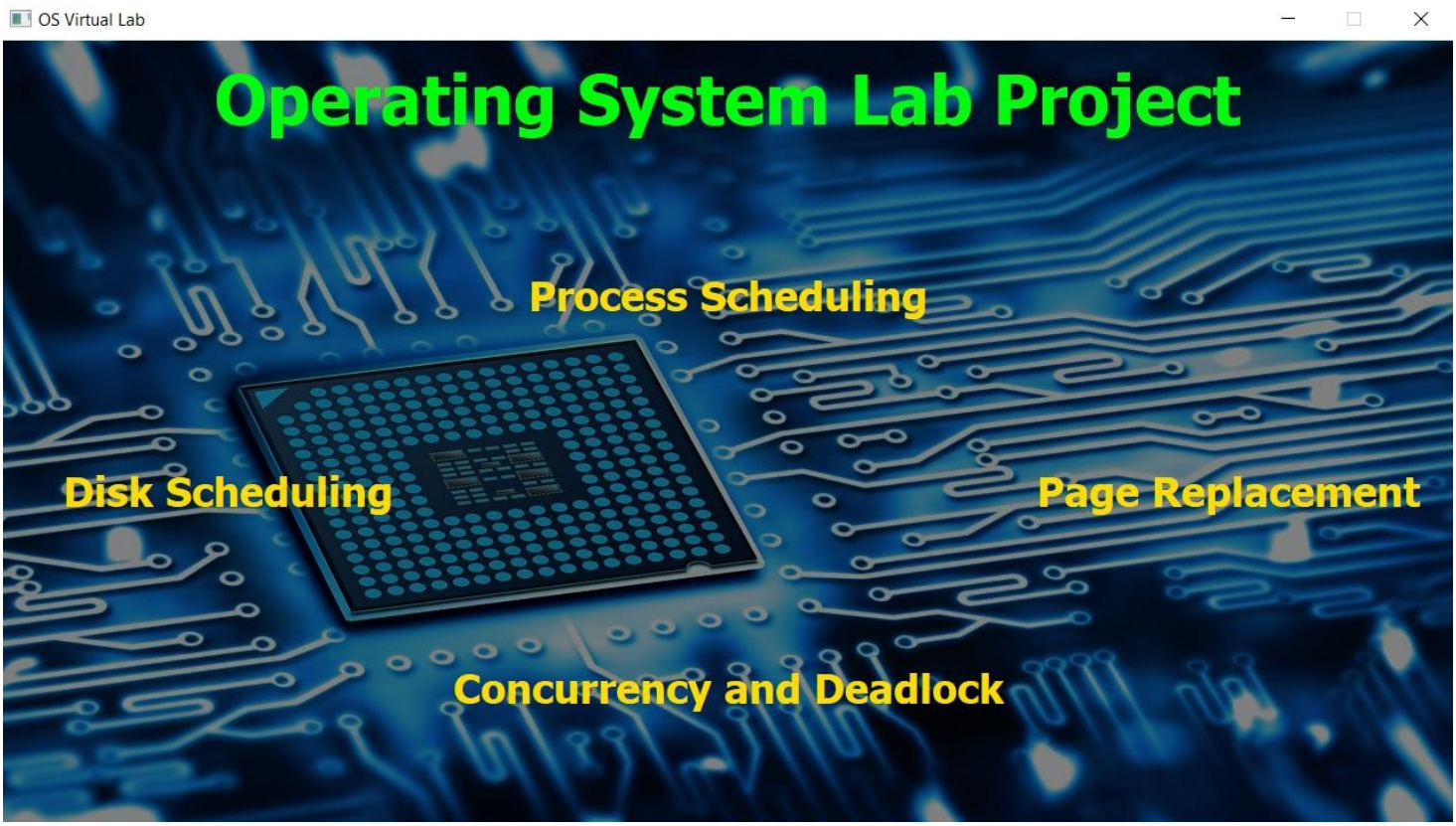


OS Virtual Lab

A Desktop Application to Visualize OS Lab Algorithms

Technical Specifications

Language: Python (with GUI Library - PyQt5)

IDE: VS Code

Installations Required

Python

Python is an interpreted, high-level and general-purpose programming language in which the whole project is designed and developed.

