

Programování ve Windows

1. cvičení

Andrea Číková

Martin Osovský

Kdo jsme?

- Ústav Výpočetní Techniky
- Oddělení vývoje systémových služeb
- Andrea – vývojářka ve Win32 API v jazyce C
- Martin – bývalý vývojář

O čem to bude

- Programování ve windows pomocí Win32 API
- Témata na wiki – v podstatě kniha Programming Applications od Jeffreyho Richtera
- Jazyk C v čisté podobě
- Naučíte se programovacímu stylu, základní struktuře windows z programátorského hlediska, předáme vám svoje zkušenosti

Co po vás budeme chtít

- Práce a účast v seminářích
 - Max 2 neomluvené neúčasti
- Domácí úlohy
 - 0–3 body
 - Odevzdat alespoň 8 úloh
- Projekt
 - Max 10 bodů
 - Odevzdat projekt
- Celkově 20 bodů

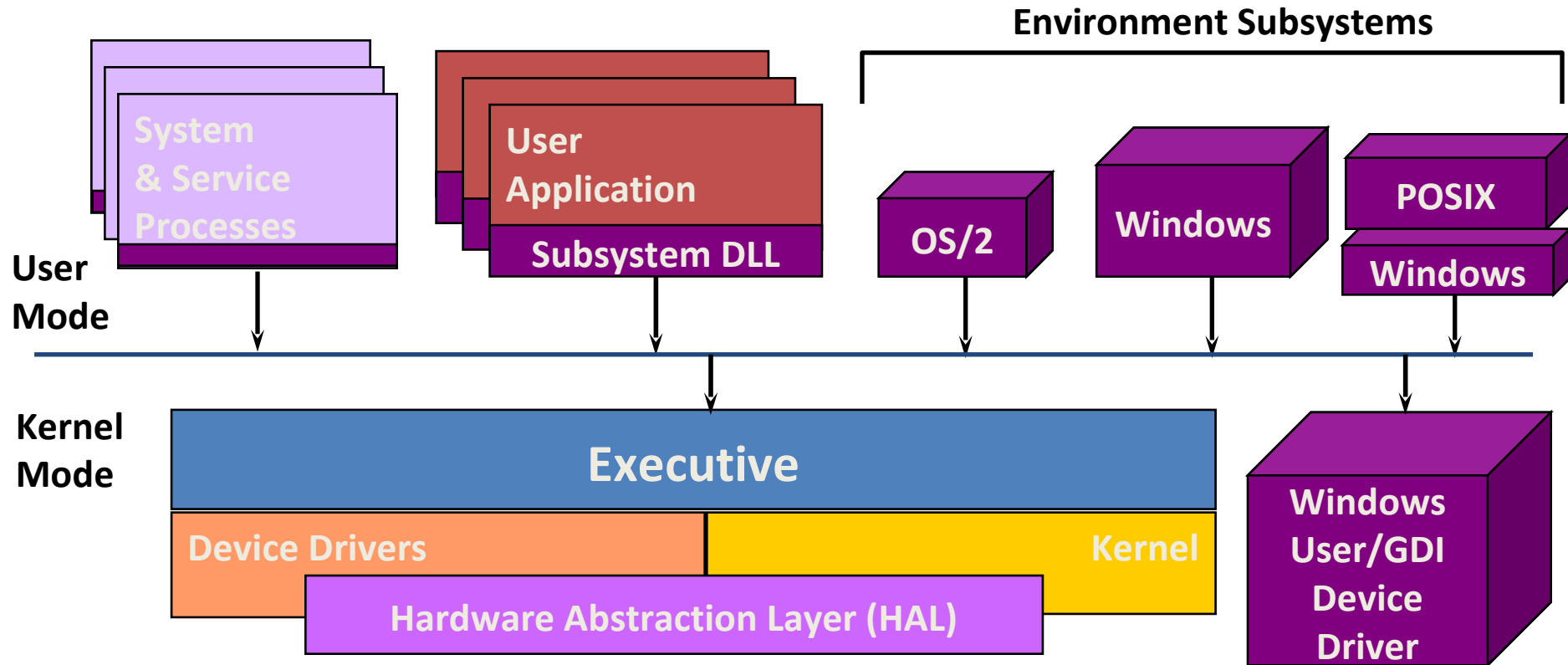
K čemu Vám to bude?

