**实验2：主机扫描与监控信息采集实验报告**

姓名：江继军 学号：21373222 班级：213921

实验日期：2024年4月3日 实验地点：一号馆406实验室 成绩：

1. **实验目的**
2. 基于Python语言和psutil库，缩写一个系统信息采集器，并存储到一个.txt文件。
3. 通过此实验，初步熟悉主机监控信息采集的Python编程。

**二、实验内容**

1. 环境安装

1. 下载并安装psutil第三库。

2. 编写系统信息采集程序、结果写入文件

1. 编制一个系统信息采集软件，收集自己的主机或虚拟机的基本信息，包含机器的基本信息和网络信息等，如加入其他信息，每种信息按照所加代码难度加1-3分，加分项一般不超过6分；（实验报告页数 ≤ 6页）
2. 对采集结果进行解析，输出到.txt文本文件，要求清晰易读。
3. 采集内容包含但不限于：
4. CPU信息：

* User time：用户时间进程百分比。
* System Time：内核进程和终端的时间百分比。
* Wait IO：由于IO等待使CPU处于空闲状态的时间百分比。
* Idle：CPU处于空闲状态的时间百分比。

1. 内存信息：

* Total (内存总数)
* Used (已使用的内存数)
* Free (空闲内存数)
* Buffers (缓冲使用数)
* Cache (缓存使用数)
* Swap (交换分区使用数)等

1. 磁盘信息

* read\_count(读IO数)
* write\_count(写IO数)
* read\_bytesIO(读字节数)
* write\_bytesIO(写字节数)
* read\_time(磁盘读时间)
* write\_time(磁盘写时间)

1. 网络信息

* bytes\_sent(发送字节数)
* bytes\_recv(接收字节数)
* packets\_sent(发送数据包数)
* packets\_recv(接收数据包数

