

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Automatická konfigurace VirtualBoxu v laboratořích KIV**

Plzeň, 2012

Milan Jirků

## PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 23. července 2012

Milan Jirků

## PODĚKOVÁNÍ

Tímto bych chtěl poděkovat všem, kteří při mně stáli během tvorby této bakalářské práce, a poskytovali mi tak nejlepší zázemí, jaké si člověk může přát. Zároveň bych chtěl velmi poděkovat svému vedoucímu bakalářské práce Ing. Ladislavu Pešičkovi za jeho velikou trpělivost, pomoc a postrčení vždy správným směrem. Poslední, kterým bych rád poděkoval jmenovitě za pomoc, jsou pánové Milan Třešňák a Robert Novák.

## ANOTACE

Bakalářská práce se zabývá technologií virtualizace a jejím využitím v počítačové učebně na Katedře informační a výpočetní techniky na Západočeské univerzitě v Plzni. Teoretická část zkoumá dostupné programy, kterými lze virtualizaci provádět, a uvádí příklady dělení na skupiny podle jejího využití. Dále je v teoretické části řešena otázka kopírování souborů o velkém objemu dat a změna přístupových práv souborů v rámci operačního systému Windows. Výstupem praktické části je potom program, který umožňuje distribuovat tyto soubory v rámci učeben KIV a importovat je do programu Oracle VM VirtualBox na základě uživatelem zvolených nastavení.

## KLÍČOVÁ SLOVA

virtualizace, virtualizační programy, virtualizační technologie, historie virtualizace, virtuální stroj, distribuce image, virtuální disk, Oracle VM VirtualBox, VboxManage

## ANNOTATION

This graduation bachelor work describes the virtualization technology and its practical use in computer classroom at Department of computer science and engineering at University of West Bohemia Plzeň. Theoretical part analysis existing virtualization programs and shows some cases, how they can be divided in terms of usage. Further, there are solved issues of copying high-volume data files and change of files' access rights under Windows operation system. Outcome of the practical part is a computer program, which allows these files to be distributed within the classrooms of KIV and imported to Oracle VM VirtualBox program according to user-chosen settings.

## KEY WORDS

virtualization, virtualization programmes, virtualization technologies, virtualization history, virtual machine, image distribution, virtual disk, Oracle VM VirtualBox, VboxManage

# OBSAH

|          |   |    |
|----------|---|----|
| <b>1</b> | <b>Úvod</b> .....                               | 7  |
| <b>2</b> | <b>Virtualizace</b> .....                       | 8  |
| 2.1      | Typy Virtualizace .....                         | 8  |
| 2.2      | Důvody virtualizace .....                       | 10 |
| 2.2.1    | Využití virtualizace v komerčním prostředí..... | 10 |
| 2.2.2    | Vysoká dostupnost a automatizovaná správa ..... | 12 |
| <b>3</b> | <b>Virtualizační software</b> .....             | 14 |
| 3.1      | VMware® .....                                   | 14 |
| 3.1.1    | VMware® Player .....                            | 14 |
| 3.1.2    | VMware® Workstation .....                       | 15 |
| 3.1.3    | VMware® vSphere.....                            | 16 |
| 3.1.4    | VMware® Fusion.....                             | 16 |
| 3.1.5    | VMware® View.....                               | 17 |
| 3.2      | Microsoft® .....                                | 18 |
| 3.2.1    | Microsoft® Virtual PC.....                      | 18 |
| 3.2.2    | Microsoft® Virtual Server 2005 .....            | 19 |
| 3.2.3    | Microsoft® Hyper-V .....                        | 19 |
| 3.3      | Citrix® .....                                   | 20 |
| 3.4      | Oracle® .....                                   | 21 |
| <b>4</b> | <b>VirtualBox</b> .....                         | 22 |
| 4.1      | Instalace programu .....                        | 22 |
| 4.2      | Vytváření nového virtuálního stroje .....       | 22 |
| 4.3      | Konfigurace virtuálního stroje .....            | 24 |
| 4.4      | Import/Export virtuálních strojů.....           | 26 |
| 4.5      | VBoxManage .....                                | 27 |
| <b>5</b> | <b>Praktická část</b> .....                     | 29 |
| 5.1      | Rozbor klíčových úloh programu .....            | 29 |
| 5.1.1    | Kopírování souborů o velkém objemu dat.....     | 30 |
| 5.1.2    | Změna přístupových práv souborů.....            | 31 |
| 5.1.3    | Automatizace .....                              | 32 |
| 5.2      | Vývoj programu.....                             | 33 |
| 5.2.1    | Návrh programu .....                            | 33 |
| 5.2.2    | Návrh uživatelského rozhraní.....               | 34 |
| 5.2.3    | Požadavky na řídicí stanici a učebnu .....      | 38 |
| 5.2.4    | Realizace kopírování.....                       | 38 |
| 5.2.5    | Tvorba skriptů.....                             | 41 |

|           |   |           |
|-----------|---|-----------|
| 5.2.6     | Funkce programu.....                            | 42        |
| 5.2.7     | Části programu.....                             | 43        |
| 5.2.8     | Ověření funkcionality .....                     | 45        |
| <b>6</b>  | <b>Závěr</b> .....                              | <b>46</b> |
|           | Seznam zkratk a slovník výrazů.....             | 47        |
|           | Seznam citované literatury a zdrojů.....        | 49        |
| <b>A.</b> | <b>Uživatelská příručka</b> .....               | <b>52</b> |
| A.1       | Nastavení systému .....                         | 52        |
| A.2       | Povolení WakeOnLan.....                         | 53        |
| A.3       | Manuál programu .....                           | 53        |
|           | Základní postup, jak používat program .....     | 53        |
|           | Nastavení kopírování souborů .....              | 55        |
|           | Nastavení virtuálních strojů.....               | 57        |
|           | Uložení/Nahrání nastavení.....                  | 59        |
|           | Učebny a práce s jejich nastavením.....         | 60        |
|           | Nahrání učebny .....                            | 62        |
|           | Spuštění skriptů na stanici po kopírování ..... | 62        |

# 1 ÚVOD

Virtualizace a virtualizační technologie je trendem moderní doby, který se neustále rozrůstá a získává si stále větší procento příznivců na poli informačních technologií. Tato technologie má nejrůznější využití a velký potenciál při využívání hardwarových zdrojů fyzické serverové stanice, které by jinak „ležely ladem“. První část této práce bude zaměřena na virtualizaci a její uplatnění.

Druhou částí práce je průzkum a popis dostupných virtualizačních technologií v rámci PC stanic v počítačové učebně. Zejména pak bude kladen důraz na program Oracle VM VirtualBox [1] (dále jen VirtualBox), který je využíván v praktické části této práce.

Praktickou částí, kterou si klade tato práce za úkol, je vytvoření aplikace, která bude schopná zjednodušit přípravu virtuálního prostředí v učebně pro výuku administrace serverů. Její součástí je vhodná distribuce image virtuálních disků nebo předpřipravených virtuálních strojů na jednotlivé stanice, nebo jejich duplikace z jednoho místa na druhé v rámci každé stanice. Před distribucí souborů pro virtualizaci je vygenerovaný jednoduchý skript, který bude kopírovaný na každou stanici společně se soubory pro virtualizaci.

## 2 VIRTUALIZACE

### 2.1 TYPY VIRTUALIZACE

Virtualizace je poměrně široký pojem, který se rozšířil do povědomí veřejnosti až během posledních 10 let. Samotná technologie ale sahá mnohem dále a má kořeny již v 60. letech minulého století [2].

Existuje mnoho způsobů jak přistupovat k virtualizaci. Jedním z nich je vytvoření abstraktního hardwaru, který běží nad softwarovou virtualizační platformou na operačním systému. Tento způsob se označuje jako „**softwarová virtualizace**“. Softwarová virtualizace může být prováděna na třech hlavních úrovních.

**Emulace**, neboli plná systémová simulace či plná virtualizace s dynamickým přestavením (rekompilací) nastává, když virtuální stroj simuluje kompletní hardware dovolující provoz nemodifikovaného operačního systému na úplně jiném procesoru.

**Paravirtualizace** je virtualizací, kdy virtuální stroj nesimuluje hardware, ale místo toho nabídne speciální rozhraní API, které vyžaduje modifikace hostovaného operačního systému, aby mohl být tento OS nad virtuálním strojem spuštěn.

**Nativní (plná) virtualizace** probíhá tehdy, když virtuální stroj simuluje dostatek hardwaru, aby mohl nemodifikovaný OS běžet izolovaně od hostitelského operačního systému, ale hostitelský operační systém musí být určený pro stejný druh procesoru. Pojem *nativní virtualizace* se někdy používá ke zdůraznění, že je využita hardwarová podpora pro virtualizaci [3].

Abychom mohli používat **softwarovou virtualizaci**, potřebujeme k tomu určený software kompatibilní s operačním systémem, na kterém virtualizujeme.

Druhým způsobem je **virtualizace hardwarová**. Hlavní rozdíl je v tom, že virtualizovaný operační systém běží nad softwarovou platformou přímo nad hardware bez existujícího operačního systému.

Díky skutečnosti, že na fyzickém stroji není přítomen operační systém, je zde oproti softwarové virtualizaci jednoznačná výhoda v podobě poskytování



většiny hardwarových zdrojů pouze virtuální stanicí. To má za následek růst výkonu celého virtualizovaného systému. Jelikož nemáme operační systém, je zde zastoupen software pro virtualizaci tzv. **hypervisorem** [4], který má za úkol nabídnout hardwarové zdroje virtualizovanému operačnímu systému.

Kromě virtualizace celého systému se dají také virtualizovat například pouze aplikace. Taková aplikace pak běží ve svém malém virtuálním prostředí, které obsahuje pouze komponenty nejnужnější pro její spuštění (např. proměnné prostředí, globální objekty, soubory, položky registrů). Toto virtuální prostředí se chová jako vrstva mezi operačním systémem a aplikací zabraňující jejich vzájemným konfliktům nebo konfliktům s dalšími aplikacemi. Mezi software poskytující **aplikační virtualizaci** lze pak zařadit například Java Virtual Machine (JVM), který umožňuje spuštění aplikací na vybraných platformách, pro které je dostupný, nebo také .NET Virtual Machine (NVM), který umožňuje v prostředí MS Windows vytvářet aplikace pro jiné platformy (např. mobilní zařízení).

Virtualizace operačních systémů a aplikací je však jen zlomek toho, co se dá virtualizovat. **Virtualizovat se dají například i datová úložiště**, kdy se sloučí více fyzických disků nebo pouze jejich části do virtuálního celku, který se pak tváří jako jeden disk. Počítačové sítě, kdy můžeme vytvořit přímo virtuální logické spojení bodů, které spolu fyzicky přímo nesousedí. Tam, kde nestačí operační paměť fyzického stroje, můžeme přiřadit **paměť virtuální** a rozšířit tak adresní prostor paměti o adresy větší než jsou ty u paměti fyzické.

Pro náš případ bude nejdůležitější **softwarová virtualizace desktopu**, kdy se virtualizuje celý hardware nad virtualizační platformou na operačním systému. Takový typ virtualizace, má výhodu z práce v prostředí, ve kterém když dojde k chybě a kritickému selhání systému, nedochází následně i k chybě systému, na kterém virtualizaci provádíme. Jednou z jeho výhod je možnost využívání takzvaných **snapshotů** [5], které uchovávají nastavení virtuálního stroje se všemi nastaveními systému z doby, kdy byl snapshot vytvořen. Virtuální stroj se dá pak načíst do tohoto bodu. Tento typ virtualizace je zejména vhodný k různému testování softwaru na operačním systému nebo třeba školením, kdy se stroj nevyužívá dlouhodobě, a pak může být načten do původního nastavení nebo smazán.

Virtualizace je užitečná v mnoha oblastech informatiky. Používá se při vývoji softwaru pro nové procesory, případně k vývoji softwaru obecně, na fyzických serverech, pro běh několika menších virtuálních serverů (úspora nákladů, které by jinak byly vynaložené na koupi dalšího serveru), atd. Více důvodů, proč přejít na virtualizaci, bude uvedeno v následující kapitole.

## 2.2 DŮVODY VIRTUALIZACE

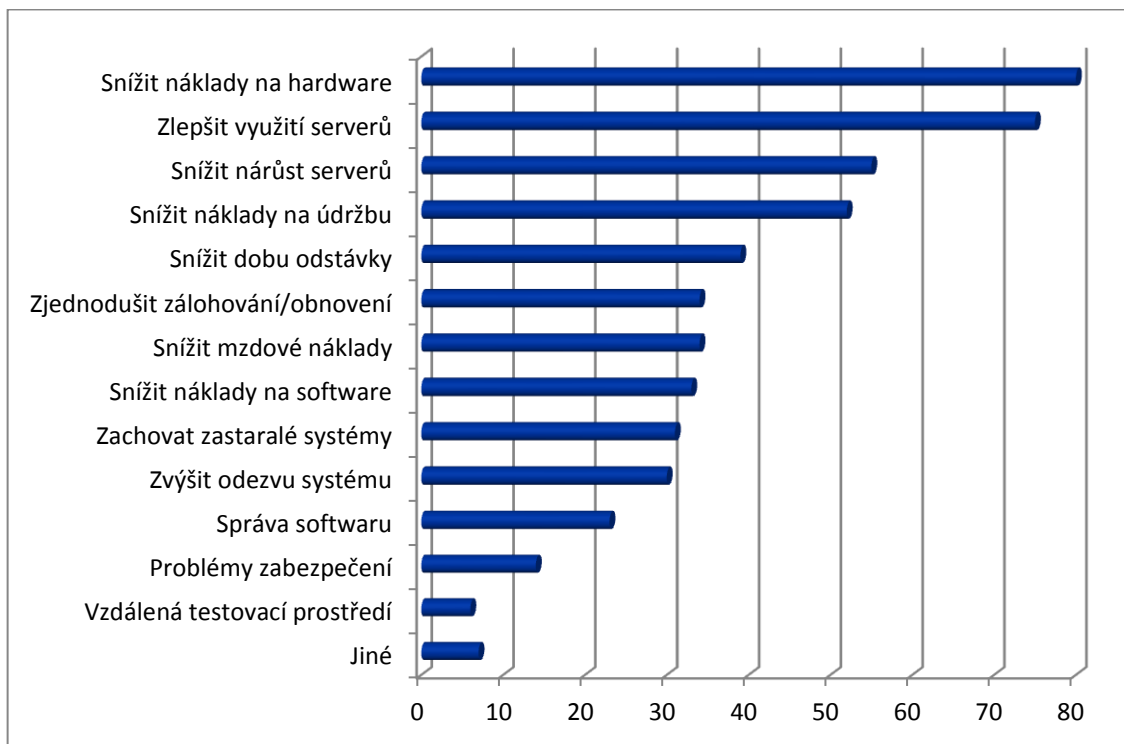
Prvním již zmiňovaným důvodem je bezesporu možnost práce v prostředí, které je naprosto nezávislé na své nadstavbě, kdy v případě chyby nebo kritického selhání není ohrožen systém, na kterém virtuální stroj běží. Virtuální stroj je také snadno přenositelný, stačí ho pouze zkopírovat nebo exportovat z programu, který ho virtualizuje. Chystáte-li se tedy například v budoucnu na nějaké školení, stačí nainstalovat jeden virtuální stroj, a ten rozkopírovat po učebně a importovat na vaší virtualizační platformě.

Předchozí odstavec naznačuje možnosti využití pro osobní potřeby nebo například potřeby školitele, který bude mít k dispozici učebnu, která mu bude přidělena v cizím prostředí. Jaký význam má ale virtualizace v širším měřítku? Řekněme například ve firmě s počtem zaměstnanců, který se blíží řádu několika tisíců? To si přiblížíme v následujících podkapitolách.

### 2.2.1 VYUŽITÍ VIRTUALIZACE V KOMERČNÍM PROSTŘEDÍ

Zde se dostáváme k jádru toho, proč je virtualizace tak populární. Jsou to peníze a úspory, které může virtualizace do podniku přinést.

**Nejčastějšími stimuly**, proč podniky přecházejí na virtualizaci, je snížení nákladů na hardware, přičemž zároveň dochází ke zlepšení využití serverů, snížení nárůstu serverů, snížení nákladů na údržbu, snížení doby odstávky, snížení mzdových nákladů, zjednodušení zálohy/obnovy a mnoho dalších, viz obr. 2.1[6].



obr. 2.1 – Důvody pro virtualizaci

Obyčejný server bez virtualizace využívá zpravidla 5-10 % svého výkonu [7]. Zároveň, dokud nebyla možnost virtualizovat, bylo možné využívat jeden server pouze na jednu věc. Díky tomu pořizovala každá firma hned několik serverů na nejrůznější úlohy. Při takovém postupu je prvním požadavkem místo, kam se servery umístí. Jelikož každý server vytváří spoustu tepla, je třeba aby místo bylo dobře větrané nebo klimatizované.

