VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

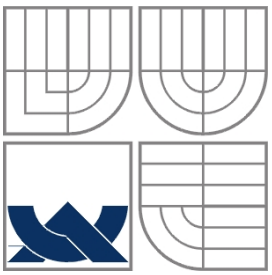
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁSTROJ PRO MONITOROVÁNÍ SYSTÉMOVÝCH PROSTŘEDKŮ OS WINDOWS

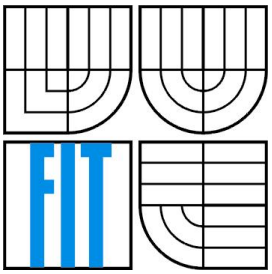
BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE ONDŘEJ HOLUB
AUTHOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁSTROJ PRO MONITOROVÁNÍ SYSTÉMOVÝCH PROSTŘEDKŮ OS WINDOWS

APPLICATION FOR MONITORING SYSTEM RESOURCES OF OS WINDOWS

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ HOLUB

VEDOUCÍ PRÁCE
SUPERVISOR

ING. LUKÁŠ GRULICH

BRNO 2008

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Holub Ondřej**
Obor: Informační technologie
Téma: **Nástroj pro monitorování systémových prostředků OS Windows**
Kategorie: Operační systémy

Pokyny:

1. Analyzujte požadavky na tento nástroj. Cílem je vytvoření aplikace, která bude v reálném zpracovávat a zobrazovat obsazení systémových prostředků (pamětí, atd.) a informace o procesech.
2. Seznamte se s WinAPI, konkrétně s částmi souvisejícími se zadáním. Zvolte vhodný nástroj pro implementaci.
3. Aplikaci implementujte, testujte. Vytvořte uživatelský manuál.
4. Diskutujte možná zdokonalení.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Grulich Lukáš, Ing.**, UITS FIT VUT
Datum zadání: 1. listopadu 2007
Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

LS

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. **Pan**

Jméno a příjmení: **Ondřej Holub**
Id studenta: 78996
Bytem: Sentice 116, 666 03 Hradčany
Narozen: 16. 06. 1985, Brno
(dále jen "autor")

a

2. **Vysoké učení technické v Brně**

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Nástroj pro monitorování systémových prostředků OS Windows
Vedoucí/školitel VŠKP: Grulich Lukáš, Ing.
Ústav: Ústav inteligentních systémů
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:
tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☒ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 8.5.2008

....

Nabyvatel Autor

Abstrakt

Práce pojednává o principu monitorování systémových prostředků v prostředí operačního systému Microsoft Windows. Soustředí se na systémové služby a funkce, které jsou využitelné ke sledování systémových zdrojů a blíže jsou rozebrány aspekty související s jejich používáním. Implementační část tvoří aplikace pro monitorování systému založená právě na těchto službách a funkcích. Zhodnocení práce a její možné rozšíření je uvedeno na závěr.

Klíčová slova

Monitorování, řízení, zatížení, výkon, systém, Microsoft Windows, služba, proces, WMI, Windows API.

Abstract

This bachelor's thesis deals with the monitoring principles of the system devices in the environment of the Microsoft Windows Operating System. The emphasis is placed on the system services and functions which are utilizable for the system resources monitoring. Some aspects related to their functioning are discussed and specified. The implementation part is represented by an application for the system monitoring based on these services and functions. The evaluation of the work and a possible extension is mentioned in the conclusion.

Keywords

Monitoring, Control, Load, Performance, System, Microsoft Windows, Service, Process, WMI, Windows API.

Citace

Holub Ondřej: Nástroj pro monitorování systémových prostředků OS Windows. Brno, 2008, bakalářská práce, FIT VUT v Brně.

Nástroj pro monitorování systémových prostředků OS Windows

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Lukáše Grulicha.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Ondřej Holub
6.5.2008

Poděkování

Rád bych poděkoval všem, kteří se podíleli na testování implementované monitorovací aplikace a všem, kteří mě při práci na ní podporovali.

Děkuji panu Ing. Lukáši Grulichovi, svému vedoucímu práce, za podporu a čas strávený při konzultacích řešeného problému.

Obsah

Úvod

Cílem této práce je seznámení s možnostmi operačního systému Microsoft Windows v oblasti systémových služeb a funkcí, které mohou být využity pro měření systémových prostředků a monitorování ukazatelů výkonu. Pozornost je věnována popisu implementace monitorovací aplikace, jež má za úkol shromažďovat naměřené hodnoty a prezentovat je vhodným způsobem uživateli. Práce je členěna do několika kapitol, které se blíže věnují problematice dané oblasti.

První kapitola je teoretickým úvodem do problematiky systémových funkcí sloužících k monitorování systémových prostředků. První její část se věnuje rozhraní Windows API, jež nabízí operační systém a možnostem jeho využití pro přístup a získávání statistických dat pro monitorování systému. Jsou zde nastíněny některé velmi často volané funkce a problémy související s jejich používáním. V druhé části této kapitoly se pak práce zaobírá oblastí služeb systému Microsoft Windows, které mají za úkol zpracovávat a následně pak prezentovat data o systémových prostředcích. Je zde kladen důraz na možnosti, jež tyto služby poskytují programátorům a uživatelům aplikací pro operační systémy Microsoft Windows. Jsou zmíněny způsoby přístupu k těmto datům a možné problémy související s těmito službami. Poslední část kapitoly se zaměřuje na popis problémů s kompatibilitou, které jsou způsobeny odlišnými vlastnostmi různých verzí systému Microsoft Windows, ale i omezeními plynoucími z uživatelských práv přidělených uživateli systémem nebo omezeními danými nástroji pro zvýšení bezpečnosti.

Druhá kapitola má za cíl přiblížit praktickou část bakalářské práce. Ve své první části se zaměřuje na jednotlivé klíčové vlastnosti aplikace a jejich přínos uživateli. Jsou zde ve stručnosti popsány základní moduly, které jsou implementovány. Následuje další část kapitoly, v níž je věnována pozornost základním konceptům vytvářené aplikace. Dále je popsána implementace zajímavých částí aplikace pro monitorování prostředků operačního systému Microsoft Windows. Pozornost je zde věnována i použitému vývojovému prostředí a cílům, které by měla navržená aplikace dosáhnout. Závěrem této kapitoly je popsán způsob testování aplikace.

V posledních kapitolách práce jsou zmíněny zkušenosti se zpracováním vybraného tématu a jsou nastíněny trendy, dalšího vývoje aplikace.

1 Prostředky systému pro monitorování

1.1 Windows API

1.1.1 Charakteristika Windows API

Windows API je systémové rozhraní, vyvinuté společností Microsoft a integrované jako jedna z nedílných součástí do operačního systému Windows. Účelem Windows API je umožnit co nejjednodušší a nejrychlejší cestou přístup k systémovým funkcím, službám a proměnným systému, jež nabízejí cenné informace. Důležitým posláním tohoto systémového rozhraní je možnost částečného odstínění systému od aplikací, které jsou díky tomu kompatibilnější a nejsou tolik závislé na používané verzi systému. Díky standardizaci Windows API je vývoj aplikací méně finančně náročný a je podstatně jednodušší a rychlejší. Všechny aplikace s výjimkou konzolových aplikací musí s operačním systémem komunikovat právě přes Windows API. Toto rozhraní bylo navrženo pro využití v programovacích jazycích C a C++, ale není problém s ním pracovat i v jiných vývojových prostředích ať již Windows API nativně podporují nebo za využití pomocných knihoven.

1.1.2 Oblasti využití Windows API

Možnosti Windows API jsou velmi široké a proto podle oblastí, kterým toto rozhraní poskytuje svoje služby, můžeme stanovit následujících 7 polí působnosti.

1.1.2.1 Administrace

Služby Windows API z oblasti správy systému umožňují úkony jako je například instalace, konfigurace nebo správa jednotlivých aplikací nebo i celého operačního systému. Do této kategorie se například řadí i možnost přístupu ke službám, z nichž se v následujících kapitolách budeme věnovat nejvíce službě WMI, s jejíž pomocí lze získat velmi komplexní informace o počítačovém systému.

1.1.2.2 Diagnostika

Diagnostické služby nabízejí možnost monitoringu aplikací a systému. Jedná se jednak o funkce sloužící pro sledování chybových událostí aplikací a zaznamenávání kritických chyb, jednak o velmi důležité funkce pro monitorování výkonu, které poskytují cenné informace a kvůli nízké náročnosti na využití systémových prostředků jsou vhodné pro sledování systému v reálném čase. Touto cestou lze získat velké množství informací o operačním systému, sítích, hardware i celkovém zatížení systému.

1.1.2.3 Grafika a multimedia

Tato součást rozhraní Windows API má na starosti také práci s grafikou a napomáhá standardizovat způsob vytváření grafického výstupu spouštěných aplikací na zobrazovacích zařízeních. Funkce grafického rozhraní zapouzdřují zpracovávání a vykreslování základních grafických entit jako jsou čáry, křivky, bitmapové obrázky, text a další standardní grafická primitiva. Rozhraní umožňuje přímý výstup na určité grafické zařízení pomocí takzvaného Device Contextu.

1.1.2.4 Sítě

V této oblasti je pomocí Windows API zapouzdřen přístup k různým síťovým službám. Windows poskytují podporu velkému množství průmyslových standardů, jako jsou například Windows Sockets, služba pro vzdálené volání procedur označovaná zkratkou RPC, protokol pro monitorování stavu sítě SNMP. Dále je umožněno přistupovat k jmenným síťovým službám DNS. Rozhraní Windows API lze využít pro komunikaci pomocí velmi často využívaných standardizovaných protokolů, jako je například FTP nebo HTTP. Systém pomocí Windows API zpřístupňuje informace o síťové komunikaci a umožňuje aplikacím nastavování a přistup k síťovému hardware, mezi který patří kromě hardware klasických sítí i bezdrátové WIFI sítě a Bluetooth zařízení.

1.1.2.5 Bezpečnost

Služby a funkce z oblasti bezpečnosti slouží ke správě oprávnění systému a systémové bezpečnosti. Jedná se o služby a funkce zabezpečující autorizovaný přístup k systémovým prostředkům a funkce pro řízení oprávnění přístupu ke službám a procesům.

1.1.2.6 Systémové služby

Funkce v oblasti systémových služeb poskytované rozhraním Windows API umožňují aplikacím přistupovat k systémovým zdrojům počítače, jako jsou například paměť, souborový systém, zařízení, procesy, vlákna a další. Aplikace mohou tyto funkce využít pro sledování a správu prostředků systému. Například velmi často využívané jsou funkce pro přidělování a uvolňování paměti, správu a synchronizaci procesů, koordinaci operací prováděných vícevláknovými aplikacemi a komunikaci mezi procesy. Rozhraní umožňuje přístup k souborovému systému a vstupně výstupním zařízením. Zapouzdřuje funkce pro přístup k registru operačního systému, službám, systémovým informacím a mnohým dalším specializovaným oblastem systému.

1.1.2.7 Uživatelské rozhraní

Pomocí funkcí pro práci s uživatelským rozhraním lze vytvářet a řídit chování jednotlivých oken a komponent. Mezi úkoly uživatelského rozhraní patří zajištění a zpracování vstupu z periferních zařízení jako jsou klávesnice a myš a jejich předání cílové aplikaci.

1.1.2.7.1. Grafické uživatelské rozhraní

Slouží pro komunikaci mezi uživatelem a počítačem, je tvořeno z interaktivních grafických prvků. Grafické uživatelské rozhraní zajišťuje jednotný vzhled aplikací běžících v operačním systému a poskytuje výstup na monitor nebo jiné výstupní zobrazovací zařízení.

1.1.2.7.2. Common Control Library

Knihovna běžných prvků umožňuje přístup k pokročilejším komponentám uživatelského rozhraní operačního systému. Jsou v ní zahrnuty vizuální komponenty jako například toolbary, záložky, tlačítka. Díky této knihovně může být standardizován vzhled aplikací běžících v operačním systému.

1.1.2.7.3. Common Dialog Library

Knihovna běžných dialogových oken systému Windows spolu s knihovnou běžných prvků sjednocuje vizuální podobu aplikací pro operační systém Windows. Zahrnuje například standardní dialogy pro výběr barvy, otevření nebo uložení souboru, dialog pro výběr tiskárny a mnohé další.

1.1.2.7.4. Windows Shell

Windows Shell je součástí uživatelského rozhraní Microsoft Windows a poskytuje přístup k velkému množství objektů nezbytných pro běh aplikace a správu systému. Mezi nejdůležitější objekty se řadí soubory a složky, nad kterými umožňuje Shell vykonávat operace jako kopírování a mazání. Objekty jsou organizovány v hierarchickém jmenném prostoru a Shell umožňuje jejich efektivní správu a zpřístupňuje je uživatelům a aplikacím.

1.1.3 Monitorování systému pomocí Windows API

Windows API nabízí mnoho nástrojů, pomocí kterých lze získávat informace o monitorovaném systému. Jsou implementovány funkce, s jejichž pomocí mohou aplikace standardizovaným způsobem získávat informace o hardwarových součástech počítače a mohou přistupovat k procesům běžícím v systému a ovládat je. Z funkcí užitečných pro monitorování systému lze jmenovat funkce pro získání zatížení paměti, procesoru a obsazení místa na pevném disku. Mezi hlavní výhody těchto funkcí patří jejich rychlost, díky které jsou ideální pro monitorování v reálném čase.

