Licence d'informatique
Module de C/Unix

Projet C

webcheck et webview — Vérification de sites Web

Jean-Luc LEVAIRE et Philippe MARQUET

Décembre 2000

Les URL utilisées dans ce document sont directement accessibles depuis la version en ligne : http://www.lifl.fr/~marquet/ens/cu/webtools/. Des copies des documents et logiciels référencés y sont disponibles localement.

1 Vérificateur de site webcheck

Le but de cette première partie est de réaliser une commande

```
webcheck url
```

qui vérifie la cohérence d'une page HTML désignée par une URL (*Uniform Resource Locator*). Une page HTML est *cohérente* si tous les *liens* qu'elle référence sont effectivement accessibles, et si ces liens sont eux-mêmes cohérents.

1.1 Syntaxe d'une URL

La syntaxe d'une URL est la suivante :

protocole ://*host*[:*port*]/*location*

protocole désigne le protocole à utiliser lors de la connexion (par exemple http, ftp, mailto, file...) ;

host est le nom de la machine abritant le serveur ;

port est le numéro de port TCP avec lequel établir la connexion (par défaut 80) ;

location désigne le nom local au serveur de la page à laquelle on accède. *location* se présente sous la forme d'un chemin à la Unix (identifiants séparés par des /), suivi d'un éventuel nom de fichier.

Tous ces composants sont optionnels, une URL vide désignant la page courante. Pour plus d'informations, on se rapportera à la définition standard de la syntaxe http ://www.ietf.org/rfc/rfc2396.txt.

1.2 Liens référencés par une page

Les liens référencés par une page HTML sont indiqués par les étiquettes HTML suivantes :

- <A ... HREF="lien" ...> ;
- <LINK ... HREF="lien" ...> ;
- <IMG ... SRC="lien" ...> ;
- <FRAME ... SRC="lien" ...> ;
- <BODY ... BACKGROUND="lien" ...> ;
- <LAYER ... SRC="lien" ...>.

Les liens apparaissant dans ces étiquettes sont ici aussi désignés par des URL. En général, ces URL sont relatives à une URL de base, qui est par défaut l'URL de base de la page où se situe le lien. Par exemple, la page http://www.lifl.fr/~levaire/C/index.html a pour URL de base http://www.lifl.fr/~levaire/C/ et un lien dans cette page dont l'URL est c.html

Id : webtools.tex,v 1.5 2000/12/08 11 :20 :40 marquet Exp

désigne en fait l'URL absolue `http://www.lifl.fr/~levaire/C/c.html`. De même, le lien `../images/button.gif` désignera l'URL absolue `http://www.lifl.fr/~levaire/images/button.gif`.

Notez que le protocole HTML permet de spécifier une URL de base dans une page par l'étiquette `<BASE HREF="base">` qui se place dans la section d'en-tête de la page (section délimitée par `<HEAD>...</HEAD>`).

Pour plus de détails sur les URL relatives, reportez vous aux sections 5.1 et 5.2 du standard `http://www.ietf.org/rfc/rfc2396.txt`.

1.3 La commande **webcheck**

La commande `webcheck url` imprime sur la sortie standard l'ensemble des liens référencés par la page désignée par `url`.

On imprimera hiérarchiquement les URL *absolues* référencées, avec pour chacune d'elles, son état :
- `ok` si l'URL est accessible ;
- un message d'erreur dans le cas contraire (`url non trouvée` ou `hôte inconnu` par exemple).

Pour implémenter la commande `webcheck`, deux principaux problèmes vont apparaître :
- le référencement d'une page par une URL : les URL sont rarement absolues, une même page peut posséder plusieurs URL. Il faut donc déterminer l'URL absolue de chaque page référencée ; c'est celle qui sera utilisée pour télécharger la page ;
- la présence de cycles dans le référencement de pages entre elles : il faut éviter de boucler sur ces cycles, et donc maintenir une structure permettant de se souvenir des pages déjà visitées.

La commande `webcheck` ne vérifie pas les liens référencés par une page qui n'ont pas le protocole `http` (par exemple, `mailto :toto@site.org`).

