

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Test grafického zobrazení identických dat na win32 a
linuxové platformě.

Štěpán Lerch

Bakalářská práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Štěpán Lerch
Osobní číslo: I08101
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Název tématu: Test grafického zobrazení identických dat na win32 a linuxové platformě.
Zadávající katedra: Katedra informačních technologií

Z á s a d y p r o v y p r a c o v á n í :

Student vyrobí jednoduché programy zobrazující identické náročné grafické objekty v kritickém časovém intervalu v programech pro win32 platformu a linuxovou platformu. Nejlépe aby programy byly portovány (technologie QT nebo i jiné technologie, ale je nutné vyrobit podobné, identické programy). Student v tabulkách a graficky v dostupných nástrojích bude analyzovat a srovnávat rychlost, paměťovou náročnost a náročnost strojového času při vykreslování na různých platformách. Programovací jazyk je na výběru studenta.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

Velký průvodce protokoly TCP/IP a systémem DNS, L. Dostálek, A.Kabelová, Computer Press

Mistrovství C++, Stephen Prata, Computer Press

OpenGL Průvodce programátora, D.Shreiner, M.Woo, J.Neider, T.Davis, Computer Press

www.MSDN.com

Vedoucí bakalářské práce:

Ing. Jaroslav Štroch

Katedra informačních technologií

Datum zadání bakalářské práce: **21. prosince 2012**


Termín odevzdání bakalářské práce: **10. května 2013**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 29. března 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.



V Pardubicích dne 10. 2. 2013

Štěpán Lerch

Poděkování

V úvodu práce bych rád poděkoval vedoucímu své bakalářské práce Ing. Jaroslavu Štrochovi za poskytnuté konzultace a rady v průběhu práce. Dále bych chtěl poděkovat své rodině za poskytnutou podporu v průběhu studií.

Anotace

Student vyrobí jednoduché programy zobrazující identické náročné grafické objekty v kritickém časovém intervalu v programech pro win32 platformu a linuxovou platformu. Nejlépe aby programy byly portovány (technologie QT nebo i jiné technologie, ale je nutné vyrobit podobné, identické programy). Student v tabulkách a graficky v dostupných nástrojích bude analyzovat a srovnávat rychlost, paměťovou náročnost a náročnost strojového času při vykreslování na různých platformách. Programovací jazyk je na výběru studenta.

Klíčová slova

Java, Windows, Linux, grafika

Title

Test of graphic display of identic data on win 32 and on linux platform

Annotation

Student creates simple programmes which show identical ambitious graphic objects in critical time interval in programmes for win32 platform and for linux platform. Best the programmes would be ported (technology QT or the other technologies, but it is necessary to create similar, identical programmes). Student in charts and graphically by available tools will analyse and compare speed, memory workload and workload of machine time by drawing on various platforms. Programme language is on choice of student.

Keywords

Java, Windows, Linux, graphics

Obsah

Obsah	14
Seznam zkratek	16
Seznam obrázků	17
Seznam tabulek	17
Úvod	18
1 Vymezení základních pojmů	19
1.1 Operační Systém.....	19
1.2 Monolitické jádro	20
1.3 Mikrojádro.....	20
1.4 Hybridní jádro	21
1.5 Microsoft Windows	21
1.6 GNU/Linux.....	22
1.7 Virtuální stroj.....	22
1.8 Open source	23
1.9 Knihovna	23
1.10 API.....	23
1.11 Procesy	23
1.12 Vlákna.....	23
2 Použité programy a technologie	24
2.1 Oracle VM VirtualBox	24
2.2 Java Virtual Machine.....	24
2.3 NetBeans.....	24
2.4 Java	24
2.5 Třída Graphics	25
2.6 Třída Graphics2D	25
2.7 Knihovna Swing	26
2.8 Windows XP.....	26
2.9 Ubuntu 12.04 Precise Pangolin – LTS	27
3 Vzhled a možnosti aplikace	27
3.1 Jednotlivé funkce aplikace.....	29

3.1.1	Vlastnosti objektu.....	29
3.1.2	Velikost objektu.....	29
3.1.3	Počet replikací	29
3.1.4	Generování.....	29
3.1.5	Ukládání do souboru.....	30
3.1.6	Průběh generování	30
3.2	Vlastní třídy a metody	30
3.2.1	Demo	31
3.2.2	GraphicsObject	31
3.2.3	GraphicsManager.....	31
3.2.4	ObjectEnum.....	31
4	Metody měření.....	31
4.1	Záznam vytížení systémových prostředků	31
4.1.1	Process Explorer.....	31
4.1.2	Linux Process Monitor	31
4.2	Způsob měření	32
4.3	Parametry spuštěných měření	32
5	Získaná data.....	32
5.1	Struktura jednotlivých měření	32
5.2	Měření 1.....	33
5.3	Měření 2.....	34
5.4	Měření 3.....	35
	Závěr.....	37
	Literatura	38
	Příloha A – Popis přílohy.....	40
	Příloha B – Zdrojový kód souboru bootstrap.php	41

Seznam zkratk

OS	Operační systém
GUI	Graphics user interface
JVM	Java virtual machine
API	Application programming interface
HW	Hardware
SW	Software
CSV	Comma-separated values

Seznam obrázků

Obrázek 1 - Graf zastoupení operačních systémů v letech 2003 až 2010	19
Obrázek 2 - Architektura monolitického jádra	20
Obrázek 3 - Architektura mikrojádra.....	20
Obrázek 4 - Architektura hybridního jádra.....	21
Obrázek 5 - Hierarchie tříd knihovny Swing.....	26
Obrázek 6 - Okno aplikace	28
Obrázek 7 - Vlastnosti objektu	29
Obrázek 8 - Velikost objektu.....	29
Obrázek 9 - Počet replikací	29
Obrázek 10 - Generování.....	30
Obrázek 11 - Ukládání dat.....	30
Obrázek 12 - Průběh generování	30
Obrázek 13 - Hraníční údaje (Měření 1)	33
Obrázek 14 - Průběh replikací (Měření 1).....	33
Obrázek 15 - Vytížení procesoru (Měření 1).....	34
Obrázek 16 - Hraníční údaje (Měření 2)	34
Obrázek 17 - Průběh replikací (Měření 2).....	35
Obrázek 18 - Vytížení procesoru (Měření 2).....	35
Obrázek 21 - Vytížení procesoru (Měření 3).....	36

Seznam tabulek

Tabulka 1 - Chronologie systémů Windows (nekompletní).....	22
Tabulka 2 - Systémové požadavky systému Windows XP	27
Tabulka 3 - Výpis procesů Windows XP	40
Tabulka 4 - Výpis procesů Ubuntu (Část 1)	41
Tabulka 5 - Výpis procesů Ubuntu (Část 2)	42
Tabulka 6 - Výpis procesů Ubuntu (Část 3)	43

Úvod

Cílem této práce je porovnání operačního systému Windows XP a Linux Ubuntu. Konkrétně v rychlosti zpracování graficky náročných úloh a vytížení procesoru při jejich provádění. Toho bylo docíleno implementací zátěžového programu, který generuje velké množství grafických objektů, přičemž ukládá statistiku o průběhu vykreslování.

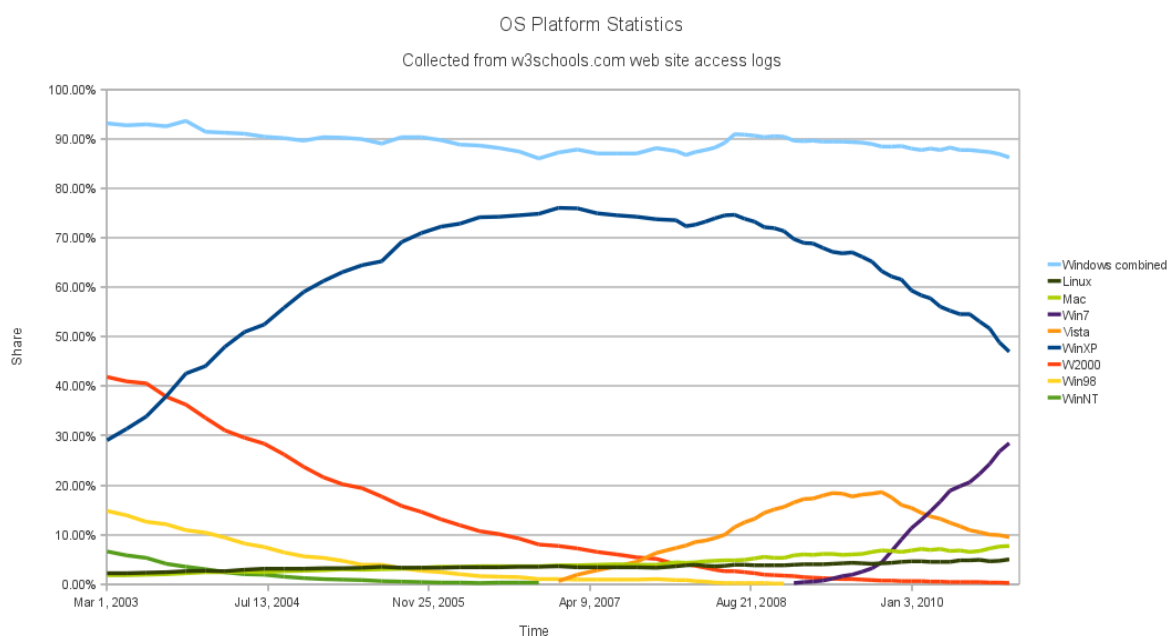
Teoretická část práce je tvořena kapitolami jedna a dva. V první kapitole budou vysvětleny základní pojmy související s prací a to především z oblasti operačních systémů a programování. Druhá kapitola obsahuje seznam a popis použitých programů a technologií.