1.1.3.1 Windows API funkce pro získání informací o systému

Základní informace o stavu systémových prostředků pro účely monitorování v reálném čase je možno získávat z operačního systému pomocí Windows API a jeho funkcí k tomuto účelu určených. Informace o využití operační paměti a velikosti stránkovacího souboru lze snadno získat voláním funkce “GlobalMemoryStatus()”, která navrací v datové struktuře “MEMORYSTATUS” požadované informace. Funkce bohužel zobrazuje korektní informace jen pro systémy, které disponují méně než 4GB operační paměti, proto se doporučuje používat její modernější variantu nazvanou “GlobalMemoryStatusEx()”, která pracuje korektně za všech podmínek.

Získání informací o procesoru je výrazně komplikovanější a zvláště u systémů s vícejádrovými procesory není triviálním úkonem. K tomuto typu informací lze přistupovat pomocí služby WMI, což ale není zcela vhodný způsob pro sledování prostředků v reálním čase kvůli velkému zatížení systému touto službou. Protože tyto informace nejsou jiným běžným a dokumentovaným způsobem za pomoci Windows API dostupné, doporučuje se k jejich získání využit některé z volně šiřitelných knihoven využívající nedokumentované funkce rozhraní. Tyto funkce využívá například integrovaný program pro sledování prostředků systému ve Windows.

1.1.3.2 Windows API funkce pro práci s procesy

V operačním systému Microsoft Windows lze přistupovat k běžícím procesům za pomoci Windows API. Jednou ze základních funkcí je funkce nazvaná "OpenProcess()", která vrací popisovač požadovaného procesu za předpokladu, že má aplikace dostatečná oprávnění k němu přistupovat. Po získání ukazatele na proces je možno dostat různorodé informace o zatížení systémových prostředků nebo nad tímto procesem provádět administrativní úkony.

Pro získání informací o využité paměti se využívá funkce "GetProcessMemoryInfo()", která tyto informace zapisuje do datové struktury nazvané "PROCESS_MEMORY_COUNTERS". Informace lze získat v případě, že máme dostatečné oprávnění přístupu k procesu. Uživateli typu administrátor je umožněn přístup ke všem procesům, standardním uživatelům jsou zpřístupněny pouze informace o procesech, jejichž jsou vlastníky. Výsledkem funkce je navrácená datová struktura naplněná informacemi o aktuálním a maximálním využití fyzické paměti a stránkovacího souboru.

Získání informací o využití procesoru je složitější, než shromažďování informací o paměti. Je nutno opakovaně vzorkovat čas procesoru spotřebovaný procesem a následně pak na základě takto nashromážděných údajů vypočítat využití procesoru dle vztahu:

$$\frac{(\text{aktualni\_casCPU} - \text{predchozi\_casCPU})}{\text{uplynuly\_}cas}$$

K sestavení kompletního seznamu všech běžících procesů lze využít funkce "CreateToolHelp32SnapShot", která vytváří obraz o aktuálním stavu procesů běžících v systému. Následně lze vyčíslit seznam procesů za použití funkcí "Process32First" a "Process32Next" a s ním získat i základní informace o procesech ve struktuře nazvané "PROCESSENTRY32". K dispozici jsou touto cestou informace o jménu procesu, počtu jeho vláken, identifikačním čísle, identifikačním čísle rodičovského procesu a o prioritě běhu procesu v systému.

Pomocí Windows API je možno za předpokladu, že disponujeme dostatečnými právy, provádět nad procesy operace, jako jsou ukončení procesu nebo změna jeho priority.

1.1.4 Možné problémy pří používání Windows API

Windows API je velmi komplexním rozhraním, které se vyvíjí v čase společně s operačním systémem Windows. Tato vlastnost s sebou přináší drobné problémy při jeho využívání, kdy některé vlastnosti mohou být různě implementovány v různých verzích Windows. Příkladem tohoto mohou být změny v rozhraní související s uvedením na trh nejnovější verze operačního systému Windows. Problém může nastat i u běžně používaných funkcí jako je funkce "OpenProcess()". V případě jejího užití v dříve vytvořených aplikací může ve Windows Vista docházet k závažným problémům, neboť Společnost Microsoft zavádí v rámci zvyšování bezpečnosti nová oprávnění k přístupu, která se ve starších verzích systému nevyskytovala, proto je nutno starší aplikace vhodným způsobem upravit a důkladně otestovat, aby byla zaručena jejich funkčnost.

1.2 Služba WMI

1.2.1 Základní informace o službě WMI

Operační systémy společnosti Microsoft obsahují již od uvedení Windows NT 4.0 (SP4) jako nativní součást službu s názvem WMI. Do nižších verzí operačních systémů je možno tuto službu dodatečně nainstalovat. WMI je zkratkou pro Windows Management Instrumentation, což je služba určená pro snazší administraci operačního systému. Tato služba není obecně mezi uživateli příliš známa, ačkoliv umožňuje velmi komfortně a bezpečně přistupovat k informacím o systému a provádět snadno jeho správu. Účelem zavedení služby WMI bylo vytvoření prostředí, které by nebylo závislé na použité verzi operačního systému a umožnilo by jednotným způsobem aplikacím přistupovat k systémovým, jinak nedostupným informacím, automatizovat administrativní úkony a celkově zjednodušit tvorbu aplikací pro správu systému. Velmi často se této služby využívá ve skriptech nástroje PowerShell. Službou WMI je možno spravovat lokální i vzdálené počítačové systémy. Služba využívá CIM objektového databázového modelu, jehož implementace je označována jako "CIM Repository". Jednotlivé části operačního systému, k nimž můžeme pomocí WMI přistupovat, jsou členěny jako jednotlivé COM objekty v databázi. Databáze je rozšiřitelná a softwarové produkty jako jsou například SQL Server, Exchange Server, Microsoft Office do ní přidávají vlastní COM objekty, nad nimiž je následně možno provádět administrativní úkony. Počet objektů poskytující WMI informace se s každou další verzí služby zvyšuje, mezi důležité objekty zprostředkující systémové informace lze bezesporu zařadit například tyto:

Win32 Provider	Zabezpečuje přístup a čerstvé informace o stavu systému, jeho nastaveních a proměnném systémovém prostředí.
Performance Counter Provider	Pomocí tohoto objektu lze přistupovat k statistickým systémovým informacím, které se týkají využití a zatížení systémových zdrojů.
Windows Installer Provider	Zprostředkovává přístup ke službám komponenty "Windows Installer" a umožňuje získávání informací o nainstalovaných aplikacích.
Session Provider	Objekt umožňující získání informací o síťové komunikaci a síťových prostředcích systému.

1.2.2 Historický vývoj WMI

Služba WMI se od dob, kdy byla poprvé implementována ve Windows NT, neustále vyvíjí a jsou přidávány nové objekty, s jejichž pomocí se zvětšuje rozsah poskytovaných informací a administrativních služeb. V původní verzi bylo implementováno pouze 15 základních objektů, s příchodem verze Windows 2000 se jejich počet zvýšil na 29, v serverové verzi Windows 2003 je jich přítomno 80 a v budoucích verzích systému je plánováno společností Microsoft jejich počet nadále zvyšovat s čímž poroste i důležitost celé této služby.

1.2.3 Vlastnosti služby WMI

Mezi hlavní rysy rozhraní vytvářeného službou WMI patří možnost využití jeho vlastností ke snadné automatizaci a dělají z něj silný nástroj při tvorbě skriptů, kdy je vývojář odstíněn od problematiky nízkoúrovňových funkcí systému, ve kterých může jednotným způsobem přistupovat k systémovým informacím a funkcím. Na informace, které poskytuje WMI se lze dotazovat pomocí WQL (Windows Management Instrumentation Query Language), což je podmnožinou SQL databázových jazyků. Tato vlastnost umožňuje využití výhod SQL, jako je například jednoduché filtrování získávaných informací nebo řazení podle zadaného kritéria.

WMI je primárně určená pro užití v aplikacích vytvořených v jazyce C nebo C++, navíc je nativně podporována frameworkem .NET, což zajišťuje její podporu u moderních vývojových nástrojů. Ovšem není problémem WMI využít i v jiných vývojových prostředích, která nejsou vybaveny vestavěnou podporou této služby. V tomto případě plně postačuje, když jsou prostředí schopna přistupovat k ActiveX prvkům.

Služby nabízené WMI umožňují snadnou cestou získat velké množství detailních informací, ale cenou za tento komfort je vysoké využití systémových prostředků. Tato nepříjemná vlastnost omezuje jejich využití v oblasti monitorování systému v reálném čase, kdy je vhodnější využít standardních funkcí Windows API. Na druhou stranu nebrání žádná z okolností využití WMI pro získávání detailních doplňkových informací nebo informací, které nejsou pomocí standardních funkcí Windows API přístupné kvůli systémem přiděleným oprávněním.

1.2.4 Monitorování systému pomocí WMI

Pomocí WMI lze monitorovat velkou škálu ukazatelů výkonu a parametrů systému. K jejich získávání se používá především jedna ze základních WMI tříd nazvaná "Win32 Classes". S její pomocí lze přistupovat k objektům, jež jsou nejzajímavější pro účel monitoringu. Zapouzdřuje velké množství podtříd, které lze rozdělit do těchto kategorií:

Hardwarové objekty počítače	Třídy v této kategorii mají na starosti správu a získávání informací o hardware počítače.
Softwarové objekty počítače	Třídy v této kategorii slouží ke správě softwarového vybavení počítače, nainstalovaných aplikací a informací o nich.
Objekty operačního systému	Třídy sloužící pro administraci operačního systému a získávání systémových informací.
Čítače výkonu	Třídy v této kategorii zapouzdřující objekty pro monitorování výkonu a využití systému. Data z čítačů výkonu jsou poskytována v předformátované i v nezpracované podobě.
Správa WMI	Třídy umožňující administraci WMI služeb.
Pomocné bezpečnostní třídy	Třídy poskytující služby a informace z oblasti zabezpečení systému.

Prvním krokem vedoucím k získání informací je připojení k WMI službě. Tato operace je závislá na použitém vývojovém prostředí. Následně je nutno zaslat službě WMI dotaz ve formě WQL dotazu, na nějž reaguje služba navrácením datové struktury naplněné požadovanými vlastnostmi objektu a jejich hodnotami. Strukturu je nutno následně vhodným způsobem zpracovat pro další využití.

1.2.4.1 Informace o systému pomocí WMI

K získání základních i pokročilých informací o systému se využívají podtřídy a objekty z třídy "Win32". Příkladem takových informací mohou být informace, jako jméno a verze použitého systému, jméno osoby a název společnosti, na kterou je systém registrován, verze integrovaného opravného balíčku a další. Pro obdržení těchto základních informací je možno využít podtřídu nazvanou "Win32_OperatingSystem".

1.2.4.1.1. Zadávání a formát WQL dotazu pro získání informací o systému

Po připojení k WMI rozhraní je třeba zaslat WQL dotaz ve formatu: "SELECT * FROM Win32_ComputerSystem", kde lze dotaz modifikovat stejně jako běžné SQL dotazy, aby navracel jen vybrané informace.

1.2.4.1.2. Výčet důležitých informací o systému získatelných pomocí WMI

Tato podtřída obsahuje velké množství informací, jako příklad je uvedeno několik nejzajímavějších:

Version (string)	Obsahuje číslo veze Windows.
InstallDate (datetime)	Obsahuje časové informace o datu instalace operačního systému.
SerialNumber (string)	Obsahuje sériové číslo operačního systému Windows.
RegisteredUser (string)	Obsahuje jméno osoby, na niž je systém registrován.
Organization (string)	Obsahuje informaci o jménu organizace, na niž je systém registrován.
OSLanguage (integer)	Obsahuje identifikační číslo jazykové varianty operačního systému.
ServicePackMajorVersion (integer)	Udává verzi nainstalovaného opravného balíčku servicepack.
ServicePackMinorVersion (integer)	Udává podverzi nainstalovaného opravného balíčku servicepack.
LastBootUpTime (datetime)	Obsahuje časové informace, udávající délku běhu systému od startu počítače.
NumberOfUsers (integer)	Udává počet uživatelů přihlášených v systému.
SystemDirectory (string)	Obsahuje cestu k systémovému adresáři system32, například: "C:\WINDOWS\SYSTEM32".
WindowsDirectory (string)	Obsahuje cestu k adresáři, v němž je nainstalován operační systém Windows.
SystemDrive (string)	Obsahuje název jednotky, na níž je nainstalován operační systém.

Informace takto získané jsou uložené v jedné struktuře, což je velmi výhodné oproti jejich zdlouhavému získávání za užití standardních funkcí Windows API. Vzhledem k tomu, že se tyto údaje v průběhu času velmi málo mění, není problémem o něco vyšší skokové zatížení systému při jejich shromažďování.

