

# C2110

# *Operační systém UNIX a základy programování*

## 2. lekce

Petr Kulhánek

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

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

# Obsah

- **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

# 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

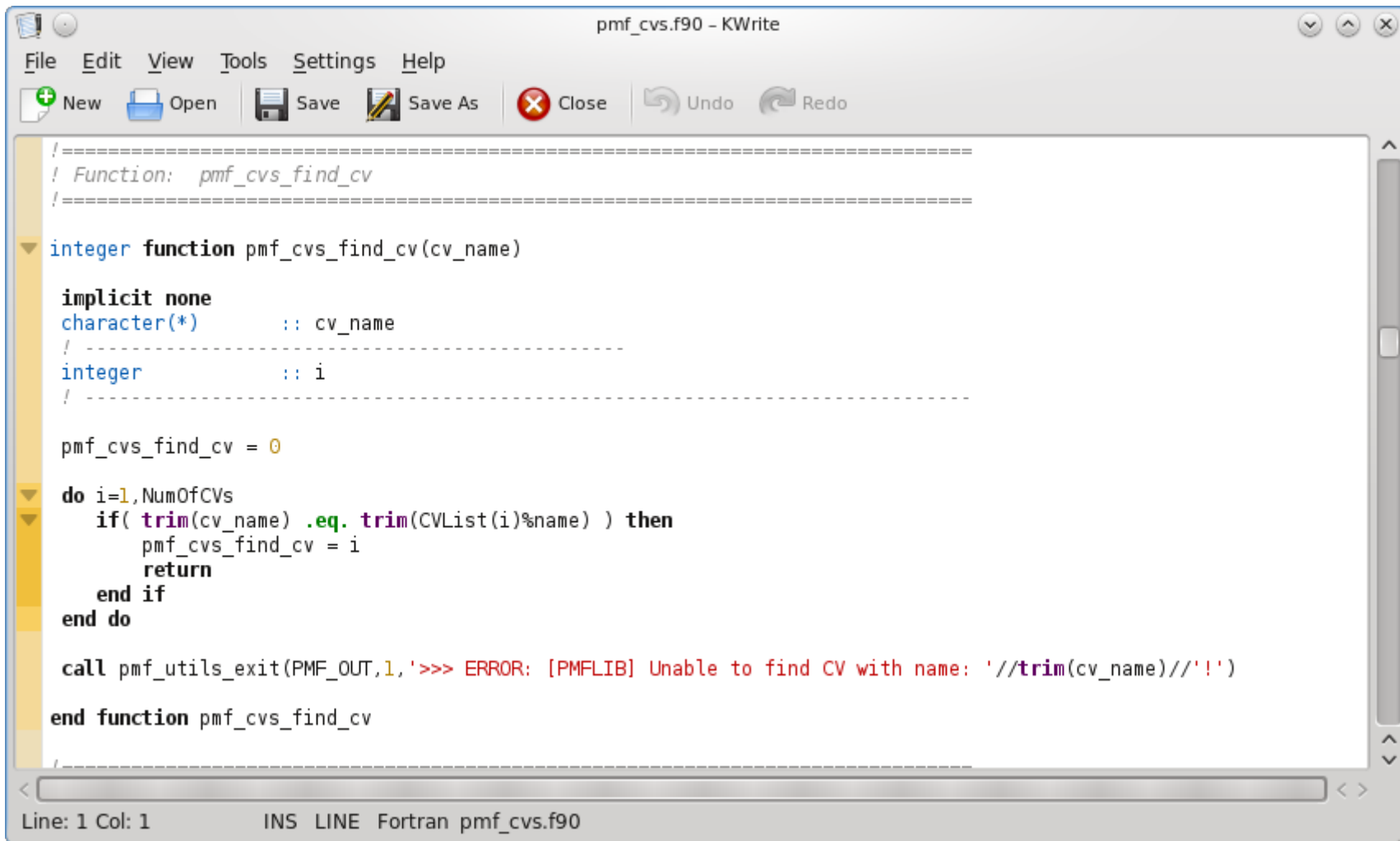
## Ukončení editoru

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

## Změny souboru

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

# kwrite



```
pmf_cvs.f90 - KWrite
File Edit View Tools Settings Help
New Open Save Save As Close Undo Redo

!=====
! Function: pmf_cvs_find_cv
!=====
integer function pmf_cvs_find_cv(cv_name)

implicit none
character(*)      :: cv_name
! -----
integer          :: i
! -----

pmf_cvs_find_cv = 0

do i=1,NumOfCVs
  if( trim(cv_name) .eq. trim(CVList(i)%name) ) then
    pmf_cvs_find_cv = i
    return
  end if
end do

call pmf_utils_exit(PMF_OUT,1,'>>> ERROR: [PMFLIB] Unable to find CV with name: '//trim(cv_name)//'!')

end function pmf_cvs_find_cv

Line: 1 Col: 1      INS LINE Fortran pmf_cvs.f90
```

# kate

```
bool result = true;

//***** LOCKED DATABASE ACCESS *****

VSServer.SelMutex.Lock();

if( VSServer.CommitSelTrans() == false ){
    VSServer.SelMutex.Unlock();
    return(false);
}

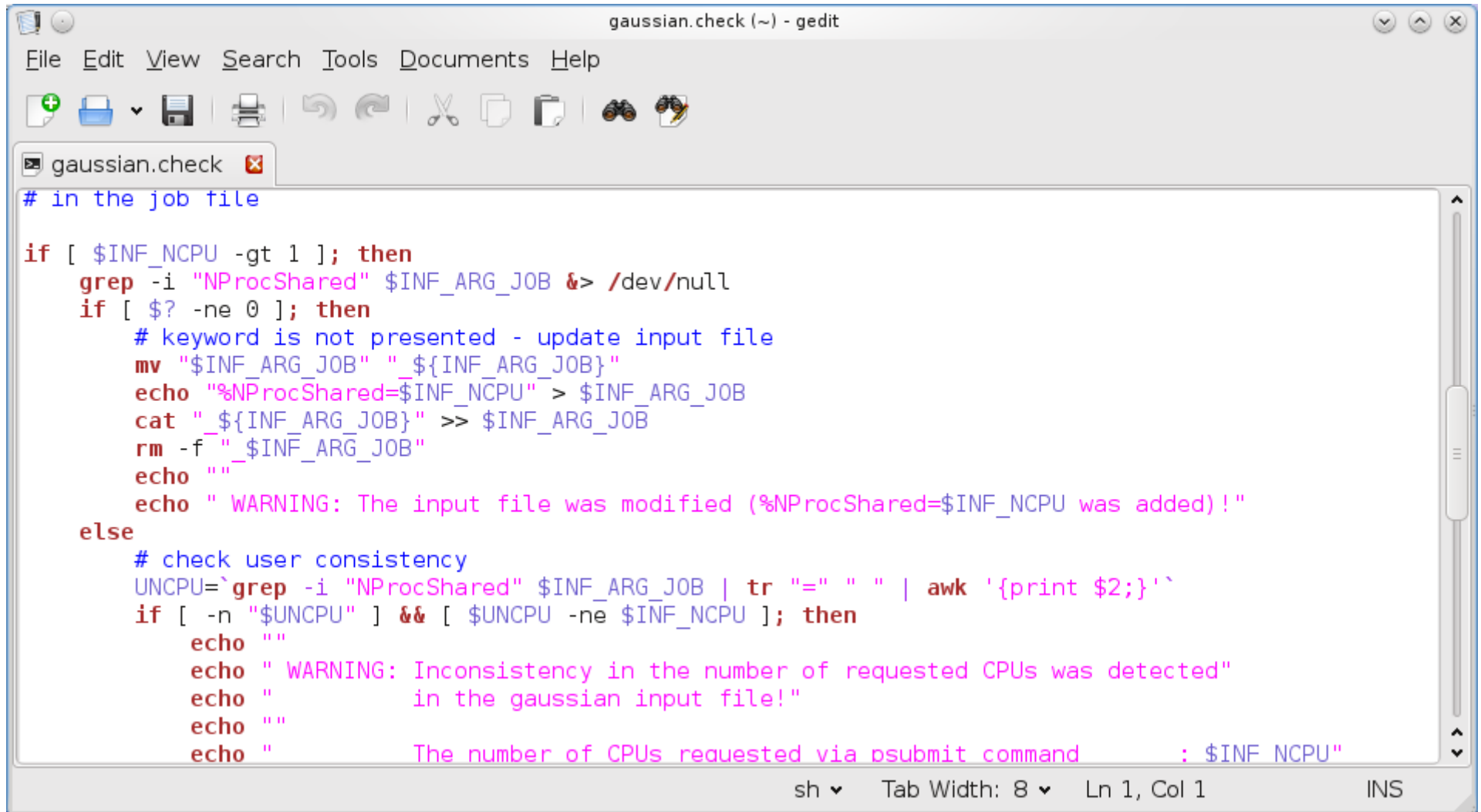
if( VSServer.SelQuery.QueryRecord() == false ){
    ResultElement->SetAttribute("molid","eof");    // no more structures
    VSServer.SelMutex.Unlock();
    return(true);
}

if( VSServer.SelQuery.GetOutputItem(0) == NULL ){
    CMD_ERROR("unable to get output item of query","CVSPProcessor::GetStructure");
    VSServer.SelMutex.Unlock();
    return(false);
}
```

