

C2110 *UNIX and programming*

Lesson 4 / Module 3

PS / 2020 Distance form of teaching: Rev1

Petr Kulhanek

kulhanek@chemi.muni.cz

National Center for Biomolecular Research, Faculty of Science
Masaryk University, Kamenice 5, CZ-62500 Brno

Text Editors

- **vi, vim, nano**
- **Graphic Text Editors (gedit, kwrite, kate)**

vi/vim, nano

Editor vi / vim is a standard text editor on UNIX-type operating systems. It only works in text mode and its use is **non-trivial**.

- It's a good idea to learn how to open a file, switch to edit mode, edit text, save your changes, and exit the editor.
- It enables scripting (use of variables, cycles, fields, associative fields), e.g., for creating automatic texts from read data.
- Despite you run the command vi in the classroom, the vim program will start automatically (Vi IMporoved)
- There is a difference in control between the original vi and vim.

Editor nano is the default text editor in some distributions (UBUNTU).

- Less versatile than vim
- More straightforward control

vi – Casics

Editor working modes



Start the editor

\$ vi start editor
\$ vi filename start editor and open file filename

Exit the editor

: q ends editor
: q! exits the editor without saving changes
: w saves file
: w filename saves file with name *filename*
:wq exits and saves the file

File changes

i text will be inserted **from** cursor position
a text will be inserted **behind** cursor position

Additional functionality - accompanying document!

