



Clase 18 – Repaso Archivos

IIC1103 Sección 9 – 2019-2

Profesor: Felipe López

03-12-2019

Resumen de la clase

Para leer archivos:

- Abrir un archivo y asignarlo a una variable:
`variable_archivo = open(nombre_archivo).`
- Para poder leer un archivo ocupar la función `readlines()` en `variable_archivo`. Esta función devuelve una lista con todas las líneas del archivo leído.
`lineas_archivo = variable_archivo.readlines()`

Para escribir en archivos:

- Abrir un archivo y asignarlo a una variable. Puede ser en modo “a” para agregar información a un archivo que ya contiene información. O “w” para crear uno nuevo con contenido nuevo (este modo sobrescribe el contenido original del archivo):
`variable_archivo = open(nombre_archivo, (modo)).`
- Escribir usando la función `write()` en la variable que contiene al archivo.
`variable_archivo.write(texto).`
- Después de escribir el texto en el archivo, usar la función `close()` en la variable que almacena el archivo.
`variable_archivo.close()`

Ejercicio 1 (P2 I2 2018-2)

Pregunta 2

El DCC ha ofrecido un gran premio al profesor más popular del curso IIC1103. Para determinar al ganador, los profesores del curso han decidido realizar una votación en cada sección, para que cada alumno vote por su profesor favorito (pueden votar por el profesor de cualquier sección). Luego de que todos votaron, notaron que varios alumnos se equivocaron: **votaron dos veces en la misma sección y/o votaron en más de una sección.**

Las votaciones de los alumnos se encuentran en N archivos de nombre `votaciones_sX.txt` (`votaciones_s1.txt`, `votaciones_s2.txt`, ..., `votaciones_sN.txt`). Cada uno de estos archivos tiene el siguiente formato en cada línea: `num_alumno,profesor`, donde `num_alumno` es un entero con el número de alumno, y `profesor` es un *string* de una lista de N elementos llamada `lista_profesores` que puedes utilizar en tu programa. Puedes suponer que cada archivo `votaciones_sX.txt` cuenta con al menos la votación de todos los alumnos de la sección X .

Además, cuentas con el archivo `alumnos.txt` que tiene el siguiente formato en cada línea: `num_alumno,seccion`, donde `num_alumno` es un entero con el número de alumno, y `seccion` es un entero desde 1 a N (inclusive). No puedes asumir que los archivos se encuentran ordenados (por número de alumno o sección). A continuación se muestra un ejemplo con $N=3$ secciones donde `lista_profesores = ["Marquinez", "Sepulveda", "Lopez"]`.

```
1,Marquinez
2,Marquinez
3,Sepulveda
4,Marquinez
5,Marquinez
```

(a)
votaciones_s1.txt

```
5,Marquinez
6,Sepulveda
7,Lopez
8,Sepulveda
8,Sepulveda
```

(b)
votaciones_s2.txt

```
8,Lopez
9,Sepulveda
10,Lopez
11,Lopez
12,Lopez
```

(c)
votaciones_s3.txt

```
1,1
2,1
3,1
4,1
5,2
6,2
7,2
8,2
9,3
10,3
11,3
12,3
```

(d)
alumnos.txt

Para este ejemplo, vemos que hubo 5 votos en cada sección. Sin embargo, existen los siguientes errores:

- El alumno 5 no pertenece a la sección 1.
- El alumno 8 votó dos veces en la sección 2.
- El alumno 8 no pertenece a la sección 3.

De esta forma, los votos válidos serían los siguientes:

s1	s2	s3
1,Marquinez	5,Marquinez	9,Sepulveda
2,Marquinez	6,Sepulveda	10,Lopez
3,Sepulveda	7,Lopez	11,Lopez
4,Marquinez	8,Sepulveda	12,Lopez

(e) votos válidos

Ejercicio 1 (P2 I2 2018-2)

Tu objetivo es consolidar los votos de todos los alumnos, corrigiendo los errores que pueden existir (considera que en caso que un alumno vota dos veces, puedes tomar cualquiera de los dos votos). Además, dado que el DCC quiere reutilizar el programa que escribas para otros semestres, debes considerar que este funcione para cualquier número de secciones (donde N corresponde al número de secciones). Deberás escribir un nuevo archivo llamado `resultados.txt` en que cada línea debe tener el formato: `profesor,nvotos` donde `profesor` es un *string* de `lista_profesores` y `nvotos` es la cantidad de votos que obtuvo cada profesor. El resultado del ejemplo anterior se muestra a continuación:

Marquinez,4
Sepulveda,4
Lopez,4

(f)
`resultados.txt`

Ejercicio 2 (P3 EX 2019-1)

Pregunta 3

Semestre tras semestre, hay gente que copia en las tareas creyendo que con solo cambiar el nombre de las variables no será detectada la copia. Por esto, se te pide hacer un programa que detecte estas copias.