Následuje praktická část, začínající třetí kapitolou, kde bude představena samotná aplikace, popsány funkce grafického rozhraní a jednotlivé třídy. Další kapitola seznamuje s metodami a způsoby měření. Pátá kapitola zahrnuje výsledky a strukturu jednotlivých měření. Na závěr budou zhodnocena získaná data a práce jako celek.

1 Vymezení základních pojmů

1.1 Operační systém

Jde o základní software, který zprostředkovává rozhraní mezi uživatelem a počítačem. Operační systém má na starost správu prostředků počítače, musí být bezpečný, stabilní a podle určení splňovat řadu dalších požadavků. Základem operačního systému je jeho jádro, tzv. kernel. Jádra dle architektury dělíme na monolitická, hybridní, mikrojádra a nanojádra. Na následujícím obrázku je zobrazeno zastoupení operačních systémů v letech 2003 až 2010. Statistika je získaná z přístupů na serveru w3schools.com. Jde však o zavádějící data, která by měla sloužit pouze pro ilustraci. Pokud bychom chtěli přesnější zobrazení, museli bychom rozdělit operační systémy z hlediska použití na vestavěné, desktopové a serverové. Mezi těmito typy by byly výrazné rozdíly, co se týká zastoupení jednotlivých značek.

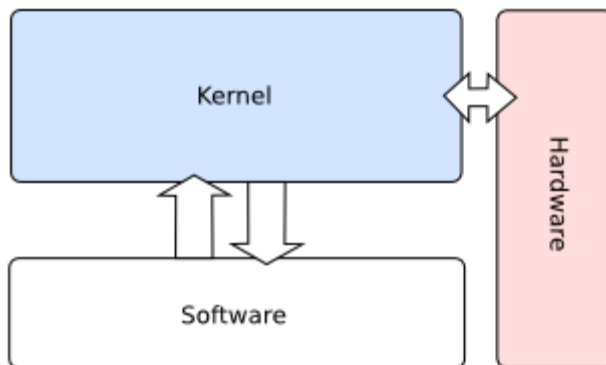


Obrázek 1 - Graf zastoupení operačních systémů v letech 2003 až 2010¹

¹ Zdroj: [1]

1.2 Monolitické jádro

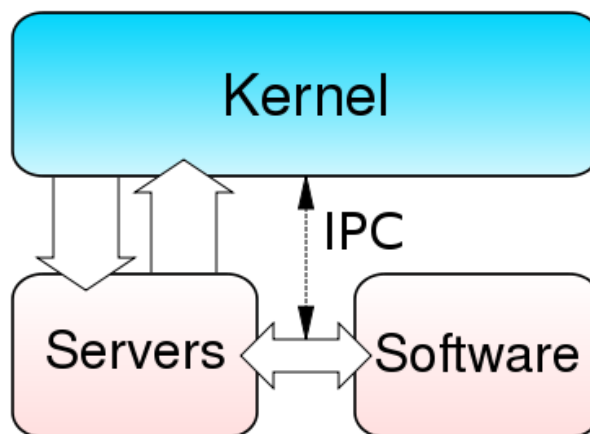
Monolitické jádro, jak už název napovídá, tvoří jeden celek, to znamená, že služby operačního systému i hlavního vlákna jádra se nacházejí ve stejné paměti. Tím se docílilo efektivního přístupu do paměti. Monolitická jádra jsou tedy považována za výkonná, avšak kvůli jednotné paměti citlivá na chyby v subsystémech. Chyba v jediném ovladači může způsobit pád celého systému. Monolitické jádro komunikuje s procesy pomocí systémových volání.



Obrázek 2 - Architektura monolitického jádra²

1.3 Mikrojádro

Tento typ jádra se vyznačuje tím, že obsahuje jen nejnütnější funkce, jako například plánování procesů nebo správu paměti. Ostatní části běží jako uživatelské procesy.

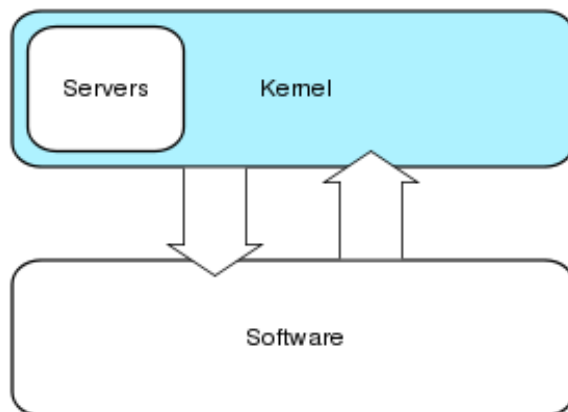


Obrázek 3 - Architektura mikrojádra²

² Zdroj: [2]

1.4 Hybridní jádro

Jde o kompromis mezi monolitickým jádrem a mikrojádro. Z mikrojádra se snaží převzít bezpečnostní výhody, ale zároveň zachovat jednoduchý koncept, který je doménou monolitických jader.



Obrázek 4 - Architektura hybridního jádra³

1.5 Microsoft Windows

Označení Windows patří operačním systémům od firmy Microsoft. První systém Windows vznikl jako nadstavba doposud používaného operačního systému MS-DOS a nesl označení Windows 1.0. Jeho hlavním přínosem byl tzv. multitasking, tedy možnost běhu více programů najednou. Další novinkou bylo grafické rozhraní. Windows 1.0 umožňoval běh většiny programů navržených pro MS-DOS, nicméně ty byly postupem času vytlačovány novými aplikacemi vytvořenými přímo pro Windows (jedním z příkladů je textový editor Microsoft Word). Teprve však Windows 3.0 a 3.1 zaznamenaly komerční úspěch. Následoval systém Windows 95, který kromě modernějšího GUI přinesl podporu TCP/IP protokolu a umožňoval tak přístup k internetu bez nutnosti instalace dalších doplňků, dále přinesl vyšší stabilitu a „Plug and Play“, tedy automatickou detekci a konfiguraci HW. Dalším významným krokem byla podpora preemptivního multitaskingu, operační systém pak neztrácel kontrolu nad počítačem a jakoukoliv úlohu mohl násilně ukončit. Tato novinka přišla až se systémem Windows NT 4.0. Následujícím systémem v řadě byl Windows 2000, v němž byla vylepšena jazyková podpora, přibyly aplikace pro handicapované uživatele a režim spánku. Systém Windows 2000, byl také posledním z řady, který byl určen pro uživatele i pro servery. Další přírůstek do rodiny Windows nesl označení Windows XP, jemuž je v této práci věnována samostatná podkapitola. Následovaly soudobé systémy jako Windows Vista, Windows 7 a Windows 8.

³ Zdroj: [2]

Tabulka 1 - Chronologie systémů Windows (nekompletní)⁴

Chronologie systémů Windows				
Rok vydání	Název	16-bit	32-bit	64-bit
1985	Windows 1.0	x		
1987	Windows 2.0	x		
1988	Windows 2.10	x		
1989	Windows 2.11	x		
1990	Windows 3.0	x		
1992	Windows 3.1	x		
1995	Windows 95	x	x	
1996	Windows NT 4.0		x	
1998	Windows 98	x	x	
2000	Windows 2000		x	
2001	Windows XP		x	x
2003	Windows Server 2003		x	x
2006	Windows Vista		x	x
2008	Windows Server 2008		x	x
2009	Windows 7		x	x
2012	Windows Server 2012			x
2012	Windows 8		x	x

1.6 GNU/Linux

V roce 1991 se rozhodl student informatiky v Helsinkách Linus Torvalds vytvořit operační systém s vlastnostmi a možnostmi Unixu, který bude zároveň dostupný pro osobní počítače. Tento systém měl na rozdíl, od tou dobou převládajícího MS-DOS, podporovat multitasking a více uživatelů. „Pokud bychom měli Linux hodnotit z hlediska současné architektury, tak je možné říci, že jde o modulární jádro, které je ale v zásadě monolitické. Od samého počátku byl podporován multitasking, z čehož plyne podpora preemptivního plánování procesů, což má pozitivní dopad na rychlost, bezpečnost i stabilitu. Jedinými oficiálně podporovanými jazyky pro programování jsou C a Asembler. V současné době se hodně používá především u serverů, kde má zcela dominantní postavení, a na mobilních telefonech, kde jej reprezentuje především Android (ale třeba také MeeGo, Maemo, LiMo a řada dalších méně známých). Na osobních počítačích je jeho podíl v řádu jednotek procent a je šířen pomocí distribucí.“⁵

1.7 Virtuální stroj

Tento pojem označuje softwarovou simulaci počítače. Tato simulace je oddělena od platformy, která ji provádí, je však omezena prostředky, které jsou jí přiděleny. Desktopová virtualizace je z hlediska využití HW neúspěšná, protože prostředky jsou

⁴ Zdroj: [3]

⁵ Zdroj: [8]

spotřebovávají nejen virtuálním strojem, ale i operačním systémem, na kterém je virtualizace prováděna. Uplatnění nalezne zejména v testování nebo vzdělávání. V současnosti patří mezi nejznámější virtualizační nástroje Bochs, QEMU, VMware a VirtualBox.

1.8 Open source

Tento výraz se v informatice užívá k označení takového programu, jehož licence umožňuje prohlížet nebo dokonce upravovat zdrojový kód.

