# Systémové programovanie Windows

Úvod do systémového programovania Windows

Andrea Číková
Martin Osovský

# Kto sme?

- Ústav výpočetní techniky MU
- Oddělení vývoje systémových služeb
- Andrea – vývojárka vo Win32 API v jazyku C
- Martin – bývalý vývojár

# O čom to bude

- Jazyk C v čistej podobe
- Programovanie vo Windows pomocou Win32 API
- Kurz je vedený podľa kníh Jeffreyho Richtera *Programming Applications for Microsoft Windows* a *Programming Server-Side Applications for Microsoft Windows*
- Naučíte sa programovaciemu štýlu, základnej štruktúre Windows z programátorského hľadiska, odovzdáme Vám svoje skúsenosti

# Čo od Vás budeme chcieť

- Prácu a účasť na seminároch
  - max. 2 neospravedlnené neúčasti
- Domáce úlohy
  - max. 2 neuznané úlohy

**Hodnotenie úloh:**

- Uznaná úloha – spĺňa zadanie
- Neuznaná úloha – neodovzdaná úloha, príp. úloha, ktorá nespĺňa zadanie

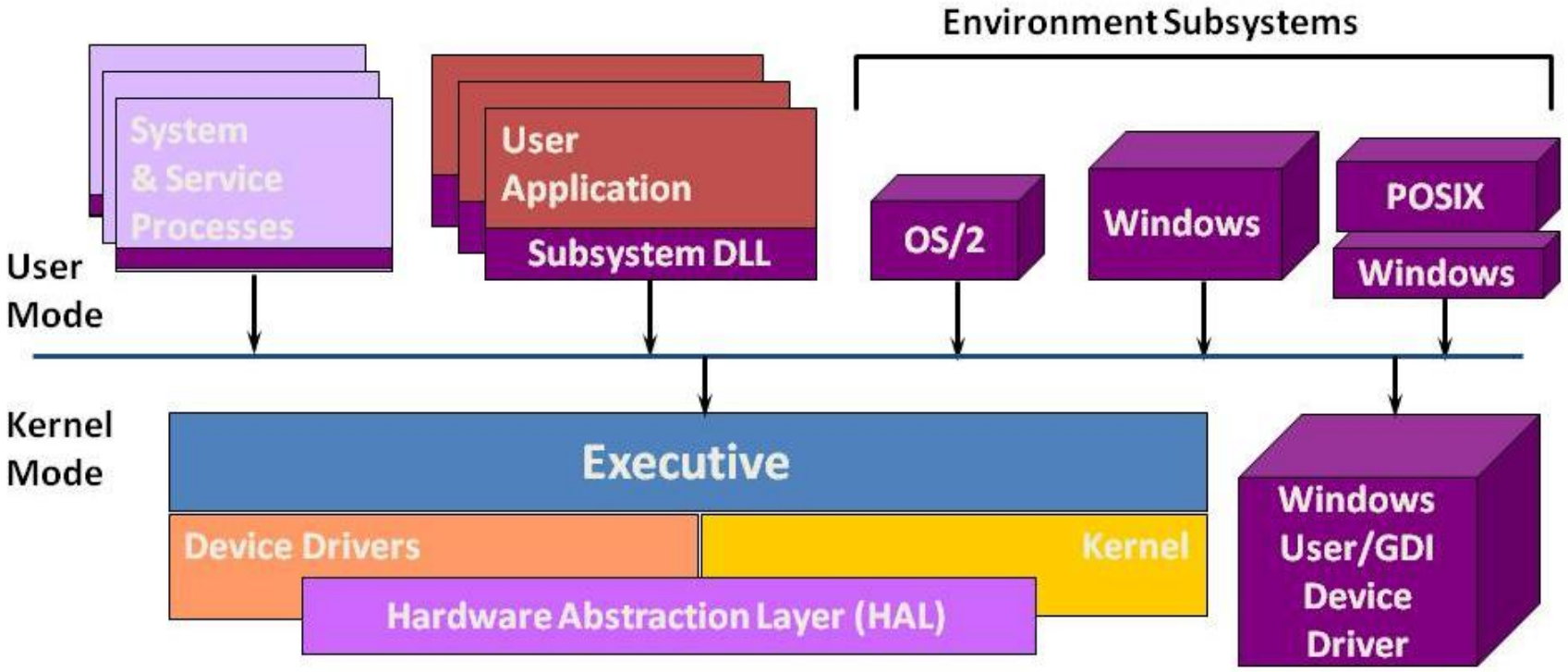
# Na čo Vám to bude?

