Univerzita Pardubice

Fakulta elektrotechniky a informatiky

# Sledování počítačů v síti s využitím HTTP serveru

Bc. Ondřej Petržilka

Diplomová práce

2011

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2010/2011**

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Bc. Ondřej PETRŽILKA** |
| Osobní číslo: | **I09379** |
| Studijní program: | **N2646 Informační technologie** |
| Studijní obor: | **Informační technologie** |
| Název tématu: | **Sledování počítačů v síti s využitím HTTP serveru** |
| Zadávající katedra: | **Katedra softwarových technologií** |

Zásady pro vypracování:

Hlavním cílem této práce je navrhnout a implementovat systém sledující vybrané běžící procesy na libovolném počítači, ať už v lokální síti, či síti internet. Programový klient, napsaný ve vzbraném jayzce. sbírá všechna data, tj. spuštění počítače, nalogování uživatele a vybrané procesy. Přičemž není účelem tajné sledování počítačů, ale závisí pouze na uživateli, která data poskytne. Klient tyto údaje zpracuje a pošle na server, kde se všechna data uloží do databáze, napsané v jazyce SQL, pro pozdější zpracování. K této databázi se přistupuje přes webové rozhraní pomocí aplikace napsané v jazyce PHP. Výsledek této práce bude využit k monitorování provozu na stanicích v síti internet, bude dávat přesný obraz toho, co se na těchto počítačích děje.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

**Žádná doporučená literatura**

Vedoucí diplomové práce: **Mgr. Josef Horálek**
KIT FIM UHK

Datum zadání diplomové práce: **27. října 2010**

Termín odevzdání diplomové práce: **20. května 2011**

prof. Ing. Simeon Karamazov, Dr.
děkan

Univerzita Pardubice
Česká republika

L.S.

doc. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 3. listopadu 2010

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury. Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

v Pardubicích dne 27. 4. 2011

Ondřej Petržilka

Poděkování:

Touto cestou bych chtěl poděkovat Mgr. Josefu Horálkovi za velmi cenné rady a připomínky k mé diplomové práci a za čas věnovaný mým konzultacím.

## ANOTACE

Práce je věnována analýze nástrojů pro sledování počítačů v síti, návrhu vlastního řešení a realizaci tohoto řešení. Teoretická část práce se zabývá historií operačních systémů a toho, jak vznikal koncept procesu běžícího v rámci operačního systému. Praktická část se zabývá návrhem a tvorbou nástrojů pro sledování počítačů v síti s využitím HTTP serveru. V rámci práce byly vytvořeny dvě hlavní aplikace, klient a server. Server poskytuje webovou službu, ke které se připojí klient a pomocí operací této služby posílá na server nasnímaná data. Klient snímá data z rozhraní WMI.



## KLÍČOVÁ SLOVA

sledování, počítač, síť, webová služba, http, web, proces, vlákno, wmi


## TITLE

Monitoring of computers in the network using the HTTP server


## ANNOTATION

The thesis is dedicated to analysis of computer monitoring tools, own design solution and implementation of the solution. The theoretical part deals with the history of operating systems and how the concept of process originated in the the operating system. The practical part deals with the design and creation of tools to monitor network computers using the HTTP server. Two main applications were created, client and server. Server provides a Web service to which the client connects and sends scanned data using the service operations. The client reads data from WMI interface.



## KEYWORDS

monitoring, computer, network, web service, http, web, process, thread, wmi

# Obsah

# Seznam tabulek

# Seznam obrázků

# Seznam použitých zkratek

| | |
|---|---|
| CD | Compact disc |
| DVD | Digital versatile disk |
| DLL | Dynamic-link library |
| CPU | Central processing unit |
| CSS | Cascade style sheets |
| DMA | Direct memory access |
| HTTP | Hypertext transfer protocol |
| I/O | Input-output |
| MIT | Massachusetts Institute of Technology |
| MVP | Model-View-Presenter |
| OS | Operating system, operační systém |
| PC | Personal computer |
| PCB | Process control block |
| SQL | Structured query language |
| UAC | User access control |
| UI | User interface |
| WMI | Windows management interface |
| XML | Extended markup language |

# Úvod

Sledování počítačů v síti lze dělit podle několika kritérií, může se jednat sledování dobrovolné, kdy uživatel dobrovolně poskytuje data nebo nedobrovolné, kdy uživatel nemá možnost ovlivnit, zda bude jeho počítač sledován či nikoli. Počítač může být sledován tajně či s vědomím uživatele.

Nedobrovolné a často i tajné sledování počítačů provádějí nebo si nechají provádět zejména společnosti, které chtějí kontrolovat, co jejich zaměstnanci dělají v pracovní době a zda skutečně pracují. Zákon říká, že zaměstnanec nesmí bez souhlasu zaměstnavatele užívat pro svou osobní potřebu počítač a telefon, případně jiné výrobní a pracovní prostředky jemu svěřené. Dodržování tohoto zákazu je pak zaměstnavatel oprávněn ze zákona přiměřeným způsobem kontrolovat.

Tajné sledování počítačů, které je za hranicí zákona provádí software, který se obecně označuje jako Malware. Malware může infiltrovat počítač pomocí webové stránky využívající bezpečnostní díry ve webovém prohlížeči nebo instalací podvodného softwaru, který si uživatel sám nainstaluje. Podvodný software si může nezkušený uživatel sám dobrovolně stáhnout a nainstalovat, často se tento program tváří jako skutečný, například antivirový, program.

Dobrovolné sledování počítačů s vědomím uživatele se často provádí pro ochranu uživatele i počítače. Tímto způsobem lze odhalit nebezpečné nebo podezřelé programy, Malware, aplikace, které příliš vytěžují procesor nebo zabírají velkou část operační paměti. Tímto způsobem lze také kontrolovat, jaký software uživatelé instalují a spouštějí.

V kapitole 2. je porovnání programů použitelných pro sledování procesů na počítačích v síti. Některé aplikace často umožňují sledovat i další věci, například služby, instalovaný software atp. Aplikace umožňují okamžité sledování stavu počítače, avšak většina aplikací neumožňuje sledovat historická data. Z porovnávaných aplikací bylo pouze velmi malé množství aplikací uzpůsobeno k automatickému sledování počítačů a vyhodnocení dat.

Do porovnávaných aplikací nebyly zahrnuty programy a systémy, které jsou nabízeny jako all-in-one řešení pro sledování počítačů a počítačových sítí. Tyto aplikace nejsou k dispozici zdarma a to většinou ani k vyzkoušení a jejich účelem je často pouze tajné sledování počítačů.

Cílem praktické části práce je vytvoření aplikací, které budou schopny zabezpečit sledování procesů, přihlašování a odhlašování uživatelů na počítači a ukládání těchto dat na HTTP server. Jedna aplikace, klient, bude na sledovaném počítači a bude odesílat požadovaná data na HTTP server. Druhá aplikace, server, bude běžet na HTTP serveru a bude zajišťovat ukládání dat do relační databáze a jejich zobrazování.

# 1 Teorie procesů

Proces je v informatice název pro spuštěný počítačový program. Proces je umístěn v operační paměti v podobě sledu strojových instrukcí vykonávaných procesorem. Obsahuje nejen kód vykonávaného programu, ale i dynamicky měnící se data. Jeden program může běžet jako více procesů s různými daty. V závislosti na operačním systému se může proces skládat z více vláken, která provádějí zpracování instrukcí současně. Správu procesů vykonává operační systém, který zajišťuje jejich oddělený běh a přiděluje jim zdroje. [1]

## 1.1 Systémy se sériovým zpracováním

Historický úvod a přehled přístupů k operačním systémům byl čerpán z [2], [3], [4].

Na nejstarších počítačích na konci čtyřicátých a počátku padesátých letech neexistoval žádný operační systém a uživatel pracoval přímo s hardwarem. Tyto stroje se skládaly z konzole, vstupního zařízení a tiskárny. Programy, které se spouštěly na těchto strojích, se nazývaly „job“. S těmito systémy byly spojeny dva hlavní nedostatky.

První problém spočíval ve způsobu plánování, v systému neexistoval žádný plánovač a uživatelé se u počítače doslova střídali. Často to fungovalo tak, že si uživatelé rezervovali počítač v určitou dobu na určitý čas. Pokud si uživatel zarezervoval hodinu času a práci dokončil rychleji, zbylý čas byl vyplýtván. Na druhou stranu, pokud uživatel práci nestihl dokončit, mohl být nucen odejít a veškerý procesorový čas, který na počítači strávil, byl vyplýtván.

Druhým nedostatkem byla doba nastavení systému před samotným spuštěním programu. Před samotným spuštěním programu mohla být vyžadována kompilace programu, to znamenalo načtení kompilátoru, zdrojových kódů, vygenerování zkompilovaného programu a uložení programu a nakonec slinkování. Každý z těchto kroků mohl vyžadovat výměnu pásky nebo děrných štítků. Pokud nastala chyba, uživatel musel typicky opakovat všechny kroky od začátku. Přípravou spuštění programu bylo tedy často vyplýtváno nezanedbatelné množství procesorového času.

## 1.2 Jednoduché dávkové systémy

První počítače byly velmi drahé, a proto bylo nutné maximalizovat jejich využití a snížit vyplýtvaný čas. Aby se zvýšilo využití těchto počítačů, byl vymyšlen koncept dávkových operačních systémů.

Za první operační systém s dávkovým zpracováním a vůbec za první operační systém je považován systém vyvinutý společností General Motors v polovině padesátých let pro počítač IMB 701. Koncept operačního systému byl zdokonalen a později implementován na počítači IBM 704. Na počátku šedesátých let se objevil operační systém IBSYS, který byl určen pro počítače IBM 7090/7094. Tímto operačním systémem bylo ovlivněno mnoho dalších operačních systémů.

Hlavní myšlenka dávkového operačního systému je použití software známého jako monitor. Monitor načítá jednotlivé úlohy do oblasti zvané “user program area” a předává jim řízení, po dokončení úlohy převezme řízení opět monitor a načítá další úlohu (typicky z děrných štítků nebo magnetické pásky). Část Monitoru starající se o načítání a spouštění úloh se jmenuje „Monitor resident“ a musí být stále načtená v paměti. Zbytek monitoru se skládá z pomocných funkcí, které mohou jednotlivé úlohy využívat.

Dávkové operační systémy mohou obsahovat další mechanismy sloužící k řízení úloh, avšak není to nezbytně nutné. Mezi tyto mechanizmy patří ochrana paměti, časovač, privilegované instrukce a přerušení. Ochrana paměti zaručuje, aby uživatelský program nemohl měnit obsah paměti monitoru, pokud se o to pokusí, je vygenerována chyba a řízení je předáno zpět monitoru. Časovač slouží k tomu, aby se omezila maximální doba spuštění jedné úlohy. Časovač se nastaví a spustí před předáním řízení dané úloze, a pokud vyprší před dokončením úlohy, je úloha zastavena a řízení je předáno zpět monitoru. Privilegované instrukce jsou instrukce, které může spouštět pouze monitor. Mezi privilegované instrukce patří veškeré I/O instrukce, aby bylo zabráněno úloze číst data nebo program jiné úlohy. Přístup k I/O se prování pomocí pomocných funkcí monitoru. Při pokusu o spuštění privilegované instrukce úlohou je vygenerována chyba a řízení je předáno zpět monitoru. Přerušení v prvních operačních systémech neexistovalo. Časem ovšem vznikla potřeba větší

flexibility při přenosu řízení systému z úlohy na monitor a zpět, a proto byl navržen a implementován koncept přerušení.

Dávkové operační systémy nabízely vyšší využití zdrojů než systémy se sériovým zpracováním, avšak stále bylo využití zdrojů velmi nízké. Přístup do paměti trval několikanásobně déle než zpracování stovky instrukcí. Tabulka 1.1 ukazuje příklad, jak dlouho mohly trvat jednotlivé operace. Využití CPU je v tomto příkladu 0,0001 / 0,0031 = 3,2%. Příklad pochází z [5].

Tab. 1.1 – Příklad doby trvání operací v dávkových systémech

| Operace | Doba trvání |
|---|---|
| Načtení jednoho záznamu ze souboru | 0,0015 s |
| Spuštění 100 instrukcí | 0,0001 s |
| Zapsání jednoho záznamu do souboru | 0,0015 s |
| **Celkem** | **0,0031 s** |

## 1.3 Multiprogramové dávkové systémy

Multiprogramové dávkové systémy se snaží řešit hlavní nedostatek jednoduchých dávkových systémů, plýtvání procesorovým časem při čekání na I/O operaci. Multiprogramové dávkové systémy umožňují spouštět více úloh najednou a při čekání na I/O operace jedné úlohy je předáno řízení jiné úloze.

Multiprogramové dávkové systémy vyžadují více operační paměti než jednoduché dávkové systémy, protože je nutné držet všechny aktuálně spuštěné úlohy v paměti. Jelikož jsou všechny úlohy v hlavní paměti programu, je zde zapotřebí určitá správa paměti. Dále je nutné implementovat určitou rozhodovací logiku, která úloha bude spuštěna, pokud je více úloh připravených k běhu.

Multiprogramové dávkové systémy mohou těžit z I/O přerušení a přímého přístupu do paměti. Pokud jsou k dispozici tyto dvě hardwarové vlastnosti systému, operační systém je může velmi efektivně využívat. Při požadavku na I/O operaci může program předat požadavek správci zařízení a pokračovat v běhu jiné úlohy. Jakmile správce zařízení dokončí I/O operaci, vyvolá přerušení procesoru, procesor přerušení obslouží pomocí obslužné rutiny a může pokračovat v běhu, buď úlohy

které čekala na dokončení I/O operace nebo kterékoli jiné, to záleží na implementované rozhodovací logice.

Multiprogramové dávkové systémy efektivně řeší čekání na I/O operace a zvyšují využití procesoru. Na rozdíl od dnešních desktopových operačních systémů jim však chybí jedna velmi důležitá vlastnost, a to schopnost reagovat na vstupy od uživatele v reálném čase.

## 1.4 Time-Sharing systémy

Time-sharing systémy se snaží na rozdíl od multiprogramových dávkových systémů minimalizovat čas odezvy. Základním principem těchto systémů je sdílení procesorového času mezi uživateli. V time-sharing systémech může více uživatelů současně přistupovat k systému skrz terminály. Operační systém přiděluje uživatelským procesům krátká časová kvanta výpočetního času. Pokud nepočítáme režii operačního systému, tak každý z uživatelů má k dispozici pouze 1/n celkového výpočetního času, kde n je počet uživatelů vyžadujících výpočetní čas.

Jeden z prvních time-sharing operačních systémů se jmenoval Compatible Time-Sharing System (CTSS) a byl vyvinut na MIT skupinou Project MAC. Systém byl původně vyvinut pro počítač IBM 709 v roce 1961, později byl přenesen na IBM 7094. CTSS pracoval s časovými kvanty o velikosti přibližně 0,2 sekundy. Přepínání úloh bylo řešeno pomocí přerušení časovače a bylo preemptivní. CTTS podporoval 32 uživatelů a velikost rezidentního monitoru byla 5000 36bitových slov.

Preemptivní přepínání úloh znamená, že za přerušení úlohy a převzetí řízení procesoru byl zodpovědný operační systém. Hlavní výhoda preemptivního plánování úloh spočívá v tom, že jedna úloha nemůže zablokovat celý systém tak, že by odmítala předat řízení operačnímu systému, jak je to v případě nepreemptivního plánování.

Time-sharing systémy musely řešit další problémy vyvstávající ze současného přístupu více uživatelů k systému. Bylo třeba vytvořit řízení přístupu k tiskárně, k velkokapacitním úložným zařízením a k souborům. Přístup k souborům musel zaručit, aby mohla být definována politika dovolující nebo naopak zakazující přístup uživatelů k souborům ostatních uživatelů.

## 1.5 Proces

Koncept procesu má zásadní význam pro strukturu operačního systému. Tento termín byl poprvé použit v šedesátých letech designéry operačního systému Multics. Jako proces je označován běžící program, instance programu běžící na počítači, entita, která může být spuštěna na procesoru a, nebo jednotka aktivity charakterizovaná jedním sekvenčním vláknem, aktuálním stavem a přidruženou množinou systémových zdrojů. Poslední definice už dnes neplatí, protože v dnešní době je běžné, že proces může mít více vláken.

V operačním systému, kde běží více procesů, je nutné procesy oddělit a zajistit, aby se procesy neovlivňovaly. Operační systém musí řešit tyto situace, nebo zajistit, aby nemohly nastat. Mezi tyto situace patří nesprávná synchronizace, selhání vzájemného vylučování, nedeterministické provádění programu a deadlock.

Program, který čeká na I/O operaci čtení, musí čekat, dokud nejsou data načtena do vyrovnávací paměti, než může pokračovat. Ve chvíli, kdy jsou data k dispozici ve vyrovnávací paměti, musí obslužná rutina přerušení poslat signál, že proces může pokračovat. Pokud se díky špatnému designu operačního systému ztratí, je poslán předčasně nebo dvakrát, proces může číst neplatná data. Obdobně může docházet k synchronizačním chybám u jiných operací. Tato chyba se nazývá nesprávná synchronizace.

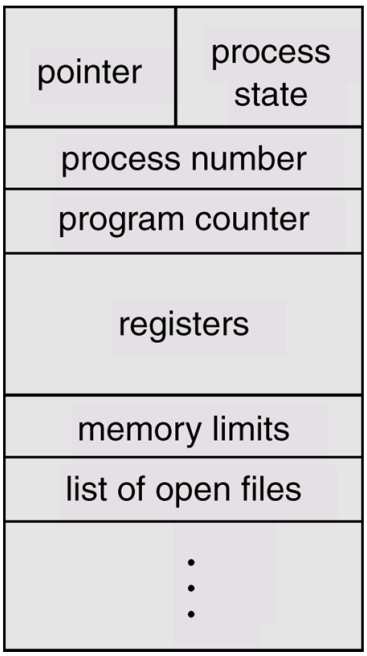
Operační systém musí zaručit, aby více procesů nepřistupovalo zároveň ke sdílenému změnitelnému prostředku. Tato ochrana se nazývá vzájemné vylučování. Vzájemné vylučování zajišťuje atomické provádění operací nad sdíleným prostředkem.

Operační systém musí zaručit deterministické provádění programu, tj. aby běh jednoho procesu neovlivňoval běh jiného procesu, pokud procesy nekomunikují. Ve chvíli, kdy se převádění jednoho procesu přeruší a poté opět obnoví, musí být zaručeno, že při obnovení běhu procesu na procesoru jsou všechny registry, které proces využívá, nastaveny na hodnoty, které byly v registrech při přerušením procesu. Dále je třeba chránit paměť procesu tak, aby nemohl jiný proces přepsat data nebo program druhého procesu.

Operační systém musí řešit problém, kdy každý ze dvou procesů drží zdroj, na který čeká druhý proces. Tato situace se nazývá deadlock. Deadlock také může nastat mezi více než dvěma procesy. Vznik deadlocku je podmíněn následujícími podmínkami: vzájemné vylučování, alokace a čekání, neodnímatelné prostředky, cyklické čekání. Vzájemné vylučování zajišťuje, že ke zdroji má přístup v jednu chvíli pouze jeden proces. Alokace a čekání je stav, kdy proces vlastní zdroj a požaduje další. Neodnímatelné prostředky jsou prostředky, které nemůže operační systém odejmout, musí být explicitně uvolněný procesem. Cyklické čekání je čekání, kdy řetěz vzájemně čekajících procesů uzavírá cyklus.

### 1.5.1 Process Control Block

Process control block je datová struktura, která obsahuje informace potřebné ke správě daného procesu. PCB obsahuje identifikátor procesu, identifikátor vlastníka procesu, registry, informace o adresním prostoru procesu, stav procesu, prioritu procesu, přiřazené zdroje procesu a informace o prostředcích meziprocesové komunikace. PCB také může obsahovat účtovací informace (accounting information) – informace o běhu procesu, jak dlouho strávil proces na procesoru, kdy naposled běžel a podobně.



Obr. 1.1 – Process control block, převzato z [6]

Identifikátor procesu, označovaný jako PID, je často číslo, které jednoznačně identifikuje proces. PCB může kromě PID obsahovat také PID rodičovského procesu.

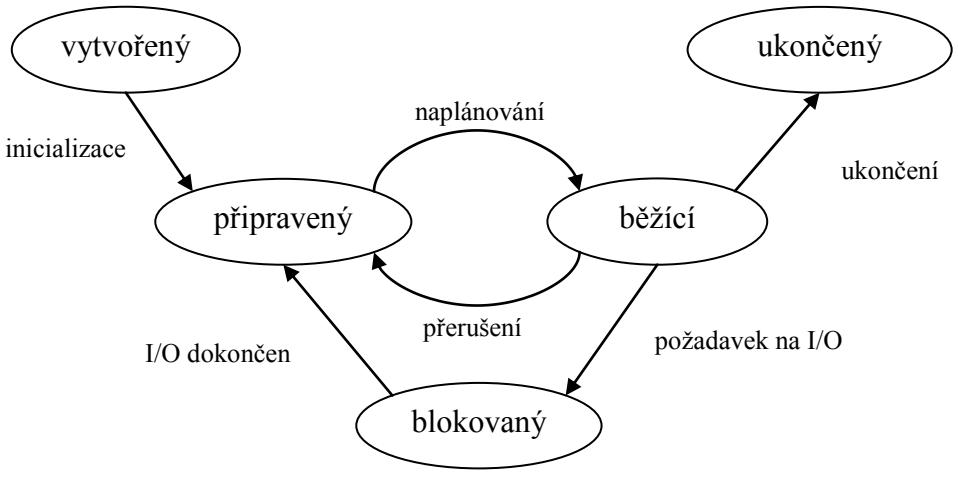
Mezi registry, které jsou uloženy v PCB, patří PC, program counter. Tento registr obsahuje adresu další instrukce, která bude načtena procesorem. Stack pointer, ukazatel zásobníku, ukazuje na vrchol zásobníku, kde jsou uloženy lokální proměnné, argumenty a adresy volaných funkcí. Condition code registry, tyto registry obsahují informace o posledních aritmetických, nebo logických operací. Jsou

zde registry indikující přetečení, znaménko a další. Také jsou zde registry příznaků přerušení, typu spuštěného kódu a další.

Informace o adresním prostoru obsahují dva registry označované jako „base“ a „limit“. Base registr určuje počáteční adresu oblasti paměti, ve které se proces nachází a limit určuje velikost paměti. Program counter a ostatní datové registry odkazující na místo v paměti odkazují relativně k hodnotě base registru a nesmějí překročit hodnotu danou limit registrem.