To nás přivádí k dalšímu faktoru, kterým je spotřeba elektrické energie. Kromě toho, že bude většinu času fungovat klimatizace, spotřebují servery další spoustu energie při běhu. Servery zpravidla zabírají kritické místo v rámci IT infrastruktury podniku, protože server by měl být schopný běžet nonstop.

Dnes je možné, díky virtualizaci, vytvořit v rámci jednoho serveru několik serverů virtuálních a na každém z nich nechat běžet něco jiného. Využití serveru tak může stoupnout až na 80%. Vždy je lepší ponechat ve výkonu nějakou rezervu, pro případy neočekávané zátěže. Tím, že je jeden server schopen obsluhovat například 8 dalších virtuálních serverů, z nichž každý plní jinou úlohu, odpadá nutnost pořizovat dalších 8 fyzických serverů. Ty by jinak zbytečně zabíraly prostor, zvyšovaly náklady na energii a celkově se chovaly nešetrně k životnímu

prostředí, což je při současném trendu společenské odpovědnosti firem nepřijatelné.

Úspory nákladů však nejsou jediným pozitivem využívání virtualizace. Může například nastat situace, kdy bude firma chtít stále používat software, který je už zastaralý a není schopen běžet na novějším hardware. V takovém případě není nic jednoduššího než vytvořit virtuální stroj s danými požadavky a software přesunout na něj.

### 2.2.2 VYSOKÁ DOSTUPNOST A AUTOMATIZOVANÁ SPRÁVA

Kromě toho, jak virtualizace dokáže znatelně snižovat náklady v serverové infrastruktuře hned na několika frontách, existují i další technologie, které je potřeba zohledňovat, protože značně ulehčují práci a správu virtuálního prostředí. Děje se tak s příchodem dalších novinek v tomto rychle rostoucím odvětví IT.

Příkladem jak zvýšit dostupnost může být použití tzv. **live migrate** [8]. Pokud má dojít k odstávce fyzického serveru, je možné pomocí živé migrace přesunout virtuální server na jiný fyzický server bez registrovaného výpadku. Virtuální server po přesunu pokračuje v činnosti, kterou prováděl před migrací a obsah jeho operační paměti je kombinací stavu před začátkem migrace a změn, které proběhly během ní.

Zajímavá je živá migrace ve spojení s prvky pro distribuovanou správu energie (**Distributed Power Management** [9]). Jedná se o další metodu, jak šetřit dostupné energetické zdroje, které spotřebují servery. DPM se stará o monitorování energetického vytížení fyzických serverů. V případě, že nejsou fyzické servery efektivně využívány, pokusí se DPM uspořádat jejich zátěž co nejefektivněji tak, aby se využívalo minimálního počtu fyzických zdrojů, a nepotřebné servery jsou vypnuty. Pokud dojde k nárůstu požadavků na výkon, může se provést opačný proces, který po vyhodnocení spustí další fyzické servery a přesune virtuální stroje mezi fyzickými servery tak, aby opět byly energetické zdroje využity co nejefektivněji.

Technologií, která se velice podobá DPM je DRS (**Dynamic Resource Scheduling** [10]). Funguje velice podobně jako DPM, ale na rozdíl od elektrické energie se kontrolují dostupné hardwarové zdroje. Uvedme si příklad, kde je jeden

server vysoce zatížen udržováním běhu virtuálních strojů a jiný oproti tomu má zátěž minimální. DRS se bude snažit tyto zdroje co nejefektivněji využívat, a podle potřeby dojde k migraci virtuálních strojů tak, aby došlo k rovnoměrnému rozložení výkonu.

Díky těmto technologiím je zaručena vysoká dostupnost virtuálních serverů téměř v jakékoliv situaci.

## 3 VIRTUALIZAČNÍ SOFTWARE

Výrobci virtualizačního software existuje celá řada. Zaměřím se ale především na čtyři hlavní a jimi poskytovaná řešení. Řeč bude především o **serverové a desktopové virtualizaci**.

Při výběru technologie, kterou budeme používat pro virtualizaci v našem prostředí, musíme zvážit, na co budeme virtualizaci používat. První otázkou by určitě mělo být, jestli si vystačíme s programy, které jsou dostupné zdarma, nebo budeme investovat do jedné z komerčních možností. Tento přehled zohledňuje obě možnosti.

### 3.1 VMWARE<sup>®</sup>

Mezi hlavní software v oboru desktopové a serverové virtualizace firmy VMware patří následující produkty – **VMware Player**, **VMware Workstation**, **VMware vSphere**, **VMware Fusion** a **VMware View** [11].

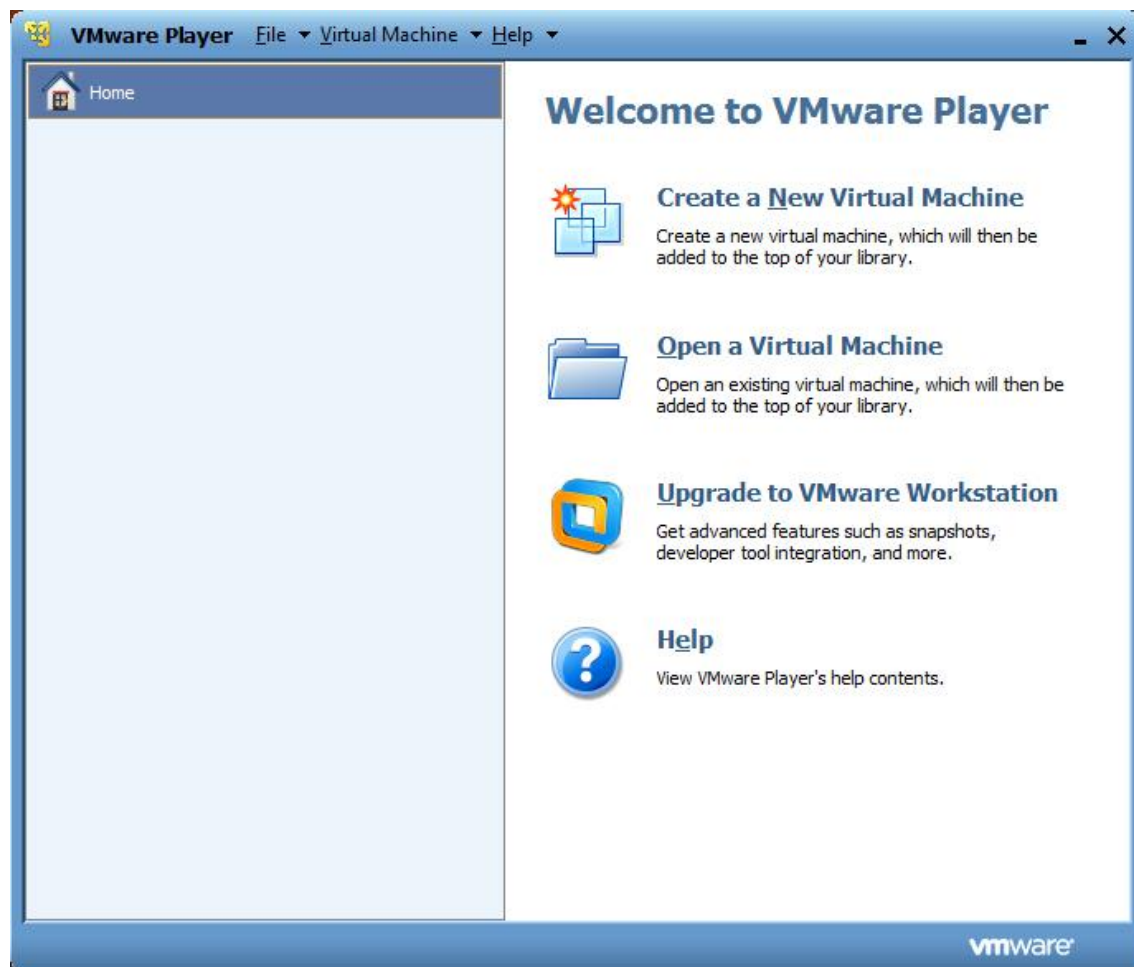
#### 3.1.1 VMWARE<sup>®</sup> PLAYER

Player je program určený pro softwarovou virtualizaci desktopových operačních systémů, dostupný pro platformy Windows a Linux. Jedná se o nižší verzi Workstation, která není nijak časově omezená a je dostupná zcela zadarmo. Mezi nabízené nástroje patří vytvoření virtuálního stroje za pomoci **.iso image** souboru, instalace z disku v mechanice nebo načtení dříve vytvořené instance.

Pokud se rozhodnete pro načtení dříve vytvořeného stroje, je možné použít i jiný formát než mají virtuální stroje vytvořené za pomoci softwaru od firmy VMware. Mezi **používané formáty** patří například **.ova** a **.ovf** soubory, které je schopen vyexportováním virtuálního stroje vytvořit program VirtualBox, nebo **.vmc** soubory, které poskytuje Microsoft Virtual PC. Nutno ovšem podotknout, že ne vždy je zaručena úplná kompatibilita a stoprocentní načtení virtuálního stroje vytvořeného na jiné platformě.

Zajímavým prvkem, který pak při vytváření nového virtuálního stroje Player poskytuje, je prvek **Easy install**. Ve většině případů Player z instalačního zdroje přímo detekuje, jaký OS je instalován a nabídne možnost vyplnit základní

informace potřebné k instalaci. Celá instalace je pak provedena automaticky, až do bodu spuštění OS a nainstalování VMware Tools (balíček ovladačů a nástrojů pro lepší a snadnější běh virtuálního stroje na hostitelském OS).



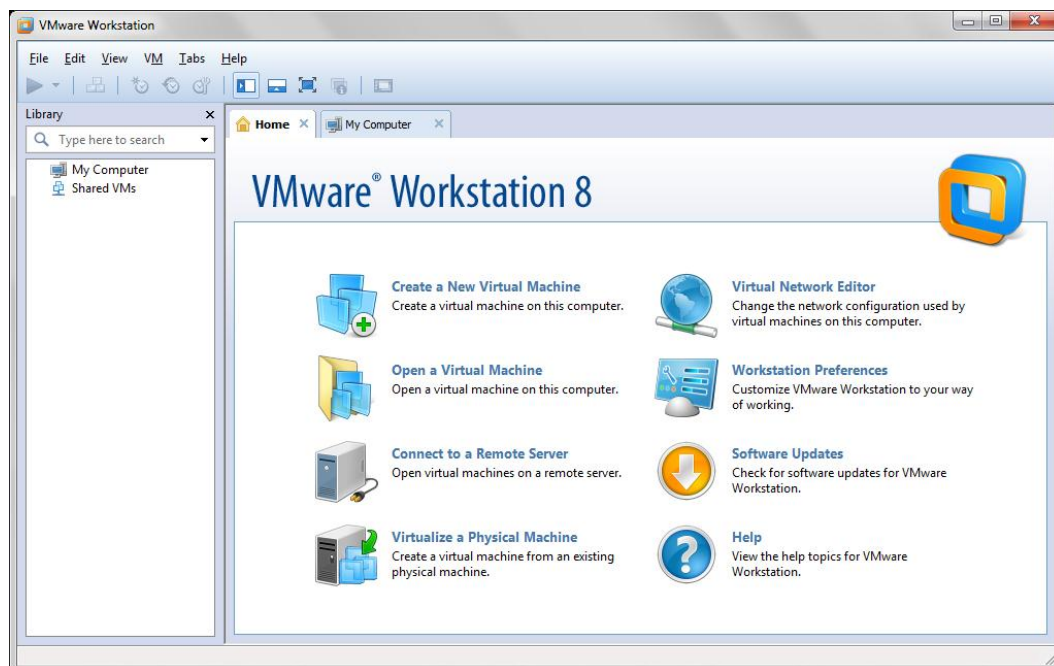
obr. 3.1 – Uživatelské rozhraní programu VMware Player

### 3.1.2 VMWARE® WORKSTATION

Workstation je jedním ze dvou hlavních virtualizačních nástrojů, které VMware nabízí. Na rozdíl od Playeru, Workstation poskytuje celou řadu dalších možností, jak nakládat s virtuálními stroji. Lze používat snapshoty, klonovat virtuální stroje a testovat aplikace v komplexních prostředích s širokou škálou operačních systémů na jednom jediném počítači.

Jedná se o velmi obsáhlou aplikaci pro softwarovou virtualizaci nad operačními systémy Windows a Linux, která je bohužel komerční a pro využití zdarma poskytuje pouze 30denní zkušební verzi. Obsahuje sadu průvodců pro snadný začátek. Kromě vytvoření nového virtuálního stroje a otevření existujícího

virtuálního stroje z lokálního disku nabízí například možnosti otevření virtuálního stroje na vzdáleném serveru nebo vytvoření virtuálního stroje z fyzického stroje, na kterém je Workstation nainstalován, viz obr. 3.2. [12]



obr. 3.2 – Uživatelské rozhraní programu VMware Workstation po spuštění

### 3.1.3 VMWARE<sup>®</sup> VSPHERE

Z produktů, které VMware nabízí, je vSphere jednoznačně tím nejkompexnějším. Je to virtualizační platforma, na které se dá postavit celá virtuální infrastruktura velkých podniků. Poskytuje nástroje na tvorbu kompletní cloudové infrastruktury, virtuálních počítačů, virtuálních úložišť a virtuálních sítí s vysokou dostupností, bezpečností a škálovatelností. Tento produkt je zde zmíněn pouze okrajově, pro více informací o vSphere se můžete podívat přímo na stránky výrobce v odkazu. [13]

### 3.1.4 VMWARE<sup>®</sup> FUSION

Na rozdíl od Player a Workstation, které jsou zaměřeny na softwarovou virtualizaci nejrozličnějších OS v prostředí Windows a Linux, je Fusion program, který umožňuje spouštět aplikace operačního systému Windows pod operačními systémy Mac.

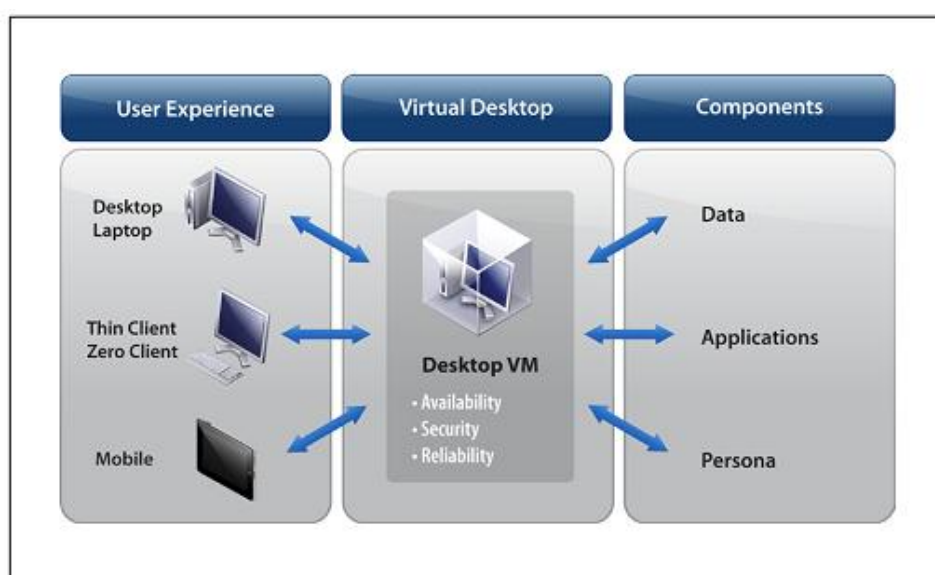


Fusion je schopen přemigrovat systém Windows a veškerý jeho obsah z klasického PC na stroj, který využívá operačního systému Mac společnosti Apple. Uživatel Macu je pak schopen spouštět veškeré aplikace z migrovaných Windows přímo v prostředí oken svého operačního systému, a zároveň přistupovat ke svým uživatelským souborům z původního počítače. Jeho další možností je schopnost využívat zařízení, která mají ovladače pouze pro Windows. [14]

### 3.1.5 VMWARE® VIEW

Program View umožňuje automatizovat správu tisíců desktopů a bezpečně poskytovat desktop vzdáleně jako službu z centrální lokace na téměř jakémkoliv zařízení.

View zapouzdřuje operační systém, aplikace, profily a data do izolovaných vrstev pro lepší správu desktopů a dynamicky skládá desktop na požadavek uživatele s personalizovaným uspořádáním jeho individuálního nastavení, viz obr. 3.3 [15].



obr. 3.3 – Zapouzdření izolovaných vrstev programu VMware View

Díky izolovaným vrstvám a skládání desktopu dynamicky až na požadavek, je nespornou výhodou aktualizace virtualizovaného operačního systému. Ta se totiž provádí jen na jednom centrálním master image, aniž by nějak byla ovlivňována uživatelská nastavení, data nebo aplikace. [15]

## 3.2 MICROSOFT®

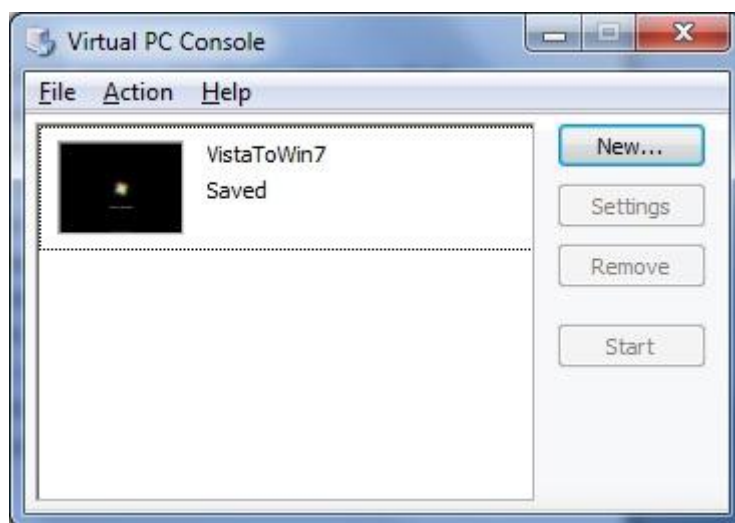
Společnost Microsoft se zabývá vývojem, výrobou, licencováním a podporou široké škály produktů a služeb, které jsou spjaté především s počítačovým softwarem.

Z produktů, které Microsoft nabízí pro virtualizaci, stojí nejvíce za zmínku produkt pro serverovou virtualizaci Microsoft **Hyper-V**. Mezi další produkty používané k virtualizaci pak řadíme z těch pro nás zajímavějších **Microsoft Virtual PC** a **Microsoft Virtual Server 2005**. Oba tyto produkty už jsou poněkud staršího rázu. Čas od času se ale dají ještě objevit v aktivním užívání.

### 3.2.1 MICROSOFT® VIRTUAL PC

Začneme u toho nejjednoduššího, co Microsoft nabízí. Microsoft Virtual PC je asi jedním z neznámějších prvotních pokusů, jak spouštět virtuální počítače v prostředí operačního systému Windows. Jedná se o program, který je zcela zdarma a nezabírá řádově ani 100 MB na disku. Jde o jednoduché rozhraní, jak co nejrychleji nainstalovat virtuální stroj a spustit.

Je to vhodné řešení softwarové virtualizace nad systémem pro začátečníky, kdy uživatel neplánuje používat nejrůznější edice systémů, ale zaměřuje se spíše na platformu Windows. Samočřejmě je zde možnost provozovat i celou řadu Linuxových distribucí, ne vždy ale fungují tak, jak by měly, a některé se vám ani nepodaří nainstalovat. [16]



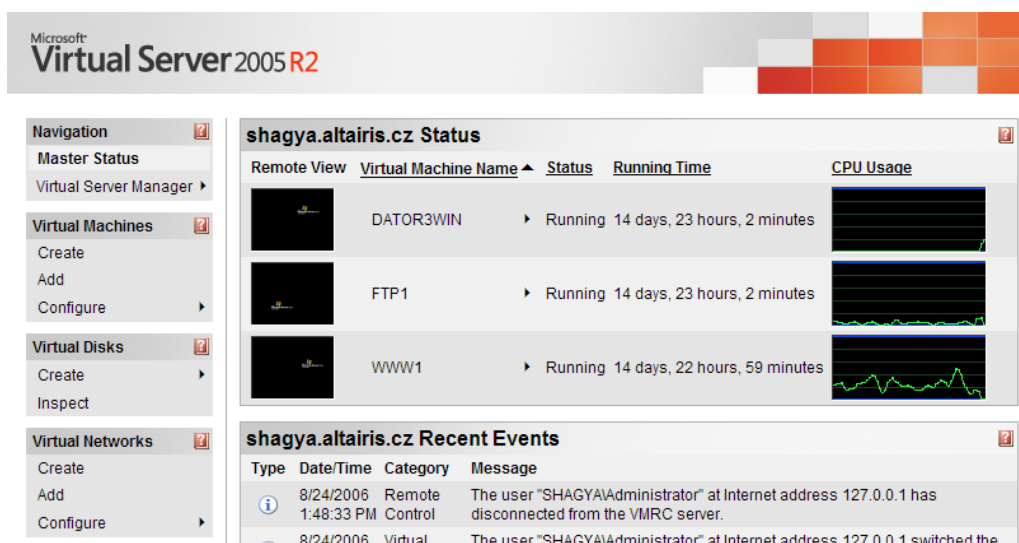
obr. 3.4 – Uživatelské rozhraní programu Virtual PC

### 3.2.2 MICROSOFT® VIRTUAL SERVER 2005

Microsoft Virtual Server je uvolněn zcela zdarma, a to i pro komerční použití. Oproti Microsoft Virtual PC je primárně zaměřen na hostování virtuálních serverových operačních systémů. Stejně jako Virtual PC je schopen používat některé distribuce Linuxu, ale nelze počítat s tím, že budou fungovat všechny. Jeho použití může sloužit k testování změn konfigurace v bezpečném virtuálním prostředí nebo přímo k ostrému provozu více virtuálních serverů na jednom fyzickém stroji.

Oproti Virtual PC se spouští jako systémová služba a virtuální servery tak mají možnost spustit se hned při startu hostitelského systému. Mezi jednu z jeho zajímavých funkcí patří Migration Toolkit, který je schopen převádět fyzické servery do virtuálního prostředí (podobně jako u VMware Workstation). Stejně jako Virtual PC je dobrým řešením, pokud se uživatel chystá s technologií virtualizace seznamovat. [17]

Prostředí Virtual Server 2005 R2 viz obr. 3.5[18].



obr. 3.5 – Uživatelské rozhraní programu Virtual Server 2005 R2

### 3.2.3 MICROSOFT® HYPER-V

Svým charakterem se dá Hyper-V považovat za nástupce Virtual Serveru 2005 R2. Poprvé se Hyper-V objevil jako Beta verze, která byla dodávána s některými edicemi Windows Server 2008. Na plnohodnotnou verzi byla automaticky aktualizována přes Windows Update zhruba v polovině roku 2008.

Přízvisko 2008 R2 si získal až poté, co nastoupila na trh verze systému Windows Server 2008 R2. Spolu s prvním ServicePackem byla uvolněna zatím současná třetí verze této virtualizační platformy, která nese název Hyper-V 2008 R2 SP1.

Hyper-V je **integrovaná část Windows Server**, která poskytuje základní stavební prvky k přechodu na cloud a zároveň umožňuje starat se o celou virtuální infrastrukturu podniku, který ho používá. Dokáže zlepšovat bezpečnost a spolehlivost, zvýšit dostupnost virtualizovaných prvků, minimalizovat výpadky, chránit data za užití zálohování za běhu, živou migraci virtuálních strojů aj., podobně jako už zmíněný VMware vSphere.

Z nabízených produktů pro komplexní virtualizační řešení podniků jsou Microsoft Hyper-V spolu s VMware vSphere nejvhodnějšími kandidáty. [19]

### 3.3 CITRIX®

Citrix Systems, Inc., je nadnárodní korporace dodávající technologie pro virtualizaci serverů, virtualizaci desktopů, počítačových sítí a cloud computingu. [20]

Citrix poskytuje řešení **Citrix Delivery Center**, které zahrnuje **XenDesktop**, **XenApp** (dříve známý jako **Citrix WinFrame Server**, **Citrix MetaFrame Server** a **Citrix Presentation Server**), **XenServer** a **NetScaler**, virtualizuje servery, desktopy a aplikace, centralizuje je v datových centrech a šíří je jako službu na vyžádání.

Tyto technologie umožňují zejména:

- vysokou flexibilitu firemního IT, rychlou reakci na požadavky kladené na IT oddělení - implementaci nových funkcí nebo změny pro koncové uživatele
- nezávislost na konkrétním koncovém zařízení a umístění uživatele, bez ohledu na lokalitu a způsob připojení
- zvýšení bezpečnosti, díky centralizaci [21]

Virtualizace u Citrixu funguje v mnoha případech jako aplikace typu server/klient. Na serveru je nainstalován virtualizovaný systém a klient k němu přistupuje, jako ke vzdálenému desktopu.

### 3.4 ORACLE®

Oracle poskytuje celé spektrum virtualizačních řešení od desktopů až po datová centra [22]. Nám však bude stačit ke zmínění jeden, který bude použitý i v praktické části této práce. Je jím program **Oracle VM VirtualBox**.

Na rozdíl od virtualizačních platforem firmy VMware je zcela zadarmo a neposkytuje jen zkušební verzi, která je po 30 dnech bez licence nepoužitelná. Zároveň není „ořezaná“ jako VMware Player a poskytuje oproti němu pro nás další důležité prvky. Software společnosti Microsoft dokáže virtualizovat jen omezenou množinu operačních systémů s plnou podporou většinou jen u svých produktů a Citrix vyžaduje nainstalování celé infrastruktury svých produktů, jelikož funguje způsobem server/klient.

Oracle VM VirtualBox je oproti tomu jednoduchý, malý a účinný software, pro softwarovou virtualizaci nad operačním systémem Windows nebo Linux. Podrobněji bude VirtualBox popsán v následující kapitole.

## 4 VIRTUALBOX

Program Oracle VM VirtualBox byl zvolený již při zadání celé práce. Mezi hlavní kritéria tohoto rozhodnutí patřil fakt, že je běžně používán k virtualizaci během výuky na KIVu, a klady, ke kterým patří schopnost virtualizovat bezproblémově téměř jakýkoliv operační systém. Další výhodou je, že program je zcela zdarma, a tak nás nic neomezuje v jeho používání.

V následující části práce rozeberu podrobně program Oracle VM VirtualBox a jeho možnosti konfigurace. Zároveň se budu zabývat problematickými místy tvorby programu pro automatickou konfiguraci VirtualBoxu a rozeberu způsoby, jakými jsou tato problematická místa řešena.

### 4.1 INSTALACE PROGRAMU

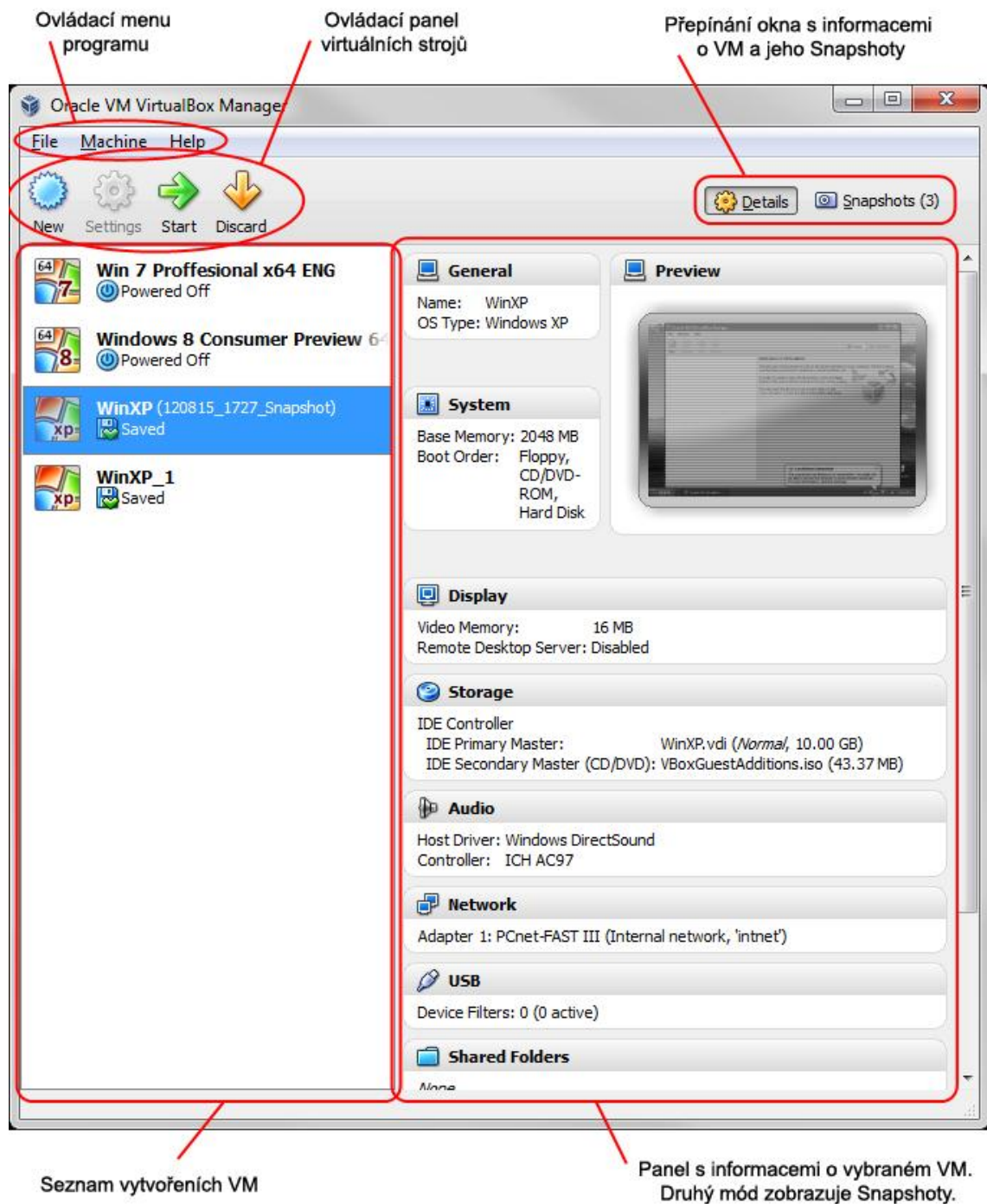
Základní požadavky na běh programu jsou operační systém Windows/Linux/Mac OS/Solaris, 140 MB na pevném disku (může se lišit v závislosti na OS, na který je VirtualBox instalován, 140 MB je pro případ instalace na systém Windows) a hardware o výkonu alespoň dvojnásobku toho, jaký je využíván při standardním běhu systému. Jinak se může stát, že při virtualizaci poběží oba systémy velmi pomalu a jakákoliv plynulá práce bude prakticky nemožná a velmi demotivační.

Soubory k instalaci programu VirtualBox jsou dostupné na webových stránkách firmy Oracle. [23]

Pro naše potřeby bude využíván VirtualBox na platformě Windows.

### 4.2 VYTVÁŘENÍ NOVÉHO VIRTUÁLNÍHO STROJE

Celá aplikace má přehledné a dobře řešené prostředí, které umožňuje snadnou navigaci během práce s ní, viz obr. 4.1.



obr. 4.1 – Uživatelské prostředí VirtualBoxu

Po kliknutí na tlačítko *New* se spustí průvodce vytvořením nového virtuálního stroje. Na výběr je ze šesti různých platform, mezi něž se řadí **Microsoft Windows, Linux, Solaris, BSD, IBM OS/2, MAC OS X** a další možnost pro vytvoření virtuálního hardwaru v úplně nejzákladnější podobě podle potřeb uživatele. Každá platforma pak ještě dává na výběr několik distribucí. Jedná se v podstatě jen o optimálně přednastavené virtuální hardwarové prostředí pro

každý vybraný operační systém. Pokud uživateli nevyhovuje základní nastavení, může je potom změnit osobně podle vlastní vůle.

Během vytváření nového stroje je uživatel vyzván k přidělení množství paměti virtuálnímu stroji a vytvoření virtuálního disku, který by mohl virtuální stroj používat. Lze použít i již dříve vytvořený **virtuální disk ve formátech**:

- **.vdi (VirtualBox Disk Image)** – formát nativně používaný VirtualBoxem
- **.vmdk (Virtual Machine Disk)** – formát nativně používaný programy VMware
- **.vhd (Virtual Hard Disk)** – formát nativně používaný programy Microsoft
- **.hdd (Parallels Hard Disk)**

Pokud se uživatel rozhodne k vytvoření nového virtuálního disku, může si vybrat z výše uvedených možností a zvolit přidělení fixní velikosti, kterou bude zabírat virtuální disk na skutečném disku na stálo, nebo možnost, kdy je místo přidělováno dynamicky (zvolí se maximální velikost a disk potom podle využitého místa roste až do svého maxima, zabírá při tom jen minimum, které skutečně využívá).

