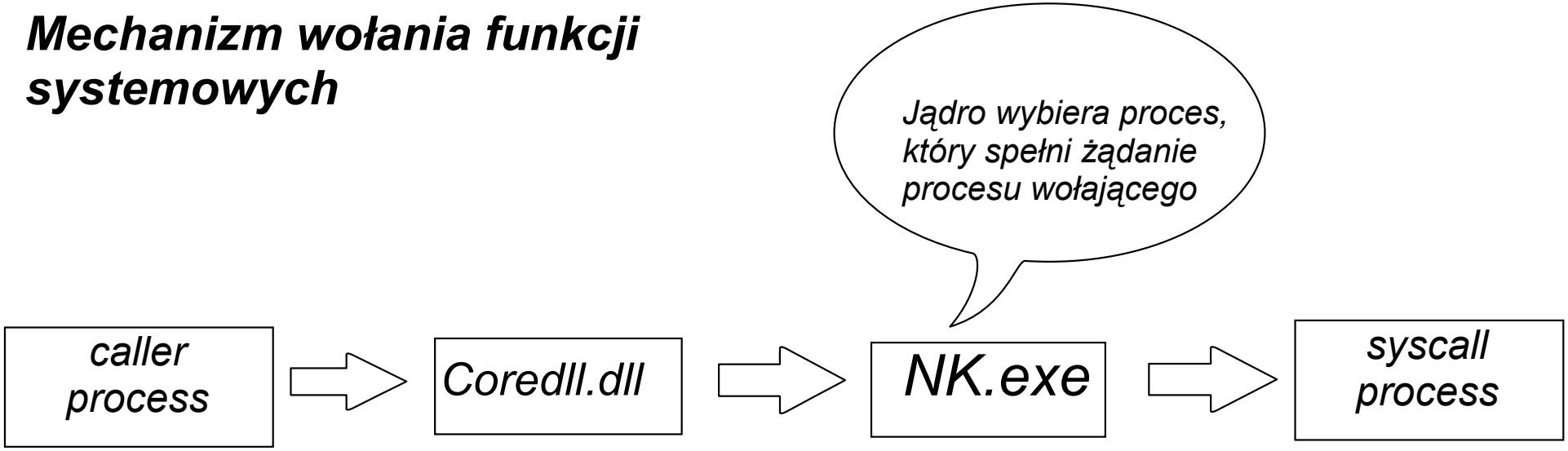
Procesy i wątki

Synchronizacja

- klasa `CriticalSection`,
- kolejki komunikatów,
- semafory,
- muteksy,
- zdarzenia (ang. *events*).

Syscalls

Mechanizm wołania funkcji systemowych



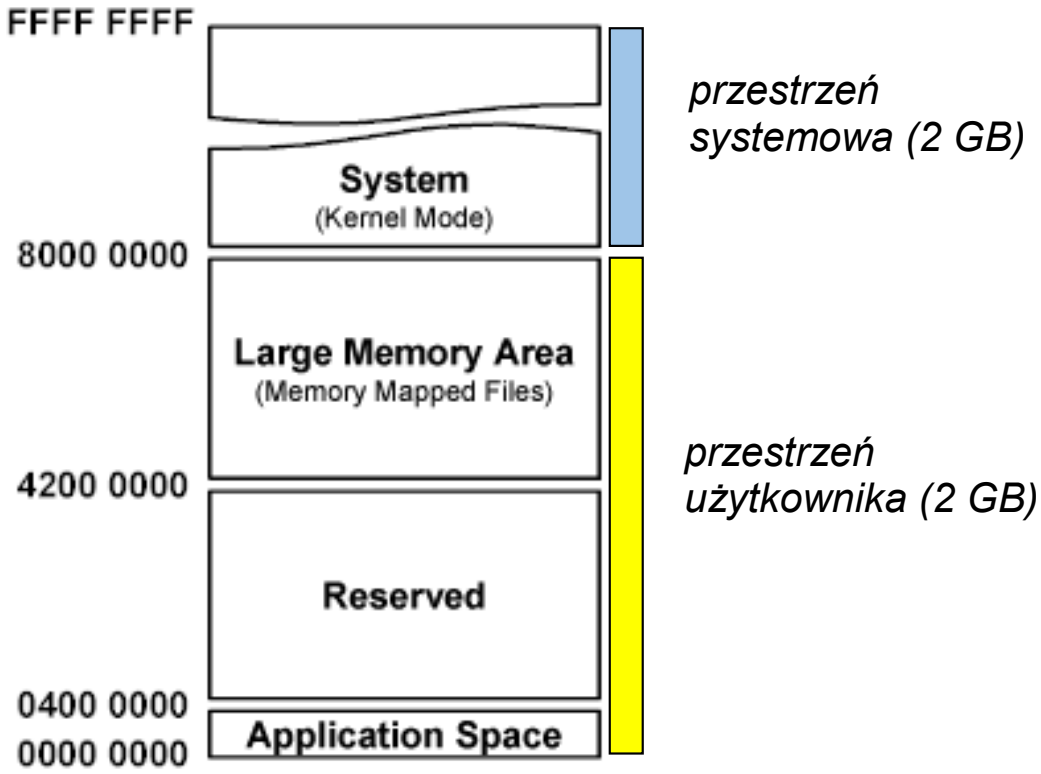
Przekazywanie wątku

wątek - repatriant

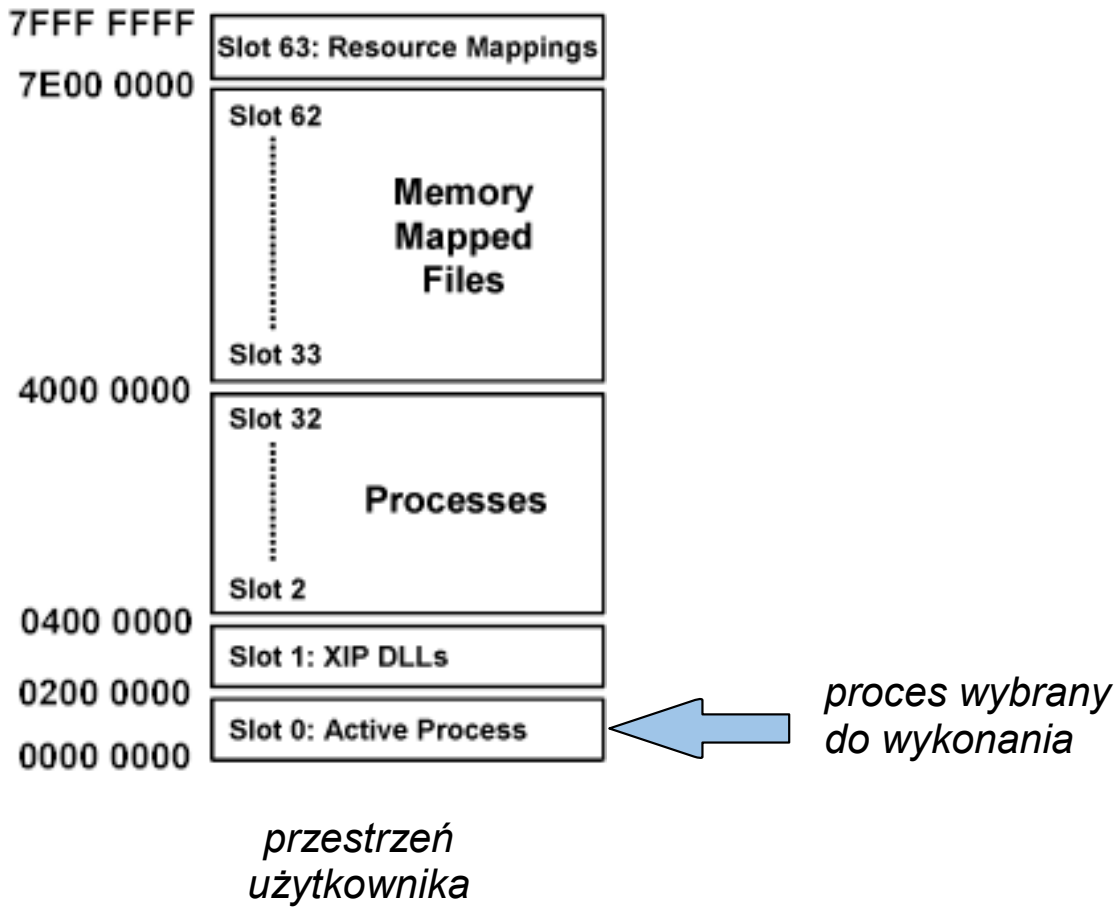
Zarządzanie pamięcią i energią

- model pamięci,
- alokacja pamięci,
- RAM i ROM,
- Persistent Storage.

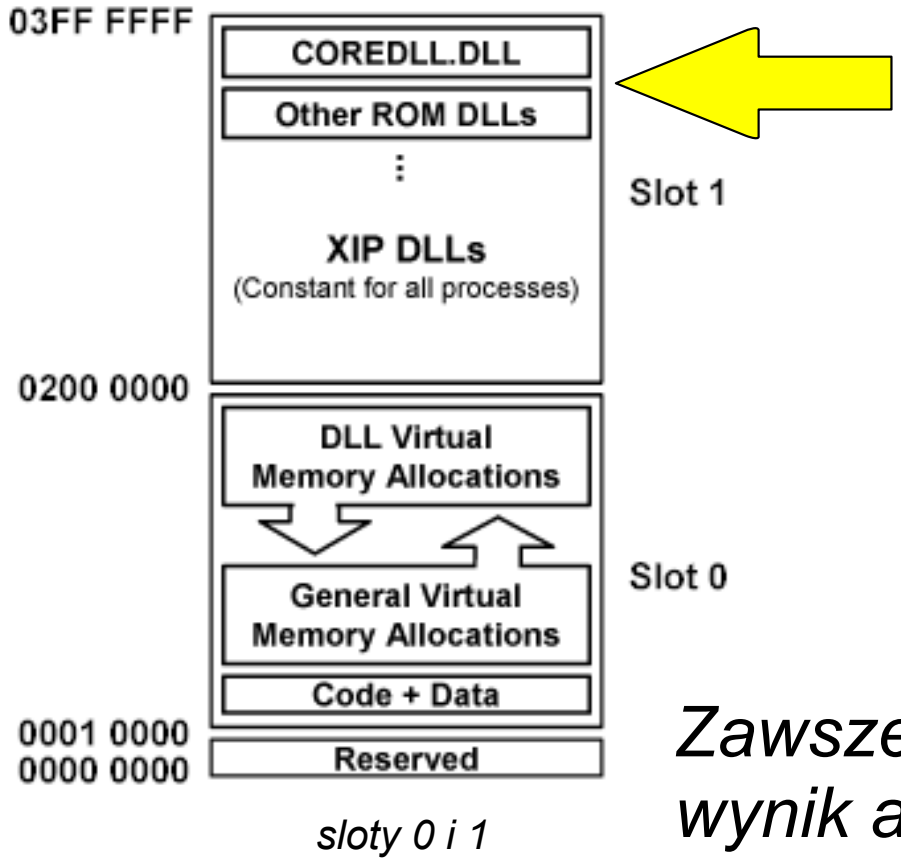
Zarządzanie pamięcią i energią



Zarządzanie pamięcią i energią



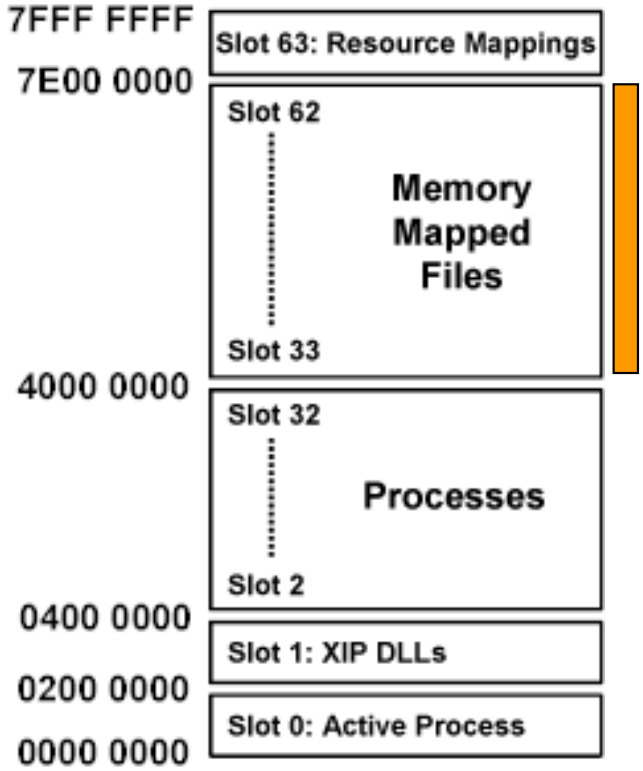
Zarządzanie pamięcią i energią



Zawsze sprawdzaj wynik alokacji

Zarządzanie pamięcią i energią

- wyrównywanie,
- większe alokacje,
- *memory-mapped files*.



przestrzeń użytkownika

Zarządzanie pamięcią i energią

Podsumowanie i wnioski

1. do 32 MB dla każdego procesu użytkownika,
2. do 32 procesów w systemie (tyle jest slotów),
3. zawsze sprawdzaj, czy alokacja się powiodła (jest ciasno),
4. staraj się przewidywać i robić od razu większe rezerwacje (wyr. do 64 KB), a następnie alokuj (wyr. do 4 KB stron),
5. łącz mniejsze DLL w większe,
6. alokacje powyżej 2 MB mają miejsce w Large Memory Area,
7. używaj mapowanych plików zamiast dużych alokacji,
8. dziel pamięć pomiędzy procesy.

