



Středoškolská technika 2016

Setkání a prezentace prací středoškolských studentů na ČVUT

Ovládací panel XAMPP

Michal Žůrek

Střední průmyslová škola elektrotechniky a informatiky, Ostrava
Kratochvílova 7/1490, 702 00 Ostrava - Moravská Ostrava

Úvod

V okamžiku kdy se někdo rozhodne a odhodlá k tomu, že se naučí tvořit webová řešení nebo vytvářet databáze či zprovoznit si vlastní FTP server. Potřebuje si potřebné servery (webový, databázový, emailový, FTP, popř. další) stáhnout a nainstalovat. Aby se však nemusel starat o správu několika serverů (přičemž se ještě každý z nich ovládá úplně jinak), tak se většinou stahují kompletní balíky, které se o vše postarají. Jedním z nejpoužívanějších takových balíčků je XAMPP [1] (Cross-platform Apache MySQL PHP Perl). XAMPP je balíkem pěti nástrojů, které jsou velmi používané a známé. S popularitou těchto nástrojů roste i popularita XAMPP a to nejen mezi pokročilými uživateli, kterým usnadňuje práci, ale i mezi začátečníky, pro které je to „stravitelnější sousto“ na začátek než ruční instalace několika serverů a jejich následná, mnohdy velmi komplikovaná konfigurace.

1. Současný ovládací panel

1.1 Server

1.1.1 Co je to server?

Slovem server se označuje buďto počítač nebo program. Proto je obecná definice [2] rozdělena na dvě části spojené spojkou nebo. Pokud serverem označujeme počítač, pak serverem rozumíme počítač, který poskytuje služby jiným zařízením v síti. Pokud tuto vlastnost splní tablet nebo smartphone (což je možné), tak i tento tablet nebo smartphone můžeme prohlásit za server. Pokud serverem rozumíme počítačový program, tak máme na mysli program, který je schopen pracovat se sítí a komunikovat s klienty v takové síti.

Obecně platí, že by mohl být počítač (hardware) serverem, tak na něm musí běžet nějaká aplikace serveru (software).

Pod pojmem *server* v této práci budu označovat server jako software (počítačový program).

1.1.2 Webový server

Webový server je jeden z nejznámějších serverů. Umožňuje zpracování webových požadavků. Nejčastěji se takový server používá pro vytváření dynamických webových stránek. Mezi nejznámější webové servery patří Apache, IIS a Tomcat.

Webové servery lze rozlišit podle toho jaké skriptovací jazyky dokáží volat. Například skriptovací a programovací jazyk PHP se dá spustit na serverech Apache a IIS. Java zase pak například na serveru Tomcat.

Balík XAMPP v sobě obsahuje Apache a Tomcat. IIS je proprietární server, vyvíjený společností Microsoft. Tento server nemůže být do balíku nasazen jednak z licenčních důvodů serveru IIS a jednak z důvodu, že XAMPP má být multiplatformní, což tento server nemůže být.

1.1.3 Databázový server

Databázový server je server, který umožňuje komunikaci mezi databází a databázovým klientem. Nejznámějšími databázovými servery jsou MySQL, Microsoft SQL Server a Oracle Database. V balíku XAMPP je zastoupena databáze MariaDb [3] což je klon fork (odnož) databáze MySQL. MariaDb je však plně kompatibilní s MySQL. Databáze od Microsoftu a Oraculu do balíku nemohou být zařazené z důvodů licenčních omezení těchto databází.

1.1.4 FTP Server

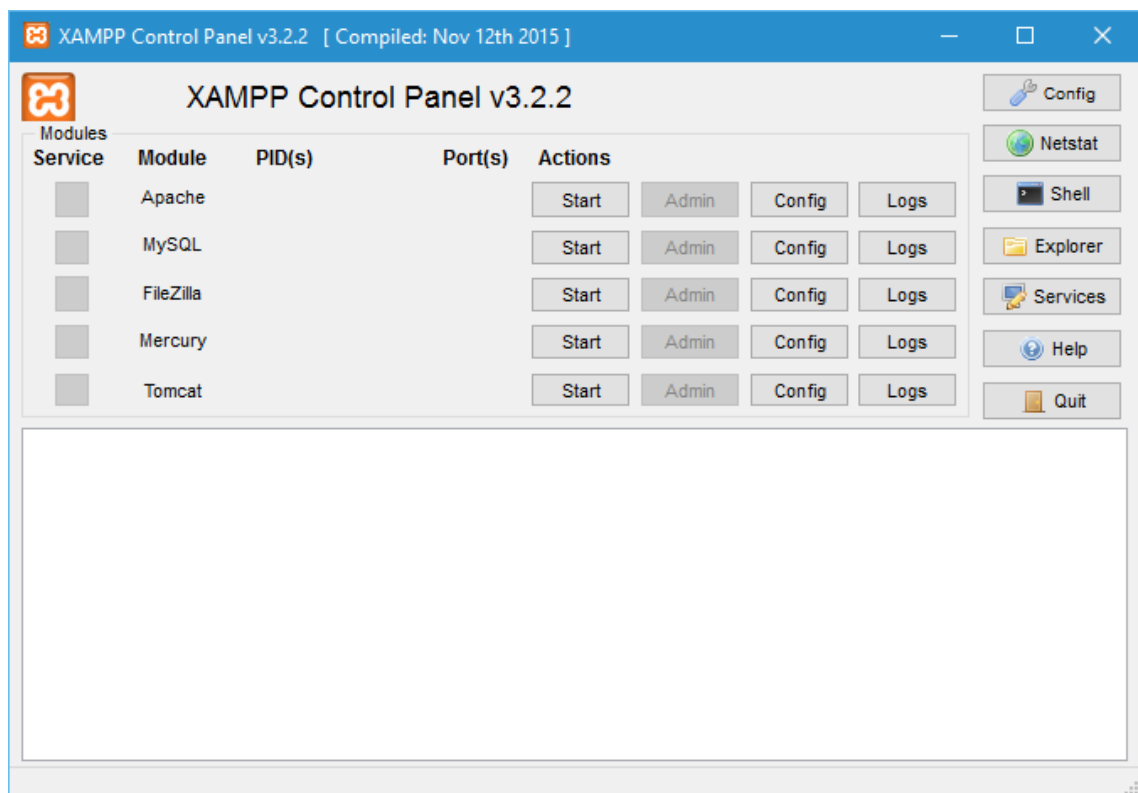
FTP [4] (File Transfer Protocol) je server, který umožní přistupovat vzdáleně k souborům na (pojmem server se teď výjimečně myslí počítač) serveru. V balíku XAMPP je zastoupen FileZilla Server

1.1.5 Emailový server

Emailové servery se rozlišují na server pro příchozí poštu a server pro odchozí poštu. V balíku XAMPP jsou oba zastoupeny aplikací Mercury Mail Transport System.

1.2 Dodávaný ovládací panel

Aby se všechny nástroje dali ovládat jednoduše, tak balík XAMPP obsahuje ovládací panel, který s každým z těchto serverů umí provádět ty stejné základní úkony, jako je například spustit a zastavit. Ovládací panel je ukázaný na obrázku 1.



