



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH TENKÉHO KLIENTA

THIN CLIENT DESIGN

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FILIP JUHAŇÁK

VEDOUCÍ PRÁCE

SUPERVISOR

ING. VIKTOR ONDRÁK, PH.D.

BRNO 2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Filip Juhaňák

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh tenkého klienta

v anglickém jazyce:

Thin Client Design

Pokyny pro vypracování:

Úvod
Vymezení problému a cíle práce
Analýza současného stavu
Teoretická východiska řešení
Návrh řešení
Zhodnocení a závěr
Seznam použité literatury
Přílohy

Seznam odborné literatury:

AGRAWAL, Neeraj, et al. Practical Handbook Of Thin-Client Implementation. [s.l.] : New Age International, 2005. 232 s. ISBN 81-2241-685-3.

BIGELOW, Stephen J. Mistrovství v počítačových sítích : Správa, konfigurace, diagnostika a řešení problémů. Brno : Computer Press, 2004. 992 s. ISBN 80-251-0178-9.

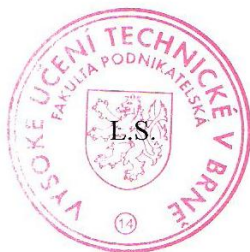
DOSTÁLEK, Libor; KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS . 5. aktualizované vyd. Brno : Computer Press, 2008. 488 s. ISBN 978-80-251-2236-5.


LIMING D., Sean. Windows XP Embedded Advanced. [s.l.] : RTC Books, 2003. 717 s. ISBN 09-2939-277-9.

MILLER P., Frederic; VANDOME F., Agnes; MCBREWSTER, John. Desktop Virtualization. [s.l.] : VDM Publishing House Ltd., 2009. 82 s. ISBN 61-3027-294-4.

Vedoucí bakalářské práce: Ing. Viktor Ondrák, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2010/11.




Ing. Jiří Kříž, Ph.D.
Ředitel ústavu


doc. RNDr. Anna Putnová, Ph.D., MBA
Děkanka

V Brně, dne 6.2.2011

Abstrakt

Tato bakalářská práce se zabývá návrhem hardwarové konfigurace a softwarového vybavení nového tenkého klienta, který má v nabídce společnosti OldanyGroup s.r.o. efektivně nahradit tenké klienty zahraničních výrobců, které společnost aktuálně nabízí svým zákazníkům. Součástí tohoto návrhu je také vývoj podpůrných programů a utilit.

Abstract

This bachelor thesis deals with designing of the hardware configuration and the software equipment of a new thin client device, which is designated to effectively replace the foreign-made thin clients in OldanyGroup company's product offering. The design also includes development of support programs and utilities.

Klíčová slova

Tenký klient, desktopová virtualizace, vzdálená plocha, PC over IP, Windows Embedded, VDI, VMware View, Microsoft .NET

Keywords

Thin client, desktop virtualization, remote desktop, PC over IP, Windows Embedded, VDI, VMware View, Microsoft .NET

Bibliografická citace této práce

JUHAŇÁK, F. *Návrh tenkého klienta*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2011. 66 s. Vedoucí bakalářské práce Ing. Viktor Ondrák, Ph.D.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně.
Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně, dne

.....

Podpis

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce, Ing. Viktoru Ondrákovi, Ph.D., za konzultace, rady a návrhy, kterými mi při tvorbě této bakalářské práce pomáhal. Rovněž bych rád poděkoval kolegovi Tomáši Andrlému za spolupráci na realizaci projektu a testování.

Obsah

1	Úvod.....	11
2	Vymezení problému a cíle práce	12
3	Analýza současného stavu	13
3.1	Informace o společnosti	13
3.2	Současný stav	13
3.3	Požadavky na návrh tenkého klienta.....	14
4	Teoretická východiska řešení.....	16
4.1	Tenký klient	16
4.1.1	Historie.....	16
4.1.2	Hardware.....	17
4.1.3	Software	17
4.1.4	Možnosti implementace tenkých klientů	18
4.2	Desktopová virtualizace a VDI	22
4.2.1	Virtualizace	22
4.2.2	Desktopová virtualizace, VDI.....	24
4.3	Protokoly RDP a PCoIP	25
4.3.1	RDP – Remote Desktop Protocol (11).....	25
4.3.2	PCoIP – protokol PC over IP	26
4.4	Přínosy VDI a tenkých klientů v podniku.....	27
4.4.1	Výhody a nevýhody řešení	27
4.4.2	TCO – Celkové náklady na vlastnictví.....	29
4.5	Modulární systémy Microsoft Windows Embedded.....	30
4.5.1	Základní vlastnosti	30
4.5.2	Edice Embedded systémů (7)	31
4.5.3	Vývojářské nástroje (2).....	32

4.6	Platforma Microsoft .NET	35
4.6.1	Microsoft .NET Framework	35
4.6.2	Microsoft Visual Studio.....	36
4.6.3	Jazyk C#.....	37
4.6.4	Jazyk VB.NET	37
5	Návrh řešení.....	38
5.1	Návrh HW konfigurace tenkého klienta	38
5.1.1	Volba komponent, dvě varianty HW	38
5.1.2	Testy stability.....	41
5.2	Operační systém tenkého klienta	44
5.2.1	Microsoft Windows Embedded – volba edice	44
5.2.2	Sestavení a konfigurace systému Windows Embedded.....	44
5.2.3	Automatizace nasazení systému	49
5.3	Podpůrné aplikace a utility.....	51
5.3.1	EWFCControl – aplikace pro ovládání funkcionality EWF.....	52
5.3.2	ReportSender – aplikace pro zasílání zpráv o chybách	53
5.3.3	Shell32 – alternativní uživatelské prostředí.....	55
5.3.4	Další doplňkové aplikace.....	56
5.4	Náklady a přínos projektu, srovnání cen.....	58
5.4.1	Náklady.....	58
5.4.2	Přínos pro firmu	58
5.4.3	Srovnání cen	59
5.5	Plánovaná a potenciální rozšíření.....	60
5.5.1	Software pro správu tenkých klientů	60
5.5.2	Možná vylepšení stávajících aplikací, odladění případných nedostatků ..	60
5.5.3	Přechod na systém Windows Embedded Standard 7	60

6	Zhodnocení a závěr	61
	Seznam použité literatury	62
	Seznam obrázků	64
	Seznam tabulek	64
	Seznam použitých zkratk	65
	Seznam příloh	66

1 Úvod

Informační technologie jsou v dnešní době nedílnou součástí prakticky jakéhokoliv podniku, bez ohledu na jeho obor, finanční prostředky či velikost. Závisí na nich nejen správné každodenní fungování podniku, ale také jeho budoucí rozvoj, schopnost udržovat vztahy se zákazníky a mnoho dalších důležitých vlastností. Aby toto vše bylo možné, jsou na informační zázemí podniku kladeny velmi vysoké nároky, a to jak na jeho spolehlivost a bezpečnost, tak i výkonnost, ale zejména na jeho efektivitu z pohledu přínosnosti pro podnik, se kterou přímo souvisí i finanční náročnost řešení této informační infrastruktury.

Existují různé možnosti a technologie, kterými lze všech výše zmíněných požadovaných kritérií dosáhnout. Jedním z moderních řešení je například možnost centralizace celé IT infrastruktury podniku, a to zejména v kombinaci s virtualizací a zavedením takzvané virtuální desktopové infrastruktury (VDI). Při správné implementaci těchto technologií lze docílit nejen výborné spolehlivosti, bezpečnosti, dostupnosti a rychlosti celého řešení podnikové IT infrastruktury, ale zejména pak snížit celkové náklady na vlastnictví díky nižším výdajům vynaloženým na údržbu a správu informačního zázemí firmy. Jistou nevýhodou zde samozřejmě mohou být vyšší pořizovací náklady, nicméně z dlouhodobého hlediska se zpravidla jedná o významné zlepšení (zejména z pohledu výše zmiňovaných vlastností) a efektivnější přístup k informačním technologiím a jejich správě v rámci podniku.

Aby bylo možné virtuální desktopovou infrastrukturu v podniku zavést, je třeba servery a klientské stanice virtualizovat. To následně umožní nahradit veškeré klientské pracovní stanice v podniku (obvykle běžná PC) menšími, úspornějšími a v neposlední řadě také levnějšími zařízeními, která budou namísto provádění veškerých výpočetních operací pouze zprostředkovávat přenos uživatelské relace. Takovéto zařízení nazýváme tenký klient. Návrh tenkého klienta (po hardwarové i softwarové stránce) pro nasazení ve virtuální desktopové infrastruktuře je předmětem této bakalářské práce.

2 Vymezení problému a cíle práce

Cílem této bakalářské práce je navrhnout tenkého klienta, kterého zařadí společnost OldanyGroup s.r.o. do svojí nabídky produktů (namísto aktuálně nabízených tenkých klientů zahraničních výrobců), zejména jako zařízení určené pro zákazníky, kteří se zajímají o řešení virtuální desktopové infrastruktury pomocí technologie VMware View, ale i pro jiné možnosti nasazení těchto zařízení, jako jsou terminálové služby, či jiná specializovaná řešení. Návrh bude odpovídat požadavkům společnosti (které vychází zejména z potřeb potenciálního zákazníka, zavádějícího desktopovou virtualizaci ve své firmě), a bude obsahovat volbu hardwarové konfigurace tenkého klienta, jeho softwarového vybavení, a následných přizpůsobení a úprav daného softwaru včetně vývoje doplňkových aplikací.

3 Analýza současného stavu

V této části práce zmíním základní údaje o společnosti, pro kterou je projekt realizován - čím se firma zabývá, stručně popíšu její charakteristiky, a následně nastíním, jak vypadá současný stav problematiky, a jaký je požadavek společnosti na výsledný výstup (respektive požadovaný produkt).

3.1 Informace o společnosti

Název: OldanyGroup s.r.o.

Právní forma: Společnost s ručením omezeným

Sídlo: Jaromírova 158/54

Praha 2 – Nusle

128 00

Předmět podnikání: Výroba, obchod a služby neuvedené v přílohách 1 až 3 živnostenského zákona

Společnost OldanyGroup působí na českém trhu od roku 2005, zabývá se moderními a inovativními technologiemi v oblasti ICT (informační a komunikační technologie), virtualizací s produkty společnosti VMware, komplexní správou systémů Windows a Linux, správou Cisco sítí a poradenstvím v těchto oblastech. Dále firma nabízí možnost outsourcingu kontinuální správy a monitorování systémů a sítí klientů, včetně podpory koncových uživatelů; firma také vyvíjí vlastní platformu pro poskytování služeb typu Cloud Computing. Firma je certifikovaným partnerem společností VMware, Microsoft, Cisco, DELL a dalších.

3.2 Současný stav

Společnost se v rámci své činnosti zabývá (mimo další činnosti zapsané v obchodním rejstříku) návrhem a poskytováním řešení desktopové virtualizace, správou sítí a systémů klientů. Pro tyto účely (zejména tedy pro účely desktopové virtualizace) je potom vhodným zařízením tenký klient, který v kombinaci s dobře realizovaným návrhem virtuální desktopové infrastruktury či terminálových služeb výhodně nahradí cenově, administračně i spotřebou energie náročnější PC.

V současné době společnost nabízí zákazníkům tenké klienty zahraničních výrobců, což není úplně vyhovujícím řešením, zejména vzhledem k vysokým cenám těchto zařízení (což může budít v zákaznících dojem zbytečně vysokých nákladů), déle trvá vyřízení případných reklamací, a také jsou zde menší možnosti, co se týče přizpůsobení a rozšiřitelnosti těchto zařízení na míru potřebám zákazníka či správce IT. Společnost se tedy rozhodla pro rozšíření své nabídky o vlastní HW tenkého klienta.

3.3 Požadavky na návrh tenkého klienta

Aby společnost mohla lépe vyhovět potřebám a požadavkům zákazníků, je třeba navrhnout a sestavit zařízení, které bude plně vyhovující (a v případě potřeby snadno upravitelné či rozšiřitelné) pro veškeré plánované a předpokládané možnosti nasazení.

Jedním ze základních požadavků jsou nízké výrobní náklady a tedy i příznivá cena pro koncového zákazníka. Nabídka tenkých klientů na trhu je poměrně rozsáhlá, a proto je třeba docílit co nejnižší možné ceny, a tím i lepší konkurenceschopnosti (zejména vzhledem k faktu, že produkt postrádá přidanou hodnotu známé a léty prověřené značky výrobce).

Dalším z požadavků je možnost rozšiřitelnosti a úpravy zařízení na přání zákazníka – může požadovat například větší operační paměť, jiný typ nebo velikost datového úložiště, adaptér pro připojení k bezdrátové síti, či jiné "nadstandardní" vybavení. Kvůli různorodosti požadavků by bylo vhodné navrhnout alespoň dvě hardwarové varianty zařízení, aby měl zákazník ihned na výběr z více možností, plus další případné úpravy, pokud by základní nabízené varianty nedostačovaly, nebo nesplňovaly jeho požadavky.

Velice důležitým požadavkem je také přizpůsobivost z pohledu softwarového vybavení a různé možnosti automatického nasazení či obnovení operačního systému tenkého klienta. Pro desktopovou virtualizaci je dále požadován návrh alternativního (zjednodušeného/omezeného) uživatelského prostředí v podobě speciální aplikace, a podpora protokolu PCoIP ve virtuální desktopové infrastruktuře s aplikací VMware View.

Mezi další požadované (ale zároveň volitelné) vlastnosti systému tenkého klienta potom patří podpora technologie Citrix HDX pro zákazníky, kteří se rozhodnou pro alternativní desktopovou virtualizaci s řešením společnosti Citrix Systems, plná podpora Microsoft Terminal Services pro možnost vzdálené relace s terminálovými službami

systemu Microsoft Windows Server, možnosti virtuální privátní sítě (VPN) pomocí základní podpory PPTP v systému Windows, případně i privátní sítě s Cisco VPN či Open VPN a podporu vzdáleného přístupu k systému pomocí VNC aplikace.

Zařízení by také mělo být možné v případě potřeby provozovat jako pracovní stanici se standardním uživatelským rozhraním a příslušenstvím obdobným jako v systému Windows XP, měla by tedy být v systému zakomponována podpora standardních aplikací systému Windows, internetu, multimédií, apod.

Důležitou součástí tohoto návrhu je také vytvoření podpůrných programů a utilit, které umožní automatické nasazení a konfiguraci systému, zjednoduší ovládání a zpřístupní uživateli jednoduché rozhraní pro odesílání zpráv o chybách.

4 Teoretická východiska řešení

V rámci této kapitoly se zaměřím na obecnou teorii tenkých klientů, jejich hardwaru a softwaru, desktopovou virtualizaci, možností nasazení a technologií s tím spojených. Stručně také zohledním ekonomický pohled na zavedení virtuální desktopové infrastruktury v podniku, klady a zápory takového řešení, a v druhé části kapitoly pak popíšu softwarové prostředky, technologie a nástroje, které budou pro realizaci návrhu a prototypu tenkého klienta použity.

4.1 Tenký klient

Tenký klient je pojem širšího významu, v této práci však pojednávám o tenkém klientovi jako o zařízení, které je ve své podstatě značně zjednodušená varianta PC, obsahující pouze základní hardware nutný pro funkčnost v rámci požadovaného prostředí (hardwarové prostředky se obvykle odvíjí od způsobu nasazení a použití těchto zařízení). Oproti běžnému PC postrádá například veškeré pohyblivé části (pevný disk, ventilátory, CD-ROM mechaniky apod.) či jakékoliv rozšiřující karty, díky čemuž je rozměrově mnohem menší a také levnější než běžné PC. Vzhledem k faktu, že uživatelské aplikace tenkých klientů běží zpravidla centrálně na terminálových serverech či ve virtuálních strojích, není u těchto zařízení obvykle vyžadován velký výkon (1).

4.1.1 Historie

Centralizace IT a provoz pracovních stanic na bázi terminálů je technologií již léta používanou – dnešní tenčí klienti jsou poměrně vzdáleným nástupcem někdejších textových terminálů, pomocí kterých se uživatelé připojovali k víceuživatelským Unix systémům - tyto byly provozovány na výkonných Mainframe počítačích. Postupem času se však tyto technologie značně zdokonalovaly – v 90. letech se začaly objevovat první terminály, které již poskytovaly grafické uživatelské rozhraní. S tímto však začínal nastávat problém nedostatečného výkonu sítí pro realizaci komplexnějších přenosů uživatelských relací, a technologie ustoupila do pozadí (značný vliv na tento útlum rozvoje měl však také výrazný pokles cen standardních PC). S vývojem počítačových sítí však jejich přenosové rychlosti rostly v řádu desetinásobků i více, díky čemuž se

v pozdějších letech stala myšlenka centralizace IT opět aktuální a zajímavou volbou. Dnešní sítě poskytují výkon pro tyto účely více než dostatečný (zejména pak uvažujeme-li podnikové LAN sítě s rychlostmi až 1Gb/s), a tato technologie tedy doznala další výrazný rozvoj a rozšíření. Současná řešení umožňují poskytnout koncovému uživateli prostřednictvím tenkého klienta plnohodnotný přístup k centrálně běžícím (často i virtualizovaným) operačním systémům a aplikacím, takřka srovnatelný s prací na lokálním PC (1).

