**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 №4**

по курсу «Операционные системы»

Студент: Камаров Лазизбек Шухрат угли

Группа: БИВ203

Преподаватель: Прокофьева   
 Екатерина Николаевна

Дата: 22.10.2022

Москва 2022

# Задание

Обновить систему до версии 10.0. Для этого перед обновлением ядра необходимо установить источник обновления версии 10.0 с помощью команды apt-repo set p10. Затем нужно обновить пакеты и ядро до новейшей версии согласно теоретическому введению.

Удалить старые версии ядра.

Провести функциональное, нагрузочное тестирование, определить время реакции на события. Прежде чем использовать утилиты для тестирования ядра, их необходимо установить с помощью команды apt-get install.

# Выполнение

# Чтобы обновиться до p10, нужно сначала подкачать все файлы p9.

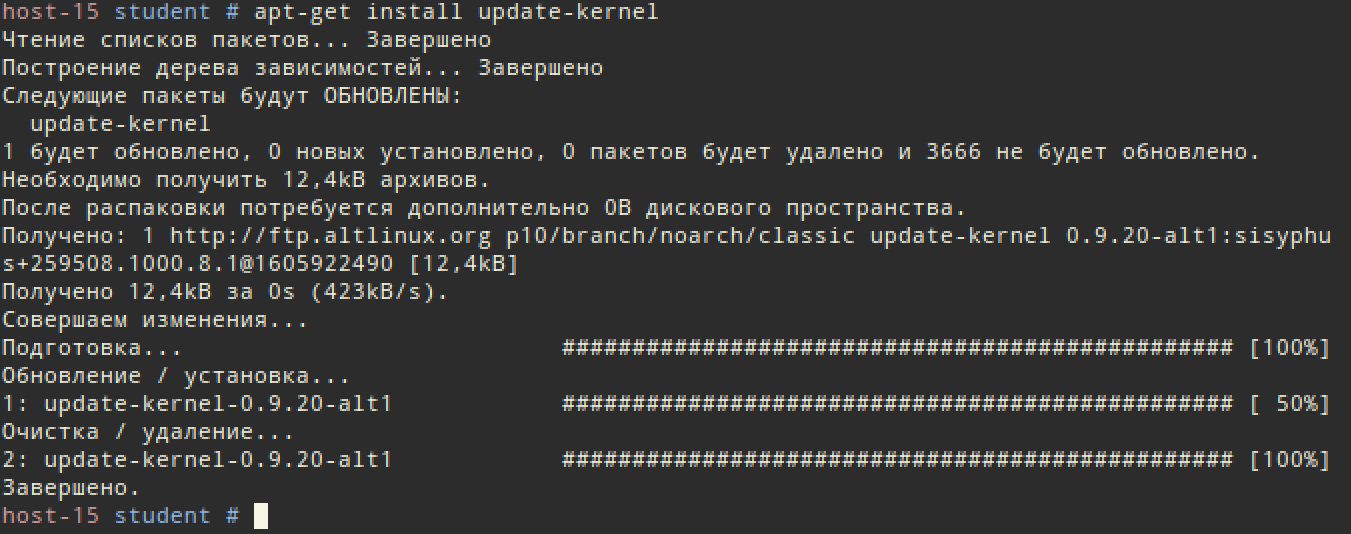
# После этого с помощью команды apt-repo set p10 установим репозиторий

