

# C2110

## *Operační systém UNIX a základy programování*

5. lekce

Petr Kulhánek

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

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

## ➤ Textové editory

vi, grafické textové editory, kwrite, gedit, kate

## ➤ Procesy II

příkazy, spouštění úloh, ukončování úloh

# Zápočtový test I

---

# Zápočtový test I

# Textové editory

---

**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)
- přiložený 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

# kwrite

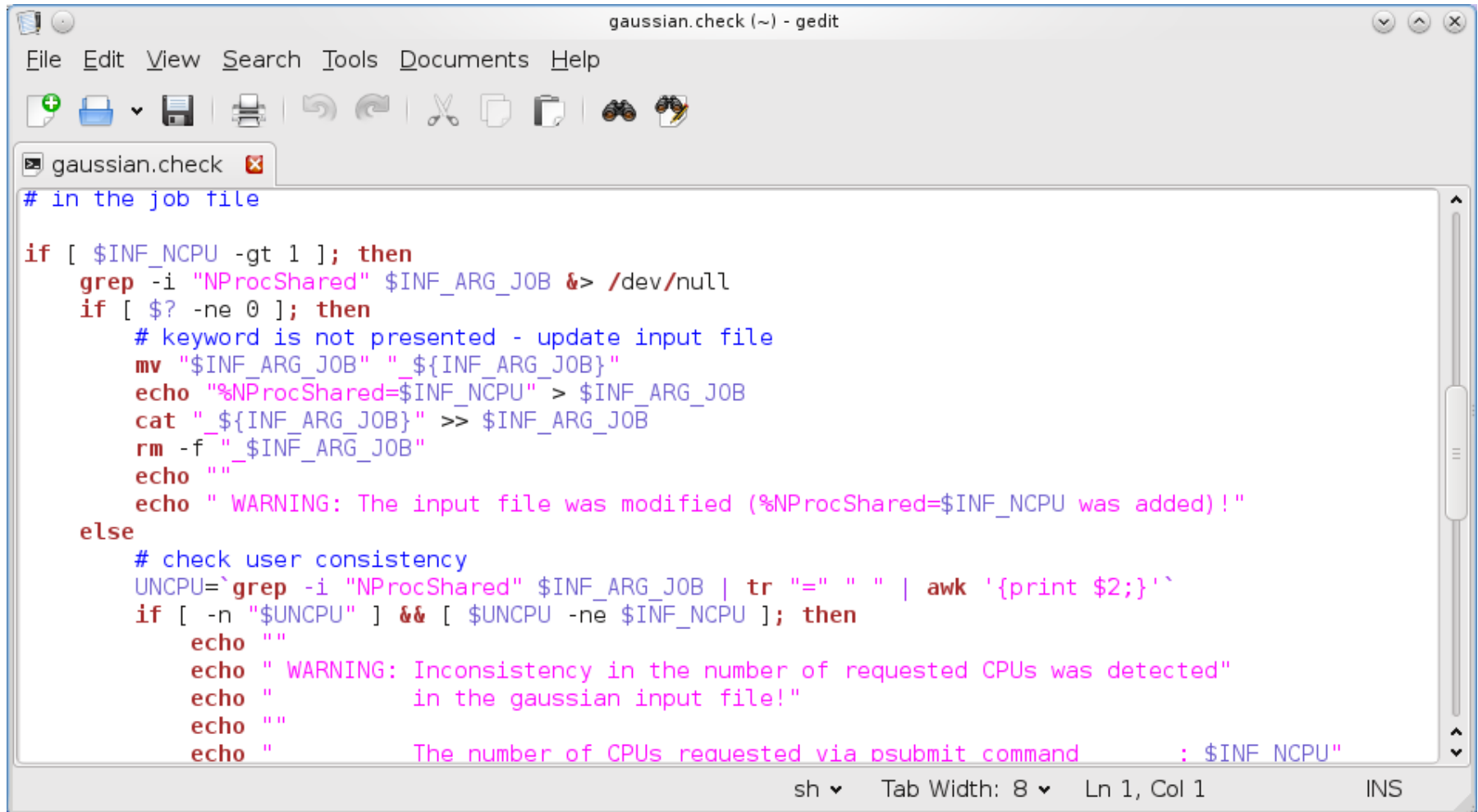
```
=====  
! Function: pmf_cvs_find_cv  
!=====  
integer function pmf_cvs_find_cv(cv_name)  
  
implicit none  
character(*)      :: cv_name  
! -----  
integer          :: i  
! -----  
  
pmf_cvs_find_cv = 0  
  
do i=1, NumOfCVs  
  if( trim(cv_name) .eq. trim(CVList(i)%name) ) then  
    pmf_cvs_find_cv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

Line: 1 Col: 1      INS LINE Fortran pmf\_cvs.f90

Rozšířená funkcionálníta: **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".

# Cvičení

1. V editoru **vi** napište text, který bude obsahovat deset řádků. Na každém řádku budou dvě a více slov. Text uložte do souboru **mojedata.txt**
2. Příkazem **wc** ověřte, že soubor **mojedata.txt** má skutečně deset řádků.
3. Za použití **rour(y)** napište sekvenci příkazů, které na obrazovku vypíší pouze počet slov v souboru **mojedata.txt**
4. V grafickém textovém editoru (dle vašeho výběru) vytvořte soubor, který bude obsahovat deset slov, každé slovo na novém řádku. Text uložte do souboru **druha\_data.txt**
5. Pomocí příkazu **paste** vytvořte soubor **vsechna\_data.txt** , který bude obsahovat obsah souborů **mojedata.txt** a **druha\_data.txt** vedle sebe.
6. Příkazem **wc** ověřte, že soubor **vsechna\_data.txt** obsahuje právě deset řádků.
7. Soubor **vsechna\_data.txt** otevřete v grafickém textovém editoru a jeho obsah ověřte vizuálně.

# Procesy II

---

# Příkazy

<b>top</b>	průběžně zobrazuje procesy setříděné podle zátěže procesoru (ukončení klávesou q)
<b>ps</b>	vypíše procesy běžící v daném terminálu nebo podle zadaných specifikací ( <code>ps -u user_name</code> )
<b>pstree</b>	vypíše procesy (stromový výpis)
<b>kill</b>	zašle signál procesu, lze použít k ukončení problematických programů
<b>nohup</b>	spustí proces bez interakce s terminálem
<b>sleep</b>	spustí proces , který čeká po zadanou dobu
<b>wait</b>	čeká na dokončení procesů na pozadí
<b>time</b>	vypíše délku běhu procesu
<b>ssh</b>	spustí příkaz na vzdáleném počítači
<b>jobs</b>	vypíše procesy na pozadí
<b>fg</b>	převede proces do popředí
<b>bg</b>	převede proces do pozadí
<b>disown</b>	odpojí proces od terminálu

# Spouštění příkazů a aplikací

## Příkazy a systémové aplikace

