



UNIVERSITÉ CATHOLIQUE DE LILLE
L2 SCIENCES DU NUMÉRIQUES

Architecture Système

Gestionnaire de tâches différées

Charles Iacopino - Guillemet Steven

2023 - 2024

1.	Introduction.....	3
1.1	Contexte	3
1.2	Objectifs	3
1.3	L'équipe et la méthode de travail.....	4
1.4	Structure du rapport.....	4
2.	Implémentation et Justification des Choix	5
3.	Discussion	6
3.1	Défi et Solution	6
3.2	Défi du Répertoire Courant :.....	6
3.3	Redirections Complexes :.....	7
3.4	Répétition Infinie et Gestion des Signaux :.....	7
3.5	Mode Ligne de Commande et Options Variées :	7
3.6	Limites du Projet	7
4.	Conclusion et Perspectives	9
5.	Annexes	10

1. Introduction

1.1 Contexte

Dans le cadre de notre cours sur l'architecture système, nous avons été chargés de développer un gestionnaire de tâches différées. Ce projet consiste à créer un planificateur de tâches en C, capable d'exécuter des commandes Bash de manière différée dans le temps. Les étapes clés de ce projet incluent :

- La mise en place d'un système pour planifier et exécuter des commandes à des intervalles spécifiés.
- L'intégration d'un mécanisme pour gérer le nombre d'exécutions de chaque commande.
- La capacité à maintenir une exécution précise en évitant la dérive temporelle.

1.2 Objectifs

Le projet débute par une version minimale répondant à des besoins essentiels, pour ensuite évoluer vers une version finale plus complexe.

La version initiale se concentre sur :

- La réception et l'exécution de commandes via les options de ligne de commande.
- La prise en charge d'un délai entre chaque exécution et du nombre d'itérations de la commande.
- L'utilisation de fonctions pour prévenir la dérive temporelle.

La version finale :

- Mode ligne de commandes
- Commande dans le répertoire courant
- Répétition infinie

1.3 L'équipe et la méthode de travail

Notre équipe, formée de Charles Iacopino et Guillemet Steven. Contrairement à notre précédent projet en algorithmique avancée, l'ampleur de celui-ci ne justifiait pas l'utilisation de GitHub. Discord s'est avéré suffisant pour planifier nos sessions de travail communes, nous permettant d'organiser efficacement plusieurs rencontres hebdomadaires.

Dès le début, nous avons défini et distribué les responsabilités : Steven a pris l'initiative sur la version minimale, cherchant à approfondir ses compétences, bien que des défis l'aient conduit à requérir l'appui de Charles pour finaliser cette étape. Dans l'élaboration de la version finale, Charles a pris en charge l'implémentation du mode ligne de commande, tandis que Steven s'est concentré sur la fonctionnalité de répétition infinie et l'exécution de commandes dans le répertoire courant, bénéficiant de l'assistance de Charles pour surmonter certains défis techniques.

1.4 Structure du rapport

- Introduction au projet et aux objectifs.
- Implémentation et Justification des Choix
- Discussion
- Conclusion et perspectives

2. Implémentation et Justification des Choix

Dans la version minimale du gestionnaire de tâches, nous avons veillé à fournir les fonctionnalités essentielles pour une expérience utilisateur efficace et sécurisée.

- **Exécution de commandes avec `<unistd.h>`** : Nous utilisons `<unistd.h>` pour son accès direct aux appels système Unix/Linux, notamment `fork()` et `exec()`, permettant de démarrer des processus enfants et d'exécuter les commandes Bash spécifiées par l'utilisateur.
- **Gestion des Entrées/Sorties avec `<stdio.h>` et `<stdlib.h>`** : L'utilisation de `<stdio.h>` pour les opérations d'entrée/sortie standard et de `<stdlib.h>` pour l'allocation dynamique de mémoire a été cruciale pour gérer les entrées utilisateur et formater les sorties du programme.

