MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 15/1975.

REAL TIME OPERATING SYSTEMS PROBABILISTIC MODELS

Adam WOLISZ

Department of Complex Automation Systems of the Polish Academy of Sciences

I. INTRODUCTION

The majority of lectures presented during this winter school dealed with mathematical problems of big computers operating systems (O.S.) for data processing or scientifical computations.

This paper aimes to discuss some problems connected with control of technological plants, one of the most rapidly growing areas for computers applications. The following considerations have been written rather from the position of a control system designer, then the computer manufacturer.

Basing on such a biased point of view some classification of real – time O.S. will be suggested and their mathematical models will be discussed.
Software used in process control systems can be divided into three parts:

- basic software, including computer oriented O.S. We shall include here programs for system generation, drivers for various I/O devices, loaders, compilers, memory allocation routines as well as interrupt handlers. All those are usually provided by a computer manufacturer, together with apropriate hardware.
  Often the computer oriented operating systems enable various modes of operation, such for example as batch processing for background activities or time sharing features.
- Utility programs, prepared for data acquisition, direct digital control optimization and so on. They are usually based on some mathematical models of technological processes, and are different for various plants, however they can be created using some languages or programme packages for process control. Those programes are usually prepared by a specialized designe office or software manufacturer in close cooperation with the user.
- Plant oriented O.S. enabling in an arbitrary time epoch to recognize the situation in the computer and in the controlled plant, so as to choose an proper program to be executed at this time.

Situation of the computing system is understood as a set of signals denoting demands for execution of various programmes, which are generated either by the system itself or more frequently by the environment. The plant oriented O.S. determines the service disciplines imposed in the system utilizing all features provided by the hardware and basic software as interrupts (including their enabling and deaseabling mechanism) real time clock etc. Implementation of the service discipline as well as switching the control from one programme to another is usually connected with some overhead caused mainly by the basic software utilization.

Implementation of any utility programme is possible using various plant oriented O.S. and various hardware together with it is basic software. The decision is usually made by the system designer.

In the following considerations we shall restrict ourselves only to the plant oriented O.S.

II. Requirements for process control O.S.

The examination of different O.S. is usually done with respect to following requirements:

a) The possibility of obtaining proper response times, which is an essential one for control applications,

b) Flexibility, understood as a set of features enabling systems developement in the sense of introducing sequentially additional functions as for example data logging, control of some technological processes, inventory control, and optimization of the full instalation. Software developement should be possible without enforcing any changes in previously debugged and well running parts.

c) Reliability and foult tolerancy. One must be prepared for existance of some bugs in new introduced programmes. System's structure should minimize their influance on the other parts of software. There can also occure some hardware demages and in this case one of the following actions should be started:

- system reconfiguration and further work in the full range of functions (in the most developed systems).
- System reconfiguration and fulfilling only the most important functions.
- Fixing all outputs in some predetermined "secure" positions and switching over to a manual control.

In any case some diagnostic tests should be executed and their results edited.
It is rather difficult to express properly requirements b) and c) in a formal way however such attempts are done mainly on the basis of graph theory and reliability theory. Some notices in this area can be also done for various structures basing on programmers and systems analysts experience.
As far as response times and utilization factor of computing systems are concerned the queuing theory approach was very usefull.

III. Real time O.S. queuing models – general remarks

An overview of deterministic congestion models was presented in paper [1] so now we shall constrain ourselves only to probabilistic ones.
An essential problem for any queuing problem is determining the type of sources that means theirs dimension N defined as the number of demands which can be generated without any service being completed and their interarrival time distribution. Investigation of various queuing disciplines for the general case of N dimensional sources is fairly difficult, and as the matter of fact not often met in practical cases. Therefore the boundary case, an infinite dimensional source which leads to an open queuing model is usually considered. In the control applications quite an important seem to be one–dimentional sources fitting perfectly to working conditions of some sensors (no technogical parameter's limit value can be twice violated without any control action being fulfilled in the meantime, no piece of material can disappear until it is taken away . . .). Both those cases give some upper and lower limit for solutions of the general N–dimensional case.
The interarrival time distributions are usually assumed to be exponential ones, because of their very convenient Markov property. The basic question if it is a good approximation for processes occuring in computer systems was investigated in [2].
The author proved positive the statistical hypothesis of interarrival times being exponentialy distributed using experimental data as well as sofisticated statistical tests, and also demonstrated that results obtained from a queuing model under such an assumption are very close to reality. The vast majority of models can be described mathematically for the case of arbitrary service distribution and as such there are generally valid.
What we usually ask for is the mean responce time. However for control applications the variance and the probability of exceeding some predetermined value can be also of great importance. Therefore the distribution of response time or more frequently it's Laplace–Stieltjes transform is exactly what we need. It is quite similar problem with the utilization factor for a computer system.

