

TSM

Modelování molekulárních struktur

Cvičení II

Petr Kulhánek

petr.kulhanek@ceitec.muni.cz

CEITEC – Středoevropský technologický institut, Masarykova univerzita,
Kamenice 5, 625 00 Brno

NCBR – Národní centrum pro výzkum biomolekul, Masarykova univerzita,
Kotlářská 2, 611 37 Brno

Základní příkazy

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
- cd
- mkdir
- rmdir
- ls
- cp
- mv
- rm

vypíše cestu aktuálního pracovního adresáře
změní aktuální pracovní adresář
vytvoří adresář
odstraní adresář
vypíše obsah adresáře
zkopíruje soubor či adresář
přesune soubor či adresář
odstraní soubor či adresář

Zjišťovací příkazy:

- hostname
- id
- whoami

vypíše jméno počítače
vypíše identifikační čísla uživatele
vypíše jméno přihlášeného uživatele

Souborový manažer

- mc

<http://www.midnight-commander.org/>

Vytvoření adresářů

- Vytvoření adresáře

```
$ mkdir jmeno_adresare
```

- Vytvoření vnořených adresářů

```
$ mkdir -p jmeno_adresare1/jmeno_adresare2
```

Kopírování

- Ke kopírování slouží příkaz “cp”

```
$ cp soubor1 soubor2
```

vytvoří kopii souboru “soubor1” s názvem “soubor2”

```
$ cp soubor1 soubor2 soubor3 adresar1/
```

kopíruje soubory “soubor1”, “soubor2”, “soubor3” do adresáře “adresar1”

```
$ cp -r adresar1 adresar2
```

vytvoří kopii adresáře “adresar1” s názvem “adresar2”; pokud adresář “adresar2” již existuje, vytvoří kopii adresáře “adresar1” jako podadresář adresáře “adresar2”

```
$ cp -r soubor1 adresar2 soubor3 adresar1/
```

kopíruje soubory “soubor1”, “soubor3” a adresář “adresar2” do adresáře “adresar1”

Přesouvání

- K přesouvání nebo přejmenování slouží příkaz “mv”

```
$ mv soubor1 soubor2
```

přejmenuje soubor “soubor1” na “soubor2”

```
$ mv soubor1 soubor2 soubor3 adresar1/
```

přesune soubory “soubor1”, “soubor2”, “soubor3” do adresáře “adresar1”

```
$ mv adresar1 adresar2
```

přejmenuje adresář “adresar1” na “adresar2”; pokud adresář “adresar2” již existuje, přesune adresář “adresar1” do adresáře “adresar2”

```
$ mv soubor1 adresar2 soubor3 adresar1/
```

přesune soubory “soubor1”, “soubor3” a adresář “adresar2” do adresáře “adresar1”

Mazání

- K mazání slouží příkaz `rm`

```
$ rm soubor1
```

odstraní soubor "soubor1"

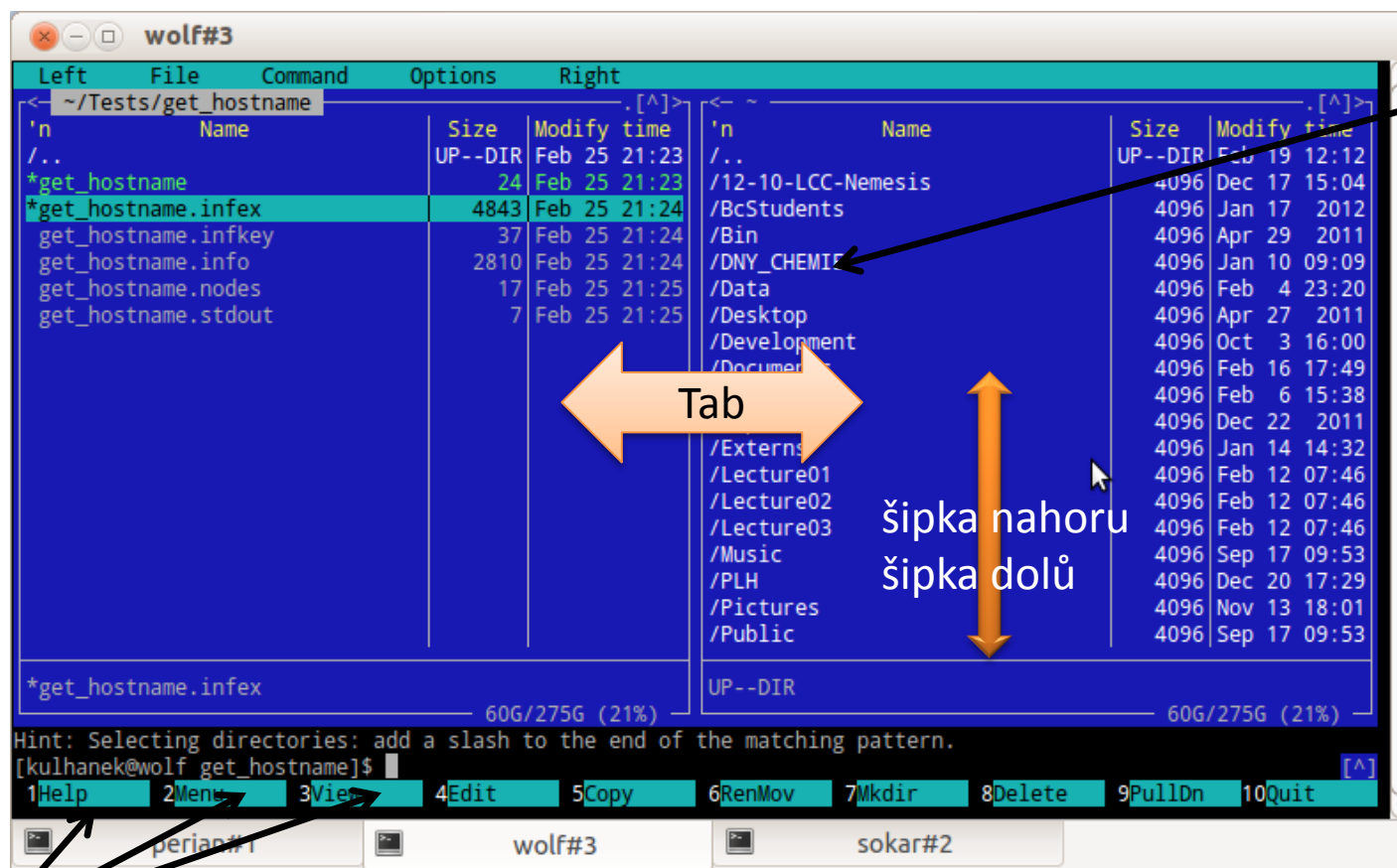
```
$ rm -r adresar1
```

odstraní adresář "adresar1"

Midnight Commander

Souborový manažer, který pracuje v terminálu.

<http://www.midnight-commander.org/>



označení více souborů - **Insert**

Tab

šipka nahoru
šipka dolů

funkční klávesy F1, F2, F3, ...

