

Design and Implementation of File Batch Processing System based on Python

Feifei Yin¹, Dong Liu^{1, a}, Bingzhe He²

¹School of North China Electric Power University, BaoDing071000, China

²State Grid Hengshui Electric Power Supply Company,05300, China

^a11206776367@yahoo.com

Abstract

Computer technology and computer software capabilities are increasing, On the one hand, software products are constantly changing and powerful, which can realize highly differentiated complex functions, On the other hand, the file processing function of the computer system is universal and powerful, but it is single and independent, and it still needs the user's hands to a certain extent. This paper is to solve the daily needs of public servants in the unit file batch processing operations, file copy, transfer, rename and other operations encapsulated into a logical chain diagram, visualization, user-defined configuration of the small program. Improve the efficiency and comfort of batch processing documents by staff in school or institution or company.

Keywords

batch processing, python, life service, small program, efficiency office.

1. Preface

In 2020, for the public servants in companies, schools or institutions, they are dealing with a variety of file business, including copying, cutting and renaming. For them, the powerful file management software on the market is highly differentiated and specialized, and it is not easy to use, On the other hand, for the existing operating system, it is not a problem to complete these tasks. Even bat batch command can be used. However, for most employees who don't understand the code, they still have to choose multiple hands-on operations to complete them. This situation has lasted for many years, after all, this is a low-income field in the field of software development, few people have set foot in it. This paper will research and develop a file batch processing program suitable for these civil servants: Office genie.

System function module design:

This system has designed four modules, namely "workspace", "datanode", "operation node", "user management module".

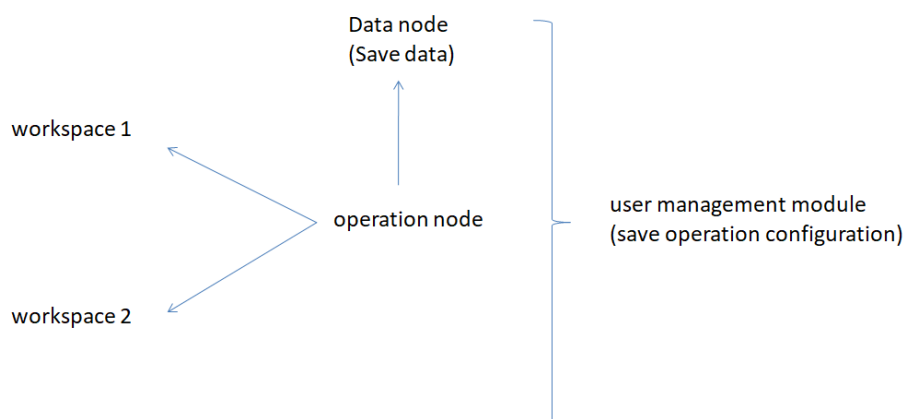


Figure 1. Frame diagram

1.1. Workspace

The workspace is the file directory selected by the user, which contains the contents of the folder that the user wants to process. The user can restrict the files in the workspace, for example, only operate in JPG format or word document format, and there is no restriction by default. Users can select multiple workspaces, and can link multiple workspaces through operation nodes.

1.2. Data Node

In reality, users may link the operations of multiple workspaces. Users can customize the data nodes to save the extracted data during the operation of one workspace, and then call them when operating the second or more workspaces. For example, variables such as file name and file number extracted from workspace 1 are used for other operations in workspace 2. There are many types of data nodes, such as integer, float, string, etc.

1.3. Operation Node

Users can extract the file name, rename, copy and paste, cut and so on. This module encapsulates the above function methods into a class. As a node in the chain diagram, it can be selected by users. Some operation node can only point to two or more workspaces.

1.4. User Management Module

Users can save their own operation configuration for further use. Users can share their operation configuration with others, greatly improving the efficiency of staff. At the same time, this module also has the humanized log management function, which is convenient for users to view.