- Programujú sa v tom
  - drivery, rýchle aplikácie, nízkoúrovňové aplikácie (bezpečnosť a podobne)
  - víry, rootkity
  - antivíry
- Získate zaujímavú a hlbokú znalosť o Windows
- Málokto to už dnes dobre vie

# Alternatívy

- Spočiatku Visual Basic, Delphi
- Niektoré funkcie systému pomocou COM
- Dnes hlavne .NET a Java
- Pre niektoré aplikácie je nevyhnutné použiť Win32 API alebo inú podobnú knižnicu
- Do .NETu je možné doimplementovať, čo tam chýba (COM interop, P/INVOKE)

# Hlavné komponenty systému



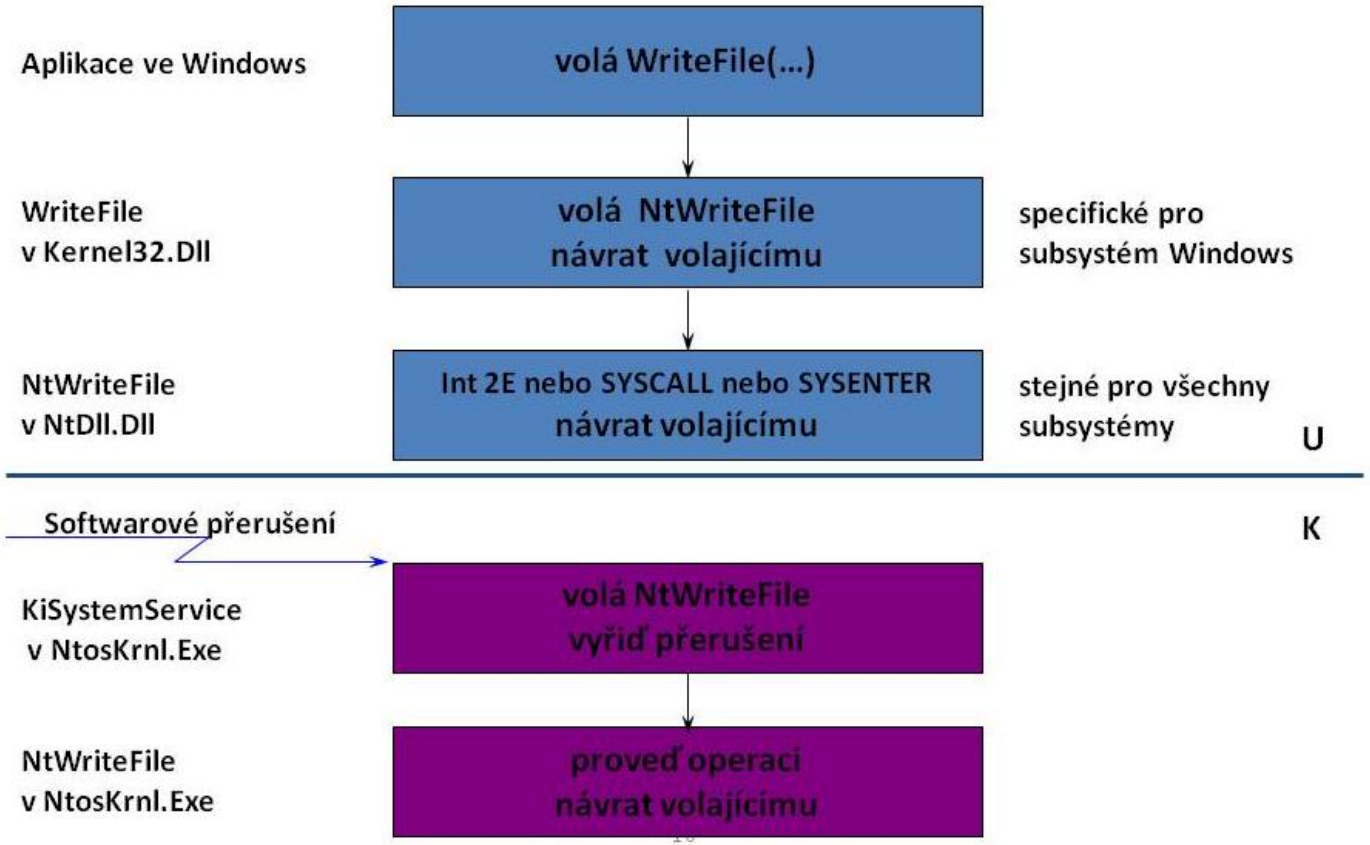
# Windows

- Subsystémy (DLL Kernel32.dll, apl. User32.dll, proces csrss.exe)
- Exekutíva (funkcie OS)
- Jadro (nízkoúrovňové funkcie, plne reentrantné, nie je to microkernel)
- Drivery zariadení (IO)
- HAL (ostatné *HW specific* veci)

