

Principy počítačů a operačních systémů

Operační systémy
Úvod, základní architektura

Zimní semestr 2007/2008

Historický vývoj OS

Počítače 1. generace (40. - 50. léta 20. století)

- technologie
 - ♦ relé, elektronky
- programování
 - ♦ strojový kód
 - ♦ propojovací desky, děrné štítky (IBM)
- operační systém
 - ♦ koncept operačního systému neexistuje
 - ♦ konstruktér/správce/programátor v jedné osobě

Historický vývoj OS

Počítače 2. generace (50. - 60. léta 20. století)

- technologie
 - ♦ tranzistory
 - ♦ pevný disk (50 disků, ø61cm, 5MB)
- programování
 - ♦ strojový kód, programovací jazyk FORTRAN
- operační systém
 - ♦ sekvenční spouštění programů, omezení doby běhu
 - ♦ běžící program má plnou kontrolu nad počítačem
 - ♦ objevuje se první time-sharing systém
 - Compatible Time Sharing System (MIT, IBM 7094 + HW mod)

Historický vývoj OS

Počítače 3. generace (60. - 70. léta 20. století)

- technologie
 - ♦ integrované obvody nízké integrace
 - ♦ myš, DRAM, ARPANET
- programování
 - ♦ BASIC, Pascal, SmallTalk, C
- operační systém
 - ♦ TSS, OS/360, MULTICS, UNICS/UNIX
 - ♦ multitasking, spooling, virtual memory & machine
 - ♦ memory mapped files, dynamic linking
 - ♦ opraváři, systémoví a aplikační programátoři

Historický vývoj OS

Počítače 4. generace (od 80. let 20. století)

- technologie
 - ♦ integrované obvody vysoké integrace
 - ♦ osobní počítače (IBM PC, ZX Spectrum, ...)
- programování
 - ♦ BASIC, Pascal, SmallTalk, C
- operační systém
 - ♦ cokoliv mezi zavaděčem programů a distribuovaným, víceuživatelským a víceprogramovým OS
 - ♦ CP/M, MS-DOS, Finder
 - ♦ UNIX, Windows, MacOS, Linux

Moderní operační systém

Hlavní role operačního systému

- extended machine
 - ♦ poskytuje abstrakce nezávislé na hardware
 - ♦ sockety, soubory, ...
- resource manager
 - ♦ přiděluje prostředky (CPU, RAM, bandwidth)
 - ♦ zajišťuje izolaci aplikací a řeší sdílení prostředků

Systémová volání operačního systému

- rozhraní mezi OS a uživatelskými programy
- mechanismus maskován knihovními funkcemi

Moderní operační systém

Architektura

- subsystémy pro různé funkce poskytované OS
 - ♦ rozdělení zodpovědnosti, zjednodušení návrhu
- různá oprávnění pro různé subsystémy
 - ♦ zvýšená odolnost proti “zlým” programům

Dilema při volbě architektury

- oddělení subsystémů vs. efektivita OS
 - ♦ zvýšená režie při komunikaci mezi subsystémy
 - ♦ souvisí s HW mechanismy pro přidělování oprávnění
- kompromis mezi odolností a efektivitou
 - ♦ základní typy architektur operačních systémů