```
$ ls -l
```

zadááme pouze jméno příkazu nebo aplikace

```
$ cp soubor.txt soubor1.txt
```

příkaz

argumenty příkazu (mění chování příkazu nebo tvoří vstupní informace pro zpracování)

## Uživatelské programy a skripty

```
$ ./muj_script
```

jméno programu nebo skriptu udáváme včetně cesty (absolutní nebo relativní)

```
$ ~/bin/my_application
```

## Zrušení výpisu do terminálu

```
$ kwrite &> /dev/null
```

přesměrování výstupu uvádíme na konec příkazu (za argumenty)

## Spouštění aplikací na pozadí

```
$ gimp &
```

na konec (za argumenty a přesměrování) příkazu uvedeme ampersand

# Spouštění příkazů a aplikací, II

## Terminál (užitečné klávesové zkratky):

- Ctrl+C** běžícímu procesu zašle signál SIGINT (Interrupt), proces je ve většině případů násilně ukončen
- Ctrl+D** zavře vstupní proud spuštěného procesu
- Ctrl+Z** pozastaví běh procesu, další osud procesu lze kontrolovat pomocí příkazů **bg**, **fg**, **disown**

## Kde se nachází systémový příkaz:


**type** vypíše cestu k systémovému příkazu nebo programu

## Příklady:

```
$ type ls  
ls is /bin/ls
```

```
$ type pwd  
pwd is a shell builtin
```

příkaz pwd je implementován jako vnitřní příkaz shellu



# Příklady

```
$ ps -u kulhanek
PID TTY          TIME CMD
...
5440 pts/8      00:00:00 bash
5562 pts/8      00:00:00 kwrite
5566 pts/8      00:00:00 ps

$ kill 5562    # ukončí aplikaci kwrite

$ kwrite      # spustí aplikace kwrite na popředí
^Z           # pozastaví aplikaci
[1]+  Stopped                  kwrite
$ jobs       # vypíše aplikace na pozadí nebo
            # pozastavené aplikace
[1]+  Stopped                  kwrite
$ bg 1      # pozastavenou aplikaci 1 spustí na pozadí
[1]+  kwrite &
$ jobs
[1]+  Running                  kwrite &
```