

# Operační systémy macOS část 1.7

## Terminál

### Úvod do Terminálu macOS

Terminál v macOS je mocný nástroj, který umožňuje komunikaci se systémem pomocí příkazové řádky. Je založen na UNIXových systémech a je podobný jako shell v Linuxu. V macOS se používá primárně shell zsh (od macOS Catalina), dříve bash.

### Stručná historie terminálu v macOS

- **Před Mac OS X:** V klasickém Mac OS nebyl standardní UNIXový terminál. Systém využíval grafické prostředí a neměl plnohodnotnou příkazovou řádku.
- **Mac OS X (2001):** Zaveden UNIXový základ (Darwin), poprvé zaveden Terminál jako aplikace s možností používat příkazovou řádku.
- **macOS Catalina (2019):** Shell změněn z Bash na Zsh.
- **Současnost:** Terminál podporuje Zsh, Bash, Fish a další shelly.

### Spuštění terminálu v macOS

- Pomocí Spotlight: `Cmd + Mezerník` a zadat "Terminal"
- Pomocí Launchpadu > Other > Terminal
- Pomocí Finderu: `/Applications/Utilities/Terminal`
- Klávesová zkratka (vlastní nastavení): např. pomocí Automatoru

### Porovnání macOS vs. Linux v terminálu

- macOS používá BSD varianty UNIX příkazů, zatímco Linux GNU varianty
- Některé příkazy se liší v syntaxi nebo možnostech (např. `ls`, `ps`)
- macOS má předinstalovaný Zsh, Linux obvykle Bash

### Příkazy

- `cd` – změna adresáře
- `cd /` – do rootu
- `cd ~` do domovského adresáře
- `cd ..` – o úroveň výš
- `cd .` – aktuální adresář
- `ls` – výpis obsahu adresáře
- `ls` – běžný výpis
- `ls -l` – detailní informace
- `ls -a` – včetně skrytých souborů
- `pwd` – zobrazí aktuální adresář
- `pwd`

- `whoami` – zobrazí jméno aktuálního uživatele
- `whoami`
- `mv` – přesouvá nebo přejmenovává soubory
- `mv star.txt novy.txt`
- `mv soubor.txt /Users/jan/Desktop`
- `mv *.txt adresar/`
- `cp` – kopírování souborů
- `cp a.txt b.txt`
- `cp soubor.txt /Users/jan/Desktop`
- `cp -r slozka/ cilova_slozka/`
- `ditto` – kopírování složek se zachováním atributů
- `ditto slozka1 slozka2`
- `ditto -V zdroj cil`
- `ditto --rsrc soubor1 soubor2`
- `df -h` – informace o disku
- `df -h` – lidsky čitelné jednotky
- `df` – běžný výpis
- `df -H /Volumes`
- `nano` – textový editor v terminálu
- `nano soubor.txt`
- `nano /etc/hosts`
- `sudo nano /private/etc/hosts`

#### Klávesové zkratky v nano:

- `Ctrl + O` – uložit
- `Ctrl + X` – zavřít
- `Ctrl + K` – vyjmout řádek
- `Ctrl + U` – vložit
- `Ctrl + W` – hledat
- `open` – otevře soubor/aplikaci v GUI
- `open .`
- `open soubor.txt`
- `open -a Safari https://www.apple.com`
- `man` – zobrazení manuálu
- `man ls`
- `man ps`
- `man zsh`

- `ping` – otestuje konektivitu
- `ping google.com`
- `ping -c 4 seznam.cz`
- `ping 8.8.8.8`
- `ifconfig` – informace o síti (macOS obdoba `ip`)
- `ifconfig`
- `ifconfig en0`
- `ifconfig lo0`
- `grep` – vyhledávání textu
- `grep "text" soubor.txt`
- `ps aux | grep Safari`
- `dmesg | grep usb`
- `traceroute` – sleduje cestu paketů
- `traceroute google.com`
- `traceroute 8.8.8.8`
- `traceroute seznam.cz`
- `dig` – DNS informace
- `dig google.com`
- `dig +short seznam.cz`
- `dig @8.8.8.8 example.com`
- `ps` – seznam procesů
- `ps`
- `ps aux`
- `ps aux | grep nano`
- `top` – informace o systémových procesech
- `top`
- `top -o cpu`
- `top | grep kernel_task`
- `kill` – ukončení procesu
- `kill 1234` – pošle procesu s PID 1234 signál `TERM` (ukončení), což je zdvořilá žádost o ukončení. Proces má šanci uklidit po sobě (např. uložit data).
- `kill -9 1234` – pošle signál `KILL`, který okamžitě ukončí proces bez možnosti reakce.
- `kill $(pgrep Safari)`
- `which $SHELL` – zjištění cesty k aktuálnímu shellu
  - Zobrazí cestu k binárnímu souboru shellu, který je nastaven jako výchozí pro uživatele.
- `echo $SHELL`
  - Výstup např.: `/bin/zsh` nebo `/bin/bash`
- `which zsh`
  - Výstup: `/bin/zsh` – zjistí, kde se nachází program `zsh`

