

Mybib

Utilitaire de manipulation de BIB_TE_X

Projet de compilation

1^{ère} année du Master d'informatique de l'Université de Rouen

Année universitaire 2015-16

Nicolas Bedon, Arnaud Lefebvre, Ludovic Mignot

Le projet consiste à écrire en C, et en utilisant flex et bison pour les parties concernant les analyses lexicales et syntaxiques, un petit outil permettant la manipulation de références bibliographiques au format BIB_TE_X.

BIB_TE_X est un outil très utilisé dans la production d'ouvrages scientifiques. Il facilite la gestion de bibliographies dans la production de documents (articles, livres, etc...) écrits en T_EX [3], L^AT_EX [4] ou dérivés. Nous renvoyons à [2] pour une rapide présentation de BIB_TE_X, mais il existe également beaucoup d'autres références.

Les fichiers de bases de données bibliographiques (fichiers `.bib`, aussi appelés *fichiers bibtex* dans la suite de l'énoncé) peuvent rapidement prendre une taille importante, et devenir, au fil des années, mal entretenus : par exemple, des doublons peuvent apparaître, certaines conventions peuvent être respectées par endroits et pas d'autres, etc. Le sujet du projet est d'écrire un petit outil de manipulation de fichiers BIB_TE_X, mybib, pour aider à résoudre ces problèmes.

1 Détails

`mybib -b toto.tex` : génère, à partir des informations contenues dans `toto.tex`, un `bibtex` restreint aux entrées réellement utilisées dans le document `toto.tex`, et construit à partir de la base de données bibliographique utilisée dans `toto.tex`. Attention, un `.tex` peut en contenir un autre (les commandes T_EX et L^AT_EX `input` et `include` permettent en effet d'inclure un fichier `.tex` dans un autre, nous supposons pour simplifier que leur argument est un nom de fichier donné en constante, et jamais par une variable T_EX ou L^AT_EX).

`mybib -c toto.bib` : vérifie que chaque entrée a une clef qui lui est propre.

`mybib -e opt regexp toto.bib` : extrait de `toto.bib` les enregistrements dont certains champs contiennent un facteur correspondant à l'expression rationnelle *regexp*. *opt* peut être *k* (key), *t* (title), *a* (authors), *n* (note), *x* (annotate), ou n'importe quelle combinaison disjonctive possible. [Dans l'implantation de cette option, vous utiliserez la bibliothèque POSIX `regex` pour rechercher des motifs à partir d'une expression rationnelle dans un texte].

`mybib -k toto.bib` : remplace les clefs de `toto.bib` par des clefs normalisées

(de la forme ABC :1984, où A, B et C sont les initiales des noms des auteurs de l'entrée, 1984 est l'année de publication. Si plusieurs entrées peuvent avoir ABC :1984 pour clef normalisée, alors on ajoute “ :x” à la suite de la clef normalisée, où x est un numéro de séquence : 1,2,3,... (ce qui donne ABC :1984 :1, ABC :1984 :2, ...).

mybib -s toto.bib : pour tous les enregistrements, extraire (s'ils existent) les champs *publisher*, *organization*, *series*, *journal*, et les définir à partir d'une chaîne de caractères BIB_{TEX} (@String), que vous devrez entièrement définir. Dans le bibtex de sortie, les @String devront être regroupés au début, et par genre, avec un paragraphe pour chaque genre (un commentaire indiquera le genre). Par exemple, sur le fichier BIB_{TEX} fourni avec ce sujet, la sortie devra être le fichier *bibtex2.bib*.

mybib -t *type* toto.bib : ne garde que les entrées de type *type*.

mybib -u toto.bib : supprime les doublons : même type, même titre (à la casse près), mêmes auteurs, même année. Les doublons retirés devront être stockés dans un fichier annexe de nom *toto.bib.doublons*.

Par défaut, la sortie est produite sur la sortie standard ; si l'option -o *fic.bib* est présente, alors la sortie est générée dans *fic.bib*, et les fichiers annexes produits ont pour préfixe de nom *fic*.

2 Ce qu'il vous est demandé

Vous devez écrire en C, et en utilisant flex et bison pour les parties concernant les analyses lexicales et syntaxiques, un programme permettant de réaliser chacune des fonctions présentées dans la section précédente.

Bien entendu, votre projet devra être le mieux écrit possible : algorithmique adaptée, code clair et commenté.

Vous pouvez étendre le projet si vous le souhaitez, en rajoutant des fonctionnalités par exemple. Cependant, ne le faites que si la base qui vous est demandée est implantée et fonctionne correctement : il est préférable d'avoir un projet qui fait correctement le minimum plutôt que d'avoir un projet étendu dont le minimum demandé ne fonctionne pas.

Votre projet devra être rendu avec un jeu d'exemples illustrant le mieux possible ses fonctionnalités, ainsi qu'un rapport contenant un rapport de développement et un manuel d'utilisation.

Il devra être développé individuellement ou par binôme, et rendu au plus tard le 11 janvier 2016 au soir, dans une archive au format tar gzippé de nom FrancoisDupontJacquesDurant.tar.gz pour un binôme (si Francois Dupont et Jacques Durant sont vos noms), envoyée en pièce jointe à un courriel de sujet “Projet MYBIB” à l'adresse de courriel de votre chargé de TP (Nicolas.Bedon@univ-rouen.fr, Arnaud.Lefebvre@univ-rouen.fr ou Ludovic.Mignot@univ-rouen.fr). Si vous avez fait le projet en binôme à cheval sur deux groupes de TP, votre projet est à envoyer aux deux chargés de TP. Vous vous mettrez en copie du courriel pour vérifier que vous n'oubliez pas la pièce jointe. L'extraction du

fichier d'archive devra produire un répertoire de nom FrancoisDupont contenant le code source de votre projet, des jeux d'exemple et un rapport de projet.

Votre projet fera l'objet d'une soutenance sur machine.

Références

- [1] Nicolas Bedon, Arnaud Lefebvre, and Ludovic Mignot. Mybib : utilitaire de manipulation de BibTeX, November 2015. Projet de compilation, 1^{ère} année du Master d'informatique de l'Université de Rouen, année universitaire 2015–16.
- [2] Jürgen Fenn. Managing citations and your bibliography with BibTeX. *The PracTeX journal*, (4), 2006.
- [3] Donald E. Knuth. *The TeXbook*. Addison-Wesley, 1986.
- [4] Leslie Lamport. *LaTeX : A document preparation system : User's guide and reference*. Addison-Wesley, 1994.