1. Introduction

My company is selling a product which is permanently generating logs tracking its activity. When a customer experiences a failure, the logs have to be checked and fully analyzed. The following program was created to be executed after customer communicates the issue and its main goal is to accelerating a troubleshooting process and speeding up a customer's solution response.
The code is executed on customer premises and provides a first log's check output and some primary conclusions which can be useful for latter and quicker decisions taken.

2. Requirements

The program was created with Python 3.6.1 under Anaconda distribution using Spyder 3.1.4 v.
The code requires the following modules:

```
from pathlib import Path
import re,glob
import psutil
import sqlite3
from datetime import datetime
```

A basic description for each module follows:
path: required when a file local paths is required.
re: (regular expressions) Required when finding and searching key-words.
glob: required when wanted to find the pathnames matching the logs files specified pattern according to the rules used by the Unix shell.
psutil: retrieves information on running processes and system utilization. Here is required for CPU percentage usage.
sqlite3: database used for historical CPU usage records.
datetime: supplies classes for manipulating the current dates and time.

Other requirements:
On the first check the program searches for some particular key-words which in case of find them on the product's logs, it will be required the execution of a second, more elaborated script developed by my company's R&D Team.
To do so a keys.txt file containing all the critical words has to be also on place for the program correct execution. The keys.txt file is attached to this report.

3. Description

The program starts asking the directory's name where the logs and the keys.txt file are located (some customers personalize the default directory where the logs are located)

Then, because we provide two main different product versions, the logs names are also different. For version 4 and lower, the system log to check is named "sysinfo.txt". For version 5 and upper the system log to be check is named "node_1.rladmin". If one exists the other doesn't. Therefore the program starts getting the existence of one of these two files for a later analysis.

The program's main goal is to find out if the product contains some particular words and therefore is in an unstable state. If so, then the program runs a second check telling us if it's safe to offer to the customer to run the program developed by R&D or not. This more elaborated program has to be executed under particular circumstances and it not safe to run it if our product does not requires it. In other words: my program has to determine if the product is unstable or not and if so checking also if it is safe running another more resource intensive program or not.

In case of these two checks are positive, then it's required to get the current server's resources levels of usage (e.g. CPU usage percentage) and inserting the values into a SQLite3 Database. The reason for this is that this DB can be consulted for future references in case of need.

For a better understanding, a summarize explanation follows:

First: the program starts asking the directory where the logs are locate (some customers require to creating a remote directory for storing the logs)

Second: determine which file exists "sysinfo.txt "or "node_1.rladmin" and store the correct one.

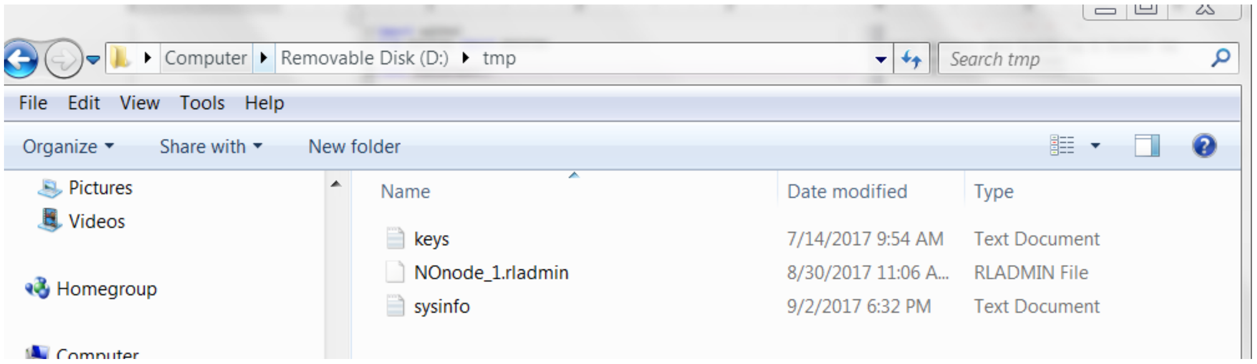
Third: open the file and determine if contains one of the key-words provided on keys.txt file. If so, then the program continues. If not, the program leaves.

Fourth: check if it's safe running the second program. This is determined if the Key word founded previously is on RUNNING or PENDING state. If RUNNING, then it is safe running the second program and the current program continues. If PENDING then it is not recommended running the second program and this one leaves.

Fifth: In case of RUNNING on previous point, it is required to getting the current CPU usage. Here it is important to get each core usage, therefore the program has to determine the value of each existing core. Then a Table on a SQLite3 DB is created, opened and sequence of values are inserted: the current server time (used as a KEY), the number of CPU Core and finally that Core usage.
This table is important for future references in case of the unstable product status high frequency: it can give us the idea of how often this happens and the sever CPU usage each time.

