Operačné systémy

Windows

Ing. Martin Vojtko, PhD.

2024/2025

História

História

Year	MS–DOS	MS-DOS based Windows	NT-based Windows	Modern Windows	Notes
1981	1.0				Initial release for IBM PC
1990		3.0			Ten million copies in 2 years
1991	5.0				Added memory management
1992		3.1			Ran only on 286 and later
1993			NT 3.1		
1995	7.0	95			MS-DOS embedded in Win 95
1998		98			
2000	8.0	Me	2000		Win Me was inferior to Win 98
2001			XP		Replaced Win 98
2006			Vista		Vista could not supplant XP
2009			7		Significantly improved upon Vista
2012				8	First Modern version
2013				8.1	Microsoft moved to rapid releases
2014				10	OS as a service

História MS-DOS

- 1980 IBM požiadala o licenciu na BASIC od Microsoft-u.
  - IBM pripravovala nové IBM PC.
  - potrebovala ho vybaviť jednoduchým OS a programovacím jazykom.
  - Microsoft nemal OS preto odporučil CP/M OS od Garyho Kildalla.
  - Kildall sa odmietol stretnúť s IBM. Jedna z najväčších chýb v computer biznise.
  - IBM preto požiadala Microsoft či nevie dodať OS.
  - Bill Gates promptne zareagoval a odkúpil DOS.
- 1981 bolo IBM PC vybavené MS-DOS. (MciroSoft Disk OS)
  - MS-DOS je 16b single-user a one-app-at-time OS.
  - je riadený príkazovým riadkom.
  - ako taký zaberal 8KB.
- 1986 IBM postavilo PC založené na procesoroch Intel 286.
  - MS-DOS v tom čase dosiahol 36KB.
  - stále je single-user a one-app-at-time OS.

História MS-DOS Windows

- 1990 MS-DOS je vybavený grafickým rozhraním - Windows.
  - Prvé verzie neboli úspešné nakoľko nebol HW.
  - Prvá úspešná verzia sa stala Windows 3.0.
- 1993 Windows 3.1
  - Verzia 3.0 a 3.1 bežala na Intel 386 procesoroch.
  - Windows je len grafické rozhranie.
  - Programy sa vykonávajú v spoločnom adresnom priestore.
  - Vzniká Win32 API pre podporu 32b aplikácií.
  - Jadrom je stále 16b MS-DOS.
- Windows 95, 98, Me
  - Doplnená virtual memory, process manažment, multiprogramming.
  - Podpora 32b aplikácií.
  - Chýba security a poriadna izolácia aplikácií a OS.

História Windows NT

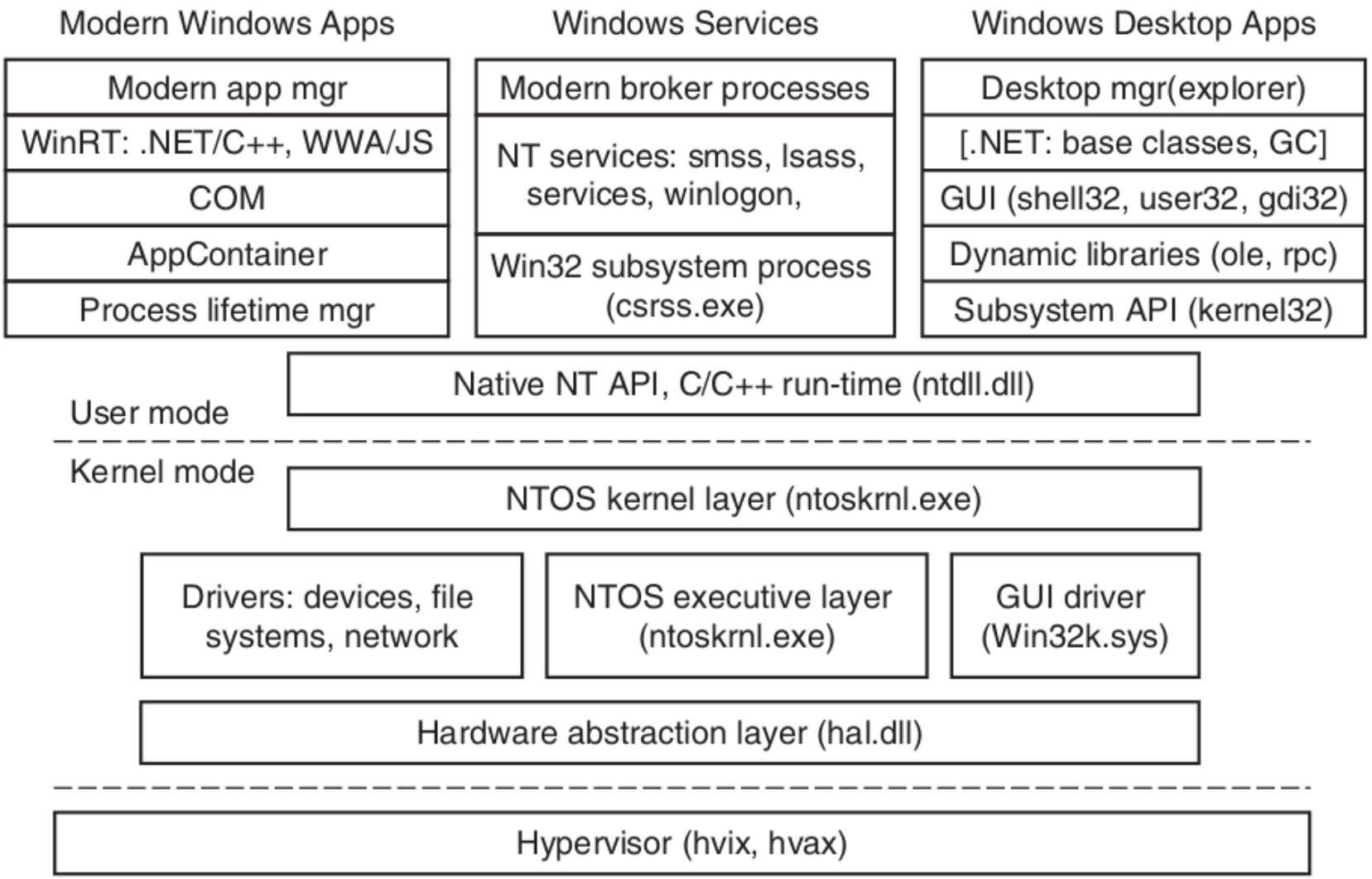
- 1993 NT 3.1 Windows (new technology).
  - Je výsledkom snahy dostať sa na trh serverových aplikácií (kde kraľuje UNIX).
  - Bol napísaný prakticky od nuly s cieľom jednoduchej prenositeľnosti.
  - Oproti Windows 3.1, NT plne podporovalo 32b aplikácie.
  - Žiaľ mal slabú podporu pre 16b aplikácie.
- Vďaka portabilite a kompatibilite s Windows 95. Mnoho spoločností jednoducho prešli na NT.
  - rovnako aj ich dátové centrá.
  - Politika Microsoftu je veľmi jednoduchá. Ak sú zamestnanci zvyknutý na windows tak...
- Windows 2000
  - plug-and-play - no more jumper config
  - network directory - zdieľanie súborov
  - power manažment - úspora batérie notebook-ov

História moderný windows

- 2010 trh sa začína meniť
  - Úspech Apple podnieti zmeny stratégie Windows.
  - Windows sa neúspešne snaží etablovať v segmente múdrych telefónov.
- 2012 Windows 8
  - Snaha o univerzálny OS na všetky zariadenia.
  - Mnoho ľudí neprehltlo chýbajúci štart a preto prichádza 8.1
- 2014 Windows 10
  - Microsoft sa transformuje na spoločnosť predávajúcu služby.

Rozhrania

Rozhrania


Modern Windows Apps
Windows Services
Windows Desktop Apps
Modern app mgr
WinRT: .NET/C++, WWA/JS
COM
AppContainer
Process lifetime mgr
Modern broker processes
NT services: smss, lsass, services, winlogon,
Win32 subsystem process (csrss.exe)
Desktop mgr(explorer)
[.NET: base classes, GC]
GUI (shell32, user32, gdi32)
Dynamic libraries (ole, rpc)
Subsystem API (kernel32)
Native NT API, C/C++ run-time (ntdll.dll)
User mode
Kernel mode
NTOS kernel layer (ntoskrnl.exe)
Drivers: devices, file systems, network
NTOS executive layer (ntoskrnl.exe)
GUI driver (Win32k.sys)
Hardware abstraction layer (hal.dll)
Hypervisor (hvix, hvax)

Rozhrania

- Windows poskytuje množstvo knižníc, ktoré slúžia ako rozhrania pre množstvo služieb OS.
- Mnohé knižnice existujú len kvôli spätnej kompatibilite.
- Rozhranie Windows funguje ako množina bežiacich služieb.
- User aplikácie komunikujú s týmito službami prostredníctvom RPCs (remote-procedure-calls).
- Aplikácie nekomunikujú priamo s jadrom na to existujú služby.
- Windows 8 už nepodporuje POSIX.
- Windows 10 rozšírený o WSL (windows subsystem for Linux) ako nepriama podpora POSIX.

Rozhrania

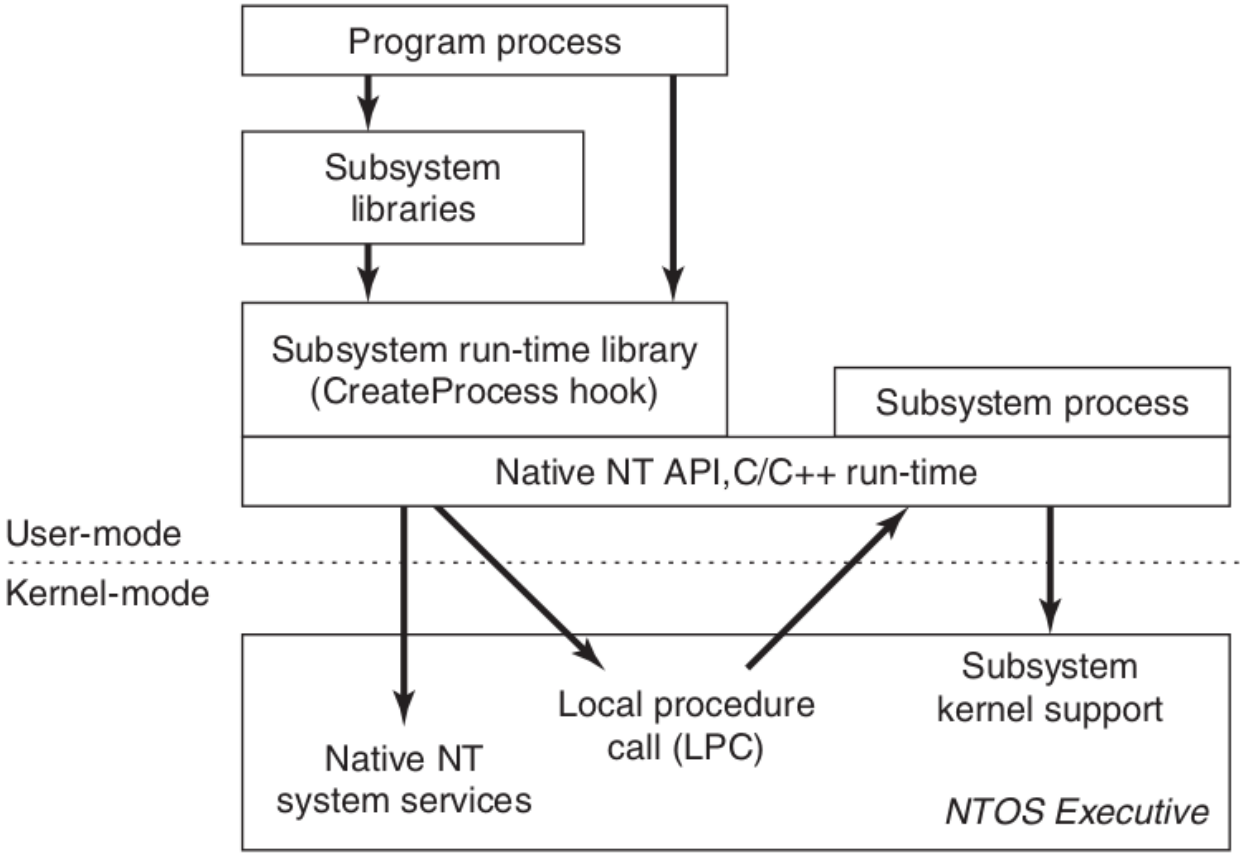
- Stavebným kameňom Windows je stále Win32 NT rozhranie.
  - Podpora vývoja aplikácií definovaná MSDK (MS Development Kit).
  - Vývoj bežných aplikácií.
- Modernejšie rozhrania ako .NET a WinRT v mnohých prípadoch dedia od Win32.
  - Bežná WinRT aplikácia neumožňuje definovať vlákna. Všetko je obstarávané API.
  - Modern Windows definuje úlohy, ktoré môže aplikácia vytvárať.
  - Práca s pamäťou je skrytá pred programátorom.
  - Aplikácie inštalované cez appStore
  - Aplikácie spúšťané cez AppContainer.
    - Je to sand-box, ktorého úlohou je separovať kód aplikácie od okolia.
    - Ak aplikácia chce komunikovať s okolím je to možné len cez tzv. broker process.
    - Broker je process s prideleným prístupom k jednotlivým prostriedkom systému.

Rozhrania - native NT

- Väčšina prostriedkov vo Windows je skrytých jadrom.
  - Windows vníma všetky prostriedky ako **objekty**.
  - Aplikácie získavajú prístup k objektom cez **handle**.
- Handle sa nedá odovzdať inému procesu priamo.
- K objektu prislúcha aj tzv. security descriptor definujúci oprávnenia.

Object category	Examples
Synchronization	Semaphores, mutexes, events, IPC ports, I/O completion queues
I/O	Files, devices, drivers, timers
Program	Jobs, processes, threads, sections, tokens
Win32 GUI	Desktops, application callbacks

Rozhrania - native NT

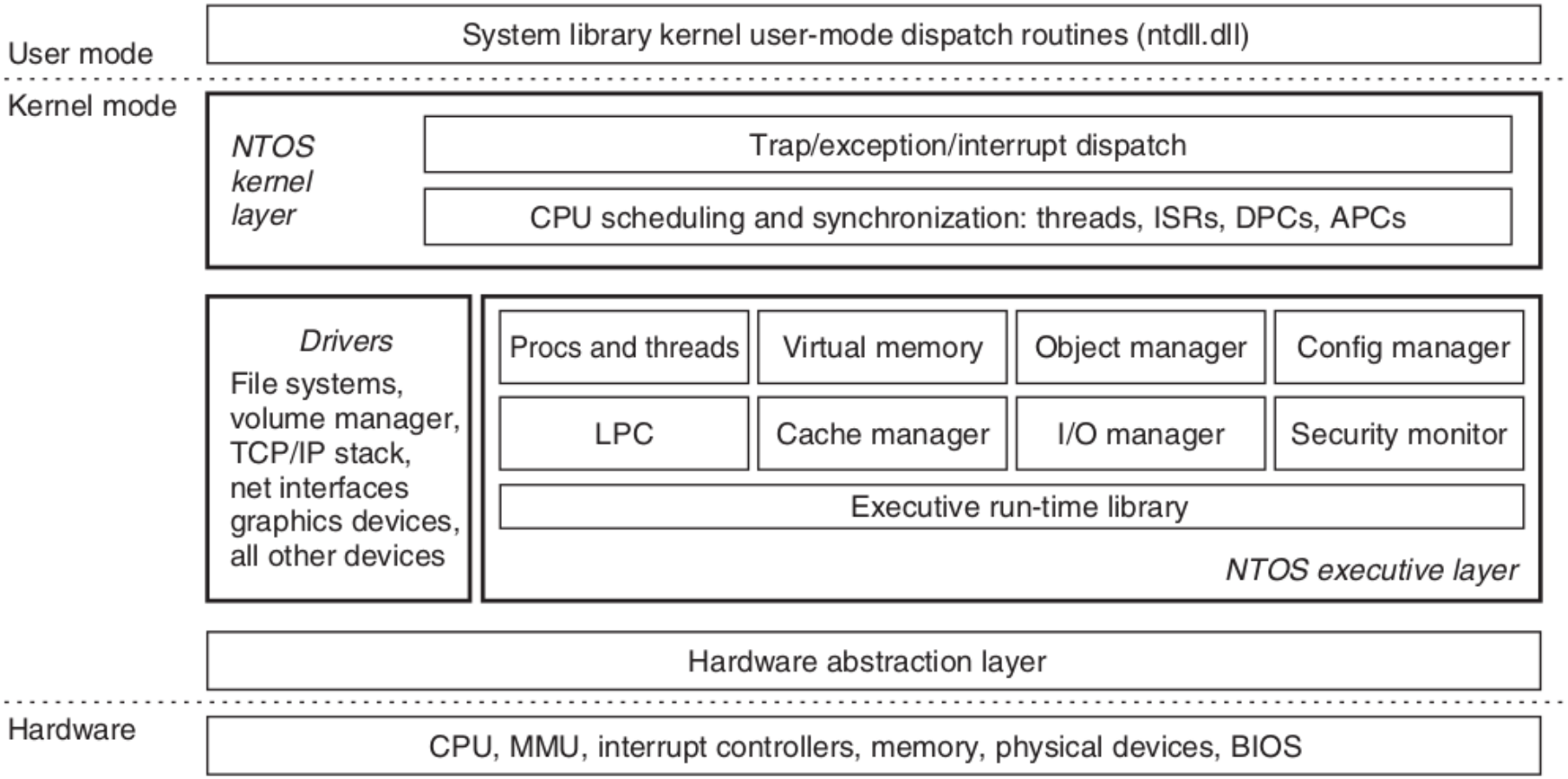


Rozhrania - Win32 API

- Je voľne dostupné a podrobne dokumentované API Windows.
- API je implementované ako knižnica procedúr.
- Všeobecne Win32 API je obalom natívneho NT subsystému.
- API sa s časom len rozširuje a len sporadicky sa staršie časti menia.
- API je veľmi komplexné.

Kernel

Kernel

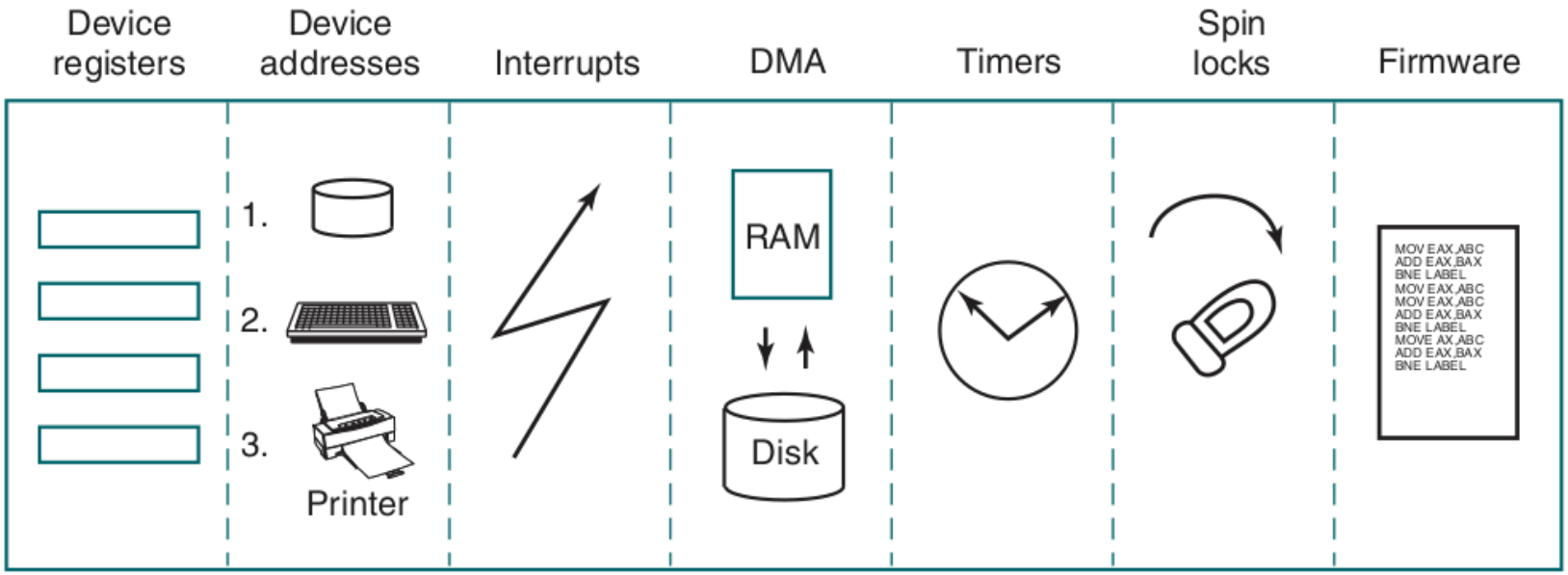

User mode
System library kernel user-mode dispatch routines (ntdll.dll)
Kernel mode
NTOS kernel layer
Trap/exception/interrupt dispatch
CPU scheduling and synchronization: threads, ISRs, DPCs, APCs
Drivers
File systems, volume manager, TCP/IP stack, net interfaces graphics devices, all other devices
Procs and threads
Virtual memory
Object manager
Config manager
LPC
Cache manager
I/O manager
Security monitor
Executive run-time library
NTOS executive layer
Hardware abstraction layer
Hardware
CPU, MMU, interrupt controllers, memory, physical devices, BIOS

Kernel - Hypervisor

- Je bežné, že jadro OS je vykonávané priamo nad konkrétnym HW
- Hypervisor je SW, ktorý slúži ako nadstavba nad HW, ktorá umožňuje zdieľať jeden HW viacerým OS.
  - je to vlastne virtuálny HW.
  - stará sa o striedanie jednotlivých OS na reálnom HW.
  - Na to aby Hypervisor fungoval je nutné aby HW podporoval virtualizáciu.
- Windows natívne podporuje virtualizovanie pomocou hypervisor-a.
  - Ak máte zapnutú virtualizáciu tak v podstate viete spúšťať separátne inštancie Windows.
  - Je možné spúšťať rôzne verzie Windows.

Kernel - HW abstraction layer (HAL)

- HAL je vrstva kódu pozostávajúca z C a asembléru.
- Obsahuje tak špecifický kód, že je vždy nutné ho prepísať pri portovaní na nový HW.



Kernel - Ovládače

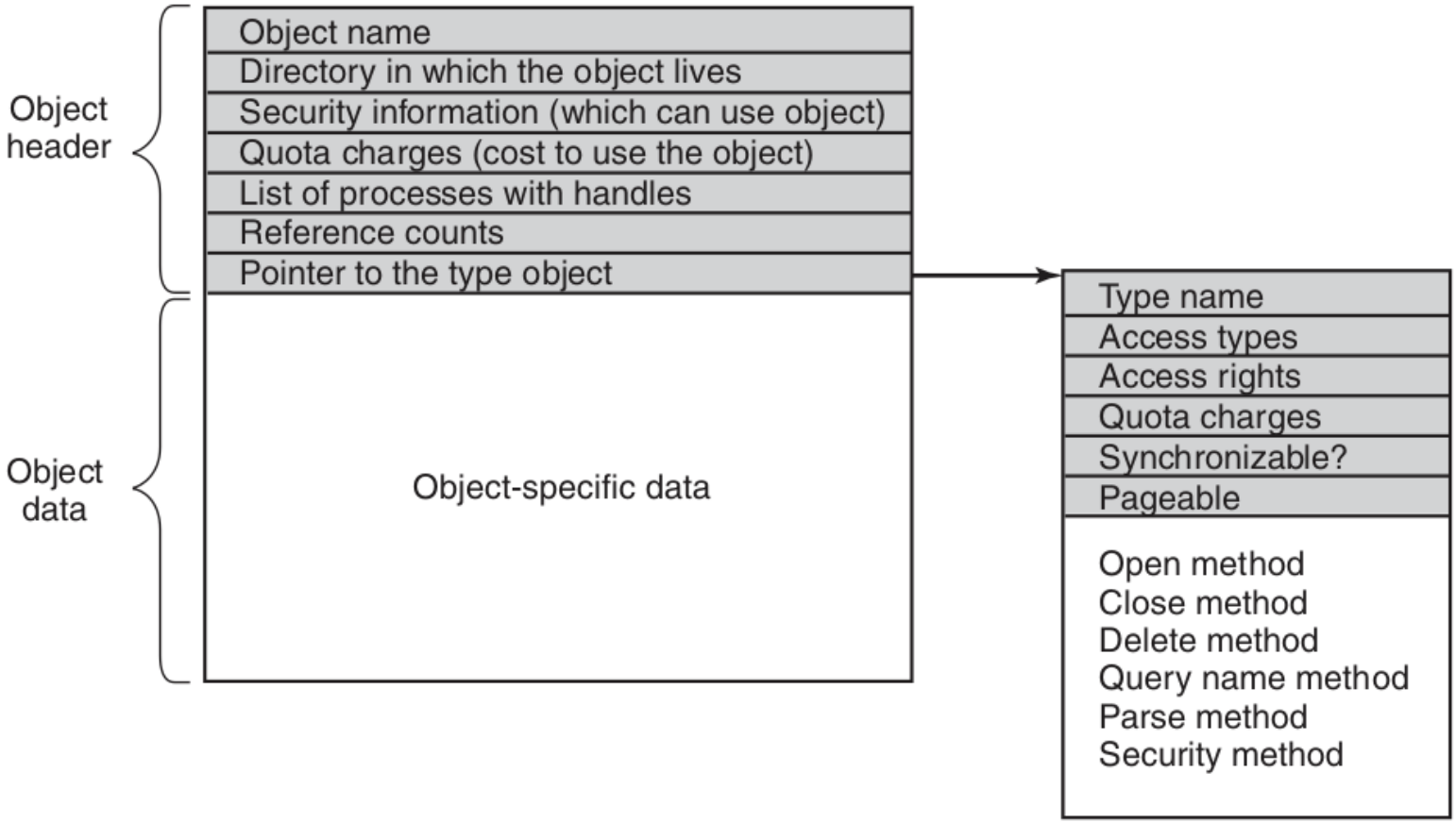
- Ovládače vo Windows môžeme písať ako kernel modul alebo ako user process.
- Keďže vo Windows existuje nespočetné množstvo ovládačov je v súčasnosti preferované implementovať ich v user vrstve.
  - V takom prípade je v kernel-i nutné minimum z kódu ovládača.
  - Zbytok kódu je implementovaný ako služba/knižnica v user vrstve.
- Ovládače sú inštalované do Windows ako DLL. Takýto interface je dostatočne všeobecný takže za ovládač sa považuje aj Win32 API
- Drvivá väčšina BSOD (Blue screen of Death) je zapríčinená ovládačom, ktorý robí to čo nemá.
  - Takže môže za to Windows alebo váš počítač?
  - Prípadne nová karta?

