

# 课程大作业三

叶增渝 519030910168

P.S.本来是想要提交相关源文件的，但是发现提交文件类型仅为 pdf，故将相关源码贴在文末

## 1.配置环境

(1) 首先我们需要下载 libvirt 环境与对应的 Python API

首先键入 `sudo apt-get install libvirt-dev libvirt-daemon libvirt-clients` 命令安装基本 `virsh` 命令。安装完成后我们键入 `virsh` 便可进入 `virsh` 命令行

```
spoilvoid@ubuntu:~$ virsh
Welcome to virsh, the virtualization interactive terminal.

Type: 'help' for help with commands
      'quit' to quit

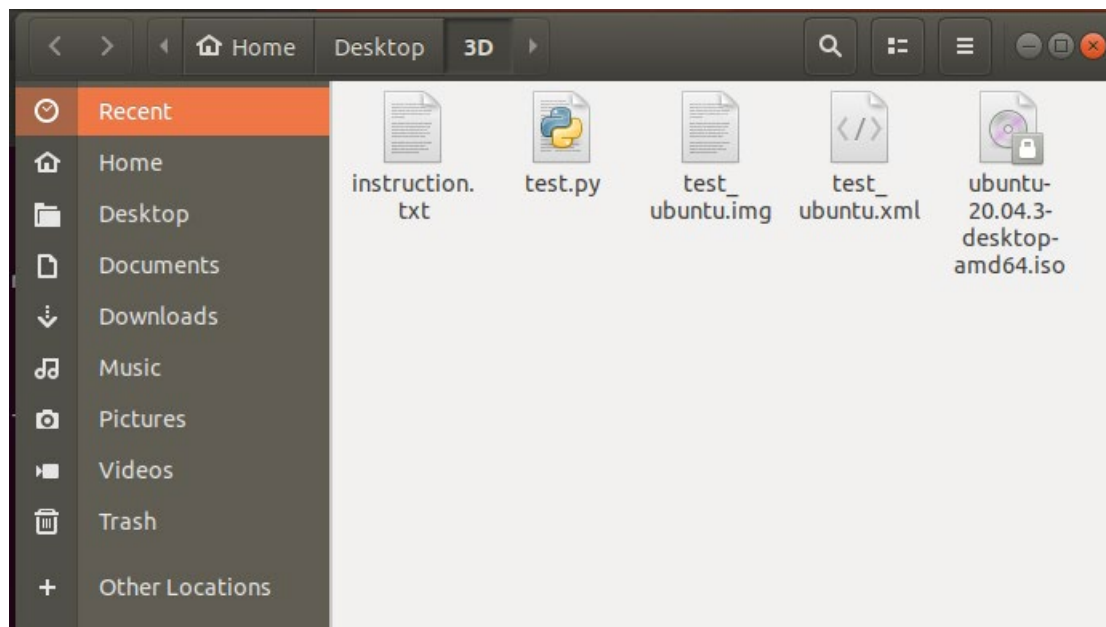
virsh #
```

再使用 `sudo apt-get install python-libvirt` 命令安装 Python API，以便后续 Python 脚本的运行

(2) 配置虚拟机对应 xml 文件（源码见文末）

对于之前使用 `qemu` 创建的虚拟机，没有相关的 xml 文件，也就无法被 `virsh` 相关命令发现，所以第一步是在 `qemu` 虚拟机文件夹下创建 xml 配置文件并进行修改(由于之前已经安装系统了，所以 `<boot dev='hd'/>` 需要设置为 `hd` 防止进入时首先进入安装界面)

Tips:我在这里选择使用 `Vncviewer` 连接重新安装了 `qemu` 虚拟机，所以与作业 2 显示的 `img` 文件名不相同



(3) 利用 `virsh` 命令行工具将其纳入 `libvirt` 管理下

首先在 `virsh` 下使用 `define test_ubuntu.xml` 命令

```
virsh # define test_ubuntu.xml
Domain test_ubuntu defined from test_ubuntu.xml
```

然后我们便可以在 list --all 命令下查看当前管理下的虚拟机

```
virsh # list --all
Id      Name                                State
-----
-       test_ubuntu                        shut off
```