4.1.2 Hardware

Od hardwaru tenkých klientů se vyžaduje v první řadě spolehlivost, která je klíčovým faktorem pro požadovanou vysokou dostupnost celého řešení a také pro delší životní cyklus samotného zařízení. Aby byla rizika hardwarových selhání minimální, neobsahují tenčí klienti žádné pohyblivé části (viz výše), které by se mohly běžným provozem (ale i neopatrnou manipulací) poškodit či opotřebovat. Díky tomuto faktu narůstá životnost a spolehlivost zařízení oproti běžnému PC (kde nejčastějším problémem bývá obvykle porucha pevného disku a ztráta lokálně uložených dat, nebo porucha ventilátoru a následné přehřátí). Velmi důležitým faktorem je zde ale také kvalita hardwaru – ta by měla být u tenkých klientů nejvyšší možná (ovšem s ohledem na cenu, která by měla být naopak co nejnižší – zde je třeba hledat vhodný kompromis) (4).

4.1.3 Software

Pro funkčnost tenkého klienta je nutné nasazení softwaru, který uživatelům umožní přístup k centrálně provozovaným operačním systémům, aplikacím a prostředkům serverů. Existují různé možnosti a řešení mnoha výrobců; mezi nejznámější z nich patří například následující (4):

- **Microsoft Terminal Services** – software umožňující vzdálený přístup k terminálovým relacím systému Microsoft Windows Server a jejich aplikacím pomocí protokolu RDP (Remote Desktop Protocol); tyto služby jsou běžnou součástí operačních systémů MS Windows.

- **VMware View** – software společnosti VMware, umožňuje přístup k virtualizovaným desktopovým systémům a jejich prostředkům běžícím na virtuální platformě VMware ESX. Toto spojení je zajištěno pomocí protokolů RDP nebo PCoIP (PC over IP).
- **Citrix XenApp/HDX** – řešení společnosti Citrix Systems, umožňuje připojení k uživatelským relacím pomocí protokolu ICA (Independent Computing Architecture).
- **Linux Terminal Server Project** – implementace terminálových služeb pro operační systémy Linux s využitím protokolu X11.
- **Další možnosti** – mimo výše zmíněné dnes existuje velké množství dalších možností a softwarových řešení pro tyto účely.

Tenký klient může (podle možností hardwaru a potřeb produkčního prostředí, ve kterém je nasazen) obsahovat i úložiště s vlastní lokální kopií operačního systému (například Microsoft Windows Embedded, speciálně upravená distribuce OS Linux, či jiné systémy), případně tento systém může být umístěn na serveru. V případě umístění systému na serveru pak musí tenký klient při spouštění nejprve pomocí prostředí PXE (Pre-boot Execution Environment) načíst systém do své operační paměti a poté jej teprve lokálně spustit (což ale skýtá značnou výhodu v rychlosti běhu OS z operační paměti zařízení a absenci datového úložiště v zařízení, je však silně penalizováno dobou načtení OS ze sítě).

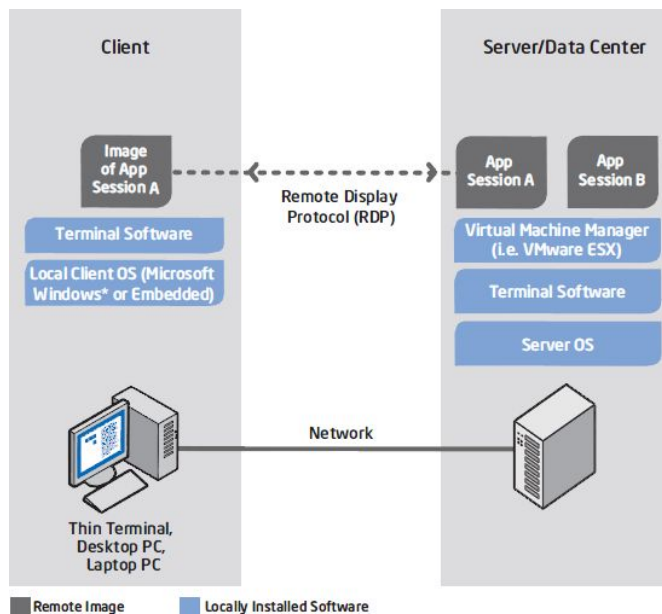
4.1.4 Možnosti implementace tenkých klientů

V praxi existuje více různých způsobů (respektive modelů), kterými lze tenké klienty v podnikové informační infrastruktuře nasadit a provozovat. Některé z nich budou popsány v rámci této podkapitoly.

Terminálové služby

Nejstarší, a časem prověřený model, který umožňuje přehlednou a efektivní správu a bezpečnost, z pohledu uživatele běžného PC však může být přechod na toto řešení nepraktický vzhledem k malým možnostem upravitelnosti prostředí, menší rychlosti a špatné mobilitě (nemožnost práce offline).

Princip: tenký klient zastává roli zobrazovacího a vstupního zařízení, veškeré výpočty a uložení dat probíhají na serveru. Pro přenosy je využíván například protokol RDP nebo ICA (5).

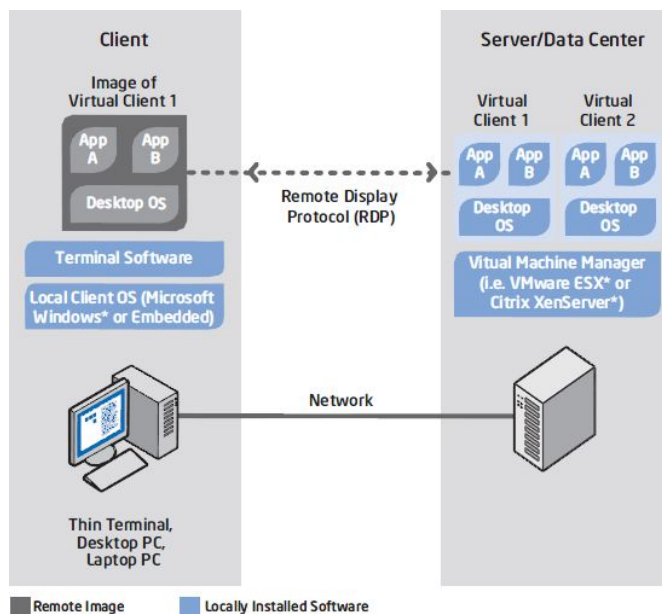


Obrázek 1 - Architektura terminálových služeb (5)

Desktopová virtualizace a VDI

Moderní model pro virtuální desktopovou infrastrukturu s poměrně rychlou odezvou (v rámci možností sítě), dobrou upravitelností OS a komfortem pro uživatele (velmi blízký používání vlastního lokálního systému stejně jako u PC). Výhodou je rovněž centralizovaná správa, bezpečnost, ale také možnost využití hardwaru tenkého klienta pro zlepšení akcelerace přenášeného obrazu a vzdálené připojení USB zařízení (toto platí pouze s protokolem PCoIP, který však skýtá i další výhody).

Princip: Všechny výpočty a ukládání dat probíhají na serveru (s výjimkou akcelerace videa u PCoIP protokolu), mezi tenkým klientem a serverem jsou přenášeny data obrazu a vstupů, případně data připojených USB zařízení. Oproti terminálovým službám má každý klient na severu vlastní virtuální stroj (nebo klonovanou instanci tohoto stroje), ke kterému se připojuje pomocí RDP či PCoIP protokolu (5).

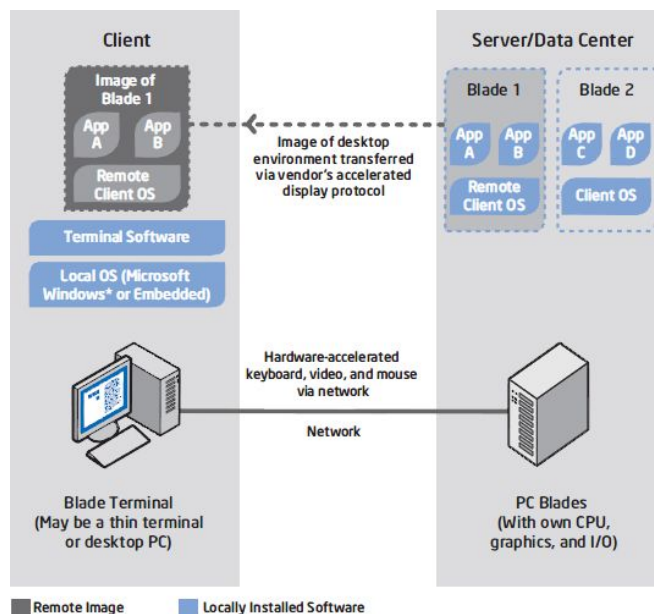


Obrázek 2 - Architektura desktopové virtualizace (5)

Blade PC

Blade PC architektura je kompromisem mezi hardwarovými PC a virtualizací – jednotlivé "Blade" jednotky (v podstatě plnohodnotné specializované počítače) jsou umístěny v datacentru a k těmto se připojují klientská zařízení (tenčí klienti). Tento přístup k implementaci tenkých klientů a centralizace je finančně velmi náročný, lze však dosáhnout vysokého výpočetního výkonu (bez rizika ovlivnění prostředků ostatních uživatelů v extrémně náročných aplikacích) při zachování centralizované správy a výborného zabezpečení.

Princip: Tenký klient slouží jako vstupní a zobrazovací zařízení, výpočty probíhají na Blade PC, data jsou ukládána na diskovém poli datacentra. Přenosy obrazu a ovládání mohou být realizovány pomocí RDP protokolu, případně s dedikovaným síťovým spojením (ve smyslu extra kanálu pro separátní přenos uživatelské relace) pro vyšší výkon a rychlejší odezvu. 1 Blade PC může využívat jeden nebo více uživatelů (tzv. One-to-One / One-to-Many architektura) (5).

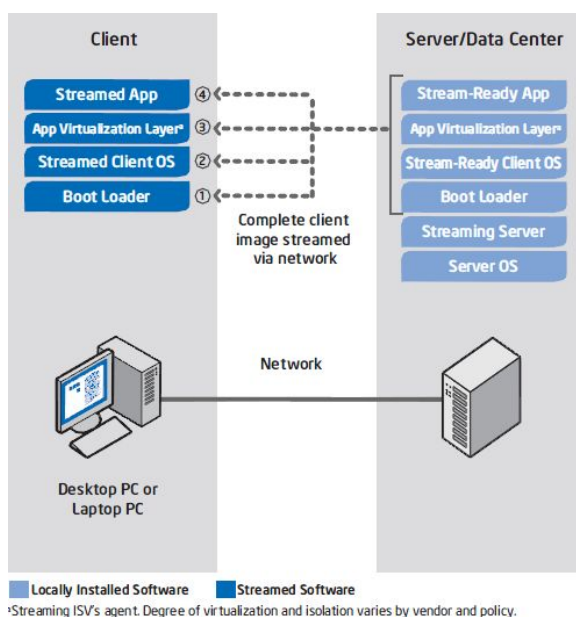


Obrázek 3 - Architektura Blade PC (5)

Streaming OS a aplikací

Tato architektura využívá tenkého klienta přímo ke zpracování dat, nejsou zde však ukládána. Výhodou je velmi rychlá odezva (díky lokálnímu běhu systému a aplikací).

Princip: Při spuštění se do paměti tenkého klienta načte operační systém a požadovaná aplikace. Práce s aplikací a OS probíhá na hardwaru tenkého klienta, data jsou však ukládána na serveru (5).



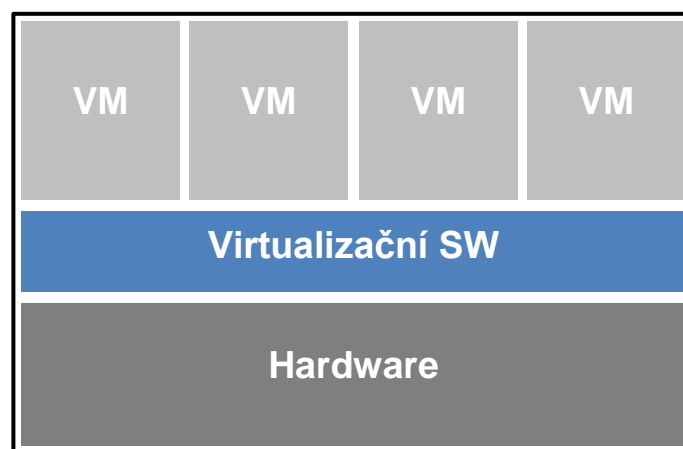
Obrázek 4 - Architektura streamování OS (5)

4.2 Desktopová virtualizace a VDI

Důležitým východiskem v této bakalářské práci je virtualizace, a to nejen serverů, ale zejména desktopových systémů. O této problematice a o virtuálních desktopových infrastrukturách pojednává tato podkapitola.

4.2.1 Virtualizace

Pojem virtualizace má poměrně široký význam, obecně však lze chápat jako vytvoření "virtuálního" ekvivalentu určitého reálného objektu. V informačních technologiích je pak virtualizace spojována převážně s abstrakcí a simulací hardwaru, kdy lze softwarovou část systému oddělit pomocí této technologie od fyzických prostředků stroje. To následně umožní například provoz více operačních systémů (a tedy logických počítačů) na jednom hardwarovém stroji (např. serveru). Tyto virtuálně spuštěné operační systémy poté fungují jako plnohodnotný počítačový systém – takovéto jednotky nazýváme virtuální stroje (virtual machine, VM) (3).



Obrázek 5 – Schéma virtualizovaného stroje (vlastní zpracování)

Virtualizace je moderní a technologie a trend ve vývoji informačních technologií, její počátky však sahají až do 60. let minulého století, kdy vznikly první experimentální projekty firmy IBM v této oblasti (3).

Dnes je virtualizace rozšiřována a zaváděna ve firmách zejména z důvodu konsolidace serverů (provozu více serverů v rámci jednoho fyzického stroje – vychází

z faktu, že běžně po většinu času je až 90% výkonu HW nevyužito), a následné úspory díky nižším nákladům na množství pořizovaného hardwaru (1 hardwarový server zastane díky virtualizaci funkci několika samostatných serverů) a z toho vyplývajících úspor energie a nižších výdajů na správu. Velkým přínosem je také možnost rozložení výkonu mezi více fyzických strojů a následná vyvážená distribuce plus možnost vysoké dostupnosti – moderní virtualizační technologie umožňují dynamicky reagovat na poruchy a výpadky serverů a automaticky přesouvat výpočty a aktivní virtuální stroje na jiné servery v reálném čase a prakticky bez výpadků.

Využití virtualizace

Virtualizace je velmi univerzální a praktická technologie s mnohými možnostmi využití a uplatnění. Mezi typické možnosti využití patří například:

- **Konsolidace HW serverů/stanic** – primární účel (viz výše)
- **Provoz nekompatibilních aplikací** – například XP Mode v systému Microsoft Windows 7 – nekompatibilní aplikace lze spustit ve virtuálním stroji se starší verzí systému, integrovaném v OS.
- **Testování a simulace** – ideální nástroj pro vývojáře aplikací či OS, možnost snadno simulovat či replikovat podmínky cílového prostředí (např. VMware Workstation, Microsoft VirtualPC, Sun VirtualBox)
- **Izolace od hostujícího OS** – v hostovaném virtuálním stroji lze spustit jakoukoliv aplikaci bez rizika dopadu na funkčnost či data hostujícího OS.

Druhy virtualizace (3)

Existují různé druhy virtualizace, které mají podle svých charakteristik rozdílné vlastnosti a využití. Tyto druhy jsou:

- **Emulace** – simuluje se celý HW, hostovaný OS může běžet v nezměněné podobě, je zde ale vysoká režie (např. Bochs, DOSEMU).
- **Virtualizace na úrovni OS** – hostovaný operační systém je provozován na úrovni kernelu (jádra) hostujícího OS, umožňuje izolovaný běh totožného operačního systému (např. FreeBSD Jail).

- **Aplikační virtualizace** – virtualizace obsahuje pouze základní části cílového systému, registry, knihovny či jiné systémové soubory (např. emulátor Wine v OS Linux)
- **Paravirtualizace** – virtualizace nesimuluje hardware, poskytuje však specializované API (aplikační programovací rozhraní); hostovaný OS musí být pro funkčnost upraven, výsledkem je potom velmi nízká režie (např. Xen).
- **Nativní virtualizace** – plná virtualizace, pouze částečná emulace hardwaru, hostovaný OS nemusí být upravován, je zde malá režie, vyžaduje však podporu přímo hardwarem (např. Microsoft Hyper-V, VMware ESX) (3)

Podpora HW virtualizace v procesorech

Pro zlepšení výkonu virtuálních strojů byly výrobci procesorů zavedeny technologie, které umožňují podporu virtualizace přímo v procesoru. Tyto funkce (Intel VT-x a AMD-V) jsou integrovány do většiny moderních procesorů a umožňují tak podporu rychlé plné virtualizace nejen u serverů, ale i běžných počítačů.

