Герман Александр, ФИТ-4

**Лабораторная работа 2**

«BashLinux»

## **Задание 01. Команды (утилиты) Linux**

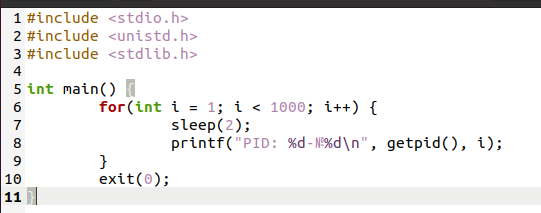
|  |  |
| --- | --- |
| Команда Linux | Краткое описание команды |
| **echo** | Вывод строки в терминал. |
| **ls** | Просмотр разрешения для файлов и папок. Если вы запустите команду без пути, то она выдаст содержимое текущего каталога. |
| **pwd** | Вывод в терминал путь к текущей папке. |
| **cd** | Смена текущую папку только для оболочки, в которой выполняется. |
| **mkdir** | Создание папок. |
| **rmdir** | Удаление папок. |
| **touch** | Устанавливает временные метки доступа и изменения каждого файла в текущее время. |
| **cp** | Копирует источник в назначение, или несколько источников в каталог. |
| **mv** | Переименовывает источник в назначение, или перемещает источник(и) в каталог. |
| **rm** | Удаляет (ссылки на) файл(ы). |
| **su** | Данная команда заменяет пользователя оболочки shell на указанного. Фактически происходит запуск нового экземпляра оболочки с указанными параметрами. Благодаря этому возможно, не выходя из системы, совершенно безболезненно повышать возможности управления операционной системой или наоборот ограничивать их. |
| **whoami** | Напечатать имя пользователя, соответствующее текущему эффективному id пользователя. |
| **man** | Интерфейс, используемый для просмотра справочных руководств системы. От слова manual. |
| **whereis** | Место расположения бинарных или исходных файлов на компьютере. |
| **whatis** | Команда whatis производит контекстный поиск заданного ключевого слова (шаблона) в базе данных, содержащей перечень системных команд с кратким их описанием. Выводятся только точные совпадения с ключевым словом. |
| **apropos** | Команда apropos осуществляет поиск переданной пользователем строки в заголовках страниц руководств. Является эквивалентом команды man -k. |
| **cat** | Являясь сокращением английского слова «concatenate» (конкатенация), она позволяет создавать, объединять, а также выводить содержимое файлов в командной строке или в другом файле |
| **less** | Команда less в Linux пригодится для просмотра больших текстовых файлов, с которыми затруднительно работать в текстовых редакторах вроде vim или с помощью утилит, загружающих весь документ сразу. |
| **head** | Печатает первые 10 строк каждого файла на стандартный вывод. |
| **tail** | Печатает последние 10 строк каждого файла на стандартный вывод. |
| **lpstat** | Печатает информацию о текущем состоянии имеющихся в системе принтеров. |
| **lpr** | Печатает файл из терминала Linux. |
| **lpq** | Просмотр состояния очереди печати, программа может просматривать состояние очереди принтера и задание печати содержит. |
| **lprm** | Удаление задания из колонки для хранения принтера. |
| **chgrp** | Позволяет задействовать соответствующую утилиту для изменения группы пользователей, владеющих файлом или директорией. |
| **chown** | Смена владельца и группы указанного файла на владельца и/или группу. |
| **chmod** | Смена режима доступа к указанным файлам. |
| **zip** | Кроссплатформенная утилита для создания сжатых архивов формата zip. |
| **gzip** | Это стандартная утилита сжатия в Unix/Linux. |
| **gunzip** | Для декомпрессии используется gunzip. |
| **bzip2** | Еще одна альтернативная утилита сжатия для Linux. Она более эффективная чем gzip, но работает медленнее. |
| **bunzip2** | Для распаковки. |
| **tar** | Стандартная утилита, с помощью которой выполняется архивирование файлов Linux. |
| **locate** | Используется для поиска файлов, расположенных на машине пользователя или на сервере. |
| **grep** | Поиск шаблонов в каждом файле. |
| **find** | Команда для поиска файлов и каталогов на основе специальных условий. |
| **history** | История использования утилит. |
| **alias** | Определение или отображение псевдонимов. |
| **unalias** | Удаляет все имена из списка определенных псевдонимов. |
| **ps** | Имеет множество опций для настройки вывода тех или иных параметров процессов в Linux. |
| **top** | Позволяет выводить информацию о системе, а также список процессов динамически обновляя информацию о потребляемых ими ресурсах. |
| **lsof** | Список всех открытых файлов всеми процессами в системе. |
| **free** | Позволяет вызывать одноименную утилиту, осуществляющую вывод информации об использовании оперативной памяти. |
| **df** | Показывает сведения о файловой системе, на которой расположен каждый из файлов, или, по умолчанию, обо всех файловых системах. |
| **du** | Суммирует использование дискового пространства набора файлов, рекурсивно с каталогами. |
| **yum** | С помощью yum пользователь или системный администратор может легко искать, устанавливать, обновлять, удалять пакеты программного обеспечения в системах Fedora linux, RHEL, CentOS. |
| **ifconfig** | С помощью нее вы можете включать или выключать сетевые интерфейсы, настраивать их параметры, переключать режимы и многое другое. |
| **ping** | Позволяет проверить доступен удаленный хост или нет.  \ |
| **traceroute** | Команда traceroute используется в Linux для отображения пути прохождения пакета информации от его источника к месту назначения. |
| **host** | Утилита host предназначена для выполнения запросов к DNS-серверам. |
| **iwconfig** | Команда iwconfig в Linux похожа на команду ifconfig , в том смысле, что она работает с резидентным сетевым интерфейсом ядра, но предназначена только для беспроводных сетевых интерфейсов. |
| **dhclient** | Для управления адресом интерфейса по протоколу DHCP |
| **ifup** | Поднять сетевой интерфейс. |
| **ifdown** | Опустить сетевой интерфейс. |
| **route** | Посмотреть таблицу маршрутизации. |
| **ssh** | С помощью утилиты можно подключиться к серверу, а также передавать файлы, выполнять скрипты удаленным способом, управлять сервером без предварительного ввода пароля. |
| **sftp** | Используйте утилиту командной строки sftp для подключения к удаленной системе sftp. |
| **scp** | С помощью команды scp вы можете не только перемещать файлы между локальной и удаленной системой, но и между двумя удаленными системами. |
| **rsync** | Позволяет копировать и синхронизировать файлы с локальной машины на удалённую — и наоборот; поддерживает копирование ссылок, файлов устройств, данные о владельцах, группах и права доступа. Rsync работает быстрее, чем scp, использует меньше пропускной способности канала передачи данных, т.к. использует компрессию и декомпрессию данных. |
| **wget** | Утилита для скачивания файлов, веб страниц и других ресурсов из интернета |
| **curl** | Утилита командной строки для скачивания и загрузки данных. |

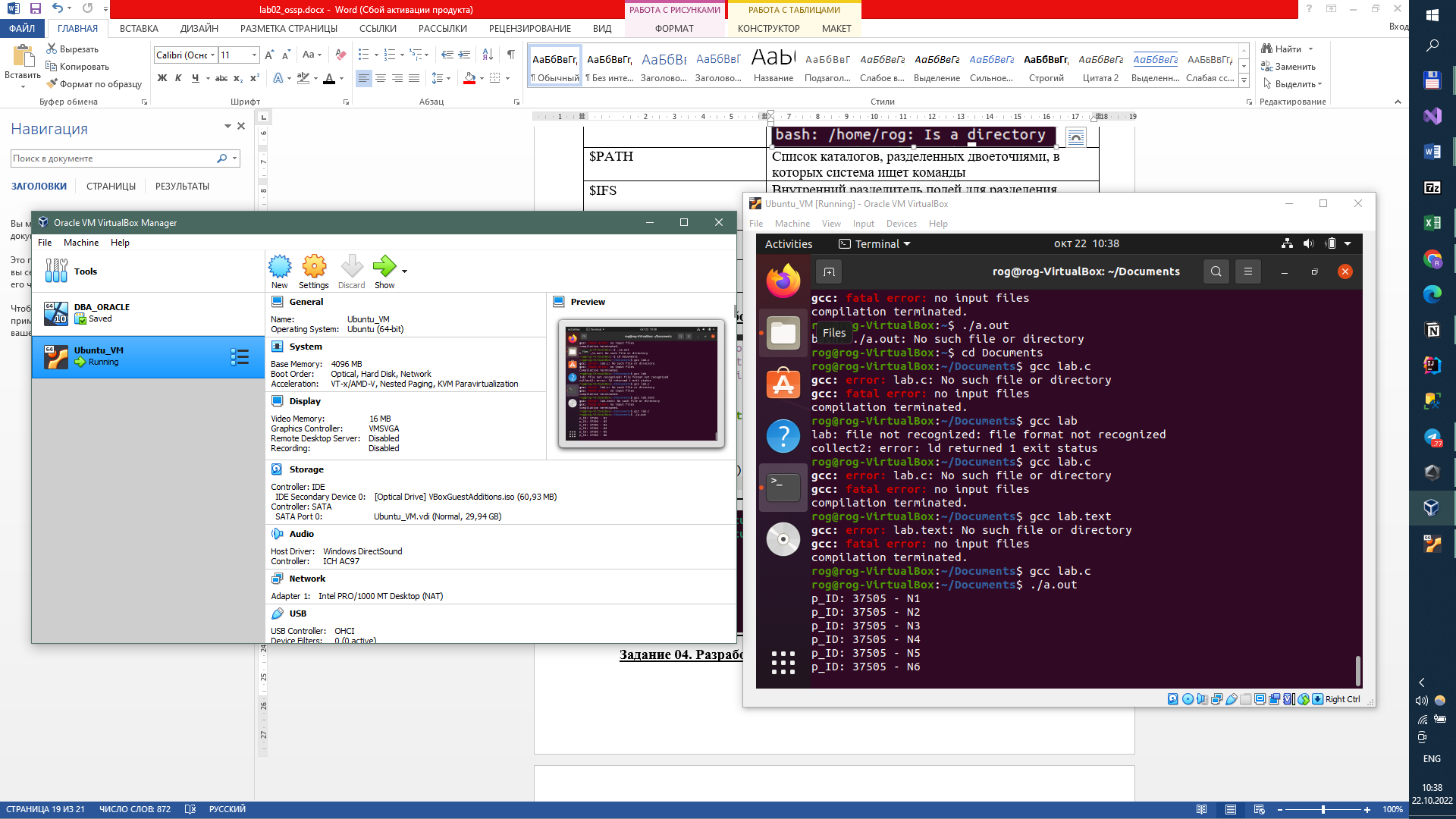
## **Задание 02. Переменные окружения среды Linux**

Командная оболочка будет искать переменную окружения с именем, соответствующим размещенной после символа доллара строке, и заменять данный символ и имя переменной на значение этой переменной (или ни на что в том случае, если переменной не существует).

|  |  |
| --- | --- |
| Переменная окружения | Краткое описание |
| $HOME | Домашний каталог текущего пользователя |
| $PATH | Список каталогов, разделенных двоеточиями, в которых система ищет команды |
| $IFS | Внутренний разделитель полей для разделения ввода в командной строке. По умолчанию это пробел. |
| $PS2 | Используется, когда команда многострочная. |
| $PS1 | Основная строка приглашения (для определения отображения приглашения оболочки) |

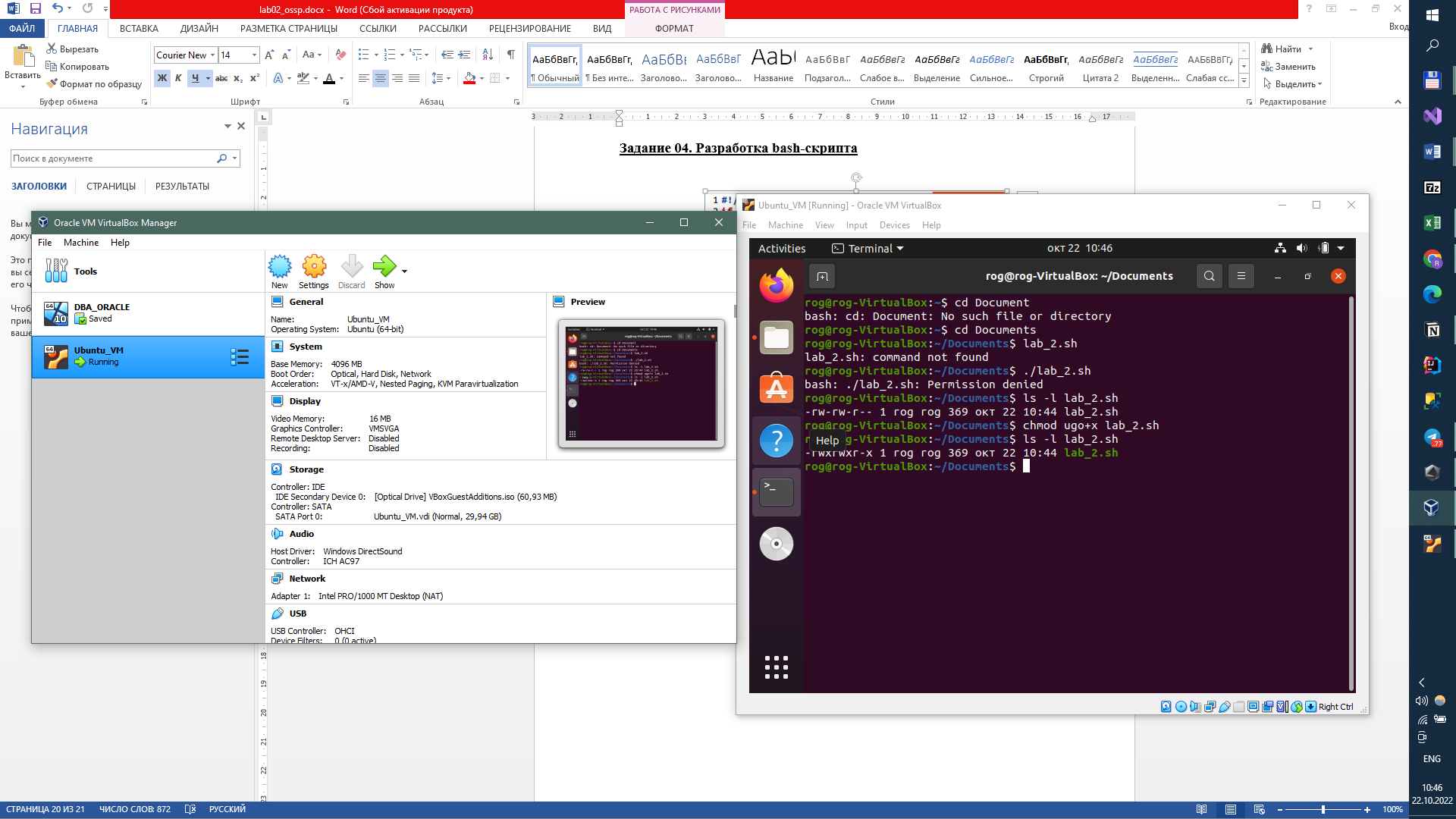
## **Задание 03. Разработка С-приложения**

****

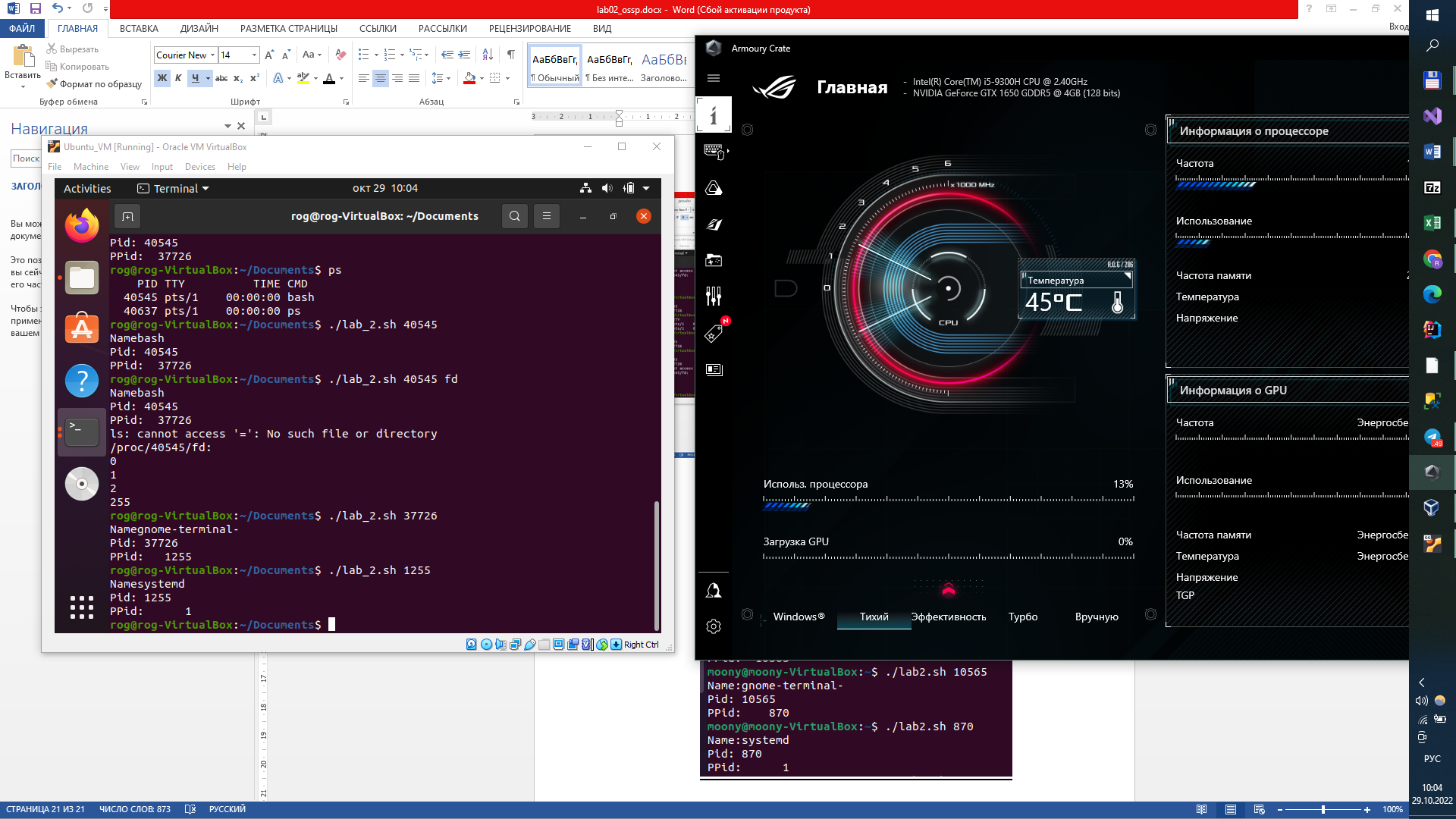


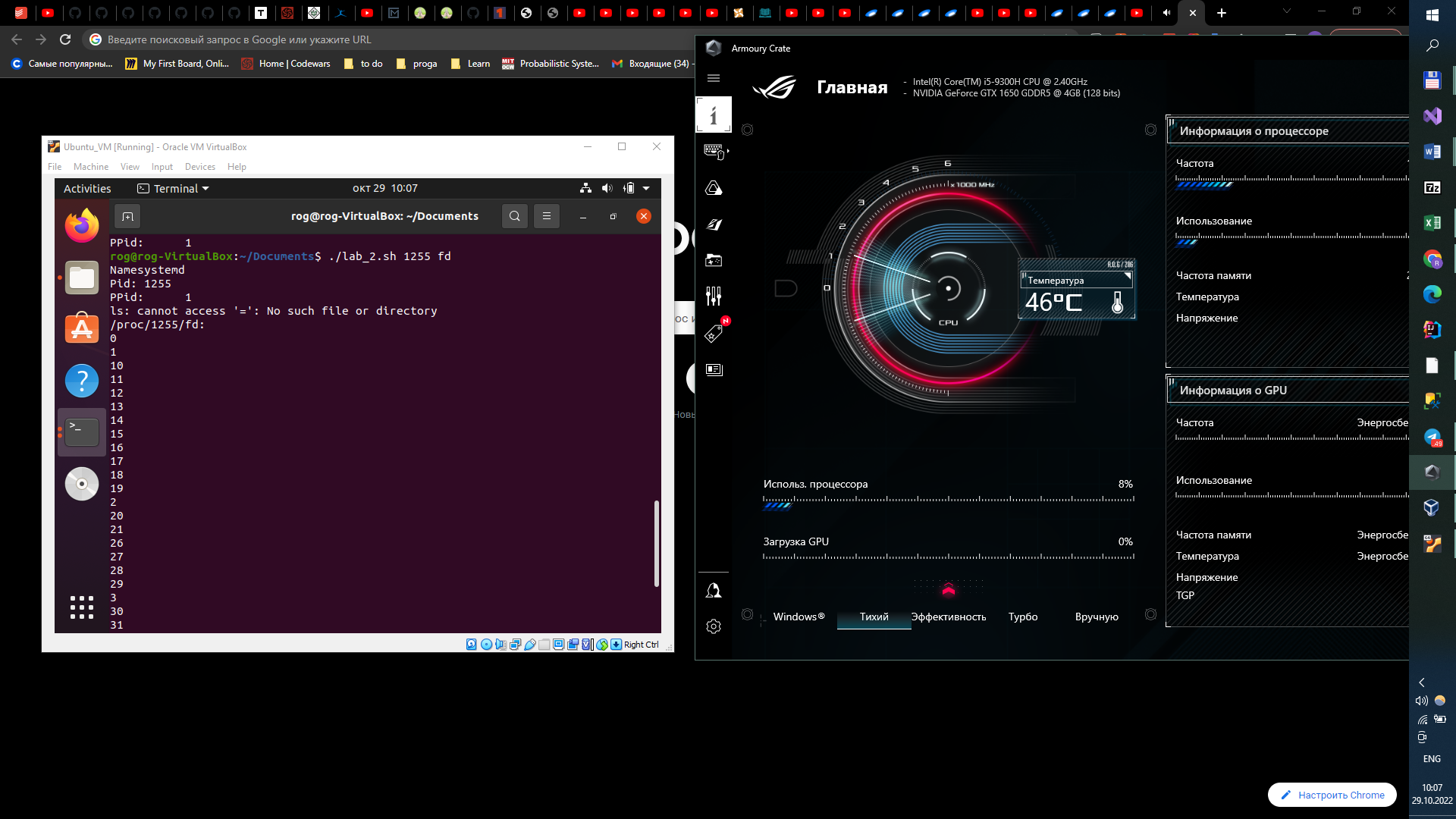
## **Задание 04. Разработка bash-скрипта**