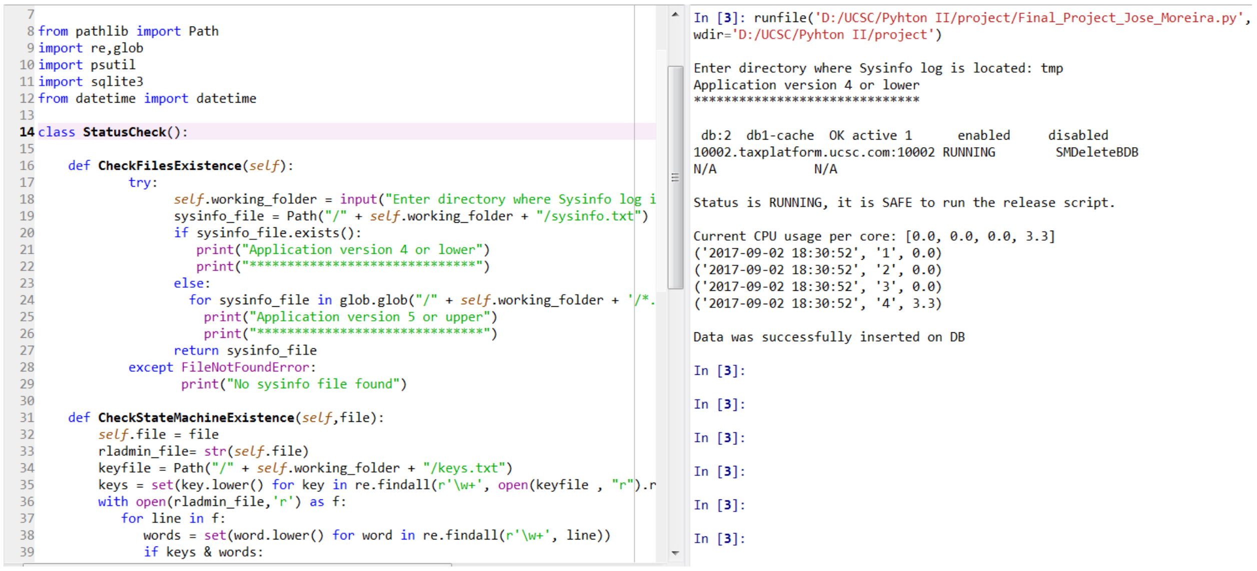
4. Screenshots of the program output

Scenario 1:
sysinfo.txt file exists (for product's version 4 and lower)
Keyword found on the logs: *SMDeleteBDB*
Second keyword: RUNNING (which means it's safe running the second program and the CPU data should be collected and stored on the DB)


Computer ▸ Removable Disk (D:) ▸ tmp
Search tmp
File Edit View Tools Help
Organize Share with New folder
Pictures
Videos
Homegroup
Computer

Name	Date modified	Type
keys	7/14/2017 9:54 AM	Text Document
NOnode_1.rladmin	8/30/2017 11:06 A...	RLADMIN File
sysinfo	9/2/2017 6:32 PM	Text Document

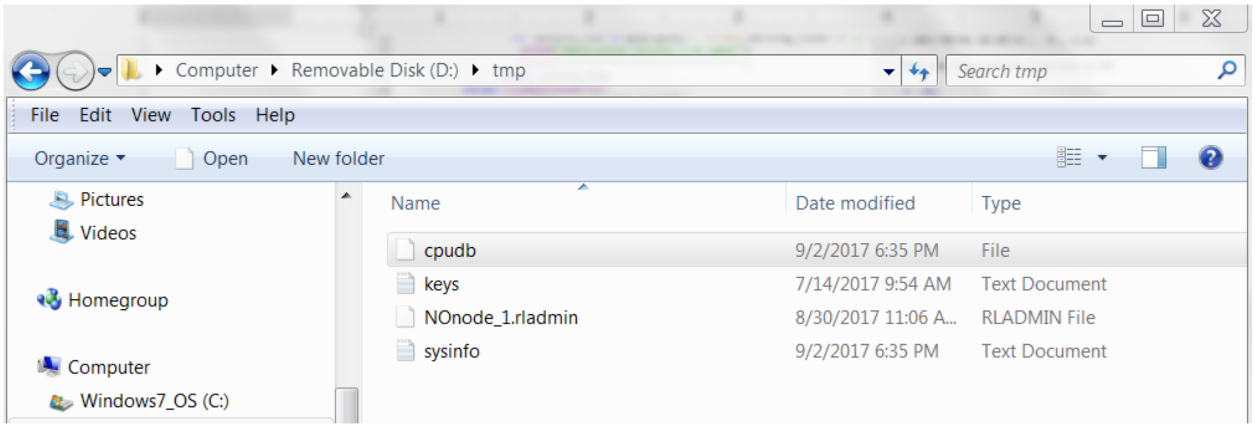
Execution:



```python

from pathlib import Path
import re,glob
import psutil
import sqlite3
from datetime import datetime

class StatusCheck():

    def CheckFilesExistence(self):
            try:
                  self.working_folder = input("Enter directory where Sysinfo log i
                  sysinfo_file = Path("/" + self.working_folder + "/sysinfo.txt")
                  if sysinfo_file.exists():
                     print("Application version 4 or lower")
                     print("******************************")
                  else:
                    for sysinfo_file in glob.glob("/" + self.working_folder + '/*.
                      print("Application version 5 or upper")
                      print("******************************")
                  return sysinfo_file
            except FileNotFoundError:
                   print("No sysinfo file found")

    def CheckStateMachineExistence(self,file):
        self.file = file
        rladmin_file= str(self.file)
        keyfile = Path("/" + self.working_folder + "/keys.txt")
        keys = set(key.lower() for key in re.findall(r'\w+', open(keyfile , "r").r
        with open(rladmin_file,'r') as f:
           for line in f:
              words = set(word.lower() for word in re.findall(r'\w+', line))
              if keys & words:
```

