

Proudy, přesměrování, roura

"Jak kulaté připojit na hranaté"

Východiska

Tvůrci Unixu *Ken Thompson* a *Dennis Ritchie* jej postavili na jednoduché filozofii:

1. Vše je soubor.
2. Každý příkaz dělá jednu věc (má vstup a výstup).
3. Pomocí roury se dají výstupy se vstupy propojovat.

Tyto vlastnosti má i jazyk C. Ostatně, příkazy v Unixu jsou v C napsané.

Proudy

Každý program **během své činnosti** pracuje se **třemi datovými toky** (proudy) neboli "deskripty":

Název	Zkratka	Číslo	Význam	Výchozí cíl
stdin	standard input	0	vstup programu	klávesnice
stdout	standard output	1	běžný výstup	obrazovka
stderr	standard error	2	chybové hlášky	obrazovka

Proudy

Tyto proudy mu přidělí **používaný shell** (to černé okno) - v našem případě je shell **bash**. Technicky to probíhá nějak takto:

1. Uživatel spustí příkaz `cat /etc/apt/sources.list`.
2. Bash najde příkaz `cat` v prohledávaných adresářích (nejčastěji `/usr/bin`).
3. Vytvoří jeho kopii v RAM, **přidělí mu ty tři proudy** a spustí jej.
4. Příkaz `cat` si ze stdin (proud **0**) načte obsah souboru `/etc/apt/sources.list` a vypíše ho na **stdout** (proud **1**).
5. **stderr** (proud 2) se v tomto případě nevyužije.
6. Poté je příkaz smazán z RAM a proudy zanikají.

Základní příkazy pro práci s textovými soubory

Unixové systémy jsou na textových souborech životně závislé - jsou na nich postavené. V nich je veškerá konfigurace, logy a mnoho dalšího. Z těchto důvodů disponují mnoha příkazy, které se soubory nějak pracují. Pro základní práci potřebujeme zejména tyto:

- `cat` .. vypíše obsah souboru tak, jak je. S přepínačem `-n` vypisuje i čísla řádků.
- `less` .. soubor prochází řádek po řádku dopředu i dozadu.
- `more` .. soubor prochází stránku po stránce dopředu i dozadu.
- `head` .. vypíše prvních 10 řádků souboru. Jiný počet definuje přepínač `-n <číslo>`.
- `tail` .. vypíše posledních 10 řádků souboru. Jiný počet definuje přepínač `-n <číslo>`.
- `sort` .. Seřadí řádky souboru podle kódů ASCII tabulky.
- `wc` .. Vypíše statistiku o souboru - počet řádků, slov a znaků. Nejčastěji se používá pro výpis pouze počtu řádků - přepínač `-l`.
- `grep <vzor>` .. Vypisuje řádky souboru, které obsahují text uvedený ve `<vzor>`.

Všechny tyto příkazy mají na **stdin** soubor.

Přesměrování výstupu

Do souboru:

```
$ du -sh /etc > etc_size
```

Výstup příkazu `du -sh` se uloží do souboru `etc_size`, pokud soubor neexistuje, vytvoří se, pokud soubor existuje, přepíše se.

Přidání na konec souboru:

```
$ du -sh /etc >> etc_size
```

Výstup se připojí na konec souboru, pokud soubor neexistuje, vytvoří se.

Přesměrování chybového výstupu:

```
$ du -sh /etc 2> etc_err
```

Chyby (stderr) se uloží do souboru `etc_err`

Přesměrování

Přesměrování obou proudů najednou:

```
$ du -sh /etc &> etc_all
```

stdout i **stderr** se uloží do jednoho souboru `etc_all`. Toto se příliš nevyužívá.

Častěji (zejména ve skriptech) se chybový výstup "zahazuje" do `/dev/null`

Soubor `/dev/null`

`/dev/null` je speciální zařízení, které „pohlcuje“ všechna data, která do něj pošleme -- černá díra Unixu. Nic neukládá, nevrací chybu – prostě všechno zahodí.

Používá se, když nechceme, aby se výstup nebo chybové hlášky nezobrazovaly na obrazovce ani ukládaly do souboru. Dá se představit jako nekonečná odpadková schránka pro nepotřebné **výstupy programů**.

 Pozor: občas se mylně zaměňuje s příkazem `rm` pro smazání souborů:

- `rm ..` odstraní soubor, tedy blok dat (bajtů), které leží na disku.
- `/dev/null ..` pohlcuje proud dat (bajtů), které odněkud "tečou" jako výstup nějakého příkazu.

Soubor /dev/null

```
$ du -sh /etc 2> /dev/null
```

Přesměruje chybový výstup do „černé díry“.

```
$ mkdir backup 2> /dev/null
```

První vykonání příkazu chybu nevyvolá. Naopak opakované spuštění by již chybovou zprávu (na stderr) vyvolalo. Takto se jí elegantně zbavíme -- pošleme ji do "černé díry".

Roura

Roura (|) propojuje výstup jednoho programu (**stdout**) se vstupem druhého (**stdin**). To je základní princip „řetězení“ příkazů. Příklady:

Vypíše počet souborů/adresářů v /etc.

```
$ ls /etc | wc -l
```

Vypíše jen soubory jejichž názvy končí (\$) na conf.

```
$ ls /etc | grep "conf$"
```

Totéž, jen zjistí jejich počet.

```
$ ls /etc | grep "conf$" | wc -l
```

Pro začátek řetězce je znak ^. Příkaz vypíše soubory začínající na s.

```
$ ls /etc | grep "^s"
```

Vypíše jen soubory začínající na s a končící na conf.

```
$ ls /etc | grep "^s" | grep "conf$"
```