

## Příkazový řádek ÚVOD

Současná počítačová periodika na období kdy MS-DOS vládl světu operačních systémů docela zapomínají. Ano, existují knihy, kde o MS-DOSu (respektive příkazovém řádku) najdete docela dost informací, ale většinou jsou jejich ceny přímo světové.

Při startu MS-DOSu vás zaujme úplně černá obrazovka a v levém horním rohu C: Ovládání počítače s tímto operačním systémem může být docela pomalé a občas to může být tvrdý oříšek, ale vyplatí se to, protože v DOSu se dají dělat opravdu husarské kousky, které jistě oceníte. Pokud se s DOSem opravdu naučíte tak zjistíte (tedy pokud nejste hráč), že v nejhorším byste se bez Windows nebo Linux klidně obešli.

Hardwarové nároky DOSu jsou opravdu velice malé a myslím, že tato věta o operační paměti vše vystihuje:  
640kB should be enough for everybody. (Bill Gates 1981)

A ještě jedna kuriozita. Několik procent počítačové veřejnosti pracuje s DOSem každý den a většinou o tom ani nevědí. Windowsy 1.0 – ME jsou v podstatě na DOSu založené. Některé příkazy mohou posloužit i hackerům jako dobrý prostředek pro získání informací nebo pro administrátory systémů pro diagnostické účely. Existuje více druhů DOSu. Například FreeDOS, OS-DOS, ale Microsoft DOS se zkrátka proslavil nejvíce.

### SPUŠTĚNÍ A OSVOJENÍ PROSTŘEDÍ

Předpokládám, že tento text čtete ve svých malovaných Windows a určitě si chcete spustit DOS. Klikněte na tlačítko Start -> Spustit -> cmd nebo můžete použít klávesovou zkratku Win + R. Když spustíte cmd, nekoukáte na čistokrevný DOS, ale na jeho nástavbu pro Windows. Můžete v něm dělat to samé, co v DOSu a ještě další funkce, které dokáže pouze Windows (příkazy net, ping,...). Zde je velice lehké nastavit barvu pozadí, velikost, barvu i font písma. Klikem na ikonku v záhlaví okna se zobrazí rozbalovací menu, v něm zvolte volbu *Vlastnosti* a poté už je to jednoduché.

Ve Windows 9x, ME je místo *cmd* příkaz *command*. Ten má mnohem menší paletu příkazů a je bližší čistokrevnému DOSu. *Command* zůstává i ve Windows XP i když jeho funkce zde je spíše symbolická. S plnou instalací DOSu na PC se již setkáte velmi zřídka, ale určitě jste se již setkali s bootovací disketou, která je v podstatě Live-distribucí DOSu. Existuje velké množství příkazů a je naprosto nemožné si pamatovat všechny operandy, které příkazy mají proto příkazový řádek nabízí přepínač */?* (např. *dir /?* zobrazí podrobné informace o příkazu *dir*)

### PŘÍKAZY DIR, CD, EXIT, CLS

*dir*

Příkaz *dir* slouží pro výpis složky, takže je asi nejpoužívanějším příkazem. Stejně jako další příkazy, které budeme probírat, jeho chování ovlivňují parametry.

Parametr = za parametry (jsou také někdy nazývány přepínače) se považují znaky za lomítkem (viz. níže). Tyto parametry ovlivňují vykonání příkazu.

Syntaxe: C:příkaz /parametry

| Parametry příkazu DIR     |  |
|---------------------------|--|
| <i>dir /p</i>             | Po zaplnění všech řádků se příkaz pozastaví a počká na stisk libovolné klávesy.        |
| <i>dir /b</i>             | Vypíše pouze názvy souborů bez záhlaví a zápatí.                                       |
| <i>dir &gt; vypis.txt</i> | Uloží výpis obsahu složky do souboru <i>vypis.txt</i> (ten bude v aktuálním adresáři). |
| <i>dir /s</i>             | Vypsány budou i podsložky.   |
| <i>dir /q</i>             | Navíc bude zobrazen sloupec vlastník souboru.  |

Změna aktuální jednotky

Přepnout z jedné diskové (nebo i virtuální) jednotky můžete příkazem: *pismeno\_disku:* (např. *A:*)  
*cd*

Pokud chcete přejít z jedné složky do druhé, je zde příkaz *cd* (Change Directory). Příkaz *cd* je zde nutný, protože pokud byste napsali jenom *system32* a stiskli ENTER, příkazový řádek by si myslel, že jde o soubor a tedy by zahlásil tuto chybu:

*system32* není názvem vnitřního ani vnějšího příkazu,  
sputitelného programu nebo dávkového souboru.

Je tedy nutné napsat *cd system32*. Nic nezkažte pokud budete psát *cd ./system32*. Tento postup by byl nutný u unixových operačních systémů (je proto dobré si na to zvykat). Pokud se chcete vrátit do nadřazené složky není nutné znovu spouštět příkazový řádek, ale k tomuto účelu slouží příkaz *cd...* Ten přejde do nadřazeného adresáře a příkazem *cd../NASM* – zabijete dvě mouchy jednou ranou, protože tento příkaz přejde do nadřazeného adresáře a současně přejde do adresáře *NASM*, který se tam nachází. Napsáním příkazu *cd* bez parametrů se vypíše aktuální složka (užitečné při dávkách).

exit

Zavírat příkazový řádek křížkem v pravém horním rohu je zde pokládáno za nekorektní ukončení. Je lepší příkazový řádek ukončovat příkazem exit.

cls

Při delší práci s příkazovým řádkem se dříve nebo později zaplní obrazovka, a proto je dobré pro větší přehlednost občas použít příkaz cls (Clear Screen), který vymaže obrazovku příkazového řádku a kurzor se posune zpět na výchozí pozici při spuštění.

### PŘÍKAZY PROMPT, DOSKEY, ECHO

prompt

Upravuje výzvu (jako výzva je v příkazové řádce C:\ – píše se za to příkazy) příkazové řádky. Vy jste zvyklí psát příkazy za C:\navez\_slozky\navez\_podrazene\_slozky> . Pomocí příkazu prompt můžete poněkud "neatraktivní" C:\ změnit třeba na 20.45 navez\_slozky a nebo jenom na > . Pro bližší informace je zde nápověda příkazové řádky, kterou vyvoláte prompt /? .

echo

Pomocí příkazu echo můžete vypisovat na obrazovku zprávy. Tento příkaz se obzvlášť hodí u dávek. Příkaz echo můžete kombinovat i s proměnnými. Zkuste napsat do příkazového řádku echo Ahoj %username% a výsledek uvidíte na své obrazovce. Takových proměnných je velké množství, ale blíže se jim budeme věnovat v kapitolách o dávkách.

doskey

Teď výjimečně začnu výklad příkladem. Spustíte příkazový řádek (cmd) a k tomu command . Zkuste v cmd něco napsat a stisknete šipku vlevo. A teď to samé udělejte v command. Jak vidíte, v cmd se stalo to samé jako v textovém editoru. Kurzor se posunul o pole vlevo, ale v command se znak vlevo od kurzoru smaže, jako kdybyste použili klávesu Backspace. Dále zkuste v command i v cmd klávesy Home, End a Insert. V command se nic nestane, ale v cmd se stane úplně to samé, jako v textovém editoru. Myslím, že jste pochopili význam doskey. Zde jsou další klávesy s kterými umí doskey pracovat. V cmd je doskey zapnut implicitně.

|        |   |
|--------|---|
| F1     | Použij znak z minule zadaného příkazu.  |
| F2     | Po stisku se budou vypisovat všechny předcházející znaky minule zadaného příkazu. |
| F3     | Vypíše všechny zbývající znaky z minulého příkazu.                                |
| F4     | Zobrazí výzvu pro zadání znaku, po který se má text vymazat                       |
| F5     | Nabídne přechází zadaný příkaz. Stejnou funkci má i šipka nahoru                  |
| F6     | Vloží znak konec souboru (Ctrl+Z)   |
| F7     | Zobrazí historii příkazů.   |
| Alt+F7 | Smaže historii příkazů  |

### PŘÍKAZY COLOR, TITLE, CHCP, COPY

color

Pomocí příkazu color můžete měnit barvu pozadí a písma v cmd. Číslem barev může být číslo od 1 – 15. Čísla 10 – 15 se tradičně píše v šestnáctkové soustavě (A, B, C, D, E). Zde jsou dvě ukázky:

|          |                             |
|----------|-----------------------------|
| color 24 | Kombinace dobrá pro vojáky. |
| color f0 | Černá na bílém.             |

title

Slouží ke změně titulku příkazového řádku. Když zadáte title Nazdar! tak v titulku bude místo cmd váš text Nazdar! .

chcp

Chcp slouží ke změně aktuální kódové stránky. Microsoft prosadil kódování nazvané Latin II nebo také CP 852. Ve Windows je toto kódování nazýváno Středoevropské kódové jazyky (DOS) . S nástupem Windows 95 zavedl kódování Windows 1250, dnes většinou nazývané jako Středoevropské jazyky (Windows). Aby toho nebylo málo, svět mimo Microsoft zavedl kódování ISO-8859-2, ve WinXP pojmenované jako Středoevropské jazyky (ISO). Příkazový řádek pracuje s kódováním CP 852 a to je také příčinou, že text importovaný z příkazového řádku je ve Windows zobrazen chybně. Nejlépe všechno vysvětlí pokusy. Importujte obsah aktuální složky do textového souboru ( dir > C:seznam.txt ) a teď to otevřete v textovém editoru. A teď otevřete pomocí příkazu type (slouží k výpisu textového souboru přímo do okna příkazového řádku) nějaký český text. A budou tam taky chyby.

