

Procesy, Soubory a Signály

Eliška Jégrová

20. 10. 2025

Obsah

Návratová hodnota příkazů

Procesy

Soubory

Signály

Návratová hodnota příkazů

- ▶ Každý příkaz po dokončení vrátí číslo — tzv. `exit code`.
- ▶ 0 = úspěch, jiná hodnota = chyba nebo jiný stav.
- ▶ Poslední návratovou hodnotu zobrazí `$?`

```
$ echo Ahoj
```

```
Ahoj
```

```
$ echo $?
```

```
0
```

```
$ ls neexistuje.txt
```

```
ls: cannot access 'neexistuje.txt': No such file or directory
```

```
$ echo $?
```

```
2
```

Samostatná práce – návratové hodnoty příkazů

1. Vytvoř soubor s názvem `prvni.txt`.
2. Zkontroluj, zda existuje soubor `prvni.txt` (`ls`)
3. Vypiš jeho návratovou hodnotu (`$?`).
4. Zkus spustit příkaz `grep "něco" soubor.txt`
5. Vypiš návratovou hodnotu.

Co je proces

- ▶ Proces je termín pro běžící program
- ▶ Každý proces má své PID (Process ID).
- ▶ Procesy lze zobrazit příkazem ps.

```
$ ps
```

PID	TTY	TIME	CMD
2180	pts/0	00:00:00	bash
2215	pts/0	00:00:00	ps

Všechny běžící procesy zobrazí: `ps -A`

Sledování procesů pomocí top

- ▶ Příkaz top zobrazuje procesy v reálném čase.
- ▶ Zobrazuje vytížení CPU, paměti a běžící procesy.
- ▶ Ukončení programu: klávesa q.

```
$ top
```

```
top - 09:42:01 up 1:03, 1 user, load average: 0.10,  
0.08, 0.02
```

PID	USER	PR	NI	VIRT	RES	S	%CPU	%MEM	COMMAND
2180	pepa	20	0	1234	532	S	0.0	0.1	bash
2250	pepa	20	0	980	400	S	0.0	0.0	sleep

Ukončení procesu

- ▶ Proces lze ukončit příkazem `kill`.
- ▶ Ukončení podle PID: `kill <pid>`
- ▶ Násilné ukončení podle PID: `kill -9 <pid>`

```
$ ps
```

```
2250 pts/0 00:00:00 top
```

```
$ kill 2250
```

Samostatná práce – procesy

1. Zapni si dva terminály, v prvním spust' top.
2. Ve druhém terminálu zjisti PID procesu top. (ps)
3. Z druhého terminálu ukonči proces top. (kill)

Práce s procesy v popředí a na pozadí

- ▶ Proces může běžet:
 - ▶ **v popředí** (foreground) – blokuje terminál, dokud neskončí,
\$ sleep 100
 - ▶ **na pozadí** (background) – běží zároveň s ostatními příkazy.
\$ sleep 100 &
[1] 1234 ← číslo úlohy a PID procesu
- ▶ Zobrazit běžící a pozastavené úlohy
\$ jobs
- ▶ Spustit proces v popředí
\$ fg
- ▶ Spustit proces v pozadí
\$ bg

Samostatná práce – práce s procesy

1. Spust': `$ sleep 1000 &`
2. Podívej se, že proces běží na pozadí. (`jobs`)
3. Přesuň úlohu do popředí. (`fg`)
4. Zastav běžící proces s `Ctrl+Z`
5. Zkontroluj, že je proces zastavený. (`jobs`)
6. Spust' proces na pozadí. (`bg`)
7. Zkontroluj, že proces běží na pozadí. (`jobs`, `ps`)

Co je soubor

- ▶ Soubor je něco, z čeho můžeme číst a/nebo do čeho můžeme zapisovat.

