Genesys test task – Application as QA Engineer – Francisco Camacho

ТЕСТ:

1. Реализовать на языке Python функциональность записной книжки с возможностью хранить в ней:

- ФИО
- номер телефона
- дату рождения

Плюсом будет реализация в записной книжке функции Reminder о днях рождения при запуске.

I - Code and explanation:

The code that was the main function in the previous version, is now the function that produces the text that is going to be the content of the message that will be automatically sent when the computer (laptop or not) is started. How is this accomplished is explained in part „II“ of this document.

I wanted to have the password in the file that contains the information needed to send the email encrypted, but I have encountered problems in Debian (I think it could be because the Python version I have in Windows is newer that the one in Debian -I have seen also differences regarding exceptions-. Anyway, this is only an exercise, so I hope you don't consider this a big flaw. I just wanted to investigate different things while learning Python).
I have kept both lines in the code, but I have commented the one that does not work in Linux, in order to send one common version that runs exaclty the same in both systems.

```
#!/usr/bin/python3

#
# Genesys test exercise - March 2015 - Francisco Camacho
# Version: 2.0
#    Added exceptions, and sending of email
#

import smtplib
import os
import sys
import time
import base64

#
# constants
#
files_path_linux="/home/fran/genesys/"
files_path_windows="D:\genesys\"

#
# auxiliary functions
#
def path_separator():
# not needed now
   if os.name=="nt":
       return "\\"
   else:
       return "/"
```

```
def path_of_files():
    if os.name=="nt":
        return files_path_windows
    else:
        return files_path_linux

def current_date():
    return time.strftime("%d/%m/%Y")

def compare_simple_dates(date1,date2):
    try:
        first_date=date1.split("/")
        d1=first_date[0]
        m1=first_date[1]
        second_date=date2.split("/")
        d2=second_date[0]
        m2=second_date[1]
    except IndexError:
        sys.stdout.write("Error in date!\n")
        return False
    return d1==d2 and m1==m2

def calculate_age(date):
    info = date.split("/")
    current_year=time.strftime("%Y")
    age = int(current_year) - int(info[2])
    # sys.stdout.write("calculate_age - age: "+str(age)+"\n")
    return str(age)

def read_login_and_pass():
    diccio_login = {}

    try:
        login_file = open(path_of_files()+"login.txt","r")
    except (OSError, IOError):
        sys.stdout.write("Login.txt not found!\n")
        # exit()
        return None

    lines = login_file.read().split("\n")

    if  os.name=="nt":
        if len(lines)!=6:
            sys.stdout.write("Login file not correct!\n")
            # exit()
            return None
    else:
        if len(lines)!=7:
            sys.stdout.write("Login file not correct!\n")
            # exit()
            return None

    diccio_login["server"]=lines[0]
    diccio_login["port"]=lines[1]
    diccio_login["login"]=lines[2]
    diccio_login["password"]=lines[3]   # "encrypted" in the file (simply encoded ...)
    diccio_login["from"]=lines[4]
    diccio_login["to"]=lines[5]         # at the moment only one receiver (yourself!)

    return diccio_login


def send_email(from_addr, to_addr_list, cc_addr_list,subject, message,
              login, password, smtpserver='smtp.gmail.com:587'):

    header  = 'From: %s\n' % from_addr
    header += 'To: %s\n' % to_addr_list
    header += 'Cc: %s\n' % cc_addr_list
    header += 'Subject: %s\n\n' % subject
    message = header + message

    server = smtplib.SMTP(smtpserver)
    server.starttls()
    server.login(login,password)
    problems = server.sendmail(from_addr, to_addr_list, message)
    server.quit()
    return problems


#
# previous reminder.py
#
def reminder_message():
    """ function that generates the content of the email """

    msg = "Today is "+current_date()+"\n\n"

    try:
        people_file = open(path_of_files()+"people.txt","r")
    except (OSError, IOError):
        sys.stdout.write("People.txt not found!")
        exit()

    lines = people_file.read().split("\n")
    cumpleanyero = []  # (spanish word for a person who has birthday :)
    cumpleanyero_phone = []
    cumpleanyero_age = []

    for line in lines:
        if not line:
            continue
        data=line.split(",")
        try:
            if compare_simple_dates(data[2],current_date()):
                cumpleanyero.append(data[0])
                cumpleanyero_phone.append(data[1])
                cumpleanyero_age.append(calculate_age(data[2]))
        except IndexError:
            sys.stdout.write("Error in user data!\n")

    people_file.close()

    if (len(cumpleanyero)==0):
        msg += " Today is nobody's birthday! No cake, sorry! \n\n"
    else:
        msg += "Today is the birthday of: \n\n"
        for i, friend in enumerate(cumpleanyero):
            msg += friend+"!("+cumpleanyero_age[i]+") - Phone number: "+cumpleanyero_phone[i]+"\n"
        msg += "\nCall your friend/s! They will be very happy! \n\n"

    return msg


def main():

    info_email=read_login_and_pass()

    if not info_email:
        sys.stdout.write("No info! Not possible to send reminder email!\n")
    else:
        subject = "Birthday reminder (v2)"
        msg = reminder_message()   # here the call to the reminder() function
        problems = send_email(info_email["from"],info_email["to"],"",
              subject,msg,info_email["login"],
              info_email["password"],info_email["server"]+":"+info_email["port"])
        # passw = str(base64.b64decode(info_email["password"]))[2:-1]
        # problems = send_email(info_email["from"],info_email["to"],"",
        #         subject,msg,info_email["login"],passw,
        #         info_email["server"]+":"+info_email["port"])
        if not problems:
            sys.stdout.write("Birthday reminder mail was sent with no problem :)\n")
        else:
            sys.stdout.write("There were problems sending the birthday reminder email!\n")


main()
```

