Univerzita Pardubice
Fakulta ekonomicko-správní

Vytvoření podpůrných nástrojů pro výuku předmětu Operační systémy

Michal Bělský

Bakalářská práce
2008

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky
Akademický rok: **2007/2008**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal BĚLSKÝ**

Studijní program: **B6209 Systémové inženýrství a informatika**

Studijní obor: **Regionální a informační management**

Název tématu: **Vytvoření podpůrných nástrojů pro výuku předmětu Operační systémy**

Zásady pro vypracování:

Bakalářská práce bude zaměřena na problematiku moderních operačních systémů. Student zpracuje vybrané oblasti z této problematiky na základě nejnovějších poznatků a přístupů. Součástí práce budou podpůrné prostředky pro výuku Operačních systémů. Důraz bude kladen na objasnění činnosti počítačových systémů, problematiku propojení hardwarových a softwarových komponentů a bezpečnost OS.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

SILBERSCHATZ A., GALVIN P.,B., GAGNE G. Operating systems concepts. John Wiley & Sons. 6th edition. Hoboken, 2003. 951 s. ISBN 0-471-25060-0.

STALLINGS, W. Operational Systems. Prentice Hall. Upper Saddle River. 5th edition. New Jersey. 2005. 818 s. ISBN 0-13-127837-1. TANENBAUM A. S. Modern operating Systems. Prentice Hall. Upper Saddle River. 2nd edition. 2001. 976 s. ISBN 0-13-031358-0.

ČADA O. Operační systémy.Grada Publishing. Praha. 1994. 377 s. ISBN 80-85623-44-7.

BIC L., SHAW A.C. Operating Systems Principles. Prentice-Hall. 1st edition. 2003. 543 s. ISBN 0-13-026611-6.

ŠAFAŘÍK, J. Operační systémy [online]. Katedra informatiky a výpočetní techniky (FAV ZČU). Plzeň. 2006. [citováno 2007-10-12]. Dostupné z URL http://www.kiv.zcu.cz/ safarikj/vyuka/os/prednasky.html

VAŘEČKOVÁ Š. Operační systémy [online]. Ústav informatiky (Slezská univerzita). Opava. 2006. [citováno 2007-10-12]. Dostupné z URLhttp://axpsu.fpf.slu.cz/ vav10ui/opsys.html

Vedoucí bakalářské práce: **Ing. Pavel Jirava**
Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **22. října 2007**

Termín odevzdání bakalářské práce: **19. května 2008**

prof. Ing. Jan Čapek, CSc.
děkan

L.S.

doc. Ing. Pavel Petr, Ph.D.
vedoucí ústavu

V Pardubicích dne 22. října 2007

Poděkování

Rád bych poděkoval svému vedoucímu práce Ing. Pavlu Jiravovi, Ph.D., za odborné vedení, náměty a připomínky, které mi poskytoval v průběhu celého období zpracovávání mé bakalářské práce.

SOUHRN

V mé bakalářské práci se zabývám oblastí operačních systémů. Operační systém v počítačovém odvětví prošel bohatou historií od příkazové řádky až po plně grafický vzhled.

V první části popisuji architektury počítačových systémů, a jakou úlohu hraje operační systém v oblasti počítačů. Práce popisuje historii a jednotlivé typy operačních systémů. Dále se snaží seznámit s bezpečností operačních systémů, která je v dnešní době nejvíce diskutovanou oblastí.

Ve druhé části jsou vysvětleny pojmy proces a vlákna, jaký je mezi nimi rozdíl a jakou důležitou roli hrají v oblasti operačních systémů. Jsou zde probírány praktické příklady, jak se jednotlivé vlákna v procesu chovají při spuštěné aplikaci. Tyto příklady budou používány ve cvičeních z operačních systémů.

KLÍČOVÁ SLOVA

Operační systém, počítačové architektury, historie operačních systémů, typy operačních systémů, bezpečnost operačních systémů, procesy, vlákna.

TITLE

Creating supporting tools to education subject of Operating system.

SUMMARY

In my bachelor work I deal with an area of operating system. Operating system in computer industry went through wide and abundant history from command line to fully graphical design.

In the first part I describe architectures of computer systems and what role operating system performs in computer branch. The work describes history and individual types of operating systems. Further it tries to acquaint with security of operating systems which is the most discussed theme at the present time.

In the second part there are explained concepts of process and threads and the differences between them, what important role they perform in the branch of operating systems. There are discussed practical examples explaining how individual threads in the process behave at the running application. These examples will be used in exercises of operating systems.

KEY WORDS

Operating system, computer architectures, history of operating system, types of operating systems, security of operating systems, processes, threads.

Obsah

Úvod

Tato bakalářská práce se zaměřuje na problematiku a principy operačních systémů. Obzvláště se autor zaměřuje na vývoj a bezpečnost operačních systémů, na procesy a vlákna a jejich fungování. Pro sledování procesů a vláken jsou využity Správce úloh systému Windows a Performace Monitor na sledování výkonu počítače.

Oblast operačních systémů určitě stojí za povšimnutí. Podle autorova názoru si většina lidí pod pojmem operační systém představí slovo Windows. Ale co se skrývá uvnitř tohoto slova už mnoho lidí neví, i když je to nesmírně důležité pro funkčnost celého počítače. Jeho bohatá historie sahá do první poloviny 20. století, kdy počítače zabíraly celou místnost, uživatel znal pouze děrné štítky a záznam na pásce.[26] V této době operační systém ještě neexistoval.[26] Postupem času se vyvinul uživatelsky přístupný operační systém, tak jak ho všichni známe z každodenního používání. Většina lidí si ani neuvědomuje jak velkou a důležitou roli hraje v oblasti počítačů. Bez operačního sytému by počítače přestaly existovat a bez počítačů by přestal existovat operační systém. A právě jeho bouřlivý vývoj se musel také neustále přizpůsobovat požadavkům a parametrům nových počítačů. Hlavně v oblasti hardware (nárůst kapacity disků). Další příčinou vývoje operačních systémů byla potřeba nových služeb, mezi které patří internet, LAN sítě a multimédia.[28] Většina lidí si svět bez internetu neumí představit.

Tak jak se zvyšovaly nároky na hardware počítače, tak se zvyšovaly i nároky na operační systém.[28] Hardware a operační systém se navzájem předháněly. Když vyšla nová verze operačního systému, tak už k tomu na trhu byl dostupný potřebný hardware.

Bezpečnost operačních systémů je v dnešní době nejvíce sledovaná oblast v oboru počítačových technologií. Ten, kdo chce odolávat útokům a bezpečnostním hrozbám, musí tuto oblast sledovat a udržovat si svůj software denně aktualizovaný.

Oblast bezpečnosti počítačů je široká a zahrnuje fyzické a administrativní řízení. Touto bakalářskou prací si upevníme význam bezpečnostních nástrojů a prozkoumáme typy hrozeb směřované na počítačovou komunikaci. Zaměříme se na speciální nástroje (tools), které zvýší bezpečnost. Dále se budeme zabývat tradičními přístupy počítačové bezpečnosti, které jsou založeny na ochranu různých počítačových prostředků, obsahujících paměť a data.[25]

Druhá část bakalářské práce pojednává o činnosti procesů a vláken ve Windows XP. Klade důraz na objasnění těchto pojmů a hlavní rozdíly mezi nimi. Sledujeme činnost ve správci

úloh a v Performance monitoru. Sledujeme, jak jednotlivé procesy vytěžují procesor a paměť a kolik obsahují vláken. Microsoft Windows používá pro pojem vlákno název podproces. V Performance monitoru monitoruje procesy a jednotlivé stavy jejich vláken. Tyto stavy si potom můžeme prohlédnout ve vygenerovaném souboru v excelu.

Autor by chtěl hlavně poukázat na význam operačního systému. Pod slovem operační systém se neskrývá jenom uživatelsky přehledná pracovní plocha na obrazovce počítače, ale také celá sada instrukcí a úkolů, které musí vykonat, aby spolu hardware a aplikace navzájem komunikovaly.

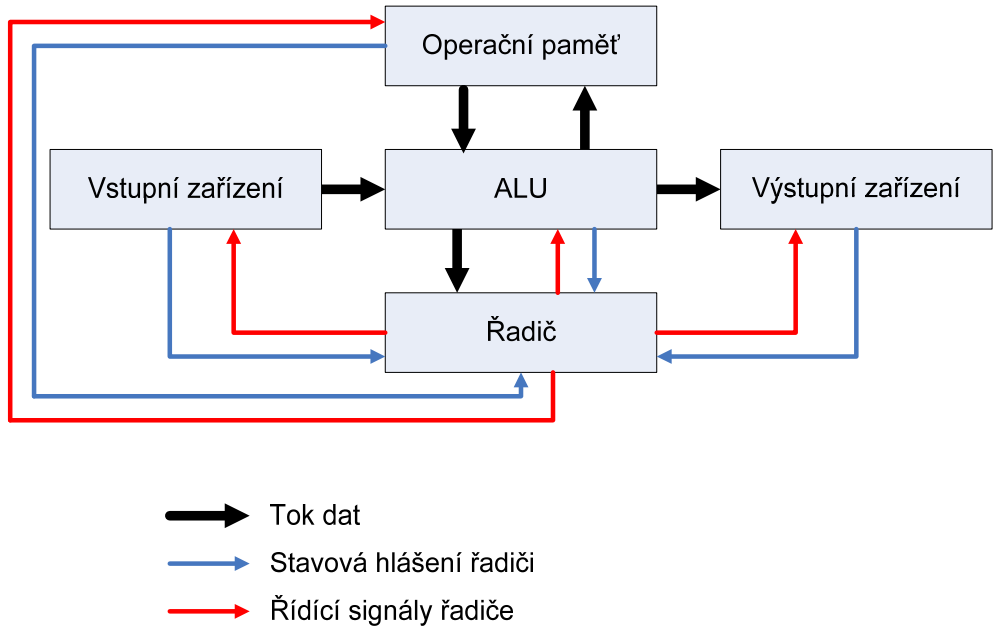
Cílem této práce je:

- Zdůraznit význam operačního systému v oblasti počítačů.
- Poukázat na bezpečnost operačních systémů a na pokročilé počítačové architektury.
- Pochopení a vysvětlení významu pojmů proces a vlákno a jakou důležitou roli hrají v oblasti operačních systémů.
- Vysvětlit a popsat práci se softwarovými nástroji Správce úloh systému Windows a Performance monitorem.
- Části této práce budou sloužit jako pomůcka ke cvičením z předmětu Operační systémy. Obzvláště pro pochopení činnosti procesů a vláken a jejich sledování ve výše uvedených softwarových nástrojích.

1. Struktura počítače

1.1. Von Neumannovo schéma počítače

Von Neumannovo schéma bylo prvním modelem samočinného počítače. Označení podle svého autora z roku 1945.[3][14][24]



Obrázek č. 1: Von Neumannovo schéma počítače [14][24]

Tato koncepce předpokládá použití společného prostoru v paměti pro umístění dat a programu. Není jednoznačně určeno, kde v paměti je uložen program a kde data.[3][29]

1.1.1. Popis obrázku Von Neumannovo schéma počítače

1. Operační paměť: Slouží k uchování dat, programů, které se zpracovávají.
2. Vstupní zařízení: Pomocí tohoto zařízení vstupují data nebo program.
3. Výstupní zařízení: Určená pro výstup výsledků, které program zpracoval.
4. ALU (aritmetickologická jednotka): Jednotka určená pro veškeré aritmetické výpočty a logické operace.
5. Řadič: Řídící jednotka, která řídí činnost všech částí počítače (řízení je prováděno pomocí řídících signálů). Reakce na řídící signály a stavy jednotlivých modulů jsou naopak zasílány zpět řadiči pomocí stavových hlášení.[14]

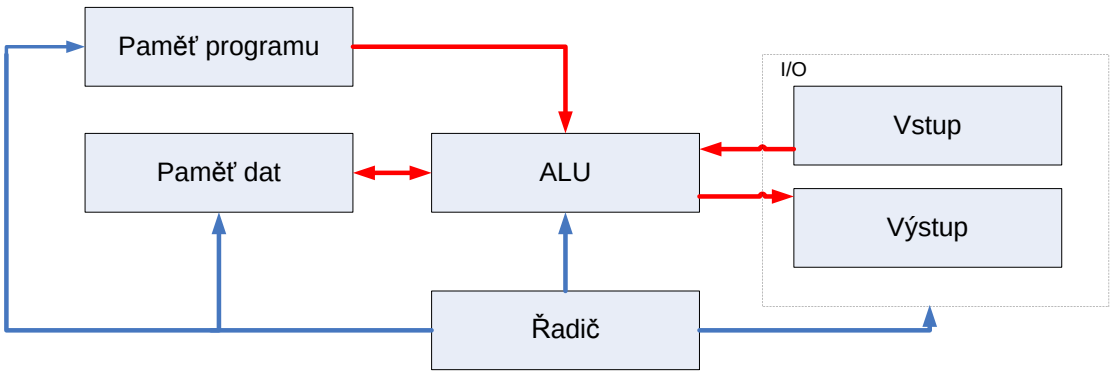
1.1.2. Popis činnosti

Data a program se přes ALU načtou do operační paměti. Zahájí se vlastní výpočet, jednotlivé kroky provádí ALU. Je řízen spolu s ostatními moduly řadičem. Průběžné výsledky výpočtu jsou ukládány do operační paměti. Po skončení výpočtu jsou poslány přes ALU na výstup.[14]

1.2. Základní odlišnosti dnešních počítačů od Von Neumannova schématu

1. Dnešní počítač může zpracovávat více programů najednou.
2. Dnešní počítač může být složen z více procesorů.
3. Existují zařízení, které zvládnou vstup i výstup dat.
4. Program se do paměti nemusí zavést celý, ale je možné zavést pouze jeho část a ostatní části zavádět až v případě potřeby.
5. Počítač by podle Von Neumannova schématu pracoval pouze v tzv. diskrétním režimu.[14]

2. Harwardské schéma



Obrázek č. 2: Harwardská koncepce [23]

2.1. Popis činnosti

Princip Harwardské koncepce je patrný z obrázku č. 2. Ze vstupního zařízení se načtou program a data přes ALU do paměti dat a do paměti programu. ALU zahájí svou činnost, začne provádět veškeré aritmetické a logické operace. Veškerou činnost kontroluje a řídí řadič. Po ukončení výpočtu jsou veškeré výsledky poslány přes ALU na výstup.

Historicky starší než Von Neumannova koncepce. Harwardské schéma používá pro program a pro data dvě nezávislé paměti. Označení podle počítače Harward Mark I, uvedeného do provozu na Harwardské univerzitě roku 1943.[8][29]

Počítač postavený na této architektuře měl instrukce uložené na děrované pásce a data na elektromechanických deskách. Architektura povoluje mít paměť stejných parametrů a vlastností jak pro data, tak pro program. Paměti mohou být vyrobeny úplně odlišnou technologií. Mohou mít různou délku slova, časování a způsob adresace.[8]

2.2. Porovnání Von Neumannova a Harwardského počítače

2.2.1. Von Neumannovo schéma

Používá se u univerzálních počítačů. V dnešní době se používá ještě u nejjednodušších kalkulaček.[29]

Výhody:

- nižší cena systému (pouze jedna paměť)
- vysoká pružnost systému (lze snadno měnit program a zpracovávat data)
- jednodušší[29]

Nevýhody:

- je nutný kompromis mezi šířkou toku programu a dat
- program může být nežádoucím způsobem ovlivněn (chybami v programu)
- rychlost (procesory mají výrazně vyšší rychlost než paměť)
- sekvenční zpracování (společná paměť, propojovací obvody)[29]

2.2.2. Harwardské schéma

Dnes se používá u specializovaných procesorů (jednočipové počítače, signálové procesory).[8]

Výhody:

- vysoká bezpečnost (paměť programu je typu ROM, takže program ji nemůže přepsat)
- u paměti programu lze použít jinou šířku slova (počet bitů) než u paměti dat

- rychlé zpracování (paralelní zpracování)[29]

Nevýhody:

- vyšší cena hardware (dvě paměti)
- malá pružnost systému (program lze změnit jen obtížně nebo vůbec ne)[29]

V moderním procesoru najdou uplatnění obě architektury. Uvnitř je použita Harwardská architektura, kde se paměť cache dělí na paměť dat a na paměť programu. Z venku se chová jako Von Neumannova architektura, protože načítá program i data z hlavní paměti naráz.[8]

3. Operační systém

Operační systém je program, který kontroluje provádění aplikačních programů a vystupuje jako rozhraní mezi aplikacemi a počítačovým hardwarem.[25] Operační systém je softwarovou nadstavbou hardware počítače.[28]

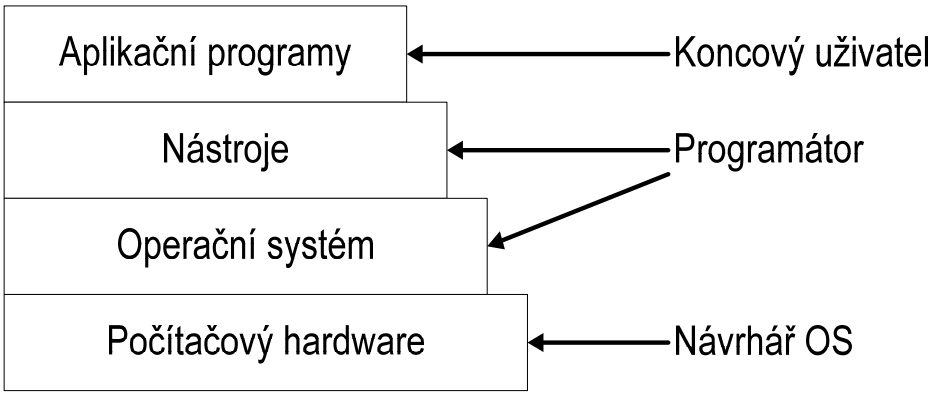
Laik neví, co si má pod pojmem operační systém představit, na druhou stranu programátoři a odborníci nemají v tomto pojmu také úplně jasno.[4]

Mnoho z nás má určité znalosti s operačním systémem, hlavně z uživatelského pohledu.[26] Každý ví, že má zmáčknout toto tlačítko na pracovní ploše na monitoru, ale k provedení tohoto úkonu je zapotřebí vykonat mnoho dalších operací, o které se stará právě operační systém. Uživatel neřeší, kdy se má spustit tento proces, kdy se má uvolnit paměť. Tyto úkoly řeší operační systém.

3.1. Na operační systém můžeme nazírat ze dvou rovin

3.1.1. Operační systém jako uživatelské rozhraní

Operační systém je software použitý k poskytování aplikací uživatelům. Na obrázku Vrstvy počítačového systému můžeme vidět členění počítačového systému.[25]



Obrázek č. 3: Vrstvy počítačového systému [25]

Operační systém je prostředník mezi aplikačními programy a počítačovým hardwarem (viz obrázek Vrstvy počítačového systému).[25] Umožňuje používat systém tzv. laickým uživatelům. Skrývá před uživatelem detaily hardwaru. Usnadňuje práci s aplikacemi koncovému uživateli.[28] OS je tzv. zjednodušující interfejs pro komunikaci mezi aplikacemi a hardwarem počítače. Např. čtení nebo zápis na disk.[9]

3.1.2. Operační systém jako správce prostředků

Moderní počítače se skládají z procesorů, paměti, myši, disků, počítačových rozhraní, tiskáren a dalších zařízení. Úlohou operačního systému je postarat se o uspořádané a kontrolované rozvržení procesorů, paměti a I/O zařízení mezi různé programy, které o ně soupeří.[26]

Z názvu vyplývá, že operační systém bude v roli správce (správce paměti, správce procesů, správce periférií, správce souborů).[9][22][30] Je zodpovědný za správnou funkci ukládání, pohybu a zpracování dat. A zároveň tyto operace kontroluje.[22][25]

Příkladem může být tisk na tiskárnu. Představme si, když budou na počítači běžet tři programy a každý z nich bude chtít současně tisknout na výstup na stejnou tiskárnu. Bez operačního systému by to bylo strašně komplikované, tiskárna by nám vytiskla několik řádků z prvního programu, pak několik z druhého a nakonec několik z třetího. Takto by se to pořád opakovalo. Díky operačnímu systému se této komplikaci vyvarujeme. Bude programy řídit.[26]

4. Historie operačních systémů, generace

Operační systém byl vyvíjen během několika desítek let. V následujících několika řádcích se podíváme na jeho hlavní mezníky vývoje.[26]

Většina vědců a inženýrů souhlasí, že byly čtyři odlišné výpočetní období. Dávkové zpracování, časové sdílení, desktop a síť. Každé období je spojeno s jednou generací počítačů.[6]

4.1. První generace (1945 – 55)

V tomto období byl sestaven první počítací stroj. Přenos byl velmi pomalý. První počítače byly elektronkové a zabíraly celé místnosti, vyznačovaly se velkou poruchovostí a nízkou rychlostí. Byly milionkrát pomalejší než dnešní nejlevnější počítač. Programovalo se strojovým jazykem. Programovací jazyk byl neznámý (dokonce jazyk assembler neexistoval). Často se zapojovaly propojovací karty ke kontrole základních strojových funkcí. Operační systém neexistoval.[26]

4.2. Druhá generace (1955 – 65)

Druhá generace oplývá tranzistory a dávkovým systémem. Tranzistorové obvody se vyznačovaly slušnou rychlostí.[6] Dochází ke zlepšení parametrů počítačů. Začíná se s nimi obchodovat. Poprvé dochází k rozdělení obsluhy na návrháře, stavitele, operátory, programátory a personál. Programátoři napsali první program na papír (Fortran nebo assembler). Potom ho pomocí děrných pásek zavedli do počítače. Výstup byl prováděn na magnetickou pásku. Na specializovaném počítači byl nakonec převeden na tiskárnu.[26]

Sálové počítače ovládly společné počítačové střediska. Tento jednoduchý sálový počítač stanovil centralizované počítače jako standardní formát výpočtu.[6]

Dávkový systém umožňoval provádění pouze jednoho programu. Pokud je v paměti zaveden pouze jeden program schopný k provádění, tak důsledkem je nedostatečné využití procesoru.[28]

4.3. Třetí generace (1965 – 80)

Třetí generace je počátkem multiprogramování a počítačů s integrovanými obvody.[9] Kroky a pokusy v polovodičové technologii integrovaly vhodné obvody do počítačů. Tyto kroky v hardwarové technologii vyvolaly minipočítačovou éru. Byly malé, rychlé, drahé a nedostupné pro koncového uživatele.[6]

Hlavní podstata multiprogramování tkví v co nejvyšším využití paměti a procesoru.[28]

Princip spočívá v umístění několika programů do paměti a v době, kdy jeden program čeká na dokončení I/O operace, může být procesor využit k provádění jiného programu. Tento princip je hlavním tématem dnešních moderních počítačů.[25]

S multiprogramováním souvisí další pojem timesharing. Několik uživatelů současně přistupuje do systému přes terminál. Procesor sdílí čas mezi více uživatelů.[25]

Timesharing je technika, která umožňuje interaktivní práci více uživatelů. Každý připojený uživatel potřebuje jenom část času CPU. Každý uživatel má pocit, že má počítač sám pro sebe, ale ve skutečnosti se programy, které využívají jednotliví uživatelé, v krátkých časových intervalech střídají.[28]

První timesharing systém se jmenoval CTSS (Compatible Time Sharing System).[25] Byl velmi primitivní v porovnání s pozdějšími systémy.[26]

Vývoj pokračoval dále a byl navržen systém, známý jako MULTICS, který podporoval stovky současně připojených uživatelů sdílejících čas procesoru.[25]

Postupně byl vyvinut operační systém UNIX, který se stal populárním po celém světě. Dále byl navržen standard pro UNIX, nazývaný POSIX, v současné době je podporován u většiny verzí UNIXU.[25]

Většina počítačových výrobců produkovala dvě odlišné skupiny počítačů. První skupinou byly vědecké počítače, které byly využívány k početním operacím. Druhou skupinou byly komerční počítače, které sloužily bankám a pojišťovnám. Firma IBM přišla s novým systémem Systém/360. Tento software byl kompatibilní, tudíž fungoval na obou skupinách počítačů. Operační systém 360 nastartoval dráhu používání integrovaných obvodů v počítačích.[26]

Dalším hlavním rysem třetí generace operačních systémů byla schopnost číst úlohy z karet na disk. Potom jakmile úloha ukončila svou činnost, operační systém načetl novu úlohu z disku do volného paměťového místa. Tato technika se nazývá spooling.[26]

4.4. Čtvrtá generace (1980 – současnost)

Čtvrtá generace je dobou osobních počítačů a mikroprocesorů. Osobní počítače byly zpočátku nazývány mikropočítače. Mikropočítače obsahovaly tisíce tranzistorů. Fenoménem této doby je osobní počítač a operační systém MS-DOS.[11][26]

V této době byly navrženy předchůdci operačního systému CP/M a MS-DOS. Tyto systémy byly založeny na zadávání příkazů z klávesnice.[26]

Bylo vyvinuto grafické uživatelské rozhraní (GUI), dále síťové operační systémy, distribuované systémy a multiprocesorové systémy. Postupem času se systém MS-DOS stával zkamenělinou.[11][26]

Čtvrtá éra je také síťovou formou počítačů. Síťová technologie předstihla procesorovou technologii během devadesátých let. 90. léta byly svědkem představení mnoha komerčních paralelních počítačů s více procesory.[6]

Lokální síť osobních počítačů a pracovních stanic začala nahrazovat sálové počítače a minipočítače. Brzy byly počítače připojeny do rozsáhlých sítí označovaných jako LAN.[6]

4.5. Současný trend

Jeden z jasných trendů je nahrazení drahých paralelních strojů více cenově příznivějšími skupinami pracovních stanic. Pronikání internetu vzbudilo zájem o síťové počítače a kabelovou síť. Tyto počítače by měly být spolehlivé, konzistentní a laciné.[6]

5. Vývoj operačních systémů

Vývoj počítačových systémů za sebou zanechal celou škálu operačních systémů. V této části si stručně nastíníme sedm z nich.[26]

1. *Mainframe operační systém*: OS se vyznačuje obrovskou kapacitou vstupních/výstupních operací. Neuměl zpracovávat více úloh najednou. Neměl interaktivní rozhraní. Nabízel tři druhy služeb: dávkové zpracování, zpracování transakce, sdílení času. Příkladem operačního systému je OS/390. Postupem času mainframe získal interaktivní přístup a grafické terminály.[12][26]

2. *Serverové operační systémy*: Zajišťují síťové služby. Patří sem UNIX, Windows 2000, Linux.[9]

   Běžely na serverech. Najednou umožňovaly práci více uživatelům, sdílení hardware a software.[26]

3. *Víceprocesorové operační systémy*: Speciální operační systém podporující spolupráci více procesorů v jednom systému. Používal se pojem paralelní počítač, multiprocesor, multipočítač. Byl variantou serverového operačního systému.[26]

4. *Operační systém osobních počítačů:* Jejich hlavní úlohou je poskytovat dobrý interfejs jednomu uživateli. Patří sem Windows, Linux, MacOS.[9][26]

5. *Operační systémy pracující v reálném čase:* Klíčovým parametrem je čas. Události v okolí systému probíhají v reálném čase. Správná funkce systému nezávisí pouze na správnosti výpočtů, ale také na tom, kdy jsou výsledky k dispozici. Pojem obvykle používáme pro technické systémy. Řízení letového provozu, řízení průmyslových procesů, řízení laboratorních experimentů.[28]

   Mezi nejznámější systémy patří VxWorks, QNX, RT-Linux.[26]

6. *Vestavěné operační systémy:* Počítačové systémy, které jsou součástí jiných technických systémů. Obvykle představují jejich řídící složku. Jsou schopné pracovat v reálném čase. Použití v PDA, TV sety, mikrovlnky, mobily. [9][28]

   Příkladem jsou PalmOS a Windows CE.[26]

7. *Operační systém čipových karet:* Nejmenší operační systém běžící v kreditních kartách, které obsahují CPU čip. Ovládají jednoduché funkce, mezi které patří elektronické platby.[26]

6. Typy operačních systémů

Operační systémy implementované v počítačích se liší svou použitelností, komplexností a orientací. Např. síťové OS musí koordinovat funkci řady počítačů a řídit komunikaci v počítačové síti. Na druhé straně jsou OS počítačů, které nejsou připojeny do sítě a zajišťují práci pro jednoho uživatele. [16][25][28][30]

1. Operační systémy dělíme podle počtu ovládaných procesorů:
   - *Jednoprocesorové (monoprocesorové)*: V daném okamžiku je v paměti aktivní jeden program (Windows s DOS jádrem).[30][16]
   - *Víceprocesorové (multiprocesorové)*: Počítač s více procesory. Operační systém dokáže naplánovat procesy s využitím všech procesorů. Umožňuje paralelní zpracování. Rozlišuje se symetrický a asymetrický multiprocessing. U symetrického může kterýkoliv proces běžet na kterémkoliv procesoru. Asymetrický multiprocessing má vyhrazen jeden procesor pro systém a uživatelské procesy běží na zbylých procesorech.[23][30]

     V současné době se používá v systémech, které jsou určeny pro více uživatelů. Unixové systémy včetně Linuxu, Windows (2000, XP, Vista).[30]
2. Podle složitosti správy uživatelů:
   - *Jednouživatelské*: V daném okamžiku umožňují práci pouze jednomu uživateli (Windows s DOS jádrem).[30]
   - *Víceuživatelské*: Mají propracovanou správu uživatelů a musí zajistit přístup do systému pro více uživatelů současně. Jedná se o tzv. serverové OS (unixové systémy, Windows XP, Windows Vista).[30]
3. Podle počtu zpracovávaných programů:
   - *Jednoprogramové*: V daném čase může být spuštěn pouze jeden program.[30]
   - *Víceprogramové*: Umožňují spouštět více programů najednou. Dále zde rozlišujeme dvě podskupiny, víceúlohové a jednoúlohové. Víceúlohové systémy (multitaskové) ještě navíc povolují sdílení prostředků mezi procesy daných programů (správa paměti, sdílení tiskárny). [30]

     Jednoúlohové systémy řeší problém odstaveného programu v paměti. Odložení veškerého paměťového prostoru na vnější medium a následné obnovení stavu ve chvíli, kdy program chce pokračovat ve své činnosti.[30]
4. Podle schopnosti práce v síti:
   - *Lokální*: Mezi lokální OS řadíme ty, které jsou určeny pro správu systémových prostředků samostatného PC. V architektuře klient-server mohou být pouze klienty. Příkladem je Windows s DOS jádrem. [30]

- *Síťové*: V architektuře klient-server mají kromě klientské části také serverovou část.[30]

5. Podle míry specializace:
   - *Speciální*: Mají jasně stanovený účel. Jsou určené pro jeden typ úloh.[30]
   - *Univerzální*: Běžné operační systémy, které jsou implementovány na počítači. Řeší různé typy úloh.[30]
6. Distribuované systémy: Systém pracuje na více počítačích v síti. Program může být rozdělen na samostatné části mezi více počítačů, které mezi sebou navzájem komunikují.[30]

   "Distribuovaný operační systém je samostatný OS běžící na síti procesorů, které nesdílejí společnou paměť, a zároveň poskytuje uživateli dojem jednoho počítače."[30]

7. Pokročilé počítačové architektury a paralelní zpracování

Počítačový architekti se vždycky snažili zvýšit výkon svých architektur. Hardwarová technologie jde neslýchanou rychlostí dopředu. Nicméně tento trend brzy přijde na konec, jelikož jsou nějaké fyzické hranice počítačové síly.[6]

Paralelní procesory jsou počítačové systémy skládající se z více základních jednotek, které jsou vzájemně propojeny přes společnou síť. Dále je potřebný software, který zajistí, aby základní jednotky pracovaly společně. Základní jednotky spolu mohou komunikovat buď přes sdílenou paměť, nebo metodou komunikace předávání zpráv.[6]

Paralelní počítač je počítač, který provádí paralelně více než jednu základní instrukci, což je zajištěno tím, že počítač obsahuje více procesorů.[10]

Hlavní důvod pro použití multiprocesorů, je vytvořit výkonný počítač jednoduše spojením více procesorů. Multiprocesor předpokládá dosažení větší rychlosti než nejrychlejší jednoprocesorový systém.[6]

7.1. Flynnova klasifikace počítačových architektur

Nejpopulárnější klasifikace počítačové architektury byla definována v roce 1960. Flynnova klasifikace je založena na představě toku informací. Do procesoru proudí dva typy informací: data a instrukce. Instrukční tok je definovaný jako sekvence instrukcí vykonaných

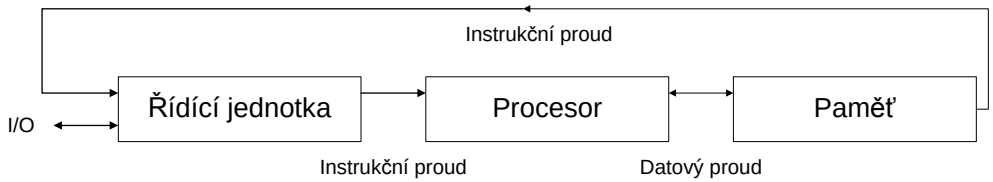
základní jednotkou. Datový tok je definován jako data, které si mezi sebou vymění paměť a základní jednotka. Podle Flynnovy architektury jsou instrukční a datové toky buď jednoduché, nebo vícenásobné. Počítačové architektury dělíme do čtyř kategorií.[6]