1.9 Knihovna

V informatice pojmem knihovna představuje soubor funkcí, procedur, datových typů, zdrojů případně objektů. Knihovna poskytuje jednotlivé části skrze rozhraní API. Čím více knihoven je pro daný jazyk k dispozici, tím snadnější je vývoj aplikací. Programátor by si měl nejdříve ověřit, zda funkce, kterou má v úmyslu zkonstruovat, není již implementována v některé knihovně, může si tak výrazně usnadnit práci. Proto je důležité, aby měla knihovna podrobnou a přesnou dokumentaci. Knihovny dělíme na statické, které jsou součástí přeloženého programu a dynamické, které jsou při překladu přilinkovány do výsledného spustitelného souboru.

1.10 API

Application programming interface (zkr. API) je rozhraní, které definuje komunikaci knihovny s vnějším světem. Nemusí však jít pouze o knihovnu, termín API se používá i ve spojitosti s jádrem operačního systému nebo jiným programem. Zjednodušeně se dá říci, že API umožňuje programátorovi využívat knihovnických funkcí, informuje ho o jejich názvu, návratové hodnotě, argumentech, účelu, případně příkladu použití. Mezi známá standardizovaná rozhraní patří například OpenGL a DirectX.

1.11 Procesy

Proces je instance běžícího programu nacházející se v operační paměti, přičemž aplikace může běžet jako více procesů. O správu procesů se stará operační systém, který zajišťuje jejich oddělený běh, přidělování prostředků a správu uživatele.

1.12 Vlákna

Vlákno představuje tzv. „odlehčený proces“. V porovnání s procesem má menší režii při přepínání kontextu či tvorbě nového vlákna. Toho je docíleno tím, že vlákna, na rozdíl od procesů, sdílejí paměť, to s sebou však nese nutnost synchronizace. Pokud by byla vlákna špatně synchronizována, mohlo by více vláken současně pracovat se stejnými daty, což by mohlo vést k nekonzistenci těchto dat. V rámci jednoho procesu může běžet více vláken.

2 Použité programy a technologie

2.1 Oracle VM VirtualBox

Jedná se o multiplatformní virtualizační nástroj, který v současnosti vlastní firma Oracle. Původně byl však vyvíjen firmou Innotek GmbH. Je distribuován pro systémy Windows, Linux/Unix, MacOS. Mezi jeho klady patří vícejazyčná podpora, snadné ovládání, open source licence a množství podporovaných operačních systémů, které je schopen virtualizovat.

2.2 Java Virtual Machine

Jde o virtuální stroj, jehož úkolem je spouštění skriptů a programů, napsaných v jazyce Java. Díky JVM je aplikace vytvořená v jazyce Java spustitelná na všech platformách, které JVM provozují. Kromě jazyku Java dokáže JVM spustit aplikace vyvinuté v jazycích jako je Groovy, Scala, Clojure.

2.3 NetBeans

NetBeans je open source vývojové prostředí primárně zaměřené na tvorbu v jazyce Java. Dále podporuje jazyky PHP, HTML 5, C/C++, Ruby. Projekt má velkou vývojářskou i uživatelskou základnu a jeho hlavním sponzorem je firma SunMicrosystems. Prostředí je naprogramováno v jazyce Java, což znamená, že je spustitelné na operačních systémech Linux, Windows, Mac OS. Jednou ze silných stránek tohoto prostředí je snadný vývoj GUI s využitím knihovny Swing.

2.4 Java

Objektově orientovaný programovací jazyk Java je od poloviny 90. let vyvíjen firmou Sun Microsystems. Jde o jeden z nejpoužívanějších programovacích jazyků na světě. Jedná se o takzvaný interpretovaný jazyk, což znamená, že místo strojového kódu je vytvářen mezikód, který je následně zpracován interpretem, tedy JVM. To s sebou přináší jisté zpomalení při překladu programu, to se však redukuje použitím metod HotSpot nebo JIT, které překládají neefektivní nebo často používané části programu do strojového kódu. Skutečnost, že je jazyk Java interpretovaný, přináší nespornou výhodu v tom, že je nezávislý na platformě. K spuštění Java aplikace je však nutné mít na daném stroji spuštěnu odpovídající JVM distribuci. Java je rozdělena na několik platform, které jsou určeny pro konkrétní účely:

- **Java SE** (Java Standard Edition) je určena pro desktopové aplikace. Mimo jiné obsahuje API knihovny AWT a Swing.
- **Java EE** (Java Enterprise Edition) platforma určená pro vývoj podnikových aplikací a informačních systémů, jde o nadstavbu Java SE.

- **Java ME** (Java Mobile Edition) byla vytvořena za účelem vývoje aplikací pro malá zařízení, jako jsou mobilní telefony, PDA, tablety.
- **JavaCard** se zaměřuje na vývoj a provoz aplikací na paměťových kartách.

Jazyk Java vychází z jazyka C a C++. Cílem bylo zjednodušit syntaxi a oprostít se od problematických konstrukcí.

„Existuje široké spektrum programovacích jazyků, z nichž jsou mnohé velmi kvalitní. Tento přehled Vám nabídne několik argumentů, proč se začít učit zrovna Javu.

- Java je syntakticky jednoduchá.
- Java se sama stará o uvolňování nepotřebné paměti.
- Javovské programy jsou přenositelné (tj. stejný program lze bez úprav pustit na libovolném operačním systému).
- Java je použitelná pro projekty každého rozsahu.
- Java je vhodná pro začátečníky – chyby v programech se dají (obvykle) snadno vyhledat.
- Java je použitelná na všech typech zařízení od vestavných po velké servery.
- Java je výkonná, zejména v porovnání s interpretovanými jazyky (PHP, Ruby).
- Java neumožňuje udělat některé programátorské zločiny proti lidskosti.
- Java je široce používaná.
- V případě nesnáží není problém nalézt někoho, kdo Vám poradí.
- Její znalost je výhodou na trhu práce.
- Vždy se můžete inspirovat kódem jednoho z tisíců otevřených projektů v Javě.
- Pro vývoj v Javě existují velmi kvalitní vývojová prostředí a nástroje usnadňující práci.⁶

Vývojová prostředí pro jazyk Java: JDeveloper, Eclipse, NetBeans, IntelliJ IDEA, BlueJ.

2.5 Java Graphics

Abstraktní třída Graphics poskytuje grafické operace s komponentami knihoven AWT a Swing.

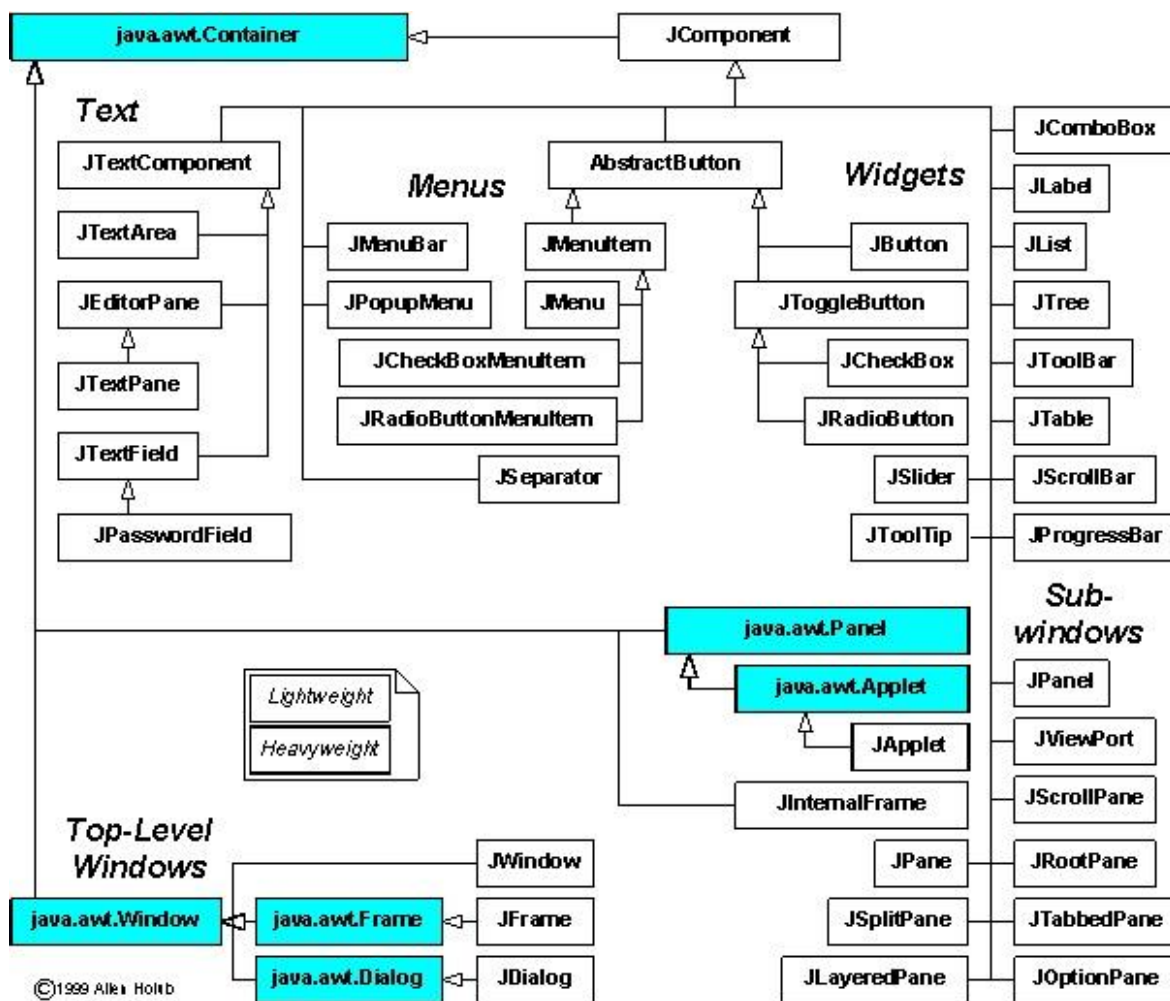
2.6 Java Graphics2D

Tato třída je potomkem třídy Graphics a poskytuje sofistikovanější podporu pro práci s geometrií, textem, souřadnicemi nebo správou barev.