### 1.5.2 Stav procesu

Každý proces je v určitém stavu. Po vytvoření se proces nachází ve stavu „vytvořený“, po tom, co je vytvořený proces inicializován, dostává se do stavu „připravený“, stav „připravený“ indikuje, že proces je připraven k provádění. Když plánovač procesů podle vnitřní logiky přidělí procesu procesor, proces se nachází ve stavu „běžící“ a běží, tedy aktivně vykonává instrukce. Pokud proces chce provést I/O operaci, je jeho stav změněn na „blokovaný“ (někdy označován také jako „čekající“) a proces čeká na dokončení I/O operace. Mezitím se provádí další proces. Po dokončení I/O je stav procesu změněn na „připravený“ a proces čeká, než na něj opět přijde řada a plánovač mu přidělí procesor. Pokud proces ve stavu „běžící“ vyčerpá přidělené časové kvantum, je přerušen a jeho stav je změněn na „připravený“. Pokud je proces ukončen, je jeho stav změněn na „ukončený“.




Obr. 1.2 – Stav procesu, převzato z [5]

### 1.5.3 Vytvoření procesu

Proces může být vytvořen na žádost operačního systému, nebo na žádost jiného procesu. Vytvoření procesu sestává z několika kroků. Nejdříve je třeba vytvořit unikátní identifikátor procesu, poté je alokováno místo pro proces. Je třeba alokovat místo pro program, pro data, zásobník a také pro PCB. Dále je nutné inicializovat PCB a zařadit proces do seznamu procesů připravených pro běh. Pokud operační systém udržuje statistické údaje o každém procesu, je nutné vytvořit datové struktury pro uchovávání těchto statistik.

### 1.5.4 Ukončení procesu

Proces může být ukončen z několika důvodů. Pokud proces dokončí veškerou svoji práci je korektně ukončen. Dále může být proces ukončen z důvodu chyby.

### 1.5.5 Přepínání procesů

Procesy mohou být přepnuty z několika důvodů, mezi důvody, proč je proces přepnut, patří tyto: přerušení časovače, I/O přerušení, page fault, trap a volání systému.

Přerušení časovače nastává ve chvíli, kdy proces vyčerpá přidělené časové kvantum. Tento čas je stanoven jako kompromis mezi tím, aby systém rychle reagoval na uživatelské vstupy a aby byla režie systému co nejnižší. Při častém přepínání procesů je vyšší režie OS, protože OS musí provádět za jednotku času více úkonů spojených s přepínáním procesů.

I/O přerušení značí dokončení vstupně-výstupní operace. V tomto případě operační systém přesune všechny procesy, jejichž I/O operace se dokončily do stavu „připravený“ a rozhodne se, který proces nyní poběží.

Page fault je stav, kdy proces chce přistupovat do paměti, která není v hlavní paměti operačního systému, ale je v paměti sekundární. Operační systém musí stránku paměti načíst do hlavní paměti, mezitím je stav procesu nastaven na „blokovaný“ a operační systém nyní aktivuje jiný proces.

Trap značí stav, kdy nastala chyba, pokud je chyba fatální, je proces ukončen a operační systém nyní aktivuje jiný proces. Pokud chyba fatální není, závisí reakce operačního systému na typu chyby a nastaveném způsobu obsluhy.

Při volání systému může být proces přepnut, pokud se jedná o přístup k I/O zařízení. Stav procesu bude změněn na „blokovaný“ a aktivován bude jiný připravený proces, protože aktuální proces bude čekat na dokončení I/O operace.

### 1.5.6 Běh procesu

Pro funkčnost operačního systému je důležité, aby nemohly procesy svévolně zasahovat do datových struktur operačního systému a do dat nebo programu jiných procesů. Aby byla splněna tato podmínka, bylo zavedeno více úrovní, ve kterých může procesor operovat. Z hlediska operačního systému a procesů jsou důležité dvě úrovně, kernel mode (mód jádra) a user mode (mód uživatele).

User mode je mód, ve kterém procesor nedovoluje spouštět privilegované instrukce a číst a zapisovat privilegované registry. Také přístup k paměti je omezen. Proces běžící v user mode, může přistupovat pouze ke svým datům, programu a k prostředkům přiděleným operačním systémem. Naopak kernel mode, je mód, ve kterém je dovoleno vše. Důvod proč existují tyto dva módy, je ochrana jádra operačního systému a ochrana procesů mezi sebou. Tímto způsobem je zaručeno, že proces bude dělat pouze to, co mu dovolí operační systém.

Pokud chce proces provést nějakou privilegovanou operaci, musí požádat jádro operačního systému. Volání jádra může být implementováno několika způsoby.

Volání jádra může být implementováno mimo proces, to znamená, že provádění procesu je přerušeno, jeho stav je uložen, procesor se přepne do kernel mode a je na něm provedeno volání obslužné rutiny. Po dokončení rutiny je proces obnoven, procesor přepnut do user mode a původní proces může pokračovat.

Druhý možný způsob je virtuálně spustit volání operačního systému uvnitř uživatelského procesu. Při volání jádra je procesor přepnut do kernel mode, provedena obslužná rutina a po jejím dokončení je proces opět přepnut do user mode. Pro volání jádra je použit druhý zásobník, který je také součástí procesu, ale používá

se pouze pro volání funkcí v kernel mode. Při využití tohoto způsobu není nutné ukládat a obnovovat stav procesu.

Další možnou implementací je mít služby jádra jako separátní procesy běžící v kernel mode. Kromě těchto procesů zde také může být malé množství kódu mimo jakýkoli proces starající se o přepínání procesů. Výhodou tohoto přístupu je, že více služeb jádra může běžet najednou, pokud je k dispozici více fyzických procesorů.

## 1.6 Vlákna

Vlákno neboli thread je odlehčený proces. Každé vlákno musí mít přiřazený proces, který je jeho vlastníkem. Pokud operační systém podporuje multithreading, může proces obsahovat více než jedno vlákno. Pokud operační systém nepodporuje multithreading, uvažujeme, že každý proces má pouze jedno vlákno, i když hranice mezi procesem a vláknem nemusí být zcela zřejmá.

Každé vlákno má vlastní Thread control block a uživatelský zásobník (případně zásobník kernelu záleží, jaký způsob volání jádra systém využívá). Thread control block je podobný Process control blocku, avšak obsahuje mnohem méně informací. Thread control block obsahuje identifikátory vlákna, stav uživatelských, řídících a stavových registrů a informace týkající se plánování vláken (např. priorita). Neobsahuje žádné informace týkající se přístupu do paměti, práv procesů, přiřazených zdrojů a podobně. Tyto informace jsou společné pro všechna vlákna procesu.

Mezi výhody multithreadingu patří rychlejší běh procesu na více-procesorových systémech, tedy za předpokladu, že daný proces obsahuje více vláken běžících současně. Důležitý předpoklad pro více-vláknové zpracování je, že práci programu lze rozdělit na několik nezávislých částí. Příkladem takového programu může být HTTP server, kde je každý požadavek obsloužen ve vlastním vlákně. Další výhodou multithreadingu je škálování. Pokud bude server přetížen, je možné zvýšením počtu procesorů dosáhnout obsloužení vyššího počtu požadavků za stejnou jednotku času bez toho, aby bylo nutné modifikovat program.

Multithreading jednoduše umožňuje, aby program reagoval na vstupy, i když zpracovává úlohu. Stačí obsluhu vstupů řešit v jednom vlákně a zpracovávat úlohu

v druhém vlákně. Bez multithreadingu to bylo také možné, avšak programátor musel sám řešit kdy přerušit úlohu, obsloužit vstupy a opět pokračovat v úloze. Navíc odezva programu na vstupy nemusela být v tomto případě okamžitá a mohla trvat i delší dobu, protože některé úlohy přerušit nelze například I/O operace.

# 2 Porovnání existujících aplikací

Aplikací pro sledování počítačů v síti je velké množství. Existují placené aplikace i aplikace dostupné zdarma. Některé aplikace nabízejí základní funkce zdarma a po zakoupení placené verze lze používat funkce pokročilé. Cena jiných aplikací se odvíjí od počtu sledovaných PC a u některých aplikací je sledování malého počtu PC zdarma.

Existují základní aplikace, které dovolují sledovat pouze procesy na jiných počítačích, některé i bez použití jakéhokoli programu na cílovém počítači (díky službě WMI). Dále jsou k dispozici aplikace, které umožňují sledovat, co se děje na ploše sledovaného počítače, a část z nich dokáže také převzít kontrolu nad počítačem. A nakonec existují i aplikace, které sledují téměř všechno, co uživatel dělá, logují navštívené webové stránky, zprávy odeslané přes instant messanger, které okno bylo kdy aktivní, jaké klávesy byly stisknuty ve kterém okně a podobně. Tyto aplikace většinou poskytují přehledné grafické statistiky, případně upozorňují obsluhu, že daný uživatel provádí nedovolené nebo podezřelé akce. Ceny těchto aplikací jsou často v řádech tisíců či desetitisíců korun.

Při používání těchto aplikací je třeba zamyslet se, zda nebude tato aplikace ohrožovat soukromí uživatele a zda není její používání v rozporu se zákonem. Záleží samozřejmě, jakou z aplikací se rozhodnete používat a jak bude nastavena. Pokud jsou pouze sledovány názvy spouštěných procesů, lze předpokládat, že soukromí uživatele není narušeno a není důvod informovat uživatele o používání této aplikace. Naopak pokud jsou sledovány zprávy zaslané instant Messengerem, jedná se o narušení soukromí a uživatel by měl být informován o tom, že na počítači, na kterém pracuje, je nasazen systém pro sledování a že může zaznamenávat různé věci.

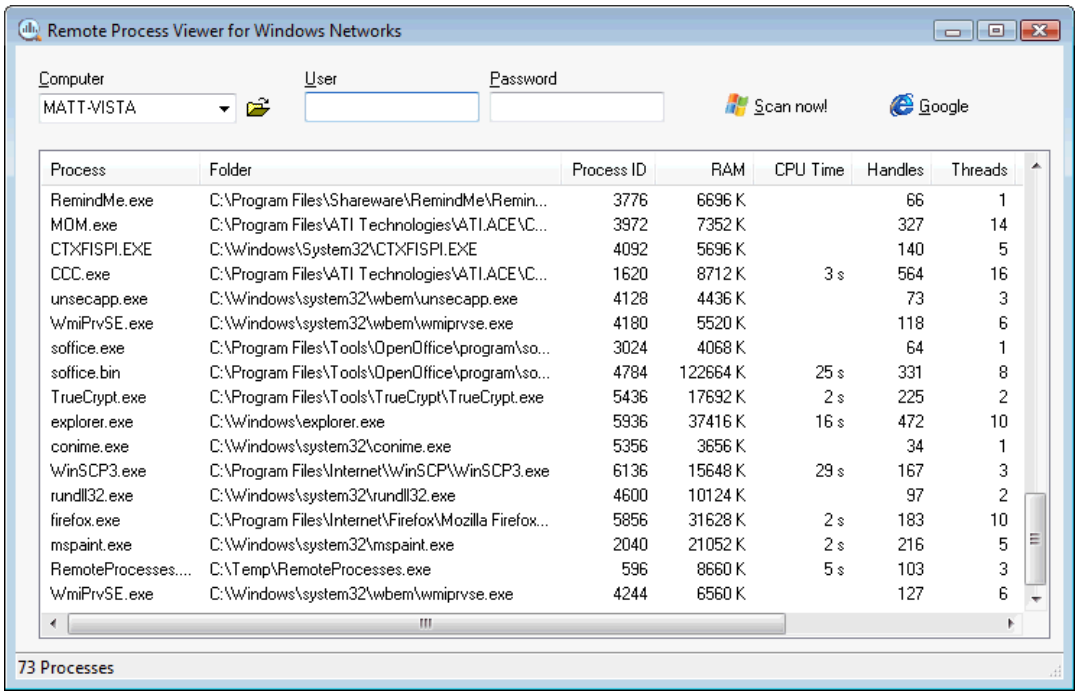
## 2.1 Remote process viewer

Remote process viewer je program, který nabízí společnost Neuber GmbH[7]. Aktuální verze je 1.2.0.39, velikost programu je přibližně 560 KB. Program je distribuován jako freeware a je určen pro operační systémy Windows NT, Windows 2000, Windows XP, Windows 2003, Windows Vista a Windows 7.

Jedná se jednoduchý program, který umožňuje sledovat procesy na libovolném počítači v síti. Na cílových počítačích není třeba instalovat žádného agenta. Systém získává informace o procesech na počítači v síti pomocí rozhraní WMI. Je třeba znát název počítače nebo IP adresu a dále přístupové údaje na cílovém počítači musí být také aktivní služba WMI a musí být dostupná po síti (povolený port ve firewallu apod.)

Program sleduje běžící procesy, ke každému procesu zobrazuje následující informace: jméno procesu, název spustitelného souboru, id procesu, velikost procesu v paměti, čas strávený na procesoru, počet vláken, prioritu procesu, a další. S procesy není možné manipulovat, nelze měnit prioritu, pozastavovat procesy ani je ukončovat, pouze je možné setřídit výpis. Program neumožňuje generovat žádné statistiky ani ukládat či zobrazovat historická data.

Výhodou programu je, že ho není třeba instalovat, je zdarma a jeho úroveň zabezpečení daná službou WMI je vysoká. Z výše uvedených informací vyplývá, že program je vhodný pro každého, kdo chce jednoduše sledovat běžící procesy na počítačích v síti.




Obr. 2.1 – Remote process viewer, převzato z [7]

## 2.2 Remote task manager

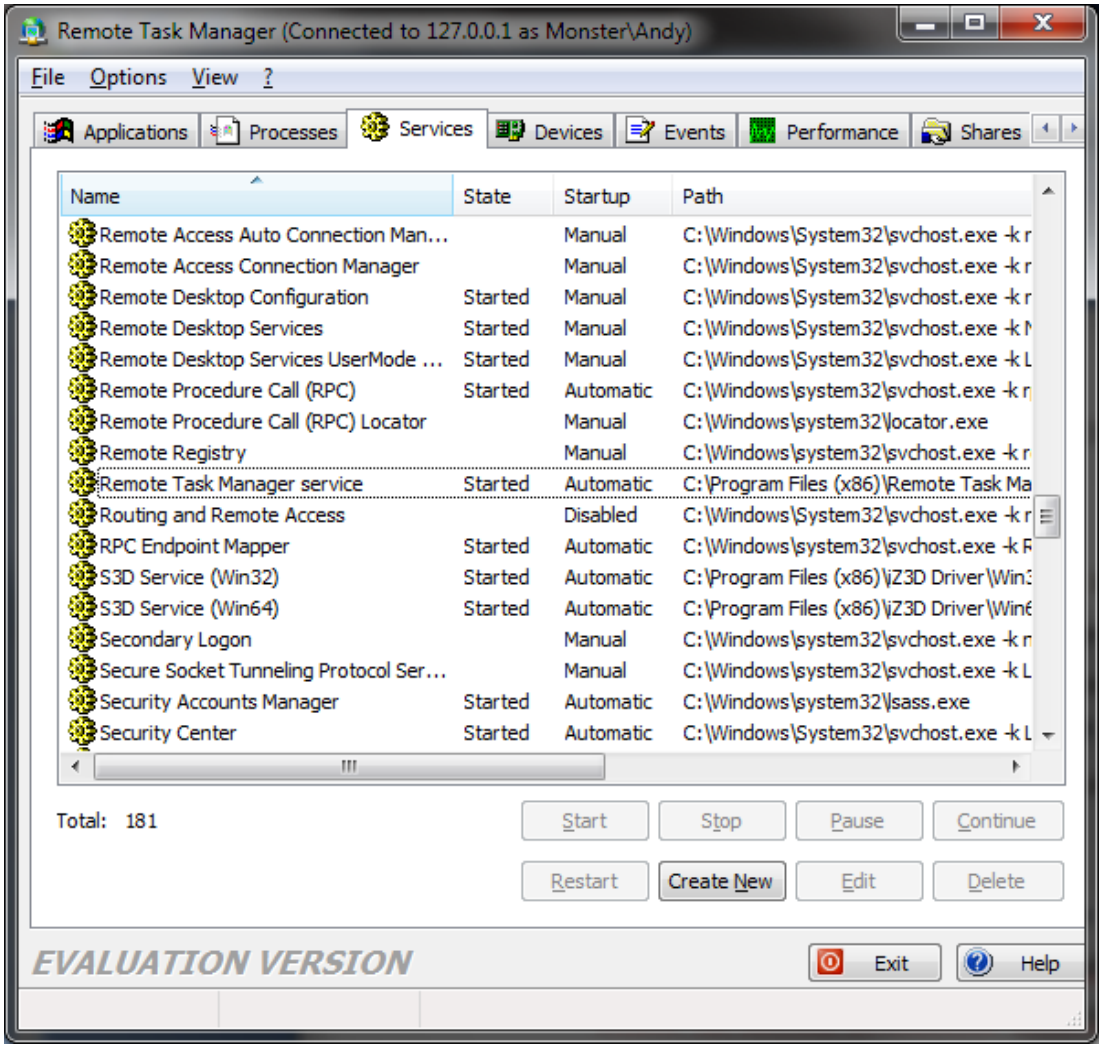
Remote task manager je program nabízený společností DeviceLock, Inc.[8]. Aktuální verze je 3.8.2, velikost programu je přibližně 2 MB. Program je nabízen pod komerční licencí. Cena jedné licence je 35,60 euro. Program lze zdarma vyzkoušet. Zkušební doba je 30 dní. Program je určen pro operační systémy Windows NT 4.0. Windows 2000, Windows XP a Windows Server 2003.

Program umožňuje sledovat běžící aplikace, procesy, služby, zařízení (device manager), event log, sdílené složky, využití CPU a paměti. Většina těchto modulů není omezena pouze na sledování. Procesy lze pozastavovat, ukončovat, měnit jejich prioritu a spřažení, služby lze zakazovat, pozastavovat, ukončovat. Lze měnit sdílené složky. Program také umožňuje vzdáleně vypínat PC nebo spouštět programy. Aktuální stav vzdáleného počítače lze exportovat do souboru, jiná možnost sledování historických dat není k dispozici.

Ke sledování počítače je nutné na cílový počítač nainstalovat službu Remote task manager. Úroveň bezpečnosti je velmi nízká, služba není po instalaci nijak chráněna, a pokud se kdokoli jiný připojí do sítě a nainstaluje si tento program, může se připojit na počítače, kde tato služba běží. Vzhledem k možnostem tohoto programu je převzetí kontroly nad počítačem nebo tajné sledování uživatelů velmi jednoduché.

Tento program je určen pro starší operační systémy a Windows Vista ani Windows 7 nejsou podporovány. Program na nich sice běží, ale vyskytují se problémy, nelze například sledovat aplikace, ani nelze zobrazit graf využití systémových prostředků a program při pokusu spustit novou aplikaci na cílovém vyhodí chybu a spadne.

Podpora pouze starších operačních systémů a nízká bezpečnost řadí tento program mezi programy nevhodné.




Obr. 2.2 – Remote task manager

## 2.3 YAPM – Yet another process monitor

YAPM je program, který vyvíjí nezávislý vývojář s přezdívkou violent_ken. Projekt je hostován na serveru SourceForge.net. [9] Poslední verze programu je 2.4.1, velikost programu je zhruba 4 MB, program je šířen pod licencí GNU GPL 3.0. Program je určen pro operační systémy Windows XP, Windows Vista a Windows 7.

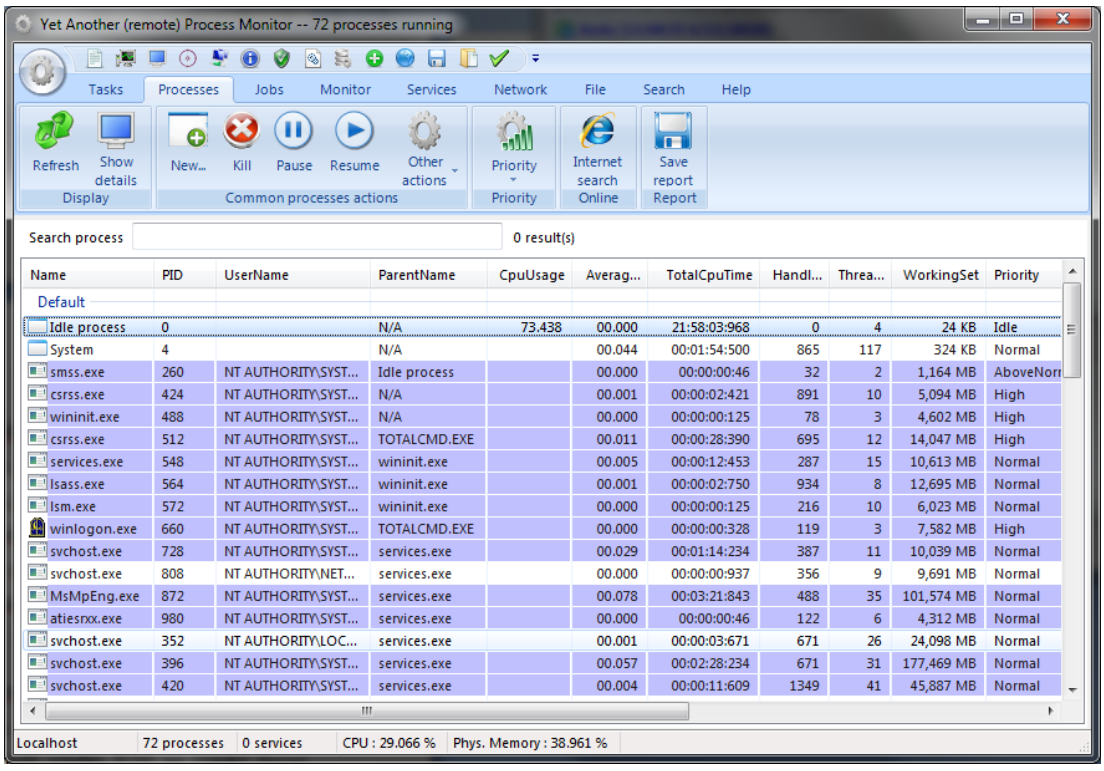
Program umožňuje sledovat běžící procesy, služby, úlohy, otevřené soubory, monitorovat síťová připojení, zobrazovat graf využití systémových prostředků, zobrazovat skryté procesy a množství dalších věcí. Informace o procesech jsou opravdu rozsáhlé. K dispozici jsou informace o vláknech, modulech, velmi podrobné využití paměti. Dokonce lze zobrazovat obsah jednotlivých bloků, dále informace o otevřených síťových spojení, informace o oknech a velmi podrobné statistiky.

Program není omezen pouze na sledování, ale spoustu věcí lze měnit, zejména co se týká správy procesů, služeb a úloh. Program má přehledné a dobře vypadající UI, které je založeno na ribbon záložkách. Lze zobrazovat informace týkající se, jak lokálního počítače, tak počítače v síti.

