# Department of Computing

**CS213: Advanced Programming**

**Class: BSCS – 4ABC**

# Lab 5: Local Spider

**Date: October 24, 2016 / October 27, 2016**

**Time: Monday (02:00PM – 05:00PM) / Thursday (10:00 AM – 01:00 PM & 02:00 PM - 05:00 PM)**

# Instructor: Fahad Ahmed Satti

# Lab 5: Local Spider

## Introduction

In this lab, students will create a spider (also known as crawler) application capable of indexing the local file system. The spider should crawl through the file system, creating an easily searchable index of file paths, file names & contents (of text files only). The user should then be able to search this index against keyword, getting back all the file paths where the keyword was encountered.

## Objectives

After performing this lab students will be able to understand:

* Spiders
* Index building and searching
* File Handling

## Tools/Software Requirement

* Preferred Programming Language: C++, Java or Python
* Any IDE
* SVN – Github

## Description

**Crawling** means extracting content without processing the results. Crawlers are rather dumb processes that fetch content supplied by Web servers answering (HTTP) requests of requested URIs, delivering those contents to other processes, e.g. crawling caches or directly to indexers.

On the other Hand **indexing** means making sense out of the retrieved contents, storing the processing results in a (more or less complex) document index. E.g. Link analysis is a way to measure URI importance, popularity, trustworthiness. Link analysis is often just a helper within the indexing process. This indexing is used to quickly fetch results against some user query.

Applying the different concepts covered during the last few lectures and your previous courses, look for a simple and smart solution. You are more than welcome to search on the WWW to look for more resources and help.

### How does Google do it?[[1]](#footnote-1)

### When you sit down at your computer and do a Google search, you're almost instantly presented with a list of results from all over the web. How does Google find web pages matching your query, and determine the order of search results? The main process that are involved in delivering search results are:

### Crawling

### Indexing

### Serving

### Crawling

### Crawling is the process by which Googlebot discovers new and updated pages to be added to the Google index. Google uses a huge set of computers to fetch billions of pages on the web. The program that does the fetching is called Googlebot. Googlebot uses an algorithmic process: computer programs that determines which sites to crawl, how often, and how many pages to fetch from each site and many more.

### Indexing

### Googlebot processes each of the pages it crawls in order to compile a massive index of all the words it sees and their location on each page. Then google processes information included in key content tags and attributes, such as Title tags. Googlebot can process many, but not all, content types. For example, it cannot process the content of some rich media files or dynamic pages.

### In the similar way your program (spider) is going to search the local file system quickly and also allows the user to search for a file quickly. Indexing can be done through many ways. Some of them can be

### • Hash Maps

### • Binary Search trees

### • Maps

### Serving results

### When a user enters a query, Google machines search the index for matching pages and return the results that are the most relevant to the user. Relevancy is determined by over 200 factors, one of which is the PageRank for a given page. PageRank is the measure of the importance of a page based on the incoming links from other pages.

## Lab Task

Your task will be to implement the following:

1. Creating a local file system crawler.
2. Index paths, names and readable/searchable contents (No exe files) for each file.
3. Allow the user to search for a keyword, in the index.
4. Find out all the files, where the keyword is being used in its path, name or content.
5. List the found files.
6. Unit tests to evaluate your code.
7. Using a Version Control System (VCS) to manage your solutions.

## Deliverables

Following the guidelines set out in your course outline, your submission must include the following:

1. Unit tests
2. A description document with separate sections on Introduction, How to run your application, Link to public GitHub repo with your submission and any other data.
3. Original Source Code

Convert your submission files to a zip folder and name it as given below and upload the zip folder on LMS.

Name – Registration No. – Section

## Grade Criteria

This lab is graded. Min marks: 0. Max marks: 10.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Minimum** | **Maximum** |
| Documentation with clearly defined understanding of the lab task and approach | Fail | Pass |
| Code clarity | 0 | 1 |
| Github (SVN) | 0 | 1 |
| Unit Tests | 0 | 1 |
| Viva | 0 | 3 |
| Lab Task | 0 | 4 |

1. https://support.google.com/webmasters/answer/70897?hl=en [↑](#footnote-ref-1)