chapter one

System Design I: Single Station

PROBLEM

Mr. K is the manager of a new Snowball Express sno-cone stand which will soon be opening for business. The stand is operated by a single attendant who sits behind a window. People come up to the window, tell the attendant which flavor sno-cone they want, wait for the attendant to fill their order, then pay and leave.

> In operating systems terminology, the attendant is the CPU and the arriving people represent the workload submitted.

When there is more than one customer at the window, a line forms so that customers are served in the order that they arrived. People always buy one and only one sno-cone at $3 each. The cost of materials to make sno-cones is negligible, so Mr. K considers all of the money taken in to be profit, except that the attendant has to be paid from the money received. A picture of the sno-cone stand is shown in Figure 1.1.

After interviewing several applicants for the attendant's position, Mr. K has narrowed his choice to two people: Fran and Bill.

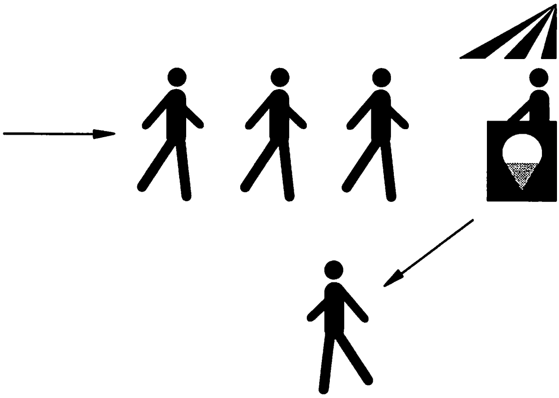


Figure 1.1 The Sno-Cone Stand

The operating systems counterpart is to choose between two CPUs of varying costs and speeds.

Mr. K has discovered (during the interviews) that Fran can complete a transaction (take the order, make the sno-cone, take payment, and make change) in 20 seconds on the average. Bill can perform the same job in 30 seconds on the average. Fran is faster than Bill but demands $12 per hour in wages, whereas Bill will work for $6 per hour.

Market research has revealed that, on the average, Mr. K can expect one customer per minute to come up to the window.

This represents the workload characterization phase.

Mr. K has also learned that his prospective customers are the kind that do not like to wait in long lines. If a customer comes to the window and there are already three people in line, he or she will promptly turn about in a huff and storm off to the Smoothy-Cream, a nearby competitor.

The buffer size is assumed to be three.

Taking all of these facts into consideration, which person is the more cost-effective to hire: Fran or Bill?

> The objective function is to maximize profit, which is the difference between extra money earned by having a faster processor and the cost of having a faster processor.

1.1 APPLICATION TO OPERATING SYSTEMS

Modern multiprogrammed computer systems comprise several devices from which processes (i.e., jobs) receive service. For example, a process may require computation service from the central processing unit (CPU), then input/output (I/O) service from a disk, and so on. Since the computer system is multiprogrammed, there are usually several processes competing for the same resources. For example, more than one job may wish to use the CPU at the same time.

In this problem, the sno-cone-stand attendant is analogous to one device (also called a "station" or "server") in the computer system, and the customers wishing to buy sno-cones are analogous to the processes demanding service from that device. The maximum line length of three is analogous to a device that has a limited amount of "waiting" space, where processes wait until they can be served by the device.

> This is also referred to as a finite-length queue or buffer.

In the design of a computer system, choices have to be made. The choice here is: Do we use a slower but cheaper device, or a faster but more expensive device? The decision is based on several things. How much demand will be placed on that device? Will the improved performance yielded by the faster device offset its additional expense?

1.2 SOLUTION

Fran is 33% faster than Bill, yet to employ Fran costs 100% more than to employ Bill. This seems to imply that Bill is the better choice. However, this natural intuitive reasoning is not always correct. Specific quantitative reasoning is needed. In order to choose Bill over Fran, we must be able to make and back up such statements as the following:

> The amount of money made while Fran operates the stand minus her pay is less than the amount of money made while Bill operates the stand minus his pay. Therefore, Bill is the better choice.

> Or, in the operating systems scenario, the improved throughput with a faster processor does not offset the extra cost of that processor.

