

C2110 *UNIX and Programming*

12th lesson

awk – second part

Petr Kulhánek

kulhanek@chemi.muni.cz

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

Content

➤ **AWK**

- **Data file analysis**
- **Regular expressions**
- **Arrays**

Process of the Script Execution

```
1 BEGIN {  
    }  
2 {  
    }  
3 /PATTERN/ {  
    }  
4 END {  
    }
```

- BEGIN block (1) is executed (if it is part of the script) before the file analysis.
 - Record is read from the file. By default, the record is one line of the analyzed file or stream. The record is split into fields. By default, the fields are individual words of the record.
 - Block (2) is executed for the given record.
 - If the record matches the pattern, the block (3) is executed.
 - ... potential execution of other blocks ...
- Block END (4) is executed (if it is included in the script) after analysis of the whole file.

Each block is enclosed in curly braces {}.
The mentioned program blocks are optional.
By default, line is set as record.

Regular Expressions

```
/PATTERN/ {  
  
}
```

If PATTERN is found in the record, the block is executed.

Pattern is a **regular expression**.

Regular expression is a language that describes the structure of the text string. It is used to search text strings and to replace parts of strings.

Examples of simple regular expressions:

- TEXT** - is fulfilled if given record contains pattern TEXT (anywhere)
- ^TEXT** - is fulfilled if given record starts with pattern TEXT
- TEXT\$** - is fulfilled if given record ends with pattern TEXT

Exercise

1. From the file **rst.out**, extract evolution of temperature in time. Plot temperature in time using **gnuplot**.

```
NSTEP =      1000    TIME(PS) =          1.000    TEMP(K) =      305.69    PRESS =          0.0
  Etot   =          907.8481    EKtot    =          160.3711    EPtot    =          747.4770
  BOND   =          40.6154    ANGLE   =          273.9238    DIHED    =          164.5827
  1-4 NB =          14.6900    1-4 EEL =          973.2602    VDWAALS  =          -67.6091
  EELEC  =          -488.9232    EGB     =          -163.0629    RESTRAINT =          0.3793
  EAMBER (non-restraint) =          747.0977
```

2. From the file **rst.out**, extract evolution of total energy (**Etot**), kinetic energy (**EKtot**) and potential energy (**EPtot**) in time. Plot individual energies in time using **gnuplot**. Verify that the sum of potential and kinetic energy equals to the total energy.

Arrays

AWK uses associative arrays. Each element of the array can be accessed by using the key. The key may have any value or type. Key may be taken from variable.

Assigning values:

```
My_array[key] = value;
```

Obtaining values:

```
value = my_array[key];
```

Examples:

```
i = 5;  
my_array[i] = 15;  
print my_array[i];  
print my_array[5];
```

```
a = "word";  
my_array[a] = "value";  
print my_array["word"], my_array[i];
```

Arrays, ...

Browsing the list of keys:

```
for( variable in array ) {  
    print array[variable];  
    ...  
}
```

Executes a loop for each key that was used to save a value to the **array**. The key value is assigned to the **variable**.

Deleting a record by key:

```
delete array[key];
```

Exercise

1. From the file **rst.out**, extract **evolution of temperature in time**. The resulting file will not contain the last two values, which are the mean value and its fluctuations. **Plot** temperature in time using **gnuplot**.
2. From the file **rst.out**, extract **evolution of temperature in time** and calculate its **average value**. Compare the calculated value with the average value printed in the file **rst.out**. **Why do the values differ?**