**Федеральное государственное автономное образовательное учреждение высшего образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

**Московский институт электроники и математики им. А.Н.Тихонова**

Департамент компьютерной инженерии

**Лабораторная работа №2**

по курсу «Операционные системы»

Студент: Камаров Лазизбек Шухрат угли

Группа: БИВ203

Преподаватель: Прокофьева   
 Екатерина Николаевна

Дата: 03.10.2022

Москва, 2022

**Задание**

*Работа с интерфейсом и элементами платформы:*

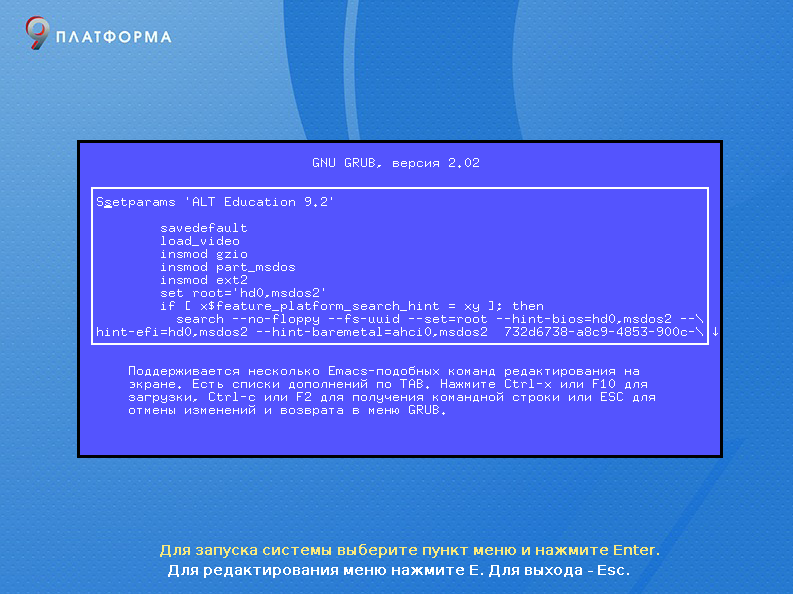
1. Попробовать вызвать редактор параметров текущего пункта загрузки системы.
2. Попробовать получить подробную информацию о шагах загрузки системы.
3. Попробовать произвести проверку оперативной памяти на возможные ошибки посредством использования инструмента Memtest при загрузке.
4. Попробовать выбрать и переключить графические оболочки XFCE и KDE5.
5. Попробовать работу с меню специальных возможностей по шрифту и контрастности.
6. Поработать с «Центром управления системой» для запуска менеджера локальных учётных записей и входа суперпользователя и создания новой учётной записи пользователя.
7. Осуществить блокировку и разблокировку экрана. Попробовать добавить элементы на панель рабочего стола и создать ссылку URL через контекстное меню для элемента.

*Работа с терминалом и базовыми командами – функциональное тестирование возможностей на платформе Альт Образование 9.2:*

Запустите терминал на платформе Альт Образование 9.2. Среди представленного в таблице 1 набора команд попробуйте запустить не менее 50 % в заданной системе (особое внимание к блокам 1,2,5,6) – выявите, какие из приведённых являются рабочими, а какие неверно представлены/ недоступны для нашей версии дистрибутива, опишите процессы посредством визуальных копий экрана с выполнением выбранных команд и аналитических комментариев.

**Выполнение**

1. Для этого запустим виртуальную машину с установленным дистрибутивом Alt Linux. Нажатием клавиши «Е» можно вызвать редактор параметров текущего пункта загрузки. В этом окне можно увидеть параметры загрузки, которые можно менять в зависимости от потребностей.

****

2. После, во время загрузки, нажимаем любую клавишу и наблюдаем дополнительную информацию по загрузке, причем пошагово.

Изображение выглядит как текст

Автоматически созданное описание

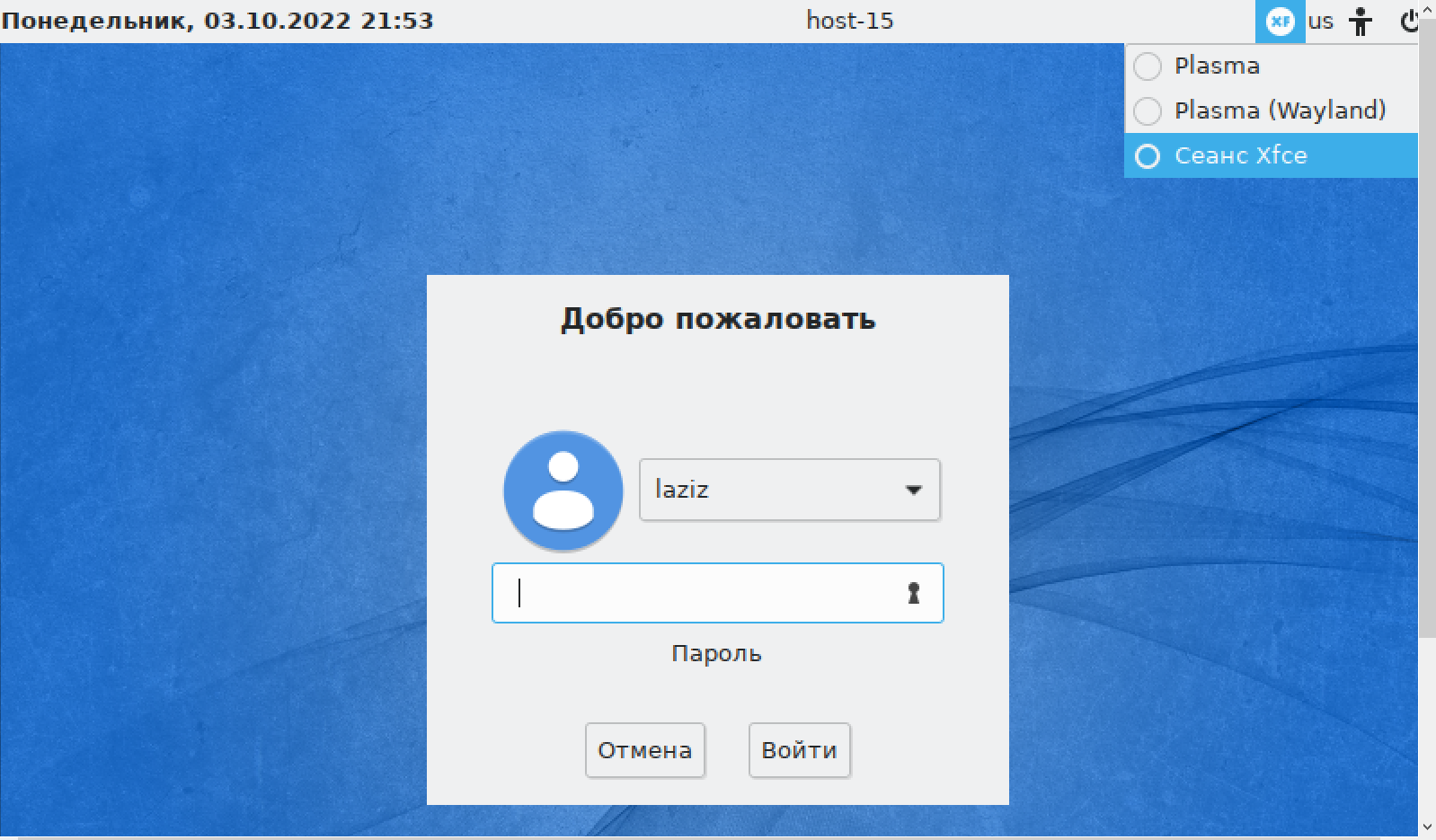
На скриншоте видно, как процессы загружаются различные менеджеры, серверы и прочее.

3. Перезапустим ВМ и, вместо запуска Alt Edu, запустим тестирование оперативной памяти MemTest.

Изображение выглядит как текст

Автоматически созданное описание

4. Запустим заново Alt Edu и посмотрим на оболочку KDE 5. В правом верхнем углу при запуске системы выберем графическую оболочку.

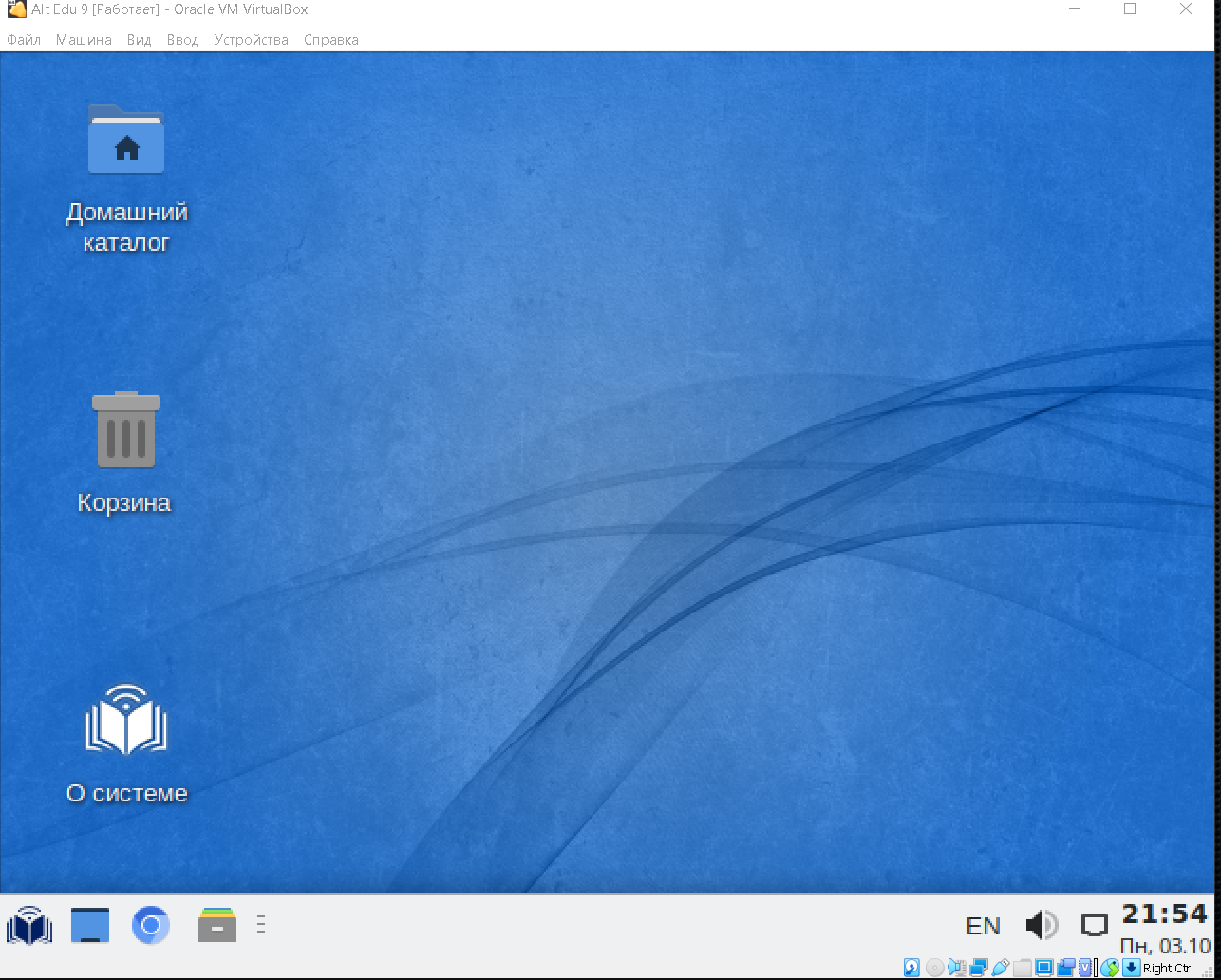


Основные отличия разных оболочек – меню пуск, рабочий стол, доп. функции на рабочем столе и файловый менеджер, поэтому сравним их.

