1. (10 points.) Redo problem #1 of HW #1 using sharpe. Submit the sharpe code and program output.

2. (20 points.) Consider a system with 3 CPUs and 2 memory modules that requires at least 1 CPU and 1 memory module to be functioning for the system to be functioning. Use sharpe to calculate the system reliability \( R(t) \) at \( t=800 \) hours. Assume that the MTTFs of a CPU and a memory module are 500 and 1000 hours, respectively. The MTTRs for a CPU and a memory module are 10 and 20 hours, respectively. Show the sharpe program listing and output. Consider the following three cases separately. For each case, you need to compute \( R(t) \) at \( t=800 \) hours. Use any model you like.

   (a) Each component (CPU or memory) has an independent repair facility.

   (b) Each subsystem (CPU or memory) has an independent repair facility that can repair failed components within the subsystem one at a time.

   (c) The whole system shares a repair facility which repairs failed components one at a time with the repair priority of memory modules over CPUs.

   (Hint: define Markov models that allow repairs to occur only when the system is still alive.)

3. (20 points.) Suppose that a network switch center has \( n = 3 \) slots to accommodate incoming high and low priority clients, with arrival rates of \( \lambda_h \) and \( \lambda_l \) and departure rates of \( \mu_h \) and \( \mu_l \), respectively. A high priority client must always occupy one full slot. A low priority client, on the other hand, can lower its quality of service (QoS) by occupying only one half of a slot, if necessary. When a low priority client occupies a full slot, we call it a low priority, high QoS client; when it occupies one half of a slot, we call it a low priority, low QoS client. Draw a Markov state transition diagram for modeling the following resource control policy:

   • If there is at least one full slot available, an incoming client always occupies an empty full slot, regardless of its priority class.

   • If there is no free slot, an incoming high priority client can lower the QoS of two low priority, high QoS clients, if they are available, after which the high priority client occupies one full slot and the two low priority, high QoS clients both become low priority, low QoS clients, with each occupying one half slot.

   • If there is no free slot, an incoming low priority client can lower the QoS of one low priority, high QoS client, if it is available, after which each occupies one half slot.

   • A client will turn away if none of the above cases is applicable.

   • A low priority, low QoS client occupying one half slot can immediately become a low priority, high QoS client occupying one full slot upon a client’s departure.

Use the representation \((x, y, z)\) where \( x \) stands for the number of low-priority, low QoS clients, \( y \) stands for the number of low-priority, high QoS clients, and \( z \) stands for the number of high-priority clients. Organize the Markov model so that when a high priority client arrives, the transition goes right, and when a low priority client arrives, the transition goes down. Label the transition rate of each transition clearly. Assume \( \lambda_h = \mu_h = 2 \) and \( \lambda_l = \mu_l = 3 \). Use Sharpe and assign rewards to states of the Markov model to obtain:

   (a) the average number of low-priority, low-QoS clients;

   (b) the rejection probability of low-priority clients;

   (c) the average response time of a high-priority client.
4. (15 points.) Consider a client-server system with a fixed number $m$ of client workstations that are connected by an Ethernet network to a database server. The server consists of a single disk and a single CPU. This leads to a closed QNM shown in the bottom figure. The Ethernet network is being modeled as a load-dependent server with service rate defined as follows:

$$\mu_{net}(1) = \frac{1}{\frac{1}{N_p}B}$$

and

$$\mu_{net}(k) = \frac{1}{\frac{1}{N_p}B + S \times C(k+1)}$$ for $k > 1$, where $C(k) = \frac{1-A(k)}{A(k)}$ is the average number of collisions per request and $A(k) = (1 - \frac{1}{k})^{k-1}$ is the probability of a successful transmission and $k$ is the number of workstations that desire the use of the network.

Assume that $N_p = 20$ (number of packets per request); $B = 10^7$ bits/sec (bandwidth of the Ethernet); $S = 5 \times 10^{-5}$ second (slot duration); $L_p = 1600$ bits (average packet length); $m=20$ (number of clients); $\mu_{client} = 0.1$/sec; $\mu_{CPU} = 10$/sec; and $\mu_{disk} = 20$/sec. Write a sharpe program to calculate the system throughput of the server subsystem, the population at the server subsystem, and the average response time per client at the server subsystem.