Zarządzanie pamięcią i energią

Wprowadzenie do Persistent Storage - RAM a ROM

- pobór energii,
- cena,
- ilość zapisów.

Zarządzanie pamięcią i energią

Persistent Storage - zalety

- bezpieczeństwo,
- żywotność

Zarządzanie pamięcią i energią

Persistent Storage - problemy

- czas dostępu,
- ograniczona ilość zapisów.

Wsparcie systemowe - `MemoryMappedFileStream`

Zarządzanie pamięcią i energią

Minimalizacja zużycia baterii

czego unikamy?

- kod obciążający CPU (aktywne oczekiwanie),
- działanie w tle.

```
case WM_ACTIVATE:
    active = LOWORD(wParam);
    if(active == WA_INACTIVE)
        // Now in background, so stop using CPU
    else if(active == WA_ACTIVE)
        // Now in foreground, start using CPU (if needed)
    break;
```

Symbian OS - Agenda

1. Historia i charakterystyka
2. Struktura systemu
3. Procesy
4. Zarządzanie pamięcią
5. Model komunikacji

Historia i charakterystyka

SIBO

↓

EPOC

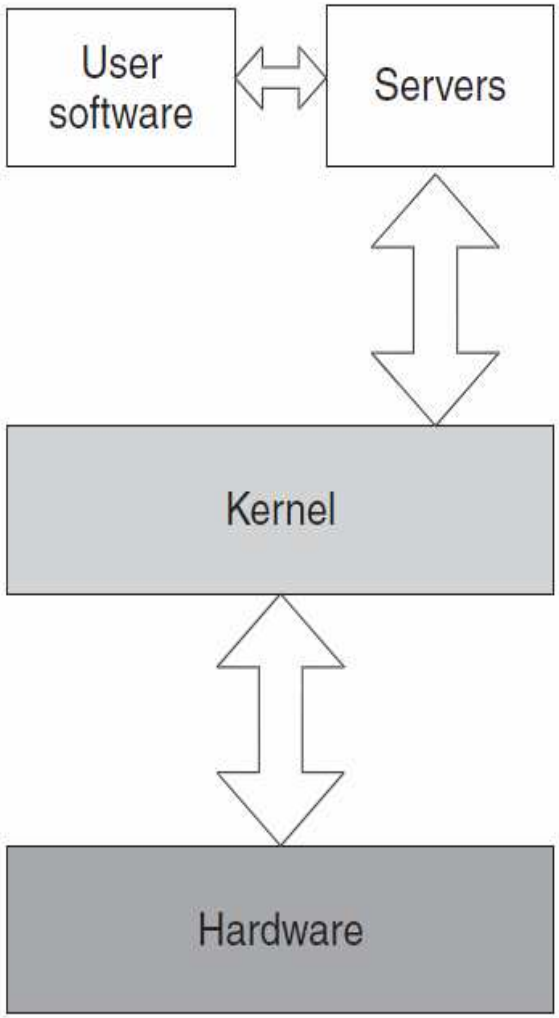
↓

SYMBIAN OS

The original Series 3, now known as the 'Classic'

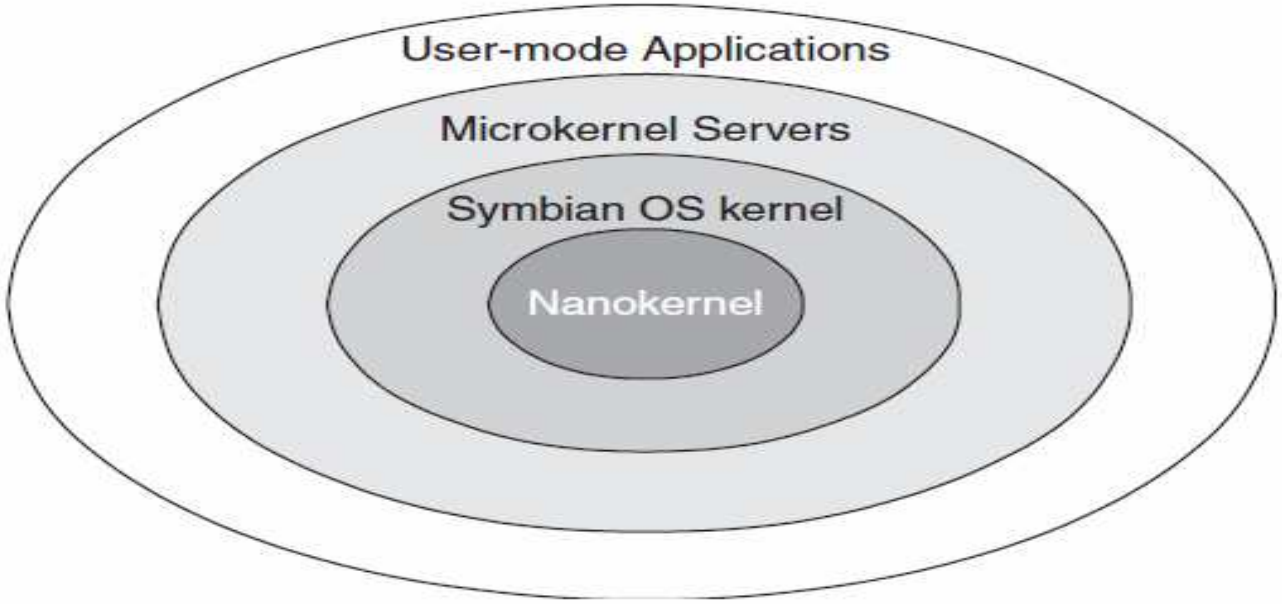
1. **SIBO**
   - Na urządzenia przenośne firmy Psion
   - 16-bit'owy
   - Silnik aplikacji
2. **EPOC**
   - 32-bit'owy
   - napisany w c++
   - zaprojektowany obiektowo
   - serwery usług
   - portowalność
3. **SYMBIAN OS**
   - dziedziczy z SIBO i EPOC
   - dedykowany Smart Phone'om