Výsledek z toho asi vyvodíte. Pro změnu kódování ve Windows slouží příkaz `chcp ciselna_hodnota`. Před exportem jakéholiv textu z `cmd` změňte kódovou stránku na `chcp 1250`.

`copy`

Slouží pro zkopírování souboru a nebo více souborů (to jedna z hlavních výhod tohoto příkazu, že můžete jedním příkazem zkopírovat `x` souborů a nemusíte je ani vypisovat). Syntaxe je následující: `C:\copy název_zdroje název_cíle`

|   |   |
|---|---|
| <code>copy zprava.txt texty</code>          | Zkopíruje soubor <code>zprava.txt</code> do složky <code>texty</code> .   |
| <code>copy zprava.txt textykopie.txt</code> | Zkopíruje soubor do složky <code>texty</code> , ale zkopírovaný soubor bude mít nový název <code>kopie.txt</code> . |
| <code>copy *.txt data</code>                | Zkopíruje všechny soubory s příponou <code>.txt</code> do složky <code>data</code> .                                |
| <code>copy *.txt novy.txt</code>            | Spojí všechny soubory s příponou <code>.txt</code> do jednoho souboru <code>novy.txt</code>                         |

### PŘÍKAZY MKDIR(MD), RMDIR, DEL, RENAME, MOVE

`mkdir(md)`

Pomocí `mkdir` (respektive `md`) můžete vytvářet složky. Při zadání příkazu `mkdir c:\data\texty\ahoj` a ani jedna ze složek nebude existovat, vytvoří příkaz i cestu.

|   |   |
|---|---|
| <code>mkdir texty</code>                | Vytvoří podsložku aktuálního adresáře.  |
| <code>mkdir c:\programy\aplikace</code> | Bez ohledu na aktuální umístění vytvoří nový adresář, který bude podsložkou adresáře <code>C:\programy</code> . |

`rmdir (rd)`

Příkaz `mkdir` (make directory) vytváří a `rmdir` (remove directory) maže složku. S přepínačem `/s` se vypíše obsah složky s dotazem na jestli skutečně chceta smazat tento adresář. Přepínače `/s /q` smaží složku bez jakýchkoli dotazů.

`del (erase)`

Příkaz `del` i `erase` mažou soubory. Pro smazání všech souborů s příponou `TXT` v aktuální složce stačí napsat `del *.txt`. Tato hvězdičková konvence se dá použít i naopak (`del fotka.*`). Také můžete smazat všechny soubory v aktuálním adresáři příkazem `erase *.*`.

`rename`

Příkaz slouží pro přejmenování souborů nebo adresářů.

`move`

Příkaz slouží k přesunutí souborů nebo složek. Můžete také použít hvězdičkovou konvenci k přesunutí více souborů (adresářů) najednou. `move *.txt adresar` přesune všechny textové soubory s touto příponou do složky `adresar`.

### PŘÍKAZY XCOPY, ATTRIB, TREE

`xcopy`

`XCOPY` rozšiřuje možnosti kopírování v příkazové řádce. Příkazem `xcopy` lze bez problémů kopírovat i složky. Při pokusu o zkopírování složky pomocí `copy`, příkaz nefungoval korektně. Syntaxe je standardní, takže `xcopy CO KAM /parametry`. Atributů tohoto příkazu je velké množství a vzhledem k jednoduchosti použití je zbytečné je vypisovat, takže vás odkazuji na nápovědu k `cmd /?`.

`Attrib`

Příkazem můžete měnit atributy souborů. Příkazový řádek zná následující atributy:

|   |               |   |
|---|---------------|---|
| R | Jen pro čtení | Soubor (složku) s tímto obsahem nelze modifikovat, lze pouze číst.  |
| A | Archivovat    | Aplikace Windows se chovají tak, že pokud do nějakého souboru zapíše data, opatří jej atributem Archivovat.   |
| H | Skrytý        | Soubor (složka) není ve Windows standardně viditelný.   |
| S | Systémový     | Tento druh atributu má asi největší utajení a Windows se jeho smazání brání zuby, nehty. Tímto atributem jsou chráněny třeba soubory jako <code>boot.ini</code> nebo <code>autoexec.bat</code> , které jsou v kořenovém adresáři disku. |

Příkaz attrib bez parametrů vypíše všechny soubory v aktuální složce s jejich atributy. Atributy můžete v cmd nastavovat takto:

|                         |  |
|-------------------------|--|
| attrib -r               | Zruší parametr jen pro čtení u všech souborů v aktuální složce.                              |
| attrib +r *.txt         | U všech souborů v aktuální složce s příponou .txt nastaví atribut Jen pro čtení.             |
| attrib +h -r            | Nastaví všem souborům v aktuální složce atribut "Skrytý" a odebere atribut "Pouze ke čtení". |
| attrib +r<br>a:*.*/s/d- | Uzamkne všechny soubory a složky na disketě. Tomu se říká SÍLA KOMBINACE                     |

tree

Zobrazí strom aktuální složky nebo disku. Příkazem C:\tree > strom.txt exportujete celý strom do souboru

strom.txt a to by se mohlo hodit

### PŘÍKAZY SUBST, SHUTDOWN, FDISK, FORMAT

subst

Pomocí tohoto příkazu můžete vytvořit logický disk. Logický disk se chová jako disková jednotka. Tento disk se dá kdykoliv zrušit a kdykoliv znovu vytvořit. Používá se hlavně pro rychlejší přístup k oblíbeným složkám. Já používám logický disk pro přístup k souboru nasm.exe, který se nalézá na umístění: C:\NASM\SRC a vždy musím zadávat tuto dlouhou cestu. Takže si radši vytvořím virtuální disk, který bude touto složkou. Upozorňuji, že jakékoliv změny ve virtuálním disku změní ekvivalentně i složku nebo i disk, jehož je virtuální disk prostředníkem. Syntaxe je: subst písmeno\_noveho\_disku cesta\_ke\_složce. Teď několik příkladů.

|                         |   |
|-------------------------|---|
| subst m: c:\            | Obsah jednotky C: bude i v jednotce m:                                    |
| subst n:<br>c:\nasm\src | Vytvoří se nový disk n: v kterém bude přístupný obsah složky c:\nasm\src. |
| subst m: /d             | Odebere logický disk m:   |

Nevýhodou je, že po restartu PC virtuální disk zmizí. Nejjednodušším postup, jak zabezpečit, aby se disk po restartu neztrácel je následující:

1. Vyvolejte v jakékoliv složce místní nabídku a zvolte možnost Nový -> Zástupce
2. Do okna "Zadejte umístění položky" zadejte: cmd /c subst p: c:\navez\_složky
3. Nakopírujte tohoto zástupce do složky C:\Documents and Settings\vase\_uzivatelske\_jmeno\Nabídka Start\Programy\Po spuštění

fdisk

Na rozdíl od subst, příkaz fdisk rozděluje disk trvale. Upozorňuji, že fdisk umí rozdělovat pouze disk se souborovým formátem FAT, FAT16, FAT32 a neporadí si s NTFS. Disky s NTFS umí jedinečně odstranit, ale neumí s nimi pracovat. Pevný disk se pomocí tohoto příkazu musí rozdělovat prázdný, protože jinak na 99 % způsobí poškození dat. Program fdisk se na spouštěcí disketě vytvořené ve Windows XP nenajdete, ale můžete si ho stáhnout z internetu nebo z bootovací diskety Win98 nebo ME. Příkaz fdisk nelze spustit z GUI Windows, ale musíte ho spouštět právě z bootovací diskety. Fdisk je tzv. externí příkaz (viz další lekce) a spustíte ho standardně příkazem fdisk. Průvodce vás poté provede rozdělením disku. Pomocí volby disku "Odstranit diskový oddíl" můžete disk i dost rychle smazat, protože fdisk narozdíl od příkazu format maže přímo partice nikoliv data. Myslím, že lepší je pro tyto účely použít nějaký specializovaný program, třeba jako Partition Manager (update 17.8.2014 – dnes by ale doporučil spíš využívat instalační DVD Windows nebo Linux distra, jelikož je to bezpečnější, rychlejší i mnohem snazší).

