

Operační systémy

5. cvičení

Jiří Zacpal

KMI/ZVT – Základy výpočetní techniky

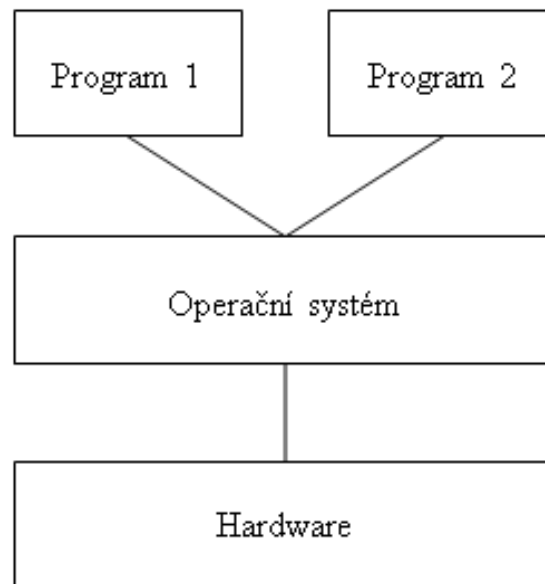
Operační systém

Co to je operační systém

- základní softwarové vybavení počítače - rozhraní mezi uživatelem a hardware počítače
- umožňuje programům (aplikacím) běh na/v počítači pomocí programového rozhraní (API) a uživatelům práci s počítačem pomocí svého uživatelského rozhraní (UI) a programů
- cíl: snadné a efektivní využití počítače (pro uživatele i aplikace)
 - víceméně protichůdné požadavky - dříve důraz na efektivitu (a vůbec možnost), nyní spíše snadnost
 - kompromis, závisí na způsobu využití a typu počítače
-> různé OS
- poskytuje abstrakci (funkcí) hardware počítače, odstiňuje uživatele a aplikace od hardware

Operační systém

- **Virtuální počítač** – OS prostředník mezi hardwarem a aplikačními programy
- **Správa prostředků** – OS přiděluje a odebírá systémové prostředky procesům



Části OS

- **jádro (kernel)** - vlastní OS,
- **základní obslužné programy** - pro práci s OS a zdroji počítače, např. administrátorské a diagnostické nástroje, diskové utility, programy pro práci se zařízeními, sítí apod.
- **uživatelské rozhraní (UI)** - součást OS (jádra?) nebo programy? -> záleží na typu OS, neinteraktivní (dávkové), interaktivní - textová konzole s interpretem příkazů (shell) nebo grafické s okenním systémem

Jádro

- Jádro je základním kamenem operačního systému. Zavádí se do operační paměti počítače při startu a zůstává v činnosti po celou dobu běhu operačního systému.
- Druhy:
 - **Monolitické jádro** – jádro je jedním funkčním celkem. Tento typ jádra je použit v operačním systému UNIX.
 - **Mikrojádro** – jádro je velmi malé a všechny oddělitelné části pracují samostatně jako běžné procesy.
 - **Hybridní jádro** – kombinuje vlastnosti monolitického jádra i mikrojádra. Tento typ jádra je použit u operačních systémů Windows.

Typy OS

- **univerzální** - pro desktopové a přenosné počítače typu PC, servery, mainframe aj.
- **embedded** - specializované pro embedded zařízení, dnes i upravené univerzální (např. Linux, MS Windows)
- **reálného času** - zaručení vyřízení požadavku/odpovědi v pevně daném čase, např. VxWorks, QNX, upravené univerzální (např. RTLinux, MS WindowsCE) i HW řešení, např. pro řízení strojů
- **distribuované** - pro běh současné na více počítačích, simulace např. jedné společné paměti, pro počítačové klastry (cluster)
- dnes nejvíce používané: na desktopových PC MS Windows (majoritní), Mac OS X, GNU/Linux, Unix, na (síťových) serverech unixové, na embedded zařízeních různé (MS Windows, Linux)

Linux

- Linux je označení pro jádro operačního systému. Na počítači se setkáte s tzv. distribucí.
- Nejznámější distribuce:
 - **Debian** – Jedna z nejstarších distribucí
 - **Fedora** – Sponzoruje firma Red Hat, jejíž distribuce je na tomto systému založena.
 - **Slackware** – Jedna z prvních distribucí, vhodná spíše pro pokročilejší uživatele.
 - **Mandriva** (dříve Mandrakelinux) – Tato distribuce používá balíčkovací systém RPM.
 - **SUSE** – Původně samostatná distribuce, později koupena firmou Novell.
 - **Ubuntu** – Tato distribuce vhodná pro začátečníky vychází z Debianu.

