

计算机原理和操作系统

数据在不通电的情况下可以长期持久地 (persistently) 存储在 磁盘 (如固态硬盘 SSD、机械硬盘 HDD) 或磁带 (常用于数据备份、长期归档) 里。但在需要呈现 (print、render、show、display、play)、计算加工 (compute、transform、analyze、machine learning、deep learning) 或编解码 (encode、decode) 时, 就需要通电的 CPU 和 内存 (硬件), 在操作系统 (软件) 里以 进程 (process) 为单元 (相互隔离) 进行处理。例如, Microsoft Word 启动后就是一个进程, 我们在 Word 进程里打开某个 .docx 文档, 将其从磁盘加载 (读取) 到内存, 然后在图形界面 (GUI) 里查看和编辑 (计算) 内存中的文档, 最后将内存数据保存 (写入) 到磁盘。同理, Python 解释器 (interpreter) 启动后也是一个进程, 她按照流程 (flow) 执行我们准备好的 Python 代码, 根据我们代码的要求, 转告 (即 调用, call) 操作系统或其他软件 (即 依赖项, dependency), 委托她们替我们执行各种 “读取——计算——写入” 等工作。我们并不需要完全理解依赖项内部的工作细节 (黑箱), 只需要清楚每个调用的主体 (即 对象, object) 是什么 类型 (type), 每个调用的输入 (即 参数, parameter/argument)、输出 (即 返回值, return value) 是什么类型, 以及调用会对内存数据、磁盘文件做什么修改, 就足以支持我们自动批量地完成工作了。

1、Fork 第 04 周打卡 仓库至你的名下, 然后将你名下的这个仓库 Clone 到你的本地计算机

git remote show origin //看远程仓库的地址

```
(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ git remote show origin
* remote origin
  Fetch URL: https://gitcode.com/hjwhjw/week04.git
  Push URL: https://gitcode.com/hjwhjw/week04.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)
```

2、用 VS Code 打开项目目录, 新建一个 environment.yml 文件, 指定安装 Python 3.12, 然后运行 conda env create 命令创建 Conda 环境
之前有 environment.yml 文件

```
(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ ls -l ../myproject
total 12514
-rw-r--r-- 1 ABCD 197121      87  3月 20 21:35 environment.yml
-rw-r--r-- 1 ABCD 197121 12811945  3月 20 23:00 EPA_SmartLocationDat
-rw-r--r-- 1 ABCD 197121     696  3月 20 22:58 main.py

(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ cat ../myproject/environment.yml
name: myproject
channels:
  - conda-forge
dependencies:
  - python=3.12
  - pandas
```

复制这个文件到当前文件夹 week04 下面

```
- pandas
(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ cp ../myproject/environment.yml ./
```

创建 conda 环境

```
(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ conda env create
Retrieving notices: done
Channels:
- conda-forge
- https://repo.anaconda.com/pkgs/main
- https://repo.anaconda.com/pkgs/r
- https://repo.anaconda.com/pkgs/msys2
Platform: win-64
Collecting package metadata (repodata.json): /
```

```
done
#
# To activate this environment, use
#
#     $ conda activate week04
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

看看是否能显示中文 cat contacts.txt

```
(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ ls -l
total 25
-rw-r--r-- 1 ABCD 197121 0 4月 1 15:19 contacts.txt
-rw-r--r-- 1 ABCD 197121 74 4月 1 15:17 environment.yml
-rw-r--r-- 1 ABCD 197121 18805 4月 1 14:53 LICENSE
-rw-r--r-- 1 ABCD 197121 2239 4月 1 14:53 README.md

(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ cat contacts.txt

(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ cat contacts.txt
白展堂 男 baizhantang@163.com
佟湘玉 女 tongxiangyu@163.com
吕轻侯 男 lvqinghou@126.com
郭芙蓉 女 guofurong@126.com
李秀莲 男 lixiulian@163.com
祝无双 女 zhuwushuang@163.com
```

(cat)读取内容

将大模型提供的代码复制粘贴进 main.py 文件

