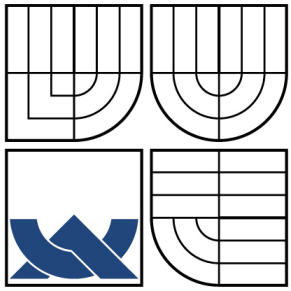
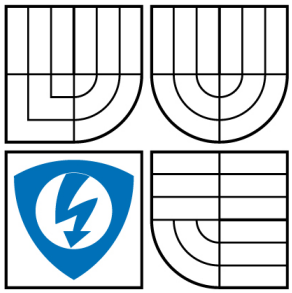


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

SOFTWARE PRO PŘEVZETÍ KONTROLY NAD POČÍTAČEM

SOFTWARE FOR ASSUMING CONTROL OVER THE COMPUTER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. JAN KOSTELNÍK

AUTHOR

VEDOUCÍ PRÁCE — Ing. MICHAL POLÍVKA

SUPERVISOR

BRNO 2008

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Jan Kostelník
Bytem: Lipovská 1162/40, 79001, Jeseník
Narozen/a (datum a místo): 6.11.1983, Jeseník

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 60200 Brno 2
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - ☐ disertační práce
   - ☑ diplomová práce
   - ☐ bakalářská práce

   jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Software pro převzetí kontroly nad počítačem
Vedoucí/školitel VŠKP: Ing. Michal Polívka
Ústav: Ústav telekomunikací
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

- ☑ tištěné formě - počet exemplářů 2
- ☑ elektronické formě - počet exemplářů 2

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

Článek 2

Udělení licenčního oprávnění

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

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

Článek 3

Závěrečná ustanovení

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

V Brně dne: .....

....

Nabyvatel Autor

ANOTACE

V úvodní části diplomové práce je stručně zmíněn historický a technologický vývoj terminálu, jeho definice, výhody a nevýhody. V další části je uveden všeobecný popis služeb a jejich význam. Jsou zde popsány terminálové služby – aplikace pro vzdálený přístup, jejich přínos a obecné výhody a nevýhody. Je provedeno rozdělení terminálových služeb. U každého typu terminálové služby jsou popsány funkce, možnosti, konkrétní výhody, nevýhody. Dále je uveden rozbor důvodu a vhodnosti komprese dat. Na demonstrační aplikaci je ukázán dopad přílišné (dobrovolné) ztráty informací o souřadnicích. V další části se práce zabývá nejpoužívanějšími aplikacemi pro vzdálené ovládání počítače. Jsou zde provedeny porovnání a testy.

Pátá kapitola je stěžejní částí diplomové práce. Je zde proveden popis návrhu a realizace pracoviště. Dále je popsán návrh a implementace systému pro dohled a ovládání. Tento systém umožňuje dohlížejícímu uživateli sledování činnosti uživatelů a ovládání jejich počítače. Dohlížející pracovník může také přistupovat k diskům a jiným paměťovým zařízením, které se nachází ve sledované stanici. Komunikace je komprimována a šifrována. Jsou zde použity symetrické a asymetrické šifrovací algoritmy. Dále jsou popsány dvě doplňkové aplikace – generování klíčů a síťový souborový manažer. V závěru mojí práce jsou uvedeny výsledky výkonnostních testů a návrhy na další rozšíření.

Klíčová slova: terminál, přenos dat, pracovní plocha, obraz, myš, klávesnice, řízení, programování, komprese, grafické rozhraní, Java

ABSTRACT

The introduction of this Master's thesis describes a historical and technological evolution of the terminal application, including its advantages and disadvantages. In the following part, a general description of the services and their understanding are mentioned. The benefits and general advantages of the terminal services are described. For the each type of the terminal service, all the functions, options, exact advantages and disadvantages are described. Consequently, the study of the suitability of the compression is presented. On the demo application, the impact of an excessive (voluntary) loss of information is demonstrated. In the next part, the thesis deals with the most frequently used remote-control applications. The tests and comparisons are made as well.

The fifth chapter is the fundamental part of this work. The design approach of the workbench is presented there. Consequently, the design approach and system implementation intended for the supervision and control is described. This system makes the user possible to observe other user's activities and also to control their workstations. The supervising user can also access their drives and other memory devices, which are located in the monitored workstation. The communication is compressed and encrypted. The symmetric and asymmetric encryption algorithms are used there. In the following part, there are two supplementary applications mentioned – application for the key generating and network file manager. In the end of my work, the results of the performance test and also the design of other improvements are presented.

Keywords: terminal**,** data transmission, desktop, image, mouse, keyboard, control, programming, compression, graphical user interface, Java

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Software pro převzetí kontroly nad počítačem“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne …… ……..

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Michalu Polívkovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce, dále pak svým rodičům za veškerou podporu, které se mi dostalo v průběhu celého studia.

V Brně dne …… ……..

(podpis autora)

Obsah

Seznam obrázků

Seznam tabulek

1 ÚVOD

V souvislosti s rozvojem moderních technologií ve 20. století se stal pojem informace běžně používaným, a to v nejrůznějších kontextech. Člověk přijímá a vydává informace již od svého narození a tato činnost je součástí každého člověka. Informace můžou mít různou povahu, význam, důležitost a cenu. Pro jednoho člověka může být konkrétní informace naprosto bezcenná, zatím co pro jiného může ta samá informace znamenat životní událost. Informací je mnoho a jsou roztroušeny všude kolem nás. V knihovnách, muzeích, na internetu atd.

Odjakživa hrál důležitou roli čas získání informace, ale také snadnost získání informace. Člověk se po celou svou existenci snažil a bude dále snažit potřebný čas zkrátit na minimum. Postupem času vymýšlel rozmanité techniky vzdáleného dorozumívání. Postupně od zvukových a kouřových signálů, přes telegrafii, dálnopis, telefonii až po počítačovou síť a mobilní komunikaci.

Tímto počínáním si člověk značně zjednodušil nejen profesní, ale i osobní život. Dnes má jakékoliv informace na dosah ruky. Jednoduchým úkonem (např. stisknutím tlačítka) překlene tisíce kilometrů a má tak možnost okamžitě komunikovat s druhým koncem světa.

1.1 Definice terminálu, jeho historie a vývoj

Obecně platí, že terminál je technické zařízení, ovládané člověkem, sloužící k ovládání nějakého objektu. Toto zařízení má mnoho podob. Od jednoduchých až po velmi komplikované. Pod pojmem terminál si můžeme postupně od historických prvotin představit například skříň či místnost, obsahující nejrůznější ovládací elementy, jako jsou například páky, otočná ústrojí, tlačítka, ale i signalizační a zobrazovací prvky ilustrující stav ovládaného objektu. Například ve slavné éře století páry můžeme terminál definovat jako mechanické zařízení sloužící k ovládání jiného technického zařízení. Obecně pracovník ovládající terminál je nazýván operátor. Takovým operátorem byl v této době například strojvůdce. Ovládaný objekt byla lokomotiva a terminál bylo místo obsahující prostředky pro ovládání a kontrolu provozu lokomotivy.

Dalším vývojem a zdokonalováním techniky byly vynalezeny komunikační terminály, které z počátku sloužily výhradně pro komunikaci mezi lidmi. Jedná se o telegraf a dálnopis – telex. Terminál pro tuto dobu lze definovat jako mechanické nebo elektromechanické zařízení sloužící k ovládání a ke komunikaci. Na obr. 1.1 je vyobrazen stolní model dálnopisu T100 firmy Siemens.

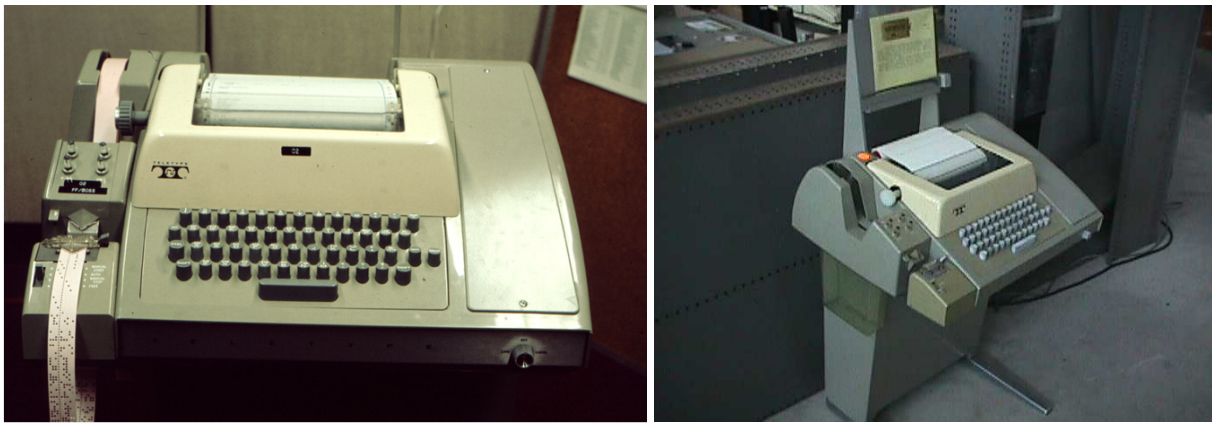
Obr. 1.1: Stolní model dálnopisného přístroje T100 firmy Siemens

Dalším důležitým obdobím v technickém pokroku je éra křemíková. Vytvoření prvních elektronických počítacích strojů a následně sálových počítačů přineslo lidem (vědcům) mnoho

nových možností, ale i problémů a komplikací s tím spojených. V době sálových počítačů (přibližně od 60. do konce 70. let) byly všechny zdroje (programy, data, výpočetní kapacity, periferie apod.) umístěny v jednom centrálním bodě. Nutno zdůraznit, že i techničtí pracovníci a vývojáři mohli vykonávat svou práci jen v bezprostředním okolí stroje.

Díky finanční náročnosti těchto velkolepých počítačů a již výše zmíněných problémů s dostupností se hledala taková řešení, aby každý pracovník, nezávisle na ostatních, mohl pohodlně zadávat počítači početní úkoly na dálku – ze své pracovny, sledovat výstupní data, stav atd. Nalezené řešení bylo inspirováno dálnopisem. Tedy první počítačové terminály byly zpočátku upravené dálnopisy. Přesto takové práce, komunikace, programování a zejména ladění programů při hledání chyb se mnohdy staly pro pracovníka noční můrou. Nástroje pro ladění, jak je známe dnes, vůbec neexistovaly a při pádu nebo nefunkčnosti programu bylo zapotřebí vytisknout obsah celé operační paměti a ručně chybu najít. Člověk si pak přál, aby ona operační paměť velká 10 KB byla podstatně menší, viz lit. [15].

Na obr. 1.2 je zobrazen „dálnopis“ sloužící ke komunikaci s centrálním počítačem. S tímto terminálem se pracovalo tak, že pracovník napsal „jednoduchý“ řádek textu pomocí klávesnice, po zmáčknutí tlačítka „návrat vozíku“ – dnešní „Enter“ se po specifické době (po zpracování vstupních dat sálovým počítačem a posláním odpovědi zpět terminálu) začala hlučně tisknout odpověď. Odpověď se vyrazila průbojníkem do štítku. Obrázky byly převzaty z lit. [12].



Obr. 1.2: Historický počítačový terminál – výstup pomocí tisku a děrování

Netrvalo dlouho a tyto terminály byly nahrazeny modernějšími a komfortnějšími. Ty obsahovaly obrazovku, klávesnici a nezbytné elektronické obvody pro jeho funkci. V obvodech těchto terminálů se vyskytovaly analogové obvody a logická hradla. Mikroprocesor se v terminálu začal vyskytovat o něco později. Mezi důvody jeho použití lze uvést zjednodušení celého terminálu, redukce elektronického vybavení, ale i rozšíření o nové možnosti. Připojení k počítači bylo realizováno sériovou linkou – RS-232. V prvních modelech byla obrazovka jednobarevná a pracovalo se pouze v textovém režimu. Text byl nejčastěji zobrazován zelenou barvou.

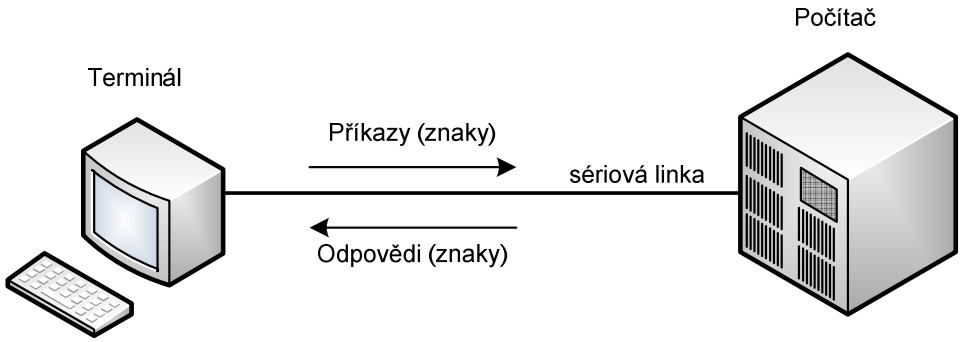
Důvodů používání textového režimu bylo několik. V počátcích sálových počítačů (spíše už „velkoskříňových“) a jejich programů nebyla navrhnuta a používána interakce s uživatelem v podobě grafického uživatelského rozhraní – GUI. Rovněž neexistoval schopný a výkonný hardware, který by toto umožňoval, a proto také terminály nebyly dostatečně „inteligentní“. Neexistence výkonného hardwaru znemožňovala přenášení grafiky do terminálu ze vzdáleného počítače. Bylo to prakticky nemožné, protože se jedná o náročný proces ve zpracovávání informací (nejen) procesorem a přenosu po síťovém vedení.

Na obr. 1.3a je zobrazen první CRT video terminál VT05 firmy DEC (Digital Equipment Corporation) z roku 1970. Obrázek 1.3b zobrazuje terminál VT100 (opět výrobce DEC) uvedený v roce 1978, který měl v sobě integrovaný 8bitový mikroprocesor 8080 firmy Intel. Tento terminál je dodnes softwarově emulován.

a) b)

Obr. 1.3: Historický terminál s obrazovkou: a) VT05, b) VT100

Vývoj terminálů a počítačů pokračoval dál. V terminálech se začaly používat podobné či stejné prvky jako u počítačů – mikroprocesory, grafické karty, záznamové mechaniky atd. Dnes jako terminál pro potřebu komunikovat se vzdálenou stranou nejčastěji používáme osobní počítač (PC), na kterém je na straně klienta spuštěna aplikace, pomocí které komunikuje se vzdálenou stranou – serverem. Na serveru běží (je spuštěna) aplikace nejčastěji typu „služba“ nebo „démon“, která vykonává přijaté příkazy od klienta a posílá zpět reakci klientovi nebo jinému zařízení, pro kterého je odpověď určena. O službách a démonech pojednává kapitola 2.1. Pro úplnost dodávám, že na protější straně nemusí být vždy nutně server, tzn. počítač, ale klient (osobní počítač v roli terminálu) může komunikovat a ovládat nějaký stroj např. lis, jeřáb, reklamní světelnou tabuli atd., který je vybaven rozhraním pro komunikaci s PC. Na obr. 1.4 je znázorněná komunikace terminálu s počítačem.



Obr. 1.4: Komunikace terminál - počítač

Moderní terminál se definuje jako elektronické nebo elektro-mechanické zařízení (rozhraní) pro zadávání příkazů a dat a zobrazování (opticky, akusticky nebo jinak) výstupních dat, která byla zpracována ve vzdáleném systému (nejčastěji počítač). Viz lit. [20].

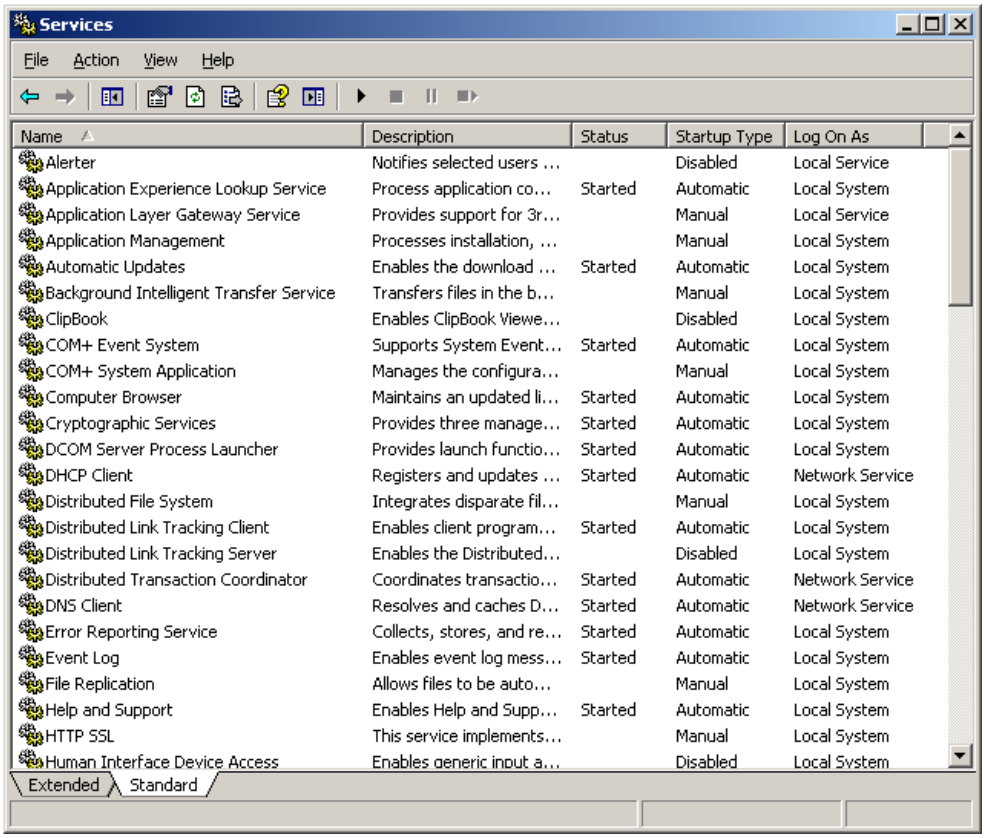
2 SOFTWAROVÉ PROSTŘEDKY

2.1 Služby a démoni

Aby mohl klient úspěšně komunikovat se serverem, je potřeba, aby na serveru běžela konkrétní aplikace, která klienta obslouží – čeká na data od klienta a pak je zpracuje. Jinak řečeno aplikace na serveru naslouchá na nějakém portu. Číslo portu rozlišuje typ služby. Jedná se o speciální aplikaci, které se označuje jako služba nebo démon. Mezi těmito názvoslovími je pouze rozdíl takový, že označení démon se používá v operačních systémech Unix-Linux a služba v operačních systémech (dále jen OS) Microsoft Windows.

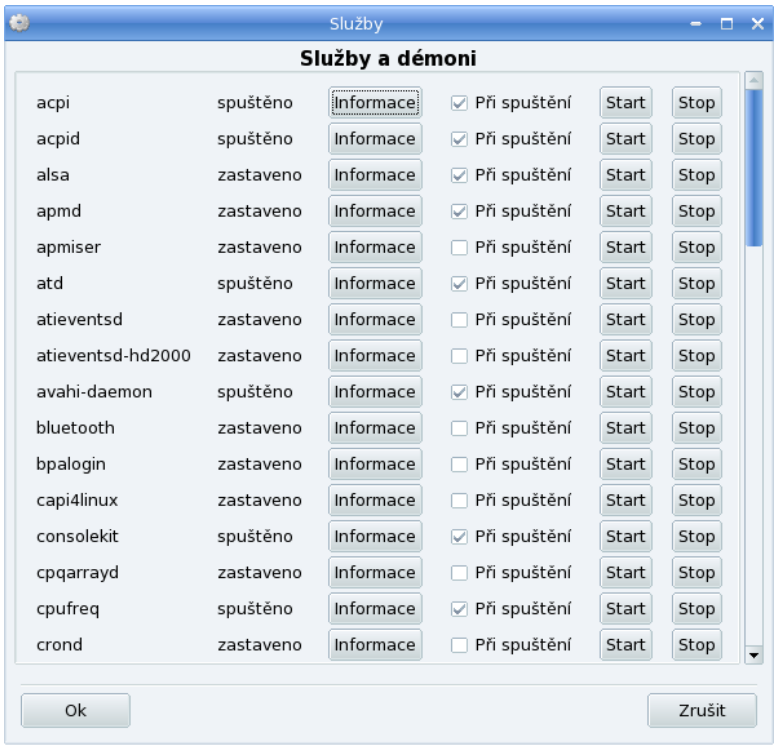
Rozdílů mezi běžnou aplikací a službou (démonem) je několik. Služba (démon) je po spuštění OS automaticky (je-li tak nakonfigurovaná) spuštěna. Správce systému nakonfiguruje potřebné služby, které mají běžet a ty se po zavedení OS ihned spustí bez potřeby přihlášení nějakého uživatele. Služba (démon) většinou nemá žádné grafické rozhraní a nijak interaktivně nekomunikuje s uživatelem (mnohdy o ni ani neví). Služby u OS MS Windows lze spouštět pouze od verze NT 4, 2000, XP a novějších. Službu (démona) lze spustit a zastavit kdykoliv si to vyžaduje situace a to bez restartování celého systému.

Pro úplnost uvádím částečný výpis služeb v OS Windows a Linux. Na obr. 2.1 je seznam některých služeb v OS Windows Server 2003 R2 Enterprise Edition, který byl vyvolán spuštěním programu `services.msc`. Pro prohlížení a správu služeb pro OS Windows existují také konzolové aplikace `net.exe` a `sc.exe`. Po standardní instalaci systému bylo z celkových 86 služeb aktivních 36.



Obr. 2.1: Stav služeb v OS Windows Server 2003 R2 Enterprise Edition

Na obr. 2.2 je zobrazen seznam některých služeb v OS Mandriva 2008. Ten byl vyvolán spuštěním programu `drakxservices`. Čistě konzolovou aplikaci pro práci se službami lze použít `services`. Po standardní instalaci systému bylo z celkových 57 služeb aktivních 30.



Obr. 2.2: Stav služeb v OS Mandriva 2008

Množství a typ používaných služeb (démonů) závisí na tom, k čemu je konkrétní počítač určen. Nepoužívané služby je vhodné zakázat (zastavit). Výsledným efektem je snížení zabraných systémových zdrojů a snížení rizika úspěšného útoku.

2.2 Terminálové služby a aplikace pro vzdálený přístup

Obecně terminálové služby jsou programové prostředky umožňující komunikaci mezi místní a vzdálenou aplikací. Uživatel (klient) prostřednictvím místní aplikace komunikuje se vzdálenou, která umožňuje **využití** – **ovládnutí** vzdálených zdrojů (serveru). Rychlost komunikace je závislá na výpočetním výkonu koncových uzlů a na přenosové schopnosti sítě, která spojuje tyto koncové uzly.

Scénář komunikace se obvykle skládá z těchto kroků:

1) Klient naváže komunikaci s protější stranou, která doposud byla v nečinnosti a naslouchala na specifickém portu. Pokud se jedná o víceuživatelské prostředí, vytvoří se nové vlákno obsluhujícího klienta. Provede se jednosměrná nebo obousměrná autentizace.
2) Druhá strana na základě přijatých přihlašovacích údajů rozhodne o přijetí nebo odmítnutí.
3) Pokud je přihlášení úspěšné, proběhne proces autorizace.
4) Pokud autentizace a autorizace proběhne v pořádku, protější strana vyčkává a následně po obdržení dat od klienta provede stanovené úkony.
5) Po splnění všech úkonů se klient „rozloučí“ s protější stranou, tzn. odhlášení, ukončení komunikace a protější strana se přepne do výchozího naslouchacího režimu, v případě víceuživatelského prostředí se obslužné vlákno aplikace ukončí.

