

# 让虚谷号定时执行任务

谢作如 浙江省温州中学

涉及学科：信息技术

在去年的疫情期间,我用虚谷号做了一个能够定时打铃、广播的创客作品——定时铃声广播系统,让家里的作息时间和学校保持一致。想不到这个简单的作品吸引了很多朋友的目光,常常有人来询问作品的编程细节问题。

从功能上分析,我们常见的创客作品大都属于“动作触发”类型,借助传感器感知外界的信息,然后根据输入的信息来执行相应的动作。而定时铃声广播系统则是根据系统内部的时间进行触发,属于“时间触发”类型。这种作品的编程重点在于如何定时执行一个个任务,即实现定时任务或者任务调度功能。

## ● “定时任务”的技术分析

我们知道,现在的计算机操作系统都支持“多任务”(Multitasking)。虽然计算机的中央处理器(CPU)在同一时刻只能运行一个程序(双核心则支持两个),但是由于CPU的速度极快,每秒能执行几十亿条机器语言指令,因此系统可以划分出微秒级的时间片,通过合理的任务调度快速切换

执行程序,从人类的角度看就是在同时运行了。

虚谷号上运行的是Linux系统,有很多方法实现任务调度。说起用Python代码实现定时执行任务,大家肯定会想到用“sleep”,代码如图1所示。

在循环中加“sleep”的做法显然不够好,最大的缺点是只能执行固定间隔时间的任务,不支持定时任务,如早上六点半播放起床铃声。另外,“sleep”是一个阻塞函数,也就是说在“sleep”这一段时间,计算机什么都不能做。因而,要实现很多任务的调度,一般会使用Python标准库中的sched模块,参

```
import time
while True:
    print("执行任务……")
    time.sleep(5)
```

图1

```
import sched
import time
# 任务
def task():
    print("执行任务……")

# 初始化 sched 模块的 scheduler 类
scheduler = sched.scheduler(time.time, time.sleep)
# 增加调度任务
scheduler.enter(10, 1, task)
# 运行任务
scheduler.run()
```

图2

考代码如图2所示。

sched模块在调度器类使用一个延迟函数,等待特定的时间执行任务,同时支持多线程应用程序,在每个任务执行后会立刻调用延时函数,以确保其他线程也能执行。但是,如果要调度的任务是周期性的,而且任务比较多,用sched模块实现依然不够方便。这时,我们需要使用功能更加强大的第三方库,如schedule或者APScheduler。

## ● Python的任务调度库

### Schedule简介

Schedule是一个非常小巧的任务调度库,其安装方法如下:

```
pip install schedule
```

Schedule的使用方法很简单,只要把要执行的任务封装为一个个函数,然后用Schedule去调用对应的函数名即可,代码可读性非常好。参考代码如下页图3所示。

看到“seconds.do”的语句,是不是立刻秒懂?这就是Schedule的最大优势之一。Schedule支持很多种

调度任务的方法,具体如下页表所示:

Schedule还支持传递参数,具体请参考Schedule的官方文档。

(地址:<https://schedule.readthedocs.io/en/stable/>)

需要强调的是,Schedule也是阻塞型的。如果要避免阻塞,我们可以借助threading模块,以多线程的方式进行工作。

## ● 定时铃声广播系统的代码编写

我做的定时铃声广播系统并没有使用其他硬件,一块虚谷号加声卡和小音箱即可。当然,因为定时广播内容来自网络,如天气预报和英语听力,所以需要设置上网。这里的代码其实适用于绝大多数类似虚谷号的迷你电脑,如树莓派、拿铁熊猫,或者其他小型的电脑主机。

### 1. 获取天气预报信息

这里采用的是“心知天气”的接口,用Requests获取,然后逐一取出需要的信息即可。参考代码如图4所示。

### 2. 获取英语听力内容

通过Requests获取一个词典网站上的“每日一句”的内容,再用正则表达式找到需要的内容。参考代码如图5所示。

函数返回两个字符型变量,分别为文本和音频地址,如图6所示。

### 3. 合成声音

虽然声音的合成可以直接用Pytts库,但如果追求语音的质量,则

```
import schedule
# 定义需要执行的任务
def job():
    print("执行任务.....")
# 设置调度的参数,每2秒执行一次
schedule.every(2).seconds.do(job)
while True:
    schedule.run_pending()
```

图3

```
#获取天气预报,参数为城市的名称,如 wenzhou
def get_weather(location):
    API='https://api.seniverse.com/v3/weather/now.json'
    KEY='S_zw8qq3BQPURlWU'
    result = requests.get(API, params={
        'key': KEY,
        'location': location,
        'language': 'zh-Hans',
        'unit': 'c'
    }, timeout=1)
    result=result.json()
    r=result["results"][0]
    s1=r["location"]["name"] #城市名称
    s2=r["now"]["text"] #天气信息
    s3=r["now"]["temperature"] #气温
    s=s1 + '今天的天气' + s2 + ', 气温' + s3
    return s
```

图4

```
def getmryj():
    url="http://dict.cn"
    r = requests.get(url)
    resx=r.text
    pattern = 'clear.dot.gif.*\s.*'
    text1 = re.findall(pattern, resx)
    pattern = '\s\w.*\t'
    text2=re.findall(pattern, text1[0])
    text3=text2[0].replace("&nbsp",".")
    pattern = 'audio=.*\S.mp3'
    text1 = re.findall(pattern, resx)
    pattern = 'http.*\S'
    text2 = re.findall(pattern, text1[0])
    text4 = text2[0].replace("'", "")
    return text3, text4
```