Pomocí třídy “Win32_OperatingSystem” lze získat i informace o velikostech a využití různých typů pamětí operačního systému, což je velmi užitečné pro monitorování systému:

FreePhysicalMemory (integet64)	Udává velikost volné fyzické paměti v bytech.
FreeSpaceInPagingFiles (integer64)	Udává velikost volného místa ve stránkovacím souboru v bytech.
FreeVirtualMemory (integer64)	Udává velikost volné virtuální paměti v bytech.
SizeStoredInPagingFiles (integer64)	Udává obsazenost stránkovacího souboru v bytech.
TotalSwapSpaceSize (integer64)	Udává celkovou velikost stránkovacího souboru v bytech.
TotalVirtualMemorySize (integer64)	Udává celkovou velikost virtuální paměti systému v bytech.
MaxProcessMemorySize (integer64)	Udává maximální velikost operační paměti, která může být přidělena procesu.

Výše zmíněné informace mají vlastnost, že se velmi často mění v průběhu času a je nutno zajišťovat jejich periodickou obnovu. Zde nastává problém s používáním WMI služby kvůli velkému využití prostředků systému. Je tedy vhodné získávat tento typ informací klasickou cestou z rozhraní Windows API.

Třída “Win32_OperatingSystem” obsahuje funkce “Shutdown“ pro vypnutí počítače, “Reboot“ pro jeho reset a “SetDateTime“ pro nastavení systémového času.

1.2.4.2 Informace o systémových prostředcích

Informace o míře využití systémových prostředků lze získávat obdobnou cestou jako v případě základních informací o systému. Odlišné jsou pouze třídy, k nimž budeme přistupovat. V následujícím textu budou stručně popsány ty třídy, pomocí nichž lze přistupovat k informacím o běžících procesech, službách a hardwarových součástech počítače.

1.2.4.2.1. Získání informací o procesech pomocí WMI

Pomocí WMI služby lze získat komplexní informace o procesech běžících v systému a to dokonce i o takových, ke kterým by nebyl poskytnut přístup při získávání informací klasickou cestou za pomoci Windows API funkcí.

Informace o procesech jsou dostupné za pomoci třídy nazvané “Win32_Process”. Jsou to informace charakterizující proces a informace, které udávají, jakou mírou se podílí na využití prostředků systému.

WQL dotaz se po připojení ke službě WMI zadává ve formátu “SELECT * FROM Win32_Process“, což má za následek odpověď systému obsahující všechny dostupné informace, jež má třída “Win32_Process“ pro všechny procesy. Tuto strukturu je dále třeba funkcí pro vyčíslení

zpracovat a rozdělit informace tak, aby odpovídaly jednotlivým procesům. Dotaz lze modifikovat stejně jako běžný dotaz v jazyce SQL, například jej lze použít ve formátu "SELECT * FROM Win32_Process where ProcessId=0", což má za následek získání informací pro proces s identifikačním číslem 0, který je v tomto případě vždy tzv. "Systém idle process".

Ze zajímavých informací o procesu zjistitelných za pomocí WMI můžeme jmenovat například tyto:

Caption (string)	Proměnná obsahující název procesu.
ExecutablePath (string)	Obsahuje úplnou cestu k exe souboru procesu. Například: "C:\WINDOWS\EXPLORER.EXE".
ProcessId (integer)	Obsahuje identifikační číslo procesu.

Výčet informací, které mohou sloužit ke sledování využití systémových prostředků a výkonu procesu je širší než v případě základních informací o systému:

Priority (integer)	Udává informaci o aktuální prioritě procesu v systému.
ThreadCount (integer)	Nese informaci o počtu vláken procesu.
WorkingSetSize (integer)	Udává informaci o velikosti fyzické paměti alokované procesem v bytech.
PeakWorkingSetSize (integer)	Udává maximální míru využití fyzické paměti systému v bytech.
VirtualSize (integer64)	Udává informaci o velikosti virtuální paměti alokované procesem v bytech.
PeakVirtualSize (integer64)	Udává maximální míru využití virtuální paměti procesem v bytech.
PageFileUsage (integer)	Udává míru využití stránkovacího souboru v bytech.
PeakPageFileUsage (integer)	Udává maximální míru využití stránkovacího souboru.
PageFaults (integer)	Nese informaci o počtu výpadků stránek stránkovacího souboru.
ReadOperationCount (integer64)	Obsahuje informaci o počtu operací čtení.
ReadTransferCount (integer64)	Obsahuje informací o množství dat přeneseném při provádění operací typu čtení.
WriteOperationCount (integer64)	Obsahuje informaci o počtu operací zápis.
WriteTransferCount (integer64)	Obsahuje informací o množství dat přeneseném při provádění operací typu zápis.
OtherOperationCount (integer64)	Obsahuje informaci o operacích provedených procesem, které nejsou zápis nebo čtení.
OtherTransferCount (integer64)	Obsahuje informaci o množství dat přeneseném při provádění operací jiných než čtení nebo zápis.

Třída “Win32_Process“ obsahuje funkce pro řízení procesů. Z nejdůležitějších je to metoda “SetPriority“ s jejíž pomocí je možno procesu nastavit jeho prioritu v systému a metoda “Terminate“, sloužící k ukončení zvoleného procesu.

1.2.4.2.2. Získání informací o službách systému Windows pomocí WMI

Pro získání seznamu služeb běžících v systému a informací o nich představuje použití WMI ideální cestu. Kromě snadného přístupu k informacím o jednotlivých službách je možno provádět snadno i jejich administraci.

Přístup ke službám je možno realizovat pomocí WMI za využití třídy “Win32_Service”, jež nabízí kromě základních informací o službě či službách i informace o stavu, v němž se momentálně nacházejí a informace o jejich dalších parametrech, které mohou sloužit k monitoringu systému.

Pro získání informací je třeba se připojit ke službě WMI a následně zaslat požadavek formou WQL dotazu ve tvaru “SELECT * FROM Win32_Service“. Odpovědí je systémem zaslaná struktura, obsahující informace o všech službách v systému, kterou je třeba funkcemi pro enumeraci vyčíslit a dále vhodně zpracovat. Pro získání informací o konkrétní službě stačí upravit WQL dotaz podobně, jako v předchozím případě doplněním klauzule “WHERE“.

Touto cestou lze opět velmi snadno získat přístup k důležitým a z hlediska správy a monitorování systému zajímavým informacím:

Name (string)	Obsahuje název služby.
DisplayName (string)	Obsahuje popis služby.
PathName (string)	Obsahuje plnou cestu k binárnímu souboru, kterým je služba implementována.
ProcessId (integer)	Nese informaci o identifikačním čísle procesu, jež je vlastníkem služby.
AcceptPause (boolean)	Udává, jestli je možno službu pozastavit.
AcceptStop (boolean)	Udává, jestli je možno službu zastavit.
ServiceType (string)	Nese informace o druhu služby.
Started (boolean)	Udává, zda je služba spuštěná nebo se nachází v nečinném stavu.
StartMode (string)	Udává informaci o režimu spouštění služby.
State (string)	Obsahuje informaci o stavu služby. Služba může být v některém ze stavů: "Stopped", "Start Pending", "Stop Pending", "Running", "Continue Pending", "Pause Pending", "Paused" nebo "Unknown".
CheckPoint (integer)	Obsahuje informaci o počtu změn běhu služby.
ErrorControl (string)	Udává informaci o režimu zaznamenávání chybových událostí do logovacího souboru.

Pro manipulaci se službami je implementováno ve třídě “Win32_service“ hned několik funkcí. Z těch, jež umožňují ovlivnit běh služeb, je možno jmenovat například metody “StartService“, umožňující spuštění a “StopService“ pro zastavení zvolené služby. Metoda “PauseService“ slouží k pozastavení služby a naopak za pomoci metody “ResumeService“ lze službu obnovit z pozastaveného stavu. Změnit režim spouštění služby je umožněno voláním metody “ChangeStartMode“. Pro manipulaci se službami jsou implementovány metody “Create“ sloužící k registraci služby, “Delete“ umožňující odstranění zvolené služby ze systému a metoda “Change“, která umožňuje změnu parametrů existující služby.

1.2.4.2.3. Získání informací o hardware pomocí WMI

WMI nabízí nepřeberné množství tříd, umožňujících získání informací o instalovaném hardwarovém vybavení. Je možno touto cestou sledovat zařízení a periferie počínaje seriovým portem za pomoci třídy “Win32_SerialPort“, přes datová uložiště počítače například za pomoci třídy “Win32_PhysicalMedia“ a monitorováním prostředků procesoru či základní desky konče.

V oblasti hardware jsou výhody této služby nesporné. Jedná se o velmi efektivní způsob přístupu k informacím o instalovaných zařízením bez nutnosti využívat nízkoúrovňové funkce, které nemusejí být navíc vždy zcela bezpečné a jejich implementace se může lišit v různých verzích systému. Velkou výhodou WMI je, že informace pro všechny druhy zařízení, pro něž jsou vytvořeny třídy je možno získat stejným způsobem, což není u nízkoúrovňových funkcí myslitelné.

1.2.5 Klady a zápory používání WMI služeb

Jak již bylo zmíněno v předchozích odstavcích, WMI je službou poskytující velmi velké množství zajímavých informací, navíc velmi komfortní a bezpečnou cestou oproti klasickému řešení pomocí Windows API, ale jak již tomu bývá, každá mince má dvě strany. Cenou za tyto výhody je vyšší zatížení systému v momentě získávání požadovaných informací, proto je nezbytné důkladně zvážit při implementaci aplikací, jaké informace budou shromažďovány za pomoci WMI a pro které bude zvolena klasická cesta, aby nedocházelo ke zbytečnému zatížení zdrojů systému a celkovému zpomalení jeho chodu. Další z drobných nevýhod WMI je její vlastnost, že je integrována jako součást systému a tak její rychlost závisí z velké části na celkové kondici systému Windows. Ovšem i přes tyto problémy je občas služby WMI vhodné a někdy i nezbytné využít. Příkladem mohou být úkony související se získáním informací týkajících se procesů, k nimž je potřeba disponovat neomezenými právy přístupu k systému. V současnosti je totiž snaha zvyšovat zabezpečení systému omezováním práv běžných aplikací, zvláště pak se tento trend projevuje u nejnovější verze operačního systému Windows Vista.

1.3 Problémy s kompatibilitou

Tato krátká kapitola je věnována problémům s kompatibilitou při využívání monitorovacích funkcí. Budeme brát v potaz pouze rodinu operačních systémů založených na technologii NT, které jsou podporovány ze strany výrobce. U rodiny systémů na bázi Windows 95 by se množina možných komplikací velmi rychle rozrostla z důvodu odlišné filozofie systému.

1.3.1 Problémy s přístupovými právy

Jako v každém systému, tak i ve Windows existují různé úrovně uživatelských oprávnění, na jejichž základě jsou poskytovány systémem služby. Existují funkce systému, ke kterým je dostatečné oprávnění standardního uživatele, ale jejich provádění bude jistým způsobem omezené. Existují ovšem i funkce, které bez nejvyššího oprávnění není možno využít. Příkladem tohoto mohou být některé funkce rozhraní Windows API, sloužící pro získání přístupu k procesům běžícím v systému. Jestliže aplikace disponuje pouze omezenými uživatelskými právy, je umožněno přistupovat pouze k procesům, jejichž vlastníkem je přihlášený uživatel, v případě že je přihlášen uživatel s neomezenými právy, je umožněn přístup ke všem procesům systému bez výjimky. Tuto vlastnost lze ovšem obejít použitím WMI namísto klasických Windows API funkcí, bohužel to není vždy možné a tento přístup není vždy efektivní. Tento aspekt je nutné zohlednit při tvorbě aplikací.

1.3.2 Problémy s bezpečnostními nástroji

Při tvorbě monitorovacích aplikací za použití systémových funkcí mohou nastávat místy až neřešitelné problémy při zabezpečení systému pomocí nových nástrojů společnosti Microsoft pro zvýšení bezpečnosti. Problém nastává v případě použití nejnovějšího operačního systému Windows Vista. Výrobcem je implementován a ihned po nainstalování systému aktivován bezpečnostní nástroj s názvem “User Account Control“, což v překladu znamená “Nástroj pro řízení uživatelských účtů“. Účelem tohoto mechanizmu je zvýšit bezpečnost systému omezením práv uživatele, kdy dochází i v případě účtu s neomezenými právy k jejich značnému omezení. V důsledku dochází k faktu, že systémem jsou všem uživatelů bez rozdílu přidělena pouze standardní uživatelská práva, provozovaným aplikacím není umožněn zápis do systémových lokací disku ani registru a přístup k systémovým funkcím je do značné míry omezen. Je omezen například přístup k procesům pomocí funkcí Windows API, použití nedokumentovaných funkcí tímto mechanismem je dokonce zcela vyloučeno.

1.3.3 Problémy s různými verzemi Windows

Jako každý software, tak i operační systém společnosti Microsoft se neustále vyvíjí a tento aspekt s sebou může přinášet jisté problémy. Při využívání funkcí systému i služeb nabízených službou WMI může nastat problém, kdy v jednotlivých verzích jsou některé vlastnosti implementovány různě, nebo je jejich výčet odlišný.

Dobrým příkladem toho může být jedna ze základních funkcí Windows API nazvaná "OpenProcess". Ve Windows Vista došlo k přidání jedné úrovně uživatelských oprávnění. Pří volání funkce "OpenProcess" se zadává požadované oprávnění jako jeden z jejích parametrů. V dřívějších verzích systému bylo oprávnění " PROCESS_QUERY_INFORMATION" dostatečným pro přistup k procesu a k následnému shromažďování informací, ve Windows Vista ovšem přibylo oprávnění "PROCESS_QUERY_LIMITED_INFORMATION", které je specifické právě pro tento systém. Bez použití tohoto oprávnění není aplikaci umožněn přístup k žádnému procesu. Tento fakt může mít za následek nefunkčnost dříve vytvořených aplikací v nové verzi systému.