Vlastnosti, kterými musí disponovat každá moderní služba pro vzdálený přístup, jsou shrnuty do následujících bodů:

- **Autentizace**. Účelem je ověření, zda uživatel (klient) má přístup k systému. To je nejčastěji zabezpečeno jménem a heslem uživatele. Je možno použít jednosměrnou nebo obousměrnou autentizace (ověření identity obou komunikujících stran).
- **Autorizace a účtování - řízení přístupu**. Po úspěšném procesu autentizace je potřeba zjistit práva přihlášeného uživatele. Jedná se např. o práva přístupu do konkrétního místa, práva modifikace, časový limit, právo využívat konkrétní službu.
- **Šifrování**. Je důležité tam, kde je zapotřebí přenášená data chránit před možným odposlechem (zcizením) a záměrným pozměněním (zfalšováním) dat během přenosu. Existuje mnoho šifrovacích technik, které tuto ochranu umožňují. Jedná se o symetrické (soukromým klíčem) a asymetrické (veřejným klíčem) šifrování.
- **Bezpečnost přenosu**. Je důležité, aby příkazy zadané uživatelem došli k cíli v přesném pořadí, tak jak byly zadány. Rovněž odpovědi z protější strany, aby byly na straně klienta interpretovány ve správném pořadí. Také je nutno zabezpečit přenášená data proti ztrátě nebo změně během přenosu. Pro kontrolu integrity zpráv existují různé algoritmy jako např. CRC, MAC, HMAC, hash – SHA1, MD5 atd. Vhodným řešením je použití přenosového protokolu TCP (*Transmission Control Protocol*).

Použití terminálových služeb má velký význam a uplatnění. Zde je popis výhod a nevýhod, které přináší terminálové služby.

Výhody:

- ✓ **Úspora financí za vylepšení (upgrade) starších počítačů.** Možnost připojení starších nevýkonných klientů. Ty pracují s aplikacemi přímo na serveru. Na klientském terminálu je minimálně zatěžován procesor. Veškeré provádění výpočtů se provádí na serveru. Klient tak získává možnost pracovat s nejmodernějšími programy instalovanými na serveru.
- ✓ **Dálkový přístup a administrace.** Správa serveru nebo klienta je možná z jakéhokoliv počítače na síti, tedy nerozhoduje fyzická vzdálenost. To administrátorům značně zjednodušuje práci a šetří čas. Uživatel je pouze omezen potřebným oprávněním. Např. administrátor vzdálený stovky kilometrů od serveru, kde nastala nějaká softwarová chyba, může tuto chybu napravit využitím mobilního zařízení (notebook, PDA, mobilní telefon) připojeného k internetu. Tento scénář je možný pouze v případě, pokud není potřeba fyzické přítomnosti, např. fyzického zasunutí datového média (CD, DVD atd.) do mechaniky.
- ✓ **Téměř neomezený počet připojených počítačů.** Pokud je na serveru provozován víceuživatelský operační systém, je možné připojit mnoho počítačů a přitom každý pracuje nezávisle na ostatních připojeních – každému uživateli se spustí vlastní relace. Jednotlivý uživatelé o sobě vůbec neví - na serveru není vizuálně nic poznat, všechny úlohy vykonávané s terminálovými službami probíhají v pozadí. Každý klient využívá své vlastní nastavení a má nainstalované vlastní aplikace. Pro připojení více počítačů je vhodné rychlé síťové připojení a především výkonný server.
- ✓ **Centralizace.** Všechny operace jsou prováděné na serveru, rovněž všechny aplikace jsou nainstalované na jediném serveru (redukce finančních nákladů na vlastnictví). Tím je získání zjednodušení administrace, dohled, zálohování a obnovy systému. Významná redukce časové náročnosti instalace a správy software. Všichni klienti mají vždy přístup ke stejné verzi programů – zjednodušení aktualizace.
- ✓ **Nezávislost hardware a software.** Na klientu/serveru může běžet rozdílný operační systém (Windows, Unix, Linux, Solaris atd.), stejně tak i architektura může být rozdílná (např. Macintosh, Alpha, SPARC, PowerPC atd.). Může být použit i např. PDA, mobilní telefon. Podmínkou je dostupnost aplikace pro konkrétní OS.

Použití terminálových služeb přináší také následující nevýhody a nebezpečí:

- **Neoprávněný přístup.** Protože se jedná o síťovou komunikaci, je zde riziko neoprávněného přístupu. Špatně zabezpečená služba a celá komunikace zvyšují pravděpodobnost zjištění identity a hesla. Útočník může provést útoky a způsobit katastrofu (znepřístupnění serveru).
- **Útoky.** Existence možností útoků na službu. Vlivem útoku může být služba nedostupná – tzv. DoS útoky, to souvisí s kvalitou zabezpečení, návrhu a implementace aplikace. Ale také díky chybě může útočník získat neomezený přístup k systému. Je vhodné nepoužívané služby vypnout, sníží se tím riziko napadění.
- **Centralizace.** Pokud je použit model centralizace, tzn. všechny aplikace, data a výpočetní kapacity jsou soustředěny na jednom místě, tak při selhání – havárii tohoto centralizovaného uzlu, jsou ochromeny všechny připojené uzly.
- **Výkonnost.** Výkonnost je závislá na mnoha faktorech. Celý proces je tak rychlý jak je nejpomalejší článek v komunikačním řetězci. Např. scénář: pomalá síť, výkonný server, nebo naopak. Ačkoliv je server velmi výkonný a provede úkon velmi „svižně“, distribucí příkazů a odpovědí vlivem pomalé sítě degraduje výkon serveru. Uživatel pak může díky netrpělivým čekáním na výsledek svého příkazu způsobovat chyby a snížit efektivitu práce.

2.3 Typy terminálových služeb (aplikací)

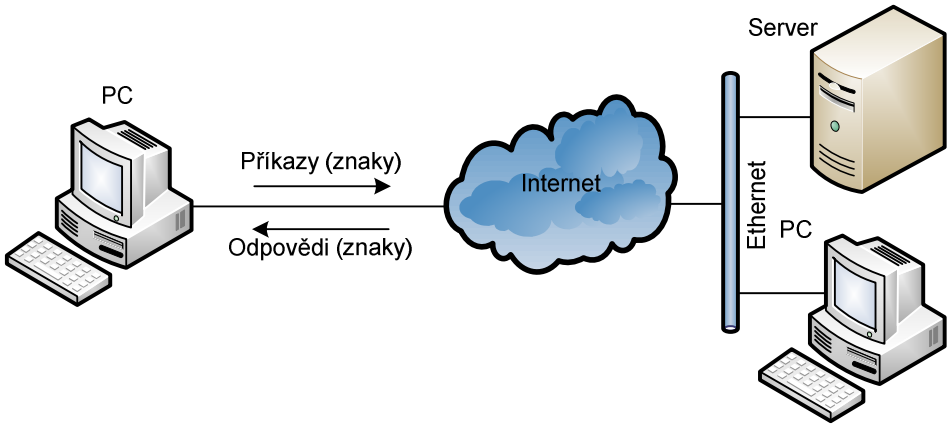
Základní rozdělení terminálových aplikací lze provést podle charakteru přenášených (a interpretovaných) informací. Rozdělení je tedy na **znakové**, **grafické** a **multimediální**.

Terminálové aplikace znakové pracují výhradně se znaky – texty a jsou nejstarším řešením. Přenáší se pouze viditelné a řídící znaky. Řídícím znakem je např. znak „nový řádek“, který se skládá z jednoho nebo dvou znaků. Jedná se o znaky CR (Carriage Return – návrat vozíku) a LF (Line Feed – posun papíru o řádek vzhůru). Tyto znaky pochází z období dálnopisné komunikace. Hlavní předností znakových terminálů je jejich snazší implementace, jednoduchost, nenáročnost na hardware a hlavně malý datový tok při vzdálené správě. Nevýhodou může být nižší komfort při práci s aplikací související s pouze textovým zobrazením. Uživatel do terminálu postupně vkládá pomocí klávesnice příkazy.

Na obr. 2.3 je znázorněná klasická komunikace počítače (emulující terminál) s protější stranou. Omezení však neplatí pouze na osobní počítač, viz níže. Jako nejčastější kódování abecedy znaků je použita normalizovaná ASCII tabulka znaků. Ovšem z pohledu aplikace nic nebrání použití modernějšího kódování, např. Unicode a příkazy psát třeba v kanji [kandži]. Nejznámějšími aplikacemi (protokoly) pro vzdálený přístup jsou telnet, rlogin, rsh a ssh.

Terminálové aplikace grafické pracují s grafickou informací. Uživatel získává obrazy („print-screeny“) od protější strany. Uživatel pomocí klávesnice a polohovacího zařízení (nejčastěji myši) posílá příkazy protější straně. Přenáší se pouze kód stisknuté a uvolněné klávesy, pozice (souřadnice) kurzoru myši na obrazovce, kód stisknutého a uvolněného tlačítka na myši a informace o „scrollovacím“ kolečku na myši. Tyto příkazy na protější straně (na ovládaném počítači) jsou provedeny jako by byly zadány na místní klávesnici a myši. Ovládaný počítač posílá zpět obrazy uživateli, kterému se zobrazují.

Z popsaného principu je vidět, že zátěž – datový tok - je značně nesymetrický. Od uživatele, který ovládá vzdálenou stanici, putuje několik bajtů až max. stovek bajtů. Druhým směrem je tok značně velký a proto je nutné použít jednu nebo několik metod pro snížení objemu dat - kompresi dat.

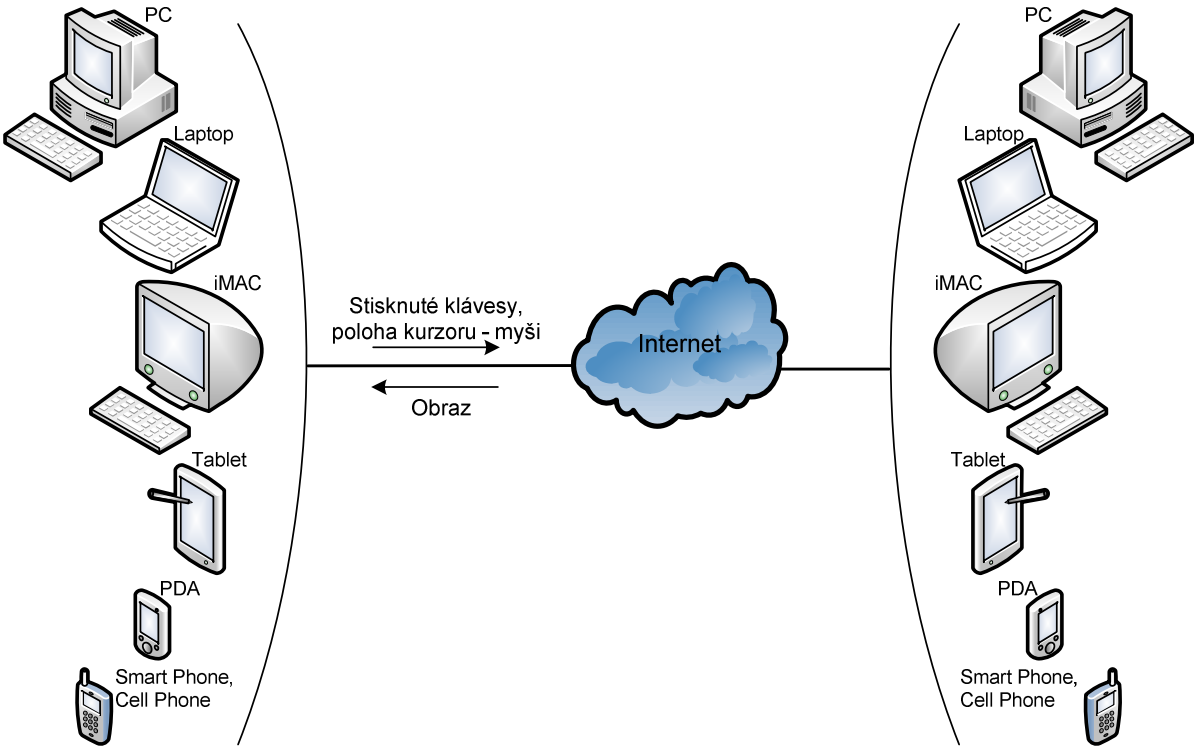


Obr. 2.3: Klasická komunikace pomocí virtuálního (textového) terminálu

Velkou výhodou grafických terminálových aplikací je komfort a daleko větší jednoduchost ovládání oproti znakovým terminálům. Uživatel nemusí znát žádné příkazy. Další významnou výhodou je možnost ovládat jakýkoliv počítač s jakýmkoliv operačním systémem, pokud existuje pro tento OS aplikace, která to umožňuje. Např. uživatel pracující v OS Windows 98, ME, XP atd. se může připojit a pracovat s počítačem, kde je spuštěn OS Linux, MAC OS X Leopard, atd.

Nevýhodou, díky přenosu obrazů, je větší přenos dat směrem k uživateli. Na obr. 2.4 je znázorněno využití moderních zařízení k ovládání protější strany. Jak je vidět, druhou stranou můžou být ty samé technické zařízení, pokud splňují technické a softwarové požadavky (připojení k internetu, podporující aplikaci, API). Např. je možné pomocí mobilního telefonu ovládat počítač a získávat z něj obraz. Ovšem v tomto případě je třeba řešit několik problémů jako např. různá rozlišení obrazovky na počítači a mobilním telefonu.

Terminálové aplikace multimediální jsou totožné s grafickými, ovšem jejich funkcionalita je rozšířená o možnost přenosu nejenom obrazu, ale i zvuku, schránky, sdílení vstupních a výstupních periferií (tiskárna, scanner) atd.



Obr. 2.4: Využití zařízení k ovládání protější strany

3 KOMPRESE DAT

Tato kapitola je úvodem do technik komprese a stručně popisuje důvody, proč je nutné použít kompresi dat a také popisuje zásady a doporučení speciálně vhodné při návrhu terminálové služby (aplikace) pracující s přenosem obrazu. Dále jsou popsány vlastnosti informačního toku, kterých lze při kompresi s výhodou užít, ale i problémy, které souvisí s konkrétními vlastnostmi.

3.1 Pojem komprese

Termíny informace, data a znalosti lze v běžném hovoru považovat za synonyma (jsou natolik příbuzné, že je prakticky nelze definovat jinak než pomocí nich samých). Informace je velkou cenností lidstva. Některé informace jsou pro člověka nepostradatelné. Postupem času našeho života, ale i vývoje lidstva všeobecně se informace hromadí, přibývají. Již odpradávna co vznikl „inteligentní" život se informace ukládala, archivovala atp. Způsobů existuje několik. Například v dávných dobách to byly malby na stěnách jeskyně, přenos znalostí z generace na generaci, knihy psané, tištěné a v moderní době digitalizace. Svět po příchodu a aplikaci výpočetní techniky nejen do běžného života přinesl nespočetná usnadnění, ulehčení práce, ale paradoxně i problémy. Jeden z problémů je například velikost informace, kterou dokážeme uchovat.

Množství informace je jeden z problémů, proto se snažíme řešit tuto situaci. S tím úzce souvisí pojem komprese. Komprese neboli komprimace je snažení se o zmenšení objemu množství informace. Výsledkem tohoto snažení je komprimovaná informace, která má menší velikost než informace původní. Tohoto stavu docílíme tak, že jsme schopni najít redundanci či irelevanci a odstranit ji.

3.2 Důvody komprese

Důvodů, proč tolik potřebnou informaci zmenšit je několik:

- komprimace šetří úložný prostor na paměťovém médiu
- přenos informace o menší velikosti je rychlejší, z toho vyplývají další výhody, jako poplatky za konektivitu

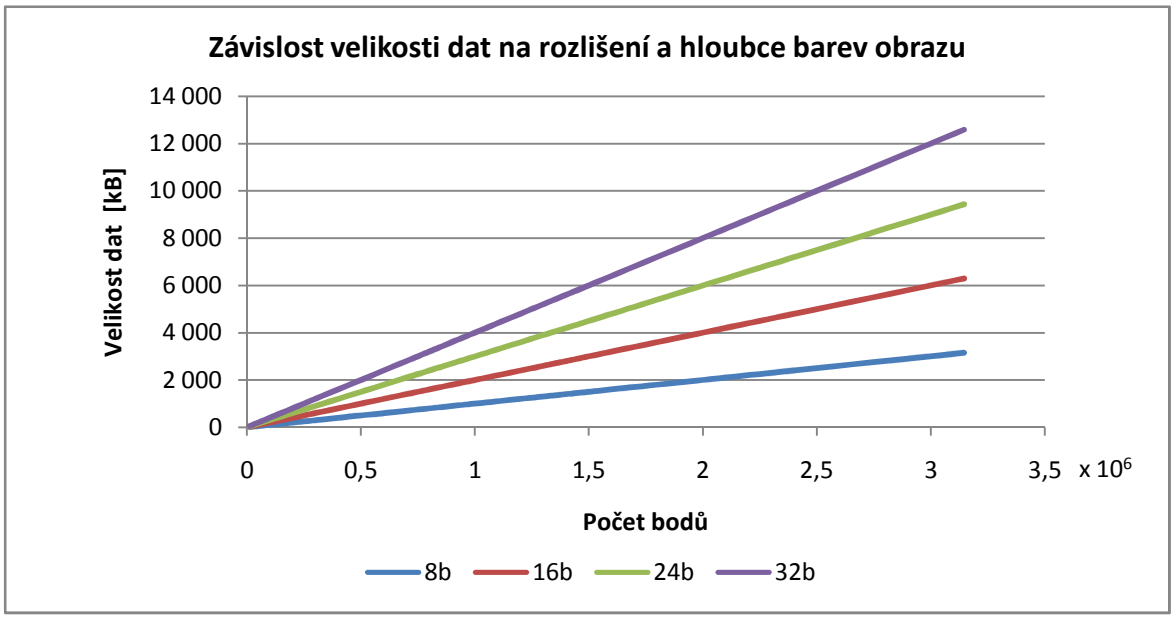
Kompresní algoritmy (programy) bývají nejčastěji obohaceny o funkce:

- ochrana dat heslem proti neoprávněnému přístupu
- ochrana dat ochranným kódem proti poškození souboru (např. CRC)
- velké množství souborů je uloženo v jediném souboru - archívu (přináší větší přehlednost v organizaci dat)
- rozdělení velkého (kapacitně) souboru na více menších

Následující tabulka (tab. 3.1) a zobrazená grafická závislost (obr. 3.1) ukazuje, jak se objem dat zvětšuje v závislosti na rozlišení obrazu (počet obrazových bodů – pixelů) a na bitové hloubce barev (počet zobrazitelných barev). Dnes se pro běžnou práci na PC používá rozlišení 1024×768 a 1280×1024 při hloubce barev 32 bitů. Přenášení těchto celých obrazů po internetu by způsobilo značné zatížení a zpomalení sítě. Nejvýrazněji by byla poznamenána mobilní zařízení s pomalým (a drahým) připojením k internetu. Potřeba snížit datový tok je nepopíratelná.

Tab. 3.1: Závislost velikosti dat na rozlišení a hloubce barev obrazu

rozlišení		počet bodů	velikost při hloubce 8b [B]	velikost při hloubce 16b [B]	velikost při hloubce 24b [B]	velikost při hloubce 32b [B]
šířka	výška					
128	96	12 288	12 288	24 576	36 864	49 152
160	120	19 200	19 200	38 400	57 600	76 800
192	144	27 648	27 648	55 296	82 944	110 592
256	192	49 152	49 152	98 304	147 456	196 608
320	240	76 800	76 800	153 600	230 400	307 200
512	384	196 608	196 608	393 216	589 824	786 432
800	600	480 000	480 000	960 000	1 440 000	1 920 000
1024	768	786 432	786 432	1 572 864	2 359 296	3 145 728
1152	864	995 328	995 328	1 990 656	2 985 984	3 981 312
1280	1024	1 310 720	1 310 720	2 621 440	3 932 160	5 242 880
1400	1050	1 470 000	1 470 000	2 940 000	4 410 000	5 880 000
1600	1200	1 920 000	1 920 000	3 840 000	5 760 000	7 680 000
2048	1536	3 145 728	3 145 728	6 291 456	9 437 184	12 582 912



Obr. 3.1: Graf závislosti velikosti dat na rozlišení a hloubce barev obrazu

3.3 Rozdělení komprese

3.3.1 Komprese bezeztrátová a ztrátová

Základní rozdělení typu komprese je na bezeztrátovou (lossless compression) a ztrátovou (lossy compression). Princip bezeztrátové komprese se používá tam, kde si nelze dovolit jakoukoli ztrátu dat – odstraňujeme redundanci. Jde například o data textová nebo binární. Ztráta jediného bitu vede ke ztrátě hromadné – data jsou nepoužitelná. Platí pravidlo, že data po dekompresi mají naprosto stejné vlastnosti jako před kompresí.

Naproti tomu ztrátová komprese odstraňuje irelevanci a aplikuje se u dat, kde nám nevadí, že nenávratně přijdeme o určité množství dat. U ztrátových kompresí se s výhodou využívá faktu, že lidské smysly (zrak, sluch) nejsou dokonalé. Např. lidské oko má omezenou rozlišovací schopnost jak ve vztahu k barevné hloubce tak i k obrysovým detailům. Barvy blízko ležících bodů oko průměruje. Obvykle ztrátová metoda komprese dosahuje výrazné úspory dat. Jedná se o data, která uchovávají statickou grafiku, video, zvuk. V případě grafiky se často používá komprimovaný formát JPG (Joint Photographic Experts Group – používá se algoritmus diskrétní kosinové transformace DCT (Discrete Cosine Transform)). Pro ztrátovou kompresi platí důležité omezení – nelze dekomprimovat do původního stavu, protože při komprimaci došlo k definitivní ztrátě některých dat.

3.3.2 Užití bezeztrátové a ztrátové komprese v grafických terminálech

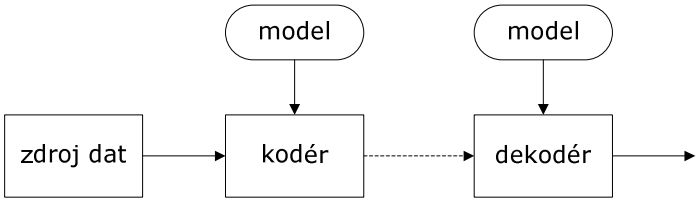
V případě grafických terminálů nelze z hlediska ztrátovosti/bezztrátovosti tyto metody bezmyšlenkovitě užít kdekoliv. Z výše popsaného je patrné, že informace přenášené od uživatele směrem k ovládané straně (serveru) nelze ztrátově komprimovat, protože se jedná např. o data popisující klávesu, která byla stisknuta, či nové souřadnice polohy ukazovátka (kurzoru) myši. Ztrátou pouhého znaku nebo informace o přesné poloze kurzoru by se stal stroj neovladatelným. Tyto data nemají žádnou irelevanci, ale můžou mít jistou míru redundance. Otázkou zůstává, jak je velká redundance a zda je nutné tyto data komprimovat.