Xfce:

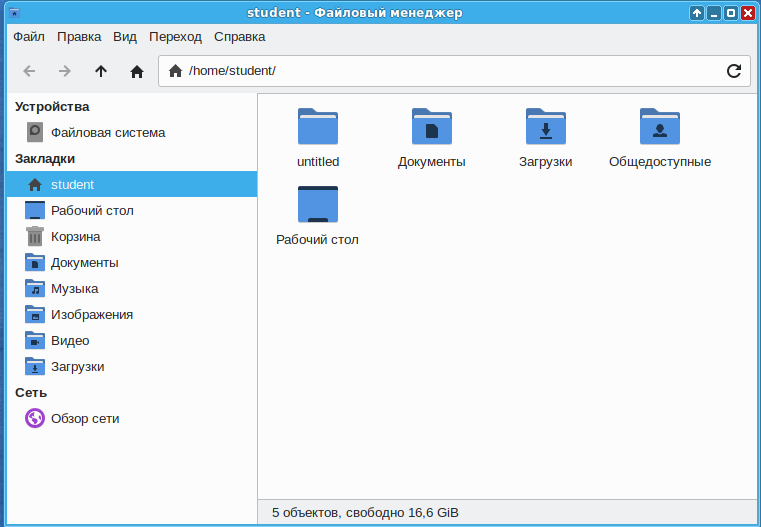
Изображение выглядит как текст

Автоматически созданное описание



Изображение выглядит как текст

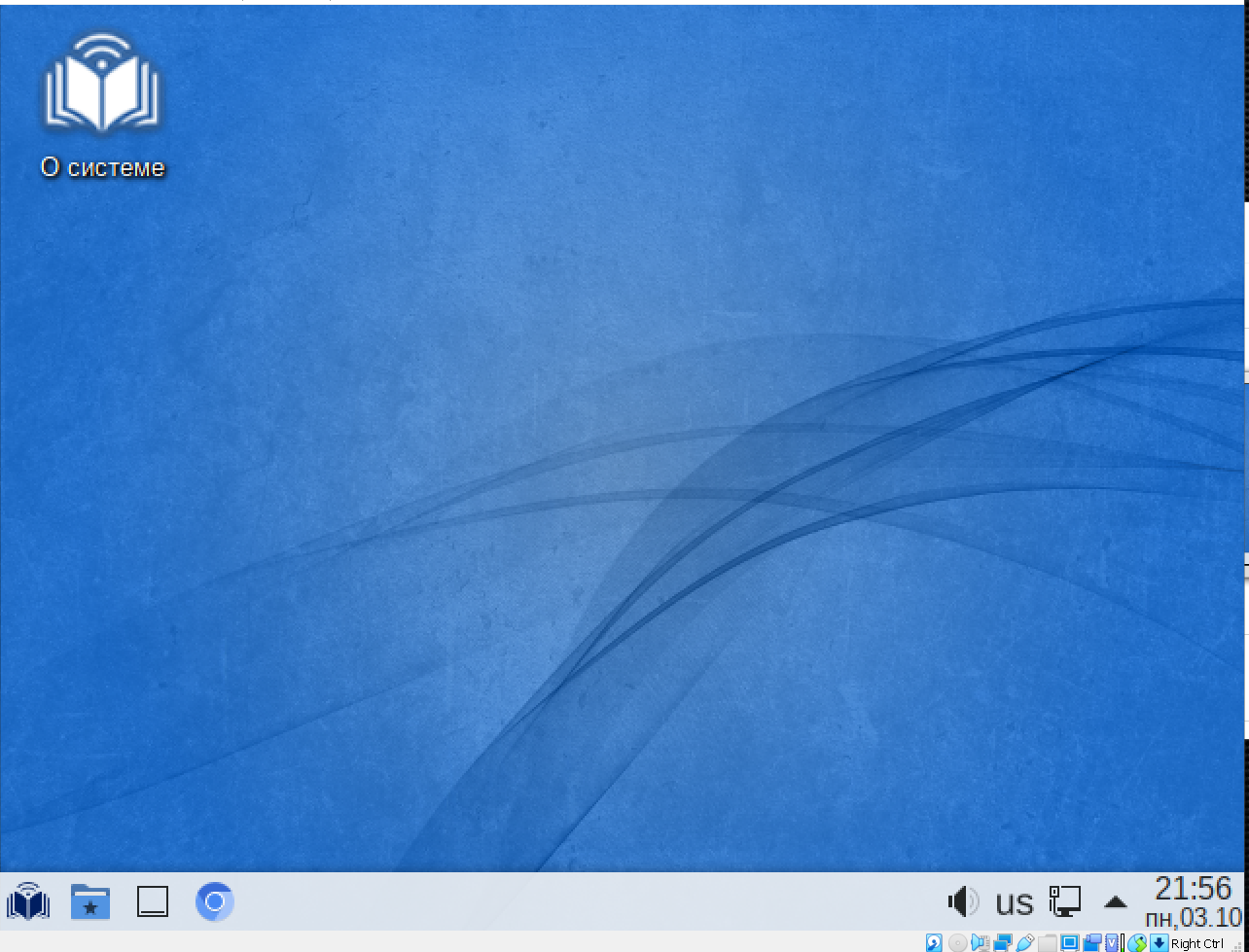
Автоматически созданное описание



KDE 5:

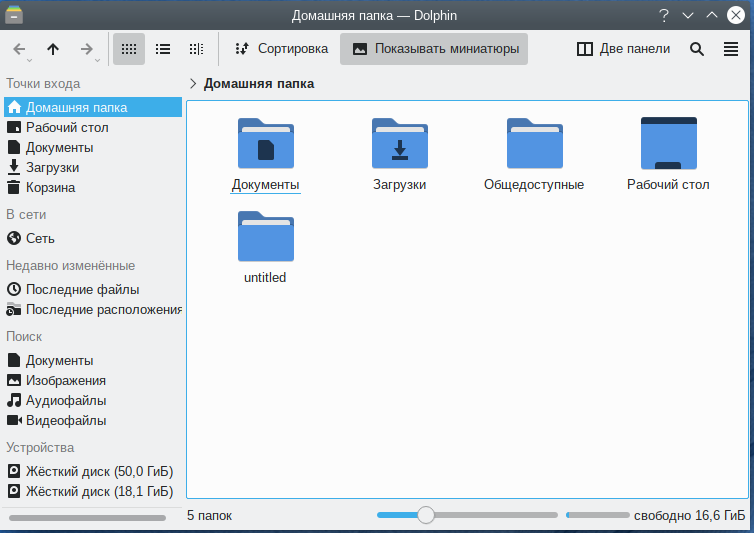
Изображение выглядит как текст

Автоматически созданное описание



Изображение выглядит как текст

Автоматически созданное описание



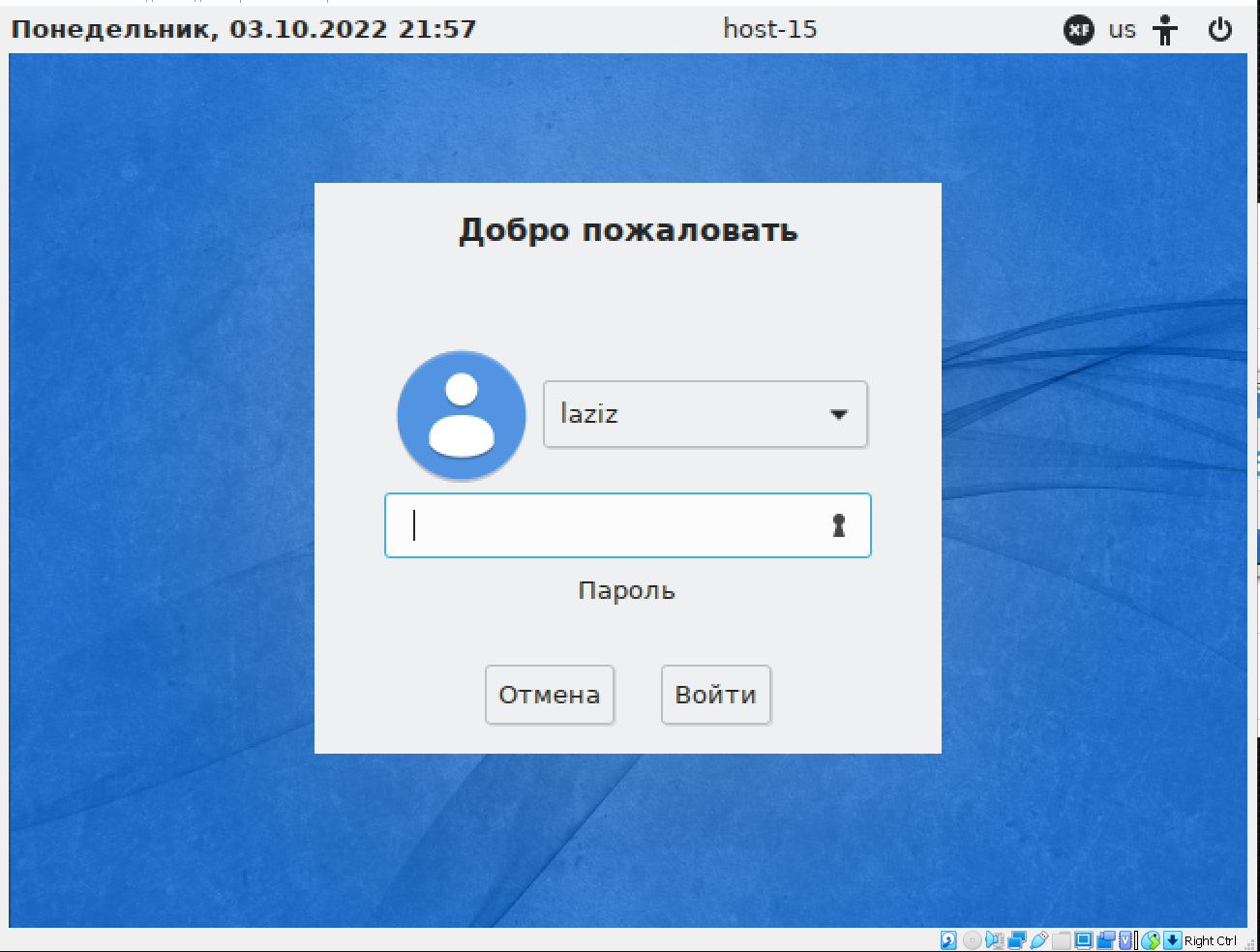
Мне больше понравилась оболочка XFCE, так как ее интерфейс более привычен.

5. Теперь воспользуемся режимом спец. возможностей и установим различный шрифт и разную контрастность. Делается это при запуске системы в правом верхнем углу.