How can such a statement be supported? The first step is to identify the quantities of interest:

1. Amount of money made while Fran works
2. Amount of money made while Bill works
3. Cost to employ Fran
4. Cost to employ Bill

In order to make calculations and comparisons with these figures, the quantities they represent must all be with respect to the same unit of time. It does not really matter what standard unit of time is chosen. What is important is that once the unit is chosen, all measurements are converted to reflect that unit. The problem above suggests *second*, *minute*, or *hour*. Arbitrarily, 1 minute is selected as the standard time unit. The statement we now wish to prove (or disprove) is:

> The average amount of money made each minute while Fran operates the stand minus her pay per minute is less than the average amount of money made each minute while Bill operates the stand minus his pay per minute. Therefore, Bill is the better choice.

Fran's and Bill's pay per minute are easily obtained by dividing each hourly payrate by 60. Fran makes \$12 per hour (i.e., 20 cents per minute) and Bill makes \$6 per hour (i.e., 10 cents per minute). These are the latter two of the four quantities required.

We now need to calculate the average amount of money taken in for each minute that each attendant works. Although the customers will come to the sno-cone stand at the same rate regardless of who is working, they are more likely to find the line of an acceptable length (fewer than three people in line before they get into line themselves) if Fran is working than if Bill is working.

The buffer length is an important parameter and must be considered.

(Remember that if the line is of length three, potential customers turn away and go to Smoothy-Cream, and that results in lost revenue.) This is because Fran is faster than Bill (on the average) and will be able to handle more customers, thus keeping the line length shorter. So, ultimately, the amount of money taken in each minute depends on the average number of customers that are actually served every minute.

The $3 cost per sno-cone is a scaling factor for throughput in this problem. In the operating systems scenario it is the weight given to a completed job relative to the negative cost of providing service.

But does the amount of extra money made when Fran works (due to customers not going to Smoothy-Cream) offset the extra cost of employing her over Bill?

Let's first assume that *exactly* one customer arrives *every* minute, on the minute. Let's also assume that Fran can service each customer in *exactly* 20 seconds, and that Bill can service each customer in *exactly* 30 seconds. Then, obviously, the line will never grow longer than one person, that being the person currently being served, regardless of whether Fran or Bill works.

That is, if we assume that the distributions of interarrival and service times are constants, Bill is the better choice.

If we made these assumptions—that the average customer arrival rate and the average service times are the same exact values for every customer—we would certainly advise Mr. K to hire Bill. Bill will get exactly one customer every 60 seconds, but he only needs exactly 30 seconds to take care of each customer. This leaves exactly 30 seconds out of every minute in which Bill can goof off, because there will be no one waiting in line.

With these assumptions, the throughput using either CPU would be the same. That is, the CPU unit is not the bottleneck and it would be best to select the cheaper, slower CPU.

As a matter of fact, there will never be a line—only the one customer being served (half the time). Since no customer will ever find the line longer than one person, no customer will turn away, and Bill will bring in just as much money as would Fran. This analysis seems to confirm the original conclusion that Bill would be the better choice.

But there is something about the previous analysis that is bothersome—the excessive use of the word *assume*. Think about it: It is very unlikely that the amount of time between when one customer walks up and when the next customer walks up is exactly the same for all customers. Also, it is very unlikely that the attendant takes *exactly* the same amount of time to make a sno-cone for each customer. Of course, such a sno-cone stand does not exist!

Modeling the variability between job arrivals and modeling the variability of the job service requirements is crucial.

The truth is that in the real world things do not tend to happen with such predictability. The one customer per minute is an *average*, as is the average number of seconds it takes each attendant to make a sno-cone. Sometimes it takes longer, and sometimes it does not take that long.

It is easy to fall into the trap of thinking that an average is the only kind of characteristic measurement needed when solving problems such as the one involving the sno-cone stand. The amount of time between customer arrivals is really random, with the average of all those random times (if you observed the sno-cone stand for several days) being an average (e.g., one customer every minute). There is a second pitfall: Even knowing that these times are random, people assume that there are just as many "long" times as there are "short" times. That is, they assume that the times are equally distributed about the average. This type of reasoning leads to the assumption that customer arrivals and the time required to make a sno-cone come from a *normal distribution*.

