https://doi.org/10.26552/com.C.2003.4.74-77

KOMUNIKÁCIE
COMMUNICATIONS

Michal Žarnay *

# DEADLOCK SOLVING IN TRANSPORT SYSTEM WITH METHODS FROM COMPUTER OPERATING SYSTEM

*In its first part, the article compares control principles in computer operating systems and in transport systems, and outlines similarities. One part of the control in both systems is handling of deadlock situations. When replacing human control by computer control in transport systems, handling of deadlock situations must be tackled as well. In its second part, the article outlines algorithms used in the operating systems field for that and discusses their potential application in the transport systems field. It comes to a conclusion that avoidance of deadlock by dynamic analysis of requested and assigned resources to processes looks to be the most perspective way generally, although in specific systems, rules for prevention or detection and recovery from deadlock can be also applied.*

## 1. Motivation

As informatics and automatic control are being implemented in the area of transport, the control part is included as well. There are attempts to replace human control by computer control.

In some parts of transport systems control, computers successfully replaced human operators. It relates especially to control of automatic operations that were defined as algorithms and that help now to produce control of higher quality.

However, in a complex transport system with complex control tasks that require complex decision-making, the human supervision is necessary. Even if computers do partial operations, some situations usually occur that cannot be solved by the computer support. This problem could be transformed into the problem of modelling of human decision-making with help of the computer.

When looking for algorithms to make models of human behaviour in transport system control, we may use several approaches. In this article, I would like to compare control in computer operating systems and in transport systems.

One part of control functions in both systems represents handling of deadlock situations. In operating systems, it is usually the machine that solves the situations automatically, without human interference. In complex transport systems it is usually a human operator who handles these situations. In the effort of replacing (at least in part) the human by a computer control, handling of deadlock situations must be created. That is why another goal of this article is to discuss application of algorithms for solving deadlock situations from the operating systems field in the transport systems field.

## 2. Control in Operating Systems

A goal of an operating system (OS) is to carry out a given set of jobs with use of a given set of resources in a given time period according to a prescribed algorithm. Both sets may change their contents in course of time.

When a job arrives to an OS, it is inserted into a queue of jobs waiting for processing. When all resources needed for a start of job processing are available, the job is removed from the waiting queue and it gets ready to be processed. The job processing can consist of one or more process threads. The process or its threads individually get assigned resources that they need. This can be at the beginning or in course of process – according to a prescribed algorithm. When the process or its thread is finished, the assigned resources that have not yet been released, are given back to the OS. After all the process threads related to a job are done, the job is removed from the system.

As an example, the resources in an OS can be a processor, internal and external memory (RAM, hard disk, floppy disk, CD-ROM), input and output devices for communication with the user (keyboard, mouse, display, printer) or with other computers (network equipment).

Among jobs, as an example, we can imagine calculation of a mathematical problem, printing of a text document, editing of a table in a table processor, scanning of a picture, copying of data from a hard disk to a floppy disk and receiving of data from the computer network.

All of these jobs process data according to a prescribed algorithm. The term data in the given examples, as it is evident, covers numerical data, a text document, tables, pictures, etc. In some processes, the data are transformed from input to output, in other cases, they are moved between two places, or both, moved and transformed.

## 3. Analogy in Control of Transportation System and Operating System

As a transportation system (TS) we can consider a network or a part of it on macroscopic or microscopic level, serving one or

* **Michal Žarnay**
Department of Transportation Networks, Faculty of Management Science and Informatics, University of Žilina, Moyzesova 20, 010 26 Žilina, Slovak Republic, Tel: +421-41-5134224, E-mail: Michal.Zarnay@fri.utc.sk

more transportation modes – for instance a passenger train station, a road crossing, air traffic network in a country, inland waterway network in a region or a terminal of combined transport.

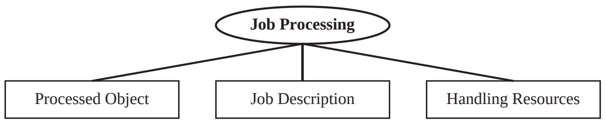

*Fig. 1. Process in Transport System*

If we take the passenger train station, "processing" of a job on a train in the station can represent a process (Fig. 1). The train enters the station, arrives to a platform, people get off and on the train, the train crew and train locomotives may exchange and the train leaves out. The handling resources in this case include infrastructure of the station, namely the tracks and the platform used, and station crew involved in the "processing" of the train. For other jobs connected with a change of train cars composition, required resources may include other types of tracks (e.g. a hump, sorting siding, depot) and workers as well as shunting locomotives.