Obrázek 1 Dodávaný ovládací panel XAMPP (na obrázku ve verzi 3.2.2)

1.2.1 Problémy současného ovládacího panelu

Omezenost schopností současného ovládacího panelu vychází již z principu, na jakém výchozí ovládací panel funguje. Současný ovládací panel má pro všechny úlohy, které se servery provádí, vytvořené dávkové soubory (jedná se o soubory s příponou *.bat). Tyto soubory ovládací panel spouští a do konzole panelu (to co je ve spodní části

okna) vypíše výstup těchto skriptů. Nevýhodou dávkových souborů však je, že nejsou schopné získat přesný výstup z aplikací, které spouští. Tento výstup pak obvykle obsahuje detailní popis vzniklé chyby. Jedinou věc, kterou jsou schopny získat je návratový kód, a ten je buď 0 (u serveru MariaDB se toleruje ještě kód -1, který je vrácený při ukončení procesu zabitím), značící že se server spustil správně a nedošlo k chybě nebo jakékoliv jiné číslo, značící že došlo k chybě. Na základě toho ovládací panel buď vypíše, že se server úspěšně spustil nebo vypíše svou univerzální chybu, z které pak uživatel může tipovat, k jaké chybě přesně došlo. Proces ovládání pomocí dávkových souborů ukazuje následující diagram.

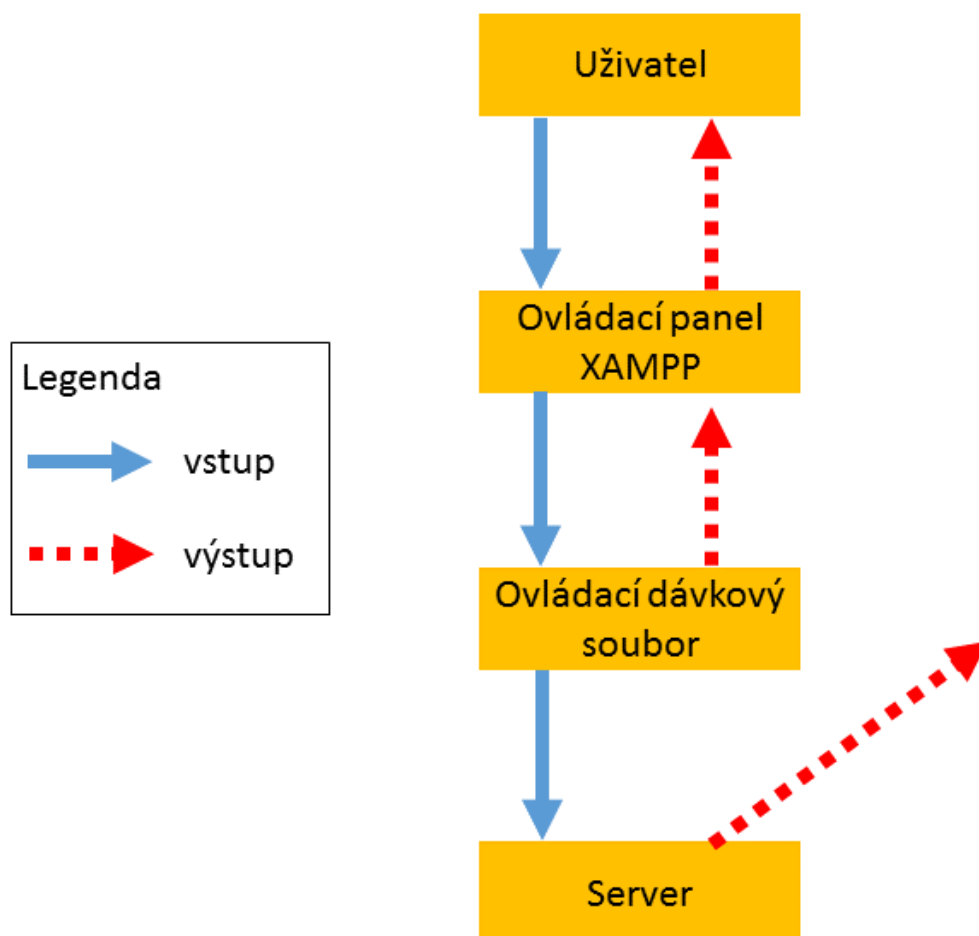


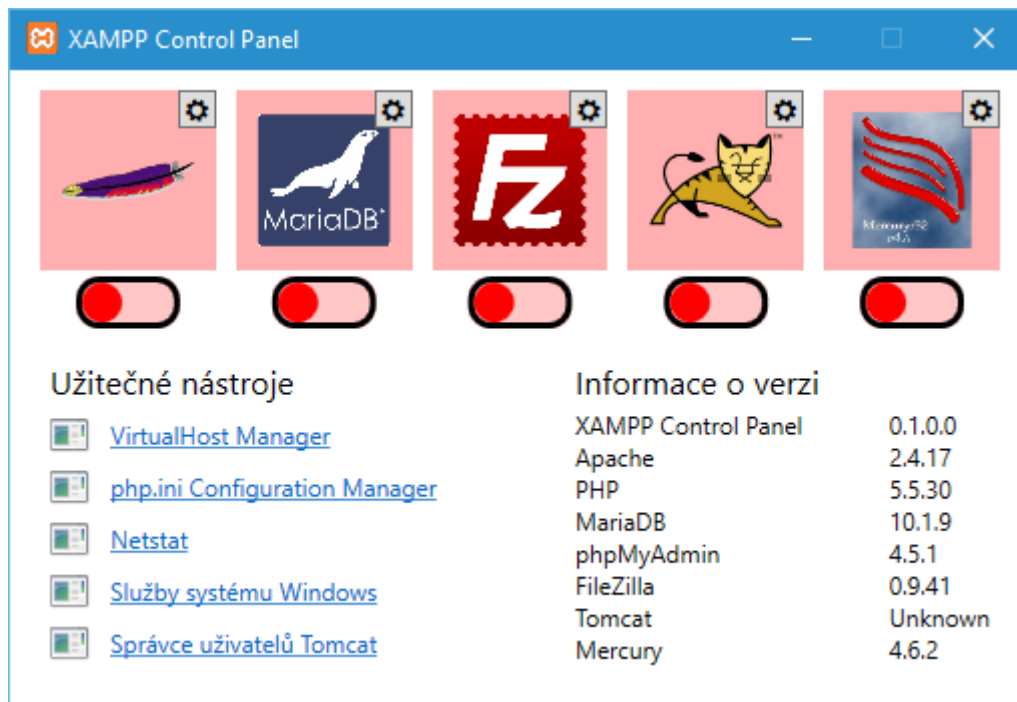
Diagram 1 Postup provádění akcí nad servery

Druhou nevýhodou ovládacího panelu je nedomulárnost jeho doplňkových součástí. V jeho pravé části se nacházejí tlačítka pro různé aplikace a ty buď spouštějí nějaké systémové části, nebo otvírají okna přímo v aplikaci, čímž však tyto zakomponované

podaplikace nejde spouštět samostatně a při například drobné aktualizaci těchto nástrojů se musí vydat aktualizace pro celý ovládací panel.