The normal distribution is also referred to as the Gaussian distribution.

The assumption of a normal distribution of the time between customer arrivals and the time it takes to make sno-cones is also not a wise one. The problem arises when one thinks of *average* and *usual* as being interchangeable words. But the truth is that the *average* time to make a sno-cone is not necessarily the time it *usually* takes to make a sno-cone.

> The difference between the *mean* of a distribution and the *median* of a distribution is the issue here. Since times are nonnegative quantities, there is a left boundary (i.e., zero) on the possible values measured. Thus it is often the case that the median is less than the mean. That is, it requires a lot of small numbers to offset a single very large number to maintain the same mean.

Studies have shown that even though the average customer arrival rate is one per minute, the amount of time between customers is usually less than that. It is the occasional lull in business that makes the *average* time between customers seem longer than that which is usually observed. The same is true of the time it takes to make sno-cones. It is those few times when the attendant has to crush more ice, or open a new bottle of cherry flavoring, or engage in any unusual task (e.g., a bathroom break) that tends to slow him or her down that makes the average time to make a sno-cone appear longer than it usually is.

As a further example, consider the list of numbers 1, 2, 3, 2, 12. The average of these numbers is 4, yet if we were to write each number on a slip of paper, mix them up in a hat, and draw one at random, there is a 4 in 5 chance that the number we draw will be less than 4, and only a 1 in 5 chance that the number will be greater than 4.

There is one special distribution, the *exponential distribution*, which does a good job of matching this phenomenon of observed times that are shorter than the average time.

> Actually, we should be using the term *negative exponential*, but since most other people use the term *exponential*, so will we.

We will not go into detail about it (you *really* don't want us to!), other than to say that the modeling technique about to be introduced assumes that the time between customer arrivals and the time it takes a person to make a sno-cone are exponentially distributed random values having a known average.

> This is stretching the truth, but it's fine for now. Cynically speaking (with some good underlying mathematical justification), making exponential assumptions is for convenience. It simplifies the analysis and happens to have most of the properties we are looking for.

For the sno-cone-stand problem and all other problems in this book, this is a reasonable assumption.

To make things easier, some new symbols and terms are needed. The amount of time between when one customer comes to the sno-cone stand and when the next customer comes to the sno-cone stand (whether or not the second customer decides to go to Smoothy-Cream) is known as the *interarrival time.* The average time between customers is known as the *mean* interarrival time. (The words *average* and *mean* are interchangeable.)

> The term *mean,* however, is used more frequently.

We can indicate the frequency with which customers arrive at the sno-cone stand either by stating the mean interarrival time or the mean arrival rate (from now on referred to simply as the arrival rate), since one is the inverse of the other. However, it is more common to use the arrival rate, which is denoted by the standard symbol λ.

> By "standard symbol" we mean a notation system generally accepted and understood by the performance evaluation community. The notation used throughout this book is consistent with that used by others.*

In our example $\lambda = 1$. That is, customers arrive at the average rate of one customer per minute.

Similarly, we can refer to the *mean service time* and *mean service rate* (simply *service rate*), which are the values associated with how long it takes the

*P. J. Denning and J. P. Buzen, "The operational analysis of queueing network models," *Computing Surveys* 10, 3(September 1979), 225–261. E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik, *Quantitative System Performance*, Prentice Hall, Englewood Cliffs, N.J., 1984.

attendant to take care of a customer on average. *Service demand* is synonymous for *service time*. You can think of a customer as "demanding" a certain amount of the attendant's time in order to be serviced. A customer's service demand is denoted by the standard symbol D. In this problem the demand of the customer depends on the attendant. The slower attendant, Bill, takes longer to service customers than does the faster attendant, Fran, meaning that customers will demand more of Bill's time than they will of Fran's. Therefore, every customer demands $D_{\text{Fran}} = \frac{1}{3}$ minute of service if Fran is the attendant and $D_{\text{Bill}} = \frac{1}{2}$ minute of service if Bill is the attendant. However, in the solution technique we are presenting, the service requirement of a customer is often conveniently expressed as a rate. Therefore, we denote the mean service rate by the symbol μ.

