Grzegorz Niemirowski

SINGULARITY – system operacyjny oparty na kodzie zarządzanym

Historia powstania i założenia

- 2003 – start projektu Singularity w Microsoft Research
- marzec 2007 – wersja 1.0
- listopad 2008 – wersja 2.0
- założenie – system niezawodny, przewidywalny, bezpieczny
- wykonanie – rezygnacja z ochrony sprzętowej na rzecz ochrony programowej, wykorzystanie bezpiecznego kodu zarządzanego

Podstawy

- brak rozróżnienia między systemem a środowiskiem uruchomieniowym
- kompilacja do kodu maszynowego przy użyciu kompilatora Bartok
- wspólny kod garbage collectora
- szybkość działania nie jest pierwszorzędnym czynnikiem, ale…

Bezpieczeństwo

- typowanie
- dostęp do pamięci i obiektów
- niezmienniki
- izolacja
- analiza statyczna
- w przyszłości typowany asembler

Jądro

- Singularity wykorzystuje mikrojądro
- prawie w całości napisane w za pomocą bezpiecznego kodu
- 90% stanowi Sing#, z czego 17% nie jest bezpieczna
- pozostałe 10% to asembler, C i C++
- brak współdzielenia pamięci ale jedna przestrzeń adresowa z programową izolacją

Jądro – cd.

- jądro zarządza dostępem do sprzętu (HAL)
- zarządza wątkami
- tworzy i niszczy SIPy
- zarządza kanałami
- posiada cztery schedulery (Rialto, Round-Robin, Laxity i Min)
- wersjonowanie jądra
- zawiera własny garbage collector

Application Binary Interface

- zawiera niskopoziomowe funkcje dostępne dla SIPów
- przesyłanie danych przez kanały
- parametry przekazywane wyłącznie przez wartość, nie przez wskaźniki
- ABI kontroluje ogólnosystemowy niezmiennik izolacji stanu
- funkcje ABI mogą być wstrzykiwane do SIPów w celu wykonania uprzywilejowanych operacji

Software-Isolated Processes

- każdy kod wykonujący się poza jądrem działa w ramach SIPa
- SIPy są wykonywane niezależnie, mogą mieć różne środowiska wykonawcze i GC
- enkapsulują część aplikacji lub systemu
- odseparowują awarię
- zamykają kod, proces nie może dynamicznie ładować ani generować wykonywalnego kodu