2 Nový ovládací panel XAMPP

Jelikož všechny problémy jsou s ovládacím panelem úzce spojeny a nelze je nějak přímo vyřešit, rozhodl jsem se, že vytvořím nový ovládací panel. Tento nový panel pak může nahradit ten původní, který je dodáváný v rámci balíku XAMPP. Nový balík jsem označil XAMPP control panel 2. Vzhledem k tomu, že balík byl vytvořen zcela znova, nebylo od věci nejen přeimplementovat celý ovládací panel a vyřešit neduhy předchozího, ale i při té příležitosti ho vylepšit. Jednou ze značných možností na jeho rozšíření nabízel samotný pravý panel, do kterého byli zakomponovaný tlačítka na další rozšířené (většinou systémové) nástroje. Do tohoto panelu, jsem pak přidal několik užitečných nástrojů, které některé běžné činnosti značně usnadňují. Protože se obsah panelu rozrůstal a pravá oblast (a ani další oblasti okna původního ovládacího panelu) není nekonečná, bylo třeba panel značně jinak uspořádat, s tím přišla úvaha, co vše je v tom panelu zbytečně velké? Třeba například každý server má u sebe 4 tlačítka, z čehož pravidelně se využívá jen jedno. Kolonky pro výpis portů, na kterých server poslouchá a ID procesu serverů, jsou při vypnutém serveru prázdné (jinými slovy: zbytečně velké) a při zapnutém procesu se může stát (třeba u serveru Mercury) že naslouchají na tolika portech, že se tam ta informace nevejde a musí být zalomena na dva řádky. Při odstraňování hlavního problému došlo i k tomu, že když ovládací panel nebude využívat dávkových souborů, tak i celé výstupní pole s výpisem výstupu těchto dávkových souborů je zcela zbytečné. Nepatrným problémem je (a týká se převážně začátečníků řešících problémy), že ovládací panel nikde nevypisuje verzi jednotlivých nástrojů. Některé verze ovládacího panelu sice vypisují verzi PHP v záhlaví okna, ale to je trochu málo od nástroje, který v sobě zabaluje několik serverů a několik interpretů. Navíc i verze PHP je do okna natvrdo vkompilevaná. Pokud tedy si ručně zaktualizují PHP, ovládací panel bude v záhlaví obsahovat pořád číslo staré verze a změnu nezaznamená. Nebylo ani od věci vytvořit nový design ovládacímu panelu. Nový ovládací panel vypadá jako na následujícím obrázku.



Obrázek 2 Nový ovládací panel XAMPP

2.1 Použité technologie

K vývoji byly použity 2 programovací jazyky, Visual Basic a C#. Samotná aplikace je psaná v jazyce Visual Basic. V C# je napsaná podpůrná knihovna, která zapouzdřuje některé operace, které jsou dostupné ve Windows API a práce s nimi by byla ve Visual Basicu komplikovaná a zdrojový kód nepřehledný. Visual Basic a C# jsou jazyky vyvíjené společností Microsoft, která zastřešuje i jejich nástroje. Oba jazyky se kompilují do mezikódu CIL [5], někdy označovaného MSIL. Díky této společné vlastnosti lze kód mezi 2 programovacími jazyky kombinovat způsobem, že část projektu je psaná ve Visual Basicu a část v C# popřípadě ještě v jiných jazycích. Aplikace běží nad knihovnou .NET Framework, konkrétně ve verzi 4.6. K tvorbě formulářů, byl využit Framework WPF (Windows Presentation Foundation). Samotný vývoj probíhal ve vývojovém prostředí Visual Studio 2015 Community. Pomocné aplikace, které byly v původním ovládacím panelu zabudované, a bylo je potřeba modularizovat, jsou napsané ve Visual Basicu. Pomocná aplikace netstat využívá systémové knihovny IpHlpApi [6], jež je implementovaná v stejnojmenné DLL knihovně, která je dodávána se systémem Windows. Řešení je rozděleno do 7 projektů, projekty jsou popsány v tabulce 1 a jednotlivých podkapitolách.

Název projektu	Programovací jazyk	Typ aplikace
Netstat	VB.NET	Doplňková
ServiceManager	VB.NET	Pomocná
Services.msc	VB.NET	Doplňková
TomcatUserManager	VB.NET	Doplňková
VirtualhostManager	VB.NET	Doplňková
WpfWindowTools	C#	Knihovna
XAMPPC2	VB.NET	Hlavní

Tabulka 1 Projekty řešení

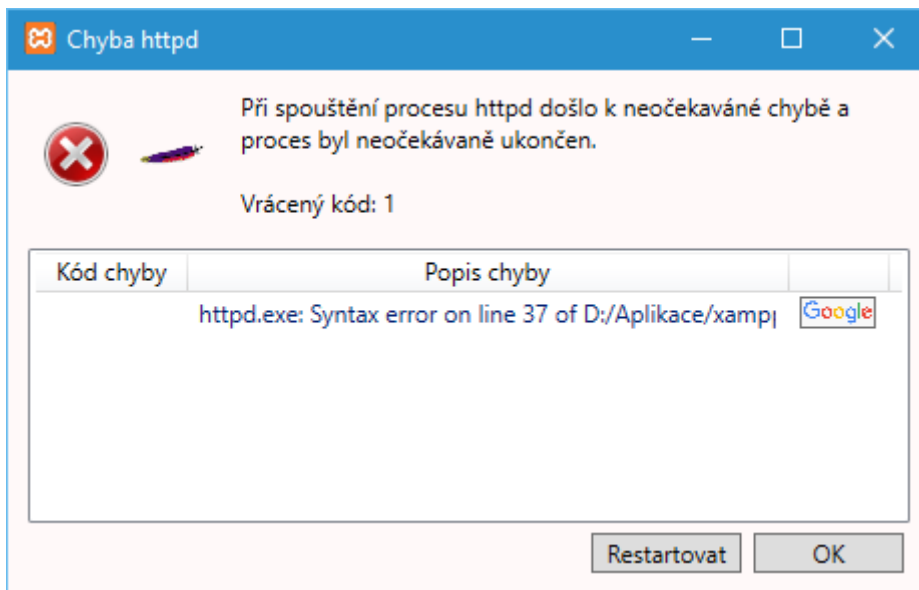
2.2 Čtení stavu serveru

Aby aplikace mohla zobrazovat aktuální stav serveru, musí pravidelně kontrolovat, co dělá server, jestli je spuštěný, jestli poslouchá nebo nějaký proces serveru skočil a při případném pádu serveru přečíst zprávu, kterou server vrátil (to je přesně to, co současný ovládací panel nedělá). Aplikace tedy po spuštění v pravidelném intervalu o délce 1 sekunda kontroluje, zdali se neobjevili nějaké nové procesy serveru (aplikace reaguje na spuštění serveru i když je spuštěn externě) či zdali nějaký server neskončil. Krom tohoto časovače ještě existuje časovač, který kontroluje, zdali server poslouchá na nějakém portu, pokud ano, vrací tuto skutečnost do tooltipu ovládacího tlačítka serveru.

Pokud nějaký proces serveru skončil, aplikace kontroluje, jaký stavový kód s ukončením server vrátil, pokud kód označuje, že k chybě v serveru nedošlo a byl ukončen standartní cestou uživatelem, aplikace server jednoduše označí za zastavený, pokud však vrátí libovolný jiný kód, aplikace přečte výstup serveru a zobrazí chybové dialogové okno informující, že v serveru došlo k chybě a popis konkrétních chyb, ke kterým v aplikaci došlo přesně podle toho, co server vrátí.