IV. Real time O.S. classification

A detailed description of O.S. structures in terms of their service disciplines will be possible only after some closer investigation of request types [3].
The demands for service can be devided into three main groups as presented on fig. 1.
The internal requests include parity checking, power failure, CPU real overflow etc. as well as requests for communication with auxiliary memory originating in the actually executed programme.
The real time clock requests have a well known interpretation. The external requests can be devided into two types.
First type includes requests which service consists only of memorizing or editing some pieces of data connected with a predetermined location.

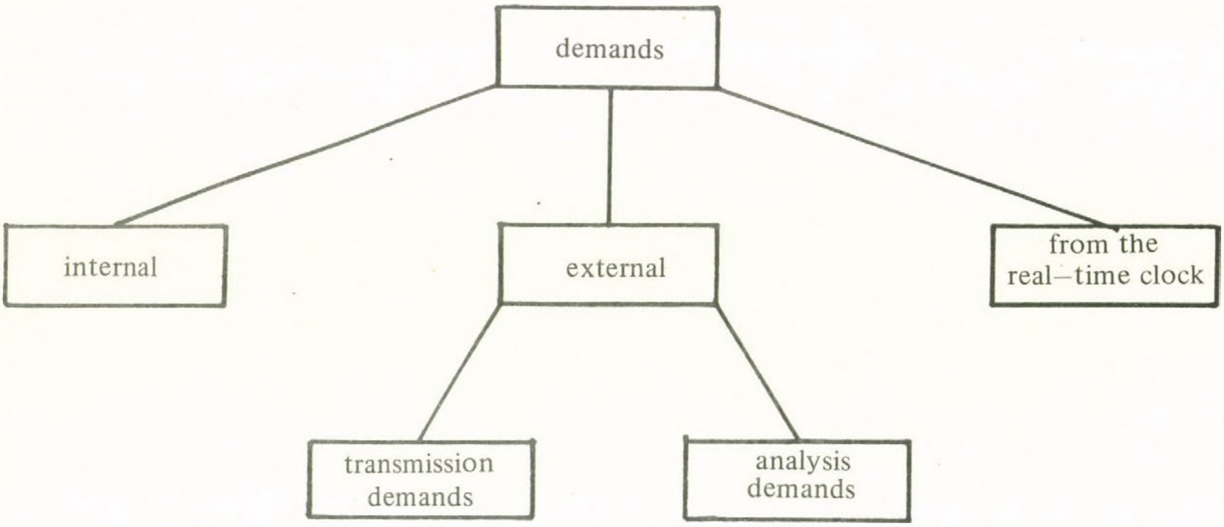


Fig. 1. Classification of demands in real time O.S.

Usually after completing of such a transmission a return to previously executed programme is desired (if no another request of higher priority occured meanwhile). These requests having a short service time shall be referred to as transmission requests.
The second type includes requests which service is not connected with data transmission but represents merely signals for activisation of a proper programme. These requests shall be referred to as analysis requests.

Due to their nature demands from two former groups have usually a preemptive priority over external calls which is imposed utilizing the interrupt feature while with regard to the third group of requests service discipline three types of O.S. can be distinguished:

1) *Event – oriented O.S.* which are nowadays the most frequently used ones. The priority of any request is evaluated immidiately after it's generation (it means when the appropriate event happens) and the highest priority request is chosen for service. Scheduling algorithms applied in control systems are usually based on the fixed priority approach in contradiction to big general purpose computers, where dynamical priorities or at least periodical reevaluation of fixed priorities are often to be spotted.
The first approximate description of an event–oriented system gives a well known congestion model with several infinite – dimensional sources served under the preemptive – resume discipline (e.g. [4]).
This model is pretty far from reality because three main factors are anticipated namely:

a) existance of nonpreemptive parts in the executed programmes

b) overhead connected with preemption

c) finite type of sources.

Taking into account the first of this factors leads to a multiphase service systems with preemptive and nonpreemptive phases successively [5].
Some attempts to consider the second factor were done also for the case of preemptive overhead [5] [6].
However in practical cases we should rather assume nonpreemptive structure of the overhead time. Some research in this area aiming at obtaining a full description of a multiphase system in which every service consists of some set of preemptive and nonpreemptive parts with nonpreemptive overhead added to every service interruption as well as resuming of the preempted programme execution was done in the Department of Complex Automation Systems, Polish Academy of Sciences, and the results would be published in the nearest future. Also an interesting simulational research reported in [7] should be mentioned here.
As far as the finite type of sources is concerned one classical system, the repairman problem has been deeply investigated. However it seems that a much more interesting case is a set of one–dimensional sources, with various parameters of the generation rate nad various service – time distributions. Such a scheme which fits to the presented also by Tomko [8] multiprogramming computer system with a constant number of non–homogeneous jobs having independent I/O facilities has been investigated in [9], [10] for the preemptive resume, head–of–the–line and discretionary priority disciplines. (Such a discipline which is preemptive before any tagget demand complets an amount of service equal to some predetermined value, and than it becomes the head of the line discipline until the service is completed is called discretionary discipline. It is a special case of a multiphase system.) The utilization factor of the system as well as the probability density function of waiting time, response time an occupation time has been found (an analysis for the preemptive – resume discipline is given in the Appendix).
All this models are studied either by the imbedded Markov chains analysis or using the supplementary variable technique. It seems to the author that the second one combined with Gaver's concept of completion time is more hopefull fo various priority disciplines. As it was shown in [4] using this method one may obtain the full description of busy period process and afterwords due to the renewal theory the general process description can be found. In that way the transient state of the system can be also investigated. Some more research in this field connected mainly with overhead time consideration especialy in the finite – dimension sources models in needed.

2) The Time – oriented O.S. which characteristic feature is an analysing the situation in the controlled process with regard to all external demands in arbitrary chosen by the scheduling algorithm time epochs independently of the moment of their generation (without making use of the interrupt feature). Among the scheduling algorithms applied in such a system one can distinguish four groups which will be discussed here.

a) Synchronious O.S.

In this solution the utility software is divided into several segments being executed in a cyclic manner. The sequence of their execution as well as time quantum for every

segment is predetermined by the system designer, while synchronization is assured by the clock. When the predetermined time quantum elapses all the vital information is stored, and control is passed to a next segment.
Time schedule for such an O.S. is plotted on Fig. 2.

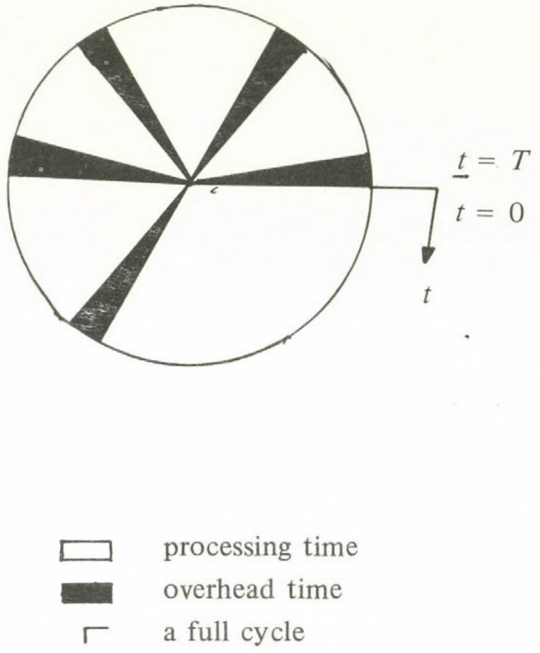
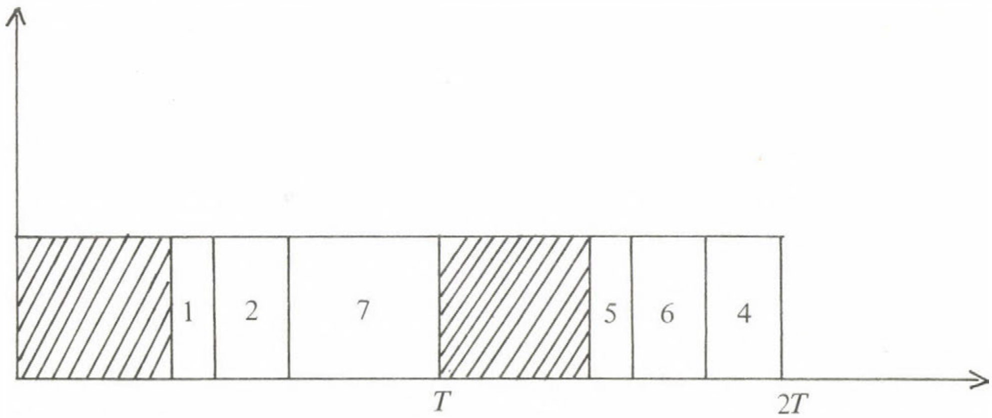