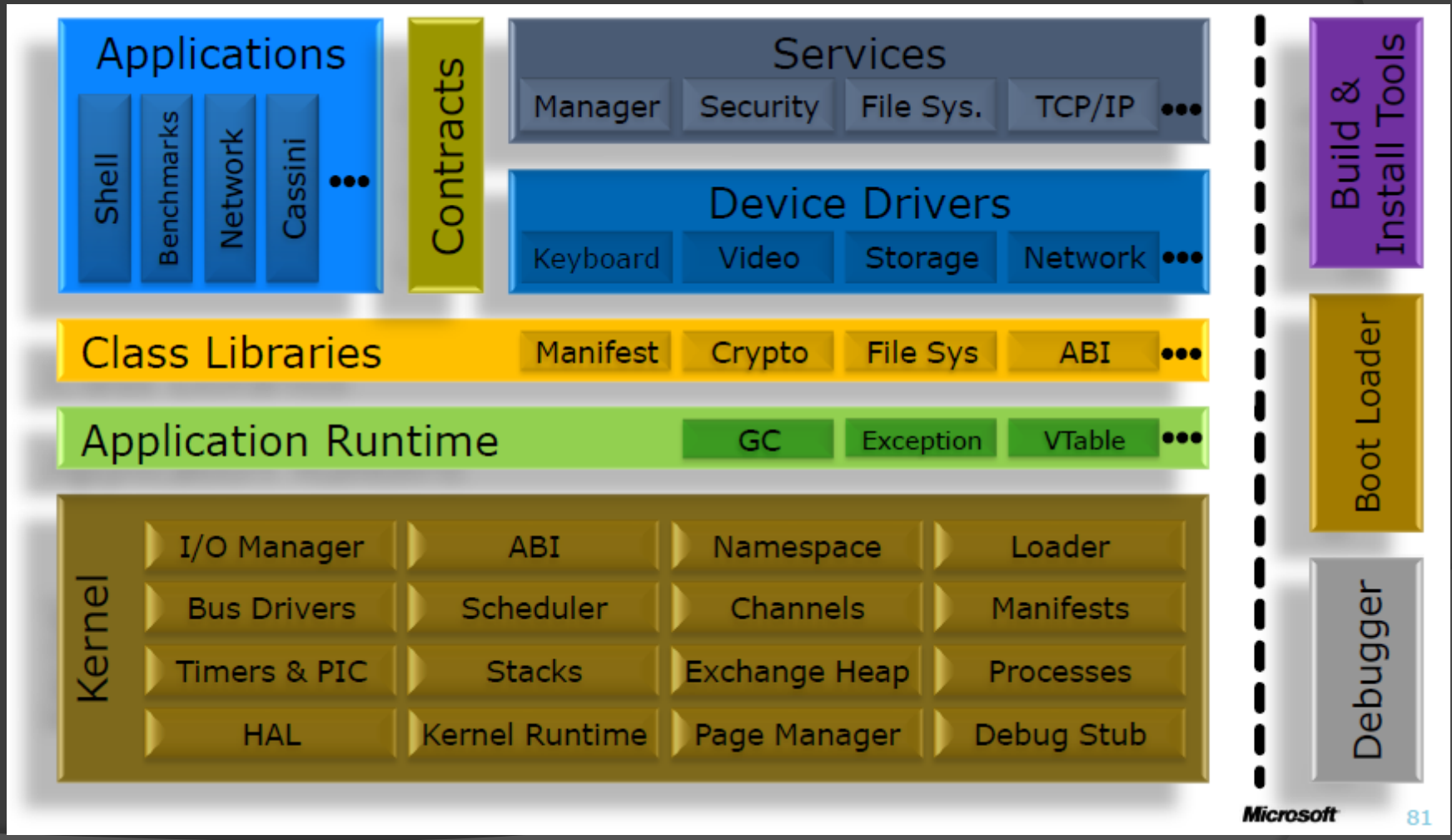
Software-Isolated Processes

- komunikacja odbywa się pomiędzy dwustronne, silnie typowane kanały wysokiego poziomu
- kanały definiowane są przez tzw. kontrakty
- tworzenie SIPa charakteryzuje się małym narzutem, podobnie komunikacja między nimi