- Programuje se v tom
 - Drivery, rychlé aplikace, nízkourovňové aplikace (bezpečnost a podobně)
 - Viry, rootkity
 - Antiviry
- Získáte zajímavou a hlubokou znalost o Windows
- Málokdo to už dneska dobře umí

Alternativy

- Zpočátku Visual Basic, Delphi
- Některé funkce systému pomocí COM
- Dnes hlavně .NET a Java
- Pro některé aplikace je stále nezbytné použít Windows API nebo jinou podobnou knihovnu
- Do .NETu je možné doimplementovat, co tam chybí (COM interop, P/INVOKE)

Hlavní komponenty systému



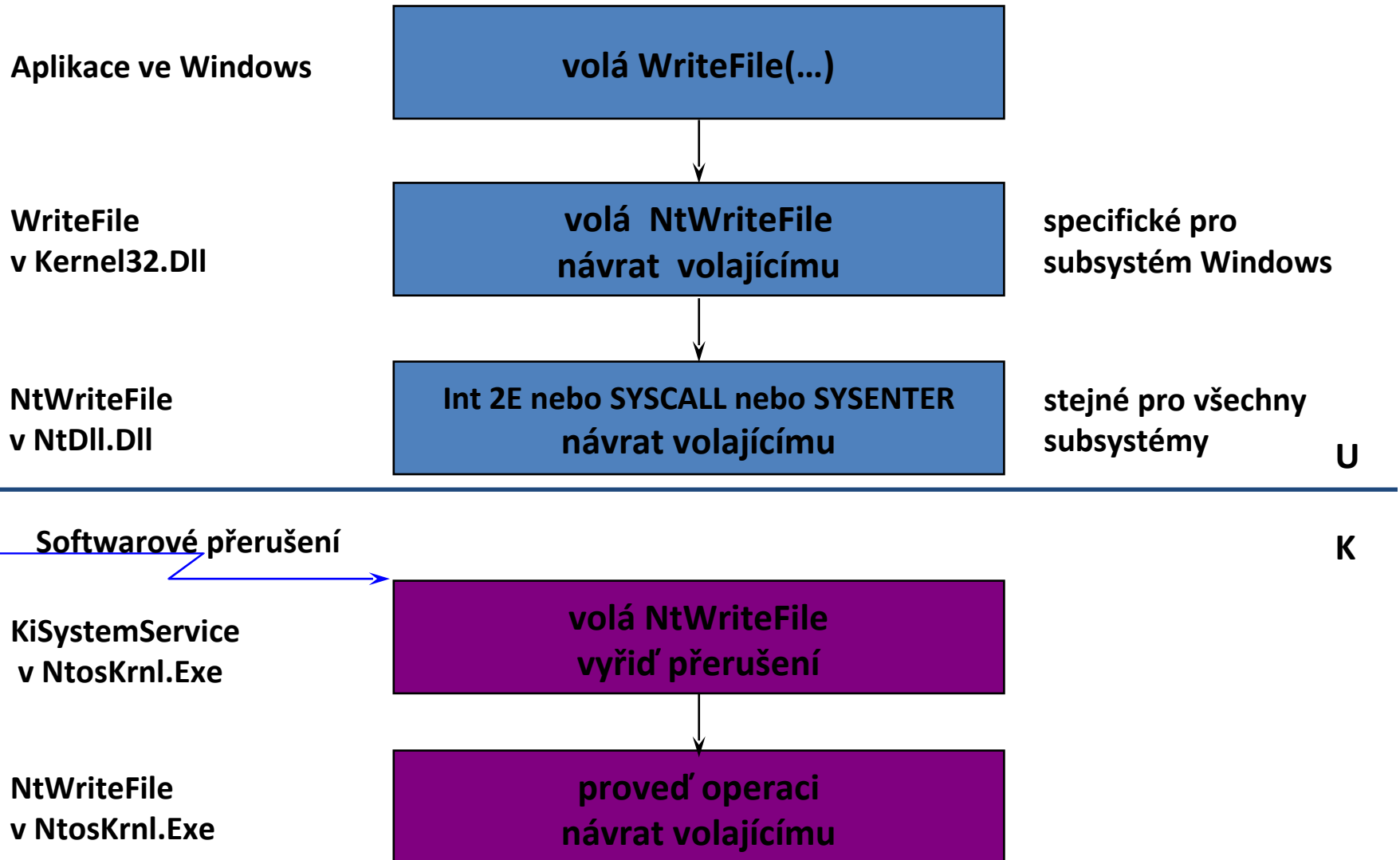
Windows

- Subsystémy (dll Kernel32.dll, User32.dll ap., proces csrss.exe)
- Exekutiva (funkce OS)
- Jádro (nízkoúrovňové funkce, plně reentrantní, není to microkernel)
- Drivery zařízení (IO)
- HAL (ostatní hw specific věci)

Volání funkce

- Proběhne celé v user space (funkce je celá implementovaná v příslušné knihovně)
- Je potřeba volat něco v kernel space – vyvolá se služba systému
- Je potřeba volat něco v procesu subsystému (csrss.exe)

Příklad – funkce používá systémovou službu



Windows API

- Označované WinBB, kde BB je 16, 32 nebo 64
- Hlavní funkcionality
 - Souborové systémy a IO
 - Správa paměti
 - Bezpečnost
 - Správa procesů
 - Meziprocesová komunikace
 - Synchronizace vláken a oken

Hlavní principy

- API zpřístupňuje především objekty jádra, pomocí referencí – handlů
- Objekt je datová struktura jádra s bezpečností informací, počtem odkazů (usage count)
 - soubory, procesy, vlákna, roury, události ...
- API poskytuje funkce pro vytvoření handlu, zavření handlu, čekání na signalizaci a další funkce pro konkrétní handly (e.g. zápis do souboru)

Hlavní principy

- Kromě těchto primitivních funkcí také pohodlné funkce, které je kombinují (jako CopyFile pro čtení a následný zápis)
- Důraz je kladen na celou řadu komunikačních a synchronizačních prostředků

