ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
Τμήμα Πληροφορικής

ΕΠΛ 421 - Προγραμματισμός Συστημάτων

ΑΣΚΗΣΗ 2 – Συλλογή και Ανάλυση Ιστοσελίδων με χρήση Τεχνητής Νοημοσύνης μέσω Προγραμματισμού Κελύφους Bash

Διδάσκων: Δημήτρης Ζεϊναλιπούρ
Υπεύθυνος Εργαστηρίου: Παύλος Αντωνίου

Ημερομηνία Ανάθεσης: Τετάρτη 15/02/23
Ημερομηνία Παράδοσης: Τετάρτη 08/03/23 και ώρα 14:00 (21 μέρες)

(η λύση να υποβληθεί σε zip μέσω του Moodle)
https://www.cs.ucy.ac.cy/courses/EPL421

I. Στόχος Άσκησης

Στόχος αυτής της άσκησης είναι η εξοικείωση με προχωρημένες τεχνικές προγραμματισμού στο κέλυφος Bash, και η εκτίμηση της ευκολίας με την οποία μπορεί κανείς να δημιουργήσει ένα σύνθετο σύστημα μέσω προγραμμάτων ωφελιμότητας (system utilities). Συγκεκριμένα, σε αυτή την άσκηση θα έχετε την ευκαιρία να χρησιμοποιήσετε έννοιες Διαχείρισης Συστημάτων (System Administration) και στη συνέχεια να κάνετε χρήση των εντολών του UNIX μέσω Προγραμματισμού Κελύφους (Bash Programming): εντολή *exec, πίνακες, συνθήκες ελέγχου, δομές επανάληψης, κανονικές εκφράσεις, επεξεργαστές ροών (sed, awk) και χρήση συναρτήσεων με τα προαναφερθέντα.*
Το θέμα της άσκησης είναι η υλοποίηση ενός προγράμματος συλλογής και ανάλυσης ιστοσελίδων HTML (Hypertext Markup Language) από τον Παγκόσμιο Ιστό Πληροφοριών (WWW) **χωρίς την χρήση κάποιου έτοιμου εργαλείου (π.χ., curl ή wget).** Οι λειτουργίες του προγράμματος σας και το αναμενόμενο αποτέλεσμα περιγράφονται αναλυτικότερα στην συνέχεια.

II. Προαπαιτήσεις

Το πρωτόκολλο HTTP ακολουθεί το μοντέλο πελάτη-εξυπηρετητή (client-server). Περισσότερες λεπτομέρειες θα δώσουμε πιο κάτω. Ο εξυπηρετητής ονομάζεται HTTP (Web) server και ο πελάτης HTTP (Web) client ή πιο απλά φυλλομετρητής (browser). Για το σκοπό της άσκησης θα χρησιμοποιήσουμε τον Apache Web server που είναι ήδη εγκατεστημένος στο VM σας από το εργαστήριο 1. Πιο συγκεκριμένα, το εγκατεστημένο πρόγραμμα ονομάζεται apache2[1] και είναι ο Apache Web server, ο οποίος είναι σχεδιασμένος να τρέχει σαν αυτόνομη διεργασία (standalone daemon process). Από προεπιλογή, ο Web server είναι ρυθμισμένος στο να «ακούει» για αιτήσεις (http requests) στην θύρα (port) 80 μιας μηχανής. Για κάθε εισερχόμενη αίτηση, ο HTTP server δημιουργεί είτε μια διεργασία (process) ή ένα νήμα (thread) για να χειριστεί την αίτηση. Οι διεργασίες και τα νήματα περιλαμβάνονται στην ύλη του μαθήματος, σε μεταγενέστερο στάδιο, και δεν θα μας απασχολήσουν στην άσκηση αυτή. Μπορείτε να διαχειριστείτε τον Apache Web server (όπως και κάθε υπηρεσία στο linux) μέσω της γραμμής εντολών.

1 Αν ο apache2 daemon δεν είναι εγκατεστημένος, δείτε ξανά το εργαστήριο 1.

Ο Apache HTTP server μπορεί να «σερβίρει» προς τον έξω κόσμο τα αρχεία και τους καταλόγους που βρίσκονται στον μονοπάτι /var/www/html (DocumentRoot) του VM σας.