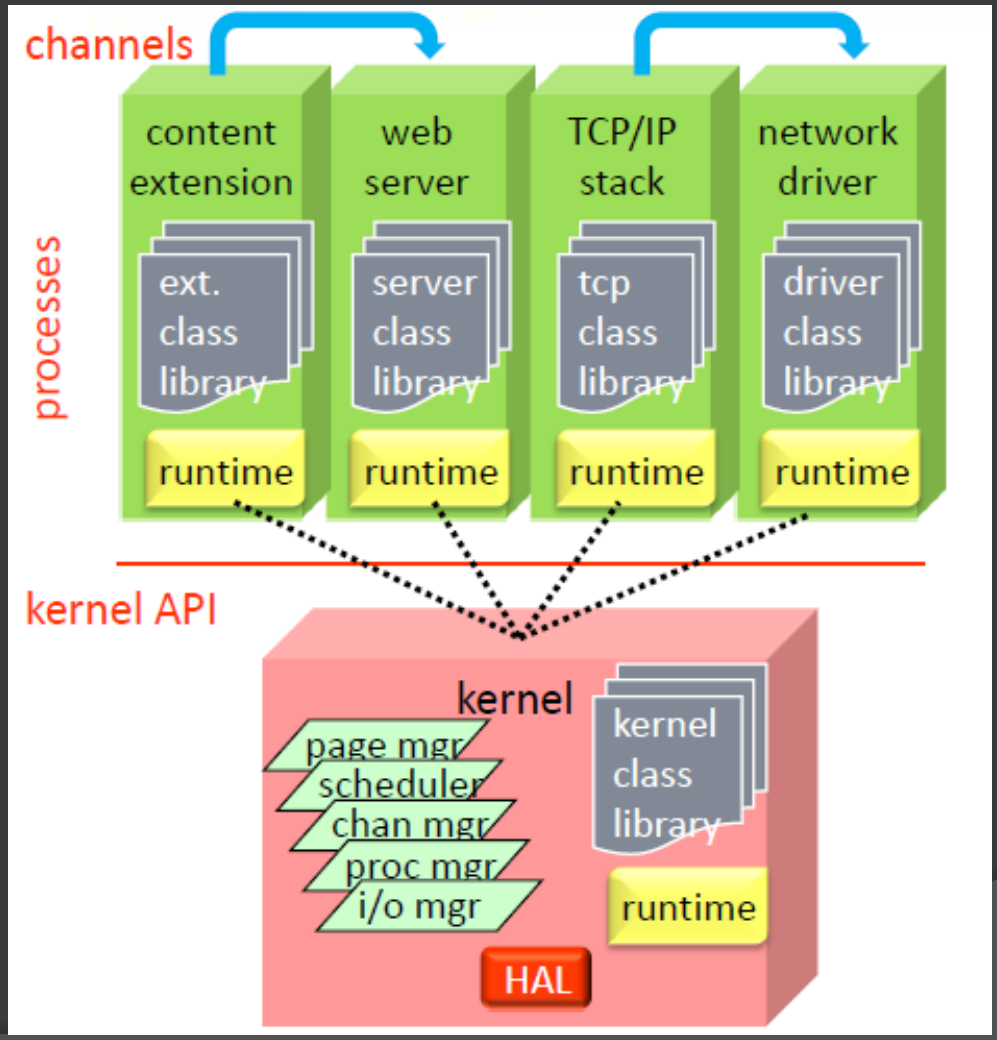
Manifesty

- kod nie może działać bez manifestu
- uruchamiany jest właśnie manifest, nie kod wykonywalny
- manifest informuje o zasobach programu, funkcjonalności oraz zależnościach
- pozwala zweryfikować aplikację
- przy instalacji MSIL jest kompilowany do kodu maszynowego

Architektura



Architektura – cd.



Obsługa sprzętu

- chipset: NVIDIA nForce4 or Virtual PC
- HAL: Legacy or ACPI [PIC or APIC]
- Networks: nForce4 and DEC Tulip
- Storage: nForce4 and PIIX4 ATA
- Video: S3Trio64 and VGA
- Audio: SB16
- Other:PnP BIOS, PCI, 8052 Keyboard

Inne sterowniki

- stos TCP/IP
- FAT12, FAT16, FAT32
- ISO9660
- SMB

Aplikacje

- powłoka
- benchmarki
- aplikacje sieciowe (klient tftp, ping, ipconfig…)
- serwer WWW Cassini
- serwer telnetu
- Serwer poczty z antywirusem

Wydajność

	Cost (CPU Cycles)			
	Singularity	**FreeBSD**	**Linux**	**Windows**
Read cycle counter	8	6	6	2
ABI call	87	878	437	627
Thread yield	394	911	906	753
2 thread wait-set ping pong	1,207	4,707	4,041	1,658
2 message ping pong	1,452	13,304	5,797	6,344
Create and start process	300,000	1,032,000	719,000	5,376,000

Kosz podstawowych operacji

Kompilacja

- kompilacja jest stosunkowo prosta
- wymagany jest .NET Framework 2.0
- uruchomienie skryptu configure.cmd
- kompilacja komendą
  msb Distro\World.proj
- otrzymujemy plik .ISO

Bootowanie

- bootować można z pliku ISO, przez sieć lub z twardego dysku
- najpierw 16-bitowy boot loader
- 32-bitowy boot loader
- HAL
- właściwe jądro
- powłoka
- system można debugować z poziomu WinDbg

Przyszłość

- brak ściśle określonego zastosowania, system eksperymentalny
- być może zostanie użyty wraz z Midori jako podstawa dla Windows 9 lub Windows Mobile 8
- eHome: urządzenia takie jak piloty, odtwarzacze, serwery mediów

Demo

Linki

- Oficjalna strona projektu: http://research.microsoft.com/en-us/projects/singularity/
- Źródła: http://singularity.codeplex.com/Release/ProjectReleases.aspx?ReleaseId=19428

Pytania

Dziękuję za uwagę