Dalton Meny

CS 470

Process Management Process

My project consists of a main function with five auxiliary functions. The main function reads in and parses the input and calls the appropriate function that corresponds with either create, destroy, wait, event, or interrupt. Interrupt is equivalent to a tick and is called by most other functions. I created a data type called process which holds all the info of a process. This included the PID, a vector of children PIDs, its parent PID, the remaining burst time, the reaming quantum time, and event number. This made it easy to pass all of this information around at once. For my ready and wait queue, I used deques of the process type. This made it easy to add elements on the back and take off the front with other elements shifting down. By using deques, it also allowed me to access any element in the deque. This made it much easier to search through them and made it possible to delete an element that is not at the front or back.

Questions

1. It can be said that the RR scheduling algorithm can be very complicated and hard to follow at time. It seems to me that the bigger the quantum size, the less complicated and easier to follow the algorithm becomes.
2. The aspect of this project that I found the most difficult was implementing the delete function. It was very complicated and required a lot thinking about how it should work. The fact that the delete cascades is a big part of this difficulty.
3. I found the event and wait functions the easiest part of process management to implement. Both of these functions were fairly straight forward and were not very complicated.
4. I would like to clean up my code some. To me, it seems to not be the most efficient in some places and I feel like I could possibly implement a few more functions to make it work better.
5. I found it interesting how much code it takes to implement process management on this small scale. I can only imagine how much it would take to do this in a much larger and more practical scale.