



# Laboratoire 3 : Traitement des données GPS

Fait par :

HOUIMLI SAFA

HOUS07608901

GAGNON MICHAEL

GAGM04129102

LEFEBVRE JONATHAN

LEFJ01119100

Cours GPA435 : Systèmes d'exploitation et programmation de  
systèmes

Remis à : **Tony Wong**

## 1- Introduction :

Dans ce laboratoire nous avons été tâché de créer un programme qui peut analyser des fichiers de GPS et ressortir des données pertinentes basée sur le trajet parcouru. Les GPS produise un fichier GPX où l'on peut trouver tout les détails de chaque point de repère du cheminement.

Le but de ce laboratoire était de nous rendre plus habile avec la programmation avancé de BASH, les filtres programmables AWK, et aussi l'analyse de performance de programmes.

Pour ce laboratoire, nous avons faits un programme qui analyse des fichiers GPX et crée une base de données à partir des informations dérivées. Avec la base de données, le programme peut ensuite calculer la vitesse moyenne, distance parcouru totale et autres statistiques pertinentes désirés.

## 2- Investigation :

### a) Décomposition du programme

Le programme est décomposé en plusieurs modules dont chacun traite une fonctionnalité spécifique. Il y'a en tout 4 sous-programmes chacun dans un fichier à part :

- 1- validation\_gpx : pour la validation du fichier GPX
- 2- creation\_bdd : pour la création de la base de données à partir du fichier gpx donné
- 3- validation\_bdd : pour valider le fichier de base de donnée fourni
- 4- calculeGPX : pour traiter les calculs de temps, distance, vitesse moyenne et élévation minimale

- **Validation du fichier GPX :**

La validation du fichier GPX est réalisé à l'aide du programme gawk valide\_gpx\_v3.nawk fournis par Tony Wong lors du cour GPA435. Aucune modification a été apporté au document excepté son nom qui, pour des fins pratiques et esthétique, a été changé pour "validation\_gpx.nawk". Un filtre regex a été ajouté afin d'analyser les résultats obtenue par ce programme. Le programme gawk fournis permet d'analyser les données balises par balises avec un système de machine d'état. Le fichier GPX doit respecter les format standard de fichier GPX, ce qui veut dire respecter la logique suivante :

- 1) Avoir un en-tête correspondant aux standards GPX
- 2) Une balise de départ GPX suivi des balises :
  - a) départ waypoint -> fin waypoint
  - b) départ route -> départ routepoint -> fin routepoint -> fin route
  - c) départ track -> départ tracksegment -> départ trackpoint -> fin trackpoint -> fin tracksegment -> fin track
- 3) La balise fin GPX.

Moindrement qu'il y a un problème lors de la séquence des balises, le programme va retourner un message d'erreur avec plusieurs informations. C'est sur ce message qu'un filtre regex est appliqué afin de détecter si le fichier est valide ou non.

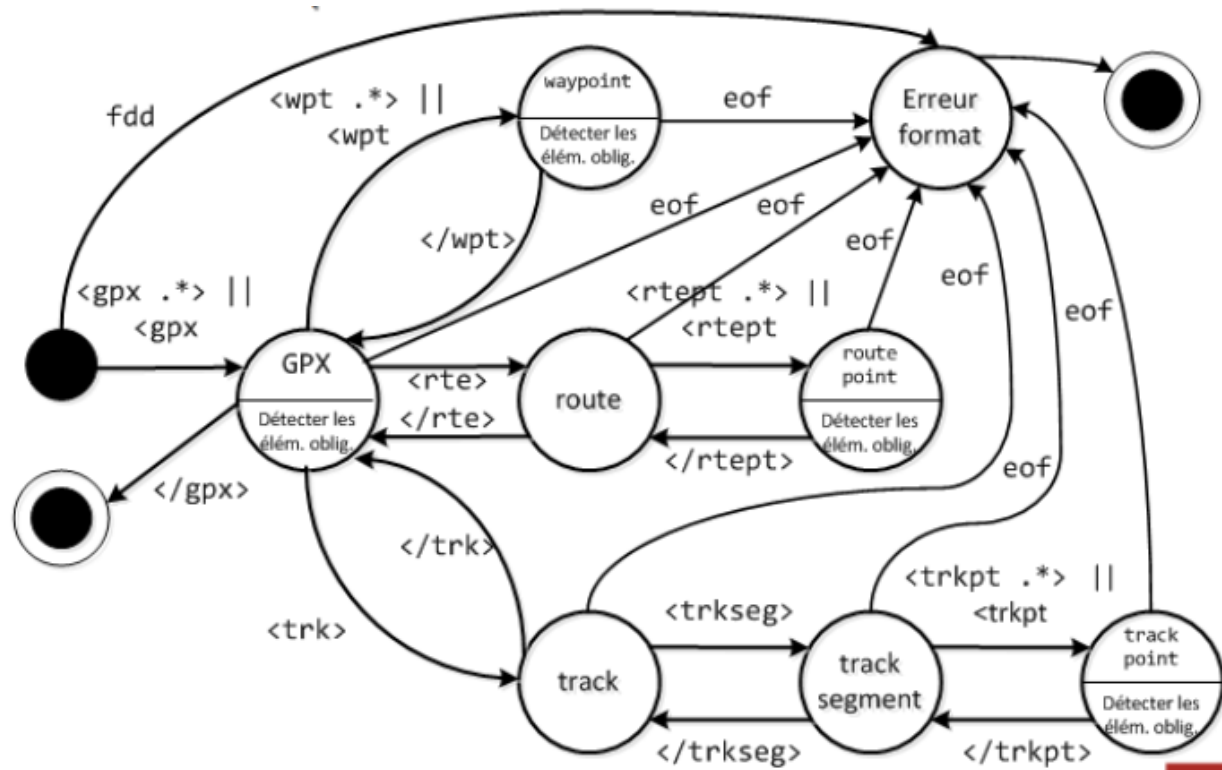


Figure 1- Diagramme machine d'état validation GPX

- **Création de la base de donnée :**

La création du fichier de base de données, "creation\_bdd.nawk", s'est fait facilement avec l'aide de monsieur Tony Wong. Son programme gawk, "valide\_gpx\_v3.nawk", qui détecte dans quelle état est rendu le programme bash, a été modifié pour qu'il puisse aussi analyser les trackpoints et créer une base de données. On a seulement apporter des modifications au début du code et dans la parti qui détecte l'état "trackpoint". Nous avons décider d'utiliser son programme pour s'assurer que le traitement de données soit fait dans la balise "trkpt".

