Федеральное агентство связи

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической Кибернетики и Информационных Технологий

Отчет по лабораторной работе

по предмету «Кроссплатформенное программирование»

на тему:

«Модифицированный web–сканер»

Выполнил: студент группы

БВТ1802

Дворянинов Павел Владимирович

Руководитель:

Ксения Андреевна Полянцева

Москва 2020

Цель работы

Изучить работу многопоточного web–сканера.

Задание

Написать программу «Web–сканер» для поиска всех ссылок, которые находятся на web–странице, в пределах указанной глубины поиска. URL–адрес, количество потоков и глубина поиска задаются в аргументах программы.

Выполнение

Class CrawlerTask.java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;

public class CrawlerTask implements Runnable {
    final static int AnyDepth = 0;
    private URLPool Pool_se;

    /** Нет "/" для поддержки https */
    private String Prefix_se = "http";

    @Override
    public void run() {
        try {
            Scan();
        } catch (IOException | InterruptedException e) {
            e.printStackTrace();
        }
    }

    public CrawlerTask(URLPool pool) {
        Pool_se = pool;
    }

    private  void  Scan() throws IOException, InterruptedException {
        while (true) {
            Process(Pool_se.get());

        }
    }

    private void Process(URLDepthPair pair) throws IOException{
        /** Установка соединения и перенаправление */
        URL url = new URL(pair.getURL());
        URLConnection connection = url.openConnection();

        String redirect = connection.getHeaderField("Местоположение");
        if (redirect != null) {
            connection = new URL(redirect).openConnection();
        }
        Pool_se.addProcessed(pair);
```

```
        if (pair.getDepth() == 0) {
            return;
        }

        BufferedReader reader = new BufferedReader(new
         InputStreamReader(connection.getInputStream()));
        String input;
        while ((input = reader.readLine()) != null) {
            while (input.contains("a href=\"" + Prefix_se)) {
                input = input.substring(input.indexOf("a href=\"" + Prefix_se) + 8);
                String link = input.substring(0, input.indexOf('\"'));
                if(link.contains(" ")){
                    link = link.replace(" ", "%20");
                }
                /** Не обрабатывает посещение одинаковых ссылок */
                if (Pool_se.getNotProcessed().contains(new URLDepthPair(link,
                 AnyDepth)) ||
                        Pool_se.getProcessed().contains(new URLDepthPair(link,
                 AnyDepth))) {
                    continue;
                }
                Pool_se.addNotProcessed(new URLDepthPair(link, pair.getDepth() - 1));
            }
        }
        reader.close();
    }
}
```

Class URLDepthPair.java

```
import java.util.Objects;

public class URLDepthPair {
    private String Url_se;
    private int Depth_se;

    public URLDepthPair(String host, int depth) {
        Url_se = host;
        Depth_se = depth;
    }

    public String getURL() {
        return Url_se;
    }

    public int getDepth() {
        return Depth_se;
    }

    @Override
    public boolean equals(Object obj) {
        if (obj instanceof URLDepthPair) {
            URLDepthPair o = (URLDepthPair)obj;
            return this.Url_se.equals(o.getURL());
        }
        return false;
    }
```

```
    @Override
    public int hashCode() {
        return Objects.hash();
    }
}
```

Class URLPool.java

```
import java.util.LinkedList;

public class URLPool {
    private LinkedList<URLDepthPair> Processed_se = new LinkedList<URLDepthPair>();
    private LinkedList<URLDepthPair> NotProcessed_se = new
        LinkedList<URLDepthPair>();
    private int Depth_se;
    private int Waiting_se;
    private int Threads_se;

    public URLPool(String url, int depth, int threads) {
        NotProcessed_se.add(new URLDepthPair(url, depth));
        Depth_se = depth;
        Threads_se = threads;
    }

    public synchronized URLDepthPair get() throws InterruptedException {
        if (isEmpty()) {
            Waiting_se++;
            if (Waiting_se == Threads_se) {
                getSites();
                System.exit(0);
            }
            wait();
        }
        return NotProcessed_se.removeFirst();
    }

    public synchronized void addNotProcessed(URLDepthPair pair) {
        NotProcessed_se.add(pair);
        if (Waiting_se > 0) {
            Waiting_se--;
            notify();
        }
    }

    private boolean isEmpty() {
        if (NotProcessed_se.size() == 0) {
            return true;
        }
        return false;
    }

    public void getSites() {
        System.out.println("Глубина поиска: " + Depth_se);
        for (int i = 0; i < Processed_se.size(); i++) {
            System.out.println(Depth_se - Processed_se.get(i).getDepth() + " " +
        Processed_se.get(i).getURL());
        }
        System.out.println("Посещённые ссылки: " + Processed_se.size());
    }
```