⁶ Zdroj: [4]

2.7 Java Swing

Knihovna Swing je určena pro tvorbu grafického rozhraní na platformě Java. Jejím předchůdcem byla knihovna AWT, jejíž nedostatky (především závislost na platformě) odstartovaly vývoj knihovny Swing. Ta je součástí Java SE od verze 1.2. a umožňuje vytvářet klasické prvky pro tvorbu GUI jako jsou: okna, rámečky, seznamy, tlačítka a další.



Obrázek 5 - Hierarchie tříd knihovny Swing⁷

2.8 Windows XP

Jedná se o operační systém z rodiny Windows, který jako první nebyl založen na MS-DOS, ale na technologiích Windows NT a Windows 2000. Windows XP využívá hybridního jádra a od svého uvedení roku 2001 se dočkal tří aktualizací balíčků (Service Pack 1 až 3) a rozšíření o 64bitovou verzi. Šlo o jeden z neúspěšnějších operačních systémů. Přišel se zcela přepracovaným grafickým prostředím, změnou menu, ovládacích panelů a řadou dalších funkcí. Díky tomu, že již nestavěl na DOS, byl mnohem rychlejší, stabilnější a umožňoval běh většího množství aplikací současně. Podporoval také DirectX, takže byl

⁷ Zdroj: [5]

vhodný pro hraní her či práci s multimédií. Obsahoval integrovaný IE, což vedlo postupně k řadě soudních sporů, které konstatovaly, že Microsoft zneužívá dominantní postavení na trhu, aby prosazoval své další aplikace.

Tabulka 2 - Systémové požadavky systému Windows XP⁸

Systémové požadavky Windows XP		
	Minimum	Doporučené
Procesor	233 MHz	300 MHz nebo vyšší
Operační paměť	64 MB RAM	128 MB RAM nebo více
Grafická karta a monitor	Super VGA (800 × 600)	Super VGA (800 × 600) a vyšší
Volné místo na HDD	1,5 GB	1,5 GB a více
Mechaniky	CD-ROM	CD-ROM nebo DVD-ROM

2.9 Ubuntu 12.04 Precise Pangolin – LTS

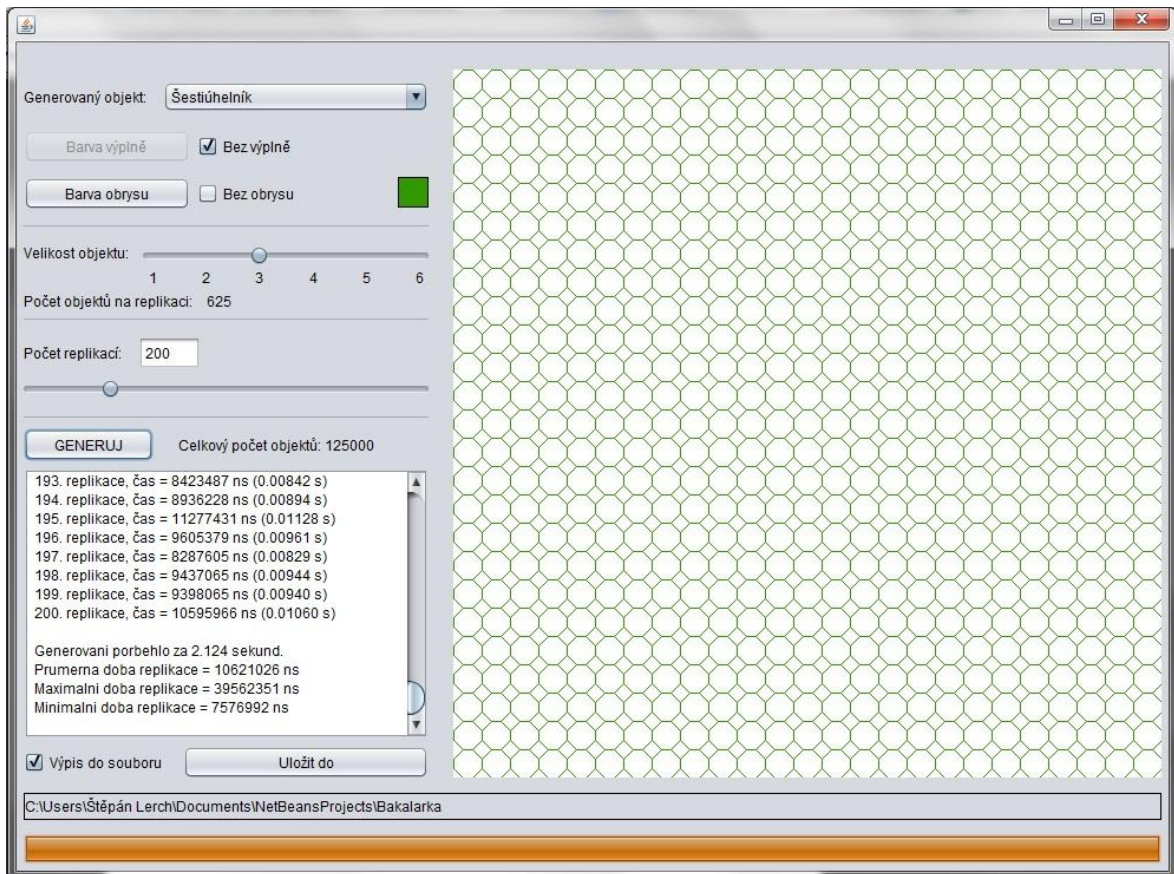
Ubuntu je linuxová distribuce, vyvíjená od roku 2004, za jejíž zrodem stojí Mark Shuttleworth. Konkrétně verze 12.04 byla vydána v dubnu roku 2012 a je tradičně pojmenována podle zvířete. Zkratka LTS naznačuje dlouhodobou podporu, v tomto případě 5 let. Ubuntu používá monolitického jádra a v případě verze 12.04 grafické rozhraní GNOME.

Podle oficiální dokumentace Ubuntu je minimální doporučená konfigurace počítače s procesorem 1 GHz Pentium 4 s 512 MB paměti RAM a 5 GB volného místa na harddisku.

3 Vzhled a možnosti aplikace

Cílem aplikace bylo zatížení systému vykreslováním grafických objektů a ukládání časových údajů o rychlosti vykreslení. Pro tyto účely bylo navrženo jednoduché grafické rozhraní, umožňující uživateli volbu vykreslovaného objektu a jeho parametrů, následně volbu počtu generovaných objektů a počet replikací. Aplikace umožňuje na jednu replikaci vykreslit 9 až 10 000 objektů, přičemž maximální počet replikací byl stanoven na 1 000. Celkový počet objektů, vykreslených na jedno generování, je v rozsahu od 9 do 10 000 000. Rozhraní nadále umožňuje specifikaci souboru, do kterého budou data ukládána a informuje uživatele o průběhu generování formou textových zpráv i samotným vykreslováním objektů na „plátno“.

⁸ Zdroj: [9]

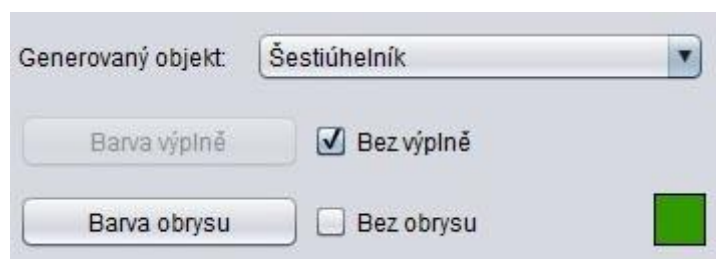


Obrázek 6 - Okno aplikace

3.1 Jednotlivé funkce aplikace

3.1.1 Vlastnosti objektu

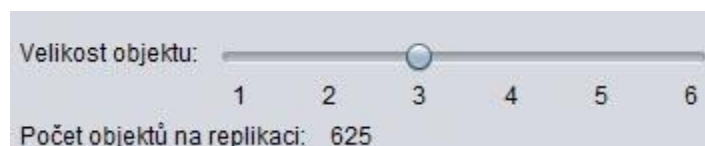
Zde si může uživatel vybrat z několika objektů, které chce generovat, následně může nastavit barvu výplně a obrysu, případně zaškrtnout jednu z možností „bez výplně“ nebo „bez obrysu“.



Obrázek 7 - Vlastnosti objektu

3.1.2 Velikost objektu

V této oblasti se nastavuje velikost generovaného objektu, od které je odvozen počet vykreslených objektů na replikaci. Velikost panelu, na něž budou objekty vykreslovány, byla stanovena na 600 x 600 pixelů. Pokud je nastavena velikost objektu na 1 má každý objekt prostor o velikosti 6x6 pixelů a jejich celkový počet na replikaci je 10 000. Je-li zvolena maximální velikost objektu, tedy 6, má každý objekt prostor 200 x 200 pixelů, což umožňuje vykreslit maximálně 9 objektů v rámci jedné replikace.