U služby WMI se s dalším vývojem neustále zvyšuje počet objektů v její databázi, na tento aspekt je při tvorbě aplikací nutno brát ohled a zabezpečit používání pouze takových objektů, jež se nachází ve všech verzích systému, pro něž je aplikace koncipována.

Z hlediska existence více verzí operačního systému je velmi nebezpečné a společností Microsoft nedoporučované používání nedokumentovaných funkcí operačního systému a skrytých funkcí rozhraní Windows API. Tyto funkce nemusí být ve všech verzích operačního systému použitelné a jejich implementace se může někdy i velmi výrazně odlišovat.

2 Aplikace pro monitorování prostředků operačního systému

2.1 Klíčové funkce vytvořené aplikace

V této krátké kapitole bude věnována pozornost klíčovým vlastnostem vytvořené aplikace během praktické části bakalářské práce, bude popsáno, jaké informace prezentují jednotlivé moduly uživateli a bude zmíněn způsob, kterým tak činí.

2.1.1 Přehled základních informací

Základní modul aplikace slouží k získání základního přehledu o počítačovém systému. Jsou zobrazeny základní informace o hardwarové i softwarové konfiguraci. Je přítomen přehled základních ukazatelů vytížení systémových prostředků, pomocí grafu je možno získat přehled o jejich průměrném vytížení během různých časových období.

2.1.2 Monitor procesů

Mezi základní implementované moduly patří správce procesů operačního systému. Jedná se o modul poskytující detailní přehled o běžících i zastavených procesech. Tento modul je tvořen seznamem procesů, ve kterém jsou prezentovány základní informace o míře, jakou zatěžují systém, pro detailnější informace je pro vybraný proces zobrazována tabulka obsahující podrobné informace. Součástí modulu je graf, díky kterému může uživatel získat přehled o činnosti různých procesů v různých obdobích. Procesy je v tomto modulu možno i spravovat. Pro ty, ke kterým má uživatel umožněn systémem přistup, je možno nastavovat prioritu jejich běhu, případně je ukončovat.

2.1.3 Monitor služeb

Modul sloužící k informování uživatele o instalovaných službách operačního systému. Nabízí jejich přehled a informace o stavu, ve kterém se nacházejí. Uživateli jsou prezentovány v tabulkové formě i detailnější informace o službách a je mu umožněno v případě dostatečných oprávnění ovlivnit jejich chod.

2.1.4 Textové výstupy

V aplikaci je implementován modul mající na starosti vytváření textového výstupu z množiny naměřených hodnot. Tento modul je konfigurovatelný a uživatel může zvolit rozsah informací, které požaduje ve výstupním souboru.

2.1.5 Alarmy

Modul alarmu slouží ke sledování hodnot zatížení systémových prostředků, stavu služeb a procesů. Je možno nastavit akce, které se provedou při splnění definovaných podmínek. Uživatel může být díky tomuto modulu včas varován o chování sledovaného ukazatele, případně může být učiněno opatření formou restartování systému.

2.2 Návrh a implementace aplikace

Cílem praktické části bakalářské práce je navržení a implementace nástroje pro monitorování prostředků operačního systému Microsoft Windows. Cílem této kapitoly je popsat a přiblížit mechanizmy a postupy jichž bylo využito při implementaci a návrhu řešení.

2.2.1 Požadavky kladené na aplikaci

Ze zadání bakalářské práce je patrno, že cílem její praktické části je vytvoření nástroje umožňujícího monitorování systémových prostředků, který by sledoval stav počítačového systému v reálném čase, zpracovával naměřené hodnoty a dále je vhodným způsobem prezentoval uživateli. Aplikace je určená pro provoz na operačních systémech Microsoft Windows.

Z tohoto hlediska je vytvořená aplikace koncipována pro operační systémy podporované společností Microsoft, což jsou Windows verze 2000 a novější. Tento aspekt má zásadní vliv na návrh celé aplikace a na výběr mechanizmů pro shromažďování informací o systému. Tím pádem lze využít moderních technologií a služeb, jejichž aplikace by na zastaralých systémech na bázi Windows 9x nebyla myslitelná.

Vzhledem ke skutečnosti, že se jedná o aplikaci stále běžící na pozadí, zásadními požadavky na navrženou aplikaci jsou dlouhodobá stabilita a nízká úroveň zatížení prostředků počítačového systému její monitorovací činností.

2.2.2 Vývojové prostředí

Jako vývojové prostředí pro tvorbu praktické části bakalářské práce bylo zvoleno Codegear Delphi. Jedná se o prostředí již dříve vyvíjené společností Borland pod novou značkou. Codegear Delphi je propracované prostředí určené pro rychlou a efektivní tvorbu aplikací s grafickým uživatelským rozhraním pro operační systém Microsoft Windows. V komerční verzi je v současnosti nabízeno pod názvem "Codegear Delphi 2007" a ve verzi určené pro volné využití pod názvem "Turbo Delphi". Hlavními rozdíly mezi volně šiřitelnou a komerční verzí jsou patrné z nabízené škály vestavěných komponent. Volně šiřitelná verze disponuje pouze několika desítkami základních komponent, naopak chybí pokročilé komponenty pro práci s databázemi, síťovými prostředky a další komponenty pro tvorbu specializovaných a pokročilých aplikací. Obě verze jsou kompatibilní s produktem Delphi, který dříve vyvíjela společnost Borland.

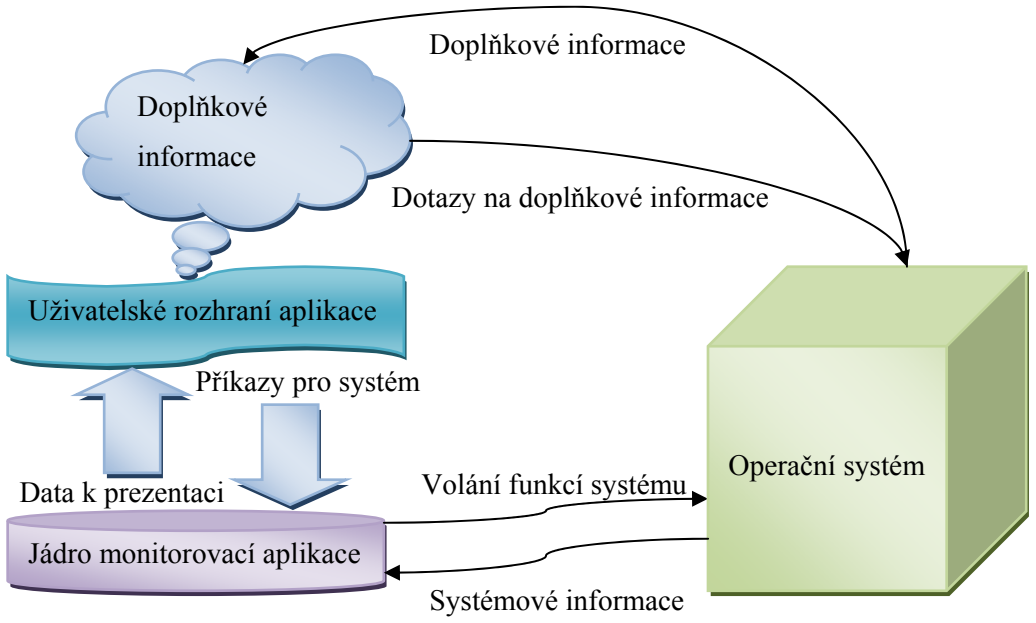
Vytvořená aplikace využívá pouze základních komponent a je tedy bez problému přeložitelná pomocí volně šiřitelné verze prostředí, jako je například "Delphi 7 Personal", kde byla její funkčnost také plně testována.

Prostředí Delphi bylo zvoleno pro svoji výše zmiňovanou uživatelskou přívětivost, snadnost návrhu a možnosti tvorby efektivního uživatelského rozhraní vytvářené aplikace. K jeho volbě přispěl i fakt, že umožňuje bezproblémový přístup k funkcím operačního systému Windows pomocí vestavěných knihoven. Přístup k informacím nabízených skrze službu WMI je možno bez zásadních problému realizovat pomocí schopnosti přistupovat k ActiveX prvkům. Mezi další nesporné výhody paří celosvětově velká podpora Delphi komunity a snadný přístup k velkému množství volně šiřitelných komponent využitelných při tvorbě aplikací.

Jako operační systém, na kterém byla aplikace vyvíjena, byl zvolen Microsoft Windows Vista v 64 bitové verzi. Po dobu vývoje nenastal žádny problém kompatibility operačního systému s vývojovým prostředím.

2.2.3 Základní koncepty navržené aplikace

Základním konceptem vytvořené aplikace je rozložení její funkcionality do 3 vrstev. Obsahem 1. vrstvy je vytvoření nezávislého jádra běžícího na pozadí, jež slouží ke shromažďování informací o počítačovém systému a k jejich zpracování. Druhá vrstva je tvořena grafickým uživatelským rozhraním, které má na starosti interakci s uživatelem a prezentaci získaných hodnot vhodným způsobem. Třetí vrstva je tvořena mechanismem sloužícím pro obstarávání a zpracovávání doplňkových informací, jež není možno získávat v reálném čase bez velkého zatížení systému.



Obrázek 1: Koncept aplikace

Tento koncept předurčuje aplikaci k využití vláken, kdy jednotlivé vrstvy pracují nezávisle na sobě. Jejich využití je takřka nezbytné v případě třetí vrstvy, kdy přichází na řadu zdlouhavé a z hlediska systémových zdrojů náročné získávání informací pomocí služby WMI. Bez implementace vláken by nastala situace, kdy by byl značně snížen komfort ovládání uživatelského rozhraní, aplikace by přestala na několik okamžiků reagovat na podněty uživatele i systému a v krajním případě by mohlo dojít k omezení schopností jádra shromažďovat informace o systému. Tento fakt by mohl způsobit nekonzistenci monitorovaných dat a zkreslit značným způsobem celou statistiku.

2.2.3.1 Koncept jádra aplikace

Jádro aplikace je koncipované jako část programu běžící ve vlastním vlákně, které má na starosti získávání informací o monitorovaném systému výhradně za užití systémových funkcí rozhraní Windows API. Získávání informací probíhá periodicky. Každou sekundu jsou obnoveny údaje o prostředcích celého systému i jednotlivých běžících procesů. Následně po získání čerstvých informací

dochází k aktualizacím statistik, jež uchovávají data pro pozdější využití. Jádro je základním kamenem aplikace a proto je nezbytné zajistit jeho bezproblémovou funkčnost a dodržení periodičnosti časových intervalů, v nichž dochází k měření.

2.2.3.2 Koncept uživatelského rozhraní aplikace

Uživatelské rozhraní monitorovací aplikace je navrženo se snahou dosáhnout dostatečné přehlednosti a uživatelské přívětivosti aplikace. Jeho hlavním úkolem je prezentovat naměřené údaje formou vhodného grafického výstupu uživateli a umožnit mu tak získání přehledu o sledovaném systému. Výstupy jsou realizovány pomocí grafů, které umožňují zobrazit monitorované údaje v různých časových obdobích. Tyto údaje jsou předzpracovány jádrem a jsou uživatelskému rozhraní předávány ve vhodném formátu k přímé prezentaci. Pro získávání doplňkových informací se využívá služeb třetí vrstvy, získané informace jsou následně uživatelským rozhraním prezentovány uživateli.

2.2.3.3 Koncept vrstvy pro získávání informací z WMI

Třetí vrstva má za úkol získávat informace s využitím služby WMI. Tato služba je bohužel v některých případech neúměrně pomalá a doba uplynulá od zadání požadavku po získání informací je při špatné kondici operačního systému v rozmezí sekund. Proto bylo nezbytné implementovat tuto vrstvu. Jedná se o čas programového kódu, který je vykonáván mimo hlavní kontext běhu aplikace. Jeho úkolem je po obdržení požadavku od hlavní aplikace vytvořit připojení k WMI službě operačního systému Windows a pomocí WQL dotazu se pokusit o získání požadovaných informací ze systému. Následně je provedena enumerace získané struktury a výsledky jsou ve vhodném formátu předány zpět hlavnímu programu a prezentovány uživatelským rozhraním. Tento mechanizmus výrazně zvyšuje uživatelský komfort aplikace a umožňuje snadnější aplikaci WMI služeb.

2.2.4 Vlastní implementace aplikace

Aplikace byla navržena a následně implementována podle výše zmiňovaného konceptu za použití vývojového studia Delphi. V této části budou přiblíženy aspekty související s implementací jejích jednotlivých částí.

První fází byla analýza problému a následoval návrh jeho řešení. V následující fázi probíhala implementace projektu, kdy bylo vytvořeno základní jádro aplikace, jež obsahovalo z počátku jen základní funkce nezbytné pro běh aplikace. Během další fáze došlo k implementaci základního uživatelského rozhraní a byla postupně přidávána další funkcionalita jádru aplikace. Následovala fáze, během které bylo zdokonalováno uživatelské rozhraní, a byly postupně přidávány další funkce a důkladně testována funkčnost celé aplikace.

