

Gestion des impressions dans un système

L'objectif de ce projet est la simulation de CUPS (Common Unix Printing System). Il s'agit de déléguer à une machine la gestion des travaux d'impression soumis par des utilisateurs connectés sur des machines du réseau.

La machine « serveur d'impression » recueille les demandes des utilisateurs. Elle les traite, puis elle transmet à l'imprimante concernée les données à imprimer.

Cette simulation met en jeu plusieurs niveaux d'activités simulant les entités concourant à la mise en œuvre du système d'impression et de son utilisation.

Il s'agit de les synchroniser et de les faire communiquer.

1 Les machines-sites

Chaque machine du système réparti sera simulée par un processus Unix.

Au sein de ce processus, de cette machine, les activités supportées seront représentées, elles, par des threads Posix. Ces activités implémenteront aussi bien les tâches permettant de mettre en œuvre le système d'impression que les tâches utilisatrices du service d'impression.

La communication entre les machines utilisera un module Communication fourni, implémentant une communication par socket Unix.

2 Les utilisateurs du service d'impression

Les tâches utilisatrices du service d'impression doivent pouvoir au minimum :

- Imprimer tout type de fichier (texte, pdf, image, etc.). On se donnera une fonction qui, à partir d'un chemin d'accès, fournit le type du contenu du fichier.
- Soumettre des travaux d'impression, en précisant un certain nombre de choix (la liste n'est pas exhaustive) :
 - Imprimante demandée.
 - Nombre de copies.
 - Recto/verso ou non.
- Connaître l'état d'avancement d'un travail d'impression précédemment soumis.
- Annuler un travail d'impression qui n'a pas encore été traité ou en interrompre un qui est en cours d'impression.
- Connaître l'état de la file d'attente d'une imprimante donnée.

D'autres services laissés à votre appréciation peuvent être proposés.

3 Le serveur d'impression

Pour simplifier, on supposera que le système ne comporte qu'un serveur d'impression, responsable d'un certain nombre d'imprimantes.

Le serveur d'impression s'exécute sur une machine-site, qui lui est dédiée. Il assure le lien entre les tâches utilisatrices qui s'exécutent sur des machines-sites et les imprimantes gérées.

Une imprimante pourra être directement connectée au serveur d'impression ou accessible à distance, via une autre machine-site.

Le projet ne vise pas à simuler le fonctionnement réel d'un serveur d'impression CUPS, mais à implémenter un modèle simplifié, décrit en annexe 1.

Le serveur d'impression assure, entre autres, les tâches suivantes :

- L'initialisation du service d'impression à partir d'un fichier de configuration décrivant l'ensemble des imprimantes gérées.
- La prise en compte des requêtes émises par les activités clientes. Ceci est assuré par la tâche CUPS Scheduler qui authentifie chaque requête par validation de l'identité de la machine émettrice. Cette requête correspond à :
 - La soumission d'un travail d'impression. Dans ce cas, un job est créé et mis en file d'attente, pour qu'il soit pris en charge ensuite par une tâche « CUPS Filters ».
 - Connaître l'état d'avancement d'une demande d'impression.
 - Connaître l'état d'une imprimante donnée.
 - Annuler une demande d'impression ou en stopper une en cours.
- Une tâche « CUPS Filter » prend en compte une requête d'impression en créant, à partir du fichier soumis, un fichier contenant une traduction que l'imprimante comprend. Le traitement que devra réaliser la tâche « CUPS filter » est donc spécifique au contenu du fichier à imprimer et au format intermédiaire qu'il faut produire. Le fichier produit est ensuite transmis à l'imprimante demandée.

Plusieurs requêtes, et en particulier plusieurs conversions de fichiers, doivent pouvoir être réalisées en parallèle. Il est évident que certaines demandes (état, annulation, arrêt, etc.) méritent d'être prises en compte le plus rapidement possible pour assurer la meilleure qualité de service possible.

4 Les imprimantes

Les travaux d'impression affectés à une imprimante sont pris en charge par une tâche dédiée spécifique à cette imprimante.

Lorsque l'imprimante est directement connectée au serveur (liaison USB, série, parallèle, etc.), cette tâche écrit directement le contenu du fichier à imprimer sur le périphérique.

Lorsque l'imprimante est accessible à distance, cette tâche transfère le contenu du fichier à la tâche localisée sur la machine-site imprimante qui assurera alors l'accès physique à l'imprimante. La communication entre les deux machines (serveur d'impression et imprimante distante) utilisera le module Communication fourni.

Il est évident que, si plusieurs impressions peuvent se dérouler en parallèle, une imprimante ne peut traiter ses impressions que l'une après l'autre.

5 Travail demandé

La finalité de ce projet n'est pas d'aligner des lignes de code C, mais plutôt de bien concevoir un système parallèle et réparti et d'explicitement la synchronisation et la communication mises en œuvre pour permettre aux activités qui le composent, d'assurer correctement le service demandé.

Une attention particulière sera donc portée à la description des structures de données utilisées, aux réponses apportées aux contraintes de synchronisation et de communication mises en évidence et au degré de parallélisme assuré.

Ce projet met en œuvre à maintes reprises le modèle producteurs/consommateurs. L'écriture d'un module implémentant une « classe générique » TamponPartagé pourrait contribuer à simplifier la réalisation du projet et sa lisibilité.

Le projet pourra être réalisé par binôme.

5.1 Etape 1

Rédiger un rapport succinct explicitant votre compréhension du cahier de charges et vos choix conceptuels pour y répondre. Un schéma devra présenter l'ensemble des activités ainsi que les mécanismes de synchronisation et de communication qui leur permettent de coopérer.

La synchronisation/communication au niveau processus s'appuiera sur le module fourni.

Pour ce qui est du niveau tâches au sein d'un processus, ce sont les concepts Posix (thread, condition, mutex, semaphore et signaux) qui seront utilisés.

5.2 Etape 2

Donner la conception détaillée de chaque « classe d'objets » utilisée pour mettre en œuvre ce projet (structure de données, fichier de spécification et source C d'implémentation).

5.3 Etape 3

Donner un makefile de votre projet, permettant sa génération et son test. Une version démo, ainsi qu'un compte-rendu, retour d'expérience, compléteront votre restitution.

6 Module fourni

Le fichier zip fourni contient une hiérarchie de fichiers qu'il faudra extraire telle quelle. Elle contient

- Dans le répertoire communication, les fichiers communication.h et communication.c qui implémentent le module de communication fourni, mais non imposé.
- Dans le répertoire serveur, le fichier serveur.c qui contient le code d'un processus serveur qui envoie à un client le contenu du fichier demandé.
- Dans le répertoire client, le fichier client.c qui contient le code d'un processus client qui demande à récupérer auprès d'un serveur un fichier donné.

Annexe 1

Projet Serveur d'impression

