

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

**Správa, konfigurace a analýza Windows 11 - video
tutoriál**

Bakalářská práce

Autor: Adam Svítek
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. Mgr. Josef Horálek, Ph.D.

Hradec Králové

duben 2024

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 25.4.2024

Adam Svítek

Poděkování:

Děkuji vedoucímu bakalářské práce panu doc. Mgr. Josefu Horálkovi, Ph.D. za cenné rady, připomínky a metodické vedení této práce.

Abstrakt

Bakalářská práce se zaměřuje na návrh praktických cvičení pro studenty s cílem podpořit tvorbu sylabů pro nové předměty související s operačním systémem Windows 11. Teoretická část práce důkladně popisuje vývoj a architekturu operačních systémů Windows, včetně principů správy a konfigurace. Zvláštní pozornost je věnována také technologii Docker a jejímu využití v kontextu operačních systémů. V praktické části byla představena série cvičení, která zahrnují konkrétní postupy a nástroje pro správu a konfiguraci systému Windows 11. Obsahují demonstrace práce s Windows Subsystem for Linux, kontejnerizaci aplikací pomocí Dockeru a diagnostiku systému Windows 11.

Klíčová slova: Windows 11, Windows Subsystem for Linux, Docker, operační systém, správa systému, diagnostika systému

Abstract

Title: Managing, configuring, and analyzing Windows 11 - video tutorial

The bachelor thesis focuses on the design of practical exercises for students to support the development of syllabi for new courses related to the Windows 11 operating system. Particular attention is also paid to the Docker technology and its use in the context of operating systems. In the practical part, a series of exercises are presented that include specific procedures and tools for managing and configuring Windows 11. They include demonstrations of working with Windows Subsystem for Linux, containerizing applications using Docker, and diagnosing Windows 11.

Key words: Windows 11, Windows Subsystem for Linux, Docker, operating system, system administration, system diagnostics

Obsah

1	Úvod.....	1
2	Cíl práce.....	3
3	Operační systém	4
	3.1 Úvod do operačního systému.....	4
	3.2 Základy operačních systémů.....	4
4	Vývoj operačního systému Windows	5
	4.1 Úvod do systému Windows	5
	4.2 Windows 7.....	5
	4.3 Windows 8.....	5
	4.3.1 Windows 8.1.....	6
	4.4 Windows 10	6
	4.5 Windows 11	6
5	Architektura Windows NT.....	8
	5.1 Obecná struktura	8
	5.2 Režim jádra	8
	5.2.1 Super administrátor	9
	5.3 Uživatelský režim	9
	5.3.1 Subsystem Windows.....	9
	5.3.2 Subsystem POSIX.....	10
	5.3.3 Windows Subsystem pro Linux	10
	5.4 Aplikace	10
	5.5 Procesy.....	11
	5.6 Procesy na pozadí.....	11
	5.7 Systémové procesy.....	12

5.7.1	Session Manager Subsystem	12
5.7.2	Local Security Authority Process	12
5.7.3	Windows Logon Process.....	12
6	Nástroje pro správu a diagnostiku systému	13
6.1	Úvod.....	13
6.2	Správce úloh	13
6.3	Process Explorer	14
6.4	Sledování prostředků.....	14
6.4.1	Processor	14
6.4.2	Paměť.....	14
6.4.3	Disk	15
6.4.4	Sít'	15
7	Docker.....	16
7.1	Úvod.....	16
7.2	Platforma Docker	16
7.3	Architektura Dockeru.....	16
7.4	Démon Docker.....	17
7.5	Klient Docker	17
7.6	Pracovní plocha Docker.....	17
7.7	Registry nástroje Docker	17
7.8	Image (obraz)	18
7.8.1	Image Hello-world	18
7.8.2	Image Nginx.....	18
7.8.3	Image MySQL	19
7.9	Kontejnery	21
7.10	Porovnání mezi virtualizací a Dockerem	21

7.10.1 Virtuální počítače.....	21
7.10.2 Docker kontejner	21
7.10.3 Podobnosti a rozdíly.....	22
8 Praktická cvičení.....	24
8.1 Úvod.....	24
8.2 První cvičení – Windows Subsystem for Linux.....	25
8.2.1 Úvod.....	25
8.2.2 Zapnutí nebo vypnutí funkce systému Windows.....	25
8.2.3 Instalace WSL.....	26
8.2.4 Dostupné distribuce Linuxu	26
8.2.5 Práce s příkazovou řádkou	28
8.2.6 Vytváření skriptů.....	29
8.2.7 Instalace grafického rozhraní XFCE	31
8.2.8 Sdílení souborů mezi Windows 11 a distribucí Ubuntu.....	31
8.2.9 Závěr.....	32
8.3 Druhé cvičení – Docker.....	32
8.3.1 Instalace Dockeru.....	32
8.3.2 Práce s Dockerem.....	32
8.3.3 Docker Hub	32
8.3.4 Připojení k webovému serveru přes image Nginx.....	34
8.3.5 Vytvoření SQL databáze.....	36
8.3.6 Závěr.....	38
8.4 Třetí cvičení – Diagnostika systému.....	39
8.4.1 Úvod.....	39
8.4.2 Konfigurace systému.....	39
8.4.3 Spuštěné aplikace po zapnutí počítače	41

8.4.4	Spuštěné služby po zapnutí počítače	41
8.4.5	Práce s Process Explorer	41
8.4.6	Práce s nástrojem Sledování prostředků	42
8.4.7	Závěr	43
9	Závěry a doporučení	44
10	Seznam použité literatury	45
11	Přílohy	49
12	Zadání práce z IS (eVŠKP)	1

Seznam obrázků

Obrázek 1 Grafická reprezentace architektury Dockeru (zdroj: https://docs.docker.com/get-started/overview/)	17
Obrázek 2 Pracovní plocha ve Windows 11 (zdroj: autor)	24
Obrázek 3 Okno Zapnout nebo vypnout funkce systému Windows (zdroj: autor)	25
Obrázek 4 Vygenerovaný seznam všech dostupných distribucí Linuxu (zdroj: autor)	27
Obrázek 5 Ukázkový skript ve rozhraní NANO (zdroj: autor)	30
Obrázek 6 Okno Připojení ke vzdálené ploše (zdroj: autor)	31
Obrázek 7 Grafického rozhraní XFCE (zdroj: autor)	31
Obrázek 8 Docker Hub (zdroj: autor)	33
Obrázek 9 Spuštění kontejneru hello-world v Dockeru (zdroj: autor)	34
Obrázek 10 Uvítací zpráva z web serveru (zdroj: autor)	35
Obrázek 11 Zobrazení všech kontejnerů v Dockeru (zdroj: autor)	35
Obrázek 12 Zobrazení vlastního projektu na web serveru (zdroj: autor)	36
Obrázek 13 Zakládání nového databázového serveru (zdroj: autor)	37
Obrázek 14 Připojování do databázového serveru přes Docker (zdroj: autor)	38
Obrázek 15 Zobrazení tabulky <i>student</i> přes Docker (zdroj: autor)	38
Obrázek 16 Rozhraní Konfigurace systému (zdroj: autor)	40
Obrázek 17 Rozhraní Process Explorer (zdroj: autor)	42
Obrázek 18 Rozhraní Sledování prostředků (zdroj: autor)	43

Seznam tabulek

Tabulka 1 Rozdíly mezi funkcemi virtuálních počítačů a kontejnerů (zdroj: https://learn.microsoft.com/cs-cz/training/modules/run-containers-windows-server/3-list-differences-between-containers-vms)	23
---	----

Seznam zkratek

API	Application Programming Interface
BIOS	Basic Input Output System
CPU	Central processing unit
DLL	Dynamic Link Library
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IBM	International Business Machines Corporation
IMAP	Internet Message Access Protocol
LSASS	Local Security Authority Process
NVMe	Non-Volatile Memory Express
POP3	Post Office Protocol 3
RAM	Random Access Memory
RDBMS	Relational Database Management System
REST	Representational State Transfer
SMSS	Session Manager Subsystem
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSD	Solid-State Drives
UNIX	UNiplexed Information Computing System
WINLOGON	Windows Logon Process
WSL	Windows Subsystem for Linux
XFCE	XForms Common Environment

1 Úvod

Cílem této bakalářské práce je navrhnout praktická cvičení pro studenty, které můžou v budoucnu pomoci s vytvářením sylabem pro nové předměty s tímto zaměřením. Cvičení se zaměřují na správu a konfigurace ve operačním systému Windows 11.

Pro naplnění cílů byly v teoretické části práce popsány a vysvětleny definice pojmů, které jsou potřebné k pochopení dílčích úloh v praktické části práce. Teoretická část práce je rozdělena do několika hlavních kapitol. V první kapitole je obecně a krátce představen pojem operační systém. V druhé kapitole je představen vývoj operačních systémů Windows od verze 7 až po současnou verzi Windows 11. Tato část přináší přehled vývoje, změn a inovací, které byly implementovány v průběhu této éry. Třetí kapitola se detailněji zabývá základní architekturou operačních systémů Windows a na principech, na kterých jsou postaveny. Tato sekce více rozebírá komponenty a vrstvy, které tvoří jádro systému Windows NT. Čtvrtá kapitola se zaměřuje na nástroje určené pro správu, konfiguraci a diagnostiku operačního systému Windows 11. Nakonec pátá kapitola pojednává o technologii Docker a jejímu využití v kontextu operačních systémů. Tato kapitola představuje kontejnerizaci, principy Dockeru a rozdíl mezi funkcemi virtuálních počítačů a kontejnerů.

Pro naplnění cílů byly v praktické části práce představeny praktická cvičení, která jsou jednotlivě rozdělená do každé kapitoly. První kapitola se věnuje cvičení, které se soustředí na Windows Subsystem for Linux (WSL), což je funkce umožňující spouštění Linuxových distribucí přímo na systému Windows 11. V této části se probírá potřebné nastavení funkcí pro instalaci WSL a následná instalace distribuce Linuxu. Dále se rozebírá práce příkazovou řádkou v Linuxovém prostředí, tvorbě skriptů, instalaci grafického rozhraní XFCE a sdílení souborů mezi Windows 11 a distribucí Ubuntu. Druhá kapitola se zaměřuje na cvičení, které představuje nástroj Docker pro kontejnerizaci aplikací. Zde je uvedeno, jak na instalaci Dockeru, jak pracovat s Docker kontejnery, jak využívat Docker Hub pro vyhledávání *images*, jak se připojit k webovému serveru pomocí Docker image Nginx a jak se připojit k SQL databázi v kontejnerovém prostředí. Poslední třetí kapitola prakticky popisuje cvičení, které se zabývá diagnostikou systému Windows 11. Tato část se zaměřuje na konfiguraci

systemu, analýzu spuštěných aplikací a služeb po startu počítače a na práci s nástroji Process Explorer a Sledování prostředků.

V rámci této bakalářské práce byl využit nástroj umělé inteligence k podpoře tvorby obsahu. Konkrétně při hledání relevantních odborných webových článků pro teoretickou část práce, což umožňovalo hlubší vhled do zkoumaných témat. Dále nástroj umělé inteligence byl využit k podpoře návrhu praktických cvičení. Důležité je zdůraznit, že samotné řešení a postup praktických cvičení byl výhradně vypracován autorem práce. Získané informace a analýzy poskytnuté nástrojem umělé inteligence sloužily jako inspirace a podpora při tvorbě obsahu, avšak samotný výsledek a implementace cvičení byla výsledkem individuálního úsilí autora.

2 Cíl práce

Cílem práce je připravit ucelený výukový materiál pro budoucí navržené předměty, které budou studentům poskytovat konkrétní praktické dovednosti v oblasti správy a konfiguraci počítačových systémů. K tomuto účelu je vytvořila série praktických cvičení, kde každé cvičení obsahuje tyto tři klíčové dokumenty:

- Dokument s úkoly: Každé cvičení je doprovázeno dokumentem obsahujícím specifické úkoly, který popisují cíle, požadavky a očekávané výstupy cvičení. Tyto úkoly jsou navrženy tak, aby studenti měli možnost prakticky vyzkoušet a aplikovat teoretické znalosti, které získali v předchozích částech kurzu.
- Dokument s postupem: Pro usnadnění studentům je vytvořen podrobný dokument s postupem, který krok za krokem vede studenty k řešení jednotlivých úkolů. Tento dokument slouží jako průvodce pro efektivní a správné provedení konkrétních úkolů správy a konfigurace ve Windows 11.
- Video tutoriál: Kromě psaného materiálu je pro každé cvičení připraven video tutoriál, který poskytuje vizuální a textový průvod, který studentům umožňuje sledovat a reprodukovat postupy uvedené v dokumentu s postupem a ověřit správnost řešení.

Všechny tyto tři dokumenty jsou dostupné na digitálním repositáři portálu GitHub.

3 Operační systém

3.1 Úvod do operačního systému

Výběr operačního systému je pro každý počítačový systém zásadním rozhodnutím, protože určuje schopnosti a funkce systému na základě jeho konfigurace. Operační systém funguje jako prostředník mezi uživatelem počítače a počítačovým hardwarem (Madhavi A., 2024). Jeho účelem je poskytnout prostředí, ve kterém může uživatel pohodlně a efektivně provádět programy (Madhavi A., 2024). Operační systém je navržen tak, aby mohl spravovat celkové zdroje a operace počítače.

3.2 Základy operačních systémů

Operační systém je základní programové vybavení počítače, které umožňuje běh programů a ovlivňuje, jak bude počítačový systém komunikovat s uživatelem (Vaňková J., 2011). Kromě počítačů používají operační systém i některá zařízení, která se za počítač nepovažuje, ale mají v sobě nějaký druh počítače zabudovaný (Vaňková J., 2011). Operační systém skrývá detaily ovládání jednotlivých zařízení v počítači (tzv. hardware) a definuje standardní rozhraní pro volání systémových služeb tak, že vytváří abstraktní vrstvu s jednoduchými funkcemi (Vaňková J., 2011). Operační systém vytváří prostředí pro běh programů a zároveň prostředí pro komunikaci s uživatelem (Vaňková J., 2011). Přitom plní především následující funkce: poskytuje funkce pro ovládání hardwaru, vytváří aplikační rozhraní pro programování (API), poskytuje uživatelské rozhraní umožňující uživateli komunikaci s počítačem nebo jiným zařízením a jeho ovládání (Vaňková J., 2011).

4 Vývoj operačního systému Windows

4.1 Úvod do systému Windows

Systém Windows je jedním z nejpoužívanějších operačních systémů na světě, který je především známý svým uživatelsky přívětivým rozhraním a neustálými inovacemi. Technický vývoj systému Microsoft Windows od Windows 7 po Windows 11 ukazuje řadu pokroků, které ovlivnily výkon, zabezpečení a celkový uživatelský komfort operačního systému. Každá verze přinesla různá vylepšení, optimalizovala využití hardwaru, zlepšila herní možnosti a posílila bezpečnostní opatření. Systém Windows 11 se svou zdokonalenou architekturou, zvýšeným výkonem a nejmodernějšími funkcemi je důkazem odhodlání společnosti Microsoft inovovat a plnit neustále se měnící požadavky moderního počítačového prostředí.