2.3 Chybové dialogové okno

Toto okno je voláno, když server vrátí chybový návratový kód, v takovém případě jednotlivý správci serverů (bloky v programu, které se servery programově pracují) zpracují výpis chyb a předají je dialogovému oknu. Při otevření dialogového okna si okno zkontroluje, zdali ho servery nezavolali v okamžiku, kdy server sice vrátil chybový stavový kód, ale nevrátil žádný popis chyby, čili okno nemá co zobrazit. V takovém případě se okno samo zavře. Okno obsahuje kód chyby (pokud ho služba nastavuje), popis chyby a tlačítko Google, které vyhledá popis chyby ve vyhledávači Google. Okno je ukázáno na obrázku 3.



Obrázek 3 Chybové okno nového ovládacího panelu

2.4 Ovládání serverů

Ovládání serverů bylo přeneseno do větších tlačítek. Barva pozadí tlačítka značí jeho stav, rozlišují se tři stavy.

- a) Server je zastaven (červená) – tento stav signalizuje červené pozadí a znamená, že momentálně neběží žádný proces serveru ani není spuštěna jeho služba Windows.
- b) Server běží, ale na žádném portu neposlouchá (žlutá) – stav signalizuje, že existuje proces serveru nebo běží jeho služba, ale tento proces neposlouchá na žádném TCP portu. K tomuto stavu obvykle dochází, když se server spouští.
- c) Server běží a poslouchá (zelená) – stav signalizuje, že existuje proces serveru nebo je spuštěna jeho služba a proces poslouchá na nějakém TCP portu.

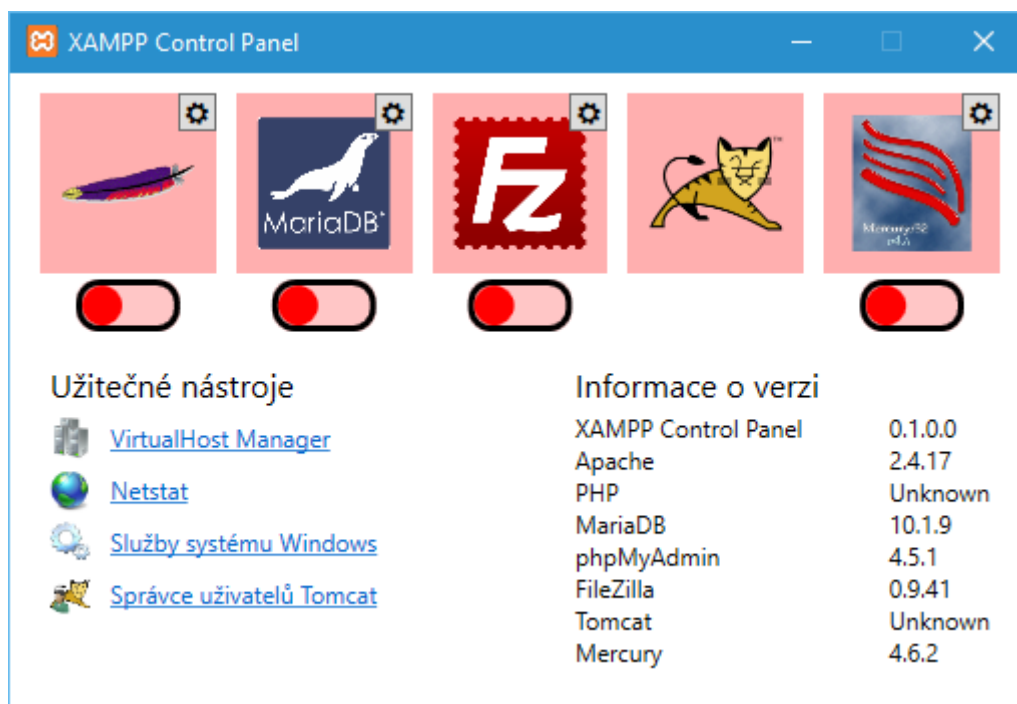
Grafické znázornění stavů serveru je zobrazeno na obrázku 4.



Obrázek 4 Vizuální zvýraznění stavu serveru

2.4.1 Nenainstalovaný server

Pokud server není v XAMPP nainstalován, nezobrazí se přepínací tlačítko. Protože server může být samozřejmě nainstalován mimo XAMPP, ovládací panel zobrazuje, jestli je server spuštěn a naslouchá na nějakém portu. Protože však server není součástí ovládacího balíku XAMPP, takto externě nainstalovaný server nelze ovládat. Příklad jak vypadá nenainstalovaný server, ukazuje obrázek 5.

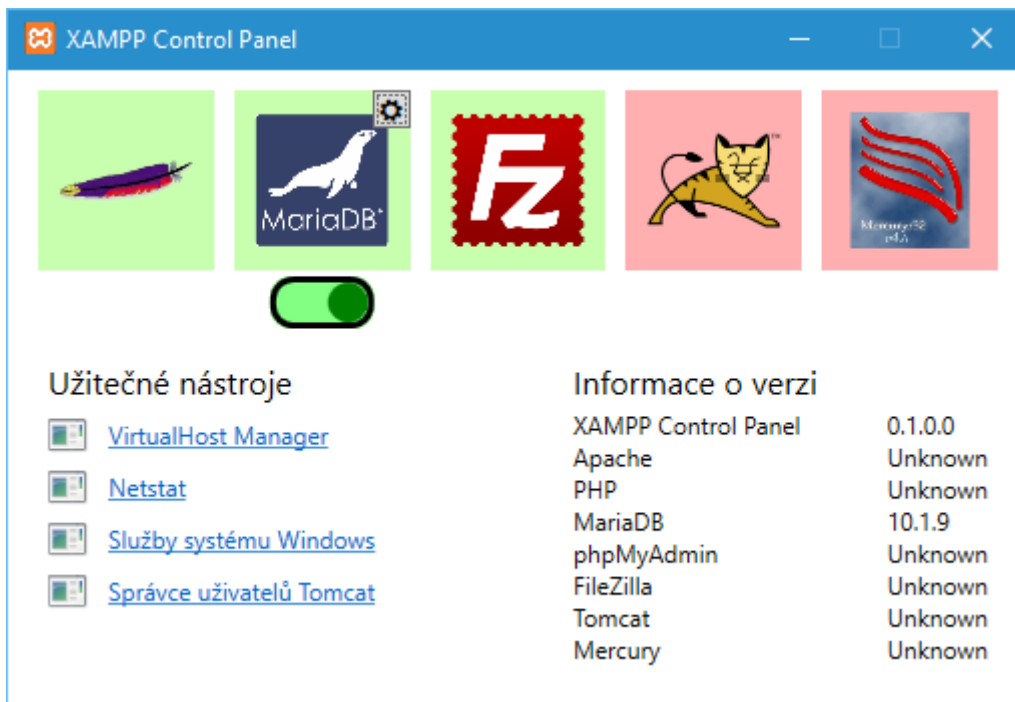


Obrázek 5 Ukázka chování ovládacího panelu pro nenainstalovaný server (zde server Tomcat)

Ovládací panel načítá procesy ze své pracovní složky. Každý server od pracovní složky musí být ještě v specificky pojmenované složce (viz. Tabulka 2), pokud jsou tyto podmínky zachovány, tak ovládací panel při spuštění z jiné složky (lze zajistit z příkazového řádku) nemá s ovládním serveru sebemenší problém. Na obrázku 6 je ukázka serveru, který byl spuštěn v složce kde je pouze server MariaDB.