shutdown

Slouží k vypnutí, restartování nebo odhlášení uživatele přes příkazový řádek. Často se používá pro vzdálené vypnutí počítače nebo pro vypnutí více počítačů najednou. Takže počítače v celé síti můžete vypnout spuštěním jedné dávky. Lze také vypnutí PC načasovat. Vše je popsáno v nápovědě příkazové řádky (update 17.8.2014 – je to také super kanadské žertík. Přidat dávku příkazu shutdown s odpočítáváním k automatickému spuštění po nastartování počítače – nejlépe s odpočítáváním).

format

Používá se k formátování diskových oddílů. Windows neumožňuje smazání aktuálního diskového oddílu, ale můžete smazat přes příkazový řádek neaktivní diskový oddíl (disketu, pevný disk). V operačním systému Windows XP si můžete vybrat jaký souborový systém chcete pro nově zformátovaný disk (parametr /FS: souborovy\_system), jmenovku, velikost stopy atd. O všem informuje nápověda k příkazové řádce. Při jakémkoliv formátování doporučuji používat parametr /u, který provede kompletní formátování i s kontrolou povrchu disku apod. Příkaz format doporučuji rovněž spouštět z bootovací diskety. Jenom bych vás chtěl upozornit, že pokud

smažete data příkazem format nebo partici příkazem fdisk, data se nesmažou, ale pouze nejsou vidět. Existují stovky programů, které dokážou data znovu obnovit. Pokud chcete disk zformátovat bez možnosti obnovy použijte některý ze specializovaných nástrojů.

### VŠECHNY PŘÍKAZY V MS-DOS

Seznam příkazů příkazové řádky jsem exportoval z nápovědy cmd. V další části je kompletní instrukční sada MS-DOSu verze 6.22. Tuto instrukční sadu jsem rovněž opsal, ale již ne z nápovědy, ale ze skvělé knihy Tomáše Baránka – "Kompletní referenční průvodce" (rok vydání 1994)

#### Interní příkazy x Externí příkazy

Interní příkazy jsou součástí kódu příkazové řádky (respektive MS-DOSu) a jsou k dispozici kdykoliv. Externí příkazy jsou programy jako ostatní (třeba Malování), ale komunikují s námi prostřednictvím příkazové řádky.

#### Instrukční sada příkazové řádky ve Windows XP

Chcete-li získat další informace o příkazu, zadejte příkaz HELP název\_příkazu.

- ASSOC Zobrazí nebo změní přiřazení koncovek souborů.
- AT Naplánuje běh příkazů a programů v počítači.
- ATTRIB Zobrazí nebo změní atributy.
- BREAK Nastaví nebo vynuluje rozšířenou kontrolu CTRL+C checking.
- CACLS Zobrazí nebo změní seznam ACL souborů.
- CALL Zavolá dávkový soubor z jiného.
- CD Zobrazí název nebo změní aktuální složku.
- CHCP Zobrazí nebo nastaví aktivní kódovou stránku.
- CHDIR Zobrazí název nebo změní aktuální složku.
- CHKDSK Zkontroluje disk a zobrazí zprávu o stavu.
- CHKNTFS Zobrazí nebo změní kontrolu disku při spuštění.
- CLS Vymaže obrazovku.
- CMD Spustí novou instanci příkazového řádku systému Windows.
- COLOR Nastaví výchozí barvu písma a pozadí konzoly.
- COMP Porovná obsah dvou souborů nebo sad souborů.
- COMPACT Zobrazí nebo změní kompresi souborů v oddílu NTFS.
- CONVERT Převede svazek FAT na NTFS. Není možné převádět aktuální jednotku.
- COPY Zkopíruje jeden nebo více souborů do jiného umístění.
  - DATE Zobrazí nebo nastaví datum.
  - DEL Odstraní jeden nebo více souborů.
  - DIR Zobrazí seznam souborů a podsložek složky.
  - DISKCOMP Porovná obsah dvou disket.
  - DISKCOPY Zkopíruje obsah jedné diskety na jinou disketu.
- DOSKEY Edituje příkazy, znovu volá příkazy a umožňuje vytvářet makra.
- ECHO Zobrazí zprávu nebo přepne stav zobrazování zpráv na zapnuto nebo vypnuto.
- ENDLOCAL Konec lokalizačních změn prostředí v dávkovém souboru.
  - ERASE Vymaže jeden nebo více souborů.
  - EXIT Ukončí program CMD.EXE (příkazový řádek).
- FC Porovná dva soubory nebo sady souborů a zobrazí rozdíly.
  - FIND Vyhledá textový řetězec v souborech.
  - FINDSTR Vyhledá textový řetězec v souborech.
- FOR Spustí určený příkaz pro všechny soubory v sadě.
- FORMAT Formátuje disk pro použití v systému Windows.
- FTYPE Zobrazí nebo změní typ souboru, použitý v přidruženích rozšíření.
- GOTO Interpret příkazového řádku přejde k vykonávání instrukcí v označeném řádku dávkového souboru.
  - GRAFTABL Povolí zobrazování sady rozšířených znaků v grafickém režimu.
- HELP Poskytne informace o příkazech příkazového řádku systému Windows.
- IF Zpracovává příkazy v dávkovém souboru na základě podmínek.
  - LABEL Vytvoří, mění nebo smaže jmenovku disku.
    - MD Vytvoří složku.
    - MKDIR Vytvoří složku.
  - MODE Konfiguruje systémové zařízení.
- MORE Zobrazí najednou pouze jednu obrazovku výstupu.
- MOVE Přesune jeden nebo více souborů z jedné složky do jiné.
- PATH Zobrazí nebo nastaví cestu vyhledávání spustitelných souborů.
- PAUSE Pozastaví zpracování dávkového souboru a zobrazí zprávu.
- POPD Obnoví předchozí hodnotu aktuální složky, uloženou příkazem PUSH.D.
  - PRINT Vytiskne textový soubor.
  - PROMPT Změní tvar příkazového řádku Windows.
  - PUSH.D Uloží aktuální složku a pak ji změní.
  - RD Smaže složku.
- RECOVER Obnoví čitelné informace z chybného nebo poškozeného disku.
- REM Označuje komentář v dávkovém souboru nebo souboru CONFIG.SYS.

REN Přejmenuje soubor nebo soubory.  
 RENAME Přejmenuje soubor nebo soubory.  
 REPLACE Přepíše soubory.  
 RMDIR Smaže složku.  
 SET Zobrazí, nastaví, nebo smaže nastavení prostředí systému Windows.  
 SETLOCAL Začátek lokalizačních změn prostředí v dávkovém souboru.  
 SHIFT Posune pozici nahraditelných parametrů v dávkovém souboru.  
 SORT Setřídí vstup.  
 START Spustí určený program nebo příkaz v novém okně.  
 SUBST Přiřadí písmenu jednotky cestu.  
 TIME Zobrazí nebo nastaví čas systému.  
 TITLE Nastaví nadpis okna relace programu CMD.EXE.  
 TREE Graficky zobrazí strukturu složek jednotky nebo cesty.  
 TYPE Zobrazí obsah textového souboru.  
 VER Zobrazí verzi systému Windows.  
 VERIFY Určuje, zda má systém Windows kontrolovat správnost zapsání souborů.  
 VOL Zobrazí jmenovku jednotky a sériové číslo.  
 XCOPY Kopíruje soubory a stromy složek.  
 Instrukční sada MS-DOS 6.22

#### interní příkazy

Příkazy pro práci se soubory

COPY kopíruje soubory  
 DEL maže soubory  
 REN provádí přejmenování souboru  
 TYPE vypíše obsah (textového) souboru