4.2 Windows 7

Systém Windows 7, vyvinutý společností Microsoft, je významnou verzí operačního systému Windows NT. Dostupný byl od 22. října 2009, po svém předchůdci Windows Vista (Britannica, 2022). Systém Windows 7 byl navržen jako dostupná aktualizace systému Microsoft Windows, jejímž cílem bylo zlepšit kritické přijetí systému Vista a zároveň zachovat kompatibilitu hardwaru a softwaru (Britannica, 2022). Rozhraní systému Windows 7 se téměř nelišilo od rozhraní systému Vista, kde největší rozdíl byl ve zřetelně zvýšení rychlosti a v nenáročných systémových požadavcích, a tato vylepšení získala pozitivní ohlasy (Britannica, 2022).

4.3 Windows 8

Systém Windows 8, vydaný v říjnu 2012, přinesl radikální odklon od tradičního rozhraní systému Windows. Nabídka *Start* byla nahrazena obrazovkou *Start* s živými dlaždicemi, optimalizovanými pro dotyková zařízení. Operační systém také znamenal zavedení obchodu Windows Store, který uživatelům nabídl centralizované místo pro stahování a instalaci aplikací. Ačkoli se systém Windows 8 setkal se smíšenými ohlasy kvůli dramatickým změnám uživatelského rozhraní. Tato verze vytvořila základ grafického rozhraní pro budoucí vývoj systému Windows.

4.3.1 Windows 8.1

V říjnu 2013 společnost Microsoft vydala systém Windows 8.1, který reagoval na některé kritické připomínky k systému Windows 8. Tato aktualizace vrátila na plochu tlačítko Start, což uživatelům tradičních desktopů usnadnilo navigaci v rozhraní. Kromě toho přinesla četná vylepšení výkonu, aktualizace funkcí a možnost spouštět systém přímo na plochu.

4.4 Windows 10

Systém Windows 10 byl vydán v červenci 2015 jako jeden z nejvýznamnějších milníků v historii systému Windows. Cílem této verze bylo sloučit nejlepší prvky systémů Windows 7 a Windows 8. Vrátila se tradiční nabídka *Start* a zároveň se zachovaly živé dlaždice pro zařízení s dotykovým ovládním (GCFGlobal.org, 2022). Systém Windows 10 byl zamýšlen jako univerzální platforma, kterou by bylo možné provozovat na široké škále zařízení, včetně počítačů, tabletů, chytrých telefonů a konzolí Xbox (GCFGlobal.org, 2022).

Systém Windows 10 představil několik významných funkcí, například virtuální asistentku Cortanu, prohlížeč Microsoft Edge a grafické rozhraní DirectX 12, které zlepšilo herní výkon (GCFGlobal.org, 2022). Systém Windows 10 navíc znamenal konec tradičního cyklu vydávání nových systému Windows, protože společnost Microsoft přešla na model průběžných aktualizací, který poskytuje pravidelné aktualizace funkcí dvakrát ročně, a však to se změnilo v roce 2021, kdy vyšla nová verze systému Windows (Microsoft, 2023).

4.5 Windows 11

Windows 11, oznámený v červnu 2021 a oficiálně vydaný v říjnu 2021, představoval významný vývoj operačního systému Windows. Tato verze se zaměřila na modernizaci uživatelského rozhraní, zvýšení produktivity a poskytování lepšího zážitku napříč zařízeními (Microsoft, 2023). Nejviditelnější změnou v systému Windows 11 byla vycentrovaná nabídka *Start* s přepracovanými živými dlaždicemi a zjednodušeným rozvržením. Hlavní panel byl také vycentrován a systémové ikony byly přemístěny tak, aby se zlepšila estetika a konzistence (Microsoft, 2023). Systém Windows 11 navíc představil nové funkce *Snap Layouts* a *Snap Groups*, které usnadňují multitasking a efektivní uspořádání oken (Microsoft, 2023). Microsoft také výrazně investoval do

vylepšení her, zavedl funkci Auto HDR pro širší škálu kompatibilních her a DirectStorage, která zrychlila načítání her využitím výkonu disků NVMe SSD (Muchmore M., 2022). Systém Windows 11 navíc předvedl obnovený důraz na podporu virtuálních ploch a dotykové interakce, čímž vyhověl stále rozmanitější škále zařízení (Microsoft, 2023).

5 Architektura Windows NT

5.1 Obecná struktura

Architektura Windows NT, řady operačních systémů vytvořených a prodávaných společností Microsoft, je vrstvený design, který se skládá ze dvou hlavních komponent - uživatelského režimu (user mode) a jádrového režimu (kernel mode). Je to preemptivní, reentrantní multi-taskingový operační systém, který byl navržen pro práci s jednoprocessorovými a symetrickými multiprocessorovými (SMP) počítači (Sajal, 2021).

Aby byla zachována kompatibilita se systémy jiných výrobců než Intel, operační systémy Windows podporují pouze dvě úrovně oprávnění – Ring 0 a Ring 3, namísto ze čtyř (Roman S., 2014). Pokud vlákno operuje v kruhu 0, říká se, že je v režimu jádra, a pokud vlákno operuje v kruhu 3, říká se, že je v uživatelském režimu. Nízkoúrovňový kód operačního systému se spouští v režimu jádra, zatímco kód uživatelských aplikací běží v uživatelském režimu (Roman S., 2014).

5.2 Režim jádra

Režim jádra systému Windows NT má plný přístup k hardwaru a systémovým prostředkům počítače. Jádrový režim zastavuje služby a aplikace v uživatelském režimu před přístupem do kritických oblastí operačního systému, ke kterým by neměly mít přístup (Sajal, 2021). Jádro Windows NT je hybridní jádro. Architektura se skládá z jednoduchého jádra, vrstvy abstrakce hardwaru (HAL), ovladačů a řady služeb (kolektivně označovaných jako Executive), které všechny existují v jádrovém režimu (Sajal, 2021).

Všechny kódy, které běží v jádrovém režimu, sdílejí jeden virtuální adresní prostor. Proto není ovladač v jádrovém režimu izolován od ostatních ovladačů a samotného operačního systému (Microsoft, 2022). Pokud ovladač v jádrovém režimu se omylem zapíše na nesprávnou virtuální adresu, mohou být ohrožena ty data, která patří operačnímu systému nebo jinému ovladači (Microsoft, 2022).

5.2.1 Super administrátor

V kontextu správy systému, Windows NT zavádí koncept super administrátora, známého také jako *System Administrator* nebo *Admin* (ServerWatch, 2001). Super administrátor má neomezená práva a privilegia, což mu umožňuje spravovat všechny aspekty systému. To zahrnuje instalaci a odstraňování softwaru, správu uživatelských účtů, změnu systémových nastavení a přístup k citlivým částem systému (ServerWatch, 2001).

5.3 Uživatelský režim

Uživatelský režim ve Windows NT je tvořen různými systémově definovanými procesy a DLL. Rozhraní mezi aplikacemi v uživatelském režimu a funkcemi jádra operačního systému kernel se nazývá *prostředí podsystému*. Windows NT může mít více než jeden z těchto podsystémů, z nichž každý implementuje jinou sadu API (Sajal, 2021). Tento mechanismus byl navržen tak, aby podporoval aplikace napsané pro mnoho různých typů operačních systémů. Žádný z podsystémů prostředí nemůže přímo přistupovat k hardwaru.

5.3.1 Subsystém Windows

Subsystém Win32 se skládá z několika základních komponent, které společně umožňují provoz 32bitových aplikací v prostředí operačního systému Windows. Hlavním procesem tohoto subsystému je *csrss.exe* (Client/Server Runtime Subsystem), který hraje klíčovou roli při správě grafické instrukční sady a zajišťuje účinnou komunikaci mezi aplikacemi a jádrem operačního systému (Pilici S., 2023).

Další důležitou komponentou jsou knihovny DLL (Dynamic Link Libraries), které subsystém Win32 využívá. Tyto knihovny poskytují sadu funkcí, které mohou být sdíleny mezi různými aplikacemi. Knihovna *ntdll.dll* je odpovědná za propojení mezi aplikacemi a jádrem systému, přičemž zprostředkovává volání do jádra (Pilici S., 2023).

Ovladač jádra *win32k.sys* je další součástí subsystému Win32. Tento ovladač zabezpečuje správnou integraci grafického uživatelského rozhraní a je odpovědný za řízení různých grafických prvků jako jsou okna, panely a další vizuální komponenty.

5.3.2 Subsystem POSIX

Subsystem POSIX je sada mezinárodních standardů, která detailně popisuje aplikační rozhraní v operačních systémech založených na Unix, s klíčovým procesem *psxss.exe*. Hlavním prvkem tohoto subsystému je komponenta běžící v jádrovém režimu s názvem *win32k.sys*. Tato komponenta zajišťuje správnou integraci a komunikaci mezi aplikacemi a jádrem operačního systému. Díky implementaci SUA mohou uživatelé v prostředí Windows využívat aplikace a služby navržené pro prostředí založené na Unixu, což zvyšuje flexibilitu a interoperabilitu mezi různými platformami.

5.3.3 Windows Subsystem pro Linux

Windows Subsystem for Linux (WSL) je funkce Windows, která umožňuje vývojářům spustit Linuxové prostředí na jejich stroji s Windows, aniž by bylo třeba samostatné virtuální stroje nebo dual booting (Microsoft, 2023) (Heddings A., 2021). WSL je navržen tak, aby poskytoval plynulý a produktivní zážitek pro vývojáře, kteří chtějí současně používat Windows a Linux (Microsoft, 2023).

Subsystem WSL umožňuje vývojářům nainstalovat linuxovou distribuci (jako Ubuntu, OpenSUSE, Kali, Debian, Arch Linux atd.) a používat linuxové aplikace, nástroje a příkazový řádek Bash přímo v systému Windows, bez modifikace, bez režie tradičního virtuálního stroje nebo dual booting nastavení (Microsoft, 2023).

5.4 Aplikace

Aplikace v operačním systému jsou softwarové programy navržené tak, aby vykonávaly specifické úkoly pro uživatele. Tyto úkoly mohou zahrnovat zpracování textu, tabulkové kalkulace, grafický design, komunikaci, zábavu a mnoho dalších funkcí. Aplikace fungují jako rozhraní mezi uživatelem a hardwarovými zdroji počítače, přičemž operační systém poskytuje nezbytné služby pro jejich spuštění, jako je správa paměti, procesů a vstupně-výstupních operací.

Operační systém Windows poskytuje abstrakci hardwaru pro programátora a uživatele, což znamená, že poskytuje generické rozhraní k službám poskytovaným základním hardwarem (Wienand I., 2004). Bez operačních systémů by každý programátor musel znát nejintimnější detaily základního hardwaru, aby cokoli spustil (Wienand I., 2004).

Dále, operační systém Windows je zodpovědný za správu zdrojů v systému. Mnoho úloh bude soutěžit o zdroje systému, jak běží, včetně času procesoru, paměti, disku a vstupu uživatele (Wienand I., 2004). Úkolem operačního systému je rozhodovat o těchto zdrojích pro více úloh a umožnit jim přístup v řádném pořadí (Wienand I., 2004).

5.5 Procesy

Proces v operačním systému je základní jednotka, která je schopna provádět instrukce a obsluhovat systémové volání. Každý proces má svůj vlastní virtuální adresní prostor, soubor kódů, data a systémové zdroje. Procesy jsou základem pro multitasking a umožňují operačnímu systému spouštět více aplikací současně bez vzájemného ovlivňování. (Turiceanu V., 2024) Windows 11 optimalizuje správu procesů tím, že dynamicky přiděluje procesorový čas a paměťové zdroje, což zajišťuje efektivní využití hardwaru a stabilní provoz systému (Turiceanu V., 2024). Procesy mohou být klasifikovány jako uživatelské nebo systémové, přičemž systémové procesy jsou nezbytné pro správnou funkci operačního systému, zatímco uživatelské procesy jsou iniciálně spuštěny uživateli. Výkon a stabilita systému Windows 11 jsou úzce spojeny s efektivním řízením procesů, které zahrnuje správu požadovaných úloh, optimalizaci počtu běžících procesů a monitorování využití CPU a paměti jednotlivými procesy (Turiceanu V., 2024). Důležité je najít správnou rovnováhu mezi plynulým chodem počítače a bezpečností.

5.6 Procesy na pozadí

Procesy na pozadí, které jsou provozovány nainstalovanými aplikacemi, ovladači, antivirem a dokonce i samotným operačním systémem Windows, jsou skryté a běží mimo dohled uživatele. Na rozdíl od aplikací, tyto procesy nemají žádné uživatelské rozhraní, ve kterém by uživatel mohl provádět akce (Khanse A., 2024).

Tyto procesy na pozadí jsou nezbytné pro správné fungování počítače. Mohou například sledovat stav systémových zdrojů, aktualizovat software, poskytování síťových služeb, provádět úkoly plánované na pozadí nebo zpracovávat data pro jiné aplikace (Singh N., 2023). I když tyto procesy nejsou viditelné pro uživatele, jsou

nezbytné pro správné fungování daného počítače a mohou výrazně ovlivnit jeho výkon a stabilitu.

5.7 Systémové procesy

Operační systém Windows obsahuje mnoho systémových procesů, které jsou přítomny při každém spuštění počítače. Tyto procesy jsou zodpovědné za mnoho věcí. Od inicializace a vytváření uživatelského rozhraní až po načítání potřebných ovladačů a knihoven DLL (Bencherchali N., 2020). Systémové procesy se odlišují od ostatních běžných procesů tím, že mají přístup do jádrového režimu.

5.7.1 Session Manager Subsystem

Session Manager Subsystem (*smss.exe*) je jedním z prvních systémových procesů vytvořených při spuštění operačního systému. Je zodpovědný za vytvoření seznamu proměnných prostředí, deskriptorů zabezpečení, které budou používány různými systémovými prostředky, a inicializaci zbytku registru (Bencherchali N., 2020).

Hlavní proces *smss.exe* vytvoří nejméně dvě instance sebe sama. Jedna v *relaci 0* (OS) bude zodpovědná za vytvoření procesu *wininit.exe* a druhá v *relaci 1* (User), která bude představovat prvního přihlášeného uživatele a vytvoří proces *winlogon.exe*. Oba vytvoří proces *csrss.exe* (Bencherchali N., 2020).

5.7.2 Local Security Authority Process

Local Security Authority Process (*LSASS*) je zodpovědný za vynucení bezpečnostní politiky v systému (Harsana L., 2023). *LSASS* ověřuje uživatele přihlašující se k počítači nebo serveru, zpracovává změny hesel a generuje přístupové tokeny (Harsana L., 2023). *LSASS* také zapisuje do Bezpečnostního protokolu Windows a pokud je proces *lsass.exe* násilně ukončen, tak systém ztratí přístup ke všem účtům, včetně *NT AUTHORITY*, což vyvolá restart počítače (Harsana L., 2023).