Windows

- Operační systém (či správně operační systémy)
Windows jsou asi nejznámější operační systémy.
- Druhy:
 - Windows pro DOS
 - 3.1x (1992)
 - Win 95, 98, Me
 - Windows NT (1993)
 - Win 2000, XP, 7,8
 - Windows CE
 - Windows Mobile, Windows Phone

Vykonávání instrukcí

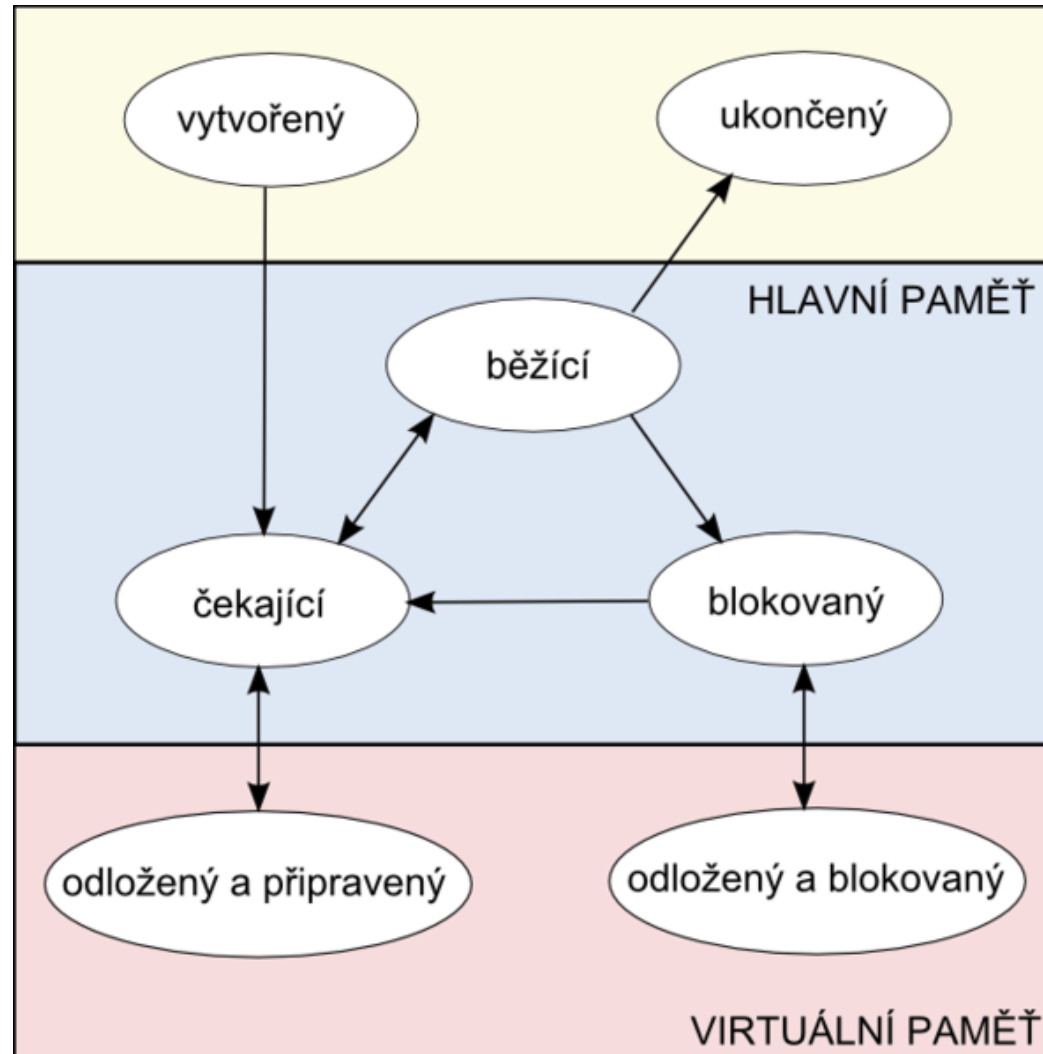
- **program** = sekvence (binárních) kódů instrukcí, registrů procesoru a dat (čísla, texty, hodnoty adres do operační paměti a vstupně/výstupních zařízení)
- stejná (RISC) nebo proměnná (CISC) délka kódů instrukcí – 1 až 2 byty
- **operandy** = parametry instrukcí, registry a data, specifický počet (obvykle 0 až 2), přípustné kombinace pro každou instrukci
- výsledek instrukce často ukládán do prvního operandu
- vykonávání instrukce
 - trvá určitý počet taktů (na vnitřní frekvenci procesoru), jednotky až desítky
 - až 7 fází: např. načtení, dekodování, načtení operandů, provedení, uložení výsledku
 - **pipelining** – částečně současné provádění instrukcí, fáze za sebou, nelze vždy, např. podmíněné skoky