```
! environment.yml U  contacts.txt U  main.py U x
main.py > ...
1 def read_contacts(file_path):
2     try:
3         with open(file_path, "r", encoding="utf-8") as file:
4             contacts = []
5             for line in file:
6                 name, gender, email = line.strip().split()
7                 contacts.append((name, gender, email))
8             return contacts
9     except FileNotFoundError:
10        print(f"错误: 未找到文件 {file_path}。")
11        return []
12    except Exception as e:
13        print(f"错误: 读取文件时发生未知错误 {e}。")
14        return []
15
16
17 def generate_emails(contacts):
18     salutations = {"男": "先生", "女": "女士"}
19     emails = []
20     for name, gender, email in contacts:
21         title = salutations.get(gender, "")
22         email_text = f"to: <{email}>\n尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。"
23         emails.append(email_text)
24     return emails
25
26
27 def sort_contacts(contacts):
28     def custom_sort_key(contact):
29         email = contact[2]
30         username, domain = email.split("@")
31         return (domain, username)
32
33     return sorted(contacts, key=custom_sort_key)
34
```

运行 python 读取代码

```
BCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (mai
$ python main.py
```

报错后修改 文本当中不能存在空行

```
BCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ python main.py
错误: 读取文件时发生未知错误 not enough values to unpack (expected 3, got 0)。
已成功生成 emails.txt 文件。
(main)
BCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ python main.py
已成功生成 emails.txt 文件。
(main)
```

```

ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ cat emails.txt
to: <guofurong@126.com>
尊敬的郭芙蓉女士，您的会员资格即将到期，请及时续费。
---
to: <lvqinghou@126.com>
尊敬的吕轻侯先生，您的会员资格即将到期，请及时续费。
---
to: <baizhantang@163.com>
尊敬的白展堂先生，您的会员资格即将到期，请及时续费。
---
to: <lixiluan@163.com>
尊敬的李秀莲先生，您的会员资格即将到期，请及时续费。
---
to: <sunyingsha@163.com>
尊敬的孙颖莎女士，您的会员资格即将到期，请及时续费。
---
to: <tongxiangyu@163.com>
尊敬的佟湘玉女士，您的会员资格即将到期，请及时续费。
---
to: <wangchuqin@163.com>
尊敬的王楚钦先生，您的会员资格即将到期，请及时续费。
---
to: <zhuwushuang@163.com>
尊敬的祝无双女士，您的会员资格即将到期，请及时续费。
---
(week04)

```

运行 `python -m pdb main.py` 命令 (作用是以调试模式 (debug mode) 启动 Python 解释器，准备执行 `main.py` 里的代码)

```

$ python -m pdb main.py
> c:\users\abcd\repo\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb)

```

在 (pdb) 提示符下练习使用 `l`(显示代码)

```

ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\abcd\repo\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) l
1  -> def read_contacts(file_path):
2      try:
3          with open(file_path, "r", encoding="utf-8") as file:
4              contacts = []
5              for line in file:
6                  name, gender, email = line.strip().split()
7                  contacts.append((name, gender, email))
8              return contacts
9      except FileNotFoundError:
10         print(f"错误：未找到文件 {file_path}。")
11         return []
(Pdb) n
> c:\users\abcd\repo\week04\main.py(17)<module>()
-> def generate_emails(contacts):
(Pdb)

```


n (执行当前行)

```
(Pdb) n  
> c:\users\abcd\repo\week04\main.py(17)<module>()  
-> def generate_emails(contacts):
```

p (打印表达式)

s (步入调用) step in 从 17 行到 27 行

```
35         return sorted(contacts, key=custom_sort_key)  
(Pdb) s  
> c:\users\abcd\repo\week04\main.py(27)<module>()  
-> def sort_contacts(contacts):
```

pp (美观打印)

c (继续执行) 等命令

在调试过程中，利用 wat-inspector (第三方软件包，需要安装) 检查 (inspect) 各种对象

```
$ ls -l  
total 34  
-rw-r--r-- 1 ABCD 197121 270 4月 1 16:04 contacts.txt  
-rw-r--r-- 1 ABCD 197121 888 4月 1 20:04 emails.txt  
-rw-r--r-- 1 ABCD 197121 95 4月 1 20:10 environment.  
done  
#  
# To activate this environment, use  
#  
#     $ conda activate week04  
#  
# To deactivate an active environment, use  
#  
#     $ conda deactivate
```

一个语句是有一个 def 定义的

```
emails = []  
for name, gender, email in contacts:  
    title = "先生" if gender == "男" else "女士"
```

For 后面是表达式，变量名也可以作为表达式

Vscode 白色是变量名

缩进代表层级