Moderní operační systém

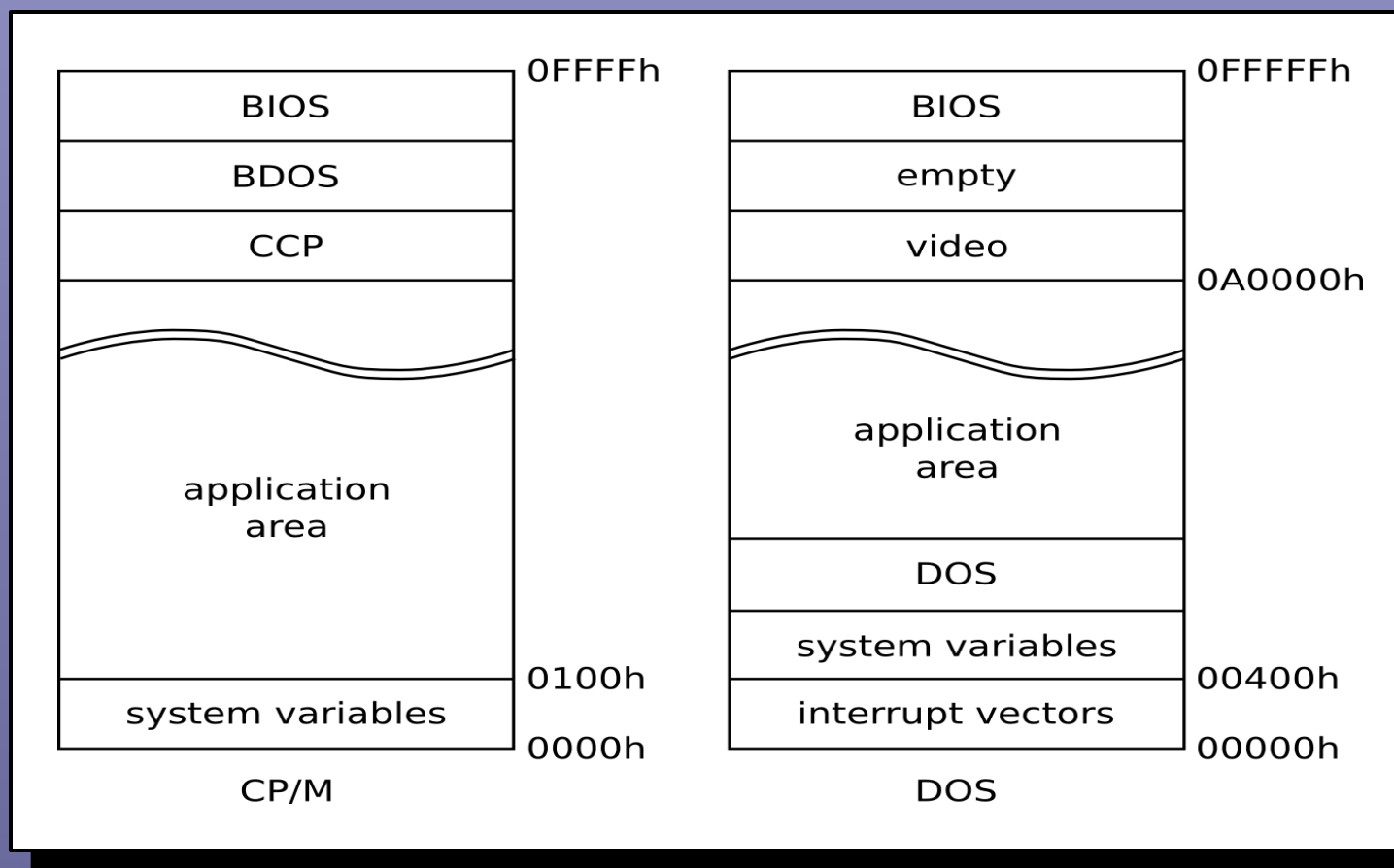
Monolitické systémy

- dávají přednost efektivitě před robustností
- všechny subsystemy mají stejná oprávnění
 - ♦ nevyklučuje modulární design
 - ♦ nízká režie na komunikaci mezi subsystemy
- uživatelské programy mají minimální oprávnění
 - ♦ speciální instrukce na systémová volání mění oprávnění
 - ♦ nepoužívá se u “zavaděčů” typu CP/M, MS-DOS
- běžně používané systémy
 - ♦ Linux, Solaris, Windows, ...