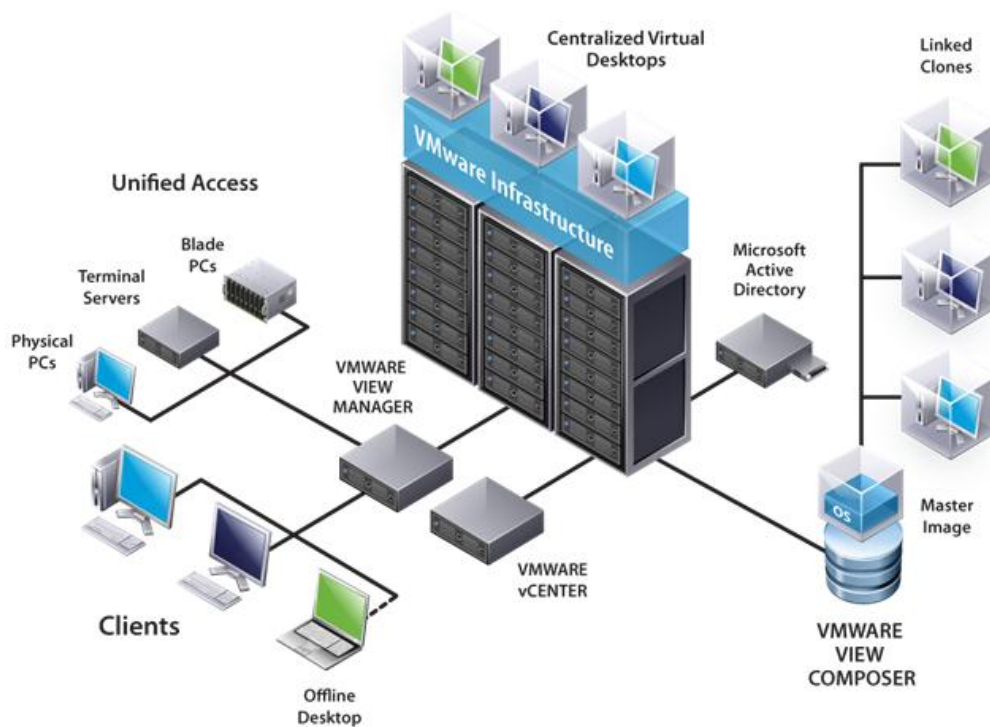
4.2.2 Desktopová virtualizace, VDI

Desktopová virtualizace je technologie, pomocí které lze oddělit operační systém a uživatelská data od klientské stanice, a namísto toho je umístit do bezpečného prostředí datacentra, kde poběží virtuálně na serveru (spolu s virtuálními stroji ostatních PC, případně i jako klonovaná instance jednoho výchozího stroje). Klientská stanice potom provozuje pouze tzv. "Connection Broker" (softwarové řešení umožňující přenos uživatelské relace mezi serverem a klientskou stanicí) a případně podřazený operační systém. V takovémto prostředí se provádí veškeré (resp. převážná většina) výpočetní úkony na serverech, a proto funkci pracovních stanic mohou zastávat tencí klienti nebo například i starší PC s výkonem již nedostatečným pro provoz požadovaných aplikací či operačních systémů (3).

Takovéto uspořádání prostředků a rozložení výpočetního výkonu se nazývá VDI (Virtual Desktop Infrastructure – virtuální desktopová infrastruktura), a z pohledu správy, bezpečnosti i nákladů poskytuje mnohé výhody (popsáno zvlášť v kapitole). Z pohledu architektury lze VDI nejvíce přirovnat k terminálovým službám – uživatel přistupuje ke svému "vzdálenému PC" obdobně jako k terminálu, avšak stále má svůj

"vlastní" (jakoby fyzický) systém a počítač, ovšem provozovaný virtuálně a na jiném místě.

Pro ilustraci: následující obrázek schematicky zobrazuje příklad, jak může vypadat jednoduchá virtuální desktopová infrastruktura realizovaná pomocí softwaru společnosti VMware.



Obrázek 6 - Virtuální desktopová infrastruktura s VMware View (12)

4.3 Protokoly RDP a PCoIP

Výsledný výstup této práce (tenký klient) je primárně určen pro nasazení ve virtuální desktopové infrastruktuře s řešením VMware View, proto zde nyní popíšu dva protokoly, které lze použít pro komunikaci koncových zařízení s virtualizovanými stroji uloženými na serveru.

4.3.1 RDP – Remote Desktop Protocol (11)

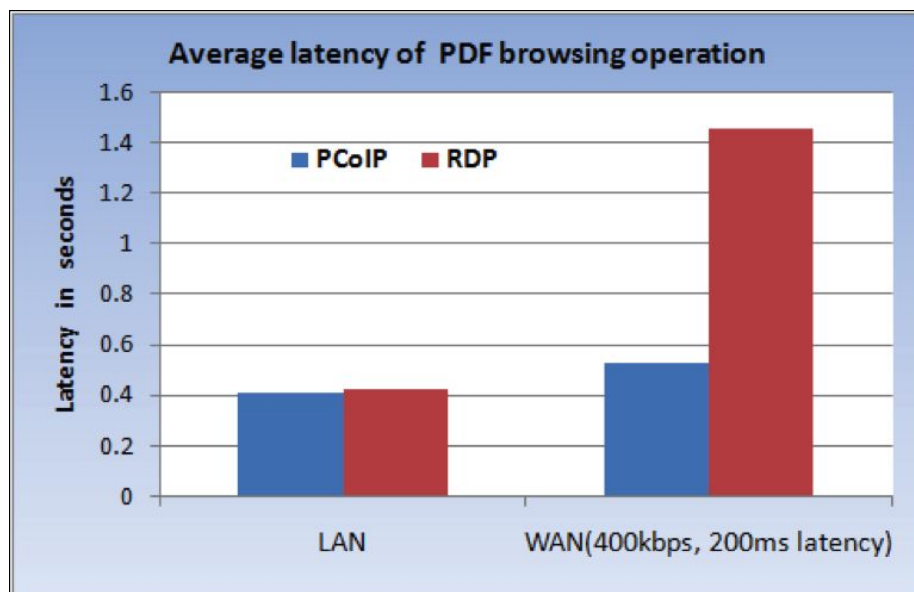
RDP je protokol navržený společností Microsoft, určený pro přenos grafického uživatelského rozhraní jednoho počítače na jiný, a umožnění vzdáleného přístupu.

Klient RDP protokolu je v základu součástí většiny operačních systému Microsoft Windows, je však dostupný i pro jiné operační systémy. Ve výchozím nastavení pracuje na portu TCP 3389. Serverová část protokolu se nazývá Microsoft Terminal Services (v poslední revizi byl již zaveden nový název – Remote Desktop Services) a byla poprvé zavedena v systému Microsoft Windows NT 4.0 Server.

Aktuální verze protokolu, 7.0, obsažená v systému Microsoft Windows Server 2008 a Windows 7 (avšak dostupná formou aktualizace i pro některé předchozí verze Windows), nabízí mimo dnes již zcela běžnou 32 bitovou barevnou hloubku, vzdálenou podporu tiskáren, audia a šifrovaného přenosu také nové funkce, jako například přesměrování multimédií, podpora více monitorů, akcelerace bitmap (pro zrychlení odezvy na pomalé síti) či podpora Aero Glass efektů v nových verzích OS Windows.

4.3.2 PCoIP – protokol PC over IP

Protokol PCoIP je stejně jako RDP určen pro zprostředkování vzdálené uživatelské relace mezi dvěma PC. Byl vyvinut společností Teradici, a existují hardwarové (PCoIP klientský procesor integrovaný např. ve speciálních monitorech) i softwarové implementace – protokol přejala společnost VMware a je implementován v nových verzích softwaru VMware View (verze 4 a vyšší). Protokol je optimalizován pro přenos obrazu s vysokým rozlišením a podporuje také vzdálené připojení USB zařízení, lokální akceleraci videa, a díky tzv. "Progressive Build" funkci umožňuje i rychlejší odezvu na pomalém spojení (viz. Obrázek 7 – srovnání s RDP při vykreslování PDF dokumentu na pomalé lince, studie VMware) (14).



Obrázek 7 - Progressive Build - srovnání PCoIP a RDP (14)

4.4 Přínosy VDI a tenkých klientů v podniku

Desktopová virtualizace a nasazení tenkých klientů v prostředí podniku je moderní, a při správném přístupu k aplikaci těchto technologií také efektivní způsob, jak realizovat informační infrastrukturu podniku. Má mnohé výhody, zejména pak po stránce úspor co se správy a spotřeby energie týče, ale také jej provází jisté nevýhody a není vhodný pro všechny druhy podniků (zejména pro velmi malé podniky nemá příliš velký smysl a přínos je obvykle minimální).

4.4.1 Výhody a nevýhody řešení

Následující tabulka poskytuje souhrnný pohled na základní kladné a záporné stránky při zavádění tenkých klientů a VDI v podniku, a potenciální důvody, proč se rozhodnout pro takovéto řešení. Tyto vlastnosti budou pak rozebrány dále.

Klady	Zápory
Centralizovaná správa	Vyšší počáteční náklady
Nízké náklady na údržbu	Odezva při uživatelské interakci
Vyšší spolehlivost klientské stanice	Nemožnost práce offline a mobility
Delší životní cyklus zařízení	Licencování
Lepší bezpečnost dat díky centralizaci	
Malá spotřeba energie	
Nižší TCO	

Tabulka 1 - Klady a zápory nasazení tenkých klientů (vlastní zpracování)

Centralizovaná správa, údržba a životní cyklus

Nasazením tenkých klientů se správa IT výrazně zjednodušuje – vše je možné spravovat centrálně pomocí speciálních nástrojů, problémy lze obvykle řešit rychleji, snadno lze instalovat nový software či aktualizace, a i vzdálená správa (ve smyslu outsourcingu IT) má v tomto případě lepší uplatnění. Údržba klientských stanic prakticky odpadá – v případě poruchy je tenký klient nahrazen bez jakéhokoliv dopadu na data uživatele (navíc tento úkon nevyžaduje čas kvalifikovaného pracovníka). Tímto se výrazně snižují personální náklady na správu a údržbu.

Životní cyklus zařízení je také delší – oproti PC, kde je vhodné provést obměnu hardwaru každé 2-3 roky, může být tenký klient využíván i dvojnásobnou dobu (teoreticky i více) a to zejména díky vyšší spolehlivosti a menším požadavkům na jejich výkon.

Bezpečnost dat

Velkou výhodou je bezesporu umístění veškerých dat na serverech – pokud jsou datacentra dostatečně zabezpečena, jsou pak data vystavena minimálním rizikům. V oblasti s běžným přístupem (kanceláře, technická pracoviště apod.) se nachází pouze tencí klienti bez jakýchkoliv podnikových dat, která by mohla být odcizena se zařízením, zničena jeho poruchou, či vystavena jiným hrozbám.

Úspora energie

Vzhledem k jednodušší hardwarové konfiguraci tenkého klienta a absenci některých komponent je spotřeba elektrické energie obvykle velmi nízká – následující tabulka

¹ TCO – Celkové náklady na vlastnictví, blíže popsáno v podkapitole 4.4.2

srovnává roční spotřebu tenkého klienta s PC (údaje jsou zde pouze orientační, počítají s určitou průměrnou spotřebou obou zařízení, provozem 8 hodin denně, 360 dní v roce a sazbou elektrické energie 4Kč/kWh).

	Tenký klient	PC	Rozdíl
Výkon zdroje	60 W	300 W	240 W
Průměrná spotřeba	45 W	120 W	75 W
Spotřeba za den (x 8h)	0,36 kWh	0,96 kWh	0,6 kWh
Spotřeba za rok (x 360d)	129,6 kWh	345,6 kWh	216 kWh
Roční spotřeba / úspora v Kč	518,40 Kč	1 382,40 Kč	864 Kč

Tabulka 2 - Srovnání spotřeby energie PC a tenkého klienta (vlastní zpracování)

Zápory

Naopak mezi záporné stránky tohoto řešení jednoznačně patří vyšší počáteční náklady (mohou i značně překročit cenu běžného řešení), které vyplývají z nutnosti nákupu výkonných serverů s dostatečnou kapacitou pro běh virtuálních strojů pro všechny pracovní stanice. Nemalou položkou jsou také licence softwaru, jelikož mimo virtualizační software a software pro podporu VDI musí navíc každá stanice mít svou vlastní přístupovou licenci a licenci operačního systému.

Značnou nevýhodou v některých situacích také může být absence možnosti mobility – vzhledem k umístění systému, dat a aplikací na serveru je práce offline (bez spojení se serverem) zcela nemožná. Jako problém se také může jevit odezva při práci se vzdáleným počítačem a tím snížený uživatelský komfort, ovšem díky rychlému spojení lokálních sítí a moderním softwarovým řešením je tento problém minimalizován.

4.4.2 TCO – Celkové náklady na vlastnictví

Analýza TCO (Total Costs of Ownership), tedy celkových nákladů na vlastnictví je jedním z prvků finanční analýzy, který může objektivně podporovat rozhodování o nových i již spuštěných projektech, souvisejících investicích, a poskytnout důležité informace zohledňující veškeré náklady, které s projektem souvisí. Tento ukazatel je velmi užitečný zejména při zavádění nových IT projektů.

"Výsledná hodnota TCO zahrnuje veškeré náklady, které musí provozovatel systému za určité období vynaložit na provoz. Nejedná se jen o pořizovací náklady, ale také o náklady na administraci, údržbu a opravy, školení, inovace apod. Celkové náklady na vlastnictví tak zahrnují všechny náklady vynaložené v průběhu celého životního cyklu provozovaného systému." (10)

Při rozhodování o nasazení VDI a tenkých klientů v podniku pak bude TCO při porovnání různých alternativ obsahovat zejména následující položky:

- Pořizovací náklady serverového a klientského hardwaru a licencí
- Náklady související s instalací a konfigurací
- Personální (kvalifikace) a časová náročnost správy IT (pro servery i pro klientské stanice)
- Plánované výdaje související s podporou koncových uživatelů (helpdesk, hotline)
- Údržba a servis celé IT infrastruktury (konfigurace, nasazování nového softwaru, řešení poruch atd.)
- Náklady na elektrickou energii
- Další náklady související s realizací projektu.

4.5 Modulární systémy Microsoft Windows Embedded

Rodina operačních systémů Microsoft Windows Embedded obsahuje systémy zaměřené na použití ve vestavěných zařízeních (odtud název "Embedded"). Takováto zařízení jsou obvykle jednoúčelová (může se jednat například o bankomat, mobilní zařízení, jukebox, kiosk či právě v tomto projektu navrhovaný tenký klient). Embedded systémy jsou flexibilní a umožňují úpravu přesně vyhovující cílovému zařízení a z toho vyplývající minimální velikost a nároky na HW (2).

4.5.1 Základní vlastnosti

Systémy Windows Embedded vychází z běžných PC operačních systémů společnosti Microsoft (aktuálně nabízené verze vychází z OS Windows XP a Windows 7), oproti nim jsou však značně upraveny a generalizovány. Veškeré součásti (od

primitiv, jako jsou dynamické knihovny a dílčí programy, až po komplexní systémové součásti) jsou rozděleny do komponent, z kterých vývojář zvolí, jaké systémové součásti bude jeho upravený operační systém obsahovat (tato volba je značně zjednodušena pomocí různých šablon dodávaných s nástroji), případně může přidat i vlastní navržené komponenty. Tímto lze dosáhnout minimálních velikostí systému, vhodných např. pro PXE zavádění systému, a úpravy systému vhodné přesně pro podmínky cílového produkčního prostředí.

Embedded systémy nejsou běžně prodejné, jsou však dostupné OEM (Original Equipment Manufacturer) výrobcům zařízení, kteří mohou poté systém prodávat jako součást svého zařízení. Licence systému je vázána na zařízení, není však (jako u běžných systémů Windows) symbolizována štítkem na zařízení, ale pouze licenčním klíčem vloženým výrobcem zařízení při kompletaci systému.

4.5.2 Edice Embedded systémů (7)

Existují různé edice a verze těchto systémů, které se vyvíjely ze standardních verzí systémů Windows. Mezi první takovéto systémy patřila Embedded verze systému Windows 3.x, určená zejména pro prodejní terminály. Později s příchodem Windows NT pak vznikl systém Windows NT Embedded a později XP Embedded. Nyní popíšu aktuálně používané edice tohoto operačního systému.

Windows Embedded CE

Edice CE (také nazývaná Compact) je jediná verze Embedded OS, která nevychází z desktopové verze Windows, ale je samostatně vyvíjeným systémem s vlastním jádrem a architekturou. Jedná se o systém určený pro mobilní zařízení typu Pocket PC či Smartphone, aktuálně ve verzi 6.0.

Windows Embedded Standard

Nejběžnější a nejvíce rozšířená varianta Embedded OS s nejširší možností využití od jednoduchých zařízení jako jsou automaty až po zařízení s plnou funkcionalitou PC. Aktuálně je tento systém dostupný ve dvou verzích:

- **Standard 2009**, který je rozšířenou a inovovanou verzí původního XP Embedded, vychází ze systému Windows XP s aktualizací SP3 a nabízí velmi podobné funkce a uživatelské rozhraní.
- **Standard 7** – odvozen od moderního systému Windows 7, nabízí nové funkce, kompatibilitu s novými zařízeními a také implementuje Microsoft .NET Framework ve verzi 3.5, je však značně náročnější na hardware.