Exekutíva

Object Manager

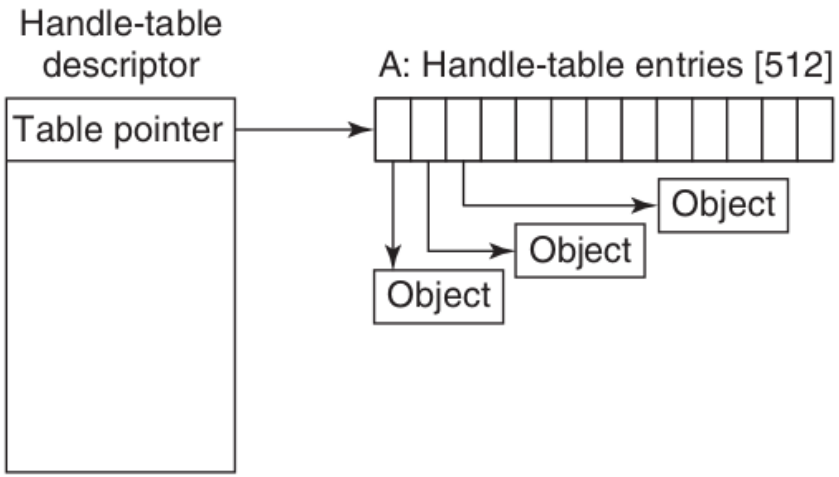
- Tak ako v Linuxe je všetko súbor. Vo Windows je všetko objekt.
- Objekt je otvorený súbor, semafór, proces, thread, sekcia pamäte, časovač, zariadenie.
- Objekt je security token, profil, transakcia.
- Výhodou objektov je ich jednoduché vytváranie, deštrukcia, definovanie quot.
- Procesy používateľa majú prístup vyjadrený pomocou tzv. Handle. (niečo ako file descriptor)
  - Handle je bezpečná náhrada pointera pretože pointer nevieme programovo overiť.
  - Handle má nedefinovanú hodnotu z pohľadu používateľa.

Object Manager

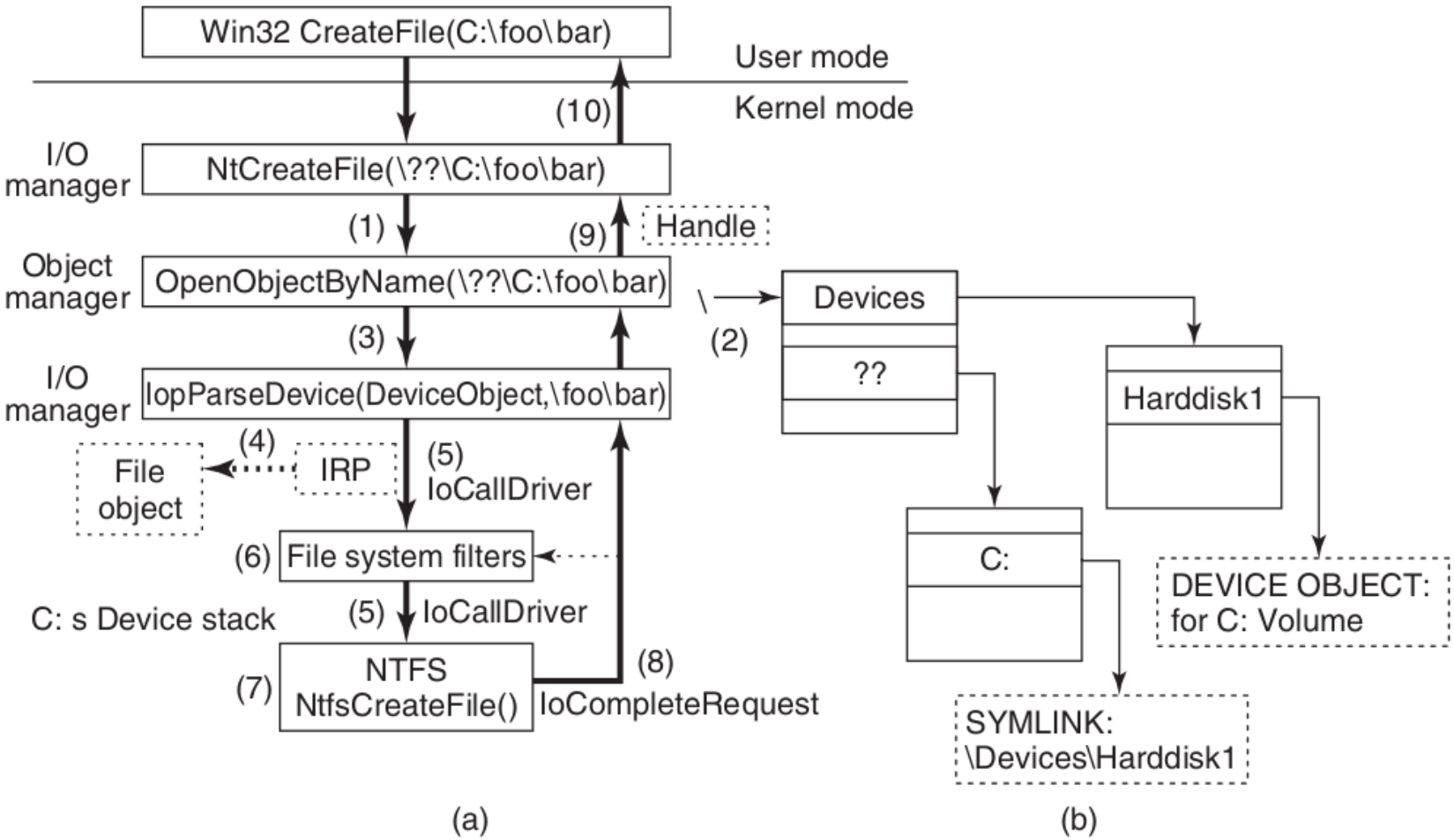


Object Manager

- Handles sú sústredené do handle tables.
- Každý proces má svoju vlastnú tabuľku.
- Tabuľka podporuje až dve úrovne mapovania objektov.