Skrytí obou panelů: Ctrl+O

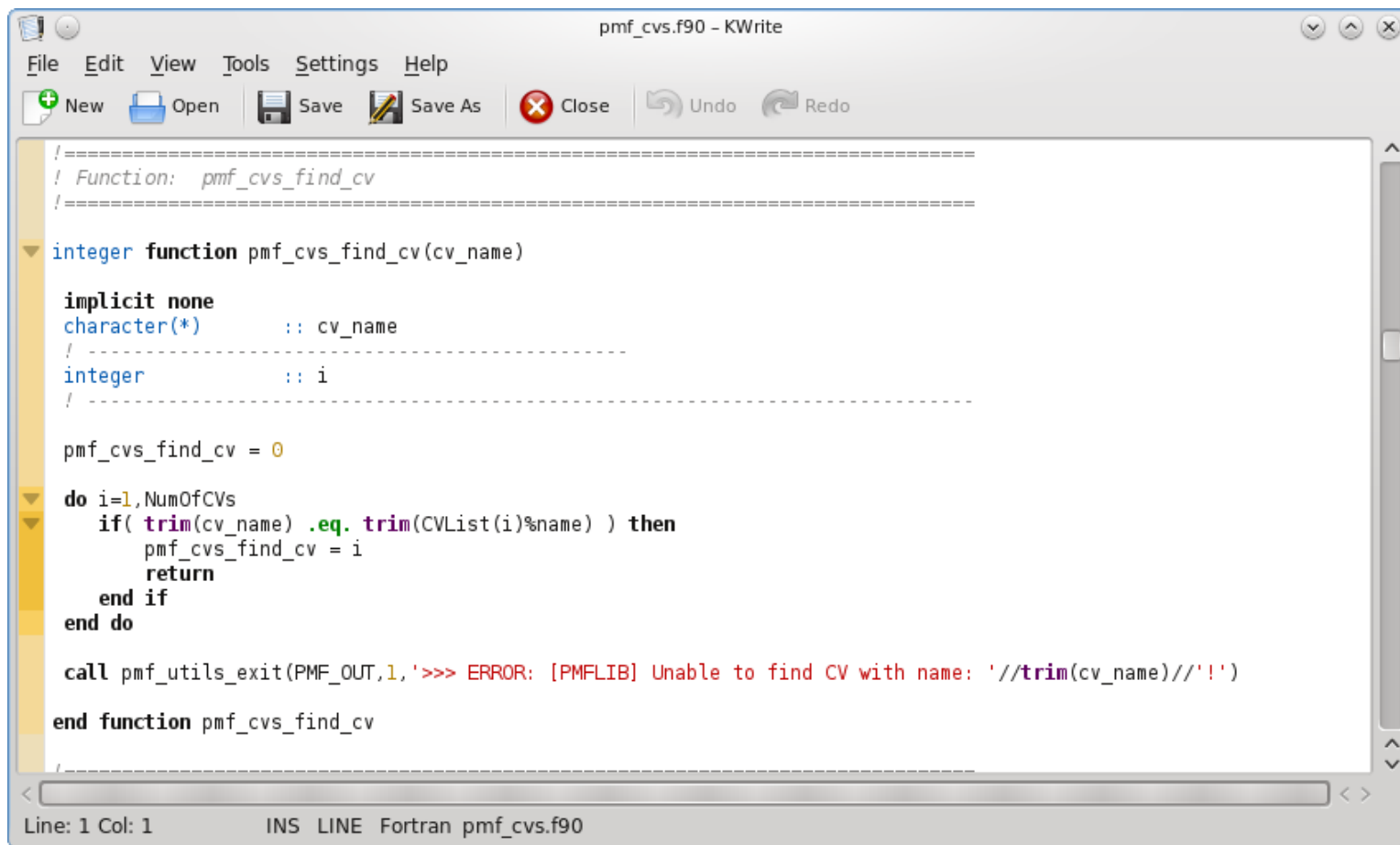
Lze použít myš.

Cvičení

1. Do adresáře **~/Downloads** si stáhněte studijní materiály (**Cvičení 2**) k předmětu TSM ze stránky:
<https://lcc.ncbr.muni.cz/~kulhanek/teaching/teaching.html>
2. Vytvořte adresář **pokus** v adresáři **/scratch/vas_login**
3. Vytvořte adresář **studmat** ve vašem domovském adresáři
4. Do adresáře **studmat** zkopírujte studijní materiály z adresáře **~/Downloads**
5. Otevřete prezentaci (**Cvičení 2**) v programu **okular**
6. Prezentaci přepokopírujte do adresáře **/scratch/vas_login/pokus**
7. V adresáři **/scratch/vas_login/pokus** prezentaci přejmenujte na **pokus.pdf**
8. Prezentaci **pokus.pdf** otevřete v programu **okular**
9. Smažte prezentace v adresáři **~/Downloads**

Textové editory

kwrite

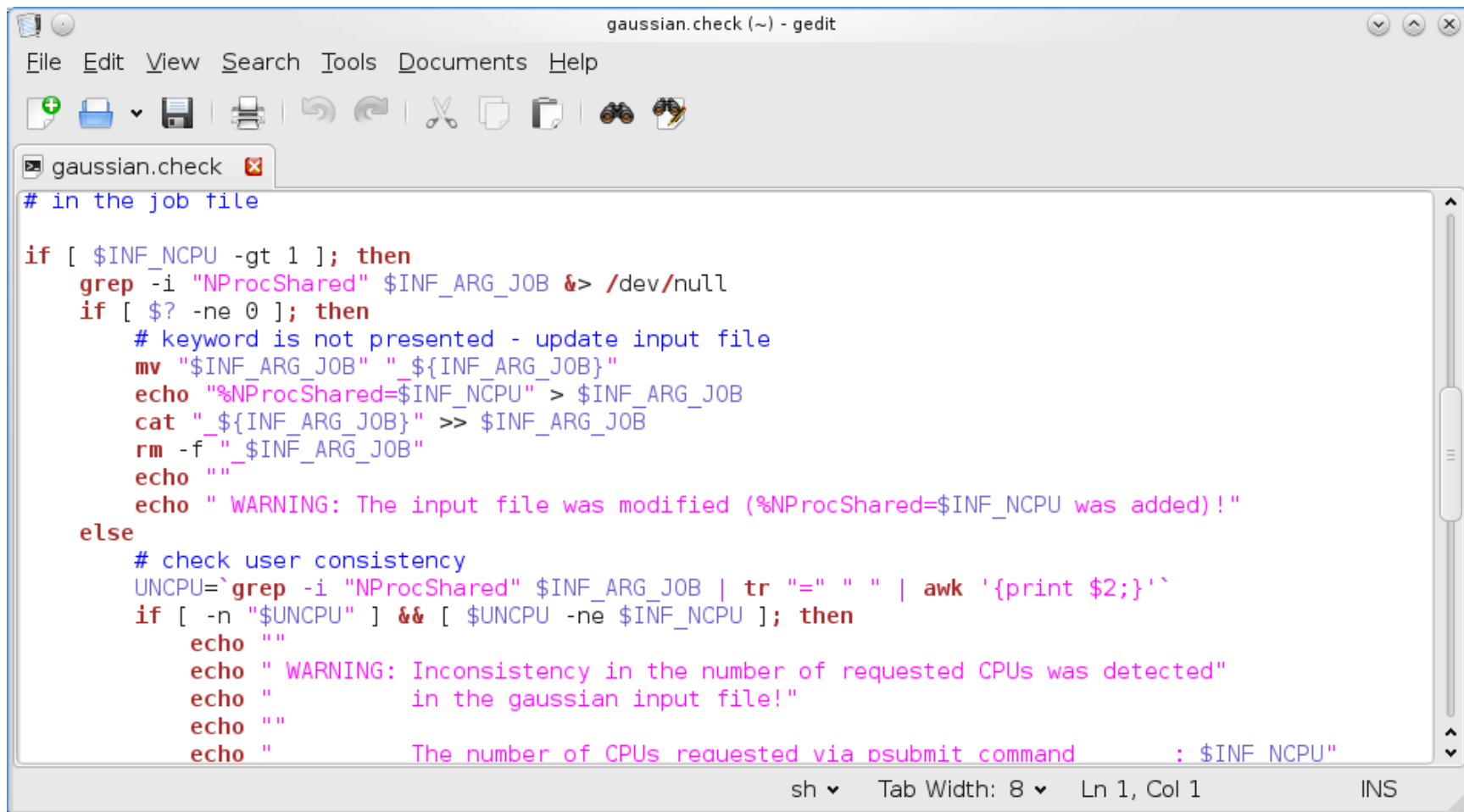


```
=====  
! Function: pmf_cvs_find_cv  
!=====  
integer function pmf_cvs_find_cv(cv_name)  
  
implicit none  
character(*) :: cv_name  
!-----  
integer :: i  
!-----  
  
pmf_cvs_find_cv = 0  
  
do i=1,NumOfCVs  
  if( trim(cv_name) .eq. trim(CVList(i)%name) ) then  
    pmf_cvs_find_cv = i  
    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  
!-----
```