Příkazy pro práci s disky a adresáři

DIR vypíše obsah adresáře  
 CD nastaví pracovní adresář (přepne do pracovního adresáře)  
 MD vytvoří adresář  
 RD vymaže prázdný adresář

Příkazy pro práci s datem a časem

DATE čte a nastavuje systémové datum  
 TIME čte a nastavuje systémový čas

Obecné řídicí a konfigurační příkazy

BREAK nastavuje způsob přerušení programu klávesami Ctrl-Break (Ctrl+c)  
 CHCP nastaví kódovou stránku používané tabulky znaků  
 CLS maže obsah textové obrazovky  
 CTTY vybírá standardní zařízení počítače pro vstupně/výstupní operace  
 EXIT ukončuje činnost stávajícího interpretu příkazů a vrací řízení na vyšší úroveň interpretu  
 LH načte program do horní paměti (upper memory)  
 VER zobrazí číslo aktuálně používané verze systému MS-DOS  
 VERIFY zapíná kontrolu ověřování správnosti při jakýchkoliv přenosech souborů  
 VOL zobrazí jméno disku a jeho sériové číslo

Proměnné prostředí a příkaz pro práci s nimi

PATH Proměnná PATH (cesta) viz.dole  
 PROMPT Proměnná PROMPT (výzva) definuje způsob zobrazení výzvy operačního systému v příkazové řádce  
 SET definuje proměnné prostředí, zobrazuje je a přiřazuje jim hodnoty.

Příkazy pro činnost dávkových souborů

CALL, FOR, GOTO, IF, REM, SHIFT o nich až dále

#### Externí příkazy

Příkazy pro práci se soubory a adresáři:

MOVE Přesouvá soubory a přejmenovává adresáře  
 UNDELETE Obnovuje vymazané soubory  
 XCOPY Kopíruje celé adresářové struktury  
 REPLACE Řídí nahrazování souborů  
 ATTRIB Zobrazuje a upravuje atributy souborů  
 TREE Zobrazuje adresářový strom disku  
 DELTREE Maže celé adresářové stromy  
 RESTORE Obnovuje soubory uložené příkazem BACKUP z dřívějších verzí DOSU  
 FC Porovnává obsah souborů  
 MSBACKUP Zálohují soubory a adresáře  
 SUBST Vytváří umělý, virtuální disk  
 SHARE Definuje podmínky pro sdílení a zamykání souborů  
 EDIT Spouští jednoduchý textový editor

EXPAND Rozbaluje soubory

Příkazy pro práci s diskem

FORMAT Formátuje disk nebo disketu  
FDISK Připravuje médium k formátování  
UNFORMAT Obnovuje data ztracená formátováním  
LABEL Připravuje disku jméno  
SYS Vytváří systémový disk  
DISKCOPY Kopíruje celé disky  
DISKCOMP Porovnává obsahy disků  
CHKDSK Kontroluje integritu souborů na disku  
SCANDISK kontroluje a opravuje chyby na disku  
SMARTDRV Vytváří vyrovnávací paměť (cache) disku  
DEFRAG Odstraňuje fragmentaci disku  
DRVSPACE Komprimuje disk

Příkazy pro práci s pamětí

MEM Zkoumá obsah paměti počítače  
MEMMAKER Optimalizuje využití paměti  
LOADFIX Načítá program do horních 64 kB paměti

Klávesnice a obrazovka, národní nastavení

DOSKEY Umožňuje editaci příkazové řádky a tvorbu příkazových maker  
KEYB Zapíná na klávesnici národní sadu znaků  
GRAPHICS Podporuje tisk obrazovky na různých tiskárnách  
COUNTRY Instaluje národní podporu

Periférie a komunikace

MODE Nastavuje základní komunikační parametry rozhraní  
PRINT Definuje parametry tisku a řídí tisk na pozadí  
MSCDEX Instaluje rozhraní pro jednotku CD-ROM  
INTERLNK Instaluje podporu komunikace dvou počítačů po LPT nebo COM  
INTERSVR Stejně jako předchozí, akorát na straně serveru

Ostatní externí příkazy a programy

MSD Diagnostický test počítače  
MORE Systémový filtr, pauzující výpisy po obrazovkách  
FIND Systémový filtr, nalézá řetězce  
SORT Systémový filtr, třídí výstupy  
POWER Snižuje spotřebu energie procesoru  
MSAV Antivirový program  
DEBUG Ladící a programovací prostředek  
QBASIC Programovací jazyk QuickBASIC  
VSAFE Rezidentní antivirový program  
HELP Návod MS-DOSu  
FASTHELP Stručná nápověda MS-DOSu  
FASTOPEN Ukládá informace o otevřených souborech do paměti, zrychluje tak přístup k nim.  
SETVER Nastavuje verzi operačního systému pro korektní programy  
CHOICE Poskytuje uživateli výběr z více odpovědí

Programová podpora Microsoft Windows

MWAVTSR, MWBACKUP, MWAB, SMARTMON, MWUNDEL - nyní už zbytečné  
Path

Path definuje cesty, ve kterých bude příkazová řádka hledat spustitelné soubory. Např. příkaz format je uložen v adresáři C:\Windows\System32, ale i když se nalézáte v kořenovém adresáři disku, přesto se format spustí. To je definováno právě příkazem PATH. Cesta C:\>PATH C:; C:\Windows; C:\Windows\System32;

C:\moje\_slozka znamená, že po napsání externího příkazu bude nejdříve prohledáván kořenový adresář disku, poté složka C:\Windows, poté složka C:\Windows\System32 a nakonc C:\moje\_slozka.

**PŘÍKAZY NET – 1. ČÁST**

Příkazy skupiny net jsou velice užitečné pro práci se sítí, uživatelskými účty a vůbec pro administrační účely. Toho také mohou využít začínající hackeři. Například pomocí dvou příkazů můžete vytvořit nový účet administrátora (cca 10 sekund). Příkazy net jsou tyto:

|              |             |           |
|--------------|-------------|-----------|
| net accounts | net help    | net share |
| net computer | net helpmsg | net start |

|              |                |                |
|--------------|----------------|----------------|
| net config   | net localgroup | net statistics |
| net name     | net stop       | net pause      |
| net continue | net print      | net use        |
| net file     | net send       | net user       |
| net group    | net session    | net view       |

Příkazy net jsou podporovány až ve Windows 2000! Než začnu popisovat, chtěl bych vám doporučit publikaci, z které jsem se hodně naučil o příkazech net. Je to kniha Jakuba Zemánka Slabá místa Windows aneb jak se bránit hackerům

#### Net Accounts

Slouží k nastavení uživatelských účtů (např. minimální požadovaná délka hesla, doba platnosti účtu, ...). Příkaz bez parametrů vypíše aktuální nastavení. Pro další vysvětlování příkazů net doporučuji vytvořit další uživatelský účet na kterém budete moci experimentovat. Zde jsou parametry:

|                             |  |
|-----------------------------|--|
| /forcelogoff: {minuty   NO} | Čas v minutách do vynuceného odhlášení uživatele. Hodnota NO znamená neomezeně a ta je nastavena implicitně.                                       |
| /minpwlen:délka             | Nastaví minimální počet znaků pro délku hesla. Můžete zadat nejvíce 14 znaků. Pokud nastavíte více bude heslo nepoužitelné. Současná hodnota je 6. |
| /maxpwage:{dny   UNLIMITED} | Doba platnosti hesla. Hodnota UNLIMITED znamená neomezeně a je nastavena implicitně  |
| /minpwage:dny               | Nastaví minimální možnou dobu před možností změny hesla.   |
| /Uniquepw:číslo             | Požaduje jedinečnost uživatelského hesla během určeného počtu jeho změn. Maximální hodnota je 24. Nikdy jsem nepoužil.                             |
| /domain                     | Řadič stávající domény. Rovněž jsem nikdy nepoužil.  |

#### net config

Zobrazí konfiguraci síťových služeb "Server" a "Workstation". Zde je ukázka výpisu.

```
Název počítače          \PCMICHAL
Celý název počítače    pcmichal
Uživatelské jméno      michal
```

Pracovní stanice je aktivní

```
Verze softwaru          Windows 2002
```

```
Doména pracovní stanice      SKUPINA
Název DNS domény pracovní stanice (null)
Přihlašovací doména         PCMICHAL
```

```
Časový limit otevření COM (s)      0
Počet bajtů odeslání COM             16
Časový limit odeslání COM (ms)     250
```

Příkaz byl úspěšně dokončen.

#### Net Statistics

Příkaz zobrazí výčet snad všech možných statistik síťových aktivit počítače. Pokud použijete tento příkaz na nějaký server zjistíte určitě zajímavé informace.