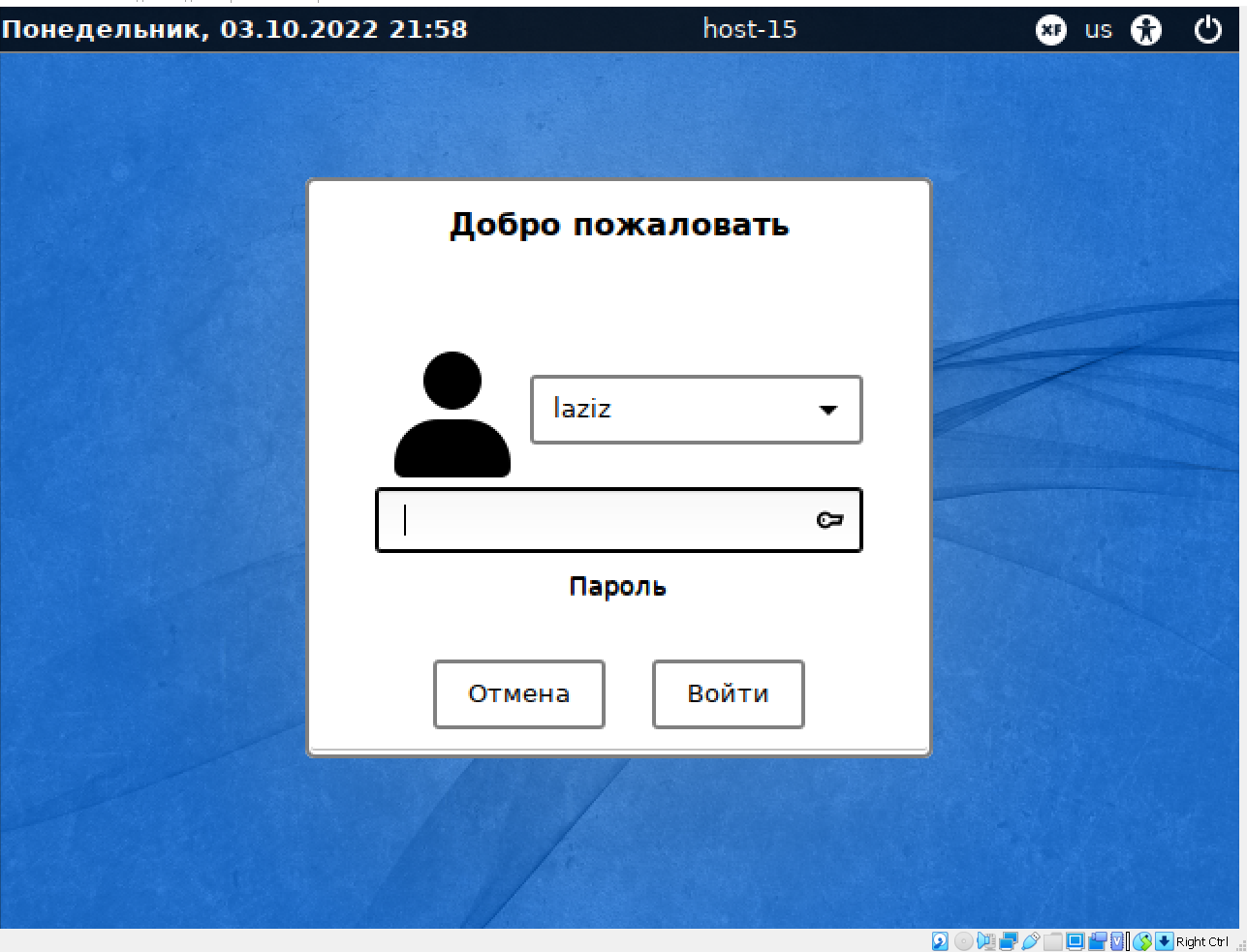
Изображение выглядит как текст

Автоматически созданное описание

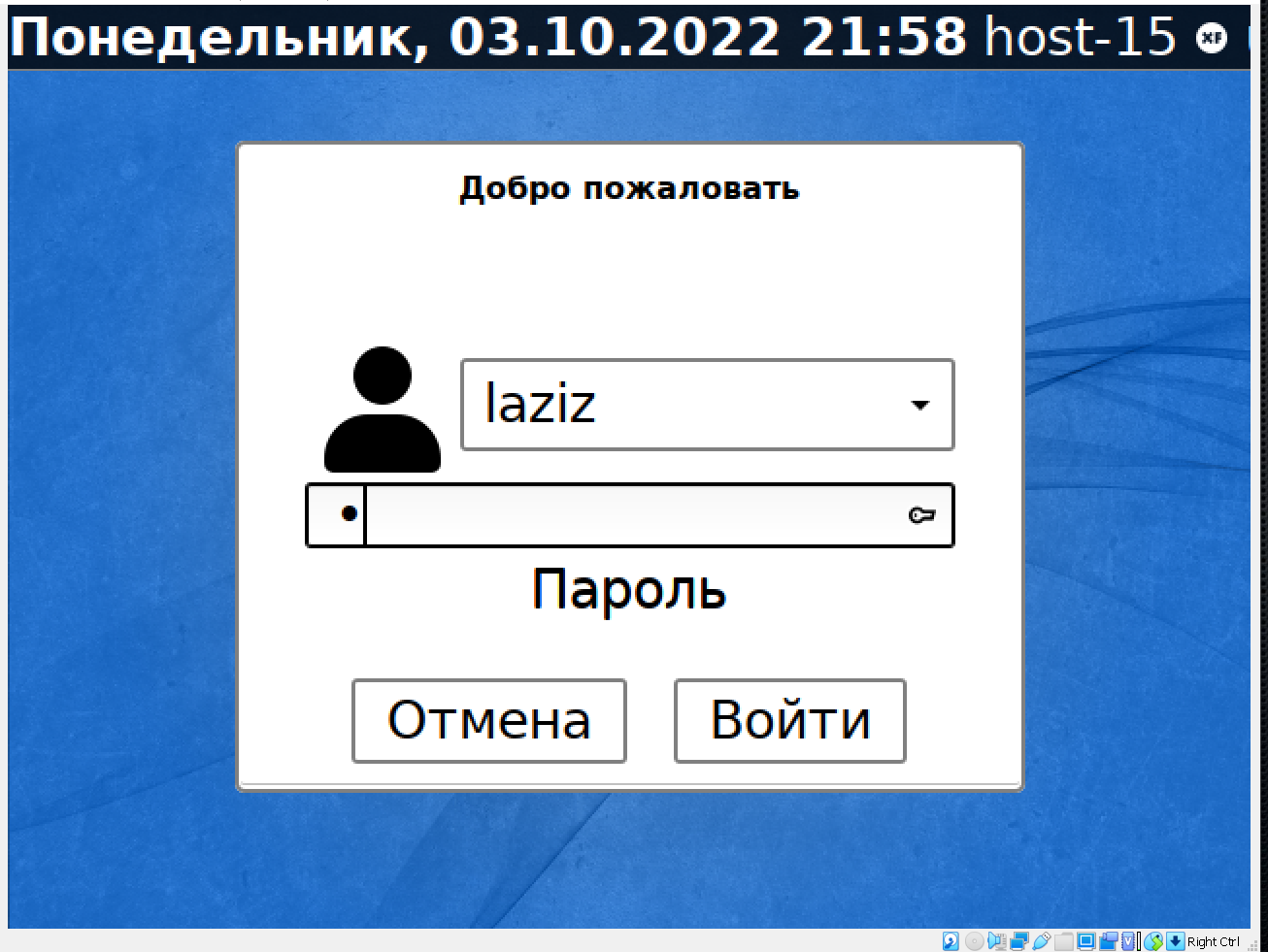
Крупный шрифт:



Высокая контрастность:



Высокая контрастность и крупный шрифт:



6. Теперь откроем центр управления системой для редактирования пользователей. Для этого в меню слева снизу выберем соответствующую программу.

Изображение выглядит как текст

Автоматически созданное описание

А дальше выберем пункт «Пользователи» - «Локальные учетные записи».

Изображение выглядит как текст

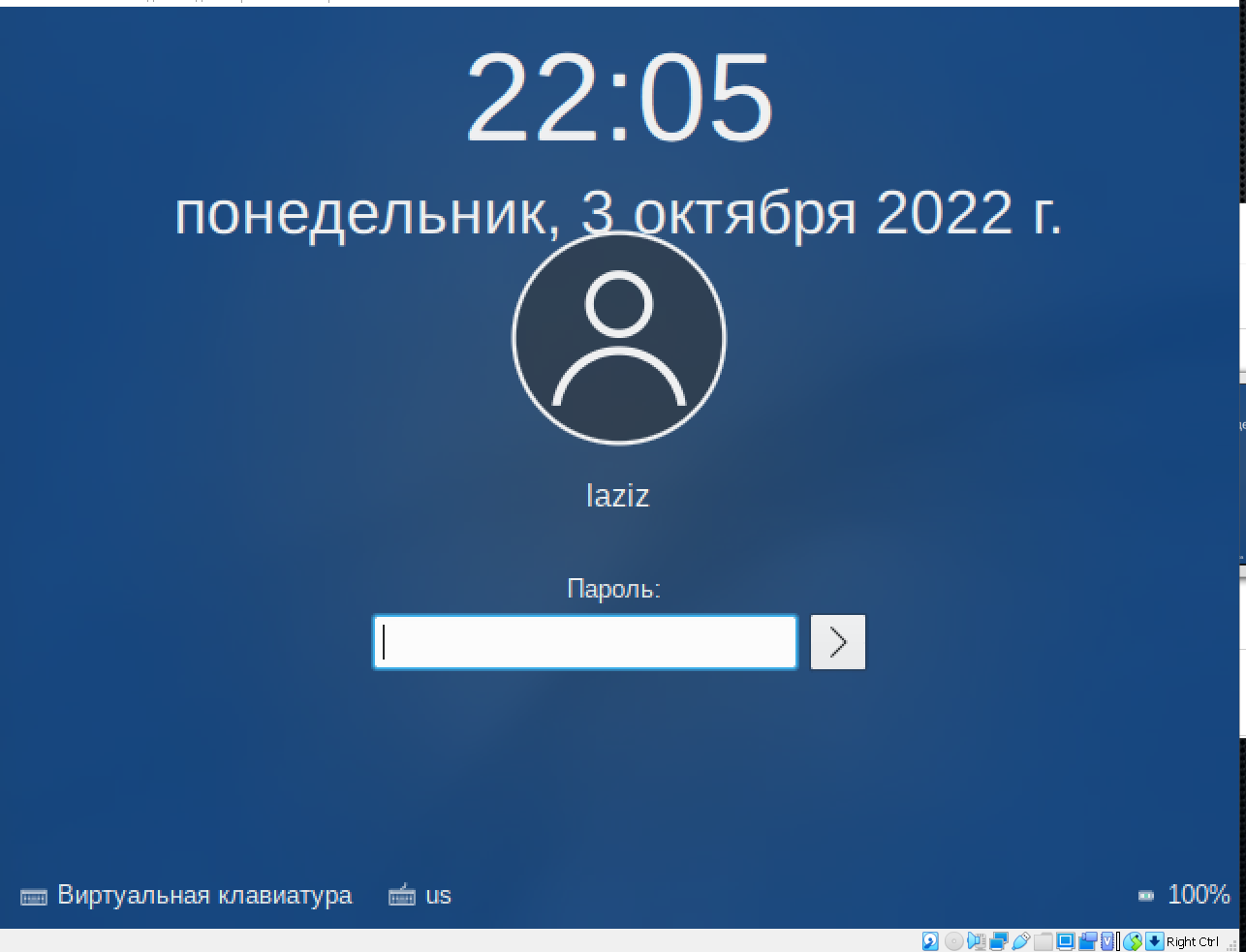
Автоматически созданное описание

Добавим новую учетную запись newuser и заполним необходимые поле с комментарием, жмём «Применить».

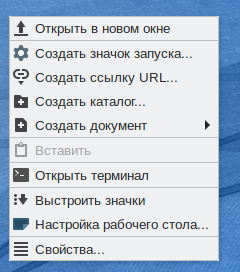
Изображение выглядит как текст

Автоматически созданное описание

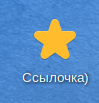
7. Заблокируем экран и попробуем его разблокировать.



Теперь создадим ярлык с помощью контекстного меню. Выбираем «Создать ссылку URL».