Nadále je používána také varianta tohoto systému s názvem **XP Embedded** – předchůdce edice Standard 2009, která stále pro mnohé účely dostačuje, neobsahuje však aktualizaci SP3 a je již zastaralá (většina nástrojů, postupů a vlastností však zůstává zachována v novější verzi 2009).

Další edice

Mimo tyto základní a nejvíce rozšířené verze jsou dále k dispozici i tyto specializované edice Windows Embedded:

- **Enterprise** – systémy Windows XP for Embedded Systems a Windows Vista for Embedded Systems – tyto verze jsou prakticky shodné s běžnými PC verzemi Windows XP a Vista pouze se speciálními licenčními podmínkami.
- **POSReady** – odvozeny od XP Embedded, tyto systémy jsou určeny pro "Point of Sale" aplikace, tedy prodejní systémy, kiosky a automaty.
- **NAVReady** – edice Windows Embedded systému pro navigační zařízení.
- **Automotive** – verze pro automobilový průmysl určená pro zařízení jako jsou dotykové panely v automobilech a jiné související systémy.

4.5.3 Vývojářské nástroje (2)

V této části se zaměřím na nástroje určené pro přípravu, návrh a úpravy image systému Microsoft Windows Embedded 2009 Standard využívané v tomto projektu. Tyto nástroje může využívat pouze OEM výrobce zařízení, který zakoupil runtime licence tohoto operačního systému a licenci vývojářského balíku Windows Embedded Standard Toolkit. Samotný softwarový balík sestává z několika hlavních součástí:

- **Windows Embedded Studio** – soubor vývojových aplikací a doplňkových utilit pro návrh, kompilaci a úpravu systémové image (viz dále)
- **Databáze komponent** – repozitář obsahující veškeré systémové soubory vyžadované pro kompletní Embedded systému + SQL databáze pro správu a evidenci balíčků (podporován je přiložený Microsoft SQL Server 2005 Express a všechny vyšší verze)
- **Služby vzdáleného spuštění** – volitelnou součástí jsou také služby PXE a TFTP pro testování spuštění systému ze sítě.

Windows Embedded Studio obsahuje množství specializovaných aplikací a utilit pro vývojáře systému – zde popíšu ty nejdůležitější z nich, jejich hlavní funkce a vlastnosti.

Target Analyzer

Aby mohl být operační systém sestaven přesně na míru danému zařízení, pro které je určen, musí vývojáři nejprve získat detailní informace o cílovém systému, architektuře hardwaru a jednotlivých zařízeních přítomných v systému. K tomuto účelu slouží konzolová utilita Target Analyzer, dostupná ve dvou verzích – spustitelná z prostředí MS DOS (16-bitová aplikace) i Windows (32-bitová aplikace). Výstupem je potom kompletní výpis vlastností hardwaru přítomného v systému, zpracovaný formou XML (Extensible Markup Language) dokumentu. Tento výstupní dokument je následně vstupem pro další aplikace (Target Designer, Component Designer), které na základě tohoto seznamu prohledají databázi komponent a zvolí ovladače potřebné pro běh systému (chybějící ovladače je nutno do databáze doplnit).

Target Designer

Aplikace Target Designer je hlavní součástí celého balíku nástrojů – obsahuje uživatelské rozhraní pro tvorbu projektů konfigurace cílového systému. V tomto rozhraní lze provádět obecná nastavení image systému (jako například cesty k základním systémovým složkám, identifikační informace či licencování), volit komponenty, které mají být obsaženy a měnit jejich nastavitelné proměnné, a v neposlední řadě také kontrolovat závislosti mezi komponenty a kompilovat výslednou systémovou image. Součástí aplikace jsou také kategoriicky třízené seznamy

komponent, základní šablony pro různé typy cílových zařízení a v neposlední řadě také funkcionality pro import dokumentů aplikace Target Analyzer.

Component Designer

Pokud chtějí vývojáři přidat do systému vlastní ovladače či software, musí tento "uživatelský" obsah nejprve upravit do formy komponent, balíčků, které zařadí do databáze komponent dostupné pro ostatní aplikace vývojového studia. K tomuto účelu slouží aplikace Component Designer – umožňuje tvorbu nových položek databáze komponent, obsahujících složky či soubory, data registrů, závislosti na jiných komponentách a jiné údaje. Tyto položky lze potom kategoricky začlenit do databáze a použít v projektu. Mimo tuto funkcionality je zde navíc rozhraní pro asistovanou komponentizaci ovladačů – aplikace podporuje import ovladače standardu WDM (Windows Driver Model) formou *.inf souboru a automatické předpřipravení vlastností komponenty.

Component Database Manager

Component Database Manager je utilita pro správu celé databáze Windows Embedded Studia. Pracuje nejen s databází Microsoft SQL serveru, ale také s místním (nebo i vzdáleným) repozitářem systémových souborů. Umožňuje přehledný přístup ke všem součástem databáze, jejich vlastnostem, a na komponenty lze nahlížet z různých perspektiv (podle platformy, balíčku, skupiny apod.), mazat je, upgradovat či importovat nové. Pomocí této aplikace je také nutno importovat vývojáři vytvořené komponenty softwaru či ovladačů, než mohou být použity v ostatních aplikacích.

SDI Loader

Pro možnost síťového spuštění systému je také součástí nástrojů aplikace SDI Loader, která umožňuje tvorbu a emulaci diskových jednotek SDI (System Deployment Image). Takto vytvořené *.sdi soubory potom mohou být použity například pro PXE spuštění systému.

4.6 Platforma Microsoft .NET

Důležitou částí této práce je také programování podpůrných aplikací a utilit tenkého klienta, pro jejichž realizaci byly využity moderní jazyky a nástroje platformy Microsoft .NET. V této kapitole uvedu základní informace o této technologii, nástrojích a dvou programovacích jazycích v projektu využívaných.

4.6.1 Microsoft .NET Framework

Microsoft .NET Framework je softwarová platforma dostupná pro operační systémy Microsoft Windows vytvořená za účelem podpory tvorby funkčně bohatých aplikací, knihoven a webových služeb založených na XML. Jedná se o technologii obdobnou například platformě Sun Java.

Mezi hlavní přínosy této technologie patří zejména:

- Stejnorodé prostředí pro objektově orientované programování v různých jazycích s možností lokálního i vzdáleného spuštění kódu
- Pokročilé služby řízení běhu kódu a správy paměti
- Minimalizace náročnosti nasazení aplikací a konfliktů v jejich verzích
- Bezpečný běh kódu, zejména pak aplikací poskytnutých třetí osobou

.NET Framework se skládá ze dvou základních komponent: CLR (Common Language Runtime) – obecného jazykového modulu, a knihovny tříd (Class Library). Hlavním účelem CLR je řízení kódu za běhu aplikace – poskytuje základní služby jako je správa paměti, řízení vláken, dohled nad kompatibilitou datových typů (jazyky platformy .NET jsou silně typové), apod. Kód spouštěný prostřednictvím CLR potom nazýváme řízený kód. Knihovna tříd je rozsáhlá kolekce objektově orientovaných datových typů a struktur, které vývojáři mohou využívat při tvorbě svých aplikací. (8)

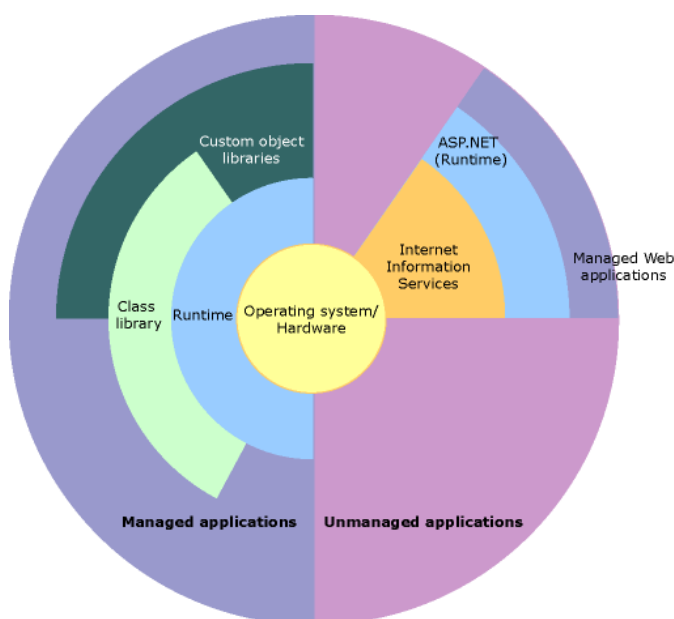
Verze a vývoj platformy .NET Framework

Od vydání první verze platformy (.NET Framework 1.0) v roce 2002 až k dnešnímu datu proběhlo mnoho aktualizací a přibylo mnoho funkcí. Zde je jejich zjednodušený chronologický přehled spolu se systémem, ve kterém byly poprvé uvedeny jako jeho součást:

Verze .NET Framework	Rok uvedení první verze	Součást systému
1.0 (SP1, SP2, SP3)	2002	Windows XP
1.1 (SP1)	2003	Windows Server 2003
2.0 (SP1, SP2)	2006	Windows Vista, Server 2008
3.0 (SP1, SP2)	2006	Windows Vista, Server 2008
3.5 (SP1)	2007	Windows 7, Server 2008 R2
4.0	2010	-

Tabulka 3 – Vývoj platformy .NET Framework (vlastní zpracování)

Následující obrázek schematicky znázorňuje začlenění funkcionality .NET do operačního systému a aplikačního prostředí.



Obrázek 8 - Schéma začlenění .NET framework v systémech (8)

4.6.2 Microsoft Visual Studio

Pro vývoj aplikací na platformě .NET jsou vývojářům dostupné různé nástroje. Mezi nejrozšířenější a nejkvalitnější patří Microsoft Visual Studio. Toto vývojové prostředí je široce využíváno již od svého prvního vydání. Postupem času (poprvé u verze s názvem "Visual Studio .NET") byla zavedena podpora a integrace s platformou .NET Framework, Visual Studio tedy umožňuje programování ve všech jazycích podporovaných touto platformou. Během následujících let vycházely aktualizace platformy a přibývaly podporované jazyky, čemuž byly nástroje přizpůsobeny a vydány

aktualizované verze. Poslední a nejaktuálnější verzí je Visual Studio 2010, které plně podporuje platformu .NET Framework verze 4.

Samotné vývojové prostředí sestává z mnoha částí, které umožňují vývojářům efektivní práci s kódem a přehledný návrh aplikace, služby, knihovny či webu. Mezi hlavní části patří tyto prvky:

- **Editor** – hlavní část vývojového prostředí, mimo editaci kódu usnadňuje programování pomocí funkce IntelliSense (integrace dokumentace v editoru)
- **Debugger** – sofistikované metody a nástroje pro odchyťávání chyb a výjimek v kódu za běhu aplikace.
- **Designer** – grafické nástroje pro návrh formulářů, webů apod.
- **Další nástroje** – mimo tyto základní prvky obsahuje Visual Studio celou řádku dalších nástrojů a utilit a také podporu uživatelských doplňků.

4.6.3 Jazyk C#

Jazyk C# je objektově orientovaný programovací jazyk vyvinutý společností Microsoft. Vychází z jazyků C++ a Java, s kterými sdílí velmi podobnou syntaxi i některé vlastnosti. Jedná se o typově bezpečný jazyk, propojený s Microsoft .NET Framework platformou – výstupem je tedy řízený kód s výhodami služeb CLR (správa paměti apod.). Díky široké kolekci tříd platformy .NET (Class Library) umožňuje tento jazyk programování s vysokou efektivitou a produktivitou – programátor se již nemusí zabývat tvorbou základních struktur, tříd, datových typů či rutin. Jazyk také podporuje XML komentáře, ze kterých lze potom snadno generovat plnohodnotnou dokumentaci v různých formátech (13).

4.6.4 Jazyk VB.NET

Jazyk Visual Basic .NET je modernizovaná verze jazyka Visual Basic, propojená s platformou Microsoft .NET Framework. Původní jazyk byl rozšířen mnoha novými funkcemi a konstrukty, jejichž výsledkem je objektově orientovaný programovací jazyk s možnostmi dědičnosti, tvorby rozhraní či přetěžování. Stejně jako C#, i VB.NET využívá obecný jazykový modul CLR, který mu poskytuje služby řízeného kódu a další výhody, rovněž podporuje XML dokumentaci a těží z výhod knihovny tříd .NET (9).

5 Návrh řešení

V této kapitole rozeberu jednotlivé kroky, postupy a rozhodnutí, které postupně vedly k vytvoření prototypu, a k následné realizaci návrhu požadavků odpovídajícího tenkého klienta. Zaměřím se zejména na volbu optimální hardwarové konfigurace pro tohoto zařízení a s tím spojené úvahy, návrh a sestavení vyhovujícího operačního systému, a veškeré související úpravy a návrhy, co se software tenkého klienta týče, včetně programování podpůrných aplikací a utilit.

5.1 Návrh HW konfigurace tenkého klienta

Jak již bylo zmíněno v požadavcích na tenkého klienta, je třeba navrhnout hardwarovou konfiguraci, která bude nejen spolehlivá, ale také cenově dostupná a ideálně i výkonnější než typický tenký klient – zejména z důvodu univerzálnosti a různých předpokládaných možností nasazení tohoto tenkého klienta. Z běžných charakteristik těchto zařízení rovněž vyplývá, že by výsledné zařízení mělo být (v rámci možností) kompaktní.

5.1.1 Volba komponent, dvě varianty HW

Nyní rozeberu, jaké HW komponenty byly vybrány pro návrh tenkého klienta, jejich základní vlastnosti a popřípadě i důvod jejich volby. S přihlédnutím k požadavkům na návrh je nutné vytvořit 2 hardwarové varianty zařízení – základní varianta s nejnižší možnou cenou (označená domluveným názvem „VECTOR“), a pokročilá varianta s vyšším výkonem a většími možnostmi rozšiřitelnosti (označená názvem „MATRIX“).

Základní deska a CPU

Pro obě varianty tenkého klienta bude použita kvalitní základní deska standardu microATX (rozměr 170x170mm) od společnosti Intel s čipovou sadou NM10 Express a integrovaným procesorem Intel Atom (avšak v rozdílných modelech pro požadované 2 varianty – viz souhrn v tabulce 4). Výhodou tohoto řešení je zejména spolehlivost (společnost Intel garantuje dlouhou životnost a rozšířenou záruku) a komplexnost – celek je přímo navržen pro tzv. „diskless“ model nasazení či vestavěné systémy, navíc obsahuje také efektivní pasivní chlazení.

Pro variantu VECTOR bude použit Intel Atom D410 - 1-jádrový procesor s podporou technologie Hyper-Threading (efektivně 2 vlákna CPU), frekvencí 1.66GHz a spotřebou 10W na desce Intel D410PT. Varianta MATRIX bude osazena vyšší edicí základní desky – Intel D510MO s výkonnějším procesorem Atom D510 disponujícím 2 jádry a rovněž podporou HT technologie (efektivně 4 vlákna CPU), frekvencí 1.66GHz a spotřebou energie 13W.

Operační paměť

V základní konfiguraci budou obě varianty tenkého klienta osazeny 1 modulem operační paměti 1GB DDR2 frekvence 667MHz – tato kapacita i rychlost jsou pro plánované účely nasazení dostatečné, v případě potřeby však bude tuto paměť možno rozšířit (obě verze základní desky mají 2 paměťové sloty a podporu až 4GB RAM).

Grafický adaptér, síťový adaptér, zvuková karta

Obě varianty tenkého klienta budou využívat grafický čip Intel GMA3150, který je integrován na základní desce. Výkon tohoto grafického čipu je dostatečný nejen pro běžnou práci, ale i základní akceleraci videa (potřebnou v tomto případě zejména pro relace protokolu PCoIP) a základní 3D aplikace – podporovány jsou API DirectX 9.0c a OpenGL 1.5. Adaptér využívá sdílenou grafickou paměť. Součástí základní desky je také zvukový čip Intel High Definition Audio, a to v 5.1 a 7.1 kanálové verzi (což vyplývá z rozdílu mezi zvolenými základními deskami). Připojení k síti je zajištěno integrovaným gigabitovým Ethernet adaptérem Realtek, 100Mbit adaptér u levnější varianty základní desky.

Úložiště dat

Tenký klient navrhovaný v rámci této práce bude mít svůj lokální operační systém, proto je nutné zabývat se i volbou vhodného úložiště dat. Toto bude (zejména kvůli ceně) řešeno pro obě varianty rozdílně.

Varianta VECTOR bude osazena interním 4GB USB flashdiskem, který je cenově nejvýhodnější přípustnou volbou (avšak za cenu pomalejšího zápisu i čtení), zatímco varianta MATRIX bude využívat 4GB CompactFlash kartu připojenou pomocí CF-to-SATA redukce – toto řešení je ve všech ohledech rychlejší, ale cenově méně příznivé.