Pour analyser chaque ligne et faire la base de données, on a commencé par créer une variable "POINT" qui compte la ligne présente. Quand le programme détectait un "trackpoint", on incrémentais la variable "POINT" et on exécutait notre code. Pour commencer, on effectuait une boucle "while" où on vérifiait, avec la fonction "match", si la ligne présente contenait les caractères "lat=", "lon=", "<ele>" ou "<time>" suivi par les chiffres dans l'ordre recherché. Avec "match" on a mis les caractères désirés dans une variable pour ensuite enlever les caractères non-voulu avec la fonction "gsub"; ceci nous empêchait de modifier le fichier GPX directement. À la fin de la boucle on utilisait la fonction "getline" pour changer la valeur de \$0 à le contenu de la prochaine ligne et ensuite on répétait la boucle juste

qu'à ce qu'on rencontrait les caractères "</trkpt>". Après la boucle on a utilisé la commande "printf" qui envoyait les variables, contenant les caractères recherchés, dans un fichier texte qu'on créait sur place en lui redirigeant les infos vers son entrée standard.

- **Validation de la base de donnée :**

Afin de valider le format des fichiers de types base de données, nous avons utilisé une approche de vérifications points par points à l'aide d'un programme gawk que nous avons nommé "validation\_bdd.nawk". Le programme considère le fichier étant valide sauf sous preuve du contraire. Afin de confirmer la validation du fichier, les points suivant sont analysés :

- En-tête : Lors de la lecture de la première ligne seulement, vérifier que l'en-tête est conforme. Comme il s'agit de la ligne d'en-tête, ignore les analyses de la position, latitude, longitude, élévation et temps.
- Position : Vérifie que la première donnée est une donnée de position valide contenant uniquement un entier positif plus grand que 0.
- Latitude : Vérifie que la deuxième donnée est une donnée de latitude valide contenant un nombre de type "float" ayant 6 nombres après le point.
- Longitude : Vérifie que la troisième donnée est une donnée de longitude valide contenant un nombre de type "float" ayant 6 nombres après le point.
- Élévation : Vérifie que la quatrième donnée est une donnée d'élévation valide de type "integer" ou "float".
- Temps : Vérifie que la cinquième donnée est une donnée de temps valide ayant le format YYYY-MM-JJThh:mm:ssZ où T et Z peuvent être minuscule.
- Contamination : Vérifie que la base de donnée n'est pas contaminée par d'autres caractères non-autorisés en comparant la ligne à tous les éléments valides. Si la ligne correspond adéquatement à ce qu'elle est supposée ressembler selon chaque élément validé, il confirme que la validité.

Une fois les vérifications terminées, le programme retourne "ECHEC" ou "VALIDE" dépendamment du résultat de l'analyse.

- **Calcul (temps, distance, vitesse moyenne) :**

Le calcul est fait selon le type donné dans la commande. L'option -c fournit le type de calcul; on a, en tout, 3 types : time (temps), distance (distance) et speed (vitesse moyenne). Le calcul dépend aussi des positions données avec l'option "-p". Cette option n'est pas obligatoire dans le calcul. Si elle n'existe pas alors le calcul va se faire du premier point jusqu'au dernier point de la base de donnée.

- 1) **Calcul du temps :**

Supposons que l'option "-p" existe avec "pos1" comme première position et "pos2" comme

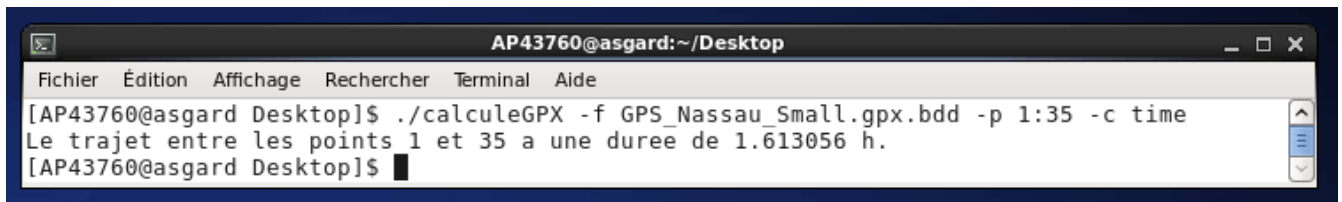
deuxième position. Ici le programme doit calculer la durée du trajet entre la position initiale et la position finale. Le calcul se fait à travers les étapes suivantes :

- extraire les temps du point initiale "pos1" et du point finale "pos2" de la colonne "time" de la base de donnée.
- formater le temps pour le transformer de la forme "YYYY-MM-DDThh:mm:ssZ" à la forme "YYYY MM DD hh mm ss"
- utiliser la fonction "mktime" de "gawk" pour transformer le temps en secondes
- faire la soustraction entre les deux temps
- afficher le résultat divisé par 3600 pour que l'unité soit en heures et non pas en secondes.

Voici les résultats de ce calcul pour le fichier de test "GPS\_Nassau\_Small.gpx.bdd" :

**`./calculeGPX -f GPS_Nassau_Small.gpx.bdd -p 1:35 -c time`**

Le programme fait la procédure de calcul de la durée du trajet entre les points 1 et 35.

