Operating Systems Project 3

Kyle Aure – KAure09@winona.edu – Version 1.0, 2019-04-19

Project Description

Create a project that simulates the inner workings of a page replacement.

Course Details

- **Course** - CS405: Operating Systems
- **Instructor** - Dr. Dennis Martin

Project Goals

- Implement both Least Recently Used (LRU) and First In First Out (FIFO) replacement alogrithms.
- Keep track of all page faults.
- Print the state of mainstore after a page fault.

Getting Started

Prerequisites

This is a Java project that uses a Maven build architecure. Prior to running this project you will need to download and install maven on your local system. Instructions on how to install maven can be found here: https://maven.apache.org/install.html

To check to see if you have Maven installed correctly run the `mvn --version` command. You should see output similar to:

```bash
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T13:33:14-05:00)
Maven home: /Applications/maven
Java version: 1.8.0_161, vendor: Oracle Corporation, runtime:
/Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.14.3", arch: "x86_64", family: "mac"
```

Installing

Since we are using Maven, all you need to do to install this program is to run the `mvn install` command. Running this command will do the following:

1. Download all dependencies.
2. Put dependencies on the project's build path.
3. Runs automated tests.
4. Build the project.
5. Package the project as a -jar.

Running the Project

Once you have installed the project you can run it by using the following command, or by opening the .jar file located in the `/target/` folder that was generated int he last step.

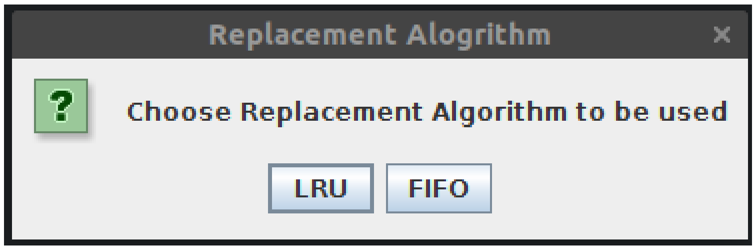
BASH

```
mvn exec:java -Dexec.mainClass="edu.winona.cs.paging.App"
```

User Interface

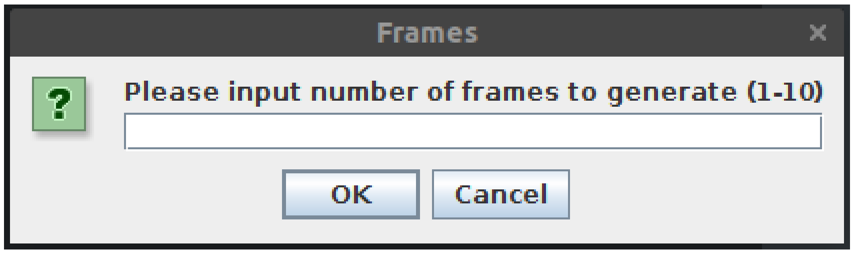
When running this project the user is prompted via a User-Interface (GUI) to choose the settings prior to running.

First the user chooses the Replacement Algorithm they want to use. LRU or FIFO:


Replacement Alogrithm

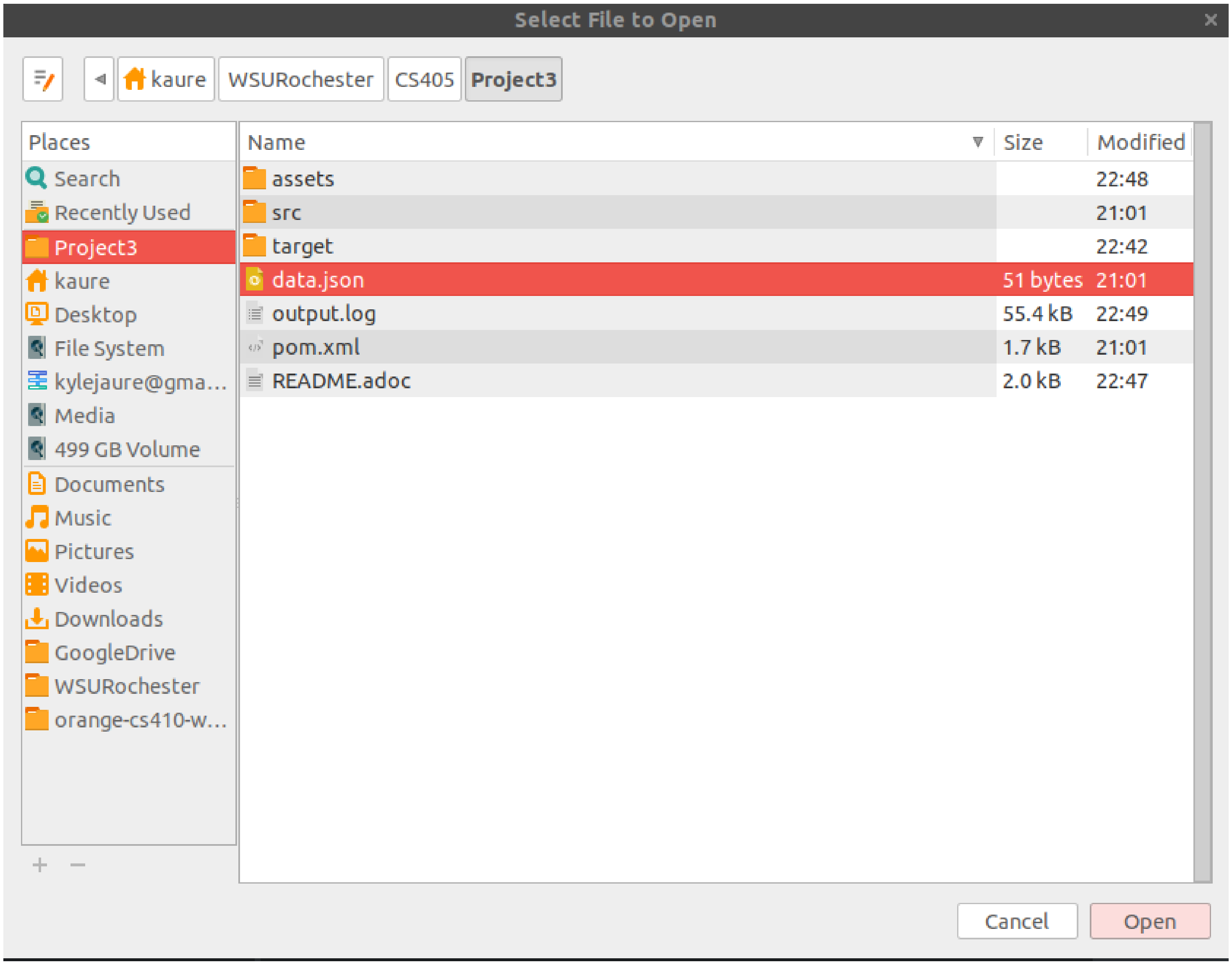
Choose Replacement Algorithm to be used

The user will then need to choose the number of frames in the main store:


Frames

Please input number of frames to generate (1-10)

Finally, the user is asked to choose a .json file that has the page information.



Name	Size	Modified
assets		22:48
src		21:01
target		22:42
data.json	51 bytes	21:01
output.log	55.4 kB	22:49
pom.xml	1.7 kB	21:01
README.adoc	2.0 kB	22:47

Output

Once the settings and file have been selected, the program creates a output.log file. This file holds the output for the program; which includes, algorithm, number of frames, and the main store after faults.

Results

For this project I used the sample data provided here:

```json
{"pages":[1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6]}
```

I have stored the resulting logs in the `/logs/` directory within this projects. Results and links to the resulting logs file here:

# of frames	Faults LRU	Log LRU	Faults FIFO	Log FIFO
3	15	LRU - 3	16	FIFO - 3
4	10	LRU - 4	14	FIFO - 4
5	8	LRU - 5	10	FIFO - 5
6	7	LRU - 6	10	FIFO - 6
7	7	LRU - 7	7	FIFO - 7

Built With

- Maven (https://maven.apache.org/) - Dependency Management
- Jackson (https://github.com/FasterXML/jackson-core) - JSON to POJO

Authors

- **Kyle Aure** - **Author** - KyleAure (https://github.com/KyleAure)

Version 1.0
Last updated 2019-04-19 23:21:55 CDT