# Volanie funkcie

- Prebehne celé v *user space* (funkcia je celá implementovaná v príslušnej knižnici)
- Je potreba volať niečo v kernel space – vyvolá sa služba systému
- Je potreba volať niečo v procese subsystému (csrss.exe)

# Príklad – funkcia používa systémovú službu



# Windows API

- Označované ako WinBB, kde BB je 16, 32 alebo 64
- Hlavné funkcionality:
  - súborové systémy a IO
  - správa pamäte
  - bezpečnosť
  - správa procesov
  - medziprocesová komunikácia
  - synchronizácia vlákien a okien

# Hlavné princípy

- API sprístupňuje predovšetkým objekty jadra, a to pomocou referencie – handle
- Objekt je dátová štruktúra jadra s bezpečnostnou informáciou a počtom odkazov (*usage count*)
  - súbory, procesy, vlákna, rúry, udalosti,...
- API poskytuje funkcie pre vytvorenie handle, zatvorenie handle, čakanie na signalizáciu a ďalšie funkcie pre konkrétnu handle (napr. zápis do súboru)

# Hlavné princípy

- Okrem primitívnych funkcií poskytuje aj pohodlné funkcie, ktoré ich kombinujú (ako *CopyFile* pre čítanie a následný zápis)
- Dôraz je kladený na celý rad komunikačných a synchronizačných prostriedkov

# Pomenovávacie konvencie

- Mená funkcií sú dlhé a popisné
  - *WaitForMultipleObjects()*
- Funkcie mávajú veľa parametrov
- Preddefinované typy sa pomenovávajú kapitálkami, a to vrátane synoným pre existujúce typy (hlavičkový súbor Windows.h)

# Príklady typov

- BOOL (32-bit integer)
- HANDLE
- DWORD
- NULL, VOID, LONG
- LPSECURITY_ATTRIBUTE
- LPCTSTR – maďarská notácia „long pointer to constant text string“ (const TCHAR *)
- Hlavičkové súbory Windows.h, Winbase.h (konštanty a typy) a Winnt.h (typy ako DWORD)

# Rozdiel medzi Win32 a Win64

- Pointre nie sú 32-bitové, ale 64-bitové (vrátane handle)
- Ostatné číselné typy sú rovnaké (DWORD), vrátane LONG
- V API funkciách nie je rozdiel

# Hlavné rozdiely s UNIXom

- Handle súboru je ukazateľ na objekt jadra, nie na malé celé číslo (deskriptory)
- Väčšina objektov jadra je poňatá rovnako (handle)
- Žiadne dané vzťahy medzi objektmi (ako parent – child pri procesoch)
- Konce riadkov nie sú LF, ale CR-LF

# Vzťah ku štandardnej C knižnici

- Je vnorená do Windows API
- Je možné ju používať, ale má obmedzené možnosti (*fopen* vs. *CreateFile*)
- Vhodná je kombinácia, aj keď ku všetkým funkciám existuje API alternatíva (*strcat* vs. *StrCat*)
- Je možné písať aj prenositeľné programy (ale zbytočné)
- Hlavičkové súbory stdlib.h, stdio.h

# Reťazce a Unicode

- Nastavenie projektu alebo define
- TEXT, TCHAR
- Inak L““, WCHAR, alebo ““ a CHAR
- Dve verzie väčšiny funkcií (A a W)

```
#ifdef UNICODE
#define CreateWindowEx CreateWindowExW
#else
#define CreateWindowEx CreateWindowExA
#endif
```

# Spracovanie chýb

- Je nutné dbať na návratové hodnoty (väčšina funkcií sa chová rôzne)
- GetLastError a SetLastError
- Niekedy ako návratová hodnota 32-bitové číslo chyby (HRESULT)
- Hlavičkový súbor WinError.h
- FormatMessage – číslo ->reťazec

# Otázky?

Ďakujeme za pozornosť☺