OOAD LAB8 | Multithreading

Details:

Name : P K Navin Shrinivas

• SRN: PES2UG20CS237

• Section: D

Lab content:

Output screenshot :

```
7.0023a3aed2c16ff8dc2786e9b9a8d2c.jpg input.txt
config.ofg L31_Virtualization.pdf
docker-compose.yml pom.xml
https://raw.githubusercontent.com/NavinShrinivas/gobees/main/README.md
https://raw.githubusercontent.com/NavinShrinivas/gobees/main/docker-compose.yml

→ learnjava java --class-path target/classes FirstPackage.App
                                                                                                                                                                                                                                      README.md
                                                                                                                                     pocker-compose.yml
→ learnjava cat L31_Virtualization.pdf
%POF-1.2
%+签
5 0 obj
 /README.md
ile created: docker-compose.yml

→ learnjava cat input.txt

tttps://i.pinimg.com/750x/7c/e2/3a/7ce23a3aed2c16ff0dc2786e9b9a8d2c.jpg
                                                                                                                                      «/congth 6 0 R/Filter /FlateDecode≫
stream
x♦m♦=o♦0
   learnjawa java --class-path target/classes FirstPackage.App
7ce23a3aed2c16ff8dc2786e9b9a8d2c.jpg
le created: 7ce23a3aed2c16ff8dc2786e9b9a8d2c.jpg
                                                                                                                                      өмө

« Ve$RZe|:48000Peirhe

носоо/8!000Ke.0Z000pe%go-0"40j)о
   learniava cat input.txt
 there, //www.cs.cmu.edu/~419-f06/lectures/L31_Virtualization.pdf

▶ learnjawa java --class-path target/classes FirstPackage.App

/L31_Virtualization.pdf

        ФФВФ\]E0007[v10*,80000/00000000;3000[00008]Aw0*8040"0(500aa7000[H0s.00] 3r0!0070KF00B

        000070KF00B

        000070KF00B

.
ile created: L31_Virtualization.pdf
→ learnjava
                                                                                                                                                                                                     U$_08n0y00Jj0_MNi0`+.00u0(004R0!00T0]IZ400z8sA00D.f0bbm750n0[0a9000k0000FJ00p&ub0@0ef
                                                                                                                                      endobj
6 0 obj
411
                                                                                                                                      411
endobj
13 0 obj
≪/Length 14 0 R/Filter /FlateDecode≫
                                                                                                                                      Stredm
x00VonA00W00M005!0
B00C0;`0$q00g 0T000L 0,00Ky0j0B%55000000I1000000 0`0t0
                                                                                                                                       eNeo>eXeoe!ee 8010000Ce>_00000Hoouec'eoLo^[[Keeemeoq$}`~eFte@;ooet[eA%1g04`ejNo'ooee
                                                                                                                                       ..
*¢ka≎≎≎≎v<şV≎L≎≎≎}JV≎}9,6Wh≎Mv≎≎≎=j≎≎EkB≎≎≎U≎≎5≎f
                                                                                                                                       V{0q-0_00=x#0,0000000000L9Z0P0,000;01060_0Q0 00C000B000.q 0060U\(\tilde{O}\)/0\(\tilde{O}\)
                                                                                                                                                                                                                                ^[Lm0'00f00r0)0A0$
"000=0\000" 09:47 30-Mar-23
[1] 0:zsh-M 1:[tmux]
```

Code:

```
import java.io.File;
import java.io.FileWriter;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
import java.util.Scanner;

class App {
   public static void main(String[] args) {
```

```
try {
         File inputfile = new File("./input.txt");
         String[] url_array;
         if (inputfile.exists()) {
            try {
               Scanner myReader = new
Scanner(inputfile);
               String url = "";
               while (myReader.hasNextLine()) {
                  url += myReader.nextLine() + "\n";
               myReader.close();
               url_array = url.split("\\r?\\n");
            } catch (Exception e) {
               System.out.println(e);
               return;
         } else {
            System.out.println("Input file not found,
please provide one.");
            return;
         for (int i = 0; i < url_array.length; i++) {</pre>
            URL url = new URL(url_array[i]);
            URLContentWriterClass ucwc = new
URLContentWriterClass(url);
            ucwc.start();
      } catch (Exception e) {
         System.out.println(e);
         return;
class URLContentWriterClass extends Thread {
   private URL url;
   URLContentWriterClass(URL url) {
```

```
this.url = url;
   public void run() {
      String[] filesplits = url.getFile().split("/");
      System.out.println("./" +
filesplits[filesplits.length - 1]);
      File myObj = new File("./" +
filesplits[filesplits.length - 1]);
      StringBuilder content = new StringBuilder();
      try {
         if (myObj.createNewFile()) {
            System.out.println("File created: " +
myObj.getName());
         } else {
            System.out.println("File already exists.");
            return;
      } catch (Exception e) {
         System.out.println(e);
         return;
      try {
         URLConnection urlconn = url.openConnection();
         BufferedReader bufferedreader = new
BufferedReader(new
InputStreamReader(urlconn.getInputStream()));
         String line;
         while ((line = bufferedreader.readLine()) !=
null) {
            content.append(line + "\n");
         bufferedreader.close();
         FileWriter fw = new FileWriter("./" +
filesplits[filesplits.length - 1]);
         fw.write(content.toString());
         fw.close();
      } catch (Exception e) {
```

```
System.out.println(e);
    return;
}
}
```

Some points to note about multithreading

•

- It is observed that the main thread is waiting for the children thread to finish without any wait class, this is not expected behaviour.
- Multithreading is achived using a extends on the Thread class.
- This class expects a run method that is overriding the parent run method.
- Once initilized, we do a .start() on the object to spawn the thread, this start method is not overriding and is defined by the parent class.
- This in turn spawns the run() method by default as a new thread.
- Multithreading unlike linear execution first spawns all the threads and lets each thread execute parellely.