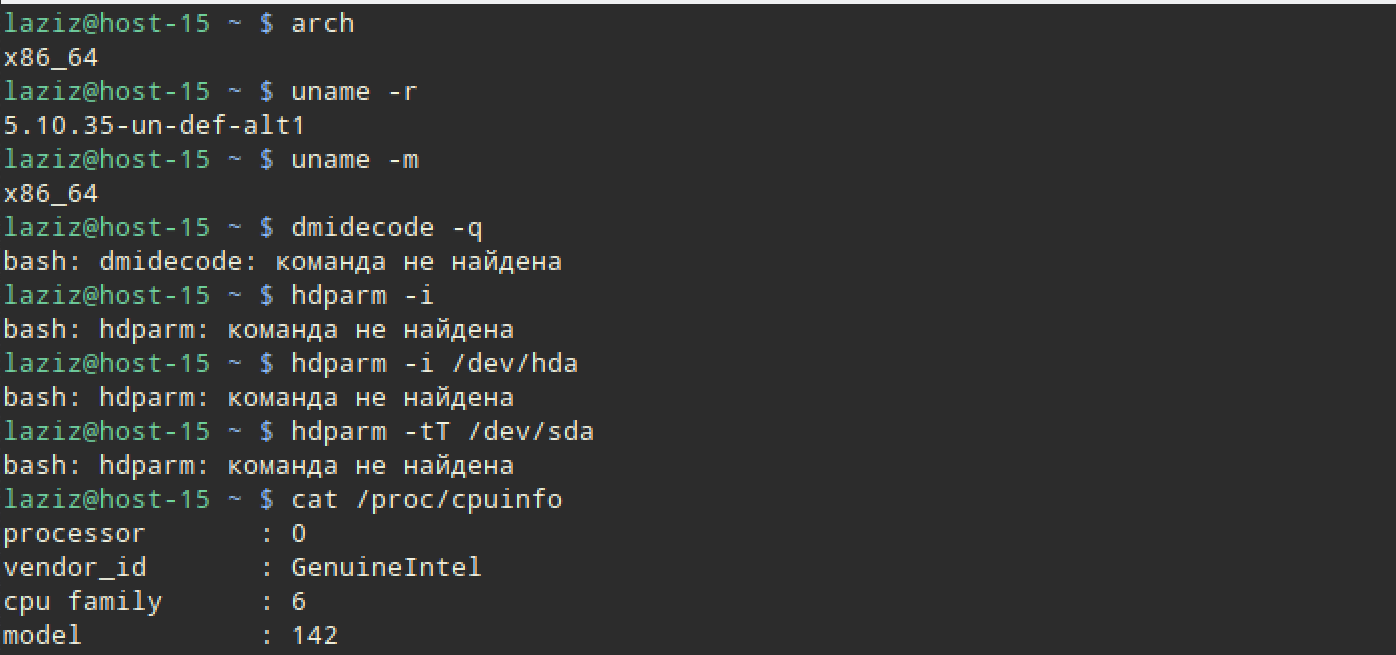
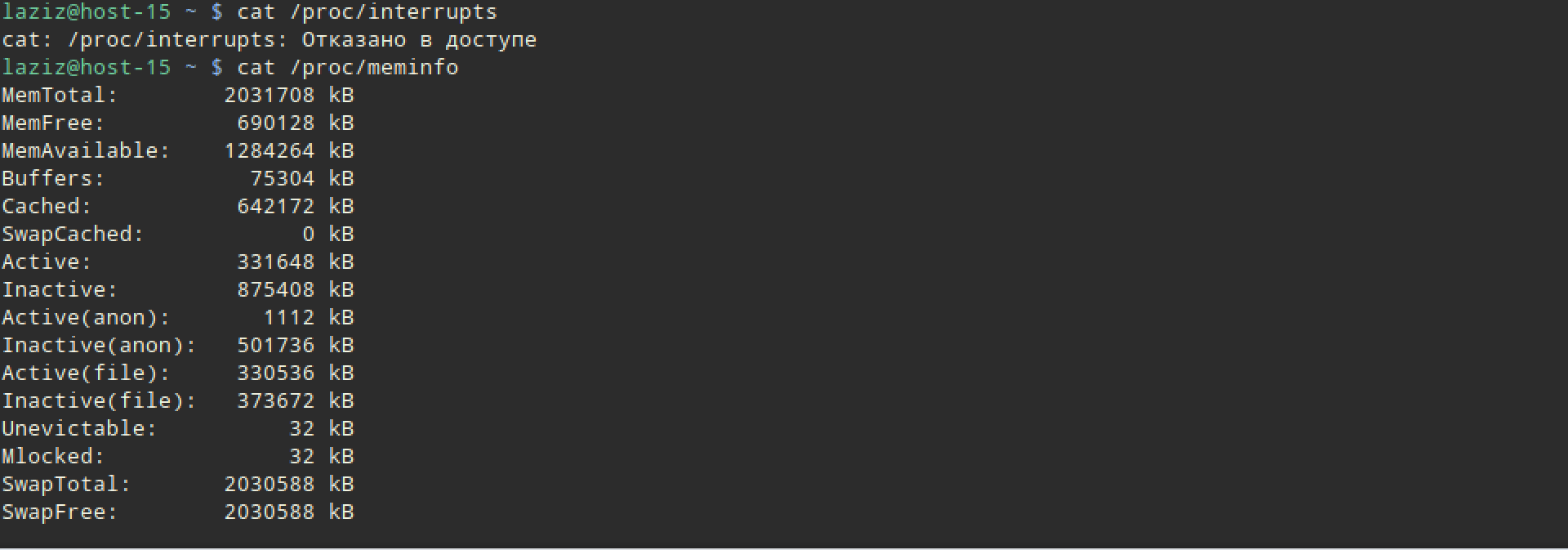
Изображение выглядит как текст

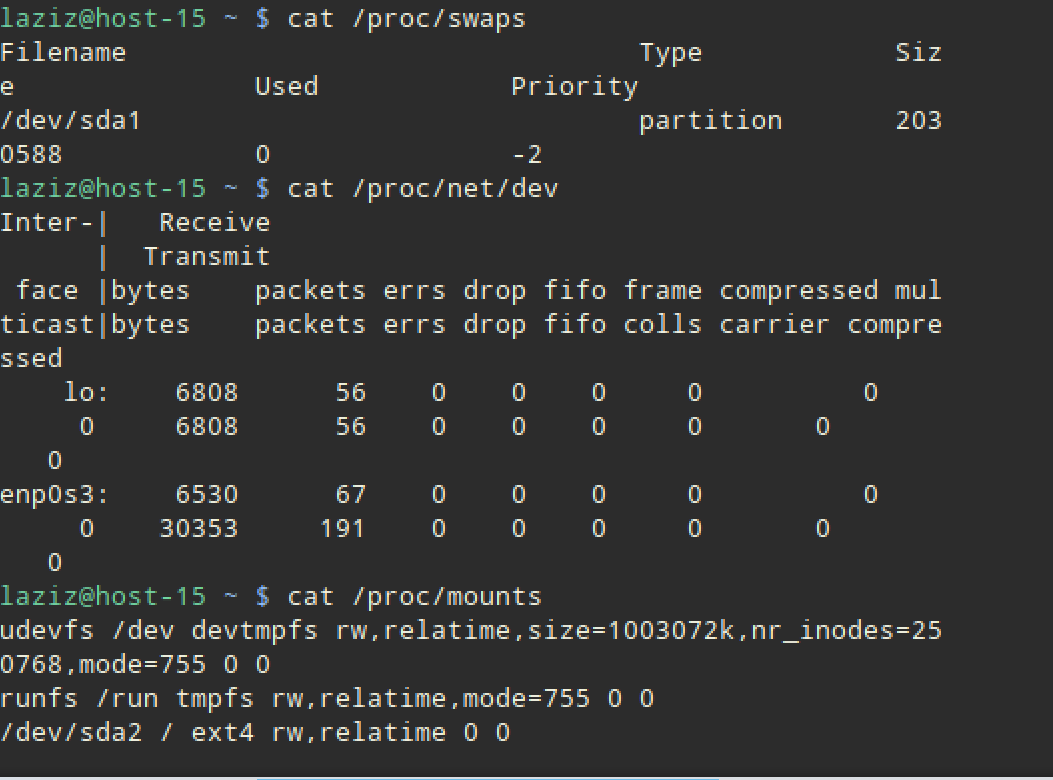
Автоматически созданное описание

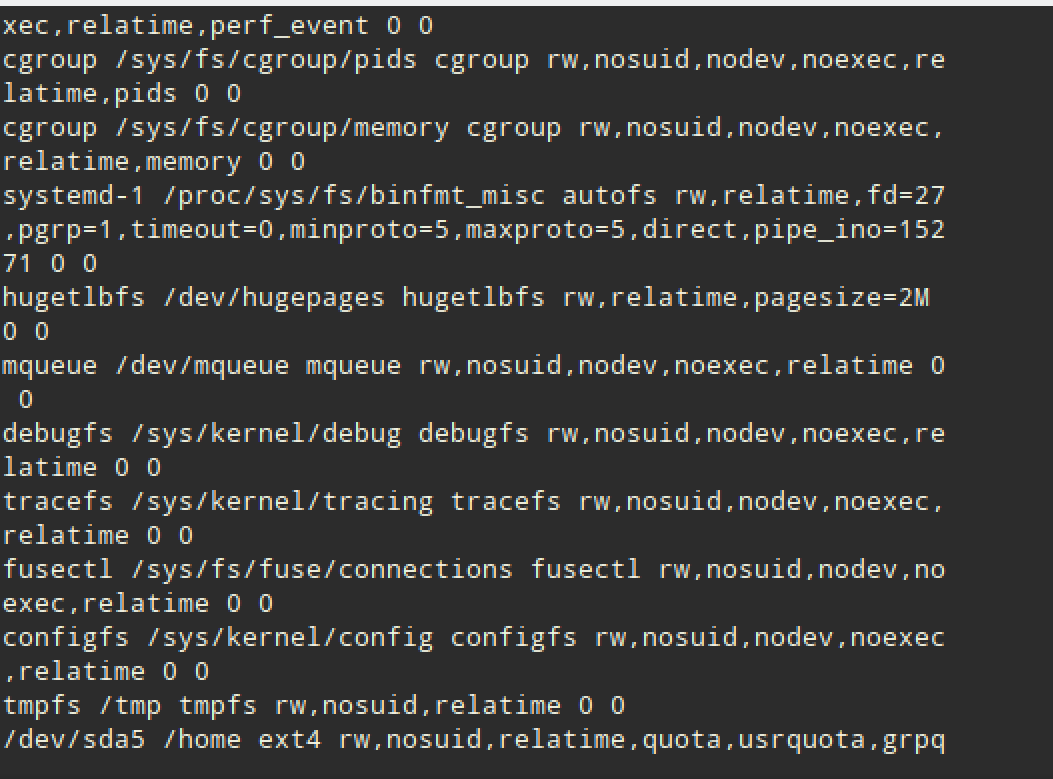
Познакомимся поближе с терминалом и базовыми командами.

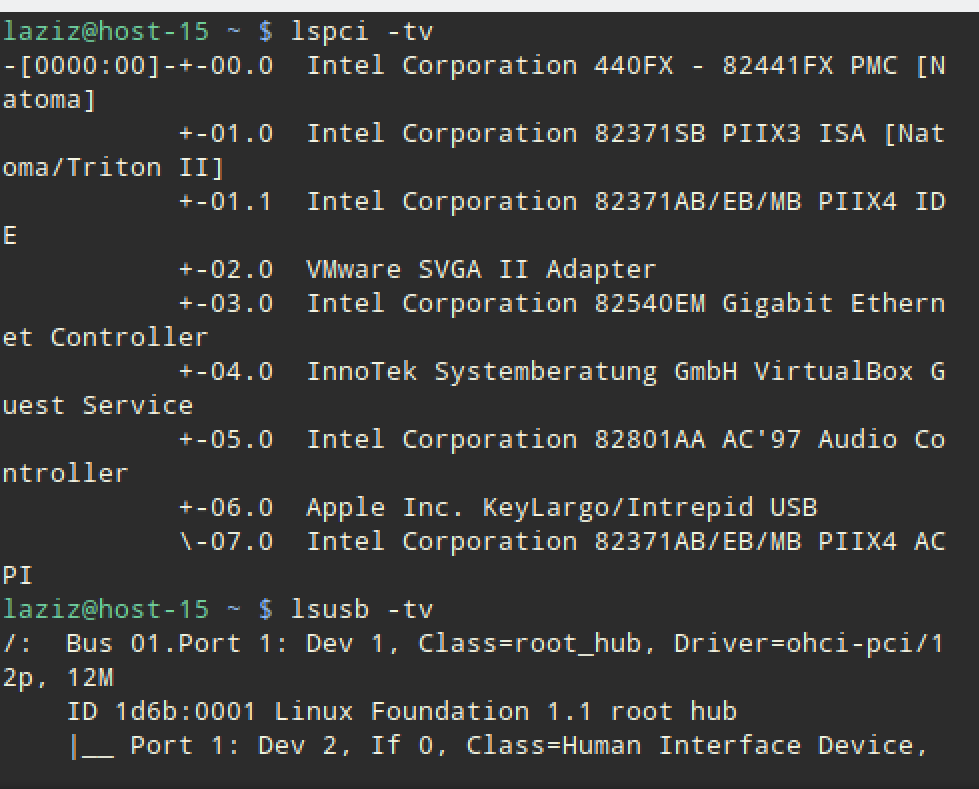
Те команды, которые не работают в нашем дистрибутиве, выделим красным.