5.7.3 Windows Logon Process

Tento proces provádí řadu kritických úloh souvisejících s procesem přihlášení do Windows (Hoffman Ch., 2017). Například při přihlášení je proces *winlogon.exe* zodpovědný za načtení vašeho uživatelského profilu do registru (Hoffman Ch., 2017). To umožňuje programům používat klíče pod *HKEY_CURRENT_USER*, které se liší pro každý uživatelský účet Windows (Hoffman Ch., 2017).

6 Nástroje pro správu a diagnostiku systému

6.1 Úvod

Při každém spuštění počítače systém Windows 11 automaticky spouští určité programy a procesy, které mohou zpomalit bootování systému. Obvykle se tyto potíže objeví až po pár měsíců či letech používání počítače, ale to záleží na uživateli, jak často instaluje a odstraňuje aplikace. Třeba odinstalování aplikace může trochu zrychlit proces spuštění systému, ale ne tak úplně, jelikož po odinstalování aplikací, zejména počítačových her, stále zůstávají soubory, které po pár letech ovlivňují spuštění systému.

Například v nástroji *Správce úloh* lze v kartě *Spouštěné aplikace* nalézt všechny programy, které se automaticky spouštějí po zapnutí počítače, a pomocí tohoto nástroje lze tyto libovolné programy zakázat, aby urychlily start systému. Pro jednotlivý program na této kartě lze vidět jeho podrobnosti jako název programu, vydavatele (společnost, která jej vytvořila), stav automatického spuštění (povolen nebo zakázán) a dopad na spuštění systému (nízký, střední, vysoký atd.).

6.2 Správce úloh

Microsoft ve svých systémech Windows vždy poskytuje *Správce úloh*, snadno použitelný nástroj pro zobrazení procesů (aplikací a služeb) spuštěných v systému, aby uživatelům pomohl získat přehled o činnostech procesů v systému. *Správce úloh* nabízí jednoduché a přímočaré rozhraní, aby pomohl uživatelům se lépe vyznat podrobnostech o systému. Umožňuje uživatelům zobrazit souhrnný seznam procesů, služeb, vytvořené uživatele, jednotlivé grafy s výkonem systému a využití sítě. *Správce úloh* je aplikace, na kterou se uživatelé obvykle obracejí, aby zjistili, proč je jejich systém pomalý, a aby případně ukončili chybné procesy (Rusinovich M., Margosis A., 2016). Často neposkytuje dostatečně hluboký přehled o tom, co je příčinou nesprávného chování procesu, ani nezobrazuje klíčové údaje, které by technickému uživateli pomohly identifikovat proces jako malware (Rusinovich M., Margosis A., 2016).

6.3 Process Explorer

Process Explorer na systému Windows je pokročilý nástroj, který poskytuje uživatelům podrobný pohled na všechny procesy běžící v operačním systému a jejich související informace. Tento nástroj zobrazuje informace o tom, které popisovače a procesy DLL byly otevřeny nebo načteny (Microsoft, 2023).

Process Explorer se skládá ze dvou dílčích oken. V horním okně se vždy zobrazí seznam aktuálně aktivních procesů, včetně názvů vlastních účtů, zatímco informace zobrazené v dolním okně závisí na režimu, ve kterém je *Process Explorer* (Microsoft, 2023). Pokud je v režimu zpracování, uvidíte popisovače, které proces vybraný v horním okně otevřel. Pokud je *Process Explorer* v režimu knihovny DLL, uvidíte knihovny DLL a soubory mapované do paměti, které proces načtl (Microsoft, 2023).

6.4 Sledování prostředků

Systém Windows obsahuje také nástroj *Sledování prostředků*, který umožňuje uživatelům monitorovat výkon a stav různých hardwarových komponent v jejich počítači (Yosifovich P., Ionescu A., 2017). Tento nástroj poskytuje informace, který zobrazuje čtyři základní systémové prostředky: procesor, disk, síť a paměť. Zobrazení podrobností těchto prostředků je na stejné úrovni jako v nástroji *Správce úloh*. Avšak tyto sekce prostředků lze rozbalit a tím získat doplňující informace.

6.4.1 Procesor

Karta *Procesor* obsahuje seznam služeb, údaje o využití procesoru těchto služeb a průměr využití na procesoru. Přidává i sloupec pro průměrné zatížení procesoru, pomocí kterého se lépe představuje, který z procesů je nejaktivnější (Yosifovich P., Ionescu A., 2017).

6.4.2 Paměť

Karta *Paměť* zobrazuje většinu stejných informací, které lze získat za pomoci nástroje *Správce úloh*, ale je organizována pro celý systém. Sloupcový graf fyzické paměti zobrazuje aktuální stav fyzické paměti, který je rozdělen na hardwarově rezervovanou, používanou, modifikovanou, pohotovostní nebo volnou paměť (Yosifovich P., Ionescu A., 2017).

6.4.3 Disk

Karta *Disk* naopak zobrazuje informace o vstupně-výstupních operacích v jednotlivých souborech způsobem, který usnadňuje při identifikaci nejvíce přistupovaných, nejvíce zapisovaných nebo nejvíce čtených souborů v systému, přičemž tyto výsledky lze dále filtrovat podle procesů (Yosifovich P., Ionescu A., 2017).

6.4.4 Síť

Na kartě *Síť* jsou zobrazena aktivní síťová připojení, procesy, které je vlastní, a množství dat, které přes ně proudí. Na základě těchto informací je možné sledovat síťovou aktivitu na pozadí, kterou by bylo jinak obtížné odhalit. Kromě toho také zobrazuje připojení TCP, která jsou v systému právě aktivní, uspořádaná podle procesů, ve kterých jsou uvedeny údaje jako například vzdálený či místní port, adresa a latence paketů (Yosifovich P., Ionescu A., 2017). Nakonec zobrazuje seznam naslouchajících portů, což správci umožňuje zjistit, které služby nebo aplikace právě čekají na připojení na daném portu (Yosifovich P., Ionescu A., 2017). Pro každý port a proces se také zobrazuje protokol a zásady brány firewall.

7 Docker

7.1 Úvod

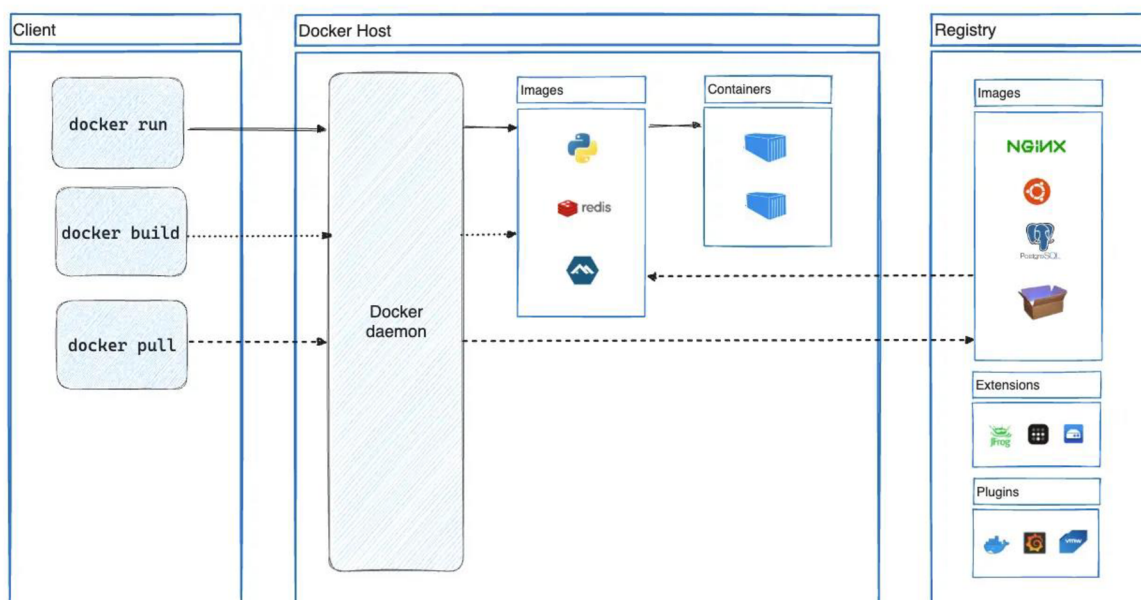
Na úvod je potřeba zmínit, že tato celá kapitola je nepřímo citována z oficiálních stránek Dockeru, kde autorem je samotná společnost vlastní platformu Docker, a taky z části nepřímo citovaná z webových stránek Microsoft Learn, kde společnost Microsoft poskytuje volně dostupné webové články či studijní programy.

7.2 Platforma Docker

Docker umožňuje zabalit a spustit aplikaci ve volně izolovaném prostředí zvaném kontejner, nezávisle na operačním systému. Izolace a zabezpečení umožňují na daném hostiteli spustit mnoho kontejnerů současně. Kontejnery jsou odlehčené a obsahují vše potřebné ke spuštění aplikace, takže se nemusíte spoléhat na to, co je nainstalováno na hostiteli. Kontejnery můžete při práci sdílet a mít jistotu, že každý, s kým je sdílíte, dostane stejný kontejner, který funguje stejným způsobem. Docker využívá jádro linuxového operačního systému k poskytování těchto funkcí.

7.3 Architektura Dockeru

Docker používá architekturu klient-server. Klient Docker komunikuje s démonem Docker, který se stará o sestavování, spouštění a distribuci kontejnerů Docker. Klient Docker a démon mohou běžet na stejném systému nebo můžete klienta Docker připojit ke vzdálenému démonu Docker. Klient Docker a démon spolu komunikují pomocí rozhraní REST API, přes sokety UNIX nebo síťové rozhraní. Dalším klientem Docker je Docker Compose, který umožňuje pracovat s aplikacemi složenými ze sady kontejnerů.



Obrázek 1 Grafická reprezentace architektury Dockeru (zdroj: <https://docs.docker.com/get-started/overview/>)

7.4 Démon Docker

Démon Docker (dockerd) naslouchá požadavkům rozhraní Docker API a spravuje objekty Docker, jako jsou images, kontejnery, sítě a svazky. Démon může také komunikovat s jinými démony a spravovat služby Docker.

7.5 Klient Docker

Klient Docker (docker) je hlavní způsob, jakým mnoho uživatelů Dockeru komunikuje s Dockerem. Při použití příkazů, jako je například `docker run`, klient odešle tyto příkazy dockerd, který je provede. Příkaz docker využívá rozhraní Docker API. Klient Docker může komunikovat s více než jedním démonem.

7.6 Pracovní plocha Docker

Docker Desktop je snadno instalovatelná aplikace pro prostředí Mac, Windows nebo Linux, která umožňuje vytvářet a sdílet kontejnerové aplikace a mikroslužby. Docker Desktop obsahuje démona Docker (dockerd), klienta Docker (docker), Docker Compose, Docker Content Trust, Kubernetes a Credential Helper. Další informace naleznete v části Docker Desktop.

7.7 Registry nástroje Docker

Registr je podobný knihovně, kde jsou image uloženy a dostupné pro vytváření kontejnerů pro spouštění služeb nebo webových aplikací. Existují privátní registry

Dockeru místně i ve veřejném cloudu. Docker Hub je veřejný registr, který spravuje Docker, společně s řešením Důvěryhodného registru Dockeru a nabízí Azure Container Registry na podnikové úrovni. AWS, Google a další mají také registry kontejnerů.

Vložení image do registru umožňuje ukládat statické a neměnné bity aplikací, včetně všech jejich závislostí na úrovni architektury. Tyto image se pak dají spravovat a nasazovat ve více prostředích, a proto poskytují konzistentní jednotku nasazení.

7.8 Image (obraz)

Image je šablona pouze pro čtení s pokyny pro vytvoření kontejneru Docker. Image je často založen na jiném obrazu s určitými dalšími úpravami. Můžete například vytvořit obraz, který je založen na obrazu ubuntu, ale nainstaluje webový server Apache a vaši aplikaci, jakož i konfigurační údaje potřebné pro spuštění vaší aplikace.

Můžete vytvářet vlastní obrazy nebo můžete používat pouze ty, které vytvořili jiní a které jsou zveřejněny v registru. Chcete-li vytvořit vlastní obraz, vytvoříte soubor Dockerfile s jednoduchou syntaxí pro definování kroků potřebných k vytvoření obrazu a jeho spuštění. Každý pokyn v souboru Dockerfile vytváří vrstvu obrazu. Když změníte soubor Dockerfile a obnovíte obraz, obnoví se pouze ty vrstvy, které se změnily. Díky tomu jsou obrazy ve srovnání s jinými virtualizačními technologiemi tak lehké, malé a rychlé.

7.8.1 Image Hello-world

Docker image *hello-world* je základním příkladem minimalizované kontejnerizace s Dockerem (Farhan Hasin Chowdhury, 2020). Image *hello-world* obsahuje jediný program, který byl zkompileován ze souboru *hello.c* (Docker, 2024). Tento program je zodpovědný za výpis zprávy, kterou uživatel vidí na svém terminálu po spuštění tohoto image za pomoci příkazu *docker run hello-world* (Docker, 2024).

7.8.2 Image Nginx

Docker image *nginx* (vyslovuje se jako *engine x*) je oficiální Docker image pro webový server Nginx. Tento image slouží jako reverzní proxy server s otevřeným zdrojovým kódem pro HTTP, HTTPS, SMTP, POP3 a IMAP protokoly (Docker, 2024). Nginx je navržen s důrazem na vysokou konkurenci, vysoký výkon a nízkou spotřebu paměti

(Docker, 2024). Je dostupný na Linuxu a je licencován pod BSD a také existují verze i na macOS a MS Windows (Docker, 2024).

7.8.2.1 Localhost

Pojem localhost spadá do počítačových sítí a dalo by se říct, že to je něco jako *počítač*, na kterém je spuštěn lokální server. Je to výchozí název používaný pro navázání spojení s počítačem pomocí síťové adresy zpětné smyčky (loopback) (Domantas G., 2023). Smyčková adresa má výchozí IP (127.0.0.1) a je užitečná pro testování programů na počítači, aniž by se informace odesílaly přes internet (Domantas G., 2023). Při volání IP adresy z počítače se obvykle kontaktuje jiný počítač prostřednictvím internetu, ale v případě adresy zpětné smyčky se volá na localhost, tedy na svůj počítač (Domantas G., 2023).

7.8.2.2 Port 80

Port 80 je číslo portu, který je přidělen k běžně používanému internetovému komunikačnímu protokolu HTTP (Hypertext Transfer Protocol) (Rouse M., 2023). Jedná se o výchozí síťový port používaný k odesílání a přijímání nešifrovaných webových stránek (Rouse M., 2023). Například, aby se klient dokázal připojit k serveru a zadat daný požadavek, musí znát jak IP adresu serveru, tak i to, která síťová služba bude použita k přenosu dat (Rouse M., 2023).

7.8.2.3 Jazyk HTML

Jazyk HTML, z anglického jazyka Hypertext Markup Language, je hypertextový značkovací jazyk určený k vytváření jednoduchých webových stránek. Jazyk HTML využívá tzv. definované značky, přesněji tagy. Tagům se přiřazují atributy a hodnoty, jež jednotlivým prvkům stránky přiřazují určitou roli (Kod'ousková B., 2021). Prostřednictvím HTML tagů tak se například určuje, kde budou odkazy a kam budou odkazovat nebo kde bude obrázek a odkud je bude prohlížeč čerpat (Kod'ousková B., 2021).

