

# 第一课：虚拟机的安装与数字符号的操作

计算机导论实验：图片隐写 作业在最下方

实验目的： 在一张图片中隐写入文字信息（其实也可以是图像，音频等），并且可以通过程序读出图片中的信息。

并且在实验过程中体会

1. 计算机科学是研究计算过程的科学
2. 计算过程是通过操纵数字符号变换信息的过程。
3. 理解计算思维的要点：无歧义的，足够精确的描述信息变换的操作序列。
4. 通过动手操作更进一步的理解，逻辑思维，算法思维和系统思维

实验原理：


一张图片在本质上是一串二进制代码，我们要做的就是将图片转化为二进制代码（事实上我们在接下来的实验中做了简化，只将图片转化成我们可以读懂的数字符号）

然后改变某一些二进制代码（比如 00011 11100，我们可以将 11100 映射为 'a'，并且记录下位置，这样在这一位读到 '11100' 就写下 'a'），然后将这些二进制代

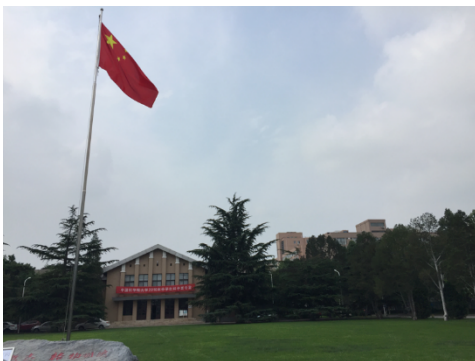
码重新转化成图片输出。这样我们就完成了隐写。

实验效果：

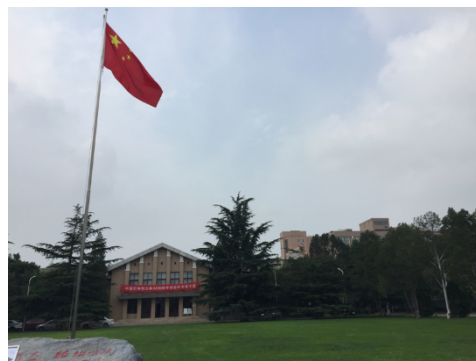
```
yu@yu-VirtualBox:~/Desktop/ComSci$ python3 example.py test.bmp hide myword.txt testnew.bmp
yu@yu-VirtualBox:~/Desktop/ComSci$ python3 example.py testnew.bmp show
the information I want to hide
```



无信息隐藏：



有信息隐藏：



当然这是我们这一个

学期将要完成的作业。在接下来的日子里，每两周都会有一个小作业，用时不会超过 45 分钟，教大家必要的编程基础从而有足够的能力来完成这一项有趣作业。

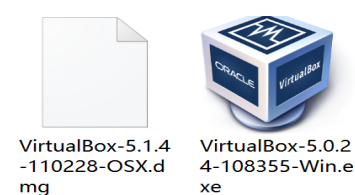
（再也不用怕妈妈查我的聊天记录啦）。有兴趣的同学可以继续学习，比如在图片里隐藏一个 bug，发给你最好的朋友。哈哈！！祝你们好运！！

说明：我们将使用 python 来解决这一有趣的任务。选择 python 的原因是它简洁，简单，而且在同学们以后的科研生活中基本上会用到 python。

今天我们只需要做一点点工作就行了：

任务一：安装虚拟机

我们使用 Oracle VM VirtualBox 这一款虚拟机。这一个安装文件会在课件中提供给大家。Windows 系统的同学使用.exe 结尾的安装；os 系统的同学使用.dmg 结尾的安装。