III. Εισαγωγή στη συλλογή και ανάλυση ιστοσελίδων

Οι μηχανές αναζήτησης, όπως για παράδειγμα το Google, χρησιμοποιούν μηχανές ανάκτησης πληροφοριών (crawlers, spiders, bots ή ants), για να συλλέξουν τις πληροφορίες (αρχεία υπερκειμένου, εικόνας, βίντεο, κτλ) οι οποίες βρίσκονται αποθηκευμένες, κατά κατανεμημένο τρόπο, στους HTTP servers. Οι πληροφορίες αυτές αποτελούν τον Παγκόσμιο Ιστό (World Wide Web). Αυτό επιτρέπει στις μηχανές αναζήτησης να ανακτήσουν στον τοπικό τους δίσκο ένα υποσύνολο του WWW. Στην συνέχεια χρησιμοποιούν μια σειρά από αλγορίθμους για να δημιουργήσουν αποδοτικά ευρετήρια (indexes), τα οποία χρησιμοποιούνται στις αναζητήσεις σας όταν επισκέπτεστε τέτοιες μηχανές αναζήτησης!
Σκοπός αυτής της άσκησης δεν είναι βεβαία η δημιουργία ενός νέου Google, αλλά η συγγραφή ενός εξειδικευμένου crawler, ο οποίος να επισκέπτεται τα αρχεία ενός ιστοχώρου που βρίσκονται σε κάποιο Web server του όποιου ξέρουμε το URL, να αποθηκεύει τοπικά το περιεχόμενο που ανακτάται, και στην συνέχεια να το επεξεργάζεται.

IV. Προγραμματισμός Ιστού με το Κέλυφος Bash

Σε αυτή την ενότητα της εκφώνησης θα δούμε πως μπορεί κανείς να διεκπεραιώσει μια αίτηση ανάκτησης ιστοσελίδας του WWW μέσω του κελύφους Bash. Στόχος μας δεν είναι να επεξηγήσουμε σε βάθος το πρωτόκολλο μεταφοράς υπερκειμένου HTTP ή την γλώσσα σήμανσης δεδομένων HTML, διότι αυτά είναι το αντικείμενο μελέτης άλλων μαθημάτων. Επιπλέον, δεν απαιτείται να γνωρίζετε τις ακριβείς λειτουργίες των πρωτοκόλλων που χρησιμοποιούνται για να διεκπεραιώσετε την άσκηση.
Ας δούμε λοιπόν τι γίνεται όταν κάποιος θελήσει να ανακτήσει μια ιστοσελίδα (ένα αρχείο του ιστοχώρου που τοποθετήσατε στο DocumentRoot του Web server σας) από το κέλυφος bash **χωρίς** την χρήση γραφικού φυλλομετρητή (browser). Τοποθετήστε στον κατάλογο /var/www/html το αρχείο test.html με το πιο κάτω περιεχόμενο:

```
<!DOCTYPE html>
<html>
  <head>
    <title>EPL421 Webpage</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

```
# Άνοιξε ένα τερματικό και εκτέλεσε την ακόλουθη εντολή, η οποία ανοίγει ένα tcp socket με τον εξυπηρετητή apache που βρίσκεται πάνω στο VM (στη διεύθυνση localhost ή 127.0.01), ακούει στη θύρα 80, για ανάγνωση/γραφή.
exec 5<>/dev/tcp/localhost/80

# Απόστειλε μια αίτηση για ανάκτηση της σελίδας /index.html
echo -e "GET /test.html HTTP/1.1\r\nHost: localhost\r\nConnection: close\r\n\r\n" >&5
# Εκτύπωσε το αποτέλεσμα στην οθόνη
cat <&5
```

Αυτό θα επιστρέψει και θα εκτυπώσει στην οθόνη το ακόλουθο κείμενο

HTTP/1.1 200 OK Date: Mon, 30 Jan 2023 20:33:33 GMT Server: Apache/2.4.41 (Ubuntu) Last-Modified: Wed, 25 Jan 2023 08:02:57 GMT ETag: "7e-5f38133a32af0" Accept-Ranges: bytes Content-Length: 126 Connection: close Content-Type: text/html	**HTTP Header**
<!DOCTYPE html> <html> <head> <title>EPL421 Webpage</title> </head> <body> Hello World </body> </html>	**Content**

Στην συνέχεια να κλείσετε το input/output redirection, για να ελευθερώσετε τον File Handler #5.

