# CptS 370 Program 2: ThreadOS Shell Report

In this report, I’ll explain the algorithm of Shell.java.

The main purpose of this project is to simulate the shell by integrating the Thread implementation class. By receiving user input information and correctly expressing Shell's working logic, that is, through "&" and ";" To perform a multiinstruction operation.

Once the project is executed, it will first display "shellName" and begin counting each operation. By reading the user input and saving it as a StringBuffer (buf). It converts to a string and keep track with the number of the command line (cmdNum). This string is then converted to several separate strings and saved in commandString. Determine whether the input statement is empty. If it is, start waiting for the input statement, and through traversing the entire input statement, if ";" , executes the current statement and executes the new input using SysLib.cin(), join the completed concurrent threads that were called earlier and then execute SysLib.join() when the delimiter is found. Count the number of commands by querying the number of each command to put in, using "&". Use syslib.exec() to determine if returns a child Thread ID on success (-1) and add the ID of the child. Use HashSet to keep track of the thread id that is currently waiting to complete. Also, use SysLib.join () to determine if the child process has been woken up and to report an error if it fails(-1). Finally, if the user input is "exit", exit the program through SysLib.exit () and output the corresponding exit statement using SysLib.cout().