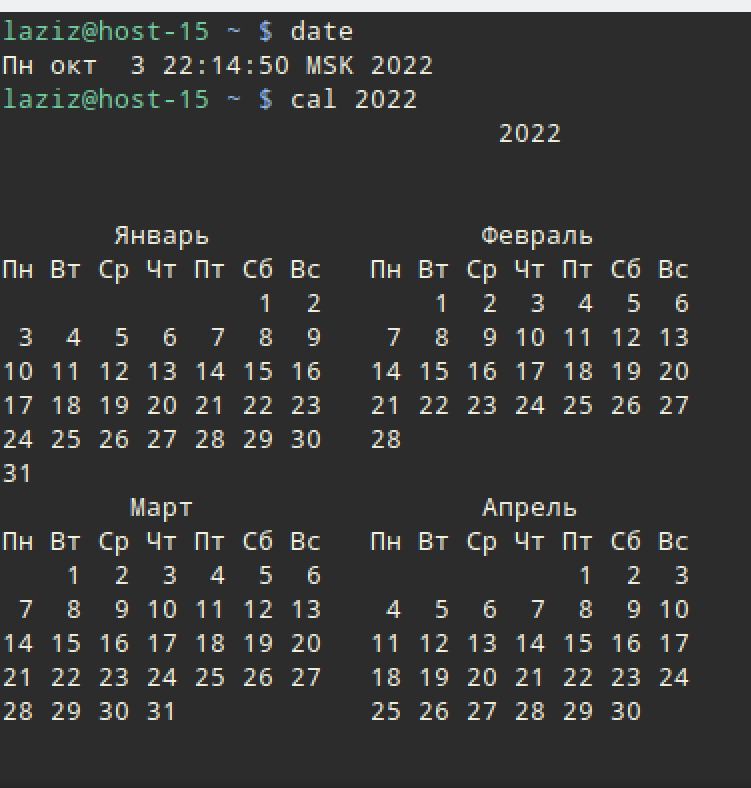
|  |  |
| --- | --- |
| **1. Системная информация** | |
| arch | отобразить архитектуру компьютера |
| uname -m |
| uname -r | отобразить используемую версию ядра |
| dmidecode -q | показать аппаратные системные компоненты - (SMBIOS / DMI) |
| hdparm -i /dev/hda | вывести характеристики жесткого диска |
| hdparm -tT /dev/sda | протестировать производительность чтения данных с жесткого диска |
| cat /proc/cpuinfo | отобразить информацию о процессоре |
| cat /proc/interrupts | показать прерывания |
| cat /proc/meminfo | проверить использование памяти |
| cat /proc/swaps | показать файл(ы) подкачки |
| cat /proc/version | вывести версию ядра |
| cat /proc/net/dev | показать сетевые интерфейсы и статистику по ним |
| cat /proc/mounts | отобразить смонтированные файловые системы |
| lspci -tv | показать в виде дерева PCI устройства |
| lsusb -tv | показать в виде дерева USB устройства |
| date | вывести системную дату |
| cal 2021 | вывести таблицу-календарь 2021-го года |
| clock -w | сохранить системное время в BIOS |



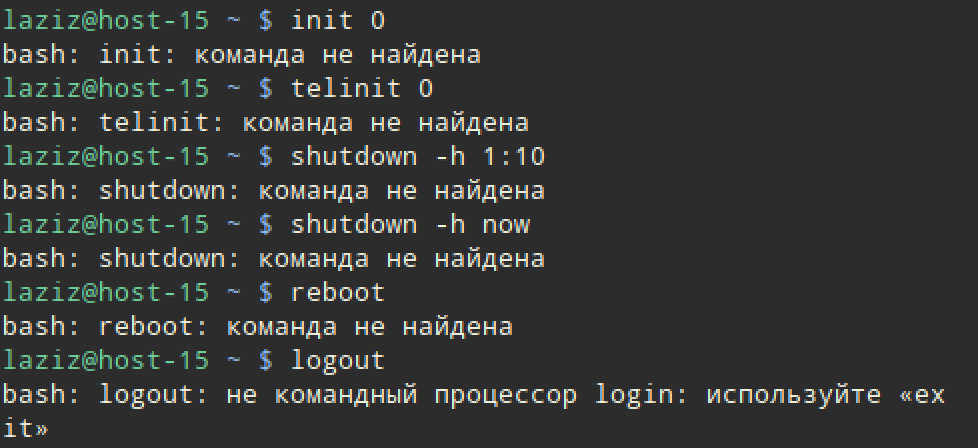




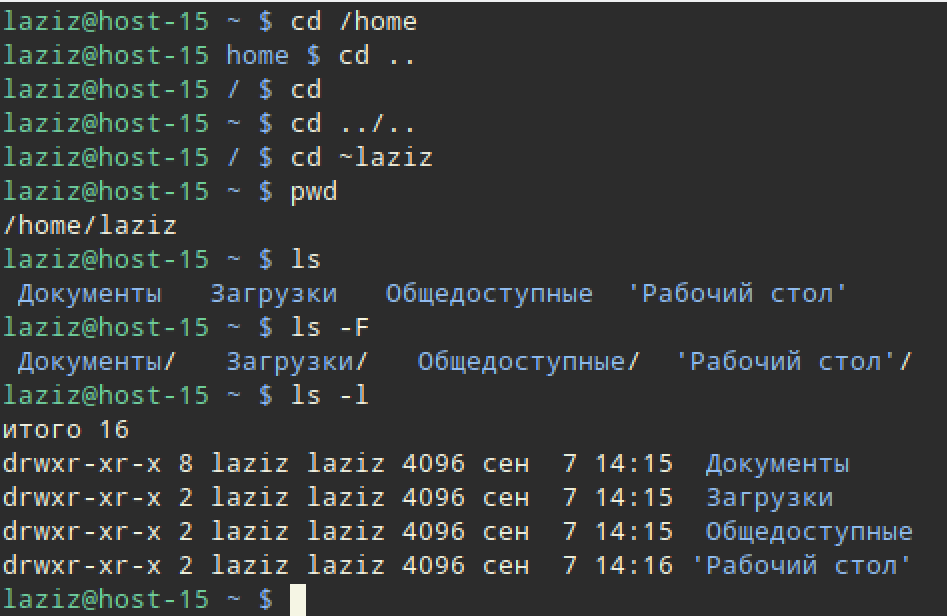


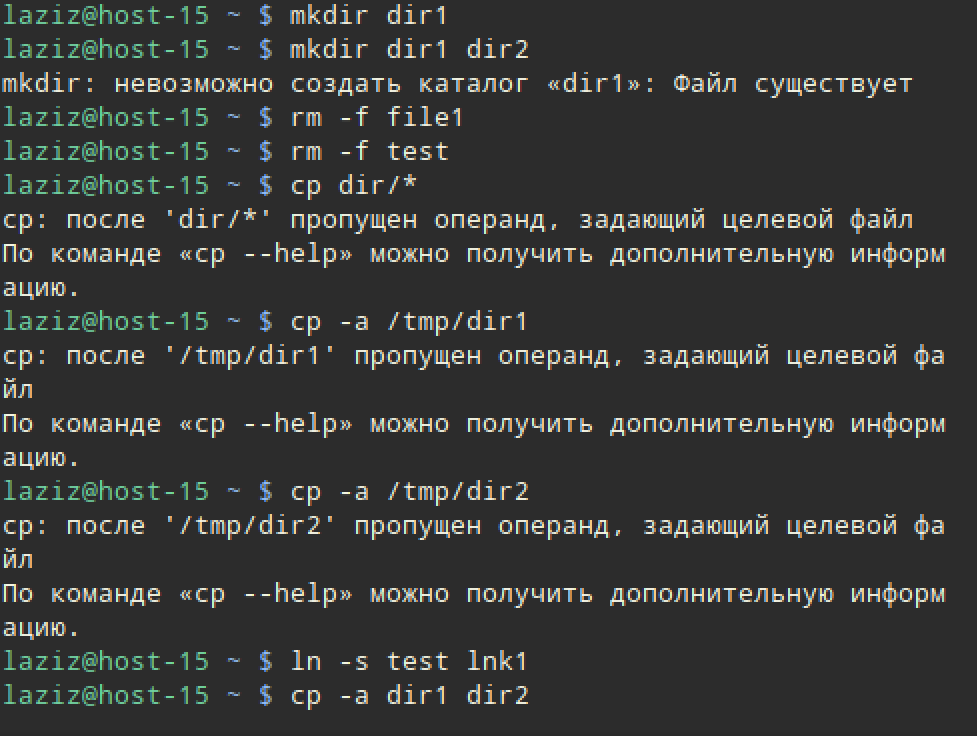


|  |  |
| --- | --- |
| **2. Остановка системы** | |
| shutdown -h now | Остановить систему |
| init 0 |
| telinit 0 |
| shutdown -h hours:minutes & | запланировать остановку системы на указанное время |
| shutdown -r now | перегрузить систему |
| reboot |
| logout | выйти из системы |

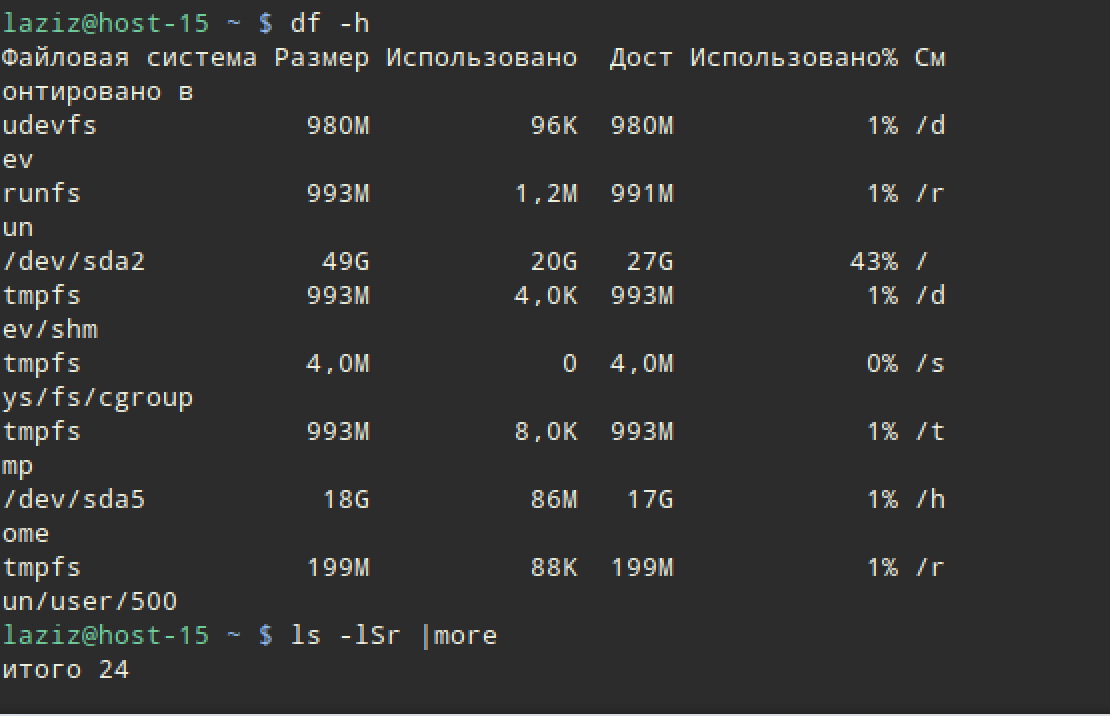


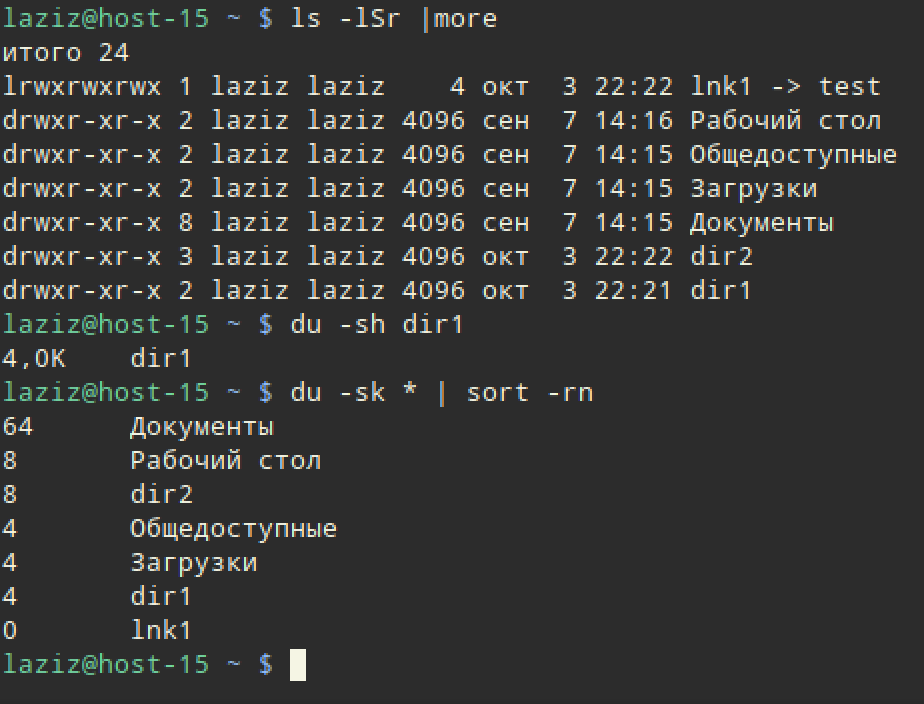
|  |  |
| --- | --- |
| **3. Файлы и директории** | |
| cd /home | перейти в директорию '/home' |
| cd .. | перейти в директорию уровнем выше |
| cd ../.. | перейти в директорию двумя уровнями выше |
| cd | перейти в домашнюю директорию |
| cd ~user | перейти в домашнюю директорию пользователя user |
| cd - | перейти в директорию, в которой находились до перехода в текущую директорию |
| pwd | показать текущую директорию |
| ls | отобразить содержимое текущей директории |
| ls -F | отобразить содержимое текущей директории с добавлением к именам символов, храктеризующих тип |
| ls -l | показать детализированое представление файлов и директорий в текущей директории |
| ls -a | показать скрытые файлы и директории в текущей директории |
| ls \*[0-9]\* | показать файлы и директории содержащие в имени цифры |
| tree | показать дерево файлов и директорий, начиная от корня (/) |
| lstree |
| mkdir dir1 | создать директорию с именем 'dir1' |
| mkdir dir1 dir2 | создать две директории одновременно |
| mkdir -p /tmp/dir1/dir2 | создать дерево директорий |
| rm -f file1 | удалить файл с именем 'file1' |
| rmdir dir1 | удалить директорию с именем 'dir1' |
| rm -rf dir1 | удалить директорию с именем 'dir1' и рекурсивно всё её содержимое |
| rm -rf dir1 dir2 | удалить две директории и рекурсивно их содержимое |
| mv dir1 new\_dir | переименовать или переместить файл или директорию |
| cp file1 file2 | сопировать файл file1 в файл file2 |
| cp dir/\* . | копировать все файлы директории dir в текущую директорию |
| cp -a /tmp/dir1 . | копировать директорию dir1 со всем содержимым в текущую директорию |
| cp -a dir1 dir2 | копировать директорию dir1 в директорию dir2 |
| ln -s file1 lnk1 | создать символическую ссылку на файл или директорию |
| ln file1 lnk1 | создать "жёсткую" (физическую) ссылку на файл или директорию |
| touch -t 2012250000 fileditest | модифицировать дату и время создания файла, при его отсутствии, создать файл с указанными датой и временем (YYMMDDhhmm) |



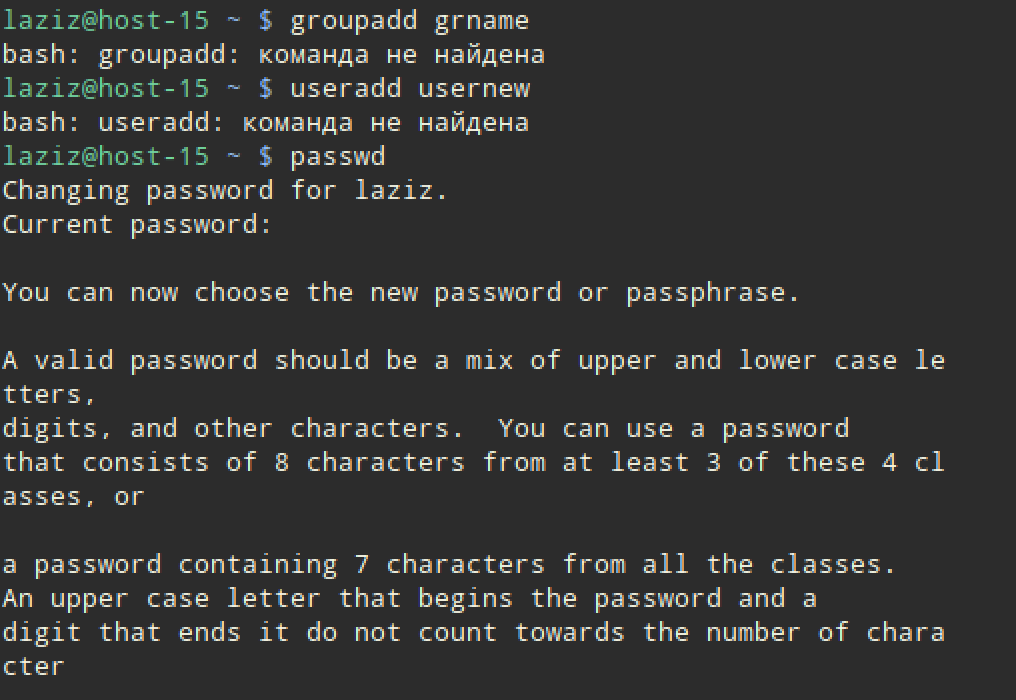


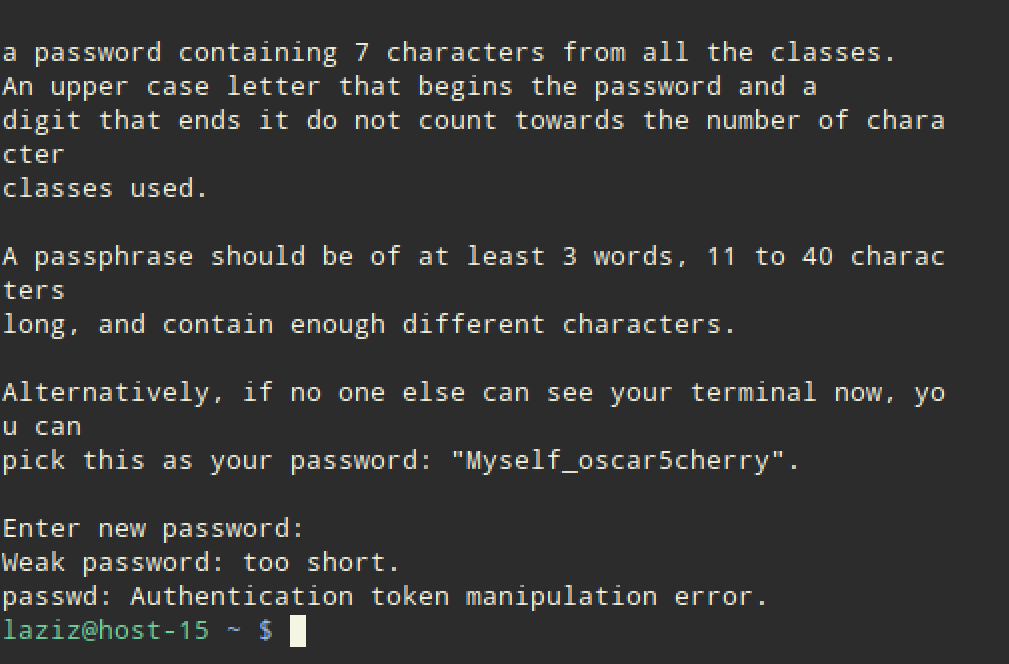
|  |  |
| --- | --- |
| **4. Дисковое пространство** | |
| df -h | отображает информацию о смонтированных разделах с отображением общего, доступного и используемого пространства *(Прим.переводчика. ключ -h работает не во всех \*nix системах)* |
| ls -lSr |more | выдаёт список файлов и директорий рекурсивно с сортировкой по возрастанию размера и позволяет осуществлять постраничный просмотр |
| du -sh dir1 | подсчитывает и выводит размер, занимаемый директорией 'dir1' *(Прим.переводчика. ключ -h работает не во всех \*nix системах)* |
| du -sk \* | sort -rn | отображает размер и имена файлов и директорий, с соритровкой по размеру |
| rpm -q -a --qf '%10{SIZE}\t%{NAME}\n' | sort -k1,1n | показывает размер используемого дискового пространства, занимаемое файлами rpm-пакета, с сортировкой по размеру (fedora, redhat и т.п.) |
| dpkg-query -W -f='${Installed-Size;10}\t${Package}\n' | sort -k1,1n | показывает размер используемого дискового пространства, занимаемое файлами deb-пакета, с сортировкой по размеру (ubuntu, debian т.п.) |



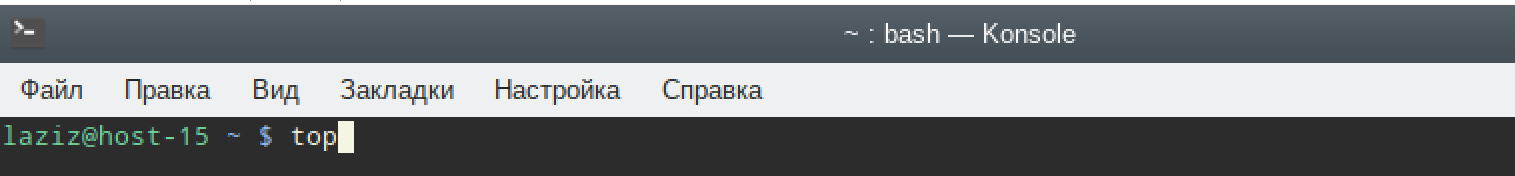


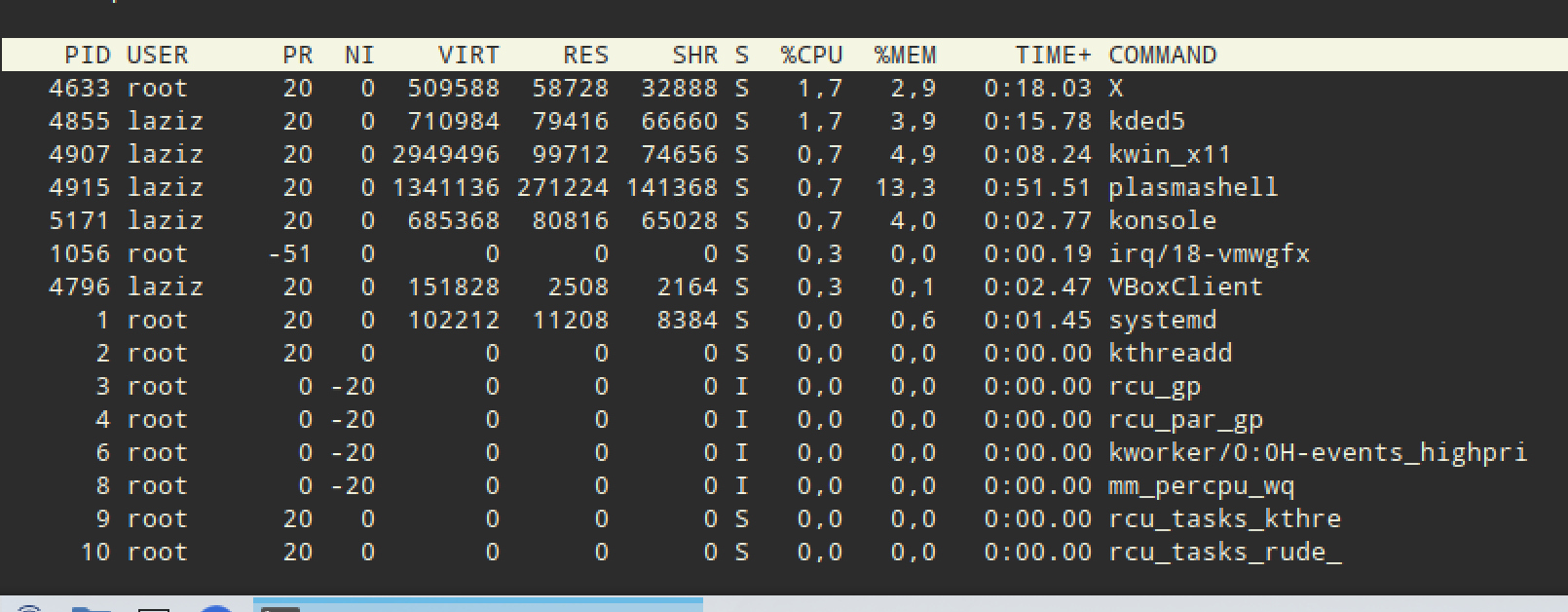
|  |  |
| --- | --- |
| **5. Пользователи и группы** | |
| groupadd group\_name | создать новую группу с именем group\_name |
| groupdel group\_name | удалить группу group\_name |
| groupmod -n new\_group\_name old\_group\_name | переименовать группу old\_group\_name в new\_group\_name |
| useradd -c "Nome Cognome" -g admin -d /home/user1 -s /bin/bash user1 | создать пользователя user1, назначить ему в качестве домашнего каталога /home/user1, в качестве shell'а /bin/bash, включить его в группу admin и добавить комментарий Nome Cognome |
| useradd user1 | создать пользователя user1 |
| userdel -r user1 | удалить пользователя user1 и его домашний каталог |
| usermod -c "User FTP" -g system -d /ftp/user1 -s /bin/nologin user1 | изменить атрибуты пользователя |
| passwd | сменить пароль |
| passwd user1 | сменить пароль пользователя user1 (только root) |
| chage -E 2005-12-31 user1 | установить дату окончания действия учётной записи пользователя user1 |
| pwck | проверить корректность системных файлов учётных записей. Проверяются файлы /etc/passwd и /etc/shadow |
| grpck | проверяет корректность системных файлов учётных записей. Проверяется файл/etc/group |
| newgrp [-] group\_name | изменяет первичную группу текущего пользователя. Если указать "-", ситуация будет идентичной той, в которой пользователь вышил из системы и снова вошёл. Если не указывать группу, первичная группа будет назначена из /etc/passwd |

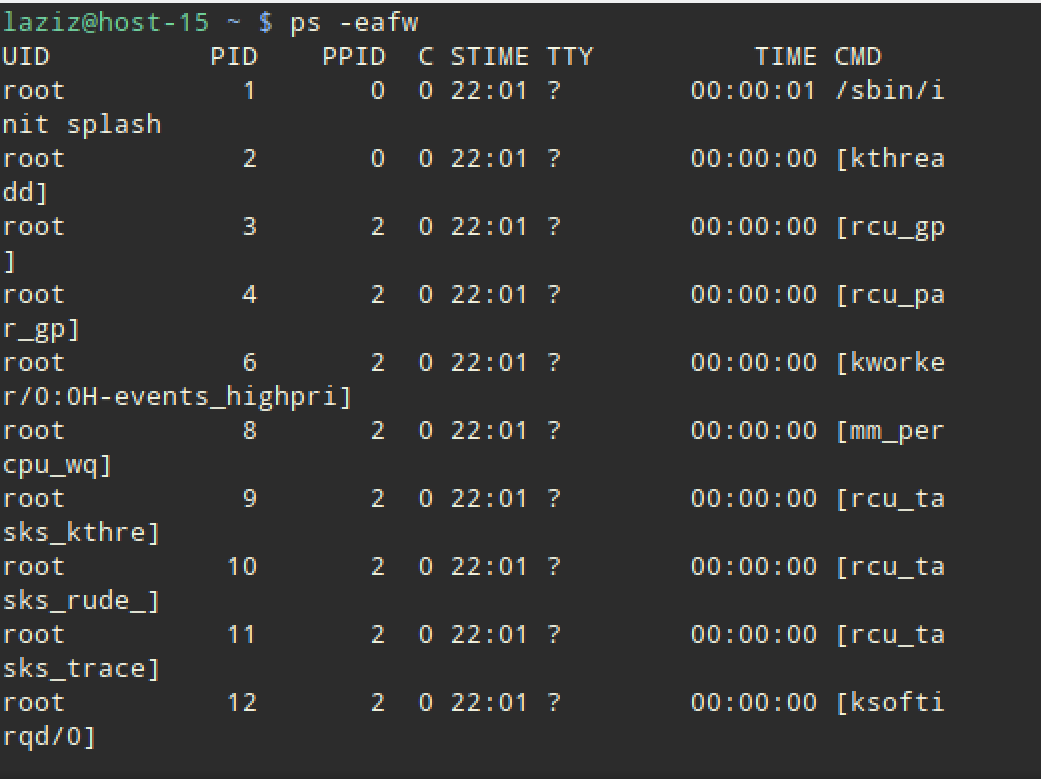


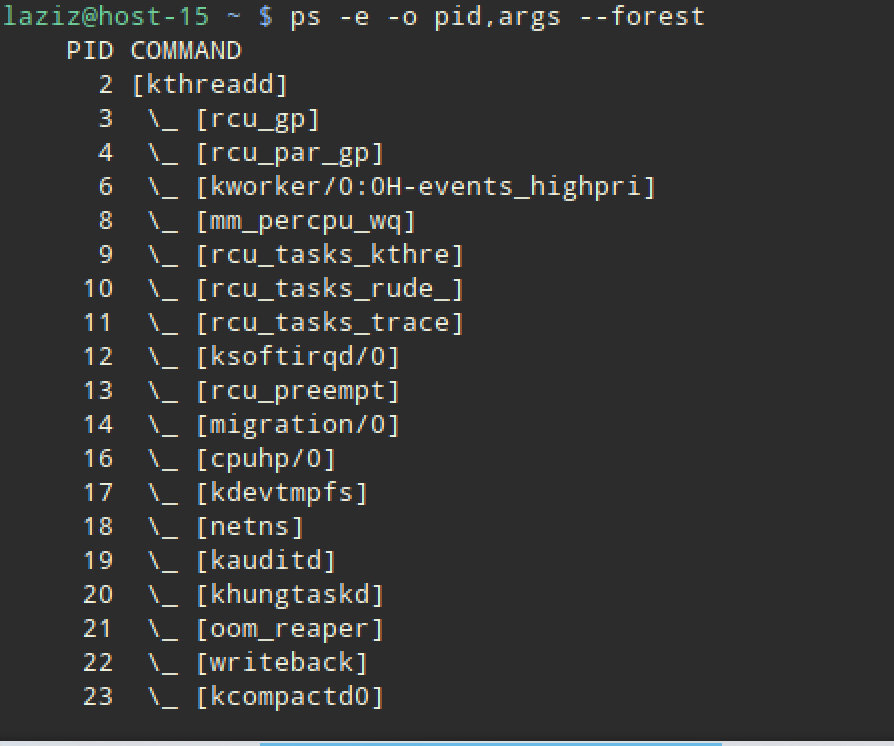


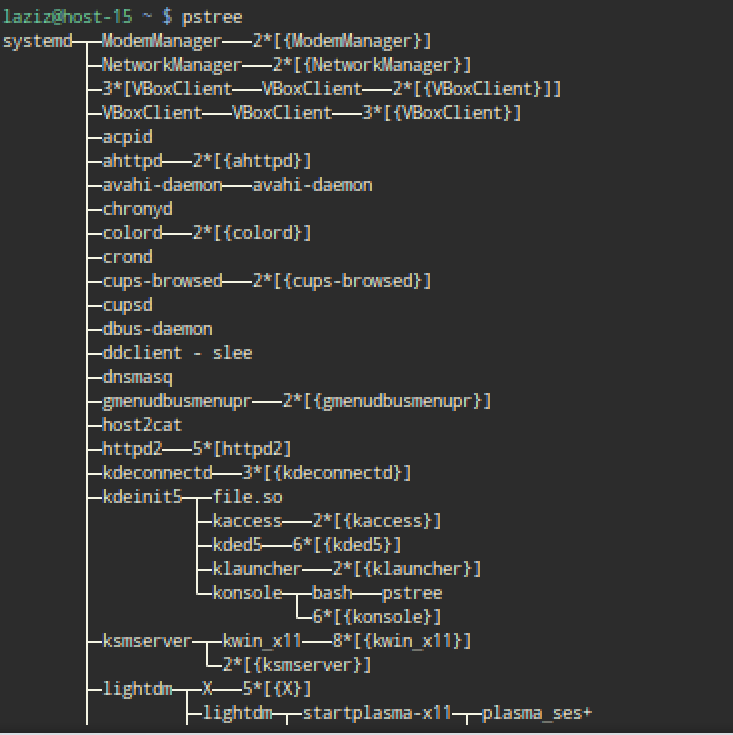
|  |  |
| --- | --- |
| **6. Мониторинг и отладка** | |
| top | отобразить запущенные процессы, используемые ими ресурсы и другую полезную информацию (с автоматическим обновлением данных) |
| ps -eafw | отобразить запущенные процессы, используемые ими ресурсы и другую полезную информацию (единожды) |
| ps -e -o pid,args --forest | вывести PID'ы и процессы в виде дерева |
| pstree | отобразить дерево процессов |
| kill -9 98989 | "убить" процесс с PID 98989 "на смерть" (без соблюдения целостности данных) |
| kill -KILL 98989 |
| kill -TERM 98989 | Корректно завершить процесс с PID 98989 |
| kill -1 98989 | заставить процесс с PID 98989 перепрочитать файл конфигурации |
| kill -HUP 98989 |
| lsof -p 98989 | отобразить список файлов, открытых процессом с PID 98989 |
| lsof /home/user1 | отобразить список открытых файлов из директории /home/user1 |
| lsof -iTCP:59302 | показать приложение, которое использует TCP-порт 59302 (не обязательно слушает) |
| strace -c ls > /dev/null | вывести список системных вызовов, созданных и полученных процессом ls |
| strace -f -e open ls > /dev/null | вывести вызовы бибилотек |
| watch -n1 'cat /proc/interrupts' | отображать прерывания в режиме реального времени |
| last reboot | отобразить историю перезагрузок системы |
| last user1 | отобразить историю регистрации пользователя user1 в системе и время его нахождения в ней |
| lsmod | вывести загруженные модули ядра |
| free -m | показать состояние оперативной памяти в мегабайтах |
| smartctl -A /dev/hda | контроль состояния жёсткого диска /dev/hda через SMART |
| smartctl -i /dev/hda | проверить доступность SMART на жёстком диске /dev/hda |
| tail /var/log/dmesg | вывести десять последних записей из журнала загрузки ядра |
| tail /var/log/messages | вывести десять последних записей из системного журнала |