Pour une version finale, nous avons fait le choix d'implémenter ces bibliothèques :

- **Gestion des options avec `<getopt.h>`** : Nous utilisons la bibliothèque `<getopt.h>` pour analyser les options de ligne de commande et configurer les tâches (intervalle d'exécution, nombre d'itérations, mode d'exécution infinie, ...).
- **Stabilité et sécurité avec `<limits.h>`** : Nous avons recours à `<limits.h>` pour gérer les limites du système et prévenir les dépassements et comportements indéfinis, garantissant ainsi une exécution sûre et stable des commandes planifiées.
- **Répétition infinie avec `<signal.h>`** : Grâce à la gestion des signaux avec `<signal.h>`, nous permettons une interruption propre du programme en mode d'exécution infinie en interceptant le signal SIGINT.

3. Discussion

3.1 Défi et Solution

La mise en place de la version minimale du gestionnaire de tâches ne présentait pas de difficultés majeures. L'utilisation de bibliothèques C standard telles que `<unistd.h>`, `<stdio.h>` et `<stdlib.h>`, ont pu nous fournir les fonctionnalités nécessaires.

Par ailleurs, après avoir discuté avec notre professeur, il s'est avéré essentiel de pouvoir toujours avoir accès au terminal après avoir lancé une commande avec notre gestionnaire de tâches.

Nous avons trouvé un moyen afin de répondre à cette exigence, mais pour visualiser la sortie des commandes, il est nécessaire de rediriger l'output vers un fichier en utilisant l'option « `-o exemple.txt` ». Cette approche permet à l'utilisateur de continuer à utiliser le terminal tout en exécutant des tâches en arrière-plan et en enregistrant leurs résultats pour une consultation ultérieure.

Quant à la version finale, notre gestionnaire de tâches a révélé plusieurs défis, notamment dans l'implémentation des fonctionnalités avancées telles que la gestion des répertoires, les redirections et la répétition infinie.

3.2 Défi du Répertoire Courant :

Lors de l'implémentation de la fonctionnalité de répertoire courant, nous avons dû gérer précisément les chemins relatifs et absolus pour permettre à l'utilisateur de spécifier le répertoire d'exécution de la commande.

Cette fonctionnalité est accessible via l'option `'-w'` suivie du chemin du répertoire de travail souhaité.

De plus, nous avons ajouté l'option `'-e'` pour permettre à l'utilisateur de spécifier le répertoire contenant l'exécutable de la commande, ici ce sera souvent « `-e /bin` ». Si aucune de ces options n'est spécifiée, le répertoire courant est utilisé par défaut pour l'exécution de la commande.

3.3 Redirections Complexes :

La redirection des sorties de commandes s'est avérée être la complexité majeure rencontrée. Pour manipuler les descripteurs de fichiers et rediriger correctement stdout et stderr sans perturber le flux d'exécution normal, nous avons utilisé `freopen()`. Par exemple, pour rediriger la sortie standard vers un fichier, nous avons utilisé la commande suivante : « `freopen(output_file, "a", stdout) ;` » // le a peut être changé si besoin

3.4 Répétition Infinie et Gestion des Signaux :

Pour la fonctionnalité de répétition infinie, nous avons intégré un gestionnaire de signaux capable d'intercepter SIGINT puis on peut presser « (CTRL+C) » pour arrêter proprement l'exécution infinie. Nous avons également ajouté un fichier de log pour tracer les actions du programme, améliorant la visibilité en cours et facilitant le débogage.

3.5 Mode Ligne de Commande et Options Variées :

Nous avons enrichi le mode ligne de commande avec diverses options pour offrir une plus flexibilité à l'utilisateur. Par exemple, '-e' permet de choisir le répertoire de l'exécutable, '-w' spécifie le répertoire d'exécution et '-o' redirige la sortie. Bien qu'il n'y ait pas toutes les options présentées ici, l'utilisateur peut utiliser './main -h' ou consulter le 'readme' pour obtenir un guide complet.

La gestion des chemins d'accès et du parsing a été nécessaire, en particulier pour les commandes contenant des espaces ou des caractères spéciaux.

3.6 Limites du Projet

Une limitation actuelle de notre gestionnaire de tâches est la difficulté à gérer plusieurs commandes sur une même ligne.

Par exemple, la commande « `./main -c "echo 'Bonjour'" -c "echo 'Monde'" -i 2 -e /bin -o test.txt` » ne prend en compte que la deuxième commande spécifiée avec `-c`.