Server nebo aplikace	Prohledávaná složka
Apache	apache
MariaDB	mysql
FileZilla	FileZillaFTP
Tomcat	tomcat
Mercury	MercuryMail
PHP	php
phpMyAdmin	phpMyAdmin

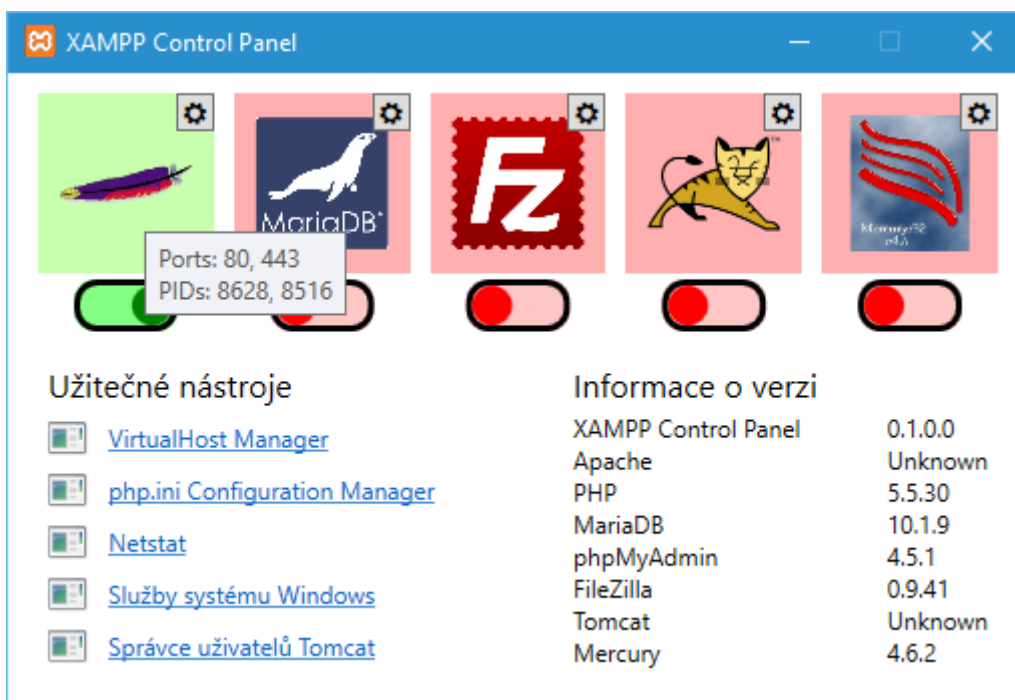
Tabulka 2 Složky, ve kterých hledá ovládací panel aplikace



Obrázek 6 Ovládací panel spuštěný v jiné pracovní složce

2.4.2 PID serveru a obsazené porty

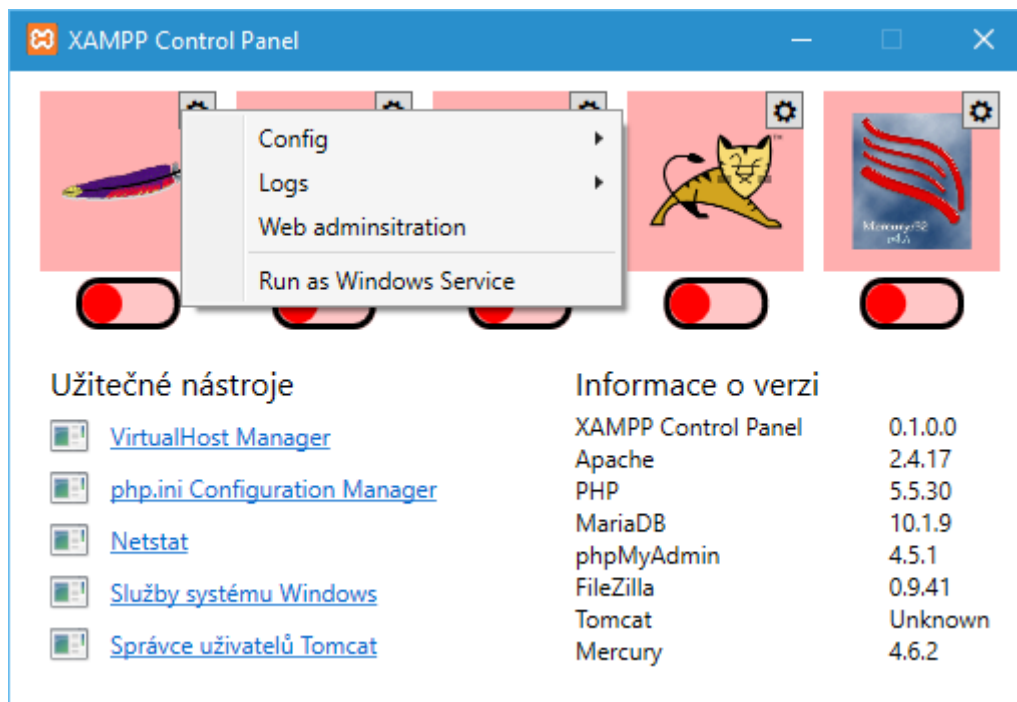
Informace o ID procesu spuštěného serveru a TCP porty, na kterých server poslouchá, se v novém ovládacím panelu zobrazí v podobě tooltipu na tlačítku. Tooltip je ukázán na obrázku 7.



Obrázek 7 Tooltip obsahující informace o běžícím serveru

2.4.3 Konfigurace serveru, logy, administrace

Konfigurace serverů, jejich logy, odkaz do administrace a možnost pouštět server jako službu se přesunuli do kontextové nabídky, kterou lze vyvolat buďto pravým kliknutím na tlačítko serveru nebo levým tlačítkem myši na symbol ozubeného kola v pravém rohu tlačítka serveru. Pokud server některou z možností nepodporuje (například Mercury nelze spouštět jako službu) je tato možnost zašedlá a nelze ji použít.



Obrázek 8 Kontextové menu s možnostmi serveru

2.5 Informace o verzi

Informace o verzi jsou vypsány v pravé spodní části obrazovky a jsou dynamicky získávány z jednotlivých serverů a její návazných částí. Získávání dat probíhá po spuštění aplikace asynchronně v nových vláknech. Obtíže při implementaci této funkce spočívali v tom, že neexistuje jednotný a zároveň rozšířený způsob jak aplikace může předat číslo své verze ven, třetím stranám. Proto nelze použít obecnou implementaci pro všechny technologie, ale pro každou je implementace nepatrně odlišná. Jako příklad se zkusme podívat na výpis verze serveru Apache a interpretu PHP. Oba programy shodně vyžadují parametr `-v`, ale výstup je zcela odlišný a co do algoritmického zpracování absolutně nepoužitelný. Zdrojový kód 1 ukazuje výstup procesu `httpd -v` a Zdrojový kód 2 obsahuje výstup procesu `php -v`. V obou ukázkách je tučně zvýrazněna hledaná informace.

```
Server version: Apache/2.4.17 (Win32)
Apache Lounge VC11 Server built: Oct 13 2015 10:54:13
Zdrojový kód 1 Výstup volání programu httpd -v
```

```
PHP 5.5.30 (cli) (built: Sep 30 2015 13:50:53)
Copyright (c) 1997-2015 The PHP Group
Zend Engine v2.5.0, Copyright (c) 1998-2015 Zend Technologies
Zdrojový kód 2 Výstup volání programu php -v
```

Některé aplikace dokonce nemají možnost získat verzi přes příkazový řádek a verze se musí zjišťovat jinou cestou. Například verze serveru FileZilla FTP se zjišťuje pomocí atributu verze u jeho spustitelného souboru. Rekordmanem je však phpMyAdmin (což je webová aplikace pro ovládání databáze), kde neexistuje jiná možnost, než naparsovat verzi z jeho konfiguračního souboru. Což je asi úplně nejhorší možné řešení, protože v okamžiku vydání nepatrně upravené verze konfiguračního souboru obsahující danou verzi se musí upravit i skript hledající označení verze.