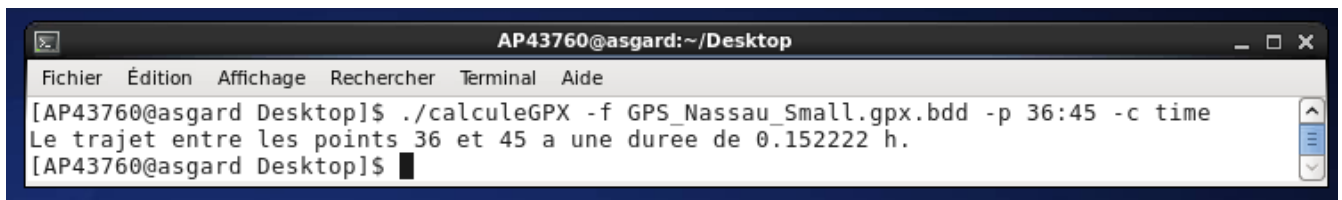


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculeGPX -f GPS_Nassau_Small.gpx.bdd -p 1:35 -c time
Le trajet entre les points 1 et 35 a une durée de 1.613056 h.
[AP43760@asgard Desktop]$
```

Figure 2- Durée du trajet entre les points 1 et 35

**`./calculeGPX -f GPS_Nassau_Small.gpx.bdd -p 36:45 -c time`**

Le programme fait la procédure de calcul de la durée du trajet entre les points 36 et 45.

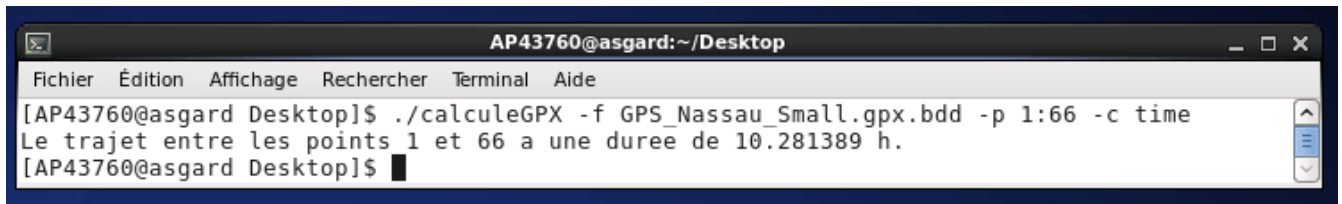


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculeGPX -f GPS_Nassau_Small.gpx.bdd -p 36:45 -c time
Le trajet entre les points 36 et 45 a une durée de 0.152222 h.
[AP43760@asgard Desktop]$
```

Figure 3- Durée du trajet entre les points 36 et 45

**`./calculeGPX -f GPS_Nassau_Small.gpx.bdd -p 1:66 -c time`**

Le programme fait la procédure de calcul de la durée du trajet entre les points 1 et 66.

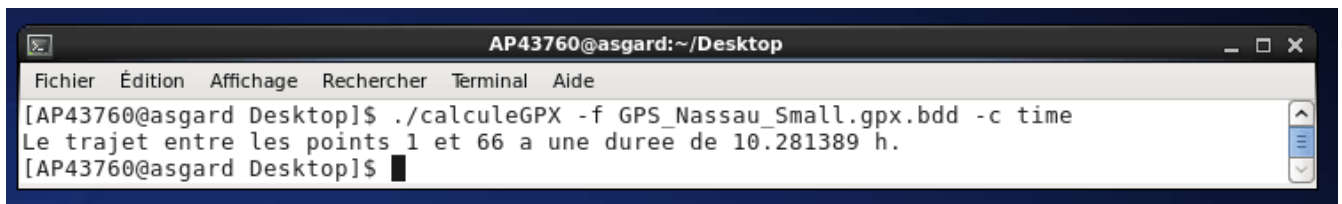


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 1:66 -c time
Le trajet entre les points 1 et 66 a une duree de 10.281389 h.
[AP43760@asgard Desktop]$
```

Figure 4- Durée du trajet entre les points 1 et 66

### **`./calculerGPX -f GPS_Nassau_Small.gpx.bdd -c time`**

Si l'option "-p" n'existe pas dans la commande, alors le programme calcule la durée du trajet du premier point (1) jusqu'au dernier point (66). On trouve le même résultat que celui des points entre 1 et 66.

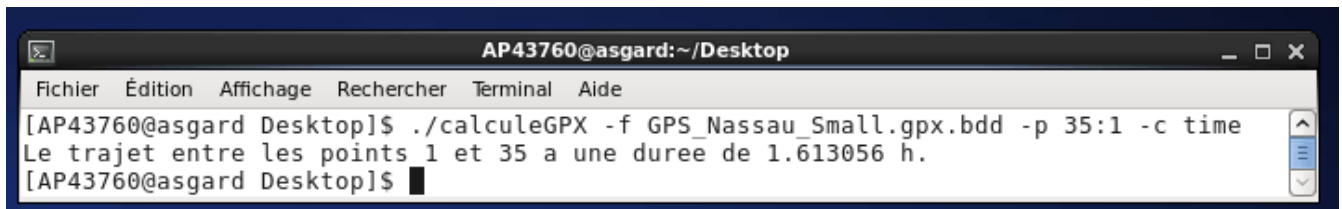


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculerGPX -f GPS_Nassau_Small.gpx.bdd -c time
Le trajet entre les points 1 et 66 a une duree de 10.281389 h.
[AP43760@asgard Desktop]$
```

Figure 5- Durée du trajet de tous les points (sans l'option -p)

### **`./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 35:1 -c time`**

Même si les positions données sont inversées, le programme les ordonne et fait le calcul. Ainsi, on a le même résultat que celui entre 1 et 35.



```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 35:1 -c time
Le trajet entre les points 1 et 35 a une duree de 1.613056 h.
[AP43760@asgard Desktop]$
```