Program umožňuje vytvořit tzv. system snapshot file, soubor, který uchovává informace o stavu sledovaného počítače ve chvíli, kdy byl snapshot vytvořen. Tento soubor lze kdykoli později v programu otevřít a projít si všechny informace o procesech, které v daný okamžik byly spuštěny. Jiný způsob zobrazování historických dat program nenabízí.

Pro sledování počítačů v síti je možné využít buď rozhraní WMI, potom však nejsou dostupné všechny možnosti programu, nebo na cílový počítač nainstalovat YAMP server. Úroveň bezpečnosti je v případě využití WMI na vysoké úrovni, avšak pokud na cílovém počítači spustíme YAMP server, může se k němu připojit kdokoli a bezpečnost je nulová.

Na základě výše uvedených informací je vhodné program používat pouze pro sledování lokálního počítače, nebo počítačů v síti přes rozhraní WMI.




Obr. 2.3 – Yet another process monitor

## 2.4 Nuclear remote control

Nuclear remote control je program od neznámého vývojáře s přezdívkou VR5. Program měl domovskou stránku hostovanou na serveru Webzdarma[10], tato stránka již není dostupná. Poslední dostupná verze programu je 1.6, velikost programu je zhruba 1 MB. Program by šířen zdarma a byl určen pro operační systémy Windows. Program lze nyní stáhnout ze serveru Slunečnice.cz. [11]

Jedná se o český program, spíše určený k ovládnutí počítače než vzdálené sledování procesů. Některé antiviry ho detekují jako trojského koně. Jedná se o starší program. Poslední verze je z roku 2002. Nyní již nefunguje ani domovská stránka, program lze stáhnout z webových stránek slunecnice.cz nebo stahuj.cz.

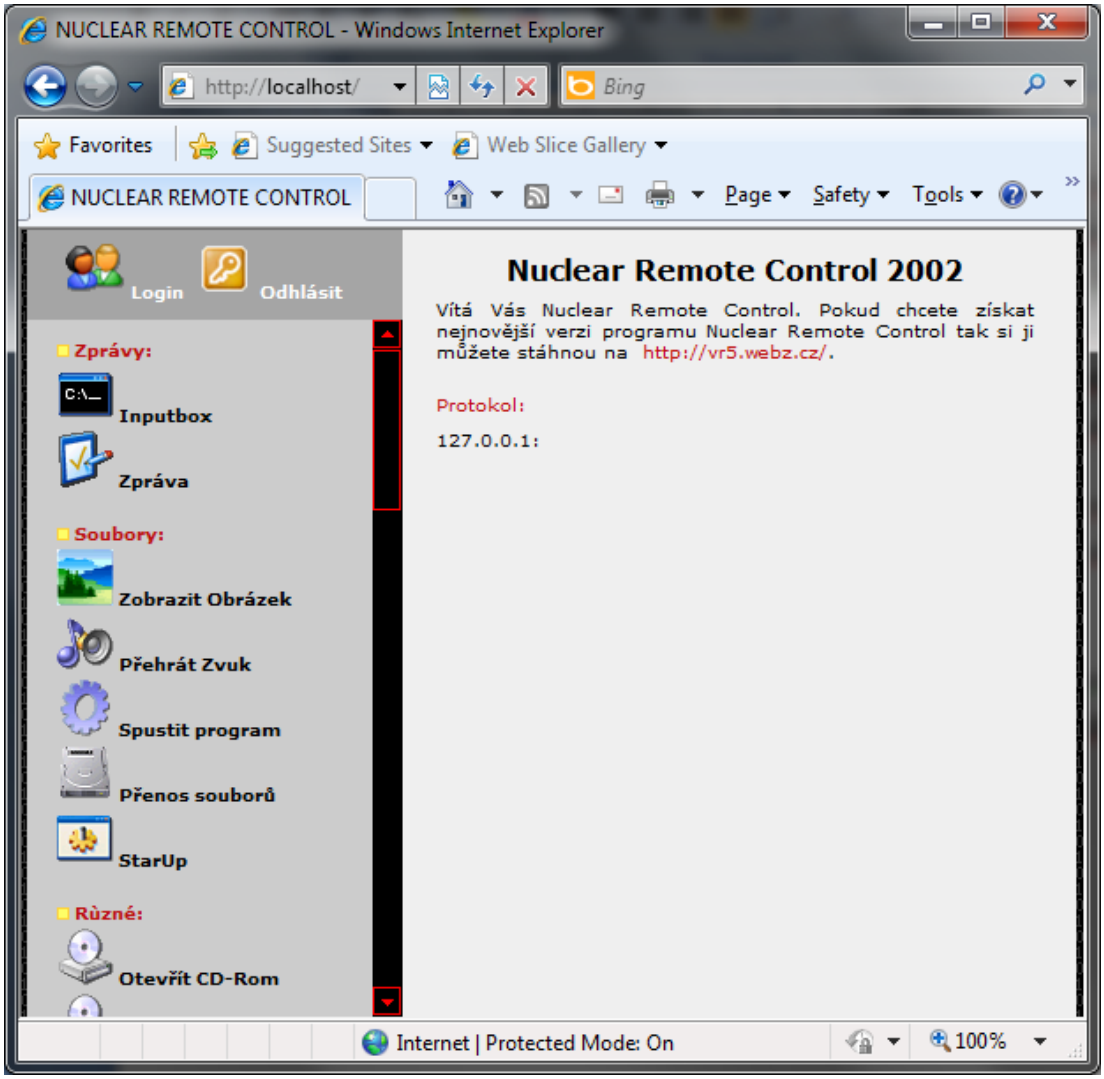
Program umožňuje sledovat běžící procesy a ukončovat je, zobrazovat využití systémových prostředků, restartovat a vypínat počítač a sledovat stisknuté klávesy. U běžících procesů jsou zobrazeny pouze základní informace. Dále program disponuje širokými možnostmi, které mají spíše zábavný či zlomyslný charakter než praktické využití. Jedná se například o prohození tlačítek myši, vysunutí CD nebo DVD mechaniky, schování taskbaru, změnu rozlišení, změnu data a času, přehrání zvuku, zobrazení obrázku, nebo zobrazení zprávy. Program dokáže běžet skrytě, zobrazuje se pouze ve výpisu procesů. Dalšího zamaskování lze docílit přejmenováním spustitelného souboru na název podobný nějakému systémovému procesu, kdy ani znalý uživatel na první pohled nezjistí, že se jedná o tento program.

Program se ovládá přes webový prohlížeč a na rozdíl od většiny předchozích programů disponuje zabezpečením. Pro připojení k počítači v síti je třeba zadat heslo, které lze zvolit pro každý počítač jiné.

Na operačních systémech Windows Vista a novějších, nějaké funkce nefungují, případně program při jejich použití spadne, to se týká zejména pokusu o ukončení nějakého běžícího procesu. Vzhledem ke stáři programu je to pochopitelné.

Jak vyplývá z výše uvedených informací, program není vhodný ke sledování procesů na počítačích v síti, protože neposkytuje o procesech skoro žádné informace.

Ostatní funkce mají spíše zlomyslný charakter, než praktické využití. Program je také zastaralý a nefunguje správně na současných operačních systémech.




Obr. 2.4 – Nuclear remote control

## 2.5 Network security taskmanager

Network security taskmanager je program, který nabízí společnostNeuber software [12]. Poslední verze programu je 1.0h, velikost programu je přibližně 3 MB. Program je nabízen pod komerční licencí, cena licence se odvíjí od toho, kolik počítačů chceme sledovat, pro sledování 5 až 19 počítačů je cena 20 amerických dolarů za jeden sledovaný počítač. S vyšším množstvím počítačů cena za jeden počítač klesá. Pro sledování jednoho až čtyř počítačů je program zdarma. Program je

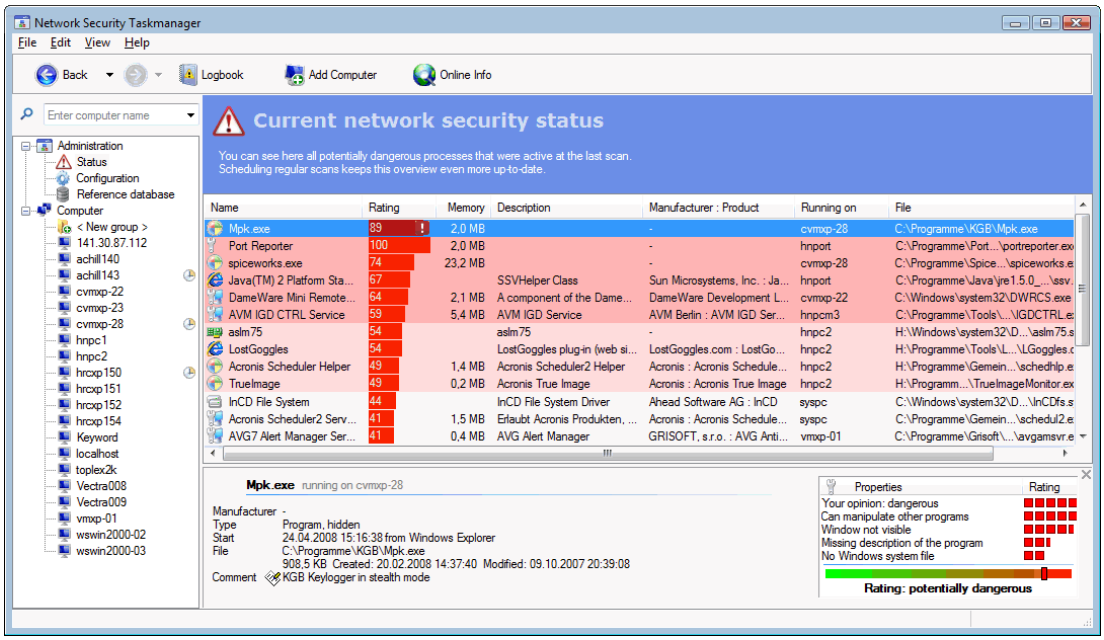
určen pro operační systémy Windows 2000, Windows XP Professional, Windows Server 2003 a Windows Vista.

Program umožňuje sledovat procesy na počítačích v lokální síti a automaticky vyhodnocovat jejich nebezpečnost. Nebezpečné procesy lze okamžitě umístit do karantény. Program umožňuje logovat, kdy vznikly procesy, jak jsou nebezpečné a upozorňovat administrátora.

Výhodou je, že program nepotřebuje instalovat žádného agenta na cílové počítače. Program se připojí k jinému počítači v síti pomocí sdílení souborů MS Windows a nahraje do sdílené složky $Admin agenta, který je posléze spuštěn. K přístupu ke sdílené složce $Admin je třeba mít administrátorské přístupové údaje k cílovému počítači, problém nastává, když je na cílovém počítači zapnuto UAC, což je defaultní nastavení Windows 7 a Windows Vista. V tomto případě nelze přistoupit ke sdílené složce $Admin ani s přístupovými údaji uživatele který má administrátorská práva. Problém by šlo pravděpodobně obejít úpravou nastavení práv na dané sdílené složce.

Program neumožňuje sledovat historická data, co se týče běhu procesů na cílových počítačích, ale umožňuje zjišťovat informace o procesech, které překročily určitou hranici nebezpečnosti, protože údaje o těchto procesech jsou logovány a lze si je kdykoli projít. Bezpečnost programu je na vysoké úrovni, protože jsou požadovány administrátorské přístupy k nahrání agenta a komunikaci s ním.

Na základě informací zmíněných výše, je program vhodný pro každého, kdo potřebuje sledovat potenciálně nebezpečné procesy v síti, avšak je třeba zvážit rizika spojená s nutností mít vypnuté UAC, případně upravit nastavení práv. Tento nástroj může být velmi užitečný správcům sítě, protože mohou jednoduše monitorovat nebezpečné procesy.




Obr. 2.5 – Network security taskmanager

## 2.6 Windows Task Manager

Windows task manager je program, který je součástí OS Windows. Jeho velikost je přibližně 220 KB a poslední verze 6.1.7600 je součástí Windows 7.

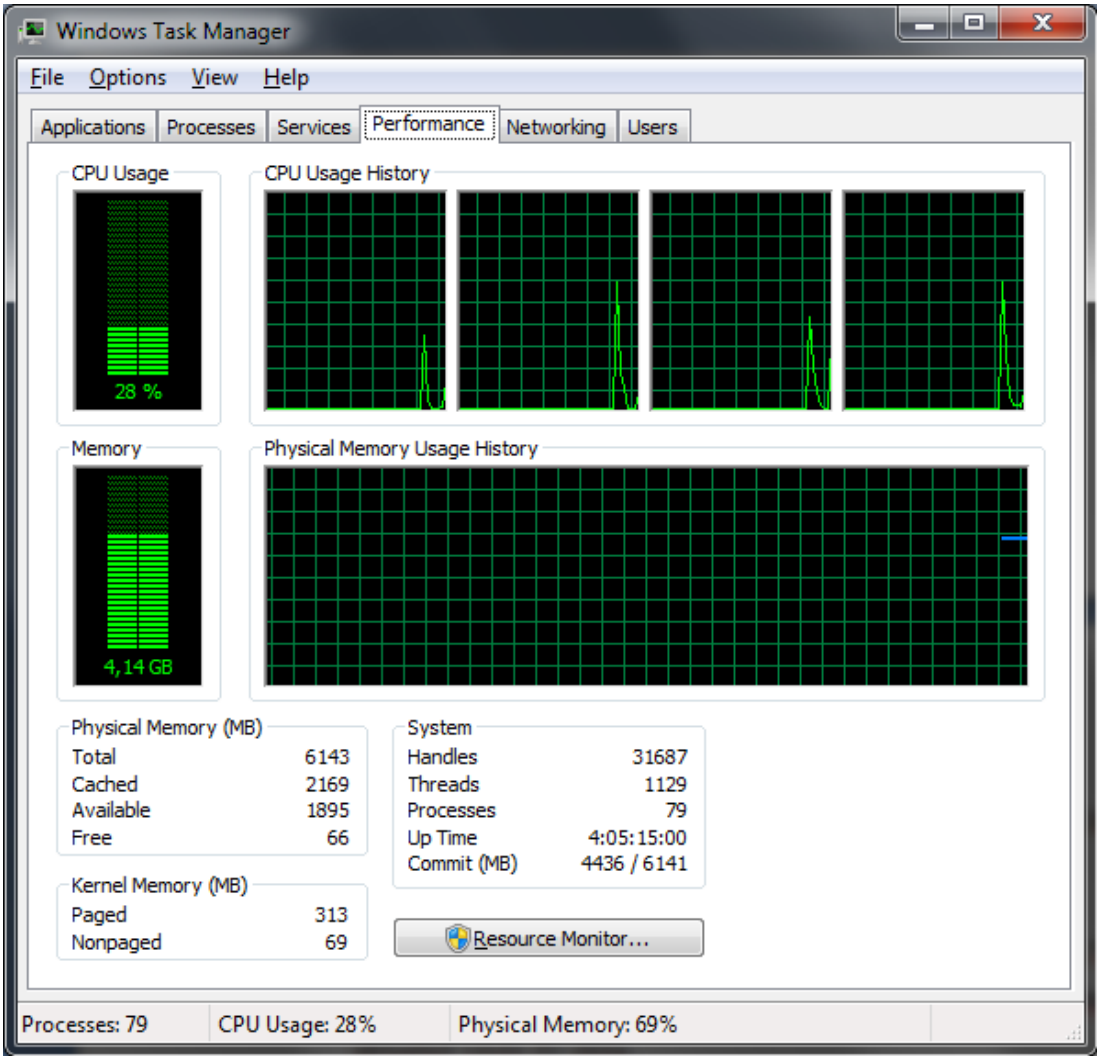
Program umožňuje sledovat aplikace, procesy, služby a využití systémových zdrojů na počítači, kde je spuštěn. Aplikace umožňuje přepínat, ukončovat, minimalizovat a maximalizovat, také dokáže najít proces dané aplikace, což se může hodit, například pokud chceme aplikaci násilně ukončit. V programu si lze zvolit, jaké informace se mají zobrazovat u každého procesu. Možností je mnoho, například jméno procesu, číslo procesu, popis procesu, využití CPU, využití paměti, počet vstupně-výstupních operací, množství přenesených dat v těchto operacích a další. U procesů lze měnit prioritu a spřažení, lze je ukončovat a lze najít spustitelný soubor odpovídající danému procesu. Služby lze spouštět a ukončovat.

Dále program nabízí přehled využití systémových prostředků, lze zjistit využití CPU jako celku a také jednotlivých jader CPU, dále lze najít v přehledu využití paměti, uptime, celkový počet procesů a vláken a využití jednotlivých síťových připojení. Program také zobrazuje aktuální přihlášené uživatele. Pokud je program spuštěn s právy správce systému, lze také přihlášené uživatele odhlásit nebo jejich sezení násilně ukončit.

Program neposkytuje historická ani statistická data s výjimkou grafů využití procesoru, paměti a sítě. Historii nelze procházet a je zobrazeno pouze to, co se aktuálně vejde do okna.

Bezpečnost programu je na velmi vysoké úrovni. Program zobrazuje pouze procesy aktuálního uživatele, procesy ostatních uživatelů lze zobrazovat pouze, pokud je spuštěn s právy správce systému.

Z výše uvedených informací vyplývá, že program je vhodný pro každého, kdo pracuje s počítačem pod operačním systémem Windows a potřebuje jednoduchý program, kterým by mohl spravovat procesy a běžící aplikace. Výhoda tohoto programu spočívá v tom, že ho není třeba instalovat, protože je součástí OS.




Obr. 2.6 – Windows Task Manager

## 2.7 Process Monitor

Process Monitor[13] je program od společnosti Microsoft, původně vyvinutý společností SysInternals, kterou Microsoft koupil. Poslední verze programu je 2.93, byla vydána v roce 2010 a velikost programu je přibližně 3 MB. Program je nabízen zdarma. Je určen pro operační systémy Windows XP a vyšší nebo Windows Server 2003 a vyšší.

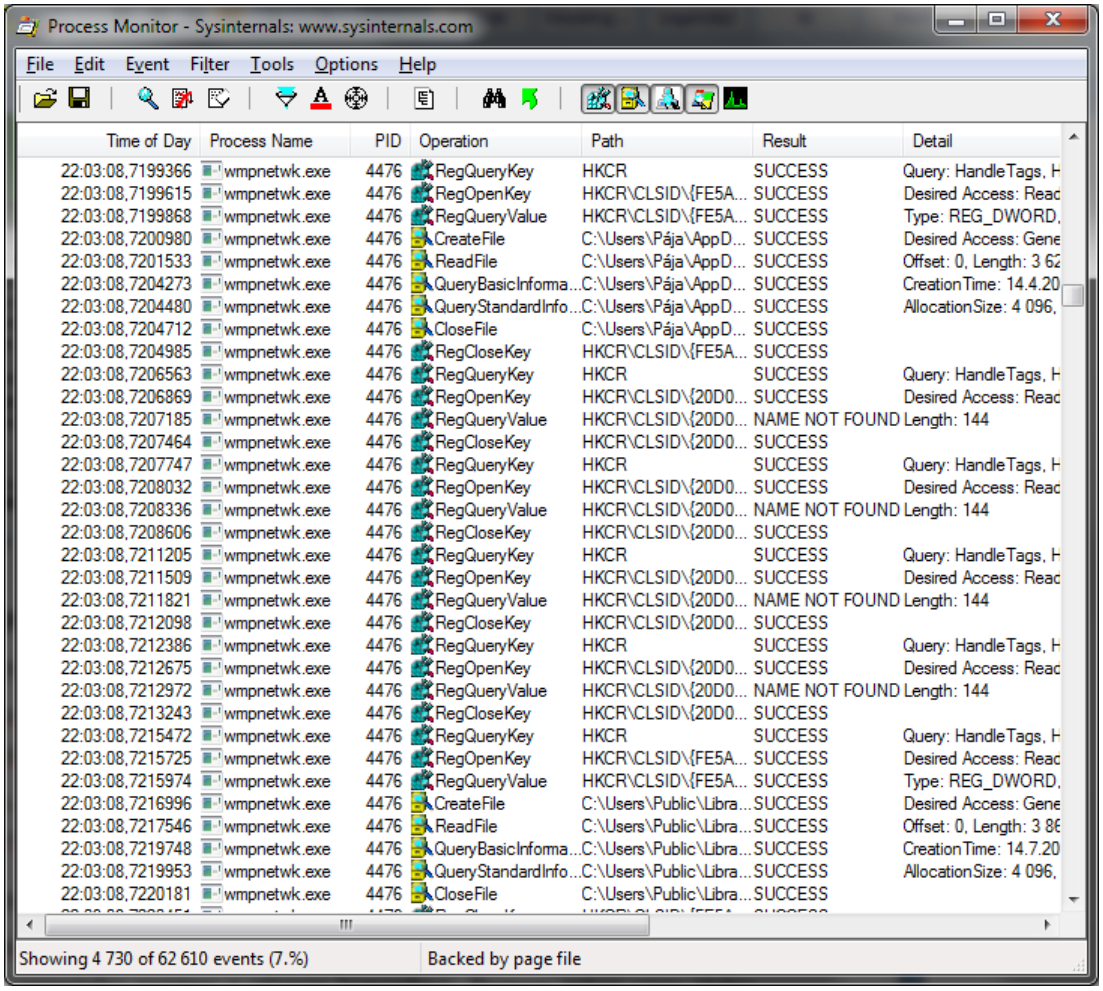
Program slouží k monitorování aktivity procesů, sleduje přístup k souborům (otevření, zavření, čtení a zápis), vytvořená a ukončená vlákna, přístup procesu do registrů operačního systému a síťovou komunikaci. Program nabízí výpis těchto událostí pro všechny aktuálně běžící procesy, avšak tento výpis se lze filtrovat podle čehokoli, co je o dané události zjištěno, například podle jména procesu, podle data, podle typu operace, podle cesty (u událostí týkajících se souborů, registrů nebo síťové komunikace) nebo podle výsledku operace (například neúspěch při zapsání do souboru a podobně). Pokud nějaká aplikace padá při určité operaci, například při uložení souboru, není nic jednoduššího než zjistit, co se pokoušela daná aplikace udělat a zda se to povedlo nebo ne. Pokud se aplikace pokouší zapsat někam, kam nemá uživatel povolen zápis a vývojář aplikace tuto chybu neošetřil, pomocí process monitoru dokážeme odhalit, co je příčinou této chyby. Program neumožňuje procesy spravovat, dokonce ani ukončovat a pro svůj běh vyžaduje oprávnění správce.

Mezi další funkce programu patří zobrazování detailu zachycených událostí. Kromě toho, že program zobrazuje popis dané události a informace o procesu, zobrazuje také její parametry (u čtení ze souboru například pozici a množství čtených dat), aktuálně připojené DLL knihovny a aktuální stav zásobníku (název volané funkce, adresu a o který modul se jedná). Díky výpisu zásobníku lze jednoduše zjistit, jaké funkce z jakých knihoven byli volány.

