

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 №8
по дисциплине «Технологии разработки программного обеспечения»

Выполнил: студент группы
БВТ1803

Маркушин Андрей Васильевич

Руководитель:

Мосева Марина Сергеевна

Москва 2020

Цель работы: расширить сканер для использования поточной обработки.

Выполнение:

MySocket:

```
import java.io.*;
import java.net.*;
public class MySocket {
    Socket socket;

    MySocket(String host,int port) throws IOException {
        socket = new java.net.Socket(host,port);
        setSoTimeout(5000);
    }

    void setSoTimeout(int timeout)throws IOException {
        socket.setSoTimeout(timeout);
    }

    InputStream getInputStream()throws IOException {
        return socket.getInputStream();
    }

    OutputStream getOutputStream()throws IOException{
        return socket.getOutputStream();
    }

    void close() throws IOException{
        socket.close();
    }
}
```

URLDepthPair:

```
import java.net.*;

public class URLDepthPair {
    private URL url;
    private int depth;

    URLDepthPair(String url, int depth){
        try {
            this.url = new URL(url);
            this.depth = depth;
        }
        catch (MalformedURLException e){
            System.err.println("MalformedURLException: " +
e.getMessage());
        }
    }

    public boolean isURL(){
        return true;
    }
    public String getUrl() {
        return url.getHost();
    }
    public int getDepth() {
        return depth;
    }
    public String toString(){
        return "|" + depth + "|" + url;
    }
}
```

Crawler:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.LinkedList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Crawler {
    URLDepthPair urlDepthPair;
    LinkedList<URLDepthPair> siteList;
    MySocket socket;
    BufferedReader bufferedReader;
    PrintWriter printWriter;
    Pattern regHTTP;
    Crawler (URLDepthPair urlDepthPair) {
        this.urlDepthPair=urlDepthPair;
    }
    LinkedList<URLDepthPair> start() throws IOException{
        try {
            socket = new MySocket(urlDepthPair.getUrl(),80);
            bufferedReader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            printWriter = new PrintWriter(socket.getOutputStream(), true);
            siteList = new LinkedList<URLDepthPair>();
            regHTTP = Pattern.compile("(http:\\/\\/\\/([\\w\\-
\\.!~?&+=\\*'|(),\\/\\/#\\:\\:]+)((?!\\<\\/\\/\\w\\>))*?");
        }
        catch (Exception exc){
            System.out.println(exc);
            return new LinkedList<URLDepthPair>();
        }
        printWriter.println("GET / HTTP/1.1");
        printWriter.println("Host: "+urlDepthPair.getUrl()+":80");
        printWriter.println("Connection: Close");
        printWriter.println();
        try{
            String line;
            while ((line=bufferedReader.readLine())!=null) {
                while(line.contains("<a")){
                    while (line.indexOf(">", line.indexOf("<a"))== -1)
line+=bufferedReader.readLine();
                    String http =
line.substring(line.indexOf("<a"),line.indexOf(">", line.indexOf("<a")));
                    if (http.contains("http://")){
                        Matcher matcher = regHTTP.matcher(http);
                        String url="";
                        if (matcher.find()) {
                            url = matcher.group();
                        }
                        else return new LinkedList<URLDepthPair>();
                        siteList.add(new
URLDepthPair(url,urlDepthPair.getDepth()+1));
                    }
                    line=line.replace(http,"");
                }
            }
        }
        catch (IOException except){
            System.out.println(except);
        }
        socket.close();
        return siteList;
    }
}
```

URLPool:

```
import java.util.LinkedList;

public class URLPool {
    private LinkedList<URLDepthPair> addSite = new LinkedList<>();
    private LinkedList<URLDepthPair> noAddSite = new LinkedList<>();

    private int finalDepth;

    private int waitThreads = 0;