Naproti tomu obraz (nebo zvuk) obsahuje díky omezené rozlišitelnosti oka (nebo ucha) irelevanci a často i redundanci. Proto je velice vhodné použít jak ztrátové tak bezeztrátové komprese. Rozhodnutí, zda užít ztrátovou nebo bezeztrátovou metodu (nebo obě dvě), závisí především na technických možnostech daného zařízení. Každý algoritmus je jinak náročný a ke svému provádění potřebuje různě výkonný procesor a dostupnou operační paměť a proto při implementaci (tvorbě programu) je nutné rozhodnout jaká metoda je nejvhodnější.

3.3.3 Adaptivní a neadaptivní komprese

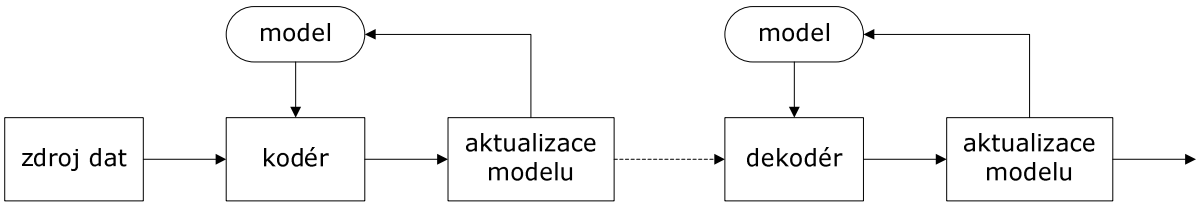
Adaptivní a neadaptivní komprimační algoritmy se rozlišují podle své schopnosti přizpůsobit se charakteru dat, se kterými manipulují.

Neadaptivní algoritmy (např. statické Huffmanovo, aritmetické kódování, LZ77) většinou obsahují předdefinované slovníky nebo řetězce znaků, o kterých je známo, že jejich pravděpodobnost výskytu v souborech dat je vysoká, nebo potřebnou informaci nejprve zjistí ze souboru. Neadaptivní algoritmus, určený pro komprimaci anglického textu, bude obsahovat slovníkové řetězce např. the, of, and, které bude nahrazovat předdefinovaným znakem. Tento způsob komprimace by však byl velmi neúčinný pro text v jiném jazyce, natož kdyby měl zpracovat grafická nebo binární data. Použitím tohoto algoritmu na správných datech docílíme výborného komprimačního poměru a zkrátíme také čas komprimace na minimum. Na obr. 3.2 je zobrazen model statické komprese a dekomprese.



Obr. 3.2: Statická metoda komprese a dekomprese

Adaptivní algoritmus (např. Lempel-Ziv-Welch, adaptivní Huffmanovo kódování, adaptivní aritmetické kódování) je naproti tomu schopen dosáhnout určité nezávislosti na komprimovaných datech. Takové algoritmy neobsahují žádné statické slovníky řetězců a ani nezjišťují dopředu žádné informace. Algoritmy si budují tyto slovníky pro každý komprimovaný soubor dat znovu dynamicky v průběhu komprese i dekomprese. Na obr. 3.3 je zobrazen model adaptivní komprese a dekomprese. Obecně platí, že adaptivní algoritmy platí za svou přizpůsobivost a větší šíři použití menší rychlostí ve srovnání se specializovanými neadaptivními algoritmy.



Obr. 3.3: Adaptivní metoda komprese a dekomprese

Díky tomu, že adaptivní kompresní algoritmy při zahájení komprese nepotřebují znát model, ani četnosti znaků, je vhodné je použít pro kompresi vstupních proudů, jako např. streamové vysílání (audio, video), real-time aplikace. Podobně jak u komprese, tak i u dekomprese, je používaný model na začátku procesu ve stejném (inicializovaném) stavu.

3.3.4 Užití adaptivní a neadaptivní komprese v grafických terminálech

Adaptivní metody oproti statickým jsou náročnější na systémové zdroje zařízení (procesor, paměť), avšak dosahují lepších kompresních ukazatelů. Proto je při návrhu programu nutné vzít v potaz rychlostní rozdíly mezi adaptivní a neadaptivní verzí, protože grafické terminály jsou real-timové aplikace a zdlouhavé zpracování informací by mělo za následek dlouhé odezvy.

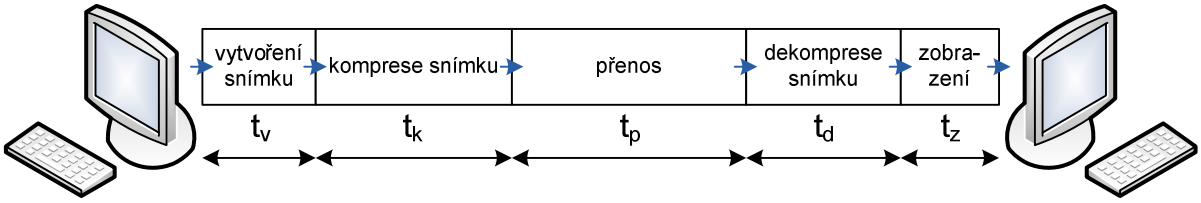
3.3.5 Symetrická a asymetrická komprese

Rozdělení na symetrické a asymetrické komprimační algoritmy je založeno na porovnání množství práce (uplynulá doba), která se podle algoritmu vykoná při kompresi a dekompresi dat. Pokud je doba (a tím většinou i počet operací) potřebná pro kompresi i dekompresi dat přibližně stejná, jedná se o symetrickou kompresi.

Některé programy jsou však záměrně konstruovány jako asymetrické. Některé algoritmy provedou větší práci při kompresi a jiné při dekompresi. Např. při archivaci dat často komprimujeme, ale dekomprimaci provádíme zřídka nebo opačně, kdy často používáme zkomprimovaná data.

3.3.6 Užití symetrické a asymetrické komprese v grafických terminálech

V případě terminálových řešení není rozhodující, zda bude užita symetrická nebo asymetrická metoda komprese. Jedná se o komunikační řetězec, kde na jedné straně se provede určitý počet kompresních operací a na straně druhé se provede určitý počet dekompresních operací. Nehraje roli, jak dlouho trvá jednotlivý proces komprese a dekomprese, ale důležitá je celková doba. Výše popsané ilustruje obr. 3.4.



Obr. 3.4: Naznačení časových operací

Na obr. 3.4 jsou naznačeny procesy:

- t_v – čas potřebný pro vytvoření snímku (uložení do paměti v nějakém vhodném datovém uspořádání, např. pole integer, viz dále)
- t_k – čas potřebný pro kompresi obrazu
- t_p – doba přenosu dat
- t_d – čas potřebný pro dekompresi obrazu
- t_z – čas pro zobrazení dat (modifikace před zobrazením, úpravy)

3.3.7 Komprese fyzická a logická

Rozdíl mezi fyzickou a logickou komprimací spočívá v tom, zda komprimační algoritmus při komprimaci přihlíží nebo nepřihlíží k logické hodnotě dat. Obvykle je výpočetně náročnější a dosahuje větší úspory. Pokud chceme použít logickou kompresi, musí pro každý specifický typ dat (text, zvuk, obraz, video atd.) existovat samostatný algoritmus. Nelze použít algoritmus pro kompresi textu na video data atd.

Fyzická komprese probíhá bez zřetele na logiku dat, se kterými algoritmus pracuje. Vytváří se nová sekvence znaků. Bez znalosti dekomprimačního algoritmu je informační hodnota komprimovaných dat nulová.

3.3.8 Užití fyzické a logické komprese v grafických terminálech

Jak bylo několikrát zmíněno, při přenosu dat v grafických terminálech se vyskytují minimálně 3 druhy dat:

- znaky (stisknuté tlačítka na klávesnici a myši)
- poloha kurzoru (souřadnice X a Y)
- obraz

V případě multimediálních (grafických, doplněné o další možnosti přenosu) to jsou např.:

- soubory
- tisk (tisková fronta – vstupní data do tiskárny)
- zvuk

V následujícím textu je popsán přístup k využití logické komprese znaků a polohy.

Vlastnosti a logická komprese informací „tlačítek“

Data, generovaná jako událost na stisknutí nebo uvolnění tlačítek klávesnice a myši, nelze ztrátově komprimovat. Po vygenerování události je nutné tato data okamžitě odeslat protější straně. Nemůže tedy docházet ke shlukování dat, např. v nějakém bufferu, který by se odeslal až po jeho naplnění. Např. pokud se uživateli zobrazí dialog, který vybízí stisk libovolné klávesy

pro pokračování v programu, tak po zmáčknutí tlačítka se ihned odešle paket. Uživatel po stisknutí tlačítka ještě tlačítko uvolní a to má za následek poslání nového (jiného) paketu.

Standardně pro přenos události jednoho znaku je zapotřebí celkem 3 bajty. Dva bajty pro znak a bajt identifikující typ události (jedná se o znaky stisknutí, uvolnění, polohu, případně něco jiného). Např. v jazyce JAVA stisknutí klávesy „SHIFT“ generuje tzv. virtual key kód 16, „F1“ kód 112, „Windows“ kód 524 a „F18“ (u multimediální klávesnice) kód 61445.

Účinnost jakékoliv použité komprese je minimální, protože aplikační náklad „tlačítek“ je tvořen pouze třemi bajty, které se pro přenos musí „zabalit“ do datové jednotky (paketu), která je několikanásobně větší. Už jen hlavička IP a TCP má dohromady minimálně 40 bajtů. Pro případnou implementaci komprese je nejvhodnější adaptivní algoritmus.

Vlastnosti a logická komprese polohy kurzoru myši

Pro přenos polohy kurzoru je zapotřebí celkem pět bajtů. Jeden bajt opět pro identifikaci a po dvou bajtech pro souřadnici X a Y. Nutnost použít právě dva bajty – datový typ short, tj. 16 bitů pro X, Y souřadnici vychází z toho, že pro vyjádření čísla, udávající počet bodů jedné strany obrazovky (šířky nebo výšky, viz tab. 1), postačují právě dva bajty. Tedy při použití dvou párů bajtů, lze adresovat polohu kurzoru myši až do 65536 × 65536. Pro kompresi dat polohy kurzoru myši lze použít tyto metody:

- klasickou statistickou kompresi (na základě pravděpodobnosti výskytu symbolů)
- snížení adresovatelného rozlišení (2 × 16 b) na skutečně používané
- přenos rozdílových hodnot souřadnic
- záměrná ztráta posunu

Snížení adresovatelného rozlišení na skutečně používané:

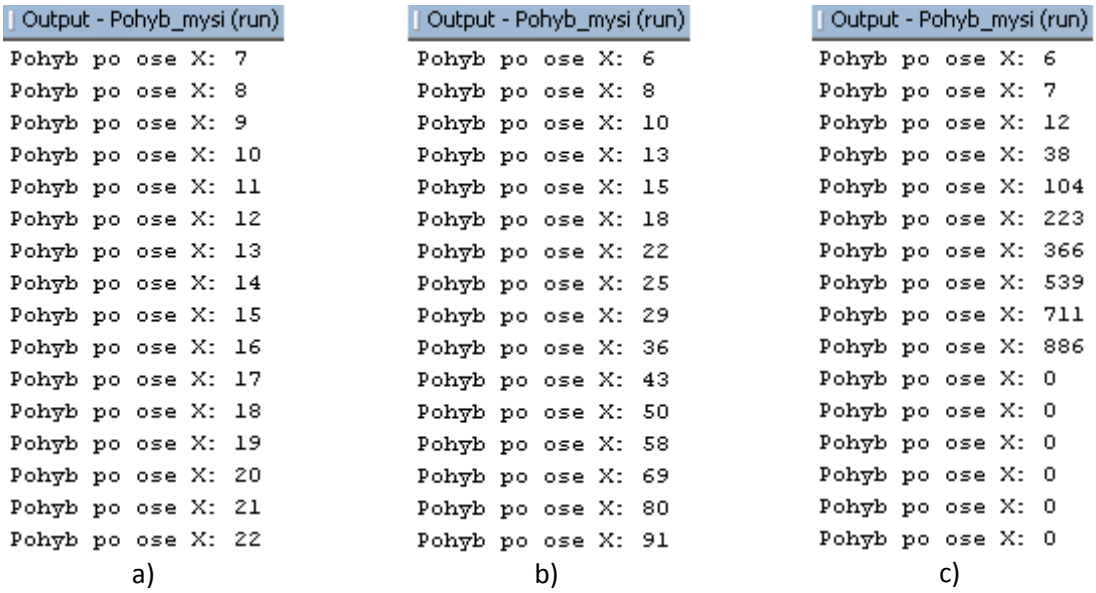
Jak bylo řečeno, pro jeden rozměr obrazovky je použito 16 bitů, které umožňují definovat polohu kurzoru kdekoliv v intervalu <0;65535> - 65536 diskrétních poloh. Pro polohu je zapotřebí dvou souřadnic – celkem 32 bitů (4 B). Dnešním standardem jsou obrazovky s daleko nižším rozlišením, běžně 1024 × 768 a 1280 × 1024, tedy pro vyjádření těchto hodnot je zapotřebí daleko méně bitů. Jako možné řešení se nabízí 12 bitů pro každou souřadnici, tj. celkem 24 bitů (3 B). Při užití 12 bitů, se může pracovat s plochou 4096 × 4096. Užitím této navržené metody by se ušetřil 1 bajt.

Přenos rozdílových hodnot souřadnic:

Tím, že se nebudou posílat absolutní ukazatele pozice myši, ale pouze rozdíly od předchozí zaznamenané hodnoty, je opět možné ušetřit několik bitů, protože číslo vyjadřující rozdíl je podstatně menší. Přenášené číslo by pak mělo formát: [znaménkový bit; rozdílové číslo]. Tuto metodu ovšem komplikuje jeden důležitý problém. Akumulace chyb. Ztrátou jediného paketu budou další rozdílové informace chybně zpracovány. V případě použití této techniky by bylo zapotřebí použít TCP protokol.

Na obr. 3.5 je demonstrativní příklad snímání pohybu myši aplikací pro různé rychlosti pohybu. V prvním testu (obr. 3.5a) aplikace zaznamenala všechny události posunu. Pro vyjádření rozdílu by stačil pouze 1 bit a to znaménkový. Ten by informoval o kladné nebo záporné jednotkové změně (posun myši vpravo nebo vlevo).

Ve druhém testu (obr. 3.5b) je největší rozdíl 11. V posledním testu (obr. 3.5c) je zaznamenán velmi rychlý pohyb. Největší rozdíl je 175 a pro jeho zakódování je potřeba 8 bitů. Jak bylo výše popsáno, nelze zaručit, že rozdíl nebude větší. Z toho plyne, že je vhodné použít více jak 8 bitů pro kódování rozdílu.



Obr. 3.5: Snímání pohybu myši: a) pomalý pohyb, b) rychlý pohyb, c) velmi rychlý pohyb

Záměrná ztráta posunu:

Touto metodou lze nejúčinněji snížit velikost přenášených dat, protože úsporou je celý paket (rámec). Princip činnosti je takový, že aplikace (přesněji vlákno aplikace) snímající posun kurzoru myši, provádí toto snímání např. periodicky v určitém intervalu. V době nečinnosti se vlákno může uspat (zvýšení výkonu). Vhodná délka uspání vlákna (po získání polohy kurzoru), tedy časový rozestup mezi snímáním, je v řádu milisekund až stovek milisekund. Čím větší hodnota, tím více se ušetří, ovšem interpretovaný pohyb na protější straně bude zkreslen. Na obr. 3.6a je zachycena dráha posunu kurzoru na vysílající a přijímací straně. Na obr. 3.6b je zobrazen interpretovaný pohyb při vzorkování 10 ms (po získání polohy se vlákno uspalo na dobu 10 ms). Na obr. 3.6c je zobrazen interpretovaný pohyb při vzorkování 500 ms. V závislosti na druhu práce (za jakým účelem je vytvořeno terminálové spojení) se různí vhodná velikost prodlevy při vzorkování. Např. při běžné správě systému (klikání na tlačítka, nastavování voleb atd.) je 200 ms dostačující, ale při úpravě obrazu (kreslení) je tato hodnota velká.

Pro snížení velikosti dat posílaných při pohybu kurzoru myši, je vhodné použít kombinaci všech tří metod. Metody pro kompresi obrazu z prostorových důvodů nemohou být uvedeny. V kap. 5 je popsána použitá metoda komprese obrazu.

Obr. 3.6: Záznam pohybu myši na protější straně při vzorkování: a) bez vzorkování, b) 10 ms, c) 100 ms

4 EXISTUJÍCÍ ŘEŠENÍ PRO VZDÁLENÉ OVLÁDÁNÍ PC

V této kapitole jsou popsána dostupná a nejčastěji využívaná řešení pro vzdálené ovládání počítače. Je zde proveden rozbor z hlediska funkcionality, schopnosti redukce objemu posílaných dat a jsou zde popsány výhody a nevýhody řešení.

4.1 Metoda testování

Pro objektivní testování aplikací z hlediska komprese datového toku jsem vytvořil metodu, která zaručí, že testované aplikace budou mít naprosto stejné výchozí prostředí a průběh komunikace bude vždy naprosto stejný. Metoda je navržena tak, aby bylo možno účinně testovat vliv využití cache paměti snímků. Do paměti cache snímku se ukládají již zpracované a poslané obrázky. Cache se nachází jak na straně vysílající (generující snímky obrazovky), tak i na straně přijímací. V případě nového snímku se testuje, zda tento snímek nebyl v minulosti již zpracován. Pokud ano, pošle se pouze informace identifikující umístění snímku v cache. Použitím cache snímku se tedy také sníží datový tok.

Smyslem metody testování je, že na vysílající straně se zobrazují snímky vhodného obsahu, pořadí a se specifickým časováním. Tím se simuluje činnost uživatele (klikání na objekty, přesun oken atd.). Délka testu je 81 sekund, během něhož se zobrazí 81 snímků.

4.2 Vzdálená plocha v MS Windows

Vzdálená plocha umožňuje v operačních systémech MS Windows provádět vzdálenou správu. To je možné provádět odkudkoliv z libovolného počítače v síti. Tento nástroj, který je přítomen v operačních systémech řady Windows Server 2003, Windows Vista a Windows XP, podporuje přihlášení ke vzdálenému systému i z klientského počítače s dřívější verzí operačního systému Windows. Tuto službu je nejprve nutno povolit, standardně po instalaci systému je deaktivovaná.

Vzdálená plocha je podmnožina funkcí, které zajišťuje Terminálová služba. Terminálové služby Microsoft používají protokol RDP (Remote Desktop Protocol), který je založen na protokolu T. 120. V následujícím odstavci jsou shrnuty vlastnosti protokolu RDP. Podrobněji viz lit [17], [19].

Vlastnosti a funkce protokolu RDP:

- multikanálový protokol – připojuje se k počítači, na kterém je spuštěna Terminálová služba
- existuje mnoho klientů pracující s RDP – např. klient pro Windows Mobile, Linux, FreeBSD, Solaris, Mac OS X
- možnost existence až 64000 kanálů
- server naslouchá na TCP portu 3389, který je ovšem možno změnit
- podporuje přenos obrazů s 32 bitovou hloubkou
- používá šifrovací algoritmus RC4 (56 a 128 bitů)
- podporuje protokol TLS (Transport Layer Security)
- přenos a přehrávání zvuku na lokálním PC
- umožňuje mapování – přesměrování souborového systému, tiskárny, sériového a paralelního rozhraní
- sdílení schránky mezi lokální a vzdálenou stanicí
- umožňuje nastavení velikosti datového toku (komprese, bitmapová cache)
- podporuje více monitorů

Pro připojení k terminálovým službám je zapotřebí klient - program, který Microsoft vyvíjí pouze pro svoji Windows platformu. Existují projekty XRDP a desktop, což jsou open source klienti protokolu RDP – díky nim je umožněno připojit se ke vzdálené ploše Windows z Linuxu.

Pro vzdálený přístup do Windows vzdálené plochy, je možno využít některé z dostupných řešení:

- aplikace Vzdálená plocha (standardně dostupná ve Windows XP: Start, Všechny programy, Příslušenství, Připojení ke vzdálené ploše)
- webové připojení ke vzdálené ploše (pomocí ActiveX) – umožňuje připojení k terminálovým službám pomocí Internet Exploreru (ostatní WWW prohlížeče nejsou podporované)
- programy třetích stran (rdesktop, xdrp, DOSRDP atd.)

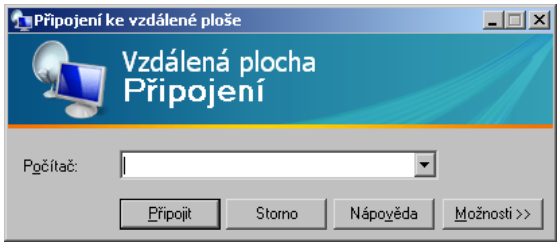
4.2.1 Aplikace Vzdálená plocha

Aplikace vzdálená plocha je standardně dostupná pro MS Windows Server 2003, Windows Vista a Windows XP. Microsoft vydal speciální aplikaci, která umožňuje připojení k systému ze starších verzí Windows. Jedná se o aplikaci `msrdpcli.exe`, která umožňuje připojení ze systémů Windows 95, Windows 98 and 98 Second Edition, Windows Me, Windows NT 4.0 a Windows 2000.

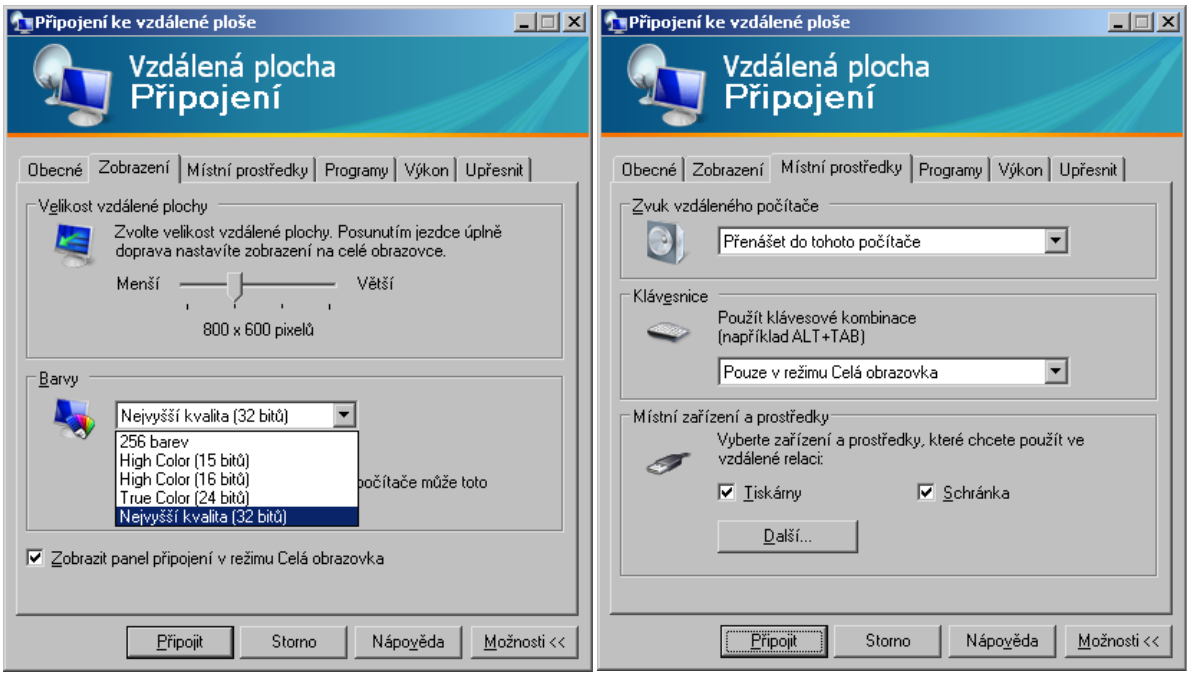
Na obr. 4.1 je zobrazeno přihlašovací okno. Zde se zadá IP adresa nebo jméno stanice (hostname). Pro nastavení parametrů připojení je možno zvolit „Možnosti“. Na obr. 4.2a je zobrazeno nastavení rozlišení obrazu a kvality barev (hloubka barev). Na obr. 4.2b jsou zachyceny možnosti voleb přenosu zvuku (zda přenášet, či nikoliv), sdílení tiskárny a schránky.

