

Operační systémy 1

Architektura a historie operačních systémů

Petr Krajča



Katedra informatiky
Univerzita Palackého v Olomouci

Architektura OS

- od operačního systému očekáváme:
 - správu a sdílení procesoru (možnost spouštět více procesů současně)
 - správu paměti (procesy jsou v paměti odděleny)
 - komunikaci mezi procesy (IPC)
 - obsluhu zařízení a organizaci dat (souborový systém, síťové rozhraní, uživatelské rozhraní)
- není žádoucí, aby:
 - každý proces implementoval tuto funkcionalitu po svém
 - každý proces měl přístup ke všem možnostem hardwaru
- \implies jádro operačního systému
- CPU různé režimy práce:
 - privilegovaný (kernel mode)
 - neprivilegovaný (user mode)
- přechod mezi režimy pomocí *systemových volání* (SW přerušení, speciální instrukce, speciální volání)

Architektura jádra (1/2)

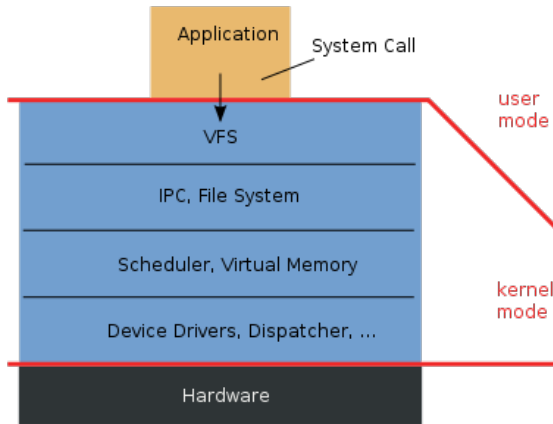
Monolitické jádro

- vrstvená architektura
- moduly
- všechny služby pohromadě \implies lepší výkon
- problém s chybnými ovladači
- Linux, *BSD

Mikrojádru

- poskytuje správu adresního prostoru, procesů, IPC
- oddělení serverů (služeb systému); běžné procesy se speciálními právy \implies bezpečnost
- možnost restartu serverů
- pomalé IPC (přepínání kontextu)
- zavedení systému
- MINIX, QNX

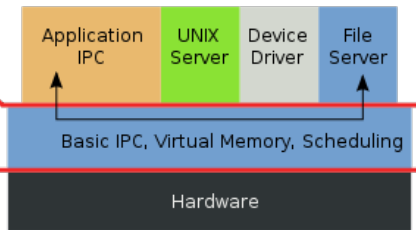
Monolithic Kernel based Operating System



Microkernel based Operating System

user mode

kernel mode



Architektura jádra (2/2)

Hybridní jádro

- kombinují prvky obou přístupů
- část funkcionality v jádře, část mimo
- Windows NT

Exokernel

- řeší jen to nejnútnejší \implies přidělování HW zdrojů
- neposkytuje HW abstrakci \implies knihovny v uživatelském prostoru

Historie operačních systémů

- 1. generace (1945-1955): relátka, elektronky a program zadrátovaný do počítače
- 2. generace (1955-1965): tranzistory, děrné štítky, dávkové zpracování a FORTRAN
- 3. generace (1965-1980): integrované obvody, IBM System/360 a minipočítače PDP
 - multiprogramming
 - timesharing (CTSS – MIT)
 - současná práce více uživatelů, ale pořád prvky dávkového zpracování
 - spooling (sdílení periferií)
 - virtuální paměť; první síť
- 4. generace (1980-současnost): vysoký stupeň integrace; Intel 8080, x86; CP/M, DOS, Windows 95/NT, Unix, GNU/Linux
- řada dalších OS, často se specifickým účelem

Problémy s návrhem OS

- je potřeba mít vizi a cíl
- definovat primitiva, datové abstrakce, izolace
- způsob provádění programů
 - algoritmicky
 - událostmi řízené
- přístup k datům
 - všechno je páska (původní FORTRAN)
 - soubor (UNIX)
 - objekt (Windows)
 - dokument (Web)
- souběžný přístup
- použitelnost SW/designu OS vs. vývoj HW
- správa hardware („nespočetné“ množství)
- přenositelnost, specifické záležitosti
- \implies KISS, jen to nejnútnejší \implies omezení počtu volání; (exec vs. CreateProcess)

Disproporce mezi vývojem HW a SW

ASCI Red

- rok výroby 1996
- určen pro jaderný výzkum
- první počítač, který překonal 1 TFLOPS (1.3 TFLOPS)
- plocha 150 m², spotřeba 850 kW
- cena 46 mil. USD
- odstaven v roce 2006

SONY PS3

- rok výroby 2006
- určen pro domácí zábavu (hry)
- výkon 1.84 TFLOPS
- velikost 325 mm × 98 mm, 5 kg, spotřeba 380 W
- cena 499/599 USD

CP/M

- pro procesory Intel 8080, 8085, Zilog 80
- basic input/output system, basic disk operating system (rezidentní)
- console command processor
- jednoduchý přístup k diskům (disketám)

MS-DOS

- navržen pro procesory Intel 8088, 8086
- jednouživatelský, jednoúlohový
- omezené možnosti práce s pamětí (problém s ovladači)
- více paměti přes rozšíření \implies DOS Protected Mode Interface

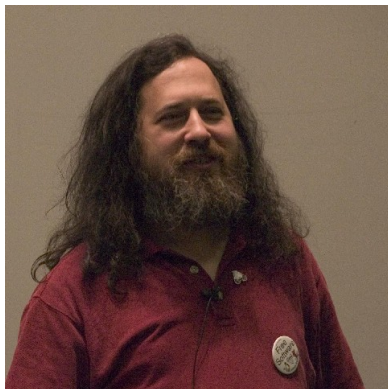
Historie unixů (1/3)

- MULTICS (MULTIplexed Information and Computing Service): MIT + Bell Labs + GE
 - „výpočetní výkon ze zásuvky“ (1964)
 - současná práce více uživatelů
 - jednotná paměť: výlučné mapování souborů do paměti; paměť procesu součástí FS
 - segmentace a stránkování
 - dynamické linkování
- Thompson \implies UNICS (UNiplexed Information Computing Service) – okleštěný MULTICS pro PDP-7
- + Ritchie, Kernighan \implies port na počítače PDP-11
- snažší psaní a portování \implies vznik C
- počítače PDP-11 populární na univerzitách UNIX
- ARPA/DARPA + UCB \implies First Berkeley Software Distribution
 - dále vyvíjené \implies 3BSD, 4BSD
 - stránkování, FS s dlouhými názvy, TCP/IP
 - řada nástrojů vi, csh, překladače

Historie unixů (2/3)

- BSD základ pro další unixy \implies SUN
- současně s BSD vydává v polovině 80.let AT&T \implies System III a System V
- vzniká řada implementací unixu – HP-UX, AIX, SunOS, atd.
- nekompatibility mezi různými verzemi unixů
- snaha o jednotnost
 - System V Interface Definition
 - POSIX (Portable Operating System Interface for UniX; IEEE 1003.1) – nepopisuje jádro, ale funkce std. knihovny a funkcí (průnik funkcionality)
 - Open Software Foundation (OSF) – snaha specifikovat i zbylé části systému (X11); neujal se
- jazyk C standardizován jako ISO i ANSI
- 1987 Tanenbaum vydává MINIX – výukový OS založený na mikrokernelu; kompatibilní s POSIX
- začátek 90.let \implies 386BSD \implies NetBSD, FreeBSD, OpenBSD

Richard M. Stallman



Historie unixů (3/3)

- 1984: Richard Stallman \implies GNU's Not Unix
- pokus o svobodnou reimplementaci Unixu
- General Public License
- vývoj základních nástrojů včetně editoru Emacs, GNU C Compileru
- dlouho chybělo jádro
- pokus použít 4.4BSD nebo vyvinout vlastní (Hurd)
- 1991: Linus Torvalds \implies jádro Linux 0.01
- původně vyvíjeno na MINIXu s GCC
- spojení GNU + Linux \implies GNU/Linux
- možnost použít i jiné kombinace GNU/Hurd, GNU/kFreeBSD
- řada aplikací portována na GNU/Linux \implies de facto standard
- 45 let unixů, Magazín KI č. 2, 2014

Základní vlastnosti unixů (1/2)

- od začátku počítá s víceuživatelským přístupem
- počítá se spolupracujícími uživateli
- počítá se zkušeným uživatelem (nejlépe programátorem)
- snaha být jednoduchý, elegantní, důsledný \implies např. všechno je soubor; (textový soubor \implies protokoly)
- snaha omezit redundanci
- možnost komponovat věci do větších celků
- transparentnost \implies debugování
- není jeden způsob, jak dělat věci správně
- Eric S. Raymond: The Art of Unix Programming

Základní vlastnosti unixů (2/2)

Rozhraní v unixu

- vrstvená architektura a pojící prvky
 - 1 systémová volání
 - 2 volání knihoven
 - 3 systémové nástroje
 - 4 uživatelské aplikace (utility – práce se soubory, filtry, vývojové nástroje, administrace)
- \implies schopnost přežít 50 let

Další unixy (1/2)

BSD

- volnější licence
- monolitické jádro
- základní nástroje vyvíjené společně; adopce GNU nástrojů
- FreeBSD
- NetBSD – podporuje 59 platforem
- OpenBSD – odvozeno z NetBSD; zaměřeno na bezpečnost
- oddělený vývoj – nejedná se o distribuce

