1986

A simulation tool to study methods of addressing of resources of a network operating system

J. Indulska
University of Wollongong

A. Goscinski
University of Wollongong

Recommended Citation

Indulska, J. and Goscinski, A., A simulation tool to study methods of addressing of resources of a network operating system, Department of Computing Science, University of Wollongong, Working Paper 86-11, 1986, 23p.
https://ro.uow.edu.au/compsciwp/33

A SIMULATION TOOL TO STUDY METHODS OF ADDRESSING OF RESOURCES OF A NETWORK OPERATING SYSTEM

by

J. Indulska
A. Goscinski

Preprint No. 86-11 October 1986

A Simulation Tool to Study Methods of Addressing of Resources of a Network Operating System

J. Indulska*, A. Goscinski

ABSTRACT

In the paper a simulation tool to study an effectiveness of different methods of addressing of resources in network operating system and simulation experiments have been presented. This model makes it possible to study network operating systems with a static addressing of resources and network operating systems with admissible dynamic change of addresses of resources in a local area network. To carry on experiments the model of the network operating system has been extended adding a model of environment demands. The later has been constructed in such a way that it is not necessary to consider an influence of different methods of resource addressing on a distribution of arrival moments of those environment demands. The simulation tool presented here has been developed on the basis of a logical model of the network operating system constructed earlier [Gos 86]. The tool described in this paper could be treated as a basic one because it is possible to use many parts of it to construct other tools making possible simulation studies of other problems of network operating system (e.g., the effectiveness of distribution of resource management, synchronization). The simulation carried out shows that the simulation tool constructed can be effectively used to study the influence of different methods of resource addressing on a performance of network operating systems.

*/ J. Indulska is with the Institute of Computer Science, St. Staszic University of Mining and Metallurgy, Al. Mickiewicza 30, 30-059 Krakow, Poland.

1. INTRODUCTION

One of the problems of the construction of network operating systems for local computer networks is a choice of an effective method of addresssing* of network resources. This problem is caused by a geographical distribution of resources which implies that demands for access to a resource could be in network nodes different from nodes to which that resource belongs. That implies a need for addressing of those demands, i.e., identification of a location of a given resource in the network. The problem of resource addressing exists in network operating systems which are extensions of centralized operating systems as well as in newly constructed network operating systems. Both types of operating systems would be called in this paper network operating systems.

The analysis of methods of resource addressing in network operating system shows their big diversity. Using the method of resource addressing as a base for classification, the solutions of the network operating systems presented in literature could be divided into the following classes:

- operating systems with the known location (address) of resources in each node of the network [Bro 82, Cab 79, Col 82, Lin 82];
- operating systems with a hierarchical access to resources - there are in the network additional data structures defining the location of network resources [Deg80, Fle 80, Wat 80];
- operating systems with distributed access to resources - in several nodes only local resources are known [Des 79, Des 80, Fri 83, Ous 80];
- distributed systems with an undefined method of resource addressing - these operating systems have been presented at much higher level of abstraction and problems of resource addressing and access to resources have been delayed up to implementation stages [Ban 82, Gui 82, Ras 81, Zim 81].

The assessment of the effectiveness of different methods of resource addressing is a complex problem. The addressing method should fulfil typical requirements of effective operating systems, e.g., a minimal time of response to events, a minimal time of demand service. Moreover, a network operating system should be reliable and reconfigureable. The addressing method chosen should be effective in the case of dynamic changes of the resource location. In practice, such a multi-criterion analysis of addressing methods is not possible using analytical methods only.

In this paper a simulation tool to study the effectiveness of methods of resource addressing is presented. The tool is constructed for real - time network operating systems and is oriented towards some methods of addressing chosen on the basis of literature as well as on our study [Gos 86]. The tool has been construced in such a way that other methods (stochastic) could be studied.

The simulation tool presented here has been constructed on the basis of the logical model of the network operating system [Gos 86, Ind 85]. That logical model has been developed to be a concept and a tool to construct simulation tools. It has been developed using ideas of the object model [Jon 78].

The paper contains five major parts. The basis of the simulation tool is presented in Section 2. Two models of communication in a network have been introduced. Section 3 contains the synthesis of the simulation tool. The following elements were considered: events controlling a network operating system, processes managing resources and functional connection between those processes. A model of demands of an environment of the network operating

*/ The addressing of a resource here means the method of access to a network resource [Wat 81]

system is introduced in Section 4. In Section 5 implementation of the simulation tool is presented. The following elements were discussed: the characteristic of attributes of objects and actions, objects of the environment, events of the simulation tool, parameters of processes managing resources and implementation factors. Tool testing and simulation experiments were presented and discussed in Section 6. The simulation carried out shows that the simulation tool can be effectively used to study the influence of different methods of resource addressing on a performance of network operating systems.

2. THE BASIS OF THE SIMULATION MODEL

The logical model used to construct the simulation tool presents the network operating system as a set of processes managing resources and definitions of connections between these processes. In the model the new types of resources of the network operating system (messages and data structures describing the location of resources) have been defined and included into a general structure of connections of processes managing resources. The model contains a mapping of demands on an environment of the network operating system into processes managing resources.

In the simulation tool the structuring of the logical model has been kept. That structuring makes possible modification of the tool in such a way that other studies of network operating system could be done (e. g., the effectiveness of distribution of resource management, synchronization).

Two models of the multi - layer architecture of communication in a network [ISO 81] have been proposed:

- the communication is modelled by a two - layer protocol; the higher layer maps an application layer of the OSI / ISO Reference Model and the lower one models six other layers of the Model (Fig. 1a);
- the communication is modelled by a three - layer protocol (Fig. 1b). The three - layer model is adequate to the case when interfaces between node computers and communication medium are intelligent enough and they can perform a part of functions connected with management of resource access (e. g., addressing of demands of access to a resource); in two - layer model these functions are performed by node (host) computers.

It should be underlined that the structuring of the logical model [Gos 86] makes possible an easy modifiability of the simulation tool such that the communication could be modelled by any number of layers, in particular by a seven layer system strictly adequate to the OSI / ISO Reference Model.