Line: 1 Col: 1      INS LINE OpGetStructure.cpp

Find in Files    Terminal

# gedit



The screenshot shows the gedit text editor window titled "gaussian.check (~) - gedit". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for file operations like opening, saving, printing, and undo. The main text area displays a shell script for "gaussian.check" with the following content:

```
# in the job file

if [ $INF_NCPU -gt 1 ]; then
    grep -i "NProcShared" $INF_ARG_JOB &> /dev/null
    if [ $? -ne 0 ]; then
        # keyword is not presented - update input file
        mv "$INF_ARG_JOB" "${INF_ARG_JOB}"
        echo "%NProcShared=$INF_NCPU" > $INF_ARG_JOB
        cat "${INF_ARG_JOB}" >> $INF_ARG_JOB
        rm -f "${INF_ARG_JOB}"
        echo ""
        echo " WARNING: The input file was modified (%NProcShared=$INF_NCPU was added)!"
    else
        # check user consistency
        UNCPU=`grep -i "NProcShared" $INF_ARG_JOB | tr "=" " " | awk '{print $2;}'`
        if [ -n "$UNCPU" ] && [ $UNCPU -ne $INF_NCPU ]; then
            echo ""
            echo " WARNING: Inconsistency in the number of requested CPUs was detected"
            echo "           in the gaussian input file!"
            echo ""
            echo "           The number of CPUs requested via psubmit command           : $INF_NCPU"
        fi
    fi
fi
```

The status bar at the bottom indicates "sh", "Tab Width: 8", "Ln 1, Col 1", and "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.

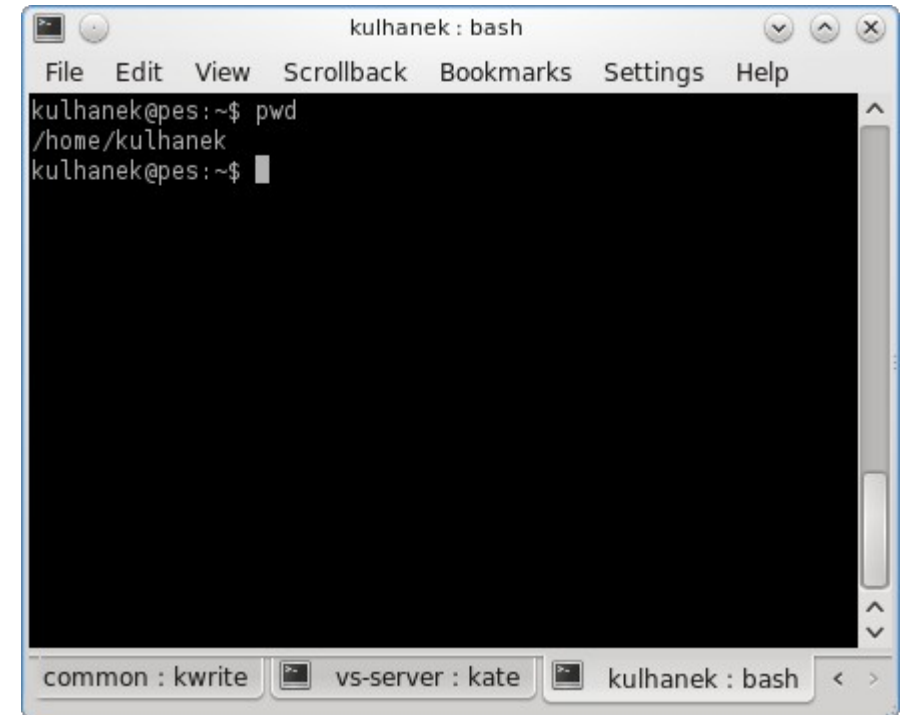
## xterm



```
kulhanek@wolf01:~ <@wolf01>
[kulhanek@wolf01 ~]$
```