Tu programa recibirá dos archivos: `codigoA.txt` y `codigoB.txt`, en donde cada uno representa un código a revisar. Los códigos son un conjunto ordenado de instrucciones, y cada una de estas instrucciones es de la forma `variable=int+int` (solo habrá ese tipo de instrucciones). Puedes asumir que en cada línea de los archivos, solo habrá una instrucción, que los nombres de las variables no se repiten dentro de un mismo archivo, que todas las operaciones tendrán solo 2 sumandos y que no habrá espacios. A continuación, se muestran dos archivos de ejemplo: el de la izquierda es el `codigoA.txt` y el del lado derecho es el `codigoB.txt`.

```
1      hola=1+2
2      mundo=2+3
3      var=12+13
4      aux=5+12
5      aux2=7+4
6      valor1=95+14
7      valor2=28+3
8      viva_chile=12+4
9      y=1+2
10     b=3+8
11     a=14+95
12     jug1=5+3
```

```
1      rango=1+2
2      y=2+1
3      contador=28+3
4      auxiliar=11+5
5      aumento=5+12
6      total=1+2
7      suma_total=13+12
8      indice=7+9
9      posicion_x=8+3
10     posicion_y=14+9
11     valor1=14+95
12     valor2=2+2
13     valor3=5+8
14     jug1=5+3
```


Ejercicio 2 (P3 EX 2019-1)

Se sospecha que dos instrucciones son copiadas si la operación es igual pero la variable se llama distinto (ejemplo: `a=10+2` y `b=10+2` son copia), o bien si la variable se llama igual y los operadores de la suma son los mismos pero están en distinto orden (ejemplo: `a=1+2` y `a=2+1` son copia).

Por último, se concluye la sospecha de que el código A fue copiado del código B si el 90% de las instrucciones de A son copiadas del código B. No importa el orden de éstas. Además de esto, si hay dos instrucciones distintas de A que coinciden con una misma instrucción de B, solo se contabiliza una de ellas.

Debes escribir un programa que lea los dos archivos, y guarde un nuevo archivo `copiaAB.txt` en donde, en cada línea del archivo se debe poner qué fila del archivo `codigoA.txt` es copia de qué fila del archivo `codigoB.txt`, y finalmente si `codigoA.txt` fue copia de `codigoA.txt` o no. Para el ejemplo proporcionado anteriormente, se tiene que:

- La instrucción 1 de A es copia de la instrucción 1 de B, pues tienen la misma operación pero nombre cambiado.
- La instrucción 2 de A no es copia de B.
- La instrucción 3 de A no es copia de B. Notar que, en particular **no** se considera copia de la instrucción 7 de B.
- La instrucción 4 de A es copia de la instrucción 5 de B, pues tienen el mismo nombre de variable pero los operadores cambiados.
- La instrucción 5 de A no es copia de B.
- La instrucción 6 de A es copia de la instrucción 11 de B, pues tienen el mismo nombre de variable pero los operadores cambiados.
- La instrucción 7 de A es copia de la instrucción 3 de B, pues tienen la misma operación pero nombre cambiado.
- La instrucción 8 de A no es copia de B.
- La instrucción 9 de A es copia de la instrucción 6 de B, pues tienen la misma operación pero nombre cambiado.
- La instrucción 10 de A no es copia de B. Notar que, en particular **no** se considera copia de la instrucción 9 de B.
- La instrucción 11 de A es copia de la instrucción 11 de B, pues tienen la misma operación pero nombre cambiado. Sin embargo, como la instrucción 11 de B ya arrojó copia con la instrucción 6, no es contabilizada como copiada y no hay ninguna otra de B que coincida y no haya sido usada ya.

Ejercicio 2 (P3 EX 2019-1)

- La instrucción 12 de A es copia de la instrucción 14 de B, pues tienen la misma operación y el mismo nombre.

De un total de 12 instrucciones, solo 6 fueron copia, lo que significa que el 50 % ($\frac{6}{12} \cdot 100$) de las instrucciones eran copia. Luego, se concluye que A no era una copia de B, pues ese porcentaje es menor a 90 %. Finalmente, el archivo `copiaAB.txt` sería el siguiente:

```
1  línea 1 copia de línea 1
2  línea 4 copia de línea 5
3  línea 6 copia de línea 11
4  línea 7 copia de línea 3
5  línea 9 copia de línea 6
6  línea 12 copia de línea 14
7  codigoA.txt no es copia de codigoB.txt
```

Links

- <https://repl.it/@FelipeLopez/IIC1103RepasoArchivos> que contiene todos los ejemplos de la clase.



Clase 18 – Repaso Archivos

IIC1103 Sección 9 – 2019-2

Profesor: Felipe López

03-12-2019