Obrázek 8 - Velikost objektu

3.1.3 Počet replikací

Jak již bylo zmíněno, program generuje uživatelem stanovený počet objektů. Další důležitou funkcí je ukládání časové hodnoty, která představuje rychlost vykreslení všech těchto objektů. Pro vyvození závěrů je však potřeba více než jedna hodnota, proto je zde možnost opakování pokusu. Počet replikací může být uživatelem nastaven v rozsahu od 1 do 1 000.



Obrázek 9 - Počet replikací

3.1.4 Generování

Tlačítkem „GENERUJ“ zahajujeme samotné vykreslování, pro přehlednost je zde umístěn popis informující o celkovém počtu objektů. Další informace získá uživatel z panelu, do

kterého jsou zapisovány údaje o jednotlivých replikacích. Po skončení generování je vypsána statistika zahrnující hodnotu nejrychlejší replikace, nejpomalejší replikace a průměrný čas jedné replikace.



Obrázek 10 - Generování

3.1.5 Ukládání do souboru

Zápis do souboru není pro spuštění vykreslování povinný, je ho možné vypnout za pomoci „checkboxu“, případně specifikovat cestu k souboru tlačítkem „Uložit do“, které vyvolá „jOpenFileDialog“ umožňující zvolení souborů s koncovkou csv. O aktuálně zvolené cestě informuje spodní řádek (JLabel).



Obrázek 11 - Ukládání dat

3.1.6 Průběh generování

Každá replikace je vykreslena na „plátno“, tedy hlavní panel. Přitom každá skončená replikace inkrementuje hodnotu spodního „JProgressBaru“, ten informuje o poměru proběhlých a zbývajících replikací.



Obrázek 12 - Průběh generování

3.2 Vlastní třídy a metody

V této podkapitole budou popsány stěžejní třídy aplikace a některé z jejich metod.

3.2.1 Demo

Nejvýše postavenou třídou v aplikaci je právě metoda Demo. Demo dědí od třídy JFrame a poskytuje grafické rozhraní. Jsou zde umístěny a obsluhovány všechny použité komponenty knihovny Swing.

3.2.2 GraphicsObject

Tato třída představuje obecný grafický objekt, který bude vykreslován, dědí od ní třídy zastupující konkrétní grafický objekt, například třída Hexagon, Rectangle. Atributy třídy určují základní vlastnosti objektu. Nejdůležitější metodou je zde metoda drawObject(Graphics2D g2d, int x, int y, int size). Tuto metodu přetěžují všechny odvozené třídy, které specifikují způsob vykreslování pro daný objekt.

3.2.3 GraphicsManager

Třída GraphicsManager je zodpovědná za samotné vykreslování objektů na „plátno“. Důležitá je především metoda drawObjects(int pocetReplikaci, List<Long> casy, JProgressBar progress), tato metoda spouští pro každou replikaci samostatné vlákno a zároveň „uzamyká“ hlavní vlákno aplikace tak, aby nebylo možné běh replikace narušit. Časy jednotlivých replikací ukládá do seznamu a průběžně informuje uživatele o stavu generování.

3.2.4 ObjectEnum

Jedná se o výčtový typ, který zastupuje všechny dosud implementované grafické objekty.

4 Metody měření

Samotné testování proběhlo v prostředí virtualizačního nástroje Oracle VM VirtualBox verze 4.1.18. Pro oba testované systémy bylo nutné nastavit stejné parametry pro virtualizaci.

Pro dané zkoumání jsou důležité především tyto hodnoty:

- operační paměť = 1024 MB
- omezení vytížení procesorů = 0 až 100 %
- video paměť = 128 MB

4.1 Záznam vytížení systémových prostředků

4.1.1 Process Explorer

Pro zachycení hodnot využití procesoru a virtuální paměti (na platformě Windows) byly využity grafy generované nástrojem Process Explorer 15.2.

4.1.2 Linux Process Monitor

Jedná se o obdobu nástroje ProcessExplorer na linuxové platformě.

4.2 Způsob měření

Pro měření byla použita metoda `nanoTime()` třídy `System`. Tato metoda vrací hodnotu typu `Long`, která reprezentuje čas systémového časovače v nanosekundách. Níže je uvedena část kódu, kde docházelo k měření.

```
startTime = System.nanoTime();

for(int i = 0; i < pocet; i++){
    for(int j = 0; j < pocet; j++){
        graphicsObject.drawObject(g2d, poziceX,poziceY,konecnaVelikost);
        poziceX += konecnaVelikos;
    }
    poziceY += konecnaVelikos;
    poziceX = 0;
}

finalTime = System.nanoTime() - startTime;
```

4.3 Parametry spuštěných měření

V následující kapitole budou zobrazena data z jednotlivých měření. Ta proběhla na operačních systémech Microsoft Windows XP (Service Pack 3) a Ubuntu 12.04 Precise Pangolin – LTS. Výpisy procesů, které byly získány při spuštění aplikace, jsou k dispozici v přílohách A a B. V průběhu měření 1 a 2 bylo generováno 10 000 objektů ve dvou stech replikacích. Měření 3 se od prvního liší v počtu generovaných objektů na replikaci, který byl stanoven na 625. Záměrně byly generovány stejné objekty, aby bylo možné adekvátně porovnat získaná data s ohledem na počet, nikoliv typ objektů.

5 Získaná data

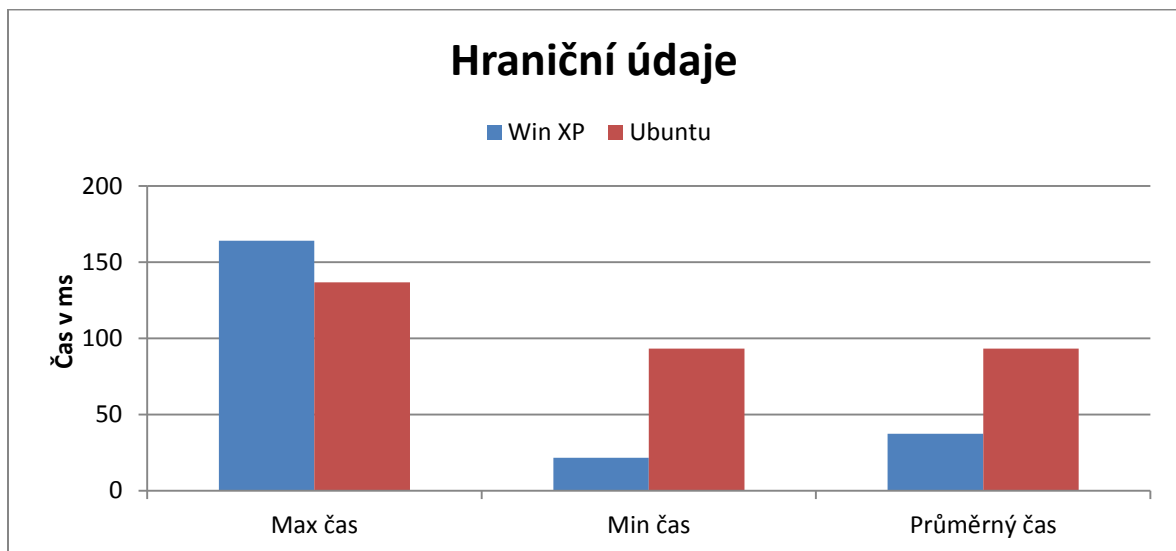
5.1 Struktura jednotlivých měření

Každé z uskutečněných měření bude rozděleno do několika částí. V první bude seznam parametrů daného měření, následuje sloupcový graf zobrazující průměrnou, minimální a maximální dobu replikace pro obě platformy. Dalším prvkem bude křivkový graf, ve

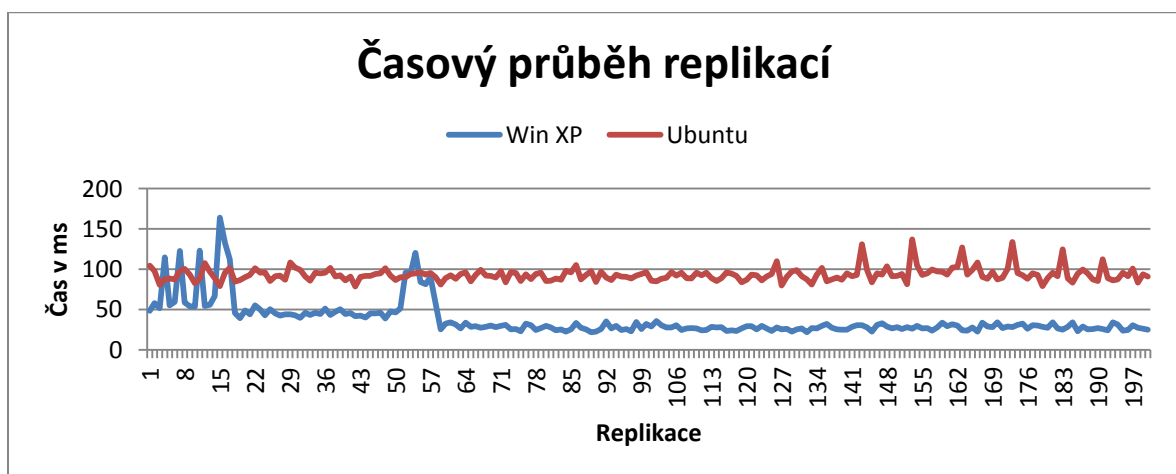
kterém budou zobrazeny časové hodnoty jednotlivých replikací. V poslední části měření nalezneme grafy vytížení procesoru. Červená barva zde indikuje procento využití procesoru v kernel-módu, zelená pak využití procesoru v user-módu.

