

Home assignment 2

Configuration management

Task

The task is to create a **Python** script that can collect basic information about remote Linux machines, and can store the information in a file.

The virtual machine that can be downloaded from the course's website was updated to have all the necessary programs (e.g. `sfc` – configuration database, `wbemcli` – command line client). Pay attention to use the new v4 version of the virtual machine!

To complete the assignment, create a test environment with two instances of the virtual machine, try out some basic queries (like the ones in the syllabus and the reading material), and then implement and test the following script.

Name and parameters of the script

```
collect-os-detail.py -m MACHINES -o OUTPUT
```

The script has to use this name and these parameters. Use named parameters! The order of the parameters should not be fixed. The definitions of the parameters are:

- `MACHINES`: the path to a CSV file that contains the list of the remote machines.
- `OUTPUT`: the path to the output file. If this file exists, then the script should exit with an error.

Examples for valid uses of the script:

```
collect-os-detail.py -m machine.csv -o output.csv
```

```
collect-os-detail.py -o os-info.csv -m computers.csv
```

Input file

The input file is an UTF-8 encoded CSV text file containing the necessary information to connect to the remote machines. Example:

```
machineName,port,protocol,user,password  
192.168.250.128,5985,http,administrator,password  
server01,5985,http,meres,password2
```

(Storing the password as clear text is not an option in production environment, this is only done here to simplify the assignment and concentrate on the configuration management topic. In real situations, the password should be encrypted or other methods should be used.)

Output file

The output file is an UTF-8 encoded CSV text file containing the summary of the collected information. For each machine in the input file there is one line in the output (except when connection error occurs, see later). The memory has to be specified in MB.

Example:

```
"Machine","NumberOfProcessor","Memory","NumberOfProcess"  
"192.168.1.1"1","1024","50"  
"server01","2","4096","43"
```

Further constraints

- Implement error handling in your script! For example, check whether the specified input file exists.
- The script should handle connection errors with a remote machine. In this case no row should be written in the output file for that machine, but a warning message should be displayed on the standard output, and the script should continue its execution.
- Add comment to your script explaining its important parts!