Moderní operační systém

Monolitické systémy

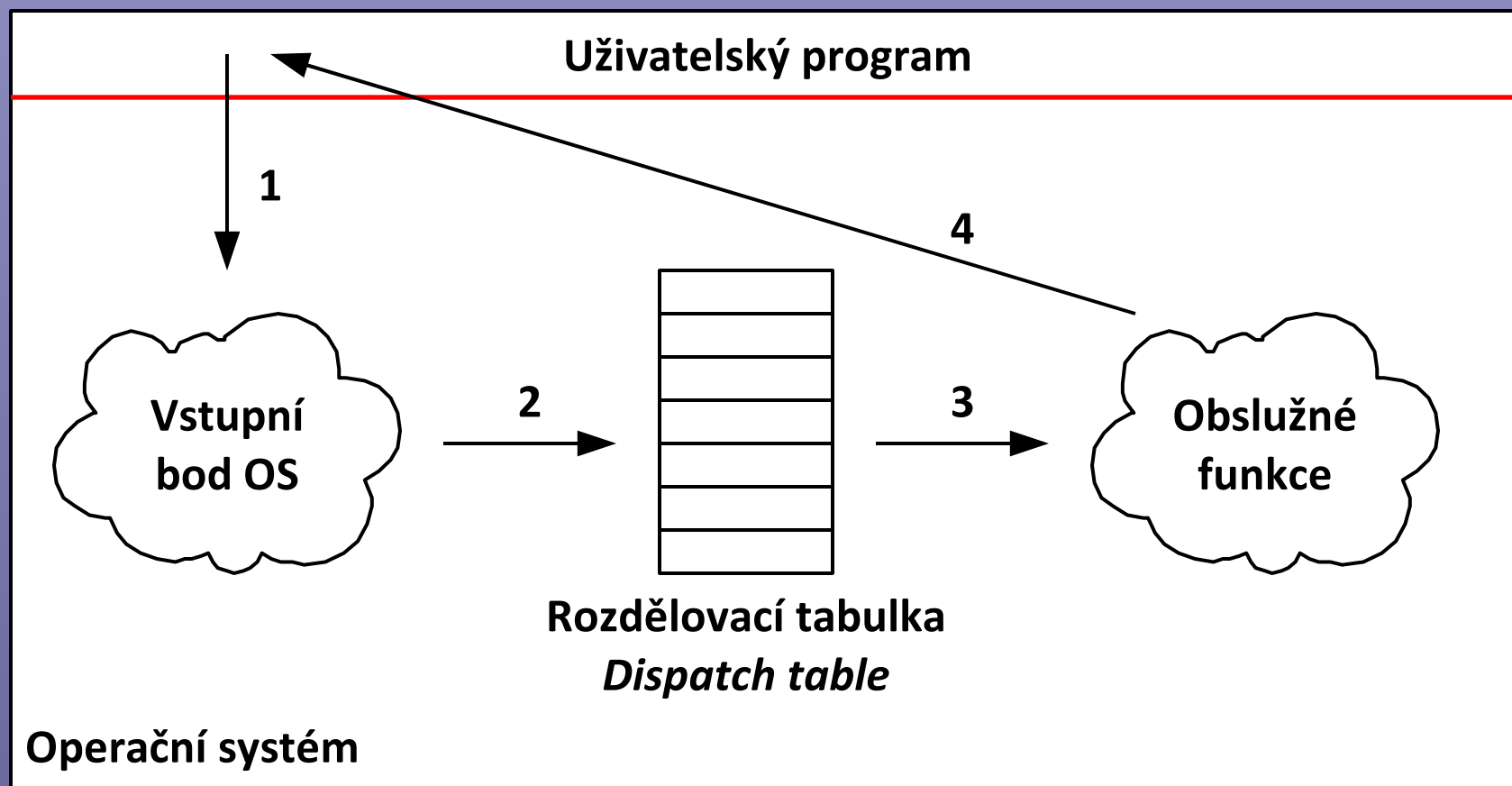
- příklad CP/M a MS-DOS



Moderní operační systém

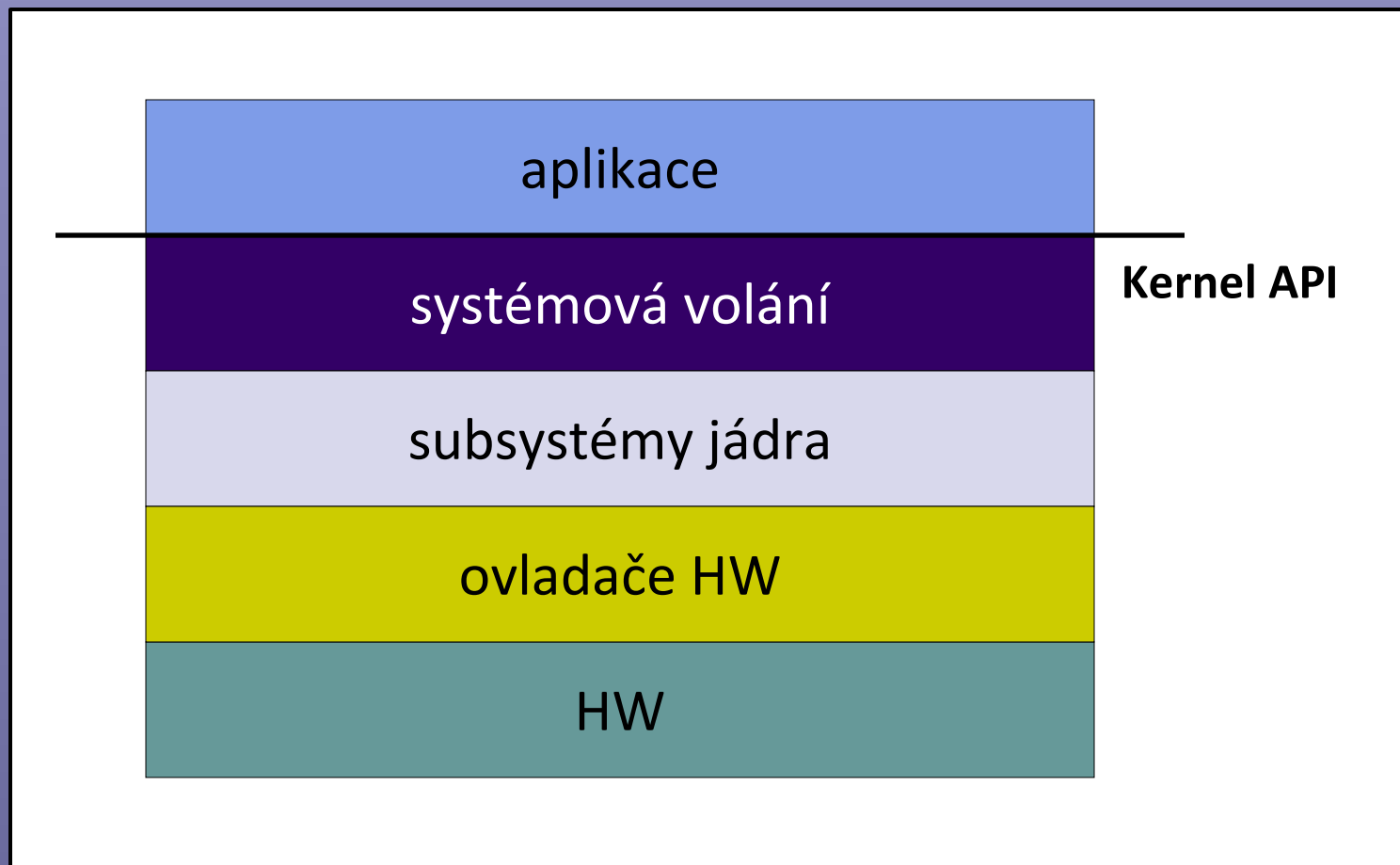
Monolitické systémy

- systémová volání



Moderní operační systém

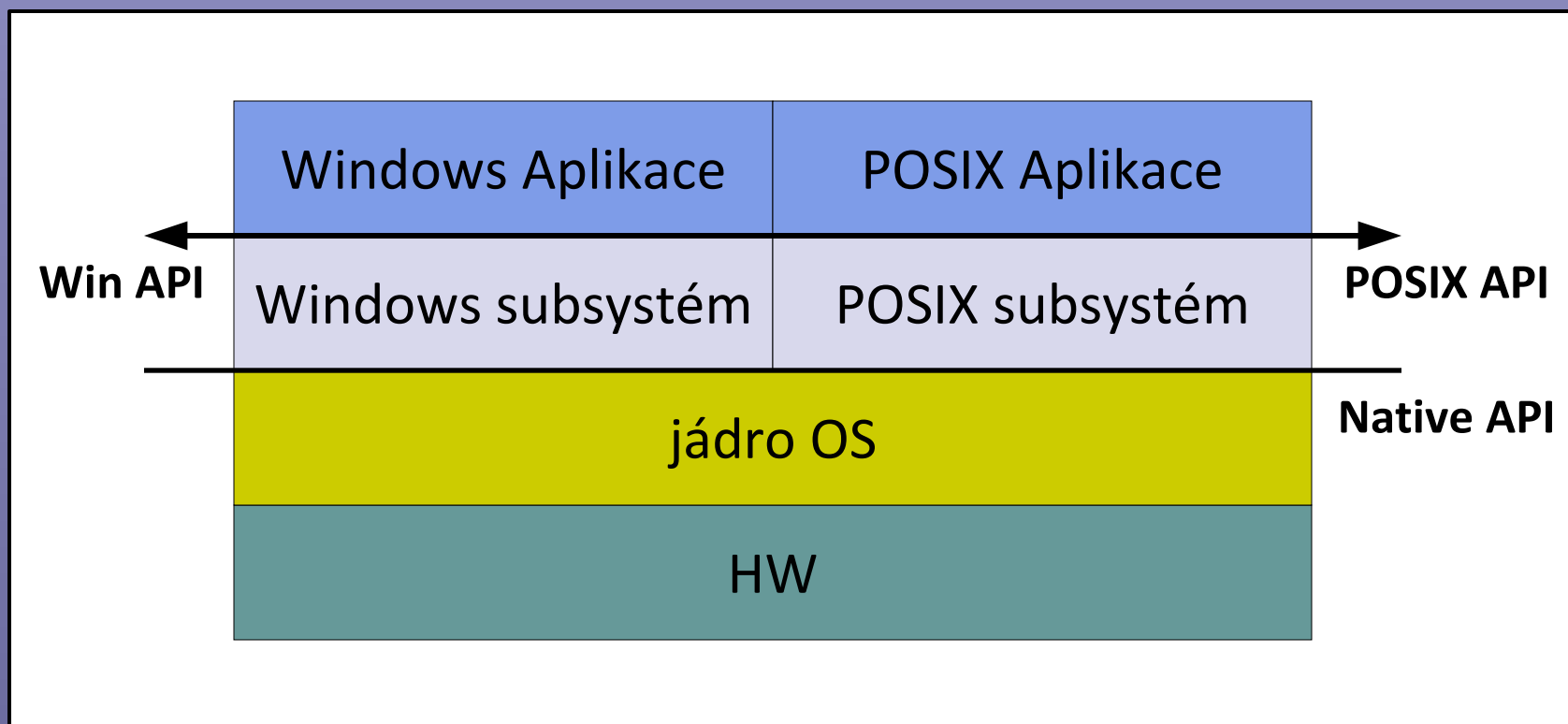
Hlavní logické vrstvy OS



Moderní operační systém

Architektura Windows NT