5.2 Měření 1

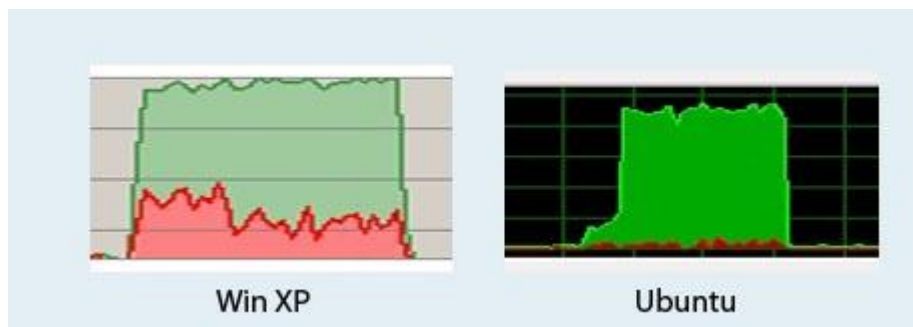
- Celkový počet generovaných objektů: 2 000 000
- Objektů na replikaci: 10 000
- Počet replikací: 200
- Typ generovaného objektu: Kruh
- Obrys/Výplň: ano/ano
- Metody pro vykreslení: fillOval(), drawOval()



Obrázek 13 - Hraniční údaje (Měření 1)



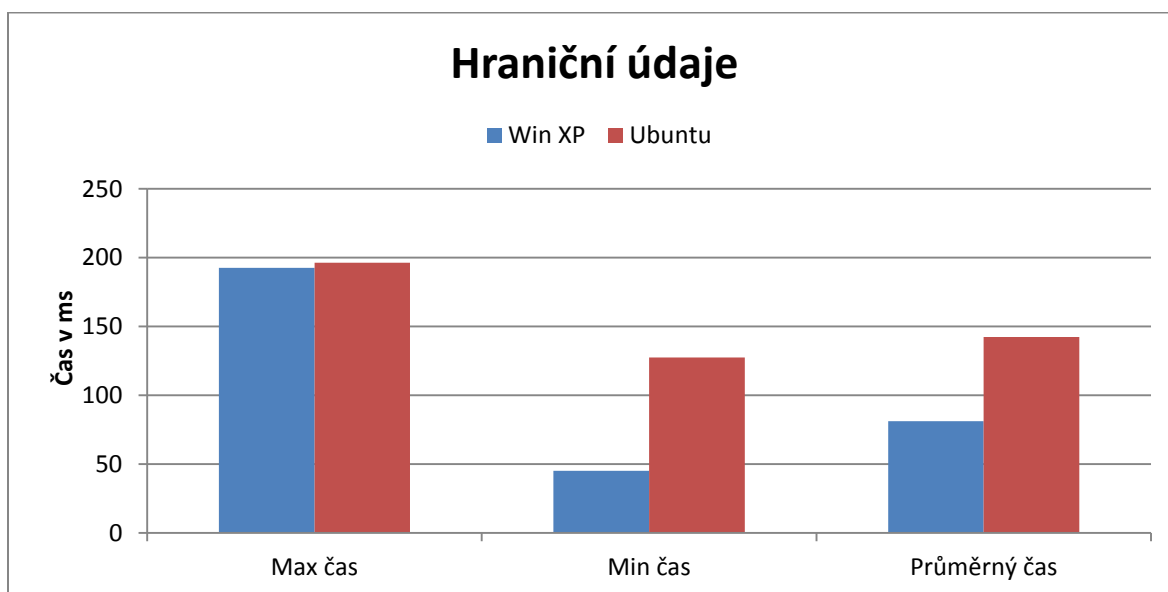
Obrázek 14 - Průběh replikací (Měření 1)



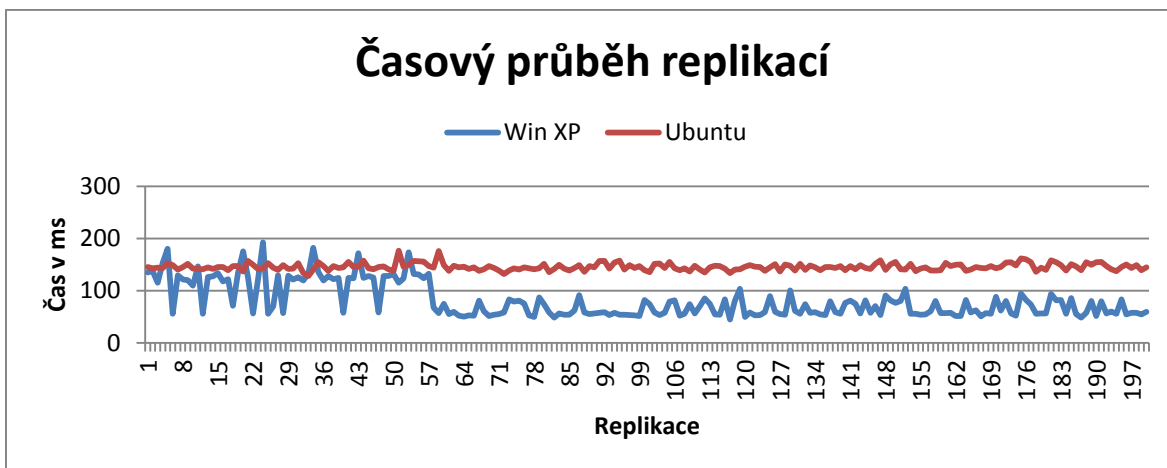
Obrázek 15 - Vytížení procesoru (Měření 1)

5.3 Měření 2

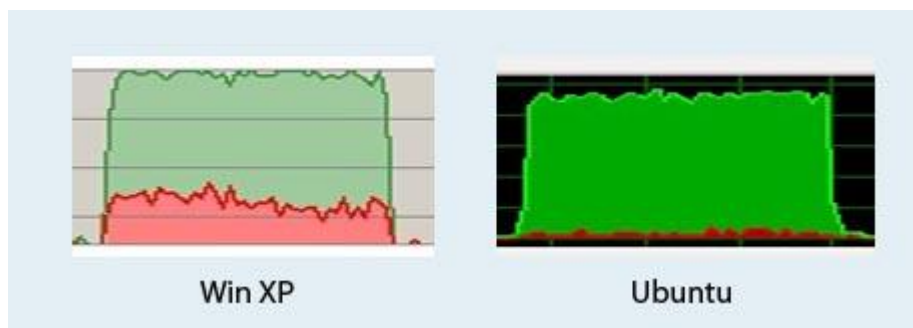
- Celkový počet generovaných objektů: 2 000 000
- Objektů na replikaci: 10 000
- Počet replikací: 200
- Typ generovaného objektu: Šestiúhelník
- Obrys/Výplň: ano/ano
- Metody pro vykreslení: fillPolygon(), fillPolygon()



Obrázek 16 - Hraniční údaje (Měření 2)



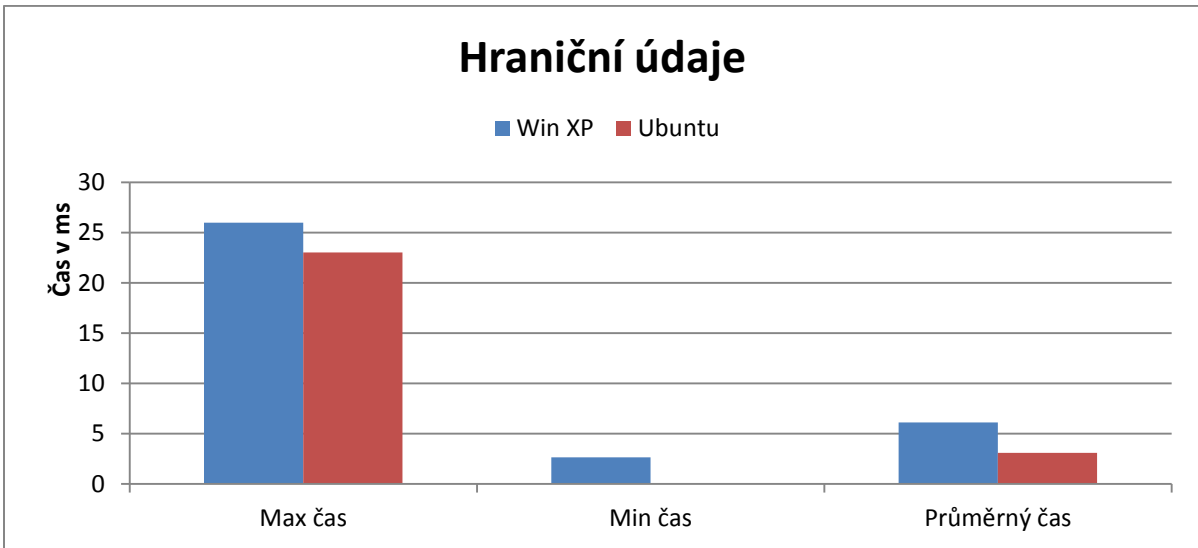
Obrázek 17 - Průběh replikací (Měření 2)



