

# Cahier des charges -- Moniteur Système

## Description

Ce document détaille le projet AISE que vous aurez à réaliser pour le semestre 2021. Vous devrez développer un outil de monitoring permettant de récupérer toutes les métriques de performance de plusieurs machines. La finalité est d'obtenir un fonctionnement proche d'outil comme `top` ou `htop` de manière distribuée. Il se construira donc en deux parties : d'un côté les capteurs (=sensors) mesurant l'activité d'une machine, d'un autre côté la ou les interfaces utilisées directement par l'utilisateur. Il est important de noter que vous avez toute liberté sur la couche de transport entre les **capteurs** et l'**interface**, libre à vous de faire du réseau, d'utiliser le système de fichiers ou autre. Les communications éventuelles entre l'interface et les capteurs se feront au moyen des méthodes I/O vues en cours. L'évaluation se fera, elle aussi, sur deux parties. Une première portera sur les composants basiques étudiés en cours, la seconde sur des aspects plus avancés. **Le projet est à réaliser en binôme**. La date de rendu est fixée au :

**Vendredi 12 Mars 2021, 23:59:59**

Le barème de cette première partie sera très proche de la répartition suivante et vous permettra d'atteindre facilement la moyenne :

Code	Composant	Description	Barème
A1	<b>Système de build</b>	Le code se construit automatiquement (Makefile, CMake...) et utilise un gestionnaire de versions (Git recommandé)	1
A2	<b>Communication entre Capteur et Interface</b>	Mettre en place un moyen de faire communiquer les capteurs et l'interface (Réseau, système de fichier, ... )	2,5
A3	<b>Premier Capteur</b>	monitoring du CPU global (charge), de la mémoire physique et de la liste des processus en cours d'exécution	3
A4	<b>Première Interface</b>	Affichage structuré sur la sortie standard des relevés des sensors connectés à intervalles réguliers. (Pris en compte à partir de 1 client).	2
A5	<b>Rapport</b>	Rapport structuré détaillant les choix d'implémentation et mesures de performances) dans un format standard type PDF.	1,5
—	<b>TOTAL</b>	—	<b>10</b>

Pour aller plus loin, et tenter d'améliorer votre score, voici un certain nombre de fonctionnalités que vous êtes encouragés à explorer. Vous êtes notés sur **27 pour une note sur 20**, il devrait être facile d'avoir une très bonne note ! Voici la liste des fonctionnalités plus complexes (verso de page) :

Code	Composant	Description	Barème
B1	<b>Client - avancé</b>	Adapter le code des capteurs pour que toutes les métriques soient regroupées par processus	3
B2	<b>Multi client</b>	Gestion de plusieurs machines en simultané	2
B3	<b>CLI uniformisée</b>	CLI complète via getopt/équivalent + help + doc	1
B4	<b>Plugins</b>	Permettre au client et au serveur de se baser sur un mécanisme de plugins. On recherchera alors dans un répertoire donné un ensemble de bibliothèques partagées qui pourront être chargées dynamiquement (dlopen) afin d'ajouter des fonctionnalités (senseur, sorties) ceci sans modification du code principal (notion de plugin).	3
B5	<b>Interactivité</b>	Action de l'utilisateur sur le programme pendant qu'il tourne (filtrage des sensors, machines, etc...)	2
B6	<b>GUI enrichi</b>	Expérience utilisateur améliorée (curses, gtk...)	2
B7	<b>Compatibilité I/O</b>	Compatible en sortie avec un outil de monitoring externe (JSON, TSDB (RRDtool), Graphana...) + preuve de validation	4
—	<b>TOTAL</b>	—	<b>17</b>

## Rendu

Vous devrez rendre une archive compressée (lisible sous Linux) nommée selon votre binôme, qui devra contenir au minimum :

- Les **sources** de l'outil, en langage C ;
- Un **README** au format texte (style « Markdown » conseillé) à la racine du projet, expliquant votre projet, la manière de le compiler et les grandes lignes d'utilisation;
- La **liste des fonctionnalités** que vous pensez avoir implémentées, en listant l'ensemble des codes des grilles ci-dessus.

La nécessité d'un réseau de machines étant requis pour évaluer correctement votre outil, il vous est fortement conseillé de travailler avec au moins deux postes par binôme, appartenant au même réseau. Un temps pourra être alloué en TD pour les étudiants ayant des difficultés à mettre cette configuration en place. Chaque archive doit nous parvenir avant la date indiquée en première page, de la manière qui vous convient (Github, email, serveur de téléchargement type WeTransfer, Torrent...). Une évaluation des projets sera faite le dernier jour (après le devoir sur table), directement par une démo de votre moniteur distribué. Vous pourrez utiliser la machine que vous souhaitez, tant que celle-ci est sous Linux. Vous pourrez avoir à nous expliquer chaque ligne de votre programme, afin que nous puissions vérifier que vous avez bien compris leur fonctionnement. Chaque archive sera sommée via `sha256sum` à réception pour vérifier qu'il n'y a pas d'altération entre la date de rendu et la démo. Nous utilisons des systèmes de détection du plagiat comme celui fourni par <http://theory.stanford.edu/~aiken/moss/>, il ne sert donc à rien de renommer les variables en copiant le code d'un autre... N'oubliez pas que vous travaillez pour vous et que la connaissance acquise au cours de ce semestre est un élément essentiel pour tout informaticien, surtout s'orientant vers le HPC.