图5

```
In [6]: s1,s2=getmryj()
        print(s1)
        print(s2)

No hesitating. No complaining. And no excuses. I want you to go out in the world and I want you to be winners!..不要犹豫,不要找借口。我要你们走向世界,我要你们成功!
http://dc.dict.cn/v/06723899/0133129d8550cb00adbbb3fdb4d66890/ddsen/1/ddsen_sun002.mp3
```

图6

```
# 输入百度 AI 的 ID、KEY 等信息
APP_ID = "151288888"
API_KEY = "BPas8Kck1B6Io9EqE0w18888"
SECRET_KEY = "AL3B7X0moRZojqFivCzvxuGYDDQZ8888"
client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)
def tts(txt):
    #播放语音之前,先播放提示音
    os.system('play ding.wav')
    time.sleep(1)
    result = client.synthesis(txt, 'zh', 1, {'vol': 5,})
    # 识别正确返回语音二进制 错误则返回 dict 参照下面错误码
    if not isinstance(result, dict):
        with open('audio.mp3', 'wb') as f:
            f.write(result)
        os.system('play audio.mp3')
```

图7

功能	范例
每小时执行	<code>schedule.every().hour.do(job)</code>
每天12:25执行	<code>schedule.every().day.at("12:25").do(job)</code>
每隔2到5分钟执行一次	<code>schedule.every(5).to(10).minutes.do(job)</code>
每星期四的19:15执行	<code>schedule.every().thursday.at("19:15").do(job)</code>
每小时的第15分钟时执行	<code>schedule.every().minute.at(":15").do(job)</code>

```
# 铃声播放
def play_wave(s):
    ling = 'wave/' + s + '.mp3'
    os.system('play ' + ling)
def tips(s):
    tts("虚谷号提醒你" + s)
# 课间英语
def learning():
    txt = "虚谷号英语学习时间："
    s1, s2=getmryj()
    tts(txt + s1)
    time.sleep(5)
    os.system('play ' + s2)
# 天气预报
def weather():
    txt = "虚谷号播放天气预报信息：" + get_weather("wenzhou")
    tts(txt)
# 起床
schedule.every().day.at("07:00").do(play_wave, s="起床")
schedule.every().day.at("07:10").do(weather)
schedule.every().day.at("07:30").do(play_wave, s="早读")
schedule.every().day.at("08:00").do(play_wave, s="上课")
schedule.every().day.at("09:00").do(play_wave, s="下课")
#趣味英语
schedule.every().day.at("09:10").do(learning)
schedule.every().day.at("09:20").do(play_wave, s="上课")
schedule.every().day.at("10:20").do(play_wave, s="下课")
schedule.every().day.at("10:40").do(play_wave, s="上课")
schedule.every().day.at("11:40").do(play_wave, s="下课")
# 午饭
schedule.every().day.at("12:00").do(tips, s="亲爱的主人，要吃午饭了！")
schedule.every().day.at("13:30").do(play_wave, s="上课")
schedule.every().day.at("14:30").do(play_wave, s="下课")
schedule.every().day.at("14:50").do(play_wave, s="上课")
schedule.every().day.at("15:50").do(play_wave, s="下课")
schedule.every().day.at("16:10").do(play_wave, s="上课")
schedule.every().day.at("17:10").do(play_wave, s="下课")
# 晚饭
schedule.every().day.at("18:00").do(tips, s="亲爱的主人，要吃晚饭了！")
schedule.every().day.at("18:10").do(play_wave, s="晚饭")
# 晚自修
schedule.every().day.at("18:30").do(play_wave, s="晚读开始")
schedule.every().day.at("19:30").do(play_wave, s="晚读结束")
schedule.every().day.at("19:50").do(play_wave, s="上课")
schedule.every().day.at("21:30").do(play_wave, s="下课")
# 睡觉
schedule.every().day.at("22:30").do(play_wave, s="就寝")

while True:
    schedule.run_pending()
    time.sleep(1)
```

图8

推荐调用百度AI平台的接口。考虑到突然间的语音播放会吓人一跳,我在播放语音前,先播放一个“叮”的提示

音。参考代码如上页图7所示。

4. 定时播放铃声

虽然核心代码看起来有点长

(如图8),但它类似于计划任务的清单,一看就明白。

相关代码已经收集在虚谷号的课程中,下载地址为:  
<https://github.com/vvlink/vvBoard-docs>。

● “定时任务”功能的应用

在中小学的创客作品中,“动作触发”类多如牛毛,而“时间触发”类却寥寥无几。其中的原因有很多。例如,之前习惯于使用Arduino,而Arduino没有内置时间模块,实现定时任务的功能有点麻烦,大家的思维被局限了。又如,定时任务往往在内部执行,在外部看不出工作流程,展示起来不直观。再如,很多作品并没有“真正”运行,仅仅停留在展示层面。“时间触发”类作品解决的是需要长期运行的问题。

生活中很多问题需要用定时执行的方式去解决,如定时打铃、定时开门、定时拍照等。当我们从Arduino、掌控板等单片机系统,逐步升级到虚谷号、树莓派、拿铁熊猫等能够运行Linux、Windows操作系统的迷你电脑时,借助于Python强大的第三方库,能够实现的功能就越来越多了。创客教师需要引导学生关注生活中的各种应用,不断提出新的需求,不断挑战新的难题。e