

Министерство образования и науки РФ  
Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и кибербезопасности  
Высшая школа программной инженерии

**Лабораторная работа №6**  
**“Сборка ядра Linux”**

Работу выполнил студент  
Кладковой Максим Дмитриевич  
Группа: 5130904/30005  
Руководитель: Петров Александр Владимирович

## Содержание

Содержание.....	2
Аппаратная платформа.....	2
Программная платформа.....	2
Задание.....	2
Подготовка к выполнению работы.....	4
Подготовка к сборке ядра.....	4
Сборка ядра:.....	4
Завершение и проверка на правильность установки:.....	5
Выполнение работы:.....	5
Заключение.....	6

## Аппаратная платформа

Honor MagicBook HUAWEI NBLK-WAX9X

CPU: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx

## Программная платформа

PRETTY\_NAME="Debian GNU/Linux 12 (bookworm)"

NAME="Debian GNU/Linux"

VERSION\_ID="12"

VERSION="12 (bookworm)"

VERSION\_CODENAME=bookworm

ID=debian

HOME\_URL="https://www.debian.org/"

SUPPORT\_URL="https://www.debian.org/support"

BUG\_REPORT\_URL="<https://bugs.debian.org/>"

## Задание

1. Установить исходный код ядра, предоставляемый вашим дистрибутивом (ванильная версия не рекомендуется).
2. Сконфигурировать и собрать ядро из установленных исходных файлов.
3. Протестировать систему с новым ядром.

4. Разработать сценарий, который запускает сборку ядра в цикле для  $-jN$  со значениями от 1 до  $2N+1$ , где  $N$  – число ядер в системе, включая виртуальные.
5. Число ядер можно узнать по `cat /proc/cpuinfo`. Сценарий возвращает только время работы сборки на процессоре (используйте `time`, а все сообщения `make-kpkg` перенаправляйте в `/dev/null`). На каждой итерации очищайте дерево исходного кода (например, `make-kpkg clean`).
6. Предоставить отчет о проделанной работе. Дополнительно необходимо предоставить файл конфигурации ядра.
7. Отчет и файл конфигурации необходимо представить в виде архива, названного в соответствии со следующим шаблоном: <первая буква имени студента><фамилия студента><номер группы студента>.
8. После согласования с преподавателем предоставить отчёт.

### Цели

1. Сконфигурированное и собранное ядро Linux.
2. Время сборки ядра при различном числе потоков сборки.
3. Нахождение оптимального числа потоков для сборки ядра.
4. Выполнение индивидуального задания.

### Задачи

1. Подготовка системы.
2. Установка исходного кода ядра.
3. Конфигурация ядра.
4. Сборка ядра.
5. Установка ядра.
6. Очищение дерева сборки.
7. Проверка работоспособности.
8. Написание сценария, собирающего ядро на потоках от 1 до  $2N+1$  и выводящего время.
9. Поиск оптимального числа потоков для сборки ядра.
10. Собрать ядро для дистрибутива, используя исходный код, предоставляемый дистрибутивом.(для ИЗ).
11. Собрать ядро для дистрибутива, используя исходный код, предоставляемый ресурсом `kernel.org`, с наложением на него заплат от дистрибутива.(для ИЗ).
12. Подведение итогов.

## Подготовка к выполнению работы

### Подготовка к сборке ядра

#### 1. Установка пакетов:

- fakeroot - Создает поддельную корневую среду.
- Xz-utils - Обеспечивает быстрое сжатие и распаковку файлов .
- Libssl-dev - Поддерживает SSL и TLS , которые шифруют данные и обеспечивают безопасность подключения к Интернету.
- Libelf-dev - Выпускает общую библиотеку для управления файлами ELF (исполняемые файлы, дампы ядра и объектный код).
- Bison - Преобразует описание грамматики в программу на языке C.
- Git - Отслеживает и записывает все изменения в исходном коде во время разработки. Это также позволяет отменить изменения.
- kernel-package используется для автоматизации рутинных шагов, необходимых для компиляции и установки другого (настраиваемого) ядра.