Content of the files with the information that the script reminder2exe.py needs:

login.txt:

```
smtp.gmail.com
587
fcamadi
<<my password>>
fcamadi@gmail.com
fcamadi@gmail.com
```

people.txt:

```
Vlad,12/12/1976

Katja,12345679,
Tester,2222222,12/11/1950
Ulyses,987654,15/03/1982

Nina,8181900900,15/03/1980
Lola,9811896666,14/01/1984
```

(I have left some incomplete lines and others empty on purpose, of course)

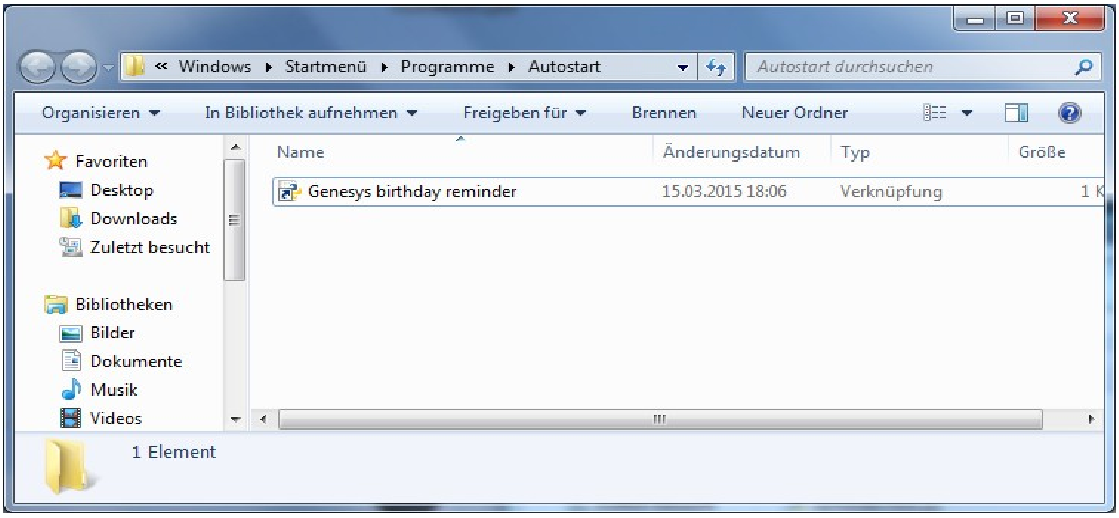
II - To make the script be automatically launched when the computer starts:

1) Added #!/usr/bin/python3 in the first line of the Python script

2)

Windows 7:

Add a link to the reminder2exe.py scrypt in the „Austostart“ folder



Name	Änderungsdatum	Typ	Größe
Genesys birthday reminder	15.03.2015 18:06	Verknüpfung	1 K

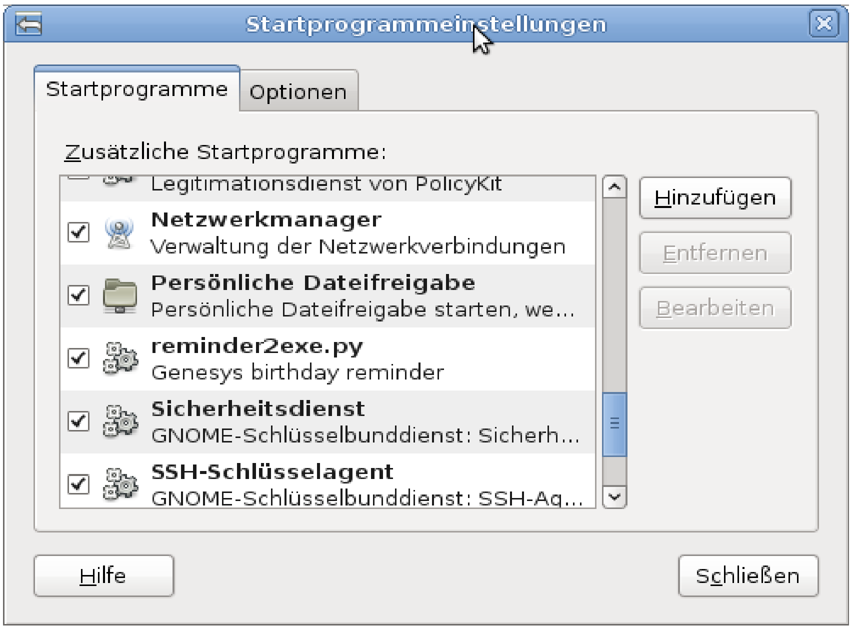
Linux (Debian 6&7):

Two ways:

a) like in Windows:

System -> preferences -> start programmes

and add the file that we want to be launched when the computer starts:



Legitimationsdienst von PolicyKit

☑ **Netzwerkmanager**
Verwaltung der Netzwerkverbindungen

☑ **Persönliche Dateifreigabe**
Persönliche Dateifreigabe starten, we...

☑ **reminder2exe.py**
Genesys birthday reminder

☑ **Sicherheitsdienst**
GNOME-Schlüsselbunddienst: Sicherh...

☑ **SSH-Schlüsselagent**
GNOME-Schlüsselbunddienst: SSH-Ag...

or

b) in the .profile of each user that needs this script, add the line:
i.e:

```
        /home/fran/genesys/reminder2exe.py
```

Note:

I have investigated also how to launch the reminder2exe.py as an SysV init script, but I don't think that this kind of script should be treated the same way as a tomcat service, or any of the different daemons that are launched when Debian starts. It was interesting anyway learning a little bit more about Linux than I knew before.
(It didn't work: it was launched to early I think).

Hagen (Germany), 15.03.2015