

Forums

Description du projet

Il s'agit de concevoir un gestionnaire de forums local à une machine. La gestion des forums est assurée par un serveur lancé une seule fois à l'initialisation du système (démon), en arrière-plan par la commande `servforum` :

syntaxe : `servforum`

Chaque utilisateur a la possibilité de demander :

- la création d'un forum sur un nouveau thème,
- son abonnement à un forum,
- de poster un message dans un forum,
- de consulter les messages non lus d'un forum auquel il est abonné
- de se désabonner
- ...

De plus le serveur doit détruire tous les messages trop anciens. On demande également de conserver une trace de toutes les opérations effectuées par le serveur.

Description détaillée

Le paradigme clients-serveur. Le serveur se trouvera en écoute sur un tube. Un client contactera le serveur et lui posera une requête en écrivant sur ce tube. Le serveur engendrera alors un processus fils pour gérer la requête. Éventuellement le fils créera des tubes supplémentaires pour gérer les communications.

Organisation logique d'un forum. C'est quoi un forum ? À vous de bien organiser cet objet.

Gestion de la concurrence. Deux fils du serveur pourront partager les mêmes données, par exemple il partageront un forum. On mettra en place un système d'exclusion mutuelle pour éviter l'accès concurrent à ces données.

Gestion des signaux. La seule façon d'arrêter le serveur sera de lui envoyer un signal (par exemple, `SIGINT`).

Que peut se passer si le serveur n'intercepte pas le signal ? Par exemple :

- un signal envoyé au serveur sera envoyé aussi au fils ...
- après terminaison, on ne voudra pas laisser des tubes nommés non utilisés ...

L'interface utilisateur. Est un objectif moins important du projet, mais il est raisonnable de se proposer cet objectif.

Nous suggérons d'implémenter une suite de petits programmes correspondant à chaque commande de l'utilisateur (il s'agit du style de programmation UNIX). Une interface utilisateur se chargera alors de présenter une vision d'ensemble de cette suite.

Modalités de travail.

On travaille impérativement par binômes. Inclure le noms du binôme dans le fichier `README` et dans chaque fichier source.

Modalités de soumission.

Préparer un répertoire contenant le code source, le fichier `README`, le fichier `Makefile`, et rien d'autre. Prendre contacte avec le responsable de votre groupe de TP (plus de détails à venir). Date limite de soumission : ???.

Modalités d'évaluation

Dans l'évaluation de votre projet importance particulière sera donnée aux critères suivants :

Documentation. Votre projet contiendra un fichier `README` contenant :

- La description des programmes adressée à un utilisateur non-expert.
- Une description concise des programmes (adressée aux évaluateurs) décrivant les choix d'implémentation, les structures de données utilisés, les résultats accomplis par rapport à chaque tâche décrite ci-dessus.

Qualité du Code C. Le code source sera bien commenté, les noms des variables, des fonctions, des structures, et des types seront bien choisis. On donnera préférence à des fonction de petite taille. On utilisera de macro-constantes aux lieux des constantes explicites.

Organisation des programmes en unités logiques. Chaque unité logique sera contenue dans un fichier séparé `nom.c` (avec `nom` est bien choisi) ayant un fichier en tête `nom.h` par défaut. La liste des unités logiques (donc la liste du code source) apparaîtra dans le fichier `README`, la compilation sera géré par la commande `make` à l'aide d'un fichier `Makefile`.

Fonctionnement de votre code. Le code source sera compilé au moment de l'évaluation, possiblement sur des machines différentes des machines SUN ou LINUX. Écrire donc votre code en suivant le standard POSIX. Votre présence n'est pas nécessaire par l'évaluation (toute explication éventuelle doit être contenue dans le fichier `README`). Le fonctionnement de votre code sera testé par rapport aux résultats déclarés accomplis dans le fichier `README`.

Remarque important

Il est impératif de soumettre le projet avant la date limite. Tout projet rendu après cette date ne sera pas considéré. Il ne sera pas possible rendre le projet pendant ou après la session d'hiver. De même, il ne sera pas possible rendre une version améliorée du projet pendant ou après la session d'hiver.