Follow the commands to install Python.

1. Select the version of Python and download Python Executable Installer from https://www.python.org/downloads/
2. Run Executable installer and make sure you select the checkboxes (if any of) '**Install launcher for all users**' and '**Add Python to PATH**'.
3. Verify Python was installed by the following steps:
    - 3.1. Navigate to the directory in which Python was installed on the system. In our case, it is C:\Users\Username\AppData\Local\Programs\Python\Python39 (in last one Python39, the numbers may change according to the version installed).
    - 3.2. Double-click **python.exe**.
    - 3.3. The output should be '**Python version Details**' (version will be like 3.9.3 and details will include time date and system details).
4. Verify Pip was installed by the following steps:
    - 4.1. Open Start menu and type '**cmd**'.
    - 4.2. Select the Command Prompt application.
    - 4.3. Enter > **pip --version** (without '>') in the console.
    - 4.4. If pip was installed successfully, output will be '**pip version from location**' (version will be like 21.0.3 and location will be where it is installed).
    - 4.5. If pip is not installed follow steps mentioned below:
        - 4.5.1. Go to https://bootstrap.pypa.io/get-pip.py. Press right click and select save as and then press enter.
        - 4.5.2. Launch windows command prompt and navigate to the folder where the above file is downloaded.
        - 4.5.3. Type > **python get-pip.py** (without '>') to start pipinstallation.
        - 4.5.4. Check pip is installed by typing > **pip --version** (without '>') and it should return the current version of the platform.
5. Add Python Path to Environment variable by the following steps **(Optional)**:
    - 5.1. Open Start menu and start the run app.
    - 5.2. Type **sysdm.cpl** and click OK. This opens the System Properties window.
    - 5.3. Navigate to the Advanced tab and select Environment Variables.
    - 5.4. Under System Variables, find and select the Path variable.

5.5. Click Edit.

5.6. Click New and paste the path where the python.exe file is located. (Eg: C:\Users\Username\AppData\Local\Programs\Python\Python39)

5.7. Click OK and close all windows.

6. Install virtual env **(Optional)**:

6.1. Open the Start menu and type '**cmd**'

6.2. Select the Command Prompt application.

6.3. Type the following pip command in the console > **pip install virtualenv** (without '>')

PyQt5

PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ **on all supported platforms including iOS and Android**.

The GUI of the application is designed using PyQt5 **QtDesigner**. Follow the commands to install PyQt5.

1. Open '**cmd**'.
2. Run command > **pip install pyqt5** (without '>').
3. Run command > **pip install pyqt5-tools** (without '>').
4. Go to your Python location (Eg: C:\Users\Username\AppData\Local\Programs\Python) and then go to:

   Python38\Lib\site-packages\pyqt5_tools\Qt\bin. The whole path looks like: C:\Users\Username\AppData\Local\Programs\Python\Python38\Lib\sitepacka ges\p yqt5_tools\Qt\bin (Replace Username with the name on your PC).
5. You can add 'C:\Users\Username\AppData\Local\Programs\Python\Python38\Lib\site- packages\p yqt5_tools\Qt\bin', this path to your system environment variable to access '**designer**' from any corner of your PC (Replace Username with the name on your PC).
6. After adding to the environment variable, just open cmd and run > **designer** (without '>') to check successful installation.

If you find this error on opening '**designer**': "**PyQt5 Designer is not working: This application failed to start because no Qt platform plugin could be initialized**".

Follow the below steps to fix it:

1. Go to > Python38\Lib\site-packages\PyQt5\Qt\plugins.
2. In plugins, copy the '**platforms**' folder.
3. Now go to > Python38\Lib\site-packages\pyqt5_tools\Qt\bin.
4. Paste the '**platforms**' folder here. Do copy and replace.
5. Open cmd and run > **designer** (without '>') to check successful installation.

Other Required Libraries

There are some other dependencies which have been used to develop the application in order to provide more functionalities with an ease.

Follow the commands to install the additional libraries.

1. Open '**cmd**'.
2. To install a library, type the command > **pip install library-name** (without '>').
3. Replace library-name with the following list of libraries to install all the required libraries.

Libraries: matplotlib, pandas, xlsxwriter, win32com.