Tyto virtuální disky mohou být snadno přeneseny z jiných virtualizačních platforem. Stačí pouze disk zkopírovat a připojit k nově tvořenému stroji. Kompatibilita však není vždy zaručena, a proto je nejlepší zůstat věrný jedné platformě a přenášet pouze disky dříve vytvořené danou platformou. To neplatí pouze pro VirtualBox, ale také pro jiné virtualizační platformy.

Jakmile je stroji přiřazen virtuální disk, následuje už pouze shrnutí, a po potvrzení je virtuální stroj připravený ke spuštění.

### **4.3 KONFIGURACE VIRTUÁLNÍHO STROJE**

Tato podkapitola je asi nejdůležitější, co se možností konfigurace VirtualBoxu týká. Program v tomto směru nabízí opravdu mnoho možností. Pokusím se být co nejstručnější.

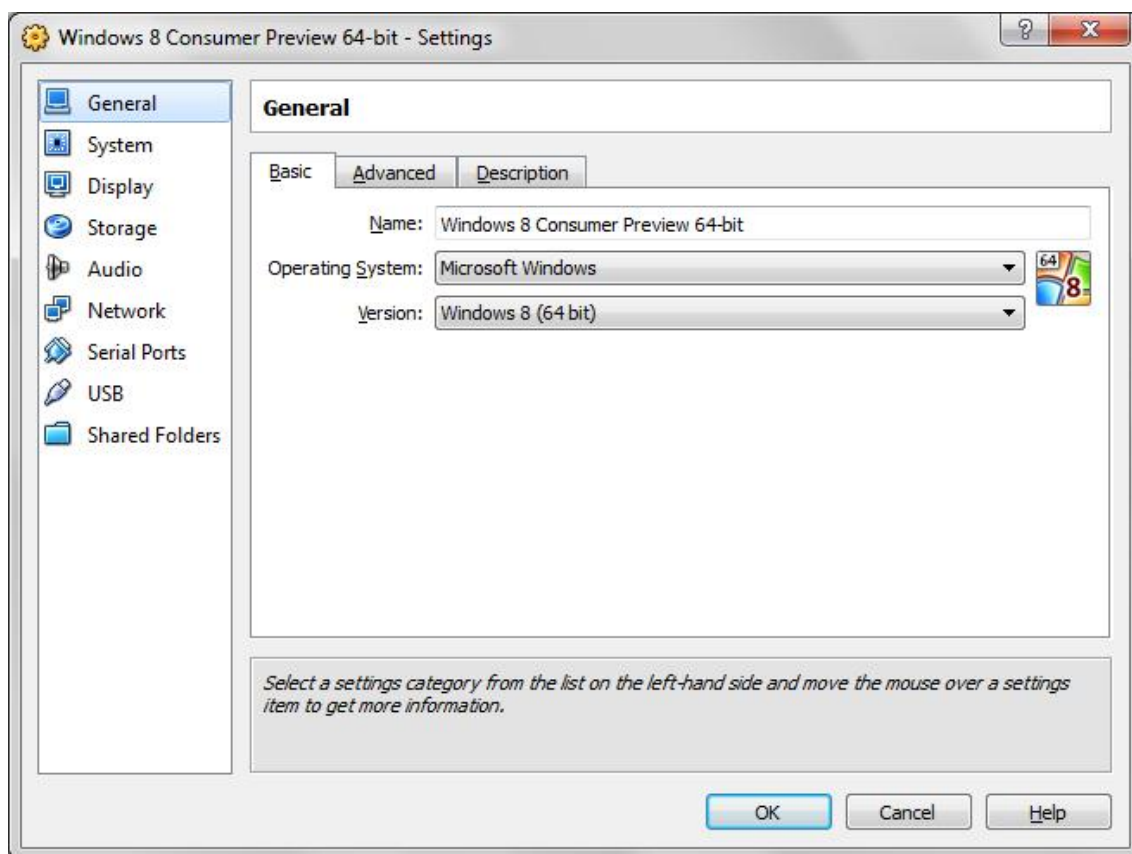
Začnu tím, že po vytvoření virtuálního stroje se v hlavním okně objeví možnost vybrat si konkrétní virtuální stroj z nabídky vytvořených, a prohlédnout si okamžitě jeho vlastnosti na panelu v pravé části okna, viz obr. 4.2.



Pokud chce uživatel okamžitě změnit některé z konkrétních nastavení, stačí vybrat položku v přehledu vedle seznamu virtuálních strojů a kliknout na její název. Po kliknutí se otevře okno, které ho navede na nastavení konkrétní položky.

Dalším způsobem je možnost vybrat virtuální stroj a kliknout na tlačítko *Settings* přímo pod hlavním menu programu. Nutno podotknout, že pokud je virtuální stroj v uloženém stavu, nebude tlačítko *Settings* aktivní. Pokud stroj běží, budou některé položky v nastavení zakázány.

Po kliknutí se otevře okno, kde má uživatel možnost nastavit nejrůznější vlastnosti. Od počtu procesorů a velikosti paměti, až po počet monitorů, na které bude virtuální stroj poskytovat výstup.



obr. 4.2 – Okno nastavení virtuálního stroje v programu VirtualBox

Abych přeci jen poskytl malý přehled, co vše je možné u virtuálního stroje nastavit, pokusím se to udělat pomocí seznamu položek, které se dají považovat za nejdůležitější:

- Obecné
  - Jméno virtuálního stroje

- Místo uložení snapshotů (ano, VirtualBox podporuje snapshoty)
- Systém
  - Velikost virtuální operační paměti
  - Bootovací pořadí zařízení
  - Počet virtuálních procesorů
  - Povolení hardwarové virtualizace
- Displej
  - Velikost video paměti
  - Počet monitorů poskytnutý virtuálnímu stroji
  - Vzdálený displej
- Úložiště
  - SATA/IDE kontrolory
  - Přidání virtuálních disků
- Audio
  - Nastavení zvuku
- Síť
  - Nastavení síťových adaptérů virtuálního stroje
- Sériové porty
  - Nastavení sériových portů
- USB
  - Povolení USB kontroleru
  - Filtrování USB zařízení
- Sdílené soubory
  - Nastavení sdílených souborů z fyzického stroje

Další a podrobnější informace lze nalézt v uživatelské příručce tohoto virtualizačního nástroje, která je dostupná online. [24]

#### **4.4 IMPORT/EXPORT VIRTUÁLNÍCH STROJŮ**

Pokud chce uživatel migrovat své projekty nebo virtuální stroje z jednoho fyzického stroje na jiný naskytají se mu v případě VirtualBoxu dvě možnosti.

První je prosté kopírování virtuálního disku. Virtuální disk se zkopíruje, na cílovém počítači se vytvoří virtuální stroj a k němu se připojí disk.

Druhá možnost je podstatně snazší. VirtualBox má možnost virtuální stroje exportovat do jednoho souboru, který obsahuje jak nastavení původního virtuálního stroje, tak i jeho virtuální disky. Tento exportovaný soubor se dá také zkopírovat, přenést a na cílové fyzické stanici s VirtualBoxem po zvolení možnosti pro import zařídí program veškerou administraci sám. Virtuální stroj je pak připraven k okamžitému použití.

## 4.5 VBoxManage

VirtualBox pro potřeby naší práce obsahuje užitečný nástroj, který nese jméno VBoxManage. Vedle možnosti nastavovat a užívat program přes uživatelské rozhraní, je možné program VirtualBox zcela a výhradně spravovat pouze prostřednictvím příkazové řádky pomocí tohoto nástroje (ukázka příkazů viz obr. 4.3 [24]). Není pak tedy problém za pomoci jakéhokoliv skriptovacího jazyka, který umí spouštět příkazy v něm uložené sekvenčně, sepsat skript, který při správné syntaxi nastaví program, tak jak uživatel potřebuje.

```
VBoxManage showvminfo <uuid>|<name> [--details]
                               [--machinereadable]
VBoxManage showvminfo <uuid>|<name> --log <idx>
VBoxManage registervm <filename>
VBoxManage unregistervm <uuid>|<name> [--delete]
VBoxManage createvm --name <name>
                    [--ostype <ostype>]
                    [--register]
                    [--basefolder <path>]
                    [--uuid <uuid>]
VBoxManage modifyvm <uuid|name>
                    [--name <name>]
                    [--ostype <ostype>]
                    [--memory <memorysize in MB>]
                    [--pagefusion on|off]
```

obr 4.3 – Ukázka příkazů VBoxManage

Přehled veškerých příkazů a jejich význam je k nalezení v již zmíněné uživatelské dokumentaci v kapitole 8. [24]

Mimo jiné se jedná o program, který je součástí instalace VirtualBoxu a je obsažen ve složce instalace jako jeden ze spustitelných souborů .exe. Takže není potřeba nic složitě doinstalovávat a nastavovat.

Uvedme si na závěr této kapitoly příklad užití programu VBoxManage, v příkazové řádce, kde bychom chtěli nastavit virtuální stroj pro Windows Server 2008. Tento stroj by měl 2048 MB RAM, 2 virtuální procesory a povolený síťový adaptér nastavený na NAT. Konfigurace by pak vypadala takto:

### **Příklad:**

```
vboxmanage createvm --name "WS2008" --register
vboxmanage modifyvm "WS2008" --memory 2048 --cpus 2 --boot1
disk
vboxmanage modifyvm "WS2008" --nic1 nat
vboxmanage modifyvm "WS2008" --cableconnected1 on
vboxmanage storagectl "WS2008" --name SATA0 --add sata
--bootable on
vboxmanage storageattach "WS2008" --storagectl SATA0 --type
hdd --mtype normal --port 0 --medium
\\localhost\Test\WS2008.vdi
```

### **Rozbor kódu:**

1. Vytvoření stroje s názvem „WS2008“ a jeho registrace do programu VirtualBox.
2. Přiřazení paměti o velikosti 2048 MB RAM, 2 virtuálních CPU a bootovací sekvence z pevného disku.
3. Nastavení síťového adaptéru na NAT.
4. Připojení síťového kabelu k síťovému adaptéru.
5. Vytvoření diskového kontroléru SATA a nastavení jeho vlastnosti jako bootovací.
6. Připojení virtuálního disku na první pozici virtuálního stroje z lokace \\localhost\Test\WS2008.vdi

## 5 PRAKTICKÁ ČÁST

Účelem praktické části je vytvoření programu, který by co nejvíce pomohl zautomatizovat přípravu učebny pro scénáře používání virtualizace operačních systémů během výuky, v laboratořích Katedry informatiky a výpočetní techniky (KIV).

### 5.1 ROZBOR KLÍČOVÝCH ÚLOH PROGRAMU

Cíle práce:

1. Zajistit kopírování souborů o velkém objemu dat v rámci učebny, včetně detekcí zda bylo úspěšné. Opakování při neúspěchu.
2. Import a nastavení virtuálního stroje v programu VirtualBox.
3. Změna přístupových práv souborů
4. Automatizace

Bod 2. je již částečně řešen v předchozí kapitole. Pro správnou automatizaci bylo nutné najít nějaký způsob, jak komunikovat s programem VirtualBox a dodávat mu příkazy, které chceme, aby vykonal. Díky programu VBoxManage je poskytnuto rozhraní, kterým můžeme VirtualBox nastavit.

Nejedná se však o jediné komplexní místo, které je potřeba řešit. Při virtualizaci desktopů je třeba starat se o práci s obrazovými soubory disků nebo počítačů, které řádově dosahují jednotek až desítek jednotek gigabyte. Chceme-li tyto soubory přenášet v rámci sítě, existuje možnost, že dojde k nějaké chybě a soubor nebude přenesen nebo bude přenesen špatně, a to je další kritické místo na které můžeme narazit.

Když si představíme případ, kdy budeme chtít připravit učebnu na výuku s využitím virtuálního prostředí, budeme určitě potřebovat nějaký předpřipravený virtuální stroj, který bude pro všechny studenty na začátku výuky stejný.

Stačí si připravit takový stroj jeden, rozkopírovat jeho obraz na ostatní počítače, a ten implementovat do virtualizačního prostředí. Když tedy virtuální stroj připravíme, budeme mít možnost použít jeho virtuální disk nebo ho exportovat a získat tak jeden soubor, který pak budeme kopírovat. To je náš

soubor, který má velikost v řádu gigabyte. Způsob jakým je řešeno kopírování dat najdete v další kapitole.

Dalším bodem, který je spíše jen formální záležitostí je změna přístupových práv k těmto souborům, které jsou kopírovány. Čistě teoreticky mohou být přístupová práva změněna uživatelem ještě před kopírováním na ostatní stroje, a tím se můžeme vyhnout nastavování dodatečně. Kdyby taková situace nastala, je jedna z následujících kapitol věnována právě této problematice pod operačním systémem Windows.

Posledním bodem, který bych chtěl zmínit je řešení automatizace. Jistě nelze udělat vše automaticky a vždy je potřeba alespoň minimální aktivita ze strany uživatele, který bude chtít mít práci v učebně zjednodušenou. Veškerá jeho aktivita při práci s naším programem by měla ale později vést ke značnému zjednodušení nastavování učebny.

### 5.1.1 KOPÍROVÁNÍ SOUBORŮ O VELKÉM OBJEMU DAT

Při analýze, jaký zvolit nástroj nebo způsob kopírování, jsem se snažil najít takový, který by splňoval hlavně spolehlivost a možnost ovládní příkazy z příkazové řádky. Důvodem je stejně jako u VBoxManage psaní skriptu, který by po svém spuštění dokázal plnit příkazy ke kopírování jeden po druhém.

Na internetu se dá najít spousta bezplatných řešení, která umí spolehlivě a rychle kopírovat velké objemy dat. Bohužel se dají většinou ovládat pouze pomocí grafického uživatelského rozhraní, a tak nejsou pro možnost ovládní skriptem vyhovující. O nejzajímavějších řešeních, která používají grafické rozhraní, a lze je použít ke kopírování velkých souborů se můžete dočíst na webové stránce v odkazu [25].

Kopírovacím nástrojem, který byl zvolen pro tuto práci je Robust File Copy neboli zkráceně **Robocopy**. Robocopy byl původně součástí rozšiřujícího balíčku nástrojů pro Windows Server 2003. Je jednoduchým řešením, protože od verze operačního systému Windows 7 je implementován v základní instalaci systému. Odpadá tím tedy další starost o dodatečnou instalaci. [26]

## Příklad:

```
robocopy C:\Windows D:\Temp
```

Příklad zobrazuje základní příkaz pro spuštění Robocopy, kdy první atribut reprezentuje zdrojovou cestu a druhý cestu cílovou. Podrobnější ukázka v kapitole 5.2.4 Realizace kopírování.

Nástroj Robocopy nabízí celou řadu parametrů, s kterými lze soubory kopírovat. Pro nápovědu o parametrech stačí spustit v příkazové řádce příkaz „Robocopy“ s parametrem „/?“.

Pro nás je důležitých několik skutečností. První je, že Robocopy umí navázat na předchozí kopírování souborů, a to i při úplném přerušení kopírování a vypnutí obou počítačů. To zjednodušuje případ, kdy dojde k výpadku kopírování a velký soubor se tak nemusí kopírovat celý znovu, ale od místa kde předešlé kopírování skončilo.

Druhá je schopnost rozpoznat chyby, které nastanou při kopírování, a psát o nich hlášení. Tato hlášení mohou být zobrazována rovnou na obrazovku nebo zapisována do logového souboru.

Další pro nás užitečnou věcí je možnost opakování s prodlevou po nastalé chybě. Pokud je kopírování přerušeno pokouší se Robocopy o navázání kopírování, podle zadané časové prodlevy a počtu opakování. Robocopy poskytuje opravdu široké možnosti kopírování souborů.

### 5.1.2 ZMĚNA PŘÍSTUPOVÝCH PRÁV SOUBORŮ

Pokud chceme měnit přístupová práva souborů pod operačním systémem Windows, stačí k tomu dostupné systémové nástroje. U Microsoft Windows je to nástroj **Cacls** a **iCacls** (Change Access Control Lists). Oba dokáží bezproblémově měnit přístupová práva souborů. Nastavovaná práva mohou být:

- R – Read (čtení)
- W – Write (zápis)
- C – Change (změna)

- F – Full Control (plná kontrola)

Aby mohl uživatel, provést změnu nastavení přístupových práv, musí mít pro měněný soubor práva nastavená na „změnu“.