7.8.3 Image MySQL

Docker image *mysql* je lehký, samostatně spustitelný balíček, který obsahuje vše potřebné ke spuštění systému MySQL (Docker, 2024). Image zahrnuje samotný software serveru MySQL, prostředí pro práci s databází, systémové nástroje, systémové knihovny a jeho nastavení (Docker, 2024).

Image *mysql* je vytvořen speciálně pro platformu Linux, ve kterém Docker pracuje (Docker, 2024). Je udržován týmem, který má na starosti MySQL ve společnosti Oracle, a je aktualizován při zveřejnění nových údržbových verzí a větších dosažených milníků při vývoji (Docker, 2024).

7.8.3.1 Jazyk SQL

Jazyk SQL (Structured Query Language) je nejrozšířenější jazyk pro práci s relačními databázemi na světě. Byl vyvinut počítačovými vědci IBM v 70. letech na základě relačního datového modelu definovaným E. F. Coddem v roce 1970 (Chamberlin D., 2009). Jazyk SQL je určen k manipulaci s daty v relačních systémech pro správu databází (RDBMS). Umožňuje uživatelům vytvářet, číst, upravovat a mazat data, která jsou uložena v tabulkové formě, a pomocí jazyka SQL lze provádět úlohy, jako je vytváření databáze, mazání databáze, načítání datových řádků, úprava a mazání datových řádků.

Jazyk SQL je také velmi efektivní při rychlém vyhledávání velkého množství dat. Poskytuje výkonnou sadu nástrojů pro filtrování, třídění a agregaci dat, což z něj činí základní nástroj pro analýzu dat. SQL také sjednocuje a aktualizuje data dotazováním při správě databáze, například vytvářením a modifikací tabulek a omezení přístupu k datům (Chamberlin D., 2009).

7.8.3.2 Program MySQL Workbench

MySQL Workbench, vyvinutý společností Oracle, je multiplatformní open-source nástroj pro návrh relačních databází, který rozšiřuje funkčnost a usnadňuje vývoj MySQL a SQL (Sulemani M., 2020). Integruje návrh, vývoj, tvorbu, správu a údržbu SQL a nabízí grafické rozhraní pro práci s databázemi ve strukturované podobě (Sulemani M., 2020).

Tento nástroj je přínosný zejména pro správce databází, databázové architekty a vývojáře MySQL (Sulemani M., 2020). Poskytuje možnosti manipulace s databázovými modely, jako je vytváření grafického modelu, zpětné inženýrství pro živé databáze do modelů (datové modelování), dopředné inženýrství modelu do skriptové/živé databáze.

7.9 Kontejnery

Kontejner se používá k zabalení aplikace spolu se všemi jejími závislostmi a abstrakci z hostitelského operačního systému, ve kterém se bude spouštět. Kontejnery poskytují jednoduché vývojové a běhové prostředí, ve kterém se dají aplikace během vývoje snadno spouštět a sdílet. Nejen že je kontejner izolovaný od hostitelského operačního systému, je také izolovaný od jiných kontejnerů. Izolované kontejnery poskytují virtuální modul runtime, který může také zlepšit zabezpečení a spolehlivost aplikací, které v nich běží.

Tradičně se softwarová aplikace vyvíjí tak, aby běžela na podporované platformě procesoru, hardwaru a operačního systému. Softwarové aplikace obvykle vyžadují další kódování, aby poskytovaly podporu pro různé platformy runtime. Díky tolika různorodým výpočetním systémům je potřeba efektivnější platforma pro vývoj a správu softwaru, která podporuje přenositelnost mezi několika výpočetními prostředími. Kontejnery pomáhají zajistit takovou přenositelnost.

7.10 Porovnání mezi virtualizací a Dockerem

Virtuální počítače i kontejnery jsou virtualizační technologie používané k poskytování izolovaných a přenosných výpočetních prostředí pro aplikace a služby.

7.10.1 Virtuální počítače

Virtuální počítače simulují celý počítač, včetně virtualizovaného hardwaru, operačního systému, uživatelského režimu a vlastního režimu jádra. Virtuální počítače jsou poměrně agilní a poskytují obrovskou podporu pro aplikace; Virtuální počítače ale bývají velké a využívají prostředky hostitelského počítače.

7.10.2 Docker kontejner

Kontejnery (jak bylo popsáno dříve) vycházejí z jádra operačního systému hostitele a obsahují proces izolovaného uživatelského režimu pro zabalenou aplikaci. Díky tomu jsou kontejnery velmi jednoduché a rychlé ke spuštění. Kromě virtuálních počítačů se kontejnery dají snadno přesunout z vývoje do produkčního prostředí. Vzhledem k tomu, že operační systém není úplně stejný, ne všechny aplikace spouštějí stejné (nebo vůbec) v prostředí kontejneru.

7.10.3 Podobnosti a rozdíly

Následující tabulka shrnuje podobnosti a rozdíly mezi funkcemi virtuálních počítačů a kontejnerů:

Funkce	VM	Kontejner
Izolace	Poskytuje úplnou izolaci od hostitelského operačního systému a dalších virtuálních počítačů.	Poskytuje jednoduchou izolaci od hostitele a dalších kontejnerů.
Operační systém	Spustí kompletní operační systém včetně jádra.	Spustí jenom část uživatelského režimu operačního systému.
Kompatibilita a hostů	Možnost spustit jakýkoli podporovaný operační systém na virtuálním počítači	Kontejnery Windows vyžadují odpovídající verze operačního systému hostitele a kontejneru. (Další podrobnosti a alternativy najdete v režimu izolace.
Nasazení	Nasazené pomocí správce Technologie Hyper-V nebo jiných nástrojů pro správu virtuálních počítačů.	Nasazené a spravované pomocí Dockeru nebo jiného modulu runtime kontejneru. Pomocí orchestrátoru, jako je Kubernetes, je možné nasadit více kontejnerů.
Trvalé úložiště	Používá soubory virtuálního pevného disku nebo sdílenou složku SMB (Server Message Block).	Data se neuchovávají, pokud už kontejner neexistuje. Pokud chcete zachovat data v kontejneru s Windows Serverem, je nutné použít trvalé úložiště.
Vyrovňování zatížení a vysoká dostupnost	Pomocí clusteru s Windows s podporou převzetí služeb při selhání nebo nástroje pro vyrovňování zatížení Windows můžete podle potřeby přesouvat	Pomocí orchestrátoru kontejnerů můžete automaticky spouštět a zastavovat kontejnery,

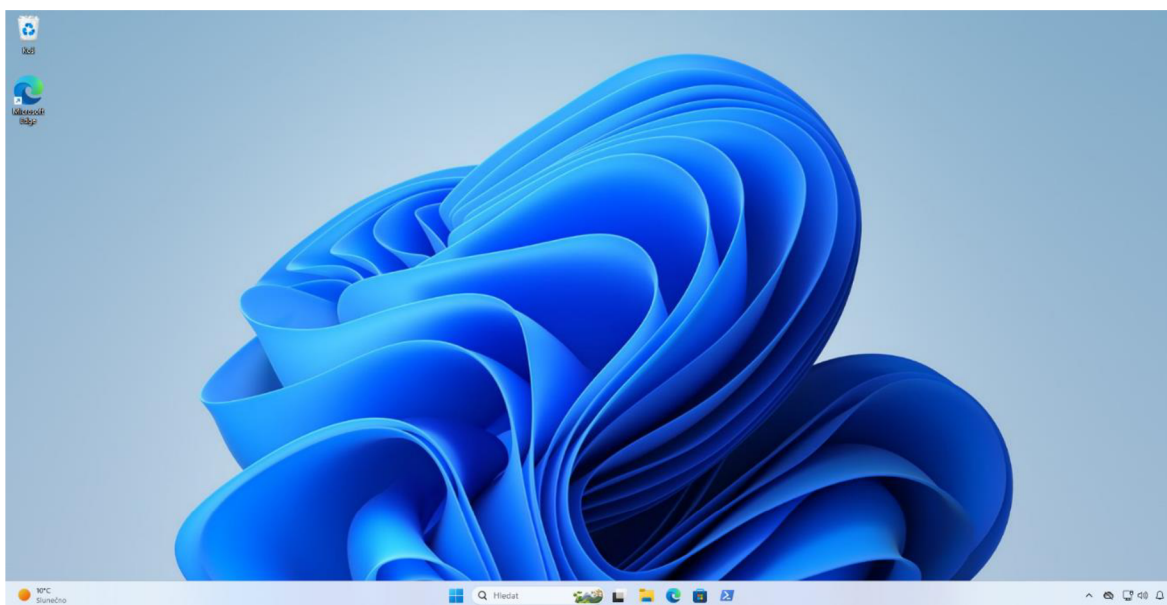
	virtuální počítače a podporovat vyšší svazek.	automaticky škálovat a podporovat další provoz.
Sítě	Používá virtuální síťové adaptéry.	V závislosti na scénáři jsou k dispozici různé možnosti sítě. Výchozí je síť překladu adres (NAT), která používá interní přepínač a mapuje porty z hostitele na kontejner.

Tabulka 1 Rozdíly mezi funkcemi virtuálních počítačů a kontejnerů (zdroj: <https://learn.microsoft.com/cs-cz/training/modules/run-containers-windows-server/3-list-differences-between-containers-vms>)

8 Praktická cvičení

8.1 Úvod

Praktická část této bakalářské práce se zabývá správou a konfigurací operačního systému Windows 11, nejnovější verze stěžejního desktopového operačního systému společnosti Microsoft. Windows 11 přináší mnoho nových funkcí a možností, které by měli znát uživatelé Windows, zejména ti, kteří pracují v hybridním pracovním prostředí. Tato část práce má za cíl poskytnout praktické znalosti a dovednosti pro efektivní využití Windows 11 v běžných i pokročilých scénářích.



Obrázek 2 Pracovní plocha ve Windows 11 (zdroj: autor)

V této praktické části se bakalářská práce zaměří na konkrétní aspekty správy a konfigurace operačního systému Windows 11. Snaha práce není pouze poskytnout teoretické znalosti, ale především vytvořit praktické prostředí pro studenty, kde si mohou osvojit klíčové dovednosti a postupy v oblasti správy a konfigurace tohoto moderního operačního systému.

Po vzájemné diskusi s vedoucím práce, a zároveň za účelem nasdílení a poskytnutí všech vytvořených cvičení, byl založen digitální veřejný repositář na portálu GitHub. Repositář obsahuje tři různá cvičení rozdělena do tří jednotlivých adresářích. V jednotlivých adresářích uživatel nalezne jednu dokumentaci se zadáním a druhou dokumentaci se řešením. Odkaz na digitální repositář je přiložen v seznamu příloh na konci práce.

8.2 První cvičení – Windows Subsystem for Linux

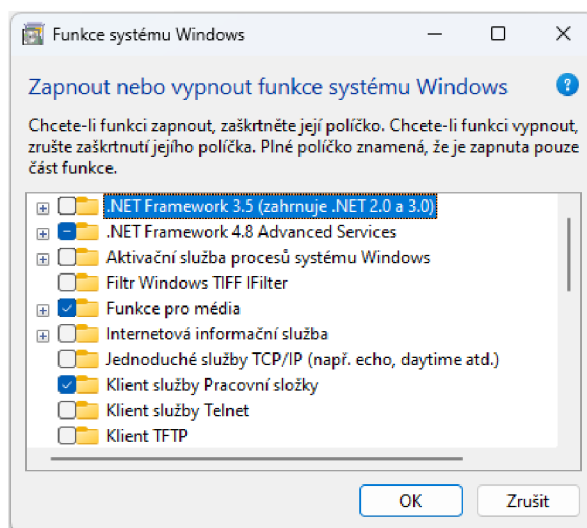
8.2.1 Úvod

První cvičení praktické části práce se zaměřuje na *Windows Subsystem for Linux* (WSL), což je technologie, která umožňuje spouštět Linuxové distribuce a aplikace přímo na Windows bez nutnosti používat virtuální stroje nebo dualboot. WSL je užitečný nástroj pro vývojáře, kteří potřebují pracovat s oběma operačními systémy současně. V tomto cvičení se naučíte, jak nainstalovat a spustit WSL na daném počítači, zkoumáním základních funkcí, jak nainstalovat Linuxovou distribuci Ubuntu, jak používat Linuxové příkazy a aplikace v rámci Windows, jak vytvořit spustitelné skripty, a jak nainstalovat a připojit se ke grafickému rozhraní XFCE přes vzdálenou plochu.

8.2.2 Zapnutí nebo vypnutí funkce systému Windows

Pro správné fungování *Windows Subsystem for Linux* (WSL) je potřeba zapnout některé funkce virtualizace ve Windows. Tento krok je nutný, protože WSL využívá virtuální stroj pro spouštění Linuxových distribucí na Windows. Následující kroky ukážou, jak zkontrolovat a případně povolit virtualizaci prostřednictvím *Zapnout nebo vypnout funkce systému Windows*:

1. Otevřete *Ovládací panely* a vyberte *Programy*
2. Klikněte na *Zapnout nebo vypnout funkce systému Windows*
3. V okně, které se otevře, zaškrtněte následující funkce: *Hyper-V*, *Virtual Machine Platform* a *Windows Subsystem for Linux*
4. Klikněte na *OK* a restartujte počítač



Obrázek 3 Okno Zapnout nebo vypnout funkce systému Windows (zdroj: autor)

Po restartu by měla být virtualizace úspěšně povolena a WSL by mělo nyní pracovat správně ve Windows 11. Tato nastavení jsou klíčová pro zajištění, že WSL může vytvářet izolovaná linuxová prostředí v systému.

8.2.3 Instalace WSL

Přes utilitu *Hledat* (hledací okno nebo ikonka lupy na hlavním panelu) se vyhledá *PowerShell* nebo *Příkazový řádek* a spustí se jeden z nich jako správce. Po otevření se nainstaluje WSL. Příkaz pro instalaci Windows Subsystem for Linux je `wsl --install`, tento příkaz zapne potřebné komponenty pro WSL, stáhne a nainstaluje nejnovější Linuxové jádro a nastaví WSL verzi 2 jako výchozí.

Po provedení těchto kroků by mělo být WSL úspěšně nainstalováno v systému. Nyní se může začít vytvářet a spravovat linuxová prostředí přímo v rámci Windows 11.

8.2.4 Dostupné distribuce Linuxu

Příkaz `wsl --list --online` je užitečný pro zjištění, jaké Linuxové distribuce jsou dostupné pro instalaci prostřednictvím online obchodu (Microsoft, 2023). Přesný seznam distribucí závisí na aktuální nabídce a aktualizacích ze strany Microsoftu a vývojářů příslušných distribucí. Tento příkaz vypíše název, verzi a velikost každé distribuce, kterou lze stáhnout a nainstalovat pomocí příkazu `wsl --install -d <Název distribuce>` (Microsoft, 2023). Tento krok je důležitý pro výběr vhodné distribuce podle potřeb a preferencí.