Procesy

Procesy

- **Proces** je spuštěný program
- **Vlákn** je prvek reprezentující vykonávání kódu procesu. Neformálně řečeno, vlákn je „vykonávání kódu“, zatímco proces je „paměť a další prostředky“.
- Stavy procesů:
 - **Vytvořený** (created)
 - **Běžící** (running)
 - **Připravený** (ready)
 - **Blokovaný** (blocked)
 - **Ukončený** (terminated)

Stavy procesů



Plánování procesů

- Plánování procesů řeší výběr, kterému následujícímu procesu bude přidělen procesor a proces tak poběží.
- Tuto úlohu má na starosti **plánovač (scheduler)**, který je součástí operačního systému.
- Plánovače používají dvě základní strategie:
 - **cyklickou obsluhu** – každému procesu je přiděleno stejné časové kvantum
 - **systém priorit** – každý proces má určitou prioritu s tím, že procesům s vyšší prioritou je přidělován procesor přednostně
- V praxi se používá kombinace obou strategií

Přepnutí kontextu

- Při přepínání procesů je nutné, aby proces po opětovném spuštění pokračoval od stejného místa, ve kterém byl přerušen a aby v procesu až na časové zpoždění nebylo poznat, že k přerušení došlo => **přepínání kontextu**
- **uložení kontextu** (anglicky context save) - kompletní uložení stavu procesu
- **obnovení kontextu** (anglicky context restore) – zpětné načtení stavu procesu do procesoru
- **změna kontextu** (anglicky context switch) - uložení kontextu jednoho a obnovení kontextu druhého

Komunikace a synchronizace

- pro procesy a pro vlákna
- procesy jsou paměťově oddělené (každý má svoji přidělenou paměť) - komunikace pomocí speciální **sdílené paměti** a **posíláním zpráv**
- vlákna sdílejí paměť procesu - komunikace pomocí sdílení paměti -> soupeření (race) o sdílenou paměť a jiné zdroje počítače, zvláště při více procesorech
- **chyba souběhu (race condition)** = chybné pořadí běhu vedoucí k nekonzistentním stavům při konfliktních operacích, např. čtení-zápis - fáze operací (i instrukcí procesoru) např. čtení z paměti, operace, zápis do paměti

Synchronizace

- určení specifického pořadí běhu
- **atomické operace** = nedělitelné, nepřerušitelné, sekvenčně prováděné (ty konfliktní), např. nastavení nebo inkrementace atomické proměnné, použití pro implementaci tzv. synchronizačních primitiv
- **synchronizační primitiva**: zámek (mutex), semafor (počítadlo), kritická sekce, událost, monitor a další
- **hardwarová podpora**: atomické instrukce procesoru (test-and-set, fetch-and-add, compare-and-swap aj.), zakázání přerušení (při jednom procesoru), preempce (při více procesorech)
- **softwarové implementace**: Dekkerův (1965), Petersonův (1981) algoritmus, vyžadují atomické uložení hodnoty do proměnné
- implementovaná a poskytovaná OS, ale i využívaná v rámci samotného OS!

