Table of Contents

User's Guide

System boot

VEOS is started automatically after the computer is started and the system BIOS is initialized. The GRUB loader menu appears on the screen, which lists the possible options for loading the operating system.

Use the Up and Down arrow keys to select the desired option. In addition to the main OS startup options from this menu, VEOS may be loaded in safe mode or memory check may be run. The default boot of the operating system (the first in the list) starts automatically after a short waiting time (usually a few seconds). By pressing the Enter key, you can start downloading immediately. By pressing the E key, you can open the parameters editor of the current download point. If the system is configured correctly, then there is no need to edit them. During the VEOS boot process, the user can monitor the boot process information, which displays the startup stages of various services and software servers in the form of separate strings on the monitor screen.

Each string begins with the word [Message] (FAILED or OK), which is a sign of normal or abnormal boot stage completion. The word Message=FAILED indicates the unsuccessful completion of the system component loading, which requires the intervention and special actions of the system administrator. Loading the operating system may take some time, depending on the performance of the computer. The main stages of VEOS loading —core boot, attaching (mounting) file systems, starting system services — can be periodically supplemented by checking file systems for errors. In this case, the waiting time may take longer than usual. Detailed information about the boot steps can be obtained by pressing Esc.

Log in to the system

The standard VEOS installation includes a basic system running in a console mode. When booting in a console mode, the VEOS boot ends with a request to enter the account login and password. If necessary, you can switch to another console by pressing Ctrl+Alt+F2. In case of successful authentication and identification procedure completion, a login will be performed. VEOS OS will switch to normal operation mode and provide access to the console for the user.

Documentation

VEOS, like all UNIX family systems, has on-screen documentation. Its texts contain documentation on system commands, resources, configuration files, etc., and can also be displayed on the screen during operation.

MAN Utility

To access the on-screen documentation, use the man command (short for manual). Each page of the manual is dedicated to one system object. In order to read the program manual page, you need to type man program_name. For example, if you want to find out what options the ls command has, you can enter the command:

```
> man ls
```

Most of the on-screen documentation is written for users who have some idea of what this command does. Therefore, most of the on-screen documentation texts contain only technical details of the command without much explanation. Nevertheless, the on-screen documentation turns out to be very valuable if you remember the name of the command, but its syntax just fell out of your memory.

The search for man descriptions is carried out by the apropos command. If you do not know exactly what the name of the program you need is, then the search is carried out by a keyword, for example, apropos ls, or by entering a word denoting the desired action after the man -k command (for example, man -k list). The word describing the action desired for you can be entered in Russian. If there is a Russian translation of the man manual pages, the search results will be displayed in the requested language. The "manual page" usually takes up more than one screen page. In order to make it easier to read, man launches a page-by-page text viewing program. The pages are scrolled with a space bar, to exit the man command description reading mode, you need to press q on the keyboard. The man man command issues a certificate on the use of the man command itself.

INFO Utility

Another source of information about VEOS and its component programs is the info reference subsystem. The manual page, despite the abundance of links of various types, remains a "linear" text, structured only logically. The info document is a real hypertext in which many small pages are combined into a tree. In each section of the info document, there is always a directory from which you can go to the desired subsection and then return back (links to navigate through text sections are marked with *). To get auxiliary information about moving through the text, use the h key. The complete info manual is called by the info info command. The info command, entered without parameters, offers the user a list of all info documents installed in the system.

Installing additional software

To install additional software in VEOS, dnf (or yum) utilities are used to download rpm packets from VEOS repositories and rpm for installation from rpm files.

> ⚠️ To install the software, the user shall have system administrator rights (root) or permission to run the sudo utility

Installing a packet from the repository

To install or update a packet from the repository, run the following command:

```
> dnf install <packet name>
```

To delete an installed packet:

```
> dnf remove <packet name>
```

Installing a packet from a file

Sometimes locally downloaded rpm packet is to be installed:

```
> rpm -ih <rpm-файл>
```

To update the packet:

```
> rpm -Uh <rpm-файл>
```

Updating installed packets

Periodically, updates of installed packets appear in the VEOS repositories, to install packet updates, you need to run:

```
> dnf update
```

During the update process, the utility will detect the latest non-installed versions for all system packets, calculate their dependencies, download them and install them after user confirmation.