Programově je funkce naimplementovaná způsobem, že formulář si asynchronně vyžaduje verzi v podobě textového řetězce od příslušných tříd, které implementují rozhraní IVersionMiner a musí tak implementovat metodu GetVersion.

2.6 Instalace serverů jako služeb Windows

Nový ovládací panel zachová možnost instalaci serverů jako služby Windows. Pouze servery, které tuto funkci nativně nepodporují (Mercury) mají tuto možnost zakázanou. Tuto akci lze vyvolat z kontextové nabídky serveru. Protože tato akce vyžaduje zvýšená oprávnění, tuto akci zajišťuje vedlejší pomocná aplikace, která pro svůj běh vyžaduje zvýšená oprávnění. Díky této aplikaci samotný ovládací panel nevyžaduje při spuštění zvýšená oprávnění a mohou jej používat uživatelé bez těchto práv. Externí aplikace se jmenuje ServiceStarter.exe. Službu lze volat 2 způsoby. První způsob vyžaduje 3 parametry příkazového řádku popsanými v tabulce 3. Druhý způsob vyžaduje 4 parametry popsané v tabulce 4. První způsob umožňuje spouštět a zastavovat službu, druhý způsob se používá k instalování a odinstalování služby.

Název parametru	Dostupné hodnoty	Popis
akce	start nebo stop	Hodnota „start“ vyvolá spuštění služby Hodnota „stop“ vyvolá zastavení služby
Jméno služby	Textová hodnota	Jméno spouštěné nebo zastavované služby
Cesta k ikoně	Textová hodnota	Cesta k obrázku, který se použije k vizualizaci probíhající úlohy

Tabulka 3 Parametry příkazového řádku aplikace ServiceStarter pro spouštění/zastavování služby Windows

Název parametru	Dostupné hodnoty	Popis
Popisek probíhající akce	Textová hodnota	Popisek zobrazený v aplikaci informující uživatele o akci probíhající na pozadí
Cesta k ikoně	Textová hodnota	Cesta k obrázku, který se použije k vizualizaci probíhající úlohy
Cesta programu	Textová hodnota	Cesta k programu, který bude na pozadí spuštěn a aplikace bude vizualizovat jeho běh
Argument programu	Textová hodnota	Argument příkazového řádku, který bude předán spouštěné aplikaci

Tabulka 4 Parametry příkazového řádku aplikace ServiceStarter pro spuštění externího programu s určitými parametry příkazového řádku

3 Doplnkové aplikace

Nový ovládací panel obsahuje v levé dolní části odkazy na aplikace, které se k správě serveru hodí zejména pokročilejším uživatelům.

3.1 Služby systému Windows

Jednoduchá utilita zapouzdřující spuštění systémového ovládacího panelu (mmc.exe) a v něm otevření systémové utility services.msc.

3.1.1 Technická implementace

Metoda Main aplikace obsahuje jeden jediný řádek zdrojového kódu, který volá statickou metodu Start objektu Process a předává jí cesty k ovládací konzoli (mmc.exe) a spouštěnému objektu (services.msc). Zdrojový kód je ukázán v ukázce Zdrojový kód 3.

```
Process.Start("C:\Windows\system32\mmc.exe",
"C:\Windows\System32\services.msc")
```

Zdrojový kód 3 Zdrojový kód utility services.exe

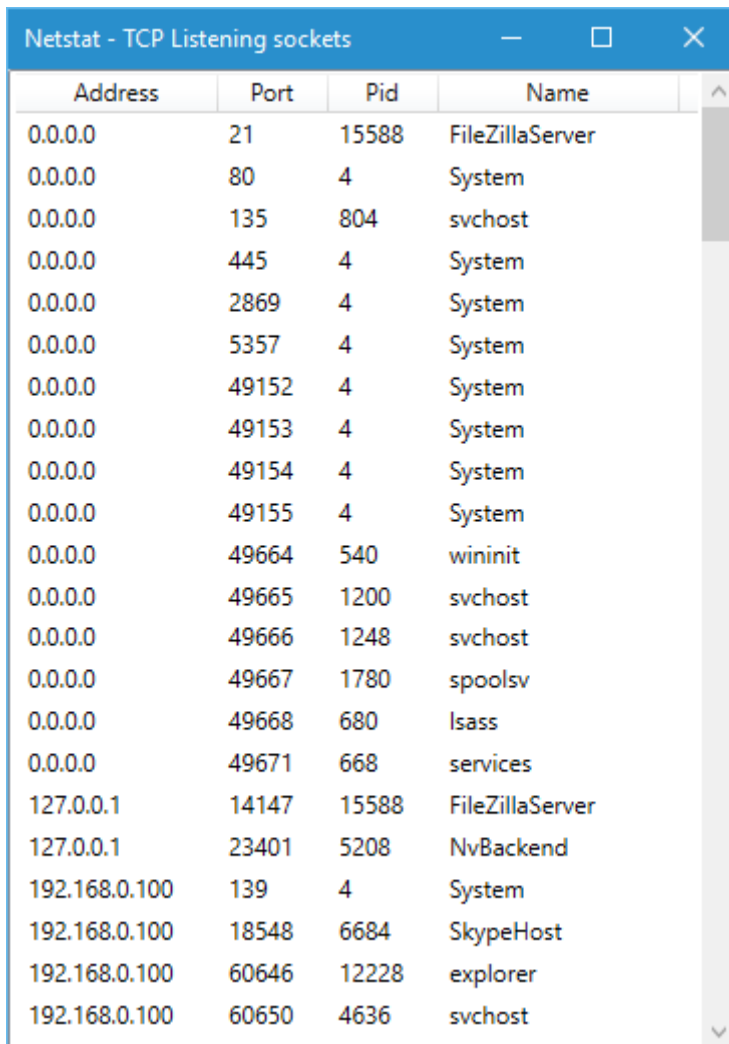
3.2 Netstat

Aplikace netstat slouží k zobrazení informace, které aplikací naslouchají na kterém TCP portů. Aplikace se hodí, zejména když je port nějakého serveru blokován jinou aplikací. Aplikace zvýrazňuje nově vzniklé poslouchání (zeleně) a zaniklá naslouchání (červeně). Obrázek 9 obsahuje screenshot z aplikace.