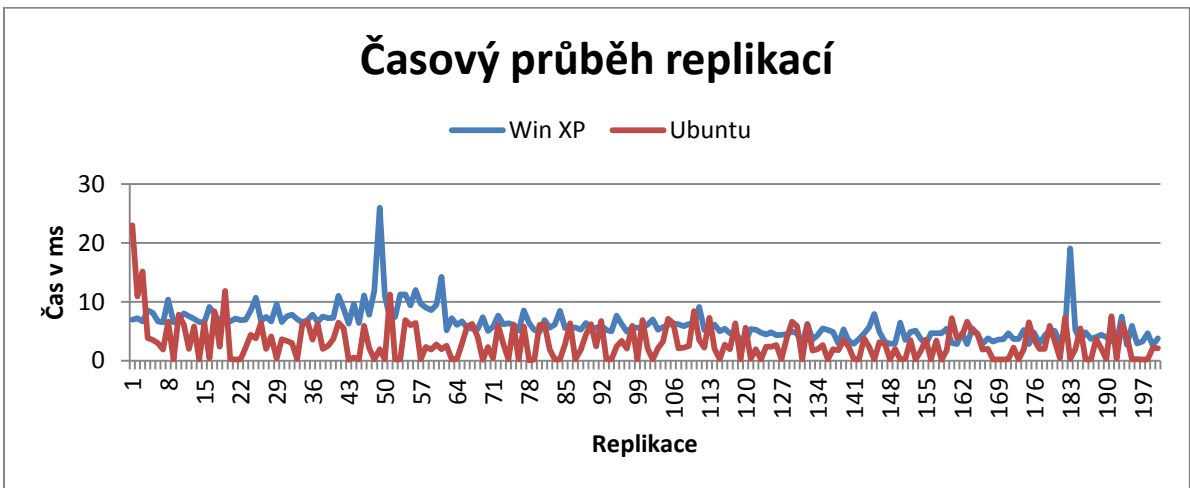
Obrázek 18 - Vytížení procesoru (Měření 2)

5.4 Měření 3

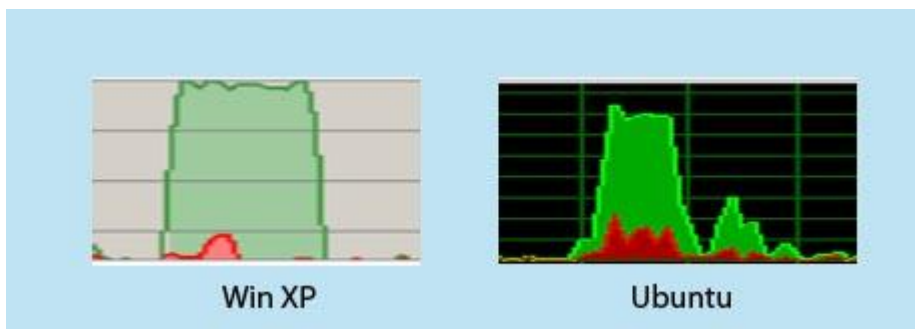
- Celkový počet generovaných objektů: 125 000
- Objektů na replikaci: 625
- Počet replikací: 200
- Typ generovaného objektu: Kruh
- Obrys/Výplň: ano/ano
- Metody pro vykreslení: fillOval(), drawOval()



Obrázek 19- Hraniční údaje (Měření 3)



Obrázek 20 - Průběh replikací (Měření 3)



Obrázek 19

Závěr

V první části práce byly přiblíženy důležité pojmy týkající se problematiky. Následně byly zvoleny dané technologie. Předně programovací jazyk Java a to především kvůli „nezávislosti“ na platformě. Dále virtualizační nástroj VirtualBox, u něhož byla rozhodující dostupnost a snadné ovládání. Jako vývojové prostředí byl zvolen program NetBeans, který obsahuje účinný a přehledný editor pro tvorbu grafického rozhraní, které je důležitým aspektem aplikace. Pro vykreslování objektů byla použita třída Graphics z nativní knihovny Java, která pro účely dané práce plně postačila.

Po instalaci a konfiguraci použitých programů započal vývoj samotné aplikace, po jehož skončení mohlo být spuštěno požadované měření. Data získaná z těchto měření ukázala, že v případě vykreslování velkého počtu objektů je Windows celkově přibližně o polovinu rychlejší než Ubuntu. Nicméně rozptyl hodnot je u Windows daleko vyšší než u systému Ubuntu, který má daleko stálější křivku průběhu, což je patrné z obrázků č. 14 a 17. K zajímavému zjištění došlo při posledním z pokusů, kde je generováno výrazně menší množství objektů. Oba průběhy jsou velmi nestálé a výsledky ukazují, že celkově je v tomto případě rychlejší operační systém Ubuntu a to cca o 50%. Z grafů vytížení procesoru vyplývá, že na platformě Windows byl procesor v průběhu všech pokusů zatížen k hranici 100%, kdežto u platformy Ubuntu bylo zatížení přibližně o 10% nižší.

Ve zkoumané oblasti, tedy zobrazení grafických objektů, se ukázal systém Windows XP konkurenceschopným ba dokonce v prvních měřeních výkonnějším, než o několik let mladší systém Ubuntu 12.04. Šlo však o pokus na virtuálním jednoprocessorovém stroji, naměřené hodnoty se tedy mohou značně rozcházet s hodnotami, které by byly získány bez využití virtualizace a na vícejádrovém stroji.

Literatura

- [1] **FÖTSCH, Michael.** Operating System Usage Statistics. In: *Realmike* [online]. 2010 [cit. 2013-05-09]. Dostupné z: <http://realmike.org/blog/2010/12/11/operating-system-usage-statistics/>
- [2] **Jádro (informatika).** In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-09]. Dostupné z: http://cs.wikipedia.org/wiki/J%C3%A1dro_%28informatika%29
- [3] **Timeline of Microsoft Windows.** In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-10]. Dostupné z: http://en.wikipedia.org/wiki/Timeline_of_operating_systems
- [4] **MIČKA, Pavel.** Java pro začátečníky. In: *Http://www.algoritmy.net* [online]. 2008-2013 [cit. 2013-05-09]. Dostupné z: <http://www.algoritmy.net/article/21340/Uvod-1>
- [5] **HOLUB, Allen.** Java Swing Classes. *Www.holub.com* [online]. 2011 [cit. 2013-05-10]. Dostupné z: <http://www.holub.com/goodies/java.swing.html>
- [6] **ECKEL, Bruce.** *Thinking in Java*. Massachusetts: Prentice-Hall, 2006. ISBN 0-13-187248-6.
- [7] **ČERNÁ, Monika a Michal ČERNÝ.** Historie operačních systémů: Od Windows NT po Vista. In: *Http://clanky.rvp.cz* [online]. 2012 [cit. 2013-05-09]. Dostupné z: <http://clanky.rvp.cz/clanek/o/g/15445/HISTORIE-OPERACNICH-SYSTEMU-OD-WINDOWS-NT-PO-VISTA.html/>
- [8] **ČERNÁ, Monika a Michal ČERNÝ.** Historie operačních systémů: rodina Linux. *Clanky.rvp.cz* [online]. 2012 [cit. 2013-05-10]. Dostupné z: <http://clanky.rvp.cz/clanek/o/g/15443/HISTORIE-OPERACNICH-SYSTEMU-RODINA-LINUX.html/>
- [9] **A history of Windows:** Highlights from the first 25 years. In: *Http://clanky.rvp.cz* [online]. 2010 [cit. 2013-05-09]. Dostupné z: <http://windows.microsoft.com/en-in/windows/history>
- [10] **Windows XP.** In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-09]. Dostupné z: http://en.wikipedia.org/wiki/Windows_XP#System_requirements
- [11] **Ubuntu 12.04: Poznámky k vydání.** In: *Http://www.ubuntu.cz* [online]. 2012 [cit. 2013-05-09]. Dostupné z: <http://www.ubuntu.cz/ziskejte/poznamkykvydani/1204poznámky>

- [12] **Class Graphics**. In: *Http://docs.oracle.com* [online]. 2011 [cit. 2013-05-09]. Dostupné z: <http://docs.oracle.com/javase/6/docs/api/java/awt/Graphics.html>
- [13] **NetBeans**. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-09]. Dostupné z: <http://en.wikipedia.org/wiki/NetBeans>
- [14] **VENNERS, Bill**. The Java Virtual Machine. In: *Www.artima.com* [online]. 1996-2013 [cit. 2013-05-09]. Dostupné z: <http://www.artima.com/insidejvm/ed2/jvm.html>
- [15] **LEAHY, Paul**. What Is Java?. *Http://java.about.com* [online]. 2013 [cit. 2013-05-09]. Dostupné z: <http://java.about.com/od/gettingstarted/a/whatisjava.htm>
- [16] **Tanenbaum, A.**: *Modern Operating Systems*. 3. vydání. Prentice-Hall USA, 2007. ISBN 978-0-13-600663-3.

