Abstract:
A simple cyclic-queue model of a multiprogramming system with a fixed number of tasks is analysed in its steady state. Expressions for queue-size distribution, average rate of job-completions, average stay-in-the-system time are derived. A measure of system efficiency alternative to processor utilisation is suggested and optimal values for the degree of multiprogramming are given for various values of the parameters.

Non-priority multiprogramming systems under heavy demand conditions from a customer's point of view

by

I. Mitrani

TECHNICAL REPORT SERIES

Series Editor: Dr. B. Shaw
Number 11
August, 1970

© 1970 University of Newcastle upon Tyne
Printed and published by the University of Newcastle upon Tyne, Computing Laboratory, Claremont Tower, Claremont Road, Newcastle upon Tyne, NE1 7RU.
Mitrani, Israel

Non-priority multiprogramming systems under heavy demand conditions from a customer's point of view. By I. Mitrani. (University of Newcastle upon Tyne, Computing Laboratory, Technical Report Series, No. 11).
Newcastle upon Tyne, University Computing Laboratory, 1970.

12

Added entries
NEWCASTLE. University. Computing Laboratory.
Technical Report Series. 11

Suggested classmarks (primary classmark underlined)
Library of Congress:
Dewey (17th):
U.D.C.:
621.381952
681.322.02

Suggested keywords
CENTRAL
PROCESSOR
CHANNEL
COMPUTING SYSTEMS
MULTIPROGRAMMING
QUEUES
TURNAROUND

Abstract
A simple cyclic-queue model of a multiprogramming system with a fixed number of tasks is analysed in its steady state. Expressions for queue-size distribution, average rate of job-completions, average stay-in-the-system time are derived. A measure of system efficiency alternative to processor utilisation is suggested and optimal values for the degree of multiprogramming are given for various values of the parameters.

About the Author
Mr. I. Mitrani is a Lecturer in the University of Newcastle upon Tyne Computing Laboratory which he joined in 1968 as a Programming Adviser after studying at the Technion, Haifa, Israel.
Introduction

This paper is concerned with a single processor, single channel computing system, with a fixed number of tasks being served by the processor and the channel on a first-in-first-out basis. A similar model was solved numerically by Wallace and Mason [1] and the effect of the system overhead on the degree of multiprogramming was studied by Lewis and Shedler [2]. Both these works are concerned with the system from the point of view of the computing manager, the quantity of interest there being the processor utilisation. If the system overhead is ignored, then a high degree of multiprogramming usually implies a high degree of processor utilisation and therefore a high rate of job completions. However, it also implies that every job (when there are no priorities) takes on the average a longer time to run. In systems where there is normally a queue of jobs waiting to be run, fast turnaround becomes an important and desirable feature. We suggest a crude and simple measure of turnaround and compute the values of the degree of multiprogramming that minimise it for different values of the parameters.

The model

The execution of a job consists of alternative intervals of 'calculating' and 'input/output'. Our queueing system consists of two servers working concurrently, which we call for convenience 'Processor' and 'Channel' or P and C respectively. When a customer (or a 'job') enters the system he joins the queue at P which is served by P in order of arrival. After receiving an interval of service by P, or 'a P-service' the customer leaves the system with probability \( \alpha \) \((0<\alpha<1)\), and joins the queue at C with probability \(1-\alpha\). The last assumption is equivalent to the assumption that the number of P-services required by a customer is geometrically distributed with parameter \( \alpha \).

\[
P(\text{No. of P-services} = m) = \alpha(1-\alpha)^{m-1}
\]

The 'Channel' also serves its queue in order of arrival and upon receiving a C-service the customers join again the queue at P. We assume that the lengths of the P-services are identically distributed random variables with exponential p.d.f. \( f_p(t) = \mu_1 e^{-\mu_1 t} \) and the lengths of the C-services are identically distributed random variables with p.d.f. \( f_c(t) = \mu_2 e^{-\mu_2 t} \), and also that these random variables are mutually independent. At any given moment there are exactly \( N \) \((N \geq 1)\) customers in the system (fixed number of tasks environment). This means that at any given moment the state of the
system is fully described by one number $n$ - the number of customers at the P-queue (including the one that is receiving service); then, the number of customers at the C-queue is $N-n$. No past history is necessary because the system is Markovian. Figure 1 illustrates the flow of customers in the system.