Pour résoudre ce problème de plusieurs commandes, une refonte du gestionnaire serait nécessaire, notamment pour gérer les dépendances entre les commandes et rediriger correctement leurs sorties. De plus L'option `-o` ne peut pas être utilisée avec le mode infini, car la redirection de la sortie vers un fichier n'est pas compatible avec une exécution ininterrompue de la tâche.

Enfin, le besoin d'accéder au shell et aux outils UNIX/Linux standards limite l'universalité de notre solution. Les utilisateurs doivent disposer d'un environnement compatible et, pour certaines commandes, de droits d'accès appropriés, ce qui pourrait nécessiter une configuration supplémentaire ou l'usage de `sudo` pour certaines opérations.

4. Conclusion et Perspectives

Le gestionnaire de tâches différées est capable de planifier et d'exécuter des commandes Bash à des intervalles spécifiés, tout en gérant le nombre d'exécutions et en évitant la dérive temporelle. On a travaillé pour créer une version minimale fonctionnelle, puis l'a étendue avec des fonctionnalités avancées telles l'amélioration de ligne de commandes, la gestion des répertoires, les redirections et la répétition infinie.

Nous avons rencontré plusieurs défis techniques, tels que la gestion des chemins relatifs et absolus, la redirection des sorties de commandes et l'intégration d'un gestionnaire de signaux pour gérer les interruptions. De plus, nous avons enrichi le mode ligne de commande avec diverses options pour offrir une plus grande flexibilité à l'utilisateur.

Cependant, certaines limitations subsistent, comme la difficulté à gérer plusieurs commandes sur une même ligne et la dépendance à un environnement compatible UNIX/Linux avec les droits d'accès appropriés. Pour améliorer notre gestionnaire de tâches, des ajustements seraient nécessaires pour prendre en charge plusieurs commandes et améliorer la portabilité.

Dans une perspective d'évolution, il serait intéressant d'ajouter les autres fonctionnalités prévues dans la version finale, ainsi que d'explorer de nouvelles idées pour enrichir notre outil. Cependant, nous sommes conscients que chaque nouvelle fonctionnalité implique un processus de développement et de débogage conséquent, ce qui peut rendre le projet de plus en plus complexe et chronophage.

Ainsi, il est important de trouver un équilibre entre l'ajout de nouvelles fonctionnalités et la gestion du temps. Une approche possible serait de prioriser les fonctionnalités les plus utiles et demandées par les utilisateurs, tout en veillant à maintenir la stabilité et la fiabilité de notre gestionnaire de tâches.

5. Annexes

Dans cette section, je présenterai quelques exemples d'utilisation du gestionnaire de tâches différées. Ces exemples montrent les différentes fonctionnalités et options disponibles, et démontrent que le gestionnaire de tâches a été testé et validé dans divers scénarios. Bien que des problèmes puissent survenir pour diverses raisons, ces exemples témoignent du bon fonctionnement du gestionnaire de tâches dans l'ensemble. Même si il n'y a pas l'exemple de toutes les commandes, évidemment.

Frame 1 : `./main -c "/bin/ls -l" -i 2 -n 2`

```
Revengeurs@DESKTOP-DUJQ0NI /cygdrive/e/FGES_COURS_2023_2024/Semestre2/Archi_sys/TETESTEST/harles_Steven_fonctionnel_je_crois
$ ./main -c "/bin/ls -l" -i 3 -n 2 -o t.txt
1 [Sat Apr 13 16:22:57 2024] /bin/ls -l : Exécution de la tâche...
2 [Sat Apr 13 16:23:00 2024] /bin/ls -l : Fin de l'exécution de la tâche.
3 [Sat Apr 13 16:23:00 2024] /bin/ls -l : Exécution de la tâche...
4 [Sat Apr 13 16:23:03 2024] /bin/ls -l : Fin de l'exécution de la tâche.
t.txt
1 total 158
2 -rwxrwx---+ 1 Revengeurs Aucun 862 Apr 8 19:35 attente.c
3 -rwxrwx---+ 1 Revengeurs Aucun 399 Apr 8 16:05 attente.h
4 -rw-rw-r---+ 1 Revengeurs Aucun 1251 Apr 13 16:18 attente.o
5 -rwxrwx---+ 1 Revengeurs Aucun 4541 Apr 11 05:56 commandes.c
6 -rwxrwx---+ 1 Revengeurs Aucun 1448 Apr 10 04:10 commandes.h
7 -rw-rw-r---+ 1 Revengeurs Aucun 4110 Apr 13 16:18 commandes.o
8 drwxrwx---+ 1 Revengeurs Aucun 0 Apr 10 08:05 docs
9 -rwxrwx---+ 1 Revengeurs Aucun 1100 Apr 11 05:13 gestion_signaux.c
10 -rwxrwx---+ 1 Revengeurs Aucun 900 Apr 11 05:03 gestion_signaux.h
11 -rw-rw-r---+ 1 Revengeurs Aucun 1381 Apr 13 16:18 gestion_signaux.o
```

Frame 2 : `./main -c "/bin/date" -o output.txt -i 3 -n 4`

```
Revengeurs@DESKTOP-DUJQ0NI /cygdrive/e/FGES_COURS_2023_2024/Semestre2/Archi_sys/TETESTEST/harles_Steven_fonctionnel_je_crois
$ ./main -c "/bin/date" -o output.txt -i 3 -n 4
[Sat Apr 13 16:25:20 2024] /bin/date : Exécution de la tâche...
[Sat Apr 13 16:25:23 2024] /bin/date : Fin de l'exécution de la tâche.
[Sat Apr 13 16:25:23 2024] /bin/date : Exécution de la tâche...
[Sat Apr 13 16:25:26 2024] /bin/date : Fin de l'exécution de la tâche.
1 Sat, Apr 13, 2024 4:25:17 PM
2 Sat, Apr 13, 2024 4:25:20 PM
3 Sat, Apr 13, 2024 4:25:23 PM
4 Sat, Apr 13, 2024 4:25:26 PM
```

Frame 3 : ./main -c "/bin/ls -l" -w /cygdrive/e/Animes -o output.txt

```
Revenagers@DESKTOP-DUJQ0NI /cygdrive/e/FGES_COURS_2023_2024/Semestre2/Archi_sys/TETESTEST/A
harles_Steven_fonctionnel_je_crois
$ ./main -c "/bin/ls -l" -w /cygdrive/e/Animes -o output.txt

[Sat Apr 13 16:27:03 2024] /bin/ls -l : Exécution de la tâche...
[Sat Apr 13 16:27:04 2024] /bin/ls -l : Fin de l'exécution de la tâche.
drwxrwx---+ 1 Revenagers Aucun      0 Aug 10 2023 bleach new season
drwxrwx---+ 1 Revenagers Aucun      0 Jan  4 20:11 Eminence in shadow
-rwxrwx---+ 1 Revenagers Aucun 3203419 Sep 29 2023 IMG_20230929_210137.jpg
-rwxrwx---+ 1 Revenagers Aucun 3350188 Sep 29 2023 IMG_20230929_210141.jpg
-rwxrwx---+ 1 Revenagers Aucun 4128267 Sep 25 2023 Omniscient_Readers_Viewpoint_-_Sing-shong_singsyong.epub
drwxrwx---+ 1 Revenagers Aucun      0 Apr 11 05:39 Oshinoko
drwxrwx---+ 1 Revenagers Aucun      0 Mar 10 2021 pre anime folder
-rwxrwx---+ 1 Revenagers Aucun 1676329 Mar 31 2023 Rapport_Reseau.zip
drwxrwx---+ 1 Revenagers Aucun      0 Aug  2 2023 Re Zero hajimera isekai sekatsu
drwxrwx---+ 1 Revenagers Aucun      0 Aug  2 2023 Re Zéro season one reworked
drwxrwx---+ 1 Revenagers Aucun      0 Mar 10 2021 Re0 season 2
```

Frame 4 : ./main -c "ls -lh" -i 5 -n 3 -w /cygdrive/e/Animes/Oshinoko -o /cygdrive/e/FGES_COURS_2023_2024/Semestre2/Archi_sys/TETESTEST/AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA/test.txt -e /bin

```
Revenagers@DESKTOP-DUJQ0NI /cygdrive/e/FGES_COURS_2023_2024/Semestre2/Archi_sys/TETESTEST/AAAAAAAAAAAAA
harles_Steven_fonctionnel_je_crois
$ ./main -c "ls -lh" -i 5 -n 3 -w /cygdrive/e/Animes/Oshinoko -o /cygdrive/e/FGES_COURS_2023_2024/Semestre2/Archi_sys/TETESTEST/AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA/test.txt -e /bin
```

FGES_COURS_2023_2024/Semestre2/Archi_sys/TETESTEST/AAAAAAAAAAAAA	
Nom	Modifié le
BBBBBBBBBB	13/04/2024 16
Gestionnaire_de_tâches_différées_Charles_Steven_finale_finale	10/04/2024 08
gestion_taches	10/04/2024 07
gestion_taches	10/04/2024 07
Gestionnaire_de_tâches_différées_Charles_Steven_finale_finale	10/04/2024 08
Gestionnaire_de_tâches_différées_Charles_Steven_finale_finale_Lucien	11/04/2024 04
Gestionnaire_de_tâches_différées_Charles_Steven_finale_finale_proche_output-presque	10/04/2024 19
Markdown to PDF	10/04/2024 08
Rapport_ARCHIIIIIIII	10/04/2024 08
ReadME_en_PDF	10/04/2024 08
test	13/04/2024 16

test - Bloc notes	
Finer Editer Format Affichage Aide	
total 7.0K	
-rwxrwx---+ 1 Revenagers Aucun 27K Jul 18 2023 avis_echeance-20230710-081836-655_58.pdf	
-rwxrwx---+ 1 Revenagers Aucun 493K Jul 12 2023 contrat.pdf	
-rwxrwx---+ 1 Revenagers Aucun 1.5K May 31 2023 Op11 no Kc S01101 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) (1).mkv	
-rwxrwx---+ 1 Revenagers Aucun 965K May 31 2023 Op11 no Kc S01107 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (41K) (1).mkv	
-rwxrwx---+ 1 Revenagers Aucun 580K May 31 2023 Op11 no Kc S01103 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 569K May 31 2023 Op11 no Kc S01104 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 558K May 31 2023 Op11 no Kc S01105 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 501K May 31 2023 Op11 no Kc S01106 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 376K May 31 2023 Op11 no Kc S01107 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 520K Jan 17 2023 Op11 no Kc S01108 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 580K Jan 17 2023 Op11 no Kc S01109 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 569K Jul 6 2023 Op11 no Kc S01110 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 549K Jul 6 2023 Op11 no Kc S01111 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
total 7.0K	
-rwxrwx---+ 1 Revenagers Aucun 24K Jul 18 2023 avis_echeance-20230710-081836-655_58.pdf	
-rwxrwx---+ 1 Revenagers Aucun 493K Jul 12 2023 contrat.pdf	
-rwxrwx---+ 1 Revenagers Aucun 1.5K May 31 2023 Op11 no Kc S01101 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) (1).mkv	
-rwxrwx---+ 1 Revenagers Aucun 965K May 31 2023 Op11 no Kc S01107 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (41K) (1).mkv	
-rwxrwx---+ 1 Revenagers Aucun 580K May 31 2023 Op11 no Kc S01103 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 569K May 31 2023 Op11 no Kc S01104 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 558K May 31 2023 Op11 no Kc S01105 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 501K May 31 2023 Op11 no Kc S01106 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 376K May 31 2023 Op11 no Kc S01107 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 520K Jan 17 2023 Op11 no Kc S01108 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 580K Jan 17 2023 Op11 no Kc S01109 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 569K Jul 6 2023 Op11 no Kc S01110 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 549K Jul 6 2023 Op11 no Kc S01111 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
total 7.0K	
-rwxrwx---+ 1 Revenagers Aucun 24K Jul 18 2023 avis_echeance-20230710-081836-655_58.pdf	
-rwxrwx---+ 1 Revenagers Aucun 493K Jul 12 2023 contrat.pdf	
-rwxrwx---+ 1 Revenagers Aucun 1.5K May 31 2023 Op11 no Kc S01101 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) (1).mkv	
-rwxrwx---+ 1 Revenagers Aucun 965K May 31 2023 Op11 no Kc S01107 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (41K) (1).mkv	
-rwxrwx---+ 1 Revenagers Aucun 580K May 31 2023 Op11 no Kc S01103 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 569K May 31 2023 Op11 no Kc S01104 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	
-rwxrwx---+ 1 Revenagers Aucun 558K May 31 2023 Op11 no Kc S01105 VOSTFR 1080p WEB x264 AAC -Tsundere-Raus (40K) .mkv	