#### Net Session

Správa síťových relací daného počítače. Lidsky řečeno příkaz vám jasně zobrazí, kdo, kdy a odkud sdílel data na vašem počítači a pomocí tohoto příkazu máte možnost sdílení okamžitě přerušit. Syntaxe je net session [\počítač]

[/delete]

## PŘÍKAZY NET – 2.ČÁST

net user

Příkaz net user je velice nebezpečná zbraň v rukou hackera a také užitečný příkaz v rukou administrátora.

net user karel 85hgj6@b /add

net localgroup administrators karel /add

Právě jsem vytvořil uživatelský účet karel, s dokonalým heslem a přidělil jsem mu administrátorská práva. To může být užitečné, když správce počítače odejde na minutku na záchod a vy máte k počítači přístup. Zde jsou nejdůležitější parametry:

|                          |  |
|--------------------------|--|
| /add                     | Přidá uživatelský účet do databáze uživatelských účtů.   |
| /delete                  | Smaže uživatelský účet.  |
| /active:{yes   no}       | Aktivuje či deaktivuje daný účet. Pokud není účet aktivován, nemůže uživatel přistupovat na server. Výchozí nastavení je yes. Tento parametr může být velice užitečný v kombinaci s příkazem net accounts, ve kterém nastavíte, že pokud uživatel třeba 5x špatně napíše heslo, tak se nastaví tato položka na no. To je často používaná metoda zabezpečení. |
| /comment:"text"          | Umožňuje napsat si poznámku k účtu (max 48 znaků)  |
| /expires:{datum   never} | Datum vypršení účtu. Formát datumu je: mm/dd/yy, či dd/mm/yy podle toho jaký je nastaven kód země.   |
| /fullname:"jméno"        | Plné jméno uživatele.  |
| /homedir:cesta           | Nastaví cestu domovského adresáře uživatele. Tato cesta musí existovat.  |
| /passwordchg:yes   no    | Určuje, zda si mohou uživatelé měnit heslo.  |
| /passwordreq:yes   no    | Určuje, zda uživatelský účet musí mít heslo.   |
| /profilepath[:cesta]     | Nastaví cestu profilu přihlášení uživatele.  |
| /times:{doby   all}      | Doby přihlášení. Doby TIMES mají formát den[-den][,den[-den]], čas[-čas][,čas[-čas]].  |

net share

Správa sdílených prostředků počítače. S jeho pomocí lze nasdílet data na HDD. Syntaxe je: net share jednotka: cesta

|              |  |
|--------------|--|
| /user:počet  | Určuje počet uživatelů, kteří se mohou současně připojit ke sdílenému prostředku.                    |
| /unlimited   | Ke sdílenému prostředku se může najednou připojit neomezený počet uživatelů – NEDOPORUČUJI!          |
| /delete      | Zastaví sdílení prostředku.  |
| Cache:Manual | Povolí ruční ukládání souborů a programů do mezipaměti klienta z této sdílené položky. NEDOPORUČUJI! |

|                |   |
|----------------|---|
| Cache:Programs | Povolí automatické ukádání dokumentů a programů do mezipaměti z této sdílené položky. |
| /cache:none    | Zakáže ukládání do mezipaměti z této sdílené položky.                                 |

net localgroup

Správa místních uživatelů počítače. Při spuštění bez parametrů vypíše všechny aktuální skupiny. V tabulce jsou uvedeny přepínače a příklady použití.

|   |  |
|---|--|
| /comment:"text"                             | Přidá poznámku k nové či existující skupině. Text je nutné uzavřít do uvozovek.                |
| /domain                                     | Provede operaci na řadiči aktuální domény. Jinak se operace provádí na místní pracovní stanici |
| /add  | Přidá jméno nebo pracovní skupinu do místní skupiny. Blíže v příkladech dole.                  |
| /delete                                     | Odstaraní skupinu nebo jméno uživatele z místní skupiny.                                       |
| net localgroup administrators karel /add    | Přidá účet Karel do skupiny administrators.  |
| net localgroup administrators karel /delete | Vymaže účet karel ze skupiny administrators.   |
| net localgroup opravari /add                | Přidá uživatelskou skupinu opravari.   |

net print

Vzdálená správa sdílených tiskáren a jejich tiskových front.

|          |   |
|----------|---|
| /hold    | Pozastavuje vytištění tiskové úlohy. Úloha zůstává ve frontě. |
| /release | Reaktivuje úlohu, která je pozastavena.                       |
| /delete  | Odstraňuje úlohu z tiskové fronty.                            |

net send

Příkaz slouží k posílání zpráv mezi uživateli. Vše ostatní obsahuje nápověda příkazové řádky.

Úplné smazání hesel ve Windows

Možná namítnete, že to do seriálu MS-DOS nepatří, ale hned vás přesvědčím, že patří a možná víc než si myslíte. To že je systém Windows děravý, je všeobecně známo, ale to že se v tomto operačním systému najde taková mezera, která zatím nebyla odstraněna mě docela zarazí (Update 18.8.2014 – již byla odstraněna, od XP, Vista, 7 nefunguje). Pokud se vám stane, že zapomenete heslo, disketu na heslo nemáte a prostě jste v háji, tak nezoufejte. Pomůže vám systém MS-DOS. Spusťte počítač z bootovací diskety a pomocí vašich současných znalostí se dostaňte do adresáře C:\windows\system32\config , kde se nalézá soubor SAM a ten když smažete tak máte vyhráno. Touto fintou se smažou hesla všech uživatelů, včetně administrátorského. Ochranou je souborový systém NTFS s kterým neumí MS-DOS pracovat, ale na internetu si můžete stáhnout NTFSDOS a s tím už tohle není problém. Můžete také použít GNU/Linux popřípadě jiný systém Windows.

**PŘÍKAZY IPCONFIG, PING, SYSTEMINFO, FTP**

ping

Jedná se o jeden z rodiny příkazů TCP/IP protokolů, který zasílá protokolem ICMP požadavek na odezvu vzdáleného počítače a přitom ukáže jeho IP adresu (jeho časté využití je k zjištění IP adresy). Použití příkazu ping je také často blokováno (např. www.microsoft.com).

|             |  |
|-------------|--|
| -t          | Opakovaně odešle určenému hostiteli žádost o odezvu. |
| -n počet    | Počet odesílaných žádostí o odezvu.                  |
| -l velikost | Velikost vyrovnávací paměti k odesílání.             |

|                     |  |
|---------------------|--|
| -f                  | V paketu nastaví nefragmentovat                                    |
| -i TTL              | Hodnota Time To Live   |
| -v TOS              | Typ služby   |
| -r počet            | Znamená cestu pro zadaný počet směrovačů.                          |
| -s počet            | Časové razítko pro zadaný počet směrovačů.                         |
| -k(-j)<br>hostitelé | Seznam hostitelů určuje všechny uzly cesty (-j hlavní uzly cesty). |
| -w čas_limit        | Časový limit čekání na odpověď (v ms).                             |

#### ipconfig

Kdo ovládá tento příkaz, ovládá připojení počítače v síti. Ipconfig je výborný zdroj informací. Nebudu o něm nic víc psát, protože bych jenom opakoval nápovědu příkazového řádku.

#### Systeminfo

Zobrazí podrobné informace o počítači, operačním systému, informace o nainstalovaných programech včetně záplat operačního systému. Nemusím snad ani říkat, jak by to mohlo být pro útočníka přínosné.

#### FTP

Spustí se jednoduchý FTP klient příkazové řádky. Útočník může odesílat data na svůj FTP server a nebo naopak z něj data přijímat. Také ho můžete využít, když zrovna nemáte jiného FTP klienta.

|                      |  |
|----------------------|--|
| -v                   | Vypne zobrazování odpovědí vzdáleného serveru.   |
| -n                   | Vypne automatické přihlášení po úvodním připojení.   |
| -<br>s:nazev_souboru | Určuje textový soubor obsahující příkazy FTP, které budou automaticky spuštěny po spuštění služba FTP. |
| -A                   | Přihlášení jako anonymní uživatel.   |
| hostitel             | určí název nebo ip adresu vzdáleného hostitele ke kterému se má příkaz připojit.                       |

#### tasklist a taskkill

První z jmenovaných zobrazí spuštěné procesy (včetně antivirového programu, firewallu atd.) a druhý z jmenovaných dokáže proces ukončit.