We assume that the system operates under heavy demand conditions, i.e. that there is a sufficient number of jobs requiring service to ensure that every time a job leaves the system another job enters it immediately (and joins at the end of the P-queue).

**Steady-state analysis of the model**

Let $p_n(t)$ denote the probability that at the moment $t$ the system is in a state $n$ ($n$ customers at the P-queue including the one receiving service). Since the system can only be in a finite number of states, the limits $p_n = \lim_{t \to \infty} p_n(t)$ exist (page 87 in Saaty [7]). Familiar techniques easily yield the following balance equations

\begin{equation}
(1-\alpha)\mu_1 p_1 = \mu_2 p_0 \\
(1-\alpha)\mu_1 p_n = [(1-\alpha)\mu_1 + \mu_2]p_{n-1} - \mu_1 p_{n-2} \quad n = 2, 3, \ldots N
\end{equation}

which, in conjunction with $\sum_{n=0}^{N} p_n = 1$ give

\begin{equation}
p_n = \frac{1 - \rho}{1 - \rho^{N+1}} \quad \rho^n \quad n = 0, 1, 2, \ldots, N
\end{equation}

where $\rho = \frac{\mu_2}{(1-\alpha)\mu_1}$. Of course the right-hand sides of (2) are defined only for $\rho \neq 1$. It is easily seen that for $\rho = 1$ \( p_n = \frac{1}{N+1} \) for all $0 \leq n \leq N$. If we want to call $\rho$ by name, we may call it 'Processor traffic intensity'; $\rho < 1$ means that the rate at which jobs join the C-queue (when $P$ is not idle) is greater than the rate at which jobs join the P-queue (when $C$ is not idle). When $\alpha$ is small (which is the case in most real systems) $\rho < 1$ means also that, in a sense, the Processor is faster than the Channel.

The Processor utilisation $U$ is equal to the proportion of time $P$ is not idle

\begin{equation}
U = 1 - p_0 = \frac{\rho - \rho^{N+1}}{1 - \rho^{N+1}}
\end{equation}

If we regard $U$ as a function of the degree of multiprogramming $N$, we can see that $U$ always increases with $N$ and when $\rho < 1$, $U$ approaches $\rho$ when $N$ approaches infinity, whereas when $\rho \geq 1$ $U$ approaches 1 when $N$ approaches infinity.
We can find now the average length $S$ of the intervals between successive departures from the system. Take a very long interval of time $I$. The Processor will be busy for a period $U.I$ during that interval. While $P$ is busy, customers leave the system at a rate of $\alpha_{1}$, therefore during the interval $I$, $\alpha_{1} UI$ departures will occur. From here we derive

$$S = \frac{1}{\alpha_{1}} \frac{1}{U} = \frac{1}{\alpha_{1} U} \frac{1}{\rho} - \frac{\rho^{N+1}}{\rho} \quad (4)$$

Our next task will be to find the average time a job spends in the system, i.e. the average interval between the admission of the job to the system and its departure from the system. For this we need three Khinchine-type results.

Let $n_{1}$ be the state of the system at the moment when the $i^{th}$ customer enters it (and the $(i-1)^{th}$ customer leaves it). Denote by $M_{1}$ the Markov chain formed by $n_{1}$; the states of $M_{1}$ are $\{1, 2, \ldots, N\}$.