Obě verze jsou srovnatelné a je v zásadě jedno, kterou pro naši úlohu použijeme. Většinou je ale lepší používat tu novější, a tou je iCacls. iCacls přináší oproti Cacls nějaká nová nastavení, ale zároveň v něm některé prvky už nejsou.

Pro potřeby naší práce je použit iCacls s tím, že jeho funkce je omezená pouze právy uživatele nad soubory, u kterých se práva mění.

### **Ukázka kódu změny přístupových práv souborů:**

```
icacls "\\localhost\TestCopy\" /t /grant Everyone:C
```

Atributy, které používá iCacls:

- Prvním atributem je adresář nebo soubor, u kterého jsou měněna práva.
- /t – změna práv je aplikována na všechny soubory a podsložky zadané cesty.
- /grant Sid:vlastnost – přiřazení práv podle systémového identifikátoru uživatele nebo skupiny a vlastnosti.

### **5.1.3 AUTOMATIZACE**

Nejčastějším způsobem jak řešit automatické úlohy je použití skriptu. Pro tyto úlohy se dá zvolit prakticky jakýkoliv programovací jazyk, který je schopen spouštět příkazy sekvenčně.

V našem případě je použitý jazyk **PowerShell**. Jedním z důvodů je jeho integrace do prostředí Windows už od verze WinXP SP2. Jedná se o rozšiřitelný shell se skriptovacím jazykem, který je od té doby integrován do instalovaných nástrojů na systému Windows. Produkt je založen na platformě .NET Framework, a z toho vyplývá i jeho odlišnost od ostatních shellů. Jako výstupy skriptu PowerShell negeneruje text, ale objekty, které pak mohou být posílány pro zpracování dále. Spolu s jazykem přichází i systémové nástroje pro tvorbu skriptů a konzole, ze které je možné skripty spouštět. [27]



Jednou z alternativ by bylo například použití skriptovacího jazyka **Python**. Důvody pro zamítnutí byly větší zkušenost s jazykem PowerShell a odpadnutí nutnosti jazyk doinstalovat a nastavit pro prostředí operačního systému, a to v rámci každé stanice zvlášť.

## 5.2 VÝVOJ PROGRAMU

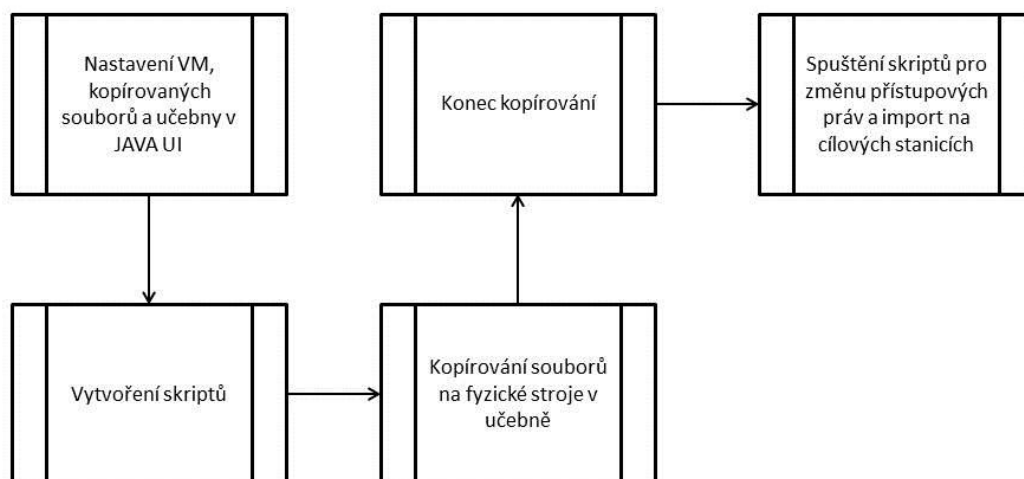
V této kapitole bude probraný návrh a vývoj programu. V předchozích kapitolách jsem naznačil, kritická místa a nástroje, které slouží pro jejich řešení, nyní popíšu, jak je program zpracován.

### 5.2.1 NÁVRH PROGRAMU

Poté co byly určeny hlavní cíle programu (kopírování souborů, nastavení virtuálního stroje, automatizace úkonů) bylo potřeba navrhnout, jak bude program naší úlohu řešit.

- Základem je ovládání, a uživatelsky nejpřívětivější a nejjednodušší formou pro práci většinou bývá možnost komunikovat s programem pomocí grafického rozhraní. Pro vytvoření grafického rozhraní byl zvolen programovací jazyk **JAVA**, který poskytuje knihovny **Swing** a **AWT**, pro tvorbu grafických komponent programů.
- Kopírování souborů je realizováno programem **Robocopy**, který je spouštěn a kontrolován pomocí skriptu napsaného v jazyce **PowerShell**.
- Automatickou konfiguraci programu **VirtualBox**, zajišťuje nástroj **VBoxManage** ve spojení s jazykem **PowerShell**.
- Přidělení přístupových práv řídí skript vytvořený v jazyce **PowerShell** ovládající program **iCacsl**.

Navržený postup funkcí programu pak vypadá takto:



obr 5.1 - Postup funkcí programu

### 5.2.2 NÁVRH UŽIVATELSKÉHO ROZHRAŇÍ

Při návrhu byl kladen asi největší důraz na to, aby bylo prostředí programu přehledné a jednoduché na ovládání. Vycházel jsem přitom ze čtyř hlavních bodů:

- Nastavit kopírované soubory s přístupovými právy
- Nastavit virtuální stroj
- Nastavit počítače v učebně, na které budeme nastavení aplikovat
- Spustit automatické nastavení

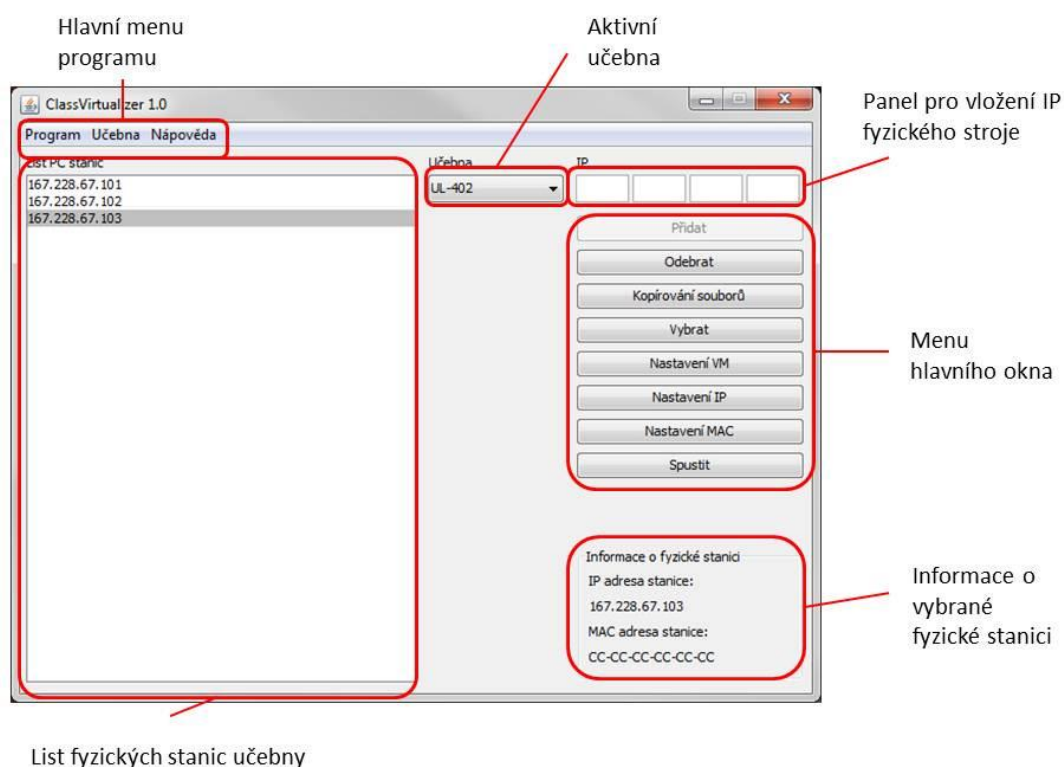
Program z těchto bodů vychází a má tři hlavní části.

- Hlavní okno s možností správy fyzických stanic
- Okno s nastavením souborů ke kopírování
- Okno s nastavením virtuálních strojů

Prvním krokem bylo navržení hlavního okna, kde by měl uživatel veškeré nástroje přehledně po ruce. Zároveň podle mě byla dobrá volba zaměřit se na fyzické stroje a kopírování souborů, o kterých program primárně je. Výsledkem

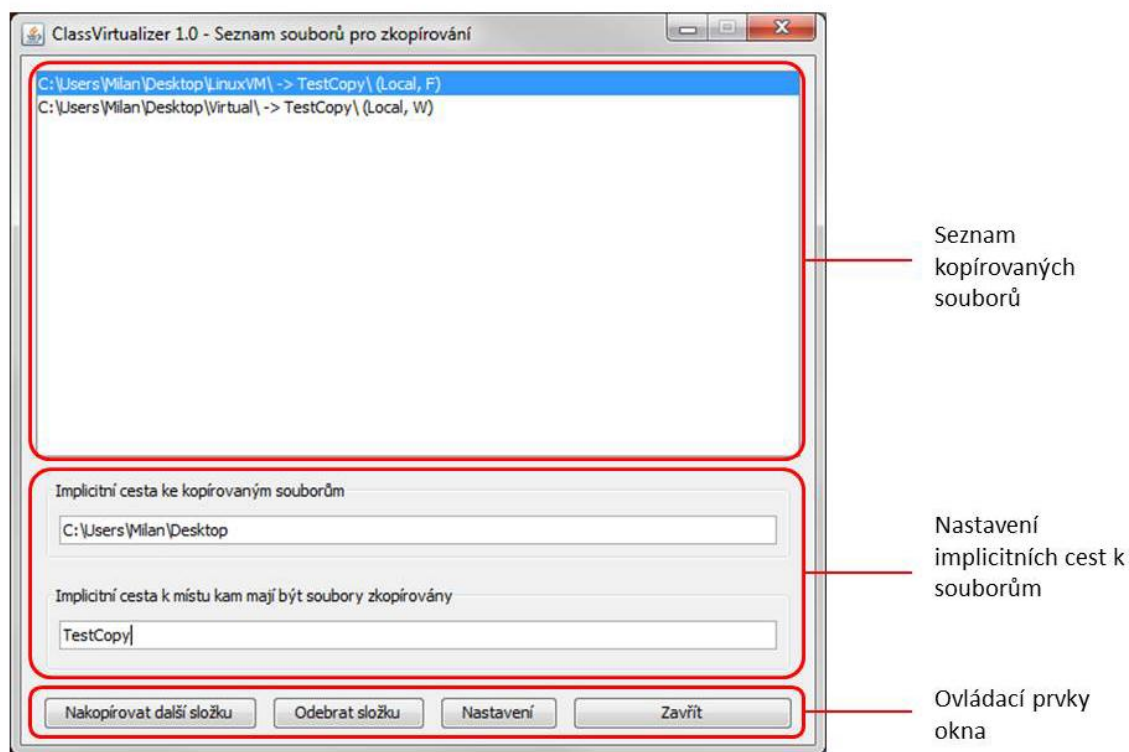
bylo okno, kde je možné přidávat fyzické stroje podle IP a MAC adresy a přes tlačítko přistupovat do samostatného okna a volit zdrojové a cílové složky ke kopírování.

Téměř ve finální části byla také do hlavního okna přidána možnost uchovávat informace o nastavení fyzických strojů více učeben. Hlavní okno viz obr. 5.2.



obr. 5.2 – Hlavní okno programu

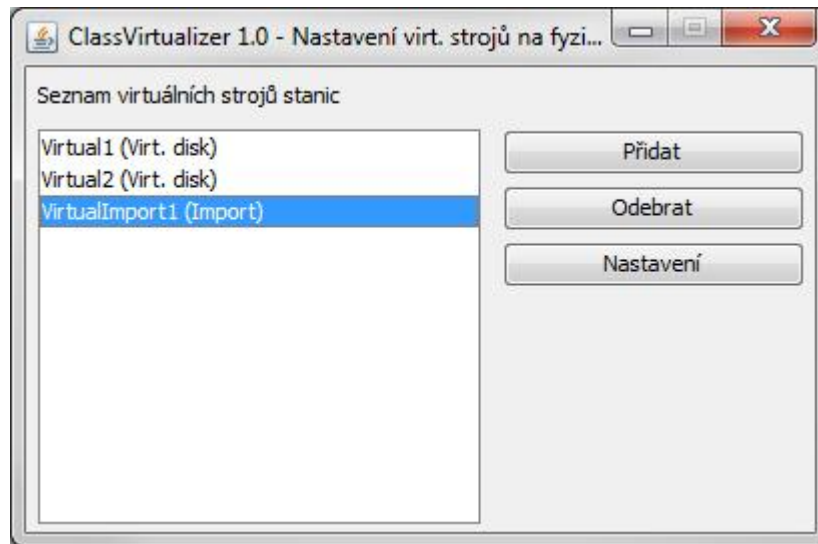
Následovalo vytvoření okna s nastavením kopírovaných souborů. Do řešení jsem zahrnul dvě možnosti jak soubory kopírovat. První bylo kopírování z lokálního úložiště na stanice v učebně a druhá, kdy jsou zdrojové soubory na fyzických stanicích již nakopírovány, ale potřebují být duplikovány. Vycházel jsem přitom z požadavků, které na mě kladl vedoucí práce. Návrh okna pro kopírování souborů viz obr. 5.3.



obr. 5.3 – Okno kopírování souborů

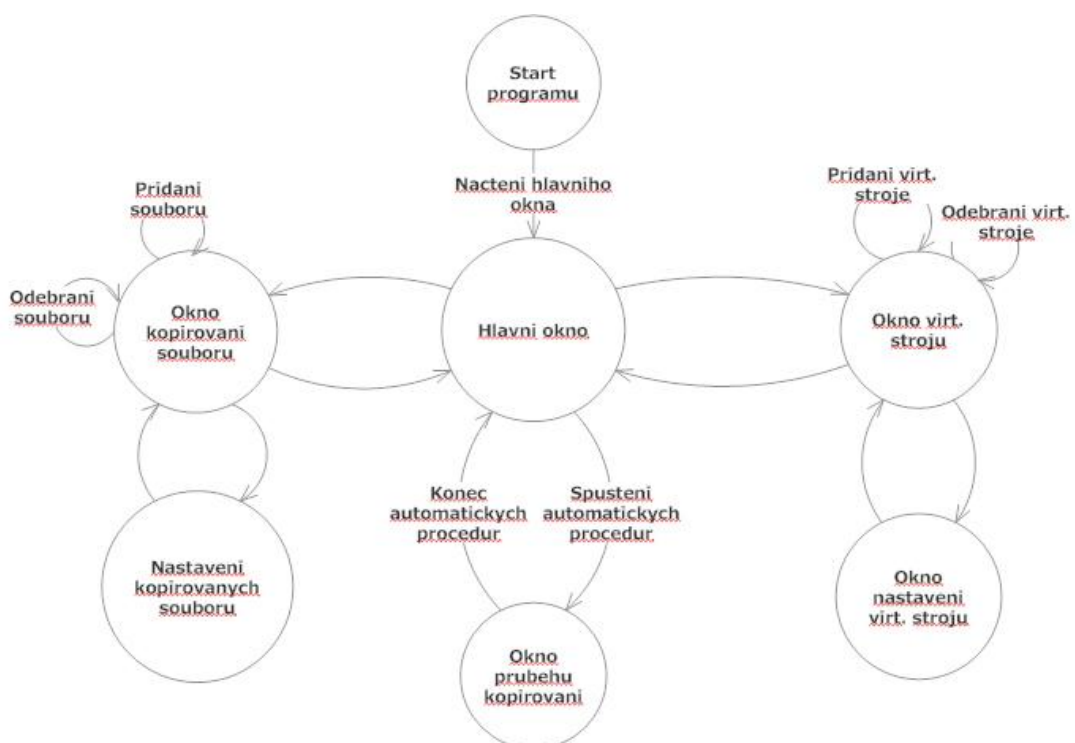
V horní části je seznam složek, které budou kopírovány. V části spodní jsou ovládací prvky okna. Program obsahuje průvodce krok za krokem, jak přidat nové soubory ke kopírování, a také okno nastavení pro případné pozdější změny. Podrobně se můžete seznámit s prvky a jejich funkcí v části „Uživatelská příručka“ v přílohách na konci práce.