#### 2. Загрузил и извлёк исходный код ядра:

- wget <https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.15.149.tar.xz> && tar xvf linux-5.15.149.tar.xz

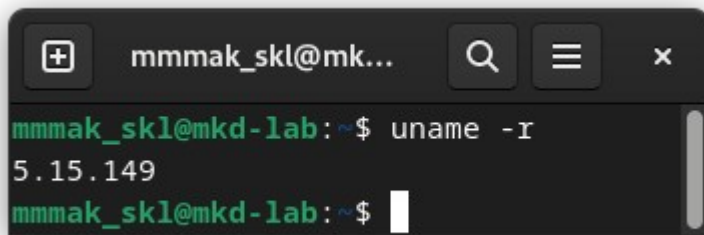
#### 3. Перешел в каталог, который будет использоваться для сборки ядра и скопировал существующий файл конфигурации:

- cd Загрузки/linux-5.15.149
- cp -v /boot/config-\$(uname -r) .config

## Сборка ядра:

1. sudo make
2. sudo make modules
3. sudo make modules\_install
4. sudo make install
5. reboot

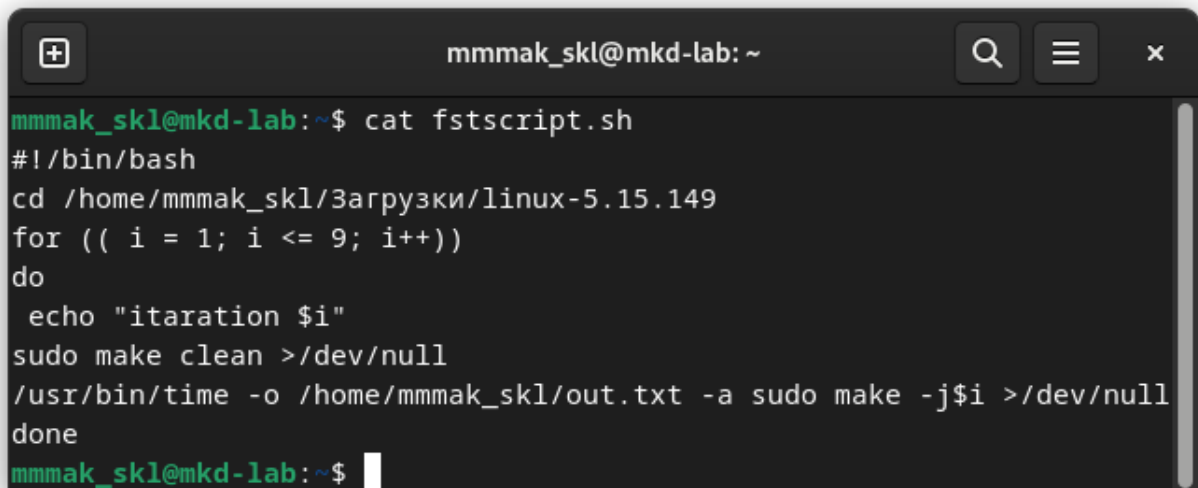
Завершение и проверка на правильность установки:



```
mmmak_skl@mkd-lab:~$ uname -r
5.15.149
mmmak_skl@mkd-lab:~$
```

Выполнение работы:

1. Написал скрипт для определения скорости сборки ядра на разном количестве потоков папо ftscrip.sh



```
mmmak_skl@mkd-lab:~$ cat ftscrip.sh
#!/bin/bash
cd /home/mmmak_skl/Загрузки/linux-5.15.149
for (( i = 1; i <= 9; i++))
do
    echo "itaration $i"
    sudo make clean >/dev/null
    /usr/bin/time -o /home/mmmak_skl/out.txt -a sudo make -j$i >/dev/null
done
mmmak_skl@mkd-lab:~$
```

Запуск скрипта:

1. `chmod +x ftscrip.sh`
2. `./ ftscrip.sh`

График соотношения количества потоков ко времени сборки:



N потоков	Время (min )
1	200:06
2	111:50
3	81:39
4	67:44
5	63:47
6	59:55
7	56:48
8	54:41
9	54:39

## Заключение

1. В результате выполнения данной работы была достигнута цель поиска наиболее подходящего количества потоков при сборке ядра. Была выполнено 9 итераций сборки ядра.
2. Прделанные действия в ходе выполнения работы:
  - Установка зависимостей для сборки;
  - Скачивание исходного кода для ядра, поддерживаемого дистрибутивом;
  - Сборка ядра;
  - Установка собранного ядра;
  - Загрузка ядра;

- Написание скрипта для определения скорости сборки ядра на разном количестве потоков;
  - Поиск оптимального количества потоков для сборки;
3. Самым оптимальным количеством потоков на аппаратной платформе является 8.