2.2.4.1 Implementace jádra aplikace

Jádro aplikace je tvořeno nekonečným cyklem, jehož účelem je získávání aktuálních informací o systému. Během tohoto cyklu je třeba vykonat mnoho úkonů. Jádro zpracovává velké množství statistických dat a k jejich uložení po zpracování používá vlastní datové struktury. Pomocí nich jsou data dále zpřístupněna dalším vrstvám, které k nim takto mohou přistupovat. Datová struktura nese informace o základních systémových prostředcích a o aktivních i neaktivních procesech. Ke každému sledovanému systémovému prostředku či procesu existuje ve struktuře pomocná struktura sloužící k uchovávání statistických dat. Data vypovídají o vytížení sledovaného prostředku v určitý čas a slouží v dalších vrstvách aplikace k vytváření grafických výstupů ve formě grafu nebo textových výstupů formou výpisu do souboru. Stavba datové struktury je patrná z níže uvedeného diagramu.

Jádro během svého cyklu sbírá informace o systému a o systémových prostředcích za využití Windows API funkcí. V první fázi jádro získá ze systému seznam běžících procesů. Tato činnost je implementována funkcí "getProcessList()" v souboru funkcí "functions.pas". Je pořízen tzv. snímek aktuálního stavu procesů za pomocí funkce "CreateToolHelp32SnapShot()". Následně je pak snímek systému vyhodnocen enumerační funkcí, všechny nalezené procesy jsou zapsány do pole datových struktur nesoucích informace o názvu procesu a jeho identifikačním čísle. Tato datová struktura obsahuje vnořený datový typ, ve kterém budou později shromažďovány statistické údaje o využití jednotlivých systémových prostředků procesem v různých časových obdobích. Celá datová struktura je navrácena jádru. Následně je získaný seznam jádrem porovnán s předešlým seznamem procesů. Dále neexistující procesy jsou ze seznamu přesunuty do seznamu neaktivních procesů, kde čekají na svoje opětovné spuštění.

V další kroku proběhne získání informací o využití procesoru procesem. Jestliže se jedná o proces, který nově přibyl do seznamu, je volána funkce, jež obstará vytvoření pomocné datové struktury nesoucí informace sloužící k výpočtu zatížení procesoru. V případě že již proces má vytvořen ukazatel z dřívější doby, dojde k volání funkce "GetCpuUsageForProcess()". Tato funkce zajistí výpočet zatížení CPU. Funkce potřebuje pro svoji činnost strukturu obsahující naposledy naměřené hodnoty. Dojde k získání aktuálního uživatelského času a času jádra věnovaného procesu. Aktuální vytížení je pak vypočítáno za pomocí vztahu:

$$\frac{(\text{aktualni\_casCPU} - \text{predchozi\_casCPU})}{\text{uplynuly\_}cas}$$

Následně je tato hodnota navrácena jádru, kde poslouží při tvorbě statistiky.

Následně přichází na řadu informace o velikostech paměti alokované procesem. K těmto informacím je umožněn přístup pomocí funkce "GetProcessMemorySize()", implementované v souboru funkcí "functions.pas". Funkce pracuje se strukturou "PROCESS_MEMORY_COUNTERS", implementovanou v rozhraní Windows API. Nejdříve dojde

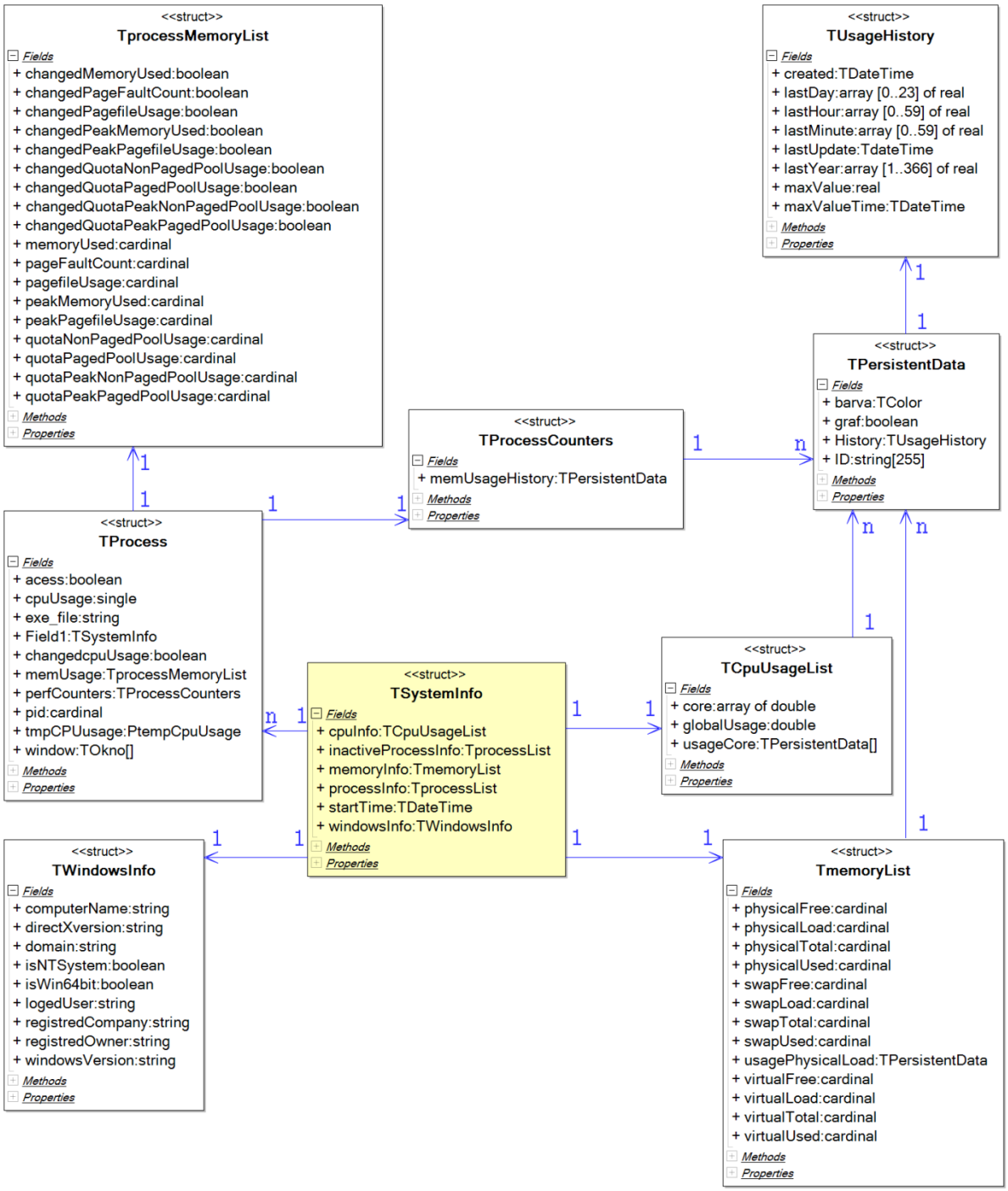
k pokusu o získání přístupu k procesu a případě úspěchu je volána Windows API funkce "GetProcessMemoryInfo()", která naplní strukturu získanými informacemi. Jsou navráceny informace o fyzické paměti a stránkovacím souboru, jejich aktuální a hraniční hodnoty.

Během dalšího kroku dojde k obnovení statistik pro ukazatele zatížení pro jednotlivé procesy. Jedná se o ukazatel využití procesoru, fyzické paměti a virtuální paměti. Tuto akci má na starosti funkce "StatCountersUpdate()" implementovaná v souboru funkcí "functions.pas". Funkce manipuluje s pomocnou datovou strukturou, která obsahuje statistické údaje zatížení systému pro jednotlivá časová období. Obsahuje celkem 4 pole. V prvním poli se archivují přesné naměřené hodnoty za posledních 60 sekund, následně je pak z těchto hodnot počítán aritmetický průměr, který dále figuruje v druhém poli o délce 60 prvků, jež reprezentuje jednotlivé minuty poslední uplynulé hodiny. Následuje obdobný postup, kdy aritmetický průměr prvků poslední hodiny tvoří prvek v třetím poli o délce 24. Toto pole obsahuje hodinové průměry naměřených hodnot za poslední den. Analogickým způsobem dochází k vytvoření čtvrtého pole, jež má délku 365 prvků a uchovává denní průměry za uplynulý rok. Návratovou hodnotou funkce je výše zmíněná struktura doplněná o časové razítko poslední změny dat.

Předposledním krokem je získání celkového vytížení procesoru a jeho jader. Je volána funkce "GetPerCoreCpuUsage()", která využívá externí volně šiřitelné knihovny "uCpuUsage.pas" z důvodu nutnosti využít nedokumentované funkce Windows API k získání informací. Po získání informací dochází, stejně jako v případě procesů, k obnovení statistických údajů, jejich přepočítání a uložení ve struktuře.

Poslední informace, které budou v cyklu získávány, jsou informace o celkovém využití stránkovacího souboru, fyzické a virtuální paměti. Za tímto účelem byla implementována funkce s názvem "GetMemoryInfo()". Funkce spolupracuje s rozhraním Windows API, pomocí kterého přistupuje k jeho funkci "GlobalMemoryStatus()", jež navrací informace o alokované paměti ve struktuře "_MEMORYSTATUS", definované v operačním systému Windows. Následuje zpracování získaných dat, obnovení statistky a navrácení struktury jako návratové hodnoty funkce.

Všechny výše zmíněné kroky jsou cyklicky opakovány každou vteřinu běhu aplikace, aby byly zaručeny aktuální informace o systému.



Obrázek 2: Diagram tříd

Diagram tříd na obrázku 2 zobrazuje stavbu datové struktury “TSystemInfo“, s níž pracuje jádro.

2.2.4.2 Implementace uživatelského rozhraní

Uživatelské rozhraní je implementováno za užití standardních komponent vývojového prostředí Delphi. Většinu času běží aplikace na pozadí operačního systému, kdy jedinou viditelnou částí je ikona v systémové liště. Při obnovení okna aplikace je k dispozici její uživatelské rozhraní. Hlavním úkolem rozhraní je prezentovat uživateli vhodným způsobem naměřená a jádrem zpracovaná data. Rozhraní je tvořeno hlavním oknem, které obsahuje v levé části tlačítkové menu aplikace a zbytkem okna, určeným pro zobrazení informací zvoleného typu.

Po stisku prvního přepínače menu zobrazuje aplikace základní informace o počítačovém systému. Pomocí služby WMI jsou získány a následně zobrazeny informace o názvu a verzi použitého operačního systému, parametrů procesoru a informace o aktuálně přihlášeném uživateli. Ve spodní části jsou zobrazeny ukazatele pro základní systémové zdroje. Tyto informace jsou výstupem jádra aplikace a jsou prezentovány za použití grafu a číselných údajů, udávajících aktuální zatížení systému. Jsou zobrazeny výsledky monitorování zatížení procesoru, jeho jednotlivých jader, stránkovacího souboru, fyzické a virtuální paměti. Jednotlivým ukazatelům je možno měnit barvu a povolit či zakázat zobrazení v grafu. Graf obsahuje hodnoty pocházející ze statistik, tvořených jádrem aplikace a je možno nastavit časové období, které bude zobrazeno.

Druhé tlačítko menu slouží k zobrazení informací o procesech běžících v systému. Opět je částečně využito informací produkovaných jádrem aplikace i doplňkových informací získaných za pomoci služby WMI. Okno je z velké části zaplněno seznamem procesů, tvořeným klasickou základní komponentou seznamu s upravenými vlastnostmi vykreslování. Tento seznam je naplněn názvy běžících procesů a základními informacemi o míře, jakou zatěžují systém. Podobný seznam je vytvořen i pro procesy, které byly ukončeny a čekají na svoje opětovné spuštění, po kterém budou přesunuty do seznamu aktivních procesů a jádro bude pokračovat ve vytváření jejich statistiky. Spodní polovina okna je věnována doplňkovým informacím o procesech a grafu. Informace zde zobrazované se periodicky získávají pomocí služeb WMI. Po zvolení procesu je vytvořeno nové vlákno aplikace, v němž dochází ke shromažďování a zpracovávání informací pomocí služby WMI, následuje synchronizace s hlavní aplikací, kde dochází k prezentaci doplňkových informací. V nejspodnější části okna se nachází graf. V seznamu procesů lze povolit či zakázat zobrazení jednotlivých procesů, nastavit jejich barvu pro vykreslování v grafu a nastavit časové období, jež bude sledováno.

Pod třetí volbou menu se skrývají informace o službách operačního seznamu. Veškeré informace použité v tomto okně jsou získány za pomoci služby WMI. Velká část okna je tvořena seznamem služeb, ve kterém figurují jejich názvy, popisy, informace o aktuálním stavu a režimu jejich spuštění. Ve spodní části jsou zobrazeny detailní informace o zvolené službě a volby, umožňující uživateli nastavovat parametry režimu spouštění a běhu služby. Všechny informace zde prezentovaná jsou získávány v separátním vláknu kvůli náročnosti služeb WMI.

