

## **An Abstract of Master Thesis**

Anton Burtsev

Thesis Title:

### **Architectural Design of Operating System Based on Distributed Objects**

(Developed in co-authorship with Leonid Ryzhyk)

July, 2002

The thesis presents the architectural design of E1 operating system, aimed to control resources of computer network.

In E1 state and functionality of all operating system components is encapsulated by distributed objects. Objects are globally accessible by their interfaces from all nodes of a system. Reliability and efficiency of distributed computations are achieved by means of object replication. A complete or partial copy of a distributed object's state can be placed in each node where the object is used. The state of an object is synchronized (replicated) among nodes. E1 allows selecting for each object the most efficient replication algorithm, which takes into account its semantics. There exist a number of universal replication strategies, e.g., active and passive replication, which can be efficiently applied to various types of objects. However, the replication strategy can be designed for a particular type of objects, which allows to maximize the efficiency of access by taking type-specific properties into account.

E1 runs on top of a microkernel which supports a minimal set of primitives like address spaces, threads, IPC, and interrupts dispatching. All operating system and application functionality is implemented by distributed objects. Microkernel-based architecture improves modularity and reliability of the system, as well as reduces control transfer costs via the kernel, which is especially important for the systems oriented at intensive communication of medium-grained objects.

All E1 objects reside within a single virtual 64-bit address space divided into isolated protection domains. Addresses of the object interfaces serve as globally unique identifiers, using which object can be invoked from any network node. Such virtual memory organization provides convenient object communication environment.

The E1 execution model is based on the migrating threads concept. In E1 a thread is not permanently bound to any specific object or domain, but transfers execution between objects on method calls. The migrating threads model simplifies object development, results in more lightweight objects, and improves the efficiency of object communication.

Following the current trend towards component software, E1 provides support for component-oriented programming. Development and use of components require a component model, defining a set of services, interfaces and conventions, all of which constitute the execution and communication environment for components. Current component models are usually implemented as middleware systems. In contrast, E1 implements component model by extending the distributed object model with component-oriented services and tools, thus avoiding the use of an additional software layer. Besides being highly efficient, such architecture simplifies the development and use of components, since in E1 distributed objects (or components) are first-class citizens, like files in UNIX. The E1 component model provides services for object protection, global object naming, dynamic class loading, garbage collection, as well as component development tools like IDL compiler and replication strategy compiler.

E1 is intended as a software platform for the clusters of workstations and distributed data processing systems, e.g., distributed CAD and DBMS software.