Struktura systemu - mikrojądro



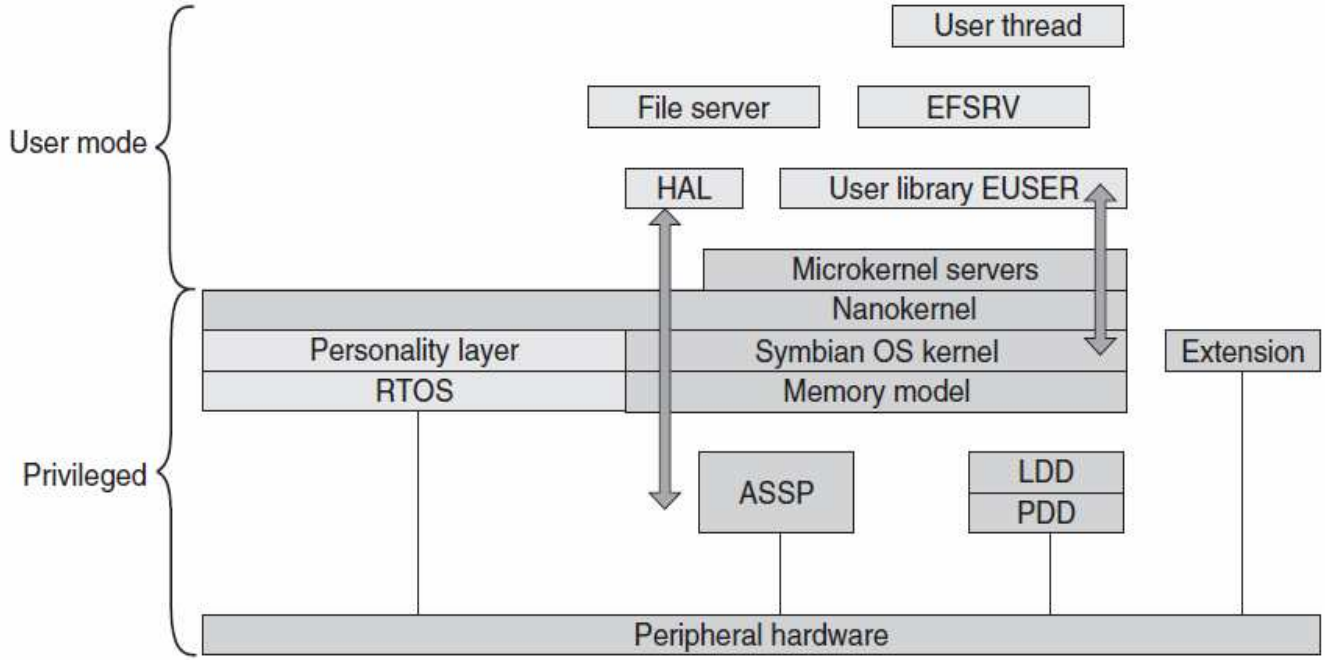
- Mniejsze wymagania pamięciowe
- Lepsze wsparcie dla podłączania nowych urządzeń
- Wolniejszy czas reakcji

Struktura systemu - warstwy



- **Nanokernel:** scheduling i synchronizacja operacji, obsługa przerwań, semafory
- **Microkernel Servers:** Serwery obsługujące konkretne funkcjonalności
- **Symbian OS kernel:** wątki użytkownika, scheduling procesów, przełączanie kontekstu, zarządzanie pamięcią, komunikacja między procesami
- **User-mode Applications:** Aplikacje użytkowe

Struktura systemu - budowa



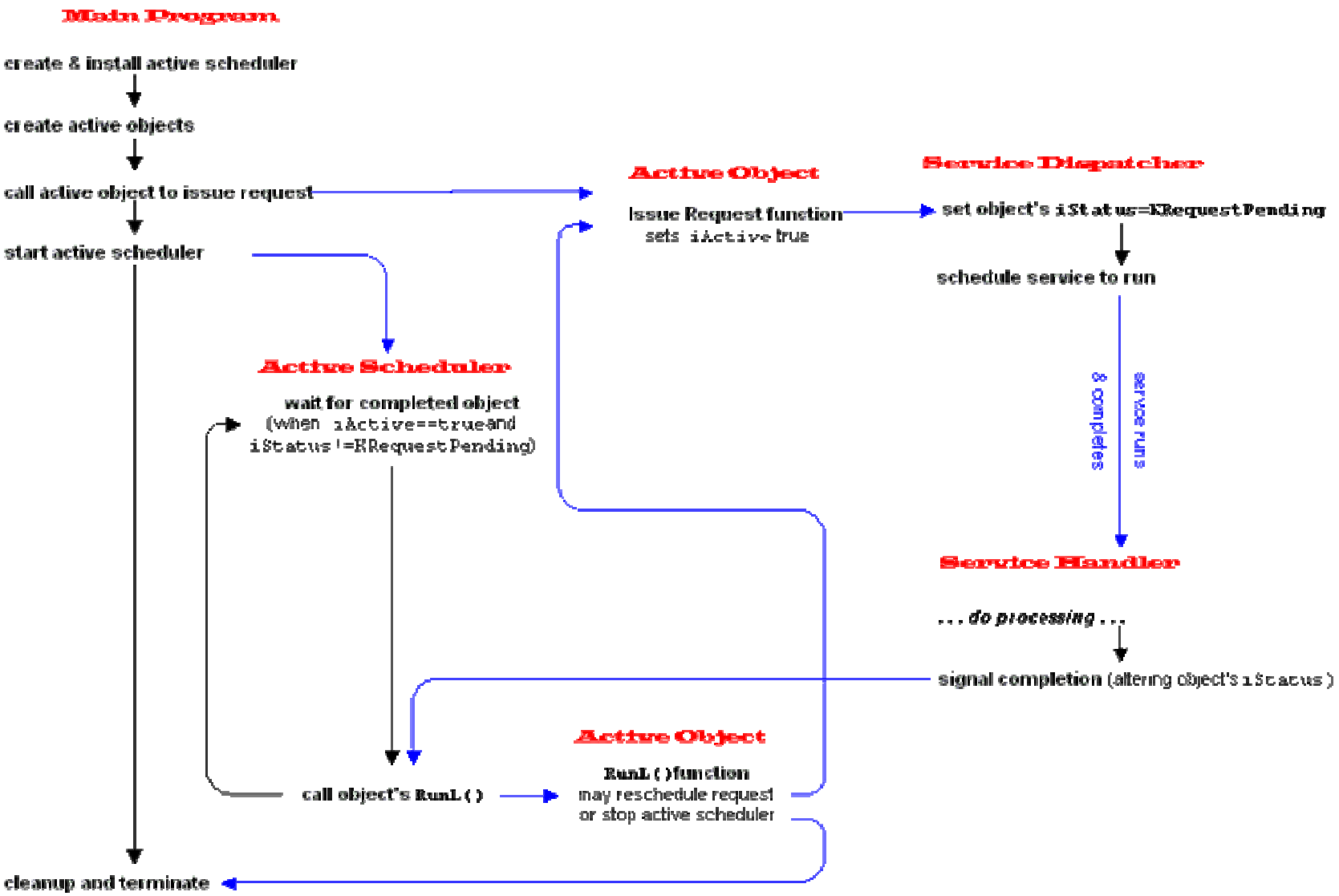
- Sterowniki podzielone na dwie części PDD i LDD
- Application-Specific Standard Product
- Memory model przedstawia organizację urządzeń
- Wyciągnięcie usług czasu rzeczywistego do oddzielnych części: Personality layer i Real-time OS

