# Main Controller Input

## Functions

a. Read File

Inputs: Text file named “Input.txt”

Outputs: Error messages if applicable

Function Purpose: The read file method takes in an input text file and uses the data within to construct proper job objects to be run in the system.

## Data Structures Used

a. ioBeginnings Vector of IO Beginning Time Integers

b. ioRequestLengths Vector of IO Request Length Integers

c. Vector of value entry Integers called Job Data

d. Integer lineCount, keeps track of which line is being read in the input file

e. Integer lastIOBeginning, keeps track of the last IO beginning on each input line

f. Boolean goodEntry, tracks whether a job should be accepted

g. ifstream readFile, used for reading from the file

h. String fileLine, used for storing entire lines from the file

i. Integer jobCount, counts the number of jobs accepted so far

j. String delim, used for creating a delimiter

k. String temp, used for temporary string storage

l. Integer temp2, used for temporary integer storage

## High-Level Description

The read file input is the primary input module in the overall system, which takes in a file named “Input.txt” in text format, parses the data, checks the data, and then constructs a job object using that data or sends out a system error message in the case that the data was invalid. If the checks pass and a job object is constructed, it will be added to the Master Array (See Main Controller System Documentation).

## Low-Level Description

The read file function takes a great many steps. Each of the steps below will run in a while loop, on the condition that there are still lines to read in the input file, and the number of jobs entered in the system is still less than 60. The variable Line Count will initialize to zero before the loop is entered.

1. Line count will be incremented, and all other data structures will be reset, meaning all vectors will be cleared, the lastIOBeginning variable will be set to zero, and the Boolean goodEntry will be set to true.
2. The entire next line in the input file will be read in for parsing and stored in the fileLine String variable.
3. In a for loop running from 0 to 12
   1. The temp String variable will hold the part of the string up until the next comma
   2. The fileLine variable will be reduced to everything appearing only after the comma mentioned in part a
   3. The temp String will be converted to an integer value and stored in the temp2 Integer value
   4. The value now held in temp2 will be added into the Job Data vector
4. For positions 3 to 7 inclusive in the Job Data vector, which will hold the IO Beginning Times
   1. Check that the current IO Beginning Time observed is a time past that of the last IO Beginning time
      1. If so, add the current IO Beginning Time observed into the IOBeginnings Vector, and set the lastIOBeginning Integer to the current IO Beginning Time observed.
5. For positions 8 to 12 inclusive in the Job Data vector, which will hold the IO Request Lengths
   1. Check that the current IO Request Length observed is between 25 and 50 inclusive
      1. If so, add the current IO Request Length observed into the ioRequestLengths Vector
6. Check the values for validity. Set the goodEntry Boolean to false and output appropriate error message if any fail.
   1. Check that the size of the Job Data vector is 13, the appropriate number of entries for a Job Object to be constructed
   2. Check that the entry at the 0 position in the Job Data vector is greater than 0, as this is the entry time value.
   3. Check that the entry at the 0 position in the Job Data vector is greater than the lastEntryTime integer variable, as no job can time travel.
   4. Check that the entry at the 1 position in the Job Data vector is between 1 and 8 inclusive, being the entry for the amount of RAM needed, as per instructions.
   5. Check that the entry at the 2 position in the Job Data vector is between 10 and 950 inclusive, as this is the total execution cycle value, as per instructions.
   6. Check that the ioBeginnings Vector and the ioRequestLengths Vector are the same size, as if they are not equal then the input clearly is not valid.
7. Finally, the Job Data information will be placed into a PCB Object, which will then be placed into the Master Array (See Main Controller System Documentation), but only if the goodEntry Boolean is still set to true.
   1. A new PCB Object will be placed onto the Master Array
   2. For the following variables, see PCB System Documentation for explanation. Each describe a variable belonging to the PCB class, and these variables will each be set in the PCB Object just placed onto the Master Array
      1. The pid variable will be set to the jobCount Integer
      2. The entryTime variable will be set to the 0 position in the Job Data Vector
      3. The ramNeeded variable will be set to the 1 position in the Job Data Vector
      4. The totalExecutionCycles variable will be set to position 2 in the Job Data Vector
      5. The currentExecutionCycle variable will be set to 0
      6. If the ioBeginnings Vector if of size greater than zero
         1. The nextIORequestNumber variable will be set to 1, referencing that the 1st IO Beginning is the next to be serviced
         2. Else, the nextIOBeginning variable will be set to 0
      7. The currentIORequest variable will be set to 0
      8. The totalIORequests variable will be set to the size of the ioBeginnings Vector
      9. For five iterations of a counter variable i starting from 0,
         1. If the ioBeginnings Vector is of size greater than i
            1. The ith position of the ioBeginnings Array in the PCB Object will be set to the ith position of the ioBeginnings Vector
         2. Else, the ith position of the ioBeginnings Array in the PCB Object will be set to -1
         3. If the ioRequestLenghts Vector is of size greater than i
            1. The ith position of the ioRequestLengths Array in the PCB Object will be set to the ith position of the ioRequestLengths Vector
         4. Else, the ith position of the ioRequestLength Array in the PCB Object will be set to -1
      10. The currentIOCycle variable will be set to -1
8. Finally, the lastEntryTime variable will be set to the 0 position of the jobData Vector if the current job observed was accepted into the system. And under the same circumstances, the jobCount Integer will be incremented.
9. If the job was rejected, an error message will be printed specifying which line of the file was rejected, and suggesting a reference to the System Documentation for proper job input.

After the loop ends, the readFile Object will be closed.