Program umožňuje sledovat historická data, dokonce umožňuje zachycené události ukládat do souboru a tyto soubory později otevírat a zobrazovat události obsažené v těchto souborech.

Z výše zmíněných informací vyplývá, že program lze doporučil komukoliv, kdo potřebuje zjišťovat, co přesně dané procesy v systému provádějí, s jakými soubory nebo klíči v registrech pracují, nebo jaké knihovny využívají. Program není

určen ke správě procesů. Výhodou tohoto programu je, že ho není třeba instalovat, ale lze ho přímo spustit.




Obr. 2.7 – Process Monitor

## 2.8 Process Explorer

Process Explorer[14] je program od společnosti Microsoft, původně vyvinutý firmou SysInternals, kterou Microsoft koupil. Poslední verze programu je 14.0, byla vydána v roce 2010. Velikost programu je přibližně 4 MB. Program je nabízen zdarma. Je určen pro operační systémy Window XP a vyšší nebo Windows Server 2003 a vyšší.

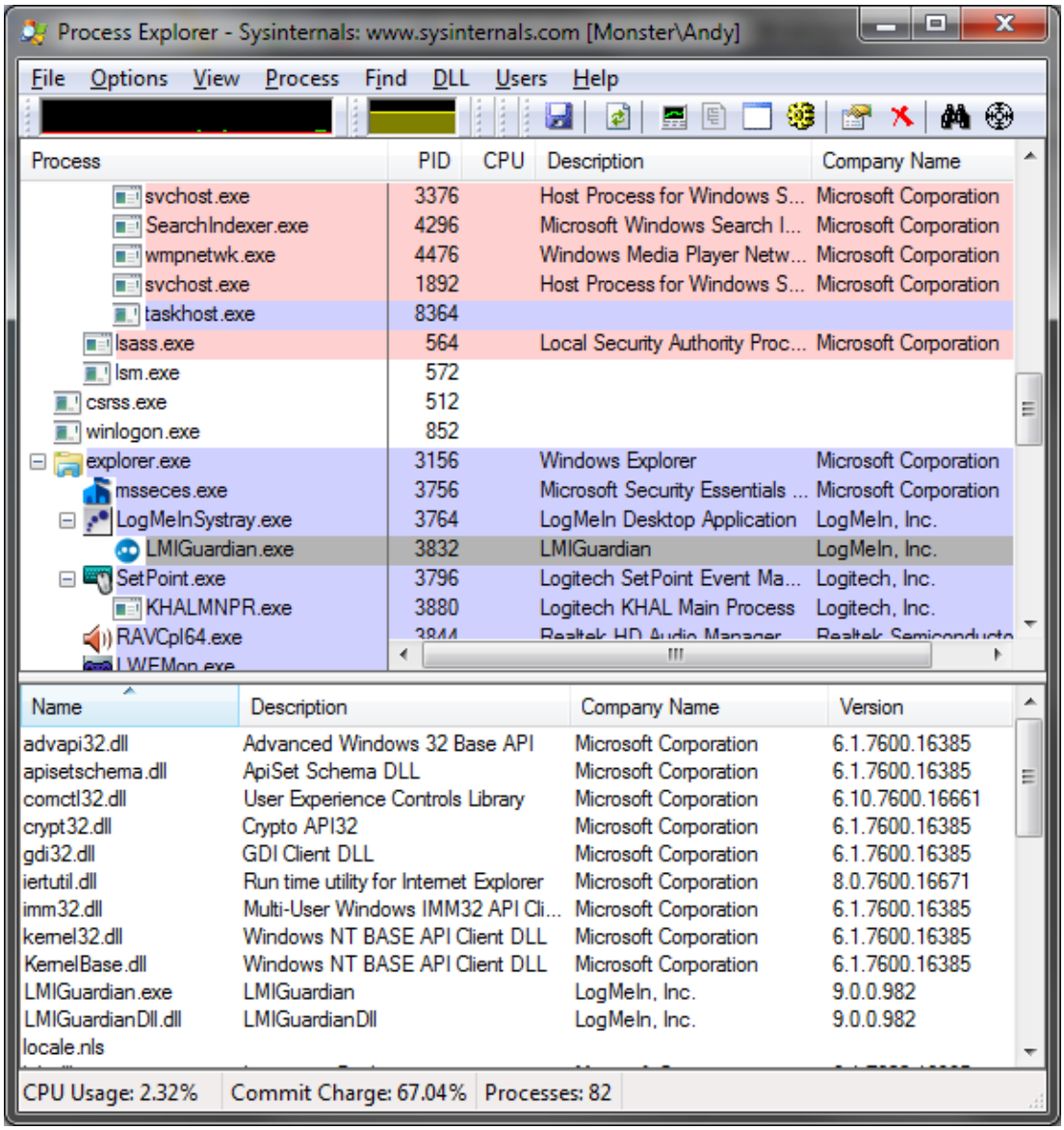
Tento program slouží ke sledování a správě procesů. Přehledným způsobem zobrazuje strom procesů, jejich popis a aktuální zátěž procesoru daným procesem. Standardní výpis lze velmi jednoduše upravit a lze zobrazovat a řadit procesy podle

využití paměti, počtu IO operací, velikosti dat přenesených během IO operací, vytížení sítě a mnoho dalších. U každého procesu lze zobrazit jeho detail. V detailu procesu je velké množství informací týkajících se procesu, například statistické údaje související s procesem, doba spuštění, doba běhu, velikost zabrané paměti, počet přenesených bytů po síti, počet bytů zapsaných do souboru, počet bytů čtených ze souboru a další. Lze si nechat vypsat všechny vlákna procesu společně s informacemi, kolik které vlákno vytěžuje operační systém, dále lze vypsat stav zásobníku vlákna, oprávnění procesu, proměnné prostředí platné pro daný proces, grafy vytížení cpu, obsazení paměti a bytů přenesených v IO operacích a mnohé další informace.

Program umožňuje spravovat procesy, umožňuje měnit prioritu, spřažení, pozastavovat, restartovat nebo ukončovat proces. Umožňuje také vytvořit tzv. dump procesu, což je uložení aktuálního stavu paměti procesu do souboru. Program nevyžaduje pro spuštění práva administrátora, avšak pouze s právy administrátor lze zobrazovat informace o procesech, které nevlastní aktuální uživatel.

Na rozdíl od programu Process Monitor tento program neukládá ani nezobrazuje historii s výjimkou grafů.

Z výše uvedených informací vyplývá, že program je vhodný pro kohokoli, kdo chce spravovat procesy a nestačí mu Windows Task Manager, který je obsažen v OS Windows. Výhodou tohoto programu je že se nemusí instalovat a že je zdarma.




Obr. 2.8 – Process Explorer

## 2.9 Process Hacker

Process Hacker[15] je open source program, jehož projekt je hostován na serveru SourceForge.net. Momentálně k projektu přispívá 5 vývojářů, vývoj probíhá přibližně 2 roky. Poslední verze programu je 2.8, vydána byla 2. 11. 2010, velikost programu je přibližně 3 MB. Program je nabízen zdarma pod licencí GNU GPL a je určen pro operační systémy Windows XP SP2 a vyšší.

Program je velmi podobný programu Process Explorer, má téměř všechny jeho funkce a navíc nabízí vyhledávání procesů pomocí regulérních výrazů, dokáže

zobrazit, na co čeká čekající vlákno a také má přehledný výpis síťových spojení. Autoři také uvádějí, že se spouští přibližně 5x rychleji a 8x méně zatěžuje procesor.

Na základě výše uvedených informací je tento program vhodný pro kohokoli, kdo chce spravovat procesy a nestačí mu Windows Task Manager, který je obsažen v OS Windows. Výhodou tohoto programu je že se nemusí instalovat, že je zdarma a že jsou dostupné zdrojové kódy.

## 2.10 Top

Top[16] je program pro sledování procesů, určený pro operační systém linux a unix. Tento program je součástí velkého množství linuxových distribucí. Program slouží jako jeden ze základních nástrojů pro správu procesů na linuxových a unixových operačních systémech. Program pracuje v textovém režimu.

Program zobrazuje procesy, které nejvíce vytěžují operační systém, program zobrazuje id procesu, název procesu, příkaz, kterým byl proces spuštěn, vlastníka procesu, prioritu, stav procesu, jak dlouho strávil proces na procesoru, aktuální procentuální vytížení procesoru, procentuální využití operační paměti daným procesem a další. Lze nastavit jaké informace má program zobrazovat pomocí argumentů, dále lze zobrazit například id procesu rodiče, id uživatele, který vlastní proces a další.

Program dále zobrazuje souhrnné informace o využití operační paměti, odkládacího souboru, využití procesoru, počty běžících, spících, zastavených a zombie procesů a další informace.

## 2.11 HTop

Htop[17] je program pro sledování a správu procesů na operačním systém linux a unix. Tento program nabízí stejnou funkčnost jako program top a nabízí další funkce navíc. Pracuje také pouze v textovém režimu. Projekt procesu je hostován na serveru SourceForge.net.

Program na rozdíl od programu top umožňuje posouvat výpis procesu jak horizontálně tak vertikálně. HTop umožňuje na rozdíl od programu Top zabít proces

nebo změnit jeho prioritu bez zadávání čísla procesu a také podporuje ovládání myší v textovém režimu.

Z výše uvedených informací vyplývá, že tento program lze doporučit komukoli, kdo potřebuje spravovat procesy a chce nástroj s pohodlnějším ovládáním než top.

## 2.12 Ps

Ps[18] je program, který vypíše aktuálně spuštěné procesy. Program je určen pro operační systém linux a unix, pracuje v textovém režimu a často je součástí distribuce operačního systému. Program na rozdíl od programů Top a HTop nezobrazuje procesy v reálném čase, ale pouze na standardní výstup vypíše seznam procesů aktuálního uživatele. Každá položka seznamu obsahuje číslo procesu, identifikátor terminálu, kde byl proces spuštěn, čas strávený na procesoru a příkaz a jméno spustitelného souboru.

Program není omezen na výpis procesů uživatele, pod kterým je spuštěn, volbou –e lze vypsat procesy všech uživatelů. Také lze vypsat více informací o jednotlivých procesech, například volbou –l. Pomocí dalších argumentů programu lze podrobně specifikovat, jaké informace o procesech se mají vypsat a také výpis procesů filtrovat podle zadaných kritérií.

Program je vhodný pro jednorázové vypsání procesů a také lze výstup tohoto program využít jako vstup pro další programy nebo skripty díky jednoduše konfigurovatelnému výpisu.

## 2.13 Pstree

Pstree[19] je program, který vypisuje aktuálně spuštěné procesy všech uživatelů na standardní výstup, výpis je organizován do stromu. Program je určen pro operační systémy linux a unix, pracuje v textovém režimu a často je součástí distribuce operačního systému. Program je podobný programu ps, ale na rozdíl od programu ps vypisuje všechny procesy a výpis zobrazuje stromově seskupen, takže je na první pohled patrné, kdo je rodičem každého procesu.

Program zobrazuje velmi málo informací o procesech, pouze název programu a volitelně lze zobrazit pouze číslo procesu, vlastníka procesu a parametry příkazu, jímž byl proces spuštěn.

Program je vhodný pro získání přehledného seznamu procesů, kde je jednoduše vidět, kdo je rodičem daných procesů a jaké mají procesy potomky.

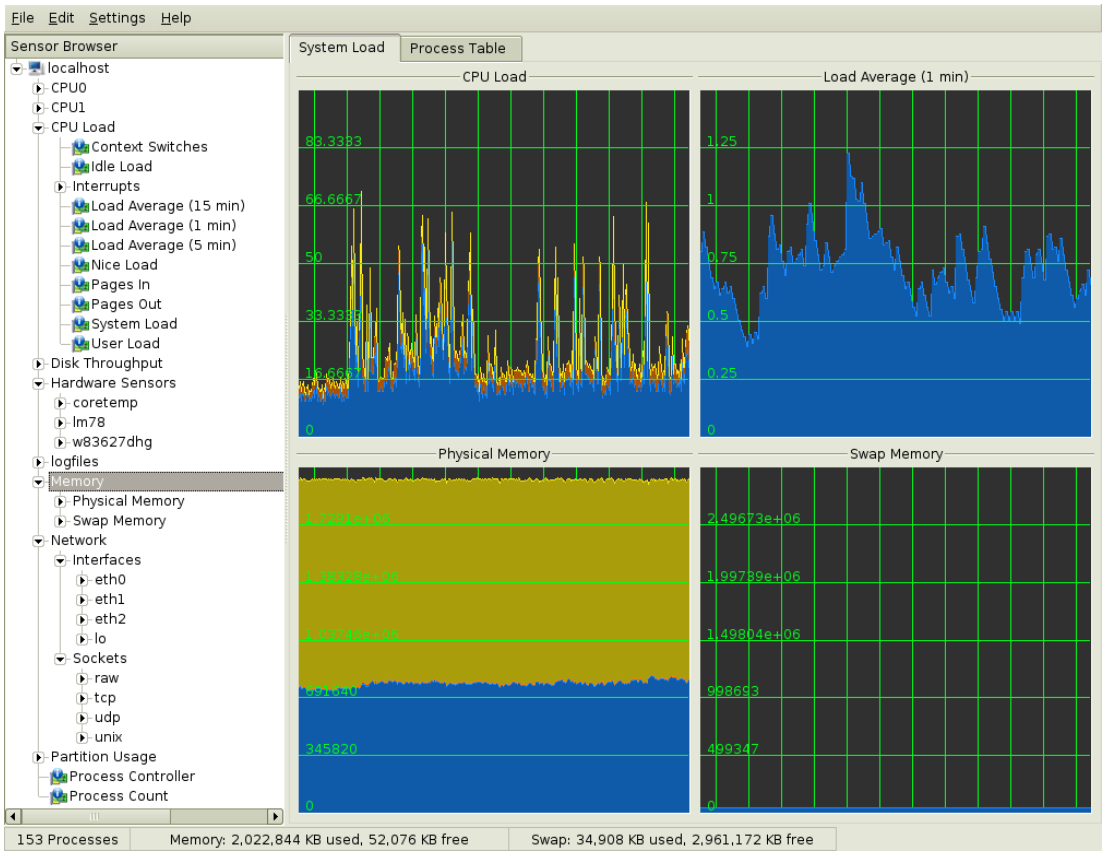
## 2.14 KDE System Guard

KDE System Guard, také známý jako KSysguard[20,21], je program, který je součástí desktopového prostředí linuxu či unixu, KDE. Aktuální verze programu je 1.2.0, program je nabízen zdarma pod licencí GPL. Program vznik jako náhrada programu KTop z KDE verze 1. Program slouží ke správě procesů a sledování využití zdrojů počítače. Program umožňuje sledovat kromě lokálního počítače také počítače vzdálené.

Program zobrazuje u každého procesu jméno, uživatele, pod jehož účtem proces běží, využití procesoru, využití paměti a titulek okna, pokud proces nějaké okno zobrazuje. Další informace o procesech lze přidat, například id procesu, terminál, na kterém proces běží, hodnotu nice, čas strávený na procesoru a další. Procesy lze zobrazit buď jako seznam nebo jako strom. Procesy lze ukončovat signálem SIGKILL. Výpis procesů lze filtrovat podle jména, titulku okna, uživatele, id procesu a dalších.

Kromě správy procesů umožňuje program sledovat využití systémových zdrojů, celkové využití paměti, procesoru, jednotlivých jader procesoru, využití swapu, vytíženost síťových rozhranní, propustnost pevného disku a další. Program lze nastavit tak, aby po překročení zadaného limitu zdroje určitého zdroje zvýraznil daný zdroj. Program umožňuje logovat do souboru vybrané informace.

Program je vhodný pro kohokoli, kdo používá desktopové prostředí KDE a potřebuje mít přehled o systémových zdrojích nebo procesech.




Obr. 2.9 – KDE System guard, převzato z [20]

## 2.15 Gnome System Monitor

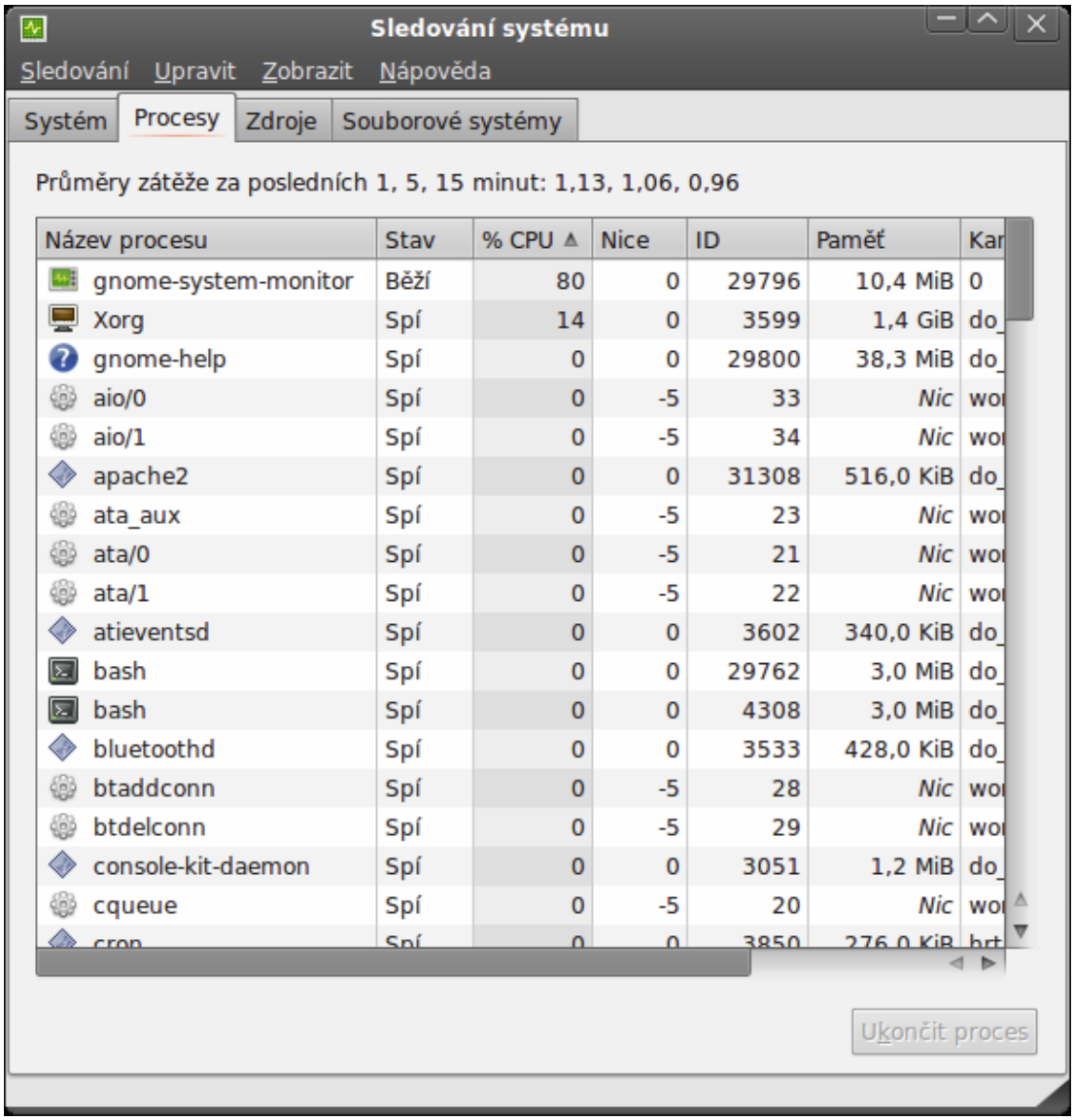
Program Gnome System Monitor[22,23] je program, které je součástí desktopového prostředí Gnome pro operační systémy Linux a Unix. Aktuální verze program je 2.28, program je nabízen zdarma pod licencí GPL. Program byl vytvořen jako náhrada programu GTop, který sloužil dříve ke správě procesů v prostředí Gnome. Program slouží ke sledování lokálního počítače, umožňuje spravovat procesy a zobrazovat využití systémových zdrojů. Dále program zobrazuje připojené souborové systémy a základní informace o nich.

Program zobrazuje u každého procesu jméno, stav, využití procesoru, hodnotu nice, id procesu, využití paměti. Lze nastavit, aby program zobrazoval i další informace o procesech, například množství vyhrazené virtuální paměti pro proces, čas strávený na procesoru, čas spuštění, parametry příkazového řádku a další. Výpis procesů lze zobrazit jako seznam nebo jako strom. U každého procesu lze zobrazit mapu paměti a lze změnit hodnotu nice. Procesy lze ukončovat standardně, signálem SIGTERM, nebo nuceně, signálem SIGKILL.

Program zobrazuje také využití zdrojů, v porovnání s programem KDE System Guard nabízí pouze zlomek informací o systémových zdrojích. Program zobrazuje grafy využití procesoru, paměti, swapu a graf síťové aktivity. Aktuální hodnoty využití těchto zdrojů zobrazuje také číselně.

Dále program zobrazuje informace o souborových systémech, zobrazeno je zařízení, adresář, do kterého je dané zařízení připojeno, typ souborového systému, celkovou velikost, volné místo a další.

Program je vhodný pro kohokoli, kdo pracuje v prostředí Gnome a potřebuje spravovat procesy nebo zobrazit využití systémových zdrojů.




Obr. 2.10 – Gnome System Monitor, převzator z [22]

## 2.16 Přehled aplikací

Všechny aplikace podporující OS Windows byly testovány na počítači s procesorem Intel Core 2 Quad disponujícím 6 GB operační paměti. Aplikace byly testovány v operačním systému Windows 7, 64-bit.

Náročnost aplikací na výkon procesoru byla přibližně stejná. Všechny aplikace vytěžovaly procesor v řádu jednotek procent. Využití procesoru bylo měřeno programem Windows Task Manager, který je součástí operačního systému. Z hlediska uživatele nebylo poznat, jak moc jsou aplikace na výkon procesoru náročné, všechny reagovali okamžitě.

Velikost všech aplikací je přibližně stejná, v řádu stovek kilobajtů až jednotek megabajtů. Ve srovnání s kapacitami dnešních pevných disků je rozdíl velikosti aplikací zanedbatelný.

Z hlediska bezpečnosti lze síťové aplikace rozdělit do dvou skupin. Aplikace s vysokou úrovní zabezpečení a aplikace, které bezpečnost neřeší. Bezpečnost je posuzována z hlediska, zda se může ke sledovanému počítači připojit kdokoli jiný bez jakýchkoli dalších informací (uživatelské jméno, heslo apod.). Aplikace využívající pro přístup ke sledovanému počítači rozhraní WMI, lze považovat za bezpečné, protože úroveň zabezpečení vychází z bezpečnosti rozhraní WMI.