Procesy - Active object

```
class CExampleActiveObject : public CActive;
    {
public:
    CExampleActiveObject(CExampleServiceProvider* aServiceProvider);
    ~CExampleActiveObject;
    void IssueRequest();
private:
    void DoCancel();
    void RunL();
    TInt RunError(TInt aError);
private:
    CExampleServiceProvider* iServiceProvider ;
    };
```

- RunL - obsługa zdarzenia
- DoCancel - funkcja przerywająca wykonanie RunL
- RunError - funkcja wywoływana gdy RunL zakończy się błędem

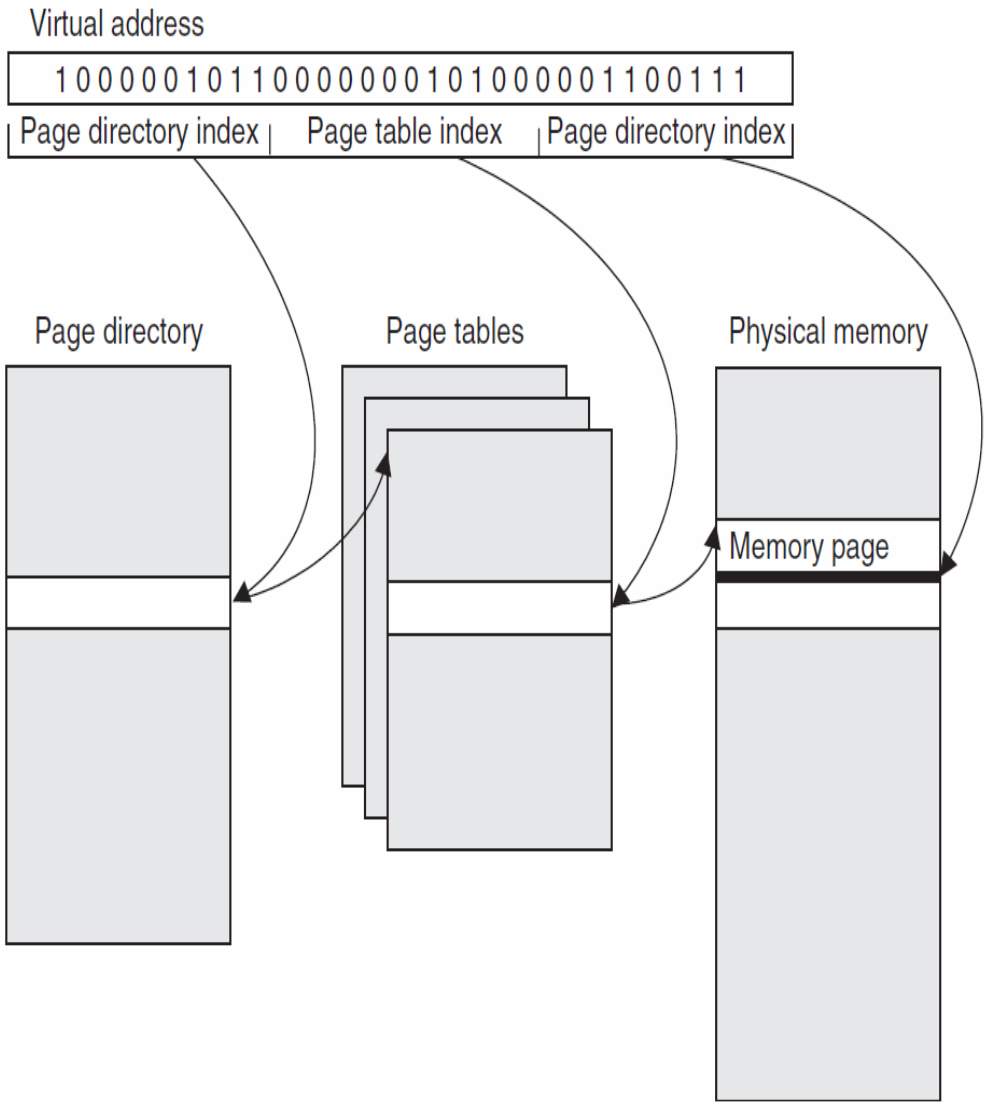
Procesy - Active object cz.2



Procesy - Scheduling i synchronizacja

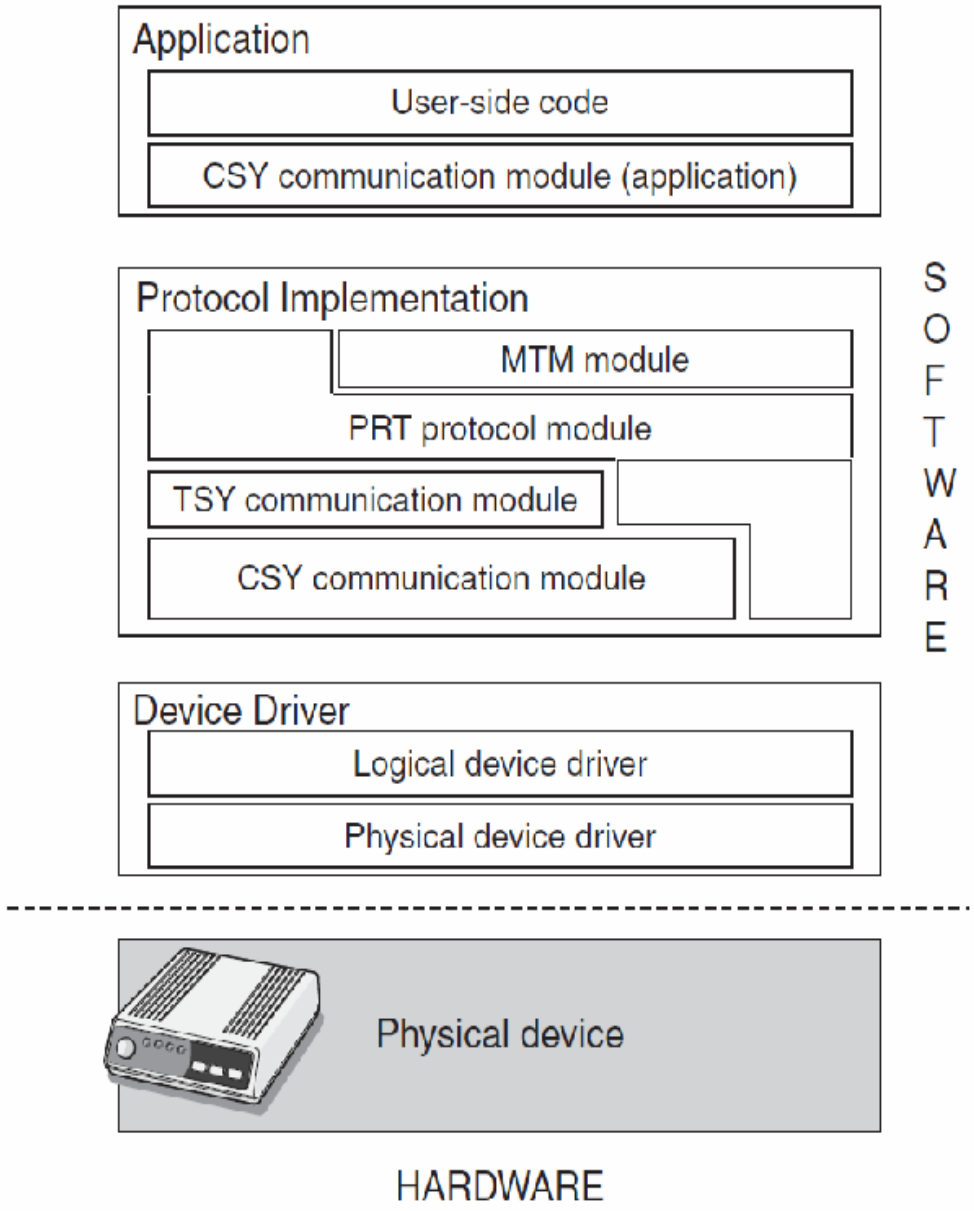
- 64 priorytety
- 64 kolejki
- 64 bitowa maska nie pustości kolejek
- Soft real-time system
- Static,monotonic scheduling
- Dwa poziomy synchronizacji
- Przydzielanie muteksów w kolejności priorytetów
- Dziedziczenie priorytetów