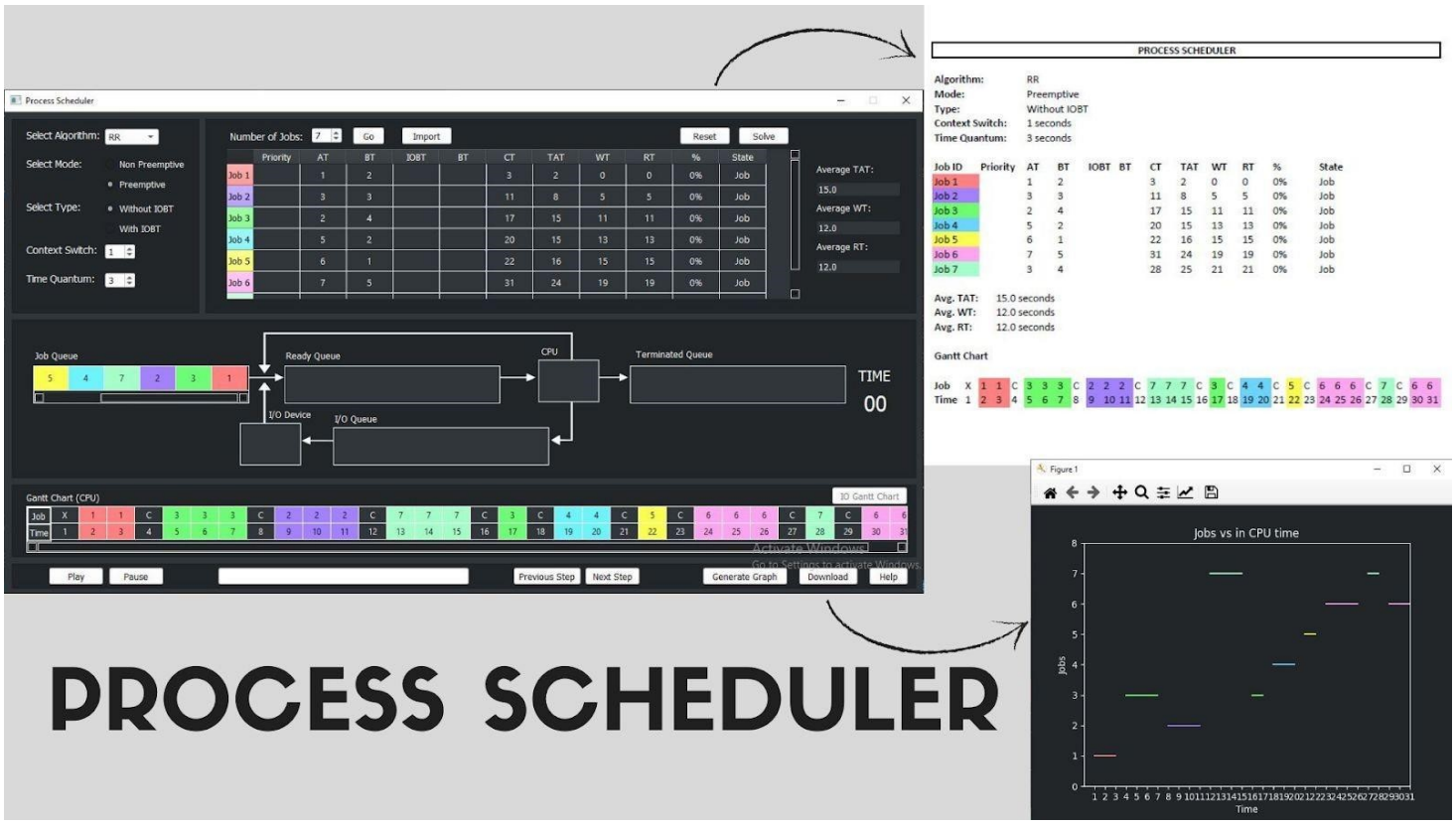
IDE (Optional)

Installation of IDE is not essential to view our project.

However, if you wish to view the code of the project, the following is a link to Visual Studio Code, which is a famous and commonly used IDE.

VSCode: https://code.visualstudio.com/download

Download and install the version suitable to your device specifications.