Object Manager - otvorenie súboru



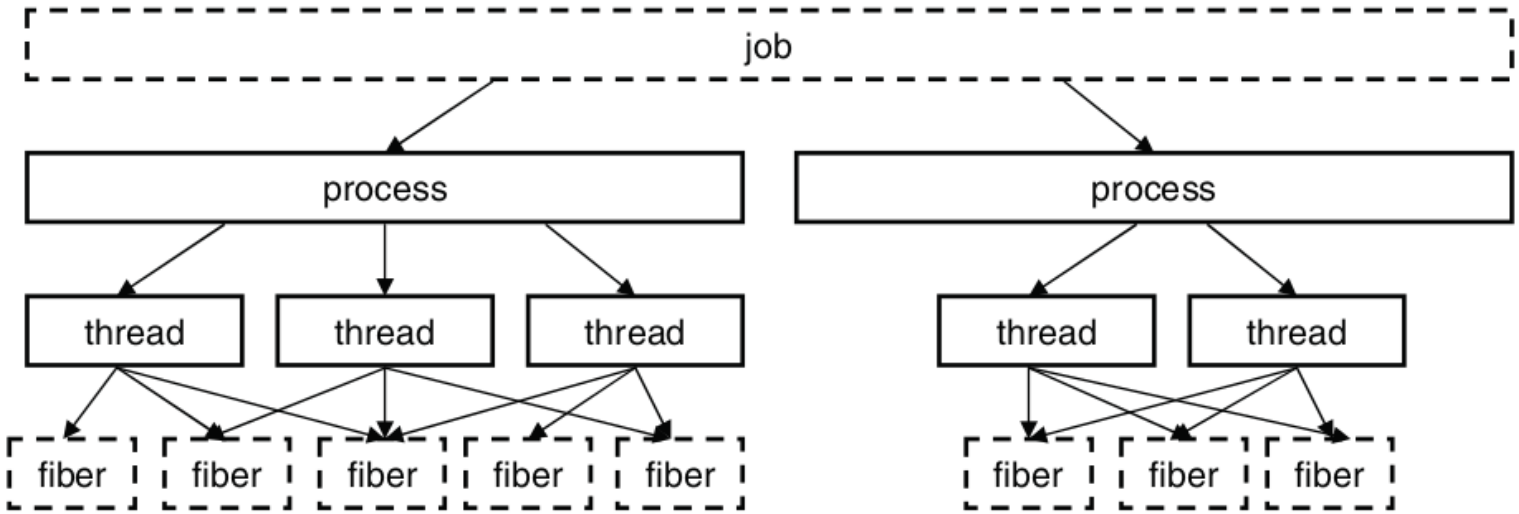
Object Manager

Type	Description
Process	User process
Thread	Thread within a process
Semaphore	Counting semaphore used for interprocess synchronization
Mutex	Binary semaphore used to enter a critical region
Event	Synchronization object with persistent state (signaled/not)
ALPC port	Mechanism for interprocess message passing
Timer	Object allowing a thread to sleep for a fixed time interval
Queue	Object used for completion notification on asynchronous I/O
Open file	Object associated with an open file
Access token	Security descriptor for some object
Profile	Data structure used for profiling CPU usage
Section	Object used for representing mappable files
Key	Registry key, used to attach registry to object-manager namespace
Object directory	Directory for grouping objects within the object manager
Symbolic link	Refers to another object manager object by path name
Device	I/O device object for a physical device, bus, driver, or volume instance
Device driver	Each loaded device driver has its own object

Process

- Process je vo Windows kontajnerom pre program a:
  - virtuálny adresný priestor.
  - zoznam handles na objekty a thread-y
  - Process Environment block (PEB) - user mode data(DLLs, EXE, pracovný adresár, heap)
- thread-y sú objektom plánovania (ako v Linux-e)
  - každý thread ma user mode a kernel mode stack.
  - Thread Environment block (TEB)
- user shared data - read-only zdieľané každým

Process


job
process
process
thread
thread
thread
thread
thread
fiber
fiber
fiber
fiber
fiber
fiber
fiber
fiber

Process

Name	Description	Notes
Job	Collection of processes that share quotas and limits	Used in AppContainers
Process	Container for holding resources	
Thread	Entity scheduled by the kernel	
Fiber	Lightweight thread managed entirely in user space	Rarely used
Thread pool	Task-oriented programming model	Built on top of threads
User-mode thread	Abstraction allowing user-mode thread switching	An extension of threads

Process - IPC

- Pipe, named pipe - byte a message verzie
- MailSlot - na rozdiel od pipe podporujú broadcast
- Socket - rozhranie pre komunikáciu po sieti
- RPC - volanie procedúry z adresného priestoru iného procesu.
- zdieľanie objektov cez handle

Process - Synchronizácia

- semafóry - objekt jadra referencovaný cez handle
  - CreateSemaphore() vytvorí a inicializuje objekt semafóra
  - DuplicateHandle() vytvorenie duplikátu handle, ktorý môžeme odovzdať druhému procesu.
  - ReleaseSemaphore() - up
  - WaitForSingleObject() - down
- mutexy - objekt jadra bez počítadla ako má semafór
- kritické regióny - podobné mutexu ale nie sú kernel objektom. Slúžia ako synchronizácia thread-ov.
- udalosti (events) - notifikácie a synchronizácie

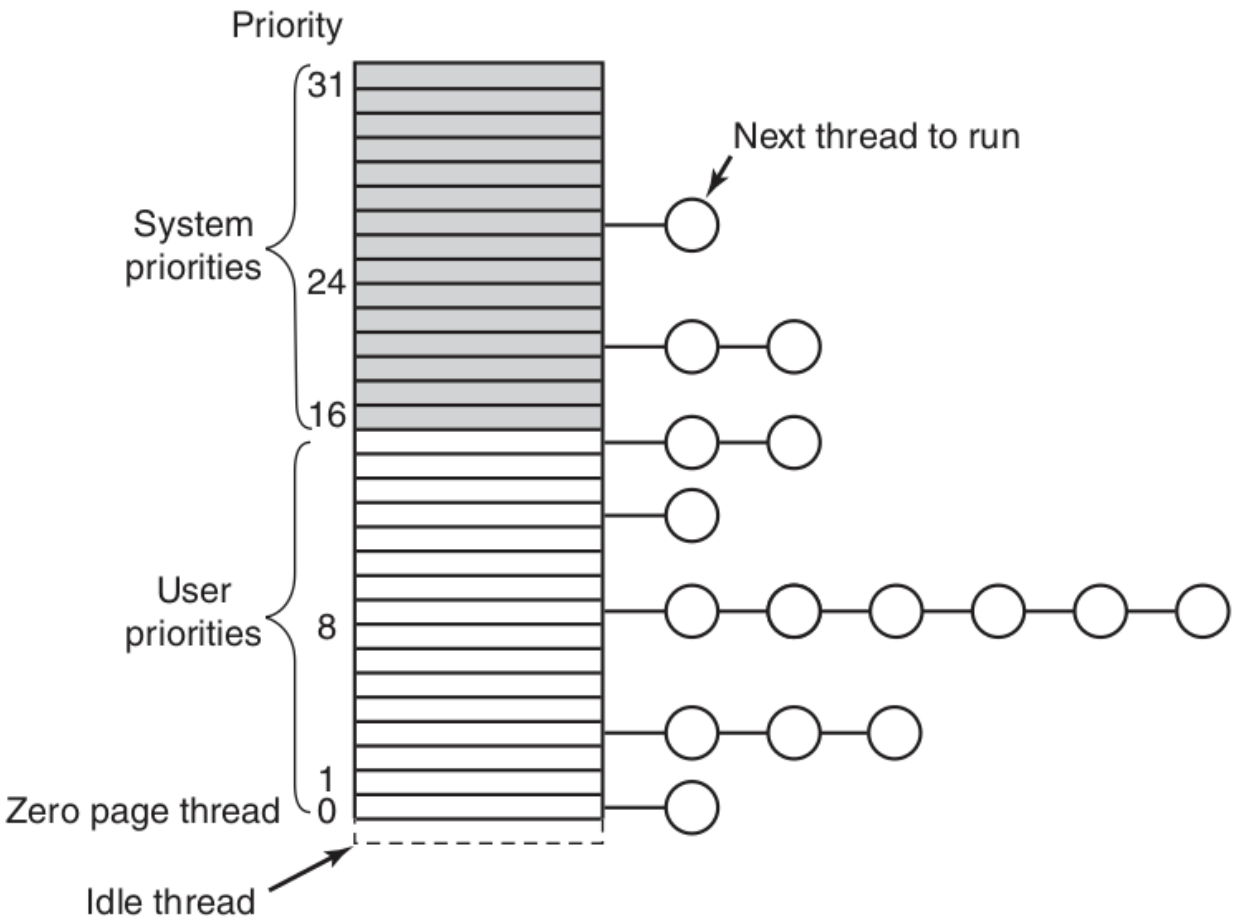
Process - Plánovanie

- Plánovanie nastáva ak:
  - vlákno je blokované, odošle signál objektu, expiruje quantum.
  - I/O operácia skončí.
  - expiruje časovač čakania.
- Dva druhy prioritných tried definuje 32 prioritných úrovní:
  - Procesy - RT, high, above normal, normal, below normal, idle
  - Thready - time critical, highest, above normal, normal, below normal, lowest, idle
- Plánovač udržiava 32 položkové pole radov thread-ov.
- thread má tzv. Baze Priority a Relative Priority.
  - priorita môže narásť ak skončilo I/O, zobudenie na objekte.
  - priority boost končí po vypršaní quanta.
  - priorita môže byť aj odovzdávaná - zamedzenie priority inversion