Uvážnutí (deadlock)

- vzájemné čekání na výlučně vlastněné zdroje (např. chráněné zámky) při modelu využívání „požadavek na přivlastnění-používání-uvolnění“
- podmínky vzniku:
 1. výlučné vlastnictví,
 2. čekání při vlastnictví jiného,
 3. vzájemné (cyklické) čekání,
 4. nemožnost preempce (násilné odebrání prostředku)
- řešení deadlocku:
 - neřešení (ignorování),
 - detekce a zotavení,
 - prevence (zamezení vzniku, tj. nenaplnění podmínek),
 - vyhýbání se (přidělování prostředků tak, aby nenastaly podmínky)
- dnešní OS neřeší (ignorují)

Správa operační paměti

Správa operační paměti

- S operační pamětí souvisí především tyto funkce, na kterých se podílí hardware a operační systém:
 - přidělování (alokace) a uvolňování (dealokace) paměti procesům na požádání
 - udržování informací o obsazení paměti
 - zabezpečení ochrany paměti – zabránění přístupu procesu k paměti mimo jeho přidělený region
 - realizace virtuální paměti

Přidělování operační paměti

- Operační systém může přidělovat procesům buď:
 - **souvislé úseky paměti**
 - přidělování pevných bloků paměti
 - přidělování bloků paměti proměnné velikosti
 - nebo malé bloky paměti – **stránky**
 - fyzický adresový prostor – **rámce**
 - logický adresový prostor – **stránky**
 - Kdykoliv proces přistupuje do paměti, je logická adresa hardwarově přeložena na fyzickou adresu

Virtuální paměť

- Operační paměť (zde se jí říká primární) rozšířena o místo na pevném disku (sekundární paměť), které je sice výrazně pomalejší, ale také výrazně lacinější
- Systém virtuální paměti funguje stejně jako stránkování, jen fyzický prostor je rozšířen o místo na disku
- Při přístupu procesu ke stránce, která není v primární paměti zajistit její nahrání do primární paměti.
- Pokud není v primární paměti volné místo, je nutno ještě nějakou stránku z primární paměti přemístit do sekundární. Tomuto procesu se říká **swapování**.

Souborový systém

Souborový systém

- Úkolem operačního systému je především:
 - Umožnit procesům ukládat velké množství informací obvykle ve formě souborů.
 - Umožnit přístup k těmto informacím všem procesům, které o to požádají.
- Aby bylo možné tyto úkoly správně plnit, musí být data ve vnější paměti nějak organizována. Organizaci těchto dat se říká **souborový systém**.
- Nejčastější způsob organizace dat je pomocí hierarchického uspořádání **souborů** a **adresářů**.

Souborový systém

- Pevné disky jsou obvykle na fyzické úrovni rozděleny na **oddíly (partitions)**.
- Souborový systém se rozkládá jen na konkrétním oddílu a ne na celém disku.
- Operační systém vnímá jednotlivé oddíly jako tzv. **svazky (volumes)**, které se do systému připojují pomocí tzv. **mountování (mounting)**.

Soubory

- **Soubor** je pojmenovaná posloupnost bytů, která je uložena na nějakém datovém médiu (pevný disk, CD disk, flash disk, ...)
- Každý soubor je charakterizován svým **jménem**.
- Část jména, která je za znakem „.“, se nazývá **přípona** názvu souboru.
- Většina operačních systémů podporuje různé druhy souborů. Nejzákladnější rozdělení je na:
 - **Normální soubory** – soubory, které obsahují uživatelské informace. Tyto se pak rozdělují na textové a binární soubory.
 - **Adresáře** – systémové soubory, které udržují strukturu souborového systému.

Atributy souboru

- Atributy - další informace o souboru:
 - délka – velikost souboru v počtech bajtů,
 - přístupová oprávnění – kdo smí se souborem pracovat (čtení, zápis, ...) pro uživatele a skupiny,
 - vlastník – uživatel vlastnící soubor (též skupina),
 - časové informace (čas vytvoření, čas posledního přístupu k souboru, čas poslední změny v obsahu souboru, ...).
- Tyto atributy jsou většinou uloženy v adresáři, ve kterém je soubor uložen

Adresáře

- **Adresář** je zvláštní druh souboru, který obsahuje množství záznamů
- Každý záznam se týká jednoho souboru (normálního nebo adresáře) a obsahuje jeho jméno, atributy a adresu, kde je na disku uložen.
- Některé operační systémy umožňují vytvořit jen jeden adresář (tento se nazývá **kořen** – root), který obsahuje všechny soubory
- Většina moderních operačních systémů však umožňuje vytvářet libovolné množství adresářů, které vytvářejí hierarchickou strukturu souborového systému.
- **Cesta** – určuje posloupnost adresářů od kořenového adresáře až k samotnému souboru

Linky

- V souborovém systému rozlišujeme dva druhy linků:
 - **Hard link** což je ukazatel na fyzické tělo souboru. Každý pojmenovaný soubor na disku je tedy vlastně hard link. Moderní operační systémy umožňují vytvářet více hard linků k jednomu souboru.
 - **Soft link** je odkaz na jiný soubor, který je specifikován jeho cestou.