#### tracert

Trasování dat směřujících k zadanému cíli. Podobně jako u pingu je k cíli zaslána žádost o odezvu (může být jiného druhu). Odpovídají však všechny počítače, kterými data procházejí, nikoli pouze cíl. Díky tomu je možné přesně určit trasu, kudy data procházela. Je velice užitečný při hledání problémových míst v síti.

#### Užitečná dávka

Toto je jednoduchá dávka, která po spuštění provede všechny tyto příkazy a uloží výsledek na disketu. Výsledky si můžete doma v klidu prohlídnout. Tuto dávku si můžete stáhnout [ZDE](#)

@echo off

```
net user administrator > a:netuser.txt
net localgroup > netloc.txt
net localgroup administrators > a:netlocad.txt
net session > a:netses.txt
net share > a:netsnar.txt
net statistics server > a:netstat.txt
ipconfig /all > a:ipcon.txt
tasklist > a:tasklist.txt
systeminfo > a:system.txt
net config server > a:netcons.txt
net config workstation > a:netconw.txt
net accounts > a:netac.txt
echo Hotovo!
```

#### **PŘÍKAZY CACLS, DISKCOMP, CONVERT**

## Cacls

Slouží k nastavení přístupových práv a výhodou příkazového řádku je, že můžete nastavit práva více uživatelům najednou. Existují tři módy přístupu:

|   |              |
|---|--------------|
| n | žádný        |
| r | čtení        |
| w | zápis        |
| c | změna        |
| f | úplné řízení |

U složek se mohou vyskytnout ještě tzv. příznaky práv: OI – Tato složka a soubory, CI – Tato složka a podsložky, IO – Vyjma aktuálního souboru/složky. Tyto příznaky se mohou i kombinovat. Např.: (OI)(CI) – Tato složka, podsložky a soubory. Pokud nebude příznak uveden týká se nastavení pouze aktuální složky. Pokud budete měnit přístupová práva tak vám mohu doporučit, abyste používali za každou změnou parametr /e. Slouží k tomu, aby původní přístupová práva zůstala zachována pro možnou obnovu.

|                            |   |
|----------------------------|---|
| cacls *.txt /t /g pnovak:r | Povolí uživateli pnovakovi čtení všech textových souborů s příponou txt v aktuální složce i v jejích podsložkách. |
| cacls *.exe /d karel       | Uživateli Karel zakáže přístup k souborům exe v aktuální složce. Ostatní přístupová práva budou odstraněna!       |

diskcomp

Porovnává obsah dvou disků (disket).

convert

Příkaz slouží pro převod mezi souborovými systémy. Pomocí tohoto příkazu nemůžete převádět aktuální disk.

Pokud se o to pokusíte bude vám nabídnuta možnost, jestli chcete tento disk převést po startu systému. Než budete převádět nějaký disk, tak doporučuji zálohovat data, protože pokud by vám při převádění došlo k přerušení (výpadek proudu, ...).

convert h: /fs:ntfs /v

Tento příkaz převede jednotku h: do systému NTFS a přepínač /vznamená, že se vám budou zobrazovány informace o průběhu. Pokud byste chtěli podrobnější informace opět vás odkazuji na nápovědu příkazového řádku.

### PROMĚNNÉ A JEJICH UŽITÍ V DÁVKÁCH

Hurá! Úspěšně jsme se prokousali k dávkám. Dávka by se dala specifikovat jako sled jednotlivých příkazů DOSu, které společně tvoří celek, který má za úkol provést určitý sled úloh nebo jenom jednu úlohu. Nejjednodušší dávky jsou vlastně jenom několik příkazů jdoucích za sebou uložených v souboru s příponou BAT. Ty složitější už obsahují podmínky a návěští, takže psaní dávek by se dalo přirovnat k jednoduchému programovacímu jazyku.

Jak a v čem tvořit dávky?

Teď budu předpokládat, že dávky už umíte a že jenom chcete, aby šli spustit. Nejjednodušším způsobem jak vytvořit dávku je napsat ji v Notepadu nebo v jiném textovém editoru (bez formátovacích znaků) a uložit soubor s příponou .bat Radši to ještě trochu upřesním. Pokud budete chtít vytvořit dávku v poznámkovém bloku zvolte možnost Soubor -> Uložit -> Uložit jako typ: Všechny soubory -> Název souboru: neco.bat. Kódování a ostatní možnosti můžete ponechat. Další možností, kde tvořit dávky jsou různé programátorské editory, které dokáží zvýraznit syntax dávkových souborů. Jedním takovým je například CodePad.

Proměnné

Proměnná je pojmenované místo v paměti, které může uchovávat určitou informaci (číslo, řetězec) se kterou pak můžeme dále pracovat. Před a za každou DOSovskou proměnnou musí být znak %. Zde je jednoduchý příklad dávky, která pouze vypíše na obrazovku "Ahoj" a počká na stisk klávesy.

@echo off

set pozdrav=Ahoj

echo %pozdrav%

pause

Deklarace proměnné

Proměnná se deklaruje příkazem SET. Tento příkaz je něco jako var v JavaScriptu. Nejdříve napíšete SET, následuje název proměnné, rovnítko a nakonec hodnota. Hodnota může být číslo, řetězec a nebo rovněž proměnná. Existují proměnné, které si nastavuje sám cmd (repektive Windows). Seznam těchto proměnných získáte příkazem SET bez parametrů. Pokud umíte trochu anglicky, tak z názvu zjistíte, co která proměnná dělá.

|      |                                 |
|------|---------------------------------|
| %cd% | Vypíše cestu k aktuální složce. |
|------|---------------------------------|

|                     |   |
|---------------------|---|
| %date%              | Obsahuje aktuální datum.  |
| %time%              | aktuální čas  |
| %random%            | Vygeneruje náhodné číslo.   |
| %username%          | Obsahuje jméno přihlášeného uživatele.  |
| %windir%,<br>%temp% | První obsahuje cestu k instalaci Windows což se dá snadno zneužít a druhá obsahuje cestu k adresáři temp. |
| %path%              | Obsahuje složky, které se prohledávají po zadání příkazu.   |

#### Číselné operace s proměnnými

Pomocí dávek můžete velké množství číselných operací. Zde jsou operace s operátory:

1. + (plus)
2. - (mínus)
3. /(děleno)
4. \* (násob)
5. % (modulo – zbytek po dělení)
6. ^ (XOR)
7. | (OR)
8. & (AND)
9. << (bitový posun vlevo)
10. >> (bitový posun vpravo)

Vždy, když budete dělat nějaké početní operace tak musíte dát o nich příkazovému řádku vědět napsáním příkazu set /a. Dále doporučuji dát celý početní výraz do uvozovek.

```
REM a = 2; b = 10
set /a "c=a+b"   výsledek je 12
set /a "c=a-b"   výsledek je -8
set /a "c=a*b"   výsledek je 20
set /a "c=(a+b*10)*(13+b)"   výsledek je 2346
```

Textové operace s proměnnými

Možná vás napadla otázka: "Když můžu do proměnných dávat řetězce tak s nima taky musí umět cmd pracovat?" Samozřejmě, že máte pravdu. Nejlépe se to vysvětlí na příkladu. Mám slovo: *LOKOMOTIVA*. Toto slovo je v proměnné %slovo%.

| PŘÍKAZ        | FUNKCE                               | VÝSLEDEK   |
|---------------|--------------------------------------|------------|
| %slovo:~2%    | Přeskočí první dva znaky             | KOMOTIVA   |
| %slovo:~-6%   | Bude použito posledních šest znaků   | MOTIVA     |
| %slovo:~0,-2% | Nebudou použity poslední dva znaky.  | LOKOMOTI   |
| %slovo:O=a%   | Všechny písmena O budou nahrazeny a. | LaKaMaTIVA |
| %slovo:O=%    | Všechny písmena O budou vypuštěna    | LKMTIVA    |

#### ZÁKLADNÍ PRVKY A PŘÍKLADY DÁVEK

pause

Příkaz pause pozastaví běh dávky a počká na stisk libovolné klávesy a poté bude pokračovat v provádění popřípadě se ukončí.

