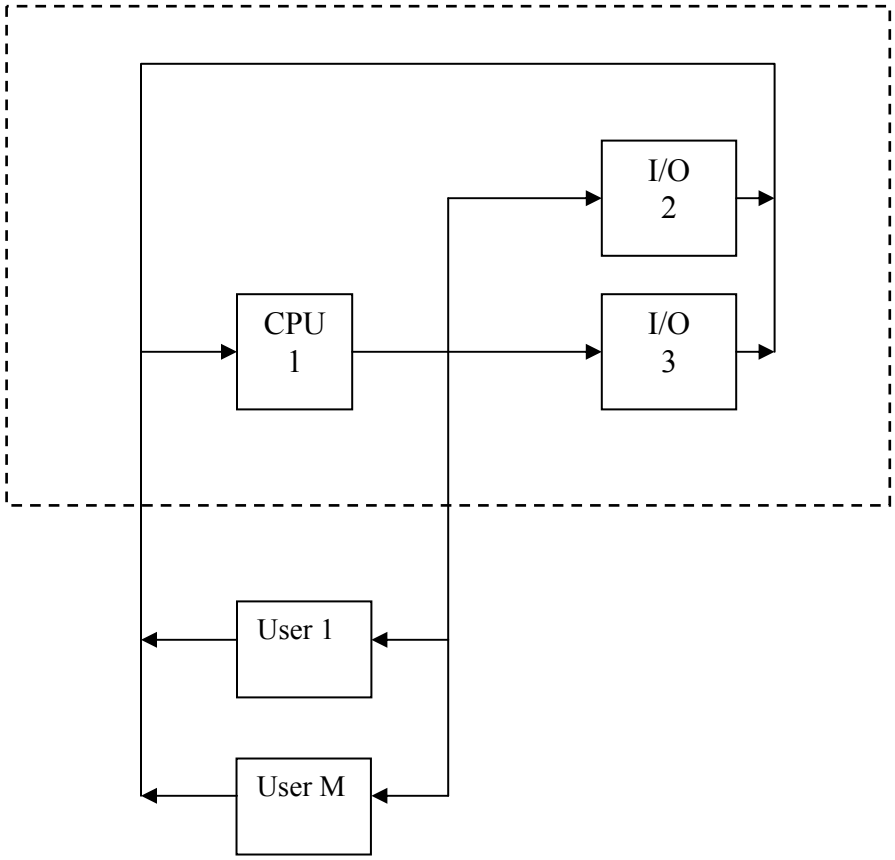
Notes on System Performance
COMP755 Advanced Operating Systems

As a program executes, it moves from device to device. Programs or tasks move between the different program states.



In the above example, the task starts by using the CPU. After an interval of using the CPU, the task will use either one of two I/O devices. After using the I/O device, it will then use the CPU. Each device may have a waiting queue where the task will have to wait if the device is busy. Eventually the task will complete. This is indicated in the diagram by a flow of control returning to a user. After some Think time, the user will enter another request and the process will repeat.

There are several quantities of the system that can be directly measured.

T	length of an observation period.
A	number of arrivals during the observation period.
B	total amount of time the system was busy.
C	number of completions during the observation period.

For a long observation period, the number of arrivals should approximately match the number of completions. Therefore A = C for long term steady state systems. We can derive several quantities from the above measured values:

λ	A/T	the arrival rate
ρ	B/T	the utilization (fraction of time the system was busy)
R	B/C	the mean system response time (how long it took to complete a task)

All of the above derived or measured quantities can apply to a single device or the system as a whole. When we refer to a device, we will put a subscript on the value. For example, λ_2 would refer to the rate at which tasks arrive at I/O device 2. A task arriving at any device may be queued before it is serviced. We will refer to the user (or any initiator of tasks) as device 0. λ_0 would refer to the rate at which tasks are submitted to (or completed by) the system.
Tasks flowing through the circular path illustrated above will not always follow the same route. Some tasks will leave the CPU and go to I/O device 2. Some will go to I/O device 3 while others will leave the system. Some devices will be visited more often than others. We can define the visit ratio as:

$$V_i = \frac{\lambda_i}{\lambda_0}$$

If we know the visit ratio for a device, we can compute the arrival rate to the device

$$\lambda_i = V_i * \lambda_0$$

This is the *Forced Flow Law*. It states that the flow in any one part of the system determines the flow everywhere in the system.

Example: Consider a transaction system where each transaction generates an average of 5 disk requests and disk throughput is measured as 75 requests/second. What is the system throughput?

Let λ_i = 75 requests/second and let Vi = 5 request/transaction.

$$\lambda_0 = \frac{\lambda_i}{V_i} = \frac{75}{5} = 15$$

Although we know very little about the system, we can determine that the system is performing 15 transactions per second.

System Response time can be calculated by:

$$R = \sum_{i=0}^{K} V_i * R_i$$

Which states that the system response time is the sum of the number of times a program visits each device times the time spent at each device.

The response time of a device, R_i includes the time it takes the device to perform the function plus any time waiting for other requests to complete. The actual time it takes a device to perform a function is known as the service time, S_i. The service time is the average time it takes to perform a function (such as read an average block from the disk) when the system is otherwise idle. The service time is always less than or equal to the response time, $S_i \leq R_i$

Little's Law states that the average number of tasks using a device or waiting for it is equal to the device's arrival rate times the time spent at the device.

$$n_i = \lambda_i * R_i$$

Assume each user of an interactive timesharing system alternates between thinking about what to do and submitting requests. Each user thinks for an average of Z time units before submitting another request. The user then waits until the system responses before thinking for Z time units. The total time for the think-wait cycle is Z+R. By *Little's Law* the average number of users, M, is

$$M = (Z + R) * \lambda_0$$

We can rearrange this equation to give the *Interactive Response Time Formula*

$$R = \frac{M}{\lambda_i} - Z$$

Example: Consider a time sharing system with the following measurements:

- Each task generates 20 disk requests.
- The disk utilization in 50%.
- The mean disk service time is 25 msec.
- There are 25 terminals.
- The average think time is 18 seconds.

We can calculate the response time after first calculating the throughput. Let subscript I refer to the disk. The forced flow and utilization laws imply

$$\lambda_0 = \frac{\lambda_i}{V_i} = \frac{\rho_i}{V_i * S_i} = \frac{0.5}{20 * 0.025} = 1$$

The system can complete 1 task per second. From the interactive response time formula

$$R = \frac{M}{\lambda_i} - Z = \frac{20}{1} - 18 = 2\,\text{sec}$$

The average response time is 2 seconds.

The utilization of a device is the fraction of time the device is busy. You can compute individual devices utilization from the devices arrival rate and the mean service time. Note that utilization is always a number between zero and one.

$$\rho_i = \lambda_i * S_i$$

In any computer system, one device will typically saturate before the other devices when the load increases. This is the bottleneck device. A device saturates when its utilization is 100%. From the service time of a device, you can calculate the arrival rate that will cause it to saturate. At saturation,

$$\lambda_i = 1 / S_i$$

Example: A disk drive can perform the average I/O request in 10.0 ms. What is the maximum number of requests per second that can be made to the disk?

$$\lambda_i = 1 / S_i = 1 / 0.010 = 100$$

The saturation point of a system occurs when any device in the system is saturated. If the average transaction makes V_i requests of device i, then the maximum number of transactions that the system can support is

$$\lambda_0 = \min(1/V_1S_1, 1/V_2S_2, ..., 1/V_nS_n)$$