Spuštění:
\$ kwrite

Rozšířená funkcionální: **kate**

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 save, print, and copy. The main text area displays a shell script for checking Gaussian input files. The script starts with a comment "# in the job file" and uses an if statement to check if the number of requested CPUs (NPROC_SHARED) is greater than 1. If so, it updates the input file with the correct number of CPUs and issues a warning. Otherwise, it checks for consistency between the number of CPUs requested via the psubmit command and the number in the Gaussian input file, also issuing a warning if there is an inconsistency.

```
# 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
```

sh Tab Width: 8 Ln 1, Col 1 INS

Spuštění:

\$ gedit

Cvičení

1. Vytvořte textový soubor **voda.xyz**, který bude obsahovat strukturu molekuly vody ve formátu xyz. Délka vazeb O-H bude 1 Å. Vazebný úhel H-O-H bude 90°. Jedná se o optimální geometrii molekuly vody? Jaký je optimální vazebný úhel H-O-H?
2. Správnost vytvořeného modelu ověřte v programu VMD. Vytvořenou strukturu vody zobrazíte následovně:

```
$ module add vmd  
$ vmd voda.xyz
```

3. Soubor ve formátu **xyz** převedte do formátu **mol2** pomocí programu Open Babel.

```
$ module add openbabel  
$ babel voda.xyz voda.mol2
```

4. Vytvořený soubor **voda.mol2** otevřete v textovém editoru. Zkuste odhadnout význam jednotlivých částí souboru.

Virtualizace

Přehled nástrojů pro virtualizaci

VirtualBox

www.virtualbox.org

Podporovaný hostitelský OS: MS Windows, Mac OS X, Linux

Licence: freeware + proprietární rozšíření pro nekomerční použití

KVM

součástí kernelu Linuxu

Podporovaný hostitelský OS: Linux

Podpůrné programy: virt-manager, qemu

Licence: freeware

VMWare

<http://www.vmware.com/>

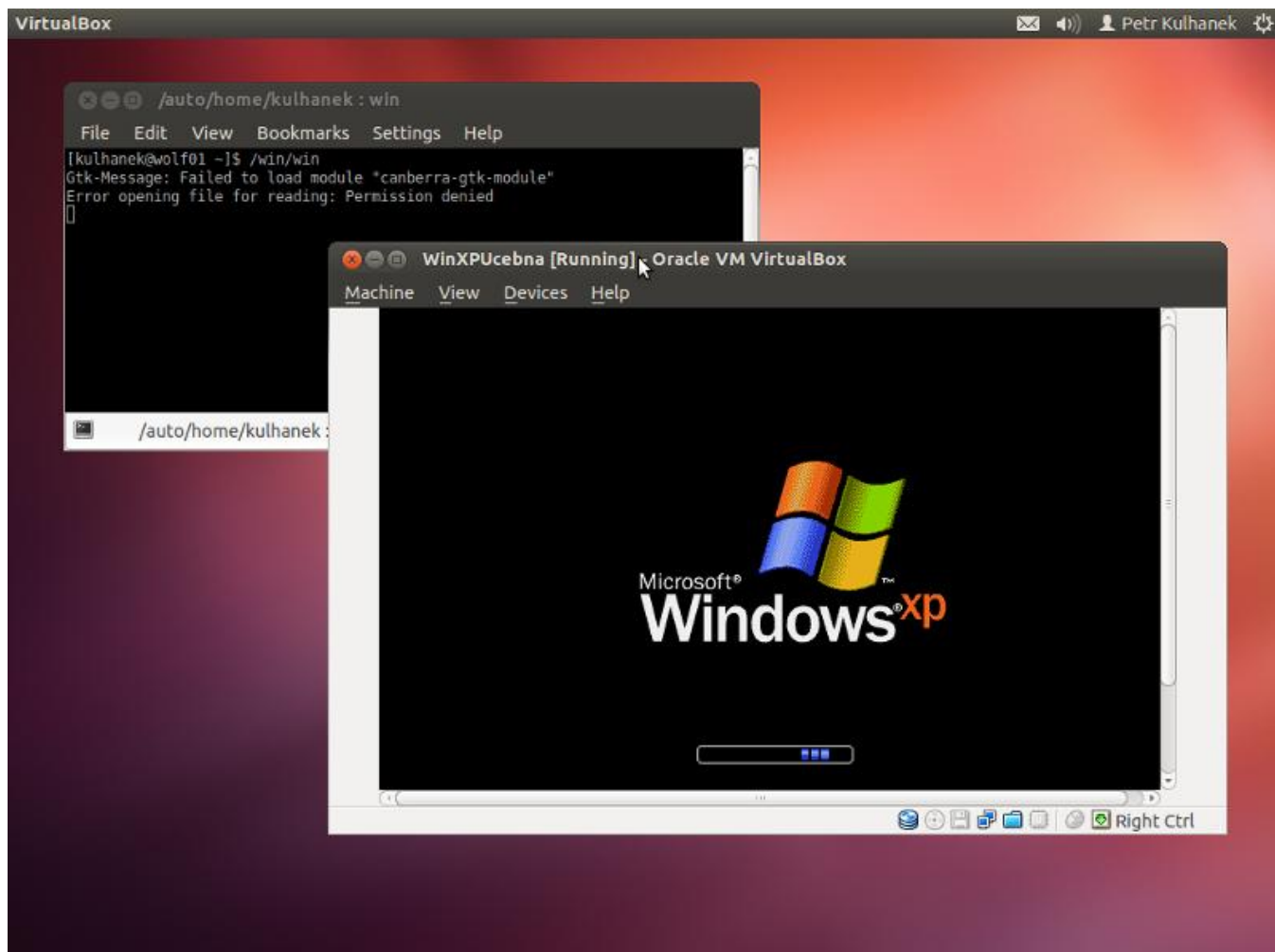
Podporovaný hostitelský OS: MS Windows, Linux

Licence: komerční

MS Windows na klastru WOLF

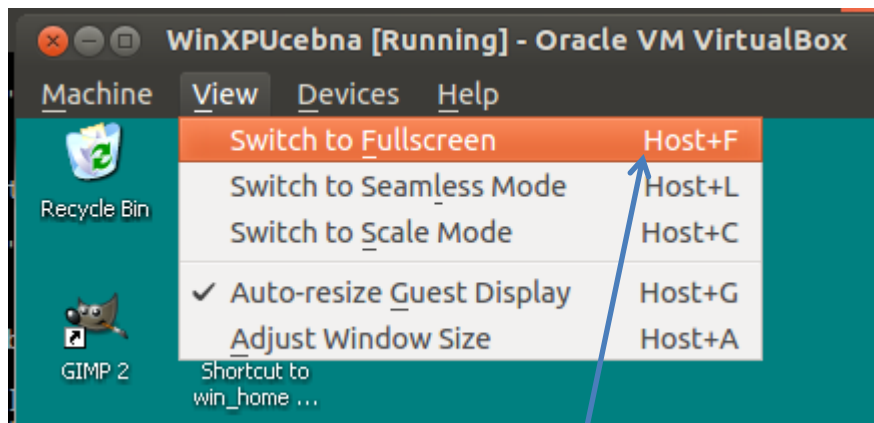
Spuštění MS Windows XP ve virtuálním stroji (hypervisor VirtualBox)