**Theorem 1.** The steady-state distribution of $M_{1}$ is given by

$$\Pi_{n_{1}} = \frac{p_{n}}{1 - p_{0}} = \frac{1 - \rho}{1 - \rho^{N}} \cdot \rho^{n-1} \quad n = 1, 2, \ldots, N \quad (5)$$

Let $n_{12}$ be the state of the system at the moment when, for the $i^{th}$ time, a customer leaves the P-queue to join the C-queue. More precisely, the moment in question is immediately after the joining of the C-queue. Denote by $M_{2}$ the Markov chain formed by $n_{12}$; the states of $M_{2}$ are $\{0, 1, \ldots, N-1\}$.

**Theorem 2.** The steady-state distribution of $M_{2}$ is given by

$$\Pi_{n_{2}} = \frac{p_{n}}{1 - p_{N}} = \frac{1 - \rho}{1 - \rho^{N}} \cdot \rho^{n} \quad n = 0, 1, \ldots, N-1 \quad (6)$$

Let $n_{13}$ be the state of the system immediately after the moment when, for the $i^{th}$ time a customer has left the C-queue and joined the P-queue. Denote by $M_{3}$ the Markov chain formed by $n_{13}$; the states of $M_{3}$ are $\{1, 2, \ldots, N\}$.

**Theorem 3.** The steady-state distribution of $M_{3}$ is given by

$$\Pi_{n_{1}} \text{ in (5). The proofs of these theorems are given in the Appendix.}$$

From theorems 1 and 3 it follows that, every time a customer joins the P-queue, there are $N_{1}$ customers there (including himself), where $N_{1}$ is obtained from
\[ N_1 = \frac{1 - \rho}{1 - \rho N} \sum_{n=1}^{N} n \rho^{n-1} = \frac{1}{1 - \rho} - \frac{N \rho^N}{1 - \rho} \]  

(7)

From theorem 2 it follows that, every time a customer joins the C-queue, there are \( N \) customers there (including himself) where \( N \) is obtained from

\[ N = \frac{1 - \rho}{1 - \rho^N} \sum_{n=0}^{N-1} (N-n) \rho^n = \frac{N}{1 - \rho^N} - \frac{\rho}{1 - \rho} \]  

(8)

The interval of time a job spends in the system consists of several (possibly none at all) full cycles of 'a stay in the P-queue followed by a stay in the C-queue' plus a final stay in the P-queue. The average number of full cycles is \( \frac{1 - \alpha}{\alpha} \), therefore, we can write the following expression for \( \varpi \)-the 'stay-in-the-system' time

\[ \varpi = \frac{1 - \alpha}{\alpha} \left( N_1 \frac{1}{1 \mu_1} + N_2 \frac{1}{\mu_2} \right) + N \frac{1}{1 \mu_1} \]  

(9)

It can be seen easily that (9) is equivalent to

\[ \varpi = \frac{N}{\alpha_1 \rho} \cdot \frac{1 - \alpha^N}{1 - \rho^N} = N.S \]  

(10)

where \( S \) is defined by (4).

We shall proceed now to define a measure of turnaround. In most computing installations, during peak hours of high demand, there is a queue of jobs, either in the form of card decks or on a spooling device, awaiting execution. The length of this queue depends largely on the number of users in the installation, and for a given time of day may be assumed to be constant. Let us denote it by \( K \). Assuming further that the user is mainly interested in having as many runs per unit time as possible (this is the case, for instance, when developing a program), the quantity he will want minimised is the time between joining the queue and leaving the system.

This time we call turnaround and it is, on the average, equal to

\[ T = (K + 1)S + \varpi \]  

(11)

where \( S \) is again the average time between successive departures from the system. In our case (4), (10) and (11) give

\[ T = \frac{(K + N + 1)}{\alpha_1 \rho} \cdot \frac{1 - \rho^N}{1 - \rho^N} \]  

(12)

Strictly speaking, in writing that the average wait in the queue is equal to \((K + 1)S\) we implicitly assume that the steady-state forward recurrence
Figure 2.

