C2110 *UNIX and programming*

Lesson 9 / Module 2

PS / 2020 Distance form of teaching: Rev3

Petr Kulhanek

kulhanek@chemi.muni.cz

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

for … in Loop

for ... in Loop

Commands in the block **do/done** (**command1,...**) are performed for each element in the list **LIST**. In then given run of the cycle, variable **VAR** contains the current element from the list **LIST.**

```
for VAR in LIST
     do
           command1 $VAR
           ...
     done
```

Compact notation:

```
for VAR in LIST; do
     command1 $VAR
     ...
done
```

for ... in Loop, Lists

```
for A in a b c; do
     echo $A
done
```

Loop cycles three times, during which it prints one by one characters **a**, **b**, **c**.

Lists of items can be created programmatically (using the commands given in inverted apostrophes).

```
for A in `ls *.eps`; do
     ./process_file $A
done
```

Command **process_fill** is executed for each file with an extension **.eps**, which is located in the current directory.

```
for A in `seq 1 0.25 10`; do
     printf "%8.3f\n" $A
done
```

Lists real numbers in the range from 1 to 10 in 0.25 increments. The numbers will be given to three decimal places, right-aligned and in an 8-character field. Documentation: man seq

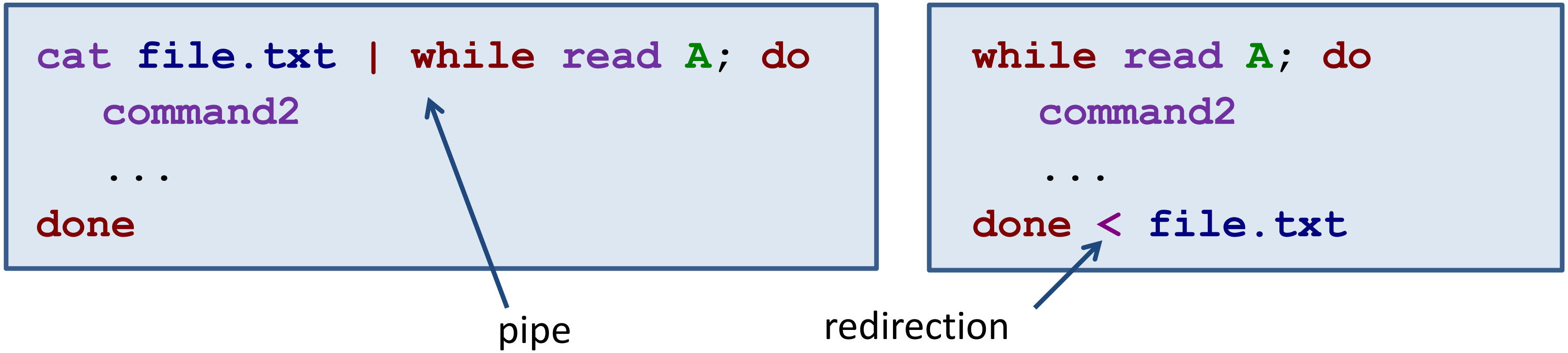
Exercise 1

1. Write a script that prints real numbers in the range from -10 to 10 in increments of 0.5. The numbers will be given including the sign, right-aligned in a field of 10 characters and with precision of one decimal place.
2. Write a script that prints the file name and the number of lines for each *.cpp file in directory /home/kulhanek/Documents/C2110/Lesson09/data1. Use for … in loop and use the ls command to list files to solve the task.
3. Modify the previous script so that for each *.f90 file in the /home/kulhanek/Documents/C2110/Lesson09/data2 directory the file name and the number of lines it contains is printed. Use for … in loop and use the find command to list files.

Loops and Redirections

Redirection and pipes - input

It is possible to transfer input data to the cycle using a pipe or redirection. Typical use is **loading file/input stream line by line**:



```
cat file.txt | while read A; do
    command2
    ...
done
```

```
while read A; do
    command2
    ...
done < file.txt
```

pipe

redirection

Redirection and pipes - input

Input can be "**consumed**"commands in the block of the loop. This can be prevented by redirecting standard input from /dev/null to a block of commands in {}.

```
cat file.txt | while read A; do
   {
      command2
      ...
   } < /dev/null
done
```

```
while read A; do
   {
      command2
      ...
   } < /dev/null
done < file.txt
```

Redirects and pipes - voutput

Redirecting output to a file:

```
for((I=1;I <= 10;I++)); do
     echo $I
done > file.txt
```

The output of all commands in the cycle is redirected to a file **file.txt**.