Oba zvolené typy úložiště jsou však citlivé na časté zápisy a přepisování, které by mohly negativně ovlivnit jejich životnost a výkon – tento problém bude v systému ošetřen zavedením ochrany EWF (Enhanced Write Filter) – popsáno dále.

Skříň a napájení

S ohledem na formát základní desky, pasivní chlazení všech prvků systému, požadované kompaktní rozměry zařízení a v neposlední řadě i cenu byla pro obě varianty zvolena skříň Thermaltake Mini-ITX 3310 (obrázek viz příloha 1), která vyhovuje nejen těmto požadavkům, ale mimo jiné také ve své ceně zahrnuje 60W externí zdroj, který je pro napájení této konfigurace více než dostatečný.

Volitelné doplňky

K oběma variantám bude dostupné již výše zmíněné rozšíření operační paměti až na 4GB, vyšší kapacita datového úložiště (Flash či CompactFlash o velikosti až 32GB, teoreticky i více) a možnost uchycení k monitoru pomocí VESA Mount držáku. Do varianty MATRIX (jejíž deska disponuje potřebným rozhraním) bude také možno nainstalovat Mini PCI Express kartu Intel Pro/Wireless pro podporu bezdrátových sítí. Následující tabulka vyobrazuje orientační souhrn zvolených HW komponent obou variant tenkého klienta a jejich základních vlastností.

Varianta tenkého klienta	VECTOR	MATRIX
Základní deska	Intel D410PT Packton	Intel D510MO Mount Olive
Procesor	Intel Atom D410 (1.66GHz)	Intel Atom D510 (1.66GHz)
	HT, 1 jádro, 2 vlákna	HT, 2 jádra, 4 vlákna
Operační paměť	1GB DDR2 (667MHz)	1GB DDR2 (667MHz)
Datové úložiště	USB flashdisk, 4GB	CompactFlash (SATA), 4GB
Grafický adaptér	Intel GMA 3150	Intel GMA 3150
Zvuková karta	Intel HDA (5.1)	Intel HDA (7.1)
Ethernet adaptér	10/100Mb/s	10/100/1000Mb/s
Volitelné příslušenství	Operační paměť až 4GB	Operační paměť až 4GB
	Datové úložiště až 32GB	Datové úložiště až 32GB
	VESA Mount	VESA Mount
		Mini PCI Express Wi-Fi karta

Tabulka 4 - HW konfigurace variant tenkého klienta (vlastní zpracování)

5.1.2 Testy stability

Před jakýmkoliv dalším postupem je nutno sestavit prototypy obou HW konfigurací a provést zátěžové testy, které prověří především stabilitu zařízení ve vysoké zátěži, ale také schopnosti pasivního chlazení udržet přípustnou teplotu během náročného provozu.

Prime95

Pro otestování stability systému a účinnosti chlazení jsem se rozhodl použít aplikaci Prime95 – tato aplikace je primárně určena pro vyhledávání nových Mersennových prvočísel, a je schopna maximálně vytížit systém, čímž mohou být odhaleny případné potíže se stabilitou či přehřívání. Přímo v aplikaci je možno spustit tzv. „Torture test“ režim (zátěžový test systému) s různými možnostmi nastavení. Pro účely testování tenkých klientů jsem zvolil profil realizující kalkulace algoritmem s velkými FFT transformacemi (rychlá Fourierova transformace). Tento test vede (podle informací v aplikaci uvedených) k maximální zátěži procesoru, který pak generuje nejvyšší teploty a spotřebu energie (takovýchto hodnot při běžném provozu zařízení nikdy zdaleka

nedosáhne) a testuje částečně i paměť RAM. Aplikace poběží pod systémem Windows XP.

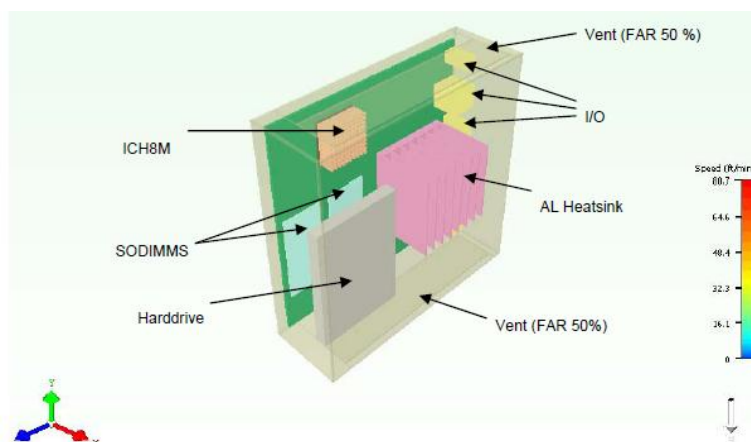
Cílové teploty a doporučení (6)

Pro referenci a zjištění přípustných teplot naměřených při testech systému vycházím z dokumentu Thermal Design Guide společnosti Intel pro procesory Intel Atom N450, D410 a D510. Tento dokument popisuje tepelné vlastnosti těchto procesorů a doporučení související s jejich chlazením a instalací. Tyto procesory jsou zde také označeny jako „vhodní kandidáti“ pro systémy s pasivním chlazením. Následující tabulka obsahuje výtah důležitých limitních hodnot, na které je třeba brát ohled.

Vlastnost	Hodnota (°C)
Maximální teplota CPU	100
Teplota vypnutí CPU	125
Nejvyšší vhodná okolní teplota	55

Tabulka 5 - Limitní hodnoty procesorů Atom (zprac. podle 6)

V dokumentu je mimo jiné také definována doporučená orientace chladiče vzhledem k větracím otvorům skříně a doporučené vlastnosti proudění vzduchu. Instalací základní desky do zvolené skříně (viz výše – volba HW konfigurace) jsou mnohá tato doporučení již splněna. Navržené řešení je rozvržením i vlastnostmi velmi podobné v rámci dokumentu popisovanému systému, který byl použit pro studii proveditelnosti pasivního chlazení procesorů Atom (viz následující obrázek).



Obrázek 9 - Rozvržení komponent ve skříní - reference (6)

Výsledky testů

Na obou prototypech tenkého klienta jsem provedl několikahodinový zátěžový test pomocí Prime95 (po jehož dobu bylo vytížení všech vláken CPU 100%), přičemž jsem zaznamenával data o teplotách pomocí aplikace Core Temp. Výsledky jsou stručně shrnuty v následující tabulce.

Varianta HW	Teplota CPU v klidu (běh OS) (°C)	Maximální teplota CPU v zátěži (Prime95) (°C)	Teplota ve skříni v klidu (°C)	Teplota ve skříni v zátěži (°C)
VECTOR	47	54	36	45
MATRIX	46	62	37	48

Tabulka 6 - Teploty naměřené při zátěžovém testu (vlastní zpracování)

Při testech nenastaly u tenkých klientů žádné potíže se stabilitou, a i přes 100% vytížení CPU, po celou dobu testu operační systém odpovídal. Výsledné naměřené teploty dokazují, že chlazení je dostatečně efektivní a k přehřátí nedojde ani při velmi náročném provozu. Rovněž se potvrzuje tvrzení z referenčního dokumentu – procesory jsou dobrými kandidáty pro pasivní chlazení – při dodržení doporučeného rozvržení lze jejich teploty bez problému udržet v bezpečných hodnotách.

Grafy teplot v průběhu doplňujícího, 1-hodinového stress-testu obou HW variant, jsou vyobrazeny v příloze č. 2.

Dodatečné testování

Mimo zátěžový test jsem prototypy tenkých klientů také testoval několikadenní prací v nainstalovaném systému Windows XP a to jak provozem relací MS Terminal Services a VMware View, tak i provozem aplikací na lokálním systému (práce s MS Office apod.). Během práce rovněž nedošlo k žádným potížím se stabilitou či přehříváním komponent.

Z předchozího vyplývá, že navržené HW konfigurace jsou plně funkční a stabilní, je tedy možné přejít dále k návrhu softwarového řešení tenkého klienta.

5.2 Operační systém tenkého klienta

Zde se budu zabývat volbou vhodného operačního systému, jeho sestavením, úpravami a možnostmi automatizace nasazení. Jak vyplývá ze stanovených požadavků na tenkého klienta, je vyžadována zejména podpora VMware View, MS Terminal Services, VPN a standardních aplikací a příslušenství systému Windows, čímž se volba značně zjednodušuje – společnost Microsoft nabízí pro tyto účely speciálně navrženou rodinu operačních systémů Windows Embedded.

5.2.1 Microsoft Windows Embedded – volba edice

Operační systémy MS Windows Embedded existují v mnoha variantách pro různé možnosti nasazení (viz teoretická východiska řešení). V případě tohoto návrhu je však vyžadována zejména běžná funkcionality Windows PC, dobrá stabilita, přizpůsobitelnost a ideálně i malá velikost systému. Těmto kritériím nejlépe vyhovují edice Standard – zde je na výběr ze dvou verzí: Standard 2009 (vycházející z operačního systému Windows XP) a Standard 7 (vycházející z moderního systému Windows 7).

Vzhledem k výše navržené hardwarové konfiguraci zařízení, stanoveným požadavkům a v neposlední řadě ceně, jsem se rozhodl pro volbu systému Windows Embedded Standard 2009, který je plně vyhovující aktuálním požadavkům (v budoucnu pak může být v případě potřeby zavedena i robustnější verze Standard 7).

5.2.2 Sestavení a konfigurace systému Windows Embedded

V této podkapitole stručně shrnu a popíšu kroky, ze kterých sestával návrh a kompletace operačního systému Windows Embedded Standard 2009 (dále jen WES 2009) pro navrhovaného tenkého klienta.

Analýza cílového HW

Aby bylo možné navrhnout a sestavit systém, který bude na cílovém zařízení korektně fungovat, je třeba nejprve provést analýzu tohoto zařízení, a získat tak potřebné údaje o hardwaru a prostředcích přítomných v systému, a jejich vlastnostech –

na základě těchto údajů potom lze potom v nástrojích Windows Embedded Studio připravit balíčky s komponentami ovladačů (viz dále).

Analýzu jsem provedl na obou verzích tenkého klienta (pod pro testovací účely nainstalovaným systémem Windows XP s aktuálními ovladači všech zařízení) pomocí utility Target Analyzer, jejímž výstupem je .pmq soubor, obsahující XML strukturu se získanými informacemi.

Příprava ovladačů

Prvotním krokem při sestavování image WES 2009 systému je zajištění dostupnosti ovladačů pro všechna zařízení, která má navrhovaný systém obsloužit – pro všechna zařízení musí být v databázi Embedded Studia komponenta s ovladačem. Pokud tomu tak není, je třeba chybějící ovladače nejprve dodat.

Pro zjištění chybějících ovladačů jsem použil aplikaci Component Designer (jedna z hlavních součástí Embedded Studia) – importováním analýzou HW získaného .pmq souboru se zde vytvoří nová makro-komponenta (tj. komponenta Embedded systému obsahující pouze závislosti na jiných komponentách – kontejner pro další komponenty), která obsahuje všechny v souboru popisované ovladače zařízení. Chybějící ovladače jsou vypsány formou varování, které obsahuje údaj o chybějícím ovladači zařízení (ve formátu jeho hardwarového ID - např. *PCI\VEN_8086&DEV_A001*), podle kterého lze dohledat a identifikovat, pro které zařízení ovladač v databázi chybí.

Po identifikaci chybějících ovladačů jsem vytvořil jejich komponenty – k tomuto účelu slouží rovněž aplikace Component Designer – importem .inf souboru požadovaného ovladače je vytvořena nová komponenta s vlastnostmi ovladače (z inf struktury jsou parserem přečteny údaje pro zápis do registrů systému, vytvářené služby, servery apod. a nastaveny jako patřičné vlastnosti nové komponenty) a odkazy na potřebné soubory (tyto bylo nutné dodat manuálně formou repositáře). Následuje přidání komponenty a jejího repositáře do balíčku, který je nutno importovat do databáze Embedded Studia pomocí aplikace Component Database Manager. Tímto je chybějící ovladač doplněn.

Závěrem této fáze sestavování operačního systému je import makro-komponent s ovladači do databáze, čímž je možno přejít k dalším krokům. Vzhledem k podobnostem v hardwaru obou konfigurací tenkého klienta jsem se rozhodl vytvořit

pouze 1 společnou makro-komponentu HW konfigurace, která obsahuje všechny potřebné ovladače pro obě varianty, tím pádem vznikne 1 systém vhodný pro oba tenké klienty.

Volba komponent OS

Operační systém WES 2009 vychází ze systému Windows XP SP3 – jedná se o velmi podobný systém, avšak je atomicky rozdělen na strom primitiv, komponent a kontejnerů – tzv. makro-komponent obsahujících závislosti na jiných komponentách, díky čemuž mohou vývojáři sestavit systém přesně na míru potřebám a s minimální možnou velikostí. Volba komponent je nejdůležitějším krokem při sestavování systému.

V základu sestává každý Embedded systém z několika výchozích komponent (které však závisí na velkém množství dalších součástí). Zde popíšu, jaké varianty těchto výchozích komponent jsem zvolil a jejich základní vlastnosti nebo účel.

- **HAL komponenta** – z termínu Hardware Abstraction Layer – součást jádra systému, která zajišťuje abstrakci nad hardwarem. Tato komponenta je závislá na cílovém HW, proto její volba vycházela čistě z údajů získaných při analýze zařízení – automaticky byla zvolena verze *ACPI Multiprocessor PC* (pozn. kvůli funkci Hyper-Threading považuje operační systém Windows i 1-jádrový Atom D410 za více jádrový procesor).
- **Shell komponenta** – tato komponenta určuje, jaké uživatelské prostředí bude image systému obsahovat; mezi dostupné možnosti patří *Command shell*, *Task Manager shell* a *Explorer Shell*. Protože je žádoucí, aby systém měl standardní grafické uživatelské rozhraní, zvolil jsem variantu *Explorer Shell*, která reprezentuje toto prostředí (pozn. příkazový řádek i správce úloh budou v systému obsaženy také, avšak ne formou primární Shell komponenty).
- **Jazykové komponenty** – lokalizace systému a regionální nastavení (např. rozložení klávesnice, jednotky, speciální znaky, časové zóny apod.). Jako primární volbu jsem vybral komponentu *Czech Language Support*, image však bude obsahovat i sekundární volbu s anglickým jazykem a klávesnicí. Přidání podpory dalších jazyků či rozložení klávesnice do projektu je kdykoliv později možné.

- **Bootloader komponenta** – zavaděč systému, výchozí volbou je *NT Loader*, zavaděč používaný systémem Windows XP. Pro účely tohoto projektu jsem však zvolil namísto základního zavaděče zavaděč *EFW NTLDR*, který vychází z původního NT Loaderu, avšak podporuje technologii Enhanced Write Filter (popsáno dále).
- **Komponenty souborových systémů** – tyto komponenty definují, které souborové systémy budou v OS podporovány, je možno přidat více těchto komponent. Přestože primárním souborovým systémem tenkého klienta bude NTFS, zvolil jsem (zejména kvůli podpoře USB flash disků či externích CD mechanik) všechny dostupné systémy, tedy *NTFS, CDFS, FAT*.
- **Logon komponenta** – komponenta zajišťující zabezpečené přihlašování uživatelů do systému a dohled nad jejich právy, použita výchozí (vyžadovaná prostředím Explorer Shell) – *Windows Logon*.

Mimo tyto základní a povinné komponenty musí každý Embedded systém obsahovat také výše zmiňované komponenty ovladačů zařízení, zjištěné analýzou HW. Jejich zavedení do systémové image spočívá pouze v přidání předpřipravené makrokomponenty s ovladači do projektu – jednotlivé ovladače se potom automaticky přidají díky závislostem v celé struktuře systému.

Takto sestavený systém je již provozuschopný, avšak jedná se pouze o základní konfiguraci s minimální funkcionalitou. Aby systém vyhovoval požadavkům, bylo nutné manuálně přidat mnoho dalších komponent a maker. Tyto jsou podrobněji (avšak stále ve značně zestručněné podobě) rozepsány v příloze č. 3.

Enhanced Write Filter (EWF)

Aby byl software tenkého klienta (OS, aplikace, nastavení) uživatelem neměnitelný, použil jsem ve Windows Embedded systému nabízenou funkcionalitu Enhanced Write Filter (EWF). Tato součást systému zajistí, že všechny změny provedené v systému budou přesměrovány a zapsány do vyhrazeného prostoru v operační paměti namísto zápisu na disk (případně lze zápisy směřovat i na jiný oddíl disku, z dále popisovaných důvodů jsem se však rozhodl pro variantu RAM). Tím je zajištěno, že s každým restartem zařízení poběží systém v kompletně původní a nezměněné podobě.

Administrátor (či systémový účet s potřebným oprávněním) může změny, které provedl v případě potřeby zapsat na disk pomocí konzolové utility **ewfmgr** (pro tyto účely bude sloužit jednoduchá utilita) – buffer s provedenými zápisy je přenesen na lokální systémový disk a změny tak zůstanou zachovány i při dalším spuštění.