Windows 系统安装示例：苹果安装与 Windows 安装方式一致就不做示范了

1.双击安装:只需要一路 yes install 就可以啦。

最后出现如下画面：



任务二：

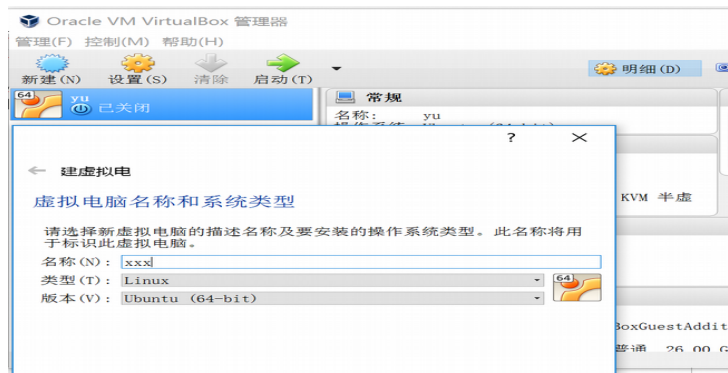
安装 Linux ubuntu 16.04 操作系统

原料：电脑，virtualbox，镜像文件

镜像文件将会发布在网上，届时同学们自行下载，是以 ios 结尾的文件。

1.打开 virtualbox 点击新建按钮

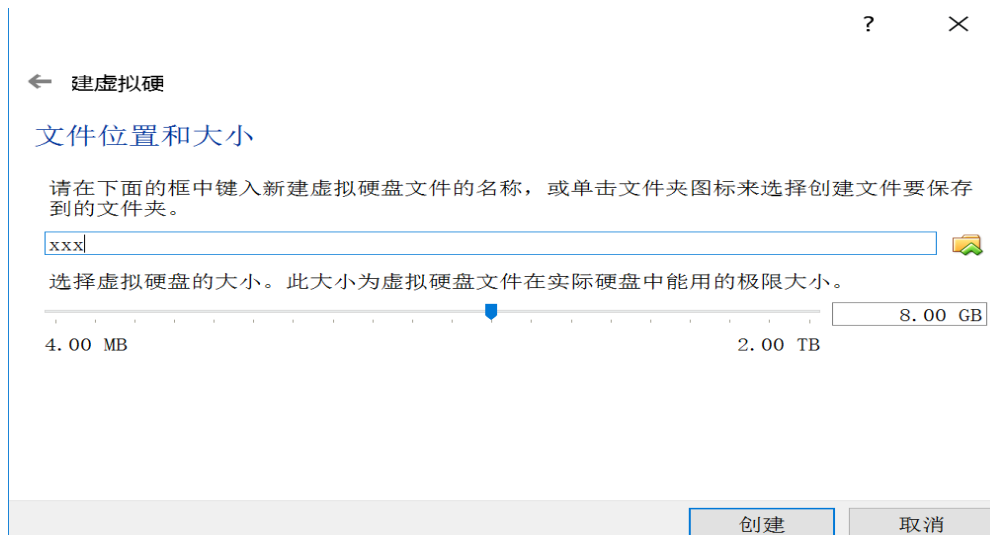
2.按照图示，选择正确的操作系统。（名称可以任意）



PS:学校发的电脑可能没有 64-bit 这一选项，需要重启电脑，按 Enter 键进入，然后按 f1 进入 bios，用上下左右移动选择 Security，向下选择 Virtualization 按 enter，将其改为 Enabled，按 F10 保存退出，启动即可。