Na obr. 4.3a jsou zobrazeny další možnosti sdílení – přesměrování periferií (sériové porty, disky, zařízení Plug & Play). Podle zvolené rychlosti připojení se nastaví další parametry, které ovlivňují zatížení linky. Uživatel si tyto volby může podle svého uvážení zvolit sám, viz obr. 4.3b.

Pro připojení ke vzdálené ploše je potřeba znát IP adresu nebo jméno (hostname) počítače. Pro úspěšné přihlášení musí být na vzdáleném počítači zřízen účet, ke kterému se vzdáleně přihlásíme. Pokud se přihlašujeme ke vzdálenému počítači (kde běží Windows XP), na kterém je již přihlášen nějaký uživatel, zobrazí se varující hlášení, které informuje o tom, že uživatel bude odhlášen (systémem). Pokud stiskneme „ano“, přihlásíme se do systému (pokud máme právo) a zobrazí se nám plocha počítače.). V případě přihlašování pomocí vzdálené plochy k Windows Server 2003, je umožněno přihlášení současně dvou uživatelů k systému (jeden fyzicky a dva vzdáleně). Fyzicky i vzdáleně přihlášení uživatelé se přitom nijak neovlivňují.

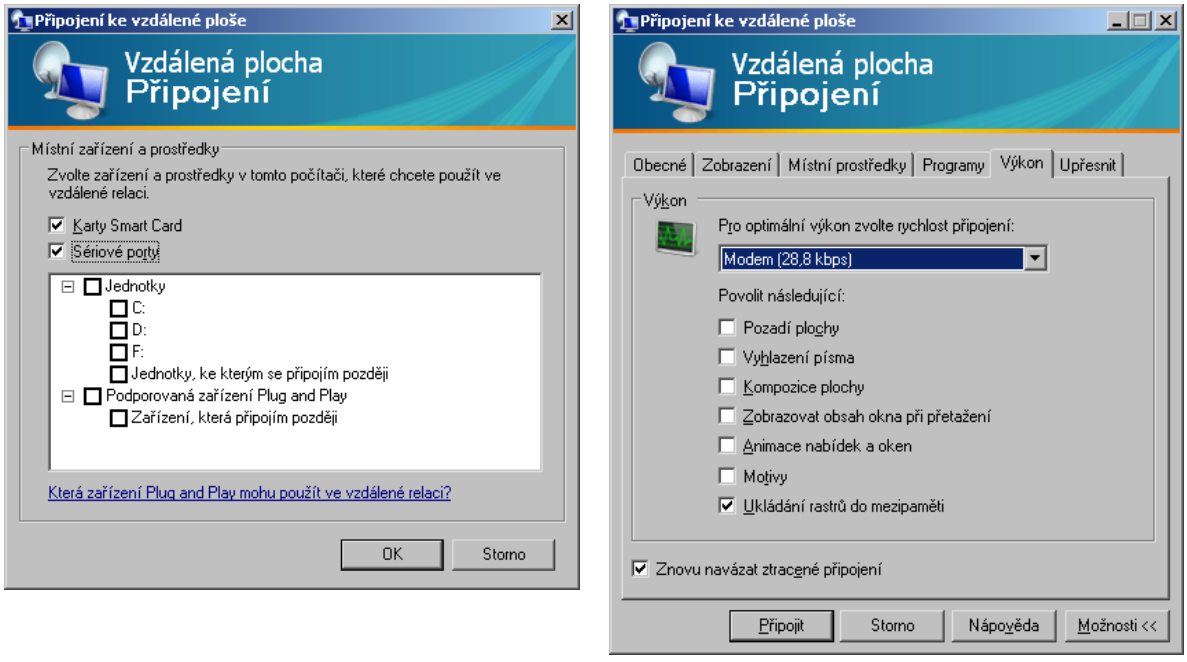


Obr. 4.1: Přihlašovací okno ke vzdálené ploše



a) b)

Obr. 4.2: Nastavení relace vzdálené plochy: a) nastavení obrazu, b) nastavení zvuku, sdílení tiskárny a schránky



a) b)

Obr. 4.3: Nastavení relace vzdálené plochy: a) přesměrování prostředků, b) nastavení zatížení linky

Kvalita a velikost přijímaného obrazu je ovlivněna předchozím nastavením. Na obr. 4.4 je zobrazeno připojení z klientského OS Windows XP do vzdáleného OS Windows XP.

V tab. 4.1 jsou zobrazeny zprůměrované hodnoty přenosů dat čtyř měření. Při testování se nepoužívala myš. Z výsledků je vidět praktický vliv použití cache (bufferu) snímků. Při testování s použitím cache snímků se snížil datový tok směrem k uživateli přibližně o 30 %.

Tab. 4.1: Porovnání velikosti přenesených dat aplikace Vzdálená plocha

Rozlišení	Hloubka barev	Přijato [KB]	Odesláno [KB]	Nastavení
1024 × 768	8	1 882	32	Modem 28,8 kbps, bez cache
1024 × 768	8	1 405	21	Modem 28,8 kbps, s cache
1024 × 768	32	2 672	38	Síť LAN, bez cache
1024 × 768	32	1 910	28	Síť LAN, s cache

Vzdálená plocha je velmi užitečná a plně funkční aplikace pro vzdálenou správu. Velkou výhodou je, oproti konkurenčním aplikacím, že po přihlášení ke vzdálenému systému, bude na vzdálené straně přihlášený uživatel odhlášen, tím je znemožněno zasahování do relace (myší, klávesnicí), či „špehování“. Rovněž možnost zvolení velikosti plochy (rozlišení) při přihlašování je velice užitečnou funkcí, kterou nenabízí některé alternativní programy. Jistou nevýhodou je, že Microsoft nepodporuje možnost přihlášení se do Windows z jiného operačního systému (např. z Linuxu). Ovšem existuje software třetích stran a open source projektů, např. *rdesktop*, *xrdp*.

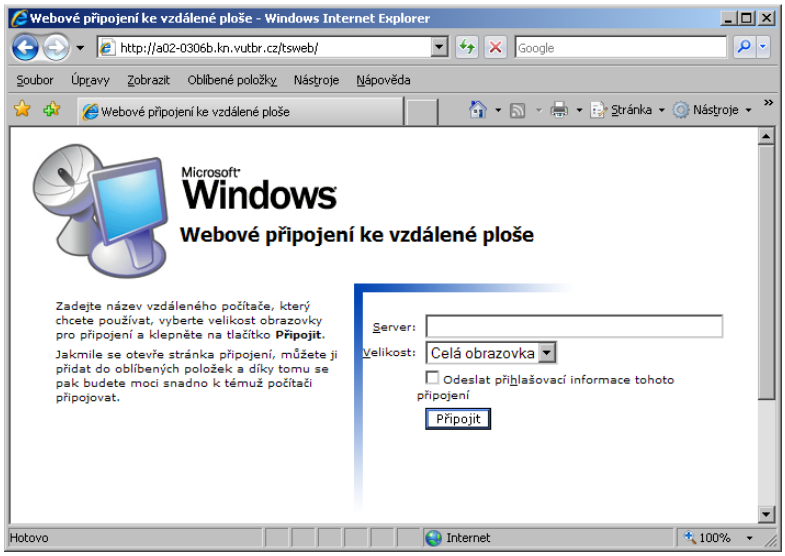
Obr. 4.4: Vzdálená správa pomocí vzdálené plochy

4.2.2 Vzdálená plocha přes webové rozhraní

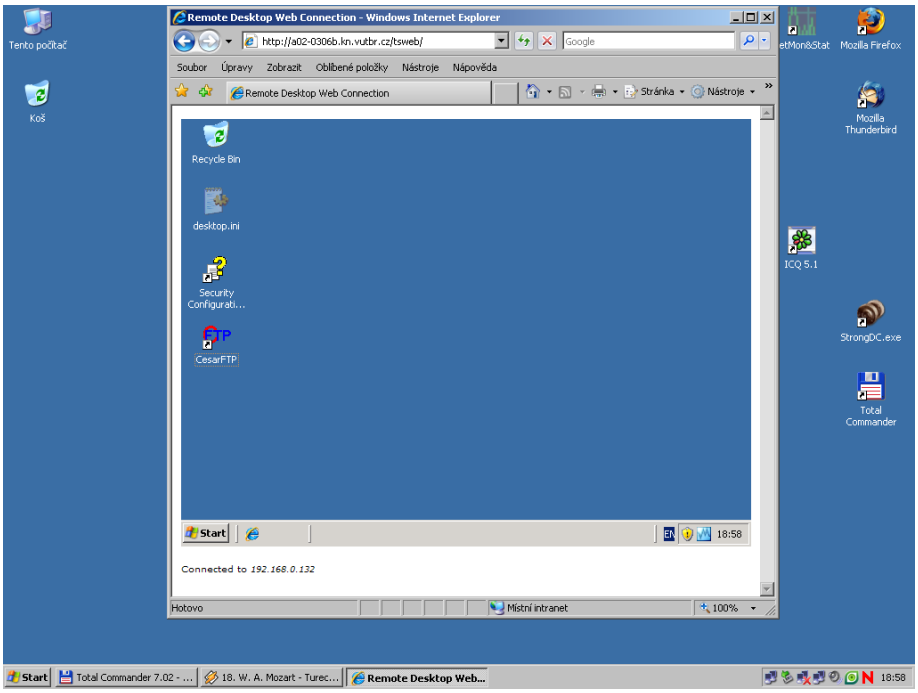
Další variantou, jak se vzdáleně přihlásit k systému, je pomocí webového rozhraní. K tomu je zapotřebí, aby na serveru byla nainstalovaná funkce „**Webové připojení ke vzdálené ploše**“. Postup instalace je dobře popsán např. na stránkách Microsoftu. Pro přihlášení je nutné použít prohlížeč Internet Explorer. Použití jiných prohlížečů není možné, stránka se sice zobrazí, ale nelze se připojit.

Pro přihlášení na vzdálenou plochu, se v Internet Exploreru zadá adresa serveru, která má tvar **http://hostname/tsweb/**. Na obr. 4.5 je zobrazené přihlašovací rozhraní. Uživatel musí zadat adresu serveru, na který se chce připojit a má možnost si vybrat rozlišení obrazovky.

Přihlášení pomocí vzdálené plochy do OS Windows Server 2003 z OS Windows XP je na obr. 4.6. Také u webové verze vzdálené plochy platí, že se lze připojit pouze z Windows do Windows.



Obr. 4.5: Webové připojení ke vzdálené ploše



Obr. 4.6: Správa systému pomocí vzdálené plochy přes webové připojení

4.3 Virtual Network Computing

VNC - Virtual Network Computing je podobně, jako Vzdálená plocha ve Windows, dalším systémem pro přenos vzdálené plochy. Jedná se také o systém typu klient – server. Klient posílá serveru údaje o polohovacím zařízení (myši) a klávesnici. Server klientovi posílá obraz. Základem VNC je protokol RFB (Remote Frame Buffer). Za vznikem VNC a RFB stojí společnost AT&T. Současná verze protokolu RFB je 3.8. Popis specifikace protokolu je volně k dispozici na internetu, viz lit. [11]. Hlavní výhodou VNC je, že je platformě nezávislý. Je napsán v mnoha jazycích, především v C++ a v Javě. Tím je možné ovládat např. vzdálený systém MS Windows z lokálního systému Linux. Obnova (přenos) obrazu je řízena klientem. Klient zasílá požadavky serveru a ten mu pošle odpověď (pošle změnu v obraze). Toto schéma umožňuje řízení rychlosti – adaptaci na různé přenosové rychlosti linky.

VNC standardně používá TCP port 5900. VNC klienty napsané v Javě porty 5800. Při zahájení komunikace se klient se serverem dohodne na formátu obrazu (použitá hloubka barev) a typu komprese. Při přenosu obrazu se posílají pouze výřezy obrazu, tj. čtverce nebo obdélníky. Formát je takový, že se pošle hlavička obsahující souřadnice výřezu a za hlavičkou následuje zkomprimovaný výřez obrazu.

Pro snížení toku dat je k dispozici několik metod kódování (komprese):

- **Raw**. Jedná se o nejjednodušší typ kódování. Obraz se přenáší jako pole pixelů – pole bajtů RGB. Nejedná se tedy o žádnou kompresi a při této metodě bude linka značně vytížená (v závislosti na zvolené hloubce barev). Výhodou této metody je nenáročnost na zpracování (není potřeba komprimovat či dekomprimovat), tedy nízké nároky na hardware.
- **CopyRect**. Tuto metodu je vhodné použít v případě, kdy se přesouvá okno nebo nějaká část obrazu. Jedinou informací, která se přenáší, jsou pouze souřadnice.
- **RRE**. Metoda RRE (Rise and Run length Encoding) je obdobou komprese RLE (Run length encoding). Jedná se o dvojrozměrnou verzi - nahrazuje sekvence stejných pixelů jedním pixelem a dál nese informaci o počtu opakování.
- **HexTile**. Celá plocha obrazu se rozloží na čtverce o rozměru 16×16 pixelů. Tyto malé plochy se posílají v předem známém pořadí. Pro kompresi (kódování) se používá RAW nebo RRE.
- **Zlib**. Pro kompresi se používá knihovna zlib (viz http://www.gzip.org/zlib/). Touto bezeztrátovou metodou komprese se značně sníží datový tok, ovšem zvýší se zatížení vlivem náročnosti metody. Náročností jsou postihnuty obě komunikující strany (komprimující strana – dekomprimující strana).
- **Tight**. Používá se, jednak funkce z knihovny zlib a také ztrátová technika komprese – algoritmus JPG. Tato metoda sníží nejvíce datový tok za cenu největší náročnosti.

Existuje mnoho klientů, jak z čistě komerční sféry, tak i freeware a open source. Nejznámější klienti jsou např. RealVNC, TightVNC, UltraVNC. Každý z klientů rozšiřuje funkcionalitu základního návrhu VNC. Výše jmenované programy jsou na velice dobré úrovni a jsou navzájem kompatibilní.

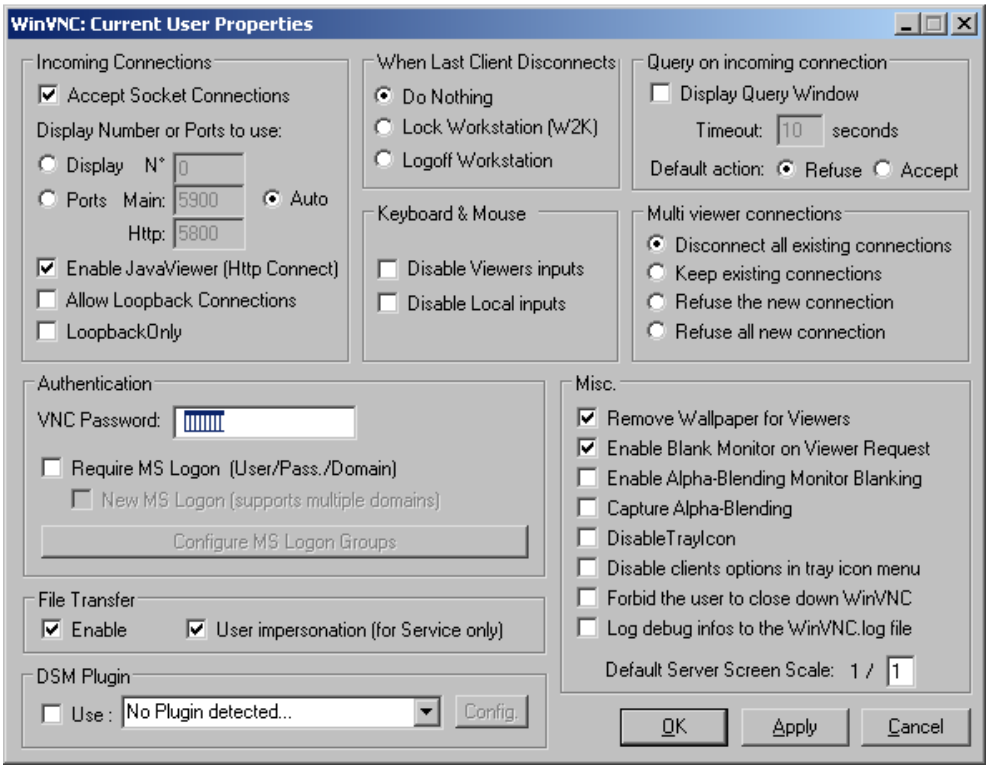
Pro popis a testování aplikace založené na VNC jsem si vybral aplikaci UltraVNC – jedná se o open source projekt.

4.3.1 UltraVNC

Aplikaci UltraVNC lze stáhnout z domovské stránky http://www.uvnc.com/. Aplikaci je možné nakonfigurovat i jako systémovou službu. Na obr. 4.7 je zobrazeno okno serverové části aplikace.

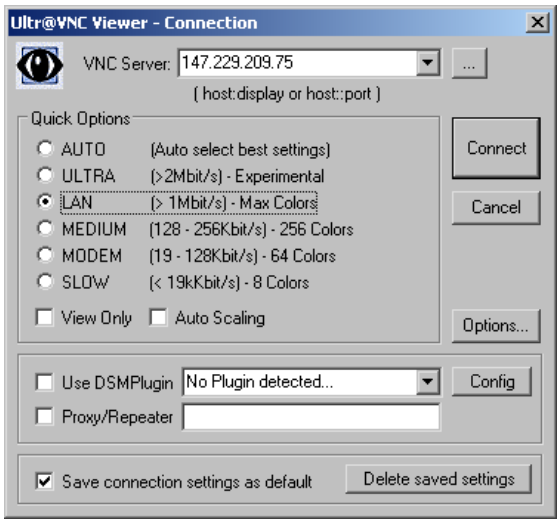
V serverové části aplikace je možné nastavit následující možnosti:

- aktivace síťové komunikace
- číslo portu, na kterém naslouchá server a číslo portu, na kterém naslouchá http server (možnost se přihlásit přes internetový prohlížeč – stáhne se Java applet)
- aktivace/deaktivace http serveru – pro přihlášení zadat adresu http://*hostname*:*port*/
- nastavení hesla
- nastavení smyčky – Loopback (pro testovací účely)
- přihlášení jako ve Windows (zadání uživatelského jména a hesla)
- možnost přenosu souborů
- DSM (Data Stream Modification) plugin – plugin ovlivňující vysílaná data, např. šifrování
- událost po odpojení: nedělat nic, zamknout stanici, odhlásit se ze systému (OS)
- více připojení (v režimu sledování – bez zásahu myší a klávesnice)
- logování do souboru
- odstranění obrázku na pozadí plochy
- zvětšení a zmenšení obrazu – scale



Obr. 4.7: Nastavení UltraVNC – serverová část

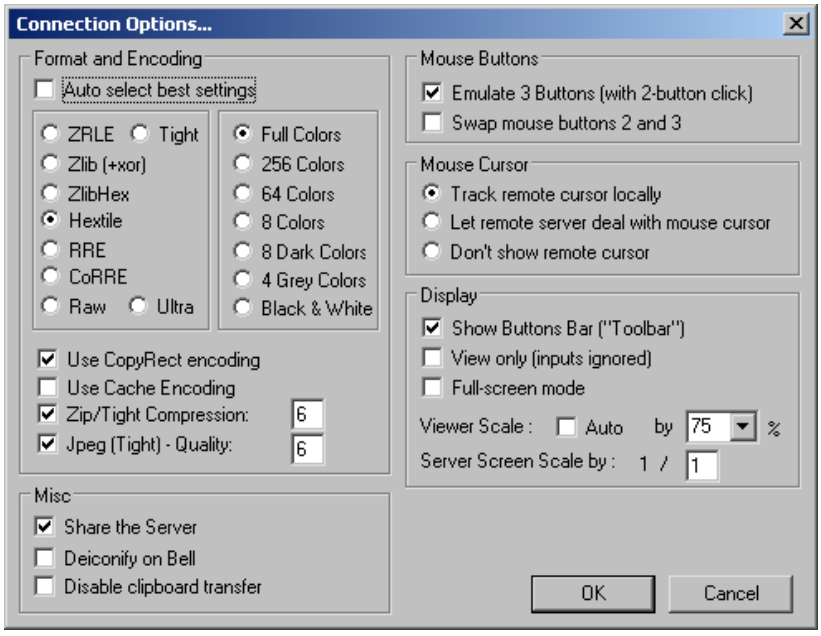
Na obr. 4.8 je zobrazeno uživatelské rozhraní klientské části aplikace. Zde se zadává adresa vzdáleného počítače, typ připojení – podle tohoto výběru se automaticky zvolí typ komprese, DMS plugin a případně proxy/repeater adresa. Z obr. 4.8 je patrná skutečnost, že typ a kvalitu komprese nastavuje klient a server se podle toho podřídí.



Obr. 4.8: Nastavení UltraVNC – klientská část

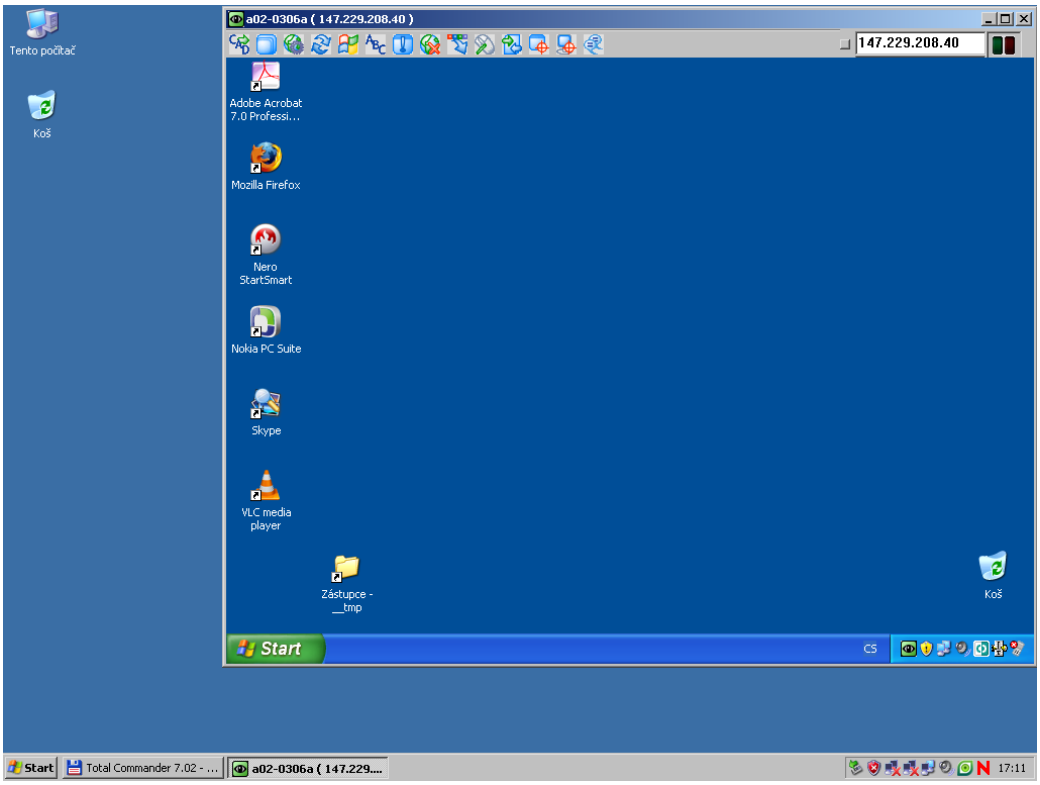
Kliknutím na tlačítko *Options* se zobrazí detailní nastavení, viz obr. 4.9. Zde jsou na výběr tyto možnosti:

- typ kompresní metody
- počet barev v obraze
- úroveň (kvalita) ZIP/JPG komprese
- sdílení schránky
- emulace třítlačítkové myši
- zobrazení vzdáleného kurzoru (sledování změny kurzoru, např. při roztažení okna, psaní textu)
- zobrazení tlačítkové nabídky
- pouze sledování obrazu (události klávesnice a myši se nebudou přenášet)
- nastavení měřítka (zvětšení nebo zmenšení obrazu – scale)



Obr. 4.9: Podrobnější nastavení v klientské části aplikace UltraVNC

Na obr. 4.10 je zobrazeno připojení pomocí UltraVNC klienta ke vzdálenému systému.