Zarządzanie pamięcią



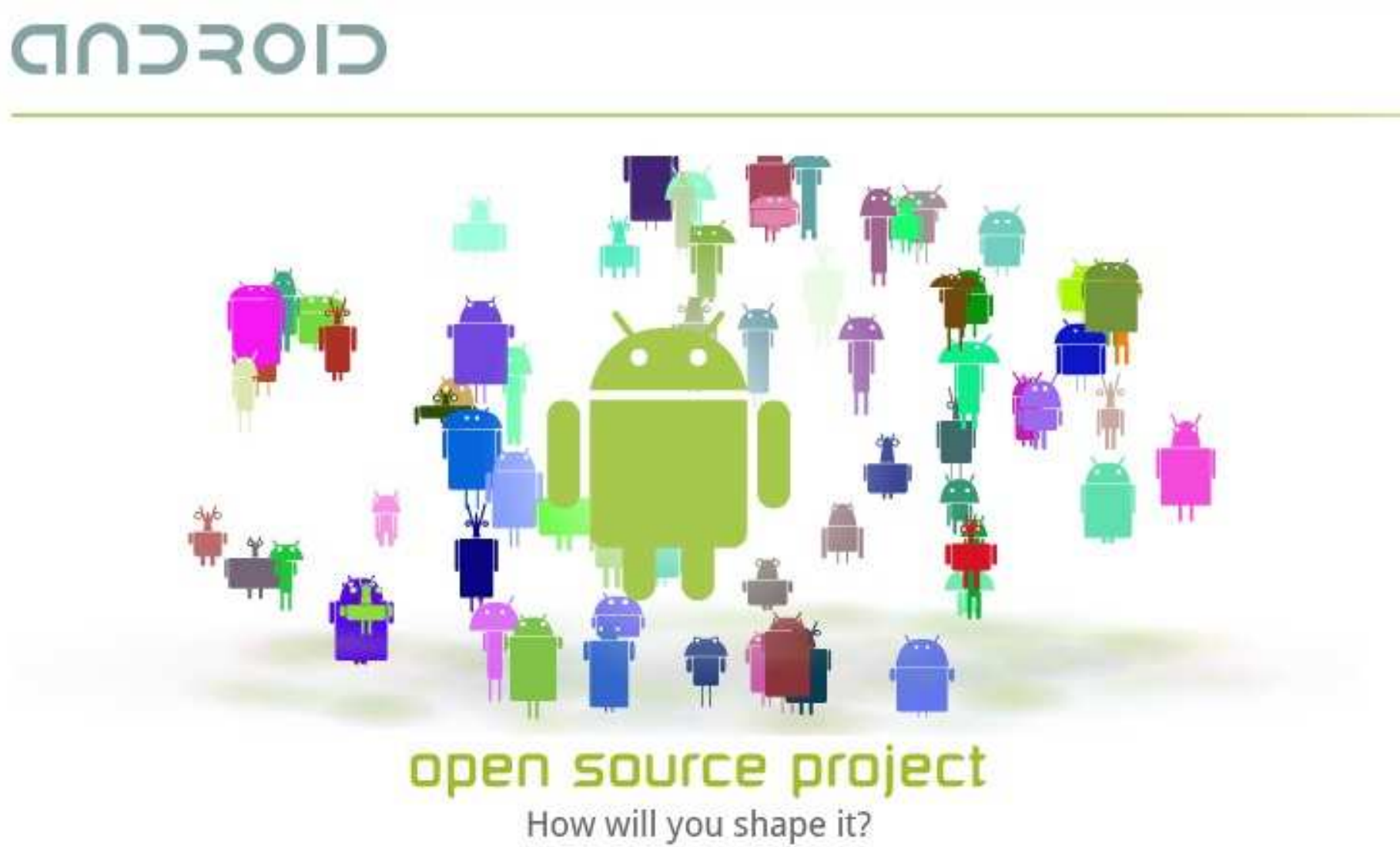
- Translation table-base register
- Strony wielkości 4KB
- Brak memory-swapping'u

Model komunikacji



- Podział na moduły względem funkcjonalności
- Ustandaryzowane API niezależnie od urządzenia

Google Android



Co to Google Android?

- **Open Handset Alliance**
  - Konsorcjum założone 5 listopada 2007, w którego skład wchodzi 34 firm z branży telekomunikacyjnej, sprzętowej i programistycznej, , między innymi: Google, Qualcomm, HTC, Intel, Samsung, Motorola, Sprint czy Texas Instruments.
  - Jego celem jest rozwój otwartych standardów dla urządzeń mobilnych.
- **Android**
  - Platforma opensource dla telefonów komórkowych oparta na systemie Linux, ogłoszona przez Google i inne firmy zrzeszone w Open Handset Alliance.
  - Pozwala programistom tworzyć aplikacje w Javie przy użyciu bibliotek opracowanych przez Google.