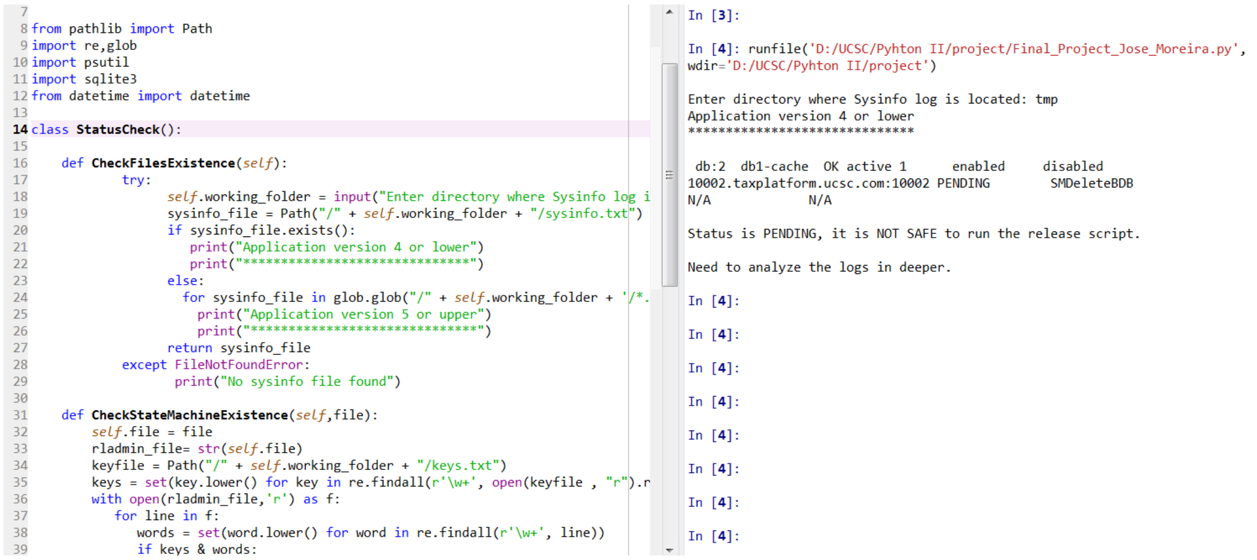
```
In [3]: runfile('D:/UCSC/Pyhton II/project/Final_Project_Jose_Moreira.py',
wdir='D:/UCSC/Pyhton II/project')

Enter directory where Sysinfo log is located: tmp
Application version 4 or lower
******************************


 db:2  db1-cache  OK active 1      enabled     disabled
10002.taxplatform.ucsc.com:10002 RUNNING        SMDeleteBDB
N/A             N/A

Status is RUNNING, it is SAFE to run the release script.

Current CPU usage per core: [0.0, 0.0, 0.0, 3.3]
('2017-09-02 18:30:52', '1', 0.0)
('2017-09-02 18:30:52', '2', 0.0)
('2017-09-02 18:30:52', '3', 0.0)
('2017-09-02 18:30:52', '4', 3.3)

Data was successfully inserted on DB

In [3]:

In [3]:

In [3]:

In [3]:

In [3]:

In [3]:
```

DB (named "cpudb") was created on /tmp directory:



Name	Date modified	Type
cpudb	9/2/2017 6:35 PM	File
keys	7/14/2017 9:54 AM	Text Document
NOnode_1.rladmin	8/30/2017 11:06 A...	RLADMIN File
sysinfo	9/2/2017 6:35 PM	Text Document

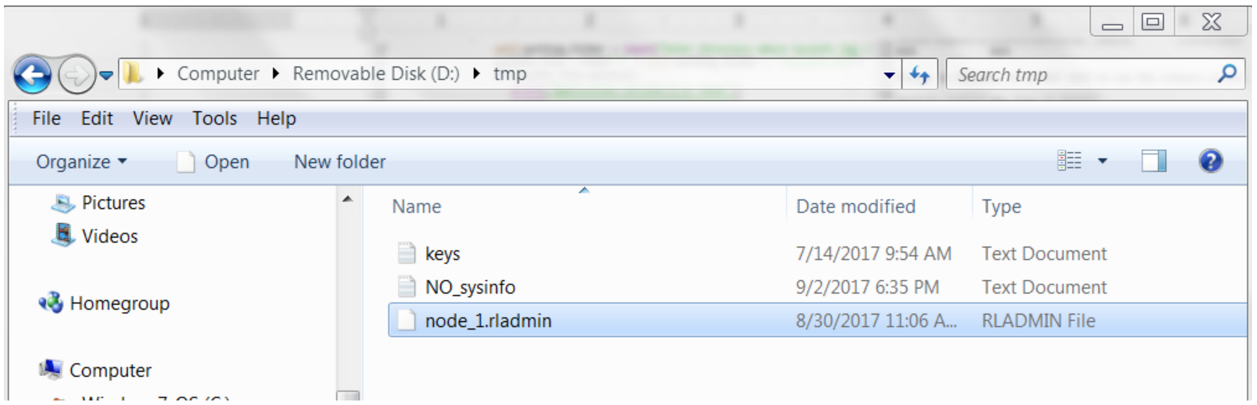
Scenario 2:
sysinfo.txt file exists (for product's version 4 and lower)
Keyword found: *SMDeleteBDB*
Second keyword: PENDING (which means it is NOT safe running the second program and program leaves)



```python

from pathlib import Path
import re,glob
import psutil
import sqlite3
from datetime import datetime

class StatusCheck():

    def CheckFilesExistence(self):
            try:
                  self.working_folder = input("Enter directory where Sysinfo log i
                  sysinfo_file = Path("/" + self.working_folder + "/sysinfo.txt")
                  if sysinfo_file.exists():
                     print("Application version 4 or lower")
                     print("******************************")
                  else:
                    for sysinfo_file in glob.glob("/" + self.working_folder + '/*.
                      print("Application version 5 or upper")
                      print("******************************")
                  return sysinfo_file
            except FileNotFoundError:
                   print("No sysinfo file found")

    def CheckStateMachineExistence(self,file):
        self.file = file
        rladmin_file= str(self.file)
        keyfile = Path("/" + self.working_folder + "/keys.txt")
        keys = set(key.lower() for key in re.findall(r'\w+', open(keyfile , "r").r
        with open(rladmin_file,'r') as f:
           for line in f:
              words = set(word.lower() for word in re.findall(r'\w+', line))
              if keys & words:
```