Figure 6- Durée du trajet entre les points 35 et 1

## **2) Calcul de la distance :**

Le calcul de la distance entre deux points donnés est plus compliqué que celui du temps. Il faut faire 'addition des distances des arcs entre chaque paire de points appartenants à l'intervalle de positions. La distance entre une paire de points dépend de la latitude et longitude des deux points concernés. Ainsi, le programme effectue les étapes suivantes pour aboutir au résultat final :

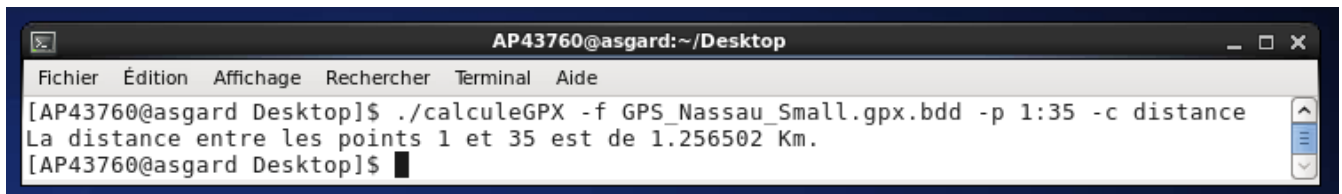
- a) extraire les latitudes et longitudes de chaque point entre l'intervalle [pos1, pos2] et les mémoriser dans un tableau

- b) faire le calcul des distances des segments grâce aux fonctions données dans l'énoncé du lab
- c) faire l'addition des distances des segments. La somme totale est la distance du trajet.

Voici les résultats de ce calcul pour le fichier de test "GPS\_Nassau\_Small.gpx.bdd" :

**./calculGPX -f GPS\_Nassau\_Small.gpx.bdd -p 1:35 -c distance**

Le programme effectue la procédure du calcul de la distance pour le trajet entre le point 1 et le point 35

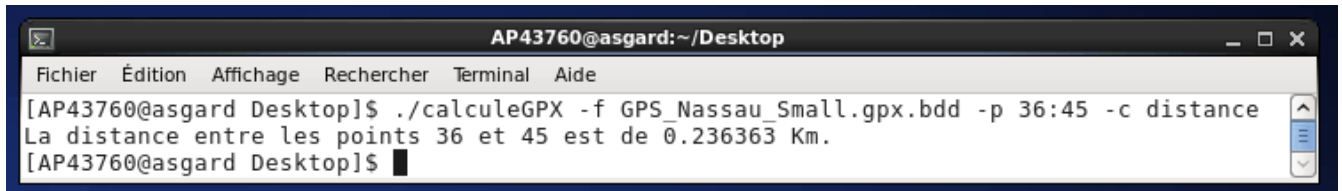


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculGPX -f GPS_Nassau_Small.gpx.bdd -p 1:35 -c distance
La distance entre les points 1 et 35 est de 1.256502 Km.
[AP43760@asgard Desktop]$
```

**Figure 7- Distance du trajet entre les points 1 et 35**

**./calculGPX -f GPS\_Nassau\_Small.gpx.bdd -p 36:45 -c distance**

Le programme effectue la procédure du calcul de la distance pour le trajet entre le point 36 et le point 45.

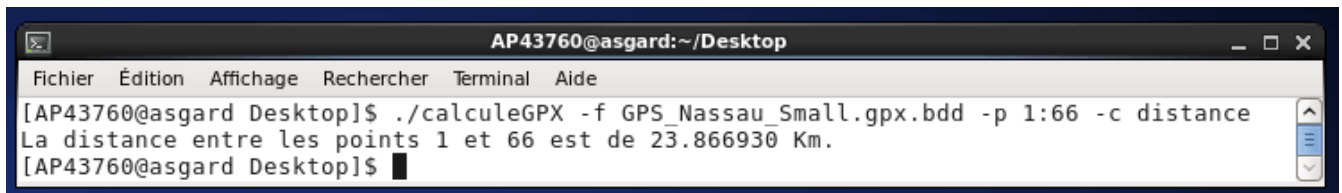


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculGPX -f GPS_Nassau_Small.gpx.bdd -p 36:45 -c distance
La distance entre les points 36 et 45 est de 0.236363 Km.
[AP43760@asgard Desktop]$
```

**Figure 8- Distance du trajet entre les points 36 et 35**

**./calculGPX -f GPS\_Nassau\_Small.gpx.bdd -p 1:66 -c distance**

Le programme effectue la procédure du calcul de la distance pour le trajet entre le point 1 et le point 66.