# 

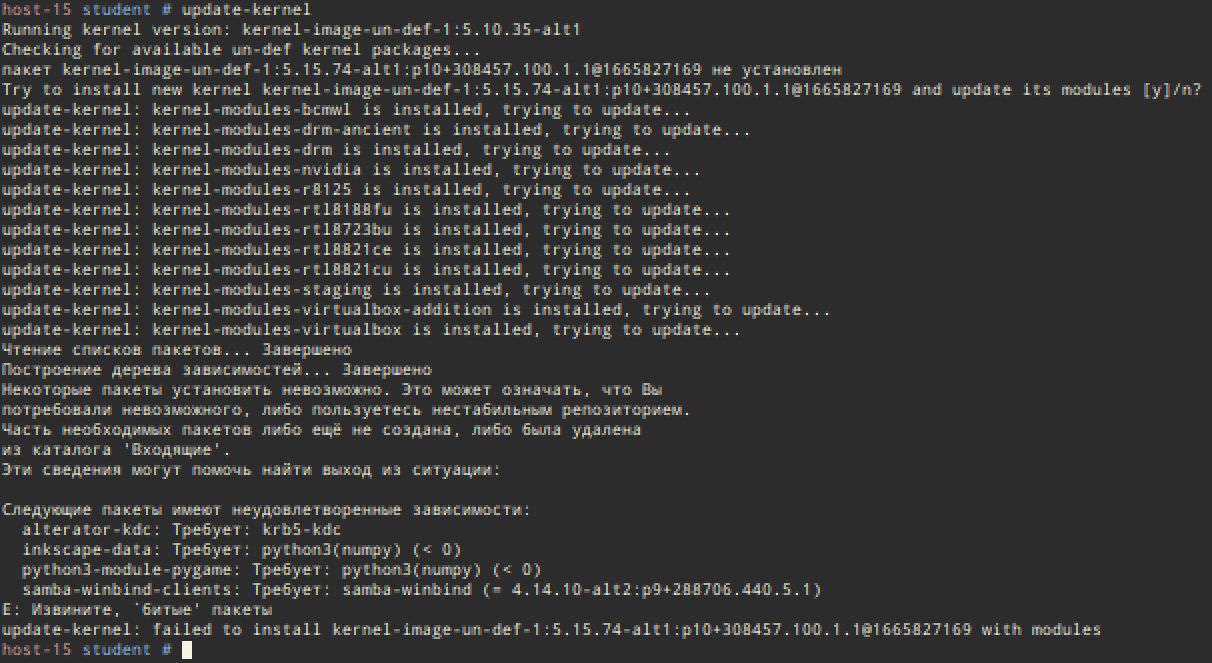
# Установим нужный список пакетов, обновим и посмотрим полные версии доступных пакетов