Čtvrtá položka menu nás přesune do okna sloužícího k exportování naměřených dat do textového souboru. Okno je tvořeno větším množstvím přepínačů, pomocí nichž lze ovlivnit množství detailů a několika seznamy sloužícími k výběru informací, které budou exportovány. Je využito implementovaných funkcí v knihovně "functions.pas", při exportu je využito podle zvoleného nastavení jak informací poskytovaných službou WMI, tak i souboru informací vytvořených jádrem aplikace.

Pod pátým tlačítkem menu se skrývá nastavení modulu alarmu. Ve vrchní části okna se nachází seznam vytvořených alarmů, které lze upravovat pomocí níže umístěných editačních polí. Do seznamu je možno přidávat další záznamy, s nimiž modul pracuje. Je možno vytvářet alarmy, které mají na starosti monitorování prostředků systému, procesů i služeb. Reakci, která bude následovat po splnění zvolené podmínky, je možno vybrat ze seznamu předdefinovaných akcí. K dispozici jsou akce informování uživatele, restart a vypnutí počítače. Spodní část okna je tvořena seznamem událostí alarmu, jež byly aktivovány. K implementaci tohoto modulu bylo využito jak funkcí WMI, tak i služeb systémového rozhraní Windows API.

2.2.4.3 Implementace mechanizmu získávání informací z WMI

Tento mechanismus je velmi důležitý pro zvýšení komfortu užívání aplikace a pro hladkost celkového jejího běhu. Je využíváno implementovaných funkcí v knihově "functions.pas" pro práci s WMI. Především se jedná o funkci "WMI_getProcessInfo()" pro zobrazování detailních informací o procesech, které jsou využity následně zejména ve správci procesů. Funkce "WMI_getServicesInfo()" je základním kamenem správce služeb. Informace o operačním systému a uživatelích, které jsou využívány v základním přehledu systémových informací a při vytváření textového souboru při exportování, jsou získávány voláním funkcí "WMI_getWindowsInfo()", "WMI_getUsersInfo()" a "WMI_getSystemInfo()". Všechny zmiňované funkce jsou poměrně náročné a zatěžují systémové prostředky, proto je nutno je používat s rozvahou. Z tohoto důvodu jsou funkce spouštěny v separátním vláknu, kde dojde k získání a zpracování informací. Následuje synchronizace s hlavní aplikací, během které dojde k prezentaci doplňkových informací do požadované části uživatelského rozhraní. Funkce se připojují ke službě WMI pomocí přístupu k prvkům ActiveX, obstarávají zaslání požadavku a následnou enumeraci získané datové struktury, jejich návratovými hodnotami jsou datové struktury obsahující požadované informace.

2.2.5 Testování aplikace

Aplikace byla velmi důkladně a dlouhodobě testována. Její funkčnost a správnost výstupů byla ověřována za pomoci dobrovolníků s použitím několika desítek testovacích systémů s různorodou konfigurací. Na základě jejich podnětů bylo upravováno jak uživatelské rozhraní, tak i jádro aplikace. Při tvorbě uživatelského rozhraní byl kladen důraz na pohodlí uživatelů při ovládání aplikace a tak jejich názory a poznatky byli velmi důležitým ukazatelem směru vývoje. Díky podnětům testovacích uživatelů docházelo i k drobným změnám ve funkcionalitě aplikace a byl jimi ovlivněn i výčet prezentovaných informací.

Aplikace byla testována na velkém spektru hardwarového i softwarového vybavení. Proběhlo testování na všech verzích operačního systému Windows počínaje jeho verzí 2000 až 64 bitovou variantu Windows Vista konče. Stranou nezůstali ani serverové operační systémy a ne příliš rozšířené varianty Windows jako jsou například Microsoft Windows Media Center Edition. Hardware testovacích systémů byl také velmi rozličný. Nejstarším testovacím hardwarem byl přenosný počítač poháněný 166MHz procesorem Intel Pentium a nejrychlejším testovacím systémem byl systém s operačním systémem Windows Vista a se čtyřjádrovým procesorem Intel Core2Quad. Na všech systémech byla aplikace bez problémů funkční, vyskytly se pouze menší problémy s výše zmíněnými omezenými uživatelskými právy u systému s operačním systémem Windows Vista se zapnutou funkcí řízení uživatelských účtů, kdy nebylo možno monitorovat všechny procesy běžící v operačním systému.

3 Závěr

Závěrem práce bych chtěl podotknout, že studium a následná implementace zvoleného tématu byly velmi zajímavé a i přes drobné nesnáze vedlo úsilí k požadovanému cíli. Práce na daném tématu byla velmi poučná a měla pro moji osobu velký přínos. Při zpracování této problematiky byly značným způsobem obohaceny moje znalosti rozhraní Windows API a byl jsem zasvěcen do problematiky související se službami systému určenými k poskytování informací a jejich užívání. Tyto zajímavé nabyté zkušenosti předurčují aplikaci k dalšímu vývoji a zdokonalování jejích funkcí.

3.1 Směr budoucího vývoje aplikace

Jako každá aplikace má i tento nástroj své chyby, které by v budoucnu měly být opraveny, či vlastnosti, které by mohly být lépe optimalizovány.

Jedná se především o možnost vylepšení práce se systémovými funkcemi, kdy by bylo možno implementovat funkce, které by umožňovaly přistupovat k systémovým zdrojům s vyšší úrovní oprávnění. Příkladem by mohl být přístup k procesům systému, který je odepřen v případě omezených práv. Východiskem by bylo vytvoření nových pomocných funkcí, používajících ke své činnosti nedokumentované funkce systémového rozhraní Windows API. Jsou to takové funkce, které jsou využívány například správcem úloh operačního systému. Jejich užití je ovšem velice komplikované už z jejich povahy. Jedná se totiž o funkce, které nejsou zcela bezpečné, jejich činnost není popsána v dokumentaci a na různých verzích operačního systému mohou být implementovány s drobnými odlišnostmi. Využití těchto funkcí by požadovalo velmi důsledné a dlouhodobé testování jejich správné funkčnosti na velkém počtu testovacích systémů.

Směrem budoucího vývoje by zajisté mohla být implementace nových modulů do monitorovací aplikace. Mohlo by se jednat například o moduly pracující se síťovými prostředky operačního systému a zaštiťující jejich monitoring. V průběhu dalšího vývoje by mohli být přidány nové moduly pro získávání informací o hardwarovém vybavení počítačového systému. Jejich implementace by byla snadno proveditelná za využití služby WMI.

Produktem budoucího vývoje by mohla být i síťová verze aplikace. Jednalo by se o verzi, která by umožňovala monitorovat počítačové systémy propojené v lokální síti. Ze specifikace služby WMI plyne její možná realizace, neboť WMI pracuje na principu vzdáleného volání procedur a tak by bylo možno přistupovat ke vzdáleným počítačům v síti bez nutnosti instalace nějaké zvláštní klientské aplikace a získávat z nich vzdáleně informace o jejich systémových prostředcích a ty následně zpracovávat aplikací a prezentovat je vhodnou formou například správci sítě.

3.2 Návaznost na ostatní zpracovávané projekty

Z hlediska ostatních prací, zpracovávaných v letošním roce se, jako velmi zajímavá témata, blízce související s touto tématikou, jeví projekty z oblasti operačních systémů. Velmi zajímavě se jeví téma "Aplikace k monitorování událostí OS Windows", jež má blízký vztah k mnou zpracovávané tématice. Další zajímavá témata pocházejí z oblasti webových aplikací. Velmi zajímavým projektem, který však není v současnosti realizován, by mohl být nástroj pro monitorování prostředků systému, disponující právě webovým uživatelským rozhraním.

Literatura

[1] Teixeira, S., Pacheco, X. *Mistrovství v Delphi 6*. Praha, Computer Press, 2002.

[2] Wilson, E. Microsoft Windows Scripting with WMI. Computer Press, 2005.

[3] Malina, P. PowerShell – Podrobný průvodce skriptováním. Computer Press, 2007.

[3] Microsoft, U.S.A.: Redmond, Microsoft Developer Network – MSDN Library, Dokument dostupný na URL http://msdn.microsoft.com (květen 2008)

Seznam příloh

Přílohy

1 Uživatelský manuál

1.1 Základní informace

1.1.1 Systémové požadavky

Monitorovací aplikace je běžným programem pro operační systém Microsoft Windows, ale vzhledem ke svému zaměření na oblast systémových služeb musejí být zajištěny určité prerekvizity.

Základním požadavkem aplikace je operační systém Microsoft Windows alespoň ve verzi 2000. Na starších verzích systému není možno aplikaci spustit z hlediska odlišnosti jader systémů.

Druhým požadavkem je nainstalovaná služba WMI, jež je součástí všech systémů založených na technologii NT. Služba musí být pro činnost monitorovací aplikace spuštěna.

Posledním požadavkem aplikace jsou dostatečná uživatelská práva přihlášeného uživatele. Uživatelům se standardními uživatelskými právy bude umožněna manipulace pouze s omezenou škálou systémových prostředků.

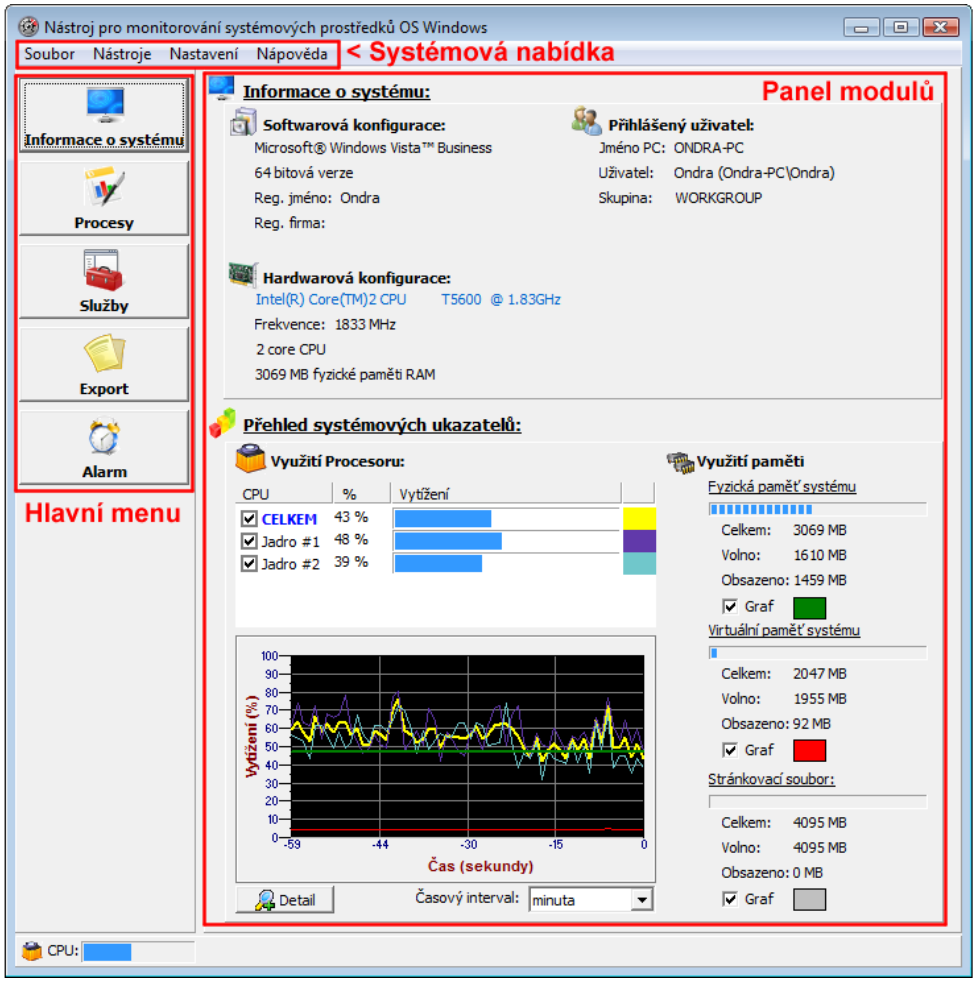
1.1.2 Spuštění aplikace

Aplikaci je možno spustit standardním způsobem jako každou jinou aplikaci pro operační systém Microsoft Windows. Aplikace nemusí být instalována, stačí pouze spustit, ale je vhodné, aby byla umístěna na uložišti, do něhož je možno zapisovat a to z důvodu archivace naměřených hodnot. V případě spuštění z media, na něž není povoleno či možno zapisovat, nebudou záznamy naměřených hodnot archivovány a po ukončení aplikace budou nenávratně ztraceny. Aplikace umožňuje pomocí svého nastavení určit způsob, jakým bude spouštěna. Je možno nastavit, aby byla aplikace automaticky spouštěna spolu s operačním systémem a definovat, zda bude po spuštění minimalizována v systémové liště, či zda bude viditelná.

1.2 Ovládání monitorovací aplikace

1.2.1 Hlavní okno aplikace

Hlavní okno aplikace je v horní části tvořeno systémovou nabídkou, s jejíž pomocí je možno s aplikací pracovat a nastavovat ji. Nabídka obsahuje položky “soubor“, “nástroje“, “nastavení“ a “nápověda“. Pod položkou “soubor“ se skrývají položky pro zálohování dat a ukončení aplikace. Zálohování dat umožňuje uložení datových souborů se statistickými daty na zvolené médium. Položka “nástroje“ obsahuje položky, jejichž aktivací se aplikace přepne na zvolený modul. Tato nabídka je ekvivalentem k postrannímu menu aplikace, jež bude zmíněno níže. Položka “nastavení“ aktivuje dialog, s jehož pomocí je možno nastavit chování aplikace. Pod položkou “nápověda“ se skrývají položky pro zobrazení informací o aplikaci a pro zobrazení tohoto uživatelského manuálu.



