Table of Contents

VEOS functional characteristics and life cycle

Functional characteristics

The VEOS operating system (hereinafter referred to as VEOS) is a set of integrated programs created on the basis of the Linux OS, and provides processing, storage and transmission of information in a secure software environment in round–the-clock operation. VEOS has the following functional characteristics:

- provides the ability to process, store and transmit information;
- provides the ability to function in multitasking mode;
- provides the ability to scale the system: it is possible to operate the OS both on physical hardware and on a virtual machine;
- provides multi-user operation mode;
- provides support for multiprocessor systems;
- provides virtual memory support;
- provides support for running virtual machines;
- provides network data processing, including access control to network packets.

VEOS supports client-server architecture and can serve processes both within one computer system and processes on other PCs via data transmission channels or network connections.

VEOS consists of a set of components designed to implement functional tasks necessary for users (officials to perform daily actions defined by job descriptions) and is supplied as a distribution kit and a set of operational documentation. The following functional elements can be distinguished in the VEOS structure:

- OS kernel;
- system libraries;
- utilities and drivers;
- information security tools;
- system applications;
- cloud and distributed computing support tools, virtualization tools and data storage systems;
- monitoring and control systems;
- executable code preparation tools;
- source code revision tools;
- subroutine libraries (SDK);
- development, testing and debugging environments;
- interactive working environments;
- software servers;
- database management systems;
- command interpreters;
- general purpose application software;
- office applications.

The VEOS kernel controls access to RAM, network, disk and other external devices. It starts and registers processes, manages the time division between them, implements the rights differentiation and defines a security policy that cannot be avoided without referring to it.

The kernel operates in the "supervisor" mode, which allows it to have access to the entire RAM and hardware task table at once. Processes run in "user mode": each is bound by the kernel to one entry in the task table, which, among other data, indicates which part of RAM this process has access to.

The kernel is constantly in memory, executing system calls – requests from processes to execute these routines.

System libraries are sets of programs (software packets) that perform various functional tasks and are intended for dynamic connection to running programs that need to perform these tasks. Server programs and applications provide the user with services (mail services, file storage, database management system, document management, user data storage, and so on) on a local or global network and ensure their execution. VEOS includes the following additional system applications:

- archivers;
- applications for managing RPM packets;
- backup applications;
- system monitoring applications;
- applications for working with files;
- applications for system configuration;
- configuring download settings;
- hardware setup;
- network setup.

Life cycle

As a technological complex for the VEOS release, an infrastructure based on the free Linux software assembly and supporting technologies is used. In addition, this infrastructure allows for integration into the operating system environment and support for software products by various developers.

The software support, development and update infrastructure includes the following components.

Software source code repository

The repository stores all the source codes along with the revision history. Several branches of the same project are maintained by both one developer and different ones, in order to support several distribution versions, as well as several distributions. The source code repository is integrated with the packet assembly system.

Packet assembly system

When assembling a packet from the source code, a virtual file system is automatically generated, which guarantees the reproducibility of the assembly regardless of the assembly system configuration. Assembling packets in a virtual environment allows to fix the assembly environment, and also ensures its security. The assembly system is integrated with the software source code repository and allows to track the relationship between the assembled packets and the source code, which simplifies testing and debugging solutions.

Assembly parallelization system

Provides distributed packet assembly on available computer capacities, which allows to achieve optimal packet assembly time and, if necessary, reassembly of the entire repository.

Unified software packet management system

Software management is carried out by a single system representing software in the form of so-called packets and controlling dependencies between packets, including packet versions. The presence of this system allows to form distributions with controlled closeness by dependencies between software components, and in the future – to organize the selective installation of packets or their updating, without violating the system integrity and without losing user settings.

Unified installation system

The installation of the distribution is carried out by a single configurable system integrated with the system object management system and with the packet management system.

Distribution generation system

This system allows to create installation images of distributions (sets of CD or DVD iso images) based on a configuration file (a set of target packets), taking into account dependencies between packets.