Fig. 2. Time schedule for an synchronious O.S.

An ettempt to present some analitical results for this system as well as evaluation of it's features is given in [11].

b) O.S. with periodical cycle

This scheduling algorithm can be considered as a special case of the previous one, but it should be discussed due to it is wide range of applications. As it is presented in the Fig. 3. every T seconds a fixed programme package is executed, which consists of some highly time dependent jobs (as for example data logging, direct digital control etc.) and the rest of time is devoted to non periodical jobs served under any introduced discipline.

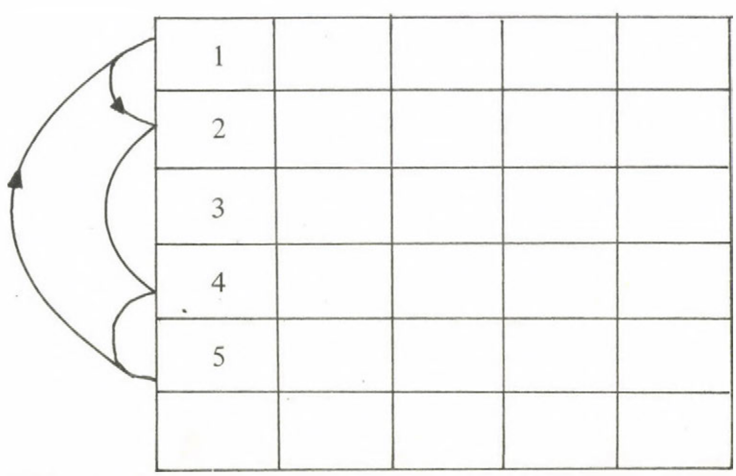


periodical jobs package
1, 2, . . . numbers of nonperiodical jobs

Fig. 3. O.S. with periodical cycle

c) The sequential O.S.

In such a structure (which is often used in chemical industry) job table is created which contains full description of all jobs, (Fig. 4.).



1, 2, . . . , job numbers

Fig. 4. Searching the job table in a sequential O.S.

We shall mention here only the most important components of a job description such as an enable / disable flag (which informs if the job is to be considered et all), time of the nearest desired activisation, period of activisation, starting adress location, continuation adress (for a preempted job).
This system also works with a predetermined cycle. Every given time Quantum it starts searching the jobs' table from the very beginning with respect to enabled jobs (so the position in the table enforces priority within a cycle).
If any of the jobs has a desired activisation time smaller then system clock state (it means that a demand is in Queue) it is executed, and it is activisation period is added to the desired activisation time.
After the time of the cycle elapses a jump to the top of the table takes place regardless of the actually being executed job's number and state. This structure has so far no elaborated queuing theory description.

d) O.S. with time slicing
The most commonly used time – sliced O.S. for real time applications are time – sharing models having a very large bibliography. For the industrial control applications a modification of them is sometimes used, namely a priority system in which service is given in quanta, and after every quantum completion a job having the highest priority is chosen for further service (sometimes it can be continuation of the previous job's service). Of course the choise of a proper Quantum is essential for this case. This case is under author's investigation.

3) Mixed structure
There exists also a class of systems which serve the transmission demands using interrupt system directly after their generation while analysis demands are considered in time epochs chosen by the scheduling algorithm, regardless of the moment of demands generation.
Scheduling algorithms for analysis demands are the same as in the case of time oriented O.S.

V. Final Remarks

Not attempting to give any exact comparison of the discussed structures one can present some outlines of their main features.
Usually the shortest response time can be obtained (for high priority demands) in the event oriented O.S.
However in this case the variance of response time can be expected to be bigger than in some time – oriented O.S. which introduce a deterministic factor in to their operation as well.
In the event oriented O.S. the ratio of overhead time changes with the systems'load and increases distinctively in heavy traffic conditions while for example the time–oriented synchronous system spends on the overhead a certain constant amount of time.