Process Scheduler

Overview

Process Scheduler is a desktop application to run and visualize the Job/Process Scheduling Algorithms. Users can enter the details of the algorithm and visualize how the algorithm works with the queue diagram animation.

Goals

1. To solve the job scheduling algorithms for the given inputs.
2. To make users visualize the working of algorithms with queue diagram animation

How to run

Follow the commands to run the application.

1. Download the zip file of the Project and extract it or download the **ProcessScheduler** folder from https://github.com/ParthPrajapati43/OS-Algorithms on your PC.
2. Open '**cmd**' and navigate into the folder of the project.
3. Run > **py ProcessScheduler.py** (without '>') and the Application Window will appear.

How to use

We have uploaded a working video of the project on YouTube which will give you all the information on how to use the process scheduler application and what all features are there.

Link of the video: https://youtu.be/CCd-H7qVybw

Even though below are the steps listed about how to use this application.

1. Select the **algorithm** in the drop-down list.
2. Select the **mode** - preemptive/non-preemptive.
3. Select the **type** - with/without IOBT.
4. Set the **context switch** with the numeric spinbox (if required). 5. Set the **time quantum** with the numeric spinbox (only for RR).
6. Set the **number of jobs** with the numeric spinbox and click on the **Go** button.
7. Fill all the **required input fields** in the job input table and click on the **Solve** button.
8. For filling down the inputs, you can also fill in the excel file **inputs** which is located in the project folder and then import them in the application with the **Import** button.
9. Now you have all the calculated results and the queues are set with the jobs for the visualization.
10. For type with IOBT algorithms, you can click on the **IO Gantt Chart** button to see the IO Gantt Chart.
11. You can click on the **Generate Graph** button to get the graph of Job vs in CPU time (and Job vs in IO Device time when type is with IOBT).

12. You can click on the **Download** button to get all the information for the given inputs into a **pdf** file which will get saved in the project folder.
13. You can click on the **Help** button to get any kind of help regarding the algorithm or the application.
14. You can play and operate the queue diagram visualization with the **Play**/**Pause**/**Next Step**/**Previous Step** buttons.
15. While the visualization is going on, a **progress bar** at the bottom of the application will indicate the progress of the algorithm in percentage.
16. After you are done with one set of inputs, you can set the algorithm information and reset the input fields with the **Reset** button and if the number of jobs differ, set the number of jobs and click on the **Go** button to start with the next set of inputs.

Specifications

There is total 7 algorithms implemented in this application:

1. First Come First Serve (FCFS)
2. Shortest Job First (SJF)
3. Shortest Remaining Time First (SRTF)
4. Round Robin (RR)
5. Longest Job First (LJF)
6. Longest Remaining Time First (LRTF)
7. Priority

For every algorithm, there are options for mode as preemptive or non-preemptive and type as with IOBT or without IOBT. You can also set the context switch time and time quantum wherever required for every algorithm.

Here is a brief description of all the features of the application which make it more innovative and unique:

Colors

For every job, a unique color is allotted which makes it easier to visualize the animation and study the Gantt Chart.

Import

A normal person is more used to MS Excel than our application. We have kept an excel sheet in the project folder where the table is already prepared, the user just needs to enter all the details over there and with the **import** button, the inputs will be filled in the application window.

Graph

After the inputs are given and the solve button is clicked, the user can click on the **generated graph** button which will display the graph of Job vs in CPU time (and Job vs in IO Device time).

Download

If any algorithm is solved, with the **download** button, all the information like the output table, Gantt chart, etc. will get downloaded in a PDF file into the project folder.

Gantt Chart

For every input when the algorithm is solved, a colorful Gantt chart is displayed at the bottom of the application window which tells at what time which job is in the CPU (or IO Device). If the algorithm is of type with IOBT, the IO Gantt chart is also displayed by clicking on the **IO Gantt chart** button.

Queue Diagram Animation

The bonus feature of the application is a timer-based animation of queue diagram which demonstrates which job is present in which state at any time of the algorithm. With the **play** button, the user can start animation and if he/she wants to manually go and examine some particular steps, he/she can pause the animation with **pause** button and traverse to particular step with **next step** and **previous step** button. And the progress of an algorithm is displayed by a **progress bar** at the bottom of the application window.