1. SISD (zpracovávají jeden instrukční a jeden datový tok)
2. SIMD (zpracovávají jeden instrukční tok a více datových toků současně)
3. MISD (zpracovávají více instrukčních toků současně a jeden datový tok)
4. MIMD (zpracovávají více instrukčních toků současně a více datových toků současně)[10]

7.1.1. SISD architektura

Běžný jednoprocesorový počítač.[11] SISD systém představuje tradiční Von Neumannův počítač.[6] Obsahuje jeden procesor, který pracuje s jedním tokem instrukcí. Instrukce jsou prováděny sekvenčně.[10]

SISD architektura se skládá z jednoho procesoru, který zpracovává jednu množinu dat jedním proudem instrukcí.[28] Tento počítač je označován jako sekvenční nebo uniprocesorový.[10]



Obrázek č. 4: SISD architektura [6]

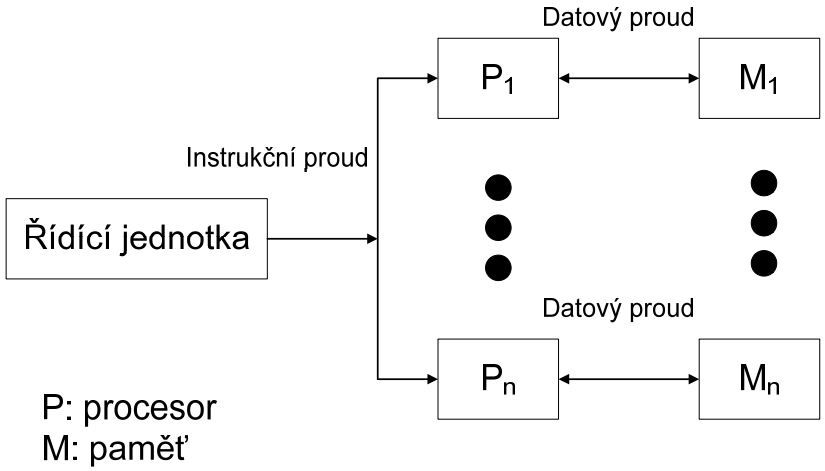
7.1.2. SIMD architektura

„Jedním proudem instrukcí se ve více procesorech zpracovává více různých množin dat. Každá instrukce programu se provede současně v n procesorech, přičemž v každém procesoru se zpracovává jiná množina dat“.[28]

SIMD model se skládá ze dvou částí. První částí je počítač Von Neumannova stylu a druhou je pole procesoru. Procesorové pole se skládá ze sady stejných synchronních elementů, které jsou schopny současně vykonávat stejné operace na různých datech.[6]

Každý procesor v poli má malou velikost lokální paměti, kde pobývají distribuovaná data, dokud je zpracovává paralelně. Pole procesoru je na začátku připojené k paměťové

sběrnici.[6] Všechny procesory jsou řízeny jedním tokem instrukcí. Provádějí v daném okamžiku stejnou instrukci, ale s různými daty.[10] Procesory buď nedělají nic, nebo přesně stejné operace ve stejný čas. V SIMD architektuře je souběžnost využívána použitím současně probíhajících operací přes obrovské toky dat.[6]



Obrázek č. 5: SIMD architektura [6]

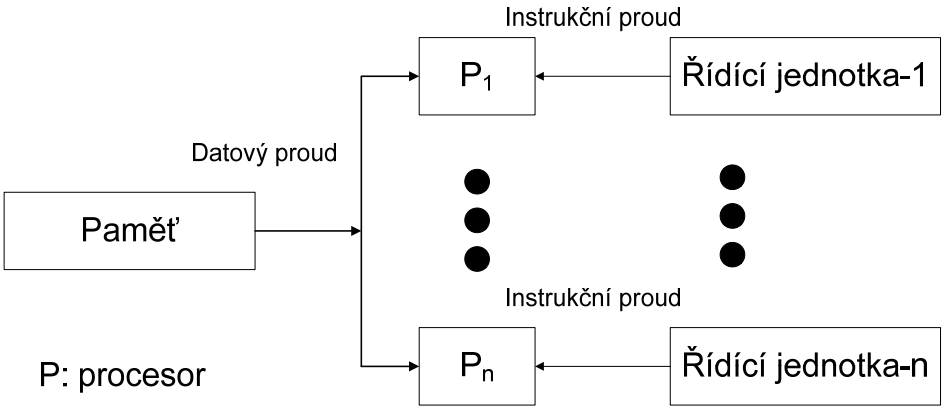
V SIMD architektuře jsou použity dvě konfigurace:

1. Každý procesor má svou vlastní lokální paměť. Procesory mohou komunikovat s každou další přes vzájemně propojenou síť. Pokud síť neposkytuje přímé spojení mezi danou dvojicí procesorů, potom si tato dvojice může vyměnit data přes mezilehlý procesor.[6] Příkladem této konfigurace je počítač Illiac IV.[10]
2. V druhém případě procesory a paměťové moduly komunikují s každou další přes vzájemně propojenou síť. Dva procesory mohou přenášet data mezi každým dalším přes mezilehlé paměťové moduly, nebo je možné přes mezilehlé procesory.[6]

7.1.3. MISD architektura

Architektura využívá jednu množinu dat, která je předána více procesorům, z nichž každý provádí jinou posloupnost instrukcí.[28]

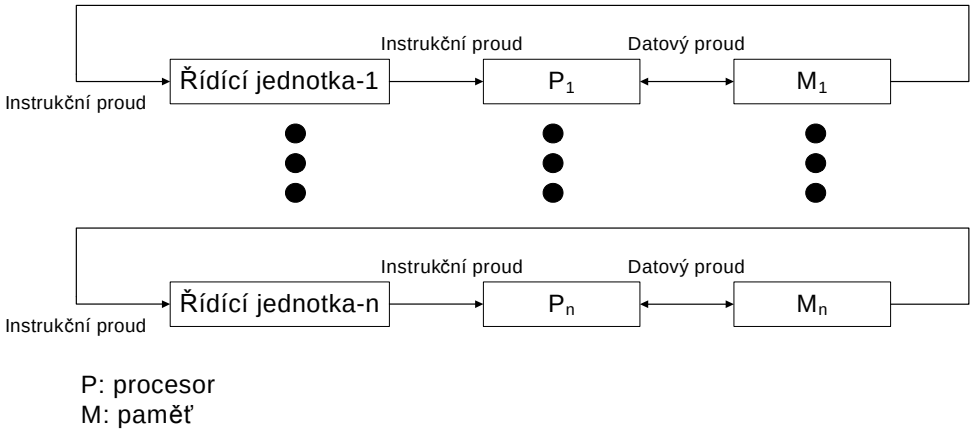
Tyto počítače se dále nerozšířily a nikdy nebyly realizovány.[10][28] V praxi není schopný MISD stroj růst, ale někteří autoři považují systolické pole procesorů za příklad MISD.[6]



Obrázek č. 6: MISD architektura [6][10]

7.1.4. MIMD architektura

Vícenásobné instrukce, vícenásobné datové toky jsou vytvořeny více procesory a vícenásobnými paměťovými moduly, které jsou spojené společně přes nějakou síť.[6] MIMD je kombinace procesorů, které souběžně (paralelně) zpracovávají odlišnými toky instrukcí odlišné množiny dat.[28] Spadají do dvou skupin. Systém se sdílenou pamětí nebo systém, který komunikuje zasíláním zpráv.[6]



Obrázek č. 7: MIMD architektura [6]

Procesory si vyměňují informace přes hlavní sdílenou paměť v systémech sdílení paměti a v systémech zasílání zpráv si vyměňují informace přes vzájemně propojenou síť. Systém sdílené paměti vykonává meziprocesorové sladění přes společnou paměť, sdílenou všemi procesory. Jsou to typické serverové systémy. Přístup do sdílené paměti je vyvážen, proto jsou tyto systémy nazývané SMP (symetrický multiprocesorový systém).[6] Všechny procesory jsou stejného typu.[28] Každý procesor má stejnou příležitost pro čtení/zápis z/do paměti a mají stejnou přístupovou rychlost.[6]

Systém zasílání zpráv (označovaný jako distribuovaná paměť) obsahuje lokální paměť a procesor v každém uzlu sítě. Toto není společná paměť, tak je nutné přesouvat data z jedné lokální paměti do další pomocí zasílání zpráv.[6]

Distribuovaná paměť je způsob, jak efektivně zvýšit počet procesorů, které jsou ovládány paralelním systémem. Jestli by rozšiřitelnost větších systémů pokračovala, měly by se používat techniky distribuované paměti.[6]

1. *Organizace sdílené paměti* – V modelu sdílené paměti procesory komunikují zápisem nebo čtením paměťových míst ze sdílené paměti, která je přístupná všem procesorům. Každý procesor obsahuje registry, vyrovnávací a dočasnou paměť a lokální paměťové bloky jako další paměťové zdroje. Při návrhu systému sdílené paměti musíme brát v úvahu řadu problémů. Musíme vyřešit řízení přístupu, synchronizaci, ochranu a bezpečnost. Model řízení přístupu vytvoří požadovanou kontrolu pro každou žádost o přístup, vydanou procesorem do sdílené paměti. Synchronizace zajistí, že informace proudí správně a že systém je funkční. Ochrana zabraňuje libovolnému přístupu procesorů do zdrojů, které patří jiným procesorům. Sdílení povoluje přístup.[6]
2. *Organizace zasílání zpráv* – Systémy zasílání zpráv jsou typem multiprocesorů, v kterých má každý procesor přístup ke své vlastní lokální paměti. Na rozdíl od systémů sdílené paměti je komunikace v systémech zasílání zpráv vykonávána pomocí poslání a přijmutí operací. Uzel se v takovém systému skládá z procesoru a jeho sdílené paměti. Procesory nesdílejí společnou paměť, každý si vybírá svůj vlastní adresový prostor.[6]
3. *Vzájemně propojené sítě* – Síť multiprocesorů může být klasifikována na základě více kritérií.[6]
    - Druh výroby (synchronní nebo asynchronní)

    V synchronním druhu je využívána jediná globální časová základna všemi komponenty v systému. Naproti tomu asynchronní systém nepožaduje globální časovou základnu. Místo toho jsou použity signály k navázání spojení, za účelem sladit operace těchto systémů.[6]

- Strategie řízení (centralizované nebo decentralizované)

V centralizovaném řídicím systému je použita jedna centrální jednotka k řízení operací všech komponent. V decentralizovaném řízení je řídící funkce rozdělena mezi různé komponenty v systému.[6]

- Spínací techniky (přepojování okruhů versus přepojování paketů)

V obvodu přepojování okruhů musí být stanovena úplná cesta mezi zdrojem a cílem před startem komunikace. V zařízení na přepojování paketů se komunikace mezi cílem a zdrojem uskutečňuje prostřednictvím zpráv, které jsou rozděleny do menších entit nazývaných pakety.[6]

- Topologie (statická nebo dynamická)

Ve statické síti je pevně stanoveno spojení mezi uzly. Oproti tomu v dynamické síti je spojení definováno jako potřebné. Spínací člen je použit k navázání spojení mezi vstupy a výstupy.[6]

8. Bezpečnost operačních systémů

V dnešní vyspělé době počítačových technologií je bezpečnost operačních systémů na prvním místě v cestě za stabilním a zabezpečeným systémem.[25] Každý uživatel, který je se svým systémem online se světem, si musí uvědomit nebezpečí útoků a hrozeb, které pocházejí ze světa internetu.

8.1. Bezpečnostní hrozby

Pro porozumění typům hrozeb, které existují, si potřebujeme definovat bezpečnostní požadavky:

1. Důvěrnost: Vyžaduje, aby informace v počítačovém systému byly přístupné pouze ověřeným stranám.
2. Integrita: Požaduje, aby počítačový systém mohl být modifikován pouze oprávněnou stranou.
3. Dostupnost: Požaduje, aby počítačový systém byl dostupný pro oprávněnou stranu.
4. Pravost (ověření): Požaduje, aby počítačový systém byl schopný ověřit identitu uživatele.[26]

Z bezpečnostního pohledu má počítačový systém tři obecné cíle, které odpovídají hrozbám v níže uvedené tabulce č. 1. Důvěrnost dat se týká tajných dat. Integrita dat znamená, že neautorizovaný uživatel by neměl být schopný modifikovat data. Dostupnost systému znamená, že nikdo nemůže narušit systém a udělat ho nepoužitelným.[26]

Tabulka č. 1: Bezpečnostní cíle a hrozby [26]

Cíl	Hrozba
Důvěrnost dat	Ohrožení dat
Integrita dat	Falšování dat
Dostupnost systému	Odmítnutí služby

8.2. Typy hrozeb

Typy útoků na bezpečnost počítačového systému nebo síť jsou nejlépe charakterizovány sledovací funkcí počítačového systému a následným poskytnutím informace. Obvykle se to děje tokem informací ze zdroje (soubor nebo část hlavní paměti) k cíli (další soubor nebo uživatel). Nyní si nastíníme čtyři obecné kategorie útoku:

1. Přerušení: Aktivní část systému je zničena nebo se stane nedostupnou (zničení části hardwaru, např. pevný disk).
2. Zachycení: Neautorizovaná strana (osoba, program) získá přístup do aktivní části systému. Příkladem je zachycení dat v síti, nezákonné kopírování souborů.
3. Modifikace (pozměnění): Neautorizovaná strana nejenom že získá přístup, ale také může manipulovat se systémem. Příkladem může být změnění hodnot v datovém souboru, pozměnění obsahu zprávy zaslané v síti.
4. Zhotovení padělku: Neautorizovaná strana vloží padělaný objekt (vložení falešné zprávy do sítě) do systému.[25]

8.3. Části počítačového sytému

Části počítačového systému můžou být kategorizovány jako hardware, software, data, přenosové linky a sítě. Hardware je nejvíce zranitelný vůči útokům. Útoky jsou směřovány k úmyslnému zničení zařízení. Klíčovým útokem na software (operační systém, utility, uživatelský program) je útok na dostupnost. Obzvláště uživatelský program je snadno

smazatelný. Software může být snadno změněn nebo poškozen. Následující tabulka ukazuje podstatu útoků na jednotlivé části. U bezpečnosti dat se jedná o neautorizované čtení z datových souborů nebo databází. Útoky na bezpečnost sítě můžeme klasifikovat na pasivní a aktivní. Pasivní útoky se pokoušejí o použití informací ze systému, ale s cílem nenarušit systém. Aktivní útoky se pokoušejí pozměnit systém nebo narušit jeho činnost.[25]

Tabulka č. 2: Části počítačového systému a útoky [25]

	Dostupnost	**Utajení**	**Integrita/Pravost**
Hardware	Zařízení je ukradeno nebo deaktivováno		
Software	Programy jsou smazány, odmítnutí přístupu uživatele	Je udělána nelegální kopie softwaru	Běžící program je modifikován, buď to způsobí chybu během vykonávání, nebo to udělá nějakou neúmyslnou činnost.
Data	Soubory jsou smazány, odmítnutí přístupu uživatele	Je vykonáno neoprávněné čtení	Existující soubor je modifikován nebo je vytvořen nový soubor
Přenosové linky	Zprávy jsou zničeny nebo vymazány, síť je nedostupná	Zprávy jsou přečteny, zprávy jsou sledovány	Zprávy jsou modifikovány, zpožděny, převytvářeny nebo okopírovány. Je vytvořena falešná zpráva.

8.4. Ochrana

Úvod do multiprogramování přinesl schopnost sdílet zdroje mezi uživatele. Toto sdílení nezahrnuje jenom procesor, ale také paměť, I/O zařízení, programy a data. Tato schopnost sdílení zdrojů je úvodem do potřeby ochrany. Operační systém nabízí celou škálu ochran.[25]

1. Bez ochrany: Toto je vhodné, když utajované programy běží v izolované době.
2. Izolace: Tento přístup naznačuje, že každý proces funguje odděleně od dalších procesů (nic nesdílí a nekomunikuje). Každý proces má vlastní adresový prostor a soubory.

