Note: Problem Statements are only indicative. You are strongly encouraged to refer internet resources for a through literature of the existing support of concepts addressed in the project(s) and their support in different OS platforms and factor them in your development. Specs listed here could require the use of fork, exec, multithreading and their related system calls in Linux. Wherever data consistency / synchronisation is required you may as of now use flock / file locking mechanisms. However students are free to use semaphores support which shall be discussed in the subsequent lectures post sharing this problem statement. Projects involving games, the given specs are only indicative and you may refer to the internet for the complete rules / instructions governing the game. The list of project topics is not exhaustive. You may in discussion with the faculty and the TA identify some topic not in the list as well. Indicate the topic (team wise selection) to the TA by 10th October.

---

(I) Develop your own command prompt in LINUX (this would be an extension to your LAB exercise on command line simulation) with all functionalities supported by the Linux Command Prompt. [Denote your command prompt in a unique manner]. Feature support to include

(i) support for execution of built in commands and execution of programs (executables) both using absolute and relative path.

(ii) support for execution of shell built in features like echo, cd etc.. as part of your command prompt

(iii) support for up and down arrow key feature to trace the predecessor and successor commands (similar to doskey command in MS DOS / CMD Prompt of Windows.

(iv) support stream indirection (input) and redirection (output) using < and > operator at your command prompt.

(v) Also support features / commands to perform device redirection / indirection ( printer / read from pen drive.). Any other features supported across different OS platforms may be factored in the project to merit additional credits.

---

(II) Develop a multithreaded version of a Text Editor which allows user to edit / manipulate documents. Your editor should be capable of accepting keystrokes, performing spell check, search for words, search and replace, etc. & other features supported by widely used editors such as vi, emacs, nano etc. You may chose to make use of inbuilt feature support in Linux (such as ispell for spell check, etc.). Given the end user natured requirement of the task it is better for the project to support a visual interface for the features supported. However a command line editor would be fine subject to the feature requirements being satisfied. (Graphical Support would merit additional credits.)

---

(III) Develop a multithreaded version of the TIC TAC TOE game in a two player scenario. Also have feature for the user playing against a system (threads to check for the optimal machine move given a current state of the board and the user moves so far.) Have support for user interaction via mouse to

make it interactive. Additional constraints such an interactive design in the tiled format may be incorporated.

---

(IV) Develop a multithreaded version of the classical 15 puzzle solver game where tiles can be slid in a frame (one empty slot). The classical version involves unjumbling a layout of numbers 1-15. Extend the solution to simulate a tile arrangement game to form an initial image that is split into 15 frames. The solution should provide an interactive option to users to move tiles on the layout.

---

(V) Develop an application (multithreaded) that implements generation of any of the four series listed in the link below. (Fibonacci,Catalan,Magic Square series,etc.). The application should be interactive and provide GUI options for the user to choose the series to be generated. Also support options for the user to cancel any of the series generation at any given point in time, and to update the display of the respective series (threads) in an incremental fashion, rather than waiting for the entire series to be generated and then finally display. The application should support the principle of responsiveness required with GUI applications. Extra credits for implementations to go beyond the stated number of series.

https://edublognss.wordpress.com/2013/04/16/famous-mathematical-sequences-and-series/

---

(VI) Develop an Interactive Matrix Package application with support for operations such as Inverse, Determinant, Equality Check, Identity Matrix, Magic Square Checking, Magic Square generation as a multithreaded application. You may refer to the link below for an exhaustive list of operations possible with matrices and chose to include a few operations from the list. (extra credits for the same shall be considered). The application should be implemented in a GUI friendly fashion supporting end user friendly fashion to input matrices, compute the required statistic and display the results. https://cran.r-project.org/web/packages/Matrix/index.html

---

(VII) Develop an interactive application (multithreaded) SCHEDULER that demonstrates the working of Pre-Emptive and Non Pre-Emptive class of scheduling algorithms. The application should provide interface for process creation and demonstrate the working of each scheduling algorithm in a GUI friendly fashion. You may also include one more option for the user to upload binaries and have them executed in a set scheduler fashion. You may implement 3 algorithms under each category mandatorily and explore more options for extra credits.

---

(VIII) Develop a mulitithreaded version of the Big Two Card game. More details / rules about the game are in this link ; https://www.pagat.com/climbing/bigtwo.html. Your application should allow for multi player scenario, scoring system and a user friendly interface for game playing.

---

(IX) Develop a parallelized search application to search for files for a user string (search key) in the disk using multithreading. The application should be interactive and provide a GUI for the user to enter the search string and also display the search process (say 4 threads at a time) and highlight the occurrence of the search string in the file in an interactive fashion. Go thru the features of GREP command in Linux and come up with a interactive and multithreaded version of the same.

---

(X) Develop an interactive multithreaded version of a Maze game incorporating treasure hunt principle. The game shall support options to reward player(s) on going thru paths that have varied scored treasures while penalize paths with obstacles / daemons. Allow users to back track from a chosen path and penalize on reaching paths with dead ends. Exploit the support of multithreading to the maximum extent for multiple path searching, scoring, display etc. You may refer the internet for more rules and constraints governing a specific version of the Maze game you choose.

---

(XI) Develop an interactive and multithreaded version of the classical Snake and Ladders Game with a scoring system for n player scenario. The application should be interactive and simulate the die rolling with a random number call in one of the threads. Support options to maintain and display the scores of individual players and include as many features possible to make the game lively and interactive.

---

(XII) Develop a interactive and multithreaded version to simulate the SUDOKU game. Support user friendly options for input and blocking wrong entries.

---

(XIII) Explore any variant (multiple variants of SUDOKU are there such as WORDKU, SUDOKU CUBE (Rubix Cube + Sudoku), KAKURO, etc. in an interactive and multithreaded version. Each game / variant would be treated as a separate project. Extend the chosen application to support single user and system v/s user (or) multiplayer scenario , which ever is appropriate for the chosen game.

---

(XIV) Develop a multithreaded CROSSWORD solver game (English language – chose a domain of interest such as sports, politics, science, etc.) Support options for ACROSS and DOWN clues and allow user to enter words in a interactive (GUI) fashion. Indicate the correct and wrong entries with distinct response to help the user.

---

(XV) Develop a multithreaded SORT package that supports threaded version of classical sorting algorithms from Comparison, Exchange, Internal and External sort algorithms category. Choose sorting algorithms that lend well to multithreading and support a user friendly GUI to demonstrate the working of the sorting algorithms. The application may mandatorily include two from each category, however explore more in each category for additional credits.

---

(XVI) Develop a multithreaded STATS package that supports computation of several parameters such as Mean, Median, Mode, SD, Variance, Range, etc. for a large sized array of user inputs. Also have support for drawing various plots of user data such as Box Plot, Stem-Leaf Plots, etc. (you may refer the internet for a detailed listing of plots for specific data types such categorical, numerical attributes, etc.). The application should compute the data required for the plots and perform the plotting (multiple data) in a multi-threaded fashion.

---

(XVII) Develop a multithreaded SEARCH package that implements different search algorithms (refer the net for advanced search strategies such as a m way search, Fibonacci search, etc.). The application must be interactive and demonstrate the working of the algorithm in a step wise fashion. Mandatory to implement 4 search strategies other than linear and binary search which is done in the lab.

---

(XVIII) Develop a multithreaded version of the N Queens problem with appropriate GUI support. Refer the internet for detailed rules concerning the N Queens

Other variants to this could be Knights tour, Wheat and Chessboard Problem, etc. would be separate project topics.

---