Poslední záležitostí bylo navrhnout způsob, jak bude okno nastavovat virtuální stroje. Řešení tohoto úkolu bylo pojato co nejjednodušeji a stejně jako okno pro kopírování souborů obsahuje možnosti přidat virtuální stroj s průvodcem, nastavit a odebrat. V okně je vidět seznam aktuálně vytvořených virtuálních strojů viz obr. 5.4.



obr. 5.4 – Okno seznamu virtuálních strojů

Stavy uživatelského rozhraní programu, do kterých se může uživatel dostat, jsou shrnuty v následujícím diagramu, obr. 5.5.



obr. 5.5 – Stavy UI programu

### 5.2.3 POŽADAVKY NA ŘÍDÍCÍ STANICI A UČEBNU

Program je vytvořen pro systém Windows 7, na kterém byl i testován. Nicméně by měl být funkční i se starší verzí systému a to nejméně Win XP SP2 (SP2, kvůli přítomnosti PowerShell).

#### Softwarové požadavky na stroj v učebně:

1. Operační systém Windows 7
2. Nainstalovaný program Oracle VM VirtualBox

Všechny ostatní prvky obsahuje Windows 7 již ve své instalaci. Pro přehled je ale vyjmenuji:

- Windows PowerShell
- Robocopy
- iCacls

**Řídící stroj** musí ke všem předchozím prvkům v učebně mít nainstalovaný JAVA Runtime Environment, kvůli spuštění konfiguračního nástroje, který je výstupem této práce.

V prostředí učebny je pak potřeba na každém fyzickém stroji nastavit:

- Bootovací sekvenci na systém Windows
- Přidat adresář s programem VirtualBox do systémové proměnné „Path“
- Nastavit pro PowerShell vlastnost ExecutionPolicy na RemoteSigned.
- Vytvořit síťově sdílenou složku „VirtualClassImport“ pro uživatele, který bude program používat s právy pro zápis.

Návod k jednotlivým bodům naleznete v přílohách v uživatelské příručce programu.

### 5.2.4 REALIZACE KOPÍROVÁNÍ

Kopírování je v programu řešeno nástrojem **Robocopy** a skriptem vytvořeným v jazyce **PowerShell**. Právě tento úsek programu je pro celou jeho

funkcionalitu nejkritičtější, protože řídí celé kopírování velkých souborů. Při tvorbě skriptu bylo vycházeno z funkcí, které poskytuje nástroj Robocopy. Ve skriptu je zohledněn fakt, že Robocopy má možnost zápisu výstupu kopírování do logového souboru. Skript je poté schopný ve smyčce s timeoutem číst soubor a kontrolovat ho pro stavy, do kterých se Robocopy dostane.

Skript je pojmenován **CopyScript.ps1**, nachází se v podsložce „Scripts“ adresáře se spouštěcím souborem programu automatické konfigurace a má pět vstupních parametrů, které jsou zadávány při jeho spuštění (zdrojovou cestu kopírovaných souborů, cílovou cestu kopírovaných souborů, MAC adresu cílového stroje, číslo pořadí kopírované složky, pravdivostní hodnotu, zda se jedná o poslední kopírovaný soubor nebo ne a cestu do lokální složky programu).

#### **Ukázka parametrů skriptu:**

```
param ($SourcePath , $DestinationPath , $MAC , $Number,  
$Last, $LogPath)
```

Znalost parametrů zdrojové a cílové cesty (`$SourcePath` , `$DestinationPath`) je potřeba pro spuštění nástroje Robocopy.

#### **Ukázka příkazu spuštění Robocopy ve skriptu:**

```
robocopy $SourcePath $DestinationPath /z /e /r:50  
/Log:$RobocopyLogFile /Tee
```

Atributy, které používá Robocopy:

- `/z` – navázání na předchozí kopírování
- `/e` – kopírování včetně podadresářů, včetně prázdných
- `/r:N` – počet opakovaných pokusů navázání kopírování, při nastalé chybě (N – počet opakování)
- `/Log:soubor` – výstup stavu do log souboru
- `/tee` – výstup zároveň jak na obrazovku, tak i do logového souboru

Ve skriptu je počítáno i s možností, že by mohlo být kopírování násilně přerušeno. Nedá se zamezit všem případům, které mohou nastat. Skript alespoň počítá s možností, že může dojít k dočasnému výpadku sítě nebo dočasné ztrátě síťové cesty na základě vypnutí fyzického stroje. Pro tento případ je třetím ze vstupních parametrů programu MAC adresa fyzického počítače v učebně.

MAC adresa se používá ve spojení s funkcí **WakeOnLan** [28]. Jedná se o funkci počítače nastartovat po přijetí signálu ze sítě. Ten je vyslán, pokud nastává jeden z případů uvedených v předchozím odstavci. Je to také další ošetření pro případ, že by uživatel zapomněl některé fyzické stroje před kopírováním nastartovat, nebo kdyby někdo omylem během kopírování stroj vypnul. Funkce WakeOnLan musí být povolena v nastavení systému BIOS.

### Funkce pro WakeOnLAN:

```
Function WakeOnLan {
param ($MACAddress)
$Target = [byte[]]($MACAddress.split('-') |% {[int]"0x$_"})
Write-Verbose "Prevod MAC adresy na HEXA : $Target"
$UDPclient = new-Object System.Net.Sockets.UdpClient
$UDPclient.Connect(([System.Net.IPAddress]::Broadcast),4000)
$packet = [byte[]](,0xFF * 102)
Write-Verbose "Packet : $Packet"
6..101 |% { $packet[$_] = $Target[($_%6)]}
Write-Verbose "Odesilam WakeON"
$UDPclient.Send($packet, $packet.Length)
}
```

### Význam skriptu:

1. Parametr funkce je MAC adresa cílového stroje, která je dále rozdělena na části a vložena do pole datového typu byte. Každá část je pak převedena z hexadecimální do desítkové soustavy.
2. Skript vytvoří socket s protokolem UDP, který se otevře pro broadcastové vysílání.
3. Skriptem je vytvořen „magický paket“, který obsahuje 102 bytů. Na prvních 6 bytech je převedená MAC adresa cílového stroje do desítkové soustavy



a na všech zbylých číslo 255. Jedná se o paket, který slouží ke spuštění funkce WakeOnLan.

4. Tento paket je broadcastově odeslán do sítě a stroj, který odpovídá cílové MAC adrese je spuštěn.

Část skriptu byla převzata z odkazu online [29].

### 5.2.5 TVORBA SKRIPTŮ

Veškerá místa, kde program používá skripty, jsou naprogramována a spouštěna na platformě **PowerShell**. Program má celkově dva předpřipravené skripty a tři další generuje za běhu. Dva předvytvořené jsou skript řídicí kopírování (minulá kapitola) a skript, který kopírování násilně ukončuje.

Další tři jsou vytvářeny hlavním programem na základě předem nashromážděných informací nastavených uživatelem. Jedná se o skript spouštějící kopírování, skript pro import souborů a skript pro změnu přístupových práv souborů.

**Skript spouštějící kopírování** využívá nastavených informací o fyzických strojích a souborech, které mají být kopírovány. Vytvořen je až po skriptech pro import a změnu přístupových práv. Oproti nim je však okamžitě spuštěn. Obsahuje seznam příkazů, z nichž každý spouští kopírovací skript s nastavenými parametry.

#### Příklad:

```
CopyScript.ps1 C:\Import \\147.228.67.104\VirtualClassImport
00-21-9B-69-B4-01 0 FALSE
CopyScript.ps1 F:\BP_ClassTestFiles\
\\147.228.67.104\TestCopy\ 00-21-9B-69-B4-01 1 TRUE
```

Význam jednotlivých parametrů pro soubor CopyScript.ps1 je popsán v předchozí kapitole „Realizace kopírování“.

**Skript pro import souborů** využívá informací o nastavení virtuálních strojů. U každého virtuálního stroje projde hlavní program dostupné proměnné a z nich vytvoří sekvenci příkazů využívající nástroj **VBoxManage**. Tím vznikne skript, který postupně nastaví veškeré virtuální stroje.

## Příklad:

```
vboxmanage createvm --name "Test 1" --register
vboxmanage modifyvm "Test 1" --memory 3242 --vram 32 --cpus 1
--boot1 disk
vboxmanage modifyvm "Test 1" --nic1 nat
vboxmanage modifyvm "Test 1" --cableconnected1 on
vboxmanage storagectl "Test 1" --name SATA0 --add sata --
bootable on
vboxmanage storageattach Test 1 --storagectl SATA0 -- type
hdd --mtype normal --medium „D:\Virtual\VHD\xp.vmdk“
vboxmanage import „D:\Virtual\Export\win7.ova“
```

Vytvoří dva virtuální stroje, jeden za užití virtuálního disku a druhý z exportovaného souboru. Více informací v kapitole 4.5 VBoxManage.

**Skript pro změnu práv souborů** využívá informací o nastavení kopírovaných souborů. Stejně jako skript pro import souborů je i zde na každé nové řádce přítomen pouze příkaz s atributy, tentokrát ale pro nástroj **icacls**. Více informací v kapitole 5.1.2 Změna přístupových práv souborů.

### 5.2.6 FUNKCE PROGRAMU

Hlavní část programu je realizována v jazyce JAVA. Jde o aplikaci s GUI, ve které uživatel nastaví informace o učebně a fyzických strojích, informace o souborech, které budou kopírovány a základní nastavení virtuálních strojů.

Aplikace je poté schopna spustit svůj hlavní automatizační proces, který vygeneruje skripty pro import virtuálních strojů do programu VirtualBox, pro změnu přístupových práv souborů a pro kopírování souborů na stroje v učebně.

Skripty jsou vytvořeny v jazyce PowerShell a jsou dávány dohromady metodami `vmImportScriptBuild()`, `caclsScriptBuild()` a `copyScriptExecute()` v hlavním programu v jazyce JAVA. Hned po svém vytvoření je spuštěn skript, který neobsahuje nic jiného než seznam příkazů spouštějících `CopyScript.ps1` s vybranými parametry.

Skript CopyScript.ps1 se dá považovat za modul, který je využíván programem automatické konfigurace. Během kopírování vytváří logové soubory o průběhu kopírování, které aplikace využívá pro zjišťování informací o průběhu kopírování. Tyto logové soubory jsou pouze dočasné a po skončení kopírování jsou smazány.

Skripty pro import a změnu práv souborů jsou kopírovány na každou stanici do sdílené složky „VirtualClassImport“, pro spuštění uživatelem po skončení kopírování.

Jakmile je kopírování dokončeno, zbývá už jen spustit vygenerované skripty na cílové stanici. Tím je konfigurace učebny dokončena.

### 5.2.7 ČÁSTI PROGRAMU

Program automatické konfigurace má dvě části a z větší je realizován v programovacím jazyce JAVA. Druhou částí jsou dva krátké skripty vytvořené v jazyce PowerShell, který slouží pouze pro ošetření kopírování souborů, a jsou využívány skripty, které jsou generovány první částí. Následující text poskytuje stručný rozbor každé z nich.

#### **Programová část v jazyce JAVA:**

Program se celkově skládá z 16 hlavních tříd, z nichž některé mají ještě své podtřídy, zde je jejich stručný přehled a popis některých proměnných:

- **TypeClass** – Datový typ, uchovávající informace o třídě s fyzickými stroji (název učebny, pole fyzických strojů)
- **TypeCopyFile** – datový typ, ukládající informace o souborech virtualních disku a souboru pro import pro VirtualBox, ve zvolené zdrojové cestě (pole souborů pro import, pole souborů virtuálních disků)
- **TypeImportDirectory** – Datový typ sloužící jako pomocný, při vyhledávání podsložek adresáře
- **TypeNetAdapter** – Datový typ uchovávající informace o virtuálním síťovém adaptéru (připojení síťového kabelu, typ adaptéru)
- **Type PC** – Datový typ uchovávající informace o fyzickém stroji (název PC, IP, MAC)

- **TypeStorageFile** – Datový typ uchovávající informace o kopírovaných souborech (zdrojová a cílová cesta, typ kopírování z lokálního nebo vzdáleného místa)
- **TypeVirtPC** – Datový typ uchovávající informace o nastavení virtuálního stroje (počet virtuálních procesorů, velikost paměti, použité virtuální disky nebo soubor pro import)
- **VirtualClass** – Klientská třída programu
- **WindowCopy** – Okno pro práci a nastavení kopírovaných souborů
- **WindowCopyProgress** – Okno pro zobrazení průběhu kopírování souborů na stroje v učebně
- **WindowGuide** – Okno s nápovědou, jak používat program
- **WindowIPSet** – Okno pro nastavení IP adresy fyzického stroje
- **WindowMAC** – Okno pro nastavení MAC adresy fyzického stroje
- **WindowMain** – Hlavní okno programu s nastavením vlastností fyzických strojů, menu programu, ovládacími prvky otevření oken pro práci s virtuálními stroji a kopírovanými soubory a tlačítkem spuštění hlavních automatických procedur
- **WindowProgress** – Okno zobrazení průběhu, pro místa čekání na vykonání akcí programu před postupem
- **WindowVM** – Okno pro přidávání, odebírání a nastavení virtuálních strojů

Rozsah LOC: 9949 řádků kódu

### Programová část v jazyce PowerShell:

Jedná se o skripty, které se starají o spuštění a ukončení kopírování. Skripty jsou uloženy v podsložce „Scripts“ adresáře s hlavním programem, soubory **CopyScript.ps1** a **CopyCancel.ps1**. CopyCancel obsahuje pouze dva řádky, které ukončují činnost procesů Robocopy a PowerShell.

```
stop-process -name robocopy
stop-process -name powershell
```

CopyScript je již popsán v kapitole 5.2.4 Realizace kopírování.

Rozsah LOC: 120 řádků kódu

### 5.2.8 OVĚŘENÍ FUNKCIONALITY

Program byl testován v laboratoři KIV ve čtyřech různých scénářích. První dva scénáře byly zaměřeny na kopírování z lokálního úložiště na vzdálený stroj. V druhých dvou scénářích byly soubory kopírovány v rámci vzdálené stanice. Jak lokální kopírování, tak kopírování v rámci vzdálené stanice bylo rozděleno ještě na případy, bez cizího zásahu a druhý na možnosti přerušování kopírování simulovaných vypnutím počítače.

V obou případech byly použity stejné soubory ke kopírování:

- Ubuntu.vdi (virtuální disk o velikosti 6,256 GB)
- Ubuntu.ova (exportovaný virtuální stroj o velikosti 2,596 GB)

Ve všech testovaných případech kopírování skončilo úspěšně. Rozdílem byl pouze čas, za který bylo provedeno.

V prvním případě, kopírování z lokálního úložiště na vzdálený stroj bez zásahu proběhlo kopírování s dobou trvání okolo 15 minut.

Druhý případ kopírování z lokálního úložiště na vzdálený stroj s vypnutím počítače. Stroj, na který bylo kopírováno, byl vypnutý celkem dvakrát a doba kopírování se prodloužila na 35 minut.

Třetí případ kopírování v rámci dvou adresářů vzdálené stanice bez zásahu proběhl s dobou trvání okolo 10 minut.

Čtvrtý případ kopírování v rámci dvou adresářů vzdálené stanice s vypnutím trval okolo 25 minut.

Ve všech případech Robocopy a PowerShell běží lokálně. Pokud tedy nedojde k vypnutí počítače, na kterém běží náš program automatické konfigurace VirtualBoxu, nebo fyzickému vypnutí síťového kabelu, který nebude zapojen do vypršení timeoutu Robocopy (celkově 30 vteřin x 50 opakování = 25 minut), mělo by kopírování proběhnout zcela bez problémů.

## 6 ZÁVĚR

Tato práce se zabývá virtualizací a nástroji pro virtualizaci na pracovních stanicích obecně. Popisuje nástroje pro virtualizaci firem VMware, Microsoft, Citrix a Oracle. Hlavním programem pro virtualizaci a jeho využití v praxi ve výuce na katedře KIV, který práce zkoumá je pak program VirtualBox.

Standardní přípravy výukového prostředí zaberou vyučujícím zbytečný čas. Během tohoto času musí každý počítač zvlášť nabootovat, nakopírovat soubory a importovat je do prostředí programu VirtualBox. Vytvořená aplikace tyto činnosti centralizuje na jedno místo, ve kterém uživatel pohodlně nastaví všechny parametry a v okamžiku, kdy je spokojen, jednoduše spustí automatickou konfiguraci učebny.

Před každou výukou je potřeba nastavit virtuální prostředí do počátečního stavu, a to stojí vyučující zbytečný čas a námahu. Z této skutečnosti vychází i má aplikace, která se snaží co nejvíce usnadnit úkony, které je potřeba pro přípravu učebny vykonat.