3. Sdílej všechno nebo nic: Vlastník objektu (např. soubor nebo paměť) prohlašuje, jestli bude veřejný (každý proces může přistoupit k objektu) nebo privátní (jenom vlastník procesu může přistoupit k objektu).
4. Sdílení přes omezený přístup: Operační systém kontroluje přístupnost každého přístupu uživatele k určitému objektu.
5. Sdílení přes dynamické schopnosti: Povoluje dynamické vytvoření práv pro objekt.
6. Omezené použití na objekt: Tato forma ochrany neomezuje právě přístup k objektu, ale který objekt smí být použit.[25]

Výše uvedené body dávají operačnímu systému možnost použití různých stupňů ochrany pro různé objekty, uživatele nebo aplikace.[25]

8.5. Útočníci

V literatuře o bezpečnosti, lidé, kteří dělají problémy tam, kde nemají byznys, jsou označovány jako útočníci a někdy protivníci.[26]

Popis jednotlivých útoků:

1. Zamaskování: Jednotlivec, který neautorizovaně použije počítač a pronikne do systému přes oprávněný uživatelský účet.
2. Zneužití: Uživatel, který zneužije práva.
3. Utajený uživatel: Jednotlivec, který dohlíží na kontrolu systému a používá tuto kontrolu ke zkoumání účtů.[25]

8.6. Techniky vniknutí

Cílem útočníka je získat přístup do systému nebo zvýšit rozsah přístupových práv do systému. Útočník chce získat informace, které jsou chráněny. V mnoha případech touto informací je uživatelské heslo. Heslo může být chráněno jedním ze dvou způsobů:

1. Obyčejné šifrování: Systém uloží pouze šifrovanou formu uživatelského hesla. Jakmile uživatel zadává heslo, systém zašifruje heslo a porovná ho s uloženými hodnotami.
2. Řízení přístupu: Přístup k chráněnému souboru je omezen jedním nebo několika účty.[25]

Techniky pro odhadnutí hesla:

- Zkusit defaultní nastavení účtů, které je dodáváno se systémem.
- Zkusit všechna krátká hesla (od jednoho do tří znaků).
- Zkusit hesla ze seznamu pravděpodobných hesel (hackerské publikace, internet).
- Posbírat informace o uživateli (jeho úplné jméno, jména rodinných příslušníků, jeho koníčky).
- Zkusit uživatelské telefonní čísla, čísla přátel, čísla místnosti a bydliště.
- Zkusit všechny oprávněné licence.
- Použít trojského koně.
- Odposlouchávat telefonní rozhovory.[25]

8.7. Vytvoření bezpečného hesla

Bezpečným heslem můžeme nazvat takové, které není snadno zjistitelné, je těžce uhodnutelné nebo jinak snadno zneužitelné.[1]

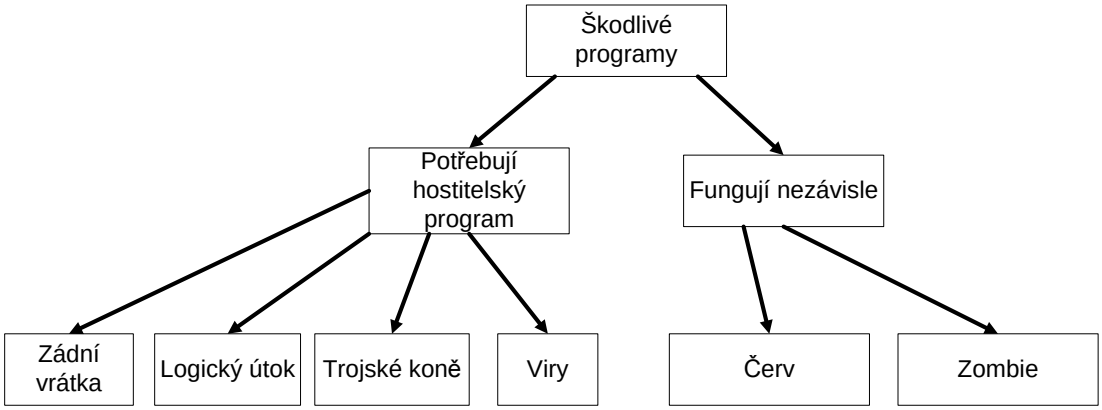
Ideální přístupové heslo by mělo být velmi dlouhé a mělo by obsahovat vybrané znaky z množiny všech přípustných znaků.[7]

Jakým způsobem vybrat heslo:

1. Vymyšlení takového hesla, které bude obsahovat malá a velká písmena, čísla nebo i zvláštní symboly a neobsahuje slovníkové výrazy.
2. Heslo můžeme vygenerovat náhodně pomocí generátoru náhodných hesel.
3. Vymyslet heslo s použitím mnemotechnických pomůcek. Vymyslet větu, která se nám bude dobře pamatovat (mám velkého bílého psa, který se jmenuje Adam). První písmena slov věty nám poslouží jako heslo (mvbpksja).[7]

8.8. Škodlivý software

Nejvíce promyšlené typy útoků na počítačový systém představují programy, které využívají zranitelnosti počítačových systémů. Týká se to jak aplikačních programů, tak utilit. Obecným názvem pro takové hrozby je škodlivý software nebo malware. Malware je software, který je navržen k tomu, aby způsobil škodu.[25]



Obrázek č. 8: Klasifikace škodlivých programů [25]

Na obrázku číslo 8 vidíme rozdělení škodlivých programů. Tyto hrozby můžeme rozdělit do dvou kategorií: které potřebují hostitelský program a které ho nepotřebují. Původem první kategorie jsou v podstatě části programů, které nemůžou existovat nezávisle na nějakém aktuálním aplikačním programu, utilitě nebo systémovém programu. Druhou kategorií jsou samostatné programy, které běží v operačním systému.[25]

Škodlivý software můžeme také rozlišit na ten, který dělá kopii a který nedělá kopii. Činitelem jsou části programů, které jsou aktivovány, když hostitelský program vyvolá nějakou určitou funkci. Tyto části programů se skládají buď z virů, nebo červů, které když se spustí, tak vytvoří jednu nebo více kopií sebe samého.[25]

8.8.1. Zadní vrátka

V praxi je tento útok znám pod pojmem back door nebo trap door. Tímto pojmem se označují chyby v nejrůznějších programech. Tyto chyby jsou buď úmyslné, nebo jsou vytvářeny záměrně. Back door jsou příležitostí pro neoprávněný průnik do zabezpečeného systému jak pro lidské útočníky, tak pro většinu virů a červů.[15]

Trap door je metoda, umožňující útočníkovi obejít běžnou autentizaci uživatele, která je požadována při vstupu do systému nebo do programu. Tyto útoky bývají skryté před běžnou kontrolou. Bezpečnost operačního systému obejdou např. tím, že se vydávají za webový prohlížeč. Může mít podobu samostatného programu nebo se jedná o modifikaci původního systému.[33] Zadní vrátka jsou oprávněně používány programátory k ladění a testování programů, které mají nějaké procedury k ověření identity uživatele.[25]

Zadní vrátka se stávají hrozbou, když jsou používány programátory k získání neautorizovaného přístupu.[25]

Je obtížné chránit operační systém proti backdoors. Bezpečnost musí být založena na vývoji programů a na aktualizaci softwaru.[25]

8.8.2. Logická bomba

Jeden z nejstarších typů útoků. Logická bomba je škodlivý kód vložený do programu, nebo může být i samostatnou aplikací, která je připravena vypuknout při splnění určitých podmínek (stisk příslušné kombinace kláves, určité datum, přítomnost nebo absence jistých souborů). Rozdíl mezi logickou bombou a virem je v tom, že vir vstupuje do počítače bez vědomí uživatele, ale bomba je do počítače dávána s nekalým záměrem.[25][20]

Jakmile se bomba spustí, změní nebo smaže data nebo celé soubory a poškodí počítač. Příkladem tohoto útoku může být zaměstnanec firmy, který umístí kód do informačního sytému před svým odchodem a cílený útok se provede po několika dnech (bomba se spustí).[25][20]

8.8.3. Trojský kůň

Trojský kůň je část programu nebo aplikace, která se tváří užitečně (nějaký spořič nebo hra). Tato část programu obsahuje skrytý kód, který když se vyvolá, tak spustí nějaké nechtěné nebo škodlivé funkce.[25][27]

Jako příklad si můžeme uvést trojského koně, který získá přístup k cizímu souboru dalším uživatelům na společně používaném systému.[25]

Může se vyskytnout v podobě volně šiřitelného spořiče, který když se spustí, tak začne otevírat porty počítače a tím poskytuje vzdálený přístup.[27]

Další běžnou motivací trojského koně je zničení dat. Tento program se jeví vykonáváním užitečné funkce, zatímco tiše maže uživatelské soubory.[25]

Mezi funkce trojského koně patří odposlouchávání přístupových jmen a hesel (sniffer), sledování znaků zadávaných z klávesnice (keylogger), sleduje uživatele při surfování na internetu (spyware), zablokuje software pro zabezpečení počítače (security software disabler).[27]

8.8.4. Viry

Virus je typ programu, který infikuje další programy a dokáže se šířit sám, tím že vytváří modifikaci sebe samotného. Modifikace zahrnuje kopii virového programu, který může pokračovat v infikování dalších programů.[25][18]

Jako virus se v oblasti bezpečnosti operačních systémů označuje program, který se dál šíří v počítači bez vědomí uživatele. Pro rozmnožení se vkládá do dalších souborů a dokumentů. Funguje podobně jako normální biologický virus, který se šíří vkládáním malých kousků genetického kódu do živých buněk.[25][18]

Šíření viru se označuje jako infekce či nakažení a napadený soubor jako hostitel. Některé viry jsou ničivé (chtějí smazat soubory) a některé jsou pouze obtěžující. Množení virů zatěžuje operační systém a plýtvá jeho zdroji.[18]

8.8.5. Červi

Červi jsou zvláštním typem počítačového viru, kteří využívají síťové připojení k rozšíření ze systému do dalšího systému. Jakmile se počítačový červ stane aktivním uvnitř systému, může se zachovat jako počítačový virus, trojský kůň nebo může vykonat několik ničivých akcí. Infikovaný systém červ využije k šíření dalších svých kopií po internetu. Červi využívají bezpečnostních děr v operačním systému nebo v software.[25][17]

Nejčastějším typem jsou e-mailové červi, kteří se šíří v přílohách elektronické pošty do dalších systémů.[17]

Síťovým červům je obtížné se bránit, nicméně síťovou bezpečností a bezpečností počítačového systému lze minimalizovat jejich útoky.[25]

8.8.6. Zombie

Zombie je program, který tajně převezme další počítače připojené k internetu a potom tyto počítače používá ke spuštění útoků. Jsou používány k útokům zvaných odmítnutí služby, typicky na webové stránky. Při tomto útoku dochází k přehlcení požadavky a má to za následek nedostupnost služby.[25][5]

8.9. Antivirové přístupy

Ideálním řešením je ochrana proti virům. Nedovolit viru dostat se do systému. Hlavním cílem je obvykle snížit počet úspěšných virových útoků. Přístupy jsou následující:

1. Odhalení: Jakmile se nákaza objeví, určit, co napadla a lokalizovat virus.
2. Identifikace: Jakmile bylo odhalení dokončeno, musí se rozpoznat konkrétní virus, který infikoval program.
3. Odstranění: Jakmile byl konkrétní virus rozpoznán, musí se odstranit všechny stopy viru v infikovaném programu.[25]

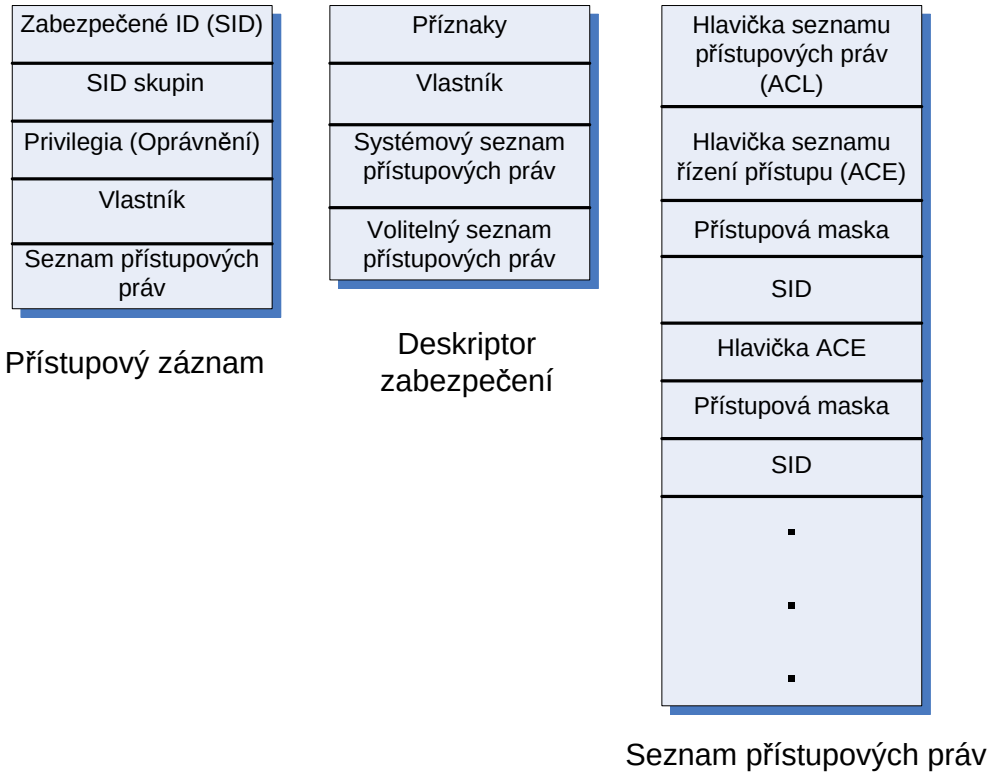
8.10. Zabezpečení Windows

Windows poskytuje jednotný prostředek pro řízení přístupu, který používá procesy, vlákna, soubory, semafory a další objekty. Řízení přístupu je řízeno dvěma entitami:

- Přístupový záznam: Souvisí s každým procesem. Ukazuje na privilegia, která patří uživateli. Povoluje každému procesu modifikovat jeho bezpečnostní parametry.
- Deskriptor zabezpečení: Souvisí s každým objektem, pro který je možný meziprocesový přístup. Hlavní částí je seznam přístupových práv, který specifikuje přístupová práva jednotlivých uživatelů a skupin pro daný objekt.[25]

8.10.1. Schéma řízení přístupu

Když se uživatel přihlašuje do systému Windows, je požadováno heslo a jméno pro ověření uživatele. Jestli je přihlášení akceptováno, vytvoří se proces pro uživatele a přístupový záznam, který souvisí s procesem.[25]



Obrázek č. 9: Struktura bezpečnosti Windows [25]

8.10.2. Přístupový záznam

Struktura přístupového záznamu zahrnuje následující parametry:

- Zabezpečené ID (SID): Jednoznačně identifikuje uživatele přes všechny počítače na síti. Zpravidla souhlasí s uživatelským přihlašovacím jménem.
- SID skupin: Seznam skupin, do kterých uživatel patří. Skupina je jednoduše soubor uživatelských ID, které jsou identifikovány, jako skupina za účelem řízení přístupu. Každá skupina má jednoznačné SID.
- Privilegia (oprávnění): Seznam bezpečnostních systémových služeb, které uživatel smí vyžadovat.
- Implicitní vlastník: Zpravidla vlastník nového procesu je stejný jako vlastník potomka existujícího procesu. Nicméně uživatel může specifikovat, který vlastník procesů bude potomkem procesu ve skupině, do které uživatel patří.
- Implicitní seznam přístupových práv: Tento počáteční seznam ochrany používá objekt, který uživatel vytvoří.[25]

8.10.3. Deskriptor zabezpečení

Má následující parametry:

- Příznaky: Definuje typ a obsah deskriptoru zabezpečení.
- Vlastník: Vlastník objektu může vykonávat nějaké akce na deskriptoru zabezpečení.
- Systémový seznam přístupových práv: Specifikuje, jaké druhy operací na objektu by měly být generovány kontrolními hláškami.
- Volitelný seznam přístupových práv: Určuje, které operace mohou uživatelé a skupiny provádět s objektem.[25]

8.11. Bezpečnost Windows XP

Firmě Microsoft je velmi často vytýkáno, že vytváří nezabezpečené systémy s bezpečnostními dírami a dalšími zranitelnostmi. Naopak poslední dobou si Microsoft vede velmi dobře a jeho systémy patří mezi nejbezpečnější.[19]

Následující kroky by měly být základem bezpečnosti počítačů s operačními systémy Microsoft Windows XP Home Edition a Windows XP Professional.[31]