Installation of VAS Experts software

By default, VEOS has access to the VAS Experts repositories configured and all the company's packets are available for installation without additional settings. For example, the installation of the DPI packet is performed as follows:

```
> dnf install fastdpi
```

Overview of the main system commands

All commands listed below can be run in a console mode. For more information, use the man command. Example:

```
> man ls
```

Note: Command parameters usually start with the "-" character, and usually after one "-" character, several options can be specified at once. For example, instead of the ls -l -F command, you can enter the ls -lF command.

User accounts

The su command

The su command allows to change the "owner" of the current session without the need to terminate the session and open a new one. Syntax:

```
> su [OPTIONS...] [USER]
```

The command can be used to replace the current user with any other, but most often it is used to obtain superuser (root) rights by the user. When entering the su - command, the superuser (root) password will be requested, and if the correct password is entered, the user will receive administrator rights. To return to user rights, enter the command:

```
> exit
```

For more information about superuser mode, see the Superuser Mode chapter

The id command

The id command displays information about the user and the groups in which they are a member for a given user or about the current user (if nothing is specified). Syntax:

```
> id [OPTIONS...] [USER]
```

The passwd command

The passwd command changes (or sets) the password associated with the login username. An general user can only change the password associated with their own login name. The command requests the old password from general users (if there was one), and then twice requests the new one. The new password shall meet the technical requirements for passwords set by the system administrator. Basic operations with files and directories.

Basic operations with files and directories

The ls command

The ls (list) command types the contents of directories to standard output. Syntax:

```
> ls [OPTIONS...] [FILE...]
```

Basic options:

- -a — view all files, including hidden ones;

- -l — display more detailed information;
- -R — output information about subdirectories recursively.

The cd command

The cd command is used to change the directory. The command operates with both absolute and relative paths. If the directory is not specified, the value of the $HOME environmental variable (the user's home directory) is used. If the directory is given a full route name, it becomes the current one. In relation to the new directory, you need to have the right to execute, which in this case is interpreted as a search permission. Syntax:

```
> cd [-L|-P] [DIRECTORY]
```

If "-" is given as an argument, then this is equivalent to $OLDPWD. If the transition was made by the $CDPATH environment variable or "-" was set as an argument and the directory change was successful, then the absolute path of the new working directory will be output to standard output. Example. While in the home directory, go to its docs/ subdirectory (relative path):

```
> cd docs/
```

Make the current directory /usr/bin (absolute path):

```
> cd /usr/bin/
```

Make the parent directory a current one:

```
> cd ..
```

Go back to the previous directory:

```
> cd -
```

Make the home directory a current one:

```
> cd
```

The pwd command

The pwd command outputs the absolute path of the current (working) directory. Syntax:

```
> pwd [-L|-P]
```

Options:

- -P — do not output symbolic links;
- -L — output symbolic links.

The rm command

The rm command is used to delete file entries. If the specified name was the last link to the file, then the file is destroyed.

Warning: If you delete a file, you will not be able to restore it!

Syntax:

```
> rm [OPTIONS...] <FILE>
```

Basic options:

- -f — never ask for confirmation;
- -i — always request confirmation;
- -r, -R — recursively delete the contents of the specified directories.

Example. Delete all html files in the ~/html directory:

```
> rm -i ~/html/*.html
```

The mkdir command

mkdir is a command for creating new directories. Syntax:

```
> mkdir [-p] [-m rights] <DIRECTORY...>
```

The rmdir command

The rmdir command deletes directories from the file system. The directory shall be empty before deleting. Syntax:

```
> rmdir [OPTIONS] <DIRECTORY...>
```

Basic options:

- -p — delete the directory and its descendants.

The rmdir command is often replaced by the rm -rf command, which allows to delete directories even if they are not empty.

The cp command

The cp command is designed to copy files from one directory to other directories. Syntax:

```
> cp [-fip] [SRC_FILE...] [DST_FILE...]
> cp [-fip] [SRC_FILE...] [DIRECTORY]
```

```
> cp [-R] [[-H] | [-L] | [-P]] [-fip] [SRC_FILE...] [DIRECTORY]
```

Basic options:

- -p — save, if possible, the time of modification and access to the file, the owner and group, access rights;
- -i — request confirmation before copying to existing files;
- -r, -R — recursively copy the contents of directories.

The mv command

The mv command is designed to move files. Syntax:

```
> mv [-fi] [SRC_FILE...] [DST_FILE...]
> mv [-fi] [SRC_FILE...] [DIRECTORY]
```

In the first syntactic form, characterized by the fact that the last operand is neither a directory nor a symbolic link to a directory, mv moves the src_file to the dst_file (the file is renamed). In the second syntactic form, mv moves the source files to the specified directory under names that match the short names of the source files. Basic options:

- -f — do not ask for confirmation of overwriting existing files;
- -i — request confirmation of overwriting existing files.

The cat command

The cat command sequentially outputs the contents of the files. Syntax:

```
> cat [OPTIONS] [FILE...]
```

Basic options:

- -n, –number — number all lines when outputting;
- -E, –show-ends — show $ at the end of each line.

If no file is specified, standard input is read. If the name "-" is present in the file list, standard input is read instead of this file.

The head command

The head command outputs the first 10 lines of each file to standard output. Syntax:

```
> head [OPTIONS] [FILE...]
```

Basic options:

- -n, –lines=[-]K — print the first K lines of each file, not the first 10;
- -q, –quiet — do not print headers with file names.

The less command

The less command allows to view the text page by page (to exit, press q). Syntax:

```
> less FILE
```

The grep command

The grep command has many options and provides the possibility to search for a character string in a file. Syntax:

```
> grep [search_template] FILE
```

The chmod command

The chmod command is designed to change the access rights of files and directories. Syntax:

```
> chmod [OPTIONS] MODE[,MODE]... <FILE>
> chmod [OPTIONS] --reference=IFILE <FILE>
```

Basic options:

- -R — recursively change the mode of access to files located in the specified directories;
- –reference=IFILE — use IFILE file mode.

chmod modifies the access rights of each specified file in accordance with the access rights specified in the mode parameter, which can be represented either in a character form or as an octal representing the bit mask of the new access rights. The character mode format is as follows:

```
> [ugoa...][[+-=][permissions...]...]
```

Here permissions are zero or more letters from the lettering "rwxXst" or one of the letters from the lettering "ugo". Each argument is a list of charactercoded commands for changing access rights, separated by commas. Each such command starts with zero or more letters "ugoa", the combination of which indicates whose access rights to the file will be changed: the user who owns the file (u), users from the group to which the file belongs (g), other users (o) or all users (a). If no letter is set, the letter "a" will be used automatically, but the bits set in the umask will not be affected.

The "+" operator adds the selected access rights to those already available for each file, "-" removes these rights. "=" assigns only these rights to each specified file. The letters "rwxXst" specify access bits for users: "r" — read, "w" — write, "x" — execute (or search for directories), "X" — execute/search only if it is a directory or a file with an already installed execution bit, "s" — set user and group ID at execution, "t" — prohibition of deletion.

Examples. Allow everyone to execute the f2 file:

```
> chmod +x f2
```

Prevent f3 file deletion:

```
> chmod +t f3
```

The chown command

The chown command changes the owner and/or group for each specified file. Syntax:

```
> chown [KEY]...[OWNER][:[GROUP]] <FILE>
```

Only the file owner or superuser can change the owner. The owner does not change if it is not specified in the argument. The group also does not change if it is not specified, but if there is a ":" symbol after the symbolic OWNER, it means changing the group to the main group of the current user. The OWNER and GROUP fields can be either numeric or symbolic. Examples. Change the owner of the /u directory to the test user:

```
> chown test /u
```

Change the owner and group of the /u directory:

```
> chown test:staff /u
```

Change the owner of the /u directory and attached files to test:

```
> chown -hR test /u
```

File Search

The find command

The find command is designed to search for all files starting from the root directory. The search can be performed by the name, type, or owner of the file. Syntax:

```
> find [-H] [-L] [-P] [-Olevel] [-D help|tree|search|stat|rates|opt|exec]
[PATH...] [EXPRESSION]
```

Search keys:

- -name — search by file name;
- -type — search by type f=file, d=directory, l=link(lnk);
- -user — search by owner (name or UID).

When the find command is executed, you can perform various actions on the found files. Basic actions:

- -exec command \; — execute the command. The command entry shall end with an escaped semicolon. The "{}" string is replaced with the current route file name;

- execdir command \; is the same as -exec, but the command is called from the subdirectory containing the current file;
- -ok command is equivalent to -exec except that before executing the command, confirmation is requested (in the form of a generated command string with a question mark at the end) and it is executed only when the answer is: y;
- -print — displays the file name on the screen.