Obr. 4.10: Vzdálená správa pomocí UltraVNC

V tab. 4.2 jsou zobrazeny zprůměrované hodnoty přenosů dat čtyř měření. Při testování se nepoužívala myš. Z výsledků je vidět praktický vliv použití cache (bufferu) snímků. Při testování s použitím cache snímků se snížil datový tok směrem k uživateli přibližně o 20 % - 25 %.

Velikost datového toku, při hloubce barev 8 bitů, se přibližně shoduje s výsledem u Vzdálené plochy. Při hloubce barev 32 bitů je datový tok několikanásobně větší.

Tab. 4.2: Porovnání velikosti přenesených dat aplikace UltraVNC

Rozlišení	Hloubka barev	Přijato [KB]	Odesláno [KB]	Nastavení
1024 × 768	8	1 550	40	slow, bez cache
1024 × 768	8	1 248	33	slow, s cache
1024 × 768	32	11 018	256	slow, full, bez cache
1024 × 768	32	8 253	186	slow, full, s cache
1024 × 768	32	11 008	254	medium, full, bez cache, jpg
1024 × 768	32	8 232	183	medium, full, s cache, jpg

Aplikace UltraVNC v porovnání se Vzdálenou plochou nabízí daleko větší možnost konfigurace. Umožňuje nastavení hloubky barev až na 1 bit (černá, bílá), nebo šeď, čímž se datový tok značně minimalizuje. Jak již bylo řečeno, je možné se připojit k VNC serveru pomocí Java aplikace. Stačí zadat v internetovém prohlížeči správnou adresu a port a povolit stažení Java appletu. Tím je získána platformová nezávislost.

Měřením datového toku jsem dokázal, že obě aplikace používají algoritmy pro kompresi obrazu založené jak na pravděpodobnostním modelu, tak i na časovém. To znamená, že při pohybu malého okna se nepřenáší celý obraz, ale pouze vzniklá změna.

5 REALIZACE VLASTNÍHO ŘEŠENÍ

Hlavním cílem mé práce bylo navrhnout a realizovat systém umožňující vzdálenou kontrolu (sledování) a obsluhu (ovládání) pracovních stanic. Za použití kompresních metod snížit objem přenášených dat a tyto data zabezpečit pomocí kryptografických metod - algoritmů.

Pro realizaci vlastního návrhu jsem se rozhodl použít programovací jazyk Java. Díky této volbě, je zajištěna platformová nezávislost a přenositelnost kódu všude tam, kde je možné provozovat JVM SE (Java Virtual Machine Standard Edition). Veškerý software jsem vyvíjel v programovacím prostředí NetBeans v. 6.0.1, které je zdarma dostupné na adrese http://www.netbeans.org. Dále při vývoji a testování byl použit JDK v. 6 (update 6) a JRE v. 6 (update 6) dostupné na adrese http://java.sun.com/javase/downloads/index.jsp. Celý systém je tedy napsán čistě v jazyce Java a z programů není explicitně použito žádných funkcí – volání specifických pro konkrétní operační systém.

Všechny programy jsou zkompilovány (soubory typu `.class`) a převedeny do balíčku (`.jar`). Spuštění programů se provede např. v konzoli pomocí příkazu `java –jar jmeno_programu.jar` nebo pouhým poklepáním myší na balíček (`.jar`).

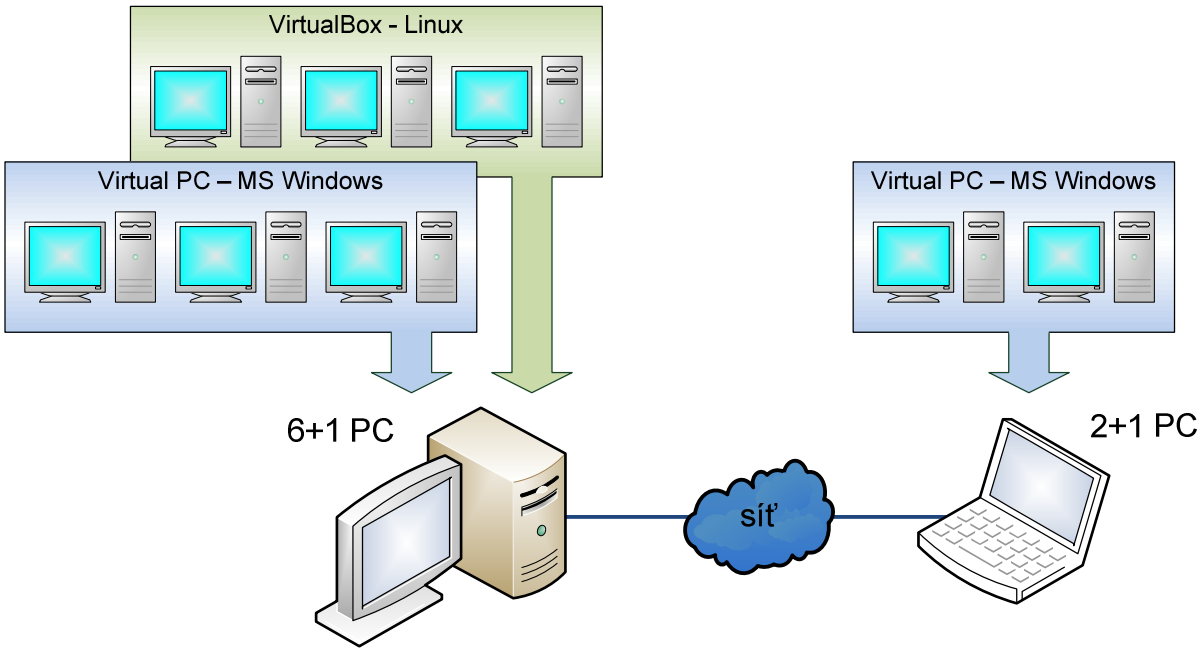
5.1 Použité prostředky

Vzhledem k rozsáhlosti stanovených úkolů (povinných i nad rámec zadání) a časovému omezení, bylo zapotřebí navrhnout takové vývojové postupy (plán a systematika práce), aby byla práce efektivní, rychlá a především časově úsporná. Řešení mé práce se skládá z několika objektů (viz následující kapitoly), které spolu vzájemně kooperují prostřednictvím počítačové sítě. Při vývoji aplikací tohoto charakteru je často zapotřebí, pro kontrolu nové funkce, nakonfigurovat a spustit všechny zainteresované objekty. Přitom je nutné zajistit, aby každý objekt měl své vlastní hardwarové (síťová karta, mac, ip adresa, volný diskový prostor) a softwarové zdroje (funkce operačního systému). Jinak řečeno, pro vývoj těchto aplikací je zapotřebí mnoho počítačů zapojených do počítačové sítě.

Postup práce, při použití samostatných fyzických počítačů (počítačová laboratoř), by byl značně časově náročný (potřeba modifikovanou aplikaci nahrát, nakonfigurovat, spustit apod. na každý počítač zvlášť), ale i nevhodný z vyplývající potřeby použít (obsadit) mnoho počítačů ve školní laboratoři. Z těchto hlavních důvodů jsem se rozhodl využít takových dostupných prostředků, které tato omezení limitují. Jedná se o virtualizační nástroje (programy), které umožňují provoz mnoha virtuálních počítačů (časté je označení „strojů“) na jediném fyzickém počítači. Tyto použité nástroje přidávají virtualizační vrstvu, která je umístěna nad konkrétním operačním systémem. Pro provoz těchto virtuálních strojů je potřeba mít na fyzickém počítači dostatek zdrojů (diskový prostor, operační paměť, procesor).

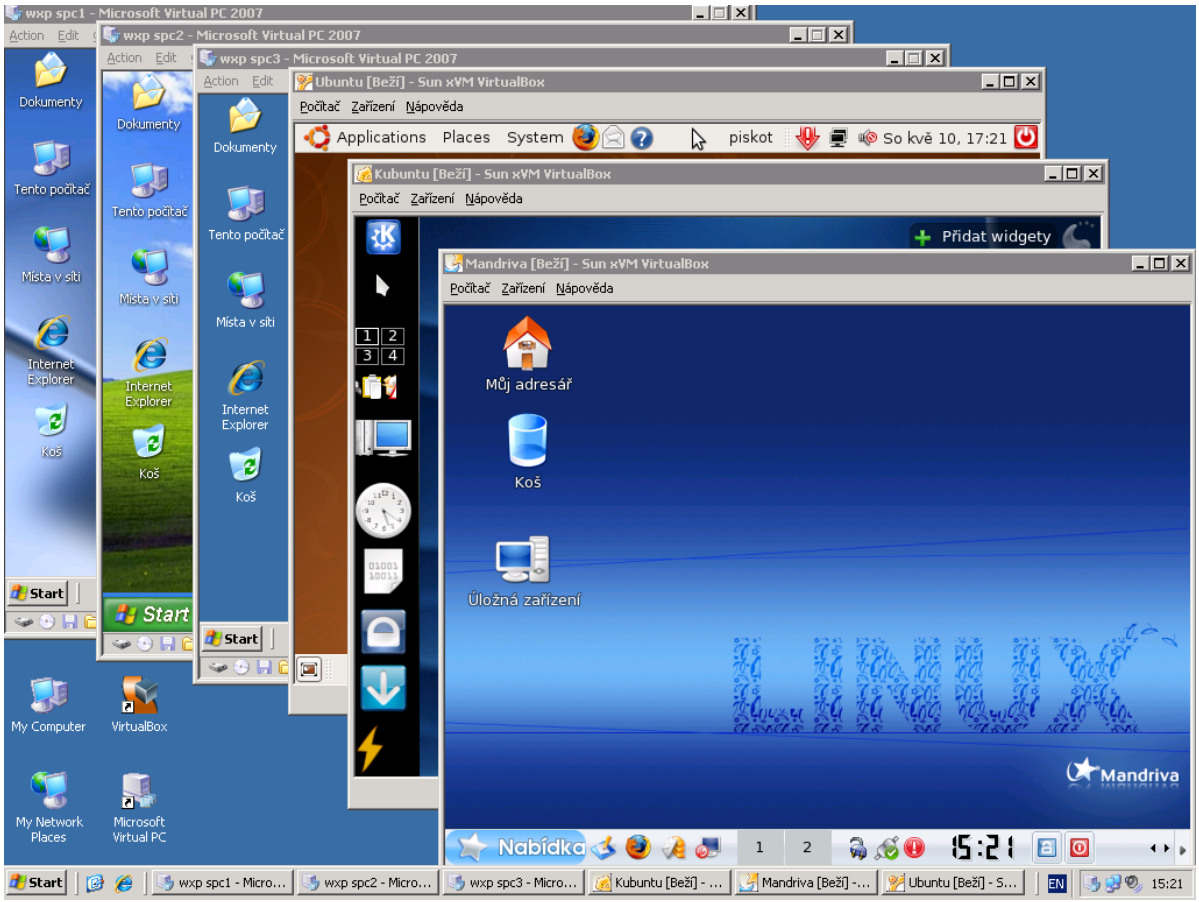
V průběhu vývoje jsem používal dva virtualizační nástroje, které jsou zdarma dostupné z internetu. Jedná se o nástroje Microsoft Virtual PC 2007 (dostupný na adrese http://www.microsoft.com/windows/downloads/virtualpc/default.mspx), který byl použit pro virtualizaci strojů s operačním systémem MS Windows XP SP3 a Sun xVM VirtualBox v. 1.6.0 (dostupný na adrese http://www.virtualbox.org), který byl použit pro virtualizaci strojů s operačním systémem Linux (distribuce Kubuntu v. 8.04, Ubuntu v. 8.04 a Mandriva v. 2008.1).

Na obr. 5.1 je zobrazeno schéma pracoviště z pohledu virtualizace – „rozmístění“ virtuálních strojů. Na výkonném desktopu, kde je provozován MS Windows Server 2003 R2, je k dispozici 6 virtuální strojů (3x MS Windows XP SP3 a 3 již zmíněné distribuce Linuxu). Na přenosném počítači, kde je provozován MS Windows XP SP3 a vývojové prostředí NetBeans, jsou k dispozici 2 virtuální stroje (oba MS Windows XP SP3). Celkem je tak k dispozici 10 počítačů, které jsou schopny, každý samostatně, provozovat požadované aplikace.



Obr. 5.1: Schéma pracoviště z pohledu virtualizace

Takto navržené a realizované pracoviště mi splnilo všechny výše jmenované požadavky a velký komfort při práci. Na obr. 5.2 je praktická ukázka dostupnosti všech šesti virtuálních strojů z jediného místa. Každé okno má v záhlaví název virtualizovaného systému. Pouhým kliknutím na požadovaný stroj je možno ihned pracovat.



Obr. 5.2: Praktická ukázka virtualizovaných strojů

5.2 Postup vývoje

Vzhledem k tomu, že vyvíjený software je postupem času složitější, rozsáhlejší a náchylnější na chyby, je také čím dál více časově náročnější přidávat nové funkce. Jinak řečeno, postupným přidáváním kódu do již rozsáhlého programu, roste potřebný čas na toto provedení, ale i čas potřebný k hledání a odstranění případných chyb. Programátor musí zdlouhavě vyhledávat chybu, která se může nacházet např.:

- v jádře rozpracované (právě vyvíjené) funkce
- v místě interakce s okolím (předávání hodnot dalším funkcím)
- v místě, které nesouvisí nebo nepřímo souvisí s právě vyvíjenou funkcí (např. problém více vláken a sdílených dat)

Při hledání chyb se může snadno stát, že programátor při běžném postupu vývoje špatně identifikuje příčinu chyby a mylně opraví část kódu, která může způsobit další chybu (či chyby) nebo chybu pouze oddálit.

Při praktické realizaci jsem postupoval po jednotlivých krocích – tvořil malé a funkční celky - tzv. moduly. Každý samostatný modul jsem po ověření funkčnosti integroval do cílového programu. Tím jsem minimalizoval riziko zdlouhavého hledání případných chyb. Některé moduly měly takový charakter (obsah), že bylo nutné je integrovat do všech cílových programů, viz kap. 5.3. V tab. 5.1 je přehled modulů, ze kterých jsou cílové programy složeny. Některé moduly nebyly do cílového programu „plně“ integrovány, ale jejich vývoj (výsledek) byl zásadní. Jedná se především o testování rychlosti různých postupů, např. přístup k datům (extrakce RGB dat z obrazu, generování atd.).

Oba použité virtualizační nástroje umožňují uložit aktuální stav stroje. To znamená, že uloží aktuální obsah operační paměti, zásobníků atd. Díky této funkci bylo možno snadněji najít a opravit chybu, která měla zdánlivě náhodný charakter. Tento druh chyby, který vznikal nepravidelně (např. důsledkem spolupráce více vláken a zpracování sdílených dat) bylo velmi složité záměrně vyvolat (z důvodu zkoumání). Ovšem díky možnosti ukládání aktuálního stavu virtuálního stroje bylo možno zachycenou (nepravidelnou) chybu uchovat a kdykoliv později zkoumat důvody vzniku a její následky.

Tab. 5.1: Seznam vytvořených modulů

Modul	Popis funkce
Blokové dešifrování	Dešifrování citlivých dat algoritmem RSA, AES.
Blokové šifrování	Šifrování citlivých dat algoritmem RSA, AES.
Časovač	Odpočítávání intervalu, periodická aktualizace (refresh).
Datagramový přijímač	Příjem dat pomocí protokolu UDP.
Datagramový vysílač	Odesílání dat pomocí protokolu UDP.
Datum a čas	Zobrazení a aktualizace aktuálního data a času.
Dekomprese	Dekomprese bloku dat do původní podoby.
Generování klíčů	Generování klíčů pro proudové (RC4) a blokové (RSA, AES) šifrování.
Komprese	Komprese původních dat.
Konfigurace	Čtení konfiguračních souborů.
Multicastový přijímač	Registrace do skupiny. Příjem dat.
Multicastový vysílač	Vysílání multicastových dat.
Nastavení	Nastavení chování aplikace, přizpůsobení, volba režimů přenosu atd.
Oprávnění	Vyhodnocení práv uživatele a reakce na specifické typy práv.
Oznamování	Oznámení chybových stavů, zobrazení dialogů.
Panel	Ukládání a zobrazování obrazu do panelu.
Plocha	Zobrazování přijatého obrazu.
Pohyb panelu	Listování a seřazování panelu na základě různých událostí.
Vyskakovací nabídka	Nabídka možností, povolení a odepření přístupu.
Přenos textových zpráv	Přenos zpráv uživatelům a příjem odpovědi.
Proudové šifrování	RC4 metoda pro šifrování obrazu.
Přenos obrazu	Rozklad, distribuce a složení obrazu.
Přenos ovládání	Přenos souřadnic a stisknutých tlačítek myši a klávesnice.
Převod čísla do pole	Převod čísla (datového typu) do pole bajtů.
Převod pole na číslo	Převod pole bajtů na číslo (datový typ).
Registrace uživatele	Zpracování nového uživatele - ověření přijatých identifikačních údajů.
Řetězce	Transformace znaků do pole bajtů, rozdělování a nahrazování znaků.
Snímání a provádění pohybu a stisku tlačítek	Rozlišování typu zdroje události (myš, klávesnice), mapování do souřadnic, provádění událostí.
Snímání obrazu	Získávání obrazu, ukládání snímků, extrakce pixelů do pole bajtů.
Souborový manažer - lokální	Procházení, kopírování, mazání, vytváření souborů / adresářů na lokálním disku.
Souborový manažer - vzdálený	Procházení, kopírování, mazání, vytváření souborů / adresářů na vzdáleném disku.
Soubory a adresáře	Práce se soubory a adresáři. Vytváření, mazání, kopírování a čtení vlastností. Výpis cest, rekurentní procházení adresáři.
Test a zpracování čísel	Test lichých a sudých čísel, zaokrouhlování, transformace.
Test konfigurace	Test správné syntaxe a existence povinných parametrů.
Test rychlosti	Test rychlosti zpracování dat různými metodami.
Výpočet haše	Výpočet haše SHA-256 ze vstupního řetězce.
Zámek	Uzamykání a odemykání aplikace. Zabezpečení pomocí hesla.
Změna rozměrů náhledu	Změna velikosti náhledu, přenos příkazů.
Změna rozměrů plochy	Změna velikosti plochy, zvětšování a zmenšování rozměrů okna.

5.3 Navržený systém

S ohledem na praktické potřeby, zásady a doporučení, jsem navrhl dynamický systém, který obsahuje všechny nezbytné náležitosti a funkce pro potřeby hromadného sledování a ovládání vzdálených stanic.

Zde je uveden výčet nejdůležitějších vlastností, které byly do systému implementovány. Výčet shrnuje praktické potřeby a doporučení pro síťově orientované aplikace a potřeby a doporučení pro hromadné sledování a ovládání vzdálených stanic.

- bezchybný přenos dat – ochrana před nedoručením nebo doručením chybných zpráv
- bezpečný přenos dat – ochrana před neoprávněným přístupem k datům
- autentizace – možnost správné identifikace uživatele, ochrana před útokem typu „muž uprostřed“
- komprese dat – eliminace přenosu velkých objemů dat
- jednoduché nasazení do praxe – jednoduchá a rychlá instalace
- jednoduchá správa (z hlediska správy sítě) – možnost rychlé a snadné modifikace a omezení konkrétních služeb
- úspornost řešení – efektivní návrh systému a použití pouze nezbytného množství prvků (PC) tam, kde je to možné a vhodné (v případě nutnosti zajistit bezpečnost a stabilitu je naopak vhodná redundance prvků)
- možnost konfigurace (z hlediska správy aplikace) – nastavení parametrů chování aplikace
- jednoduchost konfigurace – jednoduchá syntaxe příkazů, bohaté komentáře v konfiguračních souborech, příklady nastavení
- dynamické reakce na různé stavy – oznámení a uvedení důvodů vzniklých stavů, logování, ošetření následků
- snadný přístup k datům – pohodlná práce s daty
- snadný přístup ke vzdálenému zařízení – možnost užití prostředků, kterými disponuje vzdálené zařízení
- jednoduché ovládání – rychlé seznámení, intuitivní prostředí
- přehlednost – vhodné grafické uspořádání, nerušivý design
- možnost přizpůsobení – nastavení vyhovující konkrétnímu uživateli

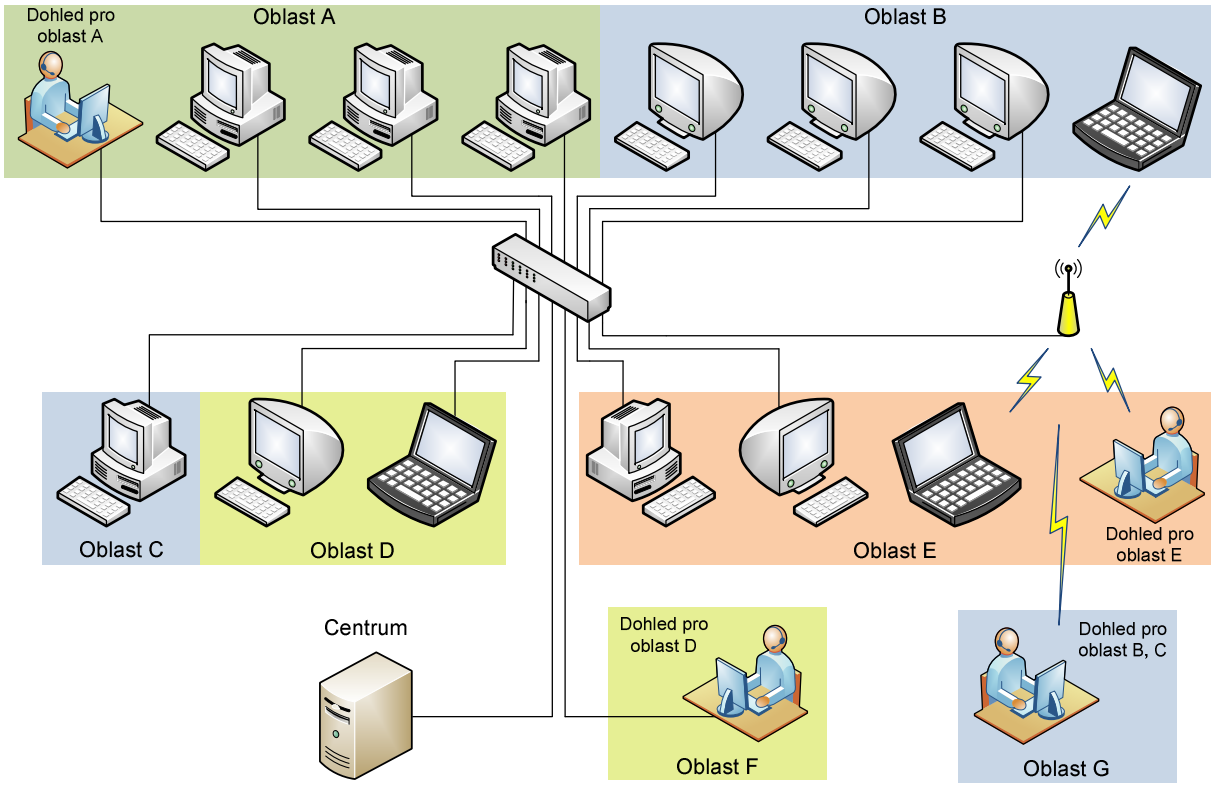
Navržený dynamický systém se skládá ze tří objektů:

- přístupový (ověřovací) server - **centrum**
- dohlížecí stanice - **dohled**
- pracovní stanice - **uživatel**

Každý z objektů v systému provádí a poskytuje funkce, které jsou specifické pouze pro konkrétní objekt. Některé funkce však musí být obsaženy ve všech objektech. Funkce každého objektu budou popsány v následujících kapitolách.

