

Gestionnaire De Tâches Différées

Le est un programme conçu pour l'exécution programmée de commandes dans un environnement UNIX/Linux. Ce README fournit les instructions nécessaires pour utiliser ce projet et présente plusieurs cas d'utilisation pour illustrer ses fonctionnalités.

Prérequis

- Système d'exploitation UNIX/Linux.
- Compiler le programme avec GCC en utilisant la commande `make all` .
- Pour exécuter le programme, utilisez `./main` ou `./main.exe`

Prise en Main

Pour bien démarrer avec le Gestionnaire De Tâches Différées, suivez ces étapes :

1. Téléchargez et installez Cygwin.
2. Assurez-vous d'inclure `gcc` , `make` , lors de l'installation de Cygwin.
3. Créez un dossier nommé `path/to/dossier` .
4. Sous ce dossier, créez un sous-dossier nommé `.vscode` .
5. À l'intérieur de `.vscode` , créez un fichier `settings.json` .
6. Copiez les configurations suivantes dans votre `settings.json` :

```
{  
  "terminal.integrated.shell.windows": "C:\\cygwin64\\bin\\bash.exe",  
  "terminal.integrated.shellArgs.windows": ["--login", "-i"],  
}
```

```
"terminal.integrated.env.windows": {  
  "CHERE_INVOKING": "1"  
},  
"C_Cpp.errorSquiggles": "Disabled",  
"files.associations": {  
  "gestionnaire_tache.h": "c"  
}  
}
```

7. Ouvrez VSCode et lancez un terminal Cygwin directement depuis l'éditeur.
8. Déplacez-vous dans le dossier que vous venez de créer.
9. Déplacez le programme dans ce dossier
10. Compiler avec `make all`
11. Lancez avec `./main.exe` et les options listées ci-dessous

Fonctionnalités Principales

- `-c` : Commande à exécuter.
- `-i` : Intervalle entre chaque exécution (en secondes).
- `-n` : Nombre d'itérations.
- `-f` : Exécution infinie (utiliser CTRL+C pour arrêter).
- `-o` : Fichier de sortie pour rediriger la sortie standard et d'erreur de la commande.
- `-w` : Répertoire de travail où la commande doit être exécutée.
- `-e` : Répertoire contenant l'exécutable de la commande (si différent du PATH).
- `-h` : Afficher l'aide et quitter.

Limitations

- L'option `-o` ne peut pas être utilisée avec l'option `-f` (exécution infinie).
- L'utilisation de plusieurs options `-c` dans une seule commande n'est pas supportée.

Mise en garde importante

- Il est essentiel de comprendre que le programme s'exécute correctement en tâche de fond, mais il est impératif d'utiliser l'option `-o` pour la redirection de sortie. Sans `-o`, les résultats de l'exécution de la commande ne seront pas visibles dans le terminal, bien que la commande soit exécutée. Voici un exemple d'utilisation avec la redirection de sortie :

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

- De plus, vous devez mettre le path dans la commande tel que `-c /bin/` n'importe quelle commande ou l'incure avec le choix du répertoire de l'exécutable tel que `-e /bin`

Commandes de Base

Exécuter une commande simple :

```
./main -c "/bin/ls -l" -i 2 -n 2 -o test.txt
```

Génération de dates à intervalles réguliers et écriture dans un fichier :

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

Utilisation Avancée

Lister le contenu d'un répertoire spécifique :

```
./main -c "/bin/ls -l" -w chemin/vers/répertoire -o output.txt
```

Liste des fichiers dans un répertoire spécifié avec redirection de la sortie vers un fichier :

```
./main -c "ls -l" -w chemin/vers/répertoire -e /bin -o output.txt -i 3 -n 3
```

Lister le contenu d'un répertoire spécifique et rediriger la sortie vers un fichier dans un répertoire différent :

```
./main -c "/ls -lh" -w chemin/vers/répertoire -o chemin/vers/répertoire/output.txt -e /bin
```

Créer un fichier dans un répertoire spécifié :

```
./main -c "/bin/touch nouveau_fichier.txt" -w chemin/vers/répertoire -i 2 -n 2
```

Lecture du contenu d'un fichier spécifié :

```
./main -c "/bin/cat chemin/vers/fichier" -i 2 -n 2
```

Liste des fichiers dans un répertoire spécifié à intervalles réguliers avec redirection de la sortie vers un fichier :

```
./main -c "date" -i 5 -f -e /bin -w /tmp >> date.txt 2>&1
```

Exécution infinie d'une commande (arrêt avec CTRL+C) :

```
./main -c "/bin/echo 'Exécution infinie test'" -f
```

Cas d'Utilisation

Créer un fichier dans un répertoire spécifié et écrire dans ce fichier :

1. Création du fichier :

```
./main -c "/touch chemin/vers/répertoire/fichier_test.txt" -w chemin/vers/répertoire -e /bin
```

2. Création du script:

```
./main -c "touch chemin/vers/répertoire/script.sh" -w chemin/vers/répertoire -e /bin
```

2.1 Création du shellbang sans ./main -c car ne fonctionne pas (j'ai tout essayé):

```
echo '#!/bin/bash' > chemin/vers/répertoire/script.sh && \  
echo 'echo "Hello World!" >> chemin/vers/répertoire/fichier_test.txt' >> chemin/vers/répertoire/script.sh
```

2.2 Donnez les droits au script :

```
./main -c "chmod +x chemin/vers/répertoire/script.sh" -w chemin/vers/répertoire -e /bin
```

3. Exécution du script et écriture dans le txt :

```
./main -c "sh chemin/vers/répertoire/script.sh" -n 3 -e /bin
```

4. Vérification dans le txt:

```
./main -c "cat chemin/vers/répertoire/fichier_test.txt" -e /bin
```

Assurez-vous de remplacer `chemin/vers/répertoire` et `chemin/vers/fichier` par les chemins réels sur votre système.