- `which bash`
  - Výstup: `/bin/bash` – zjistí, kde se nachází program `bash`
- `bash, zsh, chsh` – správa a přepínání shellů
- `zsh` – spustí nový shell typu Zsh
- `bash` – spustí nový shell typu Bash
- `chsh -s /bin/zsh` – nastaví Zsh jako výchozí shell pro uživatele
- `touch soubor.txt` – vytvoří nový prázdný soubor `soubor.txt`, pokud neexistuje
- `touch soubor1.sh soubor2.sh` – vytvoří více souborů najednou
- `touch ~/Desktop/poznamka.txt` – vytvoří soubor na ploše

### Srovnání `bash` vs. `zsh`:

Vlastnost	Bash	Zsh
Autocomplete	Jednoduché	Pokročilé
Konfigurovatelnost	Omezená	Vysoká
Podpora pluginů	Omezená	Oh-My-Zsh, atd.
Kompatibilita	Vysoká (GNU)	Vysoká (UNIX)

- `diff` – rozdíly mezi soubory
- `diff soubor1.txt soubor2.txt`
- `diff -y soubor1.txt soubor2.txt`
- `diff -q dir1/ dir2/`
- `curl` – stahování dat z webu
- `curl https://example.com`
- `curl -O https://example.com/file.txt`
- `curl -I https://google.com`
- `history` – historie příkazů
- `history`
- `history | grep nano`
- `!100` – spustí 100. příkaz z historie

## Základy skriptování v Terminálu macOS

Co je shell skript?

Shell skript je textový soubor obsahující posloupnost příkazů, které můžeme spustit jako jeden celek. V macOS se nejčastěji používá zsh nebo bash.

### Vytvoření jednoduchého skriptu

Vytvoř soubor např. myscript.sh

```
#!/bin/bash
```

```
echo "Ahoj, toto je skript."  
ls -l
```

### Nastav spustitelnost:

```
chmod +x myscript.sh
```

### Spust skript:

```
./myscript.sh
```

### Struktura skriptu

- **#! (shebang):** Určuje, který shell se má použít (#!/bin/zsh nebo #!/bin/bash)
- **Komentáře:** # Toto je komentář
- **Příkazy:** libovolné shellové příkazy

### Proměnné

```
jmeno="Honza"  
echo "Jmenuji se $jmeno"
```

### Podmínky

```
if [ $vek -ge 18 ]; then  
    echo "Plnoletý"  
else  
    echo "Neplnoletý"  
fi
```

### Smyčky

#### for smyčka:

```
for i in 1 2 3; do  
    echo "Číslo: $i"  
done
```

#### while smyčka:

```
i=1  
while [ $i -le 3 ]; do
```

```
    echo "$i"
    i=$((i+1))
done
```

## Funkce

```
pozdrav() {
    echo "Ahoj $1"
}
pozdrav "Michal"
```

## Parametry skriptu

```
#!/bin/bash
echo "Skript voláš s parametrem: $1"
```

Spuštění: `./skript.sh argument`

## Příklady užitečných skriptů

### 1. Záloha souborů

```
#!/bin/bash
src="$1"
dest="$2"

if [ -d "$src" ]; then
    cp -r "$src" "$dest"
    echo "Záloha hotova."
else
    echo "Zdroj nenalezen."
fi
```

### 2. Test připojení

```
#!/bin/bash
ping -c 1 google.com > /dev/null && echo "Internet OK" || echo "Internet nefunguje"
```

### 3. Vyhledání textu ve všech souborech

```
#!/bin/bash
if [ -z "$1" ]; then
    echo "Zadej hledaný text"
    exit 1
fi
grep -rnw . -e "$1"
```

## Tipy pro skriptování v macOS

- Pro delší skripty používej `nano`, `vim` nebo externí editor (např. VS Code)
- Testuj skripty postupně
- Sleduj chyby pomocí `set -x` nebo `bash -x skript.sh`

### **Rozdíly bash vs. zsh v kontextu skriptů**

- Zsh má lepší autokompletační, ale většina skriptů je psána v Bash kvůli kompatibilitě
- Doporučení: piš skripty pro `#!/bin/bash` pokud máš běžné požadavky, `#!/bin/zsh` pokud využíváš specifické zsh vlastnosti