#### Síťové aplikace s vysokou úrovní bezpečnosti

- Remote process monitor
- Yet another process monitor (pouze při použití rozhraní WMI)
- Nuclear remote control
- Network security taskmanager

#### Síťové aplikace s nízkou úrovní bezpečnosti

- Remote task manager
- Yet another process monitor (při použití YAPM serveru)

Tab. 2.1 – Přehled porovnávaných aplikací

| Název | OS | Síťový | Historie | Cena |
|---|---|---|---|---|
| Remote process monitor | Win NT až Win 7 | ano, WMI | ne | zdarma |
| Remote task manager | Win NT až Win XP | ano | ne | ~36 EUR |
| Yet another process monitor | Win XP až Win 7 | ano + WMI | snapshot | zdarma |
| Nuclear remote control | Win | ano | ne | zdarma |
| Network security taskmanager | Win 2000 až Win Vista | ano | částečně | ~20 USD |
| Windows Task Manager | Win | ne | ne | zdarma |
| Process Monitor | Win XP+, Win 2003+ | ne | ano | zdarma |
| Process Explorer | Win XP+, Win 2003+ | ne | ne | zdarma |
| Process Hacker | Win XP SP2+ | ne | ne | zdarma |
| Top | Unix, Linux | ne | ne | zdarma |
| HTop | Unix, Linux | ne | ne | zdarma |
| Ps | Unix, Linux | ne | ne | zdarma |
| Pstree | Unix, Linux | ne | ne | zdarma |
| KDE System Guard | Unix, Linux | ne | ne | zdarma |
| Gnome System Monitor | Unix, Linux | ne | ne | zdarma |

# 3 Návrh a realizace řešení

Cílem práce je vytvořit systém, který bude sledovat počítače v síti a informace o nich ukládat na HTTP server do SQL databáze. Tyto informace budou později zobrazovány přes webové rozhraní, aplikaci napsanou v programovacím jazyce PHP. Cílem sledování jsou především běžící procesy a služby. Systém má být konfigurovatelný, aby si sám uživatel (nebo administrátor počítače) mohl vybrat, jaká data poskytne. Systém má být založen na architektuře klient-server.

## 3.1 Použité technologie

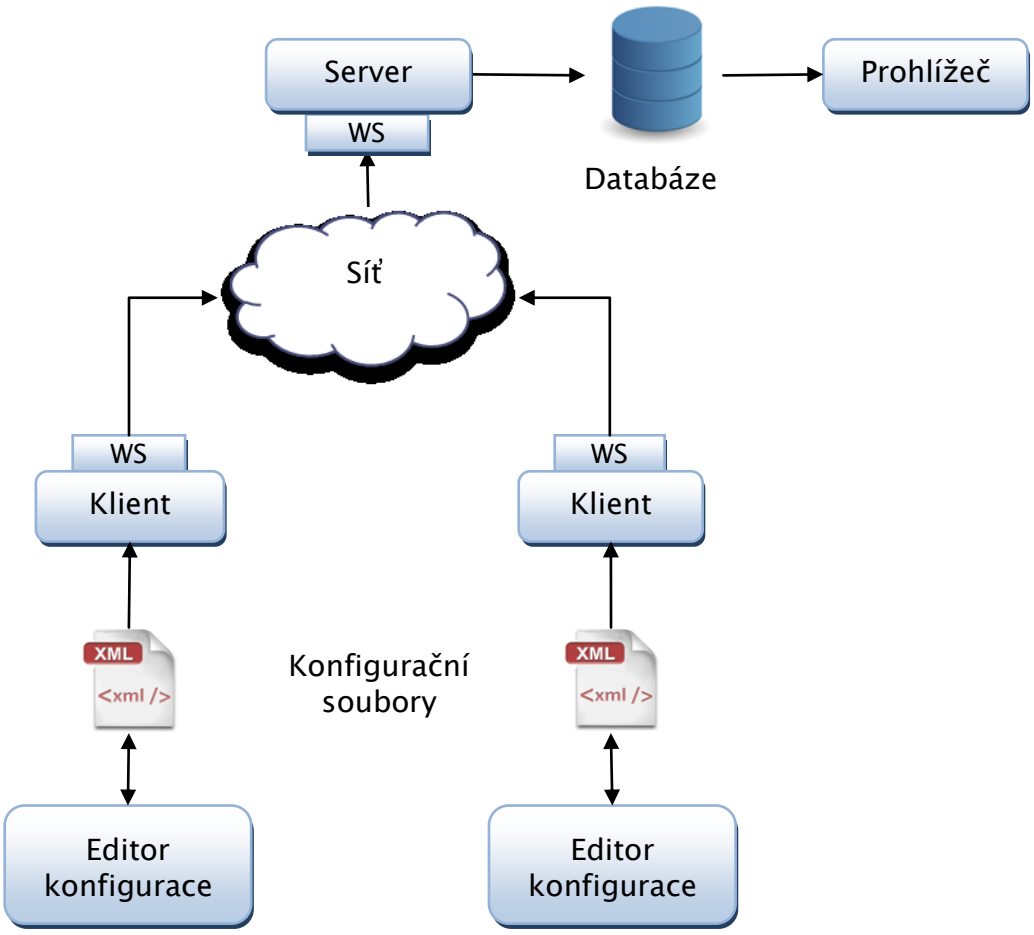
Systém má být založen na architektuře klient-server, a proto je třeba zvolit vhodnou komunikační technologii, kterou budou klienti komunikovat se serverem. Server má být napsán v jazyce PHP, a proto třeba počítat s tím, že zvolená komunikační technologie musí být v tomto jazyce podporována. Systém má ukládat informace na HTTP server, proto je volba komunikační technologie omezena pouze na protokol HTTP a jeho nadstavby. Jako nadstavbu protokolu HTTP lze využít webovou službu[24].

Webová služba je vhodným řešením, protože má standardizovaný formát přenosu informací mezi počítači, v programovacím jazyce PHP je jednoduše použitelná a je platformě nezávislá. Webová služba také sama poskytuje své schéma (WSDL), a proto je velmi jednoduché napsat klienta, který s ní komunikuje. Některé nástroje, například Visual Studio, umí vytvořit podle WSDL schématu třídy, pro komunikaci se službou a práce programátora je tak ulehčena.

Vzhledem k tomu že data mají být na serveru uložena v SQL databázi, bylo nutné rozhodnout se jaký databázový systém použít. Bylo třeba, aby aplikace napsaná v jazyce PHP mohla s tímto databázovým systémem komunikovat. Žádné další požadavky na databázový systém kladeny nebyly, a proto byl zvolen jeden z nejrozšířenějších databázových systémů používaný společně s PHP, MySQL. Výhoda systému MySQL spočívá v tom, že je zadarmo a že je dostupný na velkém množství webhostingů podporujících PHP.

Klient může být napsán v jakémkoli programovacím jazyce. Jako komunikační technologie byla zvolena webová služba, a proto je vhodné, aby programovací jazyk, v němž bude klient napsán, umožňoval připojení k webové službě. Protože na cílový programovací jazyk nebyly kladeny žádné další požadavky, byl zvolen jazyk C#.

## 3.2 Architektura




Obr. 3.1 – Architektura serverové části aplikace

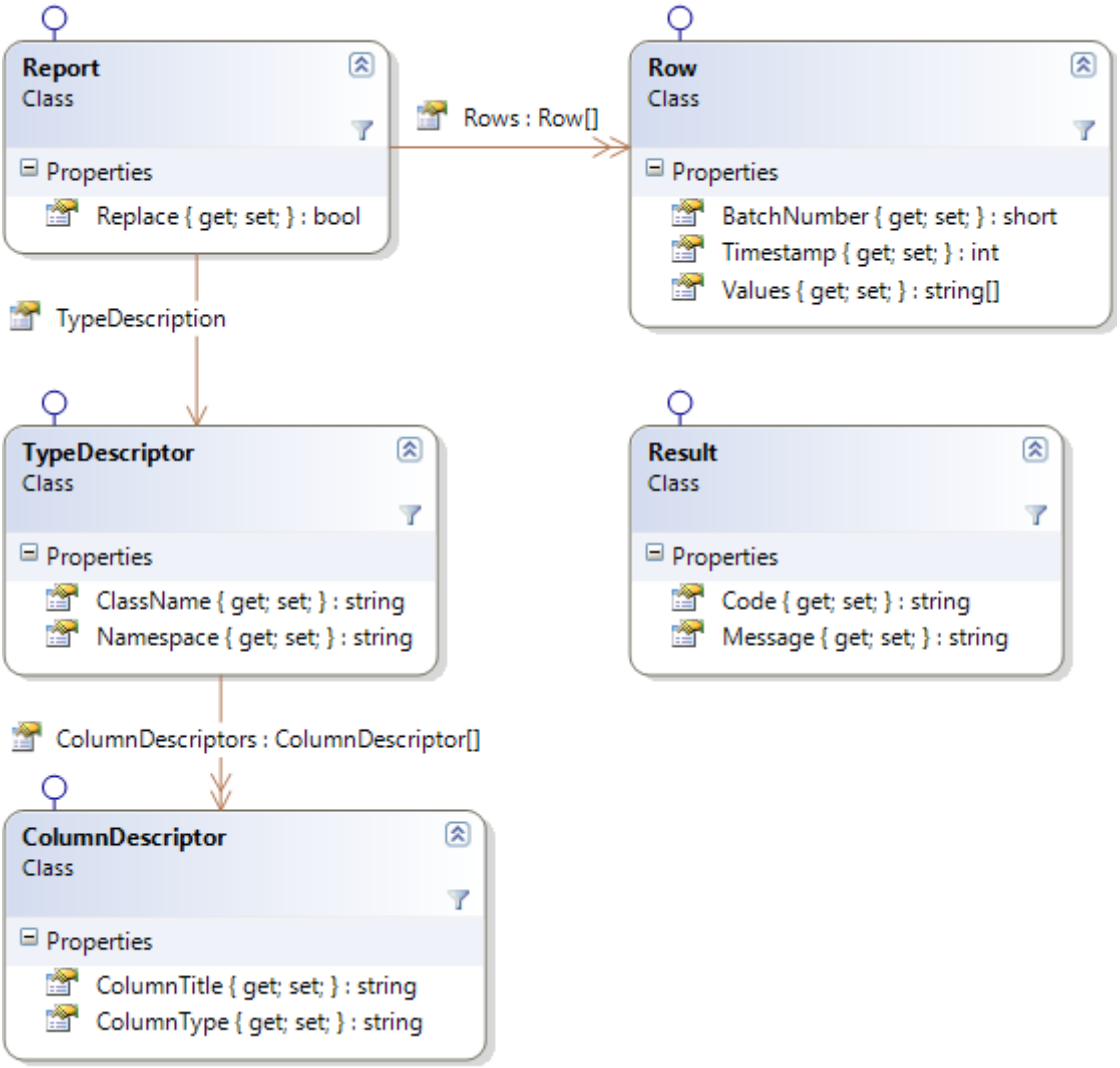
### 3.2.1 Server

Serverem je PHP aplikace, která poskytuje webovou službu. Tato webová služba přijímá hlášení neboli tzv. reporty od klientů a ukládá je do databáze. Server byl napsán ve vývojovém prostředí Eclipse for PHP Developers.

### Popis reportu

Architektura serveru byla navržena tak, aby byla dostatečně univerzální a umožňovala ukládat libovolné objekty skládající se z jednoduchých datových typů (celočíselný typ, číslo s plovoucí desetinnou čárkou, řetězec znaků, boolean a další).

Každý report, který přijde od klienta, obsahuje tabulku s nasnímanými daty. Tato tabulka obsahuje sloupce a řádky. Jeden řádek představuje jednotku nasnímaných informací. Typickým příkladem řádku jsou informace o jednom běžícím procesu na klientském počítači. V tomto případě by mohly být sloupce tabulky tyto: jméno procesu, velikost paměti obsazené procesem, název uživatele, který proces spustil a další. Jaké sloupce klient pošle je čistě na něm.




Obr. 3.2 – Diagram tříd posílaných objektů

Každý řádek obsahuje čas, kdy byly informace pořízeny a číslo skupiny. Číslo skupiny určuje, do jaké skupiny v rámci reportu řádek patří. Pokud by klient snímal informace o procesech například každých 10 sekund a report odesílal jednou za minutu, bude report obsahovat 6 skupin, v každé skupině může být různý počet procesů.

Tab. 3.1 – Příklad reportu, řádky

| Timestamp | BatchNumber | Process ID | Caption | CPU Usage |
|---|---|---|---|---|
| 1256953732 | 0 | 5764 | firefox.exe | 0.48 |
| 1256953732 | 0 | 5064 | cmd.exe | 0.01 |
| 1256953732 | 0 | 7012 | devenv.exe | 0.04 |
| 1256953742 | 1 | 5764 | firefox.exe | 0.18 |
| 1256953742 | 1 | 5064 | cmd.exe | 0.01 |
| 1256953742 | 1 | 7012 | devenv.exe | 0.34 |

Report obsahuje kromě tabulky dat také záhlaví a příznak nahrazení. Příznak nahrazení určuje, zda mají být starší reporty tohoto typu z databáze před vložením nově příchozího reportu odstraněny.

Záhlaví (třída *TypeDescriptor*) obsahuje název reportu, jmenný prostor a množinu deskriptorů sloupců (třída *ColumnDescriptor*). Název reportu a jmenný prostor si volí sám klient. Jmenný prostor reportu je název kategorie, ve které bude report k nalezení na serveru. Deskriptor sloupce popisuje jedno pole řádku. Deskriptor obsahuje název sloupce a jeho datový typ. Příklad seznamu deskriptorů sloupců pro výše uvedený report.

Tab. 3.2 – Příklad reportu, deskriptory sloupců

| ColumnTitle | ColumnType |
|---|---|
| Process ID | int |
| Caption | string |
| CPU Usage | float |

Podporované datové typy a jejich protějšek v MySQL databázi shrnuje následující tabulka.

Tab. 3.3 – Podporované datové typy

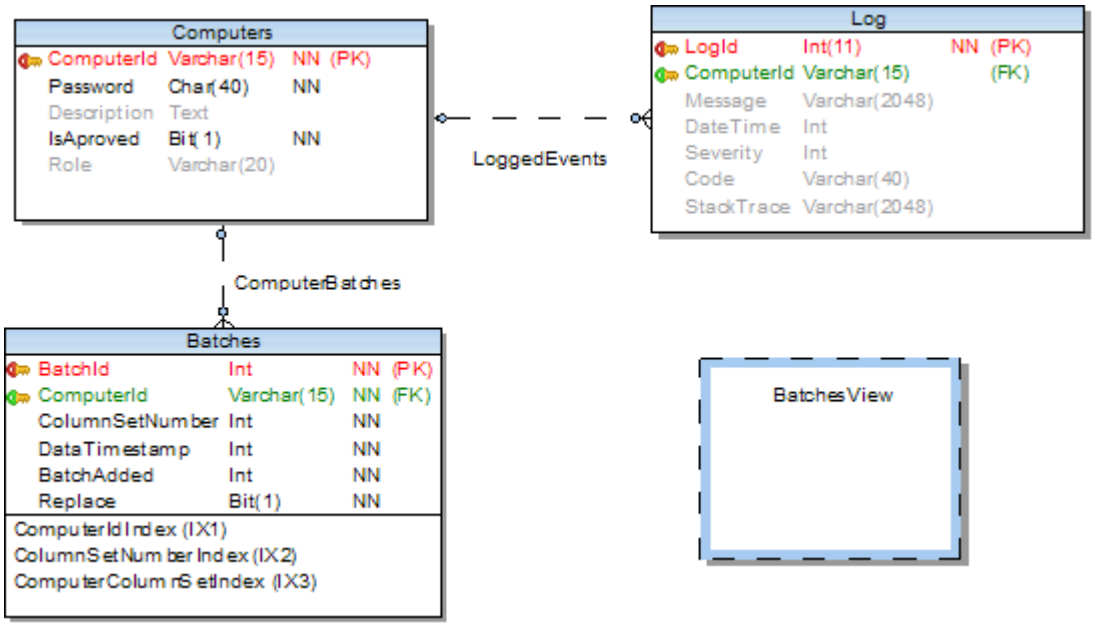
| Název typu | Typ MySQL |
|---|---|
| text | TEXT |
| string | TEXT |
| bool | BIT(1) |
| boolean | BIT(1) |
| double | DOUBLE |
| real64 | DOUBLE |
| float | FLOAT |
| real32 | FLOAT |
| uint8 | TINYINT UNSIGNED |
| byte | TINYINT UNSIGNED |
| uint16 | SMALLINT UNSIGNED |
| ushort | SMALLINT UNSIGNED |
| uint32 | INT UNSIGNED |
| uint | INT UNSIGNED |
| uint64 | BIGINT UNSIGNED |
| ulong | BIGINT UNSIGNED |
| int8 | TINYINT |
| sint8 | TINYINT |
| sbyte | TINYINT |
| int16 | SMALLINT |
| sint16 | SMALLINT |
| short | SMALLINT |
| int32 | INT |
| sint32 | INT |
| int | INT |
| datetime | BIGINT |
| int64 | BIGINT |
| sint64 | BIGINT |
| long | BIGINT |
| default | TEXT |

### Ukládání reportu

Server ukládá do databáze různé reporty. Při příchodu reportu je nutné rychle zjistit, zda se jedná o známý nebo neznámý report. Pokud server ještě daný report nezpracovával, jedná se o neznámý report, pokud už server report se stejným názvem, stejným jmenným prostorem a stejnými sloupci zpracovával, jedná se o známý report. Identifikace reportu probíhá na základě otisku záhlaví. Ze záhlaví je spočítán otisk a tento otisk je pak vyhledán v databázi. Pokud databáze daný otisk

neobsahuje, jedná se o neznámý report a je třeba pro něj vytvořit tabulku, do které budou ukládána data. Pokud tabulka již existuje, jsou porovnány existující sloupce tabulky se sloupci reportu a v případě, že se neshodují názvy a datové typy, tak je tabulka o potřebné sloupce rozšířena. Po vytvoření či modifikaci tabulky se typ reportu přidá do databáze mezi známé typy reportů a při příštím příchodu tohoto typu reportu se již žádná tabulka nevytváří ani nemodifikuje, pouze se z databáze načtou informace potřebné pro správné uložení tohoto typu reportu. Jedná se o název tabulky, názvy sloupců a další.

Databáze je rozdělena na model a meta model. Model obsahuje konkrétní tabulky reportů, například tabulku obsahující běžící procesy. Model je dynamický a je modifikován za běhu aplikace podle jejích potřeb. Kromě konkrétních tabulek reportů obsahuje také tabulku počítačů, log a tabulku, která udržuje informace o vkládaných reportech. Tato tabulka obsahuje identifikátor počítače, od kterého daný report pochází, čas vložení reportu do databáze a další informace.

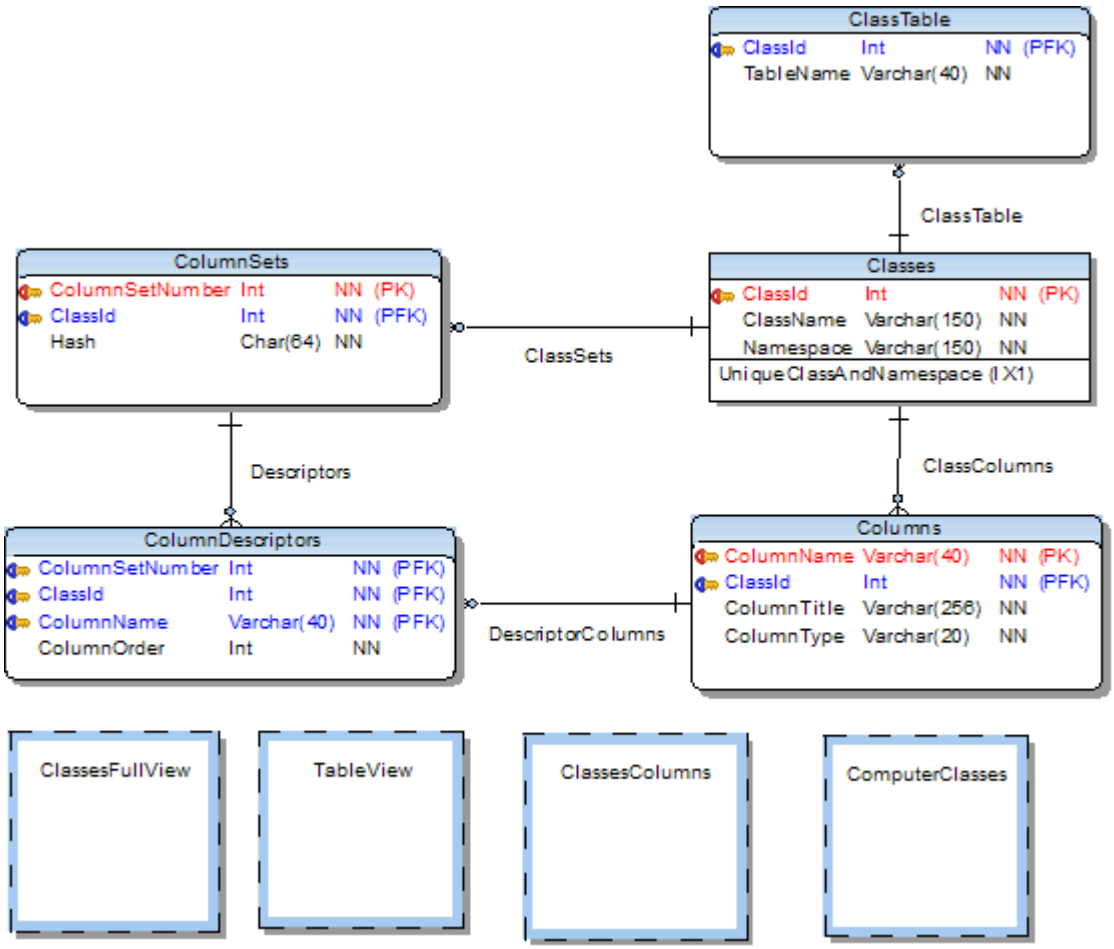



Obr. 3.3 – ER diagram modelu

Meta model je statický a obsahuje informace o tabulkách reportů. Kromě názvu tabulky obsahuje meta model také otisk záhlaví reportu, názvy sloupců jejich datové typy a další.

Název tabulky je odvozen pouze od názvu reportu a jeho jmenného prostoru. Pokud má více reportů stejný název i jmenný prostor, ale jiné sloupce, sdílejí spolu tyto reporty jednu tabulku a do nezadaných sloupců jsou vkládány hodnoty NULL. Je to z toho důvodu, aby při rozšíření reportu o další sloupce nevznikaly v databázi další velmi podobné tabulky a data byla pohromadě.