Future Work

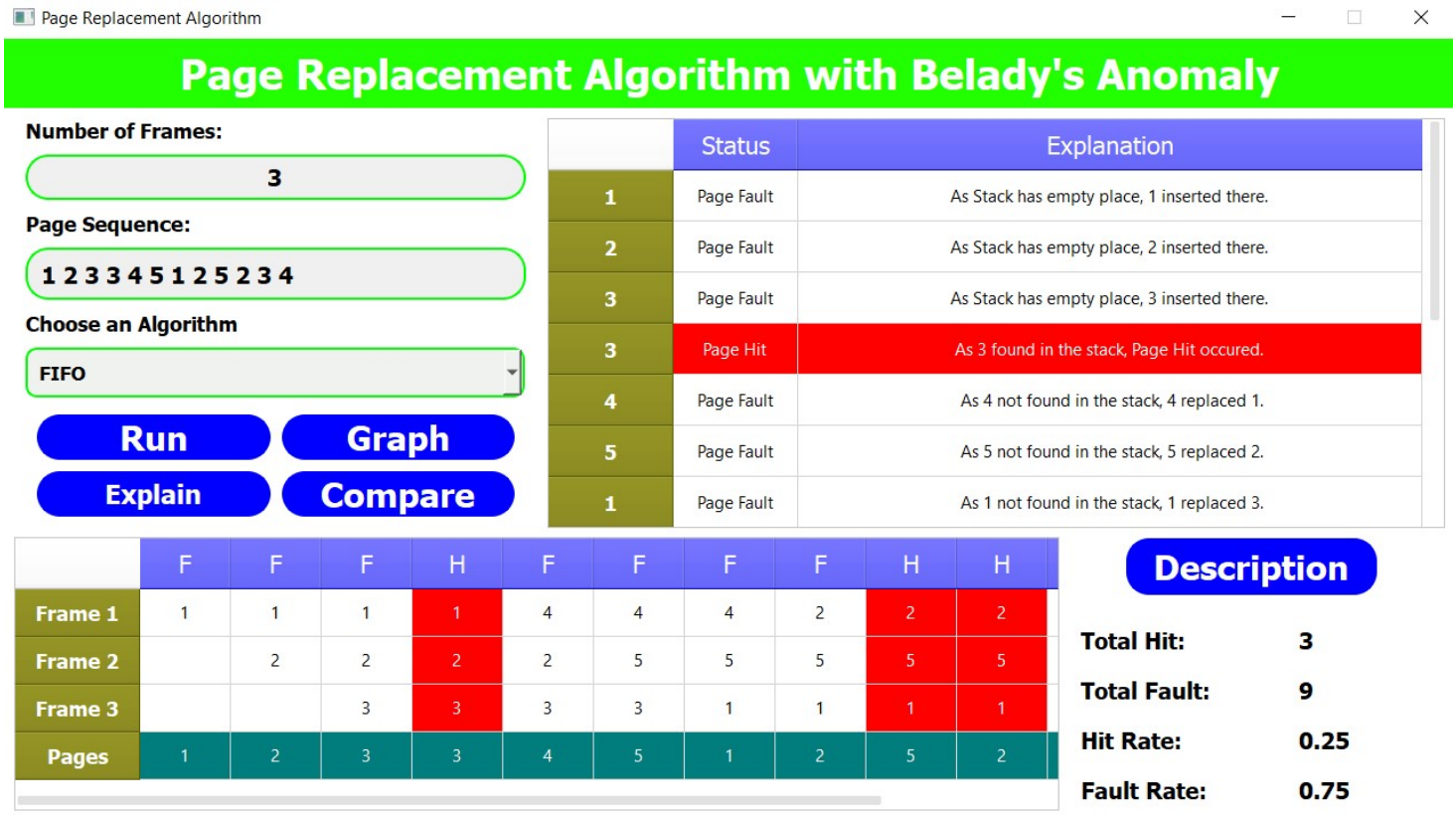
Some more features can be implemented and added to make the application better and user friendly. Here are those which can be a part of the application in future.