Tuto funkcionalitu jsem v systému zavedl také z důvodu šetření životnosti lokálních úložišť tenkých klientů. Obě varianty tenkého klienta jsou vybaveny médiem citlivým na časté zápisy, jejichž poškození je takto prakticky zamezeno díky minimalizaci četnosti zápisů a přepisování filtrem EWF.

Uživatelské účty, povinný uživatelský profil

V systému budou ve výchozí konfiguraci zavedeny 3 uživatelské účty:

- **Administrator** – typický účet lokálního administrátora s plným oprávněním a plnohodnotným rozhraním Explorer Shell.
- **Service** – skrytý služební účet pro účely automatizace některých procesů, zejména při nasazování a správě tenkých klientů (rovněž s plným oprávněním)
- **User** – běžný uživatelský účet s omezeným oprávněním a alternativním uživatelským prostředím ve formě speciální Shell aplikace (popsáno zvlášť v kapitole). Tento účet bude mít navíc tzv. "povinný profil" – uživatel tedy může provádět změny v rámci svého oprávnění, ty však budou po odhlášení ztraceny.

Pozn. Nastavení povinného uživatelského profilu bude provedeno automaticky při instalaci systému, a spočívá ve vykonání následujících kroků:

1. Přejmenování souboru s uživatelskými registry (v profilovém adresáři) z *ntuser.dat* na *ntuser.man*
2. Nastavení vlastnictví celého adresáře uživatelského profilu na skupinu *Administrators*
3. Zakázání změny hesla pro daný profil

Licencování systémů WES 2009

Při sestavování systému je důležitým krokem také zajistit jeho licencování. Oproti běžným desktopovým Windows systémům je licencování Windows Embedded systémů odlišné – nejsou zde unikátní sériová čísla pro každou kopii, štítky pro umístění na

zařízení, ani online aktivační mechanismy. OEM distributor (výrobce zařízení a sestavitel systému) má vlastní runtime klíč, který při sestavování vloží pomocí nástroje Target Designer do image systému – tímto je zajištěna licence systému a jednoznačně spojena s OEM distributorem. Pokud by v systému tento klíč vložen nebyl, fungoval by systém v režimu časově omezené testovací verze.

5.2.3 Automatizace nasazení systému

Operační systém tenkého klienta by mělo být možné velmi snadno a rychle nainstalovat, konfigurovat, či provést reinstalaci – proto je nutná automatizace tohoto instalačního procesu.

Příprava systémového oddílu úložiště

Image WES 2009 systému je pouze adresářová struktura se soubory systému Windows. Aby bylo možno systém spustit, je před nakopírováním těchto souborů na systémový oddíl nutno oddíl připravit. Pro tyto účely jsem použil programy **Diskpart** a **Bootsect**, jejichž ovládání lze snadno automatizovat. Je nutno provést následující kroky:

1. Nastavit oddíl jako primární a aktivní (v programu Diskpart příkazy *create partition primary*, pro vybrání oddílu příkazy *select disk/partition*, aktivní oddíl *active*).
2. Naformátovat oddíl systémem NTFS.
3. Zavést v oddílu master boot code pro NTFS + NT Loader (v programu Bootsect pomocí parametru /nt52 a identifikace cílového svazku).

Pozn.: Tyto kroky budou při nasazení systému provedeny automaticky pomocí skriptu.

First Boot Agent (FBA)

Stejně jako systém Windows XP (od kterého je WES 2009 odvozen), i tento operační systém je nutno na cílový stroj nainstalovat. To je zajištěno programem First Boot Agent, který se spustí automaticky při prvním startu a provede přípravu systému. Tento proces je plně automatický a probíhá v číslovaných fázích 0-20000, během kterých jsou provedeny instalace ovladačů, registrace služeb, uživatelských účtů, apod.

Při sestavování systému lze určit také vlastní příkazy, které budou v určitých fázích provedeny. Tohoto jsem využil pro automatickou instalaci a konfiguraci mnoha komponent a aplikací. Seznam těchto přidávaných příkazů je zobrazen v tabulce 7.

Fáze FBA	Příkaz / prováděné činnosti
8600	<i>install.pre-reseal.bat</i>
	VMware View
	UltraVNC
	FlashPlayer
	Citrix HDX
	Kopírování předpřipraveného uživatelského profilu*1
8605	Reseal*2
8609	<i>net.bat</i>
	Manuální instalace ovladače síť. Adaptérů*3
8610	<i>install.post-reseal.bat</i>
	Nastavení povinného uživatelského profilu
	Instalace utility EWFCControl*4
	Přejmenování stanice (konvence PC-n), n=poslední 4 znaky MAC
	Instalace alternativního uživatelského prostředí
8615	<i>installvga.bat</i>
	Manuální instalace ovladače grafické karty + restart*3
8620	<i>install.final.bat</i>
	Cisco VPN
	OpenVPN
	Doplňková nastavení registrů*5
	Korekce ovladačů USB Mass Storage*6

Tabulka 7 - Přidané příkazy v FBA (vlastní zpracování)

Poznámky k *:

1. Profil byl předem připraven (konfigurace prostředí, zástupců apod.), při nasazení je pouze nakopírován a nakonfigurován.
2. Tzv. "Reseal" je proces, kdy je systém uzamčen v aktuálním stavu instalace a vypnut, v této podobě je potom možné jej distribuovat, přičemž instalace na cílovém stroji bude kratší o již provedené kroky.
3. Některé ovladače nebylo možné instalovat automaticky (rozdílné zařízení v HW variantách tenkého klienta, příp. problémy s automatickou instalací ovladače v FBA); tyto ovladače jsem nainstaloval manuálně pomocí programu **Devcon**.
4. Utilita **EWFCControl** bude sloužit pro pohodlné ovládání funkcí EWF administrátorem (popsána zvlášť v kapitole).

5. V registrech jsem provedl drobná doplňující nastavení (odstranění rezidentních programů ovladačů zvuku/audia, zakázání balloon-tipů, skrytí servisního uživatelského účtu, apod.).
6. U varianty tenkého klienta MATRIX bylo nutno nahradit ovladač *usbstor.sys* obsažený v systému (určený pro spouštění OS z USB) za jeho běžnou verzi, aby nedocházelo k problémům s USB Mass Storage zařízeními – řešeno pomocí vlastní aplikace (popsáno zvlášť).

Nasazení systému z média s Windows PE

Aby mohly být provedeny všechny kroky potřebné k úspěšnému zprovoznění WES systému na tenkém klientovi, je nutné mít nejprve dostupné prostředí operačního systému, který umožní provoz přípravných aplikací (zde zejména programy diskpart a bootsect pro přípravu systémového oddílu). Za tímto účelem nabízí společnost Microsoft OEM sestavitelům operační systém Windows PE (Preinstallation Environment) určený pro nasazovací média a záchranné oddíly (Windows PE je licenčně přístupný se zakoupením sady nástrojů pro Windows Embedded Standard).

Tento systém bude použit pro inicializaci primárního diskového oddílu tenkého klienta (dříve zmíněnými příkazy), přenos souborů image WES systému na tento oddíl (rozbalení souborů z archivu na instalační médium na připravený systémový oddíl) a vytvoření záchranného oddílu pro možnost obnovy systému. Obsah záchranného oddílu bude prakticky téměř shodný s obsahem instalačního média.

Jako instalační médium bude sloužit USB flash disk (po úpravách oddílu pro spouštění systému), případně, pokud by k tenkému klientovi byla připojena externí mechanika, může být použito i CD s boot sektorem. Vzhledem k velikosti výsledného systému (okolo 400 MB) jsem vyloučil možnost PXE zavádění systému.

5.3 Podpůrné aplikace a utility

Při sestavování operačního systému a softwarového vybavení tenkého klienta došlo nejdříve k situaci, kdy požadované funkcionality nebylo možné dosáhnout v rámci nástrojů a komponent systému Windows Embedded. Pro tyto účely bylo nutné naprogramovat vlastní aplikace, které danou funkci zastaly. Zde popíšu, o jaké aplikace se jedná, k čemu slouží, a jak fungují.

5.3.1 EWFCtrl – aplikace pro ovládání funkcionality EWF

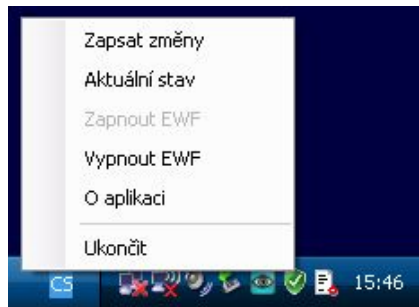
V systému zabudovaná funkcionality EWF (Enhanced Write Filter) zabraňuje zachování jakýchkoliv změn nastavení či lokálně uložených dat při vypnutí zařízení. Pokud však administrátor bude chtít provedené změny (změna konfigurace, instalace nové aplikace, apod.) zachovat, je nutné použít konzolovou utilitu s patřičnými parametry. Pro zjednodušení tohoto ovládání jsem navrhl aplikaci, která umožní základní příkazy EWF snadno aktivovat pomocí jednoduchého menu.

Popis a funkcionality aplikace

Program EWFCtrl je Windows Forms aplikace vytvořená na platformě Microsoft .NET Framework 2.0 (stejně jako všechny ostatní aplikace v tomto projektu), napsaná v jazyce C#. Po celou dobu běhu programu je hlavní formulář skryt, zobrazena je pouze ikona v system tray oblasti, přes kterou lze kliknutím vyvolat menu s ovládáním EWF. Toto menu obsahuje příkazy pro zapsání provedených změn na disk, vypnutí/zapnutí EWF, položku "O aplikaci" a ukončení. Po startu programu se ikona podle stavu EWF (zapnuto/vypnuto) nastaví na červenou nebo zelenou a indikuje tak stav, zda je filtr zápisů zapnutý nebo vypnutý.

Principy aplikace

Aplikace využívá objekty třídy `System.Diagnostics.Process` pro spouštění externích programů s různými parametry. Hlavním principem je spuštění konzolové aplikace `ewfmgr.exe` s parametry upřesňujícími určení cílové jednotky a požadované akce – celý takový příkaz pak může vypadat např. takto: `ewfmgr.exe C: -commit`. Uvedený příkaz zajistí, že provedené změny budou přeneseny z EWF bufferu na diskový oddíl. Mezi další důležité akce patří zapnutí a vypnutí filtru pomocí parametru `enable/disable`. V případě příkazu dotazujícího se na stav EWF pro daný oddíl je po spuštění programu `ewfmgr` navíc zachycen jeho konzolový výstup a následně vypsán uživateli. Všechny akce prováděné nad EWF systémem (s výjimkou zjištění stavu) vyžadují restart systému, a proto je po provedení každé takové akce uživatel formou odpovědního okna dotázán, zda chce okamžitě restartovat; při potvrzení je pak proveden restart systému příkazem `shutdown.exe /r /t 0`.



Obrázek 10 - Ovládací menu aplikace EWFCtrl (vlastní zpracování)

5.3.2 ReportSender – aplikace pro zasílání zpráv o chybách

Při používání tenkého klienta v praxi může nastat situace, kdy uživatel narazí na chybu, neočekávané a nežádoucí chování systému, nebo jiné potíže se zařízením. V takovém případě je důležité, aby se společnosti vrátila zpráva o tomto problému, a ten bylo následně možné co nejdříve vyřešit. Za tímto účelem bylo nutné navrhnout aplikaci, která umožní uživatelům snadné a intuitivní zasílání zpráv o chybách s dostatečným souhrnem informací přímo společnosti.

Popis a funkcionální aplikace

ReportSender je rezidentní aplikace, která poběží po celou dobu běhu WES 2009 systému na pozadí (opět indikována ikonou v systémové oblasti dolní lišty), a v případě stisknutí předurčené kombinace kláves se zobrazí její hlavní formulář (po dohodě byla zvolena kombinace kláves Shift+F1). V tomto formuláři pak uživatel může vyplnit údaje a informace o problému či nedostatku, které chce sdělit společnosti, a jedním stiskem tlačítka je snadno odeslat. S tímto textovým reportem je zároveň odeslán i snímek obrazovky a textový dokument, obsahující souhrn systémových informací, výpis posledních 50 systémových událostí a aktuálně spuštěných procesů (oba tyto dokumenty jsou před odesláním přístupné uživateli k nahlédnutí).

Principy aplikace

Aplikace využívá importovaných funkcí Windows API pro zachytávání stisknutých kláves, a v případě stisknutí předdefinované kombinace vyvolá hlavní formulář. Zároveň jsou v tu chvíli vygenerovány dokumenty se systémovými informacemi – je pořízen snímek obrazovky (prostřednictvím funkcí třídy `System.Drawing.Graphics`),

a jsou získány základní systémové informace (třídy `System.Windows.Forms`, `System.Environment` a `System.Diagnostics` obsahují metody pro zjištění množství vlastností systému). Tyto informace jsou dále doplněny o výpis běžících procesů a posledních 50 záznamů systémových událostí a aplikačních událostí. Oba tyto dokumenty jsou uloženy do dočasného úložiště uživatelského profilu, pod kterým je aplikace spuštěna.

Pro odesílání dat je použit protokol SMTP – zprávy jsou doručovány formou elektronické pošty s přílohami do předurčené schránky, jejíž adresa je spolu s informacemi o SMTP serveru pro odchozí poštu, jeho autentizací a formátem zprávy (předmět, tělo zprávy, apod.) načtena při spuštění aplikace z konfiguračního XML souboru (XML dokumenty jsou efektivně čteny a zpracovávány pomocí metod třídy `System.Xml.XmlDocument`). O zajištění odesílání e-mailových zpráv se pak v aplikaci stará třída `System.Net.Mail`, která obsahuje mimo jiné i komplexní funkcionalitu SMTP klienta.

Obrázek 11 - Hlavní formulář aplikace ReportSender (vlastní zpracování)

5.3.3 Shell32 – alternativní uživatelské prostředí

Při nasazení tenkého klienta ve virtuální desktopové infrastruktuře je žádoucí, aby měli uživatelé přístup pouze k přednastavenému Connection Broker programu, který jim pak již zprostředkuje relaci s jejich virtuální plochou, a naopak aby neměli přístup k ovládacím panelům a běžnému Explorer Shell rozhraní místního operačního systému tenkého klienta. Za tímto účelem byla navržena aplikace Shell32, která poskytne uživateli jednoduché a přehledné prostředí a zamezí přístup k nežádoucím prvkům systému.

Popis a funkcionálita aplikace

Aplikace Shell32 slouží jako alternativní uživatelské prostředí, spustí se tedy pokaždé, když se přihlásí výchozí uživatel User do systému (pro uživatele Administrator, Service či nově vytvořené uživatelské účty je však stále výchozím rozhraním Explorer Shell). Na hlavním formuláři této aplikace je vypsán seznam uživatelům zpřístupněných aplikací (zde bude obvykle pouze vybraný Connection Broker, správce systému ale může zvolit i jiné aplikace, které bude moci uživatel spouštět přímo na tenkém klientovi), dále základní informace jako čas a datum a vybrané ovládací prvky. Mezi tyto prvky patří možnost nastavení rozlišení monitoru, nastavení hlasitosti a odhlášení uživatele nebo vypnutí tenkého klienta.

Principy aplikace

Shell32 při spuštění načte z konfiguračního souboru cestu ke složce obsahující zástupce uživatelům přístupných programů (zvolena `C:\Shell`), a tyto zástupce pak vypíše jako položky programového menu na hlavním formuláři (tyto operace umožňují třídy `System.IO` a `System.Windows.Forms`). Při kliknutí na tyto odkazy je pak hlavní formulář skryt a spuštěna daná aplikace (po jejímž ukončení je opět zobrazen). Dále je možné měnit rozlišení – to je realizováno samostatným formulářem, ve kterém jsou vypsány dostupné grafické módy (rozlišení, barevná hloubka a frekvence). Po volbě rozlišení je změna propsána na disk, aby nebyla vyřazena EWF filtrem. Ovládání hlasitosti je realizováno prostřednictvím mixeru systému Windows (`sndvol32`). Pro provedení odhlášení uživatele či vypnutí celého systému byly použity importované funkce Windows API.



Obrázek 12 - Alternativní uživatelské prostředí Shell32 (vlastní zpracování)

Zavedení alternativního prostředí do systému

Aby při přihlášení konkrétního uživatele bylo spuštěno jiné než výchozí rozhraní, je nutné definovat tento alternativní Shell program v registrech – toho lze dosáhnout přidáním hodnoty s názvem `Shell`, obsahující řetězec s cestou k novému Shell programu do klíče `SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon` daného uživatelského účtu (soubor *ntuser.man* v povinném uživatelském profilu).

5.3.4 Další doplňkové aplikace

Mimo 3 výše uvedené aplikace, které bylo nutné naprogramovat pro splnění požadavků na tenkého klienta, bylo nutné napsat i několik menších aplikací, které řeší nedostatky zejména při automatickém nasazování systému. Jedná se o jednoduché a jednoúčelové utility, proto je zde popíšu pouze stručně.