3.2.1 Technická implementace

Technicky je aplikace implementována voláním Windows API funkce GetExtendedTcpTable [7] z knihovny IpHlpApi.lib. Výstupem této funkce je binární uspořádaní jednotlivých řádků navrácené TCP tabulky v strukturách [8][9] známých z jazyka C. Tato binární data jsou aplikací naparsována a z nich „vytěženy“ další hodnoty, které jsou dále zpracovány do uživatelsky přívětivé a srozumitelné podoby. Takto

„očesané“ data jsou pak prezentovány v grafickém uživatelském rozhraní okna aplikace. Po dokončení této akce aplikace 750 milisekund čeká a následně akci spustí znova. Po každé aktualizaci dat, pak dojde k zvýraznění nově vzniklých naslouchání a doplnění zvýrazněných zaniklých naslouchání.



Address	Port	Pid	Name
0.0.0.0	21	15588	FileZillaServer
0.0.0.0	80	4	System
0.0.0.0	135	804	svchost
0.0.0.0	445	4	System
0.0.0.0	2869	4	System
0.0.0.0	5357	4	System
0.0.0.0	49152	4	System
0.0.0.0	49153	4	System
0.0.0.0	49154	4	System
0.0.0.0	49155	4	System
0.0.0.0	49664	540	wininit
0.0.0.0	49665	1200	svchost
0.0.0.0	49666	1248	svchost
0.0.0.0	49667	1780	spoolsv
0.0.0.0	49668	680	lsass
0.0.0.0	49671	668	services
127.0.0.1	14147	15588	FileZillaServer
127.0.0.1	23401	5208	NvBackend
192.168.0.100	139	4	System
192.168.0.100	18548	6684	SkypeHost
192.168.0.100	60646	12228	explorer
192.168.0.100	60650	4636	svchost

Obrázek 9 Aplikace Netstat

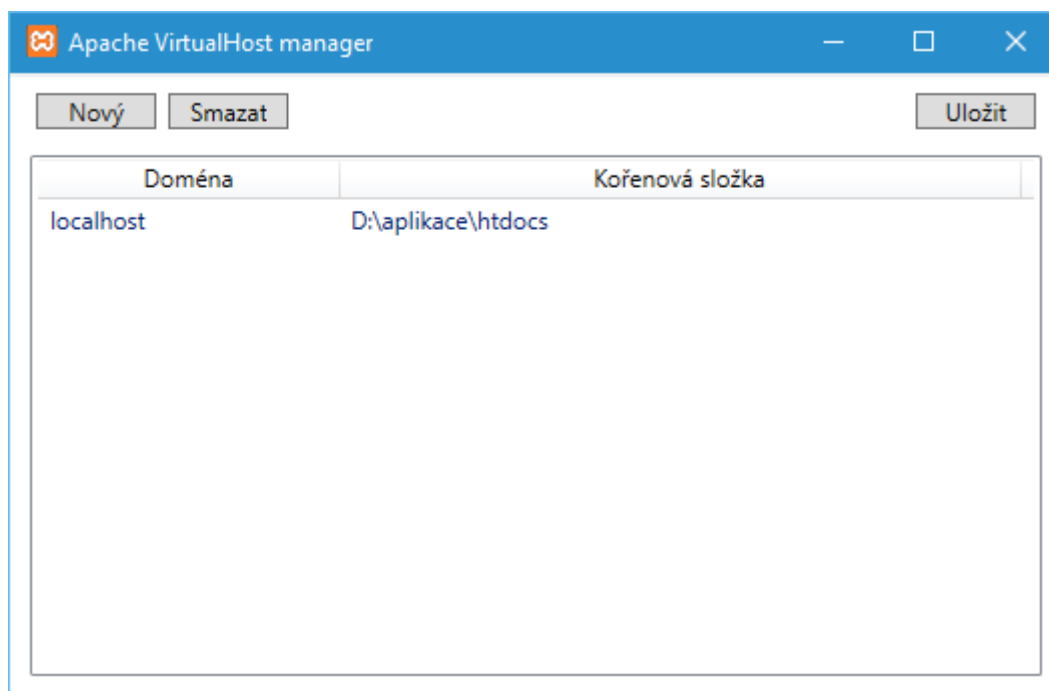
3.3 Apache VirtualHost manager

Poměrně populární praktikou serveru Apache je zavádění konfiguračních bloků VirtualHost, které umožňují, aby server Apache mohl zpracovávat více virtuálních serverů. Krom konfigurace serveru se ještě obvykle nastavuje lokální DNS záznam aby virtuální servery mohli fungovat na vlastních doménách. Oba tyto kroky aplikace zabaluje do své funkce. Protože soubor obsahující konfiguraci lokálních domén je uložen ve složce kam normální uživatel nemá přístup, aplikace při spuštění vyžaduje zvýšení oprávnění.

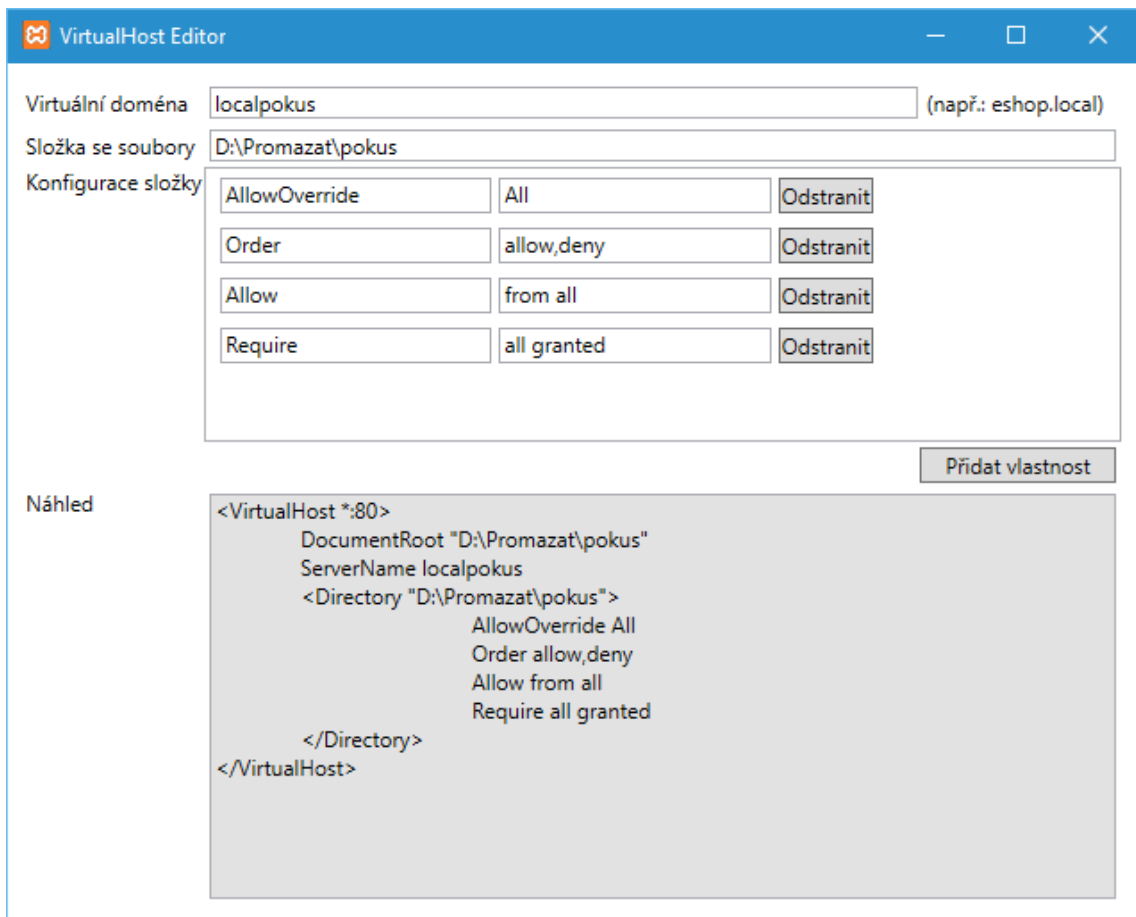
Obrázek 10 obsahuje screenshot aplikace a obrázek 11 obsahuje screenshot editoru samotného konfiguračního bloku.