Vytvořená aplikace umí na základě dodaných informací automaticky kopírovat soubory na vybrané stanice a vytvořit pro ně skript pro import daných souborů do programu VirtualBox na cílových stanicích. Zároveň poskytuje ošetření kopírování proti vypnutí strojů, na které je kopírováno, dočasným výpadkům sítě a navazuje tam, kde bylo přerušeno. Rozsah práce představuje kolem 9950 řádků kódu v programovacím jazyce JAVA, plus dalších 120 v jazyce PowerShell.

Při tvorbě aplikace jsem kladl důraz také na jednoduchost v nastavování prvků virtuálních strojů a zahrnul pouze ty nejvíce podstatné (nastavení síťových adaptérů, velikost paměti, velikost video paměti, počet CPU, tvorba stroje pomocí souboru pro import nebo virtuálního disku). Program VBoxManage ale poskytuje sadu atributů ke kompletnímu ovládní programu VirtualBox. Vytvořená aplikace by tak v budoucnu mohla být rozšířena o další prvky, které nastavení VirtualBoxu nabízí (např. detailní správu virtuálních disků, vytvoření nových virtuálních strojů s nainstalováním z .iso souboru ze sítě, atd.), nebo které přijdou s novými verzemi.

Snažil jsem se, aby aplikace byla uživatelsky co nejpřívětivější, dobře ovladatelná a jednoduchá na používání. Pevně doufám, že se mi i tato kritéria podařilo splnit a aplikace bude využívána při přípravách na výuku v prostředí KIV.

## SEZNAM ZKRATEK A SLOVNÍK VÝRAZŮ

**API** – *Application Programming Interface* – rozhraní pro programování aplikací.

**JVM** – *Java Virtual Machine* - Sada počítačových programů a datových struktur, která využívá modul virtuálního stroje ke spuštění dalších počítačových programů a skriptů vytvořených v jazyce Java.

**NVM** - *.NET Virtual Machine* - Sada počítačových programů a datových struktur, která využívá modul virtuálního stroje ke spuštění dalších počítačových programů a skriptů vytvořených v jazycích .NET Framework.

**DPM** – *Distributed Power Management* - Způsob, jak monitorovat a přidělovat energetické zdroje virtuálním strojům na základě vytížení jednotlivých fyzických serverů celé serverové infrastruktury.

**DRS** – *Dynamic Resource Scheduling* - „Distribuované plánování přidělování zdrojů“ je technologie firmy VMware, která neustále monitoruje využívání zdrojů napříč fyzickými servery a na základě obchodních potřeb inteligentně přiděluje dostupné zdroje virtuálním strojům. Nejedná se však pouze o VMware, který tuto technologii poskytuje. Stejná řešení můžeme najít i u spousty dalších výrobců virtuálních technologií.

**VM** – *Virtual Machine* – virtuální stroj

**NAT** – *Network Address Translation* – překlad síťových adres

**CPU** – *Central Processing Unit* – processor

**LOC** – *Lines Of Code* – počet řádků kódu

**Image neboli obraz disku** - jedná se o jeden soubor nebo úložné zařízení obsahující kompletní obsah a strukturu datového úložiště, jako je například: pevný disk, disketa, optický disk, USB flash disk apod.

**ZDROJ:** Jak zálohovat systém? – image disku. *LevnaPC* [online; cit 22. 7. 2012], 2010. Dostupné z: <http://www.levnapc.cz/jak-zalohovat-system-image-disku.html>

**Virtuální stroj** - Jde o pomyslně vytvořené fyzické PC nebo server na softwarové úrovni.

**Hypervisor** - Jako hypervisor se označuje engine použitý ke spuštění hardwarové virtualizace. [4]

**Snapshot** - Při práci s virtuálními stroji je jako „snapshot“ označován soubor, který obsahuje stav, disková data a konfiguraci virtuálního stroje ve specifický bod v čase. Tento snapshot se poté dá využít k navrácení nastavení do stavu v tomto bodě. [5]

**Live migrate** - Jako migrate je v případě kdy pracujeme s virtuálními servery označován přesun virtuálního stroje mezi dvěma stroji fyzickými. „live“ neboli „živá“ pak dává najevo, že se jedná o migraci, během níž nedochází k výpadku běžícího virtuálního stroje. Jedná se o plynulý přesun, kdy je virtuální stroj přesunut a v moment, kdy je připraven a aktualizován o informace, které nastali během přesunu na prvním stroji, je první stroj odstaven a druhý pokračuje v jeho běhu. [8]



## SEZNAM CITOVANÉ LITERATURY A ZDROJŮ

- [1] Oracle VM VirtualBox – program sloužící k desktopové virtualizaci. Dokumentace a přehled. *Oracle* [online; cit. 22. 7. 2012], 2012. Dostupné z: <http://www.oracle.com/technetwork/server-storage/virtualbox/overview/index.html> , vlastní překlad.
- [2] SEDMÍK, T.: Virtualizace [online; cit. 22. 7. 2012]. Dostupné z: <http://www.fi.muni.cz/~xsedmik/xsedmik.html>
- [3] BERAN, R.: Virtualizace operačních systémů. [online; cit. 22. 7. 2012], 2012. Dostupné z: <http://www.beranr.webzdarma.cz/virtualizace.html>
- [4] DANIELLE RUEST, NELSON RUEST: *VIRTUALIZACE Podrobný průvodce*. 1. vyd., Brno: Computer Press, a.s., 2010 [cit. 2012-07-22], 408 s, ISBN 978-80-251-2676-9, s. 43.
- [5] About Virtual Machine Snapshots. *Technet.Microsoft, Microsoft* [online; cit. 22. 7. 2012], 2012. Dostupné z: <http://technet.microsoft.com/en-us/library/dd851843> , vlastní překlad.
- [6] DANIELLE RUEST, NELSON RUEST: *VIRTUALIZACE Podrobný průvodce*. 1. vyd., Brno: Computer Press, a.s., 2010 [cit. 22. 7. 2012], 408 s, ISBN 978-80-251-2676-9, s. 24
- [7] DANIELLE RUEST, NELSON RUEST: *VIRTUALIZACE Podrobný průvodce*. 1. vyd., Brno: Computer Press, a.s., 2010 [cit. 22. 7. 2012], 408 s, ISBN 978-80-251-2676-9, str. 25.
- [8] ROUSE, M.: Live migration. *SearchServerVirtualization* [online; cit. 22. 7. 2012], 2006. Dostupné z: <http://searchservirtualization.techtarget.com/definition/live-migration> , vlastní překlad.
- [9] VMware Distributed Power Management: Concepts and Usage. *VMware, Inc.* [online; cit. 22. 7. 2012], 2012. Dostupné z: <http://www.vmware.com/resources/techresources/1080> , vlastní překlad.
- [10] VMware vSphere™ Distributed Resource Scheduler (DRS) - Dynamic Resource Balancing. *VMware, Inc.* [online; cit. 22. 7. 2012], 2012. Dostupné z: <http://www.vmware.com/products/drs/overview.html> , vlastní překlad.

- [11] VMware products. *VMware, Inc.* [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://www.vmware.com/products/> , vlastní překlad.
- [12] VMware Workstation. *VMware, Inc.* [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://www.vmware.com/products/workstation/overview.html>  
, vlastní překlad.
- [13] VMware vSphere. *VMware, Inc.* [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://www.vmware.com/products/vsphere/mid-size-and-enterprise-business/features.html> , vlastní překlad.
- [14] VMware Fusion. *VMware, Inc.* [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://www.vmware.com/products/fusion/overview.html> ,  
vlastní překlad.
- [15] Datasheet VMware View Modernize the Desktop. *VMware, Inc.* [online], 2012.  
Dostupné z: <http://www.vmware.com/files/pdf/view/VMware-View-Datasheet.pdf>
- [16] BERAN, R.: Virtualizace operačních systémů [online; cit. 22. 7. 2012], 2006.  
Dostupné z: <http://www.beranr.webzdarma.cz/virtualizace.html#virtualPC>
- [17] Microsoft Virtual Server. *Microsoft* [online; cit. 22. 7. 2012], 2012.  
Dostupné z:  
<http://www.microsoft.com/windowsserversystem/virtualserver> , vlastní  
překlad.
- [18] Virtualizace: Virtual Server 2005 R2. *ASPNET.cz* [online], 2011.  
Dostupné z: <http://www.aspnet.cz/files/20060825-virtualserver-hq.png>
- [19] Microsoft Server and Cloud Platform, Hyper-V. *Microsoft* [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://www.microsoft.com/en-us/server-cloud/windows-server/hyper-v.aspx> , vlastní překlad.
- [20] Citrix, About. *Citrix Systems, Inc.* [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://www.citrix.com/about.html> , vlastní překlad.
- [21] NetWin, Virtualizace Citrix. *NetWin CZ, s.r.o.* [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://www.netwin.cz/citrix>
- [22] About Oracle Virtualization. *Oracle*, [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://www.oracle.com/us/technologies/virtualization/virtualization-strategy-wp-183617.pdf> , vlastní překlad.
- [23] VirtualBox Downloads. *Oracle*, [online; cit. 22. 7. 2012] 2012.  
Dostupné z: <http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html>

- [24] VirtualBox Documentation. *Oracle*, [online] 2012.  
Dostupné z: <http://www.oracle.com/technetwork/server-storage/virtualbox/documentation/index.html>
- [25] Best Free File Copy Utility. *Gizmo's Freeware*, [online] 2012.  
Dostupné z: <http://www.techsupportalert.com/best-free-file-copy-utility.htm>
- [26] Get to Know Robocopy for More Powerful File Management. *TechNet Magazine, Microsoft* [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://technet.microsoft.com/en-us/magazine/ee851678>, vlastní překlad.
- [27] Windows PowerShell. *TechNet, Microsoft* [online; cit. 22. 7. 2012], 2012.  
Dostupné z: <http://technet.microsoft.com/en-us/library/bb978526.aspx>, vlastní překlad.
- [28] Wake on LAN – lokální i vzdálený subnet. *Samuraj-cz.com* [online], 2008.  
Dostupné z: <http://www.samuraj-cz.com/clanek/wake-on-lan-lokalni-i-vzdaleny-subnet/>
- [29] Marc van Orsouw: The PowerShell Guy – PowerShell Wake-on-lan script. *MVP Windows Admin FrameWorks* [online; cit. 22. 7. 2012], 2007.  
Dostupné z: <http://thepowershellguy.com/blogs/posh/archive/2007/04/01/powershell-wake-on-lan-script.aspx>

## A. UŽIVATELSKÁ PŘÍRUČKA

Před používáním programu je nutné mít na paměti, že pro jakoukoliv novou učebnu je potřeba provést nejprve základní nastavení. Program byl testován a vyvíjen pro operační systém Microsoft Windows 7. Jeho užití je teoreticky možné i na starší verzi systému Windows XP s doinstalovanými nástroji PowerShell v2.0, Robocopy, iCacLS a JAVA, není však zaručena plná kompatibilita a funkčnost.

Pro operační systém Windows 7 stačí mít doinstalovanou pouze podporu jazyka JAVA, stáhnutelnou ze stránek Oracle:

<http://www.oracle.com/us/technologies/java/standard-edition/overview/>

Veškeré ostatní prvky systém obsahuje už ve svém základu.

### A.1 NASTAVENÍ SYSTÉMU

Systém musí mít nainstalovaný virtualizační program Oracle VM VirtualBox a přidanou cestu k jeho cílovému adresáři mezi systémovými cestami.

1. Stáhněte a nainstalujte program Oracle VM VirtualBox ze zdroje:  
<http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html#vbox>.
2. Klikněte pravým tlačítkem na ploše nebo v menu Start na Počítač a vyberte Vlastnosti.
3. V levém panelu vyberte Pokročilé nastavení systému.
4. V záložce Pokročilé klikněte na tlačítko Systémové proměnné.
5. Ve spodním okně nalistujte položku Path a klikněte na tlačítko Změnit.
6. Do pole Hodnota proměnné přidejte středník a cestu k adresáři s nainstalovaným programem VirtualBox, standardně:  
C:\Program Files\Oracle\VirtualBox.
7. Nastavte sdílené složky, které budete využívat ke kopírování souborů.  
Př. Nastavte sdílenou složku, která ponese název „VirtualClassImport“, do této složky se bude kopírovat soubor pro import a soubor pro změnu přístupových práv souborů.

8. Spustíte PowerShell jako Administrátor. V příkazové řádce napište PowerShell a potvrďte klávesou „Enter“ nebo napište PowerShell do Startovního menu systému, klikněte pravým tlačítkem a vyberte Spustit jako Administrátor.
9. Zadejte příkaz „Set-ExecutionPolicy RemoteSigned“ (Povolení spouštění PowerShellových skriptů).
10. Nastavte bootování počítače na systém Windows.

## **A.2 POVOLENÍ WAKEONLAN**

Pro možnost spouštění vypnutých počítačů automaticky a pro ošetření opětovného spuštění v případě vypnutí počítače omylem musí mít BIOS nastaven vlastnost WakeOnLan.

1. Restartujte počítač a spustíte nastavení BIOS.
2. Každý počítač se může systémem BIOS lišit, nejčastěji však lze najít povolení prvku WakeOnLan ve správě napájení (PowerManagement).
3. Povolení WakeOnLan musí být nastaveno na „Ano“.

## **A.3 MANUÁL PROGRAMU**

### **ZÁKLADNÍ POSTUP, JAK POUŽÍVAT PROGRAM**

Aby mohl program povolit spuštění hlavních procesů, je nutné, aby uživatel učinil nastavení potřebných proměnných. Správný postup používání programu:

1. Nastavení souborů ke kopírování.
2. Nastavení virtuálních strojů.
3. Vytvoření učebny a přidání do ní náležitých fyzických strojů s IP a MAC adresou.
4. Kliknutí na tlačítko "Spustit" v hlavním okně.

## **Nastavení souborů ke kopírování**

1. V hlavním okně klikněte na tlačítko "*Kopírování souborů*". Zobrazí se okno pro práci s kopírovanými adresáři.
2. Kliknutím na tlačítko "*Nakopírovat další složku*" se spustí průvodce přidáním nových kopírovaných souborů.
3. Chcete-li odebrat některé vybrané soubory ke kopírování, označíte je v listu okna kopírování a kliknete na tlačítko „*Odebrat složku*“.
4. Chcete-li změnit učiněná nastavení u již vytvořené složky, vyberte složku z listu a klikněte na tlačítko „*Nastavení*“.
5. Jakmile dokončíte výběr souborů ke kopírování, zavřete okno křížkem, v pravém horním rohu nebo tlačítkem "*Zavřít*".

## **Nastavení virtuálních strojů**

1. Poté co jste vybrali alespoň jednu složku se soubory pro kopírování, bude zpřístupněno v hlavním okně tlačítko "*Nastavení VM*". Po jeho kliknutí se otevře okno "*Nastavení virtuálních strojů*".
2. Kliknutím na tlačítko "*Přidat*" se spustí průvodce, který vás provede přidáním nového virtuálního stroje.
3. Pro podrobnější nápovědu o práci s virtuálními stroji vyberte v okně "*Nápověda->Jak používat program*", záložku "*Virtuální stroje*".
4. Virtuální stroj je možné kdykoliv odebrat nebo nastavit kliknutím na dostupná tlačítka v okně virtuálních strojů
5. Jakmile dokončíte nastavení virtuálních strojů, zavřete okno křížkem.

## **Nastavení učebny**

1. V hlavním okně přidejte novou učebnu položkou z menu pod horní lištou okna "*Učebna->Nová učebna*" nebo "*Učebna->Nahrát učebnu*" máte-li soubor pro import učebny z listu. (Pro více informací otevřete okno "*Nápověda->Jak používat program*", záložku "*Učebny*". Zvolíte-li možnost "*Nová učebna*" pokračujte krokem 2.

2. Jakmile bude v seznamu přidána alespoň jedna učebna, můžete do ní přidat fyzickou stanici zadáním IP adresy do čtyř polí v pravém rohu okna a kliknutí na tlačítko "*Přidat*". Po přidání IP adresy budete vyzváni k vyplnění fyzické adresy MAC, kvůli pozdější možnosti spuštění počítače pomocí funkce WakeOnLAN.

