

Rapport SAE Shell

Monitoring dans la console

Présenté par

Tim Lamour & Jamel Bailleul

12 novembre 2024

Table des matières

1 Membres du groupe	2
2 Contexte	2
3 Algorithme, fichiers et fonctions	2
4 Diagramme	4
5 Commandes et options de lancement	5
5.1 Lancer l'application	5
5.2 Fichier de configuration	5
6 Exemple d'utilisation	5
7 Conclusion	5
8 Annexe : tableaux des variables configurables, des couleurs et caractères unicodes disponibles	6
8.1 Variables configurables avec une couleur :	6
8.2 Variables configurables avec un caractère UNICODE :	7
8.3 Autres variables configurables :	7
8.4 Couleurs disponibles :	8
8.5 Caractères UNICODE disponibles	9

1 Membres du groupe

- **Tim Lamour** : développement des fonctions de récupération des informations, création, écriture et mise à jour des logs, optimisations, mise en forme du code, documentation et rédaction.
- **Jamel Bailleul** : développement de l'interface graphique et du programme principal.

2 Contexte

Le projet consiste à développer une **application 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 par l'application 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'entrée `config.txt` appelé le **fichier de configuration** (configfile).

3 Algorithme, fichiers et fonctions

Le projet est séparé en 4 fichiers :

- `main.sh` : lit le fichier de configuration et lance le programme principale.
- `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` :

- `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` :

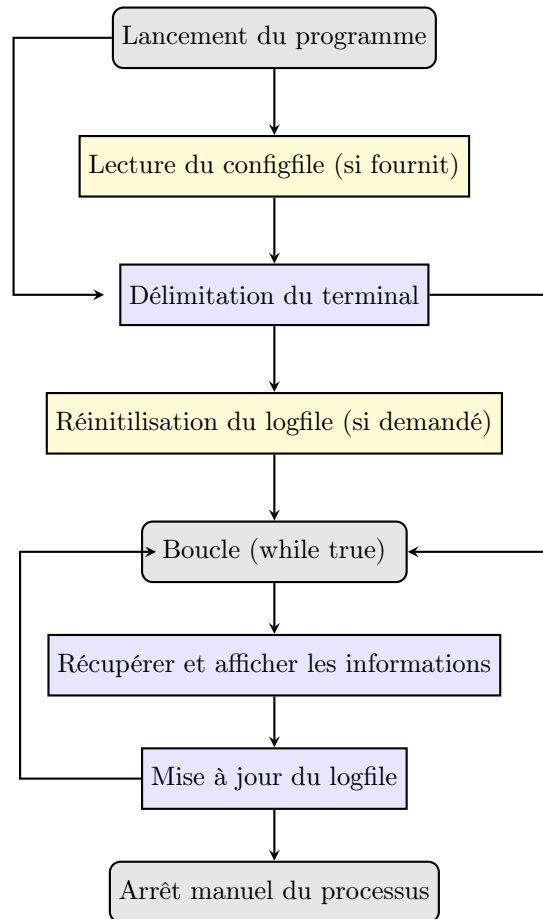
- `recup_mem` : récupère les informations sur la mémoire de la machine.
- `recup_cpu` : récupère les informations sur la carte graphique de la machine.
- `recup_gpu` : récupère les informations sur le processeur de la machine.
- `recup_disk` : récupère les informations sur les disques de la machine.
- `recup_processus` : récupère les processus en cours d'utilisation de la machine.
- `recup_network` : récupère les informations sur l'utilisation réseau de la machine.

— `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'application :



5 Commandes et options de lancement

5.1 Lancer l'application

Pour exécuter l'application, 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.

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 4) ou un **caractère UNICODE** (tableau 8.5) parmi 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
```

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, et de la manière dont les informations sont récupérées et mises à jour. Des possibles améliorations seraient par exemple d'utiliser des processus asynchrones (multi-processing), ou adapter les nombres de colonnes des processus à la taille du terminal.

On pourrait également effectuer des petits ajouts comme donner la possibilité à l'utilisateur de choisir les unités de mesure via le fichier de configuration.

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	DARK_BLACK
font_color	Couleur de la police principale	DARK_RED
border_color	Couleur de la bordure	DARK_BLUE
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)	DARK_YELLOW
full_gpu_bar_color	Couleur de la barre de progression GPU (pleine)	DARK_GREEN
full_memory_bar_color	Couleur de la barre de progression mémoire (pleine)	DARK_MAGENTA
full_disk_bar_color	Couleur de la barre de progression disque (pleine)	DARK_BLUE
empty_cpu_bar_color	Couleur de la barre de progression CPU (vide)	BRIGHT_YELLOW
empty_gpu_bar_color	Couleur de la barre de progression GPU (vide)	BRIGHT_GREEN
empty_memory_bar_color	Couleur de la barre de progression mémoire (vide)	BRIGHT_MAGENTA
empty_disk_bar_color	Couleur de la barre de progression disque (vide)	BRIGHT_BLUE

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 <code>readconfigfile.txt</code> doit être écrasé	true

TABLE 3 – Autres variables configurables

8.4 Couleurs disponibles :

Nom de la couleur	Code hexadécimal	Signification
DARK_BLACK	#000000	Noir foncé
DARK_RED	#800000	Rouge foncé
DARK_GREEN	#008000	Vert foncé
DARK_YELLOW	#808000	Jaune foncé
DARK_BLUE	#000080	Bleu foncé
DARK_MAGENTA	#800080	Magenta foncé
DARK_CYAN	#008080	Cyan foncé
DARK_WHITE	#C0C0C0	Blanc/gris foncé
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/gris clair
BRIGHT_BLACK_BIS	#555555	Noir clair bis
BRIGHT_RED_BIS	#FF5555	Rouge clair bis
BRIGHT_GREEN_BIS	#55FF55	Vert clair bis
BRIGHT_YELLOW_BIS	#FFFF55	Jaune clair bis
BRIGHT_BLUE_BIS	#5555FF	Bleu clair bis
BRIGHT_MAGENTA_BIS	#FF55FF	Magenta clair bis
BRIGHT_CYAN_BIS	#55FFFF	Cyan clair bis
BRIGHT_WHITE_BIS	#E5E5E5	Blanc/gris clair bis
DARK_RED_BIS	#800000	Fond rouge foncé
DARK_GREEN_BIS	#008000	Fond vert foncé
DARK_YELLOW_BIS	#808000	Fond jaune foncé
DARK_BLUE_BIS	#000080	Fond bleu foncé
DARK_MAGENTA_BIS	#800080	Fond magenta foncé
DARK_CYAN_BIS	#008080	Fond cyan foncé
DARK_WHITE_BIS	#C0C0C0	Fond blanc/gris foncé

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