

Mini-projet client/serveur : « monitoring pédagogique »

L'objectif de ce projet est de créer une application de monitoring de salles d'enseignement. Cette application permettra à des enseignants de suivre l'évolution de l'activité sur des postes informatiques et éventuellement d'interagir avec chaque poste. Nous distinguerons 4 niveaux d'interaction :

1. **Alerte** : un enseignant qui souhaite « suivre » une salle devra pouvoir spécifier une liste de processus (mots-clés) qu'il souhaite surveiller (par exemple « firefox »). L'objectif est que, dès qu'un tel processus sera détecté sur un des poste, l'enseignant en sera informé,
2. **Contrôle** : un enseignant peut choisir d'exécuter une commande (envoyée sous forme d'une chaîne) sur un poste de son choix. Il aura en retour la sortie de cette commande,
3. **Message** : un enseignant peut choisir de faire afficher un message sur un poste,
4. **Visuelle** : un enseignant peut choisir de recevoir périodiquement une copie d'écran réduite de tous les postes d'une salle (la fréquence de capture d'images devra être réglable). Il pourra également, à la demande, recevoir une copie d'écran en taille réelle d'un poste de son choix.

Afin de ne pas trop charger le trafic sur un poste serveur qui centraliserait tous les échanges d'images des postes de toutes les salles, vous allez créer 4 types d'agents, ce qui permettra en plus, à plusieurs enseignants, de pouvoir contrôler une même salle :

- **Master** : serveur principal de l'application, il permettra aux autres agents d'établir des connexions et de récupérer les informations utiles aux communications des autres agents.
- **Spy** : cette application sera lancée sur chaque poste des salles et aura pour mission de réaliser les 4 types d'interaction décrits ci-dessus. On peut imaginer que dans une version opérationnelle de l'application, ce programme sera démarré comme un service. Lors de la connexion du spy au master, la salle dans laquelle est le poste sera communiquée au master.
- **Contrôleur** : on utilisera un agent de type Contrôleur pour chacune des salles contrôlées. Ainsi, même si on dispose de plusieurs salles qui pourraient être surveillées, on n'aura pas forcément un agent Contrôleur en cours d'exécution pour chaque salle. En revanche, si un enseignant souhaite surveiller une salle, il devra veiller à ce qu'un agent Contrôleur soit actif pour cette salle. Lors du lancement d'un agent Contrôleur, celui-ci se connectera au master qui lui enverra la liste des postes (agents de type spy) appartenant à cette salle. L'agent Contrôleur devra alors se connecter à chacun des agent spy de cette salle, les interactions seront activées par des agents observeurs.
- **Observeur** : ce type d'agent permettra à un enseignant de surveiller une salle, on peut ainsi imaginer que plusieurs enseignants pourraient surveiller une même salle en même temps, avec des types d'interactions différents. Lorsqu'un observeur se connecte au master, il devra spécifier la salle qu'il souhaite surveiller, le master enverra en réponse les informations relative à l'agent contrôleur de la salle. Cet agent permettra ensuite à l'enseignant de spécifier les interactions de son choix et de les visualiser (l'agent spy pourra indiquer le nom de l'utilisateur connecté sur le poste).

Pour certains points techniques de ce projet, vous pourrez utiliser la fonction `system` qui prend en argument une chaîne représentant une commande et l'exécute. Consultez l'aide de la commande `import` pour les captures d'écran.

Travail à faire (par groupe de 2 ou 3 personnes) :

- écrivez la liste de tous les types de messages qui peuvent être échangés entre 2 agents. Indiquez quelles informations seront échangées avec ces messages et proposez une structure pour chaque message (ce qui définira le protocole).
- Programmez les 4 agents : en C/C++ pour le master, spy et contrôleur, dans le langage de votre choix pour l'observeur.