As far as the requirements (b) and (c) of the paragraph II are concerned it is often pointed out by various authors [12], [13] that debugging of an event – oriented system as well as it is later development is very time – consuming and enlarging it's reliability is quite difficult.

All it leads to a conclusion that the ratio of time oriented O.S. or mixed type O.S. will probably increase in the nearest future as far as the process control is concerned.

One can also state that mathematical description for plenty service disciplines is either not as detailed as required or does not exist at all so the choise of a structure is often done by the system designer basing only on his experience. Therefore some more research in this area seems to be necessary.

It should be mentioned that all results presented in this paper were obtained in cooperation with Mgr inz.Tadeusz Czachorski to whom I am also very gratefull for his valuable remarks concerning it's shape.

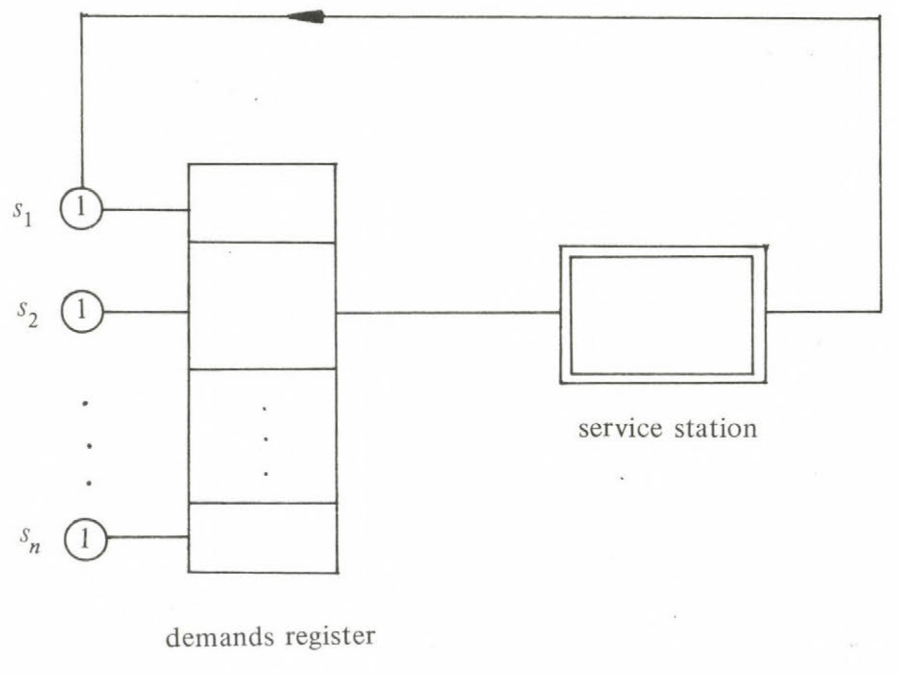
Appendix

We shall discuss here as example a system of one dimensional sources served under a preemptive – resume discipline (Fig. A. 1.). The investigation will be done using Jaiswall's method. For each source we shall assume the time a demand spends within the sonrce to have negative exponential distribution.

(1) $A_i(x) = P_r\{\tau \leqslant x\} = 1 - e^{-\lambda_i x}$ $\qquad i = 1, 2, \ldots, n$

and the service time duration probability density function

(2) $S_i(x) = P_r\{x < t < x + dx\}$ $\qquad i = 1, 2, \ldots, n$



A set of one – dimension sources

Fig. A. 1.

We shall assume that a demand from a source s_i has priority over demand from a source s_j if and only if $i<j \quad i,j = 1, 2, \ldots, n$.
The considered sheme describes also a multiprogramming system, with a constant number of nonhomogeneous jobs having independent I/O facilities, discussed by Tomko [8] in the case of preemptive resume service discipline.

Let us first investigate the operation of a service station with a single source Fig. A. 2.

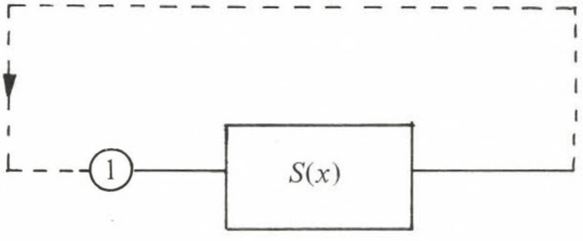


Fig. A. 2.