\$ `/win/win`



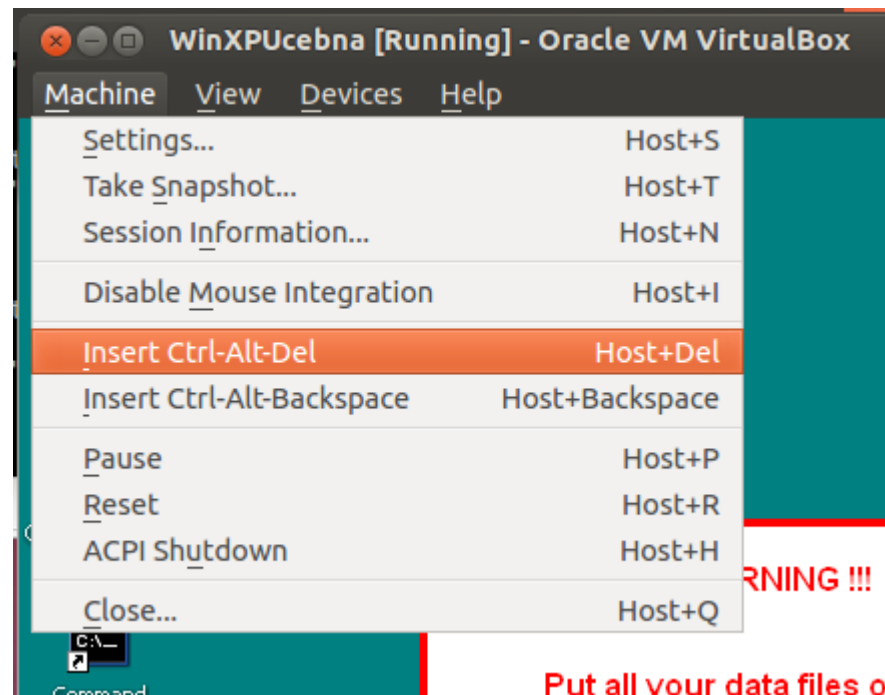
Ovládání virtuálního stroje

Přepnutí do/z Fullscreen



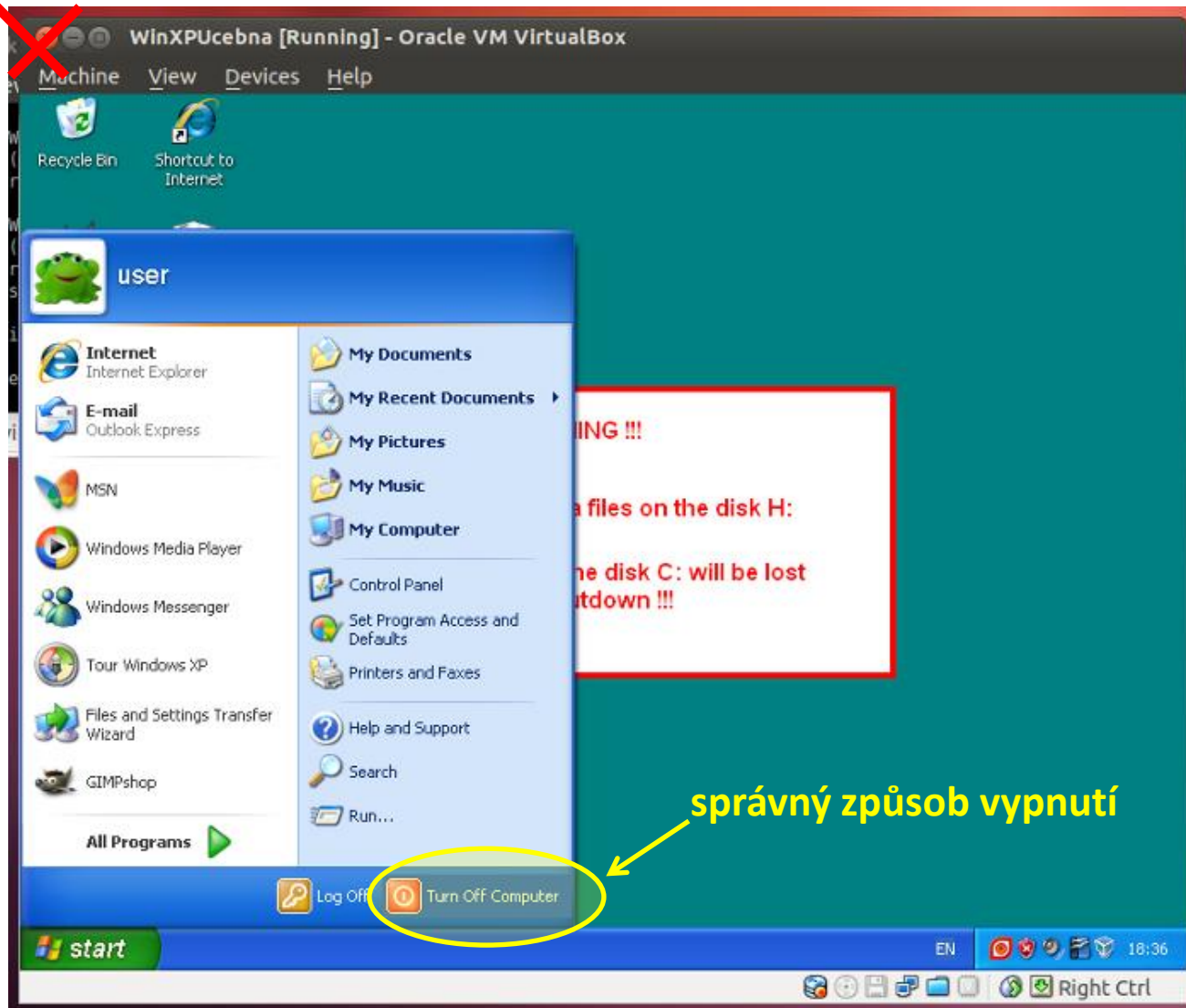
Host = (pravá klávesa Ctrl)
(pod MSWindows a Linuxem)

Zmáčknutí kláves Ctrl+Alt+Del



Vypnutí virtuálního stroje

špatný způsob
vypnutí



MS Windows jako klient

MS Windows jako klient - přehled

Přihlašování do Unixu z MS Windows (textový terminál):

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

Kopírování dat mezi Unixem a MS Windows:

WinSCP (<http://winscp.net>)
scp (např. z prostředí Cygwin; <http://www.cygwin.com/>)

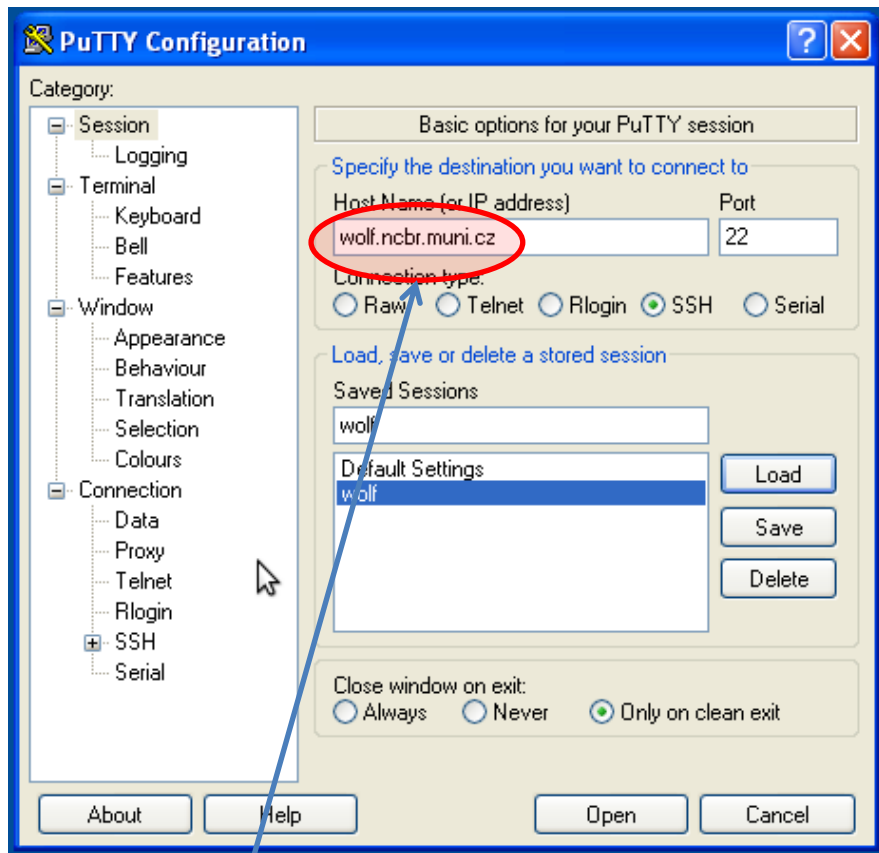
Export displeje z Unixu do MS Windows (X11 server):

Xming (<http://sourceforge.net/projects/xming/>)
cygwin (<http://www.cygwin.com/>)

Přihlašování z Unixu do MS Windows (vzdálená plocha):

