Projet : Architecture système

> Retrouvez la dernière mise à jour du projet ici : [https://github.com/Anthony-Jhoiro/Archi-Sys_sophia]

Cette année comme projet d'architecture système nous avons dû créer un programme capable de créer un démon et de communiquer avec lui. Nous avons choisi de façon arbitrair d'appeler le démon Sophia. Sophia est designée pour executer des commandes simples sans consommer beaucoup de resources. Elle est de plus facilement extensible et peut donc servir de base à une application plus grand publique.

Technologies

Nous avons utilisé des forks, des pipes et des fifo. Les forks nous ont servi à séparrer le processus principal, nous permettant ainsi de créer le démon et de séparer l'écoute de nouveaux messages en trois processus, un listener, un timer et le processus principal (voir Listener). Une FIFO est utilisé pour communiquer avec le démon et un pipe est utilisé pour transmettre le message dans le listener.

Structure de fichier

```
.
├── deamon
│   ├── deamon.c            # Fichier source principal pour le démon
│   ├── deamon.h            # Header pour les fonctions du démon
│   └── Makefile            # Makefile pour compiler le démon
├── invoker
│   ├── help.txt            # Texte à afficher avec la commande --help
│   ├── invoker.c           # Fichier source pricipal pour l'invocateur
│   ├── invoker.h           # Header pour l'invocateur
│   └── Makefile            # Makefile pour compiler l'invocateur
├── rapport.md              # Version Markdown de ce rapport
├── rapport.pdf             # Version PDF de ce rapport
├── README.md               # README du projet
└── tools
    ├── constants.h         # Constantes du projets
    ├── fifo_tools.c        # Sources pour les outils sur les fifos
    ├── log_tools.c         # Sources pour les outils sur les logs
    ├── Makefile            # Makefile pour les tools
    ├── tools.c             # Sources principale pour les outils
    └── tools.h             # Header pour les outils
```

Le dossier tools contient tout le code succeptible d'être utilisé par les 2 programmes.

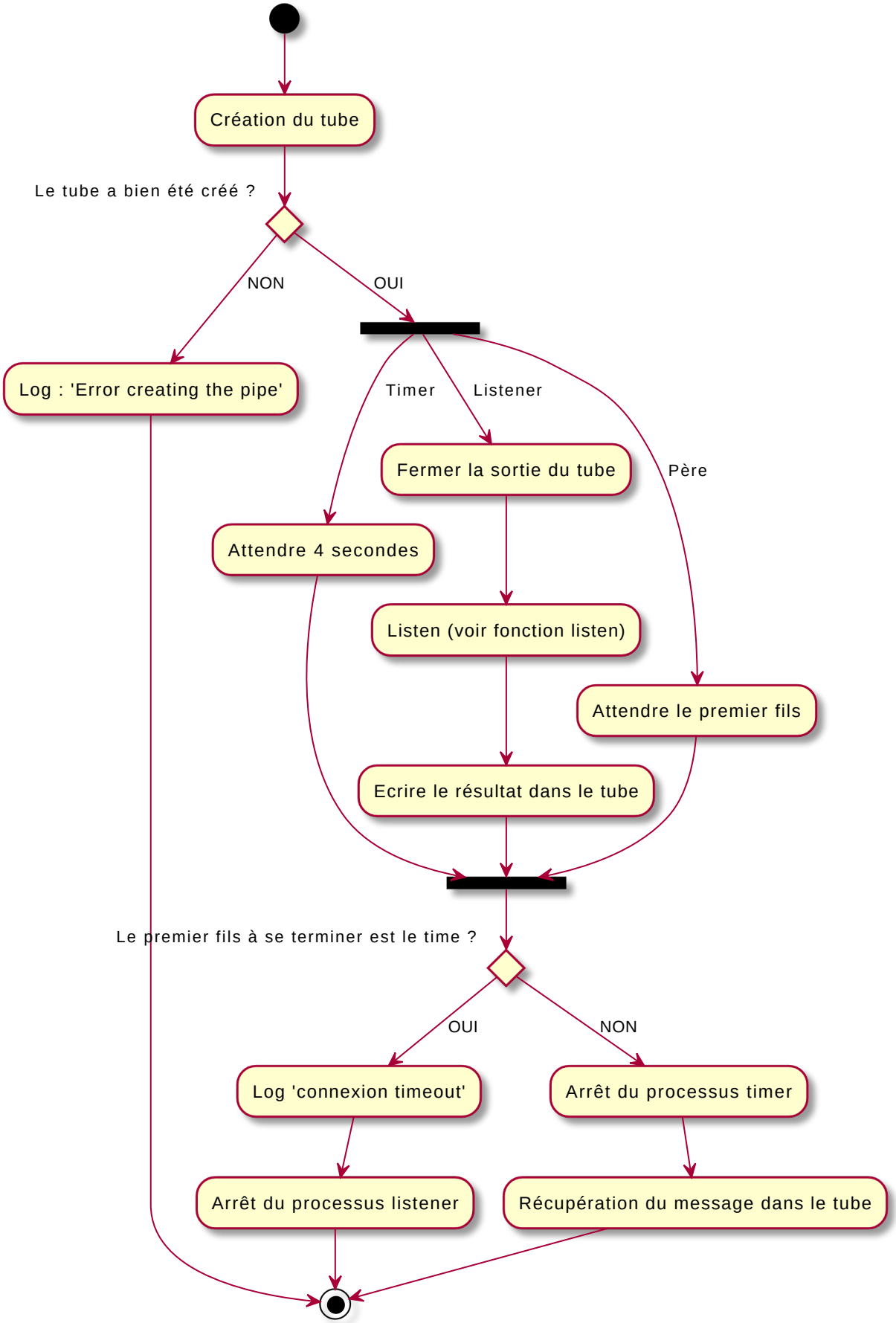
Fonctionnement principale

L'utilisateur va directement interagir avec le programme invocateur. Il va pouvoir executer différentes actions pour controller le démon ainsi qu'executer diverses actions sur le temps.

La communications avec le démon se fait par une fifo dont le nom est définit dans `tools/constants.h`.

Listener

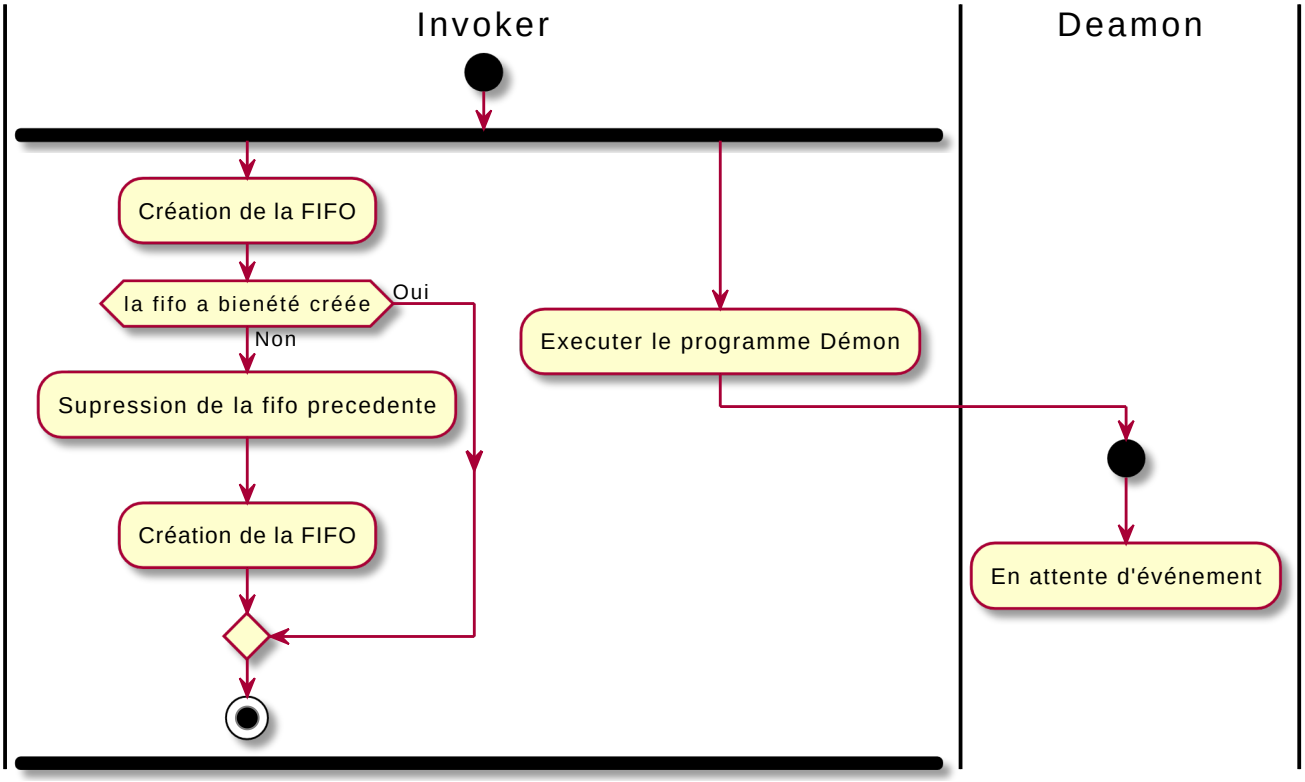
Quand le programme invocateur est en écoute, un timer est lancé en même temps. Il permet de renvoyer une erreur dans le cas où la fifo existe mais que le démon ne répond pas.


Création du tube
Le tube a bien été créé ?
NON
OUI
Log : 'Error creating the pipe'
Timer
Listener
Père
Fermer la sortie du tube
Attendre 4 secondes
Listen (voir fonction listen)
Attendre le premier fils
Ecrire le résultat dans le tube
Le premier fils à se terminer est le time ?
OUI
NON
Log 'connexion timeout'
Arrêt du processus timer
Arrêt du processus listener
Récupération du message dans le tube

Commandes de l'invocateur

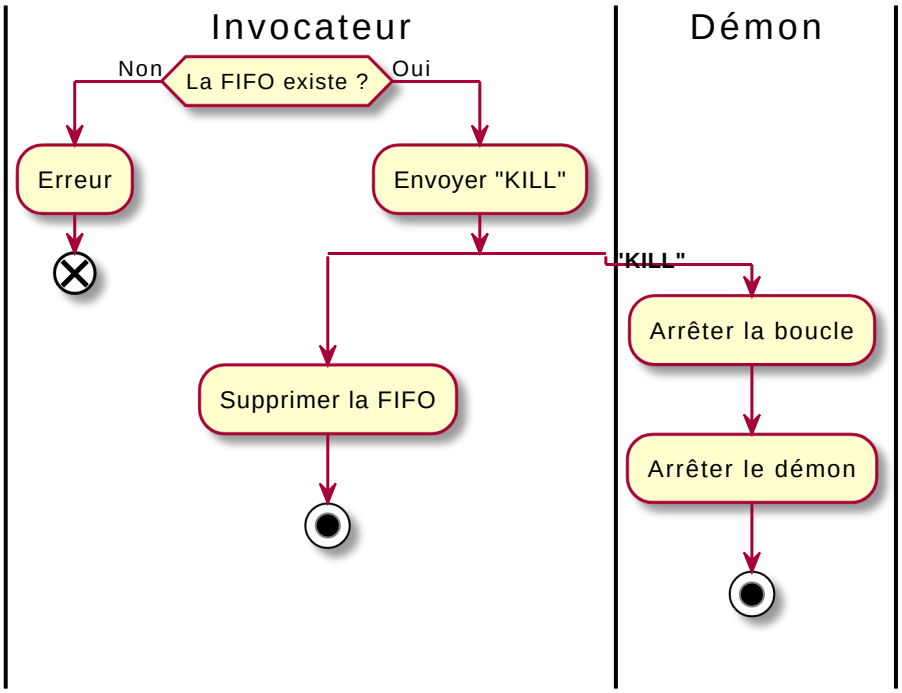
Commande `--start`

La commande `--start` permet de créer un démon. Pour ce faire, elle commence par créer un processus fils qui executera le programme démon. Puis, dans le processus père, on crée la FIFO. En cas d'échec dans la création de la FIFO, on essaie de supprimer une éventuelle FIFO existante avec la fonction `unlink()` puis on tente à nouveau de créer la FIFO, ci il y a un nouvel échec, on affiche l'erreur à l'utilisateur. Le processus père n'attend pas le fils afin que le démon puisse de nouveau écouter le programme invocateur.



