

INSA Lyon  
Enzo Boscher  
Tuoyuan Tan

# RAPPORT TP4

## C++

3IF 2019-2020

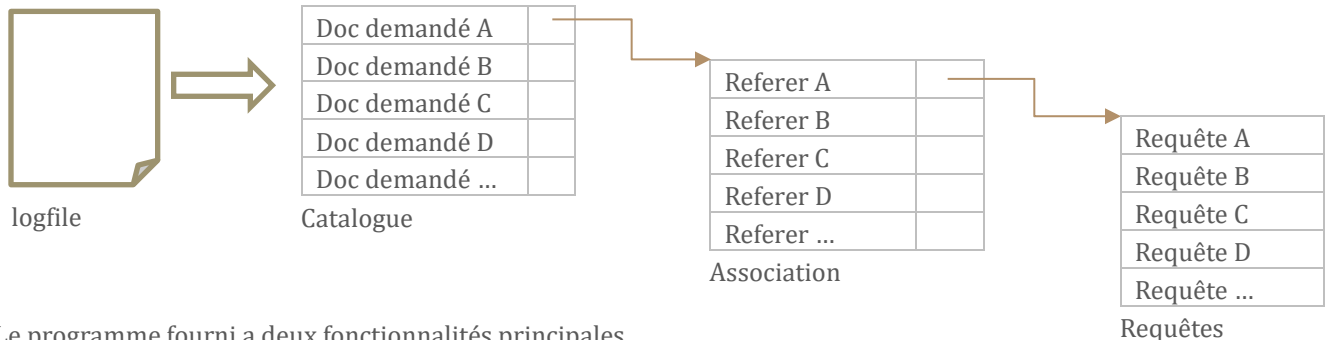
*Application analog*

# TABLE DES MATIERES

Conception et spécification	1
Gestion des erreurs	3

# Conception et spécification

## I. Schéma de données



Le programme fourni a deux fonctionnalités principales. Par défaut, il affiche les 10 documents les plus consultés. De plus, il fournira une description textuelle d'un graphe à partir du fichier de log quand l'utilisateur choisit l'option -g. Pour cette dernière fonctionnalité, il faut afficher tout d'abord tous les nœuds existant dans le fichier original et puis les relations de visite entre des nœuds.

Pour faciliter le stockage des relations entre ces nœuds, on décide de créer un couple. Le premier élément est un nœud (un document demandé), le deuxième est une autre structure qui permet de stocker tous les référencers qui ont visité ce document (pour les nœuds qui n'ont jamais été visités, cette structure sera vide). Chaque couple doit avoir une et une seule place dans notre conteneur. Donc on a besoin d'un conteneur qui contient des paires et qui assure que chaque nœud est unique. De plus, l'insertion dans ce conteneur doit être assez rapide. Par conséquent, on choisit un **map** pour stocker des paires qui prennent le document demandé comme la clé.

Ensuite, il faut trouver une structure qui permet de stocker les référencers qui réfèrent sur un même document. A priori, chacune ligne du fichier log est transformée en une **Requête**, ce dernier est un objet que l'on définit. Même s'il suffit juste de stocker le nombre de visites entre deux nœuds, on décide de stocker toutes les **Requêtes** dans notre structure pour faciliter l'implémentation de nouvelles fonctionnalités dans le futur. Pour cela, on crée de nouveau un couple. Cette fois-ci, le premier élément devient un référencer et le deuxième devient une autre structure qui stocke tous les **Requêtes** qui témoignent de la relation entre ce référencer et le document demandé. En comptant le nombre de ces **Requêtes**, on obtient le nombre de visites de ce référencer au document demandé. De même, on a besoin d'un conteneur qui contient des paires et qui assure que chaque référencer est unique. Donc on choisit un **map** en prenant le nom de référencer comme la clé.

Enfin, on a besoin d'un conteneur qui stocke les **Requêtes**. En effet, ce conteneur ne sert à rien, il n'est pas utilisé dans la suite. Il faut juste choisir un conteneur qui permet d'insérer une **Requête** à la fin très

rapidement. Donc on choisit un **vector**. Il peut y avoir plusieurs solutions possibles, on le choisit juste parce que l'on connaît très bien cette structure-là.

## II. Cas normaux

<i>Numéro cas</i>	<i>Description</i>	<i>Entrées</i>	<i>Sorties attendues</i>
0	Utilisation standard sans option	./analog <filename.log>	Top10 des objets les plus demandés
1	Ajout de l'option -g	./analog -g <filename.dot> <filename.log>	Top10 + creation d'un graph dans le fichier dot
2	Ajout de l'option -e	./analog -e <filename.log>	Top10 avec omission des fichiers .jpg, .png, .css, .js, .gif
3	Ajout de l'option -t	./analog -t <heure> <filename.log>	Top10 entre heure et heure+1
X	Toute combinaison possible des options disponibles	./analog <combinaison d'options> <filename.log>	Top10 + Traitement approprié à chaque option (voir ci-dessus)

## Gestion des erreurs

<i>Numéro cas</i>	<i>Description</i>	<i>Entrées</i>	<i>Sorties attendues</i>
20	Utilisation standard sans option mais omission du fichier de log à analyser	./analog	Usage : [options] <filename.log>
21	Ajout d'une option mais pas de '-'	./analog g <filename.dot> <filename.log>	Options accepted : <list of options> don't forget to input a '-' before any option
22	Option inconnue	./analog -z <filename.log>	Options accepted : <list of options>
23	Utilisation de -g 2 fois	./analog -g <filename.dot> -g <filename.dot> <filename.log>	You can't use -g twice
24	Utilisation de -g : pas de fichier en entrée	./analog -g <filename.log>	-g needs a file with the .dot extension
25	Utilisation de -g : mauvais format d'entrée du fichier	./analog -g <filename.zip> <filename.log>	-g File needs to have .dot extension
26	Utilisation de -e 2 fois	./analog -e -e <filename.log>	You can't use -e twice
27	Utilisation de -t 2 fois	./analog -t hour -t hour <filename.log>	You can't use -t twice
28	Utilisation de -t : pas d'heure donnée	./analog -t <filename.log>	-t needs an hour
29	Utilisation de -t : l'heure donnée n'est pas un entier positif	./analog -t ahah <filename.log>	Hour needs to be a positive number
30	Utilisation de -t : l'heure donnée n'est pas entre 0 et 23	./analog -t 24 <filename.log>	Hour needs to be between 0 & 23
31	Le fichier log n'a pas l'extension .log	./analog <filename.ouaip>	Verify your log file has the .log extension
32	Le fichier .log n'a pas pu être ouvert	./analog <filename.log>	Error when opening the log file
33	Le fichier .dot n'a pas pu être ouvert	./analog -g <filename.dot> <filename.log>	Error when opening the dot file

Note: le fichier .dot ne devrait jamais générer d'erreur étant donné qu'il est généré si jamais il n'existe pas.  
(à moins que le disque soit plein...)