jednoduché, standard na všech UNIXových systémech

## konsole



```
kulhanek : bash
File Edit View Scrollback Bookmarks Settings Help
kulhanek@pes:~$ pwd
/home/kulhanek
kulhanek@pes:~$
```

jednoduché přitom značně konfigurovatelné, dostupné v prostředí KDE

Výchozím adresářem je: **`/home/vas_login`**

# Struktura příkazové řádky

**kulhanek@wolfn ~\$**

jméno uživatele → kulhanek  
jméno počítače → wolfn  
aktuální adresář (~ znamená domovský adresář /home/vas\_login)  
typ uživatele (\$ běžný uživatel, # superuživatel)

místo pro příkaz

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 ...

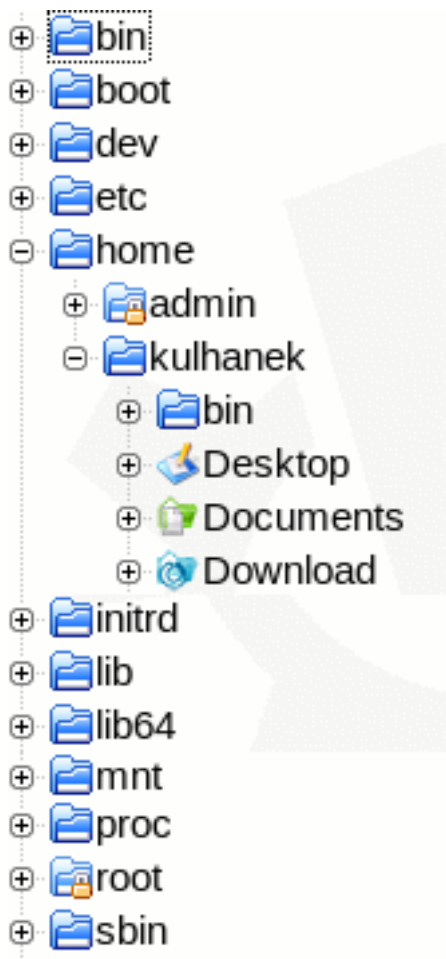
# Souborový systém

---

- **Struktura souborového systému**
- **Hlavní rozdíly vůči operačnímu systému MS Windows**
- **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**.



## 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)



# Srovnání s MS Windows

Vlastnost	Linux (ext2/ext3/ext4)	MS Windows (FAT32,NTFS)
Diskové oddíly (partitions)	Ne Diskové oddíly se připojují jako adresáře.	C:, D:, atd. Je však možné připojit i jako adresář (ntfs).
Jména	Rozlišuje malá a velká písmena (case sensitive).	Nerozlišuje malá a velká písmena (case insensitive).
Oddělování jmen	Lomítko	Zpětné lomítko
Přístupová práva	Ano POSIX	Ano (pouze NTFS) ACL
Zařízení (hardware)	Jako speciální soubory.	Ne

**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

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

↑ přístupová práva

↑ vlastník (user)  
a skupina (group)

