



## Základné príkazy vo Windows CMD

Príkazový riadok **Windows CMD** (Command Prompt) je textové rozhranie systému Windows, ktoré umožňuje zadávať príkazy priamo operačnému systému. Umožňuje tak vykonávať rôzne operácie – od práce so súborami a priečkami až po správu sieťových nastavení či systémových služieb – priamo cez textové príkazy. Mnohé úlohy sa dajú v príkazovom riadku vykonať rýchlejšie než cez grafické rozhranie a niektoré pokročilé funkcie sú dostupné iba cez príkazy.

**Ako spustiť príkazový riadok:** Vo Windows ho môžete otvoriť napríklad tak, že stlačíte **Win + R**, napíšete `cmd` a potvrdíte Enter. Vo Windows 10/11 ho viete vyhľadať aj cez menu Štart (zadajte "cmd"). **Poznámka:** Pre niektoré príkazy je potrebné spustiť príkazový riadok ako správca (pravým klikom na ikonu *Príkazový riadok* a zvoliť *Spustiť ako správca*), inak nemusia fungovať alebo vypíšu chybové hlásenie o nedostatočných právach.

Každý príkaz sa zadáva za zobrazenú výzvu (prompt), ktorá typicky zobrazuje aktuálny priečinok (napr. `C:\>`). Po napísaní príkazu stlačte **Enter** na jeho vykonanie. Nižšie uvádzame podrobný manuál základných príkazov v príkazovom riadku Windows, rozdelený do dvoch častí: jednoduché príkazy pre bežných používateľov a pokročilé príkazy pre IT odborníkov. Ku každému príkazu je uvedený popis, syntax, príklad použitia a ukázkový výstup z konzoly. V prípade potreby sú doplnené aj poznámky pre používateľa a upozornenia na riziká.

### Príkazy pre bežných používateľov

Táto sekcia obsahuje **základné príkazy**, ktoré využijú bežní používatelia pri každodennej práci so súborami a priečkami v príkazovom riadku.

#### **dir** – zobrazenie obsahu priečinka

**Popis:** Príkaz `dir` vypíše obsah priečinka (adresára). Zobrazí zoznam súborov a podpriečinkov v aktuálnom priečinku, prípadne v zadanom priečinku alebo na disku. Štandardne zobrazuje názvy, veľkosti súborov a dátumy poslednej úpravy.

**Syntax:** `dir [<cesta>] [<parametre>]`

- `<cesta>` označuje priečinok (adresár), ktorého obsah chcete zobraziť. Ak cesta nie je uvedená, zobrazí sa obsah aktuálneho priečinka.
- Medzi užitočné parametre patrí napríklad `/P` (pause – zastaví výpis po zaplnení obrazovky a pokračuje po stlačení klávesy), `/w` (wide – zobrazí výpis v širokom formáte) alebo `/A` (attributes – filtruje podľa atribútov, napr. `dir /A:H` zobrazí skryté súbory).

**Príklad použitia:** Nasledujúci príkaz zobrazí obsah priečinka `C:\Windows` vrátane súborov a priečinkov v ňom:

```
C:\> dir C:\Windows
```

ToSho



## Ilustračný výstup:

```
Volume in drive C is OS
Volume Serial Number is 1234-ABCD
```

```
Directory of C:\Windows
```

```
01.04.2025  07:30    <DIR>          .
01.04.2025  07:30    <DIR>          ..
15.03.2025  10:20    <DIR>          System32
09.02.2025  15:45                1 234 notepad.exe
12.02.2025  08:00                12 345 explorer.exe
                2 File(s)          13 579 bytes
                3 Dir(s)    100 000 000 bytes free
```

**Poznámka:** Ak je výstup dlhší než jedna obrazovka, príkaz `dir` ho automaticky zastaví po naplnení konzoly. Stlačením ľubovoľnej klávesy zobrazíte ďalšiu časť výpisu (prípadne použite parameter `/P`). Príkaz `dir` nerozlišuje malé a veľké písmená v názvoch súborov.

---

## cd – zmena aktuálneho priečinka

**Popis:** Príkaz `cd` (change directory) zmení aktuálny pracovný priečinok na iný priečinok. Umožňuje tak prechádzať medzi priečinkami v rámci diskov. Môžete zadať absolútnu cestu (napr. `C:\Windows`) alebo relatívnu cestu vzhľadom na aktuálny priečinok (napr. názov podpriečinka alebo `..` pre nadradený priečinok).

**Syntax:** `cd [<disk:><cesta>]`

- Ak zadáte názov priečinka, `cd` vstúpi do tohto podpriečinka v aktuálnej ceste.
- `cd ..` prejde o úroveň vyššie (do nadradeného priečinka).
- Zadaním názvu disku (napr. `C:`) spolu s cestou môžete prejsť priamo na iný disk alebo priečinok na inom disku. (*Poznámka: samotné prepnutie diskov napr. na `D:` dosiahnete zadaním `D:` bez príkazu `cd.`*)

**Príklad použitia:** Zmena priečinka na `C:\Windows\System32`:

```
C:\Users\Peter> cd C:\Windows\System32
C:\Windows\System32>
```

**Ilustračný výstup:** Po zadaní príkazu sa zmenila cesta vo výzve príkazového riadka, čo vidíme na novom riadku `C:\Windows\System32>` – to znamená, že aktuálnym priečinkom je teraz `C:\Windows\System32`.

**Poznámka:** Ak cesta obsahuje medzery, môžete ju v príkaze uviesť v úvodzovkách (napr. `cd "C:\Program Files"`). Príkaz `cd` bez argumentov vypíše aktuálny priečinok.



## copy – kopírovanie súborov

**Popis:** Príkaz `copy` skopíruje jeden alebo viac súborov do iného umiestnenia. Môžete kopírovať súbory v rámci rovnakého priečinka, na iný priečinok alebo dokonca na iný disk. V základnej forme príkaz `copy` pracuje so súbormi; ak chcete kopírovať celé priečinky, musíte použiť rozšírené nástroje (napr. `xcopy` alebo `robocopy` – spomínané v poznámke nižšie).

**Syntax:** `copy <zdrojový_súbor> [<ďalšie_súbory>] <cieľ>`

- **<zdrojový\_súbor>** je názov (prípadne cesta) súboru, ktorý chcete kopírovať. Môžete zadať aj viac súborov naraz (napr. `*.txt` pre všetky TXT súbory v priečinku).
- **<cieľ>** je cieľová cesta alebo názov súboru. Môže to byť buď priečinok, do ktorého sa súbor skopíruje (s pôvodným názvom), alebo nový názov súboru.

**Príklad použitia:** Skopírovanie súboru `report.pdf` z aktuálneho priečinka do priečinka `D:\Zalohy` (zálohy):

```
C:\Dokumenty> copy report.pdf D:\Zalohy
```

### Ilustračný výstup:

```
1 file(s) copied.
```

Týmto príkazom sa vytvorila kópia súboru **report.pdf** v priečinku **D:\Zalohy**. Pôvodný súbor v `C:\Dokumenty` zostal nezmenený.

**Poznámka:** Príkaz `copy` sa pýta na potvrdenie, ak už cieľový súbor existuje (zobrazí *Overwrite <filename>? (Yes/No/All)*). Potvrďte stlačením **Y** (yes) alebo zrušte **N** (no). Prepínač `/Y` zruší výzvu na potvrdenie (vynúti prepísanie) a `/-Y` ju naopak vynúti zapne.

**Poznámka:** Na kopírovanie celých priečinkov vrátane obsahu môžete použiť príkaz `xcopy` (s parametrom `/E` alebo `/S`) alebo pokročilejší nástroj `robocopy`. Tieto však nepatria medzi základné príkazy, preto tu nie sú rozoberané detailne.

---

## del – mazanie (odstránenie) súborov

**Popis:** Príkaz `del` (delete) zmaže jeden alebo viac súborov. Na rozdiel od mazania v Prieskumníkovi sa pri použití tohto príkazu súbory **nepresúvajú do Koša**, ale okamžite a natrvalo sa odstránia. Preto je dôležité používať príkaz `del` opatrne. Môžete zadať konkrétny súbor, viac súborov oddelených medzerou, alebo použiť zástupné znaky (wildcards) `*` a `?` pre zmazanie viacerých súborov podľa vzoru.

**Syntax:** `del <súbor/y> [<parametre>]`

- Namiesto konkrétneho názvu súboru môžete použiť aj napr. `*.txt` pre všetky súbory s príponou `.txt`.

ToSho



- Medzi bežné parametre patrí **/P** (prompt – pred zmazaním každého súboru si vyžiada potvrdenie od používateľa), **/F** (force – vynúti zmazanie aj súborov s atributom len na čítanie) a **/S** (zmaže zodpovedajúce súbory aj v podpriechinkoch).

**Príklad použitia:** Zmazanie všetkých súborov s príponou **.TMP** v aktuálnom priečinku:

```
C:\Temp> del *.tmp
```

**Ilustračný výstup:**

```
C:\Temp\old1.tmp  
C:\Temp\old2.tmp
```

```
Are you sure (Y/N)?
```

V našom príklade príkaz `del` našiel súbory **old1.tmp** a **old2.tmp** a pýta sa na potvrdenie zmazania (táto výzva sa objaví, ak príkaz spúšťate interaktívne bez parametra **/Q**). Po stlačení **Y** sa uvedené súbory odstránia. Pokiaľ by ste použili prepínač **/Q** (quiet), k odstráneniu dôjde bez otázok.

**Upozornenie:** Súbory zmazané príkazom `del` nie je možné obnoviť z Koša. Buďte si istí, že odstraňujete správne súbory. Tiež pozor na použitie zástupných znakov (**\***), aby ste omylom nezmazali viac súborov, než ste zamýšľali.

---

## mkdir – vytvorenie nového priečinka

**Popis:** Príkaz `mkdir` (make directory), skrátene tiež `md`, vytvorí nový priečinok (adresár) na zadanom mieste. Môžete takto vytvoriť nový priečinok v aktuálnej ceste alebo uviesť plnú cestu, kde sa má priečinok vytvoriť. Ak cesta obsahuje neexistujúce podpriečinky, `mkdir` ich vytvorí tiež (napr. `mkdir C:\Novy\Adresar` vytvorí aj priečinok **Novy**, ak neexistuje, aj **Adresar**).

**Syntax:** `mkdir <cesta_priečinka>`  
(Príkaz `md` je ekvivalentný a používa sa rovnako.)

**Príklad použitia:** Vytvorenie priečinka s názvom **Projekty** v priečinku **D:\Dokumenty**:

```
C:\> mkdir D:\Dokumenty\Projekty
```

**Ilustračný výstup:**

```
D:\Dokumenty\Projekty vytvorený.
```

(Príkaz nevypisuje žiadny podrobný zoznam – len jednoduché potvrdenie alebo nič. V niektorých prípadoch sa nemusí zobrazit žiadna správa, iba sa vytvorí priečinok.)



**Poznámka:** Ak priečinok s daným názvom už existuje, zobrazí sa chybové hlásenie "*A subdirectory or file <name> already exists.*" (Podpriečinok alebo súbor už existuje) a nový priečinok sa nevytvorí. Názvy priečinkov môžete uvádzať aj v úvodzovkách, ak obsahujú medzery.

---

## rmdir – odstránenie priečinka

**Popis:** Príkaz `rmdir` (remove directory), skrátene `rd`, zmaže (odstráni) prázdny priečinok. Štandardne funguje iba na prázdne priečinky. Ak však chcete zmazať aj priečinok, ktorý **nie je prázdny** (obsahuje súbory alebo podpriečinky), môžete použiť parameter `/s` – tým sa vymaže celý obsah priečinka vrátane podpriečinkov (rekurzívne). Ide o rýchly spôsob, ako vymazať celú štruktúru priečinkov naraz.

**Syntax:** `rmdir <priečinok> [/S] [/Q]`

- `/S` – vymaže zadaný priečinok **vrátane** všetkých súborov a podpriečinkov, ktoré obsahuje (rekurzívne mazanie).
- `/Q` – quiet mód, nevypisuje potvrdzovacie výzvy (vhodné pri skriptoch alebo v kombinácii s `/S`, aby sa nepotvrdzovalo mazanie každého súboru).

**Príklad použitia:** Odstránenie prázdneho priečinka **Test** v aktuálnom adresári:

```
C:\Work> rmdir Test
```

**Ilustračný výstup:** (Príkaz `rmdir` pri úspešnom zmazení spravidla nevypíše nič a vráti iba nový príkazový riadok. Ak by priečinok nebol prázdny, zobrazí sa chybové hlásenie.)

Napríklad, ak by **Test** obsahoval súbory, príkaz by vypísal:

```
The directory is not empty.
```

V takom prípade by ste museli použiť `rmdir /S Test` a potvrdiť mazanie celého obsahu priečinka.

**Upozornenie:** Použitie prepínača `/s` zmaže celý priečinok **vrátane všetkých súborov a podpriečinkov** bez možnosti obnovy. Pri použití s `/Q` sa nebude pýtať na potvrdenie. Buďte veľmi opatrní, aby ste nezmazali nesprávny priečinok.

## cls – vyčistenie obrazovky konzoly

**Popis:** Príkaz `cls` (clear screen) vymaže aktuálny obsah obrazovky konzoly. Odstráni všetky predtým zobrazené príkazy a výstupy a zobrazí príkazový riadok (prompt) navrchu konzoly. Tento príkaz je užitočný na prehľadnenie obrazovky, ak je zaplnená predchádzajúcimi výstupmi.

**Syntax:** `cls` (príkaz nemá žiadne parametre ani argumenty)

**Príklad použitia:**

```
C:\> cls
```

ToSho



**Ilustračný výstup:** Po vykonaní príkazu `cls` sa konzola vyčistí a zostane zobrazený iba prázdny riadok s výzvou (napr. `C:\>`). (Na obrazovke tak neuvidíte predchádzajúci text.)

---

## exit – ukončenie príkazového riadka

**Popis:** Príkaz `exit` zavrie okno príkazového riadku alebo ukončí aktuálnu **CMD** session. Ak pracujete v interaktívnom príkazovom riadku, tento príkaz zatvorí celé okno. Pri použití v skripte ukončí spracovanie skriptu (a vráti sa do volajúceho procesu, ak existuje).

**Syntax:** `exit` (príkaz nemá povinné parametre; voliteľne môžete uviesť číselný kód ukončenia, napr. `exit 0`)

### Príklad použitia:

```
C:\> exit
```

**Ilustračný výstup:** Okno príkazového riadku sa zavrie (nie je k dispozícii ďalší výstup, keďže konzola bola ukončená). Ak ste príkaz použili v skripte alebo v rámci iného programu, riadenie sa vráti naspäť volajúcemu procesu.

**Poznámka:** Ak príkaz `exit` spustíte omylom, musíte príkazový riadok otvoriť znova. Pri práci v skripte môžete k príkazu `exit` pripojiť aj číselný výstupný kód (napr. `exit 1`), ktorý môže volajúci proces využiť na zistenie dôvodu alebo výsledku ukončenia.

---

## Príkazy pre IT odborníkov (napríklad pre firemné nasadenie)

V tejto časti sú popísané **pokročilé príkazy**, ktoré sa často používajú pri správe systému, sietí a domén v profesionálnom prostredí. Tieto príkazy vyžadujú hlbšie technické znalosti a často aj administrátorské oprávnenia. Každý príkaz obsahuje detailný popis, syntax, príklad a ukážku výstupu z praxe.

### ipconfig – zobrazenie sieťových nastavení

**Popis:** Príkaz `ipconfig` zobrazí aktuálnu konfiguráciu sieťových adaptérov počítača. Výstup obsahuje IP adresu, masku podsiete, predvolenú bránu, DNS server a ďalšie informácie pre každé sieťové rozhranie. Používa sa pri diagnostike sieťových problémov a na overenie nastavení. S parametrom `/all` zobrazí podrobné informácie vrátane fyzickej adresy (MAC), DHCP stavu a pod. `ipconfig` umožňuje aj vykonávať určité akcie – napríklad `ipconfig /release` uvoľní (release) aktuálnu IP adresu získanú od DHCP a `ipconfig /renew` obnoví (vyžiada) novú IP adresu od DHCP servera. Tieto akcie sa používajú pri riešení problémov s pripojením.



**Syntax:** `ipconfig [<parametre>]`

- Bez parametrov zobrazí základný prehľad IP konfigurácie aktívnych adaptérov.
- `/all` – podrobné informácie o každom adaptéri (vrátane skrytých alebo neaktívnych).
- `/release` – uvoľní aktuálne DHCP konfigurácie (nastaví IP adresy na 0.0.0.0). (*Vyžaduje administrátorské oprávnenia.*)
- `/renew` – obnoví DHCP konfigurácie (získa novú IP). (*Vyžaduje administrátora.*)
- `/flushdns` – vymaže cache DNS resolvera (pomáha pri riešení nesprávnych záznamov DNS).

**Príklad použitia:** Zobrazenie základnej IP konfigurácie (aktívnych sieťových adaptérov):

```
C:\> ipconfig
```

## Ilustračný výstup:

```
Windows IP Configuration
```

```
Ethernet adapter Ethernet:
```

```
Connection-specific DNS Suffix . : firma.local
IPv4 Address. . . . . : 192.168.10.102
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.10.1
```

```
Wireless LAN adapter Wi-Fi:
```

```
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
```

V tomto výstupe vidíme, že počítač má napríklad jednu aktívnu *Ethernet* (káblOVú) sieťovku s IP adresou **192.168.10.102** a jednu Wi-Fi, ktorá nie je pripojená (*Media disconnected*).

**Poznámka:** Príkazy `ipconfig /release` a `ipconfig /renew` je potrebné spúšťať v príkazovom riadku spustenom ako správca, inak zmeny IP adresy neprebehnú. Tieto príkazy zvyčajne používa správca siete.

**Poznámka:** Ak potrebujete zistiť iba adresu IP, môžete použiť aj `ipconfig` s filtráciou cez `find` príkaz, napríklad: `ipconfig | find "IPv4"` vypíše len riadky obsahujúce "IPv4".

---

## ping – test sieťového pripojenia

**Popis:** Príkaz `ping` slúži na otestovanie dosiahnuteľnosti iného sieťového zariadenia alebo servera. Posiela ICMP pakety (echo request) na zadanú IP adresu alebo doménové meno a meria čas, za ktorý príde odpoveď (echo reply). Výsledkom je informácia, či je cieľ dostupný, koľko paketov sa stratilo a aký bol čas odozvy (latencia) v milisekundách. `ping` sa používa na základnú diagnostiku siete – overenie, či je konkrétny server alebo IP adresa dostupná a ako rýchlo reaguje.

**Syntax:** `ping <cieľ> [<parametre>]`

- **<cieľ>** môže byť doménové meno (napr. `example.com`) alebo IP adresa (napr. `192.168.1.1`).

ToSho



- Štandardne `ping` pošle 4 testovacie pakety. Parametrom `-t` môžete spustiť neobmedzený ping (beží dovtedy, kým ho neprerušíte klávesovou skratkou **Ctrl+C**). Parameter `-n <počet>` umožňuje zadať konkrétny počet paketov.
- Ďalšie užitočné parametre: `-l <veľkosť>` (size – veľkosť paketu v bajtoch), `-4` alebo `-6` (vynútiť použitie IPv4 alebo IPv6).

**Príklad použitia:** Otestovanie odozvy servera `example.com`:

```
C:\> ping example.com
```

## Ilustračný výstup:

```
Pinging example.com [93.184.216.34] with 32 bytes of data:
Reply from 93.184.216.34: bytes=32 time=15ms TTL=56
Reply from 93.184.216.34: bytes=32 time=16ms TTL=56
Reply from 93.184.216.34: bytes=32 time=15ms TTL=56
Reply from 93.184.216.34: bytes=32 time=14ms TTL=56

Ping statistics for 93.184.216.34:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 16ms, Average = 15ms
```

Vo výstupe vidíme štyri odpovede (*Reply from ...*) s časom odozvy približne 15ms a štatistiku, ktorá ukazuje 0% stratu paketov. To znamená, že server **example.com** je dostupný a reaguje v danom čase.

**Poznámka:** Ak `ping` nedostane žiadnu odpoveď, vypíše hlásenia ako *"Request timed out."* (vypršal časový limit požiadavky) alebo *"Host unreachable"*. To naznačuje, že cieľ nie je dostupný (buď je offline, alebo sú pakety blokované).

**Poznámka:** Príkaz `ping` môže byť blokovaný firewallom, najmä v podnikových sieťach, takže nedostupnosť pingu nemusí vždy znamenať nedostupný server – môže to byť zámerné blokovanie ICMP.

---

## netstat – výpis sieťových spojení a portov

**Popis:** Príkaz `netstat` zobrazuje aktuálne sieťové spojenia a otvorené porty na počítači. Umožňuje zistiť, ktoré porty počúvajú (LISTENING) a aké aktívne spojenia existujú (ESTABLISHED) vrátane informácií o vzdialených adresách. Je užitočný pre diagnostiku sieťovej prevádzky a bezpečnosti (napríklad zistiť, či na počítači nebeží nečakaná sieťová komunikácia). S parametrami dokáže `netstat` zobrazit' aj protokol (TCP/UDP), štatistiky a PID (identifikátor procesu) pre každé spojenie, aby ste vedeli priradiť spojenie ku konkrétnemu procesu.

**Syntax:** `netstat [<parametre>]`

- Bez parametrov zobrazí základný zoznam aktívnych TCP spojení.
- `-a` – zobrazí *všetky* spojenia a počúvajúce porty (aj tie, ktoré nekomunikujú).
- `-n` – zobrazí adresy a porty *číselne* (nevykonáva DNS ani názvové preklady – rýchlejšie a prehľadnejšie).

ToSho



- **-o** – zobrazí PID (Process ID) každého spojenia, čo umožní identifikovať proces v systéme (v kombinácii s **-a** často používané ako `netstat -ano`).
- **-b** – (vyžaduje admin) zobrazí aj názov spustiteľného súboru (procesu) pre každé spojenie.

**Príklad použitia:** Zobrazenie všetkých aktívnych spojení, vrátane otvorených portov, s číselnými adresami a PID procesov:

```
C:\> netstat -a -n -o
```

## Ilustračný výstup:

Proto	Local Address	Foreign Address	State	PID
TCP	192.168.10.102:49712	93.184.216.34:443	ESTABLISHED	3504
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	980
TCP	[::]:135	[::]:0	LISTENING	980
UDP	0.0.0.0:1900	*.*		1236

V uvedenom výstupe prvý riadok znamená, že lokálna adresa **192.168.10.102** na porte **49712** má otvorené spojenie na vzdialenú adresu **93.184.216.34** na porte **443** (HTTPS) a spojenie je v stave ESTABLISHED. PID procesu, ktorý toto spojenie drží, je **3504** (podľa PID môžete zistiť názov procesu napr. cez Správcu úloh alebo príkaz `tasklist`). Druhý a tretí riadok zobrazujú, že port 135 (typicky RPC) počúva na všetkých rozhraniach (IPv4 aj IPv6) a patrí procesu s PID 980. Štvrtý riadok je príklad UDP portu 1900, ktorý počúva (UDP spojenia nemajú stav) a patrí procesu s PID 1236.

**Poznámka:** Výpis `netstat` môže byť dosť rozsiahly na systémoch, ktoré majú veľa otvorených spojení (napr. webový prehliadač môže otvárať desiatky spojení). Pre prehľadnosť je dobré použiť parametre `-n` (vypne preklad názvov, čím urýchli výstup) a prípadne filtrovať výsledky.

**Poznámka:** Ak potrebujete zistiť, aký proces beží na konkrétnom porte, zistíte si PID cez `netstat -ano | find "<port>"` a potom použijete príkaz `tasklist /FI "PID eq <pid>"` na zistenie názvu procesu.

---

## tasklist – zoznam bežiacich procesov

**Popis:** Príkaz `tasklist` vypíše zoznam všetkých bežiacich procesov (úloh) v systéme Windows. Ide o textový ekvivalent Správcu úloh. Výstup obsahuje názov spustiteľného súboru procesu, jeho PID (identifikačné číslo procesu), názov relácie (Session) a využitie pamäte. Pomocou `tasklist` v kombinácii s filtermi môžete získať informácie o konkrétnom procese alebo dokonca o procesoch na vzdialenom počítači. Je to užitočné pre správcov na získanie prehľadu o tom, čo na systéme beží.

**Syntax:** `tasklist [<parametre>]`

- Bez parametrov vypíše všetky procesy aktuálne bežiace na lokálnom počítači.
- **/FI "<filter>"** – umožňuje filtrovať výsledky podľa zadaných kritérií, napr. podľa názvu obrazu (IMAGENAME), PID, používateľa atď. Napr. `tasklist /FI "USERNAME ne SYSTEM"` zobrazí procesy, ktoré nebežia pod účtom SYSTEM.
- **/svc** – zobrazí pre každý proces aj služby (service) bežiace v ňom (platí najmä pre procesy typu `svchost.exe`, ktoré hostujú viaceré služby).

ToSho



- `/v` – verbose, zobrazí podrobnejšie informácie o procesoch (napr. okno, ktoré proces otvoril, používateľa, atď.).
- `/m` – modules, zobrazí zoznam DLL modulov načítaných každým procesom. (Tento výstup môže byť veľmi dlhý.)

**Príklad použitia:** Zobrazenie základného zoznamu procesov na lokálnom počítači:

```
C:\> tasklist
```

## Ilustračný výstup:

Image Name	PID	Session Name	Mem Usage
System Idle Process	0	Services	8 K
System	4	Services	72 K
smss.exe	228	Services	1 200 K
csrss.exe	412	Services	4 800 K
wininit.exe	500	Services	3 600 K
explorer.exe	3456	Console	75 000 K
notepad.exe	8200	Console	5 000 K

Tento (skrátenejší) výstup zobrazuje niekoľko procesov: systémové procesy (System Idle Process, System), procesy Windows (smss.exe, csrss.exe, ...), a používateľské aplikácie ako **explorer.exe** (Prieskumník Windows) a **notepad.exe** (Poznámkový blok) s ich PID a využitím pamäte.

**Poznámka:** V zozname sa v stĺpci *Session Name* rozlišuje, či proces beží ako služba (*Services*) alebo v interaktívnom používateľskom konte (*Console* pre aktuálne prihláseného používateľa atď.).

**Poznámka:** Ak potrebujete hľadať konkrétny proces, môžete použiť `tasklist | find "názov"` alebo spomínaný parameter `/FI`. Napríklad `tasklist /FI "IMAGENAME eq notepad.exe"` zobrazí informácie len o procese notepad.exe, ak beží.

---

## taskkill – ukončenie bežiaceho procesu

**Popis:** Príkaz `taskkill` umožňuje ukončiť (zabiť) bežiaci proces podľa jeho názvu alebo PID. Je to nástroj na príkazovom riadku ekvivalentný k ukončeniu úlohy cez Správcu úloh. Umožňuje nútené zatvoriť zamrznuté programy alebo skriptovo ukončiť nepotrebné procesy. Ak proces nechce skončiť bežným spôsobom, je tu možnosť *vynúteného* ukončenia.

**Syntax:** `taskkill [<parametre>]`

- `/IM <názov_procesu>` – (Image Name) ukončí proces podľa názvu spustiteľného súboru, napr. `taskkill /IM notepad.exe`. Môžete použiť aj zástupné znaky v názve, napr. `taskkill /IM notepad*.exe`.
- `/PID <číslo_PID>` – ukončí proces podľa konkrétneho PID (identifikátora). PID získate napr. príkazom `tasklist`.

ToSho



- **/F** – (force) vynúti ukončenie procesu. Ak sa nepoužije **/F**, príkaz sa pokúsi ukončiť proces štandardne. S **/F** proces ukončíte aj v prípade, že neodpovedá (no force môže spôsobiť stratu neuložených dát).
- **/T** – ukončí aj všetky podriadené procesy daného procesu (terminate tree).

**Príklad použitia:** Ukončenie procesu Poznámkového bloku (**notepad.exe**) podľa názvu s vynútením:

```
C:\> taskkill /F /IM notepad.exe
```

## Ilustračný výstup:

```
SUCCESS: The process "notepad.exe" with PID 8200 has been terminated.
```

Týmto príkazom bol proces **notepad.exe** (s PID 8200) ukončený. Ak by bežalo viac inštancií notepad.exe, príkaz ukončí všetky (pretože sme použili **/IM** s názvom). Alternatívne by sme mohli ukončiť konkrétnu inštanciu pomocou jej PID, napríklad `taskkill /PID 8200`.

**Upozornenie:** Pri použití `taskkill /F` sa proces ukončí okamžite, čo môže spôsobiť stratu neuložených údajov v danom programe. Vždy sa presvedčte, že ukončujete správny proces. Niektoré systémové procesy nemožno ukončiť vôbec, alebo ak ich ukončíte, môže to spôsobiť nestabilitu systému.

**Poznámka:** Na ukončenie vzdialeného procesu na inom počítači v doméne môžete použiť parametre **/S** (server), **/U** (užívateľ) a **/P** (heslo) pre prihlásenie na daný stroj, pokiaľ máte potrebné oprávnenia.

---

## sc – správa systémových služieb

**Popis:** Príkaz `sc` (Service Control) je pokročilý nástroj na správu služieb v systéme Windows. Umožňuje z príkazového riadku vykonávať operácie ako vytváranie nových služieb, zmenu ich nastavení, spúšťanie, zastavovanie a zisťovanie stavu služieb. Tento príkaz je často využívaný v skriptoch na automatizáciu inštalácie a konfigurácie služieb na viacerých počítačoch. Jeho použitie je určené pre skúsených používateľov a administrátorov, pretože nesprávna manipulácia so službami môže ovplyvniť stabilitu systému.

**Syntax:** `sc <operácia> <názov_služby> [<parametre>]`

Najčastejšie operácie:

- **query** – zisťuje stav služby alebo zoznam služieb. Napr. `sc query Spooler` zobrazí stav služby Spooler (tlačový manažér). Bez špecifikácie služby (`sc query`) vypíše zoznam všetkých bežiacich služieb.
- **start** – spustí zadanú službu (ak nie je spustená). Napr. `sc start Spooler`.
- **stop** – zastaví bežiacu službu. Napr. `sc stop Spooler`.
- **config** – zmení konfiguráciu služby (napr. štartovací typ). Toto je pokročilé použitie, napr. `sc config Spooler start= disabled` by nastavilo, že služba Spooler sa nespustí pri štarte (disable). (Medzi `start=` a hodnotou musí byť medzera.)
- **create** – vytvorí novú službu (vyžaduje presnú špecifikáciu binárky a parametrov).
- **delete** – odstráni existujúcu službu.

ToSho



## Príklad použitia: Zistenie aktuálneho stavu služby **Print Spooler** (Správca tlače):

```
C:\> sc query Spooler
```

### Ilustračný výstup:

```
SERVICE_NAME: Spooler
        TYPE                : 110  WIN32_OWN_PROCESS
        STATE                 : 4    RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE       : 0    (0x0)
        SERVICE_EXIT_CODE   : 0    (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x0
```

Z výpisu vidíme, že služba **Spooler** je momentálne v stave **RUNNING** (beží). `SERVICE_NAME` uvádza systémový názov služby. Štátus **STOPPABLE, NOT\_PAUSABLE, ...** indikuje, že službu je možné zastaviť, nedá sa pauznúť atď. Tento druh detailného výpisu pomáha administrátorom zistiť, či služba beží a ak nie, tak prečo (napr. chybové kódy ak zlyhala).

**Poznámka:** Väčšina operácií príkazu `sc` vyžaduje spustiť príkazový riadok ako správca, inak dôjde k odmietnutiu prístupu.

**Poznámka:** Namiesto príkazu `sc` existujú aj jednoduchšie príkazy `net start` a `net stop` na spúšťanie a zastavovanie služieb, ktoré však neumožňujú takú širokú funkcionálnosť (napr. `net start Spooler` spustí službu `Spooler`).

**Upozornenie:** Pri manipulácii so službami buďte opatrní. Napríklad zastavenie kritických služieb (siet', bezpečnostné služby atď.) môže spôsobiť problémy v systéme. Vytváranie a mazanie služieb zase zasahuje do registru a systémových nastavení.

---

## net user – správa používateľských účtov

**Popis:** Príkaz `net user` slúži na správu používateľských účtov v prostredí Windows (lokálne na počítači alebo v doméne). Dokáže zobrazit' zoznam používateľov, vytvorit' nového používateľa, zmenit' heslo existujúceho používateľa alebo upraviť určité vlastnosti účtu. Na lokálnom počítači je užitočný napríklad na rýchle vytvorenie nového administrátorského účtu alebo aktiváciu skrytého účtu administrátora.

**Syntax:** `net user [<užívateľské_meno> [<heslo>] [<parametre>]]`

- Zadaním samotného príkazu `net user` (bez ďalších argumentov) vypíšete zoznam všetkých používateľských účtov v systéme.
- Pre vytvorenie nového používateľa použijete `net user <meno> <heslo> /add`. Napr. `net user NovyUzivatel heslo123 /add` vytvorí účet s menom *NovyUzivatel*.
- Pre nastavenie alebo zmenu hesla existujúceho používateľa: `net user <meno> <nové_heslo>` (napr. `net user NovyUzivatel novéHeslo456`).

ToSho



- Pre ďalšie nastavenia môžete použiť parametre ako `/active:yes|no` (aktivovať/deaktivovať účet), `/expires:<dátum>` (nastaviť expiráciu účtu), `/passwordchg:yes|no` (povoliť/zakázať používateľovi meniť si heslo) atď.
- `/delete` parameter zmaže účet: `net user <meno> /delete`.

**Príklad použitia:** Zobrazenie zoznamu používateľských účtov na lokálnom počítači:

```
C:\> net user
```

## Ilustračný výstup:

```
User accounts for \\MY-PC:
-----
Administrator      Guest
John                Mary
The command completed successfully.
```

Tento výstup zobrazuje, že na počítači **MY-PC** existujú účty: vstavaný **Administrator**, **Guest** (host'ovský účet) a napríklad používateľské účty **John** a **Mary**.

Ako ďalší príklad, na vytvorenie nového používateľa **TestUser** s heslom *heslo123*:

```
C:\> net user TestUser heslo123 /add
```

## Ilustračný výstup:

```
The command completed successfully.
```

Týmto príkazom sa vytvoril nový účet *TestUser*. (Predvolene bude vytvorený ako štandardný používateľ bez administrátorských práv.) Administrátor môže následne pridať tohto používateľa do skupiny Administrators, ak je to potrebné, príkazom `net localgroup Administrators TestUser /add`.

**Poznámka:** Pre zobrazenie detailov o konkrétnom používateľovi môžete použiť `net user <meno>` (bez ďalších parametrov). Napríklad `net user John` vypíše podrobnosti o používateľovi John (kedy naposledy menil heslo, či musí heslo meniť, členstvo v skupinách, atď.).

**Upozornenie:** Vytváranie a mazanie používateľských účtov či zmena hesiel vyžaduje administrátorské oprávnenia. Buďte opatrní najmä pri mazaní účtov (`/delete`), pretože tým trvalo odstránite daného používateľa a jeho profilové dáta (dokumenty) z počítača.

---

## net share – správa zdieľaných priečinkov

**Popis:** Príkaz `net share` slúži na vytváranie, zrušenie a prehľad zdieľaných priečinkov (sietí) na lokálnom počítači. Umožňuje napríklad nasdielať priečinok pre prístup cez sieť alebo zistiť, ktoré priečinky sú aktuálne zdieľané. V domácom aj firemnom prostredí sa zdieľanie priečinkov využíva na sprístupnenie súborov ostatným používateľom v sieti. Príkaz `net share` poskytuje textové rozhranie na túto funkcionality, alternatívnu k nastavovaniu zdieľania cez Prieskumníka.

ToSho



## Syntax:

- Zobrazit' všetky zdieľané priečinky: `net share`
- Vytvorit' nové zdieľanie: `net share <názov>=<cesta> [<parametre>]` (napr. `net share Dokumenty=D:\Dokumenty\PreZdieľanie /grant:everyone,full`)
- Zrušiť (odstrániť) zdieľanie: `net share <názov> /delete`

## Voliteľné parametre pri vytváraní zdieľania:

- `/users:<počet>` – obmedzí počet súčasných používateľov, ktorí môžu zdieľanie použiť.
- `/grant:<užívateľ>,<práva>` – nastaví oprávnenia pre konkrétneho užívateľa/skupinu. Napr. `everyone,read` alebo `everyone,full`.
- `/remark:"<text>"` – popis zdieľania (komentár).  
(Ak nie sú zadané oprávnenia, predvolene má prístup skupina *Everyone* s oprávnením na čítanie.)

## Príklad použitia: Zobrazenie existujúcich zdieľaní na počítači:

```
C:\> net share
```

## Ilustračný výstup:

```
plaintext
CopyEdit
Share name          Resource          Remark
-----
C$                  C:\              Default share
IPC$                C:\              Remote IPC
Public              C:\Users\Public  Zdieľaná verejná zložka
SharedDocs          D:\Dokumenty\Spoločné
                    Dokumenty pre zdieľanie
The command completed successfully.
```

Tento (hypotetický) výstup zobrazuje zoznam zdieľaných priečinkov. **C\$** a **IPC\$** sú predvolené systémové zdieľania (administrátorské a IPC), **Public** je zdieľaný priečinkov verejných dokumentov a **SharedDocs** je napríklad ručne vytvorené zdieľanie priečinka `D:\Dokumenty\Spoločné`. Stĺpec *Resource* ukazuje cestu priečinka na disku a *Remark* prípadný popis.

Ak chceme vytvorit' nové zdieľanie, napríklad nasdieľať priečinkov **D:\Projekty\Test** pod názvom **TestShare**:

```
C:\> net share TestShare=D:\Projekty\Test /grant:Users,read
```

## Ilustračný výstup:

```
TestShare was shared successfully.
```

Týmto sme vytvorili sieťové zdieľanie *TestShare*, ktoré odkazuje na priečinkov **D:\Projekty\Test**. Všetci používatelia v skupine *Users* majú právo čítania.

**Poznámka:** Zdieľanie priečinkov cez príkazový riadok vyžaduje administrátorské oprávnenia.

**Poznámka:** Predvolené zdieľania s názvom končiacim \$ (napr. C\$, IPC\$) sú skryté administrátorské zdieľania. V zozname ich vidí len administrátor.

ToSho



**Upozornenie:** Pri zdieľaní priečinka buďte opatrní, komu udeľujete prístup. Napríklad `/grant:everyone,full` sprístupní priečinok úplne každému v sieti s plnými právami (čítanie aj zápis), čo môže predstavovať bezpečnostné riziko v nedôveryhodnej sieti.

---

## systeminfo – informácie o systéme

**Popis:** Príkaz `systeminfo` vypíše detailné informácie o operačnom systéme a hardvéri počítača. Patria sem údaje ako verzia a edícia Windows, posledný dátum inštalácie (boot time), verzie BIOSu/UEFI, procesor, veľkosť pamäte (RAM), názov počítača, konfigurácia sieťovej karty, atď. Tento príkaz je užitočný pre administrátorov na rýchly prehľad o systéme, najmä pri správe viacerých počítačov.

**Syntax:** `systeminfo` (príkaz nemá povinné parametre; existujú parametre pre napr. vzdialené získanie info `/S /U /P` pre iný systém, ale základné použitie je bez parametrov)

### Príklad použitia:

```
C:\> systeminfo
```

### Ilustračný výstup:

```
plaintext
CopyEdit
Host Name:                MY-PC
OS Name:                  Microsoft Windows 10 Pro
OS Version:               10.0.19044 N/A Build 19044
OS Manufacturer:        Microsoft Corporation
System Manufacturer:     Dell Inc.
System Model:             Latitude 5580
Processor(s):             1 Processor(s) Installed.
                          [01]: Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz
BIOS Version:             Dell Inc. 1.15.0, 12/05/2020
Windows Directory:       C:\Windows
System Directory:         C:\Windows\system32
Total Physical Memory:    8 192 MB
Available Physical Memory: 3 456 MB
Virtual Memory: Max Size: 16 384 MB
Virtual Memory: Available: 10 000 MB
Virtual Memory: In Use:   6 384 MB
```

(Ukážka je skrátená – príkaz `systeminfo` vypisuje ešte ďalšie riadky týkajúce sa sieťových nastavení a aktualizácií systému.) Z uvedeného výstupu však získame podstatné informácie: názov počítača, verziu OS, výrobcu a model počítača, procesor, verziu BIOS, inštalčné priečinky Windows a informácie o pamäti.

**Poznámka:** Niektoré riadky výstupu (napríklad informácie o aktualizáciách – Hotfix(s) alebo konfigurácii sieťových kariet) môžu byť vypustené, ak príkaz nemá prístup k týmto údajom. Uistite sa, že príkaz spúšťate s dostatočnými oprávneniami (bežný užívateľ zvyčajne vidí všetko potrebné; v doméne, ak spúšťate na vzdialenom stroji cez `/S`, potrebujete admin oprávnenia na cieľ).

**Poznámka:** Príkaz `systeminfo` môže chvíľu trvať, kým zozbiera všetky informácie, najmä ak sieťové adaptéry hľadajú DHCP alebo ak je systém zaťažovaný.

ToSho



## chkdsk – kontrola disku

**Popis:** Príkaz `chkdsk` (Check Disk) slúži na kontrolu súborového systému a povrchu disku. Vyhľadáva chyby na disku (napr. nesprávne odkazy súborov, poškodené segmenty MFT atď.) a dokáže ich opraviť. Tiež môže skontrolovať fyzický povrch disku na vadné sektory. Najčastejšie sa používa s parametrom `/F` (fix), ktorý chyby automaticky opraví, a `/R` (recover), ktorý vyhľadá aj chybné sektory a pokúsi sa z nich obnoviť čitateľné údaje (tiež implicitne zahrňuje `/F`). `chkdsk` je užitočný pri podozrení na problémy s diskom alebo po nesprávnom vypnutí počítača.

**Syntax:** `chkdsk [<jednotka>:] [<parametre>]`

- Ak zadáte iba `chkdsk` bez parametrov, príkaz skontroluje aktuálny disk v režime iba na čítanie (bez opráv) a vypíše zhrnutie (či našiel chyby).
- `/F` – opraví zistené chyby na disku.
- `/R` – vyhľadá chybné sektory a obnoví čitateľné informácie (zahrňa `/F`). Táto kontrola trvá dlhšie.
- `/scan` – (Windows 8/10) vykoná online scan na bežiacom systéme (rýchlejšie, neuzamkne disk).
- `/spotfix` – (Win8+) rýchla oprava chýb pri ďalšom reštarte (kombinácia online kontroly a offline opravy).

**Príklad použitia:** Skontrolovať disk **D:** s opravou chýb:

```
C:\> chkdsk D: /F
```

### Ilustračný výstup:

```
plaintext
CopyEdit
CHKDSK is verifying files (stage 1 of 3)...
 100% complete.
CHKDSK is verifying indexes (stage 2 of 3)...
 100% complete.
CHKDSK is verifying security descriptors (stage 3 of 3)...
 100% complete.

Windows has scanned the file system and found no problems.
No further action is required.

10485759 KB total disk space.
 203456 KB in 500 files.
   320 KB in 50 indexes.
    0 KB in bad sectors.
 50000 KB in use by the system.
 102400 KB occupied by the log file.
10241983 KB available on disk.

    4096 bytes in each allocation unit.
 2621439 total allocation units on disk.
2560495 allocation units available on disk.
```

Tento výpis (skrátенý) ukazuje priebeh kontroly fáza 1-3 a nakoniec zhrnutie. V uvedenom prípade neboli nájdené žiadne problémy ("found no problems. No further action is required."). Zobrazuje tiež štatistiky o mieste na disku.

ToSho



**Upozornenie:** Ak chcete opraviť chyby na systémovom disku (napr. C:, na ktorom beží Windows), **nemôže** `chkdsk` vykonať opravu za behu systému. Ak spustíte `chkdsk C: /F` počas bežiacieho Windows, pravdepodobne vypíše správu, že disk sa nedá uzamknúť a spýta sa, či naplánovať kontrolu pri ďalšom štarte. Stlačte **Y** (Yes) a reštartujte počítač – kontrola prebehne pred naboťovaním Windows.

**Poznámka:** Spustenie `chkdsk /R` môže trvať dlho (aj niekoľko hodín) v závislosti od veľkosti disku a množstva dát. Používajte ho, ak máte podozrenie na **vadné sektory** alebo fyzické problémy disku.

**Poznámka:** Príkaz `chkdsk` vyžaduje administrátorské oprávnenia pre opravu chýb (/F alebo /R).

Jednoduchú kontrolu bez opravy (bez parametrov) môžete spustiť aj ako bežný užívateľ, ale neupraví chyby.

## sfc – kontrola systémových súborov

**Popis:** Príkaz `sfc` (System File Checker) kontroluje integritu systémových súborov Windows a v prípade nájdenia poškodených alebo chýbajúcich verzií sa ich pokúsi nahradiť správnymi verziami. Najčastejšie sa používa ako `sfc /scannow`, čo spustí kompletnú kontrolu celého systému. Tento nástroj je užitočný pri riešení problémov, kde existuje podozrenie, že niektoré systémové knižnice alebo súbory Windows boli poškodené (napr. v dôsledku malware, nečakaného vypnutia alebo nesprávnej inštalácie softvéru).

**Syntax:** `sfc <parametre>`

- `/scannow` – okamžite spustí kontrolu všetkých chránených systémových súborov a automaticky opraví/ nahradí poškodené verzie z vyrovnávacej pamäte (alebo inštaláčného média). Tento proces môže trvať niekoľko minút.
- `/verifyonly` – vykoná len overenie (nezapisuje opravy).
- `/scanfile=<cesta súboru>` – skontroluje konkrétny súbor.
- `/verifyfile=<cesta súboru>` – overí konkrétny súbor (bez opravy).  
(Príkaz bez parametrov vypíše nápovedu k použitiu.)

**Príklad použitia:** Komplexná kontrola systémových súborov:

```
C:\> sfc /scannow
```

### Ilustračný výstup:

```
Beginning system scan. This process will take some time.
```

```
Beginning verification phase of system scan.  
Verification 100% complete.
```

```
Windows Resource Protection did not find any integrity violations.
```

V tomto prípade kontrola prebehla a systém hlási, že nenašla žiadne porušenia integrity (čiže všetky systémové súbory sú v poriadku). Ak by boli nájdené poškodené súbory, ktoré SFC nedokáže opraviť, zobrazí napríklad hlásenie *"Windows Resource Protection found corrupt files but was unable to fix some of them."* a odporučí pozrieť sa do log súboru (CBS.log) pre detaily.

**Upozornenie:** Príkaz `sfc /scannow` **musí** byť spustený v príkazovom riadku s oprávneniami správcu, inak sa nespustí.

**Poznámka:** Po spustení `sfc` sa odporúča reštartovať počítač, najmä ak nástroj našiel a opravil nejaké

ToSho



súbory. V prípade, že SFC nedokáže opraviť súbory, ďalším krokom môže byť použitie nástroja DISM (`dism /online /cleanup-image /restorehealth`) v novších systémoch Windows, ktorý vie opraviť komponenty systému z online zdrojov Windows Update.

## diskpart – správa diskových oddielov

**Popis:** Príkaz `diskpart` je nástroj na správu diskov, oddielov (partícií) a zväzkov v príkazovom riadku. Umožňuje vytvárať, mazať, formátovať a upravovať oddiely na pevných diskoch, USB kľúčoch a ďalších úložiskách. Na rozdiel od iných príkazov, `diskpart` vstupuje do svojho vlastného interaktívneho prostredia so samostatnou výzvou (`DISKPART>`), v ktorom sa zadávajú ďalšie sub-príkazy. Je to veľmi mocný nástroj a mal by sa používať s opatrnosťou, pretože ním môžete neúmyselne zmazať alebo prepísať oddiely na disku.

**Syntax:** `diskpart` (po spustení prejdete do interaktívneho režimu `diskpart`, kde zadávate ďalej uvedené príkazy)

V režime `DISKPART>` sú k dispozícii desiatky príkazov. Najpoužívanejšie sú:

- **list disk** – vypíše zoznam všetkých fyzických diskov v počítači.
- **list volume** – vypíše zoznam logických zväzkov (oddielov s písmenami jednotiek).
- **select disk <číslo>** – vyberie konkrétny disk pre ďalšie operácie.
- **select volume <číslo>** – vyberie konkrétny zväzok (oddiel).
- **create partition primary [size=<MB>]** – vytvorí nový primárny oddiel na vybranom disku (voliteľne s veľkosťou v MB, ak neuviedete, použije sa celé voľné miesto).
- **format fs=<system\_súborov> quick** – naformátuje vybraný oddiel zvoleným súborovým systémom (napr. ntfs, fat32).
- **assign letter=<písmeno>** – priradí vybranému oddielu písmeno jednotky.
- **clean** – vymaže vybraný disk (odstráni všetky oddiely, efektívne vyčistí disk do stavu neparticionovaného média). (*Používajte veľmi opatrne!*)
- **exit** – ukončí `diskpart` a vráti sa do klasického príkazového riadku.

**Príklad použitia:** Spustenie nástroja `diskpart` a výpis diskov a oddielov:

```
C:\> diskpart
```

```
DISKPART> list disk
```

Disk ###	Status	Size	Free	Dyn	Gpt
Disk 0	Online	238 GB	0 B		*
Disk 1	Online	14 GB	0 B		

```
DISKPART> list volume
```

Volume ###	Ltr	Label	Fs	Type	Size	Status
Volume 0	C	System	NTFS	Partition	100 GB	Healthy
Volume 1	D	Data	NTFS	Partition	138 GB	Healthy
Volume 2	E	USBDISK	FAT32	Removable	14 GB	Healthy

```
DISKPART> exit
```

```
C:\>
```

ToSho



V tomto príklade sme spustili `diskpart`, potom príkazom `list disk` zobrazili dva disky (Disk 0 a Disk 1) a príkazom `list volume` sme vypísali oddiely. Vidíme, že na Disku 0 sú dva oddiely (C: a D:) a Disk 1 je USB kľúč s oddielom E:. Potom sme príkazom `exit` opustili `diskpart`. (Počas práce v `DISKPART`> prostredí je príkazový riadok "uzamknutý" v `diskpart` režime, z ktorého treba výjsť príkazom `exit`.)

**Upozornenie:** `diskpart` príkazy **ihneď vykonávajú zmeny** na disku. Buďte zvlášť opatrní s príkazmi ako **clean** alebo **delete partition**, ktoré nemajú undo – zmazané oddiely a dáta z nich sa nedajú jednoducho obnoviť. Pred vykonaním deštruktívnych operácií si vždy skontrolujte, že máte vybraný správny disk/oddiel (príkazom `list disk/list volume a select`).

**Poznámka:** Príkaz `diskpart` vyžaduje spustenie s administrátorskými právami. Pokiaľ sa snažíte naskriptovať viacero príkazov `diskpart` naraz, môžete ich vložiť do textového súboru a spustiť `diskpart /s skript.txt` – tým sa príkazy v skripte vykonajú sekvenčne.

---

## gpresult – výsledky zásad skupiny (Group Policy)

**Popis:** Príkaz `gpresult` zobrazuje výsledné nastavenia Zásad skupiny (Group Policy) aplikované na počítač a používateľa. V prostredí Active Directory domény správca pomocou zásad skupiny konfiguruje nastavenia pre počítače a užívateľov. `gpresult` umožňuje overiť, ktoré konkrétne politiky (GPO – Group Policy Objects) sa aplikovali na daného užívateľa a počítač, a aké nastavenia z nich vyplývajú. Je to dôležitý diagnostický nástroj pri riešení problémov s politikami (napr. keď sa nejaké nastavenie neprejaví, v `gpresult` vidíme, či politika vôbec došla a či nebola prebitá inou).

**Syntax:** `gpresult [<parametre>]`

- **/R** – (summary) zobrazí súhrnný výsledok zásad pre aktuálneho užívateľa a počítač v textovej forme priamo v konzole.
- **/S <názov\_počítača>** – získanie výsledkov pre vzdialený počítač. (Potreba admin práv na cieľovom systéme a povolené vzdialené dotazy.)
- **/user <domain\user>** – špecifikovať iného užívateľa, než aktuálne prihláseného. (Vyžaduje tiež príslušné oprávnenia.)
- **/V** – verbose, zobrazí podrobnejšie informácie (pridá zoznam nastavení).
- **/X** alebo **/H** – export výsledkov do XML alebo HTML súboru (napr. `gpresult /H report.html` vygeneruje prehľadný HTML report).

**Príklad použitia:** Zobrazenie súhrnu zásad skupiny pre aktuálneho používateľa a počítač:

```
C:\> gpresult /R
```

### Ilustračný výstup:

```
GPRESULT /R
```

```
##### Computer Policy #####
```

```
Computer name:                MY-PC
OS Version:                    10.0.19044
```

```
Last time Group Policy was applied: 4/18/2025 at 5:00:15 PM
```

```
ToSho
```



```
Group Policy was applied from:          dcl.firma.local
Computer Settings:
  Applied Group Policy Objects:
    Default Domain Policy
    OfficeSettings
  Denied GPOs:
    MyCustomPolicy (Filtering)
```

... (ďalšie výstupy) ...

```
##### User Policy #####
```

```
User name:      FIRMA\John.Doe
Domain Name:    FIRMA
Last time Group Policy was applied: 4/18/2025 at 5:00:17 PM
User Settings:
  Applied Group Policy Objects:
    Default Domain Policy
    HR-Users Policy
  Denied GPOs:
    IT-Policy (Security Filtering)
```

(Výstup bol skrátený a upravený pre ilustráciu.) V uvedenej ukážke vidno rozdelenie na **Computer Policy** (politiky aplikované na počítač) a **User Policy** (politiky pre používateľa). Vidíme mená aplikovaných GPO (napr. *Default Domain Policy*, *OfficeSettings* atď.) a ktoré boli odmietnuté (Denied) napríklad kvôli filtru bezpečnostných skupín. Taktiež sú uvedené časy posledného použitia politik a doménu, z ktorej prišli. Podrobnejší výstup by mohol ukázať aj nastavenia vynúteného prihlasovacieho skriptu, cestu k profilom, atď., ak sú nastavené politikami.

**Poznámka:** Príkaz `gpresult /R` stačí spustiť ako bežný používateľ, ak zistíte výsledky pre seba. Avšak ak chcete získať kompletné informácie o nastaveniach počítača (Computer Policy) a nie ste administrátor, niektoré informácie môžu byť neprístupné. Preto sa často `gpresult` spúšťa v zvýšenom režime (ako správca) pre úplný obraz.

**Poznámka:** Pri použití parametra `/H` sa vytvorí HTML súbor s veľmi podrobným výpisom všetkých nastavení politiky. Je to prehľadnejšie na čítanie v prehliadači.

**Upozornenie:** `gpresult` má význam najmä v prostredí Active Directory domény. Na domácom počítači mimo domény tento príkaz stále niečo ukáže (napr. *Local Group Policy* ak nejaké existujú), ale väčšina sekcií bude prázdna alebo s údajmi "N/A".

---

**Záver:** Tento manuál pokrýva základné príkazy príkazového riadku Windows a niektoré pokročilé príkazy používané správcami. Pri práci v príkazovom riadku vždy postupujte opatrne, najmä s príkazmi, ktoré menia systémové nastavenia alebo mažu dáta. V prípade potreby viac informácií ku každému príkazu môžete použiť vstavanú pomoc príkazového riadku – stačí zadať príkaz s parametrom `**/?**` (napr. príkaz `/?`) a zobrazí sa podrobná nápoveda o použití daného príkazu. Pravidelným používaním a experimentovaním v bezpečných medziach získate istotu v ovládaní Windows CMD, čo Vám umožní efektívnejšiu prácu so systémom. Veľa šťastia pri používaní príkazového riadku!

ToSho