### **Spuštění hlavní funkce programu**

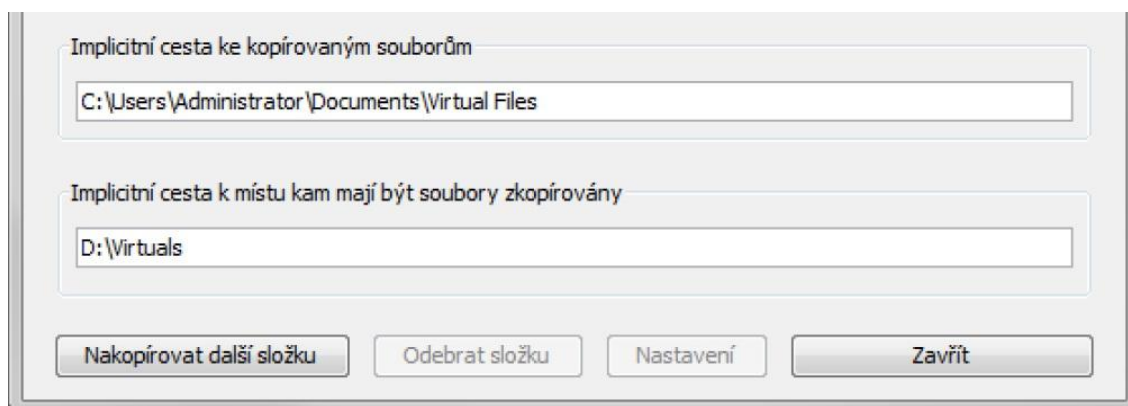
Jakmile budete mít splněny veškeré výše uvedené kroky a budete si přát spustit automatickou konfiguraci učebny, stiskněte tlačítko "*Spustit*" v hlavním okně programu.

### **NASTAVENÍ KOPÍROVÁNÍ SOUBORŮ**

Kopírování na vzdálené stanice probíhá pomocí sdílených složek nastavených na práva pro zápis pro skupinu „Everyone“. Skupina Everyone je zde uvedena, kvůli možnosti, že by učebnu na výuku chtěl připravovat kdokoliv. V případě, že bude učebnu nastavovat jen omezená skupina lidí, stačí nastavit práva pro zápis pro tuto skupinu.

Zároveň je nutností, aby sdílená složka na každém připravovaném stroji nesla stejný název. Jiný způsob se bohužel v učebně nedařilo aplikovat.

1. Do okna pro kopírování souborů se dostanete z hlavního okna pomocí tlačítka "*Kopírování souborů*".
2. Po jeho stisknutí se otevře okno, které slouží k veškerým nastavením, práci s kopírovanými soubory a zobrazuje seznam cest, které jste již vybrali nebo v budoucnu vyberete.
3. V jeho spodní části naleznete dvě pole pro vyplnění implicitní zdrojové a cílové cesty. Ty slouží pro případ, nastavení výchozích cest k souborům, abyste nemuseli pokaždé v průvodci přidáním novými soubory zadávat celou cestu znovu.



obr. A.1 – Nastavení implicitních cest kopírovacího okna

Tlačítkem "Nakopírovat další složku" spustíte průvodce přidáním nových souborů. Průvodce pro přidání nových souborů poskytuje dvě možnosti jak soubory přidávat:

### 1. Z lokálního úložiště na vzdálený stroj.

tzn. Vybrání absolutní cesty k adresáři, kde jsou skladovány soubory reprezentující virtuální disky (soubory ve formátu *.vhd*, *.vmdk*, *.vdi*, *.hdd*) nebo exportované nastavení programu Oracle VM VirtualBox (soubory ve formátu *.ovf*, *.ova*.) a cílové cesty (názevu sdílené složky) na vzdáleném stroji.

### 2. V rámci vzdáleného počítače.

Soubory jsou nakopírovány na disku na vzdálené stanici a jako zdrojová a cílová cesta se použijí cesty v rámci vzdálené stanice. (opět se jedná o sdílené složky)

V průvodci se setkáte s výběrem přístupových práv, která nejdou nastavit bohužel automaticky, ale je pro ně vygenerován skript, který se kopíruje spolu se skriptem pro import souborů do VirtualBoxu. Tento skript je účinný pouze v případě, že daný uživatel, který bude skript na cílovém stroji spouštět má dostatečná práva, aby práva u souborů mohl měnit (většinou vyučující). Práva, která se dají nastavit jsou:

- R - čtení (Read)
- W - zápis (Write)
- C - změna (Change)
- F - plná práva (Full control)



Po přidání nového adresáře do seznamu kopírovaných je nastavení reprezentováno podle vzoru v příkladu:

```
C:\Users\Milan\Desktop\Virtuals\ -> D:\Virtuals\ (Local, R)
```

1. Zdrojová cesta
2. Cílová cesta
3. Způsob kopírování
  - a. Local - z lokálního úložiště
  - b. Dist - v rámci vzdáleného počítače

Písmeno reprezentuje přístupová práva, tak jak jsou vidět o odstavce výš.

## NASTAVENÍ VIRTUÁLNÍCH STROJŮ

Přidat a nastavit nové virtuální stroje je možné pouze po kroku vybrání kopírovaných složek v okně kopírování souborů. Pro otevření okna nastavení virtuálních strojů stiskněte v hlavním okně tlačítko "*Nastavení VM*".

Pro přidání nového virtuálního stroje stiskněte v okně virtuálních strojů tlačítko "*Přidat*", které spustí průvodce přidáním nového virtuálního stroje.

Pro vytvoření virtuálního stroje v průvodci existují dvě možnosti:

### 1. Vytvoření stroje za pomoci virtuálního disku

U této možnosti mějte na paměti, že disk, který vyberete, bude sloužit jako hlavní bootovací disk virtuální stanice. Všechny ostatní nastavitelné parametry, které nebudou přítomny v průvodci, se dají nastavit v okně nastavení vybraného virtuálního stroje.

### 2. Vytvoření stroje za pomoci souboru vyexportovaného nastavení programu Oracle VM VirtualBox

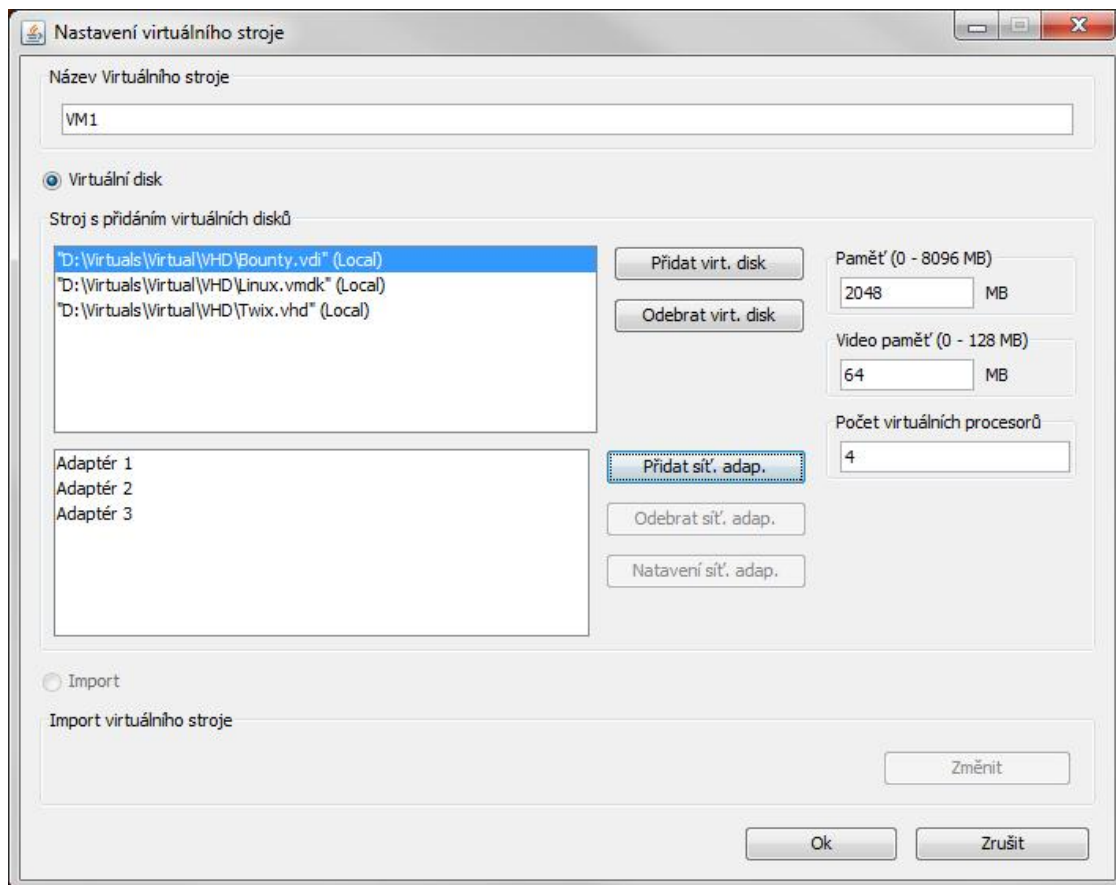
V tomto případě je možnost nastavení podstatně jednodušší než u práce s virtuálním diskem. Používá se soubor, který byl vygenerován při práci s programem Oracle VM VirtualBox. Jednoduše vyberete pouze soubor, který chcete importovat a nic dalšího již nenastavujete.

V průvodci přidáním nového virtuálního stroje zvolíte identifikační jméno, pod kterým bude virtuální stroj vidět v seznamu virtuálních stanic. Pokud se jedná o stroj, který bude využívat ke své tvorbě virtuálního disku, bude dané jméno použito pro stroj i v rámci finálního nastavení v programu Oracle VM VirtualBox.

V okně nastavení virtuálních strojů lze po vybrání již přidaného stroje, stroj odebrat nebo upravit jeho stávající nastavení pomocí tlačítek "*Odebrat*" a "*Nastavení*".

Po stisknutí tlačítka nastavení se v závislosti na předchozím nastavení stroje (virtuální disk nebo import), otevře okno kde je možné pro:

- možnost virtuálního disku, nastavit jméno, další virtuální disky (bootovací disk je pouze první ze seznamu), síťové adaptéry, počet procesorů a velikost operační a video paměti. Tyto parametry pak budou vloženy do tvorby daného virtuálního stroje při spuštění skriptu pro import na výsledném stroji.
- První list obsahuje pouze cílové cesty virtuálních disků, tak jak budou vypadat, po zkopírování tímto programem.
- Druhý list obsahuje seznam síťových adaptérů, které bude využívat virtuální stroj.
- možnost importu ze souboru, změnit soubor pro import.



obr. A.2 – Okno nastavení virtuálního stroje

## ULOŽENÍ/NAHRÁNÍ NASTAVENÍ

Ukládání a nahrávání nastavení je jednoduchá procedura, která je možná a dostupná k provedení kdykoliv během běhu programu.

### Uložení nastavení

1. V hlavním okně stiskněte položku v menu "*Program->Ulož nastavení*"
2. V okně pro zadání cesty vyberte lokaci, kam chcete soubor nastavení uložit a jak ho pojmenovat a stiskněte tlačítko "*Ulož*"

Program ukládá nastavení do textového souboru. Aby byly soubory nastavení programu nějak odlišeny od ostatních, jsou opatřeny příponou *.vcfs* (VirtualClass Save File).

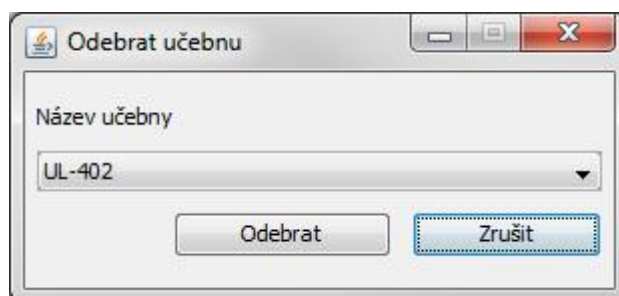
## Nahrání uloženého nastavení

1. V hlavním okně stiskněte položku v menu "*Program->Nahraj nastavení*"
2. V okně pro zadání cesty vyberte lokaci kde je soubor s nastavením, které chcete nahrát a stiskněte tlačítko "*Otevřít*"

## UČEBNY A PRÁCE S JEJICH NASTAVENÍM

Abyste mohli přidat do programu fyzické stanice, musíte nejdříve založit novou učebnu. Toho docílíte stisknutím položky v hlavním menu programu "*Učebna->Nová učebna*". Po stisknutí položky v menu zviditelníte okno, kam zadáte název učebny a potvrdíte akci stisknutím tlačítka "*Přidat*". Aby byl tento úkol snazší v případě zadávání více učeben se stejným začátkem názvu, je uchováván název naposledy přidané učebny.

Učebnu je možné i kdykoliv odebrat se všemi jejími stroji položkou v hlavním menu "*Učebna->Odebrat učebnu*". Po stisknutí položky v menu zviditelníte okno, kde bude možné učebnu odebrat z listu předem vytvořených učeben.



obr. A.3 – Okno odebrání učebny

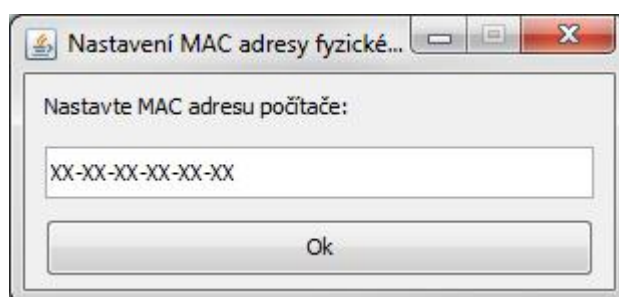
Vyberete učebnu a stisknete tlačítko "*Odebrat*".

Pokud máte alespoň jednu vytvořenou učebnu, můžete do ní přidat fyzické stroje pomocí polí IP adresy v pravém horním rohu hlavního okna programu a stisknutím tlačítka "*Přidat*". Aktivní učebna, s kterou pracujete je vidět nalevo od těchto 4 polí pro zadání IP adresy.



obr. A.4 – Náhled na část okna zadávání IP adresy

Po zadání IP adresy budete vyzváni k zadání fyzické adresy MAC, pro případ funkce WakeOnLAN, která počítač probudí z vypnutého stavu, kdyby byl někým vypnut nebo v první řadě nebyl nastartován vůbec.



obr. A.5 – Okno pro zadání MAC adresy

Jakmile je proces přidání fyzického stroje do učebny ukončen, je zobrazen pod názvem své IP adresy.

Stroj lze před spuštěním automatické konfigurace učebny označit tlačítkem „Vybrat“ a zakázat ke zpracování. Takto zakázaný stroj, na který nebude aplikováno nastavení je zobrazen v listu červeně.

Existuje však také možnost učebnu nahrát z již předem připraveného textového souboru, kde jsou stanice zobrazeny pod názvem, který vychází z formátu tohoto souboru.

## NAHRÁNÍ UČEBNY

Nahrání učebny zařídíte stisknutím položky v hlavním menu programu "Učebna->Nahrát učebnu", která otevře dialogové okno pro otevření souboru s nastavením učebny. Jedná se o textový soubor s vlastním formátováním, který aby byl odlišen od ostatních souborů používá příponu *.vclf* (Virtual Class List File).

Formát souboru je popsán na následujících řádcích:

### Příklad:

|                  |                |              |
|------------------|----------------|--------------|
| --UL-402         |                |              |
| ul402p07-kiv.fav | 147.228.67.107 | 00219B681A67 |
| ul402p08-kiv.fav | 147.228.67.108 | 00219B681A66 |
| <br>             |                |              |
| --UL-401         |                |              |
| ul401p07-kiv.fav | 147.228.67.101 | 00219B681A00 |

Soubor je rozdělen na název učebny, uvedený dvěma pomlčkami a dále pak na každé řádce obsahuje tři údaje oddělené libovolným počtem mezer:

1. Název fyzické stanice
2. IP adresu fyzické stanice
3. MAC adresu fyzické stanice

Učeben může být v listu i více, stačí použít formát, jaký je uveden v příkladu výše. Zadat název učebny a na dalších řádkách ho následovat údaji o jednotlivých fyzických strojích a tak opakovat pro každou učebnu.

## SPUŠTĚNÍ SKRIPTŮ NA STANICI PO KOPÍROVÁNÍ

Jelikož program nemá možnost spouštět skripty automaticky díky nedostatečným právům uživatele, je potřeba po dokopírování souborů na stanici spustit skript osobně.

1. Na cílové stanici spustíte PowerShellovou konzoli z nabídky Start nebo z příkazového okna příkazem PowerShell.
2. Do řádky napište příkaz „cd (název lokace vašeho sdíleného adresáře VirtualClassImport)
3. Pokud máte dostatečná práva na změnu práv nakopírovaných souborů, napište do řádky „rules.ps1“(změna přístupových práv souborů) a potvrďte.
4. Pokud práva nemáte nebo jste krok 3. již provedli, zadejte do řádky „import.ps1“ (import souborů do VirtualBoxu) a potvrďte.