Historia Androida


BusinessWeek

HOME INVESTING COMPANIES TECH INNOVATION MANAGING SMALL BIZ

AUGUST 17, 2005
NEWS ANALYSIS
By Ben Elgin

PREMIUM CONTENT
MBA Insider
BW MAGAZINE
Get Four Free Issues
Register
Subscribe
Customer Service
ONLINE FEATURES
Book Reviews
BW Video
Columnists
Interactive Gallery
Newsletters
Past Covers
Philanthropy
Podcasts

Google Buys Android for Its Mobile Arsenal

The search giant quietly acquires the startup, netting possibly a key player in its push into wireless, "the next frontier in search"

In what could be a key move in its nascent wireless strategy, Google (GOOG) has quietly acquired startup Android Inc., BusinessWeek Online has learned. The 22-month-old startup, based in Palo Alto, Calif., brings to Google a wealth of talent, including co-founder Andy Rubin, who previously started mobile-device maker Danger Inc.

lipiec 2005


NETWORKWORLD

News | Blogs & Columns | Clear Choice Tests | Videos | Events

Security | LANs & WANs | VoIP | Infrastructure Management | Wireless | Software | Data Cen

Mobile Data Services | Smart Phones | WiFi | WiMAX | WLAN Management | WLAN Security | Whitepapers

Why Google's GPhone won't kill Apple's iPhone

Mobile phone software, not hardware, is Google's focus

By John Cox, Network World, 10/08/2007

Share/Email | Buzz up! | 1 Comment | Print | Toolshed - IT ASA

Google is putting the final touches on a mobile-phone project, but unlike Apple's iPhone the so-called GPhone is all about software for mobile carriers and mobile advertisers.

What will a GPhone look like? Check out our favorite Google GPhonies

Google's goal is to extend its dominance in online advertising to the emerging mobile advertising market, which is small today but expected to grow dramatically in the years ahead, according to a news report in the *International Herald Tribune*.

HP Polyserve software for SQL Server: Download now

The report says Google is expected to unveil details later this year, and handsets with the software could appear in 2008. But about 30 prototype phones are reported to be 'in the wild' as *Network World* Microsoft Subnet blogger Alex Lewis discovered firsthand last week.

październik 2007


engadget

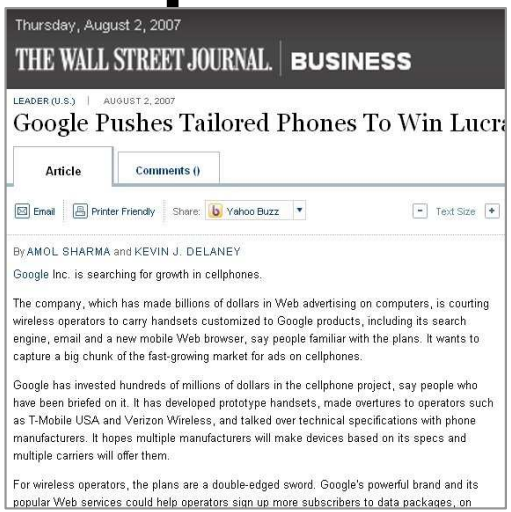
Google attacks: Android at Mobile World Congress

by Sean Cooper, posted Feb 11th 2008 at 9:06AM

Seeing as Mobile World Congress is all about the cellular experience, we weren't too surprised to bump into some Android goodness while here. Texas Instruments had the handset you see above -- from an unknown vendor -- to demo the OS running on an OMAP 3430 at 500 MHz. The interface is dead quick and rather glorious, and while we had no internet connectivity on it, we still were able to have a peek at various applications -- check the gallery for more photos. NEC / Wind River were also showing off an Android

luty 2008

sierpień 2007


Thursday, August 2, 2007

THE WALL STREET JOURNAL. | BUSINESS

LEADER (U.S.) | AUGUST 2, 2007

Google Pushes Tailored Phones To Win Lucra

Article | Comments ()

Email | Printer Friendly | Share: Yahoo Buzz | Text Size

By AMOL SHARMA and KEVIN J. DELANEY

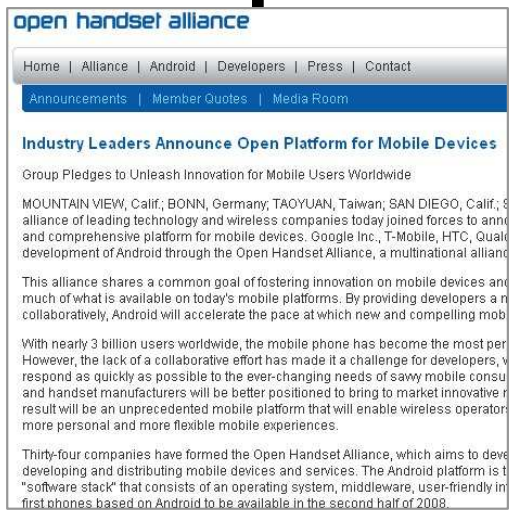
Google Inc. is searching for growth in cellphones.

The company, which has made billions of dollars in Web advertising on computers, is courting wireless operators to carry handsets customized to Google products, including its search engine, email and a new mobile Web browser, say people familiar with the plans. It wants to capture a big chunk of the fast-growing market for ads on cellphones.

Google has invested hundreds of millions of dollars in the cellphone project, say people who have been briefed on it. It has developed prototype handsets, made overtures to operators such as T-Mobile USA and Verizon Wireless, and talked over technical specifications with phone manufacturers. It hopes multiple manufacturers will make devices based on its specs and multiple carriers will offer them.

For wireless operators, the plans are a double-edged sword. Google's powerful brand and its popular Web services could help operators sign up more subscribers to data packages, on

listopad 2007


open handset alliance

Home | Alliance | Android | Developers | Press | Contact

Announcements | Member Quotes | Media Room

Industry Leaders Announce Open Platform for Mobile Devices

Group Pledges to Unleash Innovation for Mobile Users Worldwide

MOUNTAIN VIEW, Calif.; BONN, Germany; TAOYUAN, Taiwan; SAN DIEGO, Calif.;

alliance of leading technology and wireless companies today joined forces to ann

and comprehensive platform for mobile devices. Google Inc., T-Mobile, HTC, Qual

development of Android through the Open Handset Alliance, a multinational allian

This alliance shares a common goal of fostering innovation on mobile devices an

much of what is available on today's mobile platforms. By providing developers a

collaboratively, Android will accelerate the pace at which new and compelling mob

With nearly 3 billion users worldwide, the mobile phone has become the most per

However, the lack of a collaborative effort has made it a challenge for developers,

respond as quickly as possible to the ever-changing needs of savvy mobile consu

and handset manufacturers will be better positioned to bring to market innovative

result will be an unprecedented mobile platform that will enable wireless operator

more personal and more flexible mobile experiences.