Optimal value of $N$ as a function of $\phi$ for $K=20, 50, 150, 500$
APPENDIX

Proof of Theorem 1. We shall call a 'P-step' the span between the beginnings of two consecutive P-services. Thus a P-step comprises either a P-service or a P-service and a Processor idle period. We can express the transition probability matrix $Q_1$ of the Markov chain $M_1$ in the form

$$Q_1 = \sum_{k=1}^{\infty} \alpha(1-\alpha)^{k-1}Q_k$$  \hspace{1cm} (13)

where $Q_k$ is the matrix of the transition probabilities conditioned upon there being $k$ P-steps between two departures from the system. The $(i,j)^{th}$ element of $Q_k$ is the probability that just before the end of a stretch of $k$ P-steps the system will be in the state $j$, given that at the beginning of the stretch the system was in the state $i$ and no job left the system during the stretch.

It can be found directly that $Q_1$ has the form

$$Q_1 = \begin{bmatrix}
\eta & \eta(1-\eta) & \ldots & \eta(1-\eta)^{N-2} & (1-\eta)^{N-1} \\
0 & \eta & \ldots & \eta(1-\eta)^{N-3} & (1-\eta)^{N-2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & \eta & (1-\eta) \\
0 & 0 & \ldots & 0 & 1
\end{bmatrix}$$  \hspace{1cm} (14)

where $\eta = \mu / (\mu + \mu_P)$ is the probability that the P-service will complete before the C-service, provided that both P and C are working. For $k \geq 2$ we can write

$$Q_k = Q_A Q_B^{k-2} Q_1$$  \hspace{1cm} (15)

where the product is the 'row by column' matrix product, power of zero yields the identity matrix $I_N$ and $Q_A$ and $Q_B$ are given by

$$Q_A = \begin{bmatrix}
\eta+(1-\eta)\eta & \eta(1-\eta)^2 & \ldots & \eta(1-\eta)^{N-2} & (1-\eta)^{N-1} & 0 \\
\eta & \eta(1-\eta) & \ldots & \eta(1-\eta)^{N-3} & (1-\eta)^{N-2} & 0 \\
0 & \eta & \ldots & \eta(1-\eta)^{N-4} & (1-\eta)^{N-3} & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \ldots & \eta & (1-\eta) & 0 \\
0 & 0 & \ldots & 0 & 1 & 0
\end{bmatrix}$$  \hspace{1cm} (16)
and

\[
Q_B = \begin{bmatrix}
\eta(1-\eta) \eta^2 & \ldots & \eta^N & (1-\eta)^{N-1} & 0 \\
\eta & \eta(1-\eta) & \ldots & \eta^N & (1-\eta)^{N-2} & 0 \\
0 & \eta & \ldots & \eta^N & (1-\eta)^{N-3} & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \ldots & \eta & (1-\eta) & 0 \\
0 & 0 & \ldots & 0 & 0 & 0
\end{bmatrix}
\]  
(17)

The need for introducing \( Q_A \) and \( Q_B \) arises from the slightly different treatment that must be afforded to the first of the \( k \) F-steps, the next \( k-2 \) steps and the last step.

Substituting (15) in (13) we obtain

\[
Q_1 = \alpha Q_1 + \alpha(1-\alpha)Q_AQ_1 + \alpha(1-\alpha)^2Q_AQ_BQ_1 + \ldots
\]

\[
= \alpha Q_1 + \alpha(1-\alpha)Q_A\left[ \sum_{k=0}^{\infty} (1-\alpha)^k Q_B \right]Q_1
\]

\[
= \alpha Q_1 + \alpha(1-\alpha)Q_A[I_N - (1-\alpha)Q_B]^{-1}Q_1
\]

(18)