Serverová část využívá pro přístup k databázi framework *Dibi*[25]. Jedná se o tenkou vrstvu, která urychluje psaní SQL dotazů a stará se o ošetření vstupních dat. *Dibi* je open source software, dostupný pod licencí BSD, GNU GPL 2 a GNU GPL 3. Podporuje mnoho různých databázových systémů, například MySQL, MSSQL, Oracle a další.




Obr. 3.4 – ER diagram meta modelu

### Webová služba

Jedná se o službu, která běží na serveru. Tato služba zabezpečuje ukládání informací od klientů do relační databáze. Služba čeká na požadavky od klientů, tyto požadavky vyřizuje a zpět klientům posílá odpovědi.

Server vyžaduje, aby se klient před předáním reportů na server přihlásil, k tomu slouží operace *Login*. Pokud klient není přihlášen a chce provést jinou operaci než *Login*, vrátí server http kód 401, unauthorized. Na toto by měl klient reagovat přihlášením a po úspěšném přihlášení se znovu pokusit provést danou operaci.

Server ještě umožňuje druhý způsob ověřování klienta a to přes http basic authentification. Toto ověřování je méně bezpečné, protože přenáší nezašifrované heslo v každém požadavku na server. Při tomto ověřování není třeba používat operaci *Login*, stačí předávat identifikátor počítače a heslo v každém požadavku tak, jak vyžaduje http basic authentification.

Z hlediska bezpečnosti je doporučené používat metodu ověření uživatele přes operaci *Login*, ta zajistí, že je heslo přeneseno pouze jednou. Tato operace může být pro zvýšení bezpečnosti volána přes https. Díky tomu nebude přenášeno heslo přes internet v nezašifrované podobě.

Webová služba obsahuje konfiguraci v konfiguračním souboru *service/config.php*. Zde lze měnit, zda je povolen ladící mód a je možné změnit dobu, za kterou bude klient automaticky odhlášen, pokud neprovede v tomto časovém intervalu žádnou akci.

#### Operace Login

Tato operace slouží pro přihlášení klienta. Operace má dva argumenty, identifikátor klienta a heslo. Operace vrací token, který je třeba předávat v každém následujícím požadavku, aby byl klient jednoznačně identifikován. Tento token je nutné předávat v http hlavičce *token*.

#### Operace HelloWorld

Tato operace má jeden argument, textový řetězec. Operace vrací textový řetězec „Hello“, za kterým následuje předaný text. Tato operace slouží pro testování a lze pomocí ní ověřit, zda byl klient úspěšně přihlášen.

#### Operace SendReports

Tato operace předává serveru reporty, argumentem této operace je pole reportů. Tyto reporty mohou být různého typu. Reporty jsou ukládány do databáze způsobem zmíněným v předcházející kapitole. Tato operace vrací jako návratový typ pole objektů typu *Result*. Každý *Result* obsahuje kód a popis chyby. V případě úspěchu je navrácen kód „OK“. Počet objektů typu *Result* v poli odpovídá počtu předaných reportů. Pokud je v konfiguraci povolen ladící mód, případná chybová hlášení jsou přímo poslána klientovi. Pokud je ladící mód zakázán, klientovi jsou posílány pouze neurčité zprávy o tom, že se operace nezdařila. Konkrétní chybové zprávy jsou vždy zapsány do logu (tabulka *Log*).

### 3.2.2 Prohlížeč

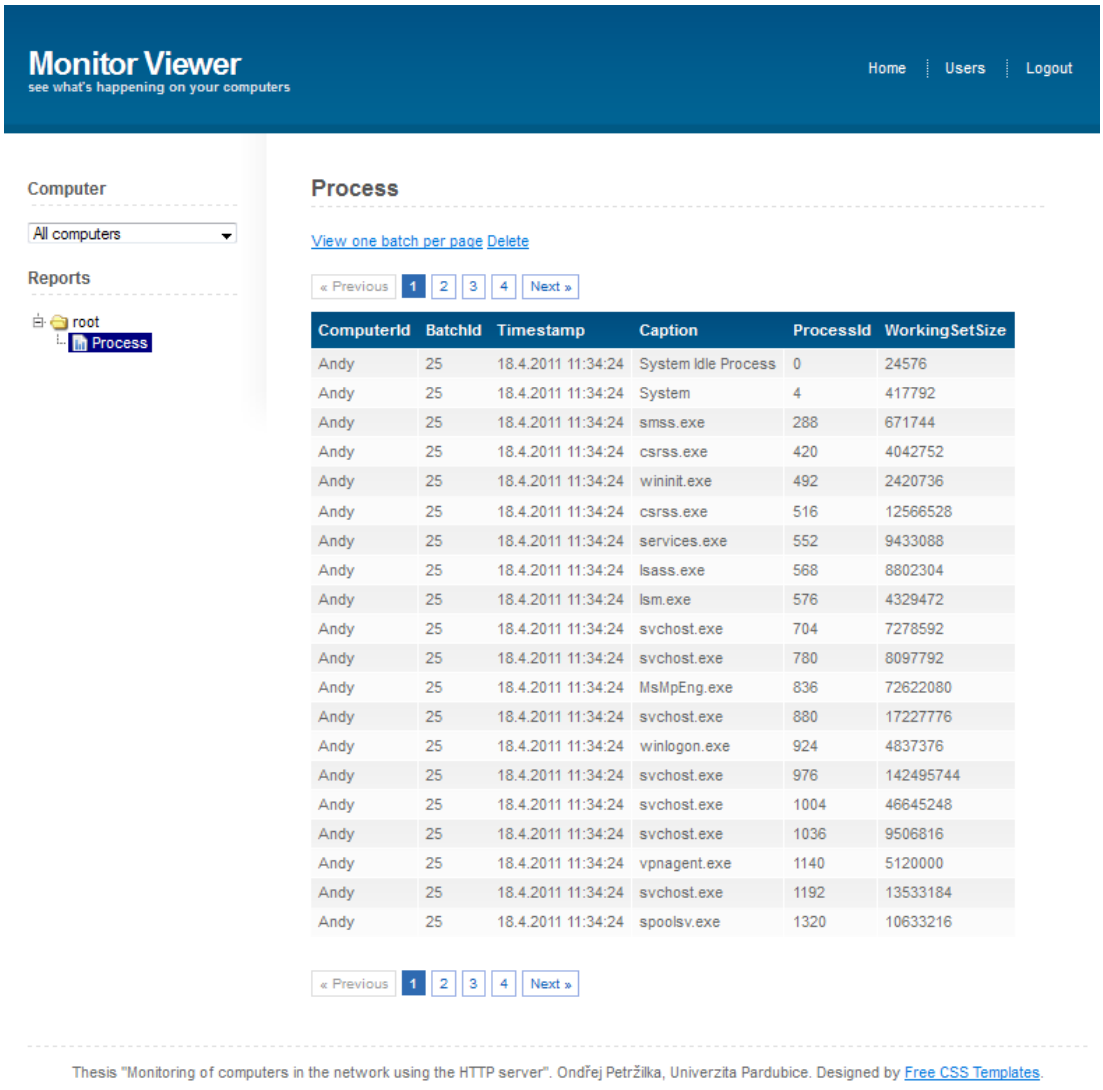
*Prohlížeč* je aplikace, která je zodpovědná za zobrazení dat uložených v databázi. Aplikace je postavena na *Nette*[26] frameworku, který je založen na architektuře MVP. *Nette* framework je open source software, dostupný pod BSD licencí. Mezi jeho hlavní přednosti patří vysoká rychlost, bezpečnost a strmá křivka učení jak pracovat s tímto frameworkem.

Pro přístup k aplikaci je třeba se přihlásit, přihlášení je provedeno zadáním identifikátoru klienta a hesla. Administrátor aplikace může spravovat klienty. Těmto klientům lze nastavit, zda budou mít přístup do prohlížeče, nebo zda budou moci pouze posílat reporty přes webovou službu. Klientům lze také přidat práva administrátora, aby mohli spravovat ostatní uživatele.

Pokud se přihlásí uživatel s právy administrátora, může zobrazovat reporty všech klientů a reporty mazat. Pokud se přihlásí klient bez práv administrátora, může zobrazovat pouze svoje reporty, mazat nemůže žádné reporty.

Prohlížeč lze konfigurovat pomocí konfiguračního souboru *config/config.php*. Lze nastavit formát data, počet řádků tabulky, do které se vypisují reporty a počet dní, po jejichž uplynutí jsou reporty automaticky mazány. Aby fungovalo automatické mazání, je nutné nastavit periodické spouštění skriptu *cron_daily.php* na spouštění jednou denně.

Jako layout prohlížeče je použit volně dostupný CSS layout z webové stránky http://www.freecsstemplates.org/. Layout je dostupný zdarma, ale je nutné zachovat v zápatí stránky odkaz, odkud daný layout pochází. Layout definuje rozložení stránky, barvy, fonty, záhlaví, zápatí, formáty odstavců, obrázků a tabulek.




Obr. 3.5 – Prohlížeč

### 3.2.3 Klient

Klient zajišťuje sbírání informací, vytváření reportů a jejich odesílání na server. Je napsán jako služba operačního systému v programovacím jazyce C#. Klient byl napsán ve vývojovém prostředí Visual Studio 2010. Dále byl použit nástroj *StyleCop*, pro dodržení jednotných pravidel psaní kódu. Klient je distribuován jako instalační balíček MSI, instalátor automaticky nainstaluje službu a nastaví její automatické spouštění při startu operačního systému. Pro instalaci klienta je zapotřebí mít práva administrátora.

Zdrojem dat pro klienta je WMI, klient čte data z WMI a vytváří z nich reporty, které pak odesílá na server. Tímto způsobem je možné číst jakákoli textová či číselná data obsažená ve WMI.

Klient je konfigurován konfiguračním souborem, tento soubor je ve formátu XML a obsahuje adresu serveru, přístupové údaje, interval odesílání reportů a definice reportů. Definice reportu se skládá ze jména třídy WMI, jmenného prostoru WMI, názvu reportu, názvu jmenného prostoru reportu, intervalu čtení dat z WMI, příznaku nahrazení a názvu sloupců. Každý sloupec může obsahovat také alias, pod kterým bude přenášen na server.

Klient po spuštění načte konfigurační soubor. Pro každý typ reportu je poté vytvořeno vlastní vlákno, které se stará o čtení dat z WMI, vytváření reportu a jeho odesílání.

Konfigurační soubor je umístěn v adresáři označovaném jako *CommonApplicationData*, jedná se o adresář, jehož cesta nemusí být vždy stejná. Ve výchozím nastavení Windows 7 se jedná o adresář *C:\ProgramData*. Celá cesta konfiguračního souboru je uvedena níže. Přístup k souboru je omezen tak, aby nemohl uživatel bez práv administrátora měnit konfiguraci.

```
[CommonAppDataFolder]\MonitorService\MonitorService.config
```

Součástí instalačního balíčku je editor konfigurace. Editor konfigurace slouží k úpravě konfigurace a je spouštěn standardně s právy administrátora, toho je docíleno použitím tzv. manifestu. Pokud by byl program spuštěn bez práv

administrátora, nebude moci měnit konfigurační soubor, protože k úpravě konfiguračního souboru je zapotřebí práv administrátora.

Celá aplikace klienta je rozdělena do několika samostatných částí, každá z částí je reprezentována DLL knihovnou nebo spustitelným souborem.

### Monitor.Client

Jedná se o assembly obsahující třídy, které byli automaticky vygenerovány Visual Studiem z WSDL schématu webové služby. Jsou to třídy zabezpečující připojení k webové službě a volání operací této služby.

Dále assembly obsahuje pomocné třídy, které zabezpečují předávání bezpečnostního tokenu v hlavičce HTTP požadavku, konverzi datového typu *Datetime* na unix timestamp a třídu zapouzdřující klienta webové služby, která zabezpečuje automatické přihlašování v případě potřeby.

### Monitor.Host

*Monitor.Host* je assembly, která obsahuje třídy a rozhraní vytvářející abstrakci zásuvného modulu a hosta. Host řídí zásuvné moduly a využívá assembly *Monitor.Client* pro připojení k webové službě a provádění operací této služby.

Do hosta lze přidat, po jeho vytvoření, zásuvné moduly, každý zásuvný modul je zodpovědný za získávání reportů, způsob, jakým zásuvný modul získává reporty, není pro hosta důležitý. Důležité je, že host je upozorněn pokaždé, když má zásuvný modul k dispozici nový report. Host přebírá od modulu tyto reporty a posílá je na server pomocí webové služby.

### Monitor.Plugins.WmiPlugin

Tato assembly obsahuje zásuvný modul, který je zodpovědný za získávání dat z WMI. Jak bylo zmíněno výše, konfigurační soubor definuje, jaké informace budou z WMI získávány a kdy. Ke čtení a úpravě konfiguračního souboru slouží assembly *Monitor.Plugins.WmiPlugin.Configuration*.

Třída *WmiPlugin* implementuje rozhraní *IMonitorPlugin* definované v assembly *Monitor.Host*. Tato třída využívá třídu *WmiFetcher* pro čtení dat z WMI.

Pro každý typ reportu je vytvořena jedna instance třídy *WmiFetcher*, která dostane při vytvoření konkrétní informace o tom, co má z WMI číst.

Třída *WmiFetcher* je přímo zodpovědná za čtení dat z WMI a vytváření reportů, které vrací třídě *WmiPlugin* a ta je předává dále.

### Monitor.Plugins.WmiPlugin.Configuration

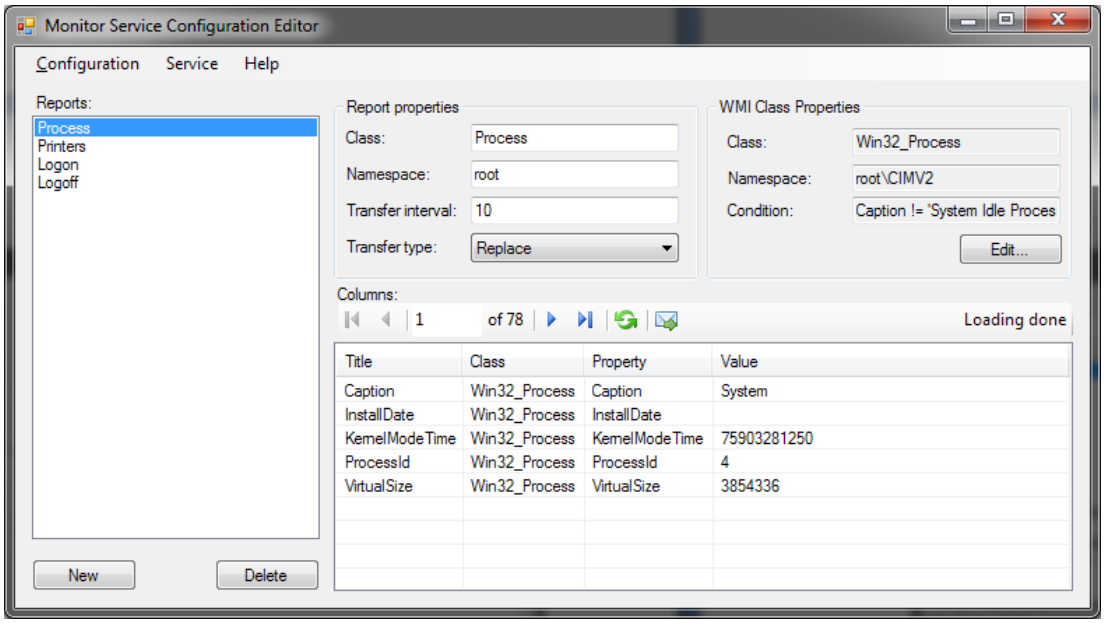
Jedná se o assembly, která je zodpovědná za čtení a úpravu konfigurace zásuvného modulu *WmiPlugin*. Přístup ke konfiguraci je založen na assembly *System.Configuration*, která je standardní součástí *.NET Frameworku*. Třídy obsažené v assembly *Monitor.Plugins.WmiPlugin.Configuration* definují vlastní sekci konfigurace a konfigurační elementy.

### Monitor.Plugins.WmiPlugin.Gui

Tato assembly je přímo spustitelným souborem a jedná se o editor konfigurace zásuvného modulu *WmiPlugin*. Tato assembly obsahuje třídy a dialogová okna potřebná ke konfiguraci zásuvného modulu.

Editor konfigurace sestává ze dvou základních dialogových oken. Hlavní dialogové okno zobrazuje existující reporty, vlastnosti vybraného reportu a náhled dat, která jsou aktuálně k dispozici. Daná třída WMI může obsahovat více než jeden objekt, a proto jsou k dispozici ovládací prvky určené k procházení kolekce těchto objektů.

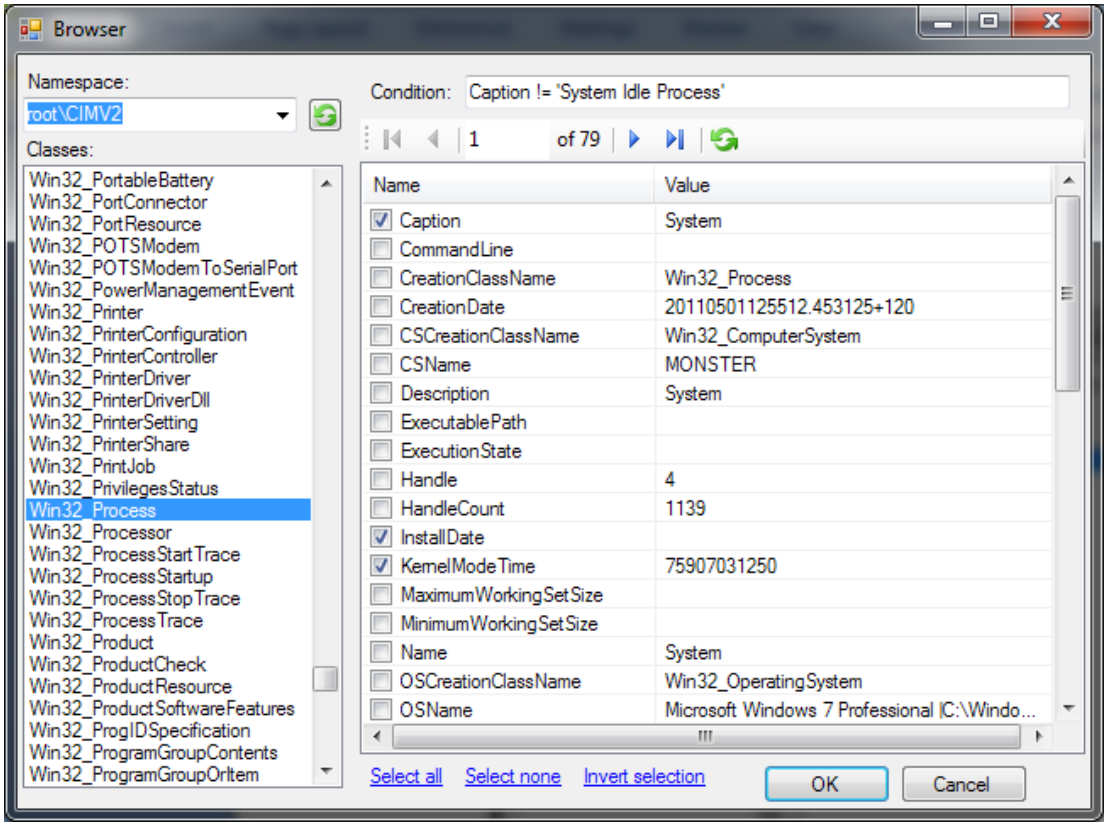
Uživatel může vytvářet, mazat a upravovat typy reportů. Uživatel také může nechat jednorázově vygenerovat report a odeslat ho na server. Tato funkcionalita slouží k tomu, aby bylo možné podívat se, jak bude daný report vypadat na serveru, až ho tam bude periodicky odesílat služba. Po upravení všech požadovaných typů reportů uživatel ukončí aplikaci, která se zeptá, zda má provedené změny uložit do konfiguračního souboru. Po uložení změn do konfiguračního souboru je třeba restartovat službu *MonitorService*, aby se tato služba začala chovat podle změněné konfigurace. Restart služby lze provést také přímo z editoru konfigurace. Slouží k tomu položka menu *Configuration/Save & Restart.* Dále je k dispozici menu *Service,* které umožňuje spouštět a zastavovat službu.




Obr. 3.6 – Editor konfigurace, hlavní okno

Druhým důležitým dialogovým oknem je tzv. *browser*. *Browser*, slouží k výběru třídy WMI a atributů této třídy. *Browser* obsahuje ovládací prvky určené k volbě jmenného prostoru a třídy WMI. Dále obsahuje seznam všech atributů dané třídy a náhled aktuálních dat. Třída může obsahovat více než jeden objekt, a proto jsou k dispozici ovládací prvky pro zobrazení dat dalších objektů dané třídy.

Uživatel vybere požadovanou třídu a zaškrtá atributy, které má obsahovat daný typ reportu. Také může doplnit podmínku ve formátu WQL. Poté výběr potvrdí tlačítkem OK.




Obr. 3.7 – Editor konfigurace, browser

### Monitor.Service

Tato assembly je službou operačního systému, která v pravidelných intervalech odesílá získané reporty webové službě. Assembly obsahuje třídu *MonitorService*, která představuje tuto službu. Třída *MonitorService* používá třídu *MonitorHost*, definovanou v assembly *Monitor.Host* a vytváří zásuvný modul *WmiPlugin*, který předává této třídě. *MonitorService* předává třídě *WmiPlugin* konfiguraci, kterou načítá pomocí assembly *System.Configuration*. Z této konfigurace je získána sekce *WmiPluginSection*, se kterou dále pracuje třída *WmiPlugin*.

Nutno zdůraznit, že třída *MonitorService* spoléhá na konkrétní zásuvný modul *WmiPlugin*. Do budoucna by bylo možné přepsat tuto třídu tak, aby hledala assembly zásuvných modulů v určitém adresáři a vytvářela instance těchto modulů za běhu. Tím by bylo docíleno nezávislosti služby na konkrétních zásuvných modulech a bylo by možné vytvářet a používat další zásuvné moduly bez zásahu do zdrojových kódů třídy *MonitorService*.

Assembly *Monitor.Service* obsahuje také třídu instalátoru zodpovědnou za instalaci služby. Tato třída byla automaticky vygenerována Visual Studiem a je využívána instalátorem.

### Setup

Setup je projekt, který je součástí zdrojových kódů a zabezpečuje vytvoření instalačního balíčku MSI. Instalační balíček obsahuje všechny výše zmíněné assembly, konfigurační soubor a zástupce editoru konfigurace, který je při instalaci umístěn do nabídky Start. Instalační balíček při instalaci nainstaluje assembly *Monitor.Service* jako službu operačního systému Windows a nastaví automatické spouštění této služby při startu operačního systému. Služba má jméno *MonitorService*.