3.3.1 Technická implementace

Aplikace při spuštění načte konfigurační soubor obsahující konfigurace VirtualHost, naparsuje jednotlivé bloky a seznam existujících konfigurací VirtualHost zobrazí v grafickém uživatelském rozhraní aplikace. Aplikace zachová i jiné části konfigurace, které načítá také, ale nikterak do nich nezasahuje. Díky tomuto načítání i dalších bloků se i tyto ostatní bloky dostanou do výstupu a konfigurační soubor tak nepříjde například o komentáře. Uživatel může jednotlivé záznamy konfigurací VirtualHost mazat, upravovat či přidávat nové. Uživatel může do konfiguračního souboru zapsat změny. Se zapsáním změn dojde i k automatické aktualizaci souboru obsahující informace o doménách. Načítání a zpracovávání konfigurace lokálních domén funguje na podobném principu jako načítání konfigurací VirtualHost. I zde aplikace zachází šetrně s komentáři.



Obrázek 10 Apache VirtualHost manager



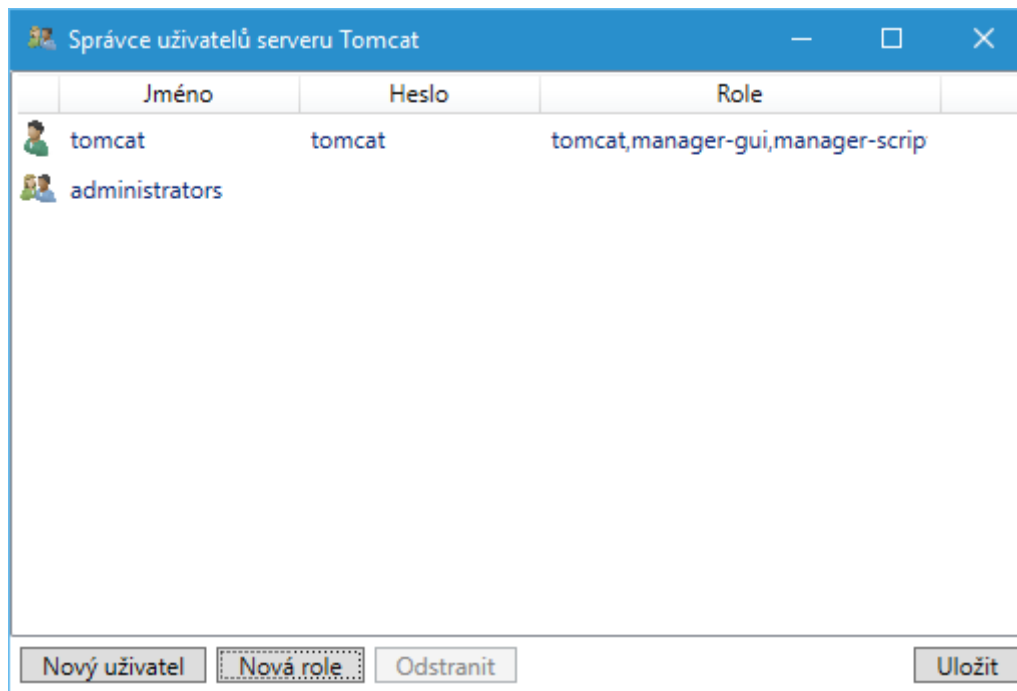
Obrázek 11 Editor bloku konfigurace VirtualHost

3.4 Správce uživatelů Tomcat

Server Tomcat slouží k provozování webových aplikací napsaných v jazyku Java. V tomto jazyce dělají obvykle komplexní, často korporátní, webové služby, které ke svému webu vyžadují správu uživatelských účtů. Některé výchozí uživatelské účty jsou pak zapsány v souboru tomcat-users.xml, což je XML soubor kde jsou v XML značkách zapsáni jak uživatelé, tak i zároveň skupiny do kterých mohou být uživatele přiřazeni. Poměrně zajímavý je fakt, že hesla jsou v tomto souboru zapsána v čistém textu, což z bezpečnostního hlediska není správné. Aplikace však toto pravidlo respektuje. Aplikace umožňuje upravovat, přidávat a odstraňovat uživatele a skupiny. Obrázek 12 obsahuje screenshot aplikace.

3.4.1 Technická implementace

Aplikace využívá nástrojů .NET Frameworku pro načítání XML Souborů do interní datové struktury. Každý objekt této struktury si s sebou nese i odkaz do XML. V konfiguračním souboru jsou uvnitř kořenového elementu zachovány i XML komentáře.



Obrázek 12 Správce uživatelů Tomcat

Závěr

Vytvořil jsem nový ovládací panel, který řeší zásadní nedostatky dosavadního ovládacího panelu, obsahuje nové nástroje, usnadňuje práci. Celý program je včetně zdrojového kódu zveřejněn pod licencí MIT na githubu pod URL <https://github.com/misaz/XAMPPC2>. Na jeho vývoji budu dále spolupracovat, stejně jako se díky otevřenému zdrojovému kódu může připojit každý.

Zdroje

- [1] XAMPP. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-03-06]. Dostupné z: <https://en.wikipedia.org/wiki/XAMPP>
- [2] Server. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-03-06]. Dostupné z: <https://cs.wikipedia.org/wiki/Server>
- [3] MariaDB. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-03-06]. Dostupné z: <https://cs.wikipedia.org/wiki/MariaDB>
- [4] File Transfer Protocol. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-03-06]. Dostupné z: https://cs.wikipedia.org/wiki/File_Transfer_Protocol
- [5] Common Intermediate Language. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-03-06]. Dostupné z: https://cs.wikipedia.org/wiki/Common_Intermediate_Language
- [6] IP Helper. Windows Dev Center [online]. Redmond (WA): Microsoft, 2016 [cit. 2016-03-06]. Dostupné z: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa366073\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa366073(v=vs.85).aspx)
- [7] GetExtendedTcpTable function. Windows Dev Center [online]. Redmond (WA): Microsoft, 2016 [cit. 2016-03-06]. Dostupné z: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa365928\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa365928(v=vs.85).aspx)
- [8] MIB_TCPTABLE_OWNER_PID structure. Windows Dev Center [online]. Redmond (WA): Microsoft, 2016 [cit. 2016-03-06]. Dostupné z: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa366921\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa366921(v=vs.85).aspx)
- [9] MIB_TCPROW_OWNER_PID structure. Windows Dev Center [online]. Redmond (WA): Microsoft, 2016 [cit. 2016-03-06]. Dostupné z: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa366913\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa366913(v=vs.85).aspx)