Příloha A – Výpis procesů systému Windows XP

Tabulka 3 - Výpis procesů Windows XP

Název procesu	PID	Název relace	Využití paměti
System Idle Process	0	Console	28 kB
System	4	Console	236 kB
smss.exe	368	Console	420 kB
csrss.exe	584	Console	4276 kB
winlogon.exe	608	Console	3896 kB
services.exe	652	Console	4448 kB
lsass.exe	664	Console	7908 kB
VBoxService.exe	820	Console	3264 kB
svchost.exe	864	Console	4980 kB
svchost.exe	952	Console	4320 kB
svchost.exe	1044	Console	21328 kB
svchost.exe	1092	Console	3592 kB
svchost.exe	1160	Console	3972 kB
spoolsv.exe	1524	Console	5208 kB
explorer.exe	1668	Console	19120 kB
vmusrvc.exe	1724	Console	2132 kB
VBoxTray.exe	1736	Console	2760 kB
NamosCtl.exe	1776	Console	6420 kB
jusched.exe	1792	Console	2320 kB
ctfmon.exe	1816	Console	3184 kB
szninstall.exe	1832	Console	2884 kB
svchost.exe	192	Console	3468 kB
msdtc.exe	244	Console	5204 kB
javaw.exe	304	Console	34292 kB
CALSnmp.exe	536	Console	12564 kB
hasplms.exe	888	Console	14140 kB
inetinfo.exe	996	Console	8028 kB
jqc.exe	1016	Console	1440 kB
sqlservr.exe	1040	Console	53752 kB
ReportingServicesService.exe	1196	Console	29864 kB
sntlkeyssrvr.exe	1340	Console	2632 kB
spnsrvnt.exe	1856	Console	3384 kB
sntlstrtsrvr.exe	1896	Console	4200 kB
sqlwriter.exe	1848	Console	3624 kB
mqsvc.exe	2068	Console	6016 kB
mqtgsvc.exe	2508	Console	4024 kB
SQLAGENT90.EXE	2612	Console	1460 kB
wmiiprvse.exe	2872	Console	6716 kB
procexp.exe	3204	Console	20532 kB
wmiapsrv.exe	3240	Console	4612 kB
wmiiprvse.exe	3260	Console	5012 kB
cmd.exe	1940	Console	2748 kB
wuauclt.exe	2332	Console	7012 kB
tasklist.exe	2692	Console	4440 kB

Příloha B – Výpis procesů systému Ubuntu 12.04

Tabulka 4 - Výpis procesů Ubuntu (Část 1)

PID	CPU	MEM	TIME	COMMAND
1	0.0	0.1	0:00	/sbin/init
2	0.0	0.0	0:00	[kthreadd]
3	0.0	0.0	0:00	[ksoftirqd/0]
5	0.0	0.0	0:00	[kworker/u:0]
6	0.0	0.0	0:00	[migration/0]
7	0.0	0.0	0:00	[watchdog/0]
8	0.0	0.0	0:00	[cpuset]
9	0.0	0.0	0:00	[khelper]
10	0.0	0.0	0:00	[kdevtmpfs]
11	0.0	0.0	0:00	[netns]
12	0.0	0.0	0:00	[sync_supers]
13	0.0	0.0	0:00	[bdi-default]
14	0.0	0.0	0:00	[kintegrityd]
15	0.0	0.0	0:00	[kblockd]
16	0.0	0.0	0:00	[ata_sff]
17	0.0	0.0	0:00	[khubd]
18	0.0	0.0	0:00	[md]
21	0.0	0.0	0:00	[khungtaskd]
22	0.0	0.0	0:00	[kswapd0]
23	0.0	0.0	0:00	[ksmd]
24	0.0	0.0	0:00	[khugepaged]
25	0.0	0.0	0:00	[fsnotify_mark]
26	0.0	0.0	0:00	[ecryptfs-kthrea]
27	0.0	0.0	0:00	[crypto]
35	0.0	0.0	0:00	[kthrotld]
37	0.0	0.0	0:00	[scsi_eh_0]
38	0.0	0.0	0:00	[scsi_eh_1]
40	0.0	0.0	0:00	[scsi_eh_2]
41	0.0	0.0	0:00	[kworker/u:4]
43	0.0	0.0	0:00	[kworker/0:2]
62	0.0	0.0	0:00	[devfreq_wq]
184	0.0	0.0	0:00	[jbd2/sda1-8]
185	0.0	0.0	0:00	[ext4-dio-unwrit]
204	0.0	0.0	0:00	[flush-8:0]
275	0.0	0.0	0:00	upstart-udev-bridge
279	0.0	0.1	0:00	/sbin/udev
323	0.0	0.0	0:00	[iprt]
341	0.0	0.0	0:00	[kpsmoused]
383	0.0	0.1	0:00	dbus-daemon
399	0.0	0.2	0:00	/usr/sbin/modem-manager
406	0.0	0.1	0:00	/usr/sbin/bluetoothd

Tabulka 5 - Výpis procesů Ubuntu (Část 2)

PID	CPU	MEM	TIME	COMMAND
450	0.0	0.0	0:00	avahi-daemon:
453	0.0	0.0	0:00	[krfcommd]
457	0.0	0.3	0:00	/usr/sbin/cupsd
462	0.0	0.4	0:00	/usr/lib/policykit-1/polkitd
495	0.0	0.1	0:00	/sbin/udevd
496	0.0	0.1	0:00	/sbin/udevd
541	0.0	0.1	0:00	/sbin/dhclient
718	0.0	0.0	0:00	upstart-socket-bridge
748	0.0	0.0	0:00	/sbin/getty
755	0.0	0.0	0:00	/sbin/getty
771	0.0	0.0	0:00	/sbin/getty
772	0.0	0.0	0:00	/sbin/getty
781	0.0	0.0	0:00	/sbin/getty
805	0.0	0.0	0:00	acpid
807	0.0	0.4	0:00	whoopsie
811	0.0	0.0	0:00	cron
812	0.0	0.0	0:00	atd
834	0.0	0.0	0:00	/usr/sbin/VBoxService
953	0.0	0.1	0:00	/usr/sbin/dnsmasq
1082	0.0	0.0	0:00	/sbin/getty
1135	0.0	0.3	0:00	lightdm
1149	1.9	6.7	0:27	/usr/bin/X
1164	0.0	0.4	0:00	/usr/lib/accountsservice/accounts-daemon
1180	0.0	0.3	0:00	/usr/sbin/console-kit-daemon
1307	0.0	0.3	0:00	lightdm
1319	0.0	0.3	0:00	/usr/lib/upower/upowerd
1386	0.0	0.1	0:00	/usr/lib/rtkit/rtkit-daemon
1479	0.0	0.3	0:00	/usr/bin/gnome-keyring-daemon
1490	0.0	0.8	0:00	gnome-session
1548	0.0	0.1	0:00	/usr/bin/VBoxClient
1555	0.0	0.1	0:00	/usr/bin/VBoxClient
1561	0.0	0.1	0:00	/usr/bin/VBoxClient
1564	0.0	0.0	0:00	/usr/bin/ssh-agent
1567	0.0	0.0	0:00	/usr/bin/dbus-launch
1568	0.2	0.2	0:03	//bin/dbus-daemon
1581	0.0	1.6	0:01	/usr/lib/gnome-settings-daemon/gnome-settings-daemon
1589	0.0	0.2	0:00	/usr/lib/gvfs/gvfsd
1591	0.0	0.3	0:00	/usr/lib/gvfs//gvfs-fuse-daemon
1597	0.2	1.2	0:03	metacity

Tabulka 6 - Výpis procesů Ubuntu (Část 3)

PID	CPU	MEM	TIME	COMMAND
1611	0.0	0.2	0:00	/usr/lib/i386-linux-gnu/gconf/gconfd-2
1614	0.1	0.5	0:01	/usr/bin/pulseaudio
1617	0.0	0.2	0:00	/usr/lib/pulseaudio/pulse/gconf-helper
1618	0.1	2.5	0:02	unity-2d-panel
1619	0.6	6.2	0:08	unity-2d-shell
1624	0.0	1.4	0:00	nm-applet
1625	0.0	1.3	0:00	/usr/lib/policykit-1-gnome/polkit-gnome-authentication-agent-1
1626	0.0	1.0	0:00	bluetooth-applet
1631	0.0	0.7	0:00	/usr/lib/gnome-settings-daemon/gnome-fallback-mount-helper
1633	0.2	2.3	0:03	nautilus
1654	0.2	1.0	0:04	/usr/lib/bamf/bamfdaemon
1674	0.0	0.4	0:00	/usr/lib/gvfs/gvfs-gdu-volume-monitor
1677	0.0	0.3	0:00	/usr/lib/udisks/udisks-daemon
1678	0.0	0.0	0:00	udisks-daemon:
1686	0.0	0.2	0:00	/usr/lib/gvfs/gvfs-gphoto2-volume-monitor
1688	0.0	0.2	0:00	/usr/lib/gvfs/gvfs-afc-volume-monitor
1711	0.0	0.3	0:00	/usr/lib/gvfs/gvfsd-trash
1714	0.0	0.2	0:00	/usr/lib/gvfs/gvfsd-burn
1716	0.0	0.2	0:00	/usr/lib/dconf/dconf-service
1722	0.0	0.1	0:00	/usr/lib/gvfs/gvfsd-metadata
1734	0.0	1.6	0:01	/usr/lib/unity/unity-panel-service
1741	0.0	0.6	0:00	/usr/lib/indicator-datetime/indicator-datetime-service
1743	0.0	0.4	0:00	/usr/lib/indicator-application/indicator-application-service
1745	0.0	0.6	0:00	/usr/lib/indicator-sound/indicator-sound-service
1747	0.0	0.9	0:00	/usr/lib/indicator-printers/indicator-printers-service
1749	0.0	0.6	0:00	/usr/lib/indicator-session/indicator-session-service
1758	0.0	0.5	0:00	/usr/lib/indicator-messages/indicator-messages-service
1785	0.0	0.2	0:00	/usr/lib/geoclue/geoclue-master
1790	0.0	0.4	0:00	/usr/lib/ubuntu-geoip/ubuntu-geoip-provider
1802	0.0	0.4	0:00	/usr/lib/indicator-appmenu/hud-service
1817	0.0	0.9	0:00	/usr/lib/unity-lens-applications/unity-applications-daemon
1822	0.0	1.1	0:00	/usr/bin/python
1823	0.0	0.5	0:00	/usr/lib/unity-lens-files/unity-files-daemon
1825	0.0	0.7	0:00	/usr/lib/unity-lens-music/unity-music-daemon
1852	0.0	0.5	0:00	/usr/bin/zeitgeist-daemon
1859	0.0	0.8	0:00	/usr/lib/zeitgeist/zeitgeist-fts
1860	0.0	0.5	0:00	zeitgeist-datahub
1875	0.0	0.0	0:00	/bin/cat
1903	0.0	0.8	0:00	/usr/lib/gnome-disk-utility/gdu-notification-daemon