Compare All

Compare All button can be added which will run all the algorithms for the same input and give the statistics of which algorithm will perform the best for the given input. This will also include a single graph with plot for all the algorithms and a PDF download with all the statistical details for why a particular algorithm is better than other for the given inputs.

Universality with multiple IOBT

It is not that a job will only go once for IO Device, it can go as many times as it requires. So, the input table can be extended with a button for each job to add multiple IOBT inputs.



Page Replacement

Overview

Page Replacement Algorithm is an integral part of Operating System. In this GUI we have made sure that if any individual visits the application gets complete knowledge about theory as well as the practical aspects of Page Replacement Algorithm. We have implemented total six algorithms, all of them were implemented using Python language and those algorithms are FIFO, LIFO, LRU, RPR, OPR and SCR. In theory part, we have explained about the basic idea and concepts of all the six algorithms. The language and layout are kept very simple so that anyone can get the concepts very easily. In practical part, the users can interact very easily with the help of userfriendly and understandable interface of the GUI. They can enter their data for any of the six algorithms and can get output with table of implementation, fault count, hit count, fault rate, hit rate, graph and explanation of each step. The overall purpose of creating this GUI is to create an online simulator of how various page replacement algorithms works.

Goals

1. To Solve and Implement the different types of Page Replacement Algorithms.
2. To Plot the graph of Hit Rate and Fault Rate and compare them with different types of other Page Replacement Algorithms.

How to run

Follow the commands to run the application.

1. Download the zip file of the Project and extract it or download the **OS-Virtual-Lab** folder from https://github.com/rgautam320/OS-Virtual-Lab on your PC.
2. Open '**cmd**' and navigate into the folder of the project or else open the extracted folder using any IDE supporting python.
3. If open with **'cmd'** Run > **py OSVirtualLab.py** (without '>') than the Application will appear.
4. Click on **Page Replacement** than the Page Replacement Application window will appear.

How to use

We have uploaded a tutorial/demo video of the project on YouTube where you can get to know on how to use the Page Replacement application and what all features are there. Link of the video : https://youtu.be/GRnetyrSwhY

Even though below are the steps listed about how to use this Application.

1. Select an **algorithm** among the one in the drop-down list.
2. Enter the number of frames in the text box labeled as **Number of Frames**.
3. Enter the Pages in the next text box labeled as **Page Sequence**.
4. Click the **Run** Button to get the Total Hits, Total Faults, Hit Rate, Fault Rate in the Right bottom of the application below Description Button.
5. By clicking in the **Run** Button, you can also get the visualization of stacks have pages in it with the Page Hit highlighted in Red Color.
6. By clicking in the **Explain** button you can also get the Explanation of Right top of the application explaining the cause why the particular page was page fault or page hit.
7. By clicking in the Graph button, we can obtain the graph of Hit Rate v/s Fault Rate of the selected Algorithm.
8. By clicking in the **Compare** button, we can obtain the Comparison graph of all the Page Replacement Algorithms.
9. By clicking in the **Description** button, you can get the clear description of the selected Algorithm in a new pop-up window.

Specifications

There is total 6 algorithms implemented in this application:

1. First in First Out (FIFO)
2. Second Chance Replacement (SCR)
3. Least Recently Used (LRU)
4. Random Page Replacement (RPR)
5. Last in First Out (LIFO)
6. Optimal Page Replacement (OPR)

For every Algorithm, we have given Graph plotting Functionality Through which the user can plot the graph for every combination of inputs also we have given the option of comparison using graph where user can compare the graph of all the algorithms mentioned. Some more features of our application are descried below.

Description

We have given a button for users where they can read about any of the given Page Replacement Algorithm in detailed and clear explanation.