SetupMonitor

Windows Forms aplikace pro sledování a zobrazování průběhu provádění .bat skriptů při FBA instalaci systému (viz tabulka č. 7). Mimo procentuální indikaci průběhu jednotlivých skriptů aplikace navíc pomocí Windows API funkcí skrývá

všechna okna příkazového řádku, aby je nebylo možné uživatelsky ukončit a narušit tak provádění skriptů.



Obrázek 13 - SetupMonitor - indikace průběhu skriptů (vlastní zpracování)

Renamepc

Konzolová utilita, jejímž účelem je v průběhu FBA instalace přejmenovat místní počítač podle stanovené konvence tak, aby název počítače byl unikátní, a obsahoval poslední 4 znaky MAC adresy síťového adaptéru tenkého klienta.

USBStorfix

Aplikace pro nahrazení zvoleného ovladače USBSTOR.SYS s podporou spuštění systému z USB za běžnou verzi tohoto ovladače – napravuje problémy s USB Mass Storage zařízeními na tenkých klientech varianty MATRIX, jejichž systémovým úložištěm není USB zařízení, ale SATA rozhraním připojená CF karta (což způsobuje potíže s odebráním USB zařízení). Typ úložiště přítomný v tenkém klientovi je detekován pomocí funkcí rozhraní WMI.

EvtClear

Po dokončení instalace systému pomocí FBA je spuštěna utilita EvtClear, která ještě před aktivací filtru EWF vyčistí systémové a aplikační události, aby byl protokol systému po spuštění čistý a neobsahoval varování vzniklá při instalaci aplikací a provádění skriptů.

5.4 Náklady a přínos projektu, srovnání cen

Návrh tenkého klienta byl realizován na základě potřeby nabídnout zákazníkům společnosti zařízení, které bude dostatečně přizpůsobivé jejich situaci a požadavkům, ale zároveň cenově dostupné a také prodejně výhodné pro společnost.

5.4.1 Náklady

S realizací tohoto projektu vznikly jednorázové náklady, které bylo nutno vynaložit zejména v souvislosti s nákupem potřebného hardwaru a specializovaného softwarového vybavení, ale také výdaje spojené se mzdovým ohodnocením vynaložené práce. V neposlední řadě pak vznikají potenciální náklady související s marketingem a produktovou prezentací, a v budoucnu bude pravděpodobně nutné realizovat také aktualizaci softwaru a opravy případných chyb. Souhrn těchto nákladů s orientačními hodnotami je zobrazen v následující tabulce.

Položka	Hodnota
Nákup SW (Windows Embedded toolkit, 2x runtime licence)	25 000 Kč
Nákup HW (komponenty dvou prototypů tenkého klienta)	7 500 Kč
Práce na vývoji tenkého klienta	20 000 Kč
Marketing a prezentace produktu	<i>(zatím nelze zohlednit)</i>
Budoucí aktualizace a korekce SW	<i>(zatím nelze zohlednit)</i>

Tabulka 8 - Orientační souhrn nákladů projektu (vlastní zpracování)

5.4.2 Přínos pro firmu

Zásadním přínosem pro firmu bude v případě nového tenkého klienta zejména možnost nastavení vyšší marže při prodeji, protože navržený tenký klient je ve výsledku (včetně licence operačního systému i práce na sestavení jednoho kusu) stále značně levnější, než dříve používaná řešení (viz. srovnání cen). Dalším přínosem pak může být rozšíření nabídky výrobků firmy o výrobek "vlastní značky", možnost kreativní propagace (výrobek může nést logo společnosti, apod.), a v neposlední řadě také vyšší užitná hodnota pro zákazníka v podobě dobrého poměru cena/výkon.

Návratnost projektu nelze v tuto chvíli přesně určit či předpovědět, jelikož zisky jsou přímo závislé na množství, a zejména pak velikosti potenciálních zakázek, ale také na velikosti marže, která bude u výsledného produktu stanovena. Pokud by například společnost získala zakázku, ve které zákazník požaduje nasazení 100 tenkých klientů, a zvolila marži 800-1000Kč, byla by návratnost prakticky okamžitá, a již při této první zakázce by byl vytvořen určitý zisk. V případě menších zakázek by pak tato doba byla delší, analogicky podle předchozích prodejů by však neměla být delší než 6-12 měsíců.

5.4.3 Srovnání cen

Zásadním požadavkem a klíčovým faktorem při návrhu tenkého klienta bylo dosažení dobrého poměru cena/výkon, ale zároveň také udržení nižší ceny oproti konkurenci. Pro srovnání jsem vybral dva tenké klienty společnosti Wyse, první se srovnatelným výkonem, druhý v podobné cenové kategorii, avšak s nižším výkonem. Rozdíly mezi konfiguracemi tenkých klientů a srovnání jejich cen jsou zobrazeny v následující tabulce (pozn. ceny jsou pouze orientační a mohou se u různých dodavatelů lišit).

Tenký klient	Wyse S50	Wyse R50L	VECTOR	MATRIX
Hardware	CPU 366MHz RAM 256MB HDD 128MB	CPU 1.5GHz RAM 1GB HDD 1GB	CPU 1.66GHz RAM 1GB HDD 4GB	CPU 1.66GHz RAM 1GB HDD 4GB
OS	Linux	Windows XP Embedded	Windows Embedded Standard 2009	Windows Embedded Standard 2009
Cena	4 700 Kč	10 100 Kč	*5 000 Kč	*6 300 Kč

Tabulka 9 - Srovnání cen a parametrů tenkých klientů (vlastní zpracování)

Pozn.: Ceny tenkých klientů VECTOR a MATRIX nezahrnují prodejní marži a náklady na sestavení 1 kusu.

5.5 Plánovaná a potenciální rozšíření

V aktuální podobě je již tenký klient připraven pro široké možnosti použití, zejména pak pro primární účel tohoto návrhu – nasazení ve VDI. Aby bylo ale nasazení tenkých klientů a jejich správa pro zákazníky ještě jednodušší a praktičtější, bylo by dobré do budoucna postupně rozšířit tento návrh o další funkcionalitu.

5.5.1 Software pro správu tenkých klientů

Pro zákazníky, kteří provozují velké množství tenkých klientů, by bylo dobré pro větší přehlednost a komfort správy těchto klientů vytvořit software, který umožní centrální správu a komplexní pohled na všechna zařízení nasazená v dané síti. Tento software by měl zahrnovat funkcionalitu pro uchovávání informací o zařízeních (logy, organizační skupiny, filtrování, aktuální stav zařízení, apod.), jejich vzdálené ovládání, zapnutí/vypnutí a zejména pak vzdálenou aktualizaci souborů operačního systému přímo ze serveru. Všechny tyto funkce by měla zajistit serverová aplikace, ke které bude možno přistupovat pomocí správcovské management konzole; generování reportů a provádění požadovaných akcí pak zajistí agent nainstalovaný na tenkém klientovi.

5.5.2 Možná vylepšení stávajících aplikací, odladění případných nedostatků

Všechny podpůrné aplikace a utility tenkého klienta jsou funkční a zastávají svůj účel, do budoucna je však bude možné rozšířit o další funkce – např. Shell aplikace může být rozšířena o podporu ovládání nakonfigurované VPN, reportování chyb může být upraveno pro zachytávání logů aplikace, kterou daný zákazník používá, apod. Je také pravděpodobné, že se při používání tenkého klienta objeví drobné nedostatky na sestaveném operačním systému či aplikacích, které bude nutno dodatečně opravit.

5.5.3 Přechod na systém Windows Embedded Standard 7

Do budoucna je možným potenciálním krokem vývoje tohoto tenkého klienta také přechod na moderní verzi Embedded systému Standard 7 – tímto by se zákazníkům zpřístupnily nejnovější technologie a aplikace obsažené v systémech Windows 7, avšak za mírně vyšší cenu a náročnost na HW. Značnou výhodou je zde nativní podpora aktuální verze platformy .NET Framework 4.0 a tedy i moderních .NET aplikací.

6 Zhodnocení a závěr

Cílem této bakalářské práce bylo navrhnout tenkého klienta, kterého bude moci společnost OldanyGroup nabídnout svým zákazníkům, a to zejména jako komplementární produkt k řešení desktopové virtualizace, či terminálových služeb.

S přihlédnutím ke stanoveným požadavkům byl vytvořen návrh dvou variant tenkého klienta a společnosti poskytnuty prototypy, které pokrývají všechny nadefinované vlastnosti, a zároveň nabízí při dodržení požadavku na nízkou cenu tenkého klienta i velmi solidní parametry jak po stránce výkonu HW, tak univerzálností a víceúčelovostí softwarového vybavení. Součástí prototypu jsou také volitelné komponenty a rozšíření, která umožní vysokou škálovatelnost a přizpůsobivost – produkt by tak měl vyhovět potřebám většiny zákazníků, finančním možnostem i různým podmínkám jejich produkčního prostředí.

Pro zajištění bezproblémového zprovoznění a funkčnosti u zákazníka byl tenký klient dostatečně otestován, zejména co se stability HW a efektivity chlazení týče, a také podroben testu bezchybnosti softwarové části. Zde se mohou samozřejmě objevit drobné nedostatky či chyby (vzhledem k úpravám operačního systému a spoléhání na software vlastního návrhu pro řešení důležitých úkonů) na ty však bude možné rychle reagovat díky zabudovaným nástrojům pro reportování chyb.

Celkově by měl navržený tenký klient splňovat aktuální požadavky společnosti, i požadavky potenciálních zákazníků, kteří jej budou využívat. Do budoucna pak bude možné tento návrh ještě dále rozšířit upgradem na modernější verzi operačního systému, vylepšením důležitých aplikací a v neposlední řadě také vytvořením softwaru pro centralizovanou správu tenkých klientů.

Seznam použité literatury

Knižní zdroje (řazeny dle příjmení prvního z autorů)

- (1) AGRAWAL, Neeraj; TIWARI, Shekhar; ANSARI, Nasimuddin. *Practical Handbook Of Thin-Client Implementation*. New Delhi : New Age International, 2005. 232 s.
- (2) LIMING D., Sean. *Windows XP Embedded Advanced*. ilustrované vydání. RTC Books, 2003. 717 s. ISBN 09-2939-277-9.
- (3) MILLER P., Frederic; VANDOME F., Agnes; MCBREWSTER, John. *Desktop Virtualization*. Mauritius : VDM Publishing House Ltd., 2009. 82 s. ISBN 61-3027-294-4.

Internetové zdroje (řazeny dle názvu webu)

- (4) *CDW-G - IT Products and Services for Government, Education and Healthcare* [online]. 2008 [cit. 2010-12-02]. Whitepaper-Thin Clients are In Again. Dostupné z WWW: <<http://webobjects.cdw.com/webobjects/media/pdf/Thin-Clients-Are-In-Again.pdf>>.
- (5) *Intel Software Development Products – White Papers* [online]. 2008 [cit. 2010-12-08]. Understanding Alternative Compute Models. Dostupné z WWW: <http://www.symantecintelalliance.com/public/pdf/Alternate_Compute_Model_WP.pdf>.
- (6) *Intel® Atom™ Processor N450, D410 and D510 for Embedded Applications* [online]. Únor 2010 [cit. 2011-03-25]. Thermal Design Guide. Dostupné z WWW: <<http://download.intel.com/embedded/processors/thermalguide/323439.pdf>>.
- (7) *Microsoft Corporation: Software, Smartphones, Online, Games, Cloud Computing, IT Business Technology, Downloads* [online]. 2010 [cit. 2011-01-18]. Microsoft Windows Embedded. Dostupné z WWW: <<http://www.microsoft.com/cze/windowseembedded>>.
- (8) *MSDN Library* [online]. 2009 [cit. 2011-01-19]. Overview of the .NET Framework. Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/zw4w595w\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/zw4w595w(v=VS.90).aspx)>.
- (9) *MSDN Library* [online]. 2011 [cit. 2011-01-19]. Visual Basic. Dostupné z WWW: <<http://msdn.microsoft.com/library/2x7h1hfk>>.

- (10) ŠVÍK, Martin. ROI, TCO a NPV: Svatá trojice. *CIO Business World.cz* [online]. 25.11.2009, 11/2009, [cit. 2011-01-24]. Dostupný z WWW: <<http://businessworld.cz/it-strategie/roi-tco-a-npv-svata-trojice-5303>>.
- (11) *Technická podpora Microsoft* [online]. 2007 [cit. 2011-01-18]. Principy Remote Desktop Protocol (RDP). Dostupné z WWW: <<http://support.microsoft.com/kb/186607/cs>>.
- (12) *Virtualizace, konzultace a správa pro řešení na VMware vSphere, Veeam, Linux a Windows | OldanyGroup* [online]. 2009 [cit. 2011-01-17]. VMware View (VDI) | OldanyGroup. Dostupné z WWW: <<http://www.oldanygroup.cz/vmware-view-virtual-desktop-infrastructure-vdi-12>>.
- (13) *Visual C# Developer Center* [online]. 2011 [cit. 2011-01-19]. The C# Language. Dostupné z WWW: <<http://msdn.microsoft.com/en-us/vcsharp/aa336809>>.
- (14) *VMware View: Desktop Virtualization and Desktop Management* [online]. 2009 [cit. 2011-01-17]. VMware View™ 4 with PCoIP Information Guide. Dostupné z WWW: <<http://www.vmware.com/files/pdf/VMware-View4-PCoIP-IG-EN.pdf>>.

Seznam obrázků

Obrázek 1 - Architektura terminálových služeb (5)	19
Obrázek 2 - Architektura desktopové virtualizace (5).....	20
Obrázek 3 - Architektura Blade PC (5)	21
Obrázek 4 - Architektura streamování OS (5)	21
Obrázek 5 – Schéma virtualizovaného stroje (vlastní zpracování).....	22
Obrázek 6 - Virtuální desktopová infrastruktura s VMware View (12).....	25
Obrázek 7 - Progressive Build - srovnání PCoIP a RDP (14)	27
Obrázek 8 - Schéma začlenění .NET framework v systémech (8).....	36
Obrázek 9 - Rozvržení komponent ve skříní - reference (6)	42
Obrázek 10 - Ovládací menu aplikace EWFCControl (vlastní zpracování)	53
Obrázek 11 - Hlavní formulář aplikace ReportSender (vlastní zpracování)	54
Obrázek 12 - Alternativní uživatelské prostředí Shell32 (vlastní zpracování).....	56
Obrázek 13 - SetupMonitor - indikace průběhu skriptů (vlastní zpracování)	57

Seznam tabulek

Tabulka 1 - Klady a zápory nasazení tenkých klientů (vlastní zpracování)	28
Tabulka 2 - Srovnání spotřeby energie PC a tenkého klienta (vlastní zpracování)	29
Tabulka 3 – Vývoj platformy .NET Framework (vlastní zpracování)	36
Tabulka 4 - HW konfigurace variant tenkého klienta (vlastní zpracování).....	41
Tabulka 5 - Limitní hodnoty procesorů Atom (zprac. podle 6).....	42
Tabulka 6 - Teploty naměřené při zátěžovém testu (vlastní zpracování)	43
Tabulka 7 - Přidané příkazy v FBA (vlastní zpracování)	50
Tabulka 8 - Orientační souhrn nákladů projektu (vlastní zpracování)	58
Tabulka 9 - Srovnání cen a parametrů tenkých klientů (vlastní zpracování)	59