Instalace pomocí vygenerovaného instalačního balíčku MSI je jednoduchá, stačí postupovat podle pokynů na obrazovce. Při instalaci je možné zvolit adresář, do kterého bude aplikace nainstalována.

# Závěr

V praktické části práce byly vytvořeny dvě aplikace, které umožňují sledovat počítače v síti. Server je platformě nezávislá aplikace napsaná v jazyce PHP, která poskytuje webovou službu. Tuto webovou službu lze využít i k jiným účelům než je sledování počítačů. Na server lze ukládat jakékoli informace, za předpokladu dodržení daného datového formátu. Do budoucna by bylo vhodné rozšířit server tak, aby umožňoval přenos dat v binární podobě. Prohlížeč dat by bylo vhodné rozšířit tak, aby umožňoval transformovat data na základě pravidel definovaných uživatelem a dále tyto data vyhodnocoval podle dalších pravidel a informoval správce pomocí emailu o definovaných skutečnostech.

Aplikace klient, napsaná v programovacím jazyce C# je platformě závislá na operačním systému Windows. Tato aplikace čte data z rozhraní WMI a odesílá je na HTTP server pomocí webové služby. Klient běží jako služba operačního systému a instaluje se pomocí instalačního balíčku MSI. Součástí instalačního balíčku je editor konfigurace, který mění konfiguraci klienta. Pomocí editoru konfigurace je možné zvolit jaké informace z WMI se budou číst a přenášet na server. Editor konfigurace umožňuje zvolit časový interval čtení dat z WMI. Konfigurace klienta je zabezpečena tak, aby ji mohl měnit pouze uživatel s právy administrátora.

Vytvořené aplikace umožňují efektivně sledovat procesy a další informace, které rozhraní WMI poskytuje. Na rozdíl od většiny porovnávaných aplikací je umožněno sledovat procesy automaticky a pohodlně na větším množství počítačů. Velmi jednoduše lze sledovat procesy, nainstalované programy, běžící služby a kdy se jaký uživatel přihlásil či odhlásil.

# Seznam příloh

# Použitá literatura


[1] *Wikipedia* [online]. 2011, last modified on 20 March 2011 [cit. 2011-04-28]. Process (computing). Dostupné z WWW: <http://en.wikipedia.org/wiki/Process_%28computing%29>.

[2] WATSON, Jon. *A History Of Computer Operating Systems : Unix, Dos, Lisa, Macintosh, Windows, Linux*. Ann Arbor : Nimble Books LLC, 2008. 60 s. ISBN 1-934840-45-9, 978-1-934840-45-0.

[3] SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. *Operating System Concepts*. 8th edition. Jefferson City : John Wiley and Sons, 2009. 992 s. ISBN 9780470128725, 978-0470128725.

[4] SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. *Operating System Concepts with Java*. 8th edition. Jefferson City : John Wiley and Sons, 2009. 1040 s. ISBN 9780470509494, 978-0470509494.

[5] STALLINGS, William. *Operating systems : internals and design principles*. 4th ed. Upper Saddle River : Prentice Hall, 2001. xviii, 779 s. ISBN 0-13-032986-X.

[6] *GITAM University : Computer Science Courseware* [online]. 2011 [cit. 2011-04-28]. Process Concept. Dostupné z WWW: <http://www.gitam.edu/eresource/comp/gvr%28os%29/4.1.htm>.

[7] *File.net* [online]. 2008 [cit. 2010-11-06]. Remote Process Viewer. Dostupné z WWW: <http://www.file.net/remote-process-viewer/index.html>.

[8] *DeviceLock* [online]. 2010 [cit. 2010-11-06]. Remote Task Manager for Windows NT/2000/XP and Windows Server 2003, the network management software. Dostupné z WWW: <http://www.devicelock.com/rtm/index.htm>.

[9] *Yet Another (remote) Process Monitor* [online]. 2009-16-12 [cit. 2010-11-06]. Yet Another Process Monitor (YAPM). Dostupné z WWW: <http://yaprocmon.sourceforge.net/index.html>.

[10] AYA CZ, spol. s.r.o. *Webzdarma.cz* [online]. 2010 [cit. 2010-11-21]. Webzdarma.cz - web, e-mail a databáze ... zdarma. Dostupné z WWW: <http://www.webzdarma.cz/>.

[11] *Slunečnice.cz* [online]. 2010 [cit. 2010-11-06]. Nuclear Remote Control 1.6. Dostupné z WWW: <http://www.slunecnice.cz/sw/nuclear-remote-control/>.

[12] *Neuber software* [online]. 2010 [cit. 2010-11-06]. Network Security Task Manager. Dostupné z WWW: <http://www.neuber.com/network-taskmanager/index.html>.

[13] Microsoft. *Microsoft TechNet* [online]. 29.9.2010 [cit. 2010-11-22]. Process Monitor. Dostupné z WWW: <http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx>.

[14] Microsoft. *Microsoft TechNet* [online]. 16.11.2010 [cit. 2010-11-22]. Process Explorer. Dostupné z WWW: <http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx>.

[15] *SourceForge.net* [online]. 22.11.2010 [cit. 2010-11-22]. Process Hacker. Dostupné z WWW: <http://processhacker.sourceforge.net/index.php>.

[16] *About.com* [online]. 2010 [cit. 2010-11-22]. Linux / Unix Command: top. Dostupné z WWW: <http://linux.about.com/od/commands/l/blcmdl1_top.htm>.

[17] *SourceForge.net* [online]. 2010 [cit. 2010-11-22]. Htop - an interactive process viewer for Linux. Dostupné z WWW: <http://htop.sourceforge.net/>.

[18] *About.com* [online]. 2010 [cit. 2010-11-29]. Linux / Unix Command: ps. Dostupné z WWW: <http://linux.about.com/od/commands/l/blcmdl1_ps.htm>.

[19] *About.com* [online]. 2010 [cit. 2010-11-29]. Linux / Unix Command: pstree. Dostupné z WWW: <http://linux.about.com/library/cmd/blcmdl1_pstree.htm>.

[20] *KDE UserBase* [online]. 2010-10-5 [cit. 2010-11-29]. KSysGuard. Dostupné z WWW: <http://userbase.kde.org/KSysGuard>.

[21] *Docs.kde.org* [online]. 2010 [cit. 2010-11-29]. The System Monitor Handbook. Dostupné z WWW: <http://docs.kde.org/stable/en/kdebase-workspace/ksysguard/index.html>.

[22] *Freshmeat* [online]. 2010 [cit. 2010-11-29]. Gnome System Monitor . Dostupné z WWW: <http://freshmeat.net/projects/gnome-system-monitor/>.

[23] *Knihovna dokumentace ke GNOME* [online]. 2010 [cit. 2010-11-29]. Příručka V2.2 k aplikaci Sledování systému. Dostupné z WWW: <http://library.gnome.org/users/gnome-system-monitor/2.28/gnome-system-monitor.html>.

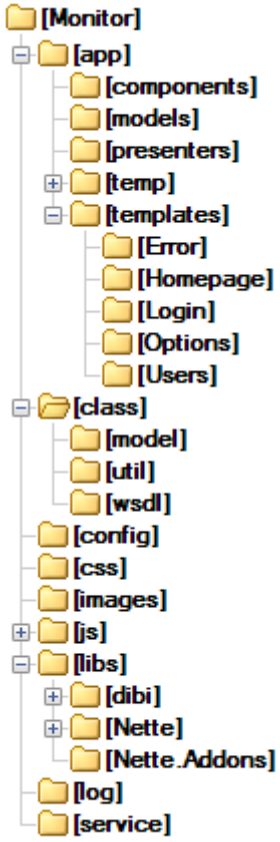
[24] *Wikipedia* [online]. 2011, last modified on 6 May 2011 [cit. 2011-05-08]. Web service. Dostupné z WWW: <http://en.wikipedia.org/wiki/Web_service>.

[25] *Dibi* [online]. 2011 [cit. 2011-05-11]. Dostupné z WWW: <http://dibiphp.com/cs/>.

[26] *Nette Framework* [online]. 2011 [cit. 2011-05-11]. Dostupné z WWW: <http://nette.org/cs/>.

# Příloha A: Adresářová struktura serverové části

| Název | Popis |
|---|---|
| app | Hlavní část prohlížeče, vnitřní struktura vychází z Nette |
| app/components | Komponenty, obsahuje ReportViewer a ReportTree |
| app/models | Modely pouze pro prohlížeč, model uživatelů pro ověřování |
| app/presenters | Presentery, hlavní logika prohlížeče |
| app/temp | Adresář pro dočasné soubory Nette |
| app/templates | Šablony stránek prohlížeče |
| class | PHP třídy společné pro prohlížeč a webovou službu |
| class/model | PHP třídy meta modelu |
| class/util | PHP třídy obsahující pomocné funkce |
| class/wsdl | PHP třídy pro objekty přenášené webovou službou |
| config | PHP soubory konfigurace, databáze a nastavení aplikace |
| css | Kaskádové styly použité v prohlížeči |
| images | Obrázky použité v prohlížeči |
| js | Javascripty použité v prohlížeči |
| libs | Využívané knihovny |
| libs/dibi | Dibi, tenká vrstva pro přístup k databázi |
| libs/Nette | Nette framework |
| libs/Nette.Addons | Pluginy Nette frameworku, stránkování |
| logs | Logovací adresář pro Nette framework |
| service | PHP třídy pouze pro webovou službu, wsdl schéma |


[Monitor]
[app]
[components]
[models]
[presenters]
[temp]
[templates]
[Error]
[Homepage]
[Login]
[Options]
[Users]
[class]
[model]
[util]
[wsdl]
[config]
[css]
[images]
[js]
[libs]
[dibi]
[Nette]
[Nette.Addons]
[log]
[service]

# Příloha B: Postup instalace serverové části

Server vyžaduje pro svůj běh libovolný webový server podporující jazyk PHP verze 5.3 nebo vyšší. Dále je vyžadován databázový server MySQL verze 5.1 nebo vyšší. Pokud není webový nebo databázový server k dispozici, je možné nainstalovat balík XAMPP, který oba výše zmíněné servery obsahuje. Balík XAMPP je součástí CD a je uložen v adresáři *Server/xampp*. Instalace se provede spuštěním souboru *xampp-win32-1.7.4-VC6-installer.exe*. Po instalaci je třeba pomocí aplikace *Xampp control* panel spustit Apache a MySQL server. K MySQL můžeme přistupovat pomocí aplikace PhpMyAdmin, která je dostupná přes webové rozhraní na adrese *http://localhost/phpmyadmin/*. Je doporučeno co nejdříve nastavit heslo administrátora MySQL serveru, ve výchozím nastavení není použito žádné heslo.

Poté co je webový server k dispozici, můžeme umístit zdrojové soubory serveru nacházející se v adresáři *Server/src* do adresáře přístupného přes webový server, v případě instalace balíku XAMPP můžeme využít adresář *htdocs*, který byl vytvořen při instalaci balíku. Adresář *htdocs* se nachází v podadresáři, kam byl balík XAMPP nainstalován. Před nakopírováním souborů je vhodné všechny ostatní soubory z tohoto adresáře odstranit. Pokud bylo vše uděláno správně, měl by být na adrese *http://localhost* dostupný prohlížeč.

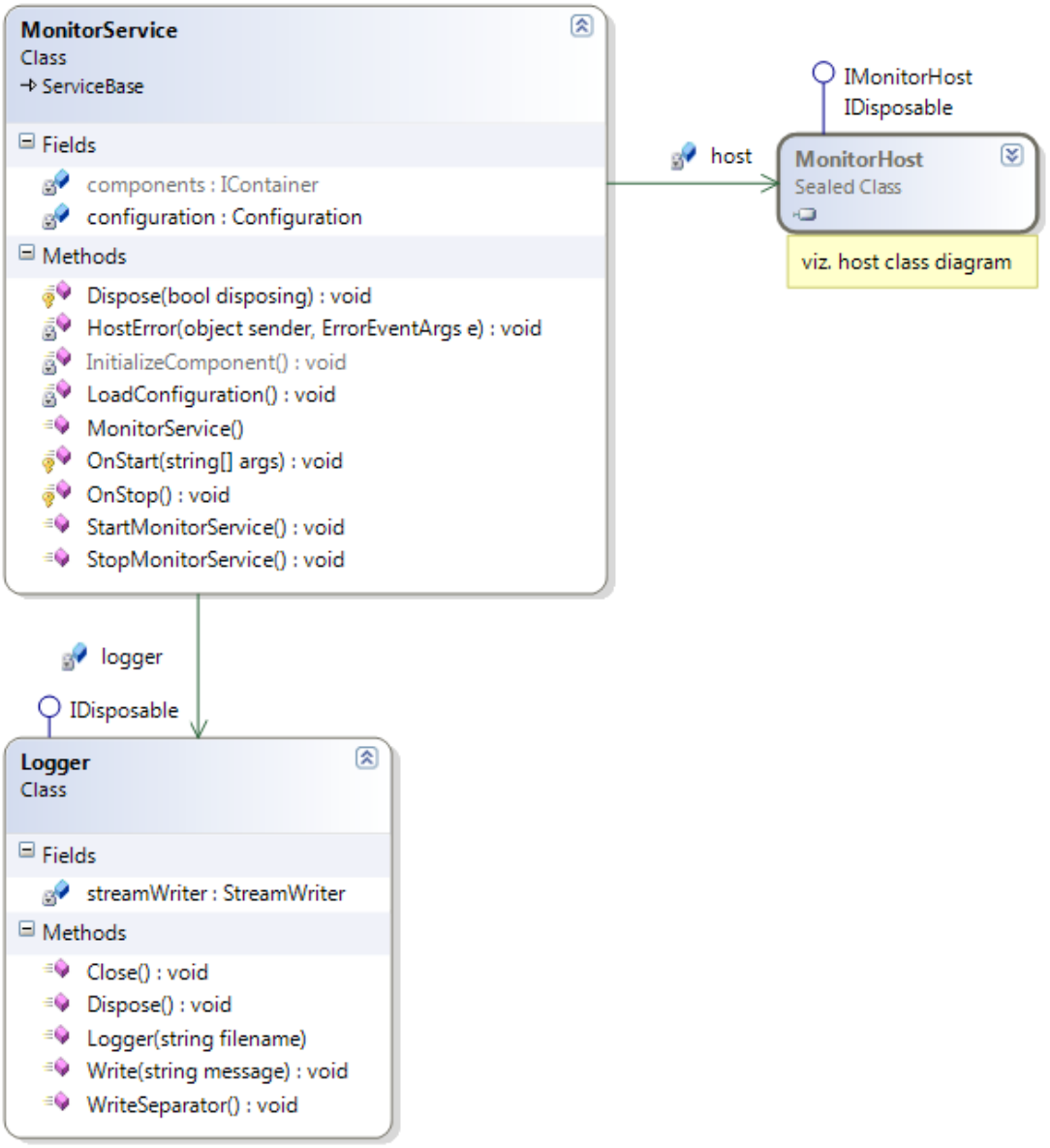
Aby byla aplikace plně funkční, je třeba vytvořit potřebné databázové objekty a nastavit připojení k databázi. K vytvoření databázových objektů slouží SQL skript *database.sql*, který se nachází v adresáři *Server/database*. Tento skript je třeba spustit na MySQL serveru například pomocí nástroje PhpMyAdmin. Pokud chceme skript spustit na již existující databází a nechceme vytvářet novou databázi, před spuštěním odebereme první 4 řádky skriptu. Po vytvoření databáze je třeba nastavit připojení k databázi, to lze provést úpravou souboru *config/database.php* (adresář *config* nacházející se v adresáři *htdocs*). V souboru nastavíme jména databáze, jméno uživatele MySQL, heslo tohoto uživatele a název nebo IP adresu počítače na kterém se nachází MySQL server. Pokud byl instalován balík XAMPP a bylo ponecháno výchozí jméno databáze, stačí v tomto souboru nastavit heslo. Pokud bylo vše provedeno správně, je možné přihlásit se do aplikace pomocí uživatelského jména

*admin* a hesla *admin*. Po přihlášení je doporučeno vytvořit vlastního uživatele a uživatele *admin* odstranit.

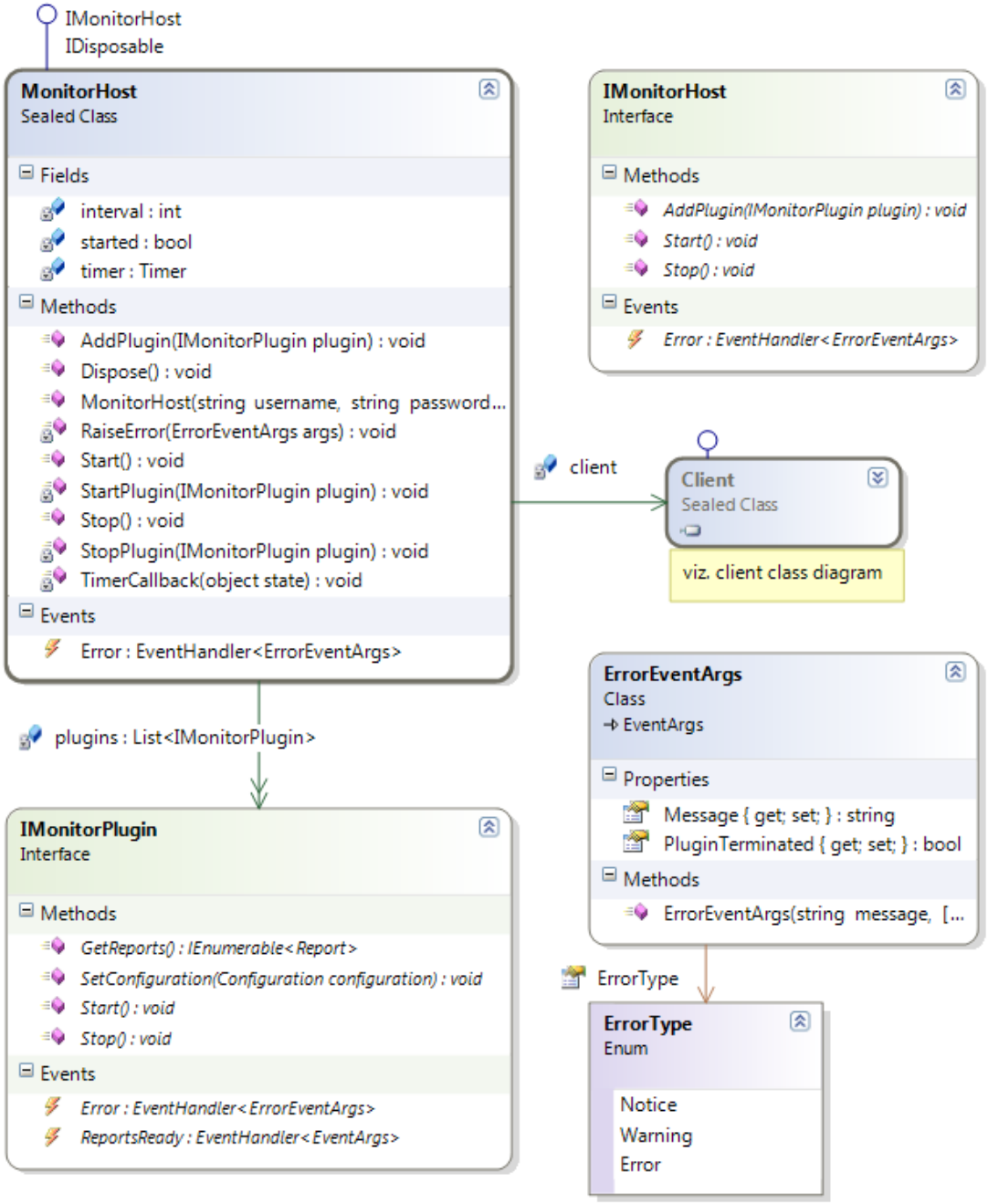
Dále je možné pomocí konfiguračního souboru *config/config.php* změnit formát výpisu data a času, počet dní, po kterých se mají reporty automaticky mazat a počet řádků na jednu stránku při výpisu reportu. Aby fungovalo automatické mazání starých reportů, je třeba zabezpečit automatické spouštění skriptu *cron_daily.php* jednou denně. Toho lze docílit pomocí nástroje *cron* nebo *Task Scheduler* (*Naplánované úlohy*).