The default path is the current subdirectory. The default expression is print. Examples. Find regular files (not directories) in the current directory which name starts with the "~" character:

```
> find . -type f -name "~*" -print
```

Find files modified later than file.bak in the current directory:

```
> find . -newer file.bak -type f -print
```

Delete all files named a.out or *.o that have not been accessed for a week:

```
> find / \( -name a.out -o -name '*.o' \) \ -atime +7 -exec rm {} \;
```

Delete all files named a.out or *.o that have not been accessed for a week:

```
> find . -size 0c -ok rm {} \;
```

The whereis command

whereis reports the path to the executable program file, its source files (if any) and the corresponding pages of the reference manual. Syntax:

```
> whereis [OPTIONS] <NAME>
```

Options:

- -b — information output only about executable files;
- -m — information output only about the pages of the reference manual;
- -s — information output only about the source files.

Process monitoring and control

The ps command

The ps command displays a list of current processes. Syntax:

```
> ps [OPTIONS]
```

By default, information about processes with the same valid UID and control terminal is displayed as those of the user's who submits the command. Basic options:

- -a — output information about processes associated with terminals;
- -f — output the "full" list;
- -l — output the "long" list;
- -p list — display information about processes with the PIDs specified in the list;
- -u list — display information about processes with listed identifiers or user names.

The kill command

The kill command allows to terminate the execution of a process or send it a signal. Syntax:

```
> kill [-s] [signal] [identifier] [...]
> kill [-l] [completion_status]
> kill [-signal_number] [identifier] [...]
```

The identifier is the PID of the leading task process or the task number, preceded by a "%" sign. Basic options:

- -l — display a list of supported signals;
- -s signal, -signal — send a signal with the specified name.

If the usual kill command does not give the desired effect, the kill command with the -9 parameter (kill -9 PID_number) shall be used.

The df command

The df command shows the amount of available disk space in the file system that contains the file transferred as an argument. If no file is specified, the available space on all mounted file systems is shown. The default sizes are specified in 1KB blocks. Syntax:

```
> df [OPTIONS] [FILE...]
```

Basic options:

- –total — calculate the total volume at the end;
- -h, –human-readable — print dimensions in a readable format (e.g. 1K, 234M, 2G).

The du command

The du command counts the disk usage by each file, for directories, counting occurs recursively. Syntax:

```
> du [OPTIONS] [FILE...]
```

Basic options:

- -a, –all — output the total amount for each given file, not just for directories;
- -c, –total — calculate the total volume at the end. It can be used to find out the total disk space

usage for the entire list of given files;

- -d, –max-depth=N — output the volume for the directory (or files, if –all is specified) only if it is N or less levels below the command string arguments;
- -S, –separate-dirs — output the size of each directory separately, not including the sizes of subdirectories;
- -s, –summarize — display only the sum for each argument.

The which command

The which command displays the full path to the specified commands or scripts. Syntax:

```
> which [OPTIONS] <FILE...>
```

Basic options:

- -a, –all — outputs all matched executable flies by content in the $PATH environment variable, not just the first one;
- -c, –total — calculate the total volume at the end. It can be used to find out the total disk space usage for the entire list of given files;
- -d, –max-depth=N — output the volume for the directory (or files, if –all is specified) only if it is N or less levels below the command string arguments;
- -S, –separate-dirs — output the size of each directory separately, not including the sizes of subdirectories;
- –skip-dot — skips all directories from the $PATH environment variable that start with a dot.

Using multitasking

VEOS is a multitasking system. In order to run the program in the background, you need to type "&" after the program name. After that, the shell will allow to run other applications. Since some programs are interactive, running them in the background is pointless. Such programs will simply stop if they are run in the background. Also, multiple independent sessions may be run. To do this, you need to type Alt and one of the keys in the range from F1 to F6 in the console. A new system prompt will appear on the screen and you can open a new session. This method also allows you to work on another console if the console you used before is unresponsive or you need to stop a hung program.

The bg command

The bg command allows to move the task to the background. Syntax:

```
> bg [IDENTIFIER ...]
```

The identifier is the PID of the leading task process or the task number, preceded by a "%" sign.

The fg command

The fg command allows to bring the task to the foreground. Syntax:

```
> fg [IDENTIFIER ...]
```

The identifier is the PID of the leading task process or the task number, preceded by a "%" sign.

File compression and packaging

The tar command

File compression and packaging is performed using the tar command, which converts a file or group of files into an archive without compression (tarfile). File packaging into an archive is most often performed with the following command:

```
> tar -cf [name of the archive file to be created] [packaged files and/or directories]
```

Example of using the archive packing command:

```
> tar -cf moi_dokumenti.tar Docs project.tex
```

Unpacking the archive contents to the current directory is performed by the command:

```
> tar -xf [archive file name]
```

Special compression programs are used to compress files: gzip, bzip2 and 7z.