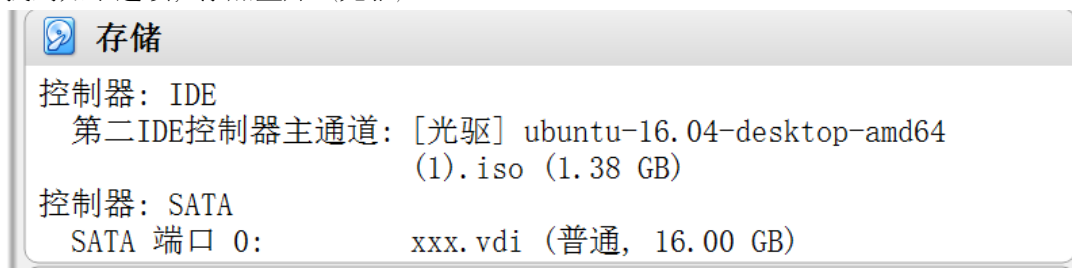
内存建议给 2G，不要低于 1.5G

3.然后一路确定到如下界面：



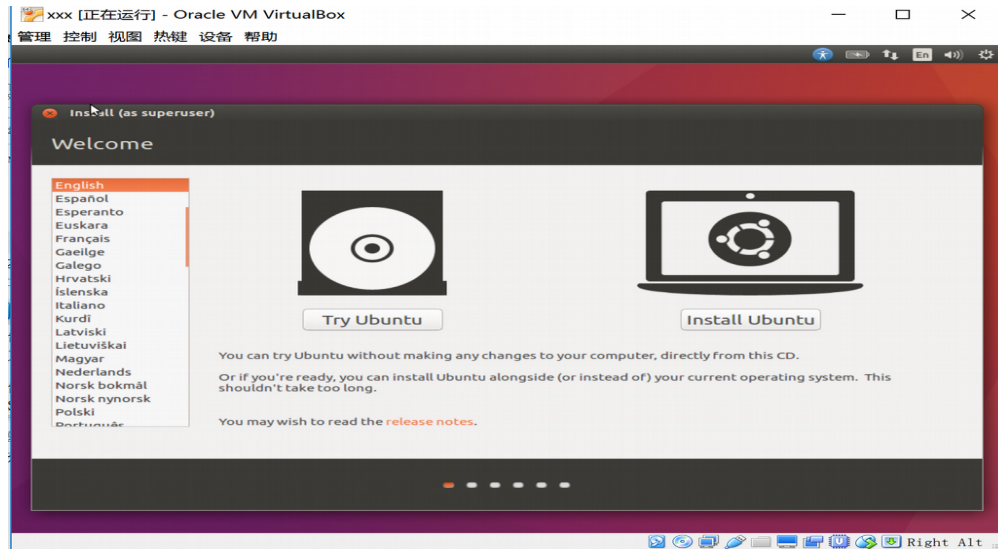
建议给 20G 虚拟硬盘

4.找到如下选项，添加盘片（光驱）



5.启动虚拟机：

安装系统：



点击: **Install Ubuntu**。如果鼠标移不出来, 尝试按右边的 **ctrl** (热键里默认的) 一路 **continue**, 然后等待安装, 语言最好使用英语。输入姓名密码继续 **continue**。安装完之后重启。它会显示没有盘片。你不要理会, 强制退出然后重启就安装好了 (可以多次重启)。

3. 点击设备, 选择安装增强效果, 按提示去做就行了。然后重新点击设备 > 安装增强效果 > 强制释放。重启电脑, 虚拟机桌面就可以随着窗口一起变化了。(ps: 不安装也可, 但操作不方便)

#### 4. 安装 python3。

Ubuntu 系统自带 python3, 我们需要装一个编译器, 以便于我们以后调试程序。

I. 安装完成后: 在 linux 桌面上右键, 选择 **Open Terminal** 打开命令行, 你会看到如下类似的界面, 通过 **cd** 命令可以退出当前目录, **cd filename** 可以进入下一级的 **filename** 的目录。

II 右键, 选择 **open Terminal**, 然后输入 **sudo apt-get install idle3** 回车, 输入密码, 在运行过程中它会问你 **yes or no** 输入 **y** 即可。

```
yu@yu-VirtualBox:~$ sudo apt install idle3
[sudo] password for yu:
```

在 **terminal** 中敲入 **idle3** 会有一个文本弹出, 借此来判断是否安装成功。

III. 运行 python: 输入 **python3**, 然后回车, 如下图。

```
yu@yu-VirtualBox:~/Desktop/computer science$ python
Python 2.7.11+ (default, Apr 17 2016, 14:00:29)
[GCC 5.3.1 20160413] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

**Ctrl+d** 可以退出 **python** 回到命令行 或者 **Ctrl+c**

#### IV. ord 函数

```
yu@yu-VirtualBox: ~  
yu@yu-VirtualBox:~$ python3  
Python 3.5.2 (default, Jul 5 2016, 12:43:10)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> ord('A')  
65  
>>> ord('a')  
97  
>>> ord('A')+ord('a')  
162  
>>>
```

帮助同学们做下一道练习

以下介绍一些 python 中的基本知识。

I以下实例 分别为计算  $(3+5)$   $(3-5)$   $(3 \bmod 5)$  ( $3$  除以  $5$  是浮点类型) ( $3$  整除  $5$  是整数类型的)  $2**3$  (表示  $2$  的  $3$  次方)