    URLPool(URLDepthPair url, int depth){
        finalDepth=depth;
        noAddSite = new LinkedList<URLDepthPair>();
        addSite = new LinkedList<URLDepthPair>();
        noAddSite.add(url);
    }

    public synchronized int getWaitThreads() {
        return waitThreads;
    }

    public synchronized URLDepthPair getURL(){
        if (noAddSite.size() == 0) {
            try {
                waitThreads++;
                this.wait();
            }
            catch (InterruptedException e) {
                System.err.println("MalformedURLException: " +
e.getMessage());
                return null;
            }
        }
        addSite.add(noAddSite.getFirst());
        return noAddSite.removeFirst();
    }

    public synchronized void addListURL(LinkedList<URLDepthPair> URLs){

        if (URLs.size() !=0){
            if (URLs.getFirst().getDepth()>=finalDepth){
                addSite.addAll(URLs);
            }
            else{
                noAddSite.addAll(URLs);
                for (int countSite=URLs.size(); countSite!=0 &&
waitThreads!=0;countSite-- , waitThreads--){
                    this.notify();
                }
            }
        }
    }

    public LinkedList<URLDepthPair> getSite(){
        return addSite;
    }
}
```

CrawlerTask:

```
import java.io.IOException;

public class CrawlerTask implements Runnable {

    private URLPool urlPool;
    private URLDepthPair urlDepthPair;
    private Crawler crawler;

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

    public void run() {
        while (true) {
            urlDepthPair = urlPool.getURL();
            try {
                crawler=new Crawler(urlDepthPair);
                urlPool.addListURL(crawler.start());
            } catch (IOException e) {
                System.out.println(e);
            }
        }
    }
}
```

main:

```
import java.io.IOException;
import java.util.LinkedList;

public class main {
    public static void main(String[] args) throws IOException {
        URLPool pool = new URLPool(new
URLDepthPair(args[0],0),Integer.parseInt(args[1]));
        LinkedList<Thread> threadList = new LinkedList<>();

        int countThread = Integer.parseInt(args[2]);

        for (int i=0;i<countThread;i++){
            CrawlerTask crawlerTask = new CrawlerTask(pool);
            threadList.add(new Thread(crawlerTask));
            threadList.getLast().start();
        }

        while (pool.getWaitThreads() !=countThread){}

        LinkedList<URLDepthPair> site=pool.getSite();

        for(URLDepthPair url: site){
            System.out.println(url);
        }

        for(Thread thread: threadList){
            thread.stop();
        }
    }
}
```

Вывод:

```
|1| http://www.l-morozko.ru/
|1| http://cspo.ru/
|1| http://www.guz.ru/
|1| http://unecon.ru/
|1| http://www.l-morozko.ru/
|1| http://www.detiplus.ru/dl/01\_01.html
|1| http://www.detiplus.ru/dl/02\_01.html
|1| http://www.detiplus.ru/dl/03\_01.html
|1| http://www.detiplus.ru/dl/04\_01.html
|1| http://www.detiplus.ru/dl/05\_01.html
|1| http://www.detiplus.ru/dl/06\_01.html
|1| http://www.detiplus.ru/dl/07\_01.html
|1| http://integration21.ru/
|1| http://www.detiplus.ru/med/01\_01.html
|1| http://www.detiplus.ru/med/02\_01.html
|1| http://www.detiplus.ru/med/03\_01.html
|1| http://www.detiplus.ru/med/04\_01.html
|1| http://www.detiplus.ru/med/06\_01.html
|1| http://www.detiplus.ru/med/07\_01.html
|1| http://www.detiplus.ru/med/08\_01.html
|1| http://www.detiplus.ru/med/09\_01.html
|1| http://www.detiplus.ru/med/10\_01.html
|1| http://www.detiplus.ru/mm/01\_01.html
|1| http://www.l-morozko.ru/
|1| http://www.my-solnyshko.com/
|1| http://www.irad.ru
|1| http://www.vuztut.ru
|1| http://www.college.msk.ru
|1| http://www.moscowschool.ru
|1| http://top100.rambler.ru/top100/
|1| http://top100.rambler.ru/top100/
|1| http://u7097.54.spylog.com/cnt?cid=709754&f=3&p=0
|1| http://top.mail.ru/jump?from=843729
|0| http://www.detiplus.ru
```

Вывод: расширили сканер для использования поточной обработки Java так, чтобы несколько веб-страниц можно было сканировать параллельно.