Další unixy (2/2)

(GNU) Mach

- „unixový“ mikrokernel
- základní jednotka je task skládající se z vláken
- komunikace přes porty (fronty zpráv) – tasky získávají oprávnění k jednotlivým portům
- podpora paralelismu
- problémy s výkonem (přepínání kontextu, validace zpráv)

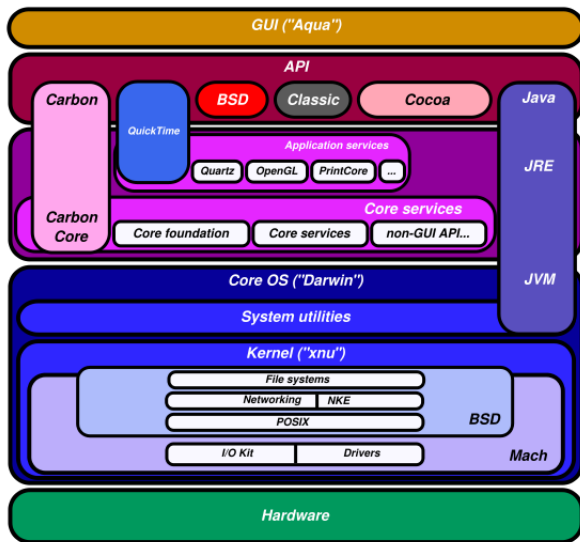
GNU Hurd

- původně zamýšleno jako jádro pro GNU
- k Mach přidává servery (sloužící jako ovladače; autorizace, spouštění aplikací, implementace FS, atd.)
- pokus přeportovat na jiný typ jádra – L4, Coyotos

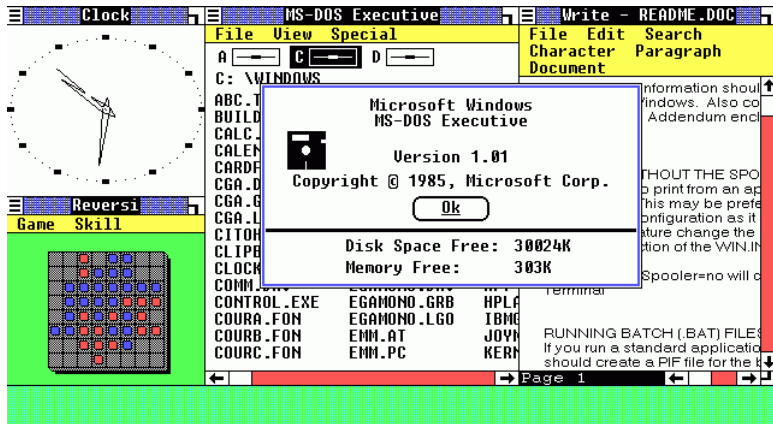
XNU/Darwin

- X is Not Unix
- část Darwinu, část MacOS X
- hybridní kernel
- slučuje jádro Mach a FreeBSD
- z Mach si bere převážně správu procesoru, paměť, IPC
- z BSD bere POSIX API, síťování, souborový systém
- macOS certifikovaný jako UNIX
- rozhraní nad jádrem (*frameworky, kity*)
 - Cocoa (Objective-C)
 - Carbon (zpětná kompatibilita)
 - Quartz 2D, OpenGL

Architektura MacOS X



Zdroj: Wikipedia.org



Historie Windows

Windows 1.0, 2.x

- nádstavba nad MS-DOSem
- kooperativní multitasking
- softwarová virtuální paměť založená na segmentaci

Windows 3.x

- přidávají lepší práci s pamětí
- lepší ovladače a další funkcionalita: práce s fonty, video, síťování (bez TCP/IP), SMB

Windows 9x

- integrace MS-DOSu + GUI
- paměť a přístup k zařízením si řeší po svém (32 bitů)
- preemptivní multitasking
- zpětná kompatibilita
- stále jednovýživatelský OS

Windows NT

- vychází z OS/2
- kompatibilita s ostatními verzemi (W9x, atd.)
- několik obecných principů
 - bezpečnost (certifikace pro armádu)
 - spolehlivost (interní testování)
 - kompatibilita s ostatními systémy (OS/2, POSIX)
 - přenositelnost (HAL)
 - rozšiřitelnost (s ohledem na vývoj HW)
 - výkon
- objektový přístup
- implementovaný v C/C++
- hybridní architektura
 - oddělené procesy pro subsystémy (mikrokernel)
 - spousta funkcionality v jaderném prostoru (monolitický kernel)