Seznam použitých zkratek

ACPI – Advanced Configuration and Power Interface

API – Application Programming Interface

ATX – Advanced Technology Extended

CDFS – CD-ROM File System

CD-ROM – Compact Disc Read-Only Memory

CF – CompactFlash

CLR – Common Language Runtime

CPU – Central Processing Unit

EFW – Enhanced Write Filter

FAT – File Allocation Table

FBA – First Boot Agent

HAL – Hardware Abstraction Layer

HDX – High Definition user Experience

HT – Hyper-Threading

HW – Hardware

ICA – Independent Computing Architecture

ICT – Information and Communication Technologies

IT – Information Technology

LAN – Local Area Network

MS – Microsoft

NTFS – New Technology File System

NTLDR – NT Loader

OEM – Original Equipment Manufacturer

OS – Operating System

PC – Personal Computer

PCI – Peripheral Component Interconnect

PCoIP – PC over IP

PDF – Portable Document Format

PE – Pre-installation Environment

PPTP – Point-to-Point Tunneling Protocol

PXE – Pre-boot Execution Environment

RAM – Random-Access Memory

RDP – Remote Desktop Protocol

SATA – Serial ATA

SDI – System Deployment Image

SMTP – Simple Mail Transfer Protocol

SP – Service Pack

SQL – Structured Query Language

SW – Software

TCO – Total Costs of Ownership

TCP – Transmission Control Protocol

TFTP – Trivial File Transfer Protocol

USB – Universal Serial Bus

VB.NET – Visual Basic .NET

VDI – Virtual Desktop Infrastructure

VESA – Video Electronics Standards Association

VM – Virtual Machine

VNC – Virtual Network Computing

VPN – Virtual Private Network

WDM – Windows Driver Model

WES – Windows Embedded Standard

WMI – Windows Management Instrumentation

XML – Extensible Markup Language

Seznam příloh

Příloha č. 1: Fotografie tenkého klienta

Příloha č. 2: Grafy zátěžových testů tenkých klientů s Prime95

Příloha č. 3: Základní strom zvolených komponent systému Windows Embedded

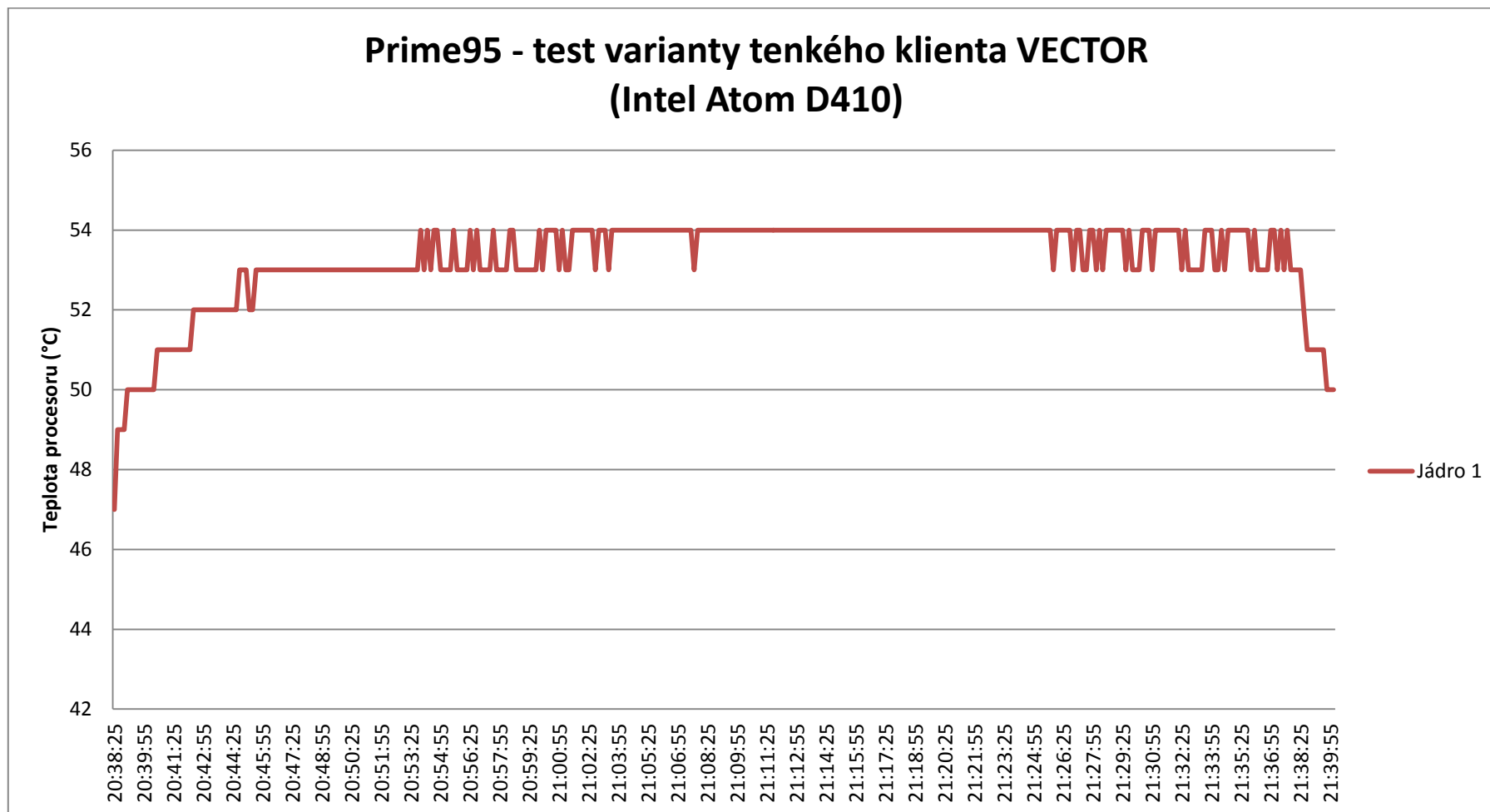
Příloha č. 4: Ukázka kódu – třída hlavního formuláře aplikace EWFCtrl

Příloha č. 1: Fotografie tenkého klienta

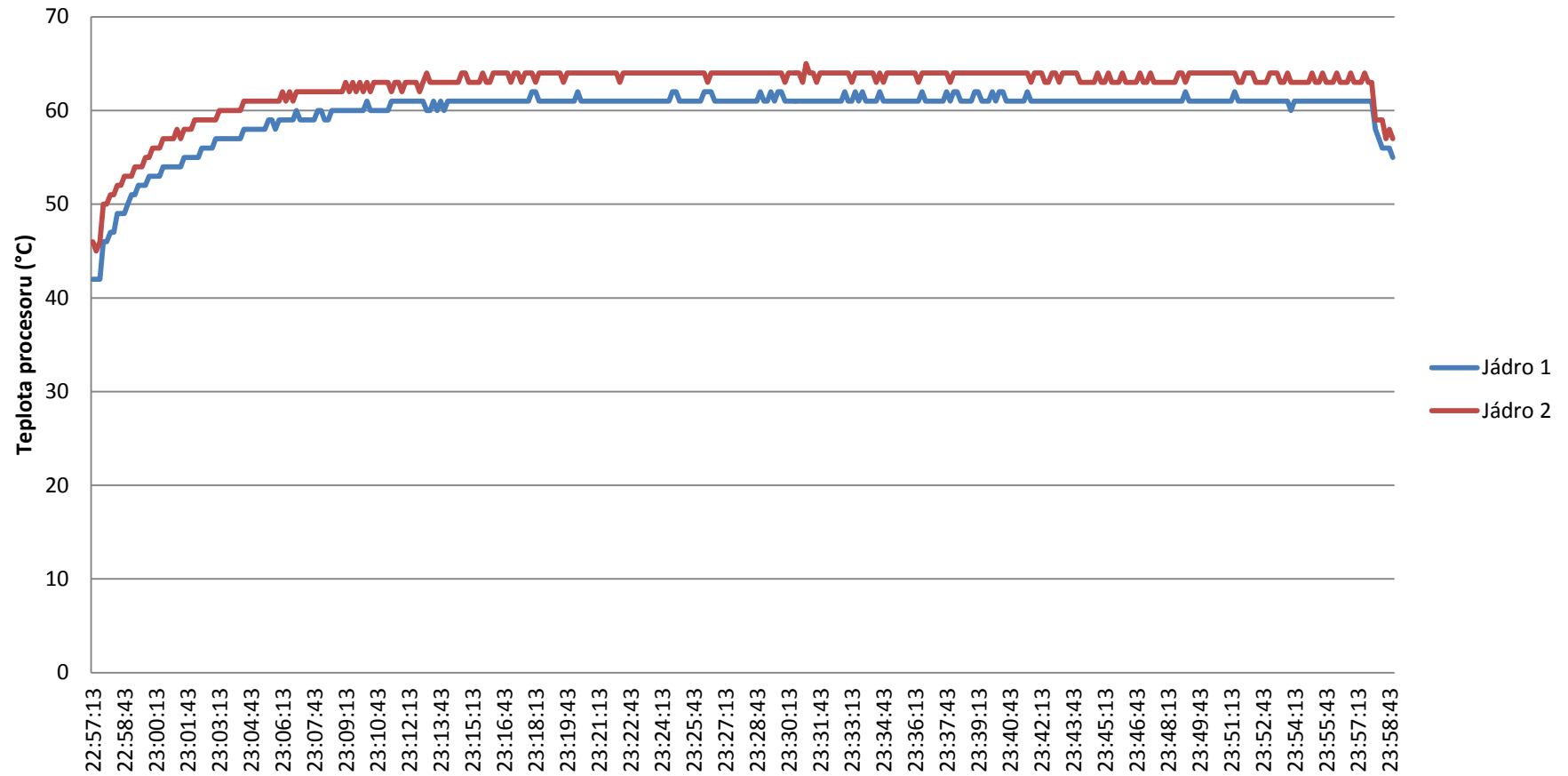
Zdroj: Produktová prezentace společnosti OldanyGroup



Příloha č. 2: Grafy zátěžových testů tenkých klientů s Prime95

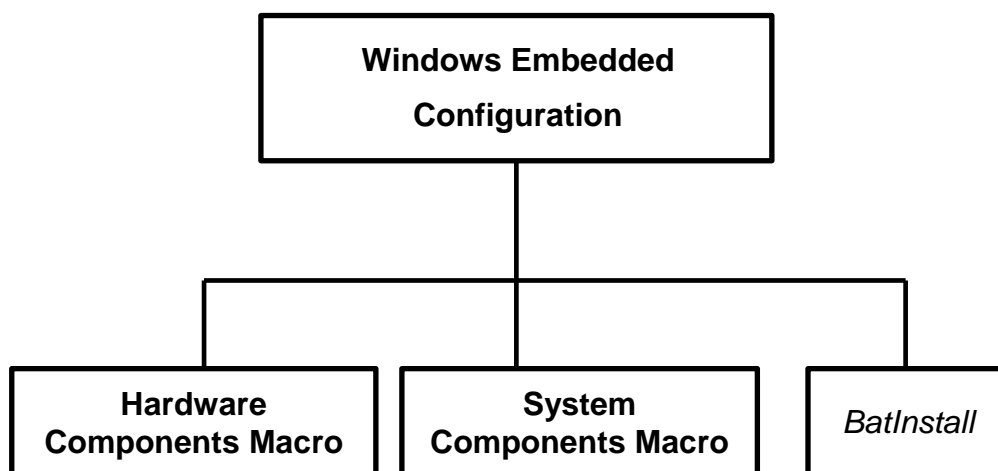


Prime95 - test varianty tenkého klienta MATRIX (Intel Atom D510)



Příloha č. 3: Základní strom zvolených komponent systému Windows Embedded 2009

Pozn.: Kvůli velkému rozsahu podrobného výpisu všech komponent strom obsahuje pouze první úroveň struktury celého projektu konfigurace Windows Embedded, **tučně** zvýrazněné komponenty zde označují makro-komponenty (tj. obsahující závislosti na dalších komponentách – slouží jako kontejnery pro množiny vyžadovaných komponent daného celku), *kurzívou* jsou označeny komponenty vlastní výroby, které byly doplněny.



- **Hardware Components Macro:** Kontejner s komponentami všech ovladačů veškerých HW zařízení a vyžadovanými položkami, na kterých tyto ovladače závisí. Tato položka byla vygenerována na základě analýzy HW tenkého klienta.
- **System Components Macro:** Hlavní skupina komponent, obsahuje veškeré důležité systémové součásti (loader, shell, apod.) i komponenty doplňující veškerou další vyžadovanou funkcionalitu systému. První úroveň obsahu tohoto kontejneru je vypsána na následující straně.
- *BatInstall:* Vlastní komponenta doplněná do databáze pro účely automatizace nasazení a konfigurace systému při jeho prvním spuštění, včetně instalace aplikací a dalších úprav.

.NET Framework 2.0

Control panel applets

System tools

Administration support tools

Automatic logon

Base support binaries

CDFS

Client printing support

CMD

Control panel command line support

Copy and compare tools

Czech language support

Device manager

Diskpart utility

Enhanced Write Filter

EWF Manager Console application

EWF NTLDR

Explorer Shell

FAT

File sharing

Fonts

IME Prototype

Internet Explorer 8

IP NAT

IP Security tools

IPX/SPX Transport stack

Kernel Audio Support

Language: English US (0409)

Local Network Bridge

Local printing support

Misc. Accessories

Misc. Command line tools

MSconfig utility

Net.exe utility

Net Inf collection

Network routing

NTFS

Other TCP/IP Services

Regional and Language options

Registry editor

RPC Services

RunAs service

Safe Mode Support

Safely Remove Hardware program

Security Shell Extension

Server Command line tools

Server printing support

Standard Start Menu Shortcuts

System Cloning Tool

Takeown utility

TAPI interface

Task manager

Task scheduler

TCP/IP Utilities

Telnet Client

Thin Client

UDFS

USB Boot 2.0

USB Printing support

User Account

WAN Miniports

Windows Installer service

Windows Logon

WMI Technologies

Windows Media Player Technologies

Příloha č. 4: Ukázka kódu – třída hlavního formuláře aplikace EWFCtrl

```
using System;
using System.Windows.Forms;
using System.Diagnostics;
using System.Resources;

namespace EWFCtrl
{
    partial class Form1 : Form
    {
        private static ResourceManager resMgr;
        private Process ewfmgr;
        private Process shutdown;
        private string consoleOutput;
        private AboutBox1 abt;

        // Konstruktor třídy formuláře
        public Form1()
        {
            // Vytvoření ResourceManager objektu pro přístup k souborům (ikony, bitmapy, apod.)
            resMgr = new ResourceManager("EWFCtrl.Properties.Resources",
System.Reflection.Assembly.GetExecutingAssembly());

            InitializeComponent();

            // Skrytí formuláře
            this.Opacity = 0;
            this.ShowInTaskbar = false;

            // Vytvoření objektů pro spouštění aplikací
            this.ewfmgr = new Process();
            ewfmgr.StartInfo.FileName = "ewfmgr.exe";
            ewfmgr.StartInfo.UseShellExecute = false;
            ewfmgr.StartInfo.RedirectStandardOutput = true;

            this.shutdown = new Process();
            shutdown.StartInfo.FileName = "shutdown.exe";

            initStatus();
        }
    }
}
```



```

// Položka menu "Ukončit"
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}

// Položka menu "Zapsat změny"
private void commitChangesToolStripMenuItem_Click(object sender, EventArgs e)
{
    ewfCommand("C: -commit", false);
    reboot();
}

// Položka menu "Aktuální stav"
private void statusToolStripMenuItem_Click(object sender, EventArgs e)
{
    ewfCommand("C:", true);
}

// Položka menu "Zapnout EWF"
private void enableEWFToolStripMenuItem_Click(object sender, EventArgs e)
{
    ewfCommand("C: -enable", false);
    reboot();
}

// Položka menu "Vypnout EWF"
private void disableEWFToolStripMenuItem_Click(object sender, EventArgs e)
{
    ewfCommand("C: -disable", false);
    reboot();
}

// Položka menu "O aplikaci"
private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (abt != null) abt.Dispose();
    abt = new AboutBox1();
    abt.Show();
}

```

```

// Metoda pro provedení pomocí programu "ewfmgr.exe" s volitelným výstupem na obrazovku
private void ewfCommand(string args, Boolean output)
{
    ewfmgr.StartInfo.Arguments = args;
    try
    {
        ewfmgr.Start();
        consoleOutput = ewfmgr.StandardOutput.ReadToEnd();
        ewfmgr.WaitForExit();
        if(output)MessageBox.Show(consoleOutput, "EWFControl", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch
    {
        // Možná výjimka při spuštění programu (nedostatečná práva)
        MessageBox.Show("Nelze spustit " + ewfmgr.StartInfo.FileName, "Chyba", MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
    }
}

// Metoda pro restart systému programem "shutdown.exe", před restartem se dotáže
private void reboot()
{
    DialogResult ans = MessageBox.Show("Pro dokončení EWF operací je vyžadován restart.\nRestartovat nyní?",
    "Restart", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (ans == DialogResult.Yes)
    {
        shutdown.StartInfo.Arguments = "/r /t 0";
        shutdown.Start();
    }
}

// Metoda pro zjištění počátečního stavu EWF, nastavení ikony a položek menu
private void initState()
{
    try
    {
        // Dotaz na status EWF pro C:
        ewfmgr.StartInfo.Arguments = "C:";
        ewfmgr.Start();
        consoleOutput = ewfmgr.StandardOutput.ReadToEnd();
    }
}

```

```

ewfmgr.WaitForExit();

//EWF je zapnuto - nastavit zelenou ikonu a položky menu
if (consoleOutput.Contains("ENABLED"))
{
    notifyIcon1.Icon = (System.Drawing.Icon)resMgr.GetObject("icon_on");
    notifyIcon1.Text = "Enhanced Write Filter je aktivní";
    contextMenuStrip1.Items[2].Enabled = false;

}

//EWF je vypnuto - nastavit červenou ikonu a položky menu
else
{
    notifyIcon1.Icon = (System.Drawing.Icon)resMgr.GetObject("icon_off");
    notifyIcon1.Text = "Enhanced Write Filter je vypnut";
    contextMenuStrip1.Items[0].Enabled = false;
    contextMenuStrip1.Items[3].Enabled = false;

}
}
catch
{
    // Možná výjimka v případě že program "ewfmgr.exe" není nalezen, nebo jej není možné spustit
    // reakcí je pak zablokování položek menu, zobrazení červené ikony a informace o chybě
    notifyIcon1.Icon = (System.Drawing.Icon)resMgr.GetObject("icon_off");
    contextMenuStrip1.Items[0].Enabled = false;
    contextMenuStrip1.Items[1].Enabled = false;
    contextMenuStrip1.Items[2].Enabled = false;
    contextMenuStrip1.Items[3].Enabled = false;
    notifyIcon1.ShowBalloonTip(3, "EWFMgr nenalezen", "Ovládání EWF není dostupné, nelze získat informace o
stavu.", ToolTipIcon.Error);
}
}
}
}
}

```