Souborové systémy

- **FAT32** je jednoduchý souborový systém, proto je podporován prakticky všemi operačními systémy
 - FAT tabulka
 - kořenový adresář
- **NTFS** byl navržen jako souborový systém pro Windows NT
 - podpora pro přidělování práv k souborům
 - kompresi na úrovni souborového systému
 - šifrování
 - diskové kvóty umožňující nastavit maximálně využitelné místo na diskovém oddíle pro konkrétního uživatele
 - struktura
 - bootovací sektor
 - master file table (MFT), což je seznam všech souborů na disku
 - systémové soubory
 - ostatní soubory.
- **UFS (Unix File Systém)** je souborový systém používaný v Unixu

Správa zařízení

Správa zařízení

- Úkolem operačního systému při správě zařízení je především:
 - **Zajistit nezávislost na zařízení.** To znamená, že například můžeme napsat program, který vytváří a ukládá soubory na libovolná zařízení (disk, flash disk, ...) a pro každé z těchto zařízení není potřeba tento program měnit.
 - **Zajistit ošetření chyb zařízení.** Při čtení nebo zápisu na zařízení se může vyskytnout chyba. Operační systém by měl tuto chybu odstranit, pokud to jde, a pokud ne, měl by tuto chybu ošetřit (pokusit se o danou akci znovu, ohlásit chybu, ...).
 - **Zajistit sdílení zařízení.** Každé zařízení může využívat více uživatelů v tu samou chvíli.
- Tyto cíle zajišťuje operační systém pomocí **ovladačů zařízení**

RAID

- RAID je metoda zabezpečení dat proti selhání pevného disku
- Zabezpečení je realizováno specifickým ukládáním dat na více nezávislých disků, kdy jsou uložená data zachována i při selhání některého z nich
- Metody:
 - Pruhování
 - Zrcadlení
- Druhy: RAID 0, RAID 1, RAID 5, RAID 6

Shell

Shell

- řádkový
 - command.com (příkaz cmd)
- grafický
 - Průzkumník

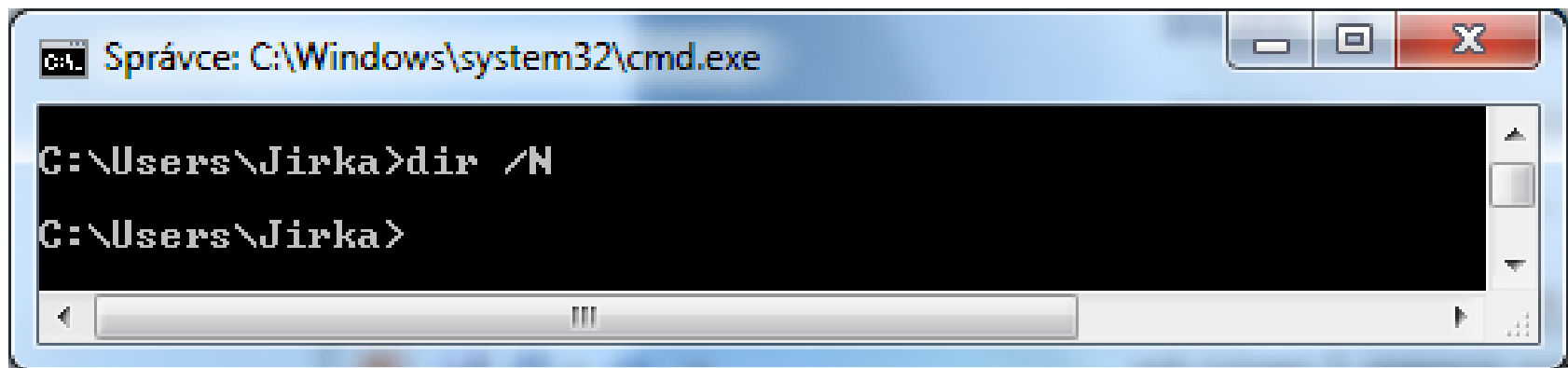
Řádkové příkazy shellu cmd.exe

- dva režimy:
 - **interaktivní režim**, ve kterém jsou vložené příkazy ihned zpracovány
 - **dávkový režim**, který provádí příkazy předem zapsané v dávkovém souboru s příponou **.BAT**
- všechny příkazy jsou provedeny po stisknutí klávesy Enter
- listování v naposledy provedených příkazech pomocí kurzorových kláves
- informační příkazy
- ostatní
- přepínače - operátory přesměrování
- obecné:
 - `/?` – nápověda
 - `help`

