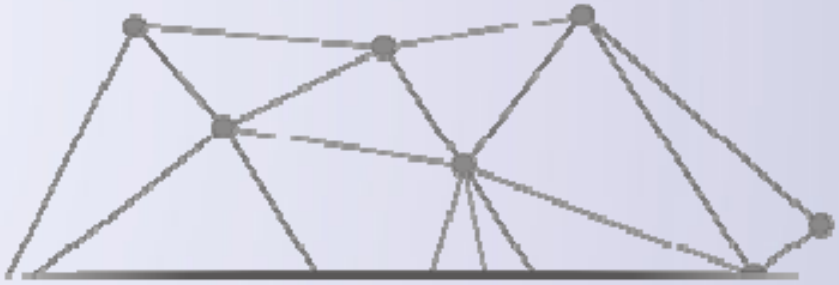
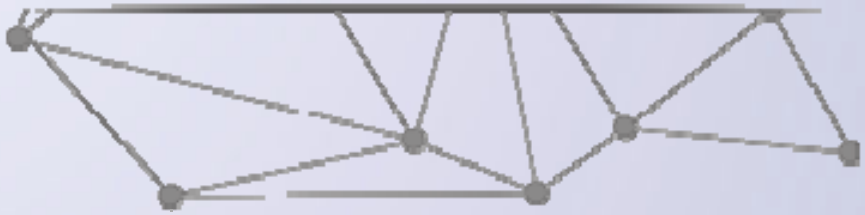


OS平台编程



嵩天

北京理工大学

python 语言程序设计

OS平台编程的需求

- 目录文件的操作

  对系统目录、文件的操作方法

- 程序定时执行

- 可执行程序的转换

  python程序向可执行程序的转换

目录文件的操作　　os库

python安装后自带的函数库，处理操作系统相关功能

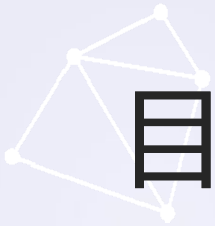
os.getcwd()　　获得当前工作目录

os.listdir(path)　返回指定目录下的所有文件和目录名

os.remove()　　删除一个文件

os.removedirs(path)　删除多个目录

os.chdir(path)　更改当前目录到指定目录



目录文件的操作　　os库

os.mkdir(path)　新建一个目录

os.rmdir(name)　删除一个目录

os.rename(src, dst)　更改文件名

python 语言程序设计

目录文件的操作　　os库

os.path 处理操作系统目录的一个子库

Os.path.isfile() 检验路径是否是一个文件

Os.path.isdir()

Os.path.exists() 判断路径是否存在

Os.path.split()　返回一个路径的目录名和文件名

os.path.splitext()　分离扩展名

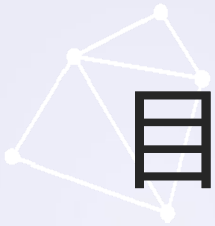
目录文件的操作　　os库

Os.path.dirname 获得路径名

Os.path.basename() 获得文件名

Os.path.getsize() 获得文件大小

Os.path.join(path, name) 返回绝对路径



目录文件的操作　　os库

os.walk(path)用于遍历一个目录，返回一个三元组

root, dirs, files = os.walk(path)

其中，root是字符串，dirs和files是列表类型，表示root中的所有目录和所有文件

目录文件的操作　　os库

例子：打印某一个目录下的全部文件

```
import os

path=input("输入一个路径:")
for root, dirs, files in os.walk(path):
    for name in files:
        print(os.path.join(root, name))
```

Os.walk会自顶向下依次遍历目录信息，以三元组形式输

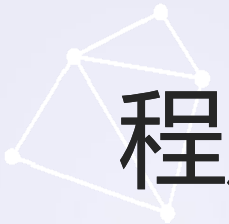
目录文件的操作　　os库

例子：将文件夹下所有文件名字后增加一个字符串_py

```
import os

path=input("输入一个路径:")
for root, dirs, files in os.walk(path):
    for name in files:
        fname, fext = os.path.splitext(name)
        os.rename(os.path.join(root, name),\
                  os.path.join(root, fname+'_py'+fext))
```

python 语言程序设计



程序定时执行　　sched库

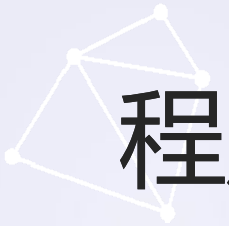
sched库用来进行任务调度

sched.scheduler()用来创建一个调度任务

当需要对一个任务进行时间调度时，用这个库

scheduler.enter(*delay*, *priority*, *action*, *argument=()*)

创建一个调度事件，argument中是action()的参数部分



程序定时执行　　sched库

scheduler.run() 运行调度任务中的全部调度事件

scheduler.cancel(event)取消某个调度事件

程序定时执行　　sched库

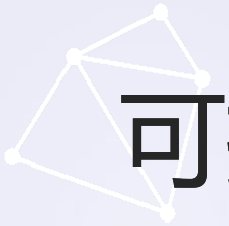
例子：函数定时执行　func_sched.py

```
import sched, time

def print_time(msg='default'):
    print("当前时间", time.time(),msg)

s = sched.scheduler(time.time, time.sleep)
print(time.time())
s.enter(5, 1, print_time, argument=('延迟5秒,优先级1',))
s.enter(3, 2, print_time, argument=('延迟3秒,优先级2',))
s.enter(3, 1, print_time, argument=('延迟3秒,优先级1',))
s.run()
print(time.time())
```

python 语言程序设计



可执行程序的转换　　py2exe库

问题：在windows平台下，使用exe文件执行，如何将python程序变成exe程序，并在没有python环境的情况下运行呢？

可执行程序的转换　　py2exe库

步骤1：确定python程序可执行， func_sched.py

步骤2：写一个发布脚本 setup.py

```
from distutils.core import setup
import py2exe

setup(console=['func_sched.py'])
```

可执行程序的转换　　py2exe库

步骤3：在windows命令行cmd下运行

```
E:\>python setup.py py2exe
running py2exe

  3 missing Modules
  -----------------
? pywintypes                            imported from -
? readline                              imported from cmd, code, pdb
? win32api                              imported from platform
Building 'dist\func_sched.exe'.
Building shared code archive 'dist\library.zip'.
Copy c:\windows\system32\python34.dll to dist
Copy C:\Python34\DLLs\unicodedata.pyd to dist\unicodedata.pyd
Copy C:\Python34\DLLs\_lzma.pyd to dist\_lzma.pyd
Copy C:\Python34\DLLs\pyexpat.pyd to dist\pyexpat.pyd
Copy C:\Python34\DLLs\_hashlib.pyd to dist\_hashlib.pyd
Copy C:\Python34\DLLs\_ssl.pyd to dist\_ssl.pyd
Copy C:\Python34\DLLs\_socket.pyd to dist\_socket.pyd
Copy C:\Python34\DLLs\select.pyd to dist\select.pyd
Copy C:\Python34\DLLs\_ctypes.pyd to dist\_ctypes.pyd
Copy C:\Python34\DLLs\_bz2.pyd to dist\_bz2.pyd
```

python 语言程序设计

可执行程序的转换　　py2exe库

步骤4：运行结果

生成两个目录：dist和__pycache__

其中，dist中包含了发布的exe程序

__pycache__是过程文件，可以删除

注意：目录dist需要整体拷贝到其他系统使用，因为，其中包含了exe运行的依赖库，不能只拷贝exe文件