1. ***Ověřit, že všechny diskové oddíly jsou naformátovány na NTFS***: NTFS oddíly nabízejí ochranu, která není dostupná pro souborové systémy FAT. Ujistěte se, že všechny oddíly na vašem počítači jsou zformátované do NTFS. Jestli je nutné převedení FAT oddílu do NTFS, tak použijte nějaký potřebný diskový nástroj (např. Partition Magic).[31]
2. ***Chránit sdílení souborů***: Standardně systém Windows XP, který není připojený do domény, používá zjednodušené sdílení souborů, ve kterém bude nutné používat účet Guest pro veškeré pokusy o přihlášení do počítače z druhé strany sítě. Toto zjednodušené sdílení souborů je určené k použití na domácí síti za firewallem.[31]
3. ***Použít službu Sdílení připojení k Internetu pro sdílení internetového připojení***: Windows poskytují schopnost k tomu, aby sdílela jednotlivé internetové spojení s více počítači na domácích sítích přes Sdílení připojení k Internetu (ICS). Jeden počítač, nazvaný ICS hostitel, spojuje přímo do Internetu a sdílí spojení se zbytkem počítačů na síti. Klientské počítače se spoléhají na hlavní počítač, který poskytuje přístup k Internetu.[31]
4. ***Povolit Bránu firewall pro připojení k Internetu***: Navrženo pro použití v domácí síti nebo v malých podnicích, windows firewall poskytuje ochranu pro počítače s Windows XP, které jsou přímo připojeny k Internetu. Funguje jako „ochranná hráz“ proti útokům zevnitř i z venku.[31][13]
5. ***Použít software Restriction Policies***: Pomocí tohoto nástroje mohou administrátoři identifikovat a ovládat běžící software. Používáním tohoto softwarového nástroje můžeme zabránit v běhu některým programům, jako jsou viry, trojské koně nebo další software, který může způsobit konflikty na počítači.[31]
6. ***Použít heslo k účtu***: Přidělením hesla k místnímu účtu odstraníme ochranu, která zabraňuje přihlášení přes síť. Také povoluje účtu přistupovat k prostředkům, oprávněných používat po přístupu, dokonce přes celou síť. Důsledkem toho je, že je lepší nechat prázdné heslo u účtu než přiřadit heslo, které by bylo slabé a tudíž i snadno odhadnutelné. Když budeme přiřazovat heslo k účtu, mělo by mít minimálně devět znaků a uvnitř prvních sedmi by mělo být interpunkční znaménko nebo netisknutelný znak ASCII.[31]

7. ***Zakázat nepotřebné služby***: Po instalaci Windows XP bychom měli zakázat síťové služby, které nepožadujeme pro počítač. Obzvláště bychom měli zvážit, zda náš počítač potřebuje nějaké webové služby. Standardně tato služba není nainstalovaná jako součást Windows XP, měla by být nainstalovaná pouze, pokud je specificky vyžadována.[31]
8. ***Zakázat nebo smazat zbytečné účty***: Zakažte nebo smažte všechny neaktivní účty, které nejsou vyžadovány.[31]
9. ***Ujistit se, že účet Guest je zakázán***: Toto nastavení se doporučuje používat pro počítače s Windows XP Professional, které náležejí do domény nebo pro počítače, které nepoužívají zjednodušené sdílení souborů. Nastavením této vlastnosti zabráníme pokusu uživatelům, kteří se chtějí přihlásit přes síť do počítače prostřednictvím účtu Guest.[31]
10. ***Nastavit silnější politiku hesel***: Microsoft navrhuje udělat následující čtyři změny:
    - Heslo by mělo být nejméně osm znaků dlouhé.
    - Nastavit minimální životnost hesla, typicky mezi 1 až 7 dny.
    - Nastavit maximální životnost hesla, nejvíce na 42 dní.[31]
11. ***Nastavit uzamykání účtů***: Microsoft doporučuje používat uzamykání účtů. Administrátor nastaví zamknutí účtu po několika neúspěšných pokusech o přihlášení. Toto nastavení umožňuje zabránit útokům hackerů, kteří se snaží odhadnout heslo.[31]
12. ***Nainstalovat antivirus a aktualizace***: Jednou z nejdůležitějších věcí pro ochranu systému je použití antivirového softwaru. Antivir by měl být aktualizován.[31]
13. ***Udržovat aktuální bezpečnostní aktualizace***: Automatické aktualizace ve Windows XP mohou automaticky zjistit a nainstalovat poslední bezpečnostní záplaty od Microsoftu.[31]

8.12. Zabezpečení uživatele

1. Volit bezpečná hesla.
2. Bezpečně uchovávat hesla.
3. Zamykat desktop při odchodu z kanceláře.
4. Nepřihlašovat se privilegovaným účtem.
5. Spouštět jen důvěryhodné programy.
6. Neotevírat podezřelé přílohy.

7. Nestát se obětí klamavých informací.
8. Dodržovat firemní pravidla informační bezpečnosti.
9. Nepokoušet se obejít bezpečnostní nastavení.
10. Hlásit podezřelé události.[2]

8.13. Přednosti a nedostatky MS Windows XP

Přednosti:

- Vylepšená ochrana kódu.
- Ochrana systémových souborů Windows.
- Šifrování systému souborů.
- Zabezpečení protokolu IP (IPSec).
- Podpora protokolu Kerberos.
- Správce pověření.
- Soubory a složky offline.
- Funkce Vzdálená pomoc.
- Brána firewall pro připojení k Internetu.[32]

Nedostatky:

- Snadné napadnutí škodlivým softwarem.
- Defaultní nastavení přihlášení jako administrátor.

8.14. Souhrn k bezpečnosti operačních systémů

Požadavky na bezpečnost jsou nejlépe určeny prozkoumáním jednotlivých hrozeb zaměřených na konkrétní organizaci. Přerušení služby je hrozbou k dosažitelnosti. Zachycení informací je ohrožení utajení. Jak modifikace oprávněné informace, tak neoprávněné zhotovení informace vedou k ohrožení integrity.[25]

Stále více hrozeb je představováno viry a jednoduchými softwarovými mechanismy. Tyto hrozby využívají zranitelnosti systému, buď získáním neautorizovaného přístupu k informacím nebo znehodnocením systémových služeb.[25]

9. Procesy v operačním systému

Počítačová platforma se skládá z procesoru, hlavní paměti, I/O modulů, čítačů a disků. Počítačové aplikace jsou vyvinuty k vykonávání úkolů (procesů). Přijmou vstup z vnějšího světa, provedou zpracovaní a generují výstup. Všechny moderní operační systémy se spoléhají na model, ve kterém spuštění aplikace odpovídá jednomu nebo více procesům.[25]

Hlavním úkolem operačního systému je:

1. Prokládat (časově) provádění mnoha procesů s cílem maximalizovat využití procesoru.
2. Přidělovat procesům požadované systémové prostředky (paměť, periferie, soubory).
3. Podporovat komunikaci procesů a vytváření nových procesů uživatelem.[28]

Všechny moderní počítače mohou vykonávat několik věcí ve stejný čas. Zatímco běží uživatelský program, počítač může číst z disku nebo tisknout.[26]

9.1. Proces

Proces je konkrétní provedení určitého programu. Označujeme ho také jako task (úkol). Průběh procesu lze trasovat. Po každé instrukci může být zastaven a analyzován.[28]

Dávka = úloha + úloha + …. + úloha[28]

Úloha = úkol + úkol + …. + úkol[28]

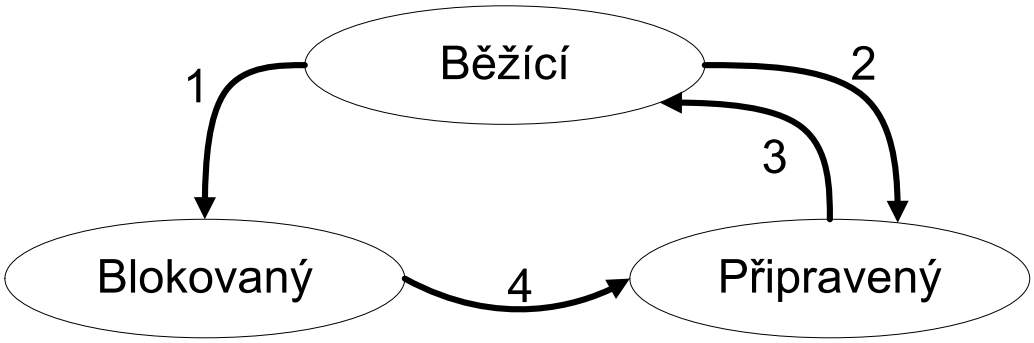
Proces je entita, která může být přidělena procesoru nebo v něm vykonána.[26] Na proces můžeme také nahlížet jako na entitu, která se skládá z několika elementů. Mezi dva základní elementy procesu patří programovací kód a sada dat, která souvisí s tímto kódem.[25] Proces můžeme jednoznačně charakterizovat:

1. Identifikátorem – Jednoznačný identifikátor související s konkrétním procesem, odlišuje ho od ostatních procesů.
2. Stavem – Stav procesu, jestli je ve stavu běžící nebo stojící.
3. Prioritou – Úroveň priority je vztažena k ostatním procesům.
4. Čítačem instrukcí – Adresa další instrukce v programu musí být vykonána.

5. Paměťovými ukazateli – Zahrnuje ukazatele programovacího kódu a dat, které souvisí s daným procesem
6. Kontextovými daty – Data, která jsou přítomna v registrech procesoru, zatímco je proces vykonáván.
7. I/O stavovou informací – Zahrnuje nesplněné I/O žádosti přiřazené procesu.
8. Evidencí informací – Zahrnuje množství procesorového času, hodinového času, časových lhůt.[25]

9.2. Stavy procesu

1. Běžící (aktuálně používaný procesorem)[26]
2. Připravený (kterýkoliv proces, který lze spustit)[28]
3. Blokovaný (proces čekající na událost, např. dokončení I/O operace)[28]



Obrázek č. 10: Stavy procesu [26]

Popis obrázku:

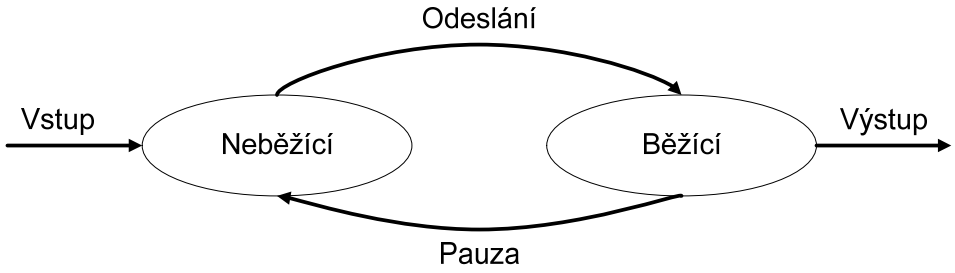
1. Proces blokován pro vstup
2. Plánovač vybírá další proces
3. Plánovač vybral tento proces
4. Vstup se stává dostupný[26]

9.2.1. Dispečer

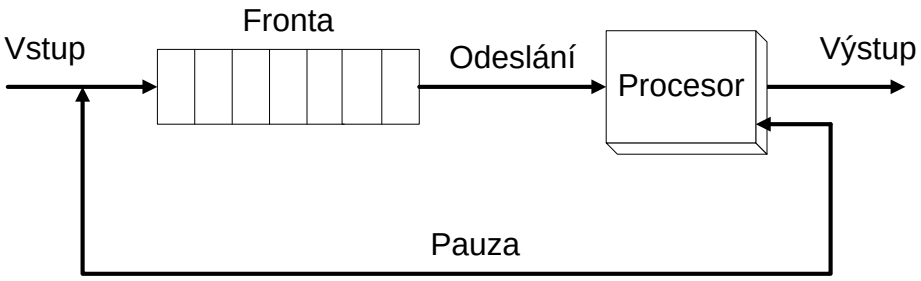
Chování procesu můžeme charakterizovat sledem instrukcí, které vykonává. Odkazujeme se na trasování procesu. Hlavní roli hraje malý program dispečer (plánovač).[25] Převádí procesor z jednoho procesu do druhého. Dispečer brání okupaci procesoru jediným procesem na nepřijatelně dlouhou dobu, na základě plánovacího algoritmu rozhoduje, který proces bude prováděn jako další. Během přepnutí z procesu A na proces B se v procesoru vždy provádí kód dispečera.[28]

9.2.2. Dvoustavový model procesu

V tomto modelu smí mít proces dva stavy (běžící a neběžící). Když operační systém vytvoří nový proces, vstoupí do systému v neběžícím stavu. Proces existuje a je znám operačním systémem, čeká, až dostane příležitost k vykonání. Občas bude současný běžící proces přerušen a dispečer vybere nějaký další proces a spustí ho. Minulý proces se dostane ze stavu běžícího do neběžícího a jeden z dalších procesů se dostane do stavu běžícího.[25]



Obrázek č. 11: Dvoustavový model procesu [25]



Obrázek č. 12: Diagram fronty [25]

Chování dispečera můžeme popsat pomocí diagramu na obrázku číslo 12. Proces, který je přerušen, je přenesen do fronty čekajících procesů. Jestli je proces ukončen nebo přerušen, tak je vyřazen existujícím systémem. Potom dispečer vybírá procesy z fronty k vykonání.[25]

Když vzniká nový proces, operační systém musí vytvořit datovou strukturu, která bude proces řídit. Musí alokovat adresní místo v hlavní paměti pro proces.[25]

Problémem dvoustavového modelu je, že neumožňuje rozlišit detaily. Dispečer nemůže jednoduše vybrat proces, který je ve frontě první na řadě, tj. čeká nejdéle.[28]

Proces vzniká:

1. spuštěním úlohy v dávce
2. přihlášením uživatele do systému
3. spuštění služby pro obsluhu požadavků (např. tiskový server, www server)
4. vytvořením potomka existujícího procesu (spawning) – každý proces může požadovat od OS vytvoření dalších procesů – potomků, tím se stává jejich rodičem[28]

Ukončení procesu:

1. uživatel se odhlásí
2. proces předá systému požadavek na ukončení
3. dojde k chybě
4. úloha v dávce vydá příkaz HALT[25][28]

9.3. Hierarchie procesů

Když v systému proces vytvoří další proces, tak rodičovský a dětský proces jsou k sobě stále v určitých způsobech přidruženy.[26] Vytvořený proces se označuje jako dětský. Tato operace nastává, když chtějí procesy spolu navzájem komunikovat a spolupracovat.[25] Dětský proces může sám vytvořit další procesy a tím formuje hierarchii procesu. Proces má pouze jednoho rodiče.[26]

9.4. Příčiny ukončení procesu

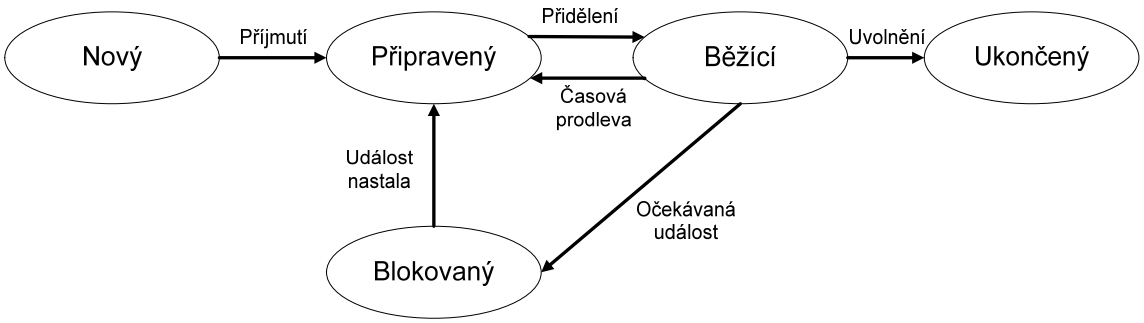
Každý počítačový systém musí poskytnout způsob pro ukončení procesu.[25]

1. Normální dokončení: Proces vykonává nějakou OS službu a ta je ukončena.
2. Vyčerpání časového limitu: Proces běží déle, než je určen maximální časový limit.