Pipe:

```
for((I=1;I <= 10;I++)); do
     echo $I
done | tee file.txt
```

The output of all commands in the cycle is connected to the standard input of command **tee**. It prints it to standard output and saves it to a file **file.txt** at the same time.

Redirects and pipes - examples

```
for((I=1;I <= 10;I++)); do
     echo $I
done > file.txt
```

same functionality

```
rm -f file.txt
for((I=1;I <= 10;I++)); do
     echo $I >> file.txt
done
```

```
for((I=1;I <= 10;I++)); do
     echo $I
     printf "N=%10d\n" $I
done > file.txt
```

different functionality

```
rm -f file.txt
for((I=1;I <= 10;I++)); do
     echo $I >> file.txt
     printf "N=%10d\n" $I
done
```

Exercise 2

1. Write a script that prints file name and the number of lines it contains for each *.f90 file in the directory /home/kulhanek/Documents/C2110/Lesson09 /data2. In the solution, use a pipe with while read and use the find command to list files.
2. Modify the shape rendering scripts (L08.M01.C01) so that the dimension of the shape is read from standard input and the resulting shape is printed to a file whose name is entered again by the user from standard input.

Homework

Homework I

Explain the different behavior of the following scripts. File data.txt contains five lines.

```
#!/bin/bash
I=0
cat data.txt | while read A; do
     I=$((I+1))
done
echo $I
```

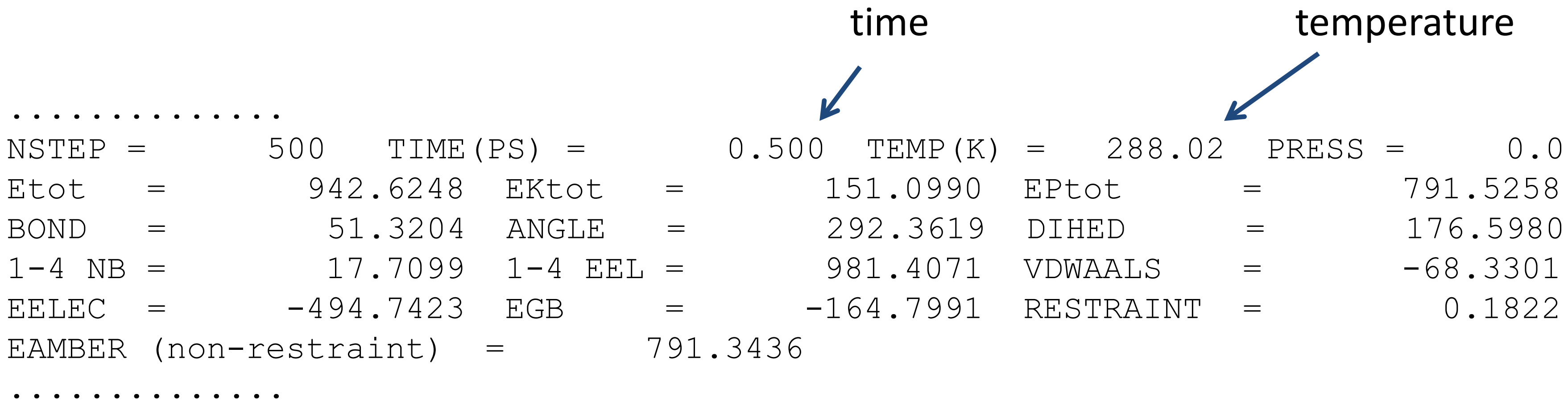
prints number 0

```
#!/bin/bash
I=0
while read A; do
     I=$((I+1))
done < data.txt
echo $I
```

prints number 5

Homework II

File rst.out (home/kulhanek/Documents/C2110/Lesson09/rst.out) contains results from molecular dynamics. The task is to extract the dependence of the temperature of the simulated system on time from the file and save this data in the temp file temp.out, which will contain exactly two columns. The first column will be the time and the second column will be the temperature.



time

temperature

```
...............
NSTEP =       500   TIME(PS) =       0.500  TEMP(K) =   288.02  PRESS =     0.0
Etot   =       942.6248  EKtot   =       151.0990  EPtot      =       791.5258
BOND   =        51.3204  ANGLE   =       292.3619  DIHED      =       176.5980
1-4 NB =        17.7099  1-4 EEL =       981.4071  VDWAALS    =       -68.3301
EELEC  =      -494.7423  EGB     =      -164.7991  RESTRAINT  =         0.1822
EAMBER (non-restraint)  =       791.3436
...............
```

ATTENTION: The script must not contain grep commands, awk nor their variants. Use command read and while loop.