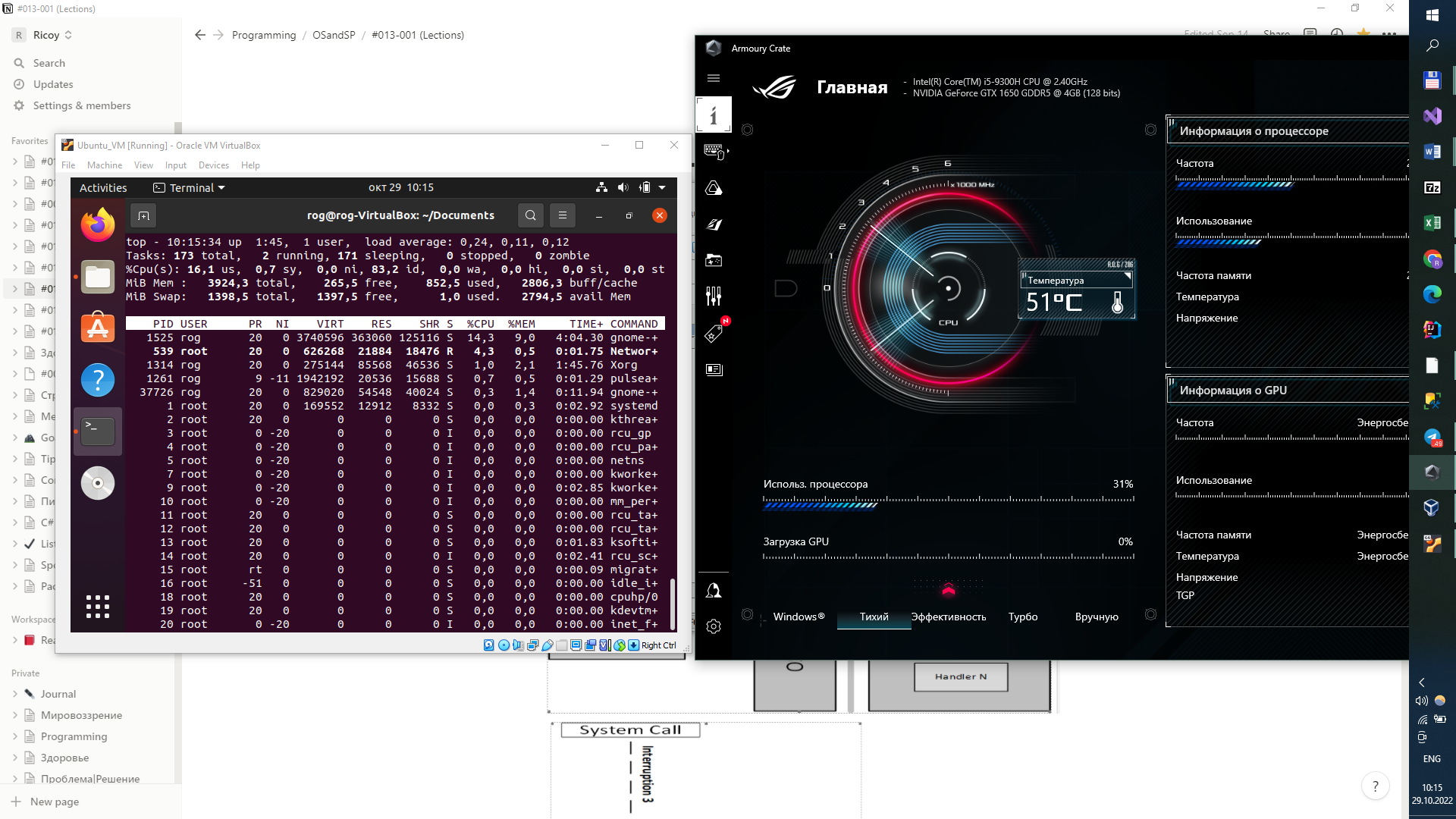
****



## **Задание 05.Построение цепочки родительских процессов**







## **Задание 06. Ответьте на следующие вопросы**

1. **Что такое фреймворк OS?**

Набор библиотек OS + API интерфейс.

1. **Что такое POSIX?**

**POSIX** — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов.

1. **Что такое аппаратное прерывание?**

**Прерывание** (англ. interrupt) — сигнал от программного или аппаратного обеспечения, сообщающий процессору о наступлении какого-либо события, требующего немедленного внимания.

**Асинхронные, или внешние (аппаратные)** — события, которые исходят от внешних аппаратных устройств (например, периферийных устройств) и могут произойти в любой произвольный момент: сигнал от таймера, сетевой карты или дискового накопителя, нажатие клавиш клавиатуры, движение мыши.

**Аппаратное** - реакция процессора на физический сигнал от некоторого устройства (клавиатура, системные часы, жесткий диск и т.д.), по времени возникновения эти прерывания асинхронны, т.е. происходят в случайные моменты времени.

