

C2110 *Operační systém UNIX a základy programování*

4. lekce

Petr Kulhánek, Jakub Štěpán

kulhanek@chemi.muni.cz

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta
Masarykova univerzita, Kotlářská 2, CZ-61137 Brno



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

➤ Souborový systém II

- speciální znaky, kvóty, přístupová práva, diskové zařízení, USB disky, vyhledávání souborů

➤ Procesy I

- proces, multitasking, základní příkazy, přesměrování, roury

Souborový systém II

- Speciální znaky
- Kvóty
- Přístupová práva
- Diskové zařízení
- USB disky
- Vyhledávání souborů

Speciální znaky

Speciální znaky (wildcards, divoké znaky) v názvech souborů nebo adresářů:

- * - cokoliv v názvu (bez skrytých souborů)
- ? - jeden znak v názvu
- [] - rozsah (jeden znak) v názvu, př. [ajk], [a,j,k], [a-j]

Rozvoj (expanzi) speciálních znaků provádí shell ještě před spuštěním samotného příkazu. Expanzi lze zabránit uvedením jména v uvozovkách nebo použitím zpětného lomítka před speciálním znakem.

Příklady

```
$ cp *.pdf Documents/
```

zkopíruje všechny pdf dokumenty z aktuálního adresáře do podadresáře Documents

```
$ rm *
```

smaže všechny soubory v aktuálním adresáři (kromě adresářů)

```
$ mv A? Tmp/
```

přesune soubory s názvem začínajícím písmenem "A" a obsahujícím dva znaky do adresáře "Tmp"

Speciální znaky

K expanzi speciálních znaků dojde pouze tehdy, pokud existuje alespoň jeden soubor vyhovující danému předpisu:

Příklady:

```
$ cd
$ echo D*
Desktop Documents Downloads
$ echo A*
A*
$ echo "D*"
D*
$ echo D\*
D*
```

Kvóty

Pro vaše domovské adresáře jsou nastaveny kvóty na využití diskového prostoru na diskovém oddílu **wolf.wolf.inet:/export/home/**. Aktuální stav zaplnění a nastavení kvót lze zjistit příkazem **quota**:

```
[kulhanek@wolfn ~]$ quota -vs
Disk quotas for user kulhanek (uid 18773):
  Filesystem  blocks      quota   limit  grace  files      quota   limit
wolf.wolf.inet:/export/home/
                1550M    1954M    2051M          20453         0         0
```

Aktuální využití

Kvóta, kterou lze dočasně překročit.

Tvrдый limit, který již nelze překročit.

Překročení kvóty může vést k **nefunkčnímu přihlášení** pomocí grafického rozhraní. V tomto případě se přihlaste v textovém terminálu (např: Ctrl+Alt+F1) a přesuňte soubory na jiný diskový oddíl (např. dočasně do adresáře /scratch/vas_login nebo smažte nepotřebné soubory).

Přístupová práva

Přístupová práva určují jaké operace může uživatel provádět se soubory či adresáři v souborovém systému.

Přístupová práva:

r	možnost číst soubor	vypsat obsah adresáře
w	možnost měnit soubor	změnit obsah adresáře
x	možnost spuštění souboru	možnost vstoupit do adresáře

Každý soubor či adresář má určeného vlastníka a skupinu. Přístupová práva se uvádějí zvlášť pro vlastníka souboru, pro skupinu a ostatní uživatele.

\$ ls -l

```
drwxrwxr-x  3 kulhanek lcc  4096 2008-10-13 09:57 bin/
drwx-----  2 kulhanek lcc  4096 2008-10-13 09:58 Desktop/
-rw-rw-r--  1 kulhanek lcc  5858 2008-10-17 11:58 distance.cpp
```

↑
přístupová práva

↑
vlastník (user)
a skupina uživatelů (group)

↑
velikost (B)

↑
čas poslední změny

↑
jméno souboru či adresáře/

Změna přístupových práv

Přístupová práva souborů a adresářů může měnit vlastník souboru nebo superuživatel příkazem **chmod**.

```
$ chmod permissions file1 [file2 ...]
```

typ: soubor (-) nebo adresář (d) → u g o
přístupová práva → drwxrwxr-x

Přístupová práva:

r	možnost číst soubor	vypsát obsah adresáře
w	možnost měnit soubor	změnit obsah adresáře
x	možnost spuštění souboru	možnost vstoupit do adresáře

Skupina práv:

u	vlastník (user)
g	skupina uživatelů (group)
o	ostatní (other)
a	všichni (all), právo se aplikuje na u,g,o

Příklad:

```
$ chmod u+x,g-w soubor
```

Přidá (+) právo pro spuštění vlastníkovi a
Odstraní (-) právo zápisu pro skupinu

Změna vlastníka, změna skupiny

Vlastníka souborů a adresářů může měnit pouze superuživatel příkazem **chown**.