1) *A Simple scheme*

Let $p(x, t)$ be the density of probability that at time t the busy period (which started at time $t = 0$ due to demands generation) is in progress and the elapsed service of the unit is between x and $x + dx$.

$b(t)$ – be the probability density that the busy period (that means a time interval during which the server remains serviving demands without any pouse) which started at $t = 0$ terminates between t and $t + dt$.

One can easy verify, that $p(x, t)$ satisfies the equation

(3) $$p(x + \Delta, t + \Delta) - p(x, t)[1 - \eta(x)\Delta] = 0$$

where

$$\eta(x) = \frac{S(x)}{1 - \int_0^x S(u)du}$$

tending with $\Delta \to 0$ we obtain an difference – differential equation

(4) $$\frac{\partial \rho(x, t)}{\partial x} + \frac{\partial \rho(x, t)}{\partial t} = - \rho(x, t) \cdot \eta(x)$$

subject to initial condition

$$\rho(x, 0) = \delta(x)$$

$$\rho(0, t) = 0 \qquad \text{for} \qquad t \neq 0$$

(4) yields

(5) $$\overline{p}(x,s) = e^{-sx}\, e^{-\int_0^x \eta(x)dx} \; *$$

(6) $$b(t) = S(t)$$

2) *A scheme with initial period process*

Let us assume, that at $t = 0$ an initial period having density function of its duration $\Omega(t)$ starts. During this period a demand can be generated, but it is service may start only after completion of the initial period, and in that case the busy period b^{Ω} is a convolution of a busy period $b(t)$ (as in the previous case) and initial period.

If no demand is generated during the initial period $b^{\Omega}(t)$ is equal to $\Omega(t)$.
It can be easily proved that

(7) $$\overline{b}^{\Omega}(s) = \overline{\Omega}(s) \cdot \overline{S}(s) + \overline{\Omega}(\lambda + s)[1 - \overline{S}(s)]$$

(8) $$\overline{p}^{\Omega}(x,s) = [\overline{\Omega}(s) - \overline{\Omega}(s+\lambda)]e^{-sx - \int_0^x \eta(u)du}$$

where b^{Ω}, p^{Ω} stand in this scheme for similar values as in the previous paragraph.

Similar to infinite dimensional source models we shall introduce the notion of completion time, defined as the duration of period that begins from the instant the service of a demand starts and ends at the instant the server becomes free to take the next unit generated from the same source.
For preemptive– resume disciplines the completion time is equal to the period between starting and finishing the service of any demand (inluding time of premptions).
The probability density function of completion time duration for demands from s_j (called j – type demands) can be found by following inference:
The probability of n preemptions during the service of i–type demands p_n is

(9) $$p_n = \frac{(\Lambda_{j-1} \cdot y)}{n!} e^{-\Lambda_{j-1} \cdot y}$$

* We shall denote the Laplace – transform of any function $\varphi(t)$ by $\overline{\varphi}(s)$

$$\overline{\varphi}(s) = \int_0^{\infty} e^{-st}\varphi(t)dt$$

where

$$\Lambda_{j-1} = \sum_{i=1}^{j-1} \lambda_i$$

y – the lenght of i – type service.

Any preemption has the same duration, with probability density function $\gamma_{i-1}(t)$ which represents the lenght of busy period for a system with $(i-1)$ sources. (As existance of lower priority sources does not influance the service of high – priority ones)

If so, than

(10) $$C_j(x) = \int_0^x \sum_{n=0}^{\infty} e^{-\Lambda_{j-1}y} \frac{(\Lambda_{j-1}y)^n}{n!} \gamma_{j-1}^{*n}(x-y) \cdot S_j(y)dy$$

where γ_{j-1}^{*n} denotes a convolution of n times $\gamma_{j-1}(t)$

(10) yields

(11) $$\bar{C}_j(s) = \bar{S}_j\{s+\Lambda_{j-1}[1-\bar{\gamma}_{j-1}(s)]\}$$

the expected value and second moment are

$$E(c_j) = E(S_j)[1 + \Lambda_{j-1}E(\gamma_{j-1})]$$

$$E[(c_j)^2] = E(S_j^2)[1 + \Lambda_{j-1}E(\gamma_{j-1})]^2 + \Lambda_{j-1}E(S_j)\acute{E}(\gamma_{j-1}^2)$$