> The use of μ, λ, and D causes no problems as long as all customers are statistically identical. Systems in which individual customers have different demands and/or different arrival rates are known as multiclass systems and must be handled more carefully. Multiclass systems are discussed in Chapter 3.

Since rates and times are inverse quantities, $\mu = 1/D$. In our example, $\mu_{\text{Fran}} = 3$ and $\mu_{\text{Bill}} = 2$, meaning that Fran can take care of 3 customers per minute on average, and Bill can take care of 2 customers per minute on average.

The next step is to determine the average number of customers processed each minute, depending on the attendant. This value is known as the *throughput* and is denoted by X. If we can find this value, then the amount of money made each minute will simply be $3X$ dollars. The goal is now to derive an expression for X for each attendant.

At any given time, there can be zero, one, two, or three people at the window. We can think of these as four states of the sno-cone-stand operation, as shown in Figure 1.2. We denote each of these states with a number, which is the same as the number of people in line when the system is in that state. For example, the system is in state 0 when there are no people in line, in state 1 when there is one person in line, and so on. Therefore, all possible states are 0, 1, 2, and 3. Associated with each state i is the probability P_i that if Mr. K were to drive by the stand at a random time (which he does quite often), he would see i people at the window. In other words, there is a P_i probability that

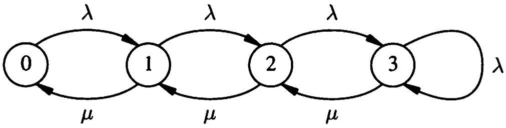


Figure 1.2 State Diagram of the Sno-Cone Stand

the system is in state i. Note that the sum of all the P_i's must be 1, since the system must be in one of the four states at any time. When the sno-cone stand is in state 0, there is no one at the window and the attendant is idle. When the sno-cone stand is in any of the other states, there is at least one person at the window and the attendant is busy (being utilized). Therefore, P_0 is the fraction of time the attendant is idle. The fraction of the time the attendant is not idle is known as the *utilization*, U. Thus $U = 1 - P_0 = P_1 + P_2 + P_3$.

Processor utilization is another performance metric often of interest. The method in which throughput will be calculated depends on knowing the utilization.

Figure 1.2 is a state transition diagram. Such diagrams are helpful in analyzing systems like our sno-cone stand.

The diagram is more formally known as a Markov diagram and its special properties arise from the fact that we have chosen to use the exponential distribution to model service and arrival times. This distribution has a so-called "memoryless" property which makes it easy to work with.

Markov processes are memoryless because, for any state the system can enter, the next state entered depends solely on the current state of the system. States visited previously to the current state and the amount of time spent in the current state or previous states have no bearing on the next transition. This allows time to be factored out of the analysis, which is a BIG help.

At any time, we think of the sno-cone stand as being in exactly one of the four states, which are represented by circles in the diagram. At every "instant" in time, one of two things will happen: Either the state of the sno-cone stand will not change, or the state will change due to the arrival or departure of a customer. For example, if the current state is state 1, the only things that can happen in the next instant of time (aside from nothing happening) are the arrival of a customer (in which case the new state is state 2) or the departure of a customer (in which case the new state is state 0). It is important to realize that at most one "thing" can happen in a single instant of time, and it takes absolutely no time to change state (traverse an arc of the diagram). More specifically, in the very next instant of time, either the state will not change, exactly one customer will arrive, or exactly one customer will depart.

This, too, is a little bit of a lie. Actually, multiple arrivals (or departures) are assumed possible, but only with a *very* small probability that can be ignored. In operating systems, this is okay since systems are driven by a common clock and at most one process is in control of the CPU at any instant.

The possibility of more than one customer arriving or departing simultaneously does not exist since we can think of the amount of time between one "instant" and the next as being as small as necessary to distinguish between the two events. This explains why there is no arc from state 0 to state 2, for example.