2. System Data Structure Implementation

The implementation of the system is based on Python language, calling the existing IO file library for implementation.

```
class DataNode:
    def __init__(self,type):
        self.nodeName = ""
        self.dataType = ""
        if type == "int":
            self.data = list<int>[]
        elif type == "string":
            self.data = list<str>[]
        else:
            self.data = list<float>[]
    def SetNodeName(self,name):
        self.nodeName = name
    ...
```

Figure 2. Data nodeClass declaration code

```
class OperationNode:
    def __init__(self,workspace1,workspace2,datanode):
        self.nodeName = ""
        self.opertationName = ""
        self.loop = 1
        self.workSpace1 = workspace1
        self.workSpace2 = workspace2
        self.dataNode = datanode

    def SetNodeName(self,name):
        self.nodeName = name
    def SetOperationName(self,name):
        self.opertationName = name
    def SetLoop(self,loop):
        self.loop = loop

    def ExtractFileName(self):
        ...
    def Copy(self):
        ...
    def Paste(self):
        ...
    def Rename(self):
        ...
    ...
```

Figure 3. Operation node Class declaration code

```
class Workspace:
    def __init__(self):
        self.nodeName = ""
        self.fileType = ""
        self.filePath = ""

    def SetPath(self,path):
        self.filePath = path
    def SetNodeName(self,name):
        self.nodeName = name
    def SetFileType(self,Type):
        self.fileType = Type
    ...
```

Figure 4. WorkspaceClass Declaration Code

3. User Operation Process

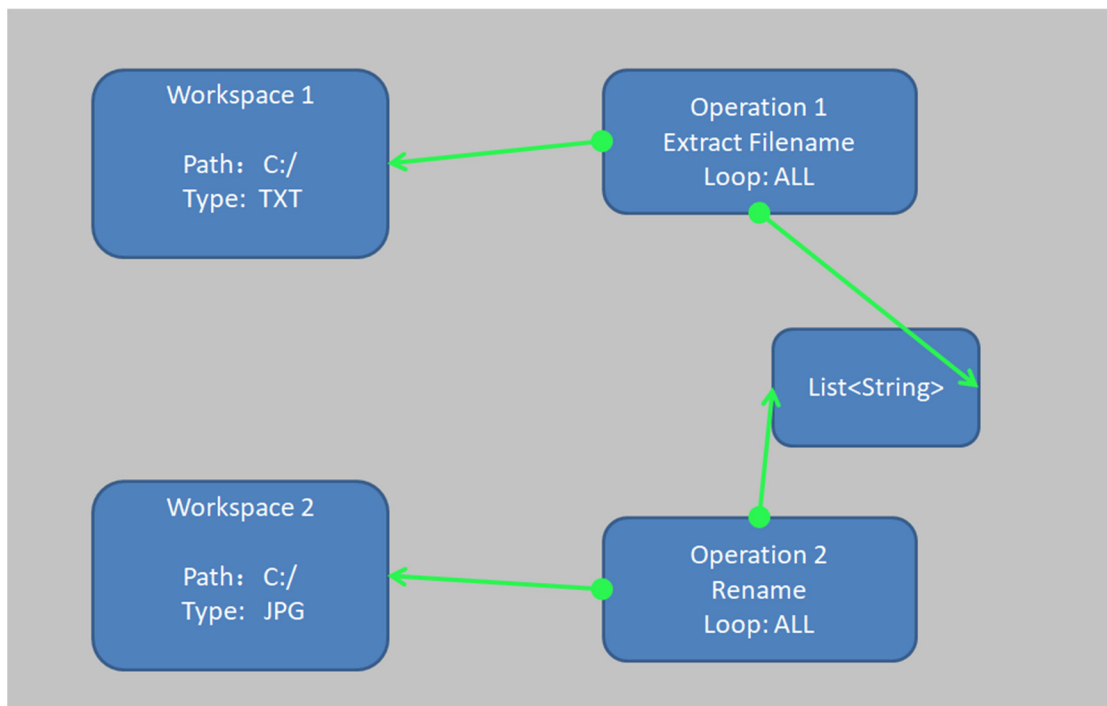


Figure 5. User Interface diagram

Step 1: the user will first create several workspace nodes, and then set the restricted file types. You can set up multiple workspaces.