Seznam může obsahovat různé distribuce, od klasických linuxových distribucí, jako je Ubuntu, až po specializované distribuce zaměřené na konkrétní účely.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\admin1> wsl --list --online
Následuje seznam platných distribucí, které je možné nainstalovat.
Nainstalovat pomocí příkazu „wsl.exe --install <Distro>“.

NAME                                FRIENDLY NAME
-----                                -
Ubuntu                               Ubuntu
Debian                               Debian GNU/Linux
kali-linux                           Kali Linux Rolling
Ubuntu-18.04                          Ubuntu 18.04 LTS
Ubuntu-20.04                          Ubuntu 20.04 LTS
Ubuntu-22.04                          Ubuntu 22.04 LTS
OracleLinux_7_9                       Oracle Linux 7.9
OracleLinux_8_7                       Oracle Linux 8.7
OracleLinux_9_1                       Oracle Linux 9.1
openSUSE-Leap-15.5                    openSUSE Leap 15.5
SUSE-Linux-Enterprise-Server-15-SP4   SUSE Linux Enterprise Server 15 SP4
SUSE-Linux-Enterprise-15-SP5         SUSE Linux Enterprise 15 SP5
openSUSE-Tumbleweed                   openSUSE Tumbleweed
PS C:\Users\admin1>
```

Obrázek 4 Vygenerovaný seznam všech dostupných distribucí Linuxu (zdroj: autor)

8.2.4.1 Instalace distribuce Ubuntu

Distribuce Ubuntu je jednou z mnoha linuxových distribucí, které jsou k dispozici v rámci Windows Subsystem for Linux (WSL). Jedná se o plnohodnotnou distribuci Ubuntu, ale je upravena pro spolupráci s WSL. Mohou se používat stejné příkazy, nástroje a aplikace jako na běžném Ubuntu, ale také integrovat s Windows souborovým systémem, terminálem, editory kódu a dalšími funkcemi.

Instalace Ubuntu na WSL je velmi jednoduchá. Pokud je zapnutá funkce WSL a je nainstalovaný nejnovější Linuxový kernel, poté se může stáhnout distribuce Ubuntu z Microsoft Storu a spustit jej z nabídky Start, nebo přes příkazovou řádku.

8.2.4.2 Vytváření uživatele

Po samotné instalaci distribuce Ubuntu WSL konzole ihned vyzve k vytvoření uživatelského jména a hesla, které bude sloužit jako výchozí uživatel a administrátor zároveň.

V průběhu lze přidat dalšího uživatele, kde se zadá příkaz `sudo adduser novy_uzivatel`, přičemž `novy_uzivatel` je preferované uživatelské jméno. Systém pak vyzve k zadání a potvrzení nového hesla pro uživatele. Další informace, jako jsou jméno, telefonní číslo a další, mohou být vyplněny, ale nejsou

povinné. Po dokončení těchto kroků je nový uživatel úspěšně vytvořen. Pokud je třeba později změnit heslo uživatele, lze použít příkaz `sudo passwd novy_uzivatel`, který opět vyzve k zadání a potvrzení nového hesla. Po dokončení těchto kroků je heslo uživatele aktualizováno.

8.2.5 Práce s příkazovou řádkou

Práce s příkazovou řádkou (konzolí) ve Windows Subsystem for Linux (WSL) Ubuntu je klíčová pro efektivní využití tohoto systému. WSL Ubuntu umožňuje uživatelům spouštět různé linuxové distribuce přímo ve Windows, což zahrnuje plnou podporu příkazové řádky (Microsoft, 2022).

Příkazová řádka je prostředí, kde uživatelé mohou zadávat textové příkazy, které systém vykoná. Tyto příkazy mohou zahrnovat instalaci a aktualizaci softwaru, práci se soubory a adresáři, a mnoho dalších operací (Microsoft, 2022).

8.2.5.1 Spouštění programů s právy superuživatele

Příkaz `sudo`, což je zkratka pro SuperUser DO, je klíčovým nástrojem v Linuxu, který umožňuje uživatelům spouštět programy s bezpečnostními právy superuživatele. Tento příkaz je nezbytný pro provádění operací, které vyžadují zvláštní oprávnění, jako je například úprava systémových souborů (Knowledge Base, 2023).

Příkaz `sudo` funguje tak, že dočasně zvyšuje oprávnění uživatele, což umožňuje vykonávat citlivé úkoly bez nutnosti přihlášení jako superuživatel (Aleksic, 2020). To je užitečné, protože Linux ve výchozím nastavení omezuje přístup k určitým částem systému, aby zabránil kompromitaci citlivých souborů (Aleksic, 2020).

8.2.5.2 Třídění a seřazování textu

Příkaz `cat` je zase používán pro výpis obsahu souborů, spojení textových souborů a další operace. Výpis seřazeného textu je často realizován pomocí příkazu `sort`, který řadí řádky textu v abecedním nebo numerickém pořadí.

Kombinace těchto příkazů pomocí symbolu *pipe* (`|`) umožňuje vytvářet silné nástroje pro manipulaci s daty. *Pipe* předává výstup jednoho příkazu jako vstup do dalšího. Například příkaz `cat`, se může použít k výpisu obsahu souboru, poté použít *pipe* k předání tohoto výstupu do příkazu `sort`, který seřadí řádky textu.

8.2.5.3 Navigování v adresářích

Navigace v adresářích je základním kamenem práce s příkazovou řádkou v Linuxu. Příkaz *ls*, což je zkratka pro *list*, se používá k výpisu obsahu adresářů. Tento příkaz poskytuje cenné informace o souborech, adresářích a jejich attributech. Příkaz *cd*, což je zkratka pro *change directory*, umožňuje uživatelům změnit aktuální pracovní adresář. Příkaz *cd ..* se používá k přesunu do nadřazeného adresáře (Garn, 2022). Tento příkaz je užitečný pro rychlé navigování zpět v adresářové struktuře (Garn, 2022). K otevření složky přes pevnou cestu, například */Plocha/Fotky*, můžete použít příkaz *cd /Plocha/Fotky* (Garn, 2022). Tento příkaz vás přesune do adresáře */Plocha/Fotky*, bez ohledu na to, kde se aktuálně nacházíte (Garn, 2022).

8.2.5.4 Textová rozhraní

Textový editor *nano* je jednoduchý a přátelský nástroj, který je součástí Unixových a Linuxových operačních systémů (Linuxize, 2023) (GNU manual, 2023). Jeho cílem je poskytnout uživatelům snadno použitelný textový editor, který zahrnuje všechny základní funkce, které byste očekávali od běžného textového editoru, jako je zvýrazňování syntaxe, více bufferů, vyhledávání a nahrazování s podporou regulárních výrazů, kontrola pravopisu, kódování UTF-8 a další (Linuxize, 2023) (GNU manual, 2023). Na rozdíl od jiných příkazových řádkových editorů, jako je *vim*, *nano* je bezmódový editor, což znamená, že můžete začít psát a upravovat text ihned po otevření souboru (GNU manual, 2023).

8.2.6 Vytváření skriptů

Vytváření skriptů v Ubuntu je základním nástrojem pro automatizaci úloh a zjednodušení složitých sekvencí příkazů (Vitux, 2022) (Vitux, 2024). Skript je soubor, který obsahuje sekvenci příkazů, které jsou vykonávány v pořadí, jak jsou napsány (Vitux, 2024). Tyto skripty mohou být použity k automatizaci různých úloh, jako je například správa systému, zpracování dat nebo dokonce vývoj softwaru (Vitux, 2022) (Vitux, 2024).


```
ubuntu@admin: /mnt/c/Users x + v
GNU nano 6.2 helloworld.txt S
#!/bin/bash
echo "Hello World!"
echo "Zadejte Vaše jméno: "
read jmeno
echo "Zadejte Váš věk: "
read vek
echo "Jmenuji se $jmeno a je mi $vek let."
if [ $vek -gt 17 ];
then echo "Ano"
else echo "Ne"
fi

[ Soft wrapping of overlong lines enabled ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

Obrázek 5 Ukázkový skript ve rozhraní NANO (zdroj: autor)

8.2.6.1 Proměnné

Proměnné v *bash* skriptech jsou užitečné pro ukládání a manipulaci s daty. Proměnná může obsahovat řetězec, číslo nebo výsledek příkazu (McKay, 2023). Proměnné jsou definovány pomocí rovnítkového znaménka (=) a jejich hodnoty jsou přístupné pomocí dolarového znaménka (\$) (McKay, 2023). Příkazy *read* a *echo* jsou často používány v kombinaci s proměnnými *read* je používán k načtení vstupu od uživatele do proměnné, zatímco *echo* je používán k vypsání hodnoty proměnné (McKay, 2023).

8.2.6.2 Podmínka if

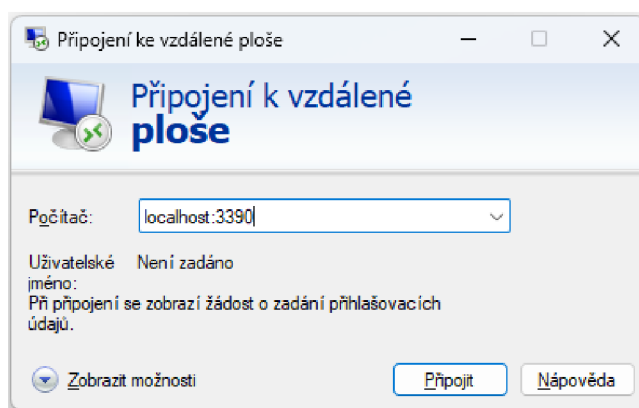
Podmínka *if* umožňuje skriptu vyhodnotit určitou podmínku a na základě toho rozhodnout, který ze dvou nebo více bloků příkazů se má vykonat. To v podstatě umožňuje zvolit odpověď na výsledek, ke kterému daná podmínka vyhodnotí.

8.2.6.3 Operační výrazy

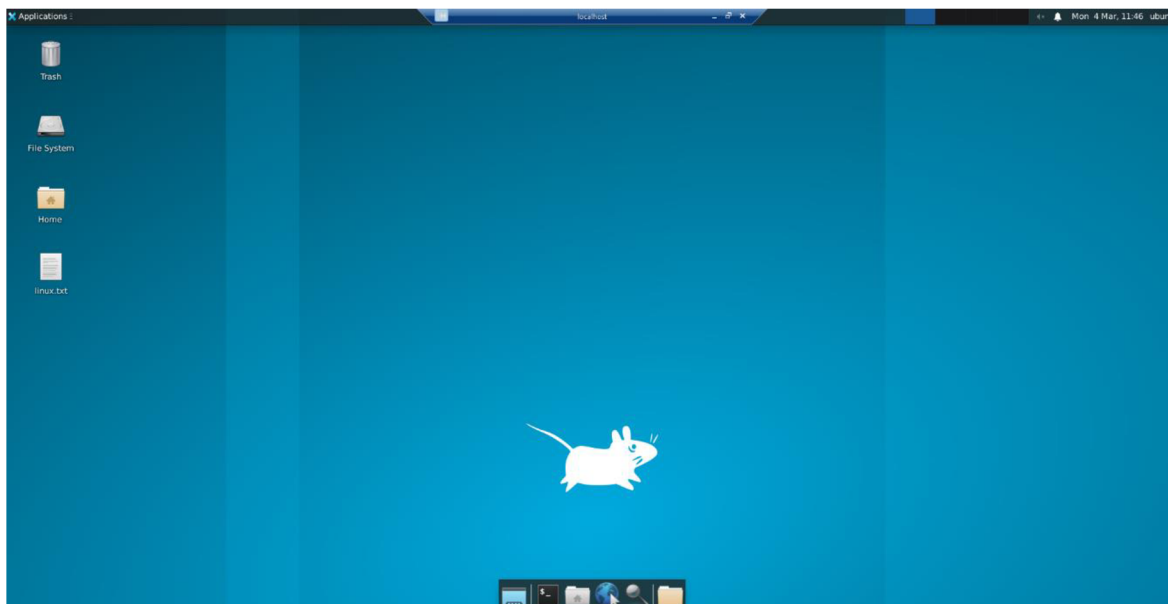
Operační výrazy, jako jsou *-eq* (==), *-le* (<=), *-gt* (>), jsou používány k porovnání dvou hodnot. Například, *-eq* kontroluje rovnost, *-le* kontroluje, zda je první hodnota menší nebo rovna druhé. Tyto výrazy jsou často používány v kombinaci s podmínkou *if* k vytvoření podmíněné logiky.

8.2.7 Instalace grafického rozhraní XFCE

Pro spuštění distribuce Ubuntu přes grafické rozhraní pomocí Windows Subsystem for Linux (WSL) je třeba nejprve nainstalovat balíčky *xfce4*, *xfce4-goodies* a *xrdp* (Pablinux, 2023). Balíček *xfce4* je lehké desktopové prostředí pro unixové operační systémy. Balíček *xfce4-goodies* obsahuje doplňkové programy a grafické rozhraní pro desktopové prostředí XFCE. Balíček *xrdp* je open-source Remote Desktop Protocol server, který umožňuje připojení k vzdálenému počítači (Pablinux, 2023). Po instalaci těchto balíčků je třeba spustit server *xrdp* a poté se připojit přes vzdálenou plochu do grafického rozhraní.



Obrázek 6 Okno Připojení ke vzdálené ploše (zdroj: autor)



Obrázek 7 Grafického rozhraní XFCE (zdroj: autor)

8.2.8 Sdílení souborů mezi Windows 11 a distribucí Ubuntu

Jako další z mnoha vlastností na WSL je vzájemné sdílení a přístup k souborového systému mezi Windows a distribucí Ubuntu. Například cesta na Windows plochu

z Ubuntu konzole je obvykle `/mnt/c/Users/<uživatelské jméno>/Desktop`. To znamená, že lze vytvářet, upravovat a mazat soubory z tohoto umístění přímo za pomoci Ubuntu konzole. Například linuxovým příkazem `touch /mnt/c/Users/<uživatelské jméno>/Desktop/nový_soubor.txt` se vytvoří nový textový soubor a jeho umístění je na Windows ploše.

8.2.9 Závěr

Po dokončení tohoto cvičení si student prohloubí svoje znalosti v oblasti operačních systémech. Naučí se další možnosti virtualizace prostřednictvím Powershellu na Windows 11. V průběhu cvičení si student také zkusí nainstalovat linuxové prostředí a vyzkouší si linuxové příkazy, pomocí kterých se v něm naviguje.

8.3 Druhé cvičení – Docker

8.3.1 Instalace Dockeru

Instalace Dockeru na Windows 11 vyžaduje několik kroků. Před instalací je důležité zajistit, aby systém splňoval všechny požadavky. To zahrnuje mít nejnovější a zároveň 64bitovou verzi Windows 11, zapnutou funkci WSL 2, minimálně 4 GB systémové RAM a povolenou hardwarovou virtualizaci v BIOSu.