The arcs of the diagram are labeled with the rates associated with the change in state they represent. For example, if the current state is state 1, a customer will arrive at rate λ, increasing the length of the line to 2, and the customer currently being served will depart with rate μ, reducing the length of the line to 0. If in state 0, no departures are possible (there is no one at the window), and if in state 3, only a departure will cause a change in state. The arc leaving state 3 that also leads to state 3 represents people who arrive, see three people in line, and go to Smoothy-Cream, leaving the state unchanged.

Suppose that we begin by letting an instant in time be defined as 1 second. This is probably small enough to exclude the possibility of more than one customer arriving or departing in an instant. If, at every instant of time, we were to note the state of the sno-cone stand and compare it with the previous state, we could determine which event occurred to cause a change in state, if any. In other words, we could tell which arc was traversed during the last instant, or determine that no arc was traversed.

The issue being addressed here is that of measurement. Values that serve as input parameters to the model must be obtained in some fashion. In this case, λ, μ_{Bill}, and μ_{Fran} are required. Even though we assumed that Mr. K knew these values a priori, in reality they came from measurement data. Mr. K may have obtained the value for λ by observing other sno-cone stands in the area and counting the number of customer arrivals during some time interval. The rates at which Fran and Bill can fill orders could have been determined by testing them during the job interview (e.g., each was asked to make 100 sno-cones and the average time was determined).

In computer systems, measurements are usually made by a special piece of hardware and/or software called a *monitor*. Monitors vary in sophistication, but usually provide data such as the number of jobs in the system at a given time, and the demands placed on system resources by various jobs.

Since our standard time unit is 1 minute, let's say that at every second we were able to total up the number of times each arc was traversed in the preceding 60 seconds and keep an average of this count for each arc. After we have observed the sno-cone stand for a long time, these averages would tend toward stable values for each arc. We would also be able to calculate the probability of being in a state by calculating the fraction of time spent in that state. These probabilities would also tend toward stable values.

The average number of times an arc is traversed per standard time unit is thought of as the amount of *flow* along that arc. It is obvious that the amount of flow along an arc is heavily dependent on being in the state from which that arc departs. The flow along an arc is simply the product of the probability of being in the state from which the arc departs and the rate associated with the arc. For example, the flow along the arc departing state 0 is λP_0. Remember that after observing the sno-cone stand for a long period of time, these flow values and state probabilities tend to settle down to stable values. This concept is known as *steady state*, meaning that the probability of being in a state is steady over a long period of time. Steady state implies that for any state, the amount of flow into that state must be equal to the amount of flow out of that state. This only makes sense. The analogy of water flowing through pipes between holding basins is valid. If the amount of flow out of a basin were larger than the flow in, conservation of flow would be violated. Over a long period of time all the water would drain out. Steady state would not exist.

This is important stuff! Read it a couple of times.

Remember that the throughput of the attendant is the average number of people that he or she processes in a minute. This is the same as summing the flows of all the arcs that are traversed due to the departure of a customer. In our example, these are all the arcs labeled with μ. To put it another way: total throughput is the sum of the individual state throughputs. The throughput in state 0 is 0, since no one is at the window. The throughput in state 1 is μ, since there is a single customer at the window who departs (i.e., is served) at rate μ. Similarly, the rates at which customers are served in states 2 and 3 are both μ. Taking all of this into consideration, we now have a way to solve for the probabilities of being in each state. Knowing these steady-state probabilities, we can determine the throughput for each attendant by calculating the sum of the flow across the arcs labeled with μ. We know the following:

- Flow along an arc is the product of the probability of being in the state from which the arc departs and the rate associated with that arc.

- The sum of all the flows into a state is equal to the sum of all the flows out of that state.
- The sum over all of the P_i's is 1, since at any time the system must be in one of the four states.
- λ and μ are known values. (λ, the customer arrival rate, is one customer per minute. μ, the service rate, is the inverse of D, which is dependent on the attendant.)