1. **Что такое программное прерывание?**

**Программные (частный случай внутреннего прерывания**) — инициируются исполнением специальной инструкции в коде программы, предназначены для выполнения некоторых действий операционной системы, являются синхронными.

1. **Что такое системный вызов?**

**Системный вызов** - механизм вызова прикладной программой функции ядра OS.

1. **Что такое процесс OS?**

**Процесс OS –** единица работы OS - объект ядра OS + адресное пространство: процессом, по существу, является программа во время ее выполнения.

1. **Что такое контекст процесса OS?**

**Контекст процесса –** данные, которые сохраняются при переключении процессов и предназначенные для продолжения работы; (адресное пространство, содержимое регистров, объекты ядра OS (объекты процессов, потоков, безопасности, файлов и пр.) стек ядра (для этого процесса)).

1. **Что такое адресное пространство процесса?**

*Адресное пространство (данные, программа, стек, куча).*

**Адресное пространство** — это диапазон адресов, обозначающих определенное место в памяти.

С каждым процессом связано его адресное пространство — список адресов ячеек памяти от нуля до некоторого максимума, откуда процесс может считывать данные и куда может записывать их.

Адресное пространство содержит выполняемую программу, данные этой программы и ее стек. Кроме этого, с каждым процессом связан набор ресурсов, который обычно включает регистры (в том числе счетчик команд и указатель стека), список открытых файлов, необработанные предупреждения, список связанных процессов и всю остальную информацию, необходимую в процессе работы программы.

Таким образом, процесс — это контейнер, в котором содержится вся информация, необходимая для работы программы.

1. **Перечислите области памяти адресного пространства процесса и поясните их назначение.**

Процессу выделяется линейное адресное пространство (размер зависит от разрядности), сегменты: code, static, data, heap, stack;



* **сегмент heap** – структура данных, с помощью которой реализована динамически распределяемая память приложения;
* **сегмент кода** – содержит машинные команды, Адресуется регистром CS;
* **сегмент данных** – содержит данные, то есть константы и рабочие области, необходимые программе. Адресуется регистром DS;
* **сегмент стека** – содержит адреса возврата в точку вызова подпрограмм. Адресуется регистром SS.

1. **Что такое стандартные потоки процесса?**

Потоки имеющие зарезервированные номера - дескрипторы (номера), поток ввода (0), поток вывода (1), поток вывода ошибок (2).

1. **Перечислите системные вызовы Windows для создания процесса?**

**CreateProcess**, которая создает новый процесс с единственным потоком. При вызове этой функции требуется указать имя файла исполняемой программы.

1. **Перечислите системные вызовы Linux для создания процесса?**

Процессы создаются через две функции Fork() и exec():

Начинается с **Fork**(), он создает точный клон вызывающего процесса, так называемый «дочерний» процесс

Менеджер исполнения **exec**() заменяет образ процесса этого клона новой программой, которая должна быть выполнена.

1. **С помощью каких утилит можно увидеть перечень процессов в Windows?**

tasklist

1. **С помощью каких утилит можно увидеть перечень процессов в Linux?**

Ps

1. **Перечислите свойства процесса OS.**

**OS: основные свойства процесса:**

* процессу соответствует исполняемый программный файл;
* у процесса есть PID;
* у процесса есть Parent PID;
* в Windows: HANDLE – идентификатор объекта OS;
* в OS есть б (родитель для всех);
* запуск и управление (создать, остановить, …) процессом осуществляется с помощью системных вызовов;
* процессы изолированы друг от друга;
* процессу выделяется линейное адресное пространство (размер зависит от разрядности), сегменты: code, static, data, heap, stack;
* контекст процесса – данные, которые сохраняются при переключении процессов и предназначенные для продолжения работы;
* процессу автоматически доступны три процесса: ввода, вывода, вывод ошибок.
* при запуске OS некоторые процессы (Windows-сервисы, Linux-демоны) загружаются и стартуют автоматически, как правило используются для внутреннего назначения;
* в составе ОS есть таблица, содержащая объекты ядра процессов (состояние, приоритет, указатели на другие объекты); есть средства OS позволяющие ее просматривать;

процесс – единица работы OS.