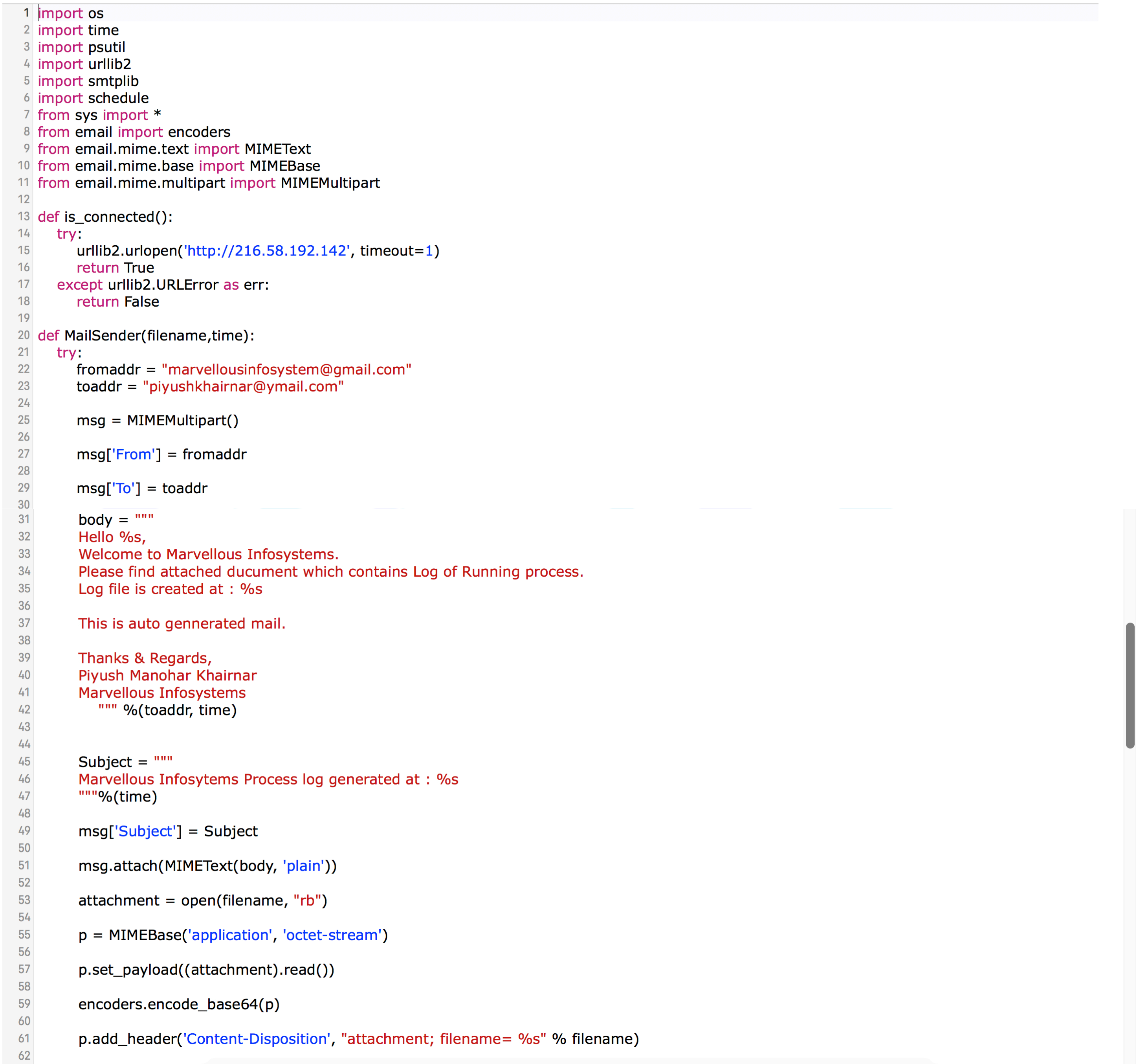
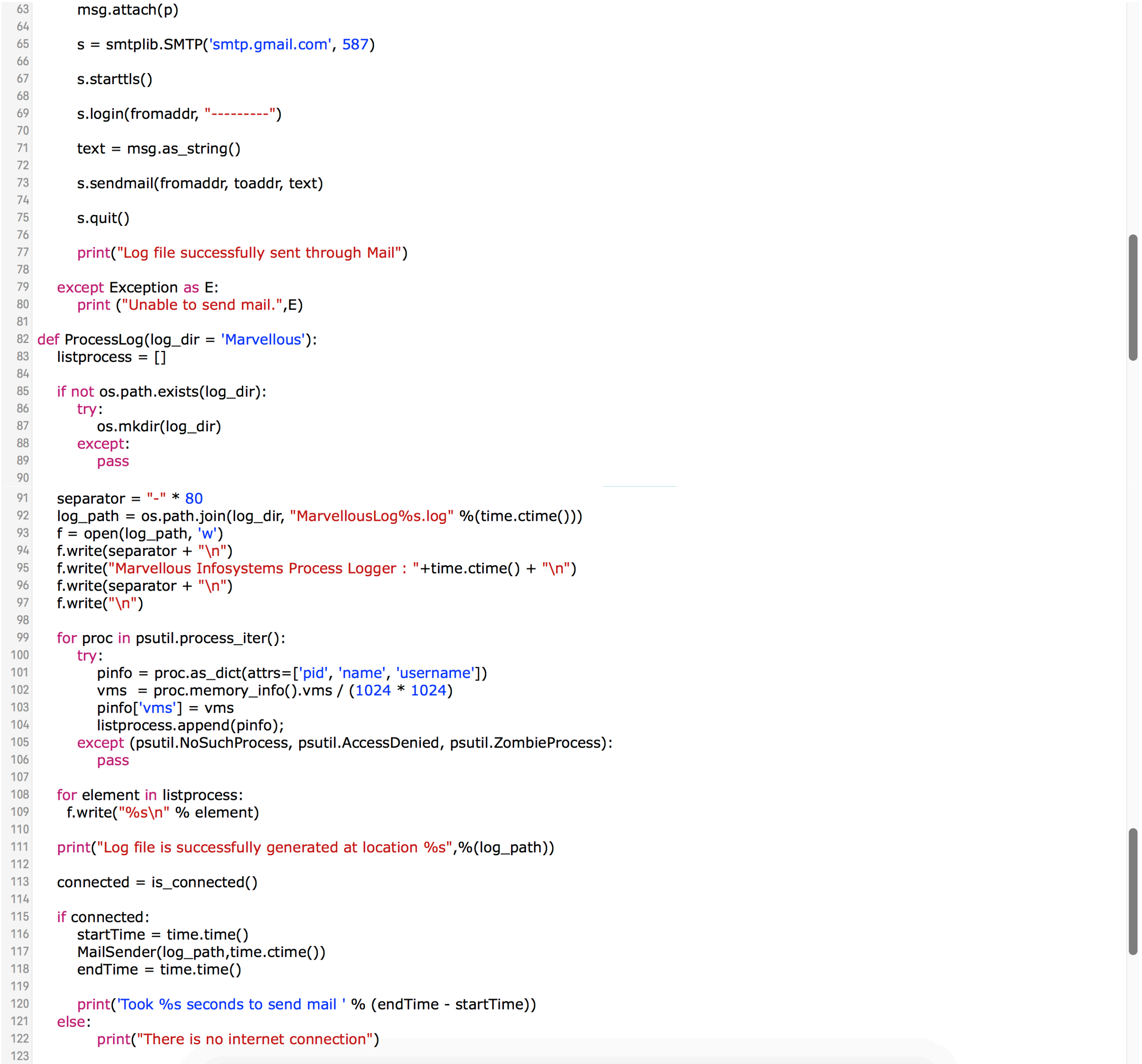
Process Monitor

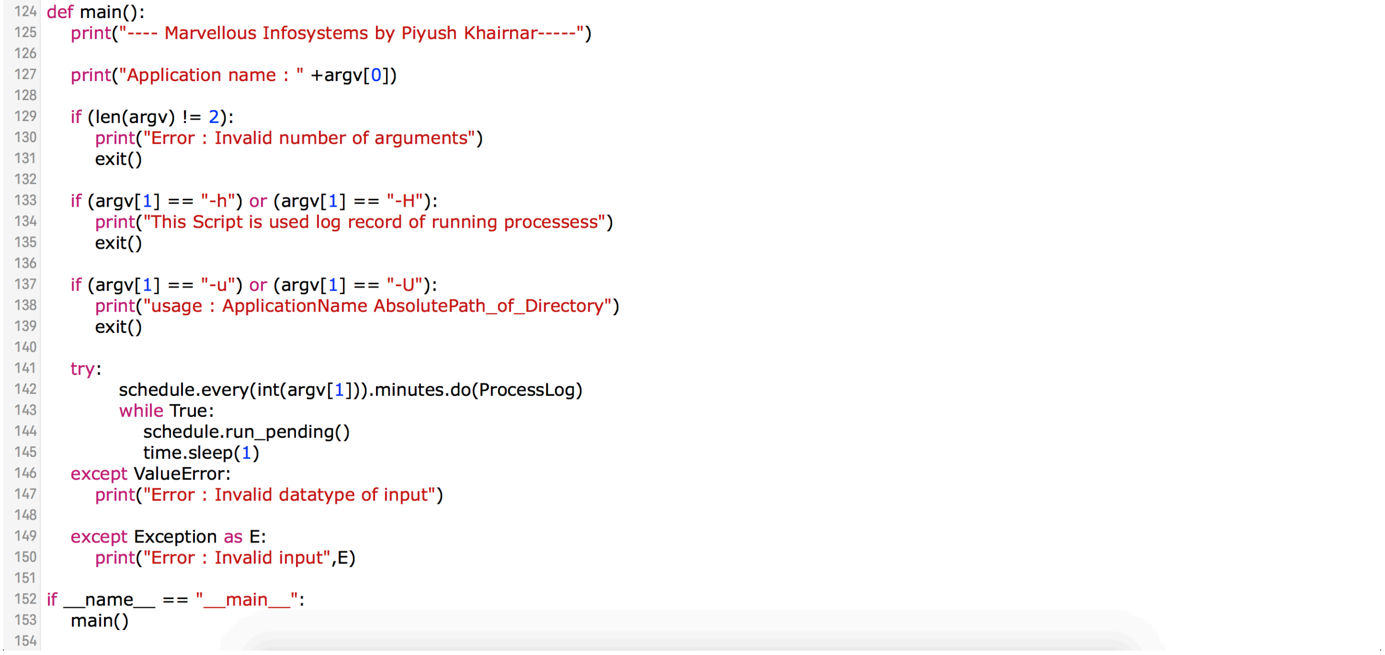
Automation script which accepts time interval from user and create log file in that Marvellous directory which contains information of all running processes.