# Příloha C: Diagramy tříd klienské části

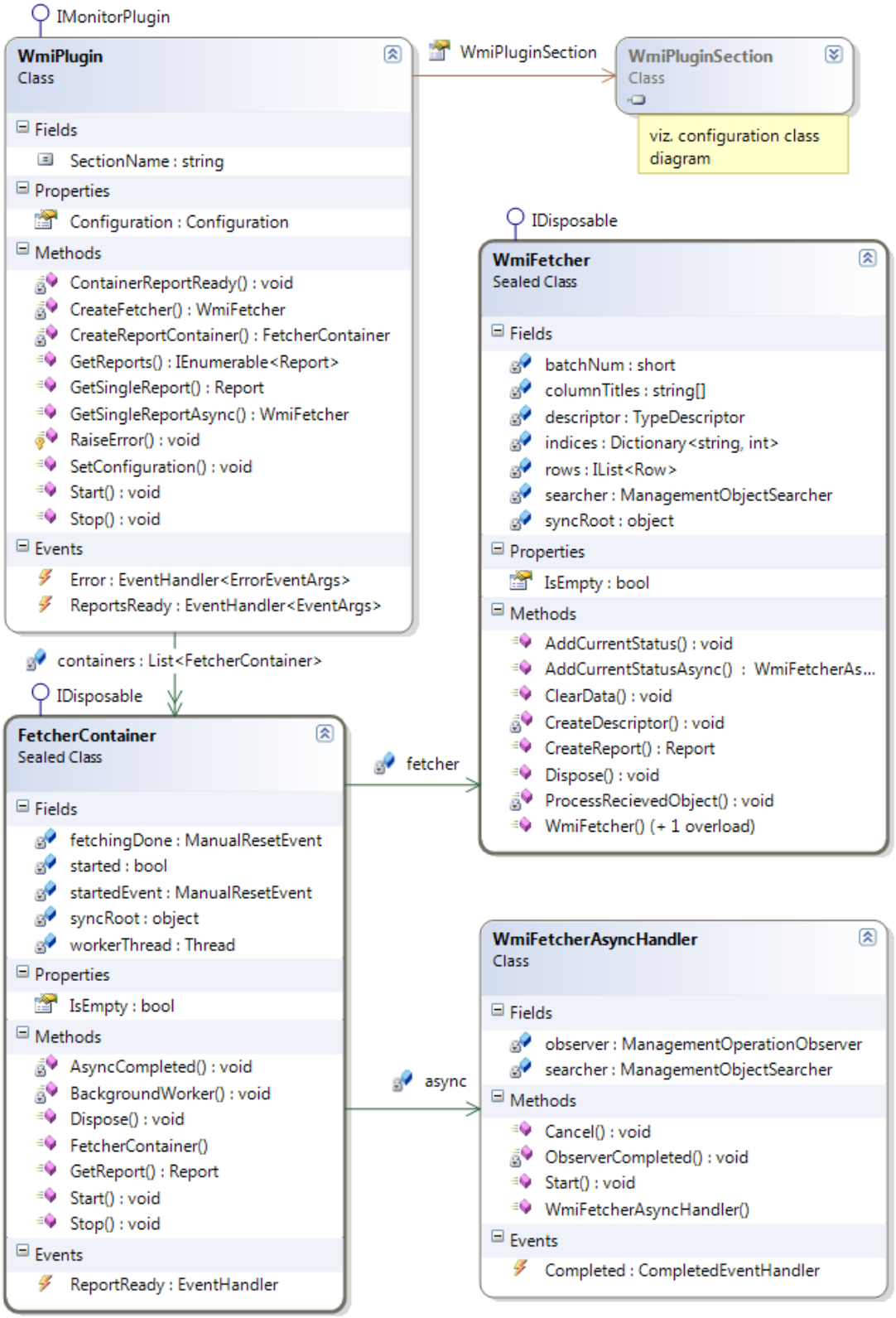
**Service class diagram**



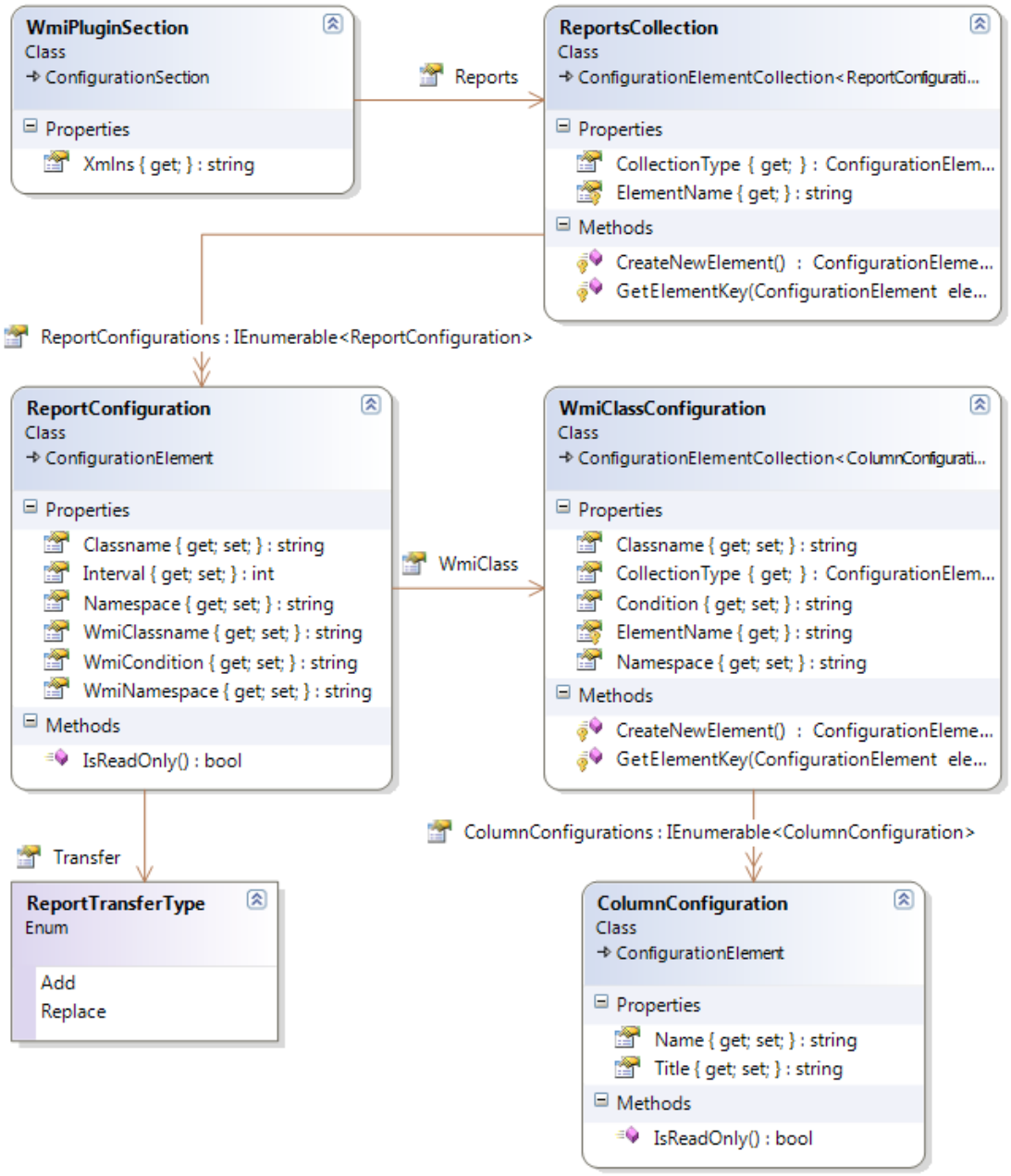
## Host class diagram



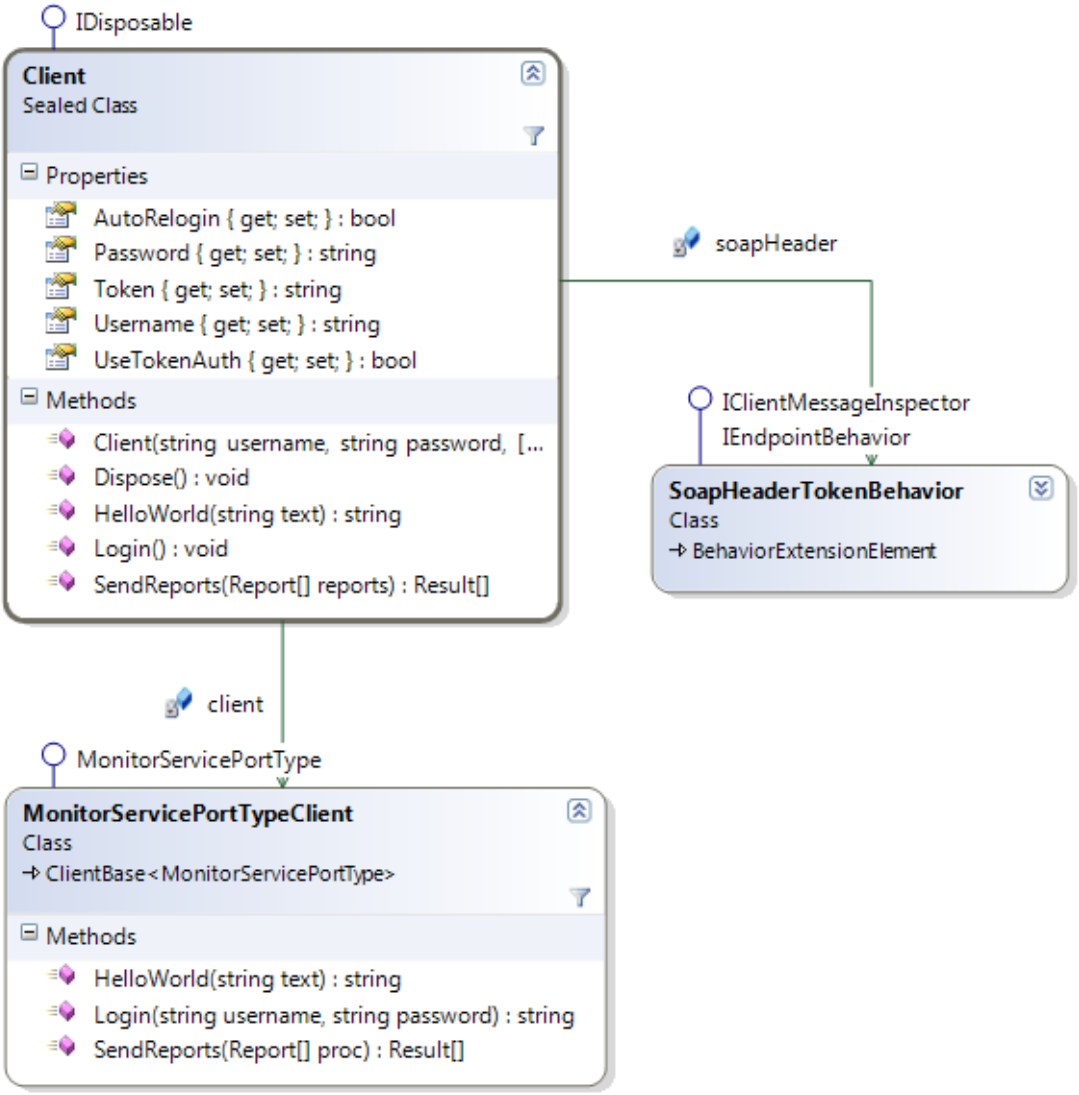
**WmiPlugin class diagram**



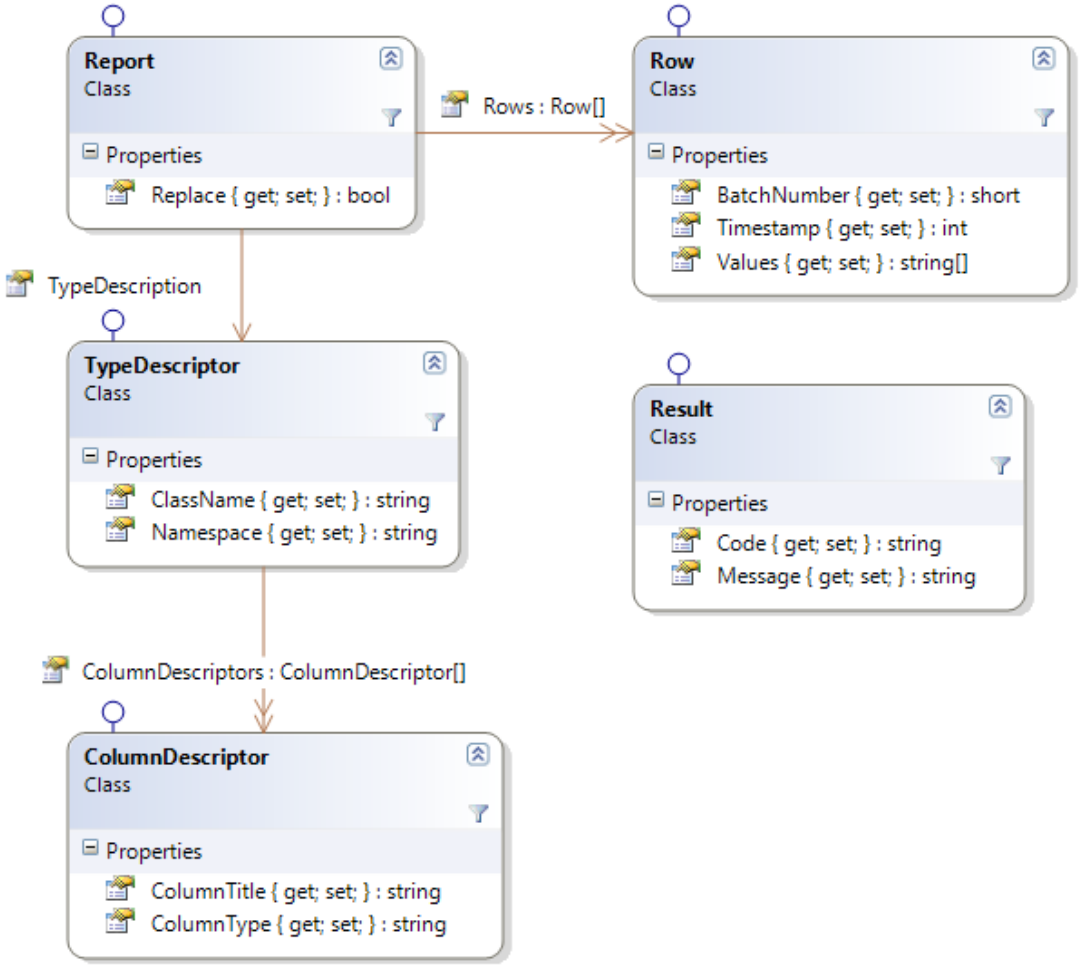
## Configuration class diagram



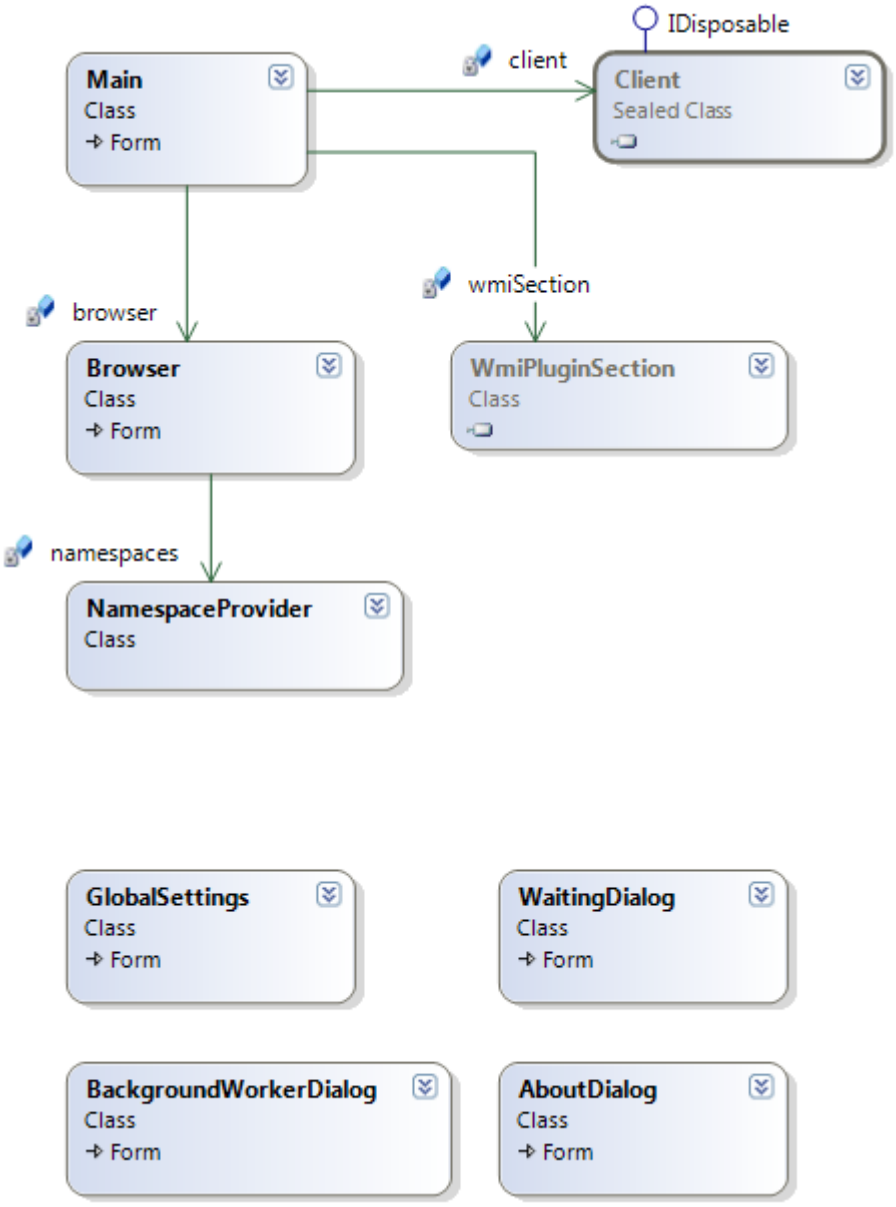
**Client class diagram – control**



**Client class diagram – data**

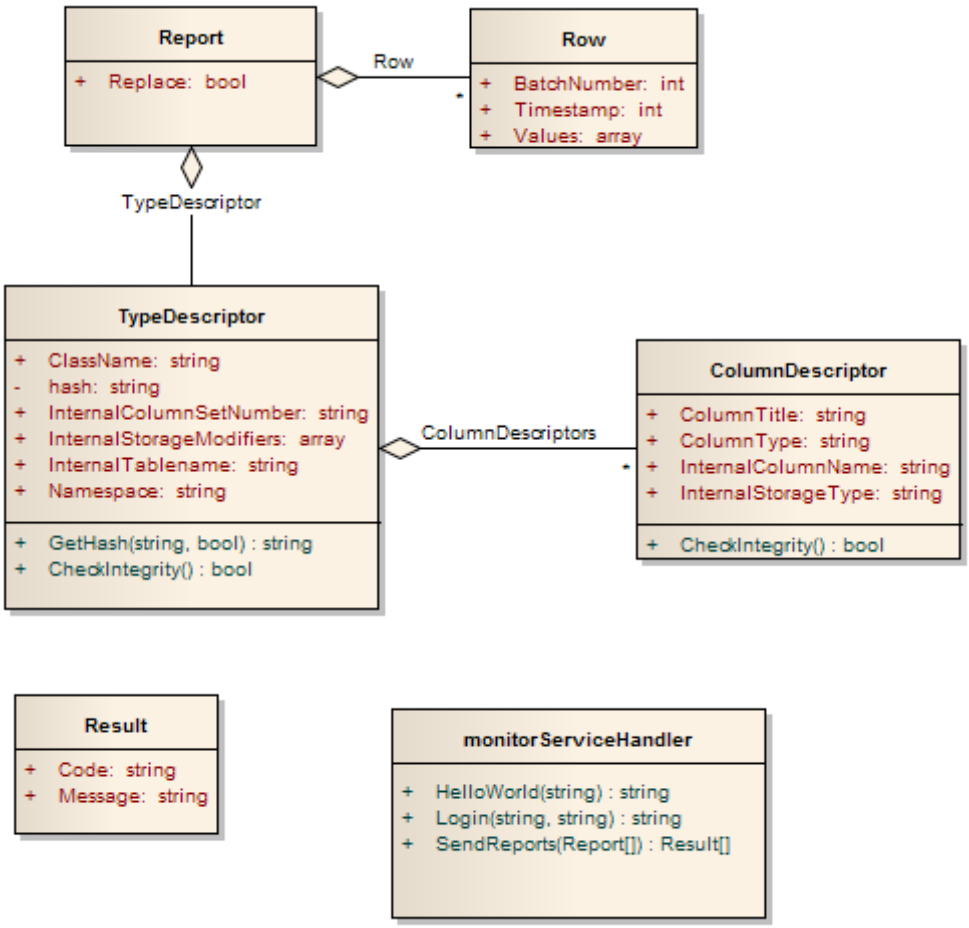


**WmiPluginGui class diagram**

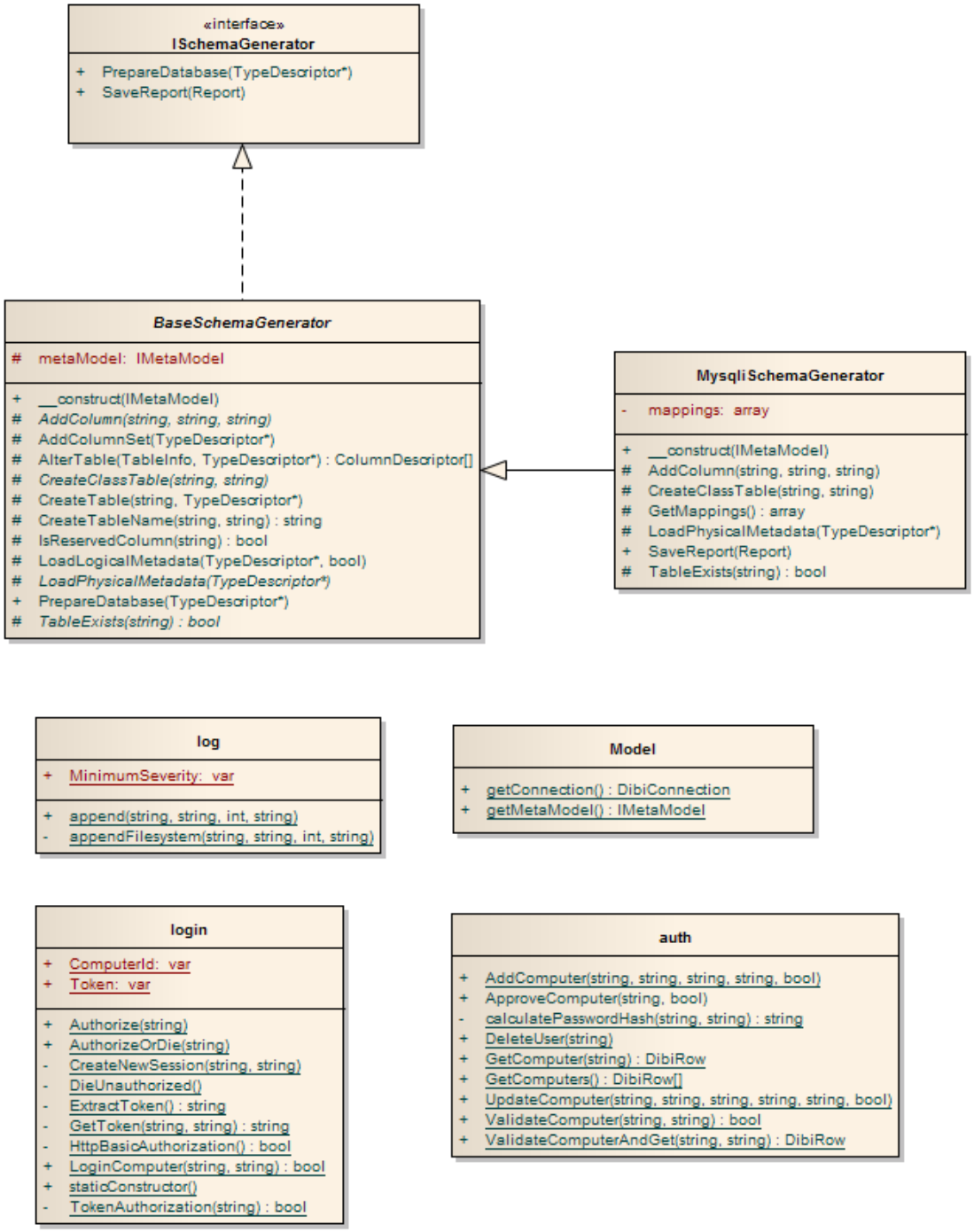


# Příloha D: Diagramy tříd serverové části

**Wsdl class diagram**



## Model class diagram



## MetaModel class diagram