Rapport SAE Shell

Monitoring dans la console

Présenté par
Tim Lamour & **Jamel Bailleul**

5 décembre 2024

Table des matières

1 Membres du groupe

— **Tim Lamour** : développement des fonctions de récupération des informations, création, écriture et mise à jour des logs, gestion des erreurs lors de la lecture du fichier de configuration, optimisations, mise en forme du code, documentation et rédaction.

— **Jamel Bailleul** : développement du programme principal (interface graphique, affichage responsive, scinder les informations à l'écran..), lecture du fichier de configuration.

2 Contexte

Le projet consiste à développer une **outil de Monitoring**, qui s'exécute dans la console. Celle-ci permet de surveiller en temps réel les ressources et les performances du système, telles que le processeur (CPU), la carte graphique (GPU), la mémoire (RAM), les disques, le réseau et les processus actifs.

Toutes les données récoltées sont écrites et mises à jour dans un fichier de sortie `logs.txt` appelé le **fichier de logs** (logfile).
Certains aspects graphiques et techniques comme les couleurs ou l'intervalle d'écriture dans le logfile sont personnalisables à l'aide d'un fichier d'entré `config.txt` appelé le **fichier de configuration** (configfile).

3 Algorithme, fichiers et fonctions

Le projet est séparé en 4 fichiers :

— `main.sh` : contient le programme principale (lecture du fichier de configuration, affichage de l'interface graphique et mise à jour du fichier des logs)

— `interface.sh` : contient les fonctions permettant d'afficher et de mettre à jour l'interface.

— `recup_info.sh` : contient les fonctions permettant de récupérer les informations des ressources du système.

— `update_log.sh` : contient les fonctions permettant de créer et de mettre à jour le fichier de logs.

Voici les fonctions principales par fichiers :

— `main.sh` :

- `read_config_file` : lit le fichier de configuration et met à jour les valeurs par défaut de l'interface.

— `interface.sh` :

- `clear_screen` : nettoie la console et dessine la bordure du terminal.
- `info_reduite` : affiche les informations sur les ressources du système, une par une si elles sont disponibles.
- `affiche_processus` : affiche les informations sur les processus à une position spécifique dans le terminal.
- `print_bar` : affiche un pourcentage sous forme de barre à une position spécifique dans le terminal.
- `info_scinder` : rassemble et scinde tous les affichages précédents.

— `recup_info.sh` :

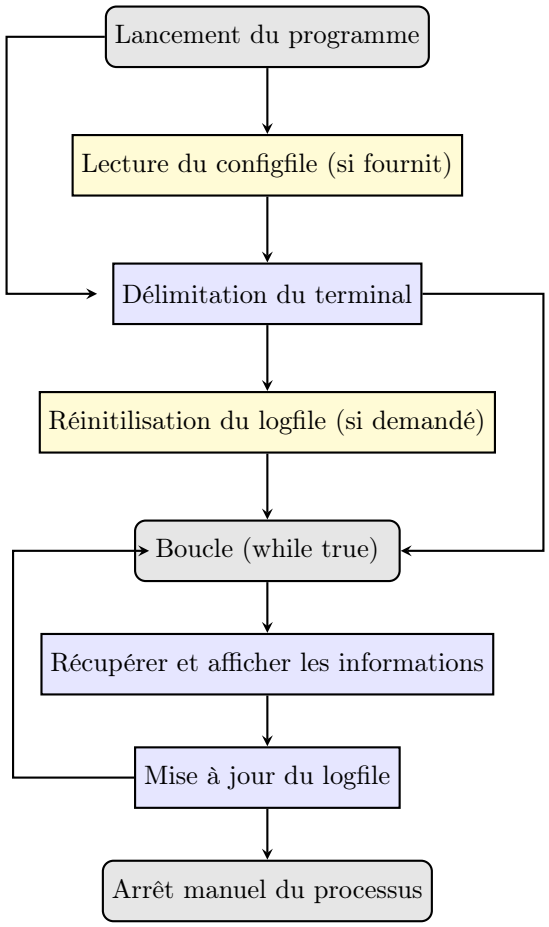
- `recup_mem` : récupère les informations sur la mémoire de la machine.
- `recup_cpu` : récupère les informations sur le processeur de la machine.
- `recup_gpu` : récupère les informations sur la carte graphiquede la machine.
- `recup_disk` : récupère les informations sur le disque courant de la machine.
- `recup_processus` : récupère les processus en cours d'utilisation de la machine.
- `recup_network` : récupère les informations sur l'utilisation réseau de la machine.
- `get_interface_name` : récupère le nom de l'interface réseau actuellement utilisé.

— `update_log.sh` :

- `create_logfile` : crée le fichier de configuration s'il n'existe pas déjà.
- `write_in_logfile` : permet d'écrire dans le fichier de configuration et donc de le mettre à jour.

4 Diagramme

Voici un diagramme montrant la logique d'exécution de l'outil :



5 Commandes et options de lancement

5.1 Lancer l'outil

Pour exécuter l'outil, vous pouvez utiliser la commande suivante dans le terminal : `./main.sh <configfile>`

5.2 Fichier de configuration

Le **configfile** est un fichier de configuration permettant de définir les couleurs de l'interface, l'intervalle pour l'écriture et la réinitialisation (ou non) du fichier de logs.
Celui-ci est optionnel car le programme a des valeurs par défaut. Il n'est donc pas nécessaire de préciser toutes les variables, vous pouvez uniquement configurer celles qui vous intéresse.

Il doit respecter la syntaxe suivante : `nom_variable=valeur`, avec une affectation par ligne. Si cette syntaxe n'est pas respectée, le processus ne se lance pas.

La majorité des variables doivent obligatoirement avoir une **couleur** (tableau 8.4) ou un **caractère UNICODE** (tableau 8.5) parmis celles et ceux disponibles.
Veuillez vous référer aux tableaux 8.1, 8.2 et 8.3 pour consulter les variables possibles ainsi que leurs valeurs.

6 Exemple d'utilisation

Voici un exemple : `./main.sh config.txt`

Contenu du fichier de configuration **config.txt** :

```
bg_color=DARK_BLACK
font_color=DARK_RED
border_color=DARK_BLUE
font_processus_color=BRIGHT_WHITE
border_char=unicode_full_block
overwrite_log=true
update_log_time=60
```

7 Conclusion

Dans l'ensemble, nous sommes assez satisfait du résultat. Cependant, il reste bien évidemment des possibilités d'optimisation, notamment au niveau de l'aspect graphique, qui n'est pas le plus esthétique ; ou encore la manière dont les informations sont récupérées et mises à jour. Une des pistes serait notamment d'utiliser des processus asynchrones (multi-processing), même si cela ne réglerait pas vraiment le problème de mono-curseur sur la console.
On pourrait également adapter le nombre de colonnes des processus à la taille du terminal ou donner la possibilité à l'utilisateur de personnaliser plus d'aspects de l'outil via le fichier de configuration, comme par exemple les unités de mesures.

Les principales difficultés rencontrées ont été de trouver ou et comment trouver les informations que l'on voulait afficher, il a fallut faire pas mal de recherche notamment dans les fichiers systèmes. Il a également était fastidieux de gérer le problème de mono-curseur, notre première version avait plusieurs secondes de latences, il a fallu ainsi optimiser nos algorithmes afin de rendre le tout le plus fluide possible visuellement.

8 Annexe : tableaux des variables configurables, des couleurs et caractères unicodes disponibles

8.1 Variables configurables avec une couleur :

Nom de la variable	**Signification**	**Valeur par défaut (couleur)**
bg_color	Couleur de fond de l'interface	BLACK
font_color	Couleur de la police principale	WHITE
border_color	Couleur de la bordure	MAGENTA
font_processus_color	Couleur de la police pour les processus affichés	BRIGHT_WHITE
full_cpu_bar_color	Couleur de la barre de progression CPU (pleine)	BLUE
full_core_bar_color	Couleur de la barre de progression CPU (pleine)	BRIGHT_BLUE
full_gpu_bar_color	Couleur de la barre de progression coeurs GPU (pleine)	CYAN
full_memory_bar_color	Couleur de la barre de progression mémoire (pleine)	GREEN
full_disk_bar_color	Couleur de la barre de progression disque (pleine)	RED
empty_cpu_bar_color	Couleur de la barre de progression CPU (vide)	WHITE
empty_core_bar_color	Couleur de la barre de progression coeurs CPU (vide)	WHITE
empty_gpu_bar_color	Couleur de la barre de progression GPU (vide)	WHITE
empty_memory_bar_color	Couleur de la barre de progression mémoire (vide)	WHITE
empty_disk_bar_color	Couleur de la barre de progression disque (vide)	WHITE

Table 1 – Variables configurables avec une couleur

8.2 Variables configurables avec un caractère UNICODE :

Nom de la variable	**Signification**	**Valeur par défaut (caractère UNICODE)**
border_char	Caractère représentant les bordures des fenêtres	unicode_full_block
full_bar_char	Caractère représentant la barre de progression pleine	unicode_dark_shade
empty_bar_char	Caractère représentant la barre de progression vide	unicode_light_shade

Table 2 – Variables configurables avec un caractère UNICODE

8.3 Autres variables configurables :

Nom de la variable	**Valeur**	**Signification**	**Valeur par défaut**
minimum_lines_width	number	Largeur minimale en lignes	30
minimum_cols_height	number	Hauteur minimale en colonnes	70
update_log_time	number	Fréquence de mise à jour du fichier de logs en secondes	60
overwrite_log	true ou false	Indique si le fichier de logs doit être écrasé	true

Table 3 – Autres variables configurables

Il n'est pas recommandé de modifier les valeurs par défaut de `minimum_lines_width` et `minimum_lines_width` car cela pourrait provoquer des problèmes d'affichage.

8.4 Couleurs disponibles :

Nom de la couleur	Code hexadécimal	Signification
BLACK	#000000	Noir
RED	#800000	Rouge
GREEN	#008000	Vert
YELLOW	#808000	Jaune
BLUE	#000080	Bleu
MAGENTA	#800080	Magenta
CYAN	#008080	Cyan
WHITE	#C0C0C0	Blanc/gris clair
BRIGHT_BLACK	#808080	Noir clair (gris foncé)
BRIGHT_RED	#FF0000	Rouge clair
BRIGHT_GREEN	#00FF00	Vert clair
BRIGHT_YELLOW	#FFFF00	Jaune clair
BRIGHT_BLUE	#0000FF	Bleu clair
BRIGHT_MAGENTA	#FF00FF	Magenta clair
BRIGHT_CYAN	#00FFFF	Cyan clair
BRIGHT_WHITE	#FFFFFF	Blanc

TABLE 4 – Couleurs disponibles

8.5 Caractères UNICODE disponibles

Nom du caractère	Code Unicode	Signification
unicode_full_block	\u2588	Bloc complet
unicode_upper_half_block	\u2580	Demi-bloc supérieur
unicode_lower_half_block	\u2584	Demi-bloc inférieur
unicode_left_half_block	\u258C	Demi-bloc gauche
unicode_right_half_block	\u2590	Demi-bloc droit
unicode_light_shade	\u2591	Ombrage léger
unicode_medium_shade	\u2592	Ombrage moyen
unicode_dark_shade	\u2593	Ombrage foncé
unicode_white_square	\u25A1	Carré blanc
unicode_black_circle	\u25CF	Cercle noir
unicode_white_circle	\u25CB	Cercle blanc
unicode_black_diamond	\u25C6	Losange noir
unicode_white_diamond	\u25C7	Losange blanc
unicode_black_star	\u2605	Étoile noire
unicode_white_star	\u2606	Étoile blanche

Table 5 – Caractères UNICODE disponibles