```
# Κλείσε το output redirection για το socket
exec 5>&-
# Κλείσε το input redirection για το socket
exec 5<&-
```

Όπως είδαμε πιο πάνω, ανακτήσαμε μαζί με το ζητούμενο αρχείο test.html (Content) και ένα HTTP Header. Το Header εκφράζει διάφορες μέτα-πληροφορίες, όπως για παράδειγμα, πότε δημιουργήθηκε το αρχείο, πόσο μεγάλο είναι κτλ. Από το HTTP Header, μας ενδιαφέρουν τα ακόλουθα:

1. Εάν βρούμε το κωδικό 200 (δηλαδή «HTTP/1.1 200 OK»), τότε η σελίδα υπάρχει στον server και επιστρέφεται κάτω από το header.
2. Εάν βρούμε οποιονδήποτε άλλο κωδικό (π.χ., HTTP/1.1 404 Not Found), τότε η σελίδα έχει κάποιο άλλο πρόβλημα (π.χ., δεν υπάρχει πλέον). Σε αυτή την περίπτωση πρέπει να προσθέσετε το URL σε ένα αρχείο (**brokenurls.txt**). Αυτό είναι χρήσιμο γιατί μπορεί κανείς να γνωρίζει στο τέλος της εκτέλεσης, όλα τα URLs τα οποία αναφέρονται μέσω του συγκεκριμένου δικτυακού χώρου και τα οποία δεν είναι προσβάσιμα πλέον (αυτό για λόγους συντήρησης του ιστόχωρου).
3. Τα "Content-Type: text/html" και "Content-Type: text/plain" υποδηλώνουν ότι πρόκειται για περιεχόμενο τύπου HTML και TEXT αντίστοιχα. Στην άσκηση θα πρέπει να επικεντρωθείτε μόνο στα δυο πιο πάνω ήδη πληροφορίας. **Οποιοδήποτε άλλο content-Type θα πρέπει να αγνοείται.**

Τα HTTP headers δεν χρειάζεται να αποθηκεύονται.
Γνωρίζοντας τώρα πως μπορείτε να ανακτήσετε μια ιστοσελίδα στον τοπικό δίσκο, θα προχωρήσουμε στην περιγραφή της αναμενόμενης λειτουργίας του συστήματος.

V. Περιγραφή Λειτουργίας Συστήματος

Το σύστημα θα πρέπει να υλοποιεί τις εξής δυο λειτουργίες:

A) Αράχνη (Crawler)

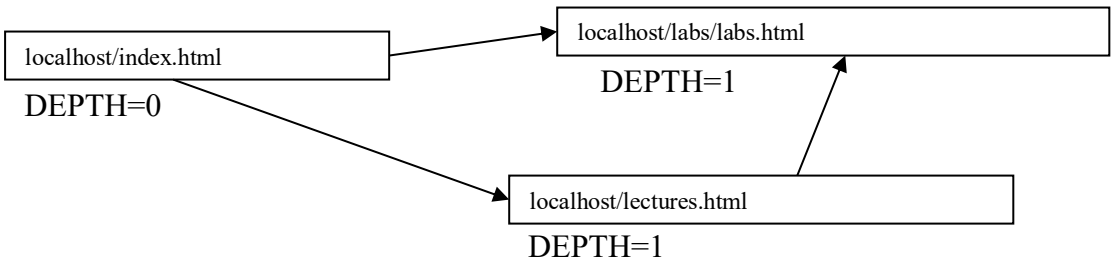
Η εντολή θα εκτελείται όπως φαίνεται παρακάτω:

```
./crawler URL DEPTH
```

όπου URL είναι η διεύθυνση του Web server στον οποίο είναι αποθηκευμένος ο ιστοχώρος που θα κάνετε crawling, και DEPTH το βάθος εξερεύνησης (δείτε πιο κάτω). Πρέπει να δίνεται μήνυμα λάθους όταν η εντολή δεν καλείται σωστά.

Για σκοπούς πειραματισμού, η δοσμένη διεύθυνση URL που θα κτυπά ο crawler σας να είναι η τοπική διεύθυνση του VM σας π.χ. 127.0.0.1 ή localhost. Μπορείτε να κατεβάσετε[2] στο DocumentRoot του Web server το περιεχόμενο ενός template για κτίσιμο ιστοσελίδων μαθημάτων που βρίσκεται εδώ (https://github.com/dmsl/course-responsive-template/archive/refs/heads/master.zip). Αποσυμπιέστε το zip file και τοποθετήστε τα όλο το περιεχόμενο του folder course-responsive-template-master (π.χ. index.htm, links.html, css/, fonts/, κτλ.) μέσα στο /var/www/html/.

Η αράχνη (crawler) θα πρέπει να κάνει μια **κατά πλάτος διερεύνηση (breadth-first-traversal)** του γράφου που εκφράζει τον ιστοχώρο που φιλοξενείται στο δοθέν URL. Δηλαδή, η αράχνη θα επισκέπτεται τις ιστοσελίδες επίπεδο-επιπεδο μέχρι να ξεπεράσει το βάθος DEPTH, όπου DEPTH είναι παράμετρος που προσδιορίζεται από τον χρήστη. Όπως φαίνεται από την πιο κάτω εικόνα, DEPTH=0 είναι το πρώτο αρχείο που θα κατεβεί (συνήθως το index.html) και DEPTH=1 όσοι σύνδεσμοι είναι μέσα στο πρώτο αρχείο που θα κατεβεί, DEPTH=2 είναι όσοι σύνδεσμοι είναι μέσα στα αρχεία του DEPTH=1 κ.ο.κ. Οποιοσδήποτε σύνδεσμος δεν εμπίπτει κάτω από τον πιο πάνω URL και οδηγεί σε άλλα domain names θα πρέπει να αγνοείται (π.χ.,http://moodle.cs.ucy.ac.cy/course/info.php?id=42, https://www.cs.ucy.ac.cy/courses/EPL222, http://goo.gl/cPEf7j, κτλ.). Ο λόγος που βάζουμε αυτό τον περιορισμό είναι για να μειώσουμε τον αριθμό τον σελίδων που ανακτά η αράχνη σας.



Για να διεκπεραιωθεί η προσπέλαση των ιστοσελίδων θα πρέπει προφανώς να έχετε κάποιο τρόπο να εξάγετε τους συνδέσμους (links) από μια ιστοσελίδα. Ένας σύνδεσμος είναι μια διεύθυνση κάποιας άλλης ιστοσελίδας. Για να εντοπίσετε ένα σύνδεσμο σε μια σελίδα πρέπει να εντοπίσετε το href= στο html κείμενο όταν περιέχεται σε σημαντήρα (tag) τύπου <a> ή <link>, όπως για παράδειγμα:

```
<a href="links.html">Links</a>
<link rel="stylesheet" href="css/font-awesome.min.css"/>
```

ή το src= στο html κείμενο όταν περιέχεται σε σημαντήρα τύπου <script>, όπως για παράδειγμα:

```
<script src="js/bootstrap.min.js"></script>
```

[2] Για τη δουλειά αυτή μπορείτε να χρησιμοποιήσετε την εντολή wget μέσω του terminal.

Οι σύνδεσμοι αυτοί μπορεί να είναι *απόλυτοι* (π.χ., `localhost/css/bootstrap.css`) ή *σχετικοί*, ως προς τα folders του Web server (π.χ., `css/bootstrap.css`). Σημειώστε ότι είναι απαραίτητο ο σύνδεσμος να εσωκλείεται μέσα σε μονά ή διπλά εισαγωγικά. Γενικότερα, επιτρέπονται και άλλοι χαρακτήρες (https://www.w3.org/TR/html4/intro/sgmltut.html#h-3.2.2) αλλά είναι εκτός του πεδίου εφαρμογής της παρούσας άσκησης. Επίσης, δεν είναι απαραίτητο να βρίσκεται σε μια γραμμή ένας σύνδεσμος (μπορεί να είναι διασπασμένο σε περισσότερες γραμμές)
π.χ.,