Více příkazů na jednom řádku

Pokud budete chtít dávat více příkazů na jeden řádek, budete muset použít znak & (ampersand). Na české klávesnici ho vytvoříte stiskem kláves pravý ALT + C. Ampersand nejvíce oceníte při tvorbě zástupců cmd (Místní nabídka -> Nový -> Zástupce). Tohoto zástupce musíte ještě přesunout do složky po spuštění. Může být užitečné pro pozdrav nebo upozornění.

```
cmd /c echo Dobrý den %username% & echo Dnes je %date% %time% & echo Nenavštěvujte prosím stránky s nelegální a nebo neetickou tematikou. & echo Vaše činnosti jsou monitorovány! echo Děkuji
```

## Správce sítě & pause

Příkazy můžeme také oddělit && (dvojitý ampersand). Následující příkaz bude vykonán pouze pokud skončil předchozí příkaz bez chyby. Dále můžete oddělovat příkazy symbolem || (dvojitá roura). Příkaz za rourami bude proveden pouze v případě, že předchozí příkaz skončil s chybou. Toto nalezne uplatnění při detekci nějakého souboru. Pokud je onen soubor nalezen bude se pokračovat a pokud ne dávka skončí. Později se ale naučíme elegantnější způsob.

```
@echo off
```

Před každou dávkou používám příkaz @echo off. Tím vypínám hlášení o průběhu dávky. Zkuste nějakou dávku napsat bez tohoto příkazu na začátku a poznáte proč je to užitečné.

Příklady dávek

Převede datum z formátu cm do formátu dd-mm-yyyy.

```
@echo off
```

```
set datum=%date%
```

```
set datum1=%date:~3%
```

```
set datum2=%date:.-=%
```

```
echo %datum2%
```

```
pause
```

Pomocí operací s texty, které cmd nabízí můžete také udělat i jednoduchou šifru.

| KODÉR   | DEKODÉR   |
|---|---|
| <pre>@echo off set text="toto je pro studijni ucely" set texta=%text:a=.% set textb=%texta:b=:% set textc=%textb:c=!% set textd=%textc:d=8% set texte=%textd:e=&gt;% set textf=%texte:f=&lt;% set textg=%textf:g=)% set texth=%textg:h=(% set texti=%texth:i=7% set textj=%texti:j=@% set textk=%textj:k=#% set textl=%textk:l={% set textm=%textl:m=}% set textn=%textm:n=\$% set texto=%textn:o=^% set textp=%texto:p=+% set textq=%textp:q=;% set textr=%textq:r=_% set texts=%textr:s=1% set textt=%texts:t=2% set textu=%textt:u=% set textv=%textu:v=?% set textw=%textv:w=&amp;% set textx=%textw:x=3% set texty=%textx:y= % set textz=%texty:z=-% echo %textz% &gt; C:vypis.txt pause</pre> | <pre>@echo off set text="2^2^ @&gt; +_^ 12/87@\$7 /!&gt;{ " set texta=%text:.=a% set textb=%texta:.=b% set textc=%textb:!=c% set textd=%textc:8=d% set texte=%textd:&gt;=e% set textf=%texte:&lt;=f% set textg=%textf:)=g% set texth=%textg:(=h% set texti=%texth:7=i% set textj=%texti:@=j% set textk=%textj:#=k% set textl=%textk:{=l% set textm=%textl:}=m% set textn=%textm:\$=n% set texto=%textn:^=o% set textp=%texto:+=p% set textq=%textp:;=q% set textr=%textq:_=r% set texts=%textr:1=s% set textt=%texts:2=t% set textu=%textt:/=u% set textv=%textu:?=v% set textw=%textv:&amp;=w% set textx=%textw:3=x% set texty=%textx: =y% set textz=%texty:=-z% echo %textz% &gt;C:vypis2.txt pause</pre> |

Tohle je jenom zlomek umění dávkových souborů MS-DOSu. V současné době je dost lidí, kteří se snaží (třeba i hádáním hesel) dostat do vašeho e-mailu. Tak ať se tam dostane a přečte si takto šifrovanou zprávu. Myslím, že bude překvapen. K tomu abyste si tuto zprávu přečetli musíte mít dekodér, který je to samé, co kodér, ale s prohozenými hodnotami. Používat kodér s těmito hodnotami už není zrovna bezpečné, a proto je lepší, když si vytvoříte svůj vlastní. Jak vydíte skoro nic za tím není. Jenom to chce trpělivost a čas. Bohužel je nutné pokud budete chtít soubor rozkódovat vložit text z něj do proměnné text v dekodéru. V předposledním řádku kodéru je místo, kam se uloží šifrovaný text a v dekodéru je zase místo, kam se uloží dešifrovaný text. Pokud bude již soubor s názvem vypis.txt nebo vypis2.txt exitovat tak bude nenávratně přepsán! Do tohoto kodéru nesmíte používat diakritiku a velká písmena. Když je tam napíšete, tak tam zůstanou tak jak jsou a nezašifrují se. Takhle by se třeba dala vytvořit i slavná Caesarova šifra (posunutí abecedy o 3 znaky) a nebo překladač z azbuky do latinky.

## PODMÍNKY A NÁVĚSTÍ

Podmínka se vykonává jenom jednou a na základě výsledku provede operaci a cyklus se vykonává dokud nějaká podmínka přetrvává. Tak třeba napíšu podmínku, když je a = 1 skoč. Dávka provede porovnání a pokud je a = 1 tak skočí. Kdybych chtěl stejnou věc udělat v cyklu tak by zápis zněl asi takto: dokud je a = 1 tak skákej.

Podmínky

Podmínky se zapisují příkazem IF. Podmínka se dá také negovat (obrátit) a to příkazem NOT. První konstrukci, kterou vás u dávek naučím je errorlevel. Používá se pokud budeme chtít vyhodnotit výsledek předchozího příkazu. Tak třeba budete chtít vytvořit nějaký soubor a cesta, kterou jste udali nebude nalezena. Potom skončí příkaz s chybou. Tady je malý příklad:

```
@echo off //vypnu režim echo
copy c:text.txt c:text2.txt //zkouším zkopírovat soubor
if errorlevel 1 goto chyba //pokud soubor neexistuje bude návratový
//kód jedna
echo Vse proběhlo v pořadku //pokud je návratový kód jiný vypiš toto
goto konec //přejdi na návěstí konec
:chyba //návěstí chyba
echo Nastala chyba!! //vypiš hlášení o chybě
:konec
pause //počkej na stisk klávesy
```

Rutinní záležitostí je porovnávání proměnných, ale ještě než ukážu jednoduchý příklad, tak bych vás chtěl ještě obeznámit s příkazem ELSE. Pokud vytvoříte podmínku (třeba jestli se a = 1 tak skoč) tak pokud neuspěje, program skončí, ale to se vám asi nelíbí a chcete, aby dávka něco udělala v případě neúspěchu a na to se používá příkaz ELSE. Teď malý příklad, který porovnává aktuální datum s 29. 4. 2006 a pokud je to dnes tak vypíše "Svátek má Robert" a v opačném případě, že neví kdo má svátek. Takto by šel vytvořit i kalendář, ale 365 dní se mi sem nechtělo psát.

```
@echo off
set datum2="%date%"
set datum="so 29.04.2006"
if %datum% EQU %datum2% (
echo Dnes je %date% a svatek ma Robert. Gratujuju!
pause
) else (
echo Dnes je %date% a nevím kdo má svátek
pause
)
```

Asi vás trochu zmátly závorky. Existují i dávky, které by fungovali i bez nich, ale tyto dávky by mohli mít jenom jeden příkaz, pokud by byla podmínka splněna. To sice dostačuje, pokud bychom chtěli jenom třeba skočit na návěstí, ale já to vždycky dělám se závorkama, a tak to nebudu měnit. Je nutné dodržovat i pořadí závorek. Dále jste si určitě všimli operátoru EQU a asi vás zaskočilo, že tam není operátor "=" jako v ostatních programovacích jazycích. Já taky nechápu jaký to má smysl, ale musíte se s tím smířit. Další operátory jsou následující:

|     |                                   |
|-----|-----------------------------------|
| EQU | rovná se (equal)                  |
| NEQ | nerovná se (not equal)            |
| LSS | menší než (less)                  |
| GTR | větší než (great)                 |
| LEQ | menší nebo roven (less or equal)  |
| GEQ | větší nebo roven (great or equal) |

Další velice užitečnou funkcí if, je ověření existence souboru. Tento příklad detekuje, jaký máte operační systém. Dávka hledá tapetu Rododendron.bmp, která je pouze ve Windows XP. Takto by se dal zjistit i antivirový program popřípadě záplaty operačního systému. Pozn.: Příklad předpokládá složku Windows na disku C:.

```
@echo off
if exist C:WindowsRododendron.bmp (
echo Na tomto pocitaci bezi Windows XP
pause
) else (
echo Na tomto pocitaci je jiny operacni system nez WinXP
pause
)
Návěstí
```

Určitě jste si všimli, že pokud v dávce nezadáte podmínku tak automaticky zpracuje následující příkaz. Ale občas se může stát, že potřebujete přejít do jiné oblasti kódu a právě začátek této oblasti se nazývá návěstí. Návěstí je

jeden řádek, který začíná dvojtečkou a za ním následuje název návěstí. Na návěstí se stejně jako ve vyšších programovacích jazycích skáče pomocí GOTO). GOTO je nepodmíněný skok a pokud z něj budete chtít udělat podmíněný, musíte ho zkombinovat s příkazem IF.

