ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

«**Конфигурация и установка ядра Linux**»

по дисциплине

«Системное программное обеспечение GNU/Linux»

**Выполнил:**

Студент гр. 23534/2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Стягов А. Ю.

**Руководитель:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фёдоров С. А.

**Аппаратная и программная платформы 2**

**Задачи и цели 3**

**Ход работы. Компиляция ядра 4**

**Ход работы. Проверка скорости сборки 5**

**Результаты и график 6**

**Выводы 7**

Аппаратная платформа

**CPU**:

***MODEL***: Intel® Core™ i7-4510U @ 2.00 GHz

***ARCHITECTURE***: x86-64

***ARCHITECTURE CODE NAME***: Haswell

***CORES***: 2

***THREADS PER CORE***: 4

***L1 CACHE***: 128 KiB

***L2 CACHE***:512 KiB

***L3 CACHE***: 4 MiB

**GPU:**

***INTEGRATED***: Intel® HD Graphics 4400

***DISCRETE***: NVIDIA GeForce GT 750M

**RAM:**

***VENDOR***:Kingston

***SIZE***: 8GