

Master 1 Informatique - Réseaux et Communications - Projet

Enregistrement de rapports d'activité dans une entreprise

Ce projet vous permet de mettre en œuvre des outils vus en cours : communications entre processus sur internet, threads, moyens de synchronisation, etc.

L'objectif est de réaliser une application client-serveur pour l'enregistrement de rapports d'activité quotidiens des employés d'une entreprise et leur suivi. Au sein de cette entreprise un employé privilégié, appelé "contrôleur" (exemple : secrétaire de la direction) est chargé de contrôler chaque jour l'avancement du travail d'une liste d'employés. En supposant que le nombre d'employés est important, seul une partie du personnel est concernée par jour. Ce personnel peut varier d'un jour à l'autre. Ainsi, chaque jour de la semaine, un employé doit vérifier en fin de journée si un rapport lui est demandé. Dans l'affirmatif, il le rédige en le sauvegardant sur un serveur. Par la suite, les rapports rédigés peuvent être consultés par le contrôleur.

Le programme comprend un serveur et deux types de clients : un unique contrôleur et des employés. Le serveur doit être capable de recevoir une liste des employés qui doivent établir un rapport, de traiter une demande de rédaction d'un rapport ; de sauvegarder le contenu du rapport d'un employé ; de le mettre au format PDF ; de renvoyer le résultat à cet employé et enfin de l'envoyer au contrôleur à sa demande. Le contrôleur de son côté doit pouvoir se connecter au serveur, lui envoyer une liste d'employés, vérifier la présence des rapports et les consulter. Enfin un employé doit être capable de se connecter au serveur pour vérifier s'il doit rédiger un rapport, pour éventuellement envoyer le contenu de son rapport et récupérer le fichier PDF.

Les étapes suivantes vous guident dans la réalisation de votre programme :

1 Première étape

Définir l'architecture de votre application et le choix des outils à utiliser. Cette partie devra paraître dans votre rapport avec justification de vos choix. Ne commencez pas à coder tant que cette étape n'est pas réalisée. La lecture intégrale et attentive de ce sujet vous permettra de faire vos choix. Vous devez notamment être attentifs aux traitements des connexions et différentes requêtes, aux accès concurrents, etc.

Cette étape vous permet aussi de répartir les tâches à réaliser.

2 Enregistrement d'un rapport d'activité

2.1 Protocole d'échanges entre les clients et le serveur

Pour que les clients et le serveur puissent "dialoguer" correctement, il faut organiser tous les échanges d'informations. Pour cela, il faut définir un protocole qui détermine le contenu des messages échangés entre un client et le serveur.

Définir un protocole pour la connexion/déconnexion des clients au serveur, les requêtes des clients, l'envoi du contenu du rapport et les réponses du serveur.

Quelques précisions :

- Un contrôleur doit être connu du serveur.
- Un employé qui se connecte au serveur doit s'identifier. Son identité est vérifiée en fonction du contenu de la liste des employés concernés par l'établissement d'un rapport. Ainsi, tant que le contrôleur n'a pas envoyé de liste, un employé ne peut rien faire.

- Le contenu d’un rapport sera composé de plusieurs blocs. Chaque bloc envoyé (par appui sur la touche entrée) devient une section du rapport après mise en forme par le serveur.
 - Vous devez éviter les problèmes de lecture/écriture de longs messages sur les sockets.
- Le protocole devra paraître dans votre rapport de projet avec justification de vos choix.

2.2 Réalisation des clients et du serveur

1. Programmer le serveur pour traiter les requêtes d’un seul client employé. Vous utiliserez le code qui vous est fourni à l’adresse <http://www.lirimm.fr/bouziane/enseignement/FMIN104/projet/> pour la mise en format PDF. Ce code permet au serveur de sauvegarder chaque rapport dans un fichier correspondant à l’employé : `nomemploye/nomemploye.tex` et d’en générer un fichier PDF. Notes : 1) ne pas considérer le transfert du fichier PDF pour le moment, 2) vous avez la possibilité de redéfinir le format d’un rapport et prendre en compte différemment les saisies du contenu par un employé, mais ce n’est pas une priorité et il ne faut pas y passer trop de temps.
2. Programmer l’employé. Dans un premier temps, vous créerez une liste d’employés manuellement sur le serveur.
3. Enrichir votre programme de manière à ce que le serveur puisse gérer plusieurs employés, puis que l’employé puisse recevoir son rapport (au format pdf) et le visualiser.
4. Programmer le contrôleur.

Quelques précisions :

- Vous devez faire attention aux éventuels accès concurrents à des données et à la synchronisation.
- Le contrôleur ne peut récupérer un rapport d’un employé tant que ce dernier n’a pas terminé la saisie. Une fin de saisie est déterminée par la déconnexion de l’employé. A ce moment, le serveur ajoute l’employé dans une liste des rapports saisis. Cette liste est consultable par le contrôleur.
- Nous nous contenterons d’interfaces textes pour les clients.
- Il est nécessaire d’implémenter un serveur concurrent en utilisant des threads. A vous de voir comment les utiliser à bon escient.

3 Améliorations (optionnelles)

Pour améliorer votre note, il est bien entendu possible de suggérer et implémenter des fonctionnalités supplémentaires. Le développement d’une interface graphique ou l’utilisation intelligente des notions vues en cours sera un plus. Toutefois, les améliorations que vous réaliserez **ne seront pas prises en compte** en cas de problèmes dans la réalisation des parties précédentes (erreurs à la compilation/exécution, cas non traités, code illisible, etc.).

4 Document à rendre

L’ensemble des documents à rendre sera sous forme d’une archive *nomEtudiant1-nomEtudiant2.tar.gz*, avec :

- un rapport (au format PDF) avec : mode d’emploi, description de l’architecture de votre application (justifiée), protocoles de d’échange (justifiés), schémas algorithmiques de vos programmes clients et serveur, difficultés rencontrées et solutions apportées (les problèmes du genre difficulté de gestion du temps de travail et de programme chargé, etc. n’ont pas à y paraître!!), etc. Le rapport ne doit pas excéder 6 pages.
- l’ensemble des sources accompagnés d’un Makefile, avec :

- un premier répertoire, appelé *partie-obligatoire* contenant les sources sans améliorations optionnelles.
- un deuxième répertoire, appelé *partie-optionnelle* contenant les sources avec améliorations optionnelles. Rappel, cette partie ne sera pas prise en compte si la première n'est ni complète ni correcte.

L'archive est à déposer en utilisant l'espace pédagogique. Si vous n'êtes pas encore inscrit dans le module à partir de cet espace, faites le.

Attention : aucun changement dans le code ne doit être fait par le correcteur avant de pouvoir exécuter vos programmes. Pensez alors au passage de paramètres !

5 Déroulement, dates et soutenances

Le travail est à effectuer par binômes ! Les binômes devront être annoncés en début de la première séance du projet (semaine du 3/12).

Votre présence en séance de TP pour la réalisation du projet est obligatoire pour votre suivi. Durant ces séances (et non en dehors), vous pouvez poser des questions à votre encadrant.

Date limite de dépôt des documents, le 29/12/2012 à 20h00. Après ce délai, AUCUN dépôt ou envoi par email ne sera accepté et la note sera 0/20.

Une soutenance orale se fera après vos examens de première session (les dates vous seront communiquées). Vous aurez 15 minutes pour faire une démonstration de votre projet en utilisant deux à trois machines des salles du bâtiment 6, et pour répondre à des questions (10 minutes de démonstration et 5 minutes de questions).