Ahmad Adil 9/4/22
Computer Science 332: Operating Systems
Professor Debnath

Lab 1: Link List Implementation

A linked list is a linear collection of data elements whose order is not given by their physical placement in memory, but by each element pointing to the next. It is a data structure consisting of a collection of nodes which together represent a sequence. Each node contains data and reference to the next node within the list. This structure allows for efficient insertion or removal of elements from any position in the sequence during iteration. The two types of lists that were implemented in this laboratory experiment were, Singly linked list and a circular linked list.

A singly linked list is a type of linked list that only allows the traversal of the list in one direction. This direction is from head node to the tail, the last node. One of main advantages of a singly linked list is that it is a dynamic data structure. This means that it can grow and shrink at runtime by allocating and deallocating memory. So there is no need to give the initial size of the linked list. Another advantage of a singly linked list is that certain operations such as insertion and deletion are quite easy to implement, as there is no need to shift elements after the insertion or deletion of an element; only the address present in the next pointer needs to be updated. One disadvantage of a singly linked list is memory usage. More memory is required in the linked list as compared to an array. Because in a linked list, a pointer is also required to store the address of the next element and it requires extra memory for itself. Another disadvantage is that random access is not possible in a linked list due to its dynamic memory allocation. One application that uses linked lists could be music players. Music Player applications need to know both the

previous song as well as the next song that needs to be played. A linked list would allow you to play songs at the beginning or end of the list.

A circular linked list is similar to a singly linked list, however the last node of the linked list points to the first node. This feature makes it circular. One of the advantages of a circular linked list is that the starting node does not matter as we can traverse each node despite whatever node we kept as the starting node. Another advantage is the use of the round robin algorithm. This allows us to effectively complete online queues without having to meet NULL suspension or reference references. One disadvantage of a circular linked list is that it can lead to an infinite loop if not set up properly. Another disadvantage of a circular linked list is that trying to directly access elements is not possible. One of the applications for circular linked lists is that data structures such as stacks and queues are implemented with circular linked lists.