If we add and subtract from the right-hand side of (18) the term \( \alpha \tilde{I}_N[I_N - (1-\alpha)Q_B]^{-1}Q_1 \) where \( \tilde{I}_N \) is the matrix

\[
\tilde{I}_N = \begin{bmatrix}
\ldots & \ldots & \ldots & \ldots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
I_{N-1} & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \ldots & \ldots & \ldots & 0(1-\alpha)I
\end{bmatrix}
\]

(19)

and noting that \( [\tilde{I}_N - (1-\alpha)Q_A] = [I_N - (1-\alpha)Q_B] \) we arrive at

\[
Q_1 = \alpha \tilde{I}_N[I_N - (1-\alpha)Q_B]^{-1}Q_1
\]

(20)

Let \( \Pi \) denote the row-vector \( (\Pi_1, \Pi_2, \ldots, \Pi_N) \) defined in (5). Since the elements of \( \Pi \) sum up to 1, theorem 1 will be proved if we show that \( \Pi_1 Q_1 = \Pi_1 \) or that

\[
\Pi_1 Q_1[I_N - (1-\alpha)Q_B] - \Pi_1 \alpha \tilde{I}_N = 0
\]

(21)
We note first that

\[
Q_1^{-1} = \begin{bmatrix}
1/\eta & -(1-\eta)/\eta & 0 & \ldots & 0 & 0 \\
0 & 1/\eta & -(1-\eta)/\eta & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 1/\eta & -(1-\eta)/\eta \\
0 & 0 & 0 & \ldots & 0 & 1
\end{bmatrix}
\]

(22)

Now, a direct substitution of (5), (17), (19) and (22) in (21) establishes it without a great deal of effort.

**Proof of Theorem 2.** The transition probability matrix \(Q_2\) of the Markov chain \(M_2\) is given by

\[
Q_2 = \begin{bmatrix}
\xi & \xi(1-\xi) & \ldots & \xi(1-\xi)^{N-2} & (1-\xi)^{N-1} \\
\xi & \xi(1-\xi) & \ldots & \xi(1-\xi)^{N-2} & (1-\xi)^{N-1} \\
0 & \xi & \ldots & \xi(1-\xi)^{N-3} & (1-\xi)^{N-2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & \xi(1-\xi) & (1-\xi)^2 \\
0 & 0 & \ldots & \xi & (1-\xi)
\end{bmatrix}
\]

(23)

where \(\xi = (1-\alpha)\mu_1/[\mu_1 + (1-\alpha)\mu_1]\) is the probability that a job from the \(P\)-queue will join the \(C\)-queue before the current \(C\)-service completes, provided that both \(P\) and \(C\) are working. Equation (23) and the value of \(\xi\) can be explained by the fact that when \(P\) is not idle, the intervals between consecutive joinings of the \(C\)-queue are distributed exponentially with parameter \((1-\alpha)\mu_1\). The first two rows of \(Q_2\) are the same because if when a job joins the \(C\)-queue the system is in the state zero, the system will have to be in the state 1 before the next job can join the \(C\)-queue.

It is easy to see that \(\pi\), the row vector of the \(M_2\) steady-state probability distribution defined by (6), satisfies \(\pi Q_2 = \pi\).

**Proof of Theorem 3.** This proof is exactly similar to the proof of theorem 2. We replace in that proof \(Q_2\) with \(Q_3\), the transition probability matrix of \(M_3\), given by
\[ \mathbf{Q}_3 = \begin{bmatrix} \xi & (1-\xi) & 0 & \cdots & 0 & 0 \\ \xi & \xi(1-\xi) & (1-\xi) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \xi^{N-1} & \xi^{N-2}(1-\xi) & \xi^{N-3}(1-\xi) & \cdots & \xi(1-\xi) & (1-\xi) \end{bmatrix} \]  

and \( \mathbf{\Pi}_3 \) with \( \mathbf{\Pi}_2 \), the steady-state probability distribution vector of \( \mathbf{M}_3 \).

References


