Département IMA / 3A (S6) / 2013 **Programmation Avancée**
Jérémie Dequidt, Rachid Elkouri, Julien Forget, Laure Gonnord et Nazim Haouchine
Sujet proposé par J. Dequidt, rédaction Laure Gonnord.

Projet de développement logiciel

Un crawler WEB

Objectif

Ce projet de programmation avancée a pour ambition de vous placer dans une situation concrète d'un développement logiciel en binôme, dans lequel vous êtes en charge du développement complet. Les spécifications sont volontairement floues, pas très précises.

Un soin particulier devra être apporté à votre programme, sa documentation, sa maintenance et son évolutivité. Pour cela, vous mettrez en oeuvre les notions, outils et bonnes pratiques vus en cours de programmation avancée : Makefile, compilation séparée, débogueur, SVN, ...

1 Plantons le décor ! - Contexte

L'objet de ce projet est de réaliser un "aspirateur" de page web, c'est-à dire un logiciel qui à partir d'une URL est capable de regarder tous les liens de la page web, et de parcourir tous ces liens, et les liens à partir de ces liens.

Il vous est demandé d'écrire les fonctions permettant de :
- Récupérer l'ensemble des liens à partir d'une page web donnée (utilisation de librairie).
- Réaliser le parcours à partir d'une URL donnée ou d'une liste d'URLs.
- Récupérer des statistiques sur ces pages (ne garder que les pages contenant un certain mot clef, par exemple).

À tous les niveaux, il faudra faire attention à la complexité des structures de données utilisées, monitorer, calculer des temps, faire des courbes (si cela est pertinent !), être critique, et surtout :

Faire Simple !

2 Figures Imposées

Pour réaliser votre tâche, vous allez vous appuyer sur :
- Votre code existant, écrit en TP de PA. Beaucoup de choses ont été fournies et écrites, notamment à partir du TP7.
- Des librairies existantes (`libcurl` et `libxml`) pour la récupération et la manipulation d'adresses et de pages web, ainsi qu'une interface à ces librairies (écrite par J.Dequidt). Des exemples de compilation sont fournies, il vous faudra les adapter pour compiler votre projet final.

– Un dépôt SVN avec accès restreint pour votre binôme sur lequel vous pourrez déposer votre projet et faire les opérations usuelles (commit, update).

La première séance de Tutorat sera consacrée à la prise en main de SVN et des librairies. **Une page du compte-rendu sera consacré à l'explication des codes `test_analyzer.c` et `test_fetch.c`. La compréhension totale du code fourni dans `curl_wrapper.c` et `html_wrapper.c` n'est pas exigée.**

Fonctionnalités obligatoires Voici la liste des fonctionnalités obligatoires de votre programme :

– Le parcours des URLs par transitivité à partir d'une page initiale passée en paramètre.
– L'utilisateur pourra rajouter un nombre max de pages vues lors de l'analyse, et un temps maximum d'exécution. Afin de ne pas surcharger le réseau, on ajoutera un timer (paramétrable) entre deux requêtes.
– Chaque choix sera discuté et argumenté, et on ne perdra pas de vue les performances (et le stockage des données pertinentes).
– Chaque nouvelle fonctionnalité sera évidemment testée au fur et à mesure, et vous rendez un projet qui compile sans warning et qui exécute correctement.

On vous demande d'au maximum découper le travail en deux parties, une pour chaque membre du binôme. Ce découpage sera expliqué dans le rapport.

Fonctionnalités pour aller plus loin Pour aller plus loin, on pourra notamment réfléchir aux questions suivantes, au choix :

– Comment visualiser le graphe des pages vues ?
– Comment calculer la pertinence des pages visitées (algorithme `pagerank`).

On vous laisse libre d'implémenter ou pas ces solutions, mais si vous vous y frottez, il conviendra d'avoir correctement implémenté les fonctions *obligatoires*.

3 SVN et consignes pour le rendu

SVN Vous travaillerez dans les dépôts qui ont été utilisés au TP de familiarisation à SVN.

Travail de chez vous Pour travailler de chez vous, vous vous rapporterez à la feuille du TP SVN. Vous pourrez avoir besoin d'installer les packets suivants : `libcurl4-gnutls-dev`, `libxml2` et `zlib1g-dev` et de modifier `Makefile.inc` en conséquence.

Rendu Nous récupérerons **le mercredi 5 juin 2013 20h** (5 points en moins par jour de retard) vos projets dans vos dépôts qui devront avoir la structure suivante :

- un fichier `README` contiendra une description rapide de votre logiciel, de ses fonctionnalités et un mode d'emploi succinct.
- un répertoire `Code` (avec Makefile). Pour faciliter la correction, le binaire s'appellera `crawler`. Chaque version éventuelle sera dans un sous-répertoire à part.
- un répertoire `Tests` qui comprendra quelques scripts de tests.
- éventuellement, un répertoire old contenant du code inutile.
- un fichier `nomdubinome.pdf` contiendra votre rapport. Le rapport ne comprendra pas plus de 6 pages, devra être clair et précis et notamment comporter les limitations de votre outil. **PAS de rapport papier SVP !**

Attention ! votre dépôt SVN devra être propre, ie ne pas comporter de fichier .o, tilde, binaire, etc.

Sur la page web du cours, on fournit un script Python qui permet de vérifier que l'arborescende de votre dépôt respecte les consignes. Des points seront enlevés aux binômes pour lesquels le script ne s'exécute pas jusqu'au bout.

Modalités d'évaluation Nous évaluerons la maîtrise des outils présentés lors du cours de Programmation Avancée, ainsi que la qualité de votre développement et de votre programme :

- les fonctionnalités, évidemment, mais aussi ...
- l'utilisation des librairies ;
- le découpage des fonctions, les commentaires, le découpage en modules, l'arborescence du SVN ;
- le travail individuel de chacun des membres du binôme en TP et l'équilibre entre les codes écrits par chacun.
- l'utilisation du SVN ;
- les aspects maintenance (la doc programmeur, la lisibilité du code) ;
- les aspects utilisateur (la doc utilisateur, les exemples, ...)
- **la gestion de ces différents points durant les séances de TP sera aussi évaluée**

Évidemment, cette liste n'est pas exhaustive ! Vous pouvez vous rapporter aux consignes du projet du semestre 5.

Enfin, rappelons à toutes fins utiles :

C'est un projet en binôme uniquement. Le partage d'algorithmes et/ou de code entre binômes différents est strictement interdit. Il est aussi interdit de partager son découpage du problème en sous-tâches. Nous voulons des solutions différentes !