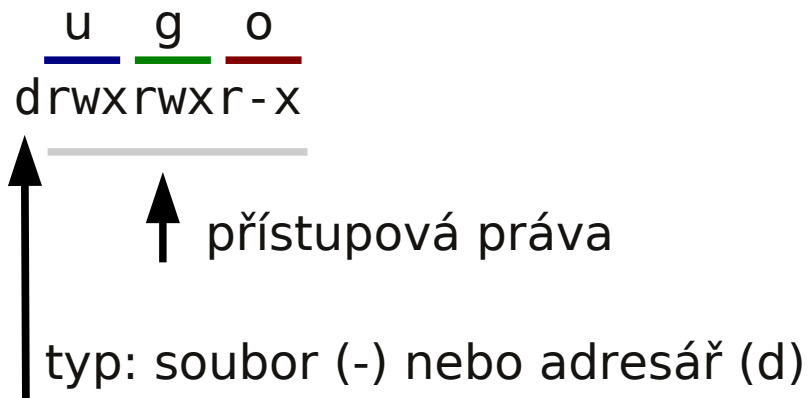
↑ velikost

↑ čas poslední změny

↑ jméno souboru či adresáře/

Přístupové práva se mění příkazy: **chmod**, **chgr**, **chown**

# Změna přístupových práv



## Přístupová práva:

<b>r</b>	možnost číst soubor	vypsat obsah adresáře
<b>w</b>	možnost měnit soubor	změnit obsah adresáře
<b>x</b>	možnost spuštění souboru	možnost vstoupit do adresáře

## Skupina práv:

<b>u</b>	vlastník (user)
<b>g</b>	skupina (group)
<b>o</b>	ostatní (other)
<b>a</b>	všichni (all), právo se aplikuje na u,g,o

## Ukázka:

\$ chmod u+x,g-w soubor	Přidá (+) právo pro spuštění vlastníkovi a Odstraní (-) právo zápisu pro skupinu
-------------------------	---



# 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
```

Soubory v adresáři jejichž jméno začíná **.** (tečkou) jsou **skryté** (používají se k ukládání konfigurace systému a aplikací. Lze je vylistovat pomocí: **ls -a**)

# Diskové oddíly

```
[kulhanek@wolfn ~]$ df -Th
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/mapper/server1-root
                ext3      15G   8.8G  4.8G  65% /
/dev/md0        ext3      190M   18M  163M  10% /boot
/dev/mapper/server1-home
                ext3     394G   47G  331G  13% /home
/dev/mapper/server1-software
                ext3      15G   13G   2.0G  87% /software/ncbr
tmpfs           tmpfs    1005M   12K 1005M   1% /dev/shm
```

**zařízení**

typ souborového systému

**přípojný bod**

```
[kulhanek@wolf02 ~]$ df -Th
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/mapper/server1-root
                ext3      20G   7.4G  12G  40% /
/dev/sda1       ext3      185M   12M  164M   7% /boot
/dev/mapper/server1-scratch
                ext3      30G   1.2G  27G   5% /scratch
/dev/mapper/server1-vbox
                ext3      20G   10G   8.7G  54% /win
tmpfs           tmpfs    1014M   12K 1014M   1% /dev/shm
wolfn.wolf.net:/home/kulhanek
                nfs      394G   47G  331G  13% /home/kulhanek
wolfn.wolf.net:/software/ncbr
                nfs      15G   13G   2.0G  87% /software/ncbr
```

**zařízení**

- diskový oddíl
- síťový disk
- atd

# 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

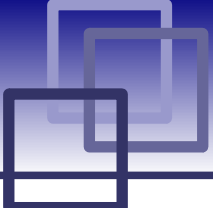
## ***Navigace v textu nápovědy:***

- 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/>

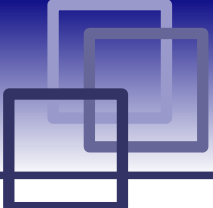




# Příkazy - přehled

## ***Souborový systém:***

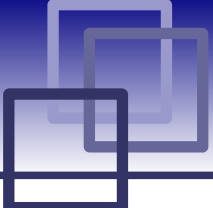
- pwd vypíše cestu aktuálního pracovního adresáře
- cd změní aktuální pracovní adresář
- mkdir vytvoří adresář
- rmdir odstraní adresář
- ls vypíše obsah adresáře
- du vypíše velikost adresáře
- cp zkopíruje soubor či adresář
- mv přesune soubor či adresář
- rm odstraní soubor či adresář
- stat vypíše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



# Příkazy - přehled

## **Procesy:**

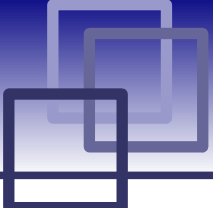
- top informace o běžících procesech v systému
- ps vypíše informace o běžících procesech v systému
- kill zašle signál procesu
- nohup spustí process bez interakce s terminálem
- time vypíše délku běhu procesu
- wait čeká na dokončení procesů na pozadí
- ssh spustí příkaz na vzdáleném počítači



# 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

## **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
- `seq` vypíše sekvenci čísel
- `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