Obr. 5.3 ilustruje strukturu a možnosti vytvořeného systému pro dohled a ovládání. Je zde naznačena nezávislost k typu datového připojení k síti. Tuto problematiku řeší nejnižší vrstvy síťového modelu. Pro ilustraci bylo použito kabelové a bezdrátové spojení. Různý design PC reprezentuje různé hardwarové (architektura 32 bit PC, 64 bit PC, Power PC atd.) a softwarové (MS Windows, Linux atd.) konfigurace. To je možné díky platformové nezávislosti jazyku JAVA. Jsou zde barevně odlišeny oblasti, které jsou na ostatních oblastech (označeny jinou barvou) naprosto nezávislé. Každý barevně odlišený rámeček označuje jednu logickou oblast, která ovšem nemusí být fyzicky na jednom místě nebo v totožné síti. Tento případ platí pro logickou oblast (označenou světle modrým rámečkem), která je rozčleněna na 3 fyzické oblasti B, C a G. Na obrázku jsou zobrazeny celkem 4 logické oblasti, které jsou složeny ze sedmi fyzických oblastí označených A až G. Fyzická oblast může být např. část místnosti, celá místnost, patro, budova, ale i různé (části) sítě atd. Díky takovému konceptu může uživatel dohlížecí

stanice – **dohled** sledovat a ovládat pracovní stanice, které nejsou fyzicky spjaty s jedním místem – s jednou sítí. Rovněž samotná dohlížecí stanice se může nacházet v libovolné oblasti. Komunikace mezi objekty probíhá prostřednictvím protokolů UDP a TCP. Pomocí portů je rozlišeno, která data patří jakému procesu. Díky rozdělení komunikace do několika spojení (specifikována konkrétním portem) je možné, z pohledu správce sítě, cílenou komunikaci omezovat - blokovat (např. zakázat přenos souborů, ale povolit přenos obrazu apod.).



Obr. 5.3: Schéma systému pro dohled a ovládání

Následující kapitoly popisují objekty a jejich funkce, které se v systému vyskytují. Funkcionalita některých částí programů je značně rozsáhlá a z prostorových důvodů nebude uveden detailní popis.

5.3.1 Přístupový server

V celém systému je zapotřebí pouze jediného přístupového serveru označovaného jako centrum. V případě, že je provozováno více systémů pro dohled a ovládání (např. v jedné organizaci), je možné jednotlivé přístupové servery umístit pouze do jediného serveru a to dvěma způsoby:

a) Provozování samostatných instancí programu, přičemž každá instance používá svou vlastní databázi (informace o účtech). Pro každou instanci je třeba vyhradit jednu IP adresu.
b) Provozování jediné instance programu, která používá jednu společnou databázi.

Výhodou sjednocení přístupových serverů do jediného serveru je úspora použitých zařízení (počítačů). Nevýhodou je, že v případě výpadku serveru dojde k nedostupnosti služby pro všechny uživatele, kteří se chtějí přihlásit. Stávající, již přihlášení uživatelé, ochromením provozu centra nebudou nijak postiženi. Výhodou uplatnění strategie popisované v bodě a) oproti b) je to, že jeden uživatel může mít v každém systému jeden účet (různá politika práv

v každém systému). Pro připojení ke konkrétnímu serveru je třeba nastavit v konfiguračním souboru příslušnou IP adresu. O možnostech konfigurace pojednává kap. 5.3.4.

Tento objekt v systému má zásadní funkci a jeho absence (výpadek) by znemožnila přihlášení nových uživatelů. Na dohled, ovládání nebo přenos dat již dříve úspěšně přihlášených uživatelů však nemá žádný vliv.

Program, který provádí činnost přístupového serveru, je uložen do souboru `centrum.jar`. Program před zahájením své činnosti nejprve zkontroluje přítomnost a integritu tří souborů, které pro svou činnost vyžaduje. Jedná se o soubory:

- `address.dat` – databáze IP adres
- `config.conf` – konfigurační soubor
- `users.dat` – databáze uživatelů

Z důvodu jednoduché konfigurace jsou všechny soubory textové. Obsahují doprovodné komentáře, popis zásad a příklady nastavení. V následujícím textu jsou použity výrazy databáze, které v tomto případě představují jednoduché textové soubory. V případě nepřítomnosti alespoň jednoho souboru nebo nalezení chyby v prohledávaném souboru, se zobrazí podrobné varovné hlášení a program se ukončí. Kromě zmíněných varovných dialogů program neobsahuje žádné další grafické prvky - okna.

Funkce přístupového serveru

Funkce přístupového serveru - centra jsou následující:

- **Zpracování nového uživatele.** Každý uživatel, který chce do systému vstoupit (přihlásit se) se musí „představit“ centru. To znamená, že musí poslat centru identifikační údaje, kterými jsou jméno a heslo uživatele. Důležitou podmínkou je jedinečnost jména uživatele v rámci jednoho systému. Toto pravidlo je prosazováno u mnohých informačních systémů, fór, e-mailových služeb atd. Po připojení uživatele k centru se vytvoří nové vlákno, ve kterém je nový uživatel obsluhován. Centrum po obdržení identifikačních údajů ověří uživatele - zjistí, zda uživatel má povolený přístup a po provedení testů oznámí uživateli výsledek testů (kladné nebo záporné vyřízení požadavku). ***Rozhodování je ovlivněno konfigurací v databázi IP adres a v databázi uživatelů. Aby se přihlašující uživatel mohl přihlásit do systému, musí projít dvěma testy.*** Uživatel, který projde testy, bude uložen do databáze „online uživatelů“. Pokud se jedná o běžného uživatele (pracovní stanici) bude uživatel uložen právě do databáze běžných uživatelů (pracovních stanic), v opačném případě do databáze dohlížejících uživatelů (stanic). Databáze obsahuje odkaz na instanci uživatele (stanice), ve kterém je uloženo jeho jméno, oblast umístění, IP adresa a pokud se jedná o dohlížecí stanici je navíc uložen veřejný klíč, v případě, že se jedná o běžnou stanici, je uloženo tajné heslo pro šifrování/dešifrování algoritmem AES.
- **Oznamování dohledu.** Po úspěšném přihlášení dohlížecí stanice, ji bude zaslán seznam jmen (a další potřebné údaje) již přihlášených uživatelů (pracovních stanic), které náleží do stejné logické oblasti. Po tomto kroku budou všechny pracovní stanice, příslušející ke stejné logické oblasti jako dohled, informovány o skutečnosti, že se do systému přihlásil dohled. Po každém úspěšném přihlášení nového uživatele bude dohled informován.
- **Oznamování uživateli.** Poté, co do systému vstoupila dohlížecí stanice, budou všichni zainteresovaní uživatelé informování. Bude jim sdělena IP adresa dohledu, ke které se následně připojí a přihlásí se dohlížejícímu uživateli.
- **Mazání uživatele a dohledu.** V případě ztráty spojení se stanicí dojde ke smazání uživatele z příslušné databáze.

Všechna citlivá data jsou při přenosu šifrována. Je zde použito symetrických (AES) a asymetrických (RSA) kryptografických algoritmů. Přístupový server vlastní tajný (RSA) – 1024 bitů dlouhý klíč. Pomocí kterého dešifruje zašifrované heslo, které si zvolili ostatní objekty v systému. Přístupový server standardně naslouchá na TCP portu 54321.

Databáze IP adres

V databázi IP adres jsou obsaženy údaje, které jsou použity při rozhodování přihlášení pracovní nebo dohledové stanice. Zde se provádí rozhodování primárně na základě IP adresy. Údaj je složen z IP adresy, identifikátoru práva, které je vyžadováno a názvu místnosti (logické oblasti), ve které se daná stanice nachází. Každý z údajů se musí nacházet na novém řádku, jednotlivé parametry jsou odděleny dvojtečkou. Struktura zápisu spolu s příkladem je zobrazena na obr. 5.4.

```
ip_adresa:typ_požadovaného_oprávnění:logické_umístění

192.168.1.105:d:PA123
147.229.209.75:x:PA456
147.229.202.55:u:E306
```

Obr. 5.4: Formát zápisu údajů v databázi IP adres

Přístupový server při testování uživatele postupuje tak, že hledá v databázi IP adres shodu IP adresy aktuálně připojeného uživatele a IP adresy v databázi. V případě nalezení shody se dále vyhodnocuje oprávnění, viz tab. 5.2. Typ oprávnění určuje, kdo a zda se může z dané IP adresy přihlásit. Existují tři typy oprávnění, které se označují znakem:

- d – ze stanice se může přihlásit pouze uživatel s právy dohledu
- u – ze stanice se může přihlásit libovolný uživatel
- x – uživatel se nemůže z této IP adresy přihlásit

V případě, že uživatel má právo se z IP adresy přihlásit, bude proveden test jména a hesla uživatele za použití databáze uživatelů. Při použití databáze IP adres je třeba zapsat každou IP adresu, která bude užita v systému. To přináší nutnost adresy vybrat a shromáždit do této databáze, což nemusí být jednoduché. Dále tato technika omezení/povolení přístupu na základě IP adresy klienta nemusí být vhodná, např. v případě, kdy je počítačům v síti je IP adresa dynamicky přidělována.

Zjednodušení správy této databáze přináší parametr `ALL_IP`. Pokud se v databázi vyskytuje toto klíčové slovo, bude při rozhodování brán zřetel pouze na údaje, které za IP adresou obsahují oprávnění typu x. To znamená, že všechny stanice projdou testem databáze IP adres mimo zakázaných. Vlivem ignorování údajů budou ignorovány i informace o logickém umístění stanice. V tomto případě se budou všechny stanice v systému nacházet v jedné (virtuální) oblasti. Dohlížet a ovládat tuto oblast bude moci pouze jeden dohlížející uživatel, který musí být specifikován v této databázi. Protože obsah tohoto souboru se (částečně) podílí v procesu testování uživatele, je nutné tento soubor zabezpečit před neautorizovanou modifikací či čtením. To je možné např. pomocí přístupových práv, které zajišťuje operační systém.

Databáze uživatelů

Uživatel, v průběhu přihlašování, pošle centru přihlašovací údaje, kterými jsou jedinečné jméno a heslo. Heslo jedinečné být nemusí. V případě úspěšného projití testem IP adres bude proveden test shody jména a hesla (získaných od uživatele) se jménem a heslem v databázi

a provede se vyhodnocení práv uživatele a práv stanice. V této databázi je také uložen údaj, který identifikuje typ práv uživatele. Jedná se o stejně značená práva `d`, `u`, `x`. Na obr. 5.5 je zobrazena struktura zápisu spolu s příkladem. Použitá hesla slouží pouze pro ilustraci, v praxi je nutné používat silná hesla (dostatečně dlouhá a složitá).

```
jméno_uživatele:přístupové_heslo_uživatele:typ_oprávnění_uživatele

xkoste04:qwerty:d
pnovak:tajneheslo:u
admin:admin:x
```

Obr. 5.5: Formát zápisu údajů v databázi uživatelů

Při vyhodnocování práv může dojít k několika jejich kombinacím. Možné kombinace práv a jejich řešení je zobrazeno v tab. 5.2. Je zde ukázáno, že uživatel mající práva dohledu se může přihlásit do systému ze stanice, která má libovolné požadavky na práva kromě typu práva `x` – v tomto případě je zakázáno se ze stanice přihlásit.

Tab. 5.2: Vyhodnocení práv

		Uživatel		
		d	u	x
Stanice	d	✓	×	×
	u	✓	✓	×
	x	×	×	×

Legenda:
✓ - lze se přihlásit
× - nelze se přihlásit

V databázi uživatelů je možné, podobně jako v databázi IP adres, použít speciální parametr, který zjednoduší konfiguraci, ale i odstraní bezpečnostní prvky. Pokud se v souboru `users.dat` nachází řetězec `ALL_USER`, bude se moci přihlásit kterýkoliv uživatel za předpokladu, že se v žádné z databází nevyskytuje omezení v podobě práva `x`, které se vztahuje k uživateli.

Organizace kódu

Funkcionalita programu je rozvržena do 7 souborů (tříd). V následujících bodech jsou stručně popsány hlavní funkce v nich obsažené:

- `Komunikace.java` – Na základě vytvoření spojení vytváří instance klientů (v novém vlákně).
- `Konfigurace.java` – Nastavení parametrů na základě obsahu konfiguračního souboru `config.conf`.
- `Main.java` – Spuštění programu. Volá funkce pro zpracování konfigurace.
- `NovyKlient.java` – Obsluha nového uživatele. Příjem přihlašovacích údajů. Volání funkce pro ověření a uložení do „online databáze“.
- `OnlineUzivatel.java` – Ukládání a načítání uživatelů z „online databáze“. Přenos seznamu přihlášených uživatelů.
- `RegSifrovani.java` – Šifrování a dešifrování komunikace. Uchování tajných hesel.
- `ZpracovaniNovehoUzivatele.java` – Ověření identifikačních údajů. Prohledávání souborů `address.dat` a `users.dat`.

5.3.2 Pracovní stanice

Pracovní stanice je druhým jmenovaným objektem navrženého systému a je hlavním bodem zájmu. Cílem celého systému je zprostředkovat komunikaci mezi dohledem (tento objekt je popsán v kap. 5.3.3) a pracovní stanicí. Pracovní stanice je, po úspěšném navázání spojení s dohledem, připravena plnit požadované příkazy, které přijala od dohledu.

Program, který provádí požadavky, posílá obraz atd., je uložen do souboru `pracovni_stanice.jar`. Program obsahuje mimo dialogových oken (zobrazující chybové oznámení) další grafické prvky. Je nutné distribuovat spolu s programem balíček `AbsoluteLayout.jar`, který musí být umístěn v adresáři `lib`. Program před zahájením své činnosti nejprve zkontroluje přítomnost a integritu konfiguračního souboru. Pokud konfigurační soubor není nalezen nebo obsahuje chyby, program zobrazí podrobnou chybovou zprávu a ukončí se.

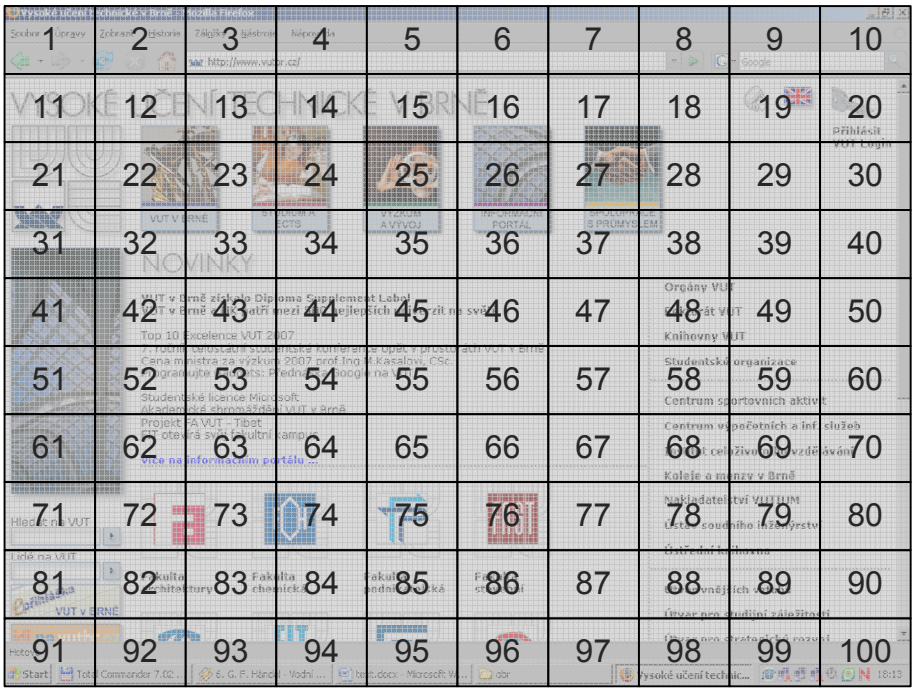
Ze stejných důvodů, které byly uvedeny v kap. 5.3.1, je i zde konfigurační soubor v textové podobě a obsahuje doprovodné komentáře, popis zásad a příklady nastavení.

Funkce pracovní stanice

Funkce pracovní stanice jsou následující:

- **Přihlášení do systému.** Pracovní stanice odešle přístupovému serveru nejprve zašifrované heslo. Dále pomocí tohoto hesla je zašifrováno a posláno jméno a heslo uživatele. Pokud jsou poslaná data platná, bude stanice (uživatel) přihlášen a uložen do databáze přihlášených pracovních stanic.
- **Přihlášení dohledu.** Po přihlášení dohlížejícího uživatele, který se nachází ve stejné logické oblasti jako uvažovaná pracovní stanice, bude této pracovní stanici sdělena IP adresa a veřejný klíč příslušející dohledu. Pracovní stanice, podobně jako při přihlašování k přístupovému serveru, zašifruje veřejným klíčem své heslo, které použije pro šifrování citlivých dat (jména a hesla pro RC4). Dohlížecí stanice má k dispozici odpovídající tajný klíč, pomocí kterého zjistí tajné heslo a pomocí tohoto hesla dešifruje přijatá data.
- **Generování hesel.** Citlivé údaje (jméno, heslo a klíč) je třeba bezpečně přenést. Pracovní stanice si sama zvolí náhodné heslo, které je **tajné**. Toto heslo je použito pro šifrování/dešifrování dat pomocí algoritmu AES. Stanice (pracovní i dohled) svoje vygenerované heslo zašifrují veřejným klíčem, který přísluší přístupovému serveru – ten vlastní odpovídající tajný klíč. Tím je zajištěna bezpečná přeprava klíče, pomocí kterého je prováděno šifrování/dešifrování. Pracovní stanice si generuje ještě jedno heslo, které bude použito v proudovém šifrování (algoritmus RC4) obrazu.
- **Vytváření a přenos náhledu obrazu.** Pracovní stanice periodicky odesílá zmenšené obrazy plochy, tzv. náhledy. Velikost náhledů a dobu periody sděluje dohlížecí stanice. Způsob přenosu a úspory je stejný jako v případě přenosu celých obrazů. Při přenosu se posílají pouze malé podoblasti obrazu, u kterých došlo ke změně oproti předchozímu snímku. Viz níže.
- **Vytváření a přenos obrazu.** Dochází k přenosu obrazu, který má původní rozměry. Není provedena žádná geometrická transformace (zmenšení) obrazu. Při přenosu se posílají pouze malé podoblasti obrazu, u kterých došlo ke změně oproti předchozímu snímku. Tato odlišná data jsou ještě před odesláním zkomprimována a zašifrována proudovou šifrou. V případě, že v dané podoblasti nenastala žádná změna, nebude zapotřebí nic posílat. Na obr. 5.6 je zobrazeno možné rozdělení plochy obrazu do menších podoblastí (rozdělení může být provedeno do čtverců nebo obdélníků). Pro kompresi všech obrazů je použita „javovská“ funkce *Deflater*.

- **Příjem a provádění ovládání.** Přenos informací, které obsahují informace o souřadnicích kurzoru a stisknutých, respektive uvolněných tlačítkách, je uskutečněn pomocí protokolu UDP.
- **Příjem, zobrazení a odesílání zpráv.** V případě, že se rozhodne dohlížející uživatel poslat zprávu (např. varování, napomenutí) uživateli, který využívá pracovní stanici, dojde po přijetí zprávy k jejímu zobrazení na obrazovce monitoru provinilé osoby. Zpráva vždy obsahuje datum, čas příjmu a znění zprávy.
- **Přenos dat.** Do aplikace je integrován souborový manažer. Pracovní stanice tak umožňuje sdílení všech disků, mechanik a paměťových médií. Je podporováno kopírování, mazání a vytváření souborů a adresářů.
- **Spánek.** V případě, že dohlížecí stanice je mimo provoz (způsobeno výpadkem ve spojení nebo uzamknutím dohlížecí aplikace), není potřeba žádné činnosti ze strany aplikace umístněné na pracovní stanici. Aplikace přejde automaticky (po ztrátě spojení s dohledem) nebo na základě nařízení od dohledu, do režimu spánku. To je stav, kdy aplikace nevykazuje žádnou činnost – nijak nezatěžuje procesor a jiné zdroje. Po příchodu „probouzejícího signálu“ se aplikace „probudí“ a je opět plně k dispozici.



Obr. 5.6: Rozdělení plochy na malé bloky

Organizace kódu

Funkcionalita programu je rozvržena do 10 souborů (tříd). V následujících bodech jsou stručně popsány hlavní funkce v nich obsažené:

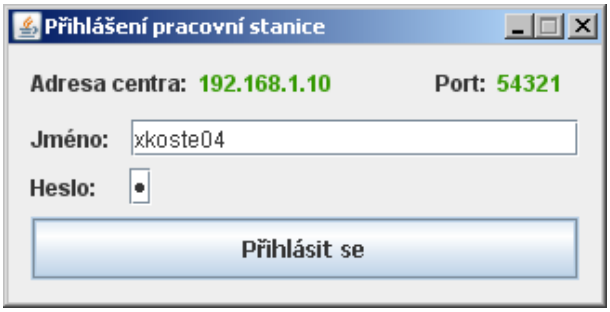
- `Komunikace.java` – Vytváření spojení a uskutečňování komunikace s centrem a dohledem na specifických portech. Příjem hromadného (multicastového) vysílání. Rozlišování (identifikace typu) příkazu a odpovědi.
- `Konfigurace.java` – Nastavení parametrů na základě obsahu konfiguračního souboru `config.conf`.
- `Main.java` – Spuštění programu. Volání funkce pro přihlášení.
- `Obraz.java` – Vytváření, zpracování a komprese náhledů a celých obrazů. Reakce na změny prostředí (změna rozlišení obrazu) a změny požadavků (perioda vysílání náhledů, rozměry náhledů atd.).