1. 用户信息

**三、实验环境**

分项列出主要实验环境（主机/操作系统、Python开发环境、Python版本、其他库版本等）：

1. 操作系统：Win11
2. Python开发环境：pycharm-community
3. Python版本：3.9
4. psuyil版本：5.9.8

**四、实验步骤**

1. 使用psutil.cpu\_times\_percent()收集CPU信息

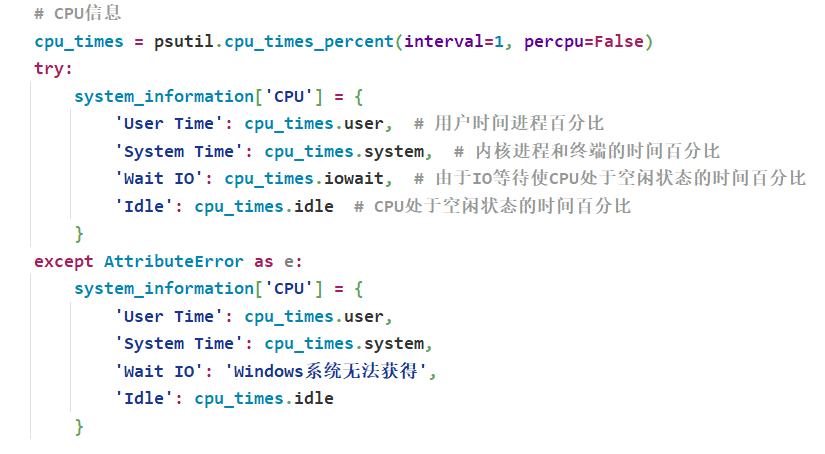


图 1 收集CPU信息

1. 使用psutil.virtual\_memory()和psutil.swap\_memory()收集内存信息



图 2 收集内存信息

1. 使用psutil.disk\_io\_counters()收集磁盘信息

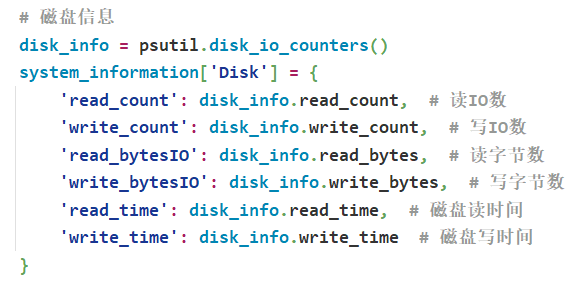


图 3 收集磁盘信息

1. 使用psutil.net\_io\_counters()收集网络信息

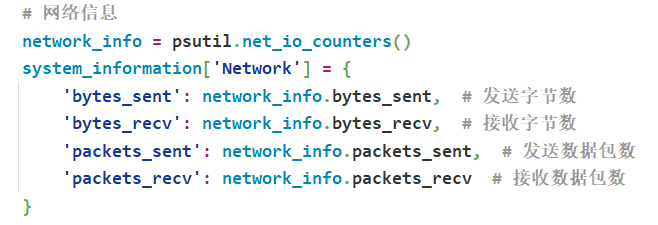


图 4 收集网络信息

1. 使用psutil.users()收集用户信息

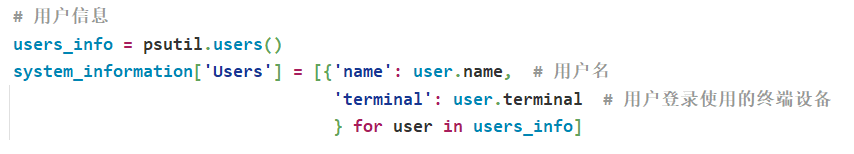


图 5 收集用户信息

1. 收集其他信息

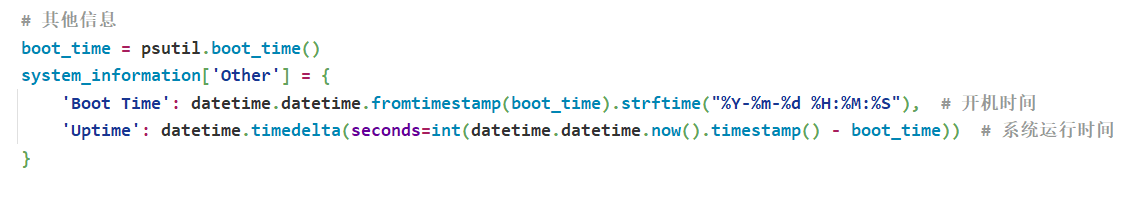


图 6 收集其他信息

1. 将收集到的信息存储为文本文件

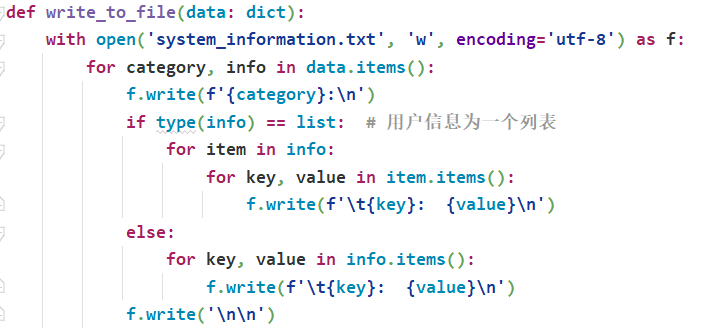


图 7 将收集到的信息存储为文本文件

**五、问题和解决方法**

1、Windows系统无法获得Wait IO、Buffers等信息，使用try关键词，可以让此程序在Linux和Windows系统下都可以运行。

2、在收集用户信息时，需考虑可能有多个用户，在存结果时用到列表，故在最后输出为文本文件时，需要对列表进行判断。

**六、实验总结**

结合本实验内容，谈一下此次实验体会，并进行实验总结。

编写主机系统信息程序的体会：

编写主机系统信息程序是一种很好的实践方式，可以帮助巩固对Python编程语言和相关库的理解和掌握。通过实际编写代码，我更深入地了解了如何使用psutil库来获取系统信息，并将其处理和展示。在编写程序时，良好的代码组织和文档是至关重要的。清晰的代码结构和注释可以让代码更易于理解和维护。特别是在处理各种系统信息时，合适的注释可以帮助他人更好地理解代码的功能和用途。此外，考虑到程序的灵活性和扩展性是很重要的。在设计程序架构时，需要考虑到后续可能添加或修改的功能，以便程序能够轻松地适应变化。

总结：

收集系统信息是一项重要的任务，它可以帮助我们监控系统的状态、诊断问题、优化性能，以及进行容量规划。本次实验让我更加深刻地认识到了系统监控的重要性。收集系统信息是系统监控和管理的重要环节，它为我们了解系统运行状态、发现问题和优化性能提供了有力支持。通过编写主机系统信息程序，我不仅加深了对Python语言的理解，还提高了自己的编程能力。在编写代码的过程中，我学会了如何设计程序架构、处理数据和异常情况，这些都是提高编程能力的重要组成部分。