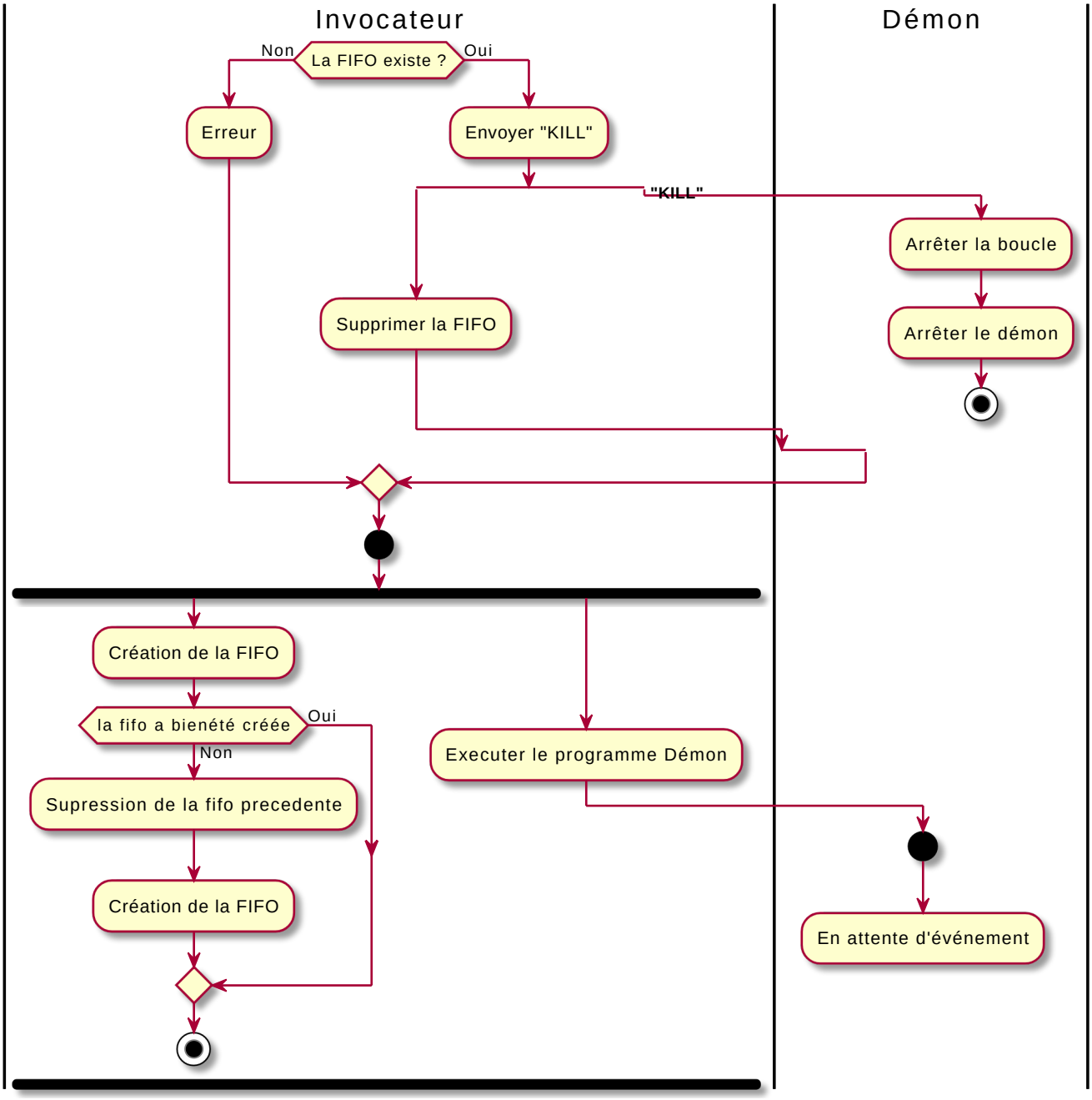
Commande `--stop`

La commande `--stop` permet d'arrêter un démon. Pour commencer, elle vérifie que la FIFO existe puis envoie *KILL* au démon avant de supprimer la fifo avec la fonction `unlink()`. Elle échoue si la FIFO n'existe pas. Il n'y a pas de vérification pour être sûr que le démon est bien arrêté, on peut utiliser la commande `--state`.