# 

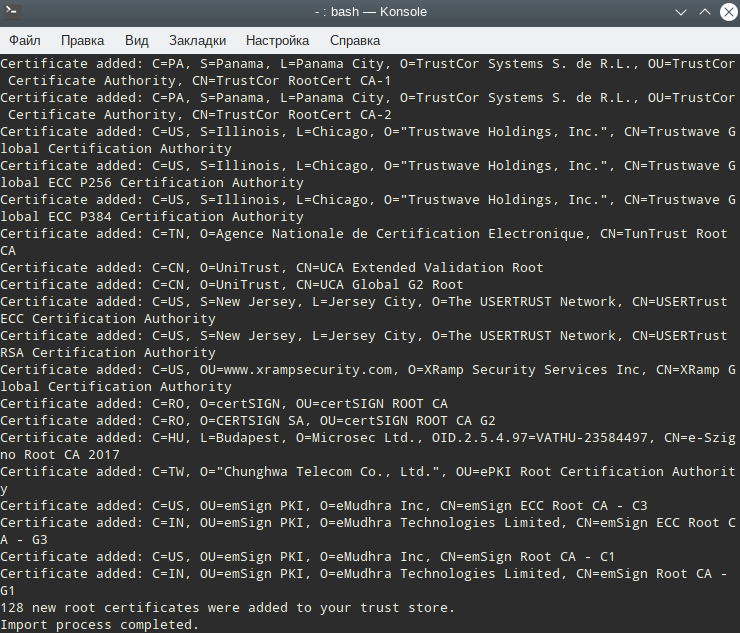
Для обновления ядра установим пакет



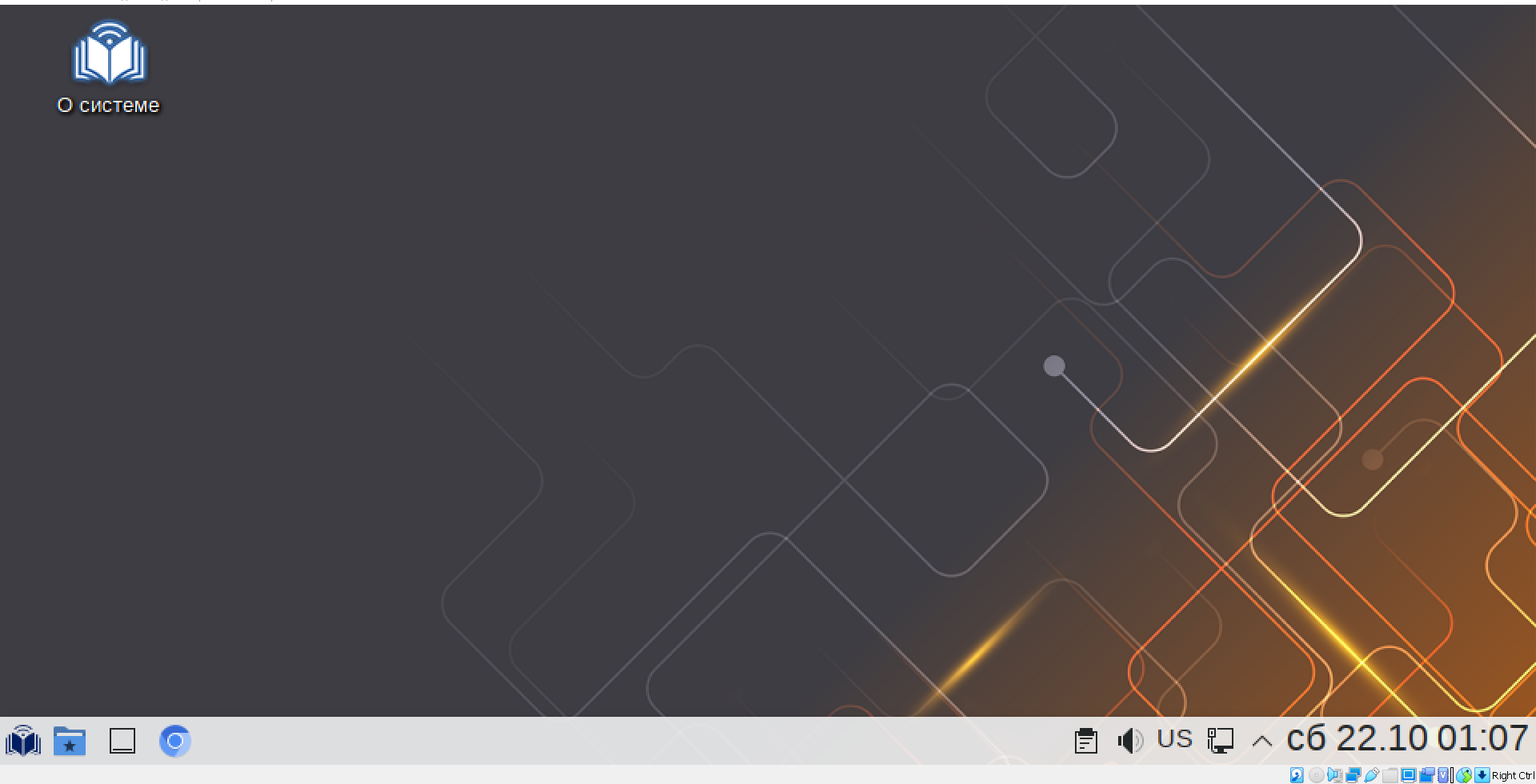
Запустим утилиту update-kernel



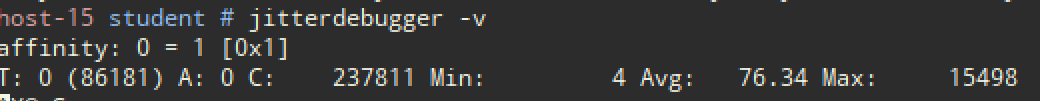
С помощью команды apt-get dist-upgrade проведем новое обновление ядра.