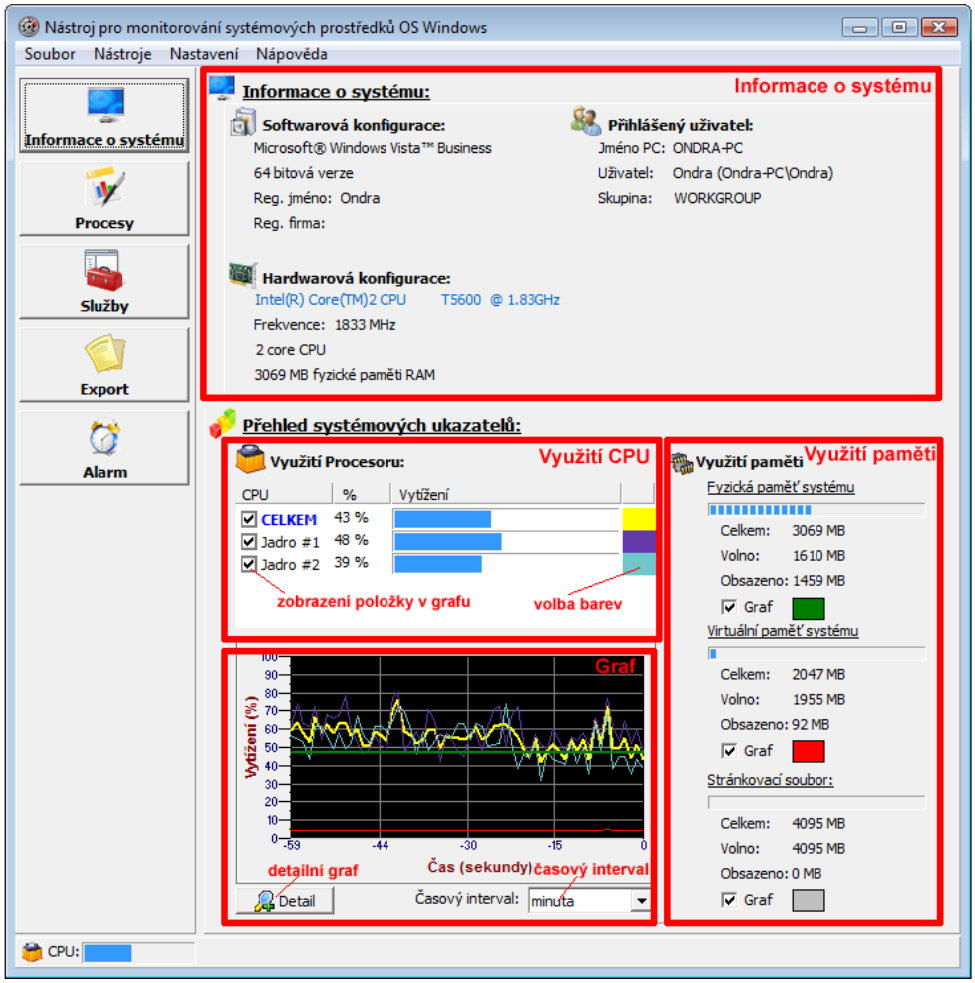
Obrázek 1: Hlavní okno aplikace

Levá část okna je tvořena pěti tlačítky menu, sloužícími k navigaci po aplikaci. Volbou tlačítka je aktivován příslušný modul a uživateli jsou prezentovány požadované informace.

Pravá část okna je určena pro zobrazení panelů jednotlivých modulů

1.2.2 Modul systémových informací

Tento modul je aktivován ihned po spuštění aplikace a uživateli jsou touto cestou prezentovány základní informace o systému.



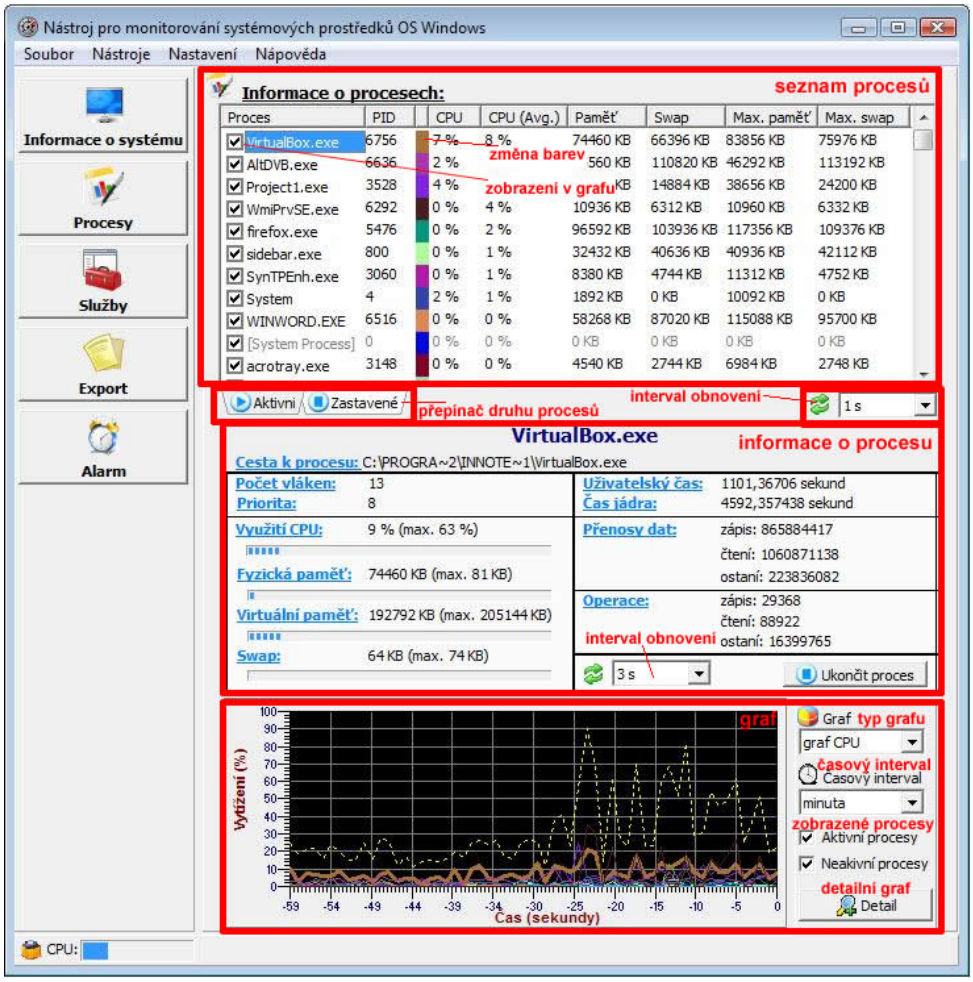
Obrázek 2: Modul systémových informací

Vrchní část panelu modulu zobrazuje informace o softwarové a hardwarové výbavě počítačového systému a o právě přihlášeném uživateli. Přesunutím kurzoru myši na modře zvýrazněné položky je možno získat podrobnější informace. Všechny tyto informace se získávají s využitím služby WMI a tak nemusí být k dispozici okamžitě po spuštění aplikace.

Dolní část panelu modulu je vyhrazena pro základní přehled ukazatelů výkonu systému. Informace o stavu procesoru a jeho jader jsou zobrazeny v tabulce. V pravé části jsou prezentovány informace o fyzické a virtuální paměti a o využití stránkovacího souboru. U všech položek je možno povolit či zakázat jejich zobrazení v grafu a barvu je možno zvolit dvojklikem myši na barevné políčko u požadované položky. Spodní část panelu je vyhrazena pro graf zobrazující stav ukazatelů výkonu. Pomocí seznamu pod grafem je možno vybrat časové období, které bude grafem zobrazeno. Pomocí kurzoru myši je možno zobrazit popisky pro jednotlivé linie grafu. Pro zobrazení detailnějšího pohledu slouží tlačítko detail, jež otevře graf v samostatném okně.

1.2.3 Modul informací o procesech

Modul informací o procesech poskytuje detailní informace o aktuálně běžících procesech i o procesech které, byly již ukončeny.



Obrázek 3: Modul informací o procesech

Vrchní část modulu je tvořena seznamem procesů, který kromě jejich názvu zobrazuje i informace o míře, kterou využívají prostředky systému. Pomocí přepínačů v seznamu lze aktivovat či deaktivovat zobrazení jednotlivých procesů v grafu, kliknutím na barevné pole u zvoleného procesu lze nastavit barvu, pomocí níž bude reprezentován v grafu. Šedou barvou jsou zobrazeny procesy, k nimž není možno s aktuálními přístupovými právy přistupovat. Stiskem pravého tlačítka myší lze vyvolat kontextovou nabídku, s jejíž pomocí lze manipulovat s procesy. Pod seznamem procesů se nachází přepínač mezi aktivními a neaktivními procesy a prvek, s jehož pomocí lze nastavit, jak často mají být informace v seznamu obnovovány.

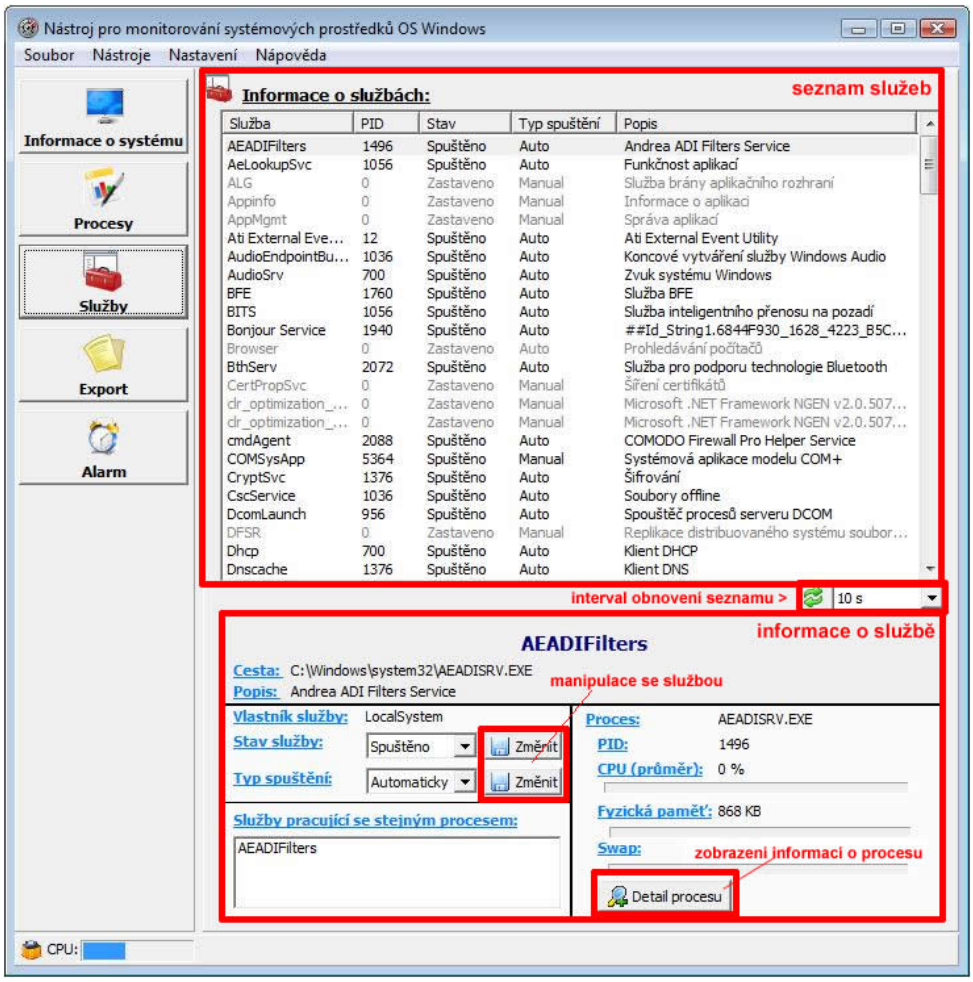
Střední část panelu modulu slouží k zobrazování detailních informací o procesu zvoleném v seznamu. K dispozici jsou ukazatele využití procesoru a pamětí, nechybí ani podrobné informace o počtech provedených operací. Tyto informace jsou získávány pomocí služeb WMI, takže jejich

zobrazení může být při špatné kondici systému časově náročnější, proto je vhodné nastavit dobu obnovování informací na takovou hodnotu, která zbytečně nezatíží systém.

Ve spodní části se nachází graf zobrazující míru využitých systémových prostředků zvolenými procesy. Pomocí ovládacích prvků v pravé části grafu je možno zvolit monitorovaný systémový prostředek a časové období, jež bude zobrazeno. Pomocí přepínačů lze nastavit, zda budou zobrazeny aktivní, neaktivní nebo oba druhy procesů. Popisky grafu lze zobrazit přesunutím kurzoru myši na požadovanou linii. Pro detailnější pohled na graf slouží tlačítko detail, jež zobrazí graf v samostatném okně, s nímž lze libovolně manipulovat.