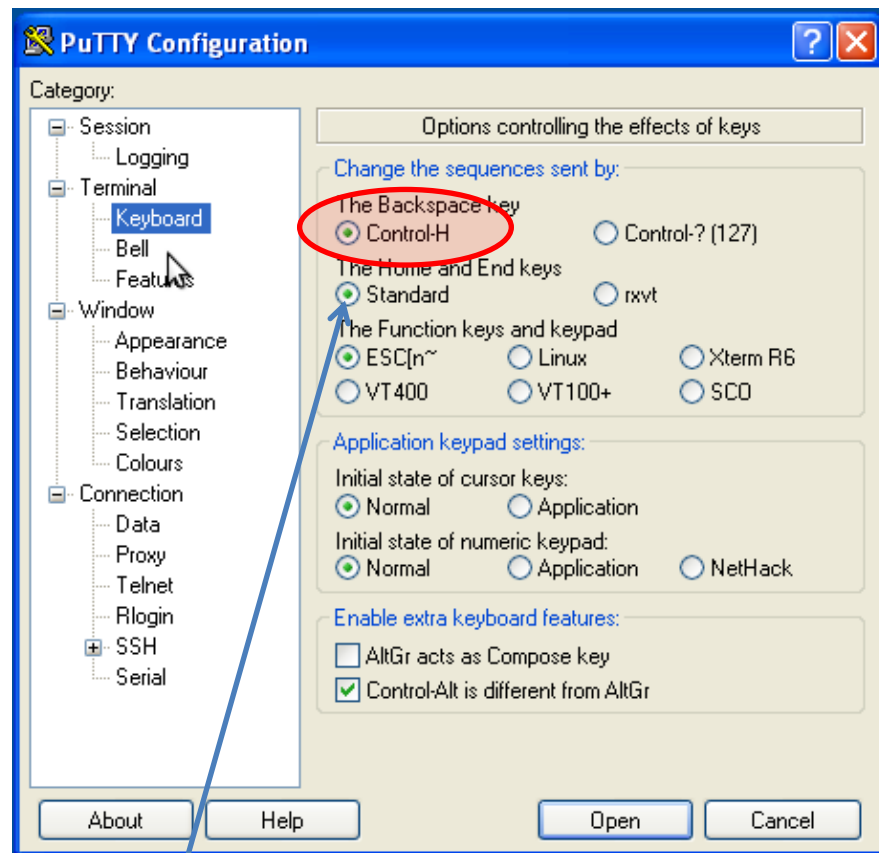
rdesktop

Putty – nastavení



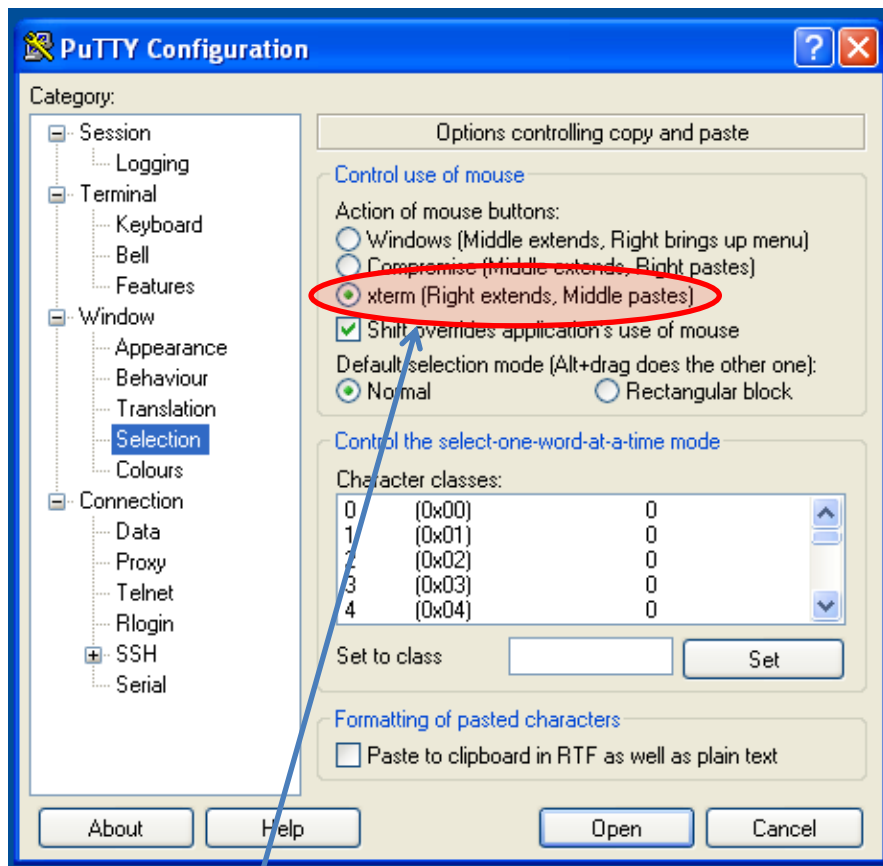
adresa vzdáleného stroje

wolf.ncbr.muni.cz

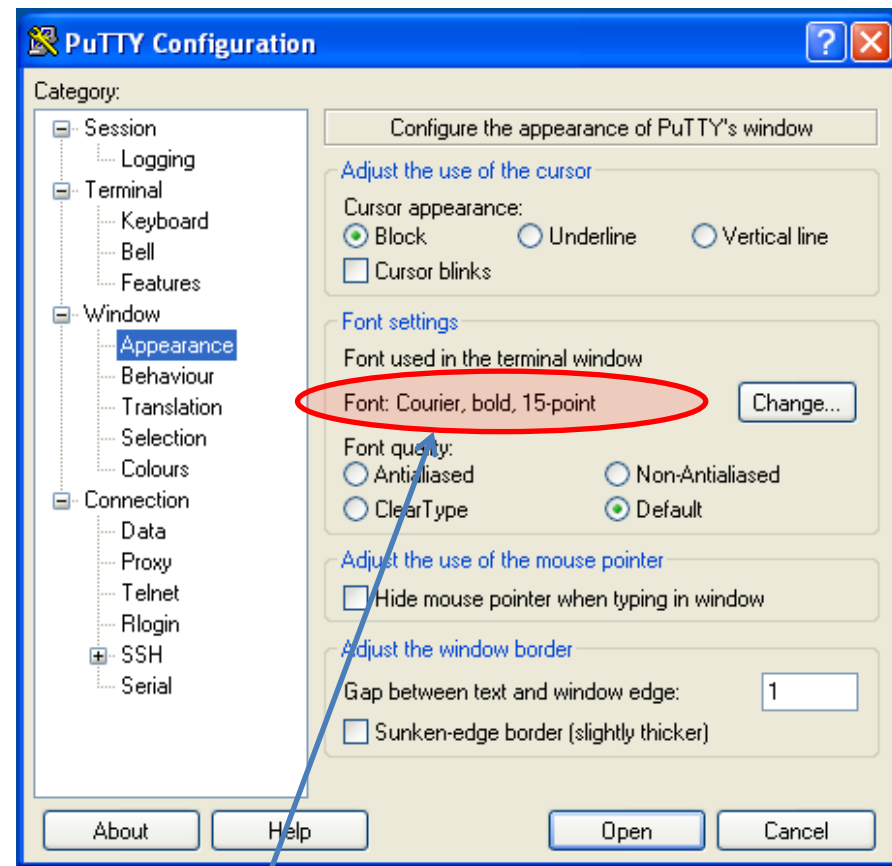


Správné fungování klávesy backspace.