Step 2: create a data node and set the data type, such as string, integer, float, string; it is a single variable or list. Users can set up multiple data transfer nodes.

Step 3: first, set up the operation node and connect it to the workspace node that the user wants to process. Then, Set the operation method: rename, copy to, transfer to, replace, etc. Finally, Set the number of times to execute (the default is one). When performing operations such as renaming, you need to specify the data source (similarly, bind it to the corresponding data node). For any node, users only need to drag and connect to use these.

Step 4: the user click to execute, the background program will process, after processing, the workspace will be opened. After finishing, users also can save this operation-configuration. Users can click the details button to view the program running log.

4. Advertising Scheme

4.1. Online Advertising can be Carried out in Two Ways:

One is, keyword promotion and search advertising. WeChat small program custom keyword promotion function, enter WeChat small program backstage for Related words selection, then submit audit about a week can bind through the keywords of the audit. According to the user's search habits, the small program combines several keywords such as "batch processing", "file sorting", "utility", "file management", "high efficiency", "visualization", etc. These can improve the probability of users to retrieve small programs.

Two is, online-Association-advertising. Communicate with schools and government departments, and use the official account numbers of the merchants to carry out online promotion. Using the official account official account to promote small programs directly, the main ways of promotion are: association display, setting entrance in the public number custom menu bar, and embedding small program cards in the article when pushing articles. For example official account entrance button is added to the public utility menu bar, pushing articles with links with small programs, setting up official account numbers, attracting fans' attention and so on.

4.2. Offline Promotion Scheme

The offline promotion program includes the use of poster units and cooperation with offline stores.

One is to put up posters and distribute leaflets (with two-dimensional code of small program) in the school or public institutions or companies, and jointly promote with school teachers, enterprise employees, public officials or efficiency office, sustainable development and other lecture activities, to issue publicity leaflets to improve office efficiency and comfort, or to design publicity posters and paste posters to teachers School building, office building, canteen, dormitory building, express station and other densely populated areas, improve the exposure rate of small program QR code on posters.

The second is to cooperate with the stores and supermarkets around the campus, so that the "my points" function of the small program can be linked with the points of store members, and the points can be used to participate in the activities of shop points reduction or points exchange for small gifts to a certain extent. Businesses can also enter the mall function in the campus app. Students can buy the products around the campus through the mall. At the same time, businesses help us to promote the small program.

5. Conclusion

Office wizard, which is based on Python language, is to help employees who are not good at computer and bat batch processing code to realize file batch processing function through visual, user-defined configuration operation steps. Users can even export their customized schemes to generate configuration files and share them with other employees in the same company for rapid task allocation and deployment, which greatly improves office efficiency. The program is compact and concise, and the expansion field is reserved. In the later stage, we can continue to integrate other software tools according to the needs, such as translation, viewing pictures, and other formats of file viewer.

Acknowledgements

The authors acknowledge the Fundamental Research Funds for the Central Universities (Grant: 2018 MS078).