Po dokončení instalace je doporučeno ověřit, zda Docker funguje za pomoci příkazu `docker info`. Pokud se proces instalace zdařil, vypíší se všeobecné informace týkající se instalace Dockeru. Zobrazené informace zahrnují verzi jádra, počet kontejnerů, images a apod (Docker, 2024).

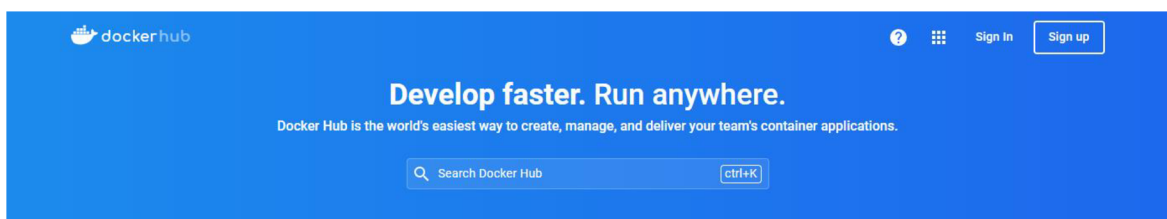
8.3.2 Práce s Dockerem

Pro práci s Dockerem se primárně používá příkazová řádka nebo *Powershell*. K dispozici je také i grafického rozhraní, které sice není určené pro psaní kódů, ale spíše slouží jako přehledný seznam všech dostupných kontejnerech, image a dalších funkcí.

8.3.3 Docker Hub

Docker Hub je služba poskytovaná společností Docker pro vyhledání a sdílení tzv. *container images*, které si může kdokoli stáhnout do jeho lokálního úložiště v Dockeru za pomoci příkazu `docker pull` (Docker, 2023). Je to zároveň největší úložiště těchto *container images* s řadou veřejných i soukromých zdrojů obsahu,

vývojářských kontejnerů, open source projektů a nezávislých softwarových prodejců (Docker, 2023).



Obrázek 8 Docker Hub (zdroj: autor)

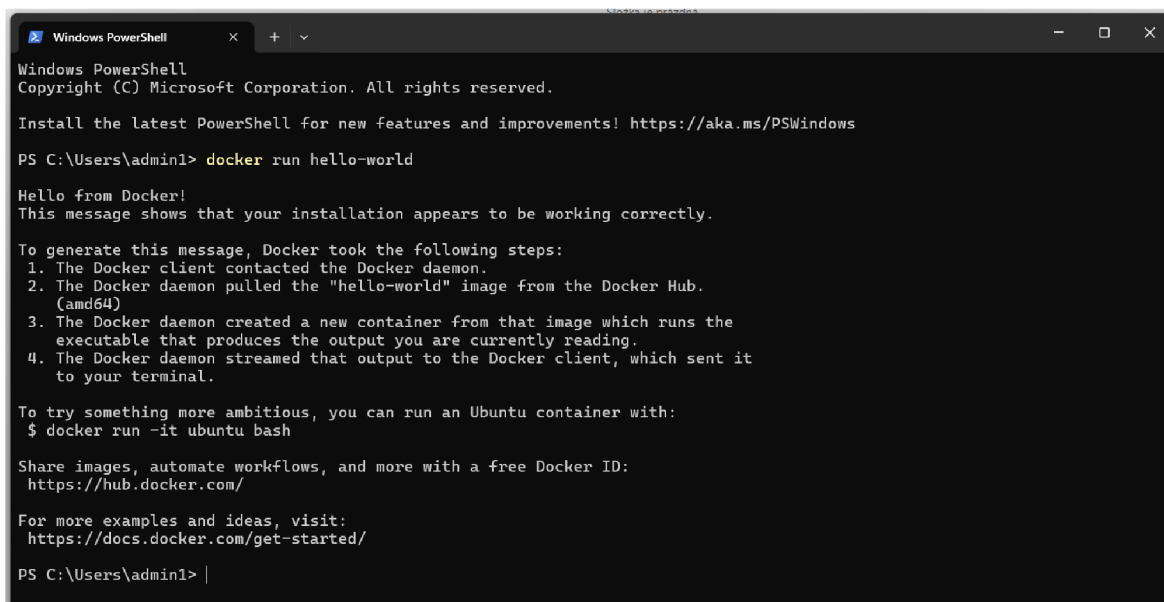
8.3.3.1 Stáhnutí image v kontejneru

Jak je již výše zmíněno, s daným veřejným imagem lze pracovat dvěma způsoby. První praktičtější způsob je si nejdříve stáhnout image příkazem `docker pull`. Image se stáhne do lokálního úložiště a lze ho zobrazit v image listu příkazem `docker ps -a`. Následně se musí image spustit příkazem `docker run`. Druhý je právě, že se image už rovnou spustí, ale daný images se musí nejdříve nakonfigurovat před jejich spuštěním. Pokud se image spustí a nebyl dříve uživatelem stáhnutý, tedy není přístupný lokálně, dojde k jeho stažení a následnému spuštění.

8.3.3.2 Stáhnutí image hello-world

Základním nástrojem je veřejná knihovna Docker Hub, která v těchto cvičeních slouží jako vyhledávání potřebných images, u kterých je také vytvořena oficiální dokumentace, která popisuje, jak příslušný image používat a případně postupovat při jeho instalaci.

Tímto imagem se ověří správná funkcionality Dockeru, zda instalace proběhla v pořádku, ale také především si uživatel vyzkouší, jak pracovat s Dockerem i práci se samotným Docker Hubem. Tedy prvním krokem je vždy si daný image najít – v tomto případě image `hello-world`. Poté si ho stáhnout příkazem `docker pull hello-world` a následně spustit kontejner s obrazem příkazem `docker run hello-world`. Tento výstup potvrzuje, že Docker klient úspěšně komunikoval s Docker démonem, který stáhl obraz `hello-world` z Docker Hubu. Následně z něj vytvořil nový kontejner a spustil v něm příslušný spustitelný soubor.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\admin1> docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

PS C:\Users\admin1> |
```

Obrázek 9 Spuštění kontejneru hello-world v Dockeru (zdroj: autor)

8.3.4 Připojení k webovému serveru přes image Nginx

Pro procvičení základních znalostí s Nginx image, je jedno cvičení určené pro vytvoření lokálního serveru na vlastním počítači a k následnému připojení k localhostu přes webový prohlížeč.

Stejně jako u image *hello-world* si musí student nejdříve vyhledat image *nginx* na Docker Hubu a obeznámit se, co je to za image a jakou má funkčnost. V dokumentaci je obsažen postup, jak docílit vytvoření vlastního lokálního serveru. Tedy nejdříve si student musí image stáhnout příkazem `docker pull nginx` a následně spustí server příkazem `docker run -p 8000:80 nginx`. Konkrétně tímto příkazem, `-p 8000:80` Docker kontejner namapuje port 8000 na hostitelském stroji na port 80, na kterým poběží image *nginx*. To znamená, že server *nginx* může být zpřístupněn přes adresu `localhost:8000`. Zadáním této adresy do webového prohlížeče se zobrazí uvítací stránka, která obsahuje zprávu, že vše bylo nainstalováno správně a naviguje uživatele na vlastní webové stránky Nginx.



Obrázek 10 Uvítací zpráva z web serveru (zdroj: autor)

8.3.4.1 Vytvoření vlastní webového serveru přes image Nginx

Po osvojení základních znalostí s image *nginx*, bylo vytvořeno další pokročilejší cvičení za účelem naprogramování vlastního obsahu na webové stránce na lokálním serveru. K docílení zobrazení této stránky je potřeba nahrát vlastní vytvořený projekt na hostitelském počítači na server. Projekt se skládá z jednoduchého HTML souboru, který obsahuje odstavec a nadpis prvního a třetího stupně.

Kontejner s image *nginx* je již spuštěn a běží na portu 80, ale pro dosažení tohoto cvičení se musí založit nový kontejner s image *nginx* s vlastním obsahem. Bohužel na stejném portu u lokálního serveru nemůžou být spuštěny dva kontejnery najednou. Musí se smazat ten starší (viz foto) a nahradit novým.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\admin1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
efe17f66f107   nginx    "/docker-entrypoint..." 28 seconds ago Exited (0) 10 seconds ago blissful_hellman
54112e647fa0   hello-world "/hello"                3 minutes ago  Exited (0) 3 minutes ago  happy_bla ck
fea526c5cfa3   hello-world "/hello"                31 minutes ago Exited (0) 31 minutes ago  naughty_c urie
PS C:\Users\admin1> docker rm efe17f66f107
efe17f66f107
PS C:\Users\admin1> |
```

Obrázek 11 Zobrazení všech kontejnerů v Dockeru (zdroj: autor)

Aby si student mohl vytvořit nový web server s vlastním obsahem, musí si nejdříve vytvořit vlastní projekt. V řešení tohoto cvičení se používá program *Visual Studio Code* a je i doporučován studentovi. V tomto programu je potřeba si založit *index.html* soubor a naplnit jej nadpisem prvního a třetího stupně a odstavcem, který obsahuje studentův login pro kontrolu, v HTML elementu *<body>*. Složka s projektem se pojmenuje *nginx-website*, která se později nahraje na server. Vlastní obsah pro web server je dokončen a nyní lze spustit image *nginx* v novém kontejneru příkazem

`docker run -d -p 8000:80 -v C:\Users\. Na první pohled se může zdát tento příkaz jako dlouhý a složitý, ale v podstatě za to může absolutní cesta ke složce s projektem. Absolutní cesta odkazuje na místo složky v počítači, která se spojí s adresářem /usr/share/nginx/html, což je cesta k Docker kontejneru. Přes tuto cestu kontejner s image nginx očekává, že nalezne studentem vytvořený obsah pro jeho zobrazení. Pokud je absolutní cesta uvedena správně, tak zadáním localhost:8000/index.html se zobrazí webová stránka s vlastním obsahem.`



Obrázek 12 Zobrazení vlastního projektu na web serveru (zdroj: autor)

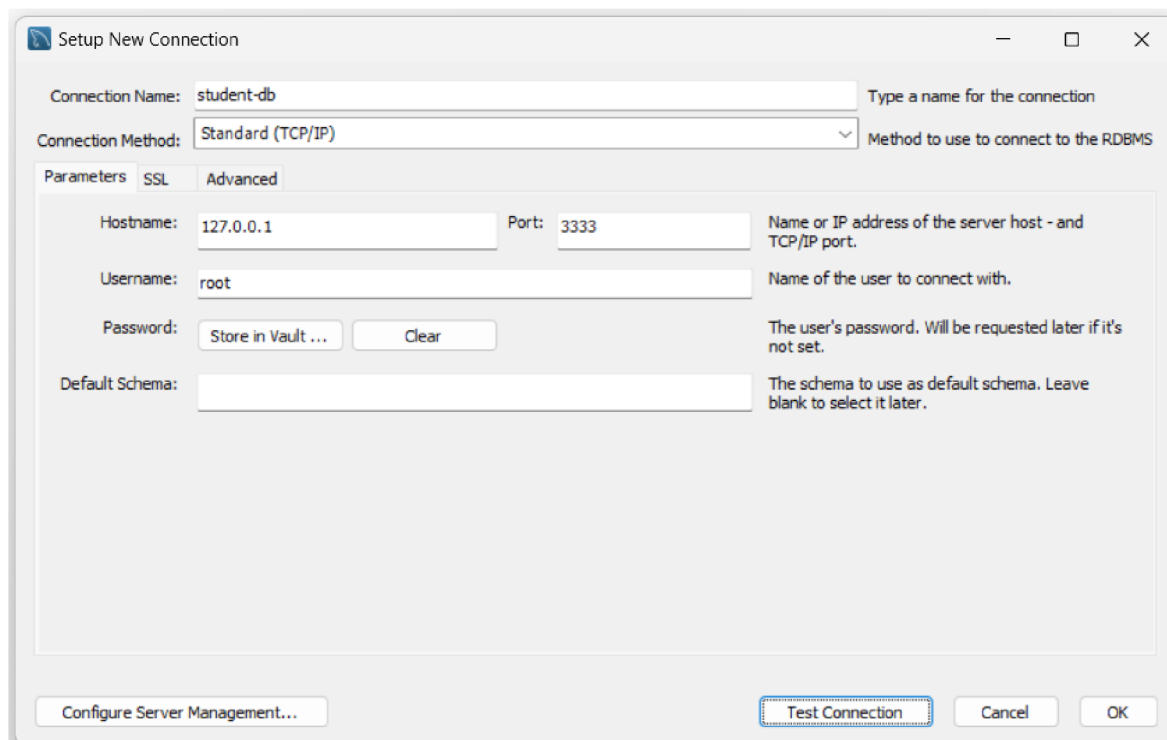
8.3.5 Vytvoření SQL databáze

Po procvičení znalostí z vytváření lokálního webového serveru, bylo vytvořeno další zajímavé cvičení, které se zaměřuje na vytvoření databázového serveru. Cílem tohoto cvičení bude vytvořit databázi, vytvořit strukturu databáze, vložit data a nakonec je vypsat v konzoli za pomoci Dockeru.

Na začátku, než se bude opět pracovat s kontejnery v Dockeru, se musí vyhledat daný image v Docker Hubu. V tomto cvičení se bude primárně pracovat s image `mysql`, který se stáhne opět příkazem `docker pull mysql` a poté se spustí příkazem `docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=root -p 3333:3306 -d mysql:latest`. `-name mysql-db` se definuje název vytvořeného kontejneru, `-e MYSQL_ROOT_PASSWORD=root` nastavuje proměnnou, která popisuje kořenové heslo serveru MySQL, uvnitř kontejneru v Dockeru na hodnotu `root` a `-p 3333:3306` mapuje port 3333 hostitelského počítače na port 3306 v kontejneru Dockeru. Nyní se musí vytvořit nové připojení.

8.3.5.1 Propojení SQL databáze s Dockerem

Během cvičení se bude používat program *MySQL Workbench*, ve kterém se bude vytvářet samotné schéma databáze a následně jejich tabulky. Pomocí tohoto programu se budou vkládat i data. V programu si student bude muset založit nové připojení, ke kterému se později připojí. Toto připojení bude běžet na localhostu na portu 3333.



Obrázek 13 Zakládání nového databázového serveru (zdroj: autor)

V novém připojení se vytvoří tabulka *student*, které nese tyto řádky: *Id_student*, *jmeno*, *prijmeni*, *vek* a *rocnik*. Dále student bude muset vyplnit tabulku minimálně pěti daty, aby bylo co vypisovat v konzoli v Dockeru. Tímto je databázový server vytvořen a naplněn daty a nyní je nutno se připojit na server přes konzoli příkazem `docker exec -it mysql-db bash`. Nyní se student dostal přímo do kontejneru *mysql-db* s linuxovým prostředím. V bashi se zadá příkaz `mysql -u root -p` a po potvrzení heslem se otevře prostředí, kde lze ovládat databázi a operovat s ní. Zde student bude muset právě zobrazit dříve vložená data v programu. Samozřejmě i zde lze přidávat nová data, ale to už je na uvážení studenta, jelikož to není naplnění cvičení.


