SINF1252 Projet 3: outil de benchmark

1. ÉNONCÉ

1.1 Description du projet

Dans le cadre de ce projet consacré au benchmarking, vous devrez implémenter en langage C une extension à l'outil de benchmark utilisé dans le cadre du cours. Ce outil, écrit par Maxime De Mol, Nicolas Houtain et Benoît Legat permet d'évaluer les performances de certains appels systèmes Unix et est disponible à l'adresse:
https://github.com/fduchene/benchmark.

1.1.1 Démarrage

La première chose à faire est de récupérer les sources de l'outil de benchmark et à installer ce dernier. Cet exercice ayant déjà été réalisé lors des travaux pratiques, il ne devrait pas vous poser de problème. Néamoins, la procédure est détaillée dans le README (fichier README.md) disponible sur le git du projet.

1.1.2 Première partie: architecture du projet

Lorsque vous disposez du projet fonctionnel sur votre machine, vous pouvez débuter la première partie du projet. Cette partie consiste à lire, analyser et comprendre l'architecture de l'outil de benchmark. Cette étape est essentielle à la bonne marche du projet, en effet, le code que vous allez produire doit s'intégrer de manière cohérente dans le reste du projet. Veillez donc à respecter les conventions utilisées dans le projet et l'architecture de ce dernier.

1.1.3 Deuxième partie: choix des scénarios

L'architecture du projet étant désormais connue, vous devez maintenant concevoir vos scénarios de tests.

Nous vous demandons d'implémenter des benchmarks pour les appels systèmes suivants:

- `struct dirent *readdir(DIR *dirp);`
- `writev`[1] à comparer avec le même comportement implémenté en utilisant `lseek`[2] `+ write`[3]

[1] ssize_t writev(int fildes, const struct iovec *iov, int iovcnt);
[2] off_t lseek(int fildes, off_t offset, int whence);
[3] ssize_t write(int fildes, const void *buf, size_t nbyte);

Lorsque les appels systèmes sont compris, préparez les scénarios que vous comptez utiliser pour évaluer ces appels systèmes. Dans le rapport, vous devrez décrire complètement vos scénarios de benchmark et justifier de la pertinence de ceux-ci.

1.1.4 Troisième partie: implémentation des benchmarks

Une fois les appels systèmes compris et vos scénarios choisis implémentez-les de manière à respecter les conventions du projet. Nous vous invitions à tirer le meilleur parti de l'outil git lors de votre développement.

2. DÉLIVRABLES

Vous devez remettre votre code dans une archive tar.gz sur iCampus, cette archive devra comprendre deux choses:

2.1 Code

Un fichier .diff généré avec `git format-patch` contenant votre projet. Ce patch devra être appliquable sans erreurs au git du projet tel qu'il se trouve au 23/04/2014 (dernier commit: 8daf094).

2.2 Rapport

Votre rapport au format PDF expliquant en 3 pages (faces, 10pt double colonne) maximum vos choix de scénarios et d'implémentations. Ce rapport devra également contenir une analyse et une explication des résultats de vos benchmarks. Pensez à inclure une ou des figures générées par vos benchmarks pour les commenter.

3. ÉVALUATION

Ce projet sera évalué et comptera dans la note finale. Les critères de cotations (non-exhaustifs) sont les suivants:

- Qualité du code (indentation, respect des conventions,...);
- Fonctionnement du code;
- Commentaires;
- Pertinence des choix, justifications apportés, qualité de l'analyse dans le rapport;
- Exhaustivité et qualité des scénarios;