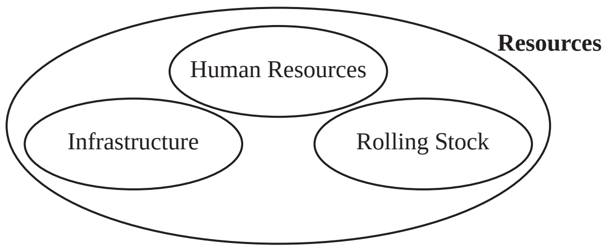

*Fig. 2. Resources in Transport System*

In other TS-s, it is similar. Jobs are represented by movement of vehicles, passengers or goods from one point to another in the system, with or without additional operations performed. This responds to the different kinds of processes including transformation or movement or both.

Resources can be in principle divided into technical equipment and human resources. Technical equipment can be further structured as infrastructure (routes and adjacent equipment) and rolling stock (moving vehicles). See Fig. 2.

The processing algorithms in TS are represented by job descriptions. They can be represented in different ways. One of them is a flow chart (Fig. 3).

## 4. Deadlock theory

Processes in OS as well as TS (the word "system" will be used further, since it may apply both OS and TS) can run in principle in two ways: successively or concurrently. When running concurrently, the system must take care of synchronization of their access to the resources.

If two or more processes use disjunctive sets of resources during the whole time period, when they run concurrently, or they use common resources with a non-exclusive assignment, there is no problem of synchronisation.

However, if they use at least one common resource that must be assigned exclusively, it may happen that one of the processes needs the common resource, while another process is using it. Thus the first process must wait for the second one to release the conflict resource. The need of the resource may arise by more processes and they get into a set of waiting processes for the conflict resource. After the resource is made available, it will be assigned to one of the waiting processes that can thus continue. After this process is finished, the conflict resource can satisfy next process from the waiting set, etc. until the set is empty.

If there are two processes $P_1$ and $P_2$ (Fig. 4) that use two or more common resources – let's assume 2 resources $R_A$ and $R_B$ for this case – and the resources are to be assigned in different order in the time period of concurrent running, the two processes can come to a situation called deadlock. It is a situation, when the $P_1$

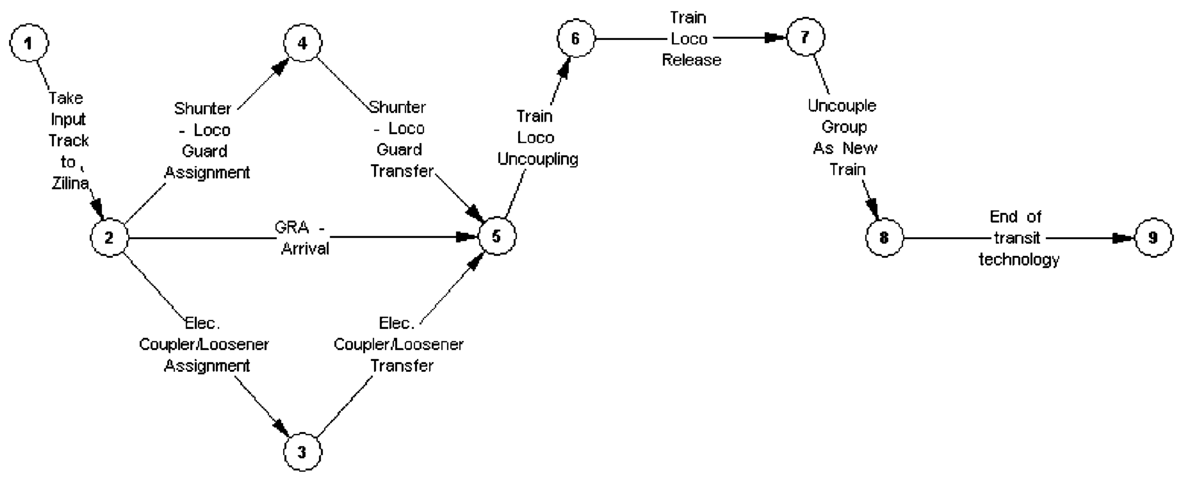
1
2
3
4
5
6
7
8
9
Take Input Track to Zilina
Shunter - Loco Guard Assignment
Shunter - Loco Guard Transfer
GRA - Arrival
Elec. Coupler/Loosener Assignment
Elec. Coupler/Loosener Transfer
Train Loco Uncoupling
Train Loco Release
Uncouple Group As New Train
End of transit technology