Druhy souborů co známe:

- ▶ Normální soubor uložený na disku.
`$ ps -Af > vystup.txt`
- ▶ Terminál – soubor reprezentující klávesnici a obrazovku.
- ▶ Roura – přenos dat mezi procesy.

```
$ ps -Af | grep -w ps
```

– výstup jednoho procesu proudí do druhého

Otevřené soubory procesu

- ▶ Každý proces si může otevírat další soubory – např. pro čtení či zápis.
- ▶ příkaz `lsof` - zjistíme, jaké soubory proces používá.
- ▶ Proměnná `$$` obsahuje PID (číslo aktuálního procesu Bashu).

```
$ echo $$  
5236
```

Příklad zkráceného výstupu: `$ lsof -p $$`

COMMAND	PID	USER	FD	TYPE	DEVICE	NAME
bash	5236	user	cwd	DIR	8,1	/home/user
bash	5236	user	txt	REG	8,1	/bin/bash
bash	5236	user	0u	CHR	136,0	/dev/pts/0
bash	5236	user	1u	CHR	136,0	/dev/pts/0
bash	5236	user	2u	CHR	136,0	/dev/pts/0

Terminál jako soubor

- ▶ Terminál má svůj název, např. `/dev/pts/0`.
V předchozím výpisu se jedná o otevřené soubory `0u`, `1u`, `2u`.
- ▶ Je to běžný soubor, do kterého lze zapisovat.

```
$ echo abc > /dev/pts/0
```

Chybový výstup

- ▶ Každý proces má:
 - ▶ 0 – standardní vstup,
 - ▶ 1 – standardní výstup,
 - ▶ 2 – chybový výstup.

```
$ cp a b
```

```
cp: cannot stat 'a': No such file or directory
```

```
$ cp a b > jiny.txt
```

– chyba se vypíše stále do terminálu

Přesměrování chybového výstupu

- ▶ Přesměrování chybového výstupu: `2>`.

```
$ cp a b 2> jiny.txt
```

- ▶ Přesměrování obojího:

```
$ ls existuje.txt neexistuje.txt > vystup.txt 2>  
chyby.txt
```

- ▶ Nebo obojí do jednoho souboru:

```
$ ls existuje.txt neexistuje.txt > jiny.txt 2>&1
```

Zahození výstupu

- ▶ Soubor `/dev/null` zahazuje vše, co se do něj zapíše.

```
$ ls existuje.txt neexistuje.txt 2> /dev/null
```

– chybové hlášky se „vyhodí“

```
$ find /var/cache > /dev/null
```

– zobrazí jen chyby (Permission denied)

Samostatná práce – práce se soubory

1. Zobraz si otevřené soubory procesu bash (ls, proměnná \$S).
2. Zjisti si z výpisu, jak se jmenuje soubor pro terminál.
3. Zapiš nějaký řetězec do souboru terminálu (echo).

Samostatná práce – přesměrování

1. Vytvoř soubor `vystup.log` a přesměruj do něj standardní výstup příkazu `$ ls / (>)`.
2. Spust' neexistující příkaz, např. `$ ls /neexistuje`, a přesměruj chybový výstup do souboru `chyby.log (2>)`.
3. Spoj výstup a chyby do jednoho souboru `vse.log (>, 2>&1)`.
4. Spust' příkaz: `$ ls /etc /neexistuje`, a přesměruj jeho standardní výstup do `/dev/null`. Co se zobrazilo?

Co je signál

- ▶ Signál je krátká zpráva
- ▶ Slouží k upozornění procesu, že se něco stalo (např. uživatel stiskl Ctrl-C).
- ▶ Typické signály:
 - ▶ SIGINT – přerušení (Ctrl + C)
 - ▶ SIGTSTP – zastavení (Ctrl + Z)
 - ▶ SIGTERM – žádost o ukončení
 - ▶ SIGKILL – okamžité ukončení

Pozn.: Seznam všech signálů najdete v \$ man 7 signal.

Zasílání signálů

- ▶ Signály se posílají příkazem `kill`.
- ▶ Syntaxe: `kill -SIGNÁL <PID>`

```
$ sleep 100 &
```

```
[1] 5423
```

```
$ kill -SIGSTOP 5423
```

- ▶ Seznam všech signálů: `kill -l`

Samostatná práce – práce se signály

1. Spust': `$ sleep 1000 &`
2. Podívej se, že proces běží na pozadí. (jobs)
3. Zastav běžící proces s pomocí `$ kill -SIGSTOP PID`.
4. Zkontroluj, že je proces zastavený. (jobs)
5. Spust' proces pomocí `$ kill -SIGCONT PID`.
6. Zkontroluj, že proces běží. (jobs, ps)
7. Ukonči proces pomocí `$ kill -SIGKILL PID`.