项目一：达内教学环境自动化管理项目
1、自动收集所有主机的 MAC 地址
2、学生机器远程开机管理
3、能够向所有的主机发送文件
4、在所有学生机上并行执行相同命令，如晚上下课关机。

get_mac.py:收集所有主机 MAC 地址

```
#!/usr/bin/env python

import subprocess
import threading
import os
import sys

def get_mac(ip):
    os.system('ping -c2 %s &> /dev/null' % ip)
    result = subprocess.Popen('arp -n %s | tail -1' % ip, shell=True, stdout=subprocess.PIPE)
    arp = result.stdout.readline()
    if ':' in arp:
        ip = arp.split()[0]
        mac = arp.split()[2]
        print ip, mac

def main():
    if len(sys.argv) != 2:
        print "Usage: %s subnet" % sys.argv[0]
        sys.exit(1)
    subnet = sys.argv[1].split('.')[:-1]
    ips = ('%s.%s' % ('.'.join(subnet), i) for i in range(1, 255))
    for line in ips:
        ip = line.strip()
        t = threading.Thread(target=get_mac, args=(ip,))
        t.start()

if __name__ == '__main__':
    sys.exit(main())
```

wake_pc.py:网络唤醒所有主机

```
#!/usr/bin/env python

import threading
import os
import sys
```

```
def wake_pc(iface, mac):
    os.system('ether-wake -i %s %s' % (iface, mac))

def main():
    if len(sys.argv) != 3:
        print "Usage: %s iface macfile"
        sys.exit(1)
    iface = sys.argv[1]
    mac_file = sys.argv[2]
    with open(mac_file) as fobj:
        for line in fobj:
            mac = line.strip()
            t = threading.Thread(target=wake_pc, args=(iface, mac))
            t.start()

if __name__ == '__main__':
    sys.exit(main())
```

mput.py: 批量向远程主机传输文件

```
#!/usr/bin/env python

import sys
import threading
import paramiko
import os

def mput(host, password, src, dst):
    base_name = os.path.basename(src)
    dst_name = os.path.join(dst, base_name)
    scp = paramiko.Transport(host)
    scp.connect(username='root', password=password)
    sftp = paramiko.SFTPClient.from_transport(scp)
    sftp.put(src, dst_name)
    sftp.close()
    scp.close()

if __name__ == '__main__':
    if len(sys.argv) != 5:
        print "Usage: %s ipfile password localfile remotefile" % sys.argv[0]
        sys.exit(1)
    ipfile = sys.argv[1]
    if not os.path.isfile(ipfile):
```

```
            print "No such file: %s" % ipfile
            sys.exit(2)
        password = sys.argv[2]
        lfile = sys.argv[3]
        rfile = sys.argv[4]
        with open(ipfile) as fobj:
            for line in fobj:
                ip = line.strip()
                t = threading.Thread(target=mput, args=(ip, password, lfile, rfile))
                t.start()
```

remote_comm.py:批量在远程主机执行命令

```
#!/usr/bin/env python

import paramiko
import os
import sys
import threading

def remote_comm(host, password, comm):
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh.connect(host, username='root', password=password)
    stdin, stdout, stderr = ssh.exec_command(comm)
    out = stdout.read()
    err = stderr.read()
    if out:
        print "[%s:out]: %s" % (host, out),
    if err:
        print "%s:Error: %s", (host, err),
    ssh.close()

if __name__ == '__main__':
    if len(sys.argv) != 4:
        print "Usage: %s ipfile password 'comm'" % sys.argv[0]
        sys.exit(1)
    ipfile = sys.argv[1]
    if not os.path.isfile(ipfile):
        print "No such file: %s" % ipfile
        sys.exit(2)
    password = sys.argv[2]
    comm = sys.argv[3]
    with open(ipfile) as fobj:
        for line in fobj:
```

```
            ip = line.strip()
            t = threading.Thread(target=remote_comm, args=(ip, password,
comm))
            t.start()
```