*Fig. 3. Flow chart of technological job description*

process has got assigned the $R_A$ resource and needs the $R_B$ resource, and the $P_2$ process has got assigned the $R_B$ resource and waits for the $R_A$ resource.

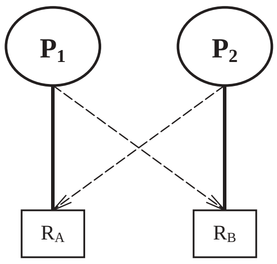

*Fig. 4. Deadlock*

There are four necessary Coffman conditions for a deadlock to occur:

- Mutual exclusion – there is more than one resource in the system that is either currently assigned to exactly one process or is available,
- Hold and wait – a process currently holding resources granted earlier can request other resources,
- No pre-emption – resources previously granted cannot be taken away from a process, but they must be explicitly released by the process holding them,
- Circular wait – there must be a circular chain of two or more processes, each of which is waiting for a resource held by next member of the chain.

All four of these conditions must be fulfilled for a deadlock to occur. If one of them is not fulfilled, no deadlock is possible.

In general, the following strategies are used for dealing with deadlocks:

- Detection and recovery – let deadlocks occur, detect them, and take a remedy action.
- Prevention – by structurally preventing one of the four conditions necessary to cause a deadlock.
- Dynamic deadlock avoidance – by a careful resource allocation.

## 5. Deadlock detection and recovery

The first approach requires the system to have:

- Algorithm for detection – to examine system state and to determine, whether the deadlock has occurred,
- Algorithm for recovery of the system from the deadlock.

The recovery from deadlock is possible in principle in two ways:

a) To notify a system operator that may handle the problem.
b) To recover automatically either by killing one or more processes in a deadlock chain or by removing one or more resources involved in deadlock from their current owner. Both methods aim to break the circular wait condition.

In computer OS, it is evident that both automatic ways (part b) may cause damage of some data that must be repaired and corresponding processes in most cases must be rerun. In TS, the recovery can have also large harmful consequences on running processes, plus the recovery itself is in most situations a very complex task to be handled automatically. Thus we can expect the recovery interventions in TS to be done by a system operator in most cases. This depends on specific TS and specific situation that occurred.

## 6. Deadlock prevention

As prevention of a deadlock, it must be assured that at least one of the four Coffman conditions is broken. Let's see, if this is possible in TS.

To prevent the Mutual exclusion condition, there must be maximum one resource that cannot be shared by more processes at the same time. All other resources must have the attribute to be shared by more processes. In TS this would mean, for instance, that the same section of route could be occupied by more than one vehicle at the same time. In general, we can say that the prevention cannot be achieved by breaking this condition.

To prevent the Hold and wait condition means to force a process to have no resources at the moment, when it requests resources. This can be achieved in two ways:

- The process uses some resources and before asking for new, it must release all, and ask for them again,
- The process asks for all resources needed at the beginning and holds each of them until there is no need for them anymore.

The first way is practically impossible in TS. Once a vehicle occupies a section of route (e.g. road, track, water or air route), it cannot disappear before asking for additional resources.

The second way leads to loss of effectiveness. The more complex is the TS, the larger may be this loss. For instance, a train coming to a station would get assigned the whole route until the departure point from the station, even if it stands in the station for a long time, while other trains could come and leave the station – some of the tracks would be assigned with no utilization for too long time.

To break the No pre-emption condition means to allow taking away previously granted resources, regardless the situation, in which the process currently is. Two principal kinds of situation could be distinguished here: whether the resource that is to be revoked is currently in use or it is only reserved to the process. In case it is only reserved, it could be taken away. However, this is only a partial solution and we cannot ensure breaking of the condition for any situation when a resource is assigned to a process. In TS, vehicles usually occupy a section of route, and this, as a resource, cannot be practically taken away.