nano

Start the editor

`$ nano` **start editor**

`$ nano filename` **start editor and opening file filename**

```
GNU nano 2.2.6          New Buffer          Modified
Toto je editor nano.
^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

More straightforward control - the menu at the bottom suggests possible actions. Use the combinations or individual letters to select an action

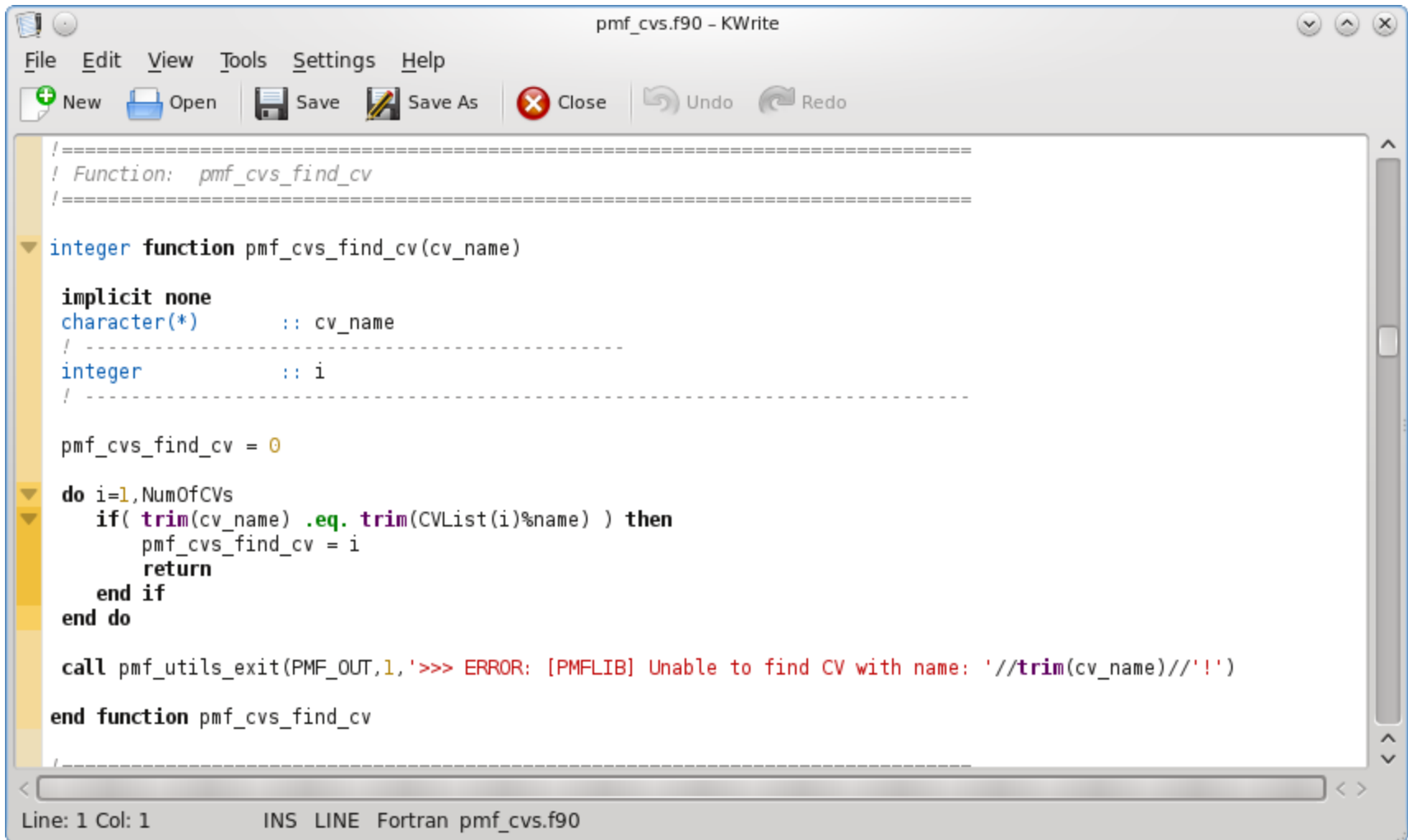
^letter – e.g. ^ X is a combination of Ctrl + X

M-letter – e.g., M-M is a combination of Alt + M

Exercise 1

1. On the WOLF cluster, create a text file using editor **vi** and insert a short text *Lorem Ipsum*.
2. Save the file as **li.txt**.
3. Copy the file to your personal computer and view it in a text editor. Solve any complications with ends of lines according to the type of OS on your computer.
4. Edit / change the file on your computer.
5. Copy the modified file to a WOLF cluster under the name **li2.txt**.
6. On the WOLF cluster, concurrently view the files **li.txt** and **li2.txt** in a text editor, each in a separate terminal.

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

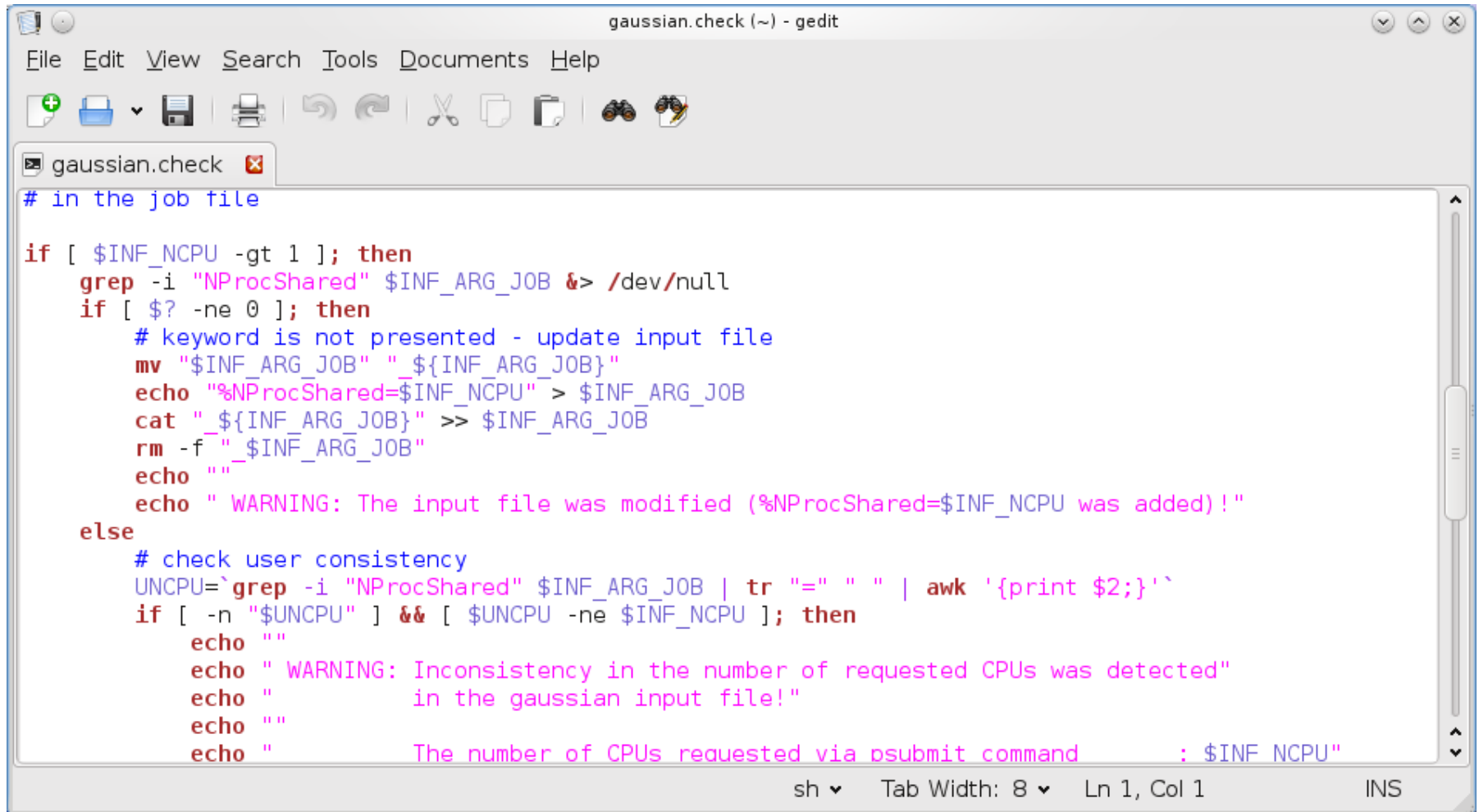
call pmf_utils_exit(PMF_OUT,1,'>>> ERROR: [PMFLIB] Unable to find CV with name: '//trim(cv_name)//'!')

end function pmf_cvs_find_cv

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

Extended functionality: **kate**

gedit



The image shows a window titled "gaussian.check (~) - gedit". The window contains a shell script 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 of the window shows "sh", "Tab Width: 8", "Ln 1, Col 1", and "INS".

Exercise 2

1. On the WOLF cluster in the editor vi, write a text that will contain ten lines. There will be two or more words on each line. Save the text to file **mydata.txt**.
2. By command wc verify that the file **mydata.txt** has ten lines.
3. Using pipe(s), write a sequence of commands that print only the number of words in the file on the screen **mojedata.txt**.
4. In a graphical text editor (of your choice), create a file that will contain ten words, each word on a new line. Save the text to a file **second_data.txt**.
5. Use the paste command to create the file **all_data.txt** which will contain the contents of the files **mydata.txt** and **second_data.txt** side by side.
6. By command wc verify that the file **all_data.txt** contains exactly ten lines.
7. Open the file **all_data.txt** in a graphical text editor and verify its contents visually.
8. Try working in individual text editors and choose the one that works best for you.

Text Editors in VM



Text Editors - Installation

Try individual text editors in your installation of Ubuntu 18.04 LTS. If they are not available, install them as follows:

```
$ sudo apt-get install vim
$ sudo apt-get install kwrite
$ sudo apt-get install kate
$ sudo apt-get install gedit
$ sudo apt-get install nano
```

If asked, enter the password for your account.

By default vi editor is installed in a compatible mode, which should be replaced by an extended version (vim). For the installation, see above.