$\gamma_j(t)$ can be obtained as

(12) $$\gamma_j(t) = \frac{\lambda_j}{\Lambda_j} B_j(t) + \frac{\Lambda_{j-1}}{\Lambda_j} B_j^{\gamma_{j-1}}(t)$$

(where $\bar{B}_j$ and $B_j^{\gamma_{j-1}}$ are given by (6), (8) with substitutions

$S(t) \longrightarrow c_j^{(t)}$

$\Omega \longrightarrow \gamma_{j-1}$

$\lambda \longrightarrow \lambda_j$

after the following inference:

The busy period γ_j may be started either by j-th type demand (with probability $\frac{\lambda_j}{\Lambda_j}$) or by any demand of the higher priority type (with probability

The substitution of completion time instead of service time originates from preemption which can occure during service. (12) yields after transformation and proper substitutions

(13) $$E(\gamma_j) = \frac{\lambda_j}{\Lambda_j} E(c_j) + \frac{\Lambda_{j-1}}{\Lambda_j} \{E(\gamma_{j-1}) + E(c_j)[1 - \bar{\bar{\gamma}}_{j-1}(\lambda_j)]\}$$

Busy periods starting epochs can be treated as regeneration points of a renewal process. The time duration between two regeneration points $f(t)$ is given by a convolution

(14) $$f(s) = \gamma_j(t) * \Lambda_j e^{-\Lambda_j t}$$

Assuming that at $t = 0$ a busy period starts, transform of the renewal density $h_j(s)$ can be expressed as

(15) $$\bar{h}(s) = \frac{\bar{f}(s)}{1 - \bar{f}(s)} = \frac{\Lambda_j \cdot \bar{\gamma}_j(s)}{s + \Lambda_j[1 - \bar{\gamma}_j(s)]}$$

Let $e(t)$ be the probability of the system being idle at time t of the general process, consisting of busy and idle periods.

(16) $$e_j(t) = \gamma_j(t) e^{-\Lambda_j t} + h(t) * \gamma_j(t) * e^{-\Lambda_j t}$$

$[1 - e(t)]$ is exactly the utility factor in any given time t For $t \to \infty$ the stationary state probability of the server being idle is

(17) $$e_j = \lim_{t \to \infty} e_j(t) = \frac{1}{1 + \Lambda_j E(\gamma_j)} \qquad (\text{if } E(\gamma_j) < \infty)$$

We shall now calculate the distribution of time duration between j–th type demands generation and the epoch when it's service starts (waiting time) denoted as $v(t)$, in the stationary state.

The probability of type j demand taking part in the busy period γ_j is

(18) $$P_{j,\gamma_j} = \frac{\lambda_j}{\Lambda_j} + \frac{\Lambda_{j-1}}{\Lambda_j}[1 - \int_0^\infty e^{-\lambda_j x} \gamma_{j-1}(x)dx] = \frac{\lambda_j}{\Lambda_j} + \frac{\Lambda_{j-1}}{\Lambda_j} \cdot [1 - \bar{\bar{\gamma}}_{j-1}(\lambda_j)]$$

as j–type demand can be served at most once during γ_j.

The conditional probability of busy period beginning with γ_{j-1} under the condition that type j demand takes part in the busy period is

(19) $$P_{c,j} = \frac{\frac{\Lambda_{j-1}}{\Lambda_j}}{\frac{\lambda_j}{\Lambda_j} + \frac{\Lambda_{j-1}}{\Lambda_j}[1 - \bar{\gamma}_{j-1}(\lambda_j)]} = \frac{\Lambda_{j-1}}{\lambda_{j-1} + \Lambda_{j-1}[1 - \gamma_{j-1}(\lambda_j)]}$$

and the conditional probability of busy period starting with j–type demands service under the condition that j–th type demands take part in the busy period

(20) $$P_{N,j} = \frac{\lambda_j}{\lambda_j + \Lambda_{j-1}[1 - \overline{\gamma}_{j-1}(\lambda_j)]}$$

Waiting time is equal to 0 if a busy period starts j–type demand service or equal to τ with probability density function $u(\tau)$ if a busy period starts with γ_{j-1} (of course we tell only about such a busy periods in which j–type demand takes part).
Aiming to find an formula for $u(\tau)$ one has to realize, that busy period is duration x for $(j - 1)$ types demands has a density function $\gamma_{j-1}(x)$ while a time period y between its start and generation of j–type demand has a density function

$$\lambda_j e^{-\lambda_j \cdot y}$$

Yet a stochastic variable $\tau = x-y$ has the density function [14]