3. THE SYNTHESIS OF THE SIMULATION TOOL

The synthesis of the simulation tool to study the effectiveness of different methods of resource addressing requires definitions of:

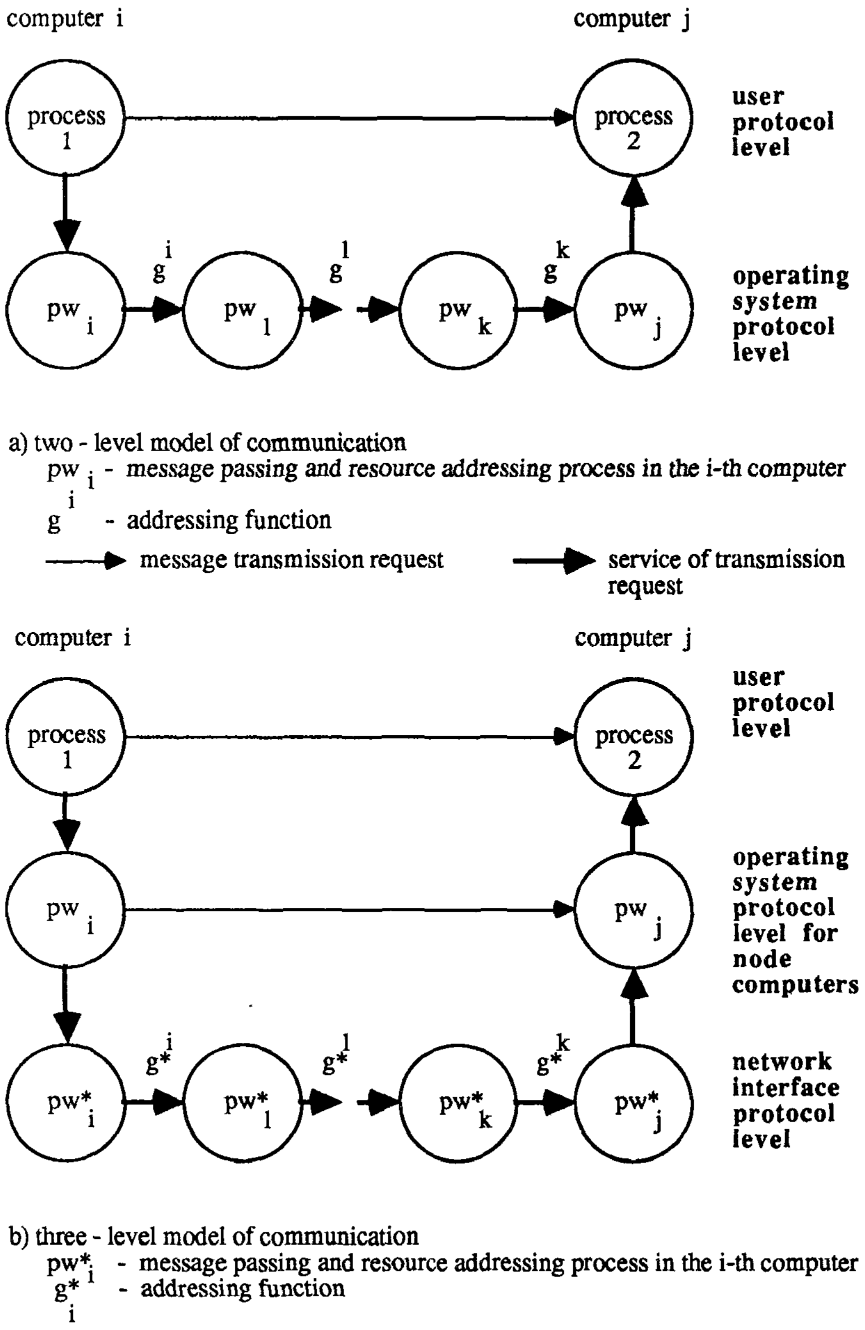
- a suitable set of processes managing resources, and
- a set of events controlling a network operating system.

The choice has been done on the basis of:

- analysis of structures of real - time centralized operating systems and their processes managing resources, and
- analysis of connections between those processes and processes managing the new defined types of resources of the network operating system [Gos 86].

3. 1. Events Controlling a Network Operating System

A network operating system is controlled by set Z of events of a local computer network. A given event $z \in Z$ is connected with a node of a network. The event is served according to



a) two - level model of communication

pw_i - message passing and resource addressing process in the i-th computer

g^i - addressing function

——▶ message transmission request ——▶ service of transmission request

b) three - level model of communication

pw^*_i - message passing and resource addressing process in the i-th computer

$g^{*\,i}_{i}$ - addressing function

Fig. 1. Models of communication in network operating systems.

an assumed method of service of events which arrived at the same time and performed by an interrupt system of a node computer. Set Z^i of events of the i-th node, $i = 1, N$, depends on an architecture of a node computer. The elements of set Z^i belong to classes Z_i, i = 1, r, distinguished in set Z by types of interrupts.

When chosing classes of events some simplifications implied by the study goal have been assumed: in the set Z of network events there are not programming errors and hardware errors of a stochasic nature.
On the other hand, all other internal and external events have been divided (on the basis of their sources and their stochastic characteristics) into the following classes:

Z_1 - random external interrupts (from an environment of the network operating system) activating user (application) processes,

Z_2 - interrupts generated by network interface,

Z_3 - clock interrupts activating cyclic processes,

Z_4 - I / O interrupts,

Z_5 - interrupts generated by macros (from processes of the network operating system and processes of an environment of the network operating system),

Z_6 - demands for a process reallocation (introduced in the simulation tool to study the effectiveness of addressing methods in the case of dynamic changes of the location of resources).

Set Z^i of events of the i-th node is a subset of the set of events defined above

$$Z^i \subset Z, \quad i = 1, N, \quad Z = \{Z_1, Z_2, Z_3, Z_4, Z_5, Z_6\}$$

In the set Z of events there are classes of events charcterized by random characteristics which do not depend on the network operating system (Z_1, Z_3, Z_6}. Because of any possible frequency of arrival of instances of these events it has been assumed that the constructed tool is adequate from the point of view of the service of these events to the service of a queue with loses . Such an approach could be explained by the fact that in the network node only one process activated by a given event could be run and a number of stored demands of activation of that process is restricted. On the other hand the service of the other types of events is adequate to the service of a queue without losses. The events leave a system after completion of service only. It is in force for communication events also - messages always arrive at their destinations.

In the simulation tool a priority uninterrupted service of interrupts has been used. On the other hand a possibility of scheduling of user processes and some subset of processes of the network operating system using a preemptive algorithm has been assumed. The method of interrupt service defined in this way makes it possible modelling of other interrupt systems used in real - time systems.

3. 2. Processes Managing Resources

In set R of processes managing resources of the network operating system two classes of processes has been distinguished: processes managing physical resources and processes managing logical resources. In the tool the processes managing memory have not been taken into consideration because of the feature of real - time operating systems, i. e., usualy processes are not removed from memory.

In set R (using the type of a managed resource as the basis of clasification) the following classes of processes managing resources have been distinguished:

R_1 - the class of processes managing message passing (exchange) in node computers,

R_2 - the class of scheduling processes,

R_3 - the class of processor dispatchers,

R_4 - the class of processes managing message passing in network interfaces (this class is used in the case of the three - layer model of the OSI / ISO architecture),

R_5 - the class of processes managing data structures describing the location of resources in the network,

R_6 - the class of processes managing parallel processors (channels) of a node computer,

R_7 - the class of processes managing clocks of a node computer,

R_8 - the class of processes connecting external events with processes of the network operating system and processes of an environment of the network operating system (a change of a state of processes activated by external events).

Processes belonging to classes R_1 - R_8 are connected with a service of interrupts and they are noninterruptable according to the type of the interrupt system. In each node there are processes belonging to classes R_1 - R_5; the presence of processes belonging to other classes depends on resources of a node. The study goal implies that definitions of processes of class R_5 could be different. Each variant is adequate to one method of resource addressing. The addressing methods have been presented in [Gos 86].

Moreover in the model of an operating system of a network node there could exist processes of classes R_9 - R_p, $p > 9$, treated here as processes of an environment of the operating system (interruptable). Processes of these classes manage logical resources such as file (an access to a file), local procedure (a call of a local procedure), etc. Processes of class R_9 are used to manage demands of reallocation of processes and implement different protocols of process reallocation from node to node. The diversity of these protocols depends on methods of resource addressing. These protocols contain two important parts: (i) reallocation processes, (ii) operations on data structures describing the resource location, and (iii) synchronization of these operations [Ind 85]. Such a definition of the processes of class R_9 makes it possible a study of an effectiveness of addressing methods in the case of dynamic changes of the resource location in the network.

Processes of classes R_{10} - R_p do not have visibly defined operations because of the level of abstraction of the description of the environment of the network operating system. The demands of an environment are oriented towards operations excluding each other on a resource pointed to by an environment.

3. 3. Functional Connections of Processes Managing Resources

Managing processes of classes R_1 - R_8 perform the service of interrupts in a network node. Functional connections of managing processes belonging to classes R_2, R_3, R_6, R_7, R_8 are an image of connections used in centralized operating systems. The service of interrupts in the tool has been extended by adding processes of classes R_1, R_4, R_5 managing message passing in the network. The connections of processes R_1 - R_8 in one network node is illustrated in Fig. 2.

Because the goal of the construction of the simulation tool is a study of the effectiveness of different methods of resource addressing it has been assumed that the processes belonging to classes R_1 - R_9 are connected with a given node of the network. When constructing simulation tools to carry out a study for different goals, e. g., the effectiveness of different methods of synchronization, a distribution of service of a resource, it is necessary to consider the necessity of:

- inclusion of additional processes managing resources,
- definition of additional connections of processes of class R_6 and R_9 - R_p (for the whole network) or
- exclusion of some classes of managing processes.

The definition of operations for the several processes managing resources is the next step when modifying this simulation tool.

4. A MODEL OF DEMANDS OF AN ENVIRONMENT OF THE NETWORK OPERATING SYSTEM

The environment of a network operating system is: hardware, user processes, the environment of a computer system (operator, production process controlled). The set of events generated by the environment of the network operating system could be divided (the classification is based on random characteristics of demands of the environment) into three groups:

(i) the events with a probability distribution of moments of demand arrival independent the network operating system (hardware errors, events signalizing demands of the environment of the computer system),
(ii) the deterministic events (completion of set up operations),
(iii) the events with a probability distribution of moments of demand arrival dependent on some strategies of the network operating system, e. g., scheduling algorithm (macros interrupts).

The constructed model of the demands on the network operating system uses the division given above and it contains:

- the sources of demands of independent events (Z_1, Z_6),
- the sources of determined demands (Z_2, Z_3, Z_4),
- a static model of user processes included declarations of dependent events (Z_5).

The static model of processes is converted by the operating system into a source of events (by supplying a processor to a process which implies instances of events declared in a program code of this process). The model of the network operating system environment has

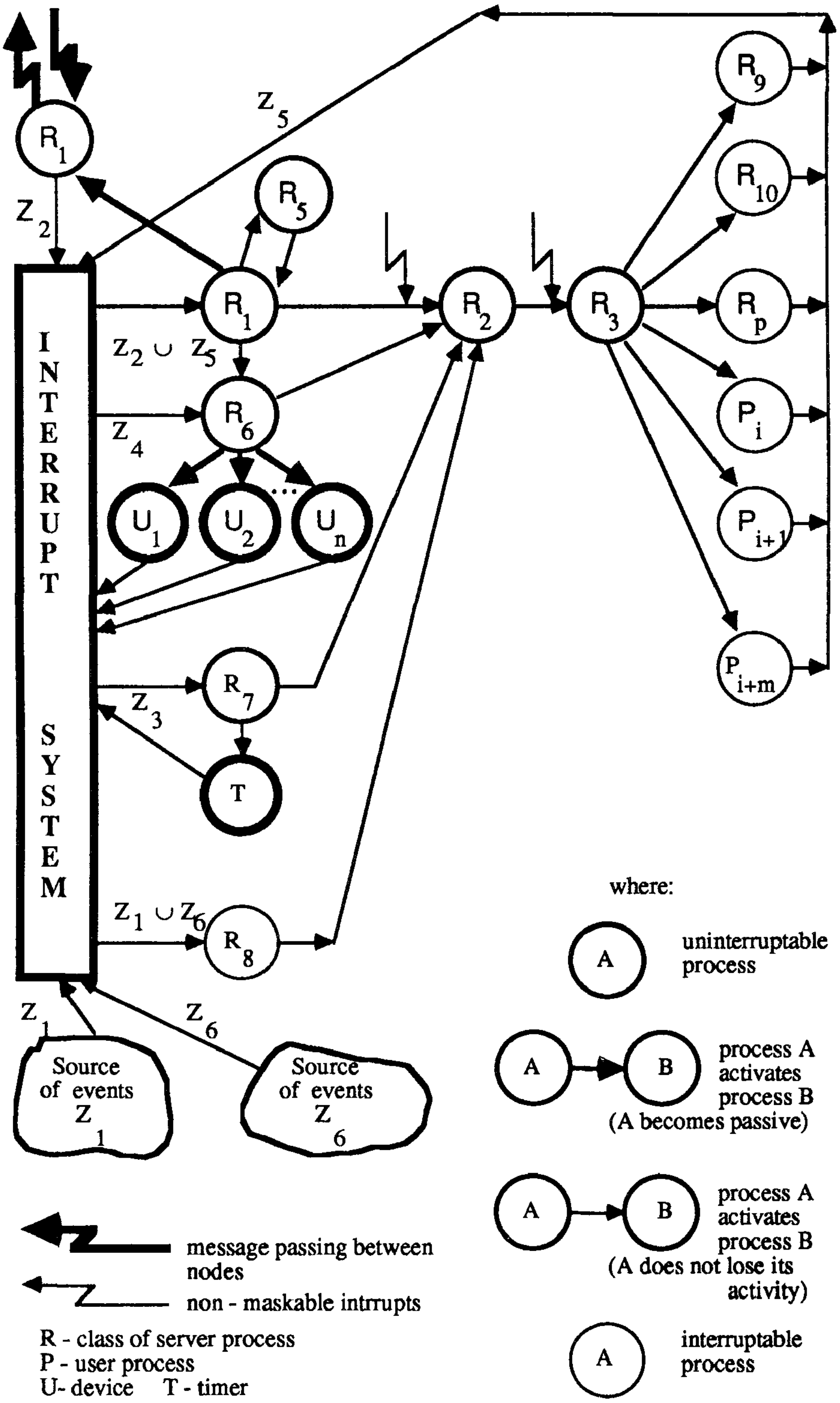


Fig. 2. Model of interconnections of the server processes in the i-th node.

been created on the basis of an analysis of characteristics of real - time user processes. The software environment of real - time network operating system contains cyclic processes activated with a given frequency and processes activated by random events. In general, a real-time software environment could be modelled (constructed) as communicating and synchronizating concurrent processes. So the set of the real - time user processes could be modelled as a set

$$P = P_T \cup P_L \cup P_N$$

where

$P = \{p_1, ..., p_m\}$, p_i - the process of the environment,
P_T - the subset of cyclic processes,
P_L - the subset of processes activated by random events,
P_N - the subset of concurrent processes activated by processes belonging to
$P_T \cup P_N$

A model of a process is a lineary ordered set of events [Paw 68]. From the point of view of an operating system, an event in a process is a demand for access to a resource of the network operating system. Between two consecutive demands (macro, a call of a subprogram of an operating system) declared in a code of a program there is a demand for access to a processor for a time Δt, $\Delta t \geq 0$.

The set of events modelling a process can contain the following types of events:

- demands of synchronization and interprocess communication,
- I / O demands (from I / O facilities, a controlled process),
- demands of procedural resources (e. g., dynamic call of a procedure).

Assumption that communication and synchronization are performed by message passing implies that there are two classes of events:

- a demand for message passing between processes of a software environment,
- a demand for performing an operation on a resource (e. g., I/O operation, a call of a process to perform a procedure) by an operating system .

Demands of the environment of the network operating system could be treated as demands of an interprocess communication between processes of the environment and processes managing resources. That possibility does not depend on an implementation of the network operating system. The demand of a communication could be synchronous or asynchronous from the point of view of a demanding process. So, declarations of events in the user processes could be modelled by atomic operations of message passing (with a nonblocked semantics [Lis 79]) which are equivalent to operations send message and receive message*.

send message (receiver, parameters)
receive mesage (sender, parameters)

where

sender, receiver - a process managing a resource or a process of an environment
parameters - a type and parameters of an operation performed on a resource or an identifier and parameters of a message.

*/ Complex protocols of interprocess communication (e. g., the communication on the basis of RPC, blocked message passing) could be presented in the terms of these primitive operations.

A model of infinitely existing user processes in the real - time operating system is an infinite sequence of message passing operations. For simplicity a repeating of processes behaviour has been assumed. That means that in the model there are repeated sequences of events in periods between events of activation and events of process suspension demands.

So, process $p_i \in P$, $i = 1, m$, is modelled by lineary ordered set of events:

- the sequence of events of message passing
  send(q, parameters)
  receive(q, parameters)

  where $q \in R \cup P$

- suspending event
  end(parameter)

The model of process p_i is extended to vector of required times of processor utilizations by this process p_i in the time horizon between two consequtive events of the process.

The allocation of processes to network nodes described as follows

$$P^i = \{p_j \in P: a(p_j) = i, j \in \{1, m\}\}, i = 1, N$$

makes possible a definition of set X^i of demands on the environment in the i-th node.

On the other hand

$$P^i \cup R^i$$

where $R^i = \{R_{hk}: a(R_{hk}) = i\}$, $i = 1, N$,
h - the class of a process managing a resource
k - the identifier of a process in a given class
defines set Z^i of possible events of the i-th node.

The presented model of the environment of the network operating system has been treated as a basic model for a simulation of sources of demands controlling a behaviour of this system. Distributions of moments of arrivals of these demands depend on demands of user processes as well as network operating system. The introduction of a stochastic model of user processes makes it possible to evoid a hard problem which is a definition of an influence of different operating systems in particular different methods of resource addressing on a distribution of moments of arrival of processes demands.

5. IMPLEMENTATION OF THE SIMULATION TOOL

The simulation tool for studying methods of resource addressing in network operating system has been constructed as an event model. Definitions of operations, except operations for new types of resources introduced in the logical model [Gos 86], have been developed on the basis of centralized operating systems. The tool described has been constructed at a level of abstraction which fulfiles requirements imposed on simulation models [Zei 76], i. e., this tool is a momomorphic mapping of a network operating system preserving properties of a system modelled.

To map the model of the network operating system and its environment into a simulation tool one must define:

- the set of objects* of the simulation tool to map static features of the system,
- the set of actions (activated by events) to represent dynamic features of the system.

Each object and each action as described in the tool by the set of attributes and their values could be set up (restrictions of the tool), or changed (decision variables or computed during a simulation).

5.1. The Characteristic of Attributes of Objects and Actions

The service of events

In this paper the service of events of the network operating system is based on a priority based interrupt system with uniterruptable service of an event.

Object(s): - lists of events
Object attributes:- N - the number of lists of events, (N - the number of network nodes)
Action: - priority based choice of an event from the i-th list of events, $1 \leq i \leq N$
Action attributes:- event pritities (decision variable),
- the vector of times of interrupts in nodes $i = 1, N$, (decision variable)

Processes managing resources

Managing processes have been mapped into objects (data structures describing a state of a resource and queues of demands), and actions on objects (operations on resources).

Processes of class R_1 - Message passing

Object(s): - the table describing resources and their addresses in the network,
- priority queues of messages in the nodes,
- the table of states of user processes
Object attributes:- number of resources,
- levels of message priorities
Action(s): - testing of the location of the process managing a resource: local or remote,
- ordering of addressing,
- message passing,
- the change of state of a local process waiting for a message
Action attributes:- addressing function (decision variable),
- the vector of times of service of a message by processes of class R_1 in all nodes, $i = 1, N$ (computed parameter),
- mean time of message passing between nodes (decision variable)

Processes of class R_2 - Process scheduling

Object(s): - priority lists of processes in the state ready for each network node
Object attributes:- the number of nodes - N,
- levels of priorities of processes (given parameter)

*/ Object, here, is understud in the sense of simulation theory. To avoid any confusion, the sense of the object model would be pointed out if necessary.

Action(s): - a choice of a process,
- a change of process states,
- inserting and removing processes from the lists of ready processes
Action attributes:- priority scheduling (given parameter),
- vectors of mean times of operation performing in all nodes, $i = 1, N$ (decision variable)

Processes of class R_3 - Processor dispatcher

Object(s): - the vector of processor states, $i = 1, N$
Object attributes:- the admissible states of a processor (given attributes: state - system, state - user process, state - wait)
Action(s): - a processor switching - a change of a processor state
Action attributes:- vectors of times of operations performed in all nodes (decision variable)

Processes of class R_4 - *Message passing at the level of a network interface*

Object(s): - the table describing resources and their addresses in the network
- priority queues of messages in network interfaces
Object attributes:- the number of resources,
- the levels of message priorities
Action(s): - addressing and message passing
Action attributes:- addressing function (decision variable),
- the vector of times of service of a message by processes of class R_4 in all nodes, $i = 1, N$ (computed parameter),
- mean time of message passing between network interfaces (decision variable)

Processes of class R_5 - Management of data structures describing the location of resources in a network

Object(s): - the table containing a description of resources and their addresses in a network,
- the table of resource states (access aspects; access blocked or not)
Object attributes:- addressing function (decision variable),
- the vector of addresses of the processes of class R_5 describing the location of several classes (generated parameter),
Action(s): - the message addressing
Action attributes:- the vector of mean times of operation completion in all nodes, $i = 1, N$ (computed parameter)

Processes of class R_6 - Management of I/O devices

Object(s): - the table containing the description of I/O devices in the network,
- the queues of demands
Object attributes:- the number of I/O devices,
- the addresses of I/O devices
Action(s): - the set up of I/O operations,
- the insertion of I/O operations to the demand queue
Action attributes:- the time of performing I/O operations (given in the model of environment),
- the vector of times of initiation of I/O operations in all nodes, $i = 1, N$ (decision variable)

Processes of class R_7 - Clock

Object(s): - the table containing states of clocks in nodes (processes, their cycles, time to the end of a cycle),

- the table of states of processes
Object attributes:- the vector containing numbers of cyclic processes in all nodes, $i = 1, N$ (computed parameter)
Action(s): - the service of a clock (actualization of the clock table)
- the change of a state of a user process connected with that event
Action attributes:- the vector of times of actualization of a description of a given process during a service of a clock interrupt in all nodes, $i = 1, N$ (decision variable),
- the vector of times of changes of processes in all nodes, $i = 1, N$ (decision variable),
- the maximum value of a cycle of a process (decision variable)

Processes of class **R_8** - The service of external events (Z_1)
Object(s): - the lists of processes activated by external events in all nodes, $i = 1, N$,

- the table of states of processes
Object attributes:- the vector containing numbers of processes activated by external events in several nodes, $i = 1, N$ (computed parameter)
Action(s): - the change of a process activated by an external event
Action attributes:- the vector of times of changes of process states in several nodes, $i = 1, N$ (decision variable)

Processes of class **R_9** - Reallocation of processes that are not connected with a node
Object(s): - the table containing costs of processes reallocation,
- the vector of accessability of resource addresses (blocked access or not blocked one),
- the queues of blocked messages,
- the queues of blocked demands of process reallocation
Object attributes:- the type and parameters of a distribution of an address reallocation,
- the type and parameters of a distribution of a demand of reallocation
Action(s): - the block of an access to a resource,
- the message passing,
- the actualization of an address and releasing of a blocking of an access to a resource
Action attributes:- the priority of processes of class R_9 (decision variable),
- the unit cost of a process reallocation (constant),
- the vector of times of preparation of one message in a process of class R_9 (decision variable)

Processes of class **R_{10} - R_p** - Management of logical resources
Object(s): - the table of processes of class R_{10} - R_p,
- the queues of demands
Object attributes:- the priorities of processes (decision variable)
Action(s): - performing an "empty" operation on a logical resource (access to a processor); this operation is interruptable taking into account a processor and uninterruptable taking into account a resource
Action attributes:- the time of performing an operation

5. 2. Objects of an Environment of the Network Operating System

The objects of an environment in the simulation tool are data structures containing a model of user processes. The processes are characterized by the following set of attributes:

- the address of a process
- the priority of a process
- the activating event
- the sequence of events modelling a process
- the word of a process state (set up during a simulation)
- the state of a process (ready, computed, suspended) - set up during a simulation.

The constant attributes of a process (address, priority, event) are generated using an uniform distribution of a given attribute.

Events modelled in process Z_5 are described by the following attributes:

- the type of an event (receive, send, end)
- the receiver or a sender of messages generated by an event
- the identifier of a message (compuetd parameter)
- the relative time of an instance of the next event
- the time of an operation on a resource
- the real time of an an instance of the event (set up during a simulation)
- the priority of an event (the priority of a message).

5. 3. The Events of the Simulation Tool

The following events of the simulation tool have been distinguished:

a) interrupts
b) completion of an interrupt service
c) completion of a scheduling
d) completion of the service of a processor dispatcher
e) measurement of given performance indices.

The diagram of events in the simulation tool is given in Fig. 3.

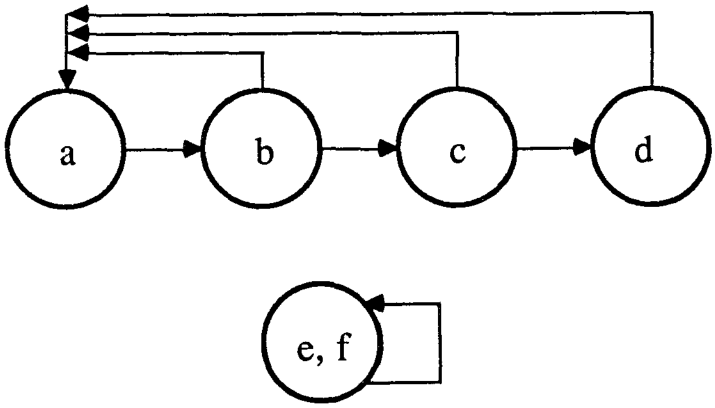


Fig. 3. Sequence of events (for a node operating system) in the simulation tool

Each event is characterized by the following attributes:

- the type of an event
- the address of an event (the number of the network node)

- the time of an instance of the event.

Moreover, all events have been described by the following parameters:

Events Z_1, Z_6

the probability of an instance of the event
the distribution of arrival moments of events
the moment of instance of the next event (computed value)

It has been assumed that random events Z_1 and Z_6 have a Poisson distribution with events intensity c.

Event Z_2

the priority of a message
the identifier of a receiver
the identifier of a sender

Event Z_3

the moment of an instance of the next event (computed value)

Event Z_4

the address of a receiver of an operation result
the moment of an instance of the next event (computed value)

Event Z_5

the attributes of events Z_5 have been characterized in a model of an environment

5. 4. Parameters of Processes Managing Resources

In the set of parameters describing processes managing resources (Section 5. 1.) one can distinguish: (i) a subset of parameters characterizing operations in processes (priorities of events, levels of process priorities, levels of message priorities, etc.) and (ii) a subset of parameters characterizing times of performing of operations. Set (i) of parameters controls the simulation tool and depends on definitions of operations in managing processes. On the other hand, set (ii) of parameters makes it possible an assesment of a service time of a demand by managing processes.

To define a mean service time $T_{ji}(t)$ in a process of class R_j performed in the i-th node at time t, the set of decision variables has been set up. These decision variables are variables characterizing computers and communication systems and they depend on times of performing the following operations:

w_{i1} - the time to accept of an interrupt in node i, i = 1, N
w_{i2} - the time to change state of a processor in node i, i = 1, N
w_{i3} - the mean time of initiation of I/O operations in node i, i = 1, N
w_{i4} - the mean time of an access to one element of a data structure describing the location of resources (the time of an access to an element of a table in node i, i = 1, N)
w_{i5} - the mean time of process scheduling in node i, i = 1, N
w_{i6} - the mean time of actualization of description of a cyclic process in node i

w_{i7} - the mean time of a change of a process state

tt_k - the mean time of message passing between two processes managing message passing, where k, k = 1, ..., 5, defines the type of addressing function.

In the simulation studies reported here, the values of these variables have been assessed on the basis of adequate operations and mean times of instruction performed in microprocessor INTEL 8080.

Vectors (w_{i1}, ..., w_{i7}), i = 1, N, and (tt_1, ..., tt_5) are the most fundamental decision variables of the simulation tool and were the basis for computation of mean times of services in the processes managing resources. The method of estimation of these mean times has been presented in [Ind 85].

5. 5. The Implementation of the Simulation Tool

The program implementing the simulation tool for the performance study of methods of resource addressing of the network operating system has a block structure based on the logical model [Gos 86].

To achieve the portability, the tool has been written in Fortran IV. The memory size is 250Kbytes.

There are the following tool restrictions:
- the maximum numer of network nodes = 20
- the maximum number of user processes = 100
- the maximum number of I/O devices = 30
- the maximum number of cyclic processes = 70

6. TOOL TESTING AND SIMULATION EXPERIMENTS

6. 1. Performance Indices

There is a lack of total performance indices for operating systems. Moreover, performance indices used and presented in the literature do not have uniform definitions. That implies that operating systems could be well assessed only in several classes of applications.

The following performnace indices have been taken into consideration:
E1 - the mean time of response to an event
E2 - the mean time of a demand service
T - the overhead of a network by the operating system
U - the utilization of processors - the relative time spent in the state "problem"
S - the relative time spent by processors in the state "system"
W - the relative time spent by processors in the state "wait"
L1 - the number of served demands of classes Z_1, Z_3, Z_5
L2 - the number of rejected demands of the activation of random processes, Z_1, Z_2
L3 - the number of rejected demands of the activation of cyclic processes, Z_3

These performance indices could be divided into three subsets:
Min = {E1, E2, T, S, L = L2 + L3} - these indices should be minimized
Max = {L1, U} - these indices should be maximized
Dep = {W} - this index should be minimized or maximized; it depends on values of indices

from both sets Min and Max.

6. 2. Simulation Experiments

Simulation experiments have been started to check the quality of the simulation tool developed and to compare network operatings system with different resource addressing methods for networks connecting a small number of computers.

The following resource addressing methods have been studied:
M1 - addresses known in each node
M2 - centralized addressing
M3 - centralized addressing with division into resource classes
M4 - local addresses known in each node
M5 - local addresses known in each node and broadcast message passing

The simulation experiments have been carried out for a chosen class of network operating systems with changing values of decision variables which influence distributions of demands: the number of network nodes, the number of user processes, the number of resources available by processes, the frequency of demands of process activation $p_i \in P_L$, the maximum cycle of process $p_i \in P_T$, the priorities of interrupts.

The class of the network operating system has been set up by a choice of decision variables $w_{i1}, ..., w_{i7}$, $i = 1, N$, which are used to evaluate service times in processes managing resources. The values of these variables are as follows:

w_{i1} = 0.50E-05, w_{i2} = 0.50E-04, w_{i3} = 0.50E-05, w_{i4} = 0.30E-04, w_{i5} = 0.90E-04, w_{i6} = 0.90E-04, w_{i7} = 0.25E-04.

It has been assumed that mean time of message passing between two nodes of the network for the k-th type of addressing function is: tt_k = 2 ms, k = 1, 5.

In each simulation run the set of user processes was generated from the set of 60 user processes constructed according to a definition of user processes (Section 5.2).

The results of simulation for the following valus:
the number of node computers N = 3, 5
the number of processes = 60,
the number of I/O devices = 4
has been presented in Tables 1, 2, 3, 4.

The results obtained could be expressed in the following way:

a) for performance indices from set Min
E1(M4) < E1(M5) < E1(M3) < E1(M1) < E1(M2)
E2(M4) < E2(M5) < E2(M1) < E2(M3) < E2(M2)
T(M4) < T(M5) < T(M3) < T(M1) < T(M2)
S(M4) < S(M5) < S(M3) < S(M1) < S(M2)
L(M4) < L(M3) < L(M1) < L(M5) < L(M2)

b) for performance indices from set Max
U(M4) > U(M5) > U(M1) > U(M3) > U(M2)

L1(M4) > L1(M5) > L1(M1) > L1(M3) > L1(M2)

c) for the performance index from set Dep
W(M1) < W(M5) < W(M4) < W(M3) < W(M2)

Table 1. Performance indices of processors (number of computers = 3, c = 0.03)

Computer	Performance Index	Addressing Method				
		1	2	3	4	5
1	S	0.284187E 00	0.128051E 00	0.133701E 00	0.163676E 00	0.182957E 00
	U	0.595263E 00	0.597080E 00	0.598100E 00	0.597650E 00	0.596948E 00
	W	0.119268 00	0.273914E 00	0.267243E 00	0.237338E 00	0.218616E 00
	T	0.477413E 00	0.214462E 00	0.223542E 00	0.273846E 00	0.306487E 00
2	S	0.196282E 00	0.889094E - 01	0.926102E - 01	0.130814E 00	0.162608E 00
	U	0.771908E 00	0.764672E 00	0.795583E 00	0.833785E 00	0.818626E 00
	W	0.306032E - 01	0.145329E 00	0.110651E 00	0.335060E - 01	0.163362E - 01
	T	0.254282E 00	0.116271E 00	0.116406E 00	0.156892E 00	0.198635E 00
3	S	0.292106E 00	0.577599E 00	0.454699E 00	0.213762E 00	0.241253E 00
	U	0.660244E 00	0.402095E 00	0.519756E 00	0.739419E 00	0.219388E 00
	W	0.463656E - 01	0.193103E - 01	0.241826E - 01	0.453287E - 01	0.375602E - 01
	T	0.442422E 00	0.142647E 01	0.874832E 00	0.289095E 00	0.335359E 00

The results show that the network operating system using resource addressing methods M4 - local addresses known in each node and M5 - local addresses known in each node and broadcast message passing are more effective. At the same time the minimal value of performance indices W(M5) and W(M1) show a big system overload. An increase of frequency of activation of processes from 0.03 to 0.3 for the same set of user processes relatively decreases the quality of the system with method M5 against to the system with method M4 (see Tables 2 and 3). The systems with different addressing methods are studied now very extensively.

6. CONCLUSION

In this paper a simulation tool to study the effectiveness of different methods of resource addressing in network operating system has been presented. The model could be used to study network operating systems with static resource addressing as well as network operating systems with a dynamic change of the location of resources in the network. The model of a network operating system has been extended by adding a model of demands of the environment of the network operating system. The later has been constructed such that it makes it possible to eliminate consideration of an influence of different methods of resource addressing on a distribution of moments of arrival of processes demands.

Table 2. Global performance indices (number of computers = 3, c = 0.03)

Performance Index	Addressing Method				
	1	2	3	4	5
L1	4610	4126	4419	4893	4878
E1	0.352220E -03	0.111643E - 02	0.598065E - 03	0.846101E - 04	0.138621E - 03
E2	0.473762E 01	0.524675E 01	0.491808E 01	0.443711E 01	0.450458E 01
L2	7	4	0	7	33
L3	49	69	55	50	31
S	0.257525E 00	0.264853E 00	0.227003E 00	0.169418E 00	0.195606E 00
U	0.675805E 00	0.587949E 00	0.637813E 00	0.723633E 00	0.711654E 00
W	0.654124E - 01	0.146184E 00	0.134026E 00	0.105391E 00	0.908374E - 01
T	0.381064E 00	0.450469E 00	0.355909E 00	0.234121E 00	0.274861E 00

Table 3. Global performance indices (number of computers = 3, c = 0.3)

Performance Index	Addressing Method				
	1	2	3	4	5
L1	3826	4797	5104	5657	5496
E1	0.391105E - 03	0.151816E - 02	0.772578E - 03	0.105841E - 03	0.157550E - 03
E2	0.567532E- 01	0.561692E 01	0.559081E 01	0.547335E 01	0.550633E 01
L2	221	228	223	197	241
L3	115	107	89	76	82
S	0.268607E 00	0.291260E 00	0.254150E 00	0.183808E 00	0.208981E 00
U	0.720748E 00	0.669490E 00	0.718135E 00	0.800391E 00	0.776235E 00
W	0.917357E - 02	0.379444E - 01	0.262203E - 01	0.138218E - 01	0.124721E - 01
T	0.372678E 00	0.435048E 00	0.353903E 00	0.229648E 00	0.269223E 00

The simulation tool has been constructed on the basis of the model of network operating system that has been developed as a tool to carry out synthesis of such tools. The simulation tool presented has been constructed in such a way that it makes it possible to study the effectiveness of methods of resource addressing as well as study other aspects of network operating system, e.g., effectiveness of synchronization primitives. That implies that the tool has been constructed at a relatively high level of abstraction, i. e., as a set of processes managing chosen resources, connection between processes and demands controlling a

Table 4. Global performance indices (number of computers = 5, c = 0.03)

Performance Index	Addressing Method				
	1	2	3	4	5
L1	4941	4937	4943	4946	4945
E1	0.274896E - 03	0.231271E - 02	0.386739E - 03	0.206041E - 04	0.469940E - 04
E2	0.453884E 01	0.462971E 01	0.454520E 01	0.453172E 01	0.450178E 01
L2	0	0	0	0	0
L3	1	0	0	0	0
S	0.146827E 00	0.171584E 00	0.112746E 00	0.716289E - 01	0.995256E - 01
U	0.364404E 00	0.364731E 00	0.364712E 00	0.364777E 00	0.364754E 00
W	0.488176E 00	0.463047E 00	0.521876E 00	0.562604E 00	0.534499E 00
T	0.402923E 00	0.470439E 00	0.309136E 00	0.196638E 00	0.272857E 00

behaviour of the network operating system. On the other hand the admissible operations in several processes managing resources have been defined at an implementation stage of the simulation tool development.

The simulation carried out shows that the simulation tool constructed can be effectively used to study the influence of different methods of resource addressing on performance of network operating systems. As has been presented, the simulation process needs some data which could be found based on the analysis of existing operating systems, results of simulation of new constructed operating systems, and parameters of computer hardware. At present the wide simulation experiments are carried on to study different methods of resource addressing as well as synchronization problems in network operating system.

ACKNOWLEDGEMENT

We wish to thank G. S. Stafford for his helpful comments.

REFERENCES

[Ban 82] Banino J. S., Fabre J. C., Distributed Coupled Actors: a CHORUS Proposal for Reliability, IEEE, 1982.

[Bro 82] Brownbridge D. R., Harshal L. F., Randell B., The Newcastle Connection of UNIXes of the World Unite!, Software - Practice and Experience, Vol.12, pp. 1147 - 1162, 1982.

[Cab 79] Cabanel J. P., Marouane M. N., Besbes R., Sasbon R. D., Diarpa A.K., A Decentralized OS Model for Aramis Distributed computer System, IEEE 1979.

[Col 82] Collinson R. P. A., The Cambridge Ring and UNIX, Software - Practice and Experience, 1982, Vol. 12, pp. 583 - 594, 1982.

[Deg 80] Degenhardt K. H., Wiesnerg., Woletz W., Distributed Control and Data Processing with Modified Real - Time Operating System, Real - Time Data Handling and Process Control, North -Holland Publishing Company, Brussel and Luxemburg, 1980.

[Des 79] Deschizeaux P., Griesner R., Ladet P., A Real - Time Operating System for Microcomputer Network, SOCOCO'79, Prague, 1979.

[Des 80] Deschizeaux P., Ladet P., Real - Time Structuration Language for Decentralized Process Control, Real - Time Data Handling and Process Control, North -Holland Publishing Company, Brussel and Luxemburg, 1980.

[Fle 80] Fletcher J., G., Watson R. W., An Architecture for Support of Network Operating System Services, Computer Networks, Vol. 4, No. 1, 1980.

[Fri 83] Friedrich G. R., Eser F. W., Management Units and Interprocess Communication in DINOS, Simens Forsch. - und Entwickl., -Ber., Bd. 12, No. 1, pp. 21 - 27, 1983.

[Gos 84] Goscinski A., Indulska J., An Object Approach to Network Operating System Model Construction, IEEE Technical Committee on Distributed Processing Newsletter, network operating system, Vol. 6, No. SI - 2, 1984.

[Gos 86] Goscinski A., Indulska J., A Model of a Distributed Operating System, Report No. 86.7, Department of Computing Science, The University of Wollongong, August 1986.

[Gui 82] Guillement M., The CHORUS Distributed Operating System: Design and Implementation, Local Computer Networks, P. C. Ravasio, G. Hopkins, N. Naffah (Editors), North-Holland Publishing Co., IFIP, 1982.

[Ind 85] Indulska J., Studies of Real - Time Distributed Operating System, Ph.D. Thesis, The St. Staszic University of Mining and Metallurgy, Krakow, 1985.

[ISO 81] ISO/TC9/SC16, Data Processing - Open Systems Intrconnection - Basic Reference Model, Computer Networks, Vol. 5, 1981.

[Jon 78] Jones A. K., The Object Model: A Conceptual Tool for Structuring Software, Operating Systems: An Advanced Cource, Springer Verlage, pp. 7 - 16, 1978.

[Lin 82] Lin M. T., Tsay D. P., Lian R. C., Design of a Network Operating System for the Distributed Double - Loop Computer Network (DDLCN), Local Computer Networks, Ravasio P. C., Hopkins G., Naffah N. (Editors), North - Holland Co., IFIP, 1982.

[Lis 79] Liskov B., Primitives for Distributed Computing, Proc. 7-th Symp. Operating Syst. Principles, Dec. 1979.

[Ous 80] Ousterhout J. K., Scelza D. A., Sindhu P. S., MEDUSA: An Experimenting Distributed Operating System Structure, CACM, Febr. 1980.

[Paw 68] Pawlak Z., On the Notation of a Computer, Logic. Math. and Phil. Sci. 3,1968.

[Pet 85] Peterson J. L., Silberschatz A., Operating Systems Concepts, Addison - Wesley Publishing Co., 1985.

[Ras 81] Rashid R. F., Robertson G. G., ACCENT: A Communication Oriented Network Operating System Kernel, Technical Report, Cornegie-Mellon University, April 1981.

[Wat 80] Watson R. W., Network Architecture Design for Back - End Storage Networks, Computer, pp. 32 - 48, February 1980.

[Wat 81] Watson R. W., Distributed System Architecture Model, Distributed Systems - Architecture and Implementation. An Advanced Course, Springer Verlag, 1981.

[Zim 81] Zimmerman H., Banino J. S., Caristan A., Guillenment M., Basic Concepts for the Support of Distributed Systems, The CHORUS Approach, Proc. of the Second International Conference on Distributed Computing Systems, France, April 1981.