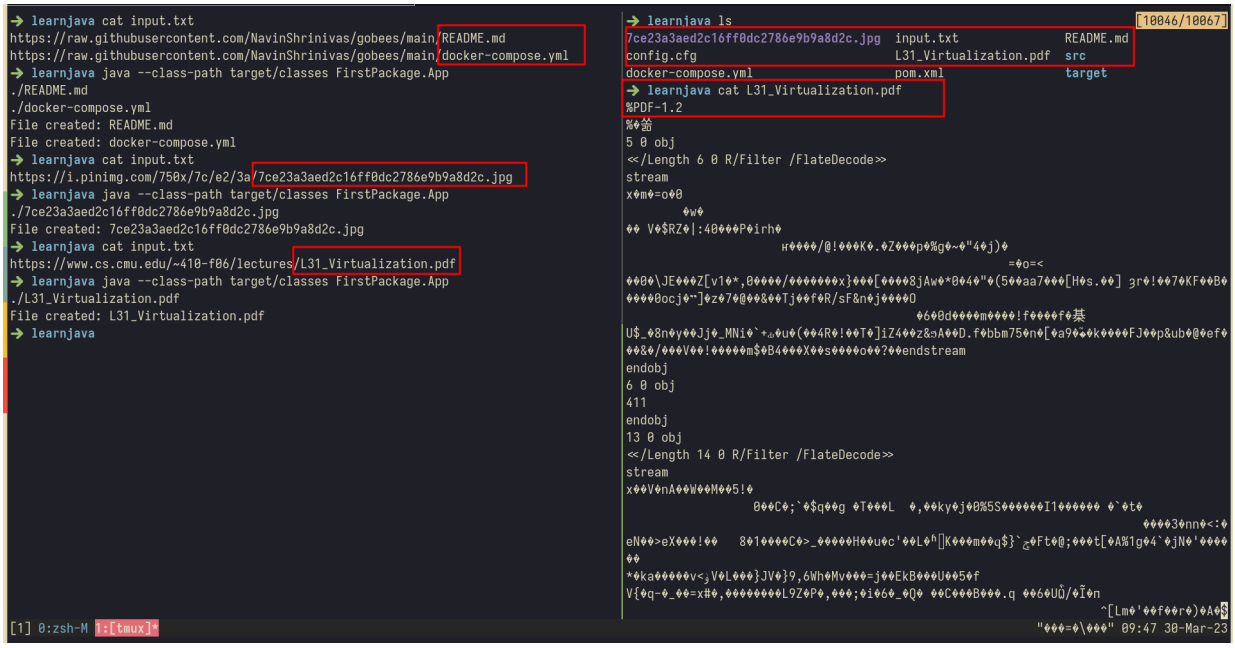
OOAD LAB8 | Multithreading

Details :

- Name : P K Navin Shrinivas
- SRN : PES2UG20CS237
- Section : D

Lab content :

- Output screenshot :



- 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++) {
            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.