Delikarl Jean-baptiste

October 9, 2016

Data structures and Algorithm II

Functional Decomposition

**poissonDistribution.c**

//This is my function to calculate factorials

double factorial(int);

//This is my math function to get the result of a "power"

double powerFunction(double, double);

//This is my math function to get summation of the first part of Po.  
double summationFunc(int, int, int);

//This is my percent idle time function to get Po  
double percentIdleTime(int, int, int);

//This is my function to get the average number of people in the system (L)  
double avgNumOfPeopleInSystem(int, int, int);

//This is my function to get average time a customer spends in the system (W)  
double avgTimeCustomerSpendsInSystem(int, int, int);

//This is my function to get average number of people in the queue (Lq)  
double avgNumOfCustomersInQueue(int, int, int);

//This is my function to get the average time a customer spends waiting in the queue (Wq)  
double avgTimeCustomerSpendsWaitingInQueue(int, int, int);

//This is my function to get the utilization factor for the system (rho)  
double utilizationFactorForSystem(int, int, int);

**linkedQueue**

//my function to see if the queue is empty

int isEmpty();

//my function to add to the queue  
void enqueue(int x);

//my function to remove from the queue  
int dequeue();  
//my queue struct function  
typedef struct Node{  
 int data;  
 struct Node\* next;  
}queueNode;  
//My head and tail pointers  
queueNode \*head, queueNode \*tail;

**priorityQueueHeap**

//struct for customers  
struct customer

//function to initialize the priority queue by setting everything to 0;  
void initPQ()

27 //function to create my customers

void createCustomer()