- `Ovladani.java` – Vykonávání pohybu kurzoru myši, stisku a uvolňování tlačítek klávesnice a myši, rolování kolečka myši.
- `PosilaniZprav.java` – Zobrazování textových zpráv (varování) od dohledu. Odesílání zpráv (odpovědí) dohledu.
- `Prihlaseni.java` – Odesílání vyplněných přihlašovacích údajů centru. Zobrazování chybových oznámení (nemožnost vytvoření spojení, odmítnutí atd.).
- `RegSifrovani.java` - Šifrování a dešifrování komunikace. Generování a uchování tajných hesel.
- `ProudoveSifrovani.java` – Generování klíče z hesla. Šifrování a dešifrování obrazových dat algoritmem RC4.
- `SouborovyManazerServer.java` – Procházení disků, práce se soubory a adresáři (kopírování, mazání a vytváření). Obousměrný přenos dat mezi pracovní stanicí a dohledem. Zobrazování chybových stavů (nemožnost provedení operace).

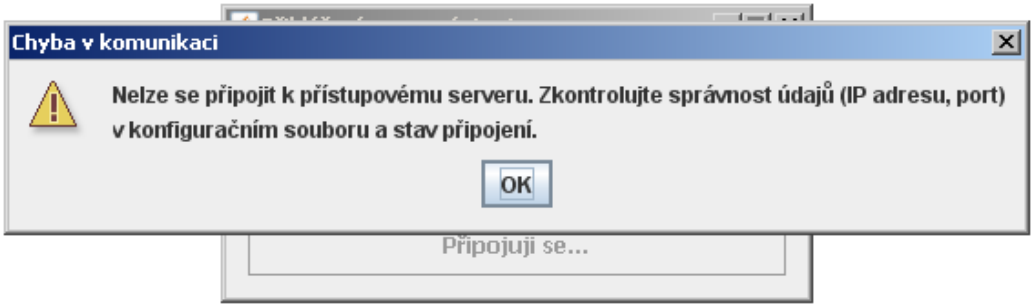
Grafické rozhraní

Aplikace je zaměřena na síťovou komunikaci. Komunikuje výhradně s centrem nebo dohledem a od místně přihlášeného uživatele (uživatele pracovní stanice) nevyžaduje žádnou interakci. Výjimku tvoří pouze fáze přihlašování do systému, dialogová chybová oznámení a rozhraní pro zobrazení zpráv od dohlížejícího uživatele. Na obr. 5.7 je zobrazeno rozhraní, které slouží pro přihlášení do systému. Kolonka „Jméno“ slouží pro zadání jména uživatele. Tato informace není tajná, ale je nutné, aby se v systému používala jedinečná jména (přístupový server si toto hlídá). Do kolonky „Heslo“ se zapisuje tajné heslo, a proto psané znaky nejsou zobrazovány (substituce černým „puntíkem“).

Důvodem, proč kolonka pro heslo má šířku pro zobrazení jednoho skrytého znaku je ten, aby případný útočník, sledující proces přihlášení, nezískal informaci o délce hesla. Tato funkce se nedá považovat za bezpečnostní opatření, ale v grafických rozhraních, určených pro přihlášení, je zvykem takto klamat případné útočníky. Jinou častěji používanou technikou je zobrazování odlišného počtu černých „puntíků“ od počtu právě zadaných znaků hesla. Na obr. 5.8 je zobrazen příklad chybového oznámení.



Obr. 5.7: Rozhraní pro přihlášení do systému



Obr. 5.8: Chybové oznámení

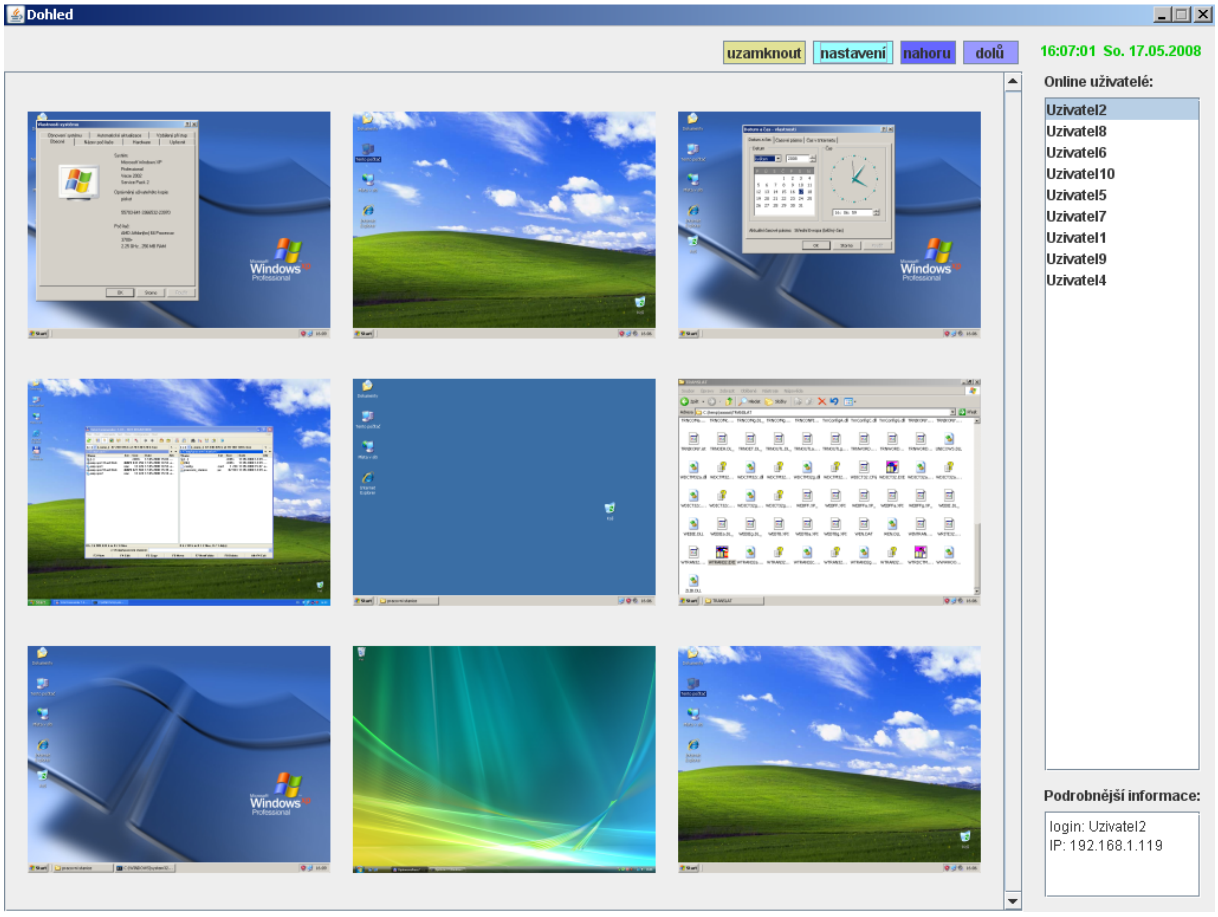
5.3.3 Dohlížecí stanice

Dohlížecí stanice – dohled je posledním jmenovaným objektem v systému. Tento dohled se může v systému vyskytovat vícekrát, viz kap. 5.3. Program, který umožňuje uživateli provádět dohlížení (sledování) a ovládání pracovních stanic, je uložen do souboru `dohlizejici_stanice.jar`. Program se výhradně ovládá pomocí grafického rozhraní a obsahuje mnoho grafických prvků. Je nutné distribuovat spolu s programem balíček `AbsoluteLayout.jar`, který musí být umístěn v adresáři `lib`. Program před zahájením své činnosti nejprve zkontroluje přítomnost a integritu konfiguračního souboru. Pokud konfigurační soubor není nalezen nebo obsahuje chyby, program zobrazí podrobnou chybovou zprávu a ukončí se.

Ze stejných důvodů, které byly uvedeny v kap. 5.3.1, je i zde konfigurační soubor v textové podobě a obsahuje doprovodné komentáře, popis zásad a příklady nastavení.

Funkce dohlížecí stanice

Tato aplikace, která umožňuje sledovat a ovládat vzdálené pracovní stanice, je ze všech aplikací nejrozsáhlejší. Na obr. 5.9 je zobrazeno hlavní okno aplikace. Jsou zde vidět náhledy obrazovek připojených pracovních stanic.



Obr. 5.9: Hlavní okno dohlížecí aplikace

Funkce dohlížecí stanice jsou následující:

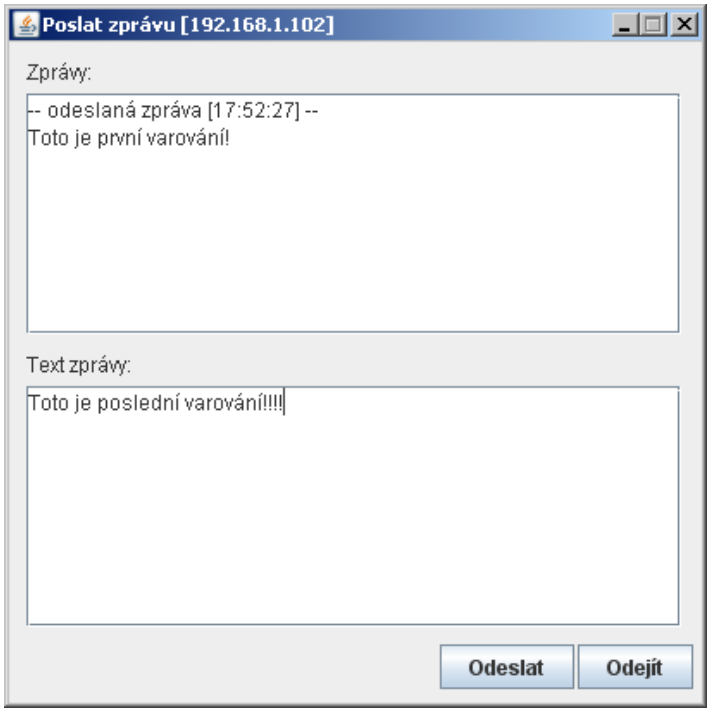
- **Přihlášení do systému.** Uživatel, který je organizací pověřen k dohlížení a správě stanic a má na přístupovém serveru vytvořen platný účet, se může přihlásit do systému

prostřednictvím rozhraní, které je velmi podobné jako na obr. 5.7. Liší se pouze v názvu okna. Pomocí tohoto rozhraní nejprve odešle přístupovému serveru zašifrované heslo, podobně jako pracovní stanice. Dále pomocí tohoto hesla je zašifrováno a posláno jméno a heslo uživatele. Pokud jsou poslaná data platná, bude stanice (uživatel) přihlášen a uložen do databáze přihlášených dohlížecích stanic. Dalším důležitým krokem v této fázi je sdělení veřejného klíče centru. Veřejný klíč si spolu s tajným klíčem dohlížecí stanice náhodně vygenerovala. Centrum si veřejný klíč uloží a pošle dohlížecí stanici seznam již přihlášených pracovních stanic. Po odeslání seznamu pak centrum všem pracovním stanicím (stávajícím i následně přihlášeným), které se nacházejí ve stejné logické oblasti jako dohlížecí stanice, odešle tento veřejný klíč.

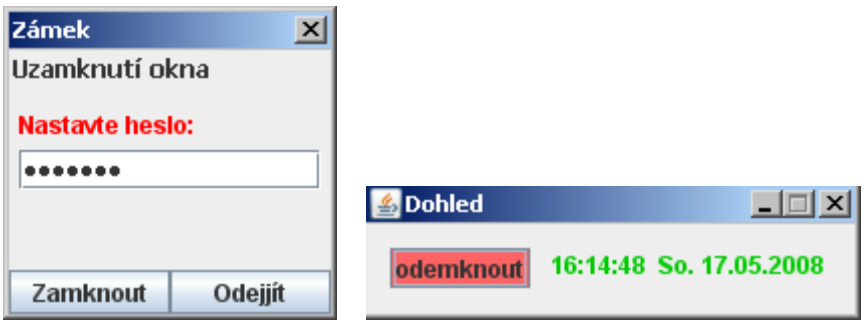
- **Zpracování (přidání) nové pracovní stanice.** Pracovní stanice, poté co získá od centra IP adresu a veřejný klíč, zašle dohledu tajné heslo, kterým bude komunikace šifrována. Po provedení fáze připojení k dohledu, bude pracovní stanice plně k dispozici. Dohlížejícímu uživateli se na obrazovce ukáže náhled pracovní plochy pracovní stanice, který se bude podle zvolené periody automaticky aktualizovat. S každou nově připojenou pracovní stanicí se vytvoří odpovídající náhled, který je vždy umístěn na konec „seznamu náhledů“.
- **Zrušení pracovní stanice.** V případě, že dojde k ukončení (přerušení) spojení mezi dohledem a pracovní stanicí, nebude možno vzdálenou stanici ovládat. Dojde k automatickému odstranění náhledu ze „seznamu náhledů“ a dalších informací, které souvisí se „ztracenou“ stanicí. Pokud smazaný náhled není na poslední pozici, vznikne prostorová mezera mezi náhledy. Proto je vždy volána funkce, která automaticky přesune poslední náhled do vzniklé mezery. Tím je vždy zaručena přehlednost a homogenní rozvržení náhledů.
- **Sledování činnosti.** Uživateli, který provádí dohled a správu, se periodicky aktualizují náhledy ploch obrazovek pracovních stanic. Na své obrazovce může v jednom okamžiku sledovat mnoho pracovních stanic. Počet zároveň zobrazitelných náhledů je určen pouze rozlišením (rozměry) obrazovky a velikosti náhledů. Čím větší rozlišení a menší rozměry náhledů budou zvoleny, tím bude možno zároveň sledovat více počítačů.
- **Ovládání.** Tato funkce umožňuje vzdálený počítač ovládat pomocí klávesnice a myši. Ovládání je možné v plnohodnotném nebo náhledovém režimu. Ovládání v náhledovém režimu je vhodné v případě potřeby něco rychle „odkliknout“, např. dialogové oznámení a nabídky v procesu hromadné (ruční) instalace softwaru. Pro dekompresi všech obrazů je použita „javovská“ funkce *Inflater*.
- **Odesílání a příjem textových zpráv.** Dohlížející má nejenom možnost sledovat a ovládat, ale i komunikovat s uživatelem prostřednictvím výměny textových zpráv. To je velmi praktické. Například dohlížející (učitel) může poslat varování studentovi, který se nevěnuje zadané činnosti. Na obr. 5.10 je zobrazeno grafické rozhraní okna pro psaní a posílání zpráv.
- **Zámek.** Tato funkce umožňuje, za pomoci hesla, bezpečně uzamknout aplikaci. Rozhraní pro zadání hesla je zobrazeno na obr. 5.11a. Při uzamknutí dojde k deaktivaci příjmu náhledů a celé grafické rozhraní se změní na malé okno, ve kterém je pouze možno se přihlásit po zadání správného hesla, viz obr. 5.11b. Pracovním stanicím je sděleno, aby přešli do režimu spánku. Uzamknutí aplikace je vhodné např. v případě, kdy dohlížející uživatel potřebuje opustit počítač (pracoviště) a nechce nebo nemůže aplikaci ukončit.
- **Přenos dat.** Do aplikace je integrován souborový manažer, který umožňuje přístup ke všem diskům, mechanikám a paměťovým médiím, které jsou sdíleny na straně pracovní stanice. Je podporováno kopírování, mazání a vytváření souborů a adresářů. Pokud nastane neočekávaná situace, např. nemožnost čtení, zápisu atd., zobrazí se dialogové okno popisující důvod nevykonání požadavku. Při přenosu souborů je zobrazen jejich název a absolutní cesta. V kap. 5.5 jsou v tab. 5.4 a 5.5 uvedeny výsledky testů rychlosti přenosu dat. Ukázka činnosti souborového manažera je na obr. 5.12.
- **Příjem obrazu.** Dohlížecí stanice přijímá pouze část obrazu, která se liší od předchozí části. Před zobrazením obrazu dochází nejprve k dešifrování a dekompresi. Po těchto

procedurách je získaná část obrazu umístěna na správné místo (souřadnice) a zobrazena. Vzhled okna je shodný s obr. 4.4.

- **Nastavení prostředí.** Aplikace pro dohled a řízení umožňuje jisté přizpůsobení. Jedná se o nastavení rozměrů náhledu, periody aktualizace, směr řazení náhledů a jiné. Některá nastavení přímo ovlivňují parametry komunikace. Grafické rozhraní nabídky nastavení je na obr. 5.13.
- **Nastavení parametrů komunikace uživatelem.** Dohlížející uživatel mění parametry ručně. Má možnost volby portů (viz kap. 5.3.4), periody aktualizace náhledů, velikost rozměrů náhledů, a posílání „uspávacích" signálů. Poslední tři zmiňované volby přímo ovlivňují zatížení sítě (objem přenášených dat) a lze ovlivňovat všechny pracovní stanice nebo jednu konkrétní.
- **Nastavení parametrů komunikace aplikací.** Podobně jako v předchozím bodě dochází ke změně parametrů. Nastavení je prováděno automaticky bez zásahu dohlížejícího uživatele. Podrobnosti jsou popsány v následující podkapitole.
- **Vyhledání a zvýraznění.** Aplikace umožňuje sledovat mnoho pracovních stanic – náhledů. V tak velkém množství je těžké z náhledů rozpoznat, o kterou pracovní stanici se jedná. Aplikace tedy obsahuje seznam jmen připojených uživatelů – stanic. Po kliknutí na konkrétní jméno se automaticky vyhledá a zvýrazní (orámuje) příslušný náhled.

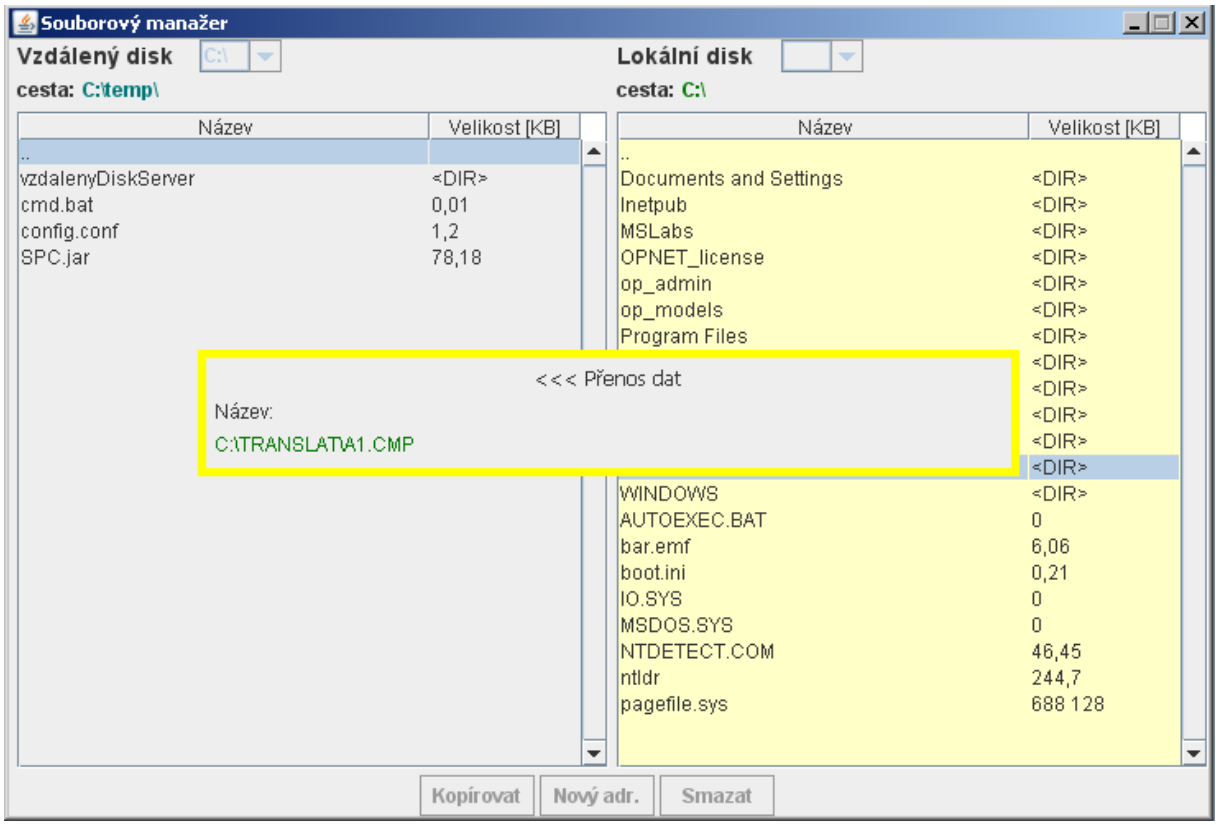


Obr. 5.10: Rozhraní pro psaní a odesílání zpráv

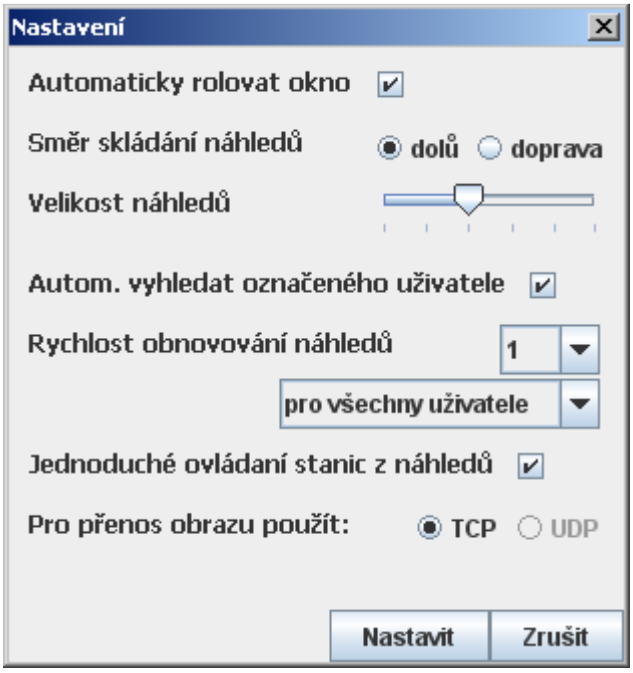


a) b)

Obr. 5.11: Zámek: a) uzamknutí aplikace, b) zamknutá aplikace



Obr. 5.12: Ukázka činnosti souborového manažera



Obr. 5.13: Možnosti nastavení

Automatické nastavení parametrů

V průběhu práce si dohlížející uživatel může nastavit různou velikost náhledu a periodu jeho obnovování. Při změně těchto parametrů jsou všechny nebo jedna (dle výběru) stanice informovány o nových parametrech. Aplikace se pak bude chovat podle přijatých příkazů. V případě přihlášení (připojení) nové pracovní stanice dohledu, budou této stanici automaticky sděleny aktuální parametry.

V případě, že dohlížející uživatel minimalizuje aplikaci, dojde k vyslání „uspávacího" signálu. Tím je zaručeno, že aplikace nebudou zbytečně vysílat nové náhledy, které nemůžou být zobrazeny – sledovány (aplikace je minimalizována do panelu).

Vzhledem k tomu, že může být sledováno a ovládáno velké množství pracovních stanic, jsou příkazy rozesílány pomocí multicastu. Multicast je odeslán všem pracovním stanicím, které patří do shodné logické oblasti jako dohlížecí stanice. V případě, že některé stanici příkaz nedojde (např. z důvodu chyby – poškození nebo nedoručení paketu), bude dodatečně znovu informována o změnách parametru, tentokrát pomocí zabezpečeného TCP protokolu.