De plus, `webcheck` ne vérifiera pas récursivement les liens suivants (on testera seulement leur accessibilité) :
- les liens dont le type de contenu (*content-type*) n'est pas `text/html` ou `text/plain`. Celui-ci est en général envoyé par le serveur dans l'entête de sa réponse, et si ce n'est pas le cas, on se limitera aux liens dont le nom de fichier a pour extension `.html` ou `.htm` ;
- les liens dont l'URL de base n'est pas un suffixe de l'URL de base passée à la commande `webcheck` ; on évite ainsi de vérifier l'ensemble du réseau !

Exercice 1
Proposez une structure de données et un algorithme pour implémenter la commande `webcheck`. Vous avez à votre disposition une fonction `Download()` qui télécharge une URL dans un fichier, et une fonction `getxref()` qui extrait d'un fichier tous les liens référencés (voir ci-dessous la section 1.4).

Exercice 2
Implémentez la commande `webcheck`. Utilisez **impérativement** la programmation modulaire pour ce faire, et commentez clairement les fichiers d'en-tête de vos modules (en plus des fichiers sources). Indiquez dans votre rapport les limitations de votre programme (par exemple, ne tient pas compte de l'étiquette `<BASE>`) et/ou les extensions (par exemple, teste les liens de protocole `ftp`).

La page `http://www.lifl.fr/~levaire/C/webtools/testweb.html` contient un petit exemple de test.

1.4 Fonctions utilitaires

Des fonctions utilitaires et deux programmes d'exemples sont à votre disposition dans `http://www.lifl.fr/~marquet/ens/cu/webtools/webutils.tgz`.

Dans le fichier `http.c`, vous trouverez la fonction

```
StatusFlags Download (char *pszHost, int nPort, char *pszFile,
                      FILE *pDest, int *statuscode,
                      char **relocation, char **contenttype) ;
```

Cette fonction

- télécharge la *location* `pszFile` depuis la machine *hôte* d'adresse `pszHost`, en utilisant le *port* TCP `nPort` ;
- le contenu de la page se trouve alors dans le fichier local `pdest` ;
- elle retourne un ensemble de bits de statut dans un champ de bits de type `StatusFlags` (voir le fichier `web.h`) ;
- en particulier le bit `got_contenttype` permet de savoir que le serveur a spécifié le type du contenu de la page. Celui-ci a alors été recopié dans le paramètre `contenttype` ;
- de même le bit `relocation` indique que la page a bougé, et sa nouvelle URL a alors été recopiée dans le paramètre `relocation` ;
- enfin le paramètre `statuscode` contient le code d'état du protocole de transfert HTTP tel que défini dans le standard `http://www.w3.org/Protocols/HTTP/1.1/rfc2616.txt.gz`

Pour tester cette fonction, compilez le programme `trydownload`, et téléchargez la page Web du module C/Unix dans le fichier local `essai.html` :

```
% make trydownload
% ./trydownload www.lifl.fr \~marquet/ens/cu/ essai.html
```

Dans le fichier `getxref.c`, vous trouverez la fonction

```
int getxref (FILE *fp, xreflist *xrefs);
```

qui extrait tous les liens référencés par une page contenue dans le fichier `fp`. Ces références sont stockées dans une structure de type `xreflist` (voir `web.h`). Pour tester cette fonction, compilez le programme `trygetxref` et essayez avec le fichier précédemment téléchargé :

```
% make trygetxref
% ./trygetxref essai.html
```

Ces fonctions sont directement dérivées de fonctions utilisées par l'utilitaire `webcrawl` écrit par Julian R. Hall. Vous pouvez le télécharger sur `http://proxad.linux.tucows.com/conhtml/adnload/8780_32654.html`, et vous inspirer des fonctions qui y sont définies (en particulier dans `url.c`).

2 Visualisateur de site `webview`

Il est intéressant de pouvoir disposer d'une représentation graphique d'un site Web (ou d'une partie d'un site Web), pour y identifier visuellement les possibilités de navigation à l'intérieur du site ; on peut aussi afficher quantité d'informations concernant les pages et les liens.

2.1 L'outil `graphlet`

Pour cela, nous allons utiliser un visualisateur de graphes ; pour nous faciliter la tâche, nous pouvons aussi laisser ce visualisateur réaliser automatiquement le placement des nœuds du graphe suivant divers algorithmes. L'outil `graphlet` implante ces fonctionnalités ; il est disponible sur le site `http://www.fmi.uni-passau.de/Graphlet`.

2.2 Exemple de résultat de `webview`

Par exemple, la figure 1 représente le site de test proposé `http://www.lifl.fr/~levaire/C/webtools/testweb.html` après avoir appliqué l'algorithme de placement automatique `Extended DAG`[1] de `graphlet`.