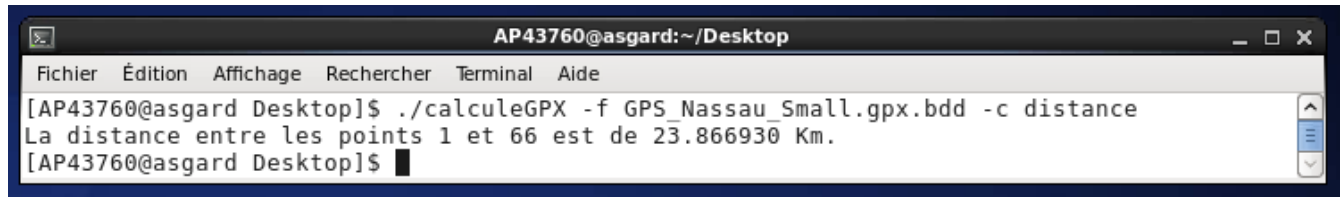


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculGPX -f GPS_Nassau_Small.gpx.bdd -p 1:66 -c distance
La distance entre les points 1 et 66 est de 23.866930 Km.
[AP43760@asgard Desktop]$
```

**Figure 9- Distance du trajet entre les points 1 et 66**

**`./calculerGPX -f GPS_Nassau_Small.gpx.bdd -c distance`**

Si l'option `-p` n'existe pas, le calcul de distance se fait pour tous les points de la base de donnée, c'est à dire du point 1 au point 66. On trouve ainsi le même résultat que le précédent.

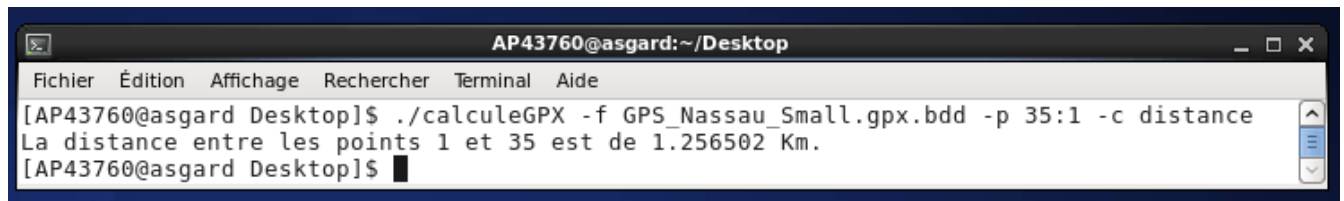


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculerGPX -f GPS_Nassau_Small.gpx.bdd -c distance
La distance entre les points 1 et 66 est de 23.866930 Km.
[AP43760@asgard Desktop]$
```

**Figure 10- Distance du trajet entre tous les points (sans l'option `-p`)**

**`./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 35:1 -c distance`**

Même si on inverse les positions, le programme les remet en ordre et fait le calcul de la distance. Ainsi, on trouve le même résultat que celui pour les points 1 et 35.



```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 35:1 -c distance
La distance entre les points 1 et 35 est de 1.256502 Km.
[AP43760@asgard Desktop]$
```

**Figure 11- Distance du trajet entre les points 35 et 1**

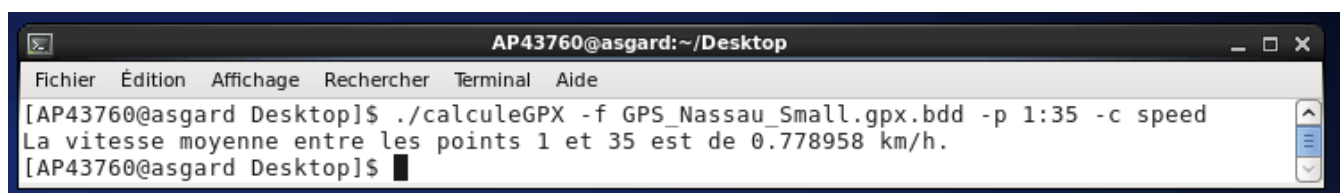
### **3) Calcul de la vitesse moyenne :**

Le programme doit calculer la vitesse moyenne dans le trajet entre les points donnés. La vitesse moyenne est égale à la distance entre ces deux points divisée par le temps du trajet. Ainsi, ce calcul va utiliser le code de calcul de la distance et celui du temps expliqués précédemment. Le résultat sera en Km/h.

Voici les résultats de ce calcul pour le fichier de test "GPS\_Nassau\_Small.gpx.bdd" :

**`./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 1:35 -c speed`**

Le calcul affiche la vitesse moyenne du trajet entre les points 1 et 35, ce qui est égale à la distance entre ces deux points divisée par le temps écoulé dans le trajet.



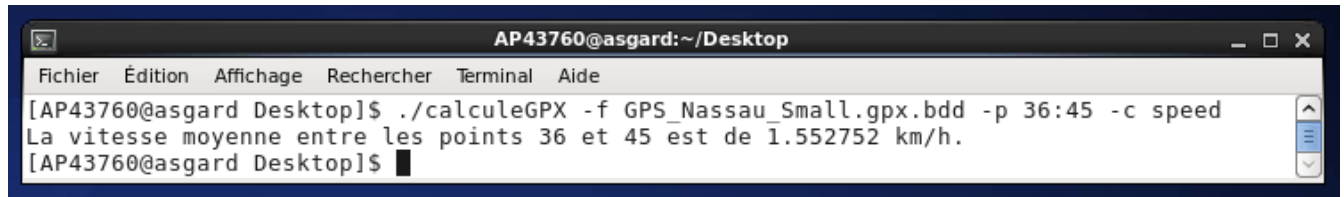
```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 1:35 -c speed
La vitesse moyenne entre les points 1 et 35 est de 0.778958 km/h.
[AP43760@asgard Desktop]$
```

**Figure 12- Vitesse moyenne du trajet entre les points 1 et 35**



**`./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 36:45 -c speed`**

Ici, on calcule la vitesse moyenne entre les points 36 et 45.

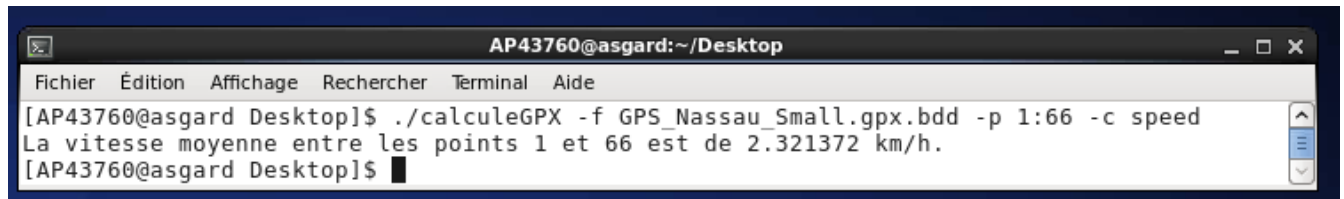


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 36:45 -c speed
La vitesse moyenne entre les points 36 et 45 est de 1.552752 km/h.
[AP43760@asgard Desktop]$
```

**Figure 13- Vitesse moyenne du trajet entre les points 36 et 45**

**`./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 1:66 -c speed`**

Dans cette commande, on demande de calculer la vitesse moyenne entre les points 1 et 66.

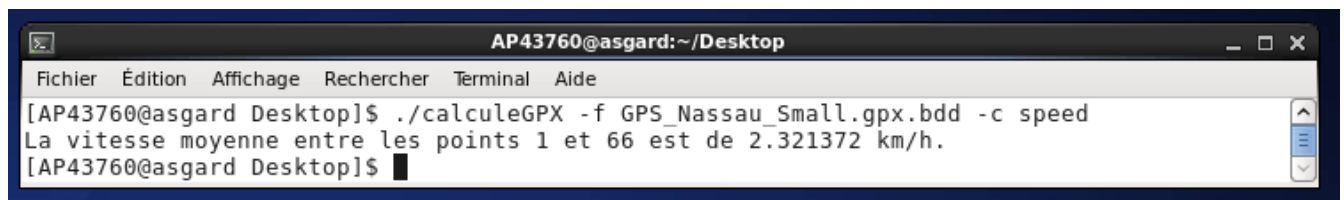


```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 1:66 -c speed
La vitesse moyenne entre les points 1 et 66 est de 2.321372 km/h.
[AP43760@asgard Desktop]$
```

**Figure 14- Vitesse moyenne du trajet entre les points 1 et 66**

**`./calculerGPX -f GPS_Nassau_Small.gpx.bdd -c speed`**

Comme dans les autres calculs, si l'option -p n'existe pas alors le calcul de la vitesse moyenne se fait du premier point jusqu'au dernier point de la base de donnée. On trouve ainsi le même résultat que celui pour les points 1 et 66.



```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculerGPX -f GPS_Nassau_Small.gpx.bdd -c speed
La vitesse moyenne entre les points 1 et 66 est de 2.321372 km/h.
[AP43760@asgard Desktop]$
```

**Figure 15- Vitesse moyenne du trajet entre tous les points (sans l'option -p)**

**`./calculerGPX -f GPS_Nassau_Small.gpx.bdd -p 35:1 -c speed`**

De même, si l'utilisateur fournit deux positions inversées dans l'ordre, le programme les re-ordonne et fait le calcul de la vitesse moyenne. On trouve le même résultat que celui des points 1 et 35.

```

AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculGPX -f GPS_Nassau_Small.gpx.bdd -p 35:1 -c speed
La vitesse moyenne entre les points 1 et 35 est de 0.778958 km/h.
[AP43760@asgard Desktop]$

```

Figure 16- Vitesse moyenne du trajet entre les points 35 et 1

### Gestion des erreurs dans le calcul

- ☐ Si les deux positions fournis sont égales alors afficher un message d'erreur et ne faire aucun calcul.

Ici, on a testé le calcul de la distance entre les points 25 et 25. Évidemment, un message d'erreur est affiché et aucun calcul n'est fait.

```

AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculGPX -f GPS_Nassau_Small.gpx.bdd -p 25:25 -c distance
Erreur: la premiere position ne doit pas etre egale a la deuxieme et la position 0 n'existe pas
[AP43760@asgard Desktop]$

```

Figure 17- Erreur: positions égales

- ☐ Si une des positions est égale à zéro alors afficher un message d'erreur et ne faire aucun calcul.

La première position dans cette commande est nulle. Le programme affiche un message d'erreur et ne fait aucun calcul.

```

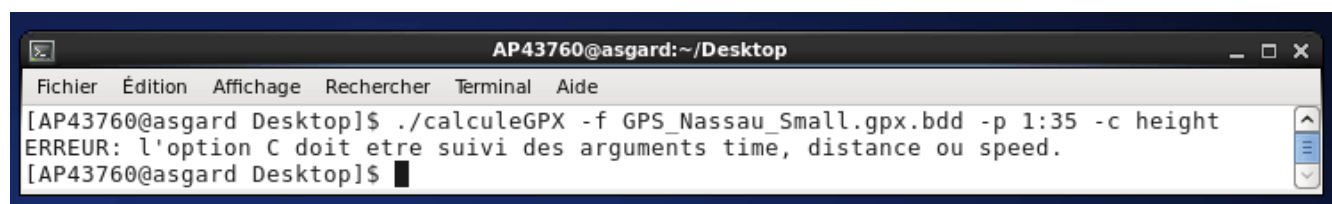
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculGPX -f GPS_Nassau_Small.gpx.bdd -p 0:25 -c distance
Erreur: la premiere position ne doit pas etre egale a la deuxieme et la position 0 n'existe pas
[AP43760@asgard Desktop]$

```

Figure 18- Erreur: position nulle

- ☐ Si le type de calcul n'est ni "time", ni "distance", ni "speed" alors afficher un message d'erreur.

Par exemple, le type de calcul dans cette commande est "height" (hauteur entre les deux positions). Le programme affiche un message d'erreur explicatif.



```
AP43760@asgard:~/Desktop
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[AP43760@asgard Desktop]$ ./calculGPX -f GPS_Nassau_Small.gpx.bdd -p 1:35 -c height
ERREUR: l'option C doit etre suivi des arguments time, distance ou speed.
[AP43760@asgard Desktop]$
```

**Figure 19- Erreur: type de calcul non reconnu**

## b) Identification des goulots d'étranglement potentiels

Afin d'analyser les goulots d'étranglements du programme, nous avons créé et utilisé le programme "mesure\_temps". Ce programme exécute X itérations le programmes et retourne la valeur en secondes calculé en nanosecondes grâce à la commande "date +%s%N".

Pour des fins pratiques, les temps mesurés sont basé sur 10 itérations du programmes.

Les plus gros goulots d'étranglements potentiels du programme "calculeGPX", incluant tous ses sous-programmes, sont :

- Le choix du fichier à traiter : Puisque le programme analyse toutes les données d'un fichier GPX ou base de données, le nombre de donnée à l'intérieur du fichier influence grandement tous le programme. Plus il y aura des données, plus le temps d'exécutions sera lent. Pour les même exécution nous avons mesurés, comme indiqué dans l'image qui suit, entre le fichier "GPS\_Nassau\_Small.gpx" (contenant 66 données) et "luzern.gpx" (contenant 32867 données) ou que le traitement de luzerne est de 100 à 160 fois plus lent à traiter .

```

TEMPS (~ /LAB_03) - gedit
Fichier  Édition  Affichage  Rechercher  Outils  Documents  Aide

TEMPS X
Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f GPS_Nassau_Small.gpx en date du 15/12/2017
Le temps ecoule pour effectuer 10 iterations du programme est 0.458372 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f GPS_Nassau_Small.gpx.bdd -a -e -c time en date du 15/12/2017
Le temps ecoule pour effectuer 10 iterations du programme est 0.296658 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f GPS_Nassau_Small.gpx.bdd -a -e -c distance en date du 15/12/2017
Le temps ecoule pour effectuer 10 iterations du programme est 0.299862 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f GPS_Nassau_Small.gpx.bdd -a -e -c speed en date du 15/12/2017
Le temps ecoule pour effectuer 10 iterations du programme est 0.30713 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx en date du 15/12/2017 a 18:30:01
Le temps ecoule pour effectuer 10 iterations du programme est 72.7804 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c time -a -e en date du 15/12/2017 a 18:30:01
Le temps ecoule pour effectuer 10 iterations du programme est 28.7123 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c distance -a -e en date du 15/12/2017 a 18:30:01
Le temps ecoule pour effectuer 10 iterations du programme est 31.1459 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c speed -a -e en date du 15/12/2017 a 18:30:01
Le temps ecoule pour effectuer 10 iterations du programme est 31.2514 secondes.
  
```

**Figure 20- Analyse des temps d'exécutions selon la taille du fichier**

- La validation du fichier : Afin de valider le fichier, le programme parcourt 2 fois le fichier. Une première fois pour vérifier s'il s'agit d'un fichier de type GPX, puis pour vérifier s'il s'agit d'un fichier de type base de données.

- La création de la base de données : La création de la base de données demande à ce qu'on valide au préalable qu'il s'agit bien d'un fichier de type GPX valide puis ensuite traite le fichier au complet. Ce qui crée un délais considérable. On voit dans l'analyse suivante que le mode création de la base de données prend de 2 à 3 fois plus de temps que les modes calculs.

```

TEMPS (~/.LAB_03) - gedit
Fichier Édition Affichage Rechercher Outils Documents Aide

TEMPS X
Lancement de mesures de temps pour 10 iterations de ./calculGPX -f GPS_Nassau_Small.gpx en date du 15/12/2017 a
Le temps ecoule pour effectuer 10 iterations du programme est 0.458372 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculGPX -f GPS_Nassau_Small.gpx.bdd -a -e -c time en da
Le temps ecoule pour effectuer 10 iterations du programme est 0.296658 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculGPX -f GPS_Nassau_Small.gpx.bdd -a -e -c distance e
Le temps ecoule pour effectuer 10 iterations du programme est 0.299862 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculGPX -f GPS_Nassau_Small.gpx.bdd -a -e -c speed en d
Le temps ecoule pour effectuer 10 iterations du programme est 0.30713 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculGPX -f luzern.gpx en date du 15/12/2017 a 18:30:01
Le temps ecoule pour effectuer 10 iterations du programme est 72.7804 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculGPX -f luzern.gpx.bdd -c time -a -e en date du 15/1
Le temps ecoule pour effectuer 10 iterations du programme est 28.7123 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculGPX -f luzern.gpx.bdd -c distance -a -e en date du
Le temps ecoule pour effectuer 10 iterations du programme est 31.1459 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculGPX -f luzern.gpx.bdd -c speed -a -e en date du 15/
Le temps ecoule pour effectuer 10 iterations du programme est 31.2514 secondes.

```

**Figure 21- Analyse des temps d'exécutions selon la création ou traitement de la base de données**

- L'utilisation des options : Chaque options ajoutent un délais au temps d'exécution excepté l'option "-p". Puisque l'option "-p" définie de quel à quel point le traitement aura lieu, il peut donc réduire considérablement le temps d'exécution selon la plage sélectionnée. Parmi les autres options, l'option "-a" est la plus gourmande puisqu'elle demande d'afficher dans la sortie standard toutes les coordonnées. On voit dans l'analyse suivant que l'utilisation de simplement l'option "-a" (27.7535s) prend plus de temps d'exécution que l'option "-c speed" combiné avec l'option "-e" (21.9559s). Concernant l'option "-c", selon ses 3 arguments possibles, l'argument "speed" demande le plus de temps à exécuter, suivi de près par l'argument "distance" et ayant l'argument "time" étant beaucoup plus rapide à exécuter comparé aux 2 autres. Finalement l'option "-e" rajoute un petit temps d'exécution similaire à l'option "-c time".

-a	-e	-c time	-c distance	-c speed
0.67432 s	0.06453 s	0.06173 s	0.23704	0.27604

**Figure 22- Temps approximatif ajouté par option pour le traitement de luzern.gpx.bdd**  
 (Les résultats sont calculé en soustrayant 2 temps du des données TEMPS. Par exemple -a est calculant la différence entre "calculGPX -f [...] -c time -a" et "calculGPX -f [...] -a")



```
TEMPS (~ /LAB_03) - gedit
Fichier  Édition  Affichage  Rechercher  Outils  Documents  Aide

TEMPS X

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f GPS_Nassau_Small.gpx en date du 15/12/2017 a 18:29:17
Le temps ecoule pour effectuer 10 iterations du programme est 0.458372 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f GPS_Nassau_Small.gpx.bdd -a -e -c time en date du 15/12/2017 a 18:29:34
Le temps ecoule pour effectuer 10 iterations du programme est 0.296658 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f GPS_Nassau_Small.gpx.bdd -a -e -c distance en date du 15/12/2017 a 18:29:37
Le temps ecoule pour effectuer 10 iterations du programme est 0.299862 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f GPS_Nassau_Small.gpx.bdd -a -e -c speed en date du 15/12/2017 a 18:29:41
Le temps ecoule pour effectuer 10 iterations du programme est 0.30713 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx en date du 15/12/2017 a 18:30:01
Le temps ecoule pour effectuer 10 iterations du programme est 72.7804 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c time -a -e en date du 15/12/2017 a 18:34:41
Le temps ecoule pour effectuer 10 iterations du programme est 28.7123 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c distance -a -e en date du 15/12/2017 a 18:36:09
Le temps ecoule pour effectuer 10 iterations du programme est 31.1459 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c speed -a -e en date du 15/12/2017 a 18:36:56
Le temps ecoule pour effectuer 10 iterations du programme est 31.2514 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c time -a en date du 15/12/2017 a 18:38:42
Le temps ecoule pour effectuer 10 iterations du programme est 28.3708 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c distance -a en date du 15/12/2017 a 18:39:36
Le temps ecoule pour effectuer 10 iterations du programme est 30.1239 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c speed -a en date du 15/12/2017 a 18:40:16
Le temps ecoule pour effectuer 10 iterations du programme est 30.5139 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c time -e en date du 15/12/2017 a 18:41:10
Le temps ecoule pour effectuer 10 iterations du programme est 21.9559 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c distance -e en date du 15/12/2017 a 18:41:56
Le temps ecoule pour effectuer 10 iterations du programme est 23.9143 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c speed -e en date du 15/12/2017 a 18:42:32
Le temps ecoule pour effectuer 10 iterations du programme est 24.0677 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c time en date du 15/12/2017 a 18:43:32
Le temps ecoule pour effectuer 10 iterations du programme est 21.2872 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c distance en date du 15/12/2017 a 18:44:11
Le temps ecoule pour effectuer 10 iterations du programme est 23.4887 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -c speed en date du 15/12/2017 a 18:44:42
Le temps ecoule pour effectuer 10 iterations du programme est 23.6546 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -a -e en date du 15/12/2017 a 18:45:19
Le temps ecoule pour effectuer 10 iterations du programme est 28.3988 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -a en date du 15/12/2017 a 18:45:56
Le temps ecoule pour effectuer 10 iterations du programme est 27.7535 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -e en date du 15/12/2017 a 18:46:38
Le temps ecoule pour effectuer 10 iterations du programme est 21.6556 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -a -e -c time -p 1:8 en date du 15/12/2017 a 18:48:36
Le temps ecoule pour effectuer 10 iterations du programme est 22.223 secondes.

Lancement de mesures de temps pour 10 iterations de ./calculeGPX -f luzern.gpx.bdd -a -e -c time -p 1:15000 en date du 15/12/2017 a 18:49:07
Le temps ecoule pour effectuer 10 iterations du programme est 25.0805 secondes.
```

### c) Proposition des contournements

Afin de réduire le temps d'exécution du programme, nous pouvons appliquer quelques améliorations tel que :

- Simplifier les validations : Au lieu d'analyser 2 fois s'il s'agit d'un fichier GPX ou base de donnée, il est possible d'analyser le nom du fichier pour savoir s'il est plus probable qu'il soit un fichier gpx ou base de donnée puis valider conditionnellement afin de ne pas faire valider le format GPX s'il a déjà été validé base de donnée. Ou vice versa.
- Imbriquer la création de la base de donnée dans la validation : Lors de la création de donnée, le programme parcourt 2 fois le fichier GPX, 1 fois pour le valider et 1 fois pour traiter les données. Pour accélérer le traitement on pourrait effectuer la validation ainsi que la création de la base de donnée en même temps. Soit en gardant les valeurs dans un tableau, ce qui consommerait énormément de mémoire, ou simplement détruire le fichier créé lorsqu'on détecte que le fichier n'est pas un fichier GPX valide.
- Traitement de la base de donnée pour les calculs : Lorsqu'il y a plusieurs options sélectionnées en même temps parmi -c -a et -e, le fichier de base de donnée est traité autant de fois qu'il y a d'options. Insérer le programme des options -a et -e dans le programme gawk afin de parcourir juste une fois la base de donnée aiderait à réduire le temps d'exécution.

## 3- Résultats et conclusion :

À la fin de ce laboratoire, nous avons réussi à produire un programme "calculeGPX" qui était capable de prendre un fichier GPX et créer une base de données à partir des valeurs de latitude, longitude, élévation et temps contenu dans le fichier. Si on lui donnait un fichier de base données, il pouvait calculer la distance totale parcourue, la vitesse moyenne, et aussi la différence de temps entre les points sélectionnés à l'aide des options qu'on lui a intégrées. Le programme était aussi capable de détecter si la commande était mal écrite, le fichier GPX était invalide ou que la base de données avait de l'information erronée, entre autres. En conclusion, nous avons appris comment programmer plus efficacement en utilisant les filtres AWK et les notions avancées de "bash" et nous avons aussi vu comment qu'on pouvait améliorer le programme pour qu'il soit plus rapide et prend moins de mémoire.

## 4- Code source :

En annexe vous trouverez dans un fichier .zip le code source sous forme de 5 fichier sous format fichier linux. Ses 5 fichiers sont;

- calculGPX (programme bash principal)
- validation\_gpx.nawk (sous-programme gawk pour calculGPX)
- validation\_bdd.nawk (sous-programme gawk pour calculGPX)
- creation\_bdd.nawk (sous-programme gawk pour calculGPX)
- mesure\_temps (programme bash utilisé pour la mesure des temps d'exécution)