Organizace kódu

Funkcionalita programu je rozvržena do 27 souborů (tříd). V následujících bodech jsou stručně popsány hlavní funkce v nich obsažené:

- `AktualizaceCasu.java` – Periodická aktualizace systémového času.
- `AktualizaceNazvu.java` – Zobrazování názvů a cest přenášených souborů.
- `CasovacOramovani.java` – Po stanoveném intervalu se zruší orámovaní náhledu.
- `HlavniOkno.java` – Hlavní pracovní okno. V tomto okně jsou zobrazeny náhledy, seznam uživatelů, podrobnější informace, tlačítka pro zpřístupnění nastavení, zámku atd. Zamezení vytvoření dalších instancí přenosu obrazu v plnohodnotném režimu, pokud už je jedna instance vytvořena. To samé u přenosu souborů (souborový manažer) a textových zpráv.
- `Komunikace.java` – Vytváření spojení a uskutečňování komunikace s centrem a pracovní stanicí na specifických portech. Hromadné (multicastové) vysílání příkazů. Rozlišování (identifikace typu) příkazu a odpovědi.
- `Konfigurace.java` – Nastavení parametrů na základě obsahu konfiguračního souboru `config.conf`.
- `Main.java` – Spuštění programu. Volání funkce pro přihlášení.
- `Nastaveni.java` – Třída pro uchování globálních nastavovacích parametrů.
- `NastaveniGUI.java` – Rozhraní, ve kterém je možno provést různá nastavení.
- `NovyAdresar.java` – Rozhraní pro vytvoření nového adresáře.
- `NovyKlient.java` – Obsluha nového uživatele. Vytvoření nové instance a vlákna. Příjem přihlašovacích údajů. Volání funkce pro ověření a uložení.
- `OdpocetSpatnehoHesla.java` – Pozastavení možnosti zadání hesla dokud se nedokončí (bezpečnostní) odpočet.
- `OvereniHesla.java` – Výpočet haše ze vstupního řetězce (hesla) a ověření shody.
- `OvladaniPCOkno.java` – Zobrazování obrazu o menších nebo větších rozměrech (vzhledem k rozměrům obrazové plochy na hostitelském systému) v plnohodnotném módu.
- `OvladaniPCPlneRozmery.java` – Zobrazování obrazu, který má shodné rozměry jako obrazová plocha na hostitelském systému, v plnohodnotném módu.
- `OvladaniPCStart.java` – Nastavení instancí a přepínání mezi zobrazením v okně a plném zobrazení.
- `PosilaniZprav.java` – Posílání a příjem textových zpráv.
- `Prenos.java` – Zobrazení dialogu, který informuje o průběhu přenosu souborů.

- `Prihlaseni.java` – Odesílání vyplněných přihlašovacích údajů centru. Zobrazování chybových oznámení (nemožnost vytvoření spojení, odmítnutí atd.).
- `ProudoveSifrovani.java` – Generování klíče z hesla. Šifrování a dešifrování obrazových dat algoritmem RC4.
- `RegSifrovani.java` – Šifrování a dešifrování komunikace. Generování veřejného a tajného klíče. Uchování tajných hesel.
- `SignalOkno.java` – Zobrazení nabídky umožňující minimalizaci okna nebo ukončení příjmu obrazu v plnohodnotném režimu.
- `SouborovyManazer.java` – Rozhraní pro správu souborů. Možnost vícenásobného označení souborů, kopírování, mazání, vytváření adresářů, výběr a procházení diskových jednotek (lokálních i vzdálených).
- `SouborovyManazerChyba.java` – Zobrazování chybových oznámení.
- `SpatneHeslo.java` – Zobrazení oznámení o špatně zadaném heslu při odemykání aplikace. Zpřístupnění možnosti nového zadání po bezpečném odpočtu.
- `UzivatelPanel.java` – Kontejner obsahující informace o konkrétní přihlášené pracovní stanici. Rovněž se do něj ukládá obraz náhledu.
- `Zamek.java` – Rozhraní pro uzamknutí a odemknutí aplikace.

5.3.4 Konfigurace

Konfigurační soubory slouží pro uchování hodnot nastavovacích parametrů. Tyto soubory mohou mít binární nebo textovou podobu. Nevýhodou při editaci binárních souborů je potřeba použití vhodného nástroje, který umožňuje konfigurační soubory modifikovat. Textové soubory je možné editovat libovolným textovým editorem. Jsou snadno čitelné a modifikovatelné. Z výše popsaných důvodu jsou konfigurační soubory textové. Obsahují doprovodné komentáře, popis zásad a příklady nastavení.

Všechny vytvořené aplikace mimo doplňkových, které jsou uvedeny v kap. 5.4, potřebují pro svou činnost přečíst informace z konfiguračního souboru `config.conf`. Pokud konfigurační soubor neexistuje, aplikace zobrazí chybové oznámení a ukončí se. V případě, že konfigurační soubor existuje, provedou se testy, které ověří korektnost a syntax zadaných hodnot. To je velmi důležité, protože člověk je tvor omylný a snadno se může splést či přehlédnout. U rozsáhlých konfigurací by pak bez zpětné vazby bylo velmi složité a zdlouhavé hledat příčinu nefunkčnosti programu. V případě nefunkčního složitého systému, který se skládá z mnoha konfigurovatelných prvků, je nalezení a odstranění problému daleko náročnější.

Všechny funkce provádějící ověření korektnosti a správné syntaxe zápisu jsou umístěny ve zdrojovém souboru `Komunikace.java`. Každý řádek, který začíná středníkem („;“), je brán jako komentář a při prohledávání souboru bude přeskočen. Rovněž také prázdné řádky budou ignorovány. V tab. 5.3 je zobrazen seznam a vysvětlení významu existujících klíčových slov – parametrů, ke kterým jsou vždy těsně za dvojtečkou („:“) umístěny náležející hodnoty. Sloupec „výskyt“ udává, v kterých aplikacích se parametr vyskytuje. Na obr. 5.14 je ukázka zápisu konfiguračních údajů.

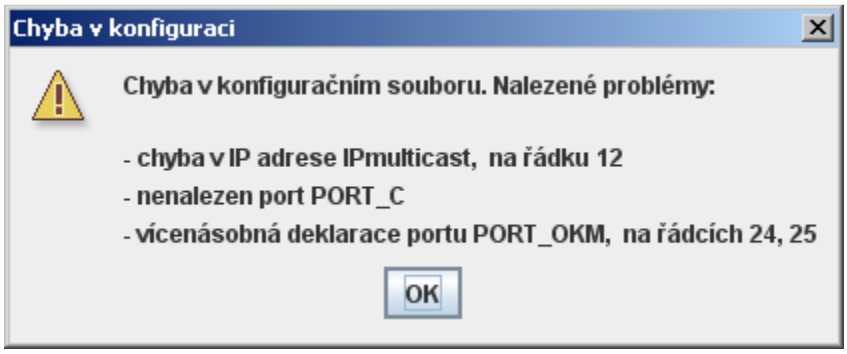
```
;IPv4 adresa centra
IP_C:192.168.1.10

;IPv4 multicast vysílaný všem sledovaným stanicím
IPmulticast:234.234.234.234

;PORT centra
PORT_C:54321
```

Obr. 5.14: Ukázka zápisu konfiguračních údajů

Každý uvedený parametr je povinný. V případě, že konfigurační soubor obsahuje alespoň jednu chybu, je zobrazeno podrobné chybové oznámení. Testovací mechanismy detekují tyto chyby:

- nesprávná hodnota parametru – port a IP mimo rozsah
- shoda čísel portů – nemožnost vícero naslouchání (serverů) na jednom portu
- neexistence parametru – nemožnost nastavení
- vícenásobná deklarace stejného parametru - nejednoznačnost

Na obr. 5.15 je ukázka podrobného chybového oznámení. To oznamuje druh chyby a umístění v konfiguračním souboru.



Obr. 5.15: Podrobné chybové oznámení

Tab. 5.3: Seznam a popis parametrů v konfiguračích souborech

Parametr	Výskyt	Význam
PORT_C	centrum, pracovní stanice, dohlížecí stanice	Port, na kterém naslouchá centrum a ke kterému se připojují ostatní stanice.
DATB_A	centrum	Cesta k databázi IP adres.
DATB_U	centrum	Cesta k databázi uživatelů.
SK_C	centrum	Tajný klíč centra.
IP_C	pracovní stanice, dohlížecí stanice	IP adresa centra.
IPmulticast	pracovní stanice, dohlížecí stanice	IP adresa multicastové skupiny.
PORT_PS	pracovní stanice, dohlížecí stanice	Port, na kterém naslouchá dohlížející stanice a ke kterému se připojují pracovní stanice.
PORT_PSmulticast	pracovní stanice, dohlížecí stanice	Port, na kterém pracovní stanice naslouchají multicastovému vysílání od dohlížející stanice.
PORT_OKM	pracovní stanice, dohlížecí stanice	Port, na kterém pracovní stanice naslouchají. Určen pro příjem ovládání (klávesnice a myši).
PORT_SM	pracovní stanice, dohlížecí stanice	Port, na kterém pracovní stanice naslouchají. Určen pro přenos souborů.
PK_C	pracovní stanice, dohlížecí stanice	Veřejný klíč centra.

5.4 Doplňkové aplikace

5.4.1 Výpočet SHA-256 a RSA klíčů

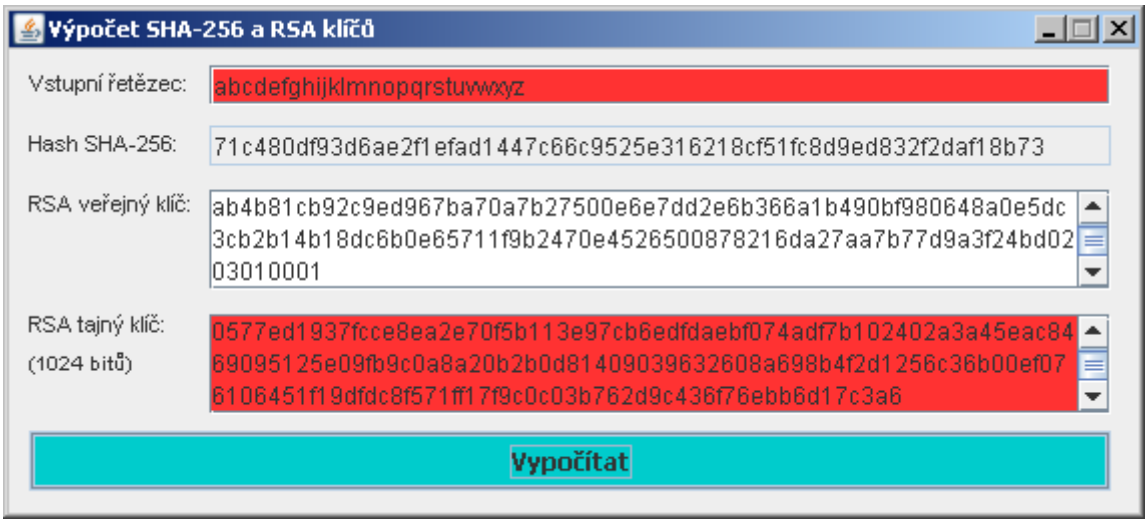
V kap. 5.3 je popsán navržený systém dohledu a ovládání stanic. Systém obsahuje bezpečnostní mechanismy založené na symetrické a asymetrické kryptografii. Volba hesla pro použití v symetrických algoritmech je velice jednoduchá. Postačí pouze znát minimální, případně maximální délku hesla a mít jistou dávku představivosti při volbě hesla. Je vhodné při volbě hesla (klíče) znát bezpečnostní doporučení, jako např. složité heslo, výskyt číslic, velkých a malých znaků atd. Tuto činnost může provést kdokoliv a kdykoliv a bez použití softwarových nástrojů. Jediným místem, kde je vyžadováno heslo od uživatele, je při zamykání a odemykání aplikace.

Výše zmíněné v případě výběru klíče u asymetrických algoritmů neplatí. Důvodem je potřeba dlouhých a rozdílných klíčů, protože jedním – veřejným klíčem jsou data šifrována a druhým – tajným klíčem jsou data dešifrována (podepisována). Tento pár klíčů je absolutně nemožné si vymyslet „jen tak z hlavy". Je třeba postupovat podle jistých matematických vztahů, které se pro každý algoritmus liší. V aplikacích *pracovní stanice* a *přístupová stanice* je použit veřejný klíč centra (přístupového serveru), který je uložen v konfiguračním souboru. Naopak *centrum* používá tajný klíč, který je také uložen v konfiguračním souboru.

Pro snadnou a pohodlnou změnu klíčů jsem vytvořil aplikaci, která toto umožňuje. Aplikace je zobrazena na obr. 5.16. Uživatel pouze zmáčkne tlačítko „Vypočítat" a vygeneruje veřejný a tajný klíč. Do textových polí, ve kterých jsou vypsány klíče, nelze psát. Tím je eliminována možnost nechtěného „přiklepnutí" dalšího znaku při kopírování klíčů do schránky. Pro potřebu vygenerování dalších klíčů stačí opět stisknout tlačítko.

Aplikace dále umožňuje vypočítat haš SHA-256 ze vstupního řetězce. Vypočítaný haš – otisk je také chráněn proti přepisu a lze jej pohodlně označit a zkopírovat do schránky. Vstupní řetězec je možné modifikovat a vypočítat tak nový haš nového řetězce. Při kopírování do schránky může dojít k nechtěné změně vstupního řetězce. Proto při modifikaci vstupního řetězce dojde k vymazání výsledného haše. Tím je zabráněno omylům, které jsou způsobené překlepy (příklepy).

Textová pole vyplněná červenou barvou označují tajné informace (vstupní řetězec je zde chápan jako heslo).



Obr. 5.16: Doplňková aplikace - výpočet SHA-256 a RSA klíčů

5.4.2 Souborový manažer

Možnost přistupovat neomezeně a transparentně ke vzdáleným datům, podobně jako k datům na lokálním disku, je často potřebná. Proto jsem se rozhodl vytvořené moduly (viz kap. 5.2), které jsou integrovány do pracovní a dohlížecí stanice, vypracovat do samostatných programů – lokální a vzdálené části. Díky těmto aplikacím lze rychle a jednoduše přistupovat ke všem diskům počítače. Grafické rozhraní se nezměnilo, pouze přibyly přihlašovací a nastavovací dialogy. Obrázek grafického rozhraní souborového manažera je na obr. 5.12.

V následující kapitole 5.5 jsou zobrazeny výsledky praktických testů rychlosti přenosu dat a vliv komprese obrazu na objem přenášených dat.

5.5 Testy

Funkčnost všech aplikací byla otestovaná ve virtuální i v reálné (školní) síti na virtuálních i reálných strojích - počítačích. V následujících tabulkách jsou zobrazeny výsledky testů rychlosti přenosu dat. Při testování byl použit volně dostupný FTP server *CesarFTP v. 0.99g* a výše jmenovaný souborový manažer.

Tab. 5.4: Porovnání rychlosti kopírování dat na vzdálený disk

Kopírování na vzdálený disk				
Protokol	**Počet souborů**	**Celková velikost [MB]**	**Čas [s]**	**Typ dat**
vlastní	2500	57	340	www stránky
FTP	2500	57	570	www stránky
vlastní	1	280	32	video
FTP	1	280	35	video

Tab. 5.5: Porovnání rychlosti kopírování dat na lokální disk

Kopírování na lokální disk			
Počet souborů	**Celková velikost [MB]**	**Čas [s]**	**Typ dat**
2500	57	45	www stránky
1	280	25	video

V tab. 5.6 jsou zobrazeny výsledky měření velikosti datového toku při přenosu obrazu. Objem přijatých komprimovaných dat je podobný jako v případě přenosu obrazu programem UltraVNC, viz pátý řádek v tab. 4.2. Měření probíhalo pomocí stejné metody, která je uvedena v kap. 4.1.

Tab. 5.6: Porovnání velikosti přenesených dat

Rozlišení	**Hloubka barev**	**Přijato [KB]**	**Odesláno [KB]**	**Nastavení**
1024 × 768	32	14 305	318	s kompresí
1024 × 768	32	82 432	1212	bez komprese

Na přiloženém CD se nacházejí video soubory, ve kterých je zachycena praktická činnost a testování. Celková délka videozáznamů je 12 minut.

5.6 Návrhy na rozšíření

Velkou snahou bylo navrhnout a implementovat takový systém, který by byl funkční, bezpečný, efektivní, jednoduchý a prakticky použitelný. **Po splnění všech bodů zadání mé diplomové práce, jsem v práci dál pokračoval. Aplikace jsem rozšiřoval a vylepšoval až do současné podoby. Zdrojové kódy všech aplikací mají celkem téměř 11000 řádků.** Vzhledem k časovému omezení práce nebylo možno realizovat všechny nápady. Prakticky vzato lze neustále vylepšovat a inovovat jakýkoliv technický produkt. V následujících bodech uvádím několik návrhů na další užitečná rozšíření:

- Ukládání (archivování) přijatých náhledů obrazovek sledovaných pracovních stanic.
- Vylepšení kompresních technik.
- Automatické provádění efektivních (přírůstkových, rozdílových) záloh dat po síti.
- Využití již existujících databází uživatelských účtů přístupovým serverem.
- Spuštění dohlížecí aplikace z internetového prohlížeče – program ve formě appletu nebo Java Web Start aplikace.
- Portování na mobilní architekturu.
- Přenos obrazu z připojených kamer (web kamer)
- Systém pro výměnu textových a multimediálních dat mezi uživateli pracovních stanic.
- Pokročilý informační systém.
- Přenos hlasu.
- Ovládání pracovních stanic hlasem.
- Ovládání dalších elektronických a elektrických zařízení, např. dataprojektoru, ozvučovací techniky, osvětlení, klimatizace atd.

6 ZÁVĚR

Cílem diplomové práce bylo prostudovat problematiku terminálových služeb se zaměřením na grafické terminály. Prostudovat jejich vlastnosti, dostupné funkce, výhody a nevýhody. Navrhnout a realizovat systém umožňující vzdálenou kontrolu (dohled) a obsluhu pracovních stanic. Dále navrhnout a realizovat řešení umožňující snížení objemu přenášených dat a jejich zabezpečení.

Téma diplomové práce mě zaujalo a po splnění všech zadaných cílů jsem v práci pokračoval a vypracovaný systém jsem dále zdokonaloval a rozšiřoval. V práci je navržen metodický postup práce, podle kterého jsem se řídil. Tento postup mi umožňoval snazší implementaci a testování nových funkcí. Dále je zde popsáno pracoviště, které bylo vytvořeno pomocí virtualizačních nástrojů. Toto pracoviště mi poskytlo značný komfort práce.

Výsledkem mé práce je dynamický systém, umožňující dohlížejícímu uživateli (dohledu) sledování činnosti uživatelů přihlášených k pracovním stanicím. Dohlížející uživatel má dále možnost stanice ovládat pomocí klávesnice a myši a přistupovat ke všem paměťovým zařízením (diskům), kterými daná vzdálená stanice disponuje. Přístup ke vzdáleným i lokálním souborům zajišťuje síťový souborový manažer. Dohlížecí uživatel má také možnost komunikovat se sledovanými uživateli prostřednictvím textových zpráv.

Dohlížející uživatel si může aplikaci přizpůsobit dle svých požadavků. Aplikace umožňuje měnit velikost rozměrů náhledů (miniatur) obrazovky vzdálené pracovní stanice, periodu aktualizace náhledů a styl jejich seřazování. V případě dočasného opuštění pracoviště je možno aplikaci pomocí hesla uzamknout.

Komunikace je komprimována a šifrována. Jsou zde použity symetrické a asymetrické šifrovací algoritmy. Všechny aplikace se konfigurují pomocí textových souborů. Dále jsem vytvořil dvě doplňkové aplikace – generování klíčů RSA a samostatný síťový souborový manažer.

V závěru práce jsou uvedeny výsledky výkonnostních testů. Měřením bylo zjištěno, že rychlost přenosu mnoha malých souborů pomocí souborového manažeru (vlastní řešení) je přibližně 2x větší než pomocí protokolu FTP a rychlost přenosu velkých souborů je stejná.

Navržený systém se skládá téměř ze 40 modulů. Rozsah všech zdrojových kódů činí téměř 11000 řádků. Všechny aplikace jsou napsané v jazyce Java, čímž je zaručena přenositelnost programů mezi různými platformami. Na přiloženém CD se nacházejí video soubory, ve kterých je zachycena praktická činnost a testování.

Literatura

[1] Kříž, L.: *Komprimační a archivační programy*. Computer Press, Praha 2002, ISBN 80-7226-757-4

[2] Vlček, K.: *Komprese a kódová zabezpečení v multimediálních komunikacích*. BEN, Praha 2004, ISBN 80-7300-134-9

[3] Čapek, J., *Fabian, P.: Komprimace dat – principy a praxe*. Computer Press, Praha 2000, ISBN 80-7226-231-9

[4] Morkes, D.: *Komprimační a archivační programy*. Computer Press, Brno 1998, ISBN 80-7226-089-8

[5] Nelson, M.: *The Data Compression Book*. IDG Books Worldwibe, Inc, ISBN 1558514341

[6] Adámek, J.: *Kódování a teorie informace*. České vysoké učení technické, Praha 1991

[7] Salomon, D.: *Data Compression – The Complete Rreference, Springer*. ISBN 0-387-40697-2

[8] Pužmanová, R.: *Moderní komunikační sítě od A do Z*, 2. vydání. Computer Press, Brno 2006, ISBN 80-251-1278-0

[9] Herout, P.: *Učebnice jazyka Java*. Kopp, České Budějovice 2006, ISBN 80-7232-115-3

[10] Herout, P.: *Java – grafické uživatelské prostřední a čeština*. Kopp, České Budějovice 2004, ISBN 80-7232-237-0

[11] The RFB Protocol. Dostupné z URL: <http://www.realvnc.com/docs/rfbproto.pdf>

[12] Wikipedie, otevřená encyklopedie. Dostupné z URL: <http://www.wikipedia.org/>

[13] Kopplin, J.: An Illustrated History of Computers. Dostupné z URL: <http://www.computersciencelab.com/ComputerHistory/HistoryPt4.htm>

[14] InTouch pro Terminálové služby. Dostupné z URL: <http://www.pantek.cz/produkty.php?id_produktu=21&produkt=intouch-pro-terminalove-sluzby&id_podkategorie=38&podkategorie=anotace>

[15] Houser, P.: Vzpomínky na dobu sálovou. Dostupné z URL: <http://archiv.cw.cz/cwarchiv.nsf/clanky/469AA88AFDB62087C1256EBC0035F132?OpenDocument>

[16] Terminálové služby. Dostupné z URL: <http://www.adminxp.cz/windows2000/index.php?aid=152>

[17] Wikipedia: Remote Desktop Protokol. Dostupné z URL: <http://en.wikipedia.org/wiki/Remote_Desktop_Protocol>

[18] Vyskočil, M.: VNC – používáme vzdálený desktop. Dostupné z URL: <http://www.abclinuxu.cz/clanky/site/vnc-pouzivame-vzdaleny-desktop>

[19] Tišer, M.: Terminálové služby. Dostupné z URL: <http://www.microsoft.com/cze/technet/clanky/terminal_services.mspx>

[20] Wikipedia: Computer terminal. Dostupné z URL: <http://en.wikipedia.org/wiki/Computer_terminal>