Skupinu uživatelů pro soubory a adresáře může měnit vlastník nebo superuživatel příkazem **chgrp**. Vlastník může použít pouze skupiny, do kterých patří (lze zjistit příkazem **id**).

```
$ chgrp group_name file1 [file2 ...]
```

```
[kulhanek@wolf01 ~]$ id  
uid=18773(kulhanek) gid=2001(lcc) groups=2001(lcc),2027(kulhanek),2030(compchem)
```

```
[kulhanek@wolf01 ~]$ ls -ld Documents/  
drwxr-xr-x 9 kulhanek lcc 4096 Feb 16 2012 Documents/
```

```
[kulhanek@wolf01 ~]$ chgrp compchem Documents/
```

← změna skupiny

```
[kulhanek@wolf01 ~]$ ls -ld Documents/  
drwxr-xr-x 9 kulhanek compchem 4096 Feb 16 2012 Documents/
```

Diskové zařízení

Přehled o využití souborových systémů, o diskových zařízeních a jejich přípojných bodech poskytuje příkaz **df**.

```
[kulhanek@wolf01 ~]$ df -Th
Filesystem                                Type      Size  Used Avail Use% Mounted on
/dev/mapper/server1-root                  ext3      20G   5.9G   13G   32% /
udev                                       devtmpfs  3.9G   4.0K   3.9G    1% /dev
tmpfs                                       tmpfs     1.6G   444K   1.6G    1% /run
none                                       tmpfs     5.0M    0     5.0M    0% /run/lock
none                                       tmpfs     3.9G   12K   3.9G    1% /run/shm
/dev/sda1                                  ext3      168M   36M   124M   23% /boot
/dev/mapper/server1-scratch                ext3      20G   1.9G   17G   11% /scratch
/dev/mapper/server1-vbox                   ext3      20G   5.6G   14G   30% /win
wolf.wolf.inet:/export/software/ncbr      nfs        93G   60G   29G   68% /software/ncbr
wolf.wolf.inet:/export/home                nfs       280G  164G  102G   62% /auto/home
```

zařízení

typ souborového systému

přípojný bod

Typy souborových systémů:

ext3,ext4 – third / fourth extended filesystem (nativní souborový systém linuxu)

nfs – network filesystem

vfat – Virtual File Allocation Table (souborový systém používaný MS Windows)

ntfs - New technology File System (vyvinul Microsoft pro svoje operační systémy)

USB disk

USB disky se **připojují** automaticky v grafickém prostředí do svazku **/media**.

```
[kulhanek@wolf01 ~]$ df -Th
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
.....						
wolf.wolf.inet:/export/home	nfs	280G	164G	102G	62%	/auto/home
/dev/sdg1	vfat	962M	841M	122M	88%	/media/B19A-1CA2

Disk lze **odpojit** v grafickém prostředí nebo příkazem **umount**. Argumentem příkazu je přípojný bod zařízení.

```
[kulhanek@wolf01 ~]$ umount /media/B19A-1CA2
```

Disk lze odpojit pouze pokud není využíván (nesmí být otevřen žádný soubor, žádný proces nesmí mít jako pracovní adresář nastaven (pod)adresář z přípojného bodu včetně přípojného bodu). Přehled procesů využívajících daný adresář (přípojný bod) lze získat příkazem **lsdf** (nebo **fuser**).

```
[kulhanek@wolf01 ~]$ lsdf /media/B19A-1CA2/
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
bash	31521	kulhanek	cwd	DIR	8,97	4096	518	/media/B19A-1CA2/GoslarFinal
bash	31893	kulhanek	cwd	DIR	8,97	4096	518	/media/B19A-1CA2/GoslarFinal
vi	32011	kulhanek	cwd	DIR	8,97	4096	518	/media/B19A-1CA2/GoslarFinal
vi	32011	kulhanek	4u	REG	8,97	12288	535	/media/B19A-1CA2/GoslarFinal/.README.swp

Vyhledávání souborů

K vyhledávání souborů lze použít příkaz **find**.

pokud není uvedeno, hledá se v aktuálním adresáři

\$ **find** [where] **what**

vyhledávání je rekurzivní (výchozí stav)

Vyhledávací dotaz (**what**) je komponován z dílčích dotazů, které je možné spojovat logickými operátory.

Nejběžnější dotazy:

- name** *pattern* najde všechny soubory, které mají jméno *pattern*
pattern může obsahovat speciální znaky: *,?,[]
(při použití speciálních znaků uvádíme *pattern* v uvozovkách)
- type** *c* najde všechny soubory typu *c* (soubor, adresář, atd. výčet typů viz. man find)

Logické operátory:

- and** levý a pravý dotaz je splněn současně
- or** je splněn levý nebo pravý dotaz

Vyhledávání souborů, příklady

```
$ find /home/ -name '*.txt'
```

v adresáři /home/ nalezne všechny soubory, které mají zakončení .txt

```
$ find ~kulhanek -name '*.txt' -or -name '*.hpp'
```

v adresáři /home/kulhanek nalezne všechny soubory, které mají zakončení .txt nebo .hpp

```
$ find -name 'D*' -and -type d
```

v aktuálním adresáři nalezne všechny podadresáře, jejichž jména začínají písmenem D

Přehled příkazů

Souborový systém:

ls	vypíše obsah adresáře
pwd	vypíše cestu k aktuálnímu pracovnímu adresáři
cd	změní aktuální pracovní adresář
mkdir	vytvoří adresář
rmdir	odstraní prázdný adresář
cp	zkopíruje soubor či adresář
mv	přesune soubor či adresář
rm	odstraní soubor či adresář
find	nalezne soubor či adresář
du	vypíše velikost adresáře nebo souborů
stat	vypíše podrobné informace o souboru či adresáři
df	vypíše informace o připojených diskových oddílech
quota	vypíše informace o nastavených kvótách pro přípojně body
scp	kopíruje soubory mezi počítači

Cvičení

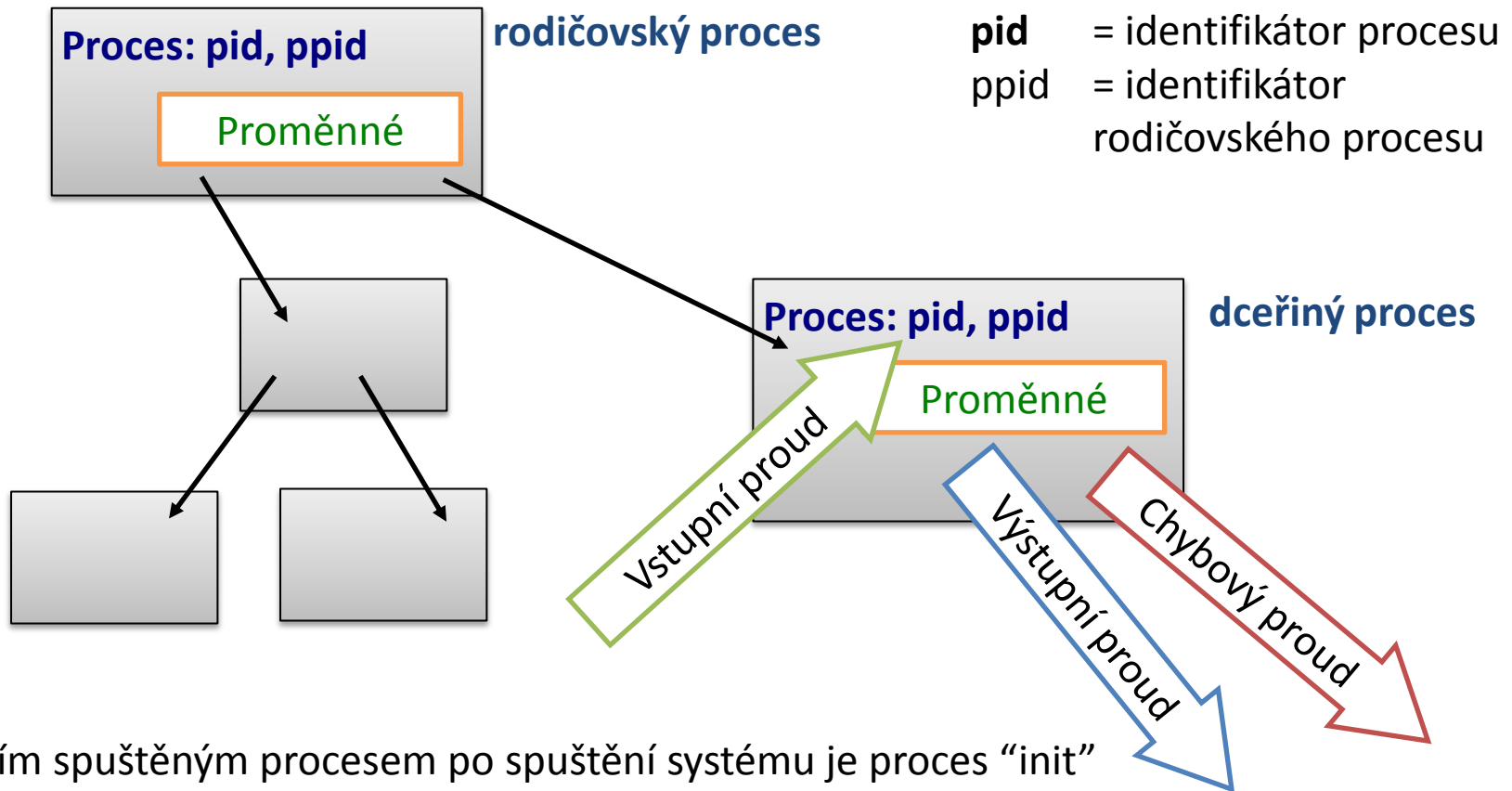
1. Ve svém domovském adresáři vytvořte podadresář **Data**
2. Do adresáře **Data** zkopírujte obsah adresáře `/home/kulhanek/Data/` včetně podadresářů
3. Nalezněte všechny soubory s koncovkou **.cpp**, které se nacházejí v adresáři **Data** (jména souborů vypište na obrazovku)
4. V adresáři `/scratch/vas_login` vytvořte adresář **Headers**
5. Do adresáře **Headers** zkopírujte všechny soubory z adresáře `/home/kulhanek/Data/dev/src`, které mají koncovkou **.h**
6. Do adresáře **Headers** přesuňte všechny soubory z adresáře `/home/vas_login/Data/dev/src`, které mají koncovkou **.cpp**
7. Jaká je velikost souboru `/home/kulhanek/Data/dev/src/GraphicsSetup.cpp` v B a kB
8. Ověřte, zda-li nepřekračujete kvótu nastavenou na váš domovský adresář
9. V adresáři **Headers** smažte všechny soubory s koncovkou **.h** a obsahující slovo **Graphics** na začátku jména souboru
10. Smažte adresář **Headers**

Procesy I

- **Proces**
- **Multitasking**
- **Základní příkazy**
- **Přesměrování**
- **Roury**

Procesy

Proces je instance běžícího **programu**. V daný okamžik může na jednom fyzickém procesoru běžet jeden proces. Operační systém pak zajišťuje rychlé přepínání běžících procesů, což vede k jejich zdánlivému souběžnému běhu (**multitasking**).



- Prvním spuštěným procesem po spuštění systému je proces "init"
- Každý příkaz spuštěný v shellu (příkazové řádce) je procesem

Výpis procesů

Procesy lze vypsat příkazy:

- top** průběžně zobrazuje procesy setříděné podle zátěže procesoru (ukončení klávesou q)
- ps** vypíše procesy běžící v daném terminálu nebo podle zadaných specifikací (`ps -u user_name`)
- pstree** vypíše procesy (stromový výpis)

```
$ ps
  PID TTY          TIME CMD
 8763 pts/5        00:00:00 bash
 8852 pts/5        00:00:00 gimp
 8857 pts/5        00:00:00 ps
```

jméno spuštěného příkazu

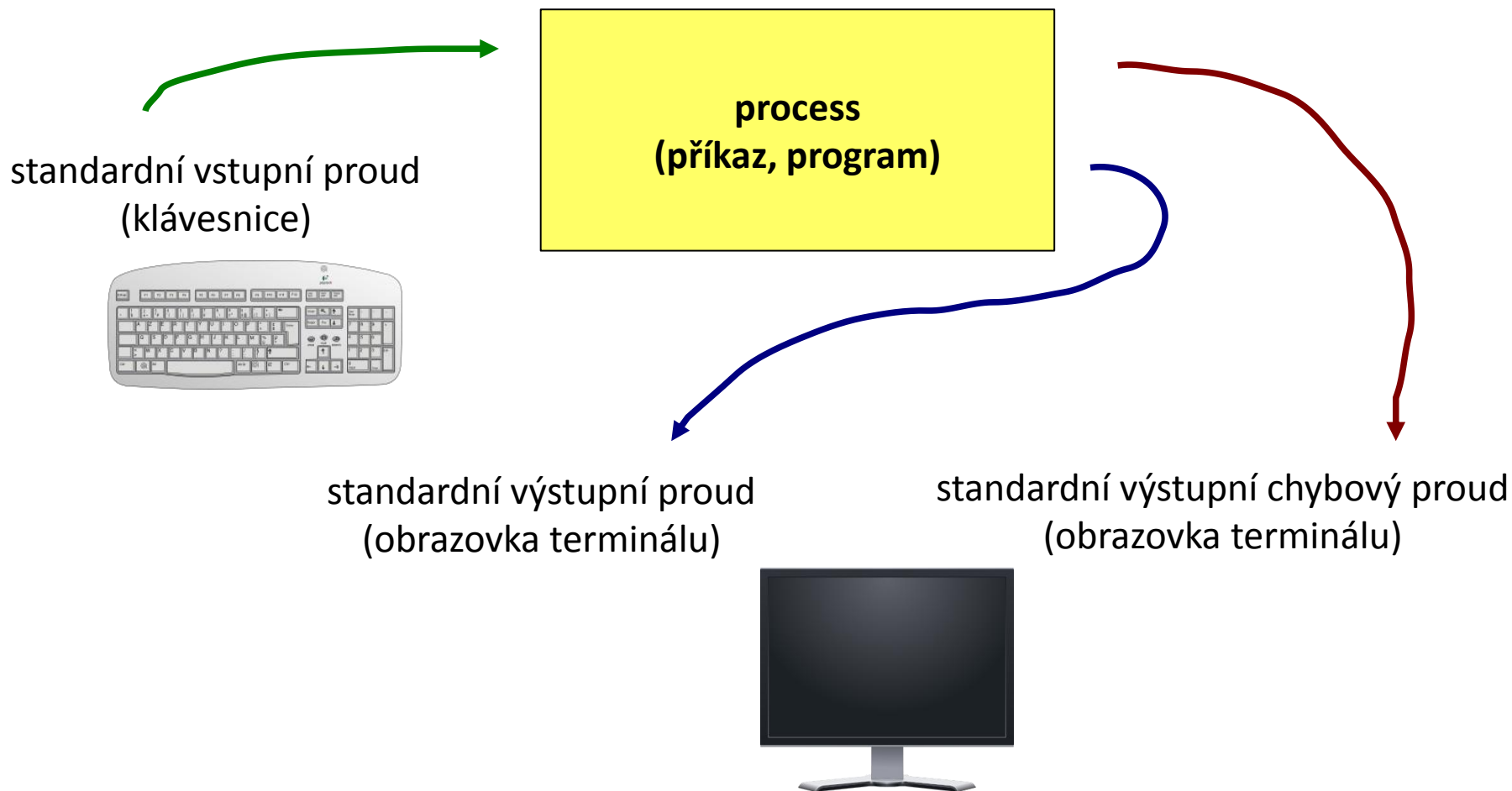
číslo procesu

terminál, ve kterém proces běží

spotřebovaný strojový čas

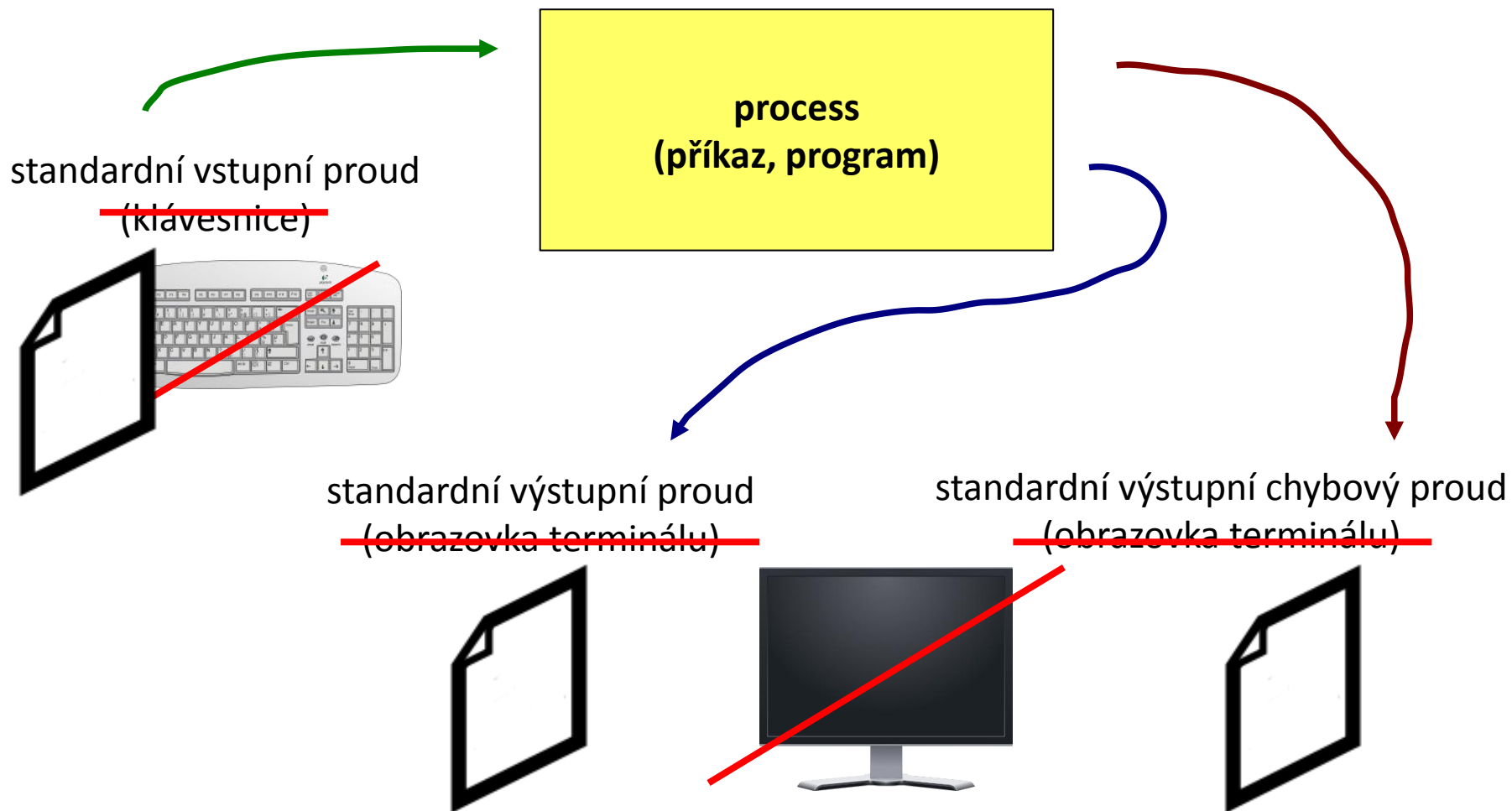
Standardní proudy

Vstupně-výstupní proudy slouží procesu ke **komunikaci** se svým okolím. Každý proces otevírá **tři standardní proudy**:



Přesměrování

Vstupně-výstupní proudy lze přesměrovat tak, aby používaly **soubory** místo klávesnice či obrazovky.



Přesměrování vstupu

Přesměrování standardního vstupu programu `my_command` ze souboru `input.txt`.

```
$ my_command < input.txt
```

Přesměrování standardního vstupu programu `my_command` ze souboru skriptu.

```
.....  
./my_command << EOF  
první radka textu  
druhá radka textu  
třetí radka textu  
EOF  
.....
```

značka určující konec vstupu
(volí uživatel)

text, který tvoří načítaný vstup

konec vstupu, značku *nesmí*
obklopotvat mezery

Tento způsob přesměrování je obzvláště výhodné používat ve skriptech, nicméně funguje i v příkazové řádce. Výhodou je expanze proměnných v načítaném textu.

Přesměrování výstupu

Přesměrování standardního výstupu programu `my_command` do souboru `output.txt`.
(Soubor `output.txt` je vytvořen. Pokud již existuje, je jeho původní obsah **smazán**.)

```
$ my_command > output.txt
```

Přesměrování standardního výstupu programu `my_command` do souboru `output.txt`.
(Soubor `output.txt` je vytvořen. Pokud již existuje, je výstup programu `my_command` **připojen** na jeho konec.)

```
$ my_command >> output.txt
```

Podobná pravidla platí pro standardní **chybový** výstup, v tomto případě se používají následující operátory:

```
$ my_command 2> errors.txt
```

```
$ my_command 2>> errors.txt
```

Spojování výstupních proudů

Standardní výstup **a** standardní chybový výstup programu `my_command` lze současně přeměřovat do souboru `output.txt`.

```
$ my_command &> output.txt
```

Výše uvedený postup nelze použít pro operátor `>>`.

```
$ my_command &>> output.txt      nefunguje
```

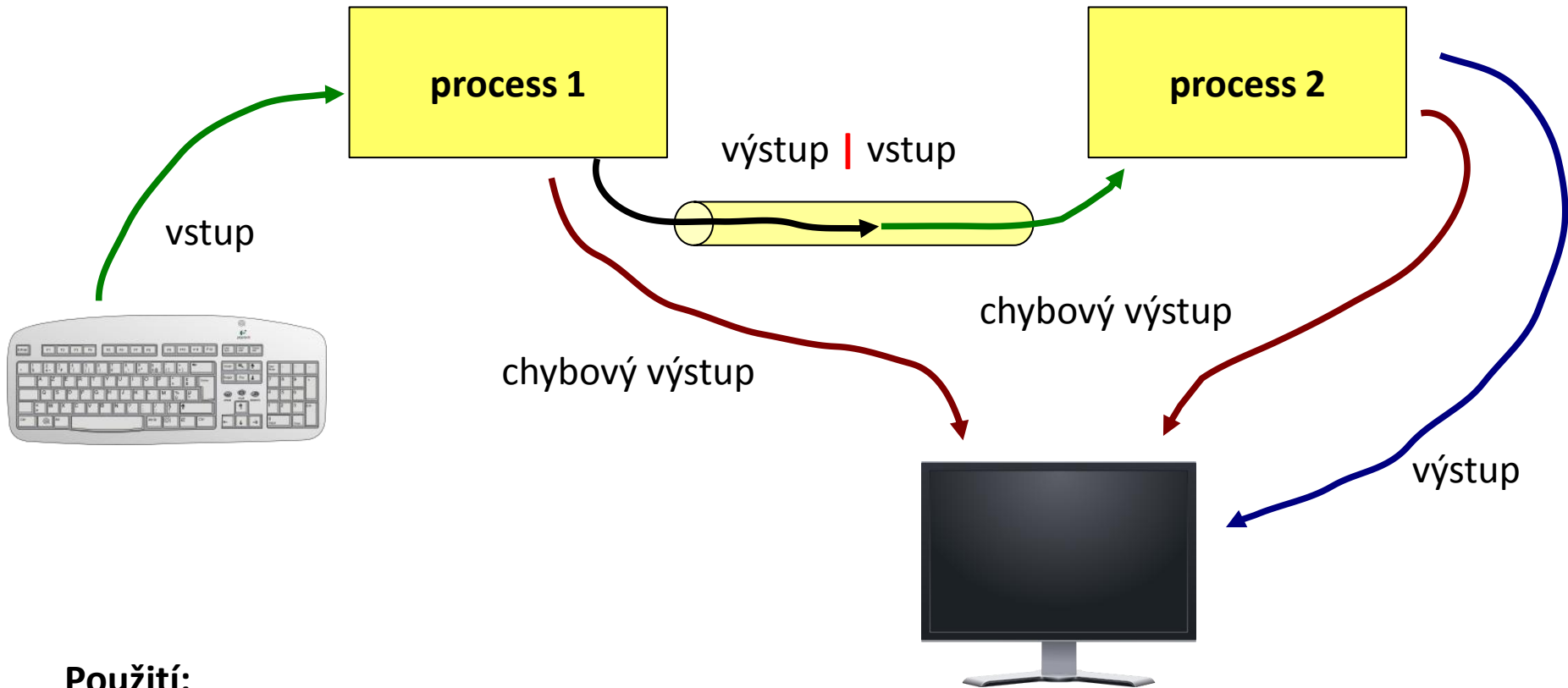
Řešení: Nejdříve je nutné **přesměrovat** standardní výstup a poté **spojit** standardní chybový výstup s výstupem standardním.

```
$ my_command >> output.txt 2>&1      pořadí je důležité!
```

```
$ my_command 2>&1 >> output.txt      nefunguje
```

Roury (pípy)

Roury slouží ke spojování standardního výstupu jednoho procesu se standardním vstupem jiného procesu.

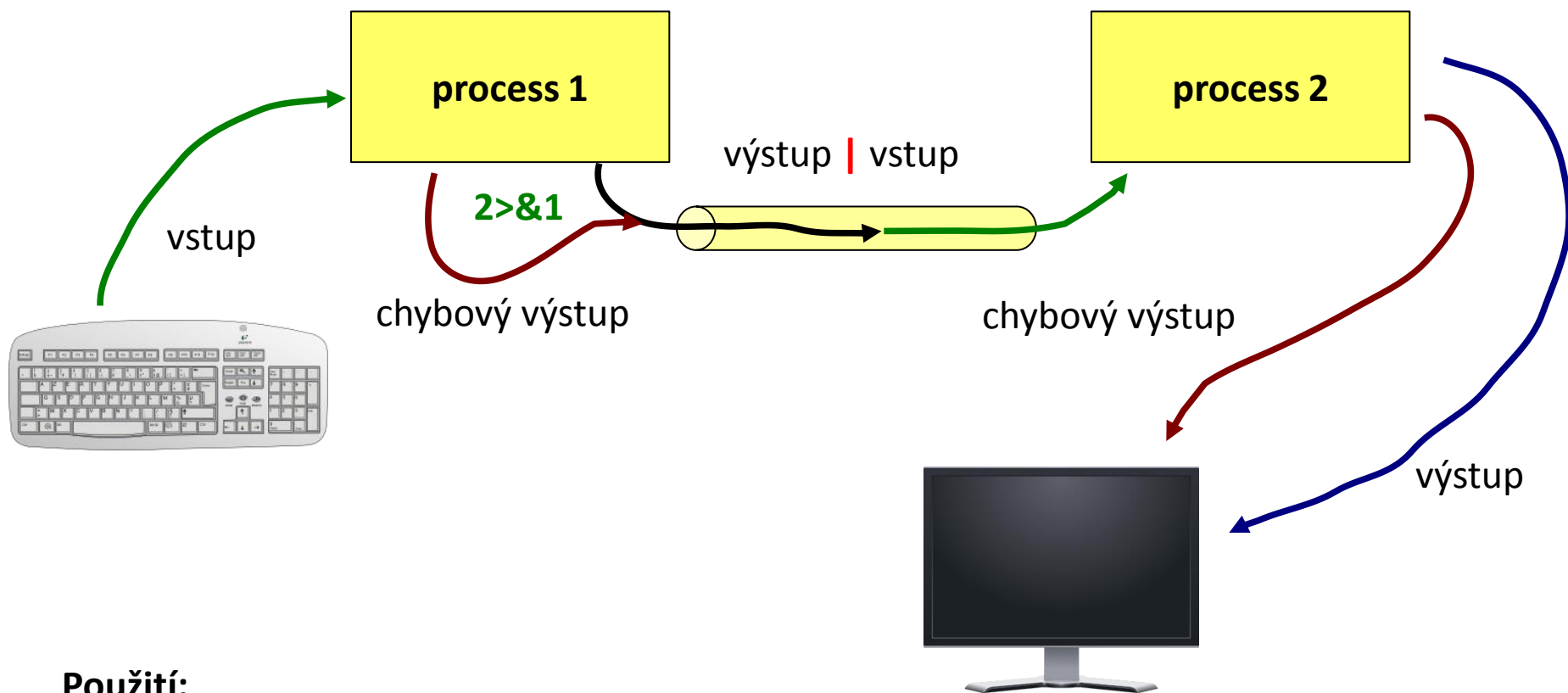


Použití:

```
$ command_1 | command_2
```


Roury a chybový proud

Přenos standardního chybového výstupu přes rouru je možné provést po jeho spojení se standardním výstupem.



Použití:

```
$ command_1 2>&1 | command_2
```

Příkazy pro cvičení

- cat** spojí obsah více souborů do jednoho (za sebe), případně vypíše obsah jednoho souboru
- paste** spojí obsah více souborů do jednoho (vedle sebe)
- wc** informace o souboru (počet řádků, slov a znaků)
- head** vypíše úvodní část souboru
- tail** vypíše koncovou část souboru

Ukázky použití:

- \$ `cat soubor1.txt soubor2.txt`
spojí obsah souborů soubor1.txt a soubor2.txt za sebe a výsledek vypíše na obrazovku
- \$ `paste soubor1.txt soubor2.txt`
spojí obsah souborů soubor1.txt a soubor2.txt vedle sebe a výsledek vypíše na obrazovku
- \$ `wc soubor.txt`
vypíše počet řádků, slov a znaků, které obsahuje soubor soubor.txt
- \$ `head -15 soubor.txt`
vypíše prvních 15 řádků ze souboru soubor.txt
- \$ `tail -6 soubor.txt`
vypíše posledních 6 řádků ze souboru soubor.txt

Cvičení – procesy, přístupová práva

1. Vypište procesy, které běží v daném terminálu (použijte příkaz **ps**)
2. Vypište všechny procesy v systému a pouze vaše procesy (použijte příkaz **ps**)
3. Vypište stromový seznam všech spuštěných procesů
4. Určete jméno prvního rodičovského procesu všech vašich procesů (použijte příkaz **ps tree**)
5. Vytvořte adresář **Procesy** ve vašem domovském adresáři
6. Do adresáře **Procesy** zkopírujte program (soubor) **/bin/ls** a přejmenujte jej na **myls**
7. Spusťte program **myls** (jedná se o nestandardní program, název programu budete muset uvést včetně cesty). Jakou funkci program vykonává?
8. Jaké přístupová práva má soubor **myls** ?
9. Souboru **myls** odstraňte všechna práva pro spuštění. Co se stane, pokud se pokusíte program **myls** spustit?

Cvičení - přesměrování

1. Nalezněte všechny soubory s koncovkou **.f90** , které obsahuje adresář **/home/kulhanek/Data/dev/src/** , seznam souborů uložte do souboru **~/Procesy/seznam.txt**
2. Kolik řádků obsahuje soubor **seznam.txt** ?
3. Vypište první dva řádky ze souboru **seznam.txt** nejdříve na obrazovku a poté do souboru **dva_radky.txt**
4. Vypište pouze třetí řádek ze souboru **seznam.txt**
5. V adresáři **/proc** nalezněte všechny soubory, které začínají písmeny **cpu** . Z výpisu odstraňte informace o nepovoleném přístupu přesměrováním chybového proudu do **/dev/null**