Report:

There are three modes of futures implemented in this program. Shared, Queue and Exclusive to be specific. The files implemented in the system were future_get , future_set, future_alloc and future_free.

The future prod sets the value using the system call future_set and future_get helps get the value for future_cons.

1 – future_alloc – This allocates memory by creating a new queue and settings the new id's.

2 – futue_free – This would free the memory allocated by future_alloc.

3 - future_get – This function gets the value that is set by future_set. This function gets the process id and enqueue itself to get_queue and suspends it which depends if the state was already future_empty or future_waiting. Once the process gets the information, it sets the state to future_waiting. If the dtate is future_valid, it sets the argument value to the value set by the function itself. It also removes the first process at the head from the set queue and resumes the process. If its in the exclusive mode, there is a process waiting for the get value, there will be a error flagged since there can't be more than one queue in the get queue.

4 - future_set – If the state is empty or waiting it adds itself onto the set_queue. If the get_queue is empty, it suspends else it deqeues from the get queue if the mode of the futures is queue or shared. If it's exclusive, there would be only one process left as it is exclusive. If the state is shared or exclusive, it would throw an error since future_set can be only called by one process once. This helps set the value in the futures, the value is passed as an argument as given in the system call. This would then set the state to future_valid.