```
<a href="labs/
   labs.html"
  >Labs</a>
```

Περιορισμοί Crawler

1. Το βάθος της διερεύνησης DEPTH, καθώς επίσης ο σύνδεσμος εκκίνησης URL, πρέπει να δίδονται από τον χρήστη σαν ορίσματα εντολής της αράχνης (command-line arguments).
2. Το σύστημα ανακτά μόνο σελίδες με **περιεχόμενο** HTML (π.χ. .html, .htm) ή TEXT (π.χ. css, js). Οι υπόλοιπες ιστοσελίδες αγνοούνται.
3. Καμιά ιστοσελίδα δεν ανακτάται από τον εξυπηρετητή περισσότερο από μια φορά, κατά την διάρκεια κάθε κύκλου ανάκτησης και επεξεργασίας.
4. Οι ιστοσελίδες πρέπει να αποθηκεύονται στο /tmp/$USER/data με την χρήση δομημένων καταλόγων, οι οποίοι είναι πανομοιότυποι στη δομή με τους καταλόγους του Web server. Δηλαδή π.χ.,
   /tmp/$USER/data/localhost/index.html
   /tmp/$USER/data/localhost/labs/labs.html
   /tmp/$USER/data/localhost/css/bootstrap.css
   ….
5. Το brokenurl.txt περιέχει όλα τα URLs τα οποία δεν είναι προσβάσιμα (λαμβάνουμε από το Web server το μήνυμα 404 Not Found). Το αρχείο έχει την μορφή :
   localhost/indexxxxx.html
   localhost/indexxYYxx.html
   ….

Β) Αναλυτής Δεδομένων

Η εντολή θα εκτελείται όπως φαίνεται παρακάτω:

```
./analyzer [options]
```

και ανάλογα με τα options θα γίνεται ανάλυση των αρχείων που έχουν αποθηκευτεί τοπικά στο /tmp/$USER/data με τις εντολές του UNIX που διδαχθήκατε στο μάθημα. Πρέπει να δίνεται μήνυμα λάθους όταν η εντολή δεν καλείται σωστά.

Options:

(α) tokenizer-stopwords-removed <filename>
Η επιλογή αυτή θα είναι ένα φίλτρο το οποίο επεξεργάζεται τις ανακτημένες ιστοσελίδες με το δοθέν filename (μόνο για αρχεία τύπου .html ή .htm) και εξάγει μια λίστα με όλες τις λέξεις (tokenize) και τις συχνότητες εμφάνισης τους σε ταξινομημένη σειρά από τη πιο συχνή στην πιο σπάνια, που πληρούν τα πιο κάτω:

- Από το αρχείο μας ενδιαφέρει μόνο ότι συμπεριλαμβάνεται στο body της ιστοσελίδας (μεταξύ των tags <body> και </body>.

- Θα αφαιρούνται τα HTML tags και οι ειδικοί χαρακτήρες HTML, όπως αυτά περιγράφονται πιο κάτω:

  1. **HTML TAG**
     Οτιδήποτε περικλείεται μεταξύ των συμβολών < >.
     π.χ., <html> <a href=ssss>, <td bgcolor="red" width="100%">

     Σε αυτή την κατηγορία περιλαμβάνονται και τα HTML σχόλια. Ένα σχόλιο ξεκινά με <!-- και τερματίζει με -->

     ```
     <!--  This is a
     Multiline comment
     Var[0];
     This should be ignored by your analysis
     -->
     ```

  2. **HTML Ειδικοί Χαρακτήρες**
     Οτιδήποτε περικλείεται μεταξύ & και ; π.χ., ***& &nsbp; Á*** Οι ειδικοί χαρακτήρες ΔΕΝ πρέπει να περιλαμβάνονται στην ανάλυση.
- Θα αφαιρούνται (δεν θα λαμβάνονται υπόψιν) οι αριθμοί.
- Με σκοπό τη βελτίωση της αναζήτησης, ορισμένες μηχανές αναζήτησης εξαλείφουν συνηθισμένες λέξεις που ονομάζονται stop words. Από τη δική σας λίστα θα αφαιρούνται οι λέξεις (της ιστοσελίδας) οι οποίες θεωρούνται stop words. Το πρόγραμμά σας θα βρίσκει τα stop words από το ένα αρχείο με το όνομα stopwords.txt (δίνεται στο as2-supplementary.zip).
- Οι λέξεις μέσα στη λίστα δεν κάνουν διάκριση μεταξύ πεζών και κεφαλαίων γραμμάτων.

Η λίστα θα αποθηκεύεται στο αρχείο /tmp/$USER/tokenizer.txt. Το αρχείο θα είναι της μορφής (ταξινομημένο ως προς τη συχνότητα εμφάνισης των λέξεων):

12 hello
7 world
3 computer

(β) show-outgoing-links
Η επιλογή αυτή θα κάνει εξαγωγή όλων των συνδέσμων (broken και non-broken) από κάθε αρχείο τύπου html ή htm και θα τα αποθηκεύει σε άλλο αρχείο /tmp/$USER/outgoinglinks.txt. Οι έγκυροι σύνδεσμοι είναι αυτοί που περιέχονται σε href= μέσα σε σημαντήρα τύπου <a> και παραπέμπουν σε άλλο διαδικτυακό τόπο. Για παράδειγμα τα http://goo.gl/8noJMv και schedule.html είναι έγκυροι διαδικτυακοί σύνδεσμοι (απόλυτος και σχετικός αντίστοιχα) ενώ το syllabus.pdf δεν είναι. Για παράδειγμα, μέσα στο αρχείο outgoinglinks.txt θα έχει γραμμές τις πιο κάτω μορφής :

index.html -> http://goo.gl/8noJMv, http://goo.gl/DLLk2x, http://www.cs.ucy.ac.cy/~csp5pa1/, http://goo.gl/a1zSFA, schedule.html

(γ) extract-keywords <filename>
Η επιλογή αυτή θα επεξεργάζεται τις ανακτημένες ιστοσελίδες με το δοθέν filename (μόνο για αρχεία τύπου .html ή .htm) και εξάγει μια λίστα με όλες τις λέξεις κλειδιά (keywords) σε ταξινομημένη αλφαβητικά σειρά που υπάρχουν μέσα στο εν λόγω αρχείο, που πληρούν τα πιο κάτω:

- Από το αρχείο μας ενδιαφέρει μόνο ότι συμπεριλαμβάνεται στο body της ιστοσελίδας (μεταξύ των tags <body> και </body>.
- Θα αφαιρούνται τα HTML tags και οι ειδικοί χαρακτήρες HTML, όπως αυτά περιεγράφηκαν πιο πάνω στο πρώτο option

Για την εξαγωγή των keywords θα επικοινωνείτε με το OpenAI API μέσω της εντολής curl. Δείτε πιο πολλές πληροφορίες εδώ: https://platform.openai.com/examples/default-keywords. Μπορείτε να χρησιμοποιήσετε τις ίδιες τιμές παραμέτρων (model, temperature, max_tokens κτλ) που φαίνεται στο παράδειγμα του πιο πάνω συνδέσμου.
Η λίστα θα αποθηκεύεται στο αρχείο /tmp/$USER/filename-keywords.txt όπου filename είναι το όνομα του αρχείου που έδωσε ο χρήστης σαν παράμετρο στην επιλογή αυτή.

VI. Γενικοί Κανόνες

1. Το σύστημα δεν αφήνει ποτέ άχρηστα και μεταβατικά αρχεία στον δίσκο, ανεξάρτητα εάν διακοπεί η λειτουργία του προγράμματος από το κλείσιμο του κελύφους. Δείτε εντολή trap.
2. Το σύστημα πρέπει να χρησιμοποιεί τεχνικές δομημένου προγραμματισμού με την χρήση συναρτήσεων.
3. Το σύστημα πρέπει να ελαχιστοποιεί την χρήση πόρων του συστήματος (αρχεία, μνήμης, κτλ).
4. Το σύστημα πρέπει να μειώνει όσο το δυνατό περισσότερο τον χρόνο διεκπεραίωσης της ανάκτησης και επεξεργασίας των δεδομένων.

Σημειώστε ότι η πιο πάνω περιγραφή θα σας επιτρέψει να δημιουργήσετε ένα σχετικά απλό crawler. Ένας πραγματικός crawler ωστόσο, είναι ένα πολύ-σύνθετο λογισμικό το οποίο πρέπει να λαμβάνει υπόψη μια πλειάδα άλλων παραμέτρων. Όπως αναφέρουν οι δημιουργοί του Google, Sergey Brin και Larry Page, στην δημοσίευση τους «*The Anatomy of a Large-Scale Hypertextual Web Search Engine» (1998):*

"Running a web crawler is a challenging task. There are tricky performance and reliability issues and even more importantly, there are social issues. Crawling is the most fragile application [in our search engine,] since it involves interacting with hundreds of thousands of web servers and various name servers which are all beyond the control of the system.

Καλή Επιτυχία !