Thirty-four companies have formed the Open Handset Alliance, which aims to deve

developing and distributing mobile devices and services. The Android platform is

"software stack" that consists of an operating system, middleware, user-friendly in

first phones based on Android to be available in the second half of 2008.

Cechy Androida (1/3)

- **Ekran**
  - Platforma obsługuje różne rozdzielczości ekranów, w szczególności zarówno VGA jak i mniejsze.
  - Biblioteki umożliwiają wykorzystanie zarówno grafiki 2D jak i 3D w oparciu o OpenGL ES 1.0.
- **Przechowywanie danych**
  - Wykorzystanie Systemu Zarządzania Bazą Danych SQLite
- **Komunikacja**
  - Android wykorzystuje wiele technologii komunikacyjnych, m.in.
    - GSM,
    - CDMA,
    - Bluetooth,
    - EDGE,
    - 3G
    - Wi-Fi.
- **Wiadomości**
  - SMS, MMS,
  - XMPP (wykorzystujące XML, w oparciu o protokół Jabbera).

Cechy Androida (2/3)

- **Przeglądarka internetowa**
  - Android używa wbudowanej przeglądarki opartej o WebKit.
  - WebKit - silnik przeglądarki internetowej rozwijany na zasadach otwartego oprogramowania.
- **Multimedia**
  - Android obsługuje formaty takie jak MPEG-4, H.264, MP3 oraz AAC, JPEG, PNG, GIF.
- **Virtual machine**
  - Oprogramowanie napisane w języku Java jest kompilowane do kodu Dalvika i uruchamiane przez Maszynę Wirtualną Dalvika, będącą wyspecjalizowaną maszyną wirtualną zaprojektowaną specjalnie dla urządzeń mobilnych. Maszyna ta nie jest kompatybilna z JVM.
- **Dodatkowe urządzenia**
  - Android potrafi w pełni korzystać z aparatu fotograficznego oraz kamery, ekranu dotykowego, GPS, kompasu, sensorów motorycznych i akceleratorów grafiki 3D.

Cechy Androida (3/3)

- **Środowisko programistyczne**
  - Zawiera emulator, debugger, diagnostykę użycia pamięci, profiler wydajności oraz wtyczkę dla Eclipse IDE.
- **Architektura**
  - ARM
  - podstawowe wsparcie dla architektury x86 (styczeń 2009)

Architektura systemu Android

Android Runtime

- **Dalvik**
  Wirtualna maszyna Javy, przeznaczona dla urządzeń o ograniczonych zasobach sprzętowych (pamięć i moc procesora).

  - Każda aplikacja jest uruchamiana jako osobny proces, wraz z jej własnym egzemplarzem wirtualnej maszyny Dalvik.
  - Dalvik kompiluje pliki Java do formatu .dex (Dalvik Executable).
  - Dalvik działa w oparciu o jądro Linuxa w zakresie niskopoziomowych funkcjonalności, takich jak zarządzanie procesami, wątkami czy niskopoziomowe zarządzanie pamięcią.

Linux Kernel

- **Linux Kernel 2.6.25 for ARM**
  - bezpieczeństwo, zarządzanie pamięcią, zarządzanie procesami, obsługa sieci i sterowniki
- **Wspiera architekturę ARM od wersji V5T**
- **Do obsługi androida wykorzystuje się szereg rozszerzeń**:
  - alarm, ashmem, binder, power management, low memory killer, kernel degugger, and logger.
- **Obsługa systemu plików FAT32**
- **Obsługa TCP/IP (TCP, UDP, etc)**

- **Obsługiwany sprzęt**
  - Platforma będzie działać na większości środowisk Linuxowych, bazujących na architekturze ARM
  - minimum 128 MB RAM
  - minimum 256 MB pamięci Flash
  - 802.11 b/g Wi-Fi
  - Standardowy interfejs USB, wraz z USB 2.0
  - Bluetooth 2.0
  - Aparat do robienia zdjęć i nagrywania filmów
  - Gniazdo pamięci

Model aplikacji

- **Pakiet android (.apk)**
  - plik zawierający kod i ewentualne zasoby aplikacji,
- **Task**
  - część pakietu, postrzegana przez użytkowników jako „aplikacja”,
- **Proces**
  - proces jądra, w którym uruchamiana jest aplikacja,
  - zwykle cały kod .apk uruchamiany w jednym procesie,
- **Wątek**
  - w ramach procesu jeden lub więcej wątków,
  - Android unika tworzenia własnych dodatkowych wątków, dopóki nie jest to konieczne

Architektura ARM

- ARM = Advanced RISC Machine
- 32-bitową architektura procesorów typu RISC
- stosowana w systemach wbudowanych i systemach o niskim poborze mocy, ze względu na ich energooszczędność
- Procesory ARM są używane między innymi w:
  - dyskach twardych,
  - telefonach komórkowych,
  - routerach,
  - kalkulatorach,
  - zabawkach dziecięcych.

Android Overview

Film do oglądnięcia pod adresem:
http://www.youtube.com/v/eNzkT9usvSY&hl=en&fs=1&autoplay=1

Niezawodność i interaktywność

- rodzaj jądra:
  - jądro monolityczne (tańsze podstawowe operacje, mniej ładowania modułów)
  - mikrojądro (servery, izolacja kodu aplikacji, mniej pamięci na kod)
  - jądro hybrydowe
- izolacja aplikacji w procesie:
  - izolacja ewentualnych błędów,
  - możliwość „ubijania” przeciążających aplikacji,
- real-time software (np. RTOS),
- standardowe mechanizmy (MMU, TLB, stronicowanie),
- wysokopoziomowe framework’i dla tworzenia aplikacji.

Bezpieczeństwo

- rodzaj jądra
- tryb jądra, tryb uprzywilejowany
- podpisywanie aplikacji
- dostęp do systemu wyłącznie przez framework'i (Android)
- mechanizm zezwoleń dla pakietu/aplikacji (Android, Symbian)
- mechanizm zezwoleń dostępu do URI (Android)
- mechanizm nadawania UID dla pakietu (Android)

Łatwość programowania

+ open-source (Android, w przyszłości Symbian)
+ IDE, SDK+emulator
+ dostarczane framework'i (Android, Windows Mobile)
+ wiele wspieranych języków
  + Symbian: C++, Java, Python, VB
  + Windows Mobile: C++ (natywnie), C# (.NET compact framework)

Choć nie zawsze:

- tylko Java i niestandardowa VM (Android)
- podpisywanie aplikacji (płatne)
- wiele wersji (Windows Mobile, Symbian)

Dziękujemy za uwagę