3. Nedostatek paměti: Proces požaduje více paměti, než systém může poskytnout.
4. Překročení mezí: Proces zkouší přistupovat do paměti, kam nemá povolen přístup.
5. Porušení ochrany: Zápis do souboru se zakázaným zápisem.
6. Chyba operace: Proces zkouší zakázanou operaci (dělení nulou, přetečení při sčítání).
7. Překročení doby čekání (Time Out): Doba čekání na událost překročila stanovené maximum.
8. Chyba I/O zařízení
9. Provedení nedovolené instrukce: Proces se pokouší provést neexistující instrukci (při pokusu o provedení dat jako programu).
10. Provedení privilegované instrukce: Proces se pokouší provést instrukci, kterou má rezervovanou operační systém.
11. Chybné použití dat: Část dat je špatného typu nebo nejsou inicializována.
12. Intervence operačního systému: Operační systém z nějakého důvodu ukončí proces (např. když detekuje deadlock).
13. Ukončení rodičovského procesu: Když dojde k ukončení rodiče, tak operační systém smí automaticky ukončit všechny potomky.
14. Požadavek rodičovského procesu: Rodičovský proces má právo ukončit svého potomka.[25][28]

9.5. Pětistavový model procesu

Jestliže jsou všechny procesy vždy připraveny k provedení, potom bude navržený dvoustavový model procesu efektivní. Tato implementace je nedostatečná, pokud některé procesy v neběžícím stavu jsou připraveny k provedení, zatímco ostatní jsou blokovány, čekají na dokončení I/O operace. Přirozenějším způsobem je rozdělit neběžící stav do dvou stavů (připravený a blokovaný).[25][28]



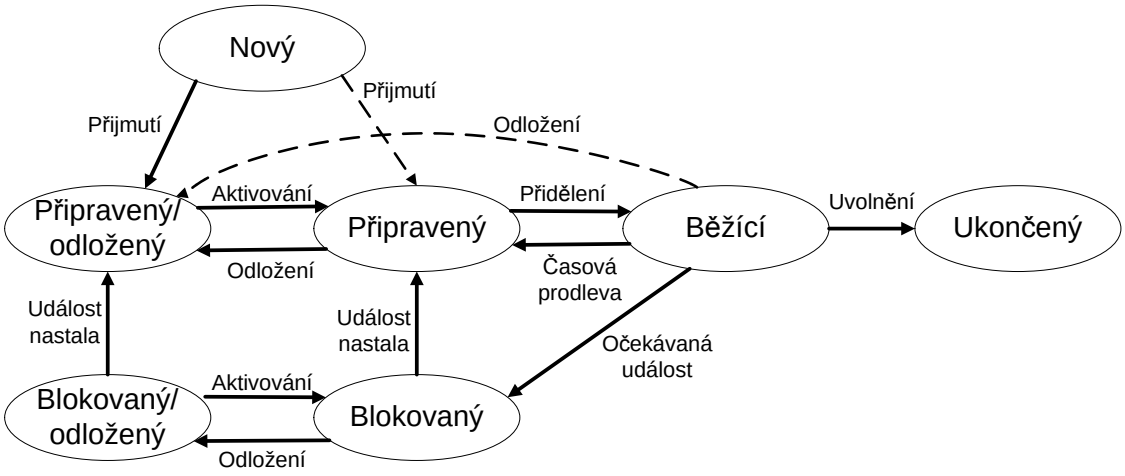
Obrázek č. 13: Pětistavový model procesu [25]

Pět stavů na obrázku 13:

1. Nový: Operační systém provedl všechny akce nezbytné pro vytvoření procesu. Proces nelze ještě zařadit mezi prováděné.[28]
2. Připravený: Proces, který je připraven k provedení, když k tomu dostane příležitost. Kterýkoliv proces, který lze spustit.
3. Běžící: Proces, který je prováděn.
4. Blokovaný: Proces, který čeká na událost. Na dokončení I/O operace.
5. Ukončený: Formálně je proces ukončen, proto ho nelze znovu spustit. Systém tento proces odstraní, jakmile to bude možné.[25][28]

9.6. Odložené procesy

Může nastat situace, kdy proces bude čekat na dokončení I/O operace. Důvodem je, že I/O operace trvá dlouho. Tyto procesy zabírají v paměti místo. Co dělat? Hlavní paměť bychom mohli rozšířit k uložení více procesů. Toto řešení je drahé. Dalším řešením je swapping (přesun) na disk. Proces je z hlavní paměti přesunut na disk a tím přejde do stavu odložený. Uvolní se nám operační paměť. Přibudou další dva stavy (blokovaný/odložený a připravený/odložený).[25][28]



Obrázek č. 14: Sedmistavový model procesu [25]

- Blokovaný/odložený: Proces je v přídavné paměti a čeká na událost.
- Připravený/odložený: Proces je v přídavné paměti, ale je schopen provedení, jakmile je načten do hlavní paměti.[25]

9.7. Implementace procesů

Operační systém musí udržovat tabulku procesů pro implementaci modelu procesu. Jeden záznam obsahuje jeden proces. Záznam obsahuje informaci o stavu procesu, čítači instrukcí, přidělené paměti a stavu.[26]

10. Procesy a vlákna

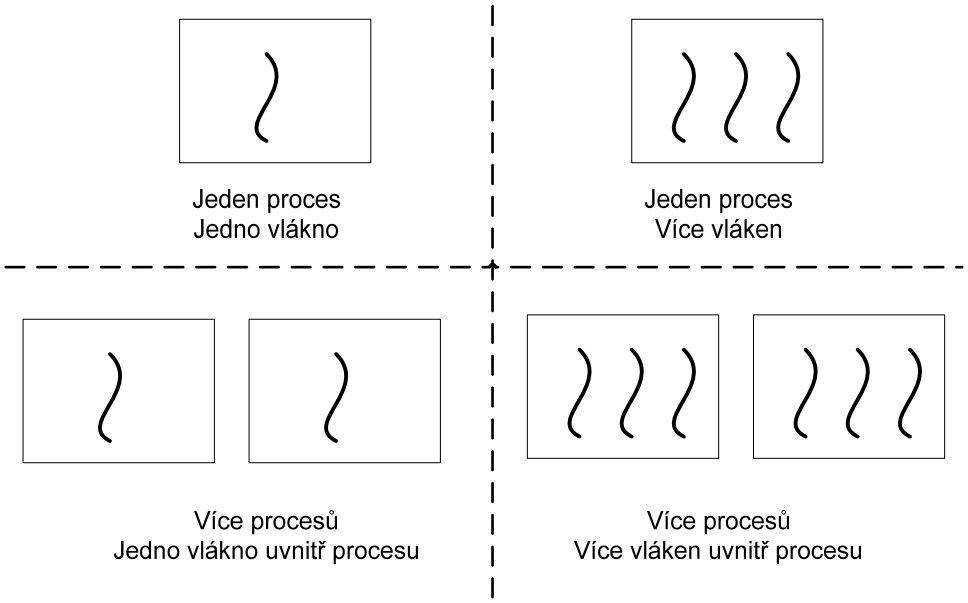
Proces ztělesňují dvě charakteristiky:

1. Vlastnictví prostředků: Proces obsahuje virtuální adresní prostor, ve kterém je uložen obraz procesu.[25]
2. Umožňuje přidělování: Proces můžeme charakterizovat jako posloupnost provádění jednoho nebo více programů. Vykonávání procesu může být přerušeno nebo proloženo vykonáváním jiných procesů.[28]

Každý operační systém může s těmito charakteristikami zacházet nezávisle. Díky rozlišení těchto dvou charakteristik můžeme definovat pojem vlákno. Vlákno je jednotka přidělování. Zatímco proces se vztahuje k vlastnictví prostředků.[25][28]

10.1. Vícevláknové procesy (Multithreading)

Schopnost operačního systému podporovat provádění více vláken uvnitř jednoho procesu. V operačním systému MS-DOS jsou implementovány pouze jednovláknové procesy. Oblíbený operační systém UNIX podporuje víceuživatelské procesy, ale procesy mohou být pouze jednovláknové. Windows, Solaris a Linux podporují vícevláknové procesy.[25][28]



Obrázek č. 15: Vlákna a procesy [25][28]

10.2. Rozdíl mezi procesem a vláknem

Proces je asociován virtuálním adresným prostorem, ve kterém je uložen obraz procesu. Dále má chráněný přístup k procesorům, dalším procesům, souborům a I/O prostředkům (zařízením). Uvnitř procesu smí být jedno nebo více vláken a každé je charakterizováno stavem provádění (běžící, připravený). Ukládají kontext, když neběží. Mají vlastní pracovní zásobník, mají vlastní statickou paměť pro lokální proměnné, mají přístup k paměti a prostředkům, které jsou sdíleny ostatními vlákny uvnitř procesu.[25][28]

10.3. Jednovláknový a vícevláknový model procesu

V jednovláknovém modelu není rozlišen pojem vlákno. Proces obsahuje řídící blok procesu a uživatelský adresní prostor. Ve vícevláknovém modelu je také s jedním procesem asociován řídící blok procesu a uživatelský adresní prostor, ale každému vláknu je přidělen zásobník.[25]

Všechny vlákna v procesu sdílejí stav a prostředky svého procesu. Pobývají ve stejném adresném prostoru a mají přístup ke stejným datům. Jestli jedno vlákno otevře soubor s právem pro čtení, tak další vlákna ve stejném procesu mohou také číst z tohoto souboru.[25]

10.4. Výhody použití vláken

1. Vytvoření nového vlákna zabere daleko méně času než vytvoření nového procesu. Výzkumy dokázaly, že vytvoření vlákna je desetkrát rychlejší než vytvoření procesu.
2. Ukončení vlákna zabere méně času než ukončení procesu.
3. Přepnutí mezi dvěmi vlákny uvnitř procesu zabere méně času než přepnutí mezi procesy.
4. Vlákna zlepšily efektivitu v komunikaci mezi programy. Protože vlákna uvnitř jednoho procesu sdílejí paměť a soubory, mohou komunikovat s každým dalším bez použití služeb jádra.[25][28]

10.5. Základní operace a stavy vláken

Jak u procesů, tak u vláken jsou stavy běžící, připravený a blokovaný. Mezi základní operace patří:

1. Vyvolání: Když je nový proces vyvolán, tak je vyvoláno i vlákno. Vlákno uvnitř procesu může vyvolat další vlákna.
2. Blokovaný: Když vlákno čeká na událost, bude blokováno.
3. Neblokovaný: Když událost, pro kterou je vlákno blokováno, nastala. Vlákno se přesune do připravené fronty.
4. Dokončený: Když je vlákno dokončeno.[25]

10.6. Implementace vláken na úrovni uživatele

Veškerá činnost vláken je řízena aplikacemi. Jádro si neuvědomuje existenci těchto vláken. Standardně aplikace začíná běžet v jednom vlákně. Aplikace a její vlákno jsou alokovány procesem, který je řízen jádrem. Kdykoliv, když běží aplikace (proces je v běžícím stavu), aplikace může vyvolat nové vlákno uvnitř stejného procesu.[25][28]

10.7. Implementace vláken na úrovni jádra

Veškerá činnost vláken je řízena jádrem.[25] Jádro má tabulku vláken, ve které sleduje všechny vlákna v systému. Když vlákno chce vytvořit nové vlákno nebo zničit existující vlákno, provede se volání jádra, které provede vytvoření nebo zničení vláken aktualizací tabulky vláken.[26] Příkladem tohoto přístupu je Windows.[25]

10.8. Kombinovaný přístup

Některé operační systémy poskytují kombinovaný přístup. Příkladem tohoto přístupu je Solaris. Vlákna se vytvářejí v uživatelském prostoru. Převážná část plánování a synchronizace probíhá uvnitř aplikace.[25][28]

11. Analýza činnosti procesoru

Analýza činnosti procesoru zahrnuje mnoho proměnných. Toto sledování si můžeme shrnout do několika kroků.[21]

- Typická pracovní zátěž vašeho sytému.
- Celkové využití procesoru (čas procesoru).
- Účinnost vašeho systému.
- Zkoumání procesů (celkový čas zabrání procesoru).
- Zkoumání podprocesů (vláken), jejich stavy, čas přidělený procesorem, zjištění jejich priority.[21]

11.1. Typické stavy podprocesů

Tabulka č. 3: Stavy podprocesů [21]

Stav podprocesu	Popis činnosti
0	Inicializován
1	Připraven (podproces chce použít procesor, ale musí na něj čekat, neboť právě není žádný volný)
2	Spuštěn (podproces používá procesor)
3	Úsporný režim (podproces se chystá použít procesor)
4	Ukončen
5	Čeká (podproces nemůže využít procesor, protože čeká na dokončení operace periferie nebo na uvolnění nějakého prostředku)
6	Přechod (podproces není připraven na spuštění)
7	Neznámý (podproces je neznámém stavu)

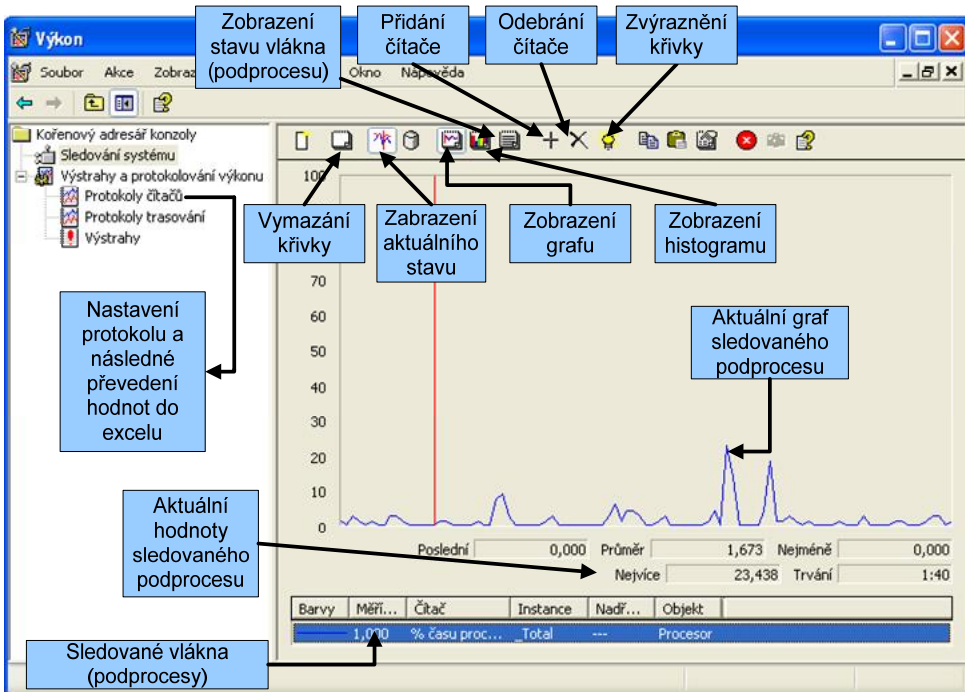
11.2. Čítače procesoru

Objekty systému, procesor, proces a podproces (vlákno) obsahují čítače, které poskytují informace o využití procesoru.[21] V nástroji Performance monitor je vlákno pod názvem podproces. Autor se ve své práci zaměří na čítače podprocesu. Bude sledovat stavy podprocesu, využití procesoru a využití paměti.

12. Nástroje na sledování výkonu

Dále pracuje se správcem úloh a s nástrojem na sledování výkonu Performance monitor. Nástroj Performance monitor je přímo implementovaný v Microsoft Windows. Spuštění sledování systému můžeme pomocí hlavní nabídky START, klepneme na položku Spustit a napíšeme perfmon a potvrdíme. Nástroj můžeme také spustit pomocí příkazového řádku ve Windows, kde napíšeme perfmon.

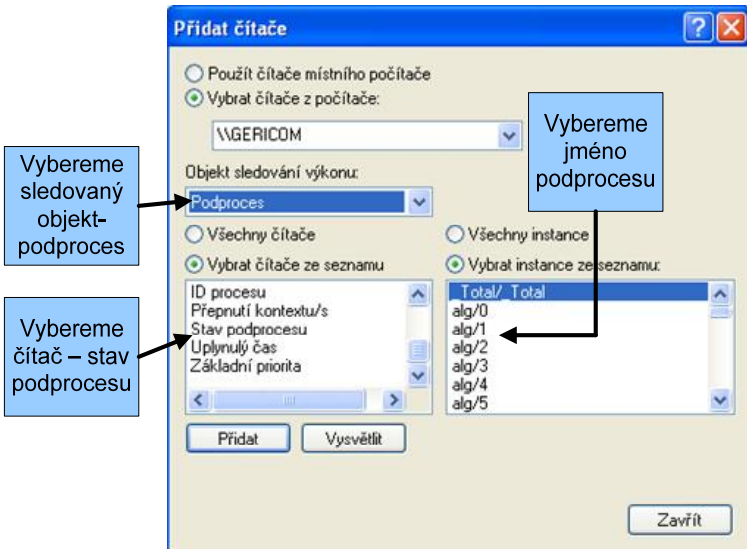
Jakmile spustíme konzolu výkonu, objeví se nám prázdný graf Sledování systému. Vlevo ve stromu Sledování systému se objeví Výstrahy a protokolování výkonu, které můžeme rozbalit a objeví se nám další položky (viz obrázek č. 16).



Obrázek č. 16: Konzola výkonu (Performance Monitor) [zdroj: vlastní]

Na obrázku č. 16 vidíme podrobně popsanou konzolu výkonu Performance monitoru.