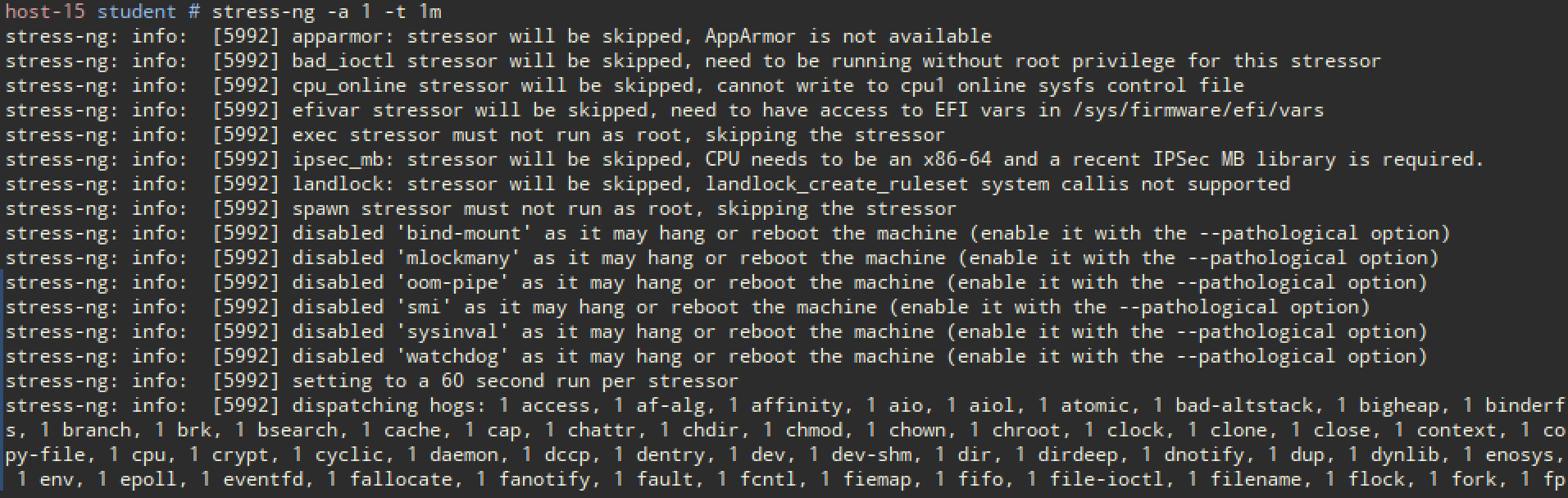
Перезагружаем виртуальную машину и наблюдаем изменение интерфейса.



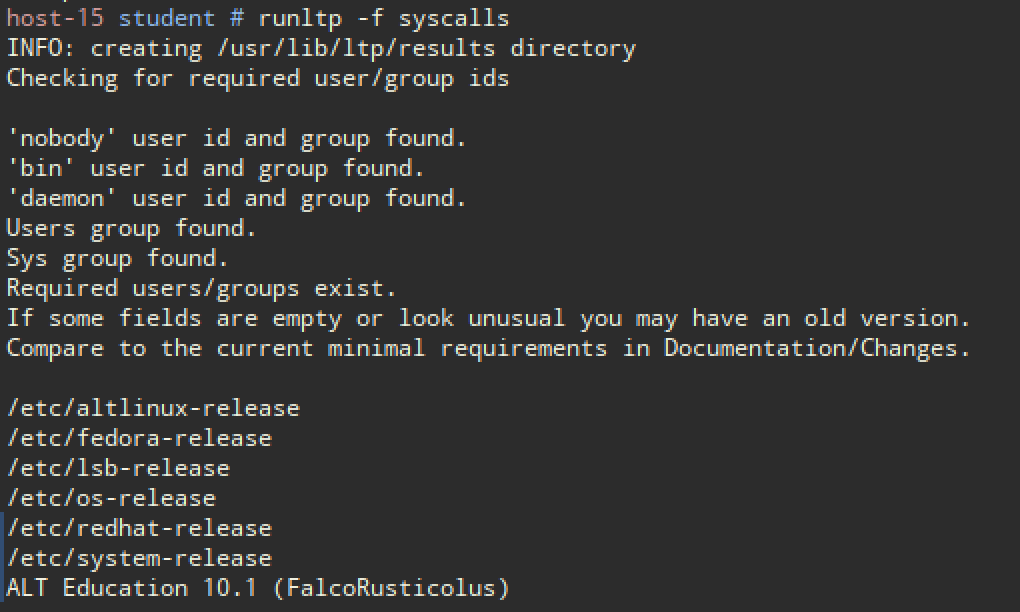
Запустим тест реакции на события через jitterdebugger



Проведём нагрузочное тестирование



Проведём функциональное тестирование



**Аналитические вопросы**

1. Опишите возможную структуру ядра, функционал и возможную динамику изменений на примере ядра Linux ОС.