Process - Plánovanie

		Win32 process class priorities					
		Real-time	High	Above normal	Normal	Below normal	Idle
Win32 thread priorities	Time critical	31	15	15	15	15	15
	Highest	26	15	12	10	8	6
	Above normal	25	14	11	9	7	5
	Normal	24	13	10	8	6	4
	Below normal	23	12	9	7	5	3
	Lowest	22	11	8	6	4	2
	Idle	16	1	1	1	1	1

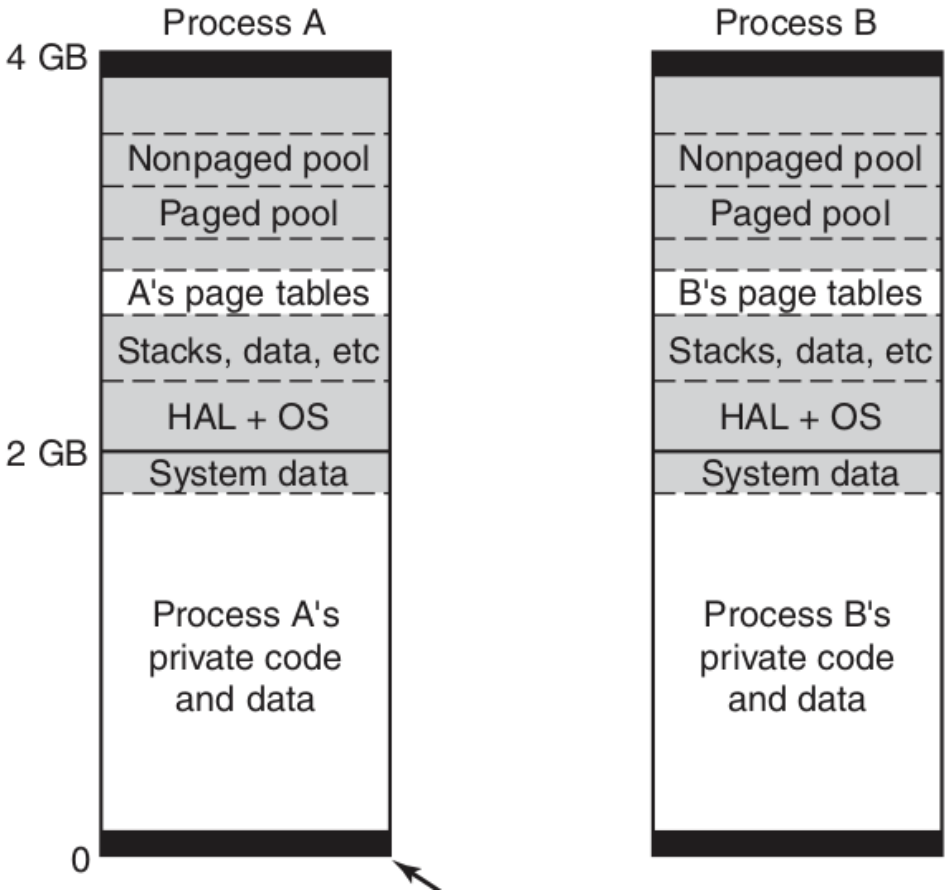
Process - Plánovanie



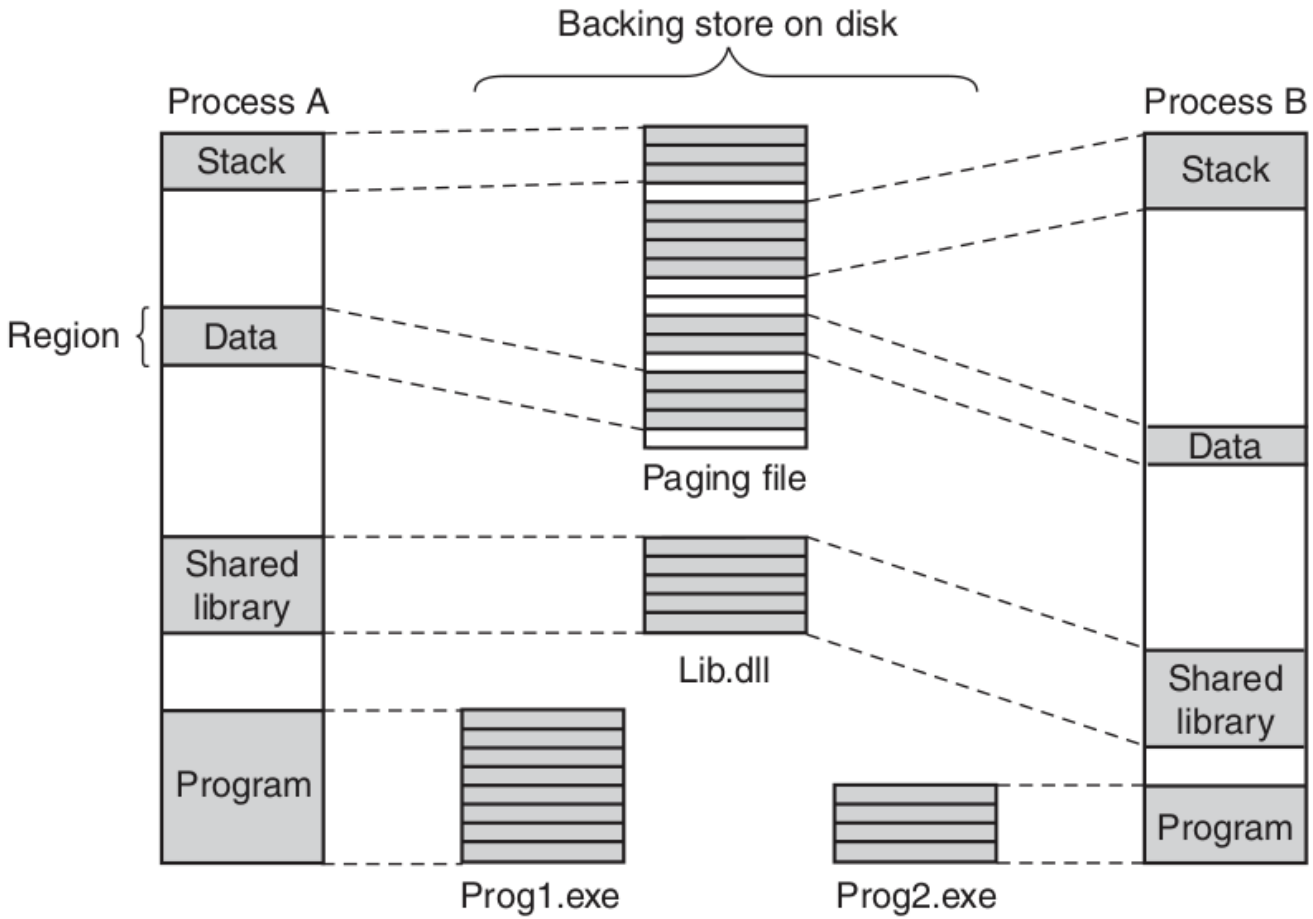
Virtuálny adresný priestor

- Každý proces má vlastný adresný priestor
  - na 32b architektúrach je rozdelený na 2GB user a 2GB kernel space.
  - používa sa Demand Paging.
  - Windows bežne používa 4KB stránky a 2MB stránky (ukladanie tabuliek stránok).
    - Veľké stránky používa kernel a veľké aplikácie.
    - Zlepšuje to page hit rate priamo v TLB.
    - Znižuje množstvo prehľadávaní v tabuľke stránok.
- Stránky, ktoré obsahujú dáta a boli odstránkované sa ukladajú do pagingFile (niečo ako swap).
- Segmentácia nie je používaná.
- Práca s pamäťou na úrovni procesov nie thread-ov