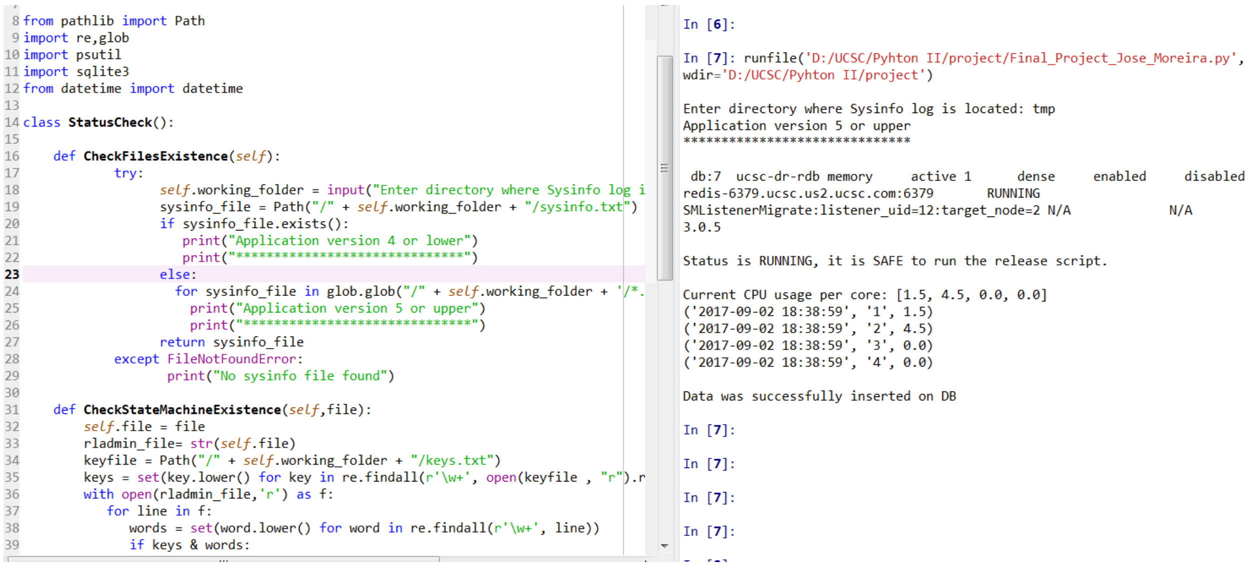
```
In [3]:

In [4]: runfile('D:/UCSC/Pyhton II/project/Final_Project_Jose_Moreira.py',
wdir='D:/UCSC/Pyhton II/project')

Enter directory where Sysinfo log is located: tmp
Application version 4 or lower
******************************


 db:2  db1-cache  OK active 1      enabled     disabled
10002.taxplatform.ucsc.com:10002 PENDING        SMDeleteBDB
N/A              N/A

Status is PENDING, it is NOT SAFE to run the release script.

Need to analyze the logs in deeper.

In [4]:

In [4]:

In [4]:

In [4]:

In [4]:

In [4]:

In [4]:

In [4]:
```

Scenario 3:
node_1.rladmin file exists (for product's version 5 and upper)
Keyword found: *SMDeleteBDB*
Second keyword: RUNNING (it is safe running the second program and the CPU data is collected and stored)



Name	Date modified	Type
keys	7/14/2017 9:54 AM	Text Document
NO_sysinfo	9/2/2017 6:35 PM	Text Document
node_1.rladmin	8/30/2017 11:06 A...	RLADMIN File