The way to prevent the Circular wait condition is to arrange all types of resources into an order and to demand processes to request resources according to ascending order of resource numbers. The order could respond to costs of assigning and release of resources from a process.

As an example, in a marshalling yard, the most expensive is most probably assigning and release of tracks (because this operation requires moving a train set that is processed between the tracks), after that, there are mobile resources (locomotives) and the least expensive are usually movements of members of crew. However, fulfilling this algorithm would imply taking the crew members away from the process in any case of a moving resource request or a track request, as well as taking the moving resource away from the process in the case of building a track route for the processed train set to move.

We can easily understand that the outlined way of breaking of the last condition results in reduced effectiveness of TS control.

## 7. Deadlock avoidance

Dynamic avoidance is an alternative to the structural prevention from deadlock. It is based on careful allocation of resources to processes after analysing information about requested and available resources. A system grants requested resources, only if it will result in a safe state of the system, i.e. the new state must not lead, under any circumstances, into circular waiting – the last of the Coffman conditions. If this is not guaranteed, the requesting process must wait until any of the assigned resources get released.

This analysis is to be carried out by each resource request. Available algorithms differ in what information they analyse. The simplest ones require from each process to state the maximum number of resources of each type needed for its execution. One example of these algorithms is the Banker's algorithm.

This method of dealing with deadlock situations is applicable in TS. The application depends on a job description and the way how the job description is represented and used in the control system. Anyhow, the control system will be obliged to extract the information about requested resources during the whole job processing.

If all the resources requested during the whole process will be taken into consideration in the beginning, it may happen that absence of a resource needed in the processing later than in its beginning will cause the waiting of the process before its start until the resource is available. That is why it looks to be a good idea not to consider resource requests for the whole process in the beginning but to divide the process into parts that are independent, and to run the analysis on each part separately before its turn comes. For instance, technological operations carried out on an airplane after its arrival and the ones on the same airplane before its departure from an airport could make up 2 parts of the airplane processing during its visit to the airport.

It could be even possible to separate two branches of the same process running in parallel and having no common resources. It could be done at the same time but not necessary. For instance, technical and traffic inspection of a cargo airplane.

Another way to make the control more effective would be finding a time estimate when a resource is supposed to be used in a given process. Duration of individual activities in a process can be estimated to a certain extent following experience and information about the processed object. This estimate can serve for creation of a time plan of resources use with possible utilization by optimisation of the resource assignment to processes as well. The time plan can be updated on-line as the information about running processes and expected jobs is received.

One of disadvantages of the Banker's algorithm may be its expensive application every time, when a request for a resource occurs. This strongly depends on a number of processes, resources and resource requests in a given time in the specific system. Higher complexity of system will result in a higher number of processes, in a higher number of resources and in more frequent resource requests, thus the algorithm will be more time demanding. TS have an advantage against computer OS in that the time needed for the algorithm to bring results should be relatively small compared to a duration of processes.

## 8. Conclusion

In this article I tried to outline analogy in control between an operating system in computers and transport systems. I focused on the deadlock theory and handling of deadlocks in the operating systems. We have found that handling methods used there can be also applied in automatic control in transport systems. From the analysis we can see that

- Detection and recovery from deadlock may be applicable, but recovery cost and complexity strongly depends on a concrete TS and a concrete deadlock situation.
- Prevention from deadlock by structural prevention of one or more conditions of deadlock is not applicable in most cases. Some of the conditions can be broken only for an unacceptable cost and loss in effectiveness of a system.
- Dynamic avoidance algorithms like the Banker's algorithm can be applied. They cost some time of a control subsystem but their effect should satisfy the need. The cost depends on dispositions of the concrete TS.

When comparing the conclusions we may understand that the most perspective way is probably the dynamic avoidance of deadlocks. However, the conclusions discussed in this article are made rather on a general level. They may vary from one transport system to another. It is also possible to apply rules that combine all three approaches to handling deadlocks - specifically for each system.

## References

[1] CENEK, P.: *Operating Systems.* VŠDS, Žilina, 1989 (In Slovak)
[2] MARTINCOVÁ, P.: *Operating Systems.* ŽU, Žilina, 1997 (In Slovak)
[3] ŽARNAY, M.: *Analysis of Control Methods in Transport Systems.* Paper for academic dissertation examination, Žilina, 2001 (In Slovak)