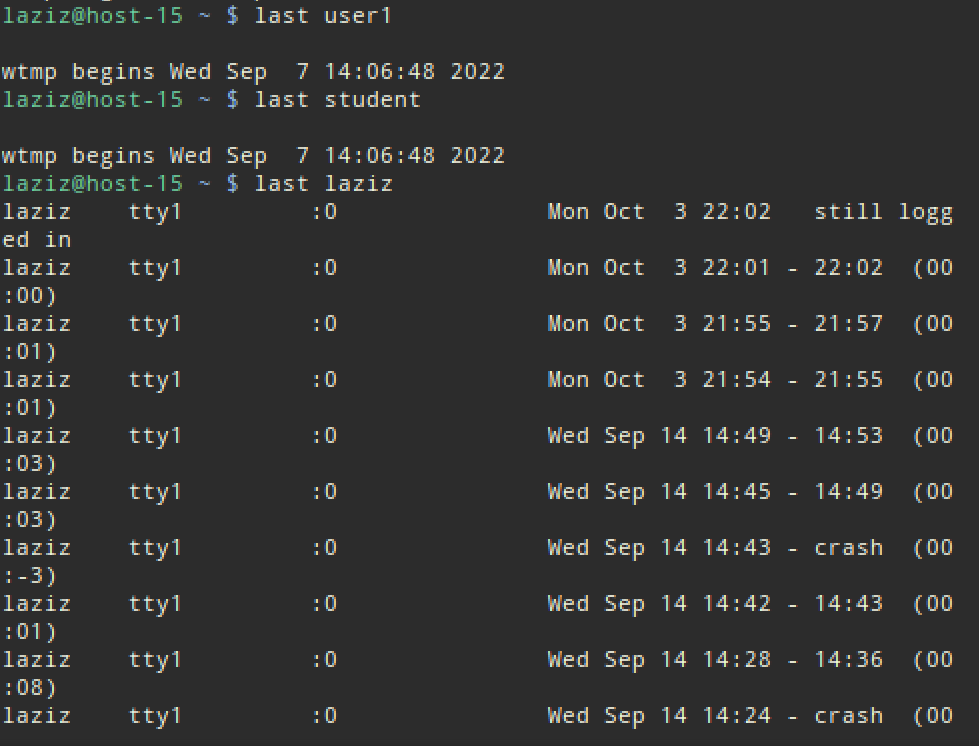


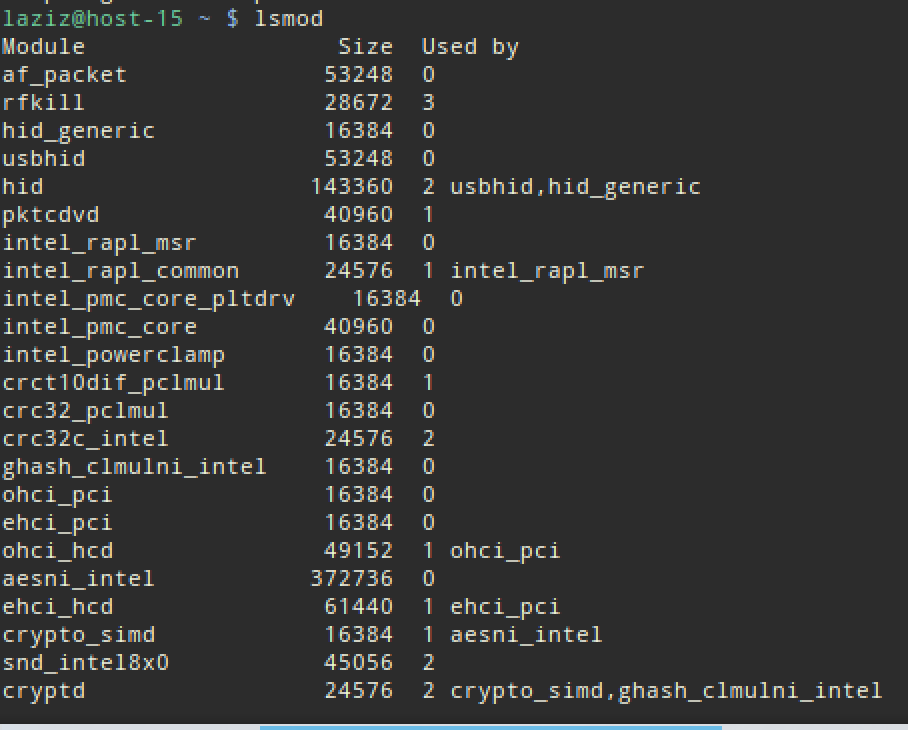




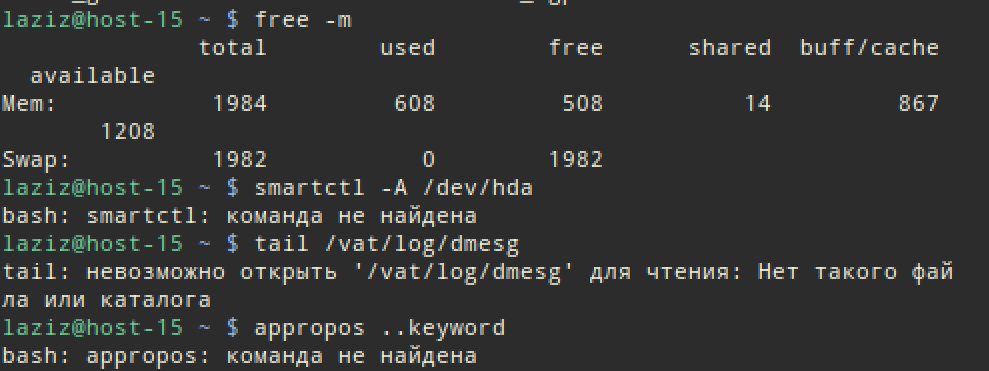






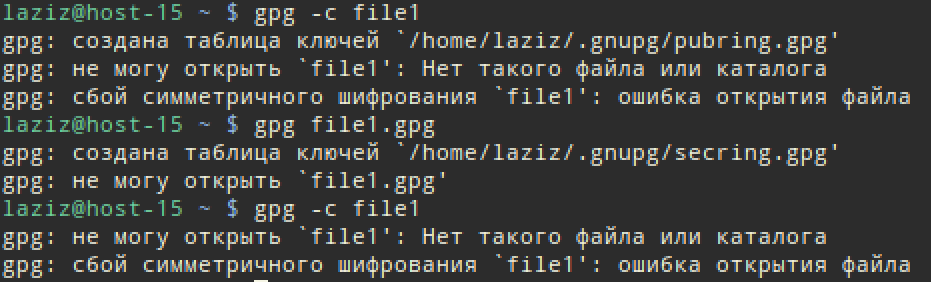


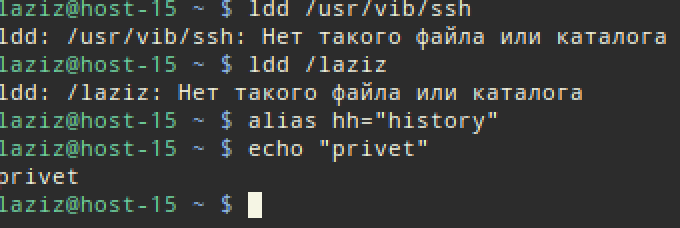
|  |  |
| --- | --- |
| **7. Дополнительные команды** | |
| apropos *…keyword* | выводит список комманд, которые так или иначе относятся к ключевым словам. Полезно, когда вы знаете что делает программа, но не помните команду |
| man ping | вызов руководства по работе с программой, в данном случае, - ping |
| whatis *…keyword* | отображает описание действий указанной программы |
| gpg -c file1 | шифрует файл file1 с помощью GNU Privacy Guard |
| gpg file1.gpg | дешифрует файл file1 с помощью GNU Privacy Guard |
| wget -r www.example.com | загружает рекурсивно содержимое сайта www.example.com |
| wget -c www.example.com/file.iso | загрузить файл www.example.com/file.iso с возможностью останова и продолжения в последствии |
| echo 'wget -c www.example.com/files.iso' | at 09:00 | начать закачку в указанное время |
| ldd /usr/bin/ssh | вывести список библиотек, необходимых для работы ssh |
| alias hh='history' | назначить алиас hh команде history |



Изображение выглядит как текст

Автоматически созданное описание





**Аналитические вопросы**

1. *Сопоставьте возможности работы с объектно-ориентированной платформой и командной строкой, какие преимущества и недостатки имеет каждый вариант, как они сочетаются?*

При работе с командной строкой ничего не отвлекает от работы, нет никаких всплывающих окон, рекламы, предложений обновления и прочего. Всё пишется белыми буквами в чёрном терминале. Минимализм и ничего лишнего. Но иногда это может быть и минусом. Например, когда вы являетесь начинающим и только знакомитесь с функционалом командной строки или терминала. В таком случае лучше начать пользоваться объектно-ориентированными платформами, в которых есть подсказки и команды быстрого доступа к некоторым функциям.

Еще одним плюсом может служить то, что командная строка весит в разы, а то и на порядок меньше, чем программы с графическим интерфейсом, особенно если они очень продвинутые. Если необходимо писать код или выполнять команды на слабых по мощности устройствах, то рационально использовать именно командную строку. Если работа всё же выполняется на компьютерах, и места на диске достаточно, то можно без проблем установить объектно-ориентированную платформу.

Похожая ситуация и с оперативной памятью. На запуск платформы с графической платформы выделяется часть памяти, которая могла быть пущена на работу процессов, запущенных через эту платформу. Командная строка сама по себе практически не тратит оперативной памяти, а потому при запуске через неё процессов всё будет выполняться быстрее.

Хорошим вариантом будет сочетание платформы с командной строкой в зависимости от выполняемых задач. Если важная высокая скорость, то стоит использовать командную строку, если требуется не так много оперативной и постоянной памяти, то выбор делается в пользу объектно-ориентированной платформы.

1. *Опишите международные или национальные стандарты, ГОСТы или рекомендации, которые на ваш взгляд наиболее применимы к сопроводительной документации различных ИТ-решений, приведите примеры.*

Есть несколько межгосударственных ГОСТов, согласно которым происходит разработка и распространение программ. ГОСТ 19.102-77 – стадии разработки программ и программной документации. ГОСТ 19.105-78 – требования к оформлению программных документов, комплексов и систем независимо от их назначения и области применения. ГОСТ 19.105-78 содержит полный перечень документации, которая должна сопровождать законченный программный продукт. К примеру, согласно ГОСТу 19.105-78 вместе с выпуском программы должны выпускаться следующие документы:

1.1. Спецификация – состав программы и документации на нее.

1.2. Ведомость держателей подлинников – перечень предприятий, на которых хранятся подлинники программной документации.

1.3. Текст программы – запись текста программы с необходимыми комментариями.

1.4. Описание программы – сведения о логической и функциональной структуре программы.

1.5. Программа и методика испытаний – требования, подлежащие проверке при испытании программы, порядок и методы их контроля.

1.6. Техническое задание – назначение и область применения программы, технические и специальные требования, необходимые стадии и сроки разработки, виды испытаний.