### KOMENTÁŘ DÁVKY, PŘÍKAZY CALL A AT

Komentář dávky

Jedinou možností, jak napsat v dávce komentář je příkaz REM. Tento postup je nepohodlný a nepružný, ale bohužel dávkové soubory jiný způsob nenabízejí. S příkazem REM musíte nakládat jako s normálním příkazem (úplně stejně jako s příkazem echo), tzn. že ho budete muset psát na zvláštní řádek. Příkaz REM se snažím užívat co nejméně, protože jeho používáním docílím jedině chauzu v dávce. Jeho využití je snad jedině v poznámkách v záhlaví nebo u komplikovaných operacích.

@echo off

```
REM -----  
REM Autor: Michal Blažek  
REM Název: mujweb.cz/www/dosms  
REM Činnost programu: snaha naučit uživatele s DOSem  
REM -----
```

pause

Příkaz call

Příkaz call volá dávku. Vysvětlení je následující: Představte si, že jste naprogramovali dávku, která má obrovskou velikost (pojmenuji jí davka.bat) a naprogramovali jste další dávku, jejíž funkce je založena na "davka.bat".

Uděláte to, že zkopírujete "davka.bat" s obrovskou velikostí do té menší a vznikne vám gigantická. A teď chcete vytvořit další dávku, která opět využívá dávku "davka.bat" a když ji tam opět nakopírujete, tak vám vzniknou dvě gigantické dávky, které mají skoro stejný obsah (lépe řečeno obě v sobě mají nakopírovanou dávku "davka.bat").

A pro tyto účely právě existuje příkaz call. Pro ty co nepochopili, co jsem se tu snažil názorně vysvětlit to teď zkusím říci po lopatě. Příkaz call spouští dávku, ke které jsme zadali cestu. Pokud patříte mezi typ lidí, kterým se říká hračkové, tak vás jistě potěší následující příklad a určitě ho budete dále rozvíjet. Příklad, který zde chci ukázat počká na stisk klávesy (1-3) a podle odpovídající klávesy vykoná příkaz a dávka se ukončí pokud stisknete klávesu 9 (tato dávka v podstatě vytváří DOS založený na DOSu). Příkaz 1 je jednoduchý dir, příkaz 2 je ver a příkaz 3 je zpráva echo obsahující aktuální datum a čas.

@echo off

```
REM -----  
REM Autor: Michal Blažek  
REM Název: dosnadosu.bat  
REM Činnost programu: Program počká na stisk klávesy 1-3 a následně ENTER  
REM a vykoná odpovídající příkaz. Program ukončíte klávesou 9  
REM -----
```

```
set prikaz1=1  
set prikaz2=2  
set prikaz3=3  
set prikaz4=9
```

echo Napoveda:

```
echo 1 - Vypise obsah aktualni slozky  
echo 2 - Vypise verzi systemu Windows  
echo 3 - Vypise aktualni datum a cas  
echo 9 - Ukonci program
```

set /p cti=DOS:

```
REM *** Přepínač /p znamená, že se bude číst z obrazovky **
```

```
if %cti% EQU %prikaz1% (call prikaz1.bat) else (goto dalsi1)
```

```
:dalsi1
```

```
if %cti% EQU %prikaz2% (call prikaz2.bat) else (goto dalsi2)
```

```
:dalsi2
```

```
if %cti% EQU %prikaz3% (call prikaz3.bat) else (goto konec)
```

```
:konec
```

echo Preji prijemny den

pause

at

Příkaz at je normálním příkazem DOSu a nepatří do kategorie if, else, goto, ... AT naplánuje provedení příkazu nebo dávky. Příkaz AT bez parametrů vypíše seznam naplánovaných úloh. Několik příkladů:

|                                  |  |
|----------------------------------|--|
| at 15:00<br>smaz.bat             | V 15:00 zavolá dávku smaz.bat a vymaže sama sebe ze seznamu naplánovaných úloh. Dávka proběhne na pozadí a vzhledem k rychlosti prováděných dávek si jí uživatel ani nevšimne. |
| at 15:00<br>every:po<br>smaz.bat | Dávka se vykoná každé pondělí v 1  |

### CYKLY

Pro cykly se v dávkách používá příkaz for a má zde velice rozsáhlou syntax a dají se s ním dělat užitečné věci.

Příkaz for – práce s množinou souborů

Velice užitečná je práce se soubory. Syntaxe je: for %x in (sada) do příkaz. Pokud příkaz použijete uvnitř dávky, je nutné proměnnou, která funguje jako řídicí, vždy uvozovat dvěma procenty! Příklad vypíše všechny soubory v aktuální složce a počká na stisk klávesy.

```
@echo off
for %%x in (*) do echo %%x
pause
```

Tento příklad by se dal lehce pozměnit, aby vypsal jenom textové soubory.

```
@echo off
for %%x in (*.txt) do echo %%x
pause
```

Pravděpodobně vás při čtení těchto řádek napadlo, že by šlo tímto způsobem získat název souboru a potom soubor zdvojit. Ano, opravdu to jde.

```
@echo off
set citac=0
for %%x in (*) do (
echo %%x
copy "%%x" "Kopie - %%x"
set /a citac+=1
REM ** Přičti pokaždé jedna k proměnné citac **
)
echo Zkopíroval jsem %citac% souborů
pause
```

Příkaz for /L – početní operace

Samozřejmě, že můžete pomocí for dělat také operace se soubory. Syntaxe je: for /I in (odkud, o kolik, kam) do příkaz . Myslím, že tento příklad mluví za vše.

```
@echo off
echo NECO Z MATEMATIKY
echo Císla se delí na suda a liché. Suda čísla jsou bezzbytku dělitelná
echo číslem dvě a liché ne.
echo Suda čísla od 1 do 20
for /L %%x in (2,2,20) do echo %%x
pause
cls
echo Liché čísla od 1 do 20
for /L %%x in (1,2,20) do echo %%x
pause
cls
echo Nasobilka 2:
for /L %%x in (0,2,20) do echo %%x
pause
cls
echo Nasobilka 7412:
for /L %%x in (0,7412,74120) do echo %%x
pause
```

Příka for /F – práce se soubory

Toto konečný příklad této lekce a věřím, že tuto funkci příkazu for v budoucnu využijete. Všechno budu vysvětlovat na příkladu. Nejdříve si vytvořte textový soubor adresy.txt, který bude vypadat následovně a potom vytvořte soubor generuj.bat, který je pod ním.

### ADRESY.TXT

```
#Toto je vzorový příklad
JMENO | BYDLISTE | POVOLANI
Karel Novak | Praha | ridic
```

ing. Jan Faust | Píbram | uředník  
#Jaroslav Mladý | Olomouc | učitel  
Bedřich Kropenka | Teplice | kuchař

### GENERUJ.BAT

```
@echo off
for /f "eol=# delims=| tokens=1,3" %%A in (adresy.txt) do (
    echo %%A, %%B
    REM ** Příkaz echo jsem u členu %%B nepsal, protože čárka značí, že se
        mají
    REM ** vypsát oba
    )
pause
```

Pro správnou funkčnost je třeba, aby dávka i soubor adresy.txt byly ve stejné složce. Myslím, že na souboru adresy.txt nemusím nic vysvětlovat, ale generuj.bat by si už několik vysvětlení zaslouhoval.

|            |   |
|------------|---|
| eol=#      | Tento příkaz znamená, že řádky začínající znakem # budou přeskočeny (samozřejmě si můžete zvolit vlastní znak).               |
| delims=    | Zde se určuje, jakým znakem jsou odděleny sloupce.  |
| tokens=1,3 | Tady si můžete určit s jakými sloupci chcete pracovat. Určitě jste si všimli, že sloupec 2 (Bydliště) není ve výsledku vidět. |

U tohoto příkazu jsem úmyslně psal strohé vysvětlení nebo vůbec žádné, protože zde jsem vás potřeboval donutit k praxi a ne abyste se něco našprtali.