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\skyfo> docker exec -it mysql-db bash
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

Obrázek 14 Připojování do databázového serveru přes Docker (zdroj: autor)

```
mysql> use studentTable;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from student;
+----+-----+-----+-----+-----+
| Id_student | jmeno | prijmeni | vek | rocnik |
+----+-----+-----+-----+-----+
| 1 | Jan | Novák | 20 | 2 |
| 2 | Marie | Svobodová | 21 | 3 |
| 3 | Petr | Dvořák | 19 | 1 |
| 4 | Eva | Novotná | 22 | 4 |
| 5 | Jiří | Kovář | 20 | 2 |
| 6 | Jan | Novák | 20 | 2 |
| 7 | Marie | Svobodová | 21 | 3 |
| 8 | Petr | Dvořák | 19 | 1 |
| 9 | Eva | Novotná | 22 | 4 |
| 10 | Jiří | Kovář | 20 | 2 |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Obrázek 15 Zobrazení tabulky *student* přes Docker (zdroj: autor)

8.3.6 Závěr

Po úspěšném testování a spuštění Docker images *hello-world*, *nginx* a *mysql* si student prohloubil své znalosti s prací v Dockeru. Toto pokročilé cvičení předpokládalo již určité zkušenosti studenta s Powershellem, což mu umožnilo efektivněji pracovat s kontejnery a jejich správou. Tímto praktickým cvičením si student rozšířil své dovednosti v oblasti kontejnerizace a nasazení aplikací, které dále může využít v reálných situacích.

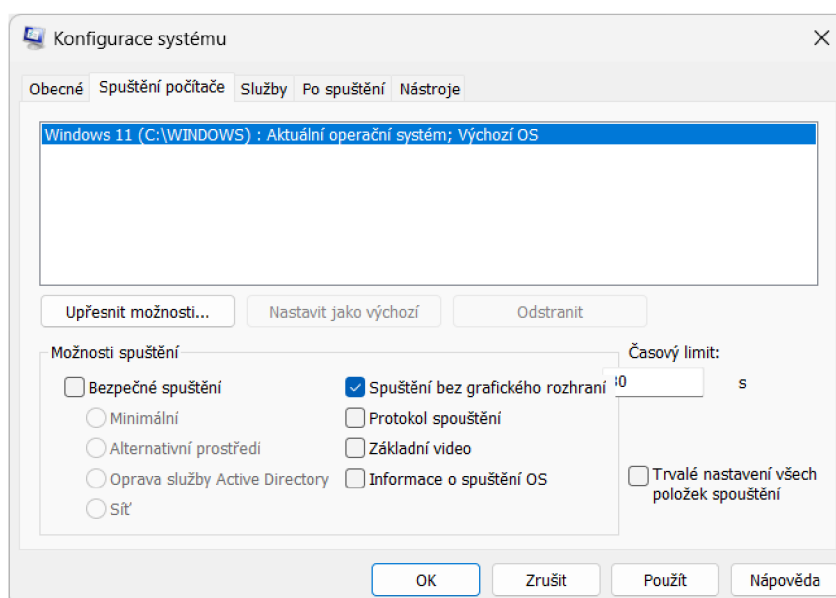
8.4 Třetí cvičení – Diagnostika systému

8.4.1 Úvod

V dnešní době je správa a diagnostika počítačových systémů důležitá pro zajištění jejich spolehlivého a efektivního provozu. S narůstající komplexitou hardwaru a softwaru se stává diagnostika systému zásadním prvkem při identifikaci a řešení potenciálních problémů. Diagnostika systému se zabývá analýzou a monitorováním různých aspektů systému, včetně hardwaru, softwaru, sítí a dalších komponent. Jejím hlavním úkolem je identifikovat příčiny problémů, optimalizovat výkon a zlepšit celkovou spolehlivost systému. Tedy třetí praktické cvičení se zabývá prozkoumáváním metod, technik a nástrojů pro diagnostiku operačního systému Windows 11 a jejich aplikaci v praxi.

8.4.2 Konfigurace systému

Operační systém Windows 11 má v sobě vestavěnou utilitu *Konfigurace systému*, neboli *msconfig*, který slouží k nastavování a úpravě různých startovacích procesů a konfigurací nastavení systému. Tato utilita poskytuje jednoduché rozhraní pro manipulaci s různými systémovými nastaveními a procesy během startu systému. Nástroj je užitečný pro diagnostiku a řešení problémů s bootováním a během běhu systému. Také obsahuje několik záložek, jako jsou Obecné, Spouštění počítače, Služby, Po spuštění a Nástroje. Například na záložce Obecné se může vybrat diagnostický start, který spustí Windows pouze s nezbytnými službami a aplikacemi. Záložka Služby umožňuje spravovat služby Windows a třetích stran a na záložce Spouštěcí oddíl lze spravovat programy, které se spouštějí při startu systému.



Obrázek 16 Rozhraní Konfigurace systému (zdroj: autor)

8.4.2.1 Možnost spuštění bez grafického rozhraní

Pro zapnutí této možnosti se nejprve otevře nástroj Konfigurace systému. To lze udělat například stisknutím kombinaci kláves *WIN + R* pro spuštění okna *Spustit* a do okna se následně zadá *msconfig*. V okně Konfigurace systému se přejde na kartu Spuštění počítače a zde se nachází možnosti pro změnu způsobu, jakým systém startuje. Zaškrtně se možnost *Spuštění bez grafického rozhraní*. Potvrzením změn tlačítkem *OK* a následným restartováním počítače se změny provedou. Po restartu se systém spustí v bezpečném režimu s minimálními ovladači, což znamená, že se bude spouštět bez většiny grafických prvků a s minimem běžících služeb a programů.

8.4.2.2 Vygenerování protokolu spuštění

Nastavení vygenerování protokolu se opět nachází v utilitě Konfigurace systému. Ve spuštěném okně *Spustit* (za pomoci kláves *WIN + R*) se napíše *msconfig* a potvrzením se následně otevře nástroj Konfigurace systému. V horní části okna Konfigurace systému se klikne na kartu nazvanou *Spouštěcí počítače*. Zaškrtnutím volby *Protokol spuštění* se povolí generování protokolu při každém startu systému. Nejdříve se ale musí potvrdit změny a potom restartovat počítač. Protokol se vygeneroval do textového souboru v adresáři *C:\Windows\ntbtlog.txt*. Soubor obsahuje seznam všech služeb, ovladačů a programů, které se spustí při startu systému, a také informace o jejich stavu a čase spuštění.

8.4.2.3 Získání informací o spuštění operačního systému

Poslední praktickým příkladem je možnost zjištění informací o spuštění operačního systému. Tato možnost se nachází v utilitě *Konfigurace systému* a postup otevření utility je stejný jako u předchozích dvou možností. Povolením se zaškrtnou možnosti *Informace o spuštění OS*, poté se potvrdí a restartuje se počítač pro provedení změn. Nyní při každém restartu počítače se zobrazí informace o spuštění operačního systému. Tyto informace budou zobrazeny během startu systému přímo na obrazovce. Obsahují detaily o jednotlivých fázích spuštění systému, načítaných službách a ovladačích.

8.4.3 Spuštěné aplikace po zapnutí počítače

Pro procvičení diagnostiky počítače je tu další vytvořené cvičení za účelem si zobrazit seznam automaticky spustitelných aplikací při každém spuštění systému. Jedná se o velmi primitivní úlohu a student bude muset využít nástroj *Správce úloh* a v něm si zobrazit výše zmíněný seznam aplikací. Jejich stav může změnit dle jeho uvážení.

8.4.4 Spuštěné služby po zapnutí počítače

Když už se studentovi podaří si zobrazit seznam automaticky spustitelných aplikací, na řadě je si zobrazit seznam automaticky spustitelných služeb při každém spuštění systému. Student opět bude muset použít na tuto úlohu nástroj *Správce úloh*, aby zjistil co více tento nástroj dokáže. Stav služeb není doporučováno měnit, pokud student není více znalý. Může se stát, že student omylem vypne důležitou systémovou službu, a pak při spuštění systému může nastat nějaká chyba.

8.4.5 Práce s Process Explorer

Utilita *Správce úloh* nemusí být v určitých situacích v diagnostice procesů dostatečná, a tak byla vytvořena další úloha s nástrojem *Process Explorer*, který nahrazuje nedostatečnou diagnostiku procesů u nástroje *Správce úloh*. Student se tedy seznámí s dalším užitečným nástrojem, ve kterém bude muset nalézt procesy jako *svchost.exe*, *Isass.exe*, *msiexec.exe*, *SecurityHealthService.exe* a určit jejich *PID*, *Description* a *Company Name*.

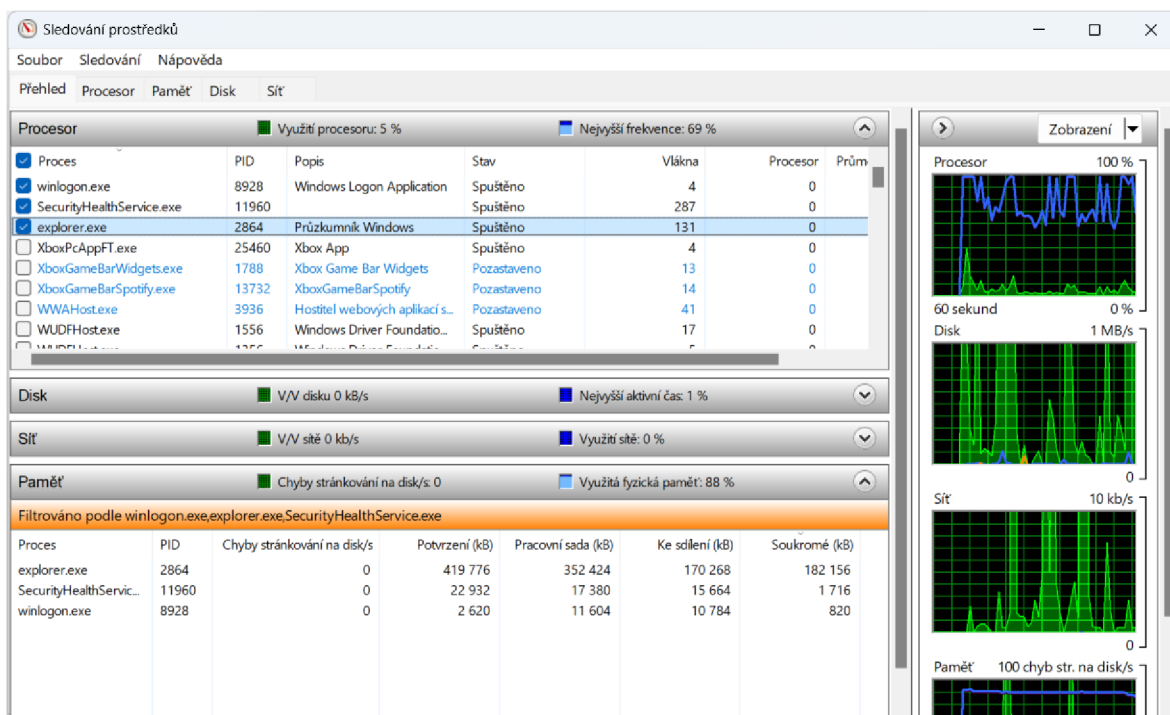
Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Secure System	Susp...	184 K	30 564 K	108		
Registry		24 096 K	42 660 K	144		
System Idle Process	98.19	60 K	8 K	0		
System	< 0.01	60 K	8 276 K	4		
Interrupts	0.38	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		1 112 K	1 180 K	628		
Memory Compression		2 200 K	66 508 K	3360		
csrss.exe		2 344 K	5 740 K	948		
wininit.exe		1 612 K	6 120 K	576		
services.exe		6 880 K	11 132 K	1032		
svchost.exe	< 0.01	24 400 K	43 716 K	1268	Host Process for Windows S...	Microsoft Corporation
RtkAudUService64.exe		12 956 K	13 420 K	5096		
WmiPrvSE.exe	< 0.01	49 392 K	18 952 K	7124		
unsecapp.exe		1 744 K	8 060 K	11376		
StartMenuExperienceHo...	Susp...	94 272 K	48 336 K	16576		
SearchHost.exe		277 916 K	232 640 K	19960		
RuntimeBroker.exe		8 328 K	27 548 K	20292		
Widgets.exe		13 264 K	27 208 K	20176		
msedgewebview2.exe		39 652 K	31 748 K	14444		
msedgewebview2...		2 136 K	7 120 K	23020		
msedgewebview2...		89 712 K	8 148 K	17872		
msedgewebview2...		11 960 K	6 644 K	19556		
msedgewebview2...		9 056 K	96 K	15144		
msedgewebview2...		144 792 K	114 228 K	22896		
msedgewebview2...		67 336 K	8 616 K	7148		
RuntimeBroker.exe		15 844 K	34 964 K	17300		
dllhost.exe		7 532 K	18 472 K	17008		
TextInputHost.exe		55 256 K	24 680 K	22112		
WidgetService.exe		5 144 K	19 812 K	6528		
ShellExperienceHost.exe	Susp...	88 548 K	30 104 K	16244		

CPU Usage: 1.73% Commit Charge: 81.59% Physical Usage: 89.42%

Obrázek 17 Rozhraní Process Explorer (zdroj: autor)

8.4.6 Práce s nástrojem Sledování prostředků

Aby student se více ponořil do diagnostiky počítače, je k tomu vytvořena pokročilejší úloha pro zjištění detailnějších informací o běžících procesech na systému. Ke samotné diagnostice se využije nástroj *Sledování prostředků*, který umožňuje monitorovat výkon a stav různých hardwarových komponent v počítači. Tato úloha se zaměřuje specificky na procesy *explorer.exe*, *winlogon.exe* a *SecurityHealthService.exe*. Student bude muset s využitím nástroje zjistit následující informace o těchto procesech: počet vláken, průměrný čas na procesoru, počet chyb stránkování na disku, potvrzenou přidělenou paměť v kB, velikost pracovní sady paměti v kB, kapacitu ke sdílení v kB a soukromou kapacitu v kB.



Obrázek 18 Rozhraní Sledování prostředků (zdroj: autor)

8.4.7 Závěr

Po vyřešení cvičení na správu, konfiguraci a diagnostiku systému se student stal schopným analyzovat a řešit potenciální problémy v prostředí operačního systému Windows 11. Práce s nástroji jako Správce úloh, Process Explorer a Sledování prostředků mu umožnila získat hlubší vhled do běhu systému a procesů. Tato praxe rozšířila jeho schopnosti v oblasti monitorování a optimalizace výkonu počítačových systémů.

9 Závěry a doporučení

Práce reflektuje úspěšné naplnění stanovených cílů v oblasti návrhu a prezentace praktických cvičení zaměřených na správu a konfiguraci operačního systému Windows 11. Teoretická část práce poskytla komplexní pohled na vývoj, architekturu a prvky operačního systému Windows, což bylo nezbytné pro porozumění praktické implementace cvičení. V praktické části práce se provedl detailní rozbor a demonstrace cvičení využívajících technologie jako Windows Subsystem for Linux (WSL), Docker a diagnostické nástroje systému Windows 11. Cvičení byla navržena tak, aby studentům poskytovala užitečné praktické dovednosti pro správu operačního systému Windows 11.

Po vzájemné diskusi s vedoucím práce došlo k závěru, že implementovaná cvičení splňují stanovené cíle a přinášejí přidanou hodnotu pedagogického materiálu. Autor práce získal hlubší porozumění problematice správy operačního systému Windows 11 a získal cenné zkušenosti v tvorbě výukových materiálů.

Jako možný směr dalšího výzkumu se nabízí rozšíření praktických cvičení o nové technologie a metody správy systémů, jako je například implementace konceptů pro správu sítí nebo automatizace konfigurace prostřednictvím skriptování.

10 Seznam použité literatury

- [1] GCFGLOBAL.ORG. How is Windows 10 different from other versions? GCFGlobal.org [online]. 2022 [cit. 2023-11-24]. Dostupné z: <https://edu.gcfglobal.org/en/windows10/windows-10-features/1/#>
- [2] MICROSOFT. Update release cycle for Windows clients. MICROSOFT. Microsoft Learn [online]. 2023 [cit. 2023-11-24]. Dostupné z: <https://learn.microsoft.com/en-us/windows/deployment/update/release-cycle>