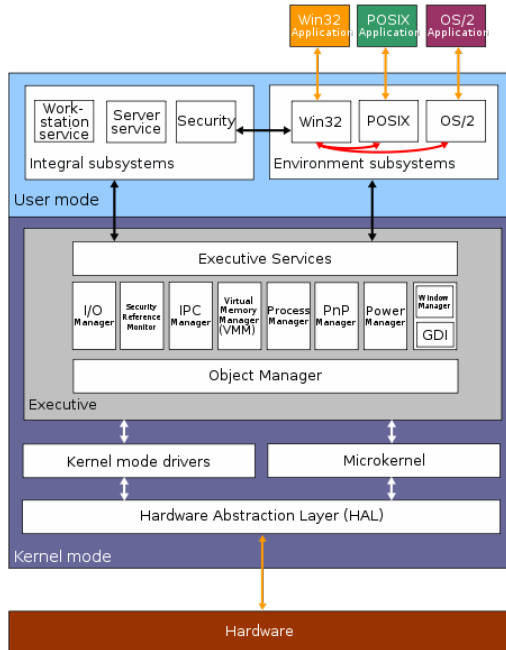
Windows NT: Architektura

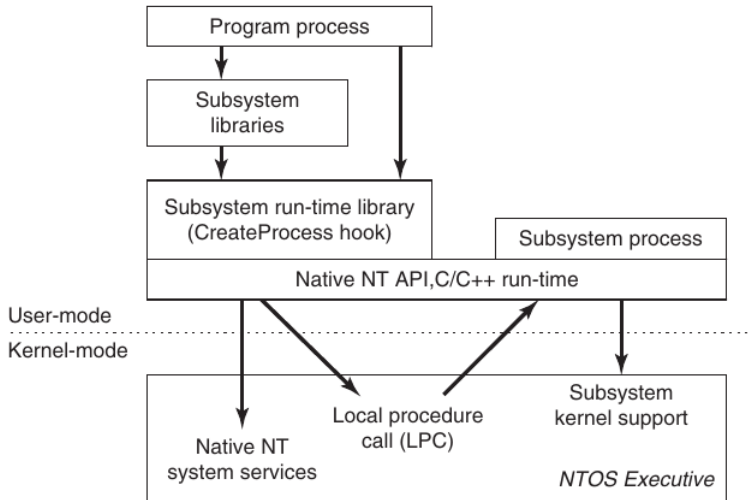
Windows Executive

- klíčová část OS: přes Ntdll.dll poskytuje funkce do uživatelského prostoru
- obsahuje jednotlivé části jádra
 - configuration manager (registry)
 - process thread manager
 - I/O manager
 - security reference manager
 - PnP manager
 - cache manager
 - memory manager
 - object manager
 - a další (mj. Windows, GDI, USER)

Subsystémy

- „pohled“ na funkce poskytované Windows executive (Tan. 794)
- jeden subsystém Windows (csrss.exe), další pro POSIX, OS/2, ...
- další systémové procesy (Session Manager—smss.exe, atd.)





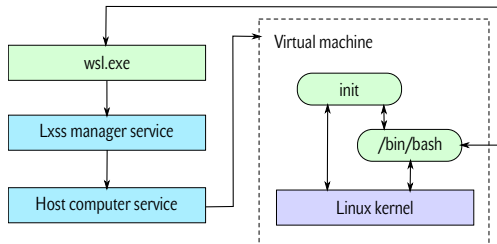
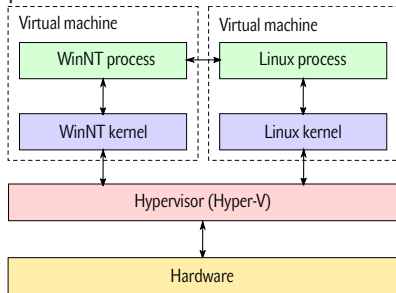
Windows Subsystem for Linux

■ WSL 1

- virtualizace pomocí pico-procesů
- pokud v pico-procesu dojde k systémovému volání, je odchyceno a předáno speciálnímu jadernému modulu
- překlad Linuxových systémových volání na volání Windows
- problematické dodržet 100% kompatibilitu (např. jak pracovat se soubory)

■ WSL 2

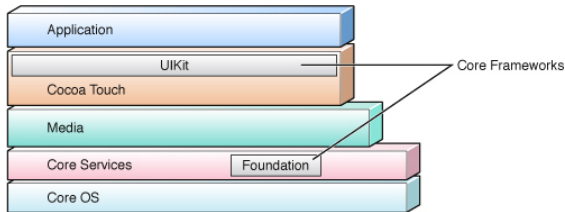
- plnohodnotná virtualizace



- Windows ♥ Linux. Magazin KI, č. 13, 2020.

Android & iOS (1/2)

- jejich jádra vychází z existujících systémů (Linux, resp. Darwin)
- jiný userland



Zdroj: developer.apple.com

Android & iOS (2/2)



Zdroj: Wikipedia.org