

# BASIC INDEXING

ESMA NUR ARSLAN  
ESLEM KURT

MAY 28, 2024

## Overview

1-Firstly, we had an unstructured file. We needed to design a structure.

We decided to use the **Delimiters method to separate fields**. And after that we created the new record file through the record.py script file. Now, we have the appropriate record.dat file for "Delimiters to separate fields".

```
1 court-cases.txt
1 Adams v. Kansas Parole Board (77584)
2 Adkerson v. Nelson (80010)
3 Aetna Life & Cas. v. Americas Truckway Systems, Inc. (74721)
4 Alberty v. Excel Corp. (77948)
5 Alkire v. Fissel (75357)
6 Almena State Bank v. Enfield (77213)
7 Amundson & Assocs. Art Studio, Ltd. v. National Council on Compensation
8 Ins., Inc. (81028)
9 Anderson v. Bill Morris Constr. Co., Inc. (78990)
10 Anderson v. Boeing Co. (78676)
```

```
1 record.dat
1 77584|Adams|Kansas Parole Board
2 80010|Adkerson|Nelson
3 74721|Aetna Life & Cas.|Americas Truckway Systems, Inc.
4 77948|Alberty|Excel Corp.
5 75357|Alkire|Fissel
6 77213|Almena State Bank|Enfield
7 78990|Anderson|Bill Morris Constr. Co., Inc.
8 78676|Anderson|Boeing Co.
9 75477|Anderson|McKune
10 76783|Aspen Products, Inc.|Global Distributors, Inc.
```

### Pseudo code for record.py

class Recording:

    constructor(case\_no, plaintiff, defendant)

    initialize attributes

method pack()

    return formatted string

function main():

    initialize empty recordings list

```

open "court-cases.txt" for reading:
for each line in file:
strip whitespace
if line is empty, skip to next iteration
extract case number from end of line an in the parantesis
if no case number, print error and skip
get remaining line content
if " v. " or " V. " in content:
split into plaintiff and defendant
else:
print error and skip
create Recording object and add to list
print matched details
open "record.dat" for writing:
for each recording in list:
write packed data to file
print written data

if script is main module:
call main()

```

2-According to your expectation, we had to use multiple indexing. We need primary index with caseNo and secondary indexes with plaintiff and defenders. And index files has to sorted.

Our primary key is **caseNo**.

Secondary keys are **plaintiff** and **defendant**.

```

case_no_index.json > ...
1  {
2    "72438": 16933,
3    "72538": 16913,
4    "72568": 11513,
5    "72709": 14372,
6    "72817": 11632,
7    "73344": 12079,
8    "73646": 17833,
9    "73785": 8941,
10   "73787": 7291,

```

view of case\_no\_index

```

plaintiff_index.json > [ ] Harmon
1  {
2    "AMCO Ins. Co.": [
3      10161
4    ],
5    "Achey": [
6      10036
7    ],
8    "Adams": [
9      0,
10     10065
11   ],
12   "Adkerson": [
13     32

```

view of plaintiff\_index

```

defendant_index.json > [ ] American Red Ball Transit Co.
1  {
2    "1985 Chevrolet Astro Van": [
3      4564
4    ],
5    "A.L. Abercrombie, Inc": [
6      5763
7    ],
8    "Abay": [
9      7516
10   ],
11   "Action Rent to Own, Inc.": [
12     11191
13   ],
14   "Allen": [
15     2438,
16     3204,
17     16689

```

view of defendant\_index

## **Pseudo code for index.py**

function build\_index():

- create empty indices

- open "record.dat"

- while not end of file:

  - read line, decode and split

  - update indices with position

- sort indices

- return indices

function save\_index(indices):

- save each index to respective json files

function load\_index():

- load indices from json files, return empty if not found

- return indices

function search\_record\_by\_plaintiff, search\_record\_by\_defendant, search\_record\_by\_case\_no(index, name):

- if name in index:

  - read records at positions from "record.dat"

  - return records

- else:

  - return None

function main():

- indices = load\_index()

- if any index is empty:

  - indices = build\_index()

  - save\_index(indices)

- while true:

  - get user input for search type and key

  - if "exit", break

  - if search by plaintiff:

    - search and print results

  - elif search by defendant:

    - search and print results

elif search by case no:

search and print result

# Demonstrate of index.py Code File Execution

```
/Users/esmanurarslan/anaconda3/bin/python "/Users/esmanurarslan/Library/Mobile Documents/com~apple~CloudDocs/6th term/File Organization/hw/index.py"
(base) esmanurarslan@Esma-MacBook-Pro hw % /Users/esmanurarslan/anaconda3/bin/python "/Users/esmanurarslan/Library/Mobile Documents/com~apple~CloudDocs/6th term/File Orga
nization/hw/index.py"
Enter in which field you want to search:
1. Plaintiff
2. Defendant
3. Case No
Type 'exit' to stop the code from running.
Your choice: 1
Value to seek: Adams
Founded records:
caseNo: 77584
plaintiff: Adams
defendant: Kansas Parole Board
=====
caseNo: 77848
plaintiff: Adams
defendant: St. Francis Regional Med. Center
=====
Enter in which field you want to search:
1. Plaintiff
2. Defendant
3. Case No
Type 'exit' to stop the code from running.
Your choice: 2
Value to seek: City of Kansas City
Founded records:
caseNo: 75707
plaintiff: International Ass'n of Firefighters
defendant: City of Kansas City
=====
caseNo: 75707
plaintiff: International Ass'n of Firefighters
defendant: City of Kansas City
=====
caseNo: 77184
plaintiff: Jackson
defendant: City of Kansas City
=====
Enter in which field you want to search:
1. Plaintiff
2. Defendant
3. Case No
Type 'exit' to stop the code from running.
Your choice: 3
Value to seek: 82095
Founded records:
caseNo: 82095
plaintiff: Hartford Cas. Co.
defendant: Credit Union 1 of Kansas
Enter in which field you want to search:
1. Plaintiff
2. Defendant
3. Case No
Type 'exit' to stop the code from running.
Your choice: █
```