(21) $$u_j(\tau) = \int_{-\infty}^{\infty} \gamma_{j-1}(x-y)\, 1(x-y) \cdot \lambda_j e^{\lambda_j y}\, 1(-y) dy$$

where

$$1(x) = \begin{cases} 1 & \text{for } x \geqslant 0 \\ 0 & \text{for } x < 0 \end{cases}$$

(21) yields

$$\overline{u}_j(s) = \lambda_j \frac{\overline{\gamma}_{j-1}(s) - \gamma_{j-1}(\lambda_j)}{\lambda_j - s}$$

Finaly

(22) $$\overline{v}(s) = P_{c,j} \cdot \overline{u}_j(s) + P_{N,j} \cdot 1$$

which after proper substitutions yields

$$E(v) = \frac{\Lambda_{j-1}}{\lambda_j + \Lambda_{j-1}[1 - \overline{\gamma}_{j-1}(\lambda_j)]} \cdot \frac{1}{\lambda_j} [\lambda_j E(\gamma_{j-1}) + \overline{\gamma}_{j-1}(\lambda_j) - 1]$$

The probability density $R(\eta)$ of the response time being equal to η can be easily calculated as

$$R_j(\eta) = v'_j(\tau) * c'_j(\tau)$$

One can also find occupation time of the server which will be not further discussed here.

References

[1] I. Labetoulle "Computer scheduling problems" preprints of this winter school.

[2] De Cegama "A methodology for computer model building " Fall Joint computer Conference, (1972), pp. 299–301.

[3] A. Wolisz, T. Czachórski, "On the reduction of interrupts in the synthesis of real–time operating systems" Podstawy sterowania vol. 4 no. 3 pp. 289–304 (in Polish).

[4] N. K. Jaiswal "Prioririty Oueues" Academic Press, New York, (1968).

[5] Б.В. Гнеденко и другие ''Приоритетные системы обслуживания'' Издателбство Московского университета, Москва, 1973.

[6] Linas Schrage, "A mixed priority queue with applications tothe analysis of real–time systems" Oper. Res. vol. 17 no. 4 pp. 728-742.

[7] А.И. Никитин, СА Шестяков ''Влияние диспетьерского времени на харакмеристики операционной системы с приори тетной дисуиплиной бслуживания'' Управляющие системы и Машины, No. 1 Vol. 1 (1972).

[8] Tomkó József "Tömegkiszolgálás elméleti modellek a számitógépek rendszerek a matematika leirásába" preprints of this winter school.

[9] A. Wolisz, T. Czachórski, "Priority Service Systems for a one dimension sources set" Podstawy sterowania vol.3 no. 4 (1973), pp. 291–32) (in Polish).

[10] T. Czachórski, A. Wolisz "The discretionary discipline of one dimensional sources service" Podstawy sterowania, vol. 4 no. 2 (1974), pp. 115–130 (in Polish).

[11] J.Golubowicz "The system of time and memory allocation for demands service" Podstawy sterowania, vol. 4 no.4 (1974), (in Polish).

[12] P. Burnet, PA. Kidd, A. Lister, "Simulation of real–time programme faults" The Computer Journal, vol. 17, no. 1 (1973).

[13] Hagnes Cookbook "Please don't interrupt me while I am computing" Computer, (december 1973).

[14] A. Papoulis, "Probabilisty, Random Varriables and stochastic Processes" Mc Graw Hill, (1965).

Összefoglaló

Real – time operációs rendszerek valószinüségszámitási modelljei

A. Wolish

A dolgozat ismerteti az operációs rendszerekben lezajló folyamatokat és az ezekre alkalmazható sorbanállási modelleket. A real – time operációs rendszerekre egy osztályozást ad a kiszolgálás módjától függően. Befejezésül egy egyszerü preemptiv kiszolgálási stratégia analitikus vizsgálatát adja.

Р е з ю м е

Вероятностные модели "real-time" операционных систем

А. Волиш

В работе излагаются процессы происходящие в операционных системах и применяемые к ним модели теории очередей. Дается классификация "real-time" операционных систем по режиму обслуживания. Наконец аналитически исследуется стратегия с автоматным приоритетом.