After creating the log file send that log file through mail.



```
import os
import time
import psutil
import urllib2
import smtplib
import schedule
from sys import *
from email import encoders
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email.mime.multipart import MIMEMultipart

def is_connected():
    try:
        urllib2.urlopen('http://216.58.192.142', timeout=1)
        return True
    except urllib2.URLError as err:
        return False

def MailSender(filename,time):
    try:
        fromaddr = "marvellousinfosystem@gmail.com"
        toaddr = "piyushkhairnar@ymail.com"

        msg = MIMEMultipart()

        msg['From'] = fromaddr

        msg['To'] = toaddr

        body = """
        Hello %s,
        Welcome to Marvellous Infosystems.
        Please find attached ducument which contains Log of Running process.
        Log file is created at : %s

        This is auto gennerated mail.

        Thanks & Regards,
        Piyush Manohar Khairnar
        Marvellous Infosystems
            """ %(toaddr, time)


        Subject = """
        Marvellous Infosytems Process log generated at : %s
        """%(time)

        msg['Subject'] = Subject

        msg.attach(MIMEText(body, 'plain'))

        attachment = open(filename, "rb")

        p = MIMEBase('application', 'octet-stream')

        p.set_payload((attachment).read())

        encoders.encode_base64(p)

        p.add_header('Content-Disposition', "attachment; filename= %s" % filename)

```



```
        msg.attach(p)

        s = smtplib.SMTP('smtp.gmail.com', 587)

        s.starttls()

        s.login(fromaddr, "---------")

        text = msg.as_string()

        s.sendmail(fromaddr, toaddr, text)

        s.quit()

        print("Log file successfully sent through Mail")

    except Exception as E:
        print ("Unable to send mail.",E)

def ProcessLog(log_dir = 'Marvellous'):
    listprocess = []

    if not os.path.exists(log_dir):
        try:
            os.mkdir(log_dir)
        except:
            pass

    separator = "-" * 80
    log_path = os.path.join(log_dir, "MarvellousLog%s.log" %(time.ctime()))
    f = open(log_path, 'w')
    f.write(separator + "\n")
    f.write("Marvellous Infosystems Process Logger : "+time.ctime() + "\n")
    f.write(separator + "\n")
    f.write("\n")

    for proc in psutil.process_iter():
        try:
            pinfo = proc.as_dict(attrs=['pid', 'name', 'username'])
            vms  = proc.memory_info().vms / (1024 * 1024)
            pinfo['vms'] = vms
            listprocess.append(pinfo);
        except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess):
            pass

    for element in listprocess:
      f.write("%s\n" % element)

    print("Log file is successfully generated at location %s",%(log_path))

    connected = is_connected()

    if connected:
        startTime = time.time()
        MailSender(log_path,time.ctime())
        endTime = time.time()

        print('Took %s seconds to send mail ' % (endTime - startTime))
    else:
            print("There is no internet connection")

```



```
def main():
    print("---- Marvellous Infosystems by Piyush Khairnar-----")

    print("Application name : " +argv[0])

    if (len(argv) != 2):
        print("Error : Invalid number of arguments")
        exit()

    if (argv[1] == "-h") or (argv[1] == "-H"):
        print("This Script is used log record of running processess")
        exit()

    if (argv[1] == "-u") or (argv[1] == "-U"):
        print("usage : ApplicationName AbsolutePath_of_Directory")
        exit()

    try:
            schedule.every(int(argv[1])).minutes.do(ProcessLog)
            while True:
                schedule.run_pending()
                time.sleep(1)
    except ValueError:
        print("Error : Invalid datatype of input")

    except Exception as E:
        print("Error : Invalid input",E)

if __name__ == "__main__":
    main()

```