**-----------------------------------------SYSTEM Module----------------------------------------------**

import os

**Cpu information**

def cpuinfo():

info='cat /proc/cpuinfo'

os.system(info)

cpuinfo()

**Cpu Architecture information**

def cpuArchiInfo():

info='lscpu'

os.system(info)

cpuArchiInfo()

**CPU Family**

def CPUFamily():

info='lscpu | grep "CPU family"'

os.system(info)

CPUFamily()

**Sockets**

def socket():

info='ss -s'

os.system(info)

socket()

**-----------------------------------------processing Module----------------------------------------------**

**no.of processing unit**

def nproc():

info='nproc'

os.system(info)

nproc()

**Processor**

def processor():

info='cat /proc/cpuinfo | grep "model name"'

os.system(info)

processor()

**Current process id**

def currentPId():

print("The current process id: ",os.getpid())

currentPId()

**Parent process id**

def PPId():

print("The parent process id: ",os.getppid())

PPId()

**Vendor id**

def vendorID():

info='lscpu | grep "Vendor ID"'

os.system(info)

vendorID()

**-----------------------------------------memory Module----------------------------------------------**

**Cache Information**

def cacheinfo():

info='lscpu | grep "cache"'

os.system(info)

cacheinfo()

**Cache Size**

def cacheSize():

info='cat /proc/cpuinfo | grep "cache size"'

os.system(info)

cacheSize()

**Ram info in kb**

def RamInfoInKB():

info='free '

os.system(info)

RamInfoInKB()

**Ram info in mb**

def RamInfoInMB():

info='free -m'

os.system(info)

RamInfoInMB()

**Ram info in gb**

def RamInfoInGB():

info='free -g'

os.system(info)

RamInfoInGB()

**-----------------------------------------memory Module----------------------------------------------**

**Date**

def date():

info='date'

os.system(info)

date()

**Calender**

def calender():

info='cal'

os.system(info)

calender()

**uptime**

def uptime():

info='uptime'

os.system(info)

uptime()

**-------------------------------------------------File Module------------------------------------------------**

**Create File:**

def createFile():

f=open("putting.txt","w+")

print("Created File Successfully!")

createFile()

**Write in File**

def writeInFile():

f=open("putting.txt","w+")

for I in range(10):

f.write("This is line %d\r\n"%(i+1))

print("Write in File Successfully!")

f.close()

writeInFile()

**read file**

def readFile():

f=open("putting.txt","r")

if f.mode=="r":

contents=f.read()

print(contents)

readFile()

**Create zip file**

from os import path

from zipfile import ZipFile

def zipFile():

if path.exists("output2.txt"):

src=path.realpath("output2.txt")

with ZipFile("output2.zip","w") as newzip:

print("Create zip file successfully")

zipFile()

**unzipfile**

from os import path

from zipfile import ZipFile

def unzipFile():

with ZipFile("output2.zip","r") as newzip:

newzip.extractall()

print("unzip file successfully")

unzipFile()

**Delete file**

import os

try:

os.remove('test.txt')

print("File delete successfully")

except IOError:

print("File not delete")

**File Renamed**

import os

fd="file1.txt"

os.rename(fd,'new.txt')

print("File Successfully Renamed")

**-----------------------------------------------------------Network Module--------------------------------------------**

**Find the ip address and Name**

def nslookup():

info='nslookup google.com'

os.system(info)

nslookup()

**Route trace of packets to take the host**

def traceroute():

info='traceroute facebook.com'

os.system(info)

traceroute()

**To find the ip and subnet mask**

def ifconfig():

info='ifconfig'

os.system(info)

ifconfig()

**Tell about the various network connection,routing table and interface statistics**

def netstat():

info='netstat'

os.system(info)

netstat()

**List all tcp ports**

def netstat():

info='netstat -at'

os.system(info)

netstat()

**List all udp ports**

def netstat():

info='netstat -au'

os.system(info)

netstat()

**information about user login**

def w():

info='w'

os.system(info)

w()