Pojmenovávací konvence

- Jména funkcí jsou dlouhá a popisná
`WaitForMultipleObjects()`
- Funkce mívají hodně parametrů
- Předdefinované typy se pojmenovávají kapitálkami a to včetně synonym pro existující typy (hlavičkový soubor `Windows.h`)

Příklady typů

- BOOL (32 bit integer)
- HANDLE
- DWORD
- NULL, VOID, LONG
- LPSECURITY_ATTRIBUTE
- LPCTSTR – maďarská notace „long pointer to constant text string“ (const TCHAR *)
- Hlavičkové soubory Windows.h, Winbase.h (konstanty a funkce) a Winnt.h (typy jako DWORD ap.)

Rozdíly mezi Win32 a Win64

- Pointry nejsou 32 ale jsou 64 bitové (včetně handlů)
- Ostatní číselné typy jsou stejné (DWORD) včetně LONG
- V API funkcích není rozdíl

Hlavní rozdíly s UNIXem

- HANDLy souborů jsou ukazatele na objekty jádra, ne malá celá čísla (deskriptory)
- Většina objektů jádra je pojímána stejně (HANDLy)
- Žádné dané vztahy mezi objekty (jako parent-child u procesů)
- Konce řádků nejsou LF ale CL-RF

Vztah ke standardní C knihovně

- Je vnořena do Windows API
- Je možno ji používat, ale má omezené možnosti (fopen proti CreateFile)
- Vhodná je kombinace, i když ke všem funkcím existuje API alternativa (StrCat proti strcat)
- Je možné psát i přenositelné programy (ale zbytečné)
- Hlavičkové soubory stdlib.h, stdio.h

Řetězce a Unicode

- Nastavení projektu nebo define
- TEXT, TCHAR a tak
- Jinak L`""`, WCHAR nebo `""` a CHAR
- Dvě verze většiny funkcí (A a W)

```
#ifndef UNICODE
```

```
    #define CreateWindowEx CreateWindowExW
```

```
    #else
```

```
    #define CreateWindowEx CreateWindowExA
```

```
    #endif
```

Zpracování chyb

- Je třeba dbát návratových hodnot (většina funkcí se chová různě)
- GetLastError a SetLastError
- 32 bitové číslo chyby, někdy jako návratová hodnota (HRESULT)
- Hlavičkový soubor WinError.h
- FormatMessage – číslo -> řetězec