

Projet de développement logiciel

Statistiques sur l'oeuvre de Shakespeare

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

La page web <http://shakespeare.mit.edu/> fournit quelques oeuvres du célèbre auteur anglais, sous forme de pages web (html).

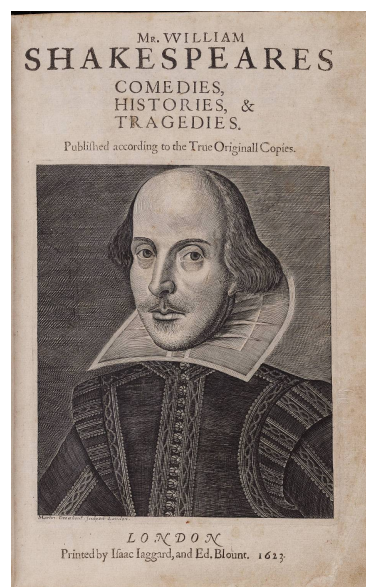


Figure 1: Shakespeare, 1564-1616, image Wikipedia (domaine public)

Nous allons réaliser des statistiques sur ces oeuvres (crédit idée : Bruno Salvy).

Il vous est demandé d'écrire les fonctions permettant de :

- Traiter des fichiers HTML en masse.
- Stocker les mots des tirades de Shakespeare.
- Manipuler ces mots et faire des statistiques.

À tous les niveaux, il faudra faire attention à la complexité des structures de données utilisées, monitorer, calculer des temps, faire des courbes, être critique.

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

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

- Le programme prendra en argument un répertoire comportant un ensemble de fichiers html à traiter et traitera tous ces fichiers.
- Le programme utilisera des tables de hachage avec résolution de collisions par liste chaînée. Plusieurs (au moins deux, dont la fonction de Bernstein¹) fonctions de hachage seront codées et comparées (théoriquement et expérimentalement). Un bonus sera accordé aux personnes utilisant la généricité (`void*`²) pour passer en paramètre les fonctions de hachage.
- Les mots seront des suites de caractères (*a* à *z*, *A* à *Z*, on oubliera virgules, tirets, ...), et on les stockera en minuscule et au singulier.
- Le programme devra être capable de donner le nombre d'occurrences de chaque mot, de donner les mots qui n'apparaissent pas dans le dictionnaire anglais du TP8, et de donner le(s) fichier(s) et la(les) ligne(s) d'apparition d'un mot donné.
- Chaque choix sera discuté et argumenté, et on ne perdra pas de vue les performances.
- 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 estimer (approximer) la taille du vocabulaire de Shakespeare ?
- Comment extraire en une seule passe une liste de 15 mots caractéristiques de Hamlet ?
- Comment comparer des pièces deux à deux en estimant leur similarité ?

et s'inspirer des pointeurs et des algorithmes décrits dans :

http://algo.inria.fr/salvy/INF431/Projet/INF431_-_Projet_Informatique.html

¹voir wikipedia

²documentation sur la page web du cours

3 SVN et consignes pour le rendu

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

Rendu Nous récupérerons **le 6 Juin 2012 20h** (5 points en moins par jour de retard) vos projets dans vos dépôts qui devront avoir la structure suivante :

- un fichier `Readme.txt` 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 `shakespeare`. 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 5 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'arborescence de votre dépôt respecte les consignes. Des points seront enlevés aux binômes pour lesquels le script renvoie `Fail`.

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 reporter 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 !