Given these facts, we can write a system of equations that can be solved to yield the steady-state values of the P_i's. For each state in the diagram there is a *balance equation*, which states that the flow into that state is equal to the flow out of that state. The balance equation for state 0 is

$$\mu P_1 = \lambda P_0$$

The left-hand side of the equation is the flow into state 0. Because only one thing can happen at a time, the only way to get into state 0 (i.e., no one at the stand) is to have a customer leave while in state 1. The probability of being in state 1 is P_1 and the rate at which customers leave state 1 is μ. Thus the flow rate into state 0 (from state 1) is μP_1. Similarly, the right-hand side of the equation is the flow out of state 0. It represents a customer arriving at the sno-cone stand when there are no customers. The balance equation for state 1 is

$$\lambda P_0 + \mu P_2 = \lambda P_1 + \mu P_1$$

The left-hand side represents the flow into state 1 and is the sum of arrivals when the system is in state 0 (i.e., λP_0) and departures when the system is in state 2 (i.e., μP_2). The right-hand side represents the flow out of state 1 due to arrivals (i.e., λP_1) and departures (i.e., μP_1).

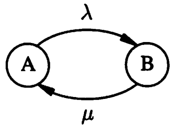
Each state has a corresponding balance equation. The other balance equations can be written in a similar manner and are given below.

These are the *global balance equations* which yield the steady-state solution.

However, the balance equations by themselves are not enough to solve the system of equations. Even though, in this case, there are four equations in four unknowns (the unknowns are the P_i's—remember that μ and λ are known values), one of the equations is redundant. The fact that all the P_i's sum to 1 provides the additional equation that allows the system to be solved. The entire system of equations for solving the diagram is

$$\begin{aligned} \mu P_1 &= \lambda P_0 && (1) \\ \lambda P_0 + \mu P_2 &= \lambda P_1 + \mu P_1 && (2) \\ \lambda P_1 + \mu P_3 &= \lambda P_2 + \mu P_2 && (3) \\ \lambda P_2 + \lambda P_3 &= \lambda P_3 + \mu P_3 && (4) \\ P_0 + P_1 + P_2 + P_3 &= 1 && (5) \end{aligned}$$

To see that one of the equations is redundant, consider the following diagram:



The balance equation for state A, flow in = flow out, is $\mu P_B = \lambda P_A$. The balance equation for state B is $\lambda P_A = \mu P_B$. These two equations are identical (i.e., one is redundant).

By using substitution of variables, the system can be solved in terms of P_0 as follows:

$$(1) \Longrightarrow P_1 = \frac{\lambda}{\mu} P_0 \qquad (6)$$

$$(2,6) \Longrightarrow P_2 = \frac{\lambda^2}{\mu^2} P_0 \qquad (7)$$

$$(3,6,7) \Longrightarrow P_3 = \frac{\lambda^3}{\mu^3} P_0 \qquad (8)$$

As you can see, equation (4) is redundant [since it reduces to $P_3 = (\lambda/\mu)P_2 = (\lambda^3/\mu^3)P_0$]. However, given equations (6) to (8) we can state (5) in terms of P_0:

$$P_0 + \frac{\lambda}{\mu} P_0 + \frac{\lambda^2}{\mu^2} P_0 + \frac{\lambda^3}{\mu^3} P_0 = 1$$

which, when rewritten as below, gives the solution to P_0.

$$P_0 = \frac{1}{1 + \lambda/\mu + \lambda^2/\mu^2 + \lambda^3/\mu^3} \tag{9}$$

Ta da! Given any values for λ and D (from which μ is calculated), (9) will yield the value of P_0, which can then be used in (6) to (8) to yield values for P_1, P_2, and P_3. Table 1.1 shows the final results when the proper values for λ and D are substituted for the cases of Fran and Bill. The steady-state probabilities $P_0 \cdots P_3$ are shown for both individuals.

> Other interesting quantities can readily be found once the steady-state probabilities P_0, P_1, P_2, and P_3 are known. For example, the average number of customers at the window is $1 \times P_1 + 2 \times P_2 + 3 \times P_3$ (i.e., 0.45 for Fran and 0.73 for Bill). Also, the rate at which customers leave in a huff for Smoothy-Cream is λP_3 (i.e., 0.025 customer per minute for Fran and 0.067 for Bill).

