A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

10/01/2020

# TP C++ : Gestion des entrées/sorties

## 3IF

NASSREDDINE Zakaria - CROWLEY Sophie -  
EDDAHABI Riad

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

## Table des matières

I.	Introduction	1
II.	Format du fichier	2
III.	Contenu du fichier demo	2
IV.	Informations supplémentaires	3
V.	Conclusion	3

### **I. Introduction**

Dans le but de nous familiariser avec la gestion des entrées/sorties, nous avons repris notre application réalisée précédemment pour l'améliorer en proposant à l'utilisateur de récupérer des données à partir d'un fichier, ou d'enregistrer des données du catalogue dans un fichier.

L'architecture (diagramme UML et héritage) est identique à celle présentée dans le cadre du TP précédent.

Cette sauvegarde (ou restitution) des données se fera selon la spécification de l'utilisateur. Nous proposons donc 4 types de sauvegarde/restitution :

- Sans critères : tous les trajets contenus dans le catalogue sont enregistrés dans un fichier, et tous les trajets enregistrés dans un fichier sont restitués sur le catalogue.
- Selon le type de trajet : selon le type de trajet, qu'il soit simple ou composé, nous restituons (respectivement sauvegardons) les données dans le catalogue (respectivement dans le fichier).
- Selon la ville de départ et/ou d'arrivée : nous restituons (respectivement sauvegardons) les données dans le catalogue (respectivement dans le fichier).
- Selon une sélection de trajet : selon un intervalle  $[n,m]$  indiqué par l'utilisateur, nous restituons (respectivement sauvegardons) les trajets d'indice  $n$  à  $m$  dans le catalogue (respectivement dans le fichier).

Pour ce qui est de la gestion de fichiers, on demande à l'utilisateur de rentrer le nom de fichier pour la sauvegarde. L'extension « .txt » est rajoutée automatiquement, et un fichier texte est créé lorsqu'il n'existe pas déjà. Lorsque le nom est déjà utilisé, on écrase l'ancien fichier et on remplit le nouveau, lorsque le fichier n'existe pas (nom non

utilisé), on le crée et on le remplit. Quant à la restitution, il faut que le nom du fichier indiqué par l'utilisateur soit exact, sinon il ne se passe rien.

## II. Format du fichier

Afin de restituer facilement les trajets, il fallait réfléchir à une structure d'enregistrement qui permettent une lecture humaine et logicielle facile.

Nous avons déjà commencé par séparer un trajet simple d'un composé à l'aide d'un tag à chaque début de ligne, TS pour trajet simple, TC pour un trajet composé.

Pour ce qui est des trajets simples, en plus du tag TS, on renseigne, dans l'ordre, la ville de départ, la ville d'arrivée, et le moyen de transport. Chaque information est séparée par un « | » qui va permettre de délimiter chaque information dans notre algorithme, et on marque la fin des informations par un « ; ». Exemple :

TS|Casablanca|Paris|Plane;

Pour ce qui est des trajets composés, en plus du tag TC, on renseigne le nombre de trajets simples qui le compose, ainsi que la ville de départ et d'arrivée. Exemple :

TC|2|Casablanca|Bordeaux;

A partir du nombre de trajets simples qui le composent, on affiche tous ces trajets simples à la suite, suivant le même modèle utilisé pour les trajets simples, décrits précédemment.

TC|2|Casablanca|Bordeaux;

TS|Casablanca|Paris|Plane;

TS|Paris|Bordeaux|Train;

----- End of elementary itineraries;

On ajoute un délimiteur (« *End of elementary itineraries* ») qui n'est pas forcément utile pour l'algorithme (la restitution se fait sans problèmes sans cette ligne) mais qui est plus utile à l'utilisateur lorsqu'il lit directement le fichier texte, il n'a pas besoin de compter le nombre de trajets simples pour savoir où s'arrête le trajet composé qui est décrit.