Virtuálny adresný priestor



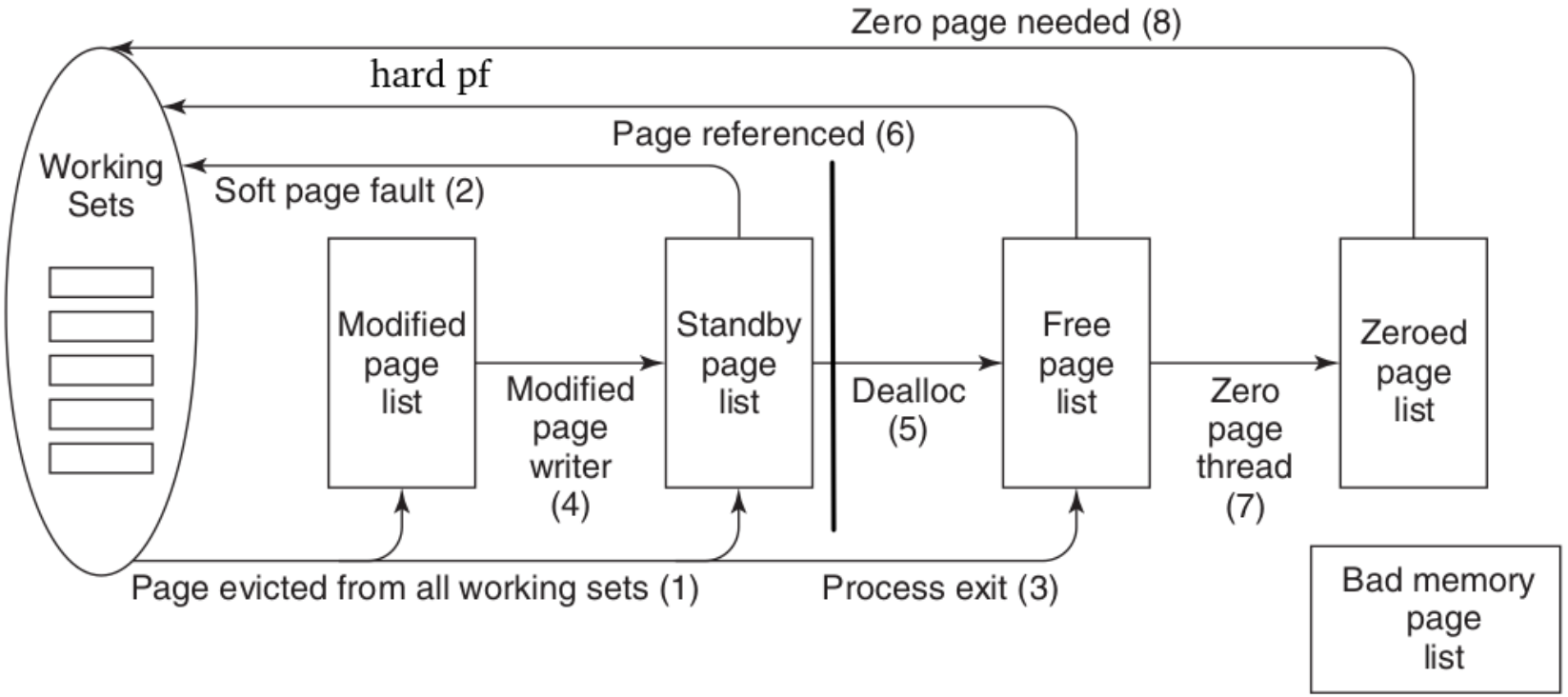
Virtuálny adresný priestor



Page Replacement

- Algoritmus založený na Working Set.
  - základná minimálna veľkosť
  - základná maximálna veľkosť
- Working Set limity sa aplikujú ak je pamäte málo
- WS manager raz za sekundu vykonáva:
  - ak je veľa pamäte - reset R bit, update age of page.
  - ak je menej pamäte - proces s prideleným počtom stránok $>$ WS nahrádza vlastné stránky.
  - ak je málo pamäte - WS manažér odoberá stránky do stanoveného maxima WS.
  - vylúčené stránky sú odložené do stand by listu alebo do modified listu

Page Replacement



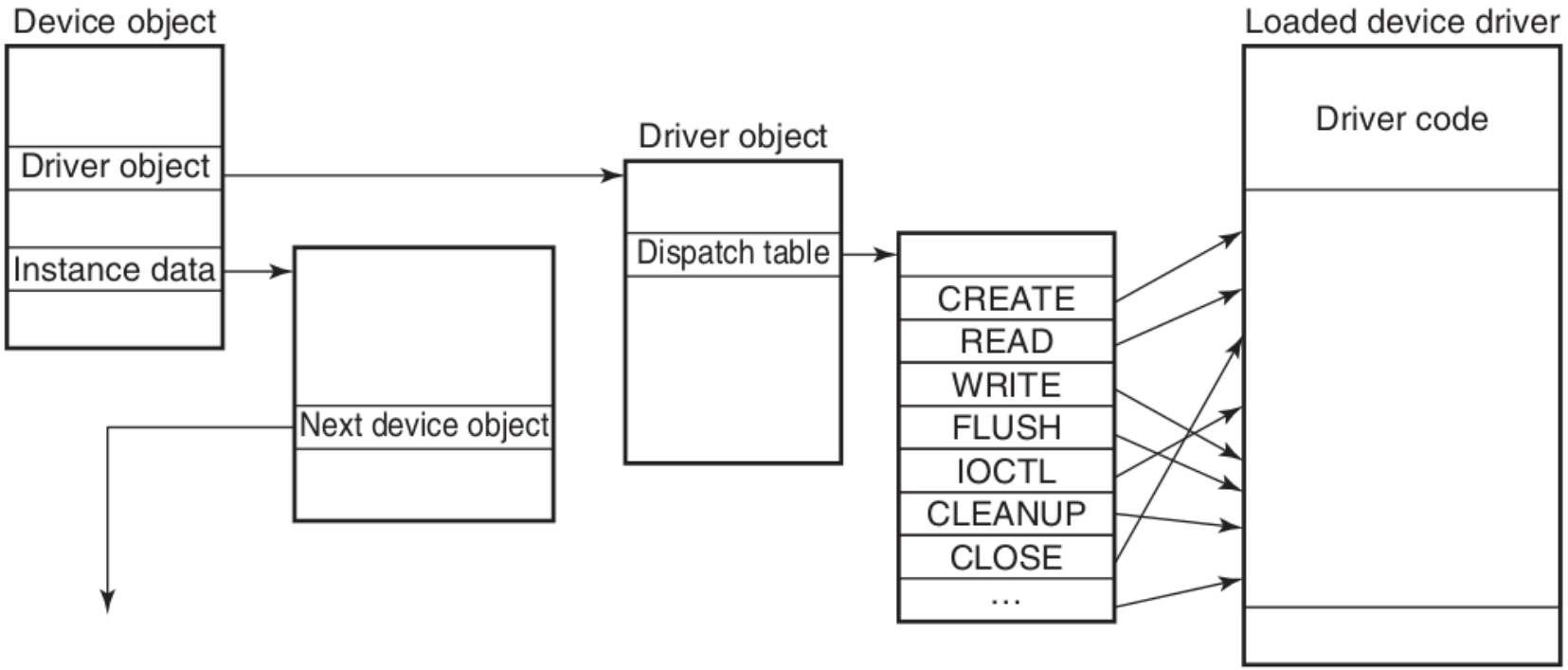
Manažment I/O

- Každé zariadenie sa pri boote alebo pri pripojení registruje.
  1. Dostane priradený Device objekt.
  2. Pre zariadenie sa vyhľadáva ovládač určený registrom.
  3. Ovládač tiež dostane pridelený Device Driver Object.
- Vo Windows je ovládačom prakticky všetko.
- Windows umožňuje jedným FS ovládať viacej partícií a aj diskov.
- Windows podporuje asynchrónne I/O
  1. Thread nie je blokovaný čakaním na I/O.
  2. Thread registruje udalosť (event) a zatiaľ môže robiť inú činnosť.