[1]Cet algorithme semble être le plus approprié pour afficher des graphes représentant des sites Web.

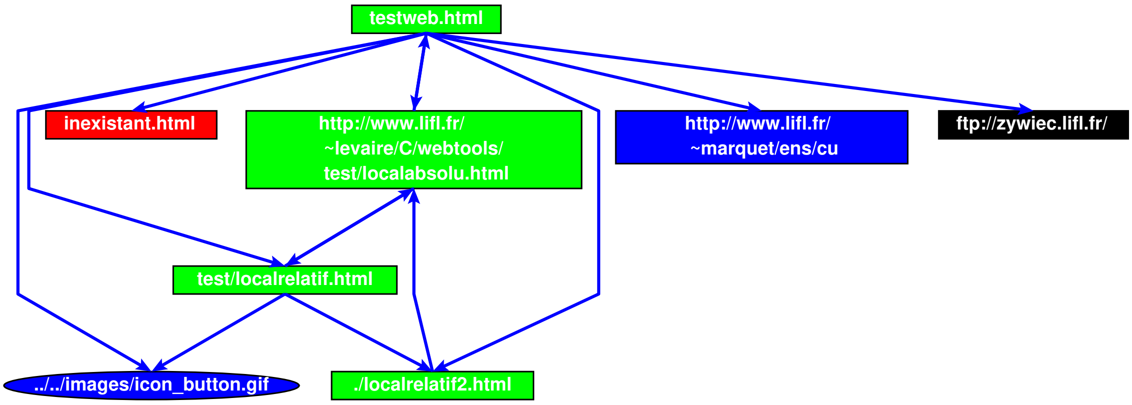

FIG. 1 – La représentation graphique du site d'essai `http://www.lifl.fr/~levaire/C/webtools/testweb.html`
En vert, les liens `ok`, en bleu les liens que l'on ne vérifie pas récursivement, en noir les liens que l'on ne vérifie pas du tout, et en rouge les liens inexistants.
(Les couleurs sont mieux identifiables dans la version en ligne du sujet, `http://www.lifl.fr/~marquet/ens/cu/webtools/`).
Le fichier GML utilisé pour produire cette représentation graphique est disponible : `http://www.lifl.fr/~marquet/ens/cu/webtools/webtools.g`. Ce fichier est un exemple de ce que peut être le résultat de votre commande `webview`.

2.3 Format des fichiers GML

Pour afficher un tel graphe, il faut générer, à partir du parcours du site, un fichier au format GML (Graph Modelling Language). Ce format est un format textuel, où l'on déclare les nœuds en leur donnant un identifiant (`id`), et les arêtes en spécifiant les identifiants du nœud source et du nœud destination. Par exemple :

```
graph [
    directed 1
    id 42
    node [
        id 1
        label "testweb.html"
        graphics [
          type "rectangle"
          fill "green"
        ]
    ]
    node [
        id 2
        label "http://www.lifl.fr/~levaire/C/webtools/test/localabsolu.html"
        graphics [
          type "rectangle"
          fill "green"
        ]
    ]
    edge [
    source 1
    target 2
    ]
]
```

déclare un graphe dirigé contenant 2 nœuds d'identifiant respectif 1 et 2, et une arête du nœud 1 vers le nœud 2. L'étiquette d'un nœud (ou d'une arête) est précisé dans le champ `label` et ses attributs graphiques dans le champ `graphics`. On ne précise aucun attribut graphique de position, simplement un type et une couleur. C'est l'algorithme de placement automatique qui modifiera en fait les attributs graphiques de position. Vous trouverez de plus amples précisions sur ce langage dans la documentation `http://www.fmi.uni-passau.de/Graphlet/download/misc/GML.ps.gz`.

2.4 La commande **`webview`**

Exercice 3
Proposez une structure de données pour conserver (après sa vérification) l'ensemble des liens à l'intérieur d'un site, ainsi que leur état (au sens de `webcheck`). On utilisera le même algorithme que précédemment pour vérifier un site.

Exercice 4
Proposez un algorithme qui parcourt la structure de données que vous avez proposée et génère un fichier au format GML représentant le site. Vous êtes libre de choisir la représentation que vous voulez, si vous la décrivez bien dans votre rapport. Vous pouvez utiliser la représentation suggérée à la figure 1.