Color

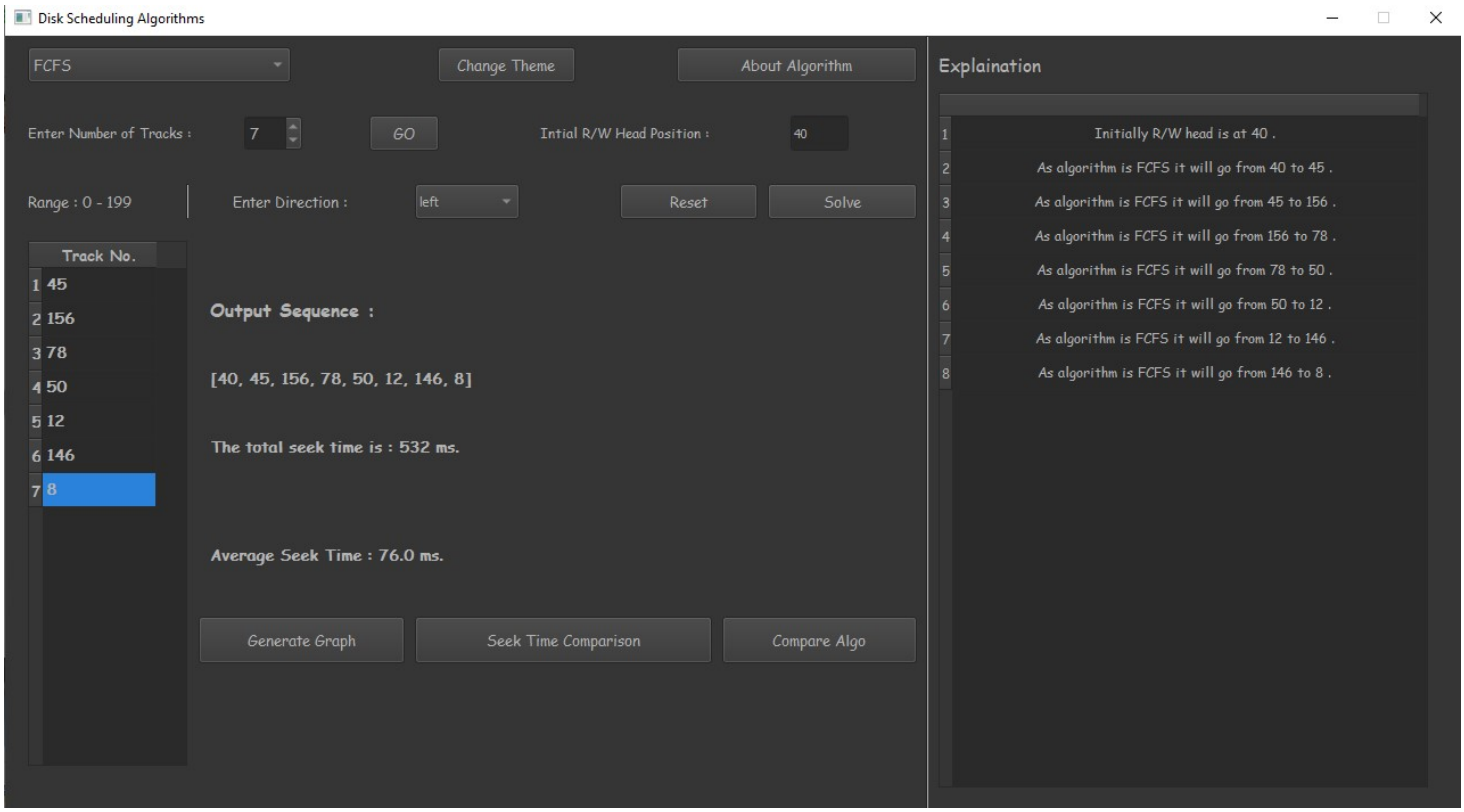
Each algorithm has been given different color while comparison so as to differentiate easily between algorithms.

Visualization

Every Algorithm is nicely explained with the help of understandable tables as well as highlighting the Page Hits.

Future Work

- More algorithms can be implemented like MRU, clock etc.
- The implemented algorithms are perfectly executed but the animation parts can still be made better and attractive.
- The GUI can be made more powerful by adding concepts of speech recognition, so that the GUI can run on the voice commands as well.
- Deadlock and Concurrency tab is also given so that someone in future can implement algorithms of Deadlock and Concurrency and complete the Entire Application.