此时 test\_ubuntu 仍然处于关闭状态，我们使用 start test\_ubuntu 命令开启虚拟机

```
virsh # start test_ubuntu
Domain test_ubuntu started
```

(在这之后此时我还使用了 vncviewer 使用 GUI 界面打开了虚拟机，但没有后续操作，我认为这是不必要的，故不加入步骤中)

```
virsh # list
Id      Name                                State
-----
3       test_ubuntu                        running
```

此时我们仅仅通过 list 命令就可以看到该虚拟机的 name 与 id 了

2.编写 test.py 脚本文件获取需要数据（源码见文末）

我们使用 python libvirt-test.py 2>/dev/null 命令执行 Python 脚本

这里的实现原理:通过自己设定好的虚拟机名字使用 conn.lookupByName(name)进行查找，查询结果如下，观察结果与在 xml 中文件中设置的内容，不难发现我们成功获取了虚拟机的 ID 为 3、name 为"test\_ubuntu"、最大内存大小为 1024MB 与 vCPU 个数为 2，任务完成。

```
spoilvoid@ubuntu:~/Desktop/3D$ python test.py 3>/dev/null
---Get domain info via libvirt python API---
-----Connection is created successfully-----

----- get domain info by name -----
('Dom id: ', 3)
('name: ', 'test_ubuntu')
('memory: ', 1024, 'MB')
('vCPUs: ', 2)
Connection is closed
```

test\_ubuntu.xml 文件源码:

```
<domain type='kvm'>
  <name>test_ubuntu</name> //虚拟机名称
  <memory>1048576</memory> //最大内存, 单位 k
  <currentMemory>1048576</currentMemory> //可用内存, 单位 k
  <vcpu>2</vcpu> //虚拟 cpu 个数
  <os>
    <type arch='x86_64' machine='pc'>hvm</type>
    <boot dev='hd'/> //光盘启动
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='localtime'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
    <emulator>/usr/bin/kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2'/>
      <source file='/home/spoilvoid/Desktop/3D/test_ubuntu.img'/> //目的镜像路径
      <target dev='hda' bus='ide'/>
    </disk>
    <disk type='file' device='cdrom'>
      <source file='/home/spoilvoid/Desktop/3D/ubuntu-20.04.3-desktop-
amd64.iso'/> //光盘镜像路径
      <target dev='hdb' bus='ide'/>
    </disk>
    <interface type='bridge'> //虚拟机网络连接方式
      <source bridge='virbr0'/> //当前主机网桥的名称
      <mac address='00:16:3e:5d:aa:a8'/> //为虚拟机分配 mac 地址, 务必唯一, 否
则 dhcp 获得同样 ip,引起冲突
    </interface>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='19' autoport='yes' listen = '0.0.0.0' keymap='en-
us'/> //vnc 方式登录, 端口号自动分配, 自动加 1, 可以通过 virsh vncdisplay 来查询
  </devices>
</domain>
```

test.py 文件源码:

```

#!/usr/bin/python
import libvirt
import sys

def createConnection():
    conn = libvirt.openReadOnly(None)
    if conn == None:
        print('Failed to open connection to QEMU/KVM')
        sys.exit(1)
    else:
        print('-----Connection is created successfully-----')
        return conn

def closeConnection(conn):
    print("")
    try:
        conn.close()
    except:
        print('Failed to close the connection')
        return 1
    print('Connection is closed')

def getDomInfoByName(conn, name):
    print("")
    print('----- get domain info by name -----')
    try:
        myDom = conn.lookupByName(name)
    except:
        print('Failed to find the domain with name %s',name)
        return 1
    print("Dom id: ", myDom.ID())
    print("name: ", myDom.name())
    print("memory:  ",int(myDom.maxMemory()/1024), "MB")
    print("vCPUs: ", myDom.maxVcpus())

if __name__ == '__main__':
    name1 = "test_ubuntu"
    print("---Get domain info via libvirt python API---")
    conn = createConnection()
    getDomInfoByName(conn, name1)
    closeConnection(conn)

```