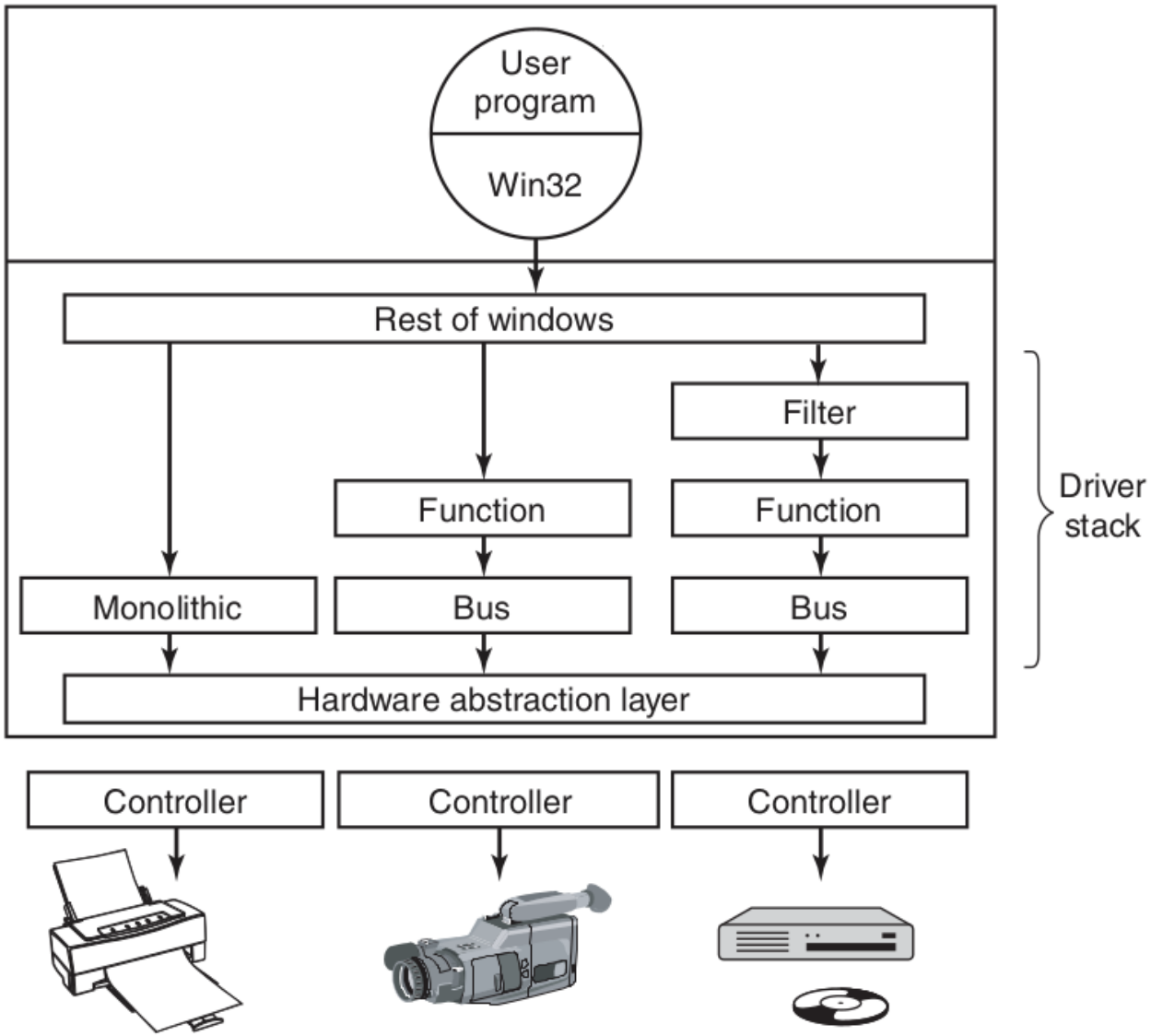
Manažment I/O - Windows Driver Model (WDM)

- Každý ovládač pre Windows musí spĺňať požiadavky WDM.
- Windows Driver Kit (WDK) dokumentuje rozhranie ovládačov.
- Aby sme to mali ešte jednoduchšie tak nad WDM ešte existuje Windows Driver Foundation (WDF)
- A keď stále nestačí: User-Mode Driver Framework (UMDF) a Kernel-Mode Driver Framework (KMDF)

Manažment I/O - Windows Driver Model (WDM)



Manažment I/O - Windows Driver Model (WDM)



Boot

Registre

- Windows udržiava všetky špecifické konfigurácie v tzv. registroch.
- V registroch sú uložené informácie o nastavení ovládačov, boote alebo aplikácie.
- Je to neslávne známa feature Windows.
  - Problém je, že registre nemajú ucelené manuály a preto sa s nimi pracuje nepredvídané.
  - Mnohé user aplikácie majú dostatočné práva na úpravu registrov.
  - Niektoré aplikácie s nimi narábajú dosť nešetrne a môžu poškodiť celý systém.
- Registre sú špeciálnym úložiskom na rozmedzí FS a databázy.
- organizujú sa v tzv. hives. Čo sú vlastne separátne FS

Registre

- Pri boote sa načítava SYSTEM hive registrov.
  - Obsahuje nastavenia OS.
  - Nastavenia programov, ktoré sa spustia pri inicializácií.
  - Priradenie zariadení a ovládačov.
  - Špecifické konfigurácie zariadení.
  - Identifikácia boot zariadení.
- Security Access Manager SAM - obsahuje login dáta o používateľoch.

Registre

Hive file	Mounted name	Use
SYSTEM	HKLM\SYSTEM	OS configuration information, used by kernel
HARDWARE	HKLM\HARDWARE	In-memory hive recording hardware detected
BCD	HKLM\BCD*	Boot Configuration Database
SAM	HKLM\SAM	Local user account information
SECURITY	HKLM\SECURITY	lsass' account and other security information
DEFAULT	HKEY_USERS\.DEFAULT	Default hive for new users
NTUSER.DAT	HKEY_USERS\<user id>	User-specific hive, kept in home directory
SOFTWARE	HKLM\SOFTWARE	Application classes registered by COM
COMPONENTS	HKLM\COMPONENTS	Manifests and dependencies for sys. components

Boot - UEFI

- Unified Extensible Firmware Interface je vylepšením BIOS.
  - Podobne ako BIOS používa malý program bootstrap (MBR)
  - bootstrap program hľadá BootMgr v NTFS.
- BootMgr
  - identifikuje stav systému. (Hibernácia, Stand-by, boot)
  - ak systém zobúdzame spúšťa sa WinResume.exe
  - ak systém štartujeme spúšťa sa WinLoad.exe

Boot - Windows

- WinLoad.exe načíta
  - kernel windows-u ntoskrnl.exe a hal.dll
  - system hive registre a ovládače.
  - Win32k.sys (kernel časť win32).
- HAL init, NTOS init
  - Linkovanie ovládačov.
  - spustenie smss.exe (niečo ako init v Linuxe)

Boot - HELP

- Windows podporuje tzv. safe-boot
- system recovery z posledného bodu obnovy

Zhrnutie

Zhrnutie

- Windows ako taký je mladší SW ako UNIX.
- Špecifickým znakom Windows je jeho monštruóznosť
  - na jednu stranu neustále mení svoju štruktúru,
  - na stranu druhú si zachováva spätnú kompatibilitu,
  - tieto navzájom nesúrodé praktiky spôsobujú, že je to obluda.
- Windows je úspešný OS.
  - keďže je úspešný, tak podstatný podiel nekalého SW je mierený naň,
  - keďže je používaný podstatnou časťou populácie pravdepodobnosť odhalenia chýb je vyššia,
  - vzniká situácia kedy používateľ 'frfle' ale na iný OS neprejde.
- Len čas ukáže kam sa posunie Windows je možné, že keď sa Microsoft plne pretransformuje budeme platiť ročné predplatné.
- https://gs.statcounter.com/os-market-share/

Čo robiť do skúšky

- Prednášky, semináre a vaše data si nezabudnite skopírovať.
- Prelúskať materiály k predmetu:
  - prednášky - 1-11
  - cvičenia - príklady z 10-12
  - literatúra - Tannenbaum to čo sme prešli.
- Skúška v AIS forma textovej odpovede:
  - teoretické otázky - vymenujte, napíšte, opíšte, uveďte, ukážte, navrhnite
  - výpočet príkladov - **výpočet**, analýza, výsledok