1.2.4 Modul informací o službách

Tento modul slouží k získání základního přehledu o instalovaných službách v systému.



Obrázek 4: Modul informací o službách

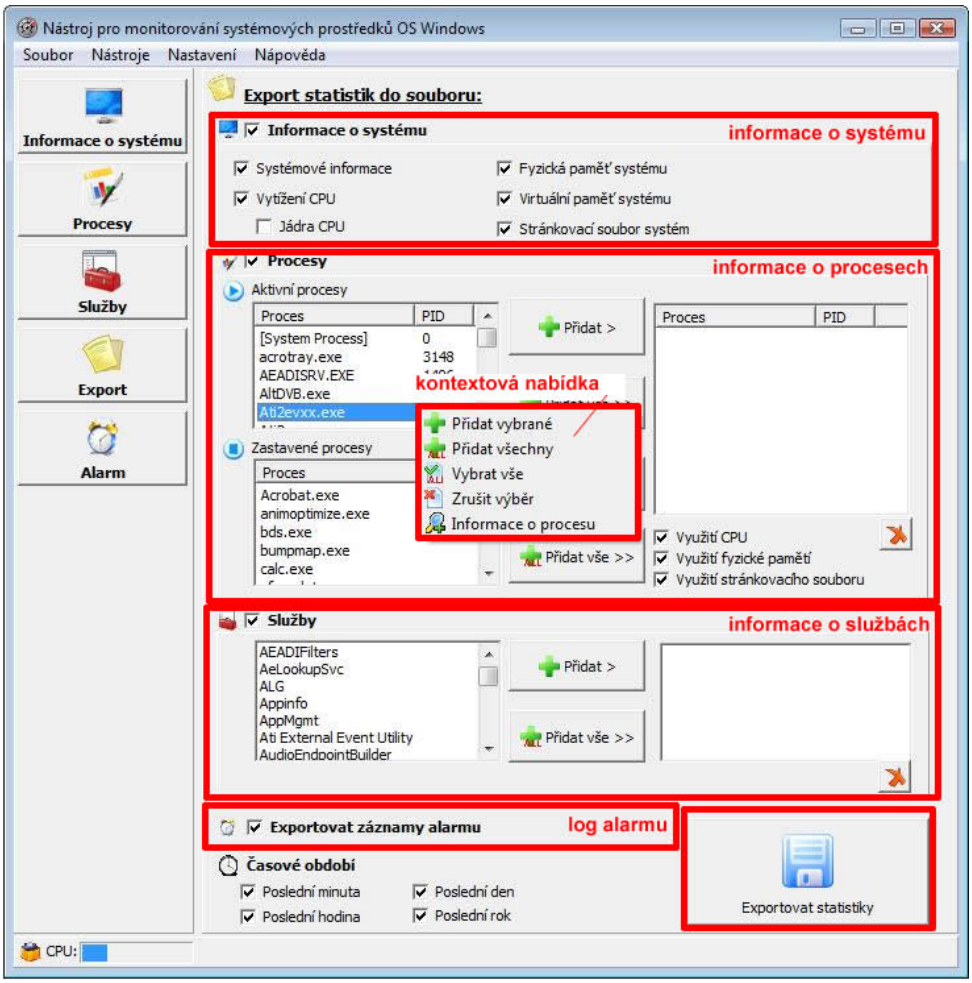
Vrchní část panelu modulu obsahuje seznam všech nainstalovaných služeb v systému a informace o jejich režimu spouštění a o jejich aktuálním stavu. Šedou barvou písma jsou zobrazeny služby, které jsou zastaveny. Pravým tlačítkem myši lze vyvolat kontextovou nabídku, jež zobrazí prostředky pro nastavení režimu nebo změnu stavu zvolené služby. Pod tímto seznamem se nachází prvek, s jehož pomocí lze nastavit časový interval obnovy seznamu služeb. Je vhodné nastavit

časovač na delší časový interval, neboť získávání informací o službách operačního systému do značné míry zatěžuje systémové prostředky.

Spodní část panelu modulu je využita tabulkou, zobrazující podrobnější informace o vybrané službě. Pomocí tlačítka “Detail procesu“ je možno přesunout se do modulu informací o procesech a zobrazit tak detailní informace o procesu, kterým je služba provozována.

1.2.5 Modul exportu do textového souboru

Modul exportu do textového souboru slouží k vytvoření textového výstupu z naměřených statistických dat.



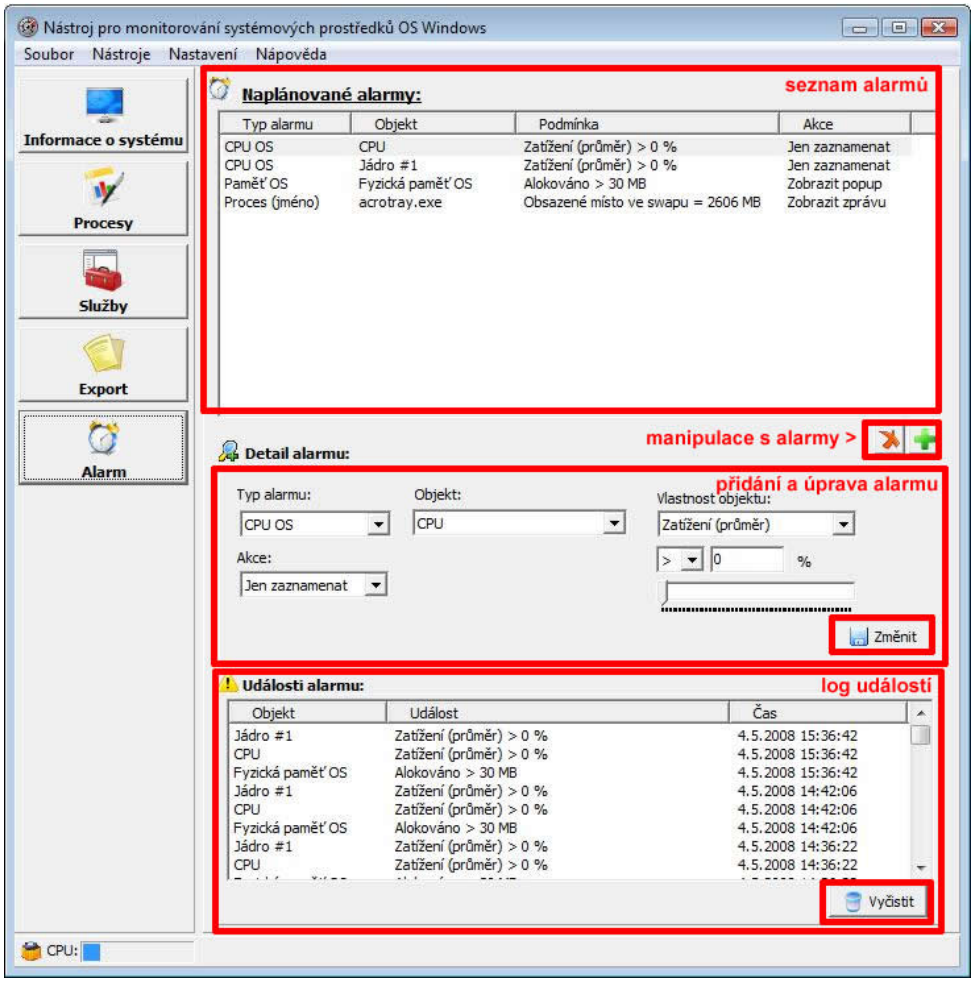
Obrázek 5: Modul exportu do textového souboru

Panel modulu je zaplněn několika skupinami přepínačů, s jejichž pomocí lze vybrat okruhy informací, jež budou uloženy do textového souboru. Je možno zakázat či povolit export informací z celých modulů nebo jejich částí. Pro snadnější manipulaci s položkami modulu procesů a služeb je možno využít kontextových nabídek, které jsou přístupné pomocí stisku pravého tlačítka myši.

Export informací do textového souboru je možno zahájit stisknutím tlačítka “Export statistiky“. Po jeho aktivaci je nutno zadat umístění, kam bude výsledný soubor uložen.

1.2.6 Modul alarmu

Pomocí modulu alarmu je možno stanovit, které systémové prostředky budou aplikací sledovány a jaké akce nastanou při splnění definovaných podmínek.



Obrázek 6: Modul alarmu

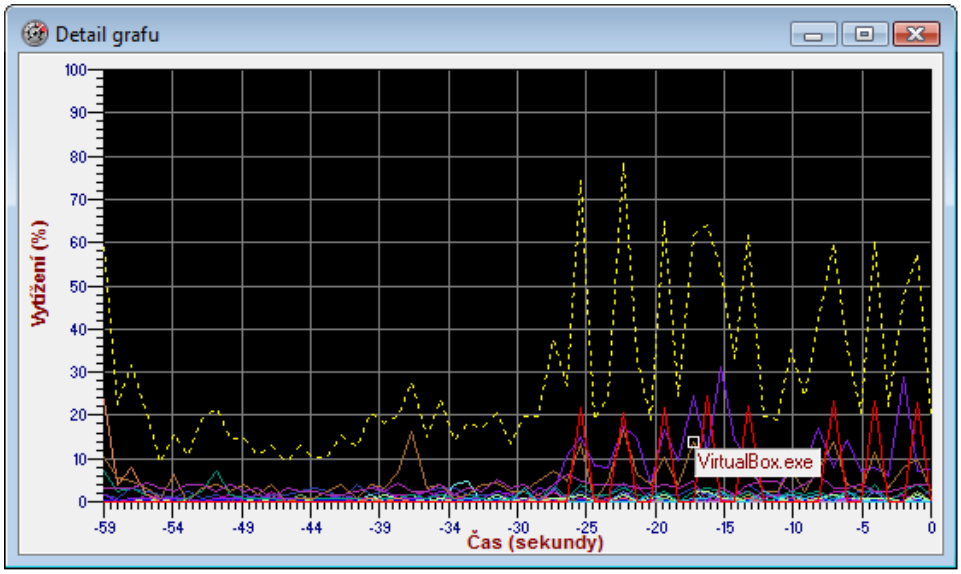
Vrchní část panelu modulu alarmu je tvořena seznamem již nadefinovaných alarmů. Pomocí kontextové nabídky je možno s jednotlivými alarmy manipulovat.

Střední část panelu obsahuje textová pole a seznamy, s jejichž pomocí lze upravovat zvolený alarm nebo vytvořit nový. Na výběr je několik objektů, jež mohou být sledovány. Jedná se o procesor, paměť operačního systému, procesy a služby. Pro každý objekt je možno nastavit další parametry jako je například název procesu pro objekt typu proces nebo typ paměti pro objekt typu paměť. Dalším krokem je nastavení podmínky, na jejíž splnění bude aplikace reagovat zvolenou akcí. Nabízí se hned několik typů akcí. Lze zvolit zaznamenání do logujícího souboru, zobrazení varovného okna, zobrazení informace ve tvaru balonku nebo restartování či vypnutí počítače.

Spodní část panelu obsahuje seznam událostí alarmu, k nimž došlo. Položky tohoto seznamu je možno exportovat v modulu exportu.

1.2.7 Detailní zobrazení grafu

Okno detailního zobrazení je možno otevřít v modulu systémových informací a informací o procesech. Graf zobrazovaný v tomto okně je identický s grafem ve zdrojovém modulu. Okno detailního grafu lze libovolně zvětšovat, pomocí přesunutí kurzoru myši na požadovanou linii lze zobrazit popisky grafu.

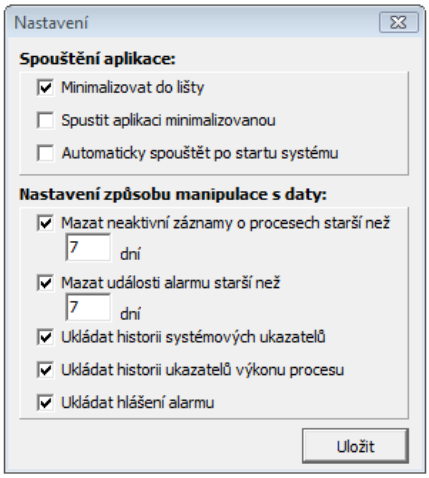


Obrázek 7: Detail zobrazení grafu

1.2.8 Nastavení

Okno nastavení je možno zobrazit pomocí volby položky "nastavení" v hlavní nabídce aplikace. V tomto dialogovém okně lze nastavit chování aplikace při minimalizaci, režim jejího spouštění a jejího chování po spuštění.

Následuje nastavení způsobu manipulace s daty. Lze zvolit, která statistická data budou ukládána a archivována a je možno nastavit dobu, po kterou se uchovávají informace o neaktivních procesech a dobu, po niž jsou archivovány události alarmu.



Obrázek 8: Nastavení

1.3 Ukládání dat

Data shromažďovaná aplikací i její nastavení jsou uchovávány v několika datových souborech v adresáři “data“, který se nachází ve stejném místě, z kterého je aplikace spuštěna. Jednotlivé moduly ukládají data do samostatných souborů. Zálohu dat je možno provést prostým okopírováním souborů nebo za pomocí položky “Zálohovat data“ z nabídky soubor. Data se ukládají periodicky každých 5 minut a též při ukončení aplikace.