

VSTUPY, VÝSTUPY, ROURY, KOLONY



Vstupy a Výstupy (datové proudy)

- **Používají se pro komunikaci programu s okolím**
 - **Řešeno na úrovni OS, standardní proudy**
 - **stdin** – standardní vstup (proud 0)
 - nejčastěji klávesnice
 - **stdout** – standardní výstup (proud 1)
 - napojen na terminál
 - **stderr** – standardní chybový výstup (proud 2)
 - napojen na terminál
- > rm nic ---> rm: nelze odstranit „nic“: Adresář nebo soubor neexistuje (na stderr)
- > **cat** (čte ze stdin posílá na stdout)



Přesměrování I

- **Přesměrování standardního výstupu**

- **příkaz > soubor** přesměruje stdout do souboru
 - Vytvořen buď nový soubor, nebo přepsán starý
 - `ls -l > vypis.txt`
 - `yes > /dev/null`
 - Skript:

```
#!/bin/bash
for pid in $(ls /proc/ | grep -E "^[[:digit:]]+$"); do
echo $pid
done > log.txt
```
- **příkaz >> soubor** přidání na konec již existujícího, nebo vytvoří nový
 - `ls -l >>vypis.txt`



Přesměrování II

- **Přesměrování standardního vstupu**

- **příkaz < soubor** čte ze souboru místo stdin

- `cat < log.txt` (cat log.txt funguje stejně)

- HERE DOCUMENT

- **příkaz <<"TAG" document TAG**

- `cat << "END"`

- `mama mele maso`

- `END ---> mama mele maso`

- HERE STRING

- **příkaz <<< řetězec**

- `#!/bin/bash`

- `for pid in $(ls /proc/ | grep -E "^[[:digit:]]+$" | sort -n)`

- `do`

- `cat <<< "toto je proces pid : $pid "`

- `done`



Přesměrování III

- **Přesměrování stderr**

- příkaz `2> soubor`
 - `rm nic 2> /dev/null` (zahazuje chybové hlášky)

- **Slučování vstupů**

- stderr s stdout: `2>&1`
 - `rm nic > log.txt 2>&1`
- stdout s stderr: `1>&2`
 - `echo "chyba" 1>&2`

- **Rozdvojení proudu `tee` (T)**

- Zapisuje jak do souboru, tak na stdout
 - `ls -l | tee /dev/pts/1`



Roury I

- **Komunikační mechanismus umožňující předávat výstup jednoho programu na vstup druhého.**
 - Je implementována jako fronta
 - Synchronizovaná
 - Pokud je plná (64kB) je producent zastaven
 - Pokud je prázdná je zastaven konzument
- **Anonymní |**
 - > dmesg | grep usb
- **Pojmenované**
 - vytvoření pomocí **mkfifo**: > mkfifo roura
 - práce jak s „běžným souborem“
 - ls -l -R / >roura 2> /dev/null (na jednom terminálu)
 - cat roura (na druhém terminálu)
 - Zacyklení (nejprve mkfifo roura1)
 - echo "baf" | cat - roura > roura1 &
 - cat <roura1 > roura



Roury II

- **Kolony-spojení více programů pomocí rour**

- `>dmesg | grep usb | wc -l`
- Užitečné programy
 - `tr` – transformace znaků
 - `cut` – ořezání znaků
 - `>cat /etc/passwd | cut -d ":" -f 1 | sort | tr [[:lower:]] [[:upper:]]`
 - `wc` – počítá slova, řádky, znaky
 - `tail`, `head` – vrací poslední, první řádky
 - `grep`, `awk` – filtruje pomocí regulárních výrazů
 - `sort` – třídí
 - `uniq` – zadržuje vícenásobný výskyt
 - `xargs` – pro předávání velkého množství parametrů, v některých starších verzích linuxu není možné předat libovolně dlouhou příkazovou řádku, je nutné rozsekat na dílčí části
 - `>find / -type f -name "*.cs" 2> /dev/null | xargs file`