For example, if Mr. K were to drive by when Fran is working, he would see Fran idle $\frac{27}{40}$ (i.e., 67.5%) of the time and would see three customers at the window $\frac{1}{40}$ (i.e., 2.5%) of the time. Similarly, when Bill is working, he is idle only $\frac{8}{15}$ (i.e., 53.3%) of the time and has 3 customers $\frac{1}{15}$ (i.e., 6.7%) of the time.

Throughput, which is the average number of customers processed per minute, is calculated by summing the amount of flow across arcs labeled with μ. That is, $X = \mu P_1 + \mu P_2 + \mu P_3$. Another way to calculate throughput is as follows. The percentage of time that an attendant is idle (in steady state) is P_0. Thus the percentage of time an attendant is working (i.e., utilization) is $U = 1 - P_0$. While the attendant is working, customers are being pumped out of the system at rate μ. Therefore, if utilization is known and the service demand is known, the throughput of the attendant is given by $X = \mu U = U/D$. This relationship, known as the *utilization law*, is stated as $U = XD$. The utilization law is useful because, given any of the two variables U, X, or D, the third is easily computed. The amount of revenue per minute is simply the product of the throughput for each attendant and the amount of money made per customer, which is \$3. The cost of each employee per minute is subtracted from the revenue gained per minute to yield the profit per minute. These calculations are shown in Table 1.1.

Given these results, we would advise Mr. K to hire Fran instead of Bill since she will bring in a net profit of 3¢ more per minute than Bill. The moral of the story is that first impressions are not always valid and that the assumptions made must be clearly understood since the outcome depends upon them.

TABLE 1.1 Results of Comparison of Fran and Bill

Results	Fran	Bill
λ (arrival rate)	1	1
D (service demand)	1/3	1/2
Steady state probabilities:		
P_0	27/40	8/15
P_1	9/40	4/15
P_2	3/40	2/15
P_3	1/40	1/15
U (utilization)	$1 - P_0$ $= 1 - 27/40$ $= 13/40$	$1 - P_0$ $= 1 - 8/15$ $= 7/15$
X (throughput)	U/D $= 13/40 \times 3$ $= 39/40$	U/D $= 7/15 \times 2$ $= 14/15$
Revenue per minute	(39/40)($3) ≈ $2.93	(14/15)($3) ≈ $2.80
Employee cost per minute	$0.20	$0.10
Profit per minute	$2.93 – $0.20 = $2.73	$2.80 – $0.10 = $2.70

1.3 SUMMARY

Table 1.2 gives a brief summary of the notation and important formulas introduced in this section.

TABLE 1.2 Summary

λ	arrival rate
D	service demand
μ	service rate ($\frac{1}{D}$)
U	utilization
X	throughput
$U = XD$	utilization law

EXERCISES

1.1 * Bill really wants this job, so he tells Mr. K that he will work for less than $6 per hour. What pay rate will he have to accept in order to be competitive with Fran?

1.2 ** Mr. K has interviewed another prospective employee, Bob. Bob has experience with sno-cones and can complete a transaction in 10 seconds. However, he charges

$18 per hour for his expertise. How much profit does Bob make per minute, and how does he rank with respect to Fran and Bill?

1.3 ** Suppose that Bill improves his service time by 5 seconds. Thus, he completes a transaction in 25 seconds. How does he compare to Fran now?

1.4 **** Assuming that all other problem parameters are as originally stated, what is the minimum amount of time by which Bill must improve his service time in order to be competitive with Fran?

1.5 * Mr. K decides that his price is too high and lowers it. How much will he have to charge per sno-cone in order for Bill and Fran to be competitive? (Note: all other problem parameters are as originally stated.)

1.6 ** Suppose that the customer arrival rate changes to 1 customer every 2 minutes. Which applicant is more profitable, Bill or Fran? By how much?

1.7 **** For what arrival rate is Bill competitive with Fran? What are the ranges of arrival rates for which a) Bill is more cost-effective than Fran; b) Fran is more cost-effective than Bill?

1.8 *** What if the maximum line length changes from 3 to 4? Who is more cost-effective, Fran or Bill? By how much?

1.9 *** Assuming all other parameters are as originally stated, for what range of line lengths will: (a) Bill be more cost-effective; (b) Fran be more cost-effective? Justify your answer.