Putty – nastavení II



selektce myši kompatibilní s Unixovými terminály

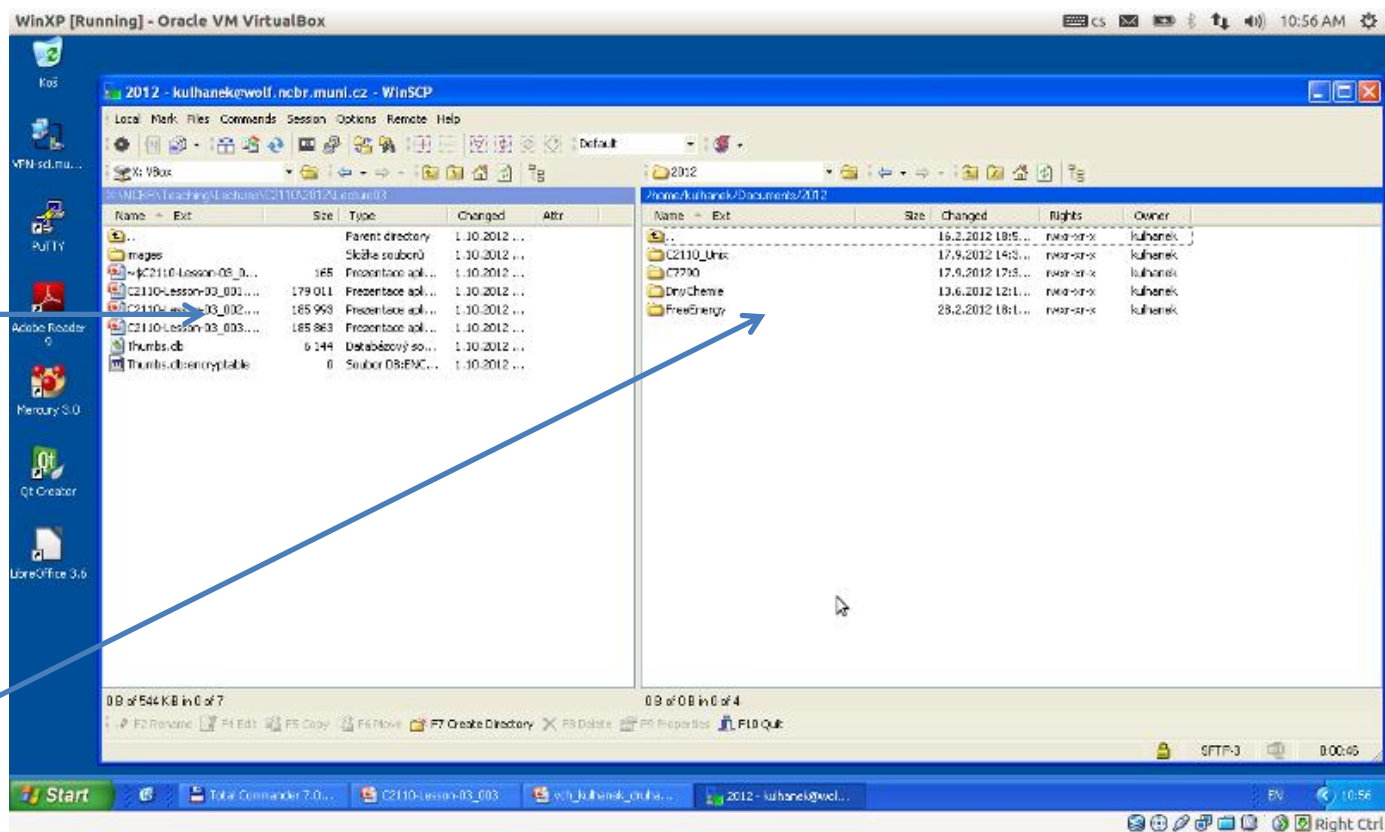


neproporcionální písmo
(všechny znaky mají stejnou šířku)

WinSCP

WinSCP <http://winscp.net/eng/docs/lang:cs>

Program pro přenos souborů mezi MS Windows a počítači podporující SFTP či SCP protokoly (převážně unixového a linuxového typu).



lokální stroj

vzdálený stroj

Textové soubory MS Win ↔ Linux

Textové soubory vytvořené pod MS Windows a Linuxem **nejsou** zcela **kompatibilní**, protože každý operační systém používá jiné kódování **konce řádku**.

Linux: \n (line feed 0x0A)

MS Windows: \r+\n (carriage return 0x0D, line feed 0x0A)

Ke konverzi souborů lze použít programy **d2u** a **u2d** (na klastru WOLF).

1) Aktivace modulu cats

```
$ module add cats
```

2) Konverze MS Windows => Linux

```
$ d2u soubor.com
```

3) Konverze Linux => MS Windows

```
$ u2d soubor.log
```

Cvičení

1. Spusťte virtuální stroj s MS Windows XP (**/win/win**).
2. Spusťte aplikaci Putty (Aplikaci stáhněte z internetu).
3. Pomocí terminálu Putty se přihlaste na stroj **wolf.ncbr.muni.cz**.
4. Monitorujte, kdo je na stroj wolf.ncbr.muni.cz přihlášen (příkazy **w** nebo **who**).
5. Zkuste v terminálu Putty spustit aplikaci **kwrite**. Proč spuštění aplikace selže?
6. Spusťte aplikaci WinSCP (Aplikaci stáhněte z internetu jako **Portable executables**).
7. Do virtuálního stroje stáhněte soubor **voda.xyz**. Soubor otevřete v programu Poznámkový blok (Notepad). Zobrazí se obsah souboru správně?
8. Opravte kódování konců řádků v souboru **voda.xyz** (na straně Linuxu) a soubor znovu otevřete ve virtuálním stroji v programu Poznámkový blok.

Vzdálené přihlášení

Vzdálené přihlášení

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

[] - možno vynechat

Syntaxe:

```
$ ssh [user@]hostname [command]
```

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

jméno počítače

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

Příklady použití:

```
$ ssh wolf.wolf.inet
```

```
$ ssh wolf01.wolf.inet w
```

Prvotní vzdálené přihlášení

```
[kulhanek@wolf01 ~]$ ssh wolf02
```

The authenticity of host 'wolf02 (10.251.28.102)' can't be established.

ECDSA key fingerprint is **1f:9d:f3:d3:1d:24:28:12:56:30:99:ef:2d:68:d2:cf**.

Are you sure you want to continue connecting (yes/no)? **yes**

Warning: Permanently added 'wolf02,10.251.28.102' (ECDSA) to the list of known hosts.

```
[kulhanek@wolf02 ~]$
```

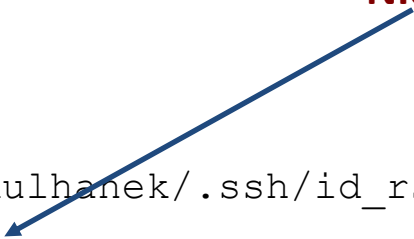
Při prvním přihlášení je nutné potvrdit autenticitu stroje, na který se hlásíme. Ve věrohodné síti můžeme otisku palce potvrdit. V nezabezpečeném prostředí je však vhodné otisk palce stroje ověřit nezávislou cestou (např. zasláním otisku palce poštou od správce vzdáleného stroje).

Přihlašování bez hesla (uvnitř klastru WOLF)

1. Vytvoření dvojice veřejného a soukromého klíče:

```
[kulhanek@wolf01 ~]$ cd .ssh
[kulhanek@wolf01 .ssh]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kulhanek/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kulhanek/.ssh/id_rsa.
Your public key has been saved in /home/kulhanek/.ssh/id_rsa.pub.
The key fingerprint is:
e9:07:0b:fc:17:23:b3:c5:1a:8a:0c:1a:98:8f:fe:28 kulhanek@wolf01.wolf.inet
```

Nic se nezadává!



2. Vložení veřejného klíče do seznamu autorizovaných klíčů:

```
[kulhanek@wolf01 .ssh]$ cat id_rsa.pub >> authorized_keys
```

Výhody:

- nemusí se neustále zadávat heslo
- bezpečnější použití příkazů ssh a scp ve skriptech
- urychlení práce

Nevýhody:

- v případě kompromitace jednoho počítače, jsou kompromitovány všechny počítače se vzájemně autorizovanými veřejnými klíči.

Podrobnější popis: man ssh

Cvičení

1. Aktivujte si přihlašování bez hesla v rámci klastru WOLF.
2. Ověřte funkčnost přihlašování bez hesla. Přihlaste se na uzel wolf01. Kdo je na uzlu přihlášen (příkaz **w** nebo **who**)?