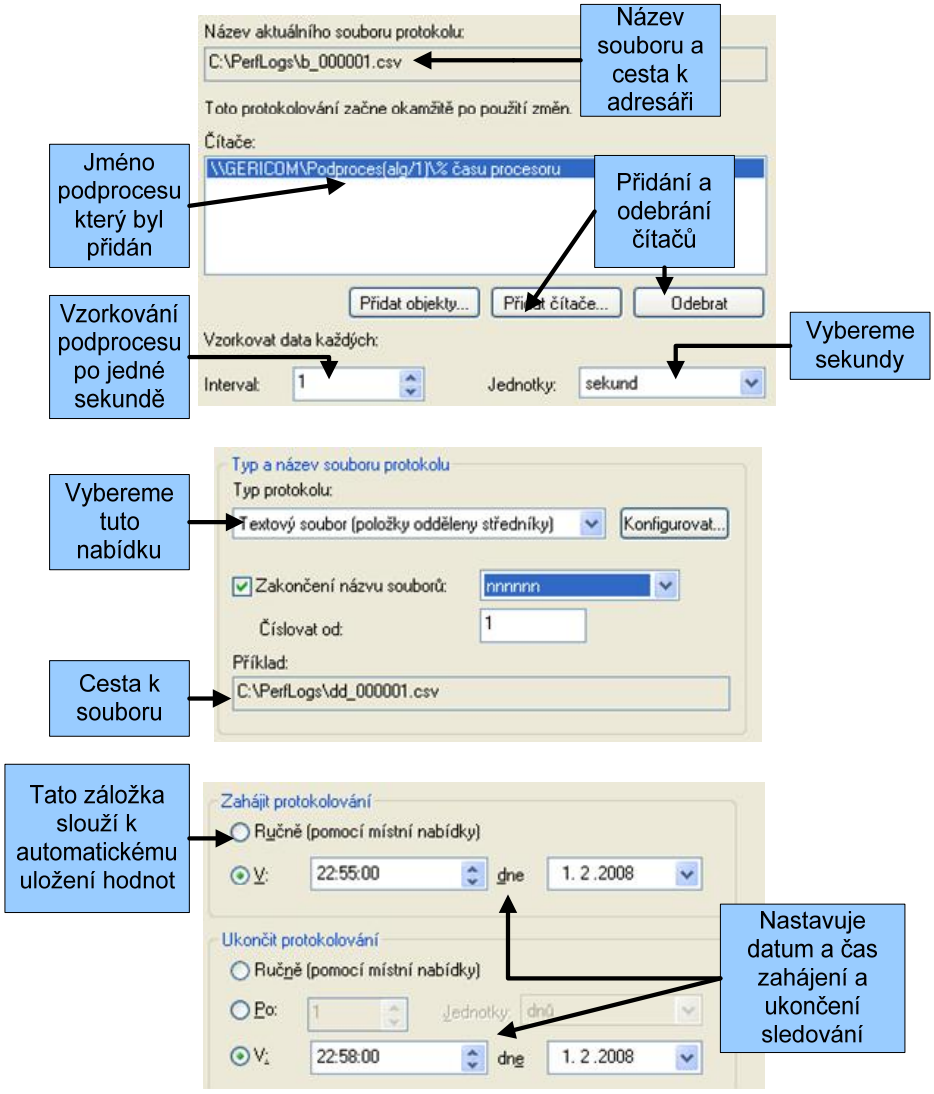
Pomocí tlačítka Přidání čítače z předcházejícího obrázku je možné přidat čítač podprocesu, který bude sledován. Na obrázku č. 17 můžeme vidět, co nám toto okno nabízí.



Obrázek č. 17: Přidání čítače [zdroj: vlastní]

Položka protokoly čítačů umožňuje vytvořit nový protokol pro sledování procesů a vláken, naplánovat si čas, po který bude podproces sledován, a potom následné uložení do souboru s příponou csv, který můžeme exportovat do excelu. Pro vytvoření nového protokolu,

klikneme pravým tlačítkem myši do pravého okna a z nabídky vybereme nové nastavení protokolu, zadáme jméno protokolu a objeví se nám okno se třemi kartami: obecné, soubory protokolů a plán. Nastavení a práci s jednotlivými kartami vidíme na obrázku č. 18. V kartě obecné nastavujeme podprocesy, které chceme sledovat a jejich vzorkování. Karta soubory protokolů konfiguruje název a typ protokolu. V poslední nabídce plánujeme čas sledování jednotlivých podprocesů. V této poslední nabídce efektivně nastavíme čas začátku a konce plánování. Na pevném disku se automaticky vytvoří adresář PerfLogs, do kterého jsou ukládány soubory s příponou csv.

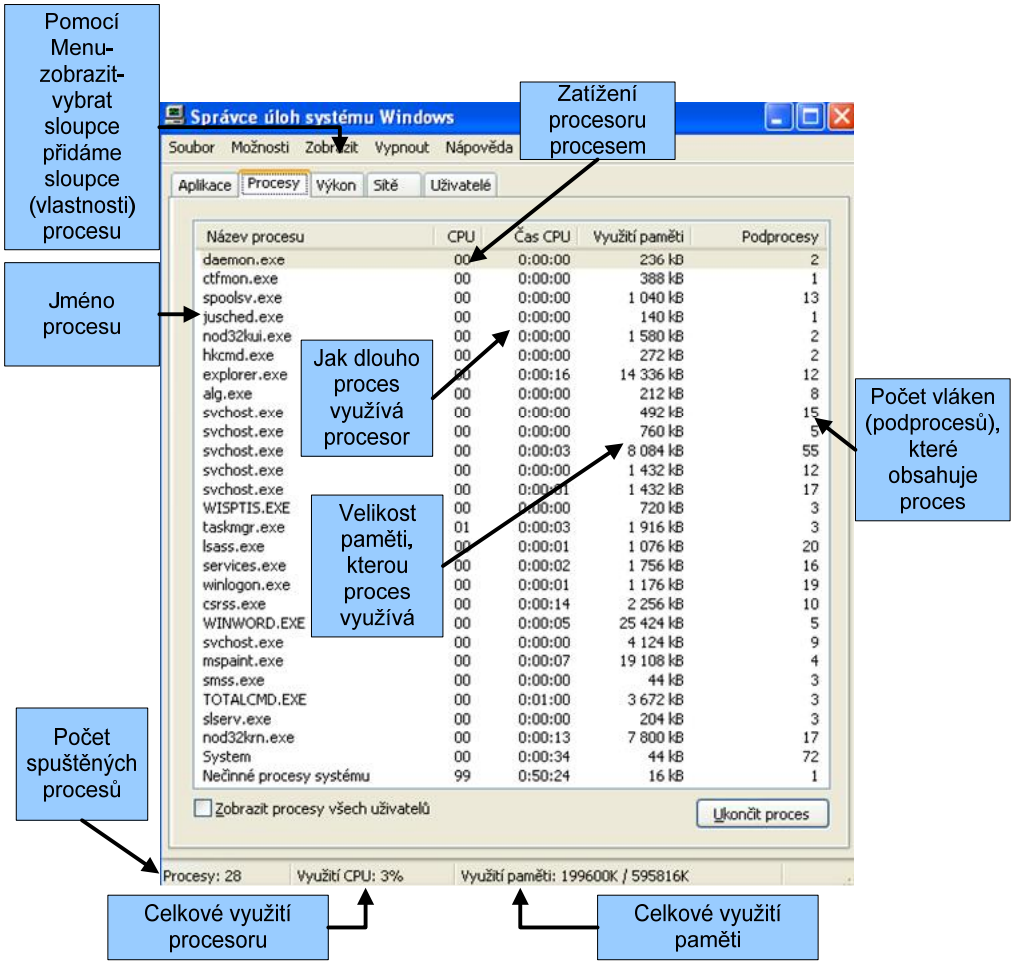


Obrázek č. 18: Protokol čítače [zdroj: vlastní]

12.1. Správce úloh (Task manager)

Správce úloh systému je přímo implementován ve Windows. Spustíme ho pomocí kombinace kláves Ctlr+Alt+Delete. Po tomto stisknutí se nám objeví okno s nabídkou. Správce má celkem pět záložek (aplikace, procesy, výkon, sítě, uživatelé). V aplikacích vidíme aktuálně spuštěné aplikace, v procesech spuštěné procesy. V záložce výkon sledujeme využití procesoru a paměti a v uživatelích je přihlášený uživatel. Na kartě sítě sleduje stav sítě.

Autor se ve své práci zaměří na záložku procesy. Na kartě procesy je možno sledovat všechny vlastnosti procesů. Můžeme zde přidávat a odebírat sloupce, pomocí nabídky menu-zobrazit-vybrat sloupce. Na obrázku č. 19 vidíme, jak proces zatěžuje procesor a paměť, jak dlouho využívá procesor a kolik obsahuje vláken. Proces ukončíme, když na něj klikneme pravým tlačítkem myši a dáme ukončit.



Obrázek č. 19: Správce úloh [zdroj: vlastní]

13. Praktický příklad sledování vláken a vytížení CPU při spuštěném programu

13.1. Program Ad-Aware SE Personal

Autor začal s testováním široké škály programů. Zkoušel od komprimovacích programů přes antivirové až po systémové programy. Mezi nejčastější stavy vláken patřily čekající a připravený. Jako nejlepší program se jevil Ad-Aware SE Personal, freeware nástroj na odstranění škodlivého spyware z počítače. Po spuštění tohoto nástroje, si spustíme správce úloh a v záložce procesy vidíme proces Ad-Aware.exe. Přidáme si sloupce podprocesy, využití paměti a čas procesoru pomocí menu-zobrazit-vybrat sloupce. Ze správce vyčteme, že proces při nabíhání aplikace zabírá od 50 do 98 % procesor, využití paměti se postupně zvyšuje od 10 MB do 60 MB, aplikace využila procesor 10 sekund a proces obsahuje dva podprocesy. Když je aplikace spuštěná a nic s ní neděláme, tak je využití procesoru téměř nulové a využití paměti klesá. Pokud aplikaci necháme testovat náš počítač, využití procesoru rapidně stoupne až do 90 %, v některém okamžiku až do maximálního využití. Využití paměti začne mírně stoupat. Ve sloupci Čas CPU se nám počítá doba, jak dlouho proces využívá procesor (čas, po který aplikace testuje náš počítač).

Dále přistoupíme k testování vláken pomocí Performance monitor. Nástroj spustíme pomocí nabídky start-spustit a napíšeme příkaz perfmon. Pro lepší orientaci nejdříve vymažeme aktuálně spuštěné čítače. Označíme čítače (vpravo dole) tlačítkem myši a zmačkneme vymazat čítače (černý křížek v panelu nástrojů). Pracovní plocha je popsaná na obrázku č. 16. Dalším krokem bude přidání čítačů, které budou obsahovat podprocesy. Z nabídky vybereme přidat čítače (popsáno na obrázku č. 17). V záložce Objekt sledování výkonu vybereme podproces, záložku Vybrat čítače ze seznamu ponecháme a ve Vybrat instance ze seznamu vybereme podprocesy sledovaného procesu (Ad-Aware/0, Ad-Aware/1). Potom přistoupíme k vytvoření protokolu čítače. Naplánujeme si dobu, po kterou budeme chování podprocesů sledovat. Klikneme na Výstrahy a protokolování výkonu – protokoly čítačů. Pravým tlačítkem myši klikneme na protokoly čítačů a dáme Nové nastavení protokolu, napíšeme libovolné jméno protokolu. Po tomto kroku se nám objeví nově vytvořený protokol vpravo na obrazovce. Bude zatím označený červeným symbolem (protokol je v nečinném stavu). Když se protokol spustí, tak se jeho označení změní na zelenou barvu. Další postup bude spočívat v nastavení podprocesů a naplánování doby sledování. Klikneme pravým tlačítkem myši na nově vytvořený protokol a

dáme vlastnosti. Objeví se nám nabídka nastavení protokolu, tak jak ji vidíme na obrázku č. 18. Nastavení si můžeme rozčlenit do třech kroků.

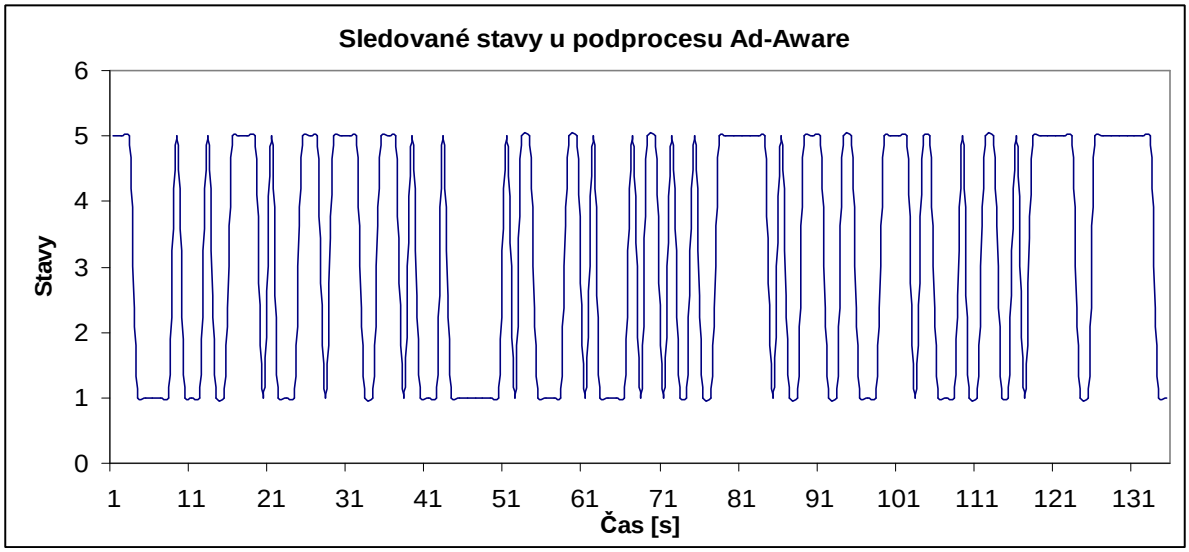
1. V první záložce Obecné přidáme sledované čítače tlačítkem Přidat čítače. V záložce Objekt sledování výkonu vybereme podproces. V levé části okna vybereme vlastnost podprocesu, kterou budeme chtít sledovat. Vybereme stav podprocesu. V pravé části okna označíme název podprocesu, který chceme vzorkovat. V našem případě Ad-Aware/0 a Ad-Aware/1. V políčku Interval nastavíme 1 a v políčku Jednotky sekundy. Toto nastavení nám umožňuje sledovat stav podprocesu v každé sekundě.
2. V záložce Soubory protokolů nastavujeme typ souboru, do kterého se budou hodnoty podprocesů ukládat. Typ protokolu označíme Textový soubor (položky odděleny středníky). V tomto okně ještě vidíme jméno adresáře a cestu, kam se bude soubor ukládat na disk.
3. Poslední záložka Plán nám poskytuje naplánování času sledování podprocesu. Umožňuje nastavit vzorkovací čas podprocesu, začátek, konec sledování a den. Zadáme čas na pět minut (např. od 11:00 do 11:05). Den zahájení a ukončení protokolování zadáme stejný.

Těmito kroky jsme nastavili celé sledování stavů podprocesů. Teď nám zbývá akorát vyčkat do začátku protokolování a naši aplikaci v tomto čase spustit, nechat testovat počítač. Až naše protokolování skončí, můžeme aplikaci zavřít.

Naše poslední akce spočívá v prohlédnutí naměřených výsledků. Otevřeme si Microsoft excel a naimportujeme náš soubor (menu – data – importovat externí soubor – importovat data). V první nabídce necháme zaškrtnutý oddělovač, v druhé zaškrtneme čárku a v poslední necháme formát Obecný.

V následující tabulce excelu vidíme stavy podprocesů v jednotlivých záznamech. V prvním sloupci je čas sledování po sekundách. V druhém a třetím sloupci jsou podprocesy a jejich stavy v každé sekundě. Nejčastějším stavem je čekající (5) a připravený (1). U prvního podprocesu se tyto dva stavy střídají a druhý zůstává pouze ve stavu čekajícím.

Na obrázku č. 20 vidíme celý vzorkovací průběh stavů sledovaného podprocesu Ad-Aware. Podproces je nejdříve ve stavu čekajícím, po několika sekundách přejde do stavu připraveného.