Syntaxe příkazů

příkaz /přepínač

- př: `dir /N`
- přepínače – slouží k nastavení parametrů příkazu
- nápověda k každému příkazu:
 - `/?`
 - `help`
 - př: `dir /?`



Pohyb v souborovém systému

- informace o aktuálním místě , kde se v souborovém systému nacházíme je před znakem „>“
- **cd**- změna adresáře
 - Příklady:
 - `cd ..` – přesune se do nadřazeného adresáře
 - `cd test` – přesune se do podadresáře test aktuálního adresáře
 - `cd c:\windows` – přesune se do adresáře c:\windows
- **změna svazku** – napsat kořenový adresář svazku
 - Příklady:
 - `N:` – přesune se na svazek N

Práce s adresáři

- **dir** - vypíše obsah adresáře
 - Příklady:
 - `dir` – vypíše obsah aktuálního adresáře
 - `dir c:\windows` – vypíše obsah adresáře c:\windows
 - `dir c:\windows |more` – odstránkuje výpis
- **md** - vytvoření adresáře
 - Příklady:
 - `md pokus` – vytvoří adresář pokus v aktuálním adresáři
 - `md o:\zacpalj\pokus` – vytvoří adresář pokus v adresáři n:\zacpal
- **rd** - odstranění adresáře

Úkol

1. Přejděte do vašeho domovského adresáře na disku N:
2. Vytvořte zde adresáře fotky, dokumenty a test.
3. V adresáři dokumenty vytvořte podadresář osobní, pracovní.
4. Smažte adresář test.

Práce se soubory

- **copy** – kopírování souborů
 - Příklady:
`copy test.txt n:\test\` – zkopíruje soubor test.txt z aktuálního adresáře do adresáře n:\test\
`copy test.txt n:\test1.txt` – zkopíruje soubor test.txt z aktuálního adresáře do adresáře n:\ a přejmenuje ho na test1.txt
- **xcopy** - kopírování i včetně podadresářů
 - Příklady:
`xcopy O:\Links Linky\` – zkopíruje obsah adresáře O:\Links do adresáře O:\Linky
- **del** - smazání souboru
 - Příklady:
`del test.txt` – smaže soubor test.txt v aktuálním adresáři
`del Linky\` – smaže všechny soubory z adresáře Linky
- **type** - vypíše textově obsah souboru

Úkol

1. Zkopírujte libovolný soubor z adresáře
c:\windows do adresáře dokumenty ve vašem
domovském adresáři.
2. Smažte tento soubor.

Procesy

- **tasklist** – zobrazí běžící procesy systému
- **shutdown** - vypnutí nebo restart počítače
`shutdown -r -f -t 1` - restart PC za 1s s vynuceným ukončením aplikací
- **tskill** - ukončení procesu podle PID či jména, i na vzdáleném PC
- **taskkill** - silnější příkaz pro ukončení procesu, může používat filtry, ukončovat více procesů naráz

Informace

- **systeminfo** - přehled základních info o systému včetně instalovaných hotfixů
- **ver** - vypíše verzi operačního systému
- **time** - zobrazí aktuální čas

Operátory přesměrování

- **> (znak větší)** - přesměruje výstup příkazu do souboru nebo zařízení (tiskárna), standardně směřuje do okna příkazového řádku
 - Příklady:
`dir c:\windows > o:\vypis.txt` - uloží výpis obsahu adresáře do souboru na disk
- **>> (dvakrát znak větší)** - obdoba příkazu >, ale provádí přidání do uvedeného souboru
 - Příklady:
`systeminfo >> o:\vypis.txt` - připojí systémové informace na konec souboru vypis.txt
- **& (ampersand)** - spojí dva příkazy, spustí nejprve jeden a pak druhý
 - Příklady:
`dir c:\ & dir d:` - postupně vypíše obsah adresáře c: a d: