

# Úvod do Linuxu

Petr Kulhánek

[kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz)

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

# Obsah

- **Klastr WOLF**  
struktura, pravidla používání, správci
- **Přihlašování**  
místní a vzdálené přihlášení, export displeje, změna hesla
- **Programové vybavení**  
systémové aplikace, vědeckotechnické aplikace
- **Textové editory**  
vi, grafické textové editory
- **Příkazová řádka**  
terminály, struktura, historie a automatické dokončování
- **Souborový systém**  
struktura, absolutní a relativní cesty, práva, speciální soubory, diskové oddíly
- **Příkazy**  
manuálové stránky, přehled příkazů
- **Procesy**  
procesy, standardní vstup a výstup, přesměrování, roury
- **Tipy a triky**  
užitečné kombinace příkazů

# Klastr WOLF

---

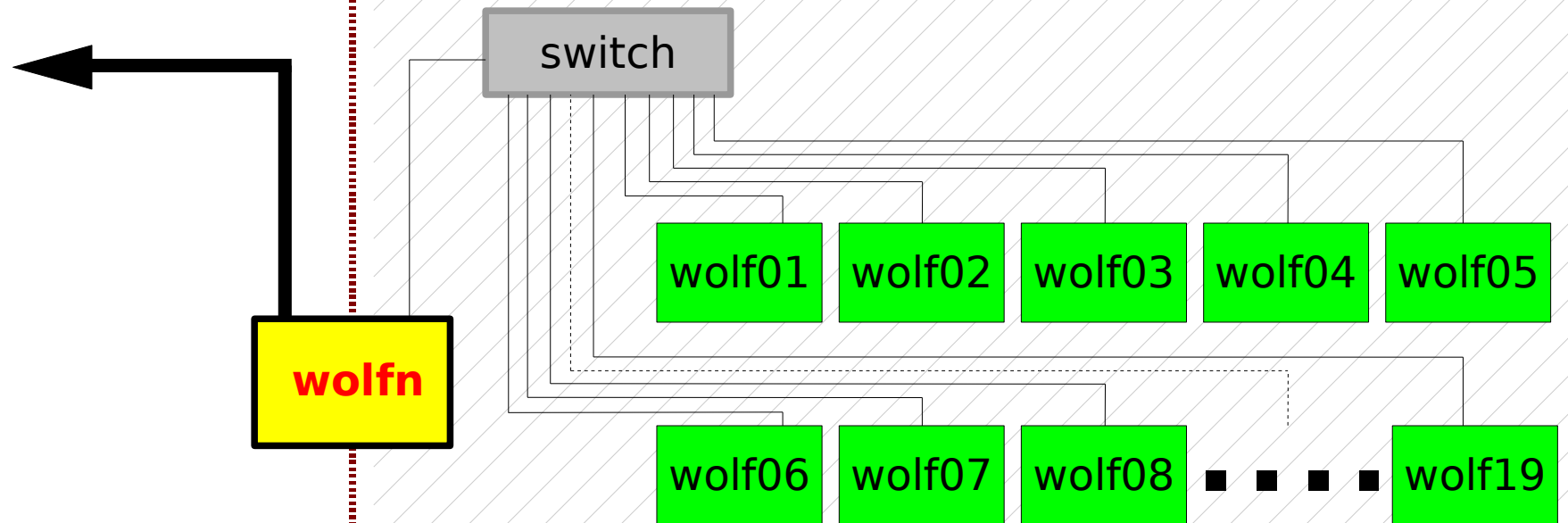
- **Struktura klastru**
- **Pravidla používání klastru**
- **Správci klastru**

# Struktura klastru WOLF

univerzitní síť  
**wolfn.chemi.muni.cz**

místní síť  
**wolfn.wolf.net**

veřejně **nedostupná** doména:  
**wolf.net**



server

pracovní stanice (výpočetní uzly)

**Operační systém:** CentOS 5

<http://www.centos.org/>



# Pravidla používání

**Klaster WOLF je určen výhradně pro účely výuky či pro vědeckou práci v rámci Národního centra pro výzkum biomolekul.**

## **Klaster WOLF je vysloveně zakázáno používat pro:**

- stahování nelegálního obsahu (autorsky chráněná díla apod.)
- rozesílání virů, spamů, nevyžádané pošty a podobných materiálů
- prolomení ochrany jiných počítačů
- čtení, mazání či změnu nechráněného obsahu souborů jiných uživatelů

## **Dodržují se pravidla užívání počítačové sítě Masarykovy univerzity:**

[https://is.muni.cz/auth/do/1499/normy/smernicerektora/Smernice\\_rektora\\_2-2003.pdf](https://is.muni.cz/auth/do/1499/normy/smernicerektora/Smernice_rektora_2-2003.pdf)



# Správci klastru

Než kontaktujete správce, prodiskutujte svůj problém s vaším kolegou či vyučujícím nebo školitelem.

## Hlavní správce klastru WOLF:

Petr Padrta ([padrta@chemi.muni.cz](mailto:padrta@chemi.muni.cz))

- zakládání nových účtů
- změna zapomenutých hesel
- správa hardware
- správa systému a systémových aplikací

## Správce aplikací pro vědecko-technické výpočty (moduly):

Petr Kulhánek ([kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz))

# Přihlašování

---

- Místní přihlášení
- Vzdálené přihlášení
- Export displeje
- Přihlašování z a do systému MS Windows
- Změna hesla



# Místní přihlášení

## Místní terminály:

- šest **textových terminálů** (F1 ... F6)
- jeden **grafický terminál** (F7)
- přepínání pomocí kláves Ctrl+Alt+F1 ... Ctrl+Alt+F7

## Přihlášení:

- nutno zadat **přihlašovací jméno** (login) a **heslo** (password)

**Přihlašovací jméno a heslo tvoří identitu uživatele a proto ji zásadně nesdělujeme třetím osobám, neukládáme či zasíláme (např. e-mailem) v nešifrované formě!**

Operační systém Linux je velmi flexibilní a umožňuje i jiné způsoby místního přihlášení, např. pomocí identifikačních karet či kryptografických klíčů. Více lze nalézt v nápovědě systému PAM (Pluggable Authentication Modules).

\$ man 8 pam





# Místní přihlášení

## Textový terminál (F1-F6):

- zpřístupňuje příkazovou řádku (CLI – command line interface)
- grafické aplikace (X11) lze spouštět jen s **exportem displeje** na vzdálený grafický terminál
- odhlášení příkazem **exit**

## Grafický terminál (F7):

- spouští X11 server a v něm okenního správce (KDE, GNOME, atd.)
- okenního správce (WM – window manager) **lze zvolit před** vlastním přihlášením
- umožňuje přímé spouštění grafických programů (GUI – graphical user interface)
- příkazová řádka je dostupná pomocí speciálních aplikací
  - xterm
  - **konsole**
- odhlášení **pomocí nabídky v menu** okenního správce

# Místní přihlášení

Grafický terminál (F7):



výběr okeního  
manažeru

# KDE

Koš

Nejčastěji používané aplikace

- Terminal
- Terminál
- KRandRTray
- NVIDIA X Server Settings

Všechny aplikace

- Dokumentace
- Grafika
- Internet
- Kancelář**
  - Kontakt
  - J-Pilot
  - KAddressBook
  - Kalendář
  - KBibTeX
  - KOrganizer
  - LyX-qt
  - Prezentace
  - Scribus
  - Sešit
  - Správa projektů
  - Textový processor
  - Úkoly
  - Vývoj databáze
- Multimédia
- Nastavení
- Nástroje
- Správa
- Systém
- Vývoj
- Wine
- Ztráty a nálezy
- Můj adresář
- Najít soubory/složky
- Nápověda
- Ovládací centrum

Činnosti

- Spustit...
- Uzamknout relaci
- Odhlásit se...

kulhanek@wolf16:~ - Shell - Konsole

Relace Úpravy Pohled Záložky Nastavení Nápověda

[kulhanek@wolf16 ~]\$

Shell

Centos 5

16:35  
neděle  
2008-10-19

# GNOME

Aplikace Místa System 16:38

Grafika  
Internet  
Kancelář  
Programování  
Příslušenství  
Systémové nástroje  
Wine  
Zvuk a video  
Přidat/Odebrat Software

J-Pilot  
KAddressBook  
Kalendář  
KBibTeX  
KOrganizer  
LyX-qt  
Presentace  
Scribus  
Sešit  
Správa projektů  
Textový processor  
Úkoly  
Vývoj databáze

kulhanek@wolf16:~  
Soubor Upravit Zobrazit Terminál Karty nápověda  
[kulhanek@wolf16 ~]\$ ksnapshot

kulhanek  
Soubor Upravit Zobrazit Místa nápověda  
Desktop tmp kde.png  
kulhanek 3 položky, Volné místo: 354,8 GB

CentOS 5

USA



# Vzdálené přihlášení

Existuje několik možností vzdáleného přihlášení (rsh, XDMCP, apod.) avšak nepoužívanějším a **nejbezpečnějším** je použití příkazu **ssh** (secure shell).

## Syntaxe:

\$ **ssh** [user@]hostname [command]                    [] - možno vynechat

jméno uživatele;  
pokud není uvedeno,  
použije se jméno  
přihlášeného uživatele

jméno počítače

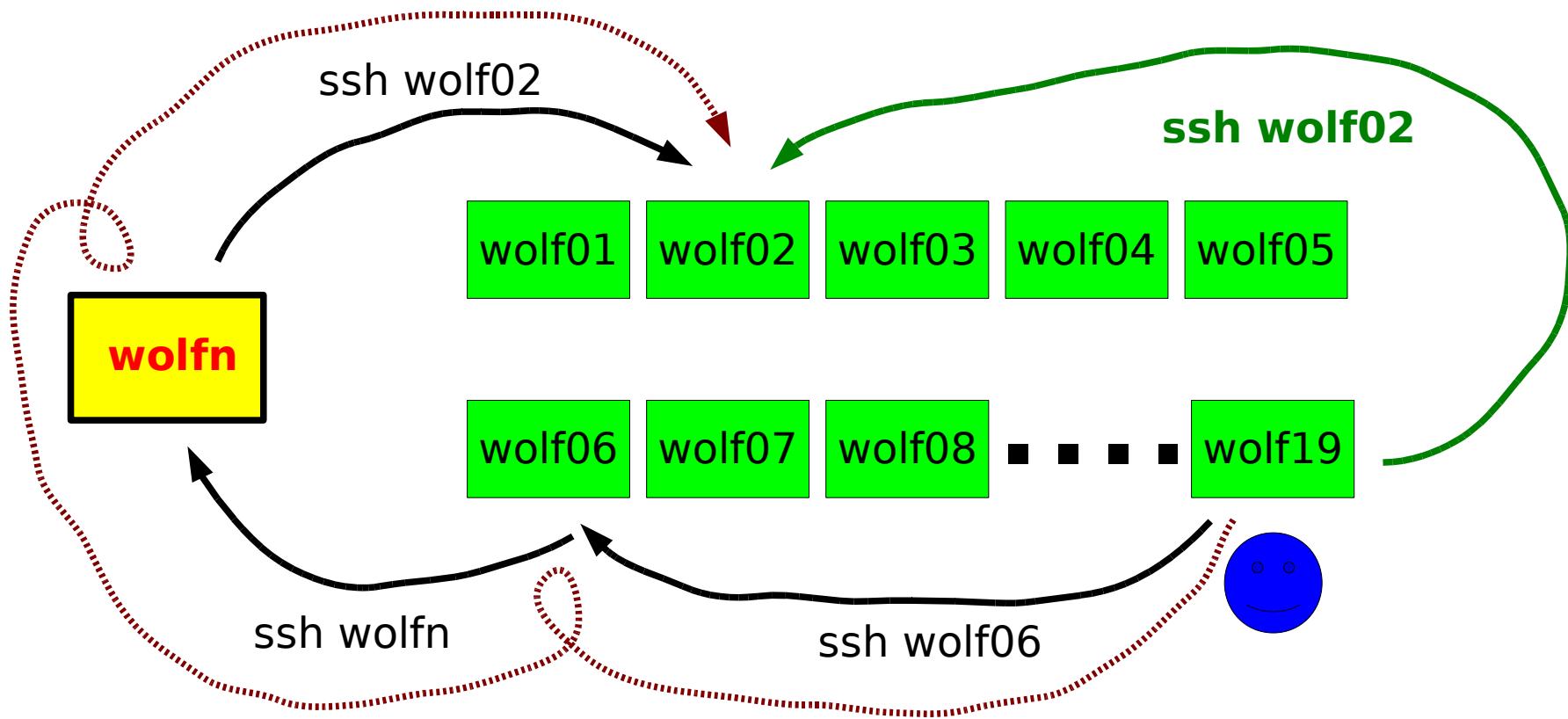
příkaz, který se má vykonat;  
pokud není uveden, zpřístupní  
se příkazová řádka

## Příklady použití:

\$ ssh wolfn.wolf.net

# Vzdálené přihlášení

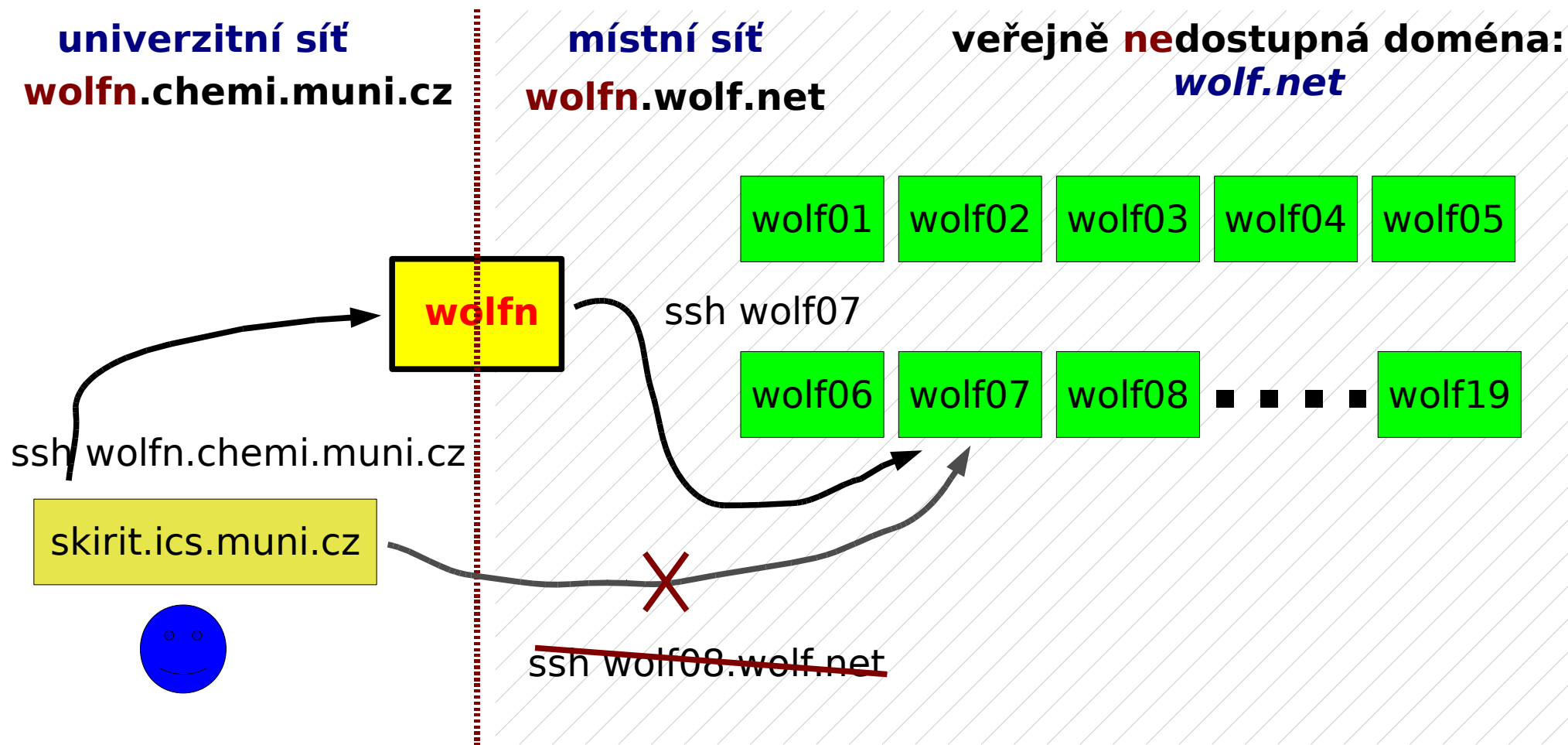
Pomocí příkazu ssh je možné provést **vnořené vzdálené přihlášení**.



S každou novou úrovní vzdáleného přihlášení **roste režie**, proto, pokud je to možné, použijeme **nejpřímější vzdálené přihlášení**.

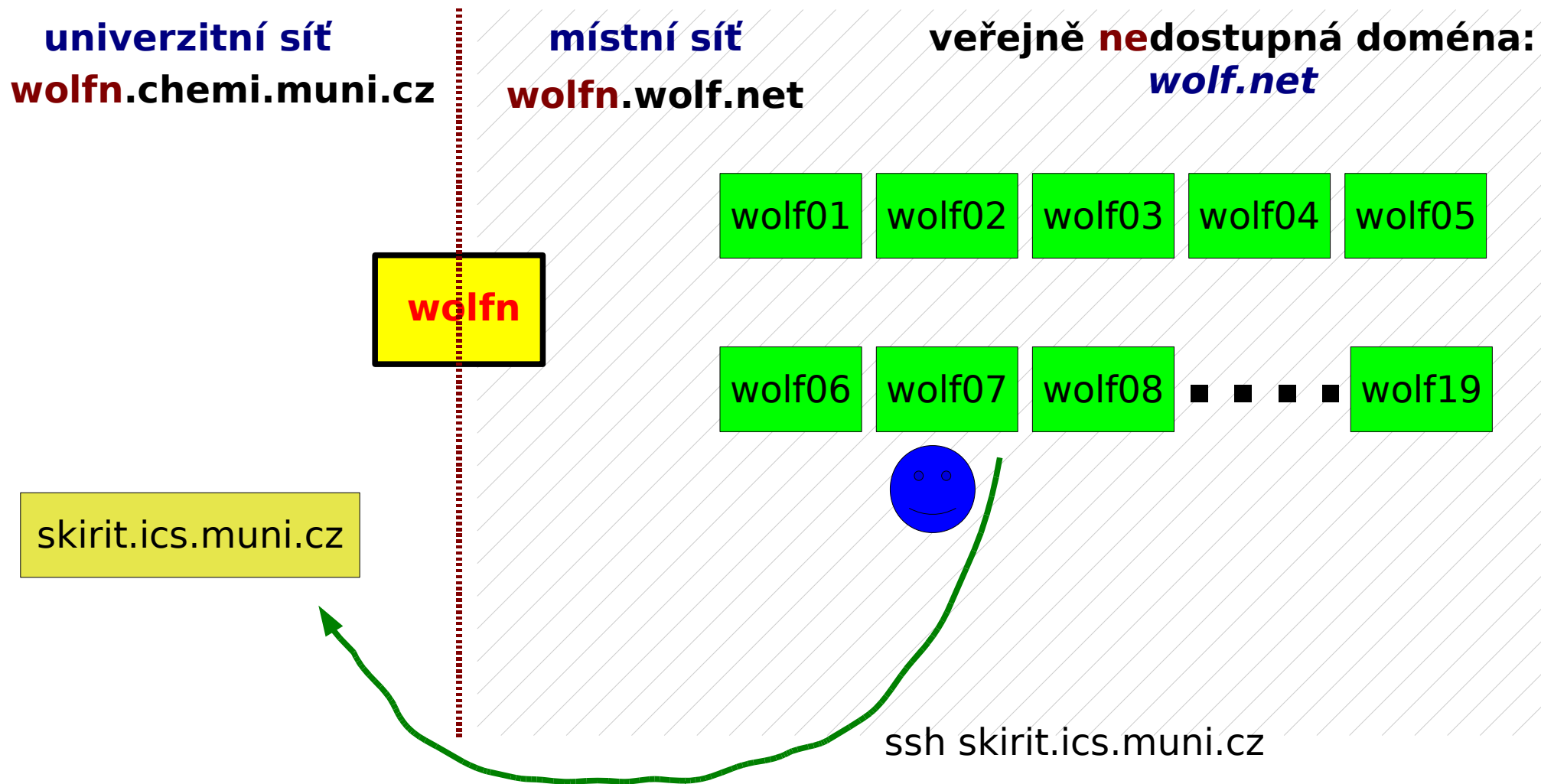
# Vzdálené přihlášení

Vnořené vzdálené přihlášení je **nutné použít** pro přístup k počítačům v místních neveřejných sítích.



# Vzdálené přihlášení

Vnořené vzdálené přihlášení je **nutné použít** pro přístup k počítačům v místních sítích.

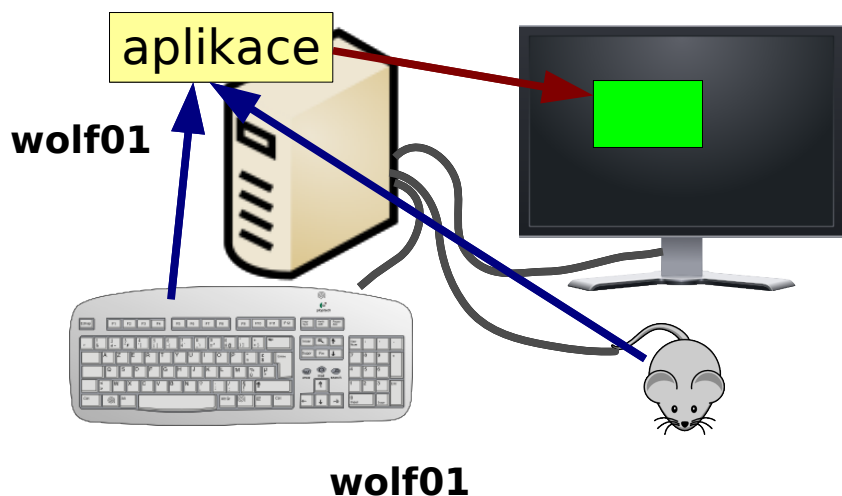




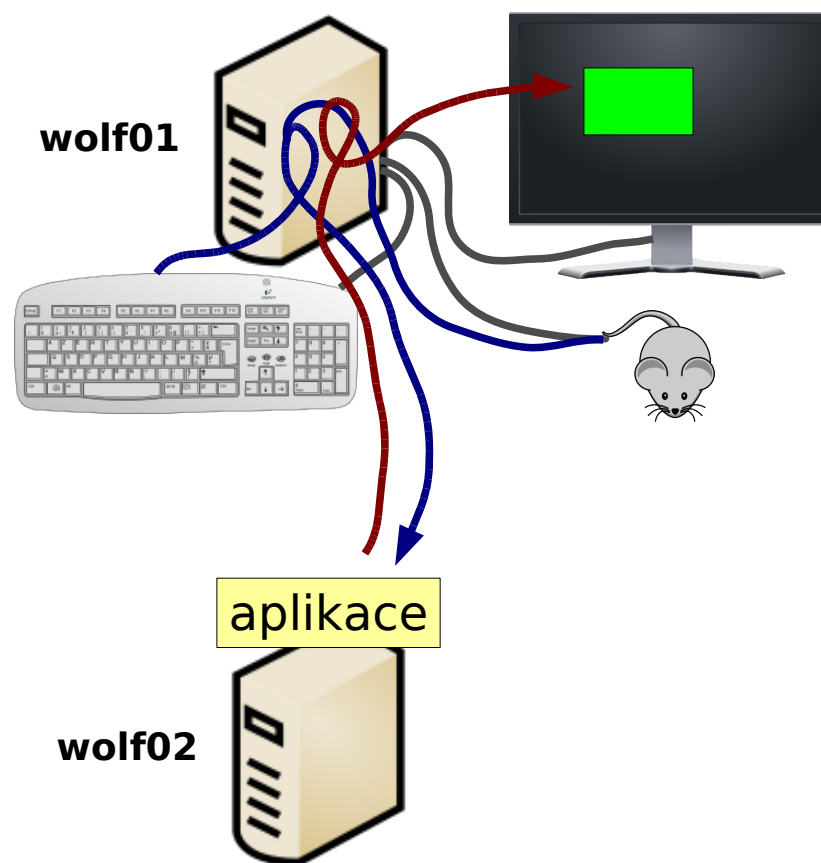
# Export displeje

Grafické aplikace je možné spouštět přímo v prostředí X11 (grafickém terminálu) nebo s exportem displeje na vzdálenou plochu prostředí X11.

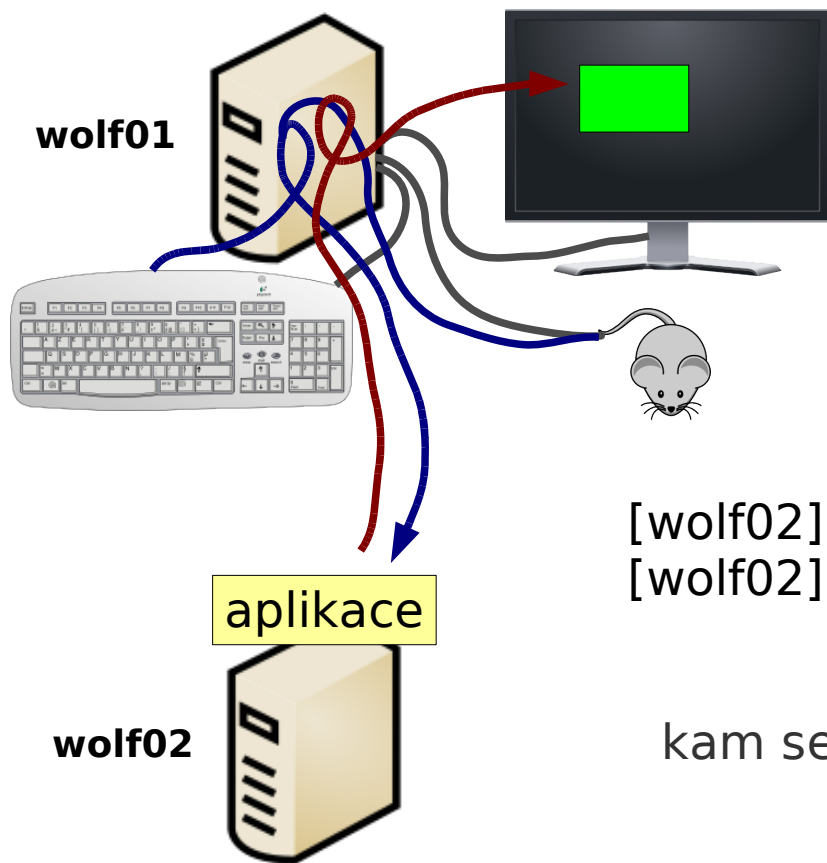
## Přímé spuštění



## Export displeje



# Export displeje



```
[wolf01] $ xhost wolf02.wolf.net
```

povolí používat místní X server  
ze stroje wolf02

```
[wolf02] $ export DISPLAY=wolf01.wolf.net:0.0  
[wolf02] $ ./my_application
```

kam se má displej exportovat

číslo obrazovky

číslo desktopu

Pokud se na stroj wolf02 přihlašujeme ze stroje wolf01 pomocí příkazu **ssh**, celá procedura proběhne automaticky při použití volby **-X**.

```
[wolf01] $ ssh -X wolf02  
[wolf02] $ ./my_application
```

## Přihlašování do Linuxu z MS Windows (textový terminál):

- **putty** (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>)
- **ssh** (např. z prostředí Cygwin; <http://www.cygwin.com/>)

## Export displeje z Linuxu do MS Windows (X11 server):

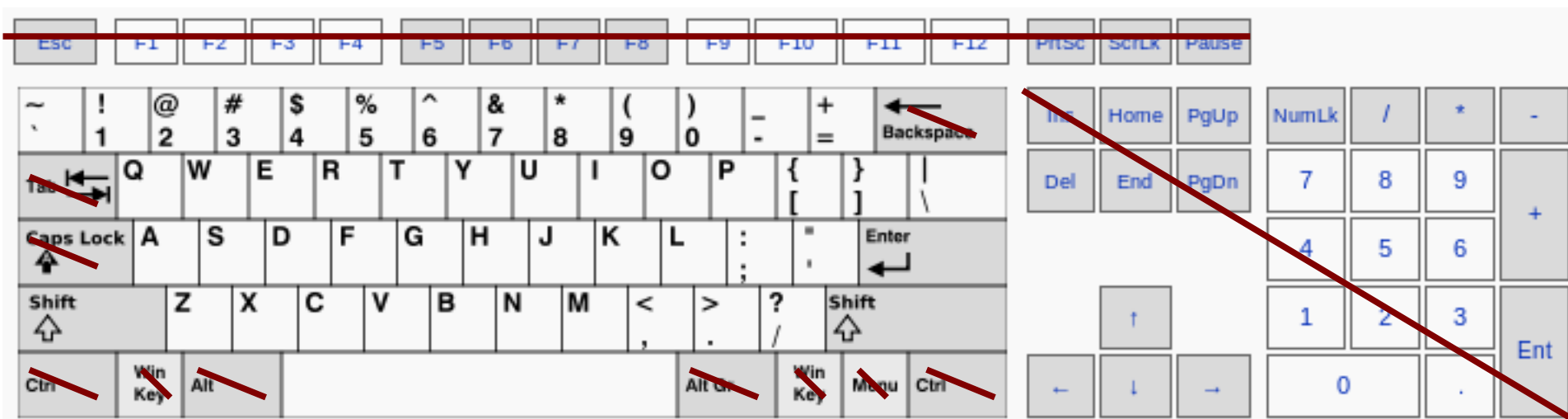
- **Xming** (<http://sourceforge.net/projects/xming/>)
- **cygwin** (<http://www.cygwin.com/>)
- celá řada komerčních řešení

## Přihlašování z Linuxu do MS Windows (vzdálená plocha):

- **rdesktop**

# Změna hesla

- Heslo se mění po přihlášení do systému z příkazové řádky pomocí příkazu **passwd**.
- Po jeho spuštění jste dotázáni na **současné platné heslo** a poté na **heslo nové**. Nové heslo se je nutné zadat dvakrát, aby se omezily možné překlepy.
- Změna se projeví na **všech počítačích** klastru WOLF.
- Heslo by mělo být dostatečně silné. Mělo by obsahovat kombinaci písmen (malých a velkých), číslic a speciálních znaků.
- Při zadávání hesla nepoužíváme numerickou část klávesnice, speciální klávesy a přemykače kromě klávesy Shift. Vyvarujete se tak problémům se zadáváním hesla na jiných počítačích.





# Nebezpečné trojhmoty

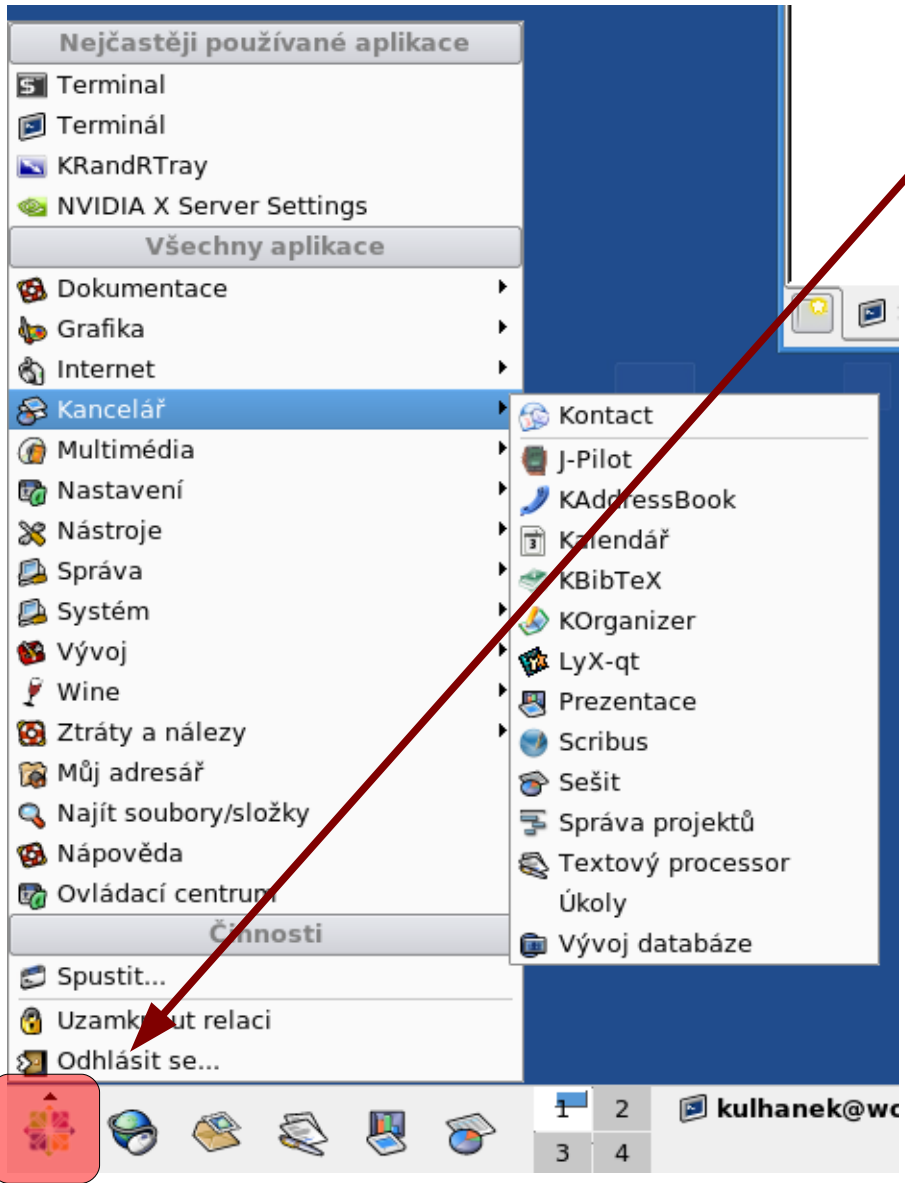
- **Ctrl + Alt + Del** (odhlásí uživatele nebo provede měkký restart počítače)
- **Ctrl + Alt + Backspace** (provede **tvrdý** reset X serveru)

# Programové vybavení

---

- **Systemové aplikace**
- **Vědecko-technické aplikace (moduly)**

# Systemové aplikace



Aplikace dostupné v menu okenního manažeru. Aplikace lze spouštět i z příkazové řádky.

**Vědeckotechnické aplikace**, které jsou instalovány v několika verzích (verze aplikace, typ kompilace, paralelní verze), jsou dostupné ve formě **modulů**. Před použitím aplikace je nutné příslušný modul aktivovat.

**Přehled dostupných aplikací:**

\$ **module**

**Přehled dostupných verzí aplikace:**

\$ **module versions amber**

**Aktivace aplikace:**

\$ **module add amber**

**Spustění aplikace z modulu **amberu**:**

\$ **xlaep**



# Textové editory

---

- **vi (vim)**
- **grafické textové editory (kwrite, kate, gedit, emacs, nedit atd.)**

**Standardní textový editor** v operačních systémech UNIXového typu. Pracuje pouze v textovém módu a jeho používání je **netriviální**.

Je vhodné se naučit, jak otevřít soubor, přejít do editačního módu, upravovat text, uložit provedené změny a editor ukončit.

### **Dokumentace:**

- manuálové stránky (man vi)
- příložený přehled příkazů

# vi - základy

## Pracovní módy editoru



## Spuštění editoru

**vi** start editoru  
**vi filename** start editoru a **otevření souboru** filename

## Ukončení editoru

**:q** ukončení editoru  
**:q!** ukončení editoru bez uložení změn  
**:w** uložení souboru  
**:w filename** uložení souboru po jménem *filename*  
**:wq** ukončení s uložením souboru

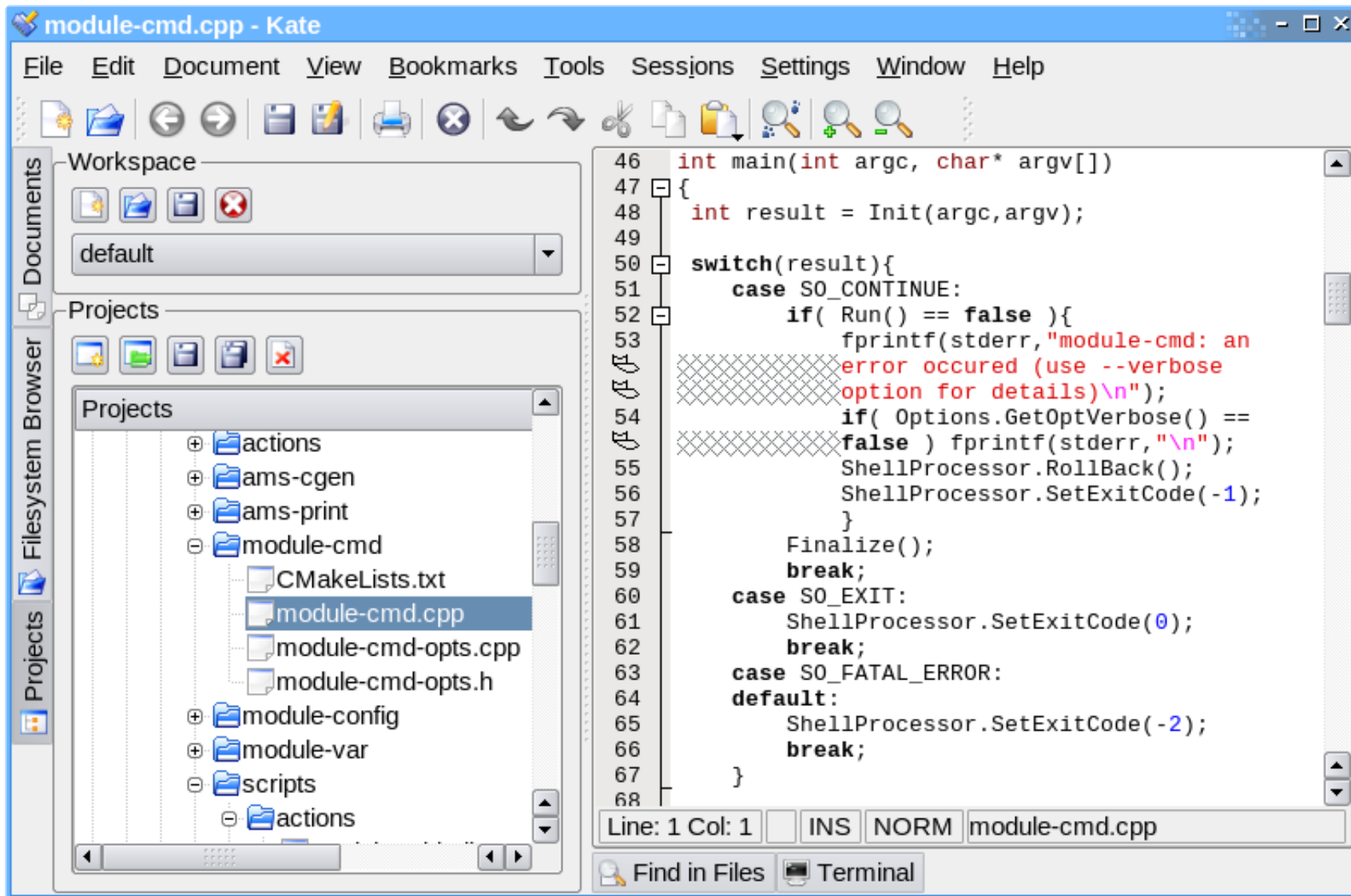
## Změny souboru

**i** text bude vkládán **od** pozice kurzoru  
**a** text bude vkládán **za** pozici kurzoru

# kwwrite

```
abf_core.f90 - KWrite
File Edit View Bookmarks Tools Settings Help
[Icons]
217 p_cvalue2 => p_cvalue3
218 p_cvalue3 => p_tmp
219
220 ! save coordinate value to history
221 do i=1,fnitem
222     p_cvalue3(i) = free_item_list(i)%value
223 end do
224
225 ! calculate abf force to be applied -----
226 la(:) = 0.0d0
227
228 if( gi .gt. 0 ) then
229     ! get scaling factor for ABF force
230     sc_ramp = abf_ramp_scaling(gi)
231     do i=1,fnitem
232         if( accumulator%nsamples(gi) .gt. 0 ) then
233             ! we are inside of ABF region - calculate force
234             la(i) = sc_ramp * accumulator%abfforce(i,gi)/(real(accumulator%nsamples(gi)))
235         end if
236     end do
237 end if
238
239 ! project abf force along coordinate -----
240
241 do j=1,fcatom
242     imass = fmass_inv(j)
243     do m=1,3
244         v = 0.0d0
245         do i=1,fnitem
```

Line: 1 Col: 1 INS NORM

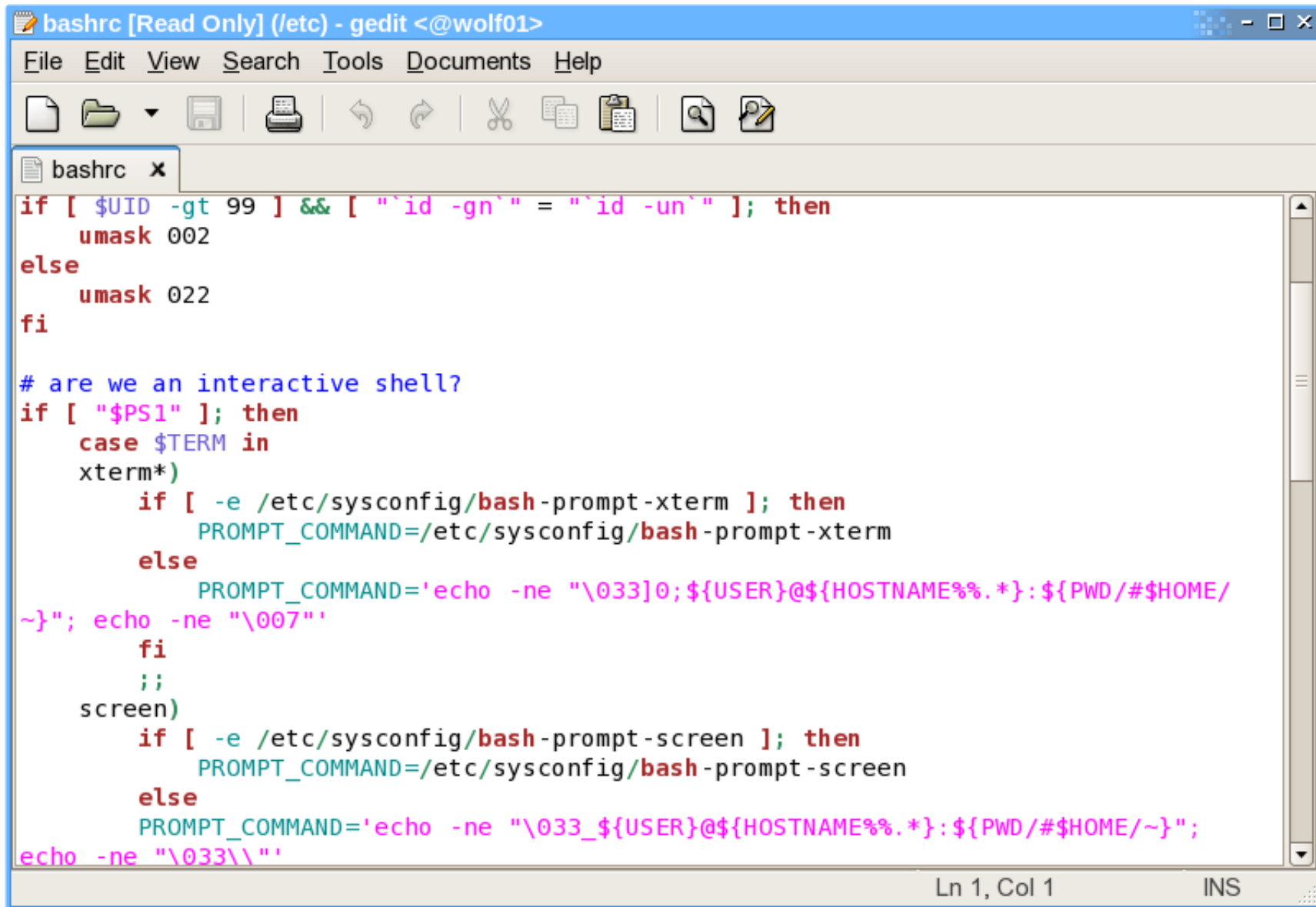


```
46 int main(int argc, char* argv[])
47 {
48     int result = Init(argc,argv);
49
50     switch(result){
51         case S0_CONTINUE:
52             if( Run() == false ){
53                 fprintf(stderr,"module-cmd: an
XXXXXXXXXXerror ocured (use --verbose
XXXXXXXXXXoption for details)\n");
54                 if( Options.GetOptVerbose() ==
XXXXXXXXXXfalse ) fprintf(stderr,"\n");
55                 ShellProcessor.Rollback();
56                 ShellProcessor.SetExitCode(-1);
57             }
58             Finalize();
59             break;
60         case S0_EXIT:
61             ShellProcessor.SetExitCode(0);
62             break;
63         case S0_FATAL_ERROR:
64         default:
65             ShellProcessor.SetExitCode(-2);
66             break;
67     }
68 }
```

Line: 1 Col: 1 | INS | NORM | module-cmd.cpp

Find in Files | Terminal

# gedit



```
bashrc [Read Only] (/etc) - gedit <@wolf01>
File Edit View Search Tools Documents Help
bashrc x
if [ $UID -gt 99 ] && [ "`id -gn`" = "`id -un`" ]; then
    umask 002
else
    umask 022
fi

# are we an interactive shell?
if [ "$PS1" ]; then
    case $TERM in
    xterm*)
        if [ -e /etc/sysconfig/bash-prompt-xterm ]; then
            PROMPT_COMMAND=/etc/sysconfig/bash-prompt-xterm
        else
            PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME%.*}:${PWD/#$HOME/~}"; echo -ne "\007"'
        fi
        ;;
    screen)
        if [ -e /etc/sysconfig/bash-prompt-screen ]; then
            PROMPT_COMMAND=/etc/sysconfig/bash-prompt-screen
        else
            PROMPT_COMMAND='echo -ne "\033_${USER}@${HOSTNAME%.*}:${PWD/#$HOME/~}";
echo -ne "\033\\""
        fi
    esac
fi
```

Ln 1, Col 1      INS

# Příkazová řádka

---

- Terminály
- Struktura příkazové řádky
- Historie a automatické doplňování

# Terminály

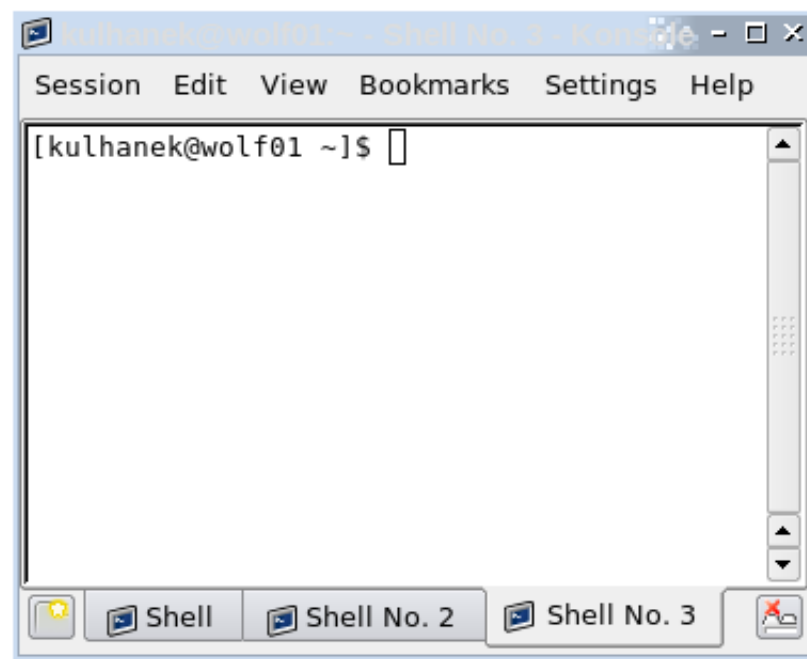
Příkazová řádka je přístupná přímo z textových terminálů. V grafickém prostředí X11 je nutné spustit vhodnou aplikaci emulující textový terminál.

## xterm



jednoduché, standard na všech UNIXových systémech

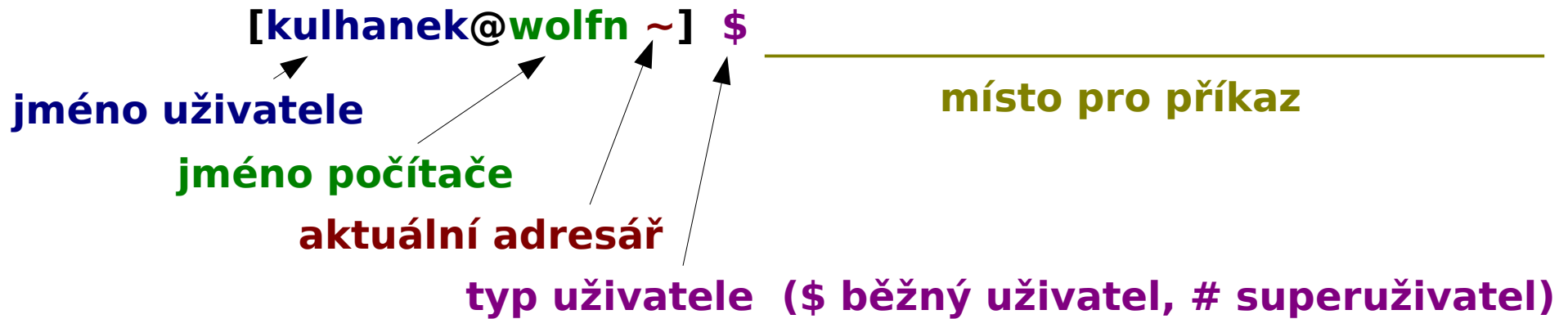
## konsole



jednoduché přitom značně konfigurovatelné, dostupné v prostředí KDE



# Struktura příkazové řádky



Příkaz se vykoná zmáčknutím klávesy **Enter**.

Na příkazové řádce lze uvést několik příkazů, je však nutné je oddělit znakem ;.

```
[kulhanek@wolfn ~]$ echo "jedna"; echo "dve"; echo "tri"  
jedna  
dve  
tri  
[kulhanek@wolfn ~]$
```

Formát příkazové řádky lze změnit, viz. man bash (kapitola PROMPTING).



# Historie a doplňování

**Historie:** pomocí kurzorových šipek nahoru a dolů lze procházet seznamem již zadaných příkazů. Příkaz z historie lze znovu použít nebo upravit a upravený použít.

**Automatické doplňování:** zmáčknutím klávesy Tab (tabulátor) se interpret příkazové řádky snaží dokončit rozepsané slovo. Doplňují se jména příkazů, cesty a jména souborů ...

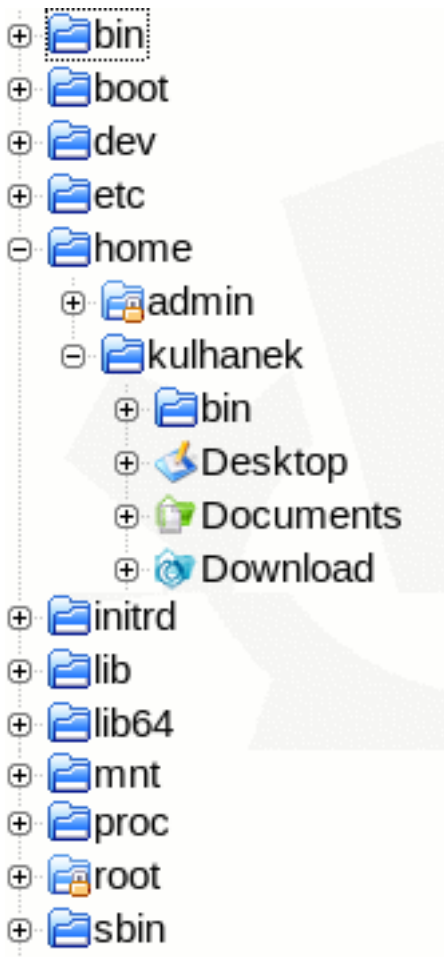
# Souborový systém

---

- **Struktura souborového systému**
- **Absolutní a relativní cesty**
- **Přístupová práva**
- **Speciální soubory**
- **Diskové oddíly a souborový systém**

# Struktura

Linux (jako UNIXový operační systém) využívá **hierarchický adresářový souborový systém** složený z adresářů a souborů. Všechny adresáře a soubory leží v **jediném kořenovém adresáři**.



## Srovnání s MS Windows:

- jednotlivé diskové oddíly (C:, D: apod.) jsou připojeny jako podadresáře
- CD-ROM a jiné zařízení (USB disky) se opět připojují jako podadresáře

## Domovský adresář:

- adresář sloužící pro **data a nastavení uživatele**
- obvykle v cestě **/home/jmeno\_uzivatele** (k určení cesty je však lepší používat tildu ~, viz. absolutní cesty)

**Cesta** k adresáři nebo souboru může být uvedena jako **absolutní** nebo **relativní**. Jména adresářů a souborů se oddělují **lomítkem /**.

**Absolutní cesta** je vždy uvedena vzhledem ke kořenovému nebo k domovskému adresáři. Musí tedy začínat buď lomítkem **/** nebo tildou **~**.

## Příklady absolutních cest:

`/home/kulhanek/Documents`

`/home/kulhanek/Documents/domaci_ukol.txt`

`~/Documents` → `/home/kulhanek/Documents`

`~alois/Documents` → `/home/alois/Documents`

## Použití tildy:

`~` domovský adresář přihlášeného uživatele

`~user_name` domovský adresář uživatele `user_name`

**Relativní cesta** je cesta uvedená k aktuálnímu pracovnímu adresáři.  
(Absolutní cestu k pracovnímu adresáři lze získat příkazem pwd.)

## Příklady relativních cest:

```
Documents/domaci_ukol.txt
```

```
../alois/Documents
```

```
./muj_script
```

## Speciální jména adresářů:

- .** (tečka)            aktuální adresář (nemá žádnou spojitost s pracovním adresářem)
- ..** (dvě tečky)      rodičovský adresář

# Přístupová práva

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

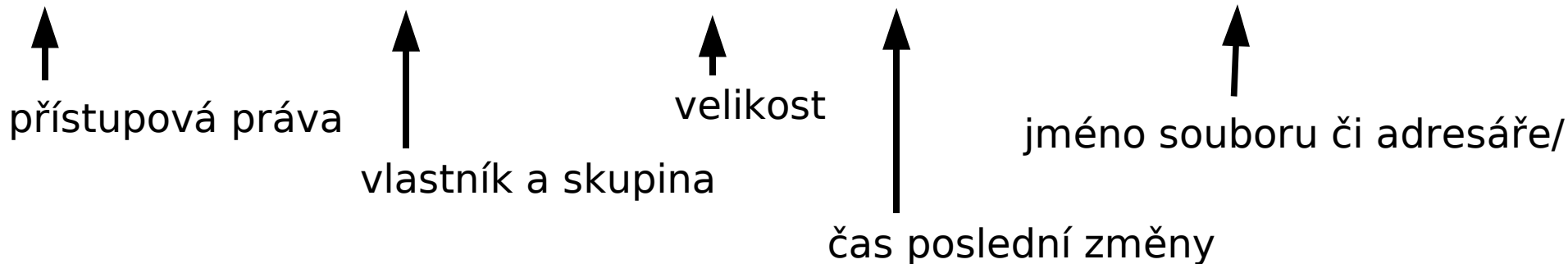
## Přístupová práva:

<b>r</b>	možnost číst soubor	vypsát obsah adresáře
<b>w</b>	možnost měnit soubor	změnit obsah adresáře
<b>x</b>	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
```





# Speciální soubory

Některé soubory v souborovém systému mají **speciální charakter**. Jedná se například o přístupové body k hardware počítače (obsah adresáře **/dev**). Některé soubory obsahují informace o aktuálním stavu operačního systému (obsah adresáře **/proc**) a podobně.

## Užitečné speciální soubory:

**/dev/null** soubor poskytuje prázdný obsah, veškerý obsah do něj zapsaný je okamžitě odstraněn vhodný pro přesměrování nechtěného standardního vstupu či výstupu

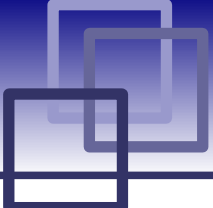
**/proc/cpuinfo** informace o dostupných procesorech

## Příklad použití:

```
$ cat /proc/cpuinfo
```

```
$ ls 2> /dev/null
```





# Diskové oddíly

# Příkazy

---

- **Manuálové stránky**
- **Přehled příkazů**

# Příkazy - nápověda

## Manuálové stránky (aneb co dělat, když si nevím rady):

- `man` vypíše manuálovou stránku příkazu

```
$ man [section_number] topic
```



jméno příkazu, funkce, tématu, kapitoly apod.

## Dostupné sekce:

- *Section 1* user commands
- *Section 2* system calls
- *Section 3* library functions
- *Section 4* special files
- *Section 5* file formats
- *Section 6* games
- *Section 7* conventions and miscellany
- *Section 8* administration and privileged commands
- *Section L* math library functions
- *Section N* tcl functions

Číslo sekce je nutné udávat u témat se stejným jménem zařazených do různých sekcí.

```
$ man 1 printf      manuálová stránka příkazu printf
```

```
$ man 3 printf      manuálová stránka funkce printf() jazyka C
```



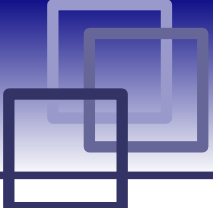
# Příkazy - nápověda

## ***Navigace v textu nápovědy:***

- posun v textu po řádcích (kurzorové šipky nahoru a dolů nebo klávesy **j** a **k**)
- posun v textu po stránkách (**PgDn** a **PgUp** nebo klávesy **f** a **b**)
- vyhledávání ( **/hledaný\_text** , klávesa **n** pro další vyhledávání )
- zavření nápovědy (klávesa **q**)

## ***On-line manuálové stránky ve formátu HTML:***

<http://linux.die.net/man/>



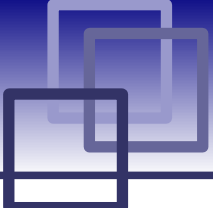
# Příkazy - přehled

## **Zjišťovací příkazy:**

- `uname` vypíše informace o systému
- `date` vypíše aktuální čas a datum
- `hostname` vypíše jméno počítače
- `who` vypíše, kdo je přihlášen do systému
- `id` vypíše identifikační čísla uživatele
- `whoami` vypíše jméno přihlášeného uživatele

## **Různé:**

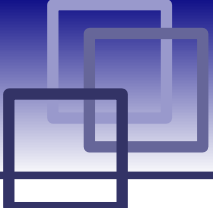
- `echo` vypíše text
- `printf` vypíše formátovaný text
- `type` vypíše cestu k příkazu
- `read` načte text ze standardního vstupu a uloží do proměnné
- `dirname` vyextrahuje jméno adresáře z úplného jména souboru
- `basename` vyextrahuje jméno souboru z úplného jména souboru



# Příkazy - přehled

## **Manipulace s textovými soubory:**

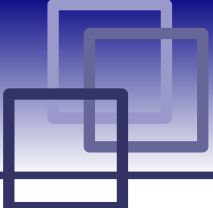
- cat vypíše obsah souboru
- more vypíše obsah souboru po stránkách
- paste spojí obsah více souborů do jednoho (vedle sebe)
- cut vypíše zvolený sloupec ze souboru
- tr transformace souboru
- wc informace o souboru (počet řádků, slov a znaků)
- grep vypíše řádky ze souboru obsahující klíčové slovo
- head vypíše úvodní část souboru
- tail vypíše koncovou část souboru
- sed neinteraktivní editor
- sort setřídí obsah souboru
- awk interpret jazyka pro manipulaci s textovými soubory



# Příkazy - přehled

## ***Souborový systém:***

- pwd vypíše cestu aktuálního pracovního adresáře
- cd změní aktuální pracovní adresář
- mkdir vytvoří adresář
- rmdir odstraní adresář
- ls vypíše obsah adresáře
- du vypíše velikost adresáře
- cp zkopíruje soubor či adresář
- mv přesune soubor či adresář
- rm odstraní soubor či adresář
- stat vypíše informace o souboru či adresáři
- find nalezne soubor
- df vypíše informace o připojených diskových oddílech



# Příkazy - přehled

## **Procesy:**

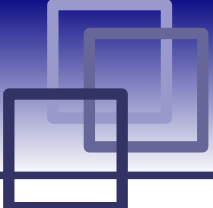
- top informace o běžících procesech v systému
- ps vypíše informace o běžících procesech v systému
- kill zašle signál procesu
- nohup spustí process bez interakce s terminálem
- time vypíše délku běhu procesu
- wait čeká na dokončení procesů na pozadí
- ssh spustí příkaz na vzdáleném počítači



# Procesy

---

- **Procesy, multiprocessing, procesory**
- **Komunikace s procesy**
- **Přesměrování**
- **Roury**



# Procesy

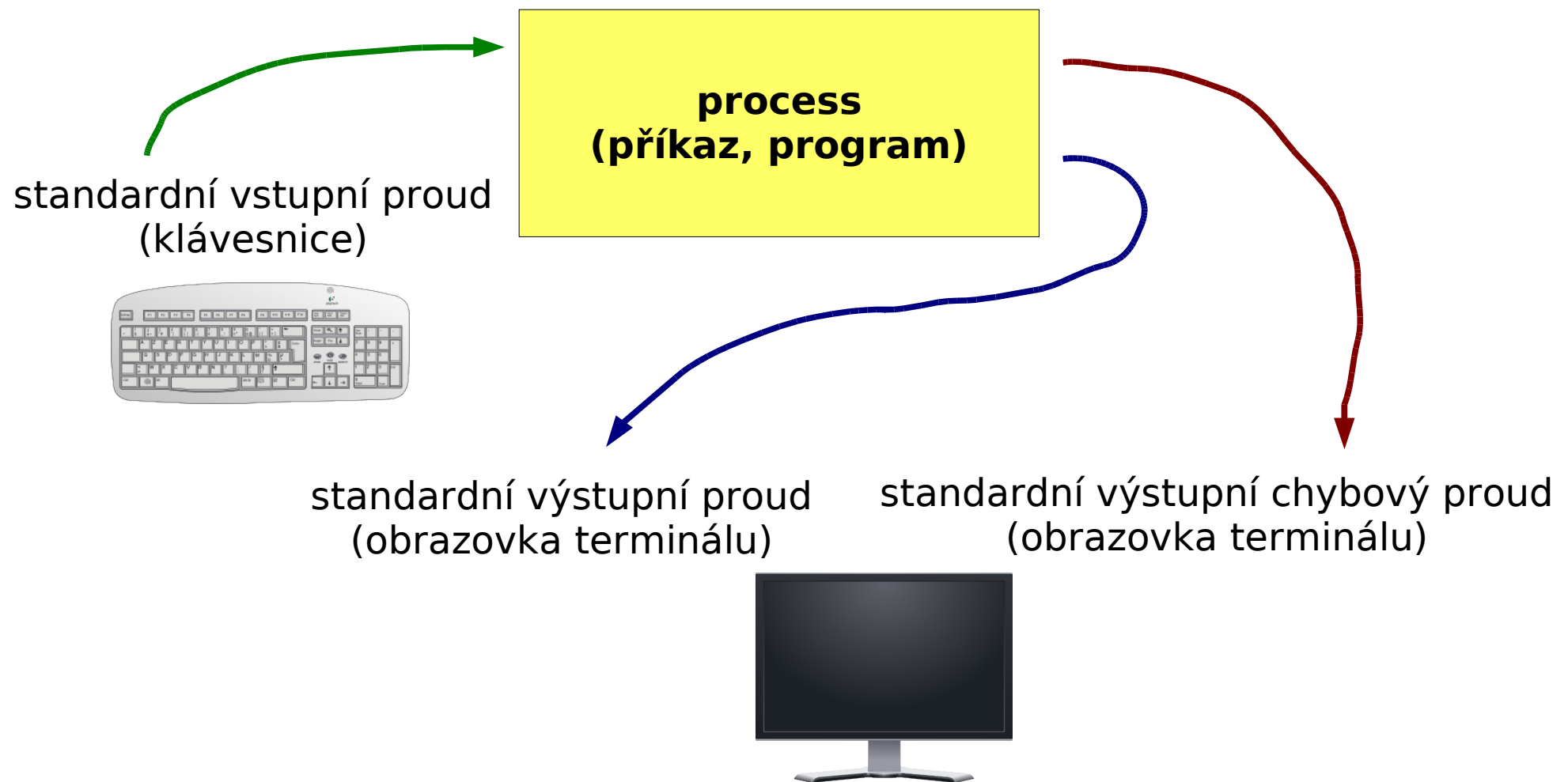


# Komunikace s procesy

# Standardní proudy

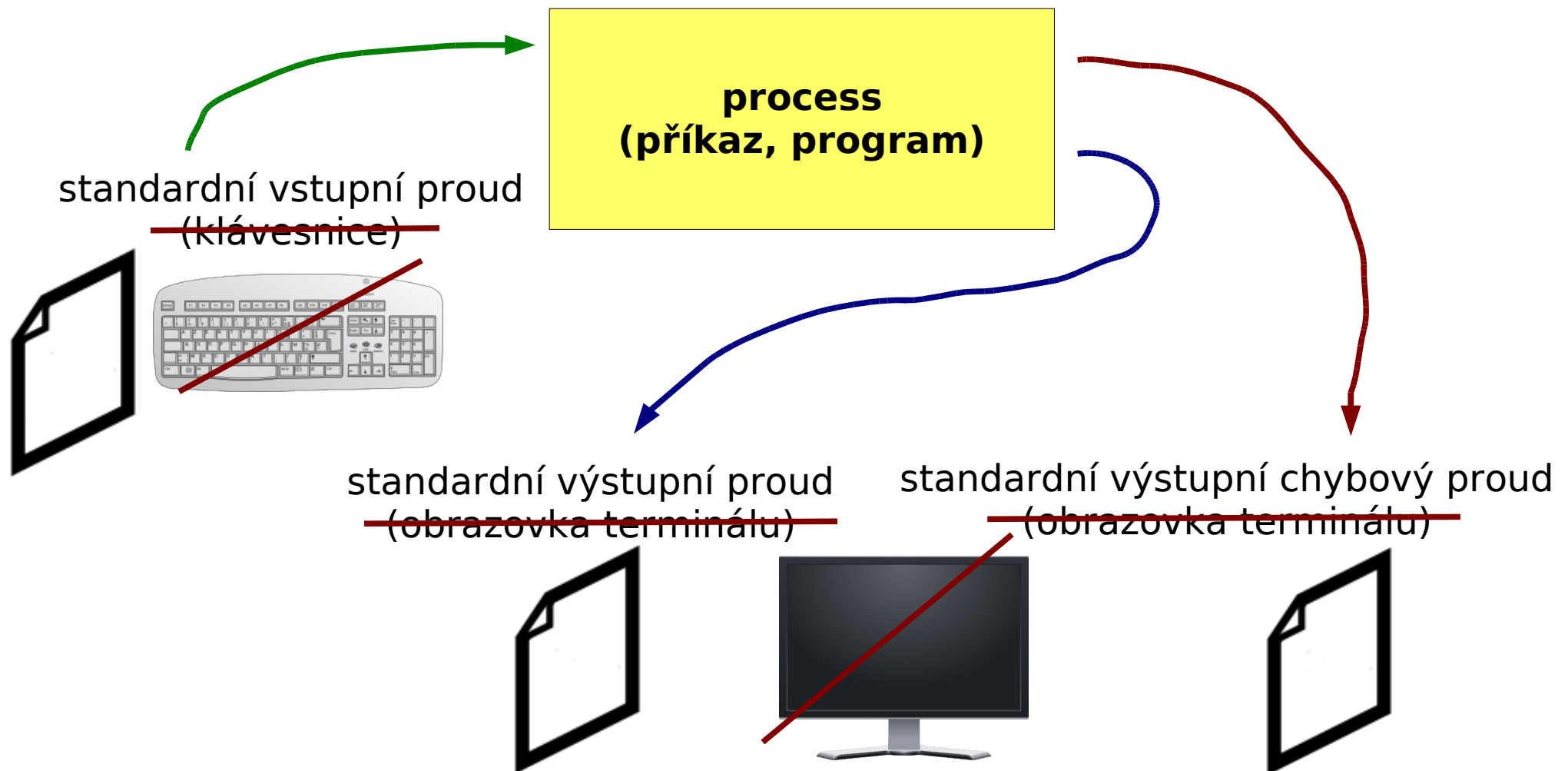
**Vstupně-výstupní proudy** slouží procesu ke **komunikaci** se svým okolím.

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  
druha radka textu  
treti radka textu  
EOF  
.....
```