```
    public void addProcessed(URLDepthPair pair) {
        Processed_se.add(pair);
    }

    public LinkedList<URLDepthPair> getProcessed() {
        return Processed_se;
    }

    public LinkedList<URLDepthPair> getNotProcessed() {
        return NotProcessed_se;
    }
}
```

Class ScannerApp.java

```
import java.io.IOException;

public class ScannerApp {
    public static void main(String args[]) throws IOException, InterruptedException {
        URLPool url_pool = new URLPool(args[0], Integer.parseInt(args[1]),
         Integer.parseInt(args[2]));
        for (int i = 0; i < Integer.parseInt(args[2]); i++) {
            CrawlerTask search_url = new CrawlerTask(url_pool);
            new Thread(search_url).start();
            System.out.println("Поиск " + i + " запущен");
        }
    }
}
```

Результат работы

Ссылка: https://www.kia.ru/, глубина поиска: 1, количество потоков: 5

```
Поиск 0 запущен
Поиск 1 запущен
Поиск 2 запущен
Поиск 3 запущен
Поиск 4 запущен
Глубина поиска: 1
0 https://www.kia.ru/
1 https://www.kia.ru/press/news/103/
1 https://www.kia.ru/press/news/113/
1 https://www.kia.ru/buy/special/
1 https://www.kia.ru/buy/fleet/
1 https://youtu.be/tdJ7VdWGMuY
1 https://www.kia.ru/service/special/
1 https://www.kia.ru/buy/accessories/
1 https://www.kia.ru/about/info/
1 https://www.kia.ru/kiamotorsrus/magazine/
1 https://www.kia.ru/used_cars/
1 https://kia.ru/kiamotorsrus/vacancies/
1 https://www.kia.ru/personal/loyalty/
1 https://vk.com/kiamotorsrus
1 https://ok.ru/group/57054967365668
1 https://www.youtube.com/user/KIAMotorsRussia
1 https://www.instagram.com/kiamotorsrussia/
1 https://zen.yandex.ru/kia
1 https://www.kia.ru/request/order_to/
1 https://www.facebook.com/kiamotorsrus
Посещённые ссылки: 20
```

Ссылка: https://www.kia.ru/, глубина поиска: 2, количество потоков: 5

```
2 https://www.facebook.com/skulinarii/?ref=py_c
2 https://www.facebook.com/QuorisArt/
2 https://www.facebook.com/kia/
2 https://www.facebook.com/AustralianOpen/
2 https://www.facebook.com/podarizhizn/
2 https://www.facebook.com/kia.aljabr/
2 https://www.facebook.com/KiaMotorsMozambique/
2 https://www.facebook.com/KIA.Motors.Thailand/
2 https://www.facebook.com/cartimes/
2 https://www.facebook.com/kiaofficial.kz/
2 https://www.facebook.com/pages/Kia-Optima/104055846296302
2 https://www.facebook.com/autoplusTV/
2 https://www.facebook.com/kiamotorsmongolia/
2 https://www.facebook.com/KiaLebanon/
2 https://www.facebook.com/KiaMotorsNZ/
2 https://www.facebook.com/KIAPLATINUMCUP/
2 https://www.facebook.com/kiamotorsgreece/
2 https://www.facebook.com/kiapalestine/
2 https://www.facebook.com/privacy/explanation
2 https://www.facebook.com/ad_campaign/landing.php?placement
2 https://www.facebook.com/help/cookies?ref_type=sitefooter
2 https://www.facebook.com/kia.sg/
2 https://www.facebook.com/kiamotorsrus/?hc_ref=ARRywgjK-J2X
 .ARDTb6eE1Vs16mv67IQ8YYGcpuvieYhjm8S9ShyRy2XO4iDL1fli9mW2zB
 -mrX7SMCZ4Esip3n3zjzNSnm2Nli0fU95uyvbMEJqQInXQxhr5hdspCWxMm
 -1pxILO5nmvcfsqE0NyWpa79JZg&amp;__tn__=kC-R
2 https://www.facebook.com/ivi.ru/?ref=py_c
Посещённые ссылки: 114
```

Вывод

Изучили работу многопоточного web–сканера. Написали программу «Web–сканер» для поиска ссылок на web–странице. Многопоточный web–сканер работает быстрее однопоточного web–сканера, что можно заметить в процессе работы программы.