References

- [1] YangJipeng, WenQiangqiang. Software engineering technology analysis in the process of system software development [J]. Technology and market,2020,27(08):102-103.
- [2] Andrea Salandin,David Soler,Michele Bevivino. An Integer Linear Programming approach to minimize the cost of the refurbishment of a façade to improve the energy efficiency of a building[J]. Mathematical Methods in the Applied Sciences,2020,43(14).
- [3] Zhouchun Huang,Qipeng Phil Zheng. A multistage stochastic programming approach for preventive maintenance scheduling of GENCOs with natural gas contract[J]. European Journal of Operational Research,2020,287(3).
- [4] Rui Peng Liu,Alexander Shapiro. Risk neutral reformulation approach to risk averse stochastic programming[J]. European Journal of Operational Research,2020,286(1).
- [5] Hieu T. Nguyen, Frank A. Felder. Generation expansion planning with renewable energy credit markets: A bilevel programming approach[J]. Applied Energy,2020,276.
- [6] Jiuping Xu,Qian Huang,Fengjuan Wang. Co-combustion of municipal solid waste and coal for carbon emission reduction: A bi-level multi-objective programming approach[J]. Journal of Cleaner Production,2020,272.
- [7] Science - Operations Science; Reports Outline Operations Science Study Results from Shanghai Maritime University (A Chance Constrained Fuzzy Goal Programming Approach for Perishable Pharmaceutical Supply Chain Network Design)[J]. Journal of Technology & Science,2020.
- [8] Scientific Programming; Research on Scientific Programming Described by Researchers at Chongqing Technology and Business University (A Grey Target Group Decision Method with Dual Hesitant Fuzzy Information considering Decision-Maker's Loss Aversion)[J]. Science Letter,2020.
- [9] Silicon Storage Technology Inc.; Patent Issued for Method Of Programming A Split-Gate Flash Memory Cell With Erase Gate (USPTO 10,714,489)[J]. Computers, Networks & Communications, 2020.
- [10] SanDisk Technologies LLC; Patent Issued for System And Method For Programming Non-Volatile Memory During Burst Sequential Write (USPTO 10,714,169)[J]. Computers, Networks & Communications, 2020.
- [11] Jing Wang,Lixia Kang,Yongzhong Liu. Optimal scheduling for electric bus fleets based on dynamic programming approach by considering battery capacity fade[J]. Renewable and Sustainable Energy Reviews,2020,130.
- [12] Ben C. Yee,Samuel S. Olivier,Terry S. Haut,Milan Holec,Vladimir Z. Tomov,Peter G. Maginot. A quadratic programming flux correction method for high-order DG discretizations of S N transport[J]. Journal of Computational.
- [13] Di Zhang,Donghui Lv,Changfang Yin,Guilian Liu. Combined pinch and mathematical programming method for coupling integration of reactor and threshold heat exchanger network[J]. Energy, 2020, 205.
- [14] Nianyuan Wu,Xiangyan Zhan,Xingyi Zhu,Zhihui Zhang,Jian Lin,Shan Xie,Chao Meng,Liuxuan Cao,Xiaonan Wang,Nilay Shah,Xuyue Zheng,Yingru Zhao. Analysis of biomass polygeneration integrated energy system based on a mixed-integer nonlinear programming optimization method [J]. Journal of Cleaner Production,2020,271.
- [15] Yi-Chi Wang,Min-Chi Chiu,Toly Chen. A fuzzy nonlinear programming approach for planning energy-efficient wafer fabrication factories [J]. Applied Soft Computing Journal,2020,95.

- [16] Guanxiu Yuan, Yan Gao, Bei Ye, Ripeng Huang. Real-time pricing for smart grid with multi-energy microgrids and uncertain loads: a bilevel programming method[J]. International Journal of Electrical Power and Energy Systems, 2020, 123.
- [17] Rahul Hanmant Waliv, Umakanta Mishra, Harish Garg, Hemant Pandurang Umap. A Nonlinear Programming Approach to Solve the Stochastic Multi-objective Inventory Model Using the Uncertain Information[J]. Arabian Journal for Science and Engineering, 2020, 45(8).
- [18] Nana Wan. Option Contract Design for a Multi-period VMI Supply Chain[J]. Arabian Journal for Science and Engineering, 2020, 45(8).
- [19] Danielle Davidov, Heather M. Bush, Emily R. Clear, Ann L. Coker. Using a Multiphase Mixed Methods Triangulation Design to Measure Bystander Intervention Components and Dose of Violence Prevention Programs on College Campuses[J]. Journal of Family Violence, 2020, 35(6).
- [20] Fu Zhijun, Wang Jingjing. Discussion and Research on Python language learning [J]. Chinese Journal of multimedia and network teaching (first xun Yao), 2019(01):8-9.