značka určující konec vstupu  
(volí uživatel)

text, který tvoří načítaný vstup

konec vstupu, značku  
**nesmí** obklopovat 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 **přesměrovat** 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.

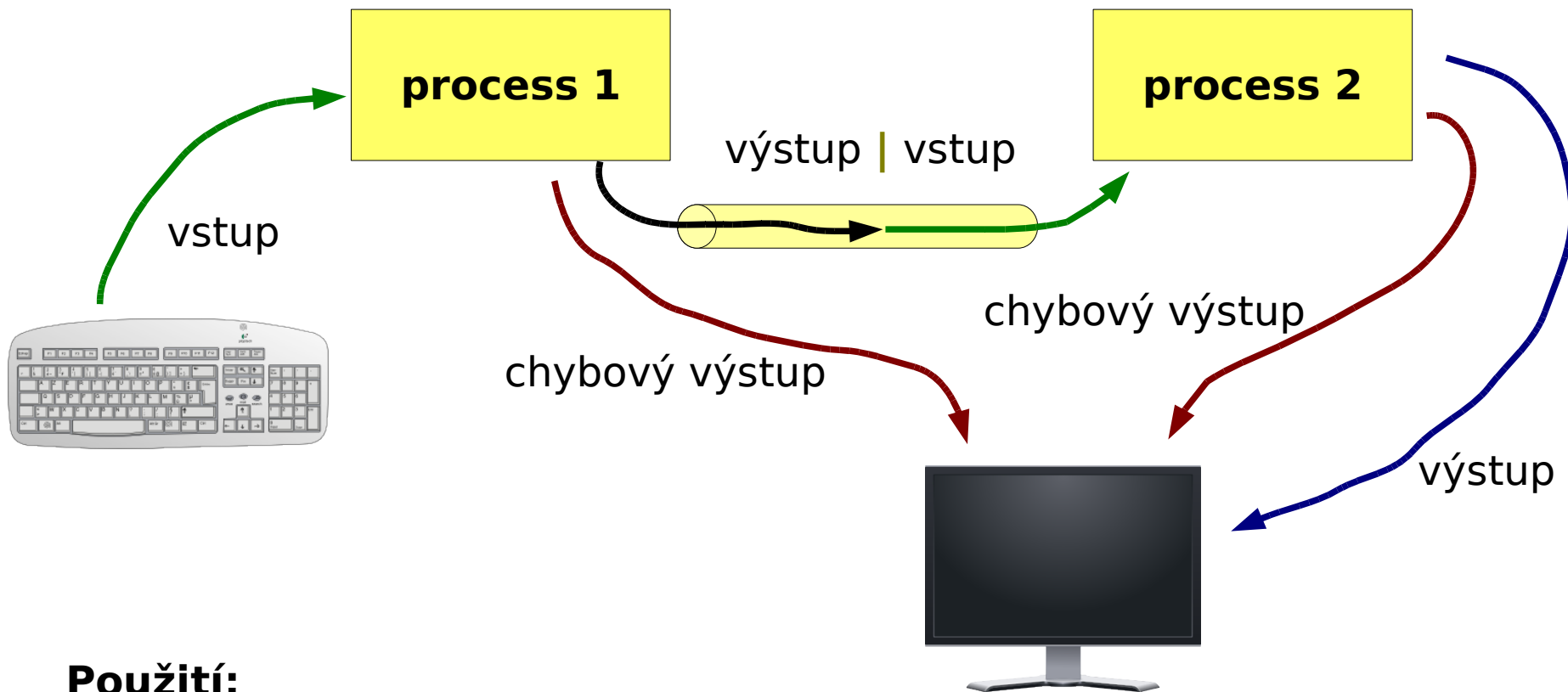
```
$ my_command >> output.txt 2>&1      pořadí je důležité!
```

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



# Roury (pípy)

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

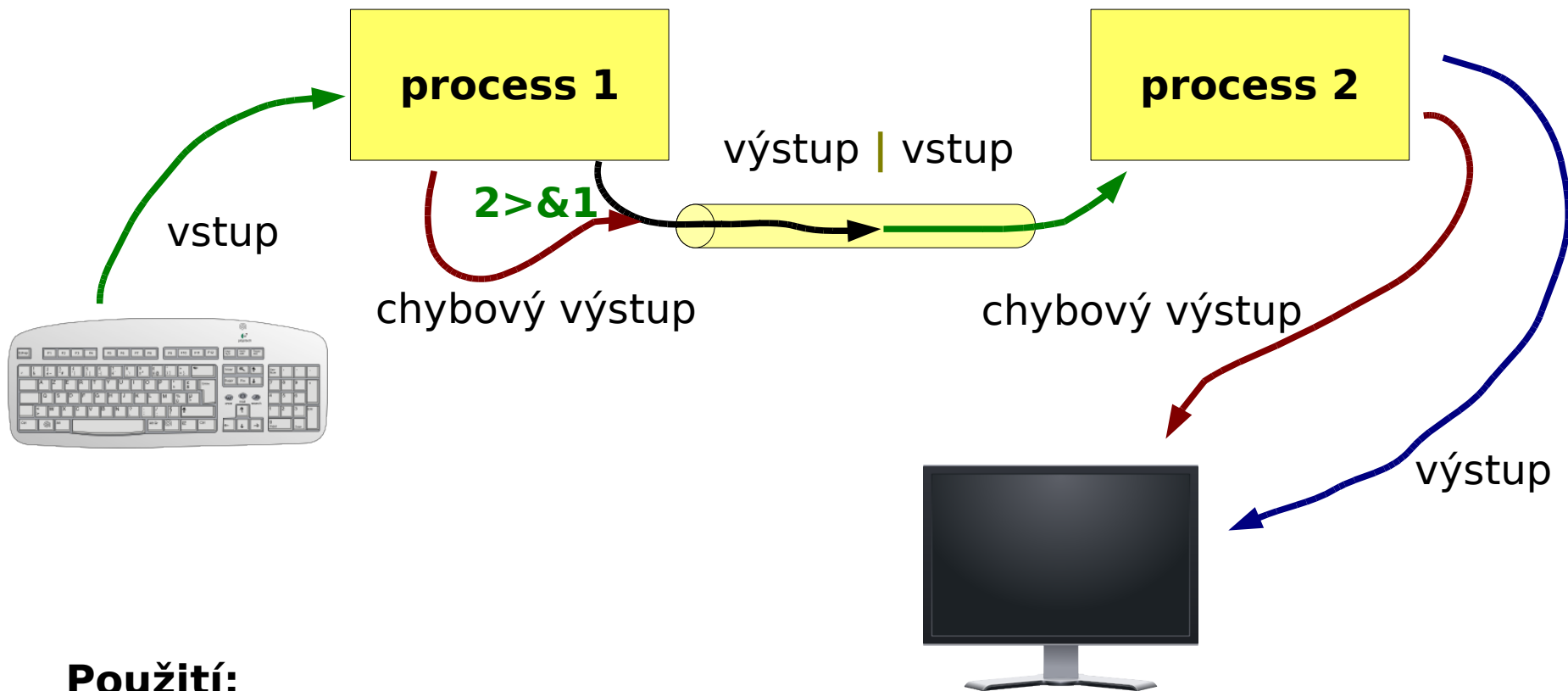


## Použití:

```
$ 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
```

# Tipy a triky

---

- **aneb užitečné kombinace příkazů**

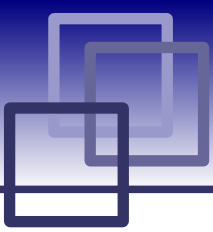


# Tipy a triky

# Literatura

---

- zdroje užitečných informací



# Literatura

# Poděkování

---