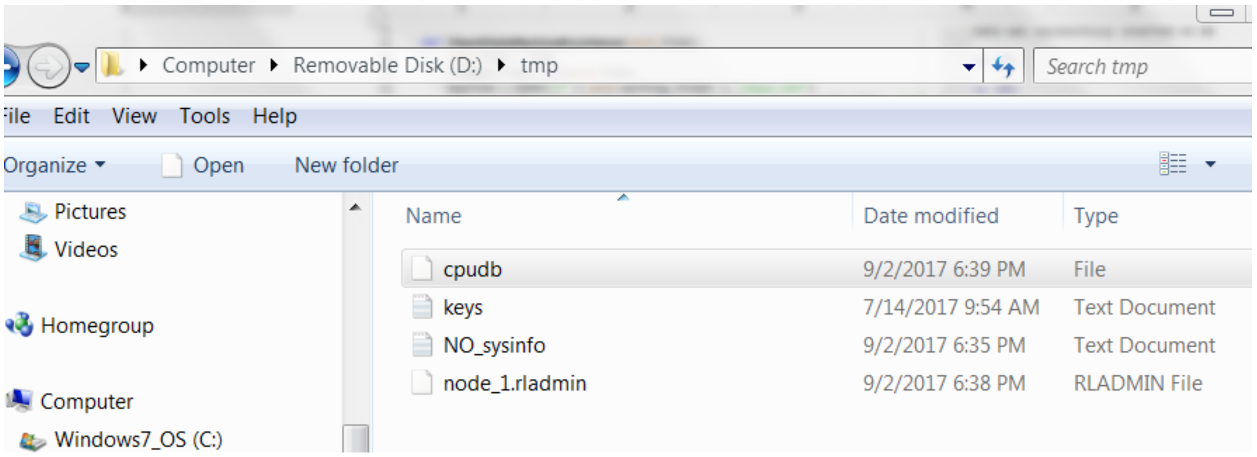
Execution:



```python
from pathlib import Path
import re,glob
import psutil
import sqlite3
from datetime import datetime

class StatusCheck():

    def CheckFilesExistence(self):
            try:
                  self.working_folder = input("Enter directory where Sysinfo log i
                  sysinfo_file = Path("/" + self.working_folder + "/sysinfo.txt")
                  if sysinfo_file.exists():
                     print("Application version 4 or lower")
                     print("*******************************")
                  else:
                    for sysinfo_file in glob.glob("/" + self.working_folder + '/*.
                      print("Application version 5 or upper")
                      print("*******************************")
                  return sysinfo_file
            except FileNotFoundError:
                   print("No sysinfo file found")

    def CheckStateMachineExistence(self,file):
        self.file = file
        rladmin_file= str(self.file)
        keyfile = Path("/" + self.working_folder + "/keys.txt")
        keys = set(key.lower() for key in re.findall(r'\w+', open(keyfile , "r").r
        with open(rladmin_file,'r') as f:
           for line in f:
              words = set(word.lower() for word in re.findall(r'\w+', line))
              if keys & words:
```

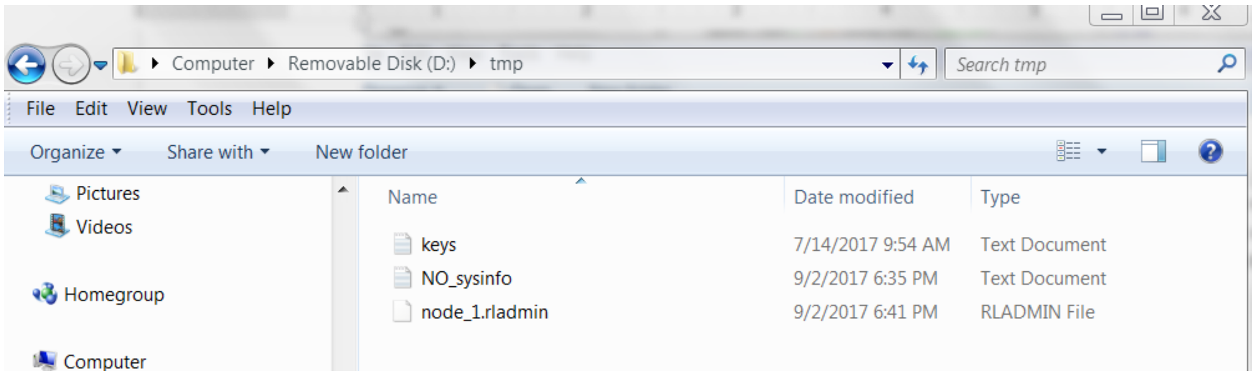
```
In [6]:

In [7]: runfile('D:/UCSC/Pyhton II/project/Final_Project_Jose_Moreira.py',
wdir='D:/UCSC/Pyhton II/project')

Enter directory where Sysinfo log is located: tmp
Application version 5 or upper
******************************

 db:7  ucsc-dr-rdb memory     active 1      dense     enabled     disabled
redis-6379.ucsc.us2.ucsc.com:6379       RUNNING
SMListenerMigrate:listener_uid=12:target_node=2 N/A             N/A
3.0.5

Status is RUNNING, it is SAFE to run the release script.

Current CPU usage per core: [1.5, 4.5, 0.0, 0.0]
('2017-09-02 18:38:59', '1', 1.5)
('2017-09-02 18:38:59', '2', 4.5)
('2017-09-02 18:38:59', '3', 0.0)
('2017-09-02 18:38:59', '4', 0.0)

Data was successfully inserted on DB

In [7]:

In [7]:

In [7]:

In [7]:
```

DB is created:



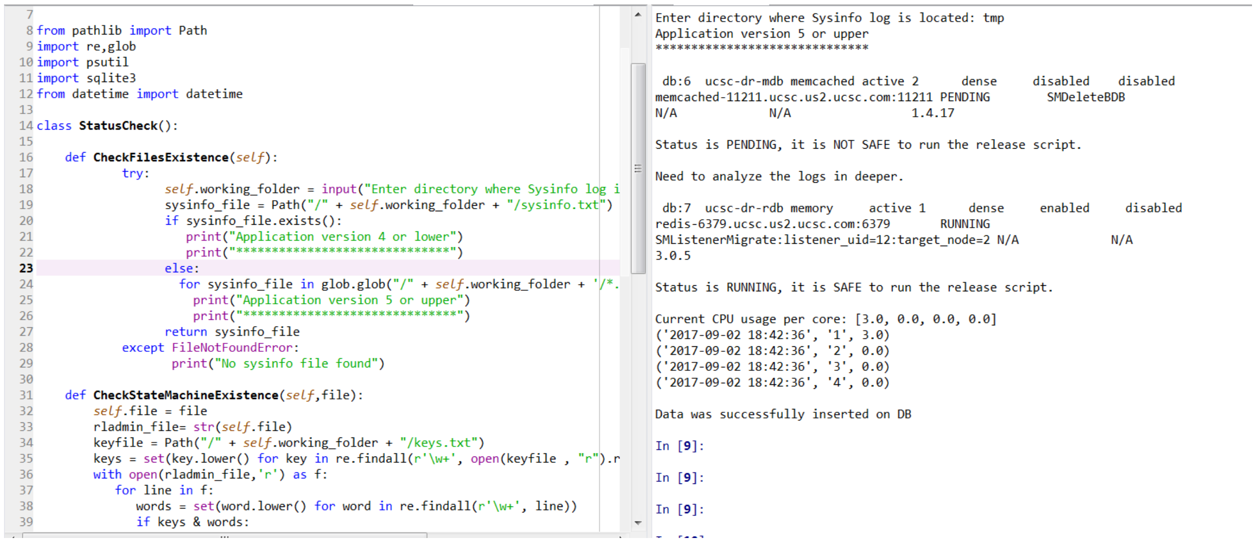
Name	Date modified	Type
cpudb	9/2/2017 6:39 PM	File
keys	7/14/2017 9:54 AM	Text Document
NO_sysinfo	9/2/2017 6:35 PM	Text Document
node_1.rladmin	9/2/2017 6:38 PM	RLADMIN File

Scenario 4:
node_1.rladmin file exists (version 5 and upper)
Keywords found: *SMDeleteBDB* and *SMDeleteDB* (2 key-words were found now)
Second keyword: both found RUNNING and PENDING.



Name	Date modified	Type
keys	7/14/2017 9:54 AM	Text Document
NO_sysinfo	9/2/2017 6:35 PM	Text Document
node_1.rladmin	9/2/2017 6:41 PM	RLADMIN File

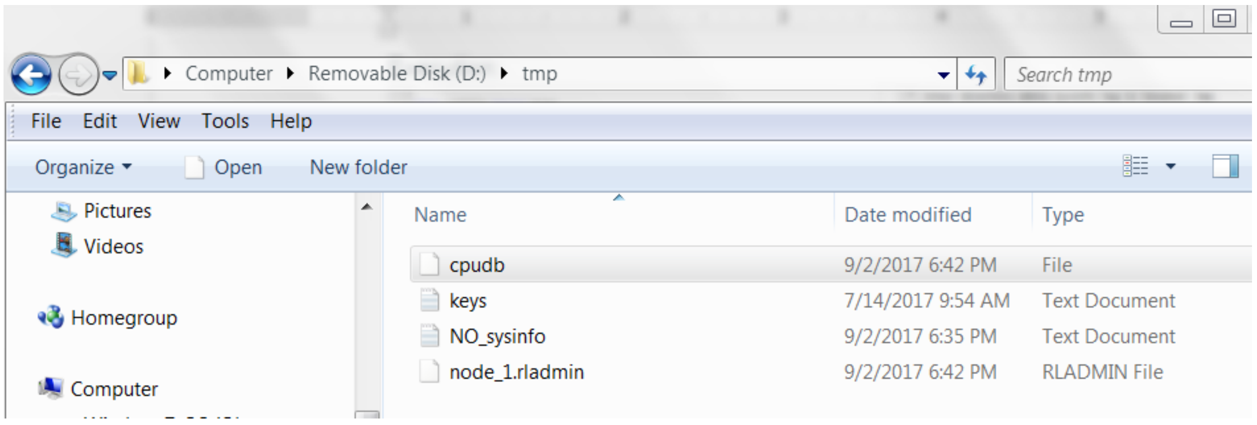
Execution:



```python

from pathlib import Path
import re,glob
import psutil
import sqlite3
from datetime import datetime

class StatusCheck():

    def CheckFilesExistence(self):
            try:
                  self.working_folder = input("Enter directory where Sysinfo log i
                  sysinfo_file = Path("/" + self.working_folder + "/sysinfo.txt")
                  if sysinfo_file.exists():
                     print("Application version 4 or lower")
                     print("*****************************")
                  else:
                    for sysinfo_file in glob.glob("/" + self.working_folder + '/*.
                      print("Application version 5 or upper")
                      print("*****************************")
                  return sysinfo_file
            except FileNotFoundError:
                     print("No sysinfo file found")

    def CheckStateMachineExistence(self,file):
        self.file = file
        rladmin_file= str(self.file)
        keyfile = Path("/" + self.working_folder + "/keys.txt")
        keys = set(key.lower() for key in re.findall(r'\w+', open(keyfile , "r").r
        with open(rladmin_file,'r') as f:
           for line in f:
              words = set(word.lower() for word in re.findall(r'\w+', line))
              if keys & words:
```

```
Enter directory where Sysinfo log is located: tmp
Application version 5 or upper
******************************

 db:6  ucsc-dr-mdb memcached active 2      dense     disabled    disabled
memcached-11211.ucsc.us2.ucsc.com:11211 PENDING        SMDeleteBDB
N/A             N/A                 1.4.17

Status is PENDING, it is NOT SAFE to run the release script.

Need to analyze the logs in deeper.

 db:7  ucsc-dr-rdb memory     active 1      dense     enabled     disabled
redis-6379.ucsc.us2.ucsc.com:6379       RUNNING
SMListenerMigrate:listener_uid=12:target_node=2 N/A            N/A
3.0.5

Status is RUNNING, it is SAFE to run the release script.

Current CPU usage per core: [3.0, 0.0, 0.0, 0.0]
('2017-09-02 18:42:36', '1', 3.0)
('2017-09-02 18:42:36', '2', 0.0)
('2017-09-02 18:42:36', '3', 0.0)
('2017-09-02 18:42:36', '4', 0.0)

Data was successfully inserted on DB

In [9]:

In [9]:

In [9]:
```

All states are found and according to each of them, the program made the suggestions.



Name	Date modified	Type
cpudb	9/2/2017 6:42 PM	File
keys	7/14/2017 9:54 AM	Text Document
NO_sysinfo	9/2/2017 6:35 PM	Text Document
node_1.rladmin	9/2/2017 6:42 PM	RLADMIN File

5. **Conclusion**

The program utilizes and implements the following Python programming resources:
Functions, Classes, Try-except, regular expressions, data type cast, file input and output, db creation, opening and inserting, list comprehension, input data and list process.

The resulted script is just the beginning of a future larger script assigned by my own manager as a first quick and visual check for a real customer’s issues.
It is planned to keep improving the current code and also keep adding extra functions/features for a better quick initial understanding of real product problems.
As example, in the future it’s planned to be able to run the similar analysis but on every servers involved on executing the product; getting more information from a other different logs; adding to the SQLite3 DB more server resources consumption such as memory; sending the result via email to the customer’s contact with a better formatted version than the current one; etc.

6. **Python program and other files required below:**

6.1 Code:

```
"""
Created on Thu Aug 31 2017
Author: Jose Moreira
Course: Python for Programmers - UCSC- Extension
"""

from pathlib import Path
import re,glob
import psutil
import sqlite3
from datetime import datetime


class StatusCheck():

  def CheckFilesExistence(self):
      try:
         self.working_folder = input("Enter directory where Sysinfo log is located: ")
         sysinfo_file = Path("/" + self.working_folder + "/sysinfo.txt")
         if sysinfo_file.exists():
           print("Application version 4 or lower")
           print("******************************")
         else:
          for sysinfo_file in glob.glob("/" + self.working_folder + '/*.rladmin'):
            print("Application version 5 or upper")
            print("******************************")
         return sysinfo_file
```

```
            except FileNotFoundError:
                print("No sysinfo file found")


    def CheckStateMachineExistence(self,file):
        self.file = file
        rladmin_file= str(self.file)
        keyfile = Path("/" + self.working_folder + "/keys.txt")
        keys = set(key.lower() for key in re.findall(r'\w+', open(keyfile , "r").readline()))
        with open(rladmin_file,'r') as f:
          for line in f:
            words = set(word.lower() for word in re.findall(r'\w+', line))
            if keys & words:
                print("\n",line, end='')
            if ' PENDING ' in line:
                print("\nStatus is PENDING, it is NOT SAFE to run the release script.")
                print("\nNeed to analyze the logs in deeper.")
            elif ' RUNNING ' in line:
                print("\nStatus is RUNNING, it is SAFE to run the release script.")
                self.CheckResources()


    def CheckResources(self):
        cpu_dict = []
        cpu_dict = psutil.cpu_percent(interval=1, percpu=True)
        print("\nCurrent CPU usage per core:",cpu_dict)
        number_cores = len(cpu_dict)
        try:
            for x in range(number_cores):
                n1=(cpu_dict[x])
                conn = sqlite3.connect('/tmp/cpudb')
                c = conn.cursor()
                c.execute('drop table if exists cpurecords')
                conn.commit()
                c.execute('create table cpurecords ([timestamp] timestamp, cpu_number text,cpu float)')
                c.execute('insert into cpurecords values(?,?,?)',(datetime.now().strftime("%Y-%m-%d
%H:%M:%S"),x+1,n1))
                conn.commit()
                c.execute ('select * from cpurecords')
                for row in c:
                    print(row)
            print("\nData was successfully inserted on DB")
        except:
            print("ERROR. It was unable to write on DB")
        finally:
```

```
        c.close()

st = StatusCheck()
file = st.CheckFilesExistence()
st.CheckStateMachineExistence(file)
```

6.2 keys.txt:

```
SMCreateBDB,SMDeleteBDB,SMListenerFailover,SMUcscFailover,SMUcscMigrate,SMSlaveUcscMigrate,S
MListenerMigrate,SMStandaloneUcscMigrate,SMReshard,SMUpdateBDB
```

6.3 node_1.raldmin:

```
UcscLabs Node Information
Wed Aug 30 14:06:06 EDT 2017
------------------------------------------------------------
rladmin status extra all:
CLUSTER:
OK. Cluster master: 1 (10.192.16.51)
Cluster health: OK, [0, 0.06666666666666667, 0.016666666666666666]
failures/minute - avg1 0.00, avg15 0.07, avg60 0.02.

CLUSTER NODES:
NODE:ID ROLE   ADDRESS      EXTERNAL_ADDRESS          HOSTNAME MASTERS SLAVES SHARDS CORES
FREE_RAM      VERSION        RACK-ID STATUS
*node:1 master 10.192.16.51 10.192.16.51,10.192.144.51 rlec1    0      0      0/1024 6     6.9GB/7.64GB
4.5.0-40.rhel7  -       OK
node:2  slave  10.192.16.52 10.192.16.52,10.192.144.52 rlec2    1      0      1/1024 6     6.96GB/7.64GB
4.5.0-40.rhel7  -       OK
node:3  slave  10.192.16.53 10.192.16.53,10.192.144.53 None     0      1      1/1024 6     7.07GB/7.64GB
4.3.0-219.rhel7 -       OK, OLD VERSION
node:4  slave  10.192.16.54 10.192.16.54,10.192.144.54 None     2      0      2/1024 6     7GB/7.64GB
4.3.0-219.rhel7 -       OK, OLD VERSION
node:5  slave  10.192.16.55 10.192.16.55,10.192.144.55 None     0      0      0/1024 6     6.53GB/7.64GB
4.3.0-219.rhel7 -       OK, OLD VERSION

DATABASES:
DB:ID NAME          TYPE      STATUS SHARDS PLACEMENT REPLICATION PERSISTENCE ENDPOINT
EXEC_STATE EXEC_STATE_MACHINE                         BACKUP_PROGRESS MISSING_BACKUP_TIME
UCSC_VERSION
```

```
db:6  ucsc-dr-mdb memcached active 2      dense     disabled    disabled    memcached-
11211.ucsc.us2.ucsc.com:11211 PENDING       SMDeleteBDB                              N/A           N/A
1.4.17
db:7  ucsc-dr-rdb memory     active 1     dense     enabled     disabled    ucsc-6379.ucsc.us2.ucsc.com:6379
RUNNING    SMListenerMigrate:listener_uid=12:target_node=2 N/A           N/A              3.0.5

ENDPOINTS:
DB:ID  NAME           ID          NODE    ROLE    SSL      WATCHDOG_STATUS
db:6   ucsc-dr-mdb   endpoint:10   node:3  master  No       OK
db:7   ucsc-dr-rdb   endpoint:11   node:3  slave   No       OK
db:7   ucsc-dr-rdb   endpoint:12   node:2  master  No       OK

SHARDS:
error: cluster is not responding, please try again.
```

6.4 sysinfo.txt:

```
UcscLabs Node Information
Tue Jun 13 20:33:35 CDT 2017
------------------------------------------------------------
OS and platform:
Linux USSLTC1910V.DTTSOL-EAST.COM 3.10.0-327.18.2.el7.x86_64 #1 SMP Fri Apr 8 05:09:53 EDT 2016
x86_64 x86_64 x86_64 GNU/Linux
Red Hat Enterprise Linux Server release 7.2 (Maipo)
------------------------------------------------------------
Cluster status:
CLUSTER:
OK. Cluster master: 2 (10.25.65.47)
Cluster health: OK, [0, 0.0, 0.13333333333333333]
failures/minute - avg1 0.00, avg15 0.00, avg60 0.13.

CLUSTER NODES:
NODE:ID ROLE   ADDRESS     EXTERNAL_ADDRESS SHARDS MASTERS SLAVES CORES FREE_RAM
VERSION         RACK-ID STATUS
node:1  slave  10.25.65.46 10.25.65.46      2      2       0      8     61.75GB/62.76GB 4.3.0-219.rhel7 rack1   OK
node:2  master 10.25.65.47 10.25.65.47      0      0       0      8     61.69GB/62.76GB 4.5.0-22.rhel7  rack2   OK
node:3  slave  10.25.65.48 10.25.65.48      2      0       2      8     61.79GB/62.76GB 4.5.0-22.rhel7  rack3   OK

DATABASES:
DB:ID NAME       TYPE  STATUS SHARDS REPLICATION PERSISTENCE ENDPOINT
EXEC_STATE EXEC_STATE_MACHINE            BACKUP_PROGRESS MISSING_BACKUP_TIME
```

```
db:2  db1-cache  OK active 1      enabled     disabled    10002.taxplatform.ucsc.com:10002 RUNNING
SMDeleteBDB                         N/A           N/A
db:3  db2-opaop  OK active 1      enabled     disabled    10044.taxplatform.ucsc.com:10044
db:4  db3-nnnn  OK active 1      enabled     disabled    10777.taxplatform.ucsc.com:10777

ENDPOINTS:
DB:ID       NAME          ID            NODE     ROLE     WATCHDOG_STATUS
db:1       db0-config     endpoint:1     node:2   slave    OK
db:1       db0-config     endpoint:2     node:3   master   OK
db:2       db1-cache      endpoint:3     node:2   master   OK
db:2       db1-cache      endpoint:4     node:3   slave    OK

SHARDS:
DB:ID NAME       ID      NODE   ROLE   SLOTS  USED_MEMORY BACKUP_PROGRESS RAM_FRAG
WATCHDOG_STATUS STATUS
db:1  db0-config ucsc:1 node:1 master 1-4096 8.1MB       N/A           -3711.1KB  OK            OK
db:1  db0-config ucsc:2 node:3 slave  1-4096 2.62MB      N/A           1012.55KB  OK            OK
db:2  db1-cache  ucsc:3 node:3 slave  1-4096 2.46MB      N/A           885.71KB   OK            OK
db:2  db1-cache  ucsc:4 node:1 master 1-4096 5.06MB      N/A           -1596.73KB OK            OK
----------------------------------------------------------
```