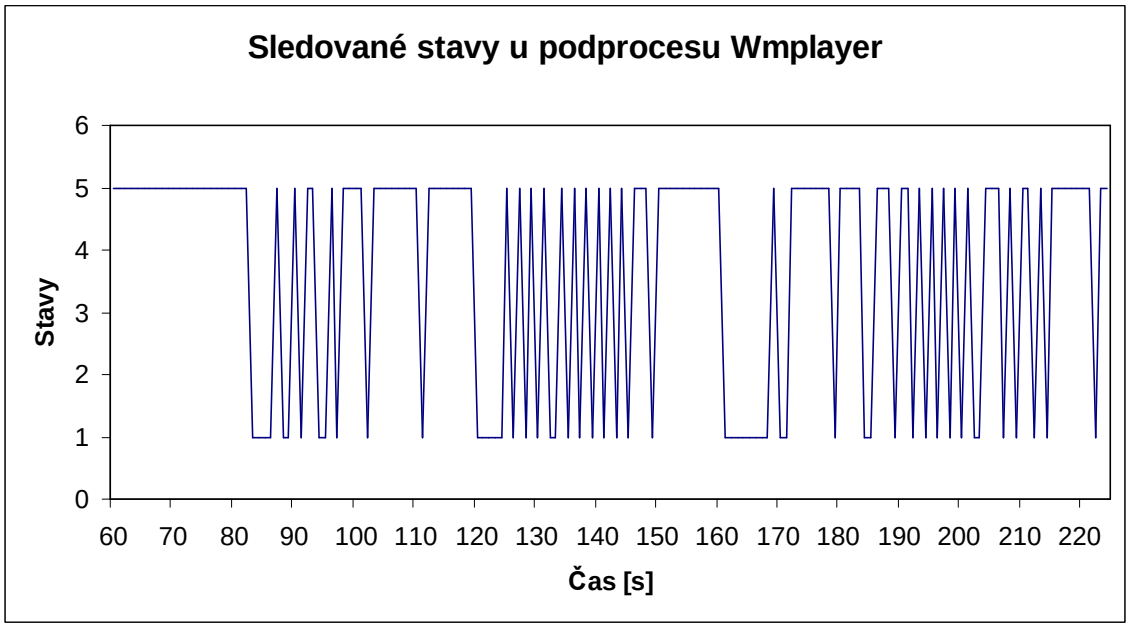
Obrázek č. 20: Graf stavů podprocesu Ad- Avare [zdroj: vlastní]

13.2. Program Windows Media Player

Dalším programem, který si autor vybral na testování, je Windows Media Player na přehrávání videa a hudby. Tato aplikace je standardním vybavením operačního systému Windows. Při testování budeme postupovat stejně jako v předcházejícím příkladu.

1. Spustíme si program Windows Media Player pomocí nabídky start ve Windows.
2. Spustíme si správce úloh (přidáme si potřebné sloupečky: podprocesy, využití paměti a čas procesoru pomocí menu – zobrazit – vybrat sloupce).
3. Ve správci úloh vidíme tuto aplikaci jako proces wmplayer.exe, který obsahuje 16 podprocesů (počet se může měnit v závislosti na využití programu)
4. Budeme sledovat využití paměti a procesoru.
5. Ve sloupečku využití paměti vidíme číslo od 15 do 17 MB (závisí to, na tom, co zrovna s programem děláme).
6. Program při úvodním spuštění skoro procesor nevytěžuje (od 1 do 5 %). Když začneme v programu přehrávat film, tak se vytížení procesoru začne zvyšovat.
7. Dále přistoupíme k Performance monitor (spustíme ho pomocí příkazu perfmon v nabídce start – spustit).

8. Smažeme všechny aktuálně spuštěné objekty (vidíme je vpravo dole). Označíme je a zmačkneme tlačítko Delete nebo pomocí tlačítka Odstranit (takového křížku) v nabídkové liště.
9. Přidáme si aktuálně spuštěné podprocesy procesu wmplayer. Buď pomocí klávesové zkratky CTRL+I nebo pomocí tlačítka Přidat (takové plus) v nabídkové liště.
10. V záložce Objekt sledování výkonu vybereme podproces. V okně Vybrat čítače ze seznamu vybereme Stav podprocesu a v okně Vybrat instance ze seznamu vybereme podprocesy wmplayer/0 až wmplayer/16. Potvrdíme tlačítkem Přidat a dáme Zavřít.
11. Nyní přistoupíme k naplánování času sledování podprocesů. V levém okně dáme Výstrahy a protokolování výkonu a protokoly čítačů. V menu klineme na akce – nové nastavení protokolu, zadáme jméno protokolu a potvrdíme. Objeví se nám nabídkové okno (popsané na obrázku číslo 18). Budeme postupovat podle tohoto obrázku.
12. V záložce Obecné přidáme podprocesy wmplayer pomocí tlačítka Přidat čítače. Dále nastavíme interval testování na 1 sekundu.
13. V záložce Soubory protokolů nastavíme typ protokolu na textový soubor (položky odděleny středníky). V záložce Plán nastavíme po jaký čas budeme testovat podprocesy. Doporučoval bych nechat testovat po dobu pěti minut.
14. V programu spustíme nějaký film a necháme testovat. Budeme sledovat ve správci úloh, jak se mění využití paměti a procesoru po dobu testování. Můžeme vidět, že využití procesoru se mění od 20 do 50 %. Využití paměti se mění od 9 MB do 12 MB.
15. Po skončení testování si spustíme excel a naimportujeme do něj získaná data z testování.
16. V excelu vidíme, že nejčastějším stavem podprocesů je 1 (připravený) nebo 5 (čekající).
17. Nakonec si uděláme graf z podprocesu wmplayer/16. V grafu vidíme, jak podproces přechází z jednotlivých stavů.



Obrázek č. 21: Graf stavů podprocesu Wmplayer [zdroj: vlastní]

Závěr

V této části svou práci zhodnotím jako celek a vyzvednu její význam a přínos pro společnost. Téma *Vytvoření podpůrných nástrojů pro výuku předmětu Operační systémy* jsem si zvolil záměrně, neboť se zabývám nastavením, instalací a konfigurací operačního systému skoro každý den.

Jsem přesvědčen o tom, že má bakalářská práce přispěje k lepšímu pochopení a porozumění principům a funkcím operačního systému.

Chci zdůraznit, že všechny cíle mé bakalářské práce byly naplněny. Jsou zde popsány základní charakteristiky počítačových systémů, stručná historie operačních systémů, jednotlivé typy a bezpečnost operačních systémů, která je podle mého uvážení jednou z nejdůležitějších pasáží v dnešním světě informačních technologií a každý uživatel by ji měl věnovat nejvíce času. Ve své práci dále pojednávám o pojmu proces a vlákno a jejich praktické aplikaci na konkrétní spuštěné programy. Toto testování je prováděno pomocí softwarových nástrojů Správce úloh v systému Windows a Performance monitoru, které jsou přímo implementovány v systému MS Windows.

Proč je to právě sledování procesů a vláken ve Windows, které je tématem druhé části mé práce? Odpověď je jednoduchá, již v samotném zadání, vedoucí mé bakalářské práce apeloval na nepochopení rozdílu mezi pojmem proces a vlákno u většiny studentů. Dal jsem si za úkol vnést do tohoto problému jasno. Má práce osvojí tyto pojmy a popíše práci pomocí výše zmíněných nástrojů.

Považuji procesy a vlákna za velmi důležité a klíčové pojmy v oblasti operačních systémů. Je nutné pochopit rozdíl mezi nimi a jejich podstatu. Ve své práci jsem se k výše zmíněnému snažil přispět a má práce by bezpochyby mohla sloužit jako studijní pomůcka ke cvičením z operačních systémů. Dále je užitečné používání nástrojů Správce úloh systému Windows a Performance Monitoru na sledování výkonu počítače.

Za největší nevýhodu nástroje Performance monitoru považuji neschopnost nastavení menší doby vzorkování vláken než je jedna sekunda. Důsledkem této nevýhody je, že u testovaných aplikací jsem zachytil pouze stavy vláken čekající (1) a připravený (5). Ostatní stavy proběhly v době menší než jedna sekunda, proto je nástroj nezachytil.

Výsledkem této práce bude studijní opora ke cvičením z předmětu Operační systémy. Každý student by měl být schopen, po přečtení této studijní opory, aplikovat tyto softwarové

nástroje na kteroukoli aplikaci, nejenom na výše uvedené v textu.

Použité zdroje

1. *Bezpečné heslo* [online]. 2008, 20. 3. 2008 [citováno 2008-04-06]. Dostupný z WWW: <http://www.earchiv.cz/a94/a421c120.php3>.
2. *Bezpečnost softwarových systémů* [online]. 2004 [citováno 2008-04-18]. Dostupný z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=32250>.
3. BIC L., SHAW A.C. Operating Systems Principles. Prentice-Hall. 1st edition. 2003. 543 s. ISBN 0-13-026611-6.
4. ČADA O. *Operační systémy*. Grada Publishing. Praha. 1994. 377 s. ISBN 80-85623-44-7.
5. *Denial of Service* [online]. 2008 [citováno 2008-04-12]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Denial_of_Service>.
6. EL-REWINI, H., ABD-EL-BARR, M. *Advanced computer architecture and parallel processing.* New Jersey: John Wiley & Sons, 2005. 272 s. ISBN 0-471-46740-5.
7. HÄRING, D. *Jak zvolit bezpečné heslo?* [online]. 2002 [citováno 2008-04-06]. Dostupný z WWW:<http://www.linuxzone.cz/index.phtml?ids=1&idc=398>.
8. *Harwardská architektura* [online]. 2007 [citováno 2007-11-8]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Harvardsk%C3%A1_architektura>.
9. HUDEC, T. *Operační systémy*. (přednáška**)** Pardubice: Univerzita Pardubice, 2005.
10. JANDOŠ, J. *Technické prostředky informačních systémů II*. Praha: Vysoká škola ekonomická, 1995. 85 s. ISBN 80-7079-821-1.
11. KOLÁŘ, P., *Operační systémy* (přednáška) Liberec: Technická univerzita Liberec, 2005.
12. *Mainframe* [online]. 2007 [citováno 2007-11-9]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Mainframe>.
13. *Novinky v zabezpečení Windows XP Service Pack 2* [online]. 2004 [citováno 2008-04-12]. Dostupný z WWW: <http://www.zive.cz/default.aspx?article=118933>.
14. PELIKÁN, J. *Technické vybavení počítačů.* Masarykova univerzita. Brno. 2001. 207 s.
15. PETERKA, J. *Trap door* [online]. 1994 [citováno 2008-04-11]. Dostupný z WWW: <http://www.earchiv.cz/a94/a421c120.php3>.

16. PETRLÍK, L. *Základy operačních systémů*. (přednáška) Plzeň: Západočeská univerzita, 2007.

17. *Počítačový červ* [online]. 2008 [citováno 2008-04-12]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%BD_%C4%8Derv>.

18. *Počítačový virus* [online]. 2008 [citováno 2008-04-12]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%BD_virus>.

19. *Překvapení od Symantec: Windows jsou nejbezpečnější* [online]. 2007 [citováno 2008-04-12]. Dostupný z WWW: <http://www.root.cz/zpravicky/prekvapeni- od-symantec-windows-jsou-nejbezpecnejsi/>.

20. PŘIBYL, T. *Logické bomby v počítačích* [online]. 2005 [citováno 2008-04-11]. Dostupný z WWW: <http://www.automatizace.cz/article.php?a=481>.

21. ROUBÍČEK, L. *Microsoft Windows 2000 Server*. Computer Press. 1. vydání. Praha, 2000. 577 s. ISBN 80-7226-291-2.

22. ŠAFAŘÍK, J. *Operační systémy*. (přednáška) Plzeň: Západočeská univerzita, 2006.

23. ŠESTÁK, Z. *Generace počítačů* [online]. Střední odborná škola. Třinec. 2007. [citováno 2007-11-9]. Dostupný z WWW <www.sostrinec.cz/sestak/files/uvodHW.pps>.

24. SILBERSCHATZ A., GALVIN P.,B., GAGNE G. *Operating systems concepts.* John Wiley & Sons. 6th edition. Hoboken, 2003. 951 s. ISBN 0-471-25060-0.

25. STALLINGS, W. *Operational Systems.* Prentice Hall. Upper Saddle River. 5th edition. New Jersey. 2005. 818 s. ISBN 0-13-127837-1.

26. TANENBAUM A. S. *Modern operating Systems*. Prentice Hall. Upper Saddle River. 2nd edition. 2001. 976 s. ISBN 0-13-031358-0.

27. *Trojský kůň (program)* [online]. 2008 [citováno 2008-04-11]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Trojsk%C3%BD_k%C5%AF%C5%88_(program)>.

28. VANĚK, A. *Operační systémy*. (přednáška**)** Pardubice: Univerzita Pardubice, 2003.

29. VANĚK, A. *Úvod do počítačů*. (přednáška**)** Pardubice: Univerzita Pardubice, 2003.

30. VAŘEČKOVÁ Š. *Operační systémy* [online]. Ústav informatiky (Slezská univerzita). Opava. 2006. [citováno 2007-10-12]. Dostupný z WWW <http://axpsu.fpf.slu.cz/ vav10ui/opsys.html>.

31. *Windows XP Baseline Security Checklists* [online]. 2003 [citováno 2008-04-12]. Dostupný z WWW: <http://www.microsoft.com/technet/archive/security/ chklist/xpcl.mspx?mfr= true>.

32. *Windows XP Professional: Stručné informace* [online]. 2008 [citováno 2008-04-18]. Dostupný z WWW: <http://www.microsoft.com/cze/windows/xp/pro/ evaluation/ features.mspx>.

33. *Zadní vrátka* [online]. 2008 [citováno 2008-04-11]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Zadn%C3%AD_vr%C3%A1tka>.

Seznam obrázků

Seznam tabulek

Použité zkratky

Zkratky	České vysvětlení	Anglické vysvětlení
ACE	Seznam řízení přístupu	Access Control Entry
ACL	Seznam přístupových práv	Access Control List
ALU	Aritmeticko-logická jednotka	Arithmetic And Logic Unit
CP/M	Operační systém, který se používal v době osmibitových počítačů	Control Program for Microcomputers
CPU	Centrální procesorová jednotka	Central Processing Unit
CTSS	Systém se sdílením času	Compatible Time Sharing System
DOS	Diskový operační systém	Disk Operating System
FAT	Jednoduchý souborový systém	File Allocation Table
GUI	Grafické uživatelské rozhraní	Grafical User Interface
IBM	Výrobce elektronických číslicových počítačů	International Business Machines
ICS	Sdílení připojení k internetu	Internet Connection Sharing
I/O	Vstupní/výstupní	Input/Output
IP	Datový protokol	Internet Protocol
LAN	Lokální síť	Local Area Network
MacOS	Operační systém pro počítače Macintosh firmy Apple	Macintosh Operating System
MIMD	Kombinace procesorů, které souběžně zpracovávají odlišnými posloupnostmi instrukcí odlišné množiny dat	Multiple Instruction Multiple Data

MISD	Jedna množina dat je předána více procesorům, z nichž každý provádí jinou posloupnost instrukcí	Multiple Instruction Single Data
MS	-	Microsoft
MS-DOS	Jednoúlohový a jednouživatelský operační systém	Microsoft Disk Operating System
MULTICS	Operační systém, historický předchůdce Unixu	Multiplexed Information And Computing Service
NTFS	Moderní souborový systém	New Technology File System
OS	Operační systém	Operating System
PalmOS	Operační systém pro PDA a komunikátory	Palm Operating system
PC	Osobní počítač	Personal Computer
PDA	Osobní digitální pomocník	Personal Digital Assistant
POSIX	Přenositelné rozhraní pro operační systémy	Portable Operating System Interface
QNX	Komerční sytém reálného času	
RT-Linux	Operační systém reálného času	Real Time- Linux
ROM	Paměti, které jsou určené pouze pro čtení informací	Read Only Memory
SID	Bezpečnostní identifikátor vlastníka objektu	Security Identifier
SIMD	Jedním proudem instrukcí se ve více procesorech zpracovává více různých množin dat	Single Instruction Multiple Data

SISD	Jeden procesor zpracovává jednu množinu dat jedním proudem instrukcí	Single Instruction Single Data
SMP	Symetrický multiprocesing	Symmetric Multiprocessing
UNIX	Víceúlohový a víceuživatelský operační systém	Unary Information And Computing Service

ÚDAJE PRO KNIHOVNICKOU DATABÁZI

Název práce	VYTVOŘENÍ PODPŮRNÝCH NÁSTROJŮ PRO VÝUKU PŘEDMĚTU OPERAČNÍ SYSTÉMY
Autor práce	Michal Bělský
Obor	Regionální a informační management
Rok obhajoby	2008
Vedoucí práce	Ing. Pavel Jirava, Ph.D.
Anotace	V mé bakalářské práci se zabývám oblastí operačních systémů. Operační systém v počítačovém odvětví prošel bohatou historií od příkazové řádky až po plně grafický vzhled. V první části popisuji architektury počítačových systémů a jakou úlohu hraje operační systém v oblasti počítačů. Práce popisuje historii a jednotlivé typy operačních systémů. Dále se snaží seznámit s bezpečností operačních systémů, která je v dnešní době nejvíce diskutovanou oblastí. Ve druhé části jsou vysvětleny pojmy proces a vlákna, jaký je mezi nimi rozdíl a jakou důležitou roli hrají v oblasti operačních systémů. Jsou zde probírány praktické příklady, jak se jednotlivé vlákna v procesu chovají při spuštěné aplikaci. Tyto příklady budou používány ve cvičeních z operačních systémů.
Klíčová slova	Operační systém, počítačové architektury, historie operačních systémů, typy operačních systémů, bezpečnost operačních systémů, procesy, vlákna.