Disk Scheduler

Overview

Disk Scheduling is one of the cores and integral concepts of the Operating systems. In this Graphical user interface, we have tried our best to cover maximum of the theory and implementation part of the disk scheduling algorithms. In total, we have implemented seven disk scheduling algorithms which includes FCFS, SSTF, LIFO, SCAN, LOOK, C SCAN and C LOOK. We have provided all the information about those algorithms as well. The language and layout are kept very simple so that anyone can get the concepts very easily. In practical part, the users can interact very easily with the help of user-friendly and understandable interface of the GUI. We have also featured the comparison of all the algorithms through which one compare the parameters like seek count and order of arrival of all the algorithms in one graph. The overall purpose of creating this GUI is to create an online simulator of how various Disk Scheduling algorithms works.

Goals

1. To get the output sequence of track number for a particular input given by the user.
2. To plot the graph of track no. vs order of arrival in order to understand the algorithm.

3. To compare the seek count of all the algorithms for a particular input in a single graph to understand the efficiency of all the algorithms.

How to run

Follow the commands to run the application.

1. Download the zip file of the Project and extract it or download the **OS-Virtual-Lab** folder from https://github.com/Priyank2912/Visual-Implemantion-of-Opreating-System-Algorithms on your PC.
2. Open '**cmd**' and navigate into the folder of the project or else open the extracted folder using any IDE supporting python.
3. If open with **'cmd'** Run > **py OSVirtualLab.py** (without '>') than the Application will appear.
4. Click on **Disk Scheduling** than the Page Replacement Application window will appear.

How to use

We have uploaded a tutorial/demo video of the project on YouTube where you can get to know on how to use the Disk Scheduling application and what all features are there.
Link of the video : https://youtu.be/L2iqhflYLZg

You can also follow the below given steps in order to use and implement disk scheduling:

1. On the top left, you can see a select algorithm button is there through which you can select any of the seven algorithms mentioned above.
2. Below it, there is an enter number of tracks button. Exactly, besides it there is a GO button. So, when you enter the number of tracks as much you want and click on go button, the number of rows will be created in the below table as per your input.
3. Next step is to enter the input track numbers one by one in the corresponding rows that were created.
4. Besides Go Button, you can see a label named Initial r/w head position. Next to it you have to enter the position you want.
5. If have selected any algorithm which needs the direction to be mentioned, then you also need to enter the direction by selecting either left or right from the drop down.
6. In case you forgot to enter the direction, right will be taken as the default direction of the algorithm.
7. After all this, when you click on the solve button, the output sequence and total seek time and average seek time will be displayed.
8. On the right side of the window, the explanation part basically shows the trach of the read/write head.
9. Generate graph button will generate the graph of order of arrival of a particular track number in the input.

10. Compare Algo will compare all the algorithms in one single graph.
11. Seek time comparisons shows the bar graph comparing the seek time of all the algorithms.
12. We have also added the change theme button which basically allows the user to switch between the light mode and dark mode.
13. About algorithm will show the complete information about the algorithm selected by a user.
14. At last, the reset button will completely the reset the disk scheduling window and the previous data will be lost.

Specifications

There is total 7 algorithms implemented in this application:

1. First come First serve (FCFS)
2. Shortest Seek Time First (SSTF)
3. Last in First Out (LIFO)
4. SCAN
5. C - SCAN
6. LOOK
7. C - LOOK

For every Algorithm, we have given Graph plotting Functionality Through which the user can plot the graph for every combination of inputs also we have given the option of comparison using graph where user can compare the graph of all the algorithms mentioned. Some more features of our application are descried below.

Color

Each algorithm has been given different color while comparison in the graph so as to differentiate and analyze easily between algorithms.

Graph

After the inputs are been provided, we provide three types of graph. First is generate graph for graph of the provided algorithm. Second is the seek time comparison which compares the seek time of all the algorithms through bar graph and last one is compare algo which basically plots the order of arrivals of all the algorithms in a single graph for a specific input. All those graphs could all also be downloaded in your device.

Future Work

- The implemented algorithms are perfectly executed but the animation and graphics part can still be made better and attractive.
- Also, a 2-D or 3-D Disk visualization could be added in order to make it more user friendly and understandable.
- The GUI can be made more powerful by adding concepts of speech recognition, so that the GUI can run on the voice commands as well.
- Also, this application could be connected to the database in order to save all our data at one place.
- Deadlock and Concurrency tab is also given so that someone in future can implement algorithms of Deadlock and Concurrency and complete the Entire Application.