- více “personalit” nad nativním API



Moderní operační systém

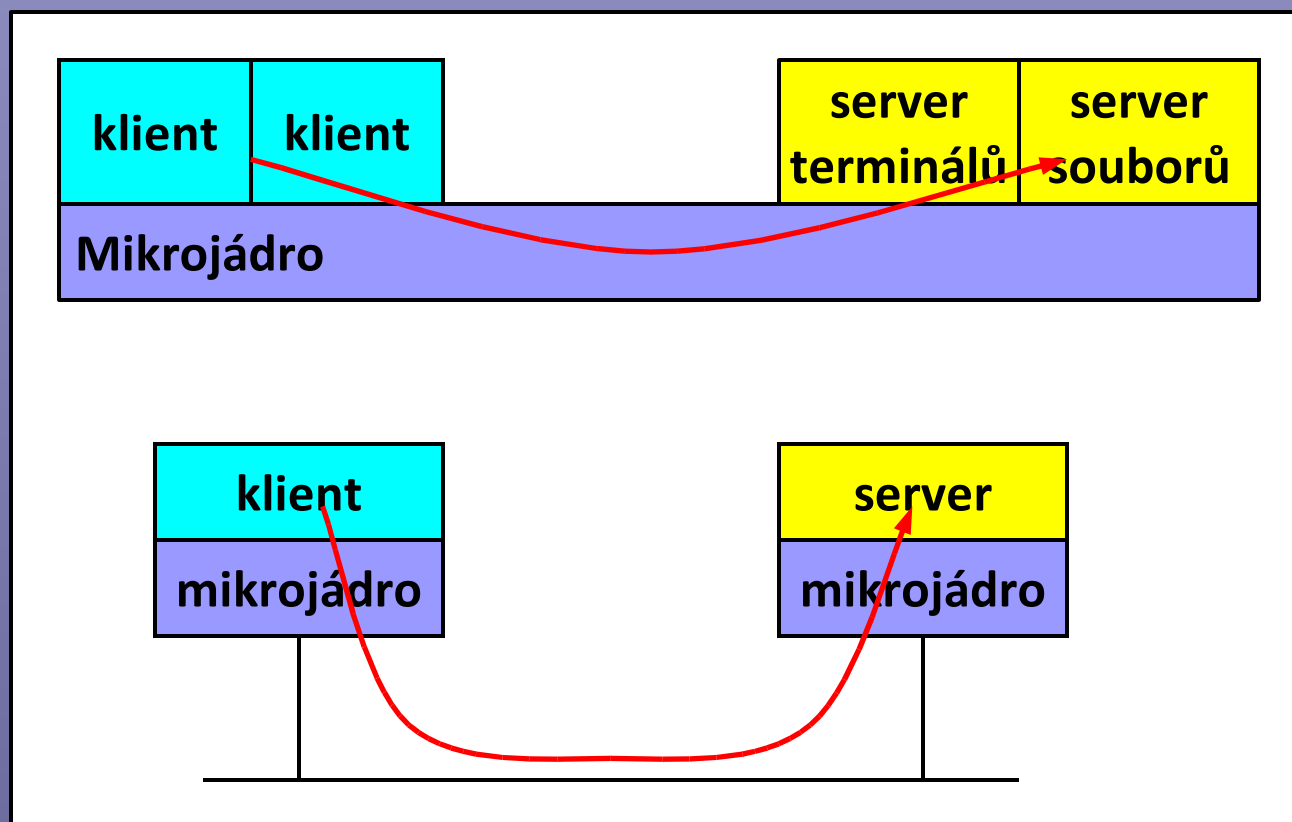
Systémy založené na mikrojádře

- dává přednost robustnosti před efektivitou
- subsystémy mají pouze minimální oprávnění
 - ♦ příliš se neliší od uživatelských programů
 - ♦ výsledkem malé jádro + sada systémových programů
- jádro poskytuje pouze nejnutnější funkce
 - ♦ obsluha přerušení, komunikace mezi programy
 - ♦ komunikace zajišťuje kontrolu oprávnění
- výzkumné a embedded systémy
 - ♦ MACH, Spring, Minix, ..., QNX, L4, Fiasco, ...
- důraz na spolehlivost: Trusted Computing Base

Moderní operační systém

Systémy založené na mikrojádrě

- transparentnost komunikace



Moderní operační systém

Virtuální stroje

- virtualizace HW pro běh více OS
 - ♦ snaha o maximální využití prostředků stroje
- původní podoba
 - ♦ IBM System/360, IBM OS/360
 - ♦ abstrakce HW + plánování běhu virtuálních strojů
- dnešní podoba
 - ♦ abstraktní stroj bez závislosti na HW
 - AS 400, Java, CLR (.Net)
 - ♦ virtualizace na holém HW i v hostitelském OS
 - Xen, VMware, Virtual PC, ..., IBM 390

Moderní operační systém

Virtuální stroje

- IBM 360, IBM 390