- [3] MICROSOFT. Windows 11 features to explore. MICROSOFT. Microsoft [online]. 2023 [cit. 2023-11-25]. Dostupné z: www.microsoft.com/en-us/windows/learning-center/windows-11-features-to-explore
- [4] ZIFF DAVIS, LLC. Windows 11: Biggest Changes and New Features. ZIFF DAVIS, LLC. PCMag [online]. 2022 [cit. 2023-11-25]. Dostupné z: <https://www.pcmag.com/news/everything-you-need-to-know-about-windows-11>
- [5] ROMAN, Steven. MICROSOFT. Windows Architecture. MICROSOFT. Microsoft [online]. 2014 [cit. 2023-12-03]. Dostupné z: [https://learn.microsoft.com/en-us/previous-versions/cc768129\(v=technet.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/cc768129(v=technet.10)?redirectedfrom=MSDN)
- [6] PADAKUU. Environmental Subsystems. PADAKUU. PadaKuu [online]. 2023 [cit. 2023-12-03]. Dostupné z: <https://padakuu.com/environmental-subsystems-143-article>
- [7] SAJAL. The NT Architecture of Windows. Certcube [online]. 2021 [cit. 2023-12-03]. Dostupné z: <https://blog.certcube.com/the-nt-architecture-of-windowss/>
- [8] BUCK, Alex, Amy VIVIANO a Mark HOPKINS. User mode and kernel mode. Microsoft [online]. 2023 [cit. 2023-12-03]. Dostupné z: <https://learn.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/user-mode-and-kernel-mode>
- [9] PILICI, Stelian. MALWARETIPS LLC. Csrss.exe: What It Is & Should I Remove It? MalwareTips [online]. 2023 [cit. 2023-12-04]. Dostupné z: <https://malwaretips.com/blogs/csrss-exe-what-it-is-should-i-remove-it/>
- [10] PILICI, Stelian. MALWARETIPS LLC. Win32k.sys BSOD: What It Is & How To Fix It? MalwareTips [online]. 2023 [cit. 2023-12-04]. Dostupné z: <https://malwaretips.com/blogs/win32k-sys-bsod-what-it-is-how-to-fix-it/>
- [11] BENCHERCHALI, Nasreddine. Windows System Processes — An Overview For Blue Teams. Medium [online]. 2020 [cit. 2023-12-05]. Dostupné z: <https://nasbench.medium.com/windows-system-processes-an-overview-for-blue-teams-42fa7a617920>
- [12] HOFFMAN, Chris. What Is Windows Logon Application (winlogon.exe), and Why Is It Running on My PC? How-To Geek [online]. 2017 [cit. 2023-12-05]. Dostupné z: <https://www.howtogeek.com/322411/what-is-windows-logon-application-winlogon.exe-and-why-is-it-running-on-my-pc/>
- [13] Windows 7. ENCYCLOPEDIA BRITANNICA. Britannica [online]. 2022 [cit. 2023-12-05]. Dostupné z: <https://www.britannica.com/technology/Windows-7>

- [14] HEDDINGS, Anthony. What is Windows Subsystem For Linux (WSL), and How Do You Use It? How-To Geek [online]. 2021 [cit. 2023-12-05]. Dostupné z: <https://www.howtogeek.com/devops/what-is-windows-subsystem-for-linux-wsl-and-how-do-you-use-it/>
- [15] LOEWEN, Craig, Matt WOJCIAKOWSKI, Will BJORN a Alma JENKS. How to install Linux on Windows with WSL. Microsoft [online]. 2023 [cit. 2023-12-05]. Dostupné z: <https://learn.microsoft.com/en-us/windows/wsl/install>
- [16] LOEWEN, Craig, Matt WOJCIAKOWSKI a Alma JENKS. Basic commands for WSL. Microsoft [online]. 2023 [cit. 2023-12-05]. Dostupné z: <https://learn.microsoft.com/en-us/windows/wsl/basic-commands>
- [17] LOEWEN, Craig a Matt WOJCIAKOWSKI. Getting started with Linux and Bash. Microsoft [online]. 2022 [cit. 2023-12-06]. Dostupné z: <https://learn.microsoft.com/en-us/windows/wsl/tutorials/linux>
- [18] ALEKSIC, Marko. How to Use the sudo Command in Linux. PhoenixNAP [online]. 2020 [cit. 2023-12-06]. Dostupné z: <https://phoenixnap.com/kb/linux-sudo-command>
- [19] KNOWLEDGE BASE. About Unix sudo and su commands. University Information Technology Services [online]. 2023 [cit. 2023-12-06]. Dostupné z: <https://kb.iu.edu/d/amyi>
- [20] LINUXIZE. How to Use Nano, the Linux Command Line Text Editor. Linuxize [online]. 2023 [cit. 2023-12-06]. Dostupné z: <https://linuxize.com/post/how-to-use-nano-text-editor/>
- [21] GNU NANO. The GNU nano text editor. GNU nano [online]. 2009, 2023-01-08 [cit. 2023-12-06]. Dostupné z: <https://nano-editor.org/manual.html>
- [22] GARN, Damon. 8 essential Linux file navigation commands for new users. Red Hat [online]. 2022 [cit. 2023-12-08]. Dostupné z: <https://www.redhat.com/sysadmin/Linux-file-navigation-commands>
- [23] MCKAY, Dave. How to Work with Variables in Bash. How-To Geek [online]. 2023 [cit. 2023-12-16]. Dostupné z: <https://www.howtogeek.com/442332/how-to-work-with-variables-in-bash/>
- [24] DIGITALOCEAN. How to Use if-else in Shell Scripts? DigitalOcean [online]. 2022 [cit. 2023-12-16]. Dostupné z: <https://www.digitalocean.com/community/tutorials/if-else-in-shell-scripts>
- [25] VITUX. How to create and run a shell script in Ubuntu 22.04. VITUX [online]. 2024 [cit. 2024-01-20]. Dostupné z: <https://vitux.com/ubuntu-shell-script/>
- [26] VITUX. How to Write a Shell Script in Ubuntu. VITUX [online]. 2022 [cit. 2024-01-20]. Dostupné z: <https://vitux.com/how-to-write-a-shell-script-in-ubuntu/>
- [27] PABLINUX. Jak nainstalovat Ubuntu s grafickým rozhraním na Windows díky WSL2, nebo ještě lépe Kali Linux. Ubuunlog [online]. 2023 [cit. 2023-12-20]. Dostupné z: <https://ubunlog.com/cs/jak-nainstalovat-ubuntu-s-grafickým-rozhraním-ve-windows-díky-wsl2-nebo-ještě-lépe-kali-linux/>
- [28] MICROSOFT. Jak používat Vzdálenou plochu. Microsoft [online]. 2021 [cit. 2023-12-20]. Dostupné z: <https://support.microsoft.com/cs-cz/windows/jak-používat-vzdálenou-plochu-5fe128d5-8fb1-7a23-3b8a-41e636865e8c>

- [29] DOCKER INC. Docker overview. Docker [online]. 2024 [cit. 2024-01-25]. Dostupné z: <https://docs.docker.com/get-started/overview/>
- [30] SERVERWATCH STAFF. SERVERWATCH. Windows NT Architecture. TECHNOLOGYADVICE. ServerWatch [online]. 2001 [cit. 2024-02-10]. Dostupné z: <https://www.serverwatch.com/guides/windows-nt-architecture/>
- [31] MADHAVI, Aluka. Introduction of Operating System. GEEKSFORGEEEKS. GeeksForGeeks [online]. 2024, 2024-01-19 [cit. 2024-02-10]. Dostupné z: <https://www.geeksforgeeks.org/introduction-of-operating-system-set-1/>
- [32] RATHORE, Prashant. Resource Management in Operating System. GEEKSFORGEEEKS. GeeksForGeeks [online]. 2023, 2023-01-25 [cit. 2024-02-10]. Dostupné z: <https://www.geeksforgeeks.org/resource-management-in-operating-system/>
- [33] VAŇKOVÁ, Jana a Michal ČERNÝ. Úvod do operačních systémů. Metodický portál RVP.CZ [online]. 2011 [cit. 2024-02-11]. Dostupné z: <https://clanky.rvp.cz/clanek/c/g/13995/UVOD-DO-OPERACNICH-SYSTEMU.html>
- [34] MICROSOFT. Spouštění kontejnerů na Windows Serveru. MICROSOFT. Microsoft Learn [online]. 2023 [cit. 2024-02-24]. Dostupné z: <https://learn.microsoft.com/cs-cz/training/modules/run-containers-windows-server/>
- [35] DOCKER INC. Hello-world. DOCKER INC. Docker Hub [online]. 2023 [cit. 2024-03-02]. Dostupné z: https://hub.docker.com/_/hello-world/
- [36] DOCKER INC. Nginx. DOCKER INC. Docker Hub [online]. 2023 [cit. 2024-03-02]. Dostupné z: https://hub.docker.com/_/nginx/
- [37] MICROSOFT. Registry, image a kontejnery Dockeru. MICROSOFT. Microsoft Learn [online]. 2024 [cit. 2024-03-04]. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/architecture/microservices/container-docker-introduction/docker-containers-images-registries>
- [38] CHOWDHURY, Farhan Hasin. The Docker Handbook. HonKit [online]. 2020 [cit. 2024-03-05]. Dostupné z: <https://docker-handbook.farhan.dev/en/>
- [39] DOMANTAS, G. What Is Localhost? Hostinger [online]. 2023 [cit. 2024-03-25]. Dostupné z: <https://www.hostinger.com/tutorials/what-is-localhost>
- [40] ROUSE, Margaret. Port 80. Techopedia [online]. 2023 [cit. 2024-03-25]. Dostupné z: <https://www.techopedia.com/definition/15709/port-80>
- [41] KOĐOUSKOVÁ, Barbora. HTML pro začátečníky aneb jak začít psát web. RASCASONE S.R.O. Rascasone [online]. 2020 [cit. 2024-03-25]. Dostupné z: <https://www.rascasone.com/cs/blog/html-pro-zacatecniky-jak-psat-web>
- [42] DOCKER INC. Mysql. DOCKER INC. Docker Hub [online]. 2024 [cit. 2024-04-03]. Dostupné z: https://hub.docker.com/_/mysql/
- [43] DOCKER INC. Mysql/mysql-server. DOCKER INC. Docker Hub [online]. 2024 [cit. 2024-04-03]. Dostupné z: <https://hub.docker.com/r/mysql/mysql-server/>
- [44] CHAMBERLIN, Don. SQL. In: Encyclopedia of Database Systems. Springer US, 2009, s. 2753–2760. ISBN 978-0-387-35544-3.
- [45] SULEMANI, Maryam. MySQL Workbench tutorial: Complete guide to the RDBMS tool. EDUCATIVE INC. Educative [online]. 2020 [cit. 2024-04-09]. Dostupné z: <https://www.educative.io/blog/mysql-workbench-tutorial>

- [46] RUSSINOVICH, Mark a Aaron MARGOSIS. Process Explorer. In: Troubleshooting with the Windows Sysinternals Tools. 2. vydání. Microsoft Press, 2016, s. 87–184. ISBN 978-0735684447.
- [47] YOSIFOVICH, Pavel, David A. Solomon, David A. SOLOMON a Mark E. RUSSINOVICH. Concepts and tools. In: Windows Internals: System architecture, processes, threads, memory management, and more, Part 1. 7. vydání. Microsoft Press, 2017, s. 22–82. ISBN 978-0735684188.
- [48] SINGH, Nitin. 13 Ways to Reduce Background Processes in Windows 11. TECHWISER MEDIA PRIVATE LIMITED. Techwiser [online]. 2023 [cit. 2024-04-11]. Dostupné z: <https://techwiser.com/ways-to-reduce-background-processes-in-windows-11/>
- [49] KHANSE, Anand. How to stop Background Processes in Windows 11/10. THE WINDOWS CLUB. The Windows Club [online]. 2024 [cit. 2024-04-11]. Dostupné z: <https://www.thewindowsclub.com/how-to-stop-background-processes-in-windows>
- [50] TURICEANU, Vlad. How Many Processes Should Be Running on Windows 11? Windows Report [online]. 2024 [cit. 2024-04-12]. Dostupné z: <https://windowsreport.com/windows-11-how-many-processes-should-be-running/>
- [51] MICROSOFT. Process Explorer v17.05. MICROSOFT. Microsoft Learn [online]. 2023 [cit. 2024-04-12]. Dostupné z: <https://learn.microsoft.com/cs-cz/sysinternals/downloads/process-explorer>
- [52] WIENAND, Ian a Mark E. RUSSINOVICH. The role of the operating system. In: Computer Science from the Bottom Up. Creative Commons Attribution-ShareAlike License, 2004, s. 98-100. ISBN N/A.

11 Přílohy

Cvičení 1 – 3 a jejich řešení jsou k dispozici na digitálním repositáři GitHub (viz příloha č. 1).

<https://github.com/AdamSvit/UHK-Bakalarska-prace--Sprava-konfigurace-a-analyza-Windows-11>

12 Zadání práce z IS (eVŠKP)



Zadání bakalářské práce

Autor: Adam Svítek

Studium: I2000418

Studijní program: B1802 Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Název bakalářské práce: **Správa, konfigurace a analýza Windows - video tutorial**

Název bakalářské práce AJ: Managing, configuring, and analyzing Windows 11 - video tutorial

Cíl, metody, literatura, předpoklady:

Cílem bakalářské práce je vytvořit podpůrné materiály v oblasti správy, konfigurace a analýzy desktop implementací OD Windows v podobě video tutoriálů. V teoretické části autor představí a podrobně popíše postupy a řešení dílčích úloh správy, konfigurace a analýzy desktop Windows. V praktické části pak autor vytvoří praktická řešení dílčích úloh ve formě video tutoriálů.

Zadávací pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: doc. Mgr. Josef Horálek, Ph.D.

Datum zadání závěrečné práce: 15.10.2021