## III. Contenu du fichier demo

Selon le jeu d'essai décrit dans le sujet, le contenu du fichier demo.txt obtenu à la suite d'une sauvegarde serait le suivant :

TS|Lyon|Bordeaux|Train;

TC|2|Lyon|Paris;

TS|Lyon|Marseille|Bateau;

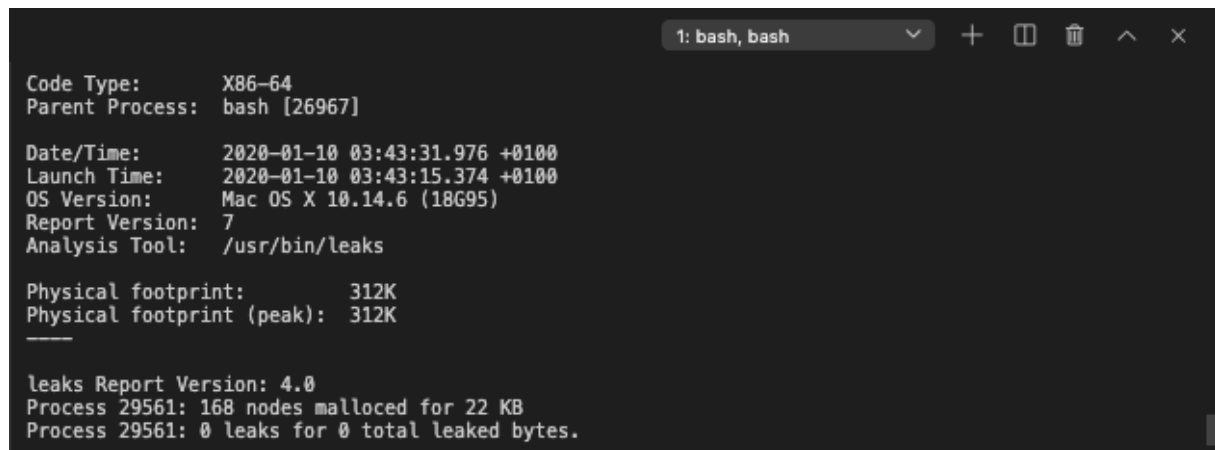
TS|Marseille|Paris|Avion;

----- End of elementary itineraries;

TS|Lyon|Paris|Auto;

## IV. Informations supplémentaires

Un des problèmes rencontrés lors de la réalisation de ce projet était la gestion des fuites. Nous avons réussi à faire en sorte que notre programme ne présente pas de fuites de mémoires, grâce à l'utilisation de l'outil "leaks" sur MacOS X.



```
1: bash, bash
Code Type:           X86-64
Parent Process:      bash [26967]

Date/Time:           2020-01-10 03:43:31.976 +0100
Launch Time:         2020-01-10 03:43:15.374 +0100
OS Version:          Mac OS X 10.14.6 (18G95)
Report Version:      7
Analysis Tool:        /usr/bin/leaks

Physical footprint:   312K
Physical footprint (peak): 312K
-----

leaks Report Version: 4.0
Process 29561: 168 nodes malloced for 22 KB
Process 29561: 0 leaks for 0 total leaked bytes.
```

## V. Conclusion

En conclusion, cette amélioration nous a permis d'en apprendre plus sur les entrées et sorties, la gestion des flux ... Et notre application n'en est que plus complète.

Concernant les problèmes rencontrés, il y a tout d'abord eu la gestion de la lecture des lignes du fichier lors de la restitution. Nous sommes passés par la commande *substr()* pour récupérer la donnée située entre nos délimiteurs. Nous avons également été amené à discuter et faire des choix pour se mettre d'accord sur les conventions à adopter, compte tenu de l'ambiguïté du cahier des charges. Nous ne nous attendions pas à devoir retravailler la gestion de la mémoire mais ayant eu recours à des allocations dynamiques dans les algorithmes de critères de sélection, cela s'est avéré nécessaire.

Enfin, pour ce qui est des améliorations possibles, premièrement, il est toujours envisageable d'implémenter une interface graphique à notre application, qui la rendrait plus intuitive, à l'aide de la bibliothèque *gtkmm* par exemple. Il aurait également été beaucoup plus pertinent de pouvoir combiner plusieurs filtres (ou critères de sélection) à la fois comme dans de vraies applications de parcours de trajets.