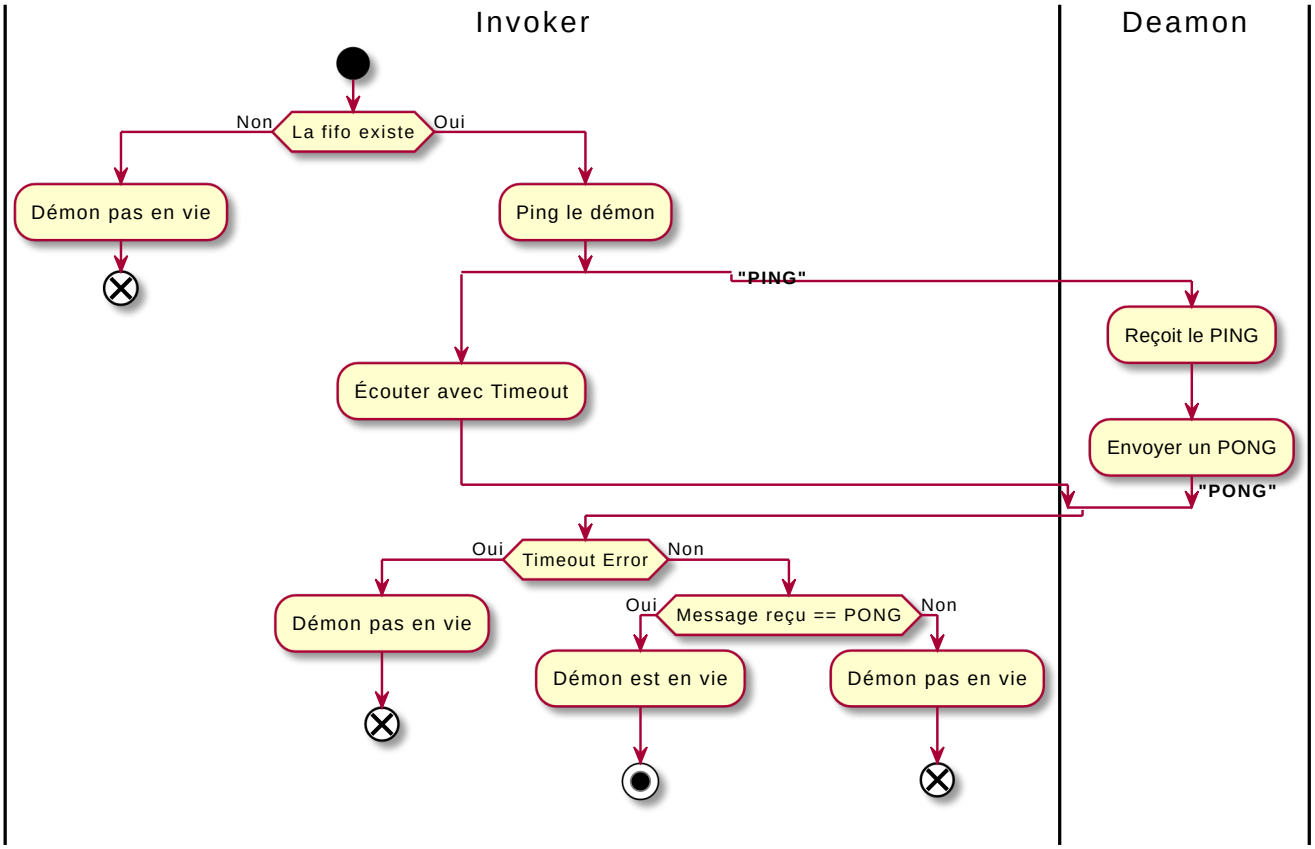
Commande `--restart`

La commande `--restart` permet de relancer un démon. Pour ce faire, on lance la fonction de la commande `--stop` puis cette de la commande `--start`. La première fonction n'est pas blocante donc si le démon n'était pas actif, il sera simplement créé.



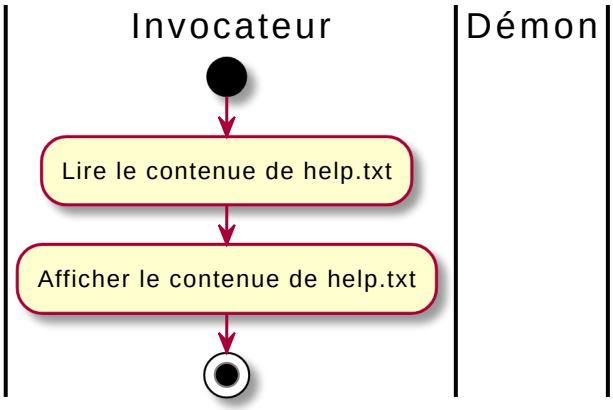
Commande `--state`

La commande `--state` permet de savoir si un démon est actif ou non. Elle commence par vérifier si la FIFO existe puis elle envoie un *PING* au démon par cette FIFO. En recevant un *PING*, le démon va renvoyer un *PONG*, il y a un timeout de 200 microsecondes pour permettre à l'invocateur de passer en écoute. Il aurait été possible de mettre l'invocateur en écoute avec un thread ou un fork mais après des tests, nous nous sommes rendu compte que c'était encore plus long. Une fois le *PONG* reçu, l'invocateur informe l'utilisateur que le démon est actif.



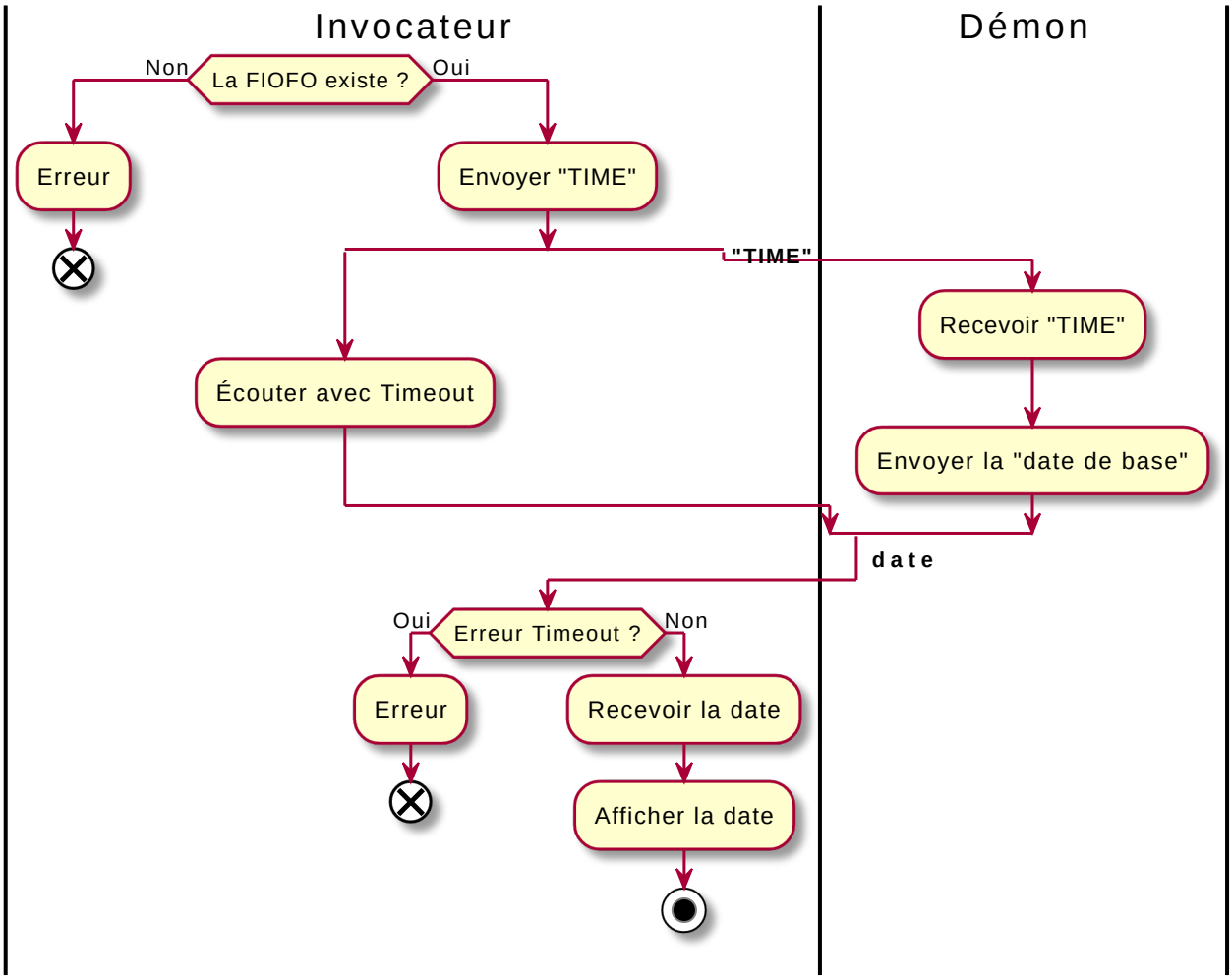
Commande `--help`

La commande `--help` sert à afficher l'aide pour utiliser l'invocateur. Elle affiche simplement le contenue du fichier `help.txt`situé dans le dossier contenant les sources de l'invocateur.



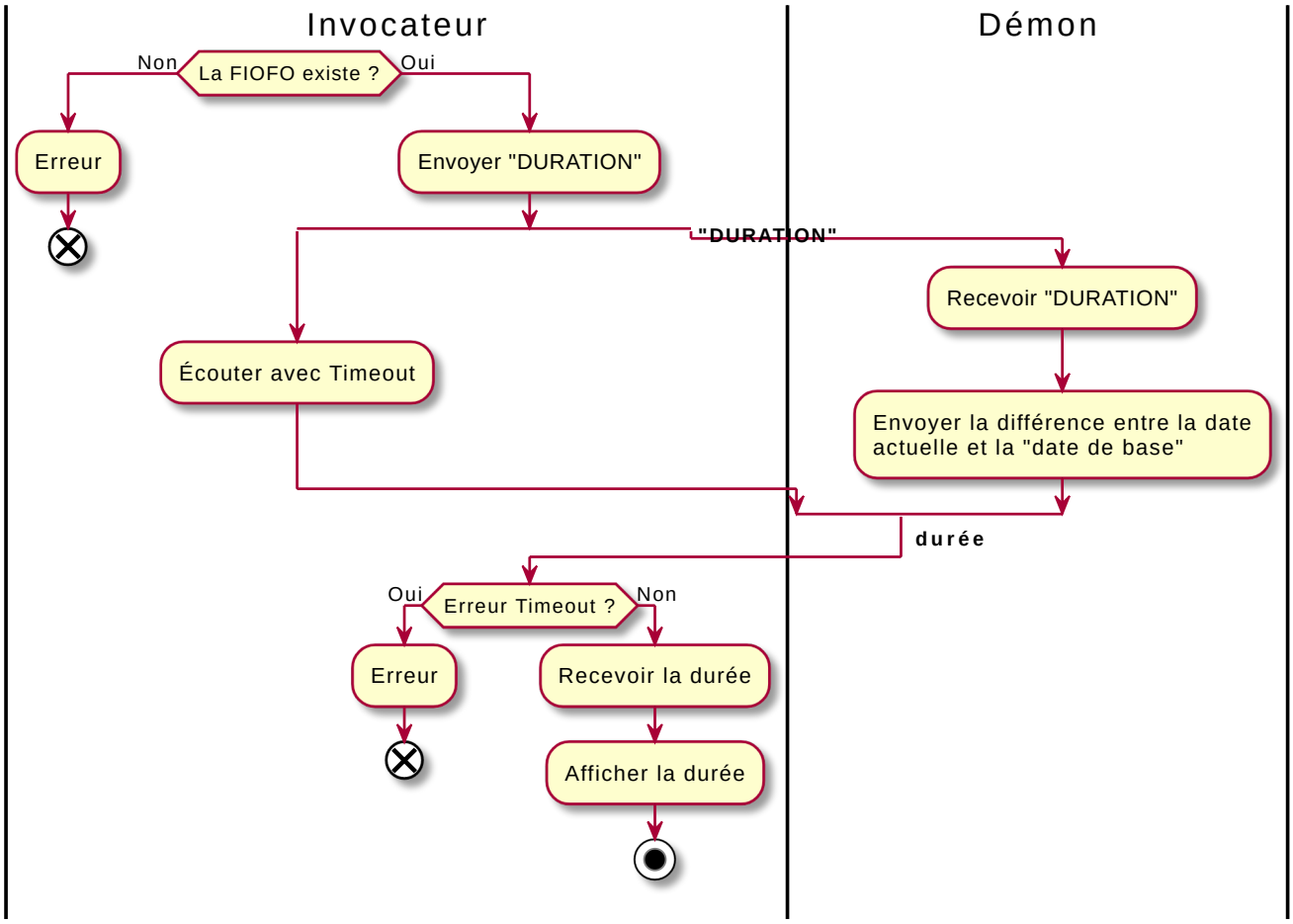
Commande `--date`

La commande `--date` sert à récupérer la date depuis l'invocateur par lé démon. Pour ce faire, le programme invocateur vérifie que la FIFO existe puis envoie *TIME* au démon. Quand le démon reçoit *TIME*, il renvoie la date à l'invocateur par la FIFO. L'invocateur affiche ensuite la date envoyée par le démon.



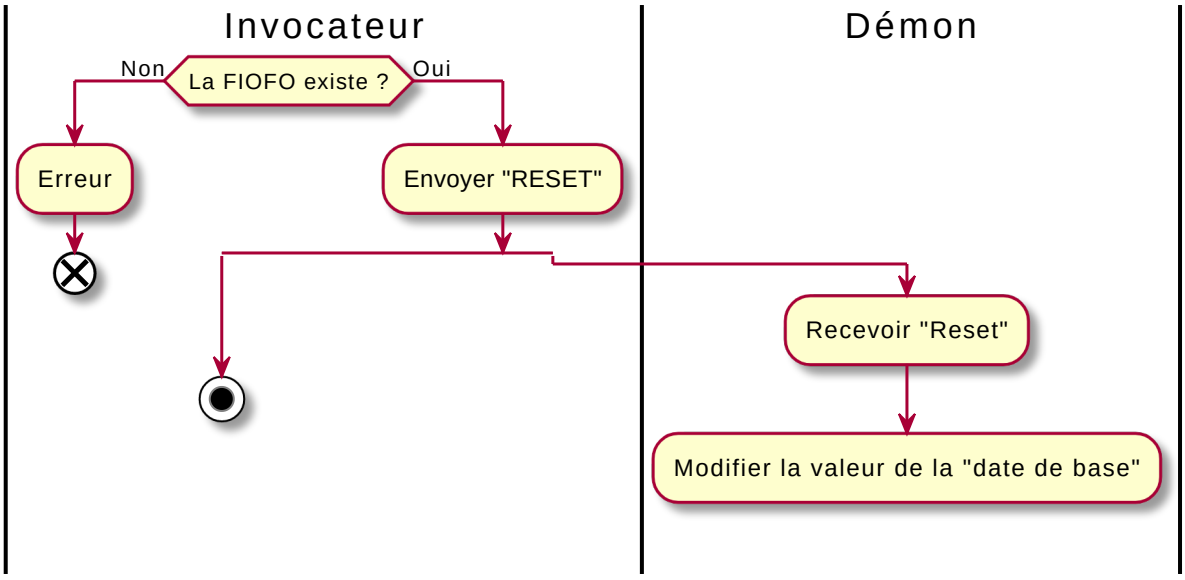
Commande `--duration`

La commande `--duration` sert à obtenir la durée depuis laquelle le démon est en vie ou alors la durée depuis le dernier "reset" (voir commande `--reset`). Pour ce faire, le programme invocateur vérifie que la FIFO existe puis envoie *DURATION* au démon. Quand le démon reçoit *DURATION*, il renvoie la différence entre la "date de base" et la date actuelle à l'invocateur par la FIFO. Cette différence est exprimée en seconde. L'invocateur affiche ensuite la durée envoyée par le démon.



Commande `--reset`

La commande `--reset` permet de modifier la "date de base" du démon. Pour ce faire, le programme invocateur vérifie que la FIFO existe puis envoie *RESET* au démon. Quand le démon reçoit *RESET*, il modifie sa "date de base" pour qu'elle corresponde à la date actuelle.



Partage des tâches

Anthony

- fonctions de base de l'invocateur
  - `--start`
  - `--stop`
  - `--restart`
  - `--state`

Kilian

- Le démon
- La gestion du temps
  - `--date`
  - `--duration`
  - `--reset`

Commun

- Le rapport avec les diagrammes
- Les fonctions pour lire et écrire dans la fifo