Dávkové systémy

Dávkové zpracování

Dávkové zpracování je vykonávání série programů (tzv. dávek) na počítači bez účasti uživatele. Dávky jsou připraveny předem, takže mohou být zpracovány předány bez účasti uživatele. Všechna vstupní data jsou předem připravena v souborech (skriptech) nebo zadána pomocí parametrů na příkazovém řádku. Dávkové zpracování je opakem interaktivního zpracování, kdy uživatel až teprve za běhu programu poskytuje požadované vstupy.

Výhody dávkového zpracování

- sdílení zdrojů počítače mezi mnoha uživateli a programy
- odložení zpracování dávek do doby, kdy je počítač méně vytížen
- odstranění prodlev způsobeným čekáním na vstup od uživatele
- maximalizace využití počítače zlepšuje využití investic (zejména u dražších počítačů)

zdroj: www.wikipedia.cz, upraveno

Nástroje pro dávkové spouštění

➤ **OpenPBS**

<http://www.mcs.anl.gov/research/projects/openpbs/>

➤ **PBSPro**

<http://www.pbsworks.com>

➤ **Oracle Grid Engine**

<http://www.oracle.com/us/products/tools/oracle-grid-engine-075549.html>

➤ **Open Grid Scheduler**

<http://gridscheduler.sourceforge.net/>

open source

➤ **Torque**

<http://www.adaptivecomputing.com/products/open-source/torque/>

je použit jako dávkový systém v MetaCentrum VO, na klastrech SOKAR a WOLF

Infinity

<https://lcc.ncbr.muni.cz/whitezone/development/infinity/>

Přehled příkazů

Správa software:

- site aktivace logických výpočetních zdrojů
- module aktivace/deaktivace software

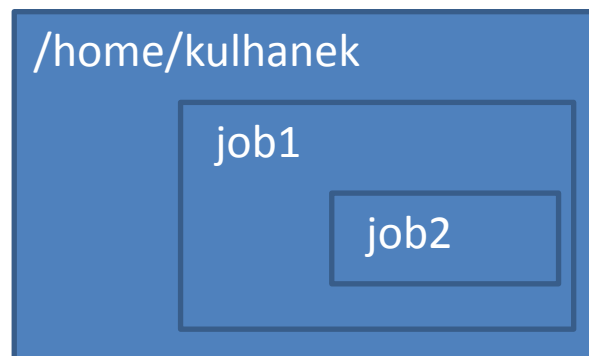
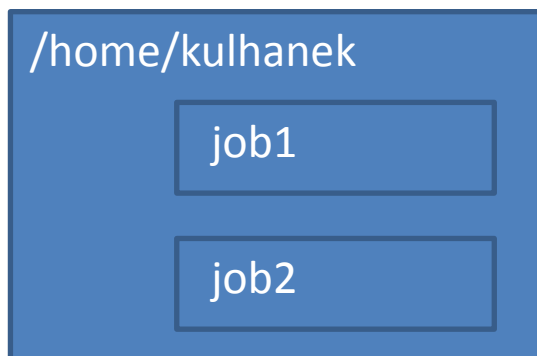
Správa úloh:

- pqueues přehled front z dávkového systému dostupných uživateli
- pnodes přehled výpočetních uzlů dostupných uživateli
- pqstat přehled všech úloh zadaných do dávkového systému
- pjobs přehled úloh uživatele zadaných do dávkového systému
- psubmit zadání úlohy do dávkového systému
- pinfo informace o úloze
- pgo přihlásí uživatele na výpočetní uzel, kde se úloha vykonává
- paliases definování aliasů

Úloha

Úloha **musí splňovat** následující podmínky:

- každá úloha se spouští v samostatném adresáři
- všechny vstupní data úlohy musí být v adresáři úlohy
- adresáře úloh nesmí být do sebe zanořené
- průběh úlohy je řízen skriptem nebo vstupním souborem (u automaticky detekovaných úloh)
- skript úlohy musí být v bashi
- ve skriptu úlohy se nesmí používat absolutní cesty, všechny cesty musí být uvedeny relativně k adresáři úlohy



Spuštění úlohy

Úlohu spouštíme **v adresáři úlohy** příkazem **psubmit**.

```
psubmit destination job [resources] [syncmode]
```

destination (kam) je buď:

- název_fronty
- název_uzlu@název_fronty

job je buď:

- název skriptu úlohy
- název vstupního souboru pro automaticky rozpoznávané úlohy

resources jsou požadované zdroje pro úlohu, pokud není uvedeno, požaduje se běh na 1 CPU

syncmode určuje způsob kopírování dat mezi adresářem úlohy a výpočetním uzlem, výchozím módem je "sync"

Monitorování běhu úlohy

K monitorování průběhu úlohy lze použít příkaz **pinfo**, který se spouští buď v adresáři úlohy nebo v pracovním adresáři na výpočetním uzlu. Dalšími možnostmi jsou příkazy **pjobs** a **pqstat**.

Pokud je úloha spuštěna na výpočetním uzlu, je možné použít příkaz **pgo**, který se naloguje na výpočetní uzel a změní aktuální adresář do pracovního adresáře úlohy.

Servisní soubory

V adresáři úlohy vznikají při zadání úlohy do dávkového systému a dále v průběhu života úlohy a po jejím ukončení servisní soubory. Jejich význam je následující:

- *.info kontrolní soubor s informacemi o průběhu úlohy
- *.infex vlastní skript (wrapper), který se spouští dávkovým systémem
- *.infout standardní výstup z běhu *.infex skriptu, **nutno analyzovat při nestandardním ukončení úlohy**
- *.nodes seznam uzlů vyhrazených pro úlohu
- *.gpus seznam GPU karet vyhrazených pro úlohu
- *.key unikátní identifikátor úlohy
- *.stdout **standardní výstup z běhu skriptu úlohy**

Cvičení I

1. Vytvořte textový soubor obsahující následující text. Soubor nazvěte **get_hostname** a uložte jej do samostatného adresáře **uloha01**.

```
#!/bin/bash
hostname
```

2. Nastavte souboru práva pro spuštění:

```
$ chmod u+x get_hostname
```

3. Vytvořený skript spusťte v terminálu. Co se stane?

```
$ ./get_hostname
```

4. Skript zadejte ke spuštění do dávkového systém. K zadání úlohy použijte frontu **short**.

```
$ psubmit short get_hostname
```

5. Průběh úlohy monitorujte příkazem **pinfo**.

6. Na kterém výpočetním uzlu se úloha spustila? Analyzujte obsah souboru **get_hostname.stdout** v textovém editoru.

Cvičení II

1. Vytvořte textový soubor obsahující následující text. Soubor nazvěte **long_job** a uložte jej do samostatného adresáře **uloha02**. Co dělá příkaz **sleep** (použijte manuálové stránky)?

```
#!/bin/bash
hostname
sleep 240
```

2. Skript zadejte ke spuštění do dávkového systém. K zadání úlohy použijte frontu **short**.

```
$ psubmit short long_job
```

3. Průběh úlohy monitorujte příkazem **pinfo**. Úlohy ostatních uživatelů příkazem **pqstat**. Obsazení jednotlivých výpočetních uzlů pak příkazem **pnodes**.

Programy pro molekulové modelování

Nemesis

<https://lcc.ncbr.muni.cz/whitezone/development/nemesis/>

Program pro stavbu a vizualizaci molekul. Alfa verze pro Linux. Testovací verze pro MS Windows na vyžádání.

Avogadro

http://avogadro.openmolecules.net/wiki/Main_Page

Program pro stavbu a vizualizaci molekul. Volně dostupný pro MS Windows a Linux.

Přehled funkcionality: <https://www.youtube.com/watch?v=xdmLoBILmqs>

VMD

<http://www.ks.uiuc.edu/Research/vmd/>

Program pro vizualizaci molekul. Po bezplatné registraci dostupný pro MS Windows a Linux.

Molekel

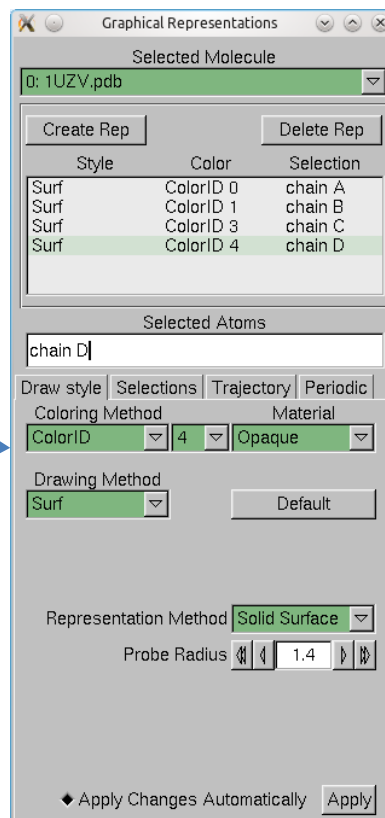
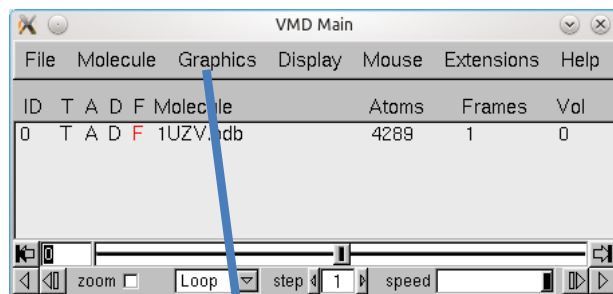
<http://molekel.cscs.ch/wiki/pmwiki.php>

Program pro vizualizaci molekulárních struktur.

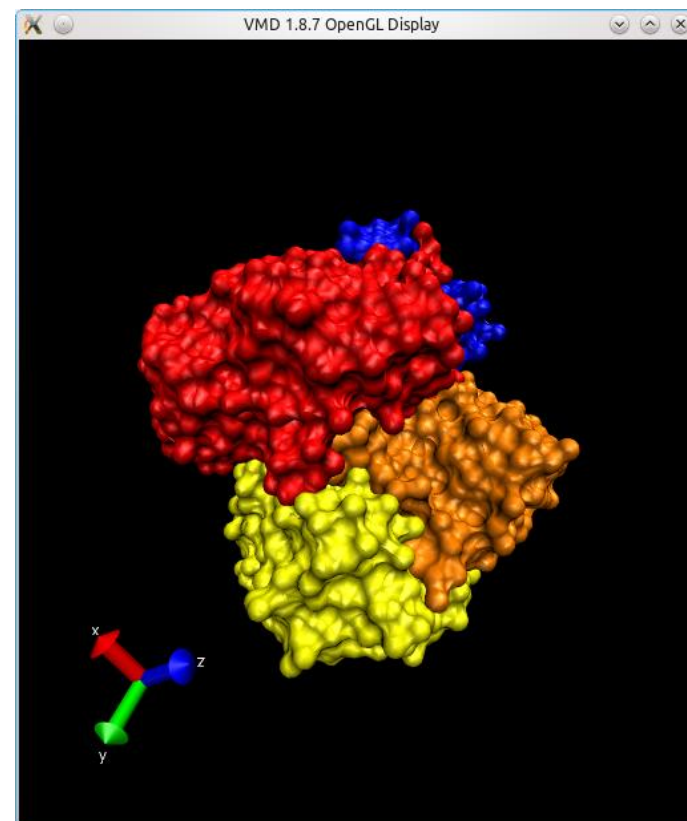
Program VMD

<http://www.ks.uiuc.edu/Research/vmd/>

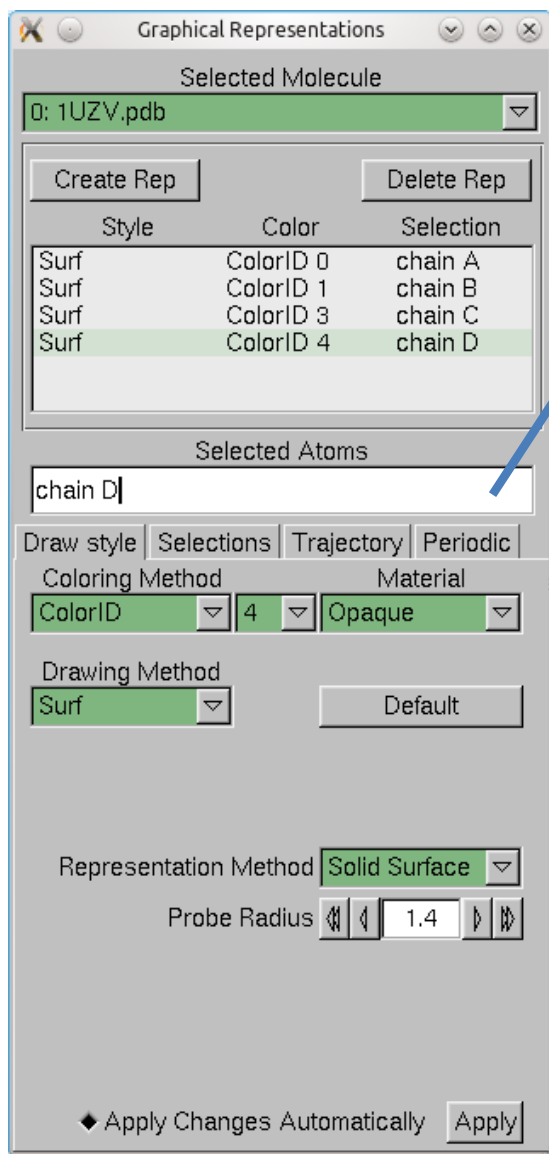
Program slouží k vizualizaci (bio)molekul a k analýze výsledků molekulárně dynamických simulací. Program je volně dostupný (vyžaduje registraci) a je dostupný i pro operační systém MS Windows.



Representation



Program VMD



Selekcce (volba) části molekuly:

- protein – zvolí všechny aminokyseliny
- water – zvolí všechny molekuly vody
- chain X – zvolí řetězec X
- resname X – zvolí residuum s názvem X
- resid X – zvolí residuum s číslem X

Příklady:

chain A
chain A B C
resname ASP GLU
resid 1
resid 1 to 100

Bližší informace:

C2150 Zpracování informací a vizualizace v chemii

PDB Databáze

www.pdb.org

The screenshot displays the PDB website interface in a Mozilla Firefox browser window. The address bar shows the URL <http://www.pdb.org/pdb/explore.do?structureId=1UZV>. The main content area features the entry title: **HIGH AFFINITY FUCOSE BINDING OF PSEUDOMONAS AERUGINOSA LECTIN II: 1.0 A CRYSTAL STRUCTURE OF THE COMPLEX**, with the ID **1UZV** prominently displayed. Below the title, the DOI is listed as [DOI:10.2210/pdb1uzv/pdb](https://doi.org/10.2210/pdb1uzv/pdb). A 'Primary Citation' section provides the full citation: **High affinity fucose binding of Pseudomonas aeruginosa lectin PA-III: 1.0 A resolution crystal structure of the complex combined with thermodynamics and computational chemistry approaches.** by Mitchell, E., Sabin, C.D., Snajdrova, L., Pokorna, M., Gautier, S., Perret, C., Hofr, C., Gilboa-Garber, N., Koca, J., Imberty, M., and Wimmerovaa. The citation is from *Journal: (2005) Proteins: Struct., Funct., Bioinf.* 58: 735. It also includes a PubMed link (15573375) and a DOI (10.1002/prot.20330). A 'Biological Assembly' section shows a 3D ribbon diagram of the protein structure, colored in shades of green and blue, with a fucose molecule bound to it. The left sidebar contains navigation links such as 'Home', 'News & Publications', and 'Deposition'.

Obsahuje struktury biomolekul určené metodami rentgenové strukturní analýzy, nukleární magnetické rezonance, teoretické modely.

Cvičení

PDB:

- Najděte protein s PDB kódem 1UZV

VMD:

- Zobrazte strukturu 1UZV
- Zvýrazněte jednotlivé monomerní jednotky komplexu (model: surf, selekce: chain A, chain B, ...)
- Zobrazte sekundární strukturu komplexu (NewCartoon, Color: Secondary Structure). Který strukturní element ve struktuře převládá?
- Zobrazte navázaný ligand (resname FUC). Kolik ligandů je v komplexu obsaženo?
- Zobrazte vápenaté ionty (resname CA). Kolik iontů je v komplexu obsaženo?