Ядро **Linux** состоит из пяти основных подсистем:

* **Process Scheduler** (SCHED) – *планировщик процессов*, отвечает за контроль над доступом процессов к CPU. Планировщик обеспечивает такое поведения ядра, при котором все процессы имеют справедливый доступ к центральному процессору.
* **Memory Manager** (MM) – *менеджер памяти*, обеспечивает различным процессам безопасный доступ к основной памяти системы. Кроме того, MM обеспечивает работу [виртуальной памяти](https://rtfm.co.ua/linux-tipy-pamyati/), которая позволяет процессам использовать больше памяти, чем реально доступно в системе. Выделенная, но неиспользуемая память вытесняется на файловую систему, и при необходимости – возвращается из неё обратно в память (swapping).
* **Virtual File System**(VFS) – *виртуальная файловая система*, создаёт абстрактный слой, скрывая детали оборудования, предоставляя общий файловый интерфейс для всех устройств. Кроме того, VFS поддерживает несколько форматов файловых систем, которые совместимы с другими операционными системами.
* **Network Interface** (NET) – *сетевые интерфейсы*, обеспечивает работу с различными сетевыми стандартами и сетевым оборудованием.
* **Inter-Process Communication** (IPC) – *межпроцессная подсистема*, поддерживающая несколько механизмов для *process-to-process* связей в единой **Linux**-системе.

Ядро **Linux** представляет собой своего рода виртуальную машину для процессов. Процессы работают без всякой информации про оборудование компьютера – ядро абстрагирует весь уровень оборудования в единый совместимый виртуальный интерфейс. Кроме того, ядро реализует многозадачность прозрачно для всех процессов – каждый процесс “думает”, что он является единственным процессом в системе, и имеет полные и эксклюзивные права на память и другие ресурсы оборудования компьютера. Фактически же – ядро выполняет несколько процессов одновременно, и оно ответственно за распределение ресурсов оборудования таким образом, что бы каждый процесс получил достаточный доступ к этим ресурсам.

1. Опишите, как навыки формирования модулей ядра и запуска приложений применимы в прикладной информационно-технической деятельности и в организации инженерных работ.

Думаю, что в дальнейшей прикладной информационно-технической деятельности одним из важных навыков формирования ядра является возможность его «пересборки» под собственные нужды. Можно устанавливать приложения, необходимые для инженерных работ, такие как Qucs (моделирование электрических цепей), AutoCAD и др.

1. Опишите известные Вам утилиты для тестирования ядра, сложности, которые возникли в ходе выполнения задания.

**Linux Test Project** – отвечает за функциональное тестирования. Данный тест предназначит для подтверждения поведения ядра Linux, а также glibc (является библиотекой Си, которая обеспечивает системные вызовы и основные функции, такие как open, malloc, printf и т.д.). Возникла трудность в использовании данной утилиты, так как ее встроенные пакеты не хотели скачиваться. Пришлось использовать официальный сайт Linux с доступными утилитами и после загрузки пакета, вызвать его в терминале.

**Stress-ng –** данная утилита предназначена для нагрузочного тестирования. Данный тест производит нагрузочный стресс-тест, который направлен на тестирование аппаратного обеспечения различными способами. С этой утилитой проблем не возникло, через терминал удалось активировать встроенные пакеты и вызвать тест.

**Jitterdebugger** измеряет латентность. Эта утилита запускает поток на каждом CPU, который программирует таймер и измеряет время, которое проходит с момента истечения срока действия таймера до повторного запуска потока, установившего таймер.

Сложности возникли с утилитами **Linux Kernel Selftests** и **cyclictest**, так как их пакеты не удалось загрузить.

1. Опишите различия, информацию о которых Вам удалось найти по ОС версии 9.2 и 10 Альт Образование.

* Более удобный и красивый интерфейс.
* Возможность поддержки устройств на базе процессоров Байкал-М.
* Введен модуль настройки  многотерминального режима alterator-multiseat.
* Различные версии программного обеспечения, такие как ядро Linux, офисный пакет LibreOffice-still, редактор растровой графики, редактор векторной графики и т.д.

1. Опишите различия, которые возникают при формировании обновлённой, более совершенной версии ядра.

Формирование обновлённой версии ядра заняло около 1.5 часов



Сам процесс формирования я пропустил, но при перезагрузке системы изменился интерфейс.