```
yu@yu-VirtualBox:~$ python3  
Python 3.5.2 (default, Jul 5 2016, 12:43:10)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 3+5  
8  
>>> 3-5  
-2  
>>> 3%5  
3  
>>> 3/5  
0.6  
>>> 3//5  
0
```

II. 以下实例，是赋值符号

$x=1$   $x=x+1$   $x$   $y=1$   $x=x+y+1$   $x$  (注意理解等号的意义: 如  $x=x+1$  把原来  $x$  的值加上  $1$  之后赋值给  $x$ 。这和我们平时理解的等号很不一样。)

```
>>> x=1  
>>> x=x+1  
>>> x  
2  
>>> y=1  
>>> x=x+y+1  
>>> x  
4
```

III以下是打印 (显示处理结果)

`print(x)` `print('abcdef')`

```
>>> print(x)  
4  
>>> print('abcdef')  
abcdef
```

说明: 似乎 `print(x)` 与 `x` 的效果是一样的, 其实不然。这是交互式界面, `print(x)`是打印, 而 `x` 是告诉你 `x` 是什么。`'abcdef'`是字符串类型。比如`'1' != 1`。( `!=`表示不等于)。  
还有`'1'+ '1' == '11'` 而 `1+1=2` `1+'1'` 是错误的。同学们可以自己尝试一下。

IV. 以下是接收键盘输入

例子: `name=input('please in put your name')`

```
>>> name=input('please iput your name:')
please iput your name:xxx
>>> name
'xxx'
```

V. 我们首先来介绍一下 python 的一些在本次实验中会用到的基本的数据类型。这些不会耗费大家太多时间，其实我们在上一次小作业中已经用到大部分了。

1. 字符串 string: 简单的说是用 ' ' 引起来的东西。字符串可以如下操作:

'gfxg'[0] 是 'g' 请记住它是从 0 开始的;

另外介绍一个函数 `len(string)` 是获取 string 的长度。

```
>>> '123456'[1]
'2'
>>> len('123456')
6
```

2. 整数 (int) 就是整数 没啥好说的。

3. 浮点数 (float)

19. 注意这里有一个点 .098 -2e3 表示  $-2 \times 10^3$

4. 类型转化

`str(object='')` 将整数或者浮点类型转换成为字符串

`int(x)` 将数字字符串或数值转换为整数 (是十进制的)

`float(x)` 将字符串或数值转化为浮点数

```
>>> str(123456)
'123456'
>>> int(12.5)
12
>>> float('12.3')
12.3
```

7.列表 (list)

可以把大量数据放在一起，列表中可以包含任意数据类型，也可以包含另一个列表。像字符串一样，可以用序号访问其中的成员。对于列表有很多操作函数有兴趣的同学可以自己查一查。

```
>>> []
[]
>>> x=[1,2,3]
>>> x[1]
2
>>> x.append(list())
>>> x
[1, 2, 3, []]
```

6.元组(tuple)

元组可以看成是特殊的列表，但是元组一旦建立就不可以更改。是完全不可以更改。但是可以读取。

```
>>> x=(1,2,3)
>>> x[0]
1
```

### 姓名编码编程练习

考虑一个简单的计算过程：通过特定的符号变换产生姓名编码。这个符号变换是：每个同学将自己的姓名的汉语拼音的字符相加，并打印出结果数值。该练习涉及三类符号，即汉语拼音字符（**Xu Zhi Wei**）、整数数值数据（**861**）、数值字符（**8, 6, 1**）。

实例：字符相加。我们可以将“汉语拼音的字符相加”简化等同于这些字符的 **ASCII** 编码值相加，即忽略拼音的声调。例如，“徐志伟”同学的姓名编码是“**861**”，因为“徐志伟”同学的汉语拼音共包含 **10** 个字符（注意不要忽略两个空格），即 **Xu Zhi Wei**。采用 **ASCII** 编码，其相加之和是  $88 + 117 + 32 + 90 + 104 + 105 + 32 + 87 + 101 + 105 = 861$ 。

练习：请用 **Python** 实现自己的姓名编码符号变换。【参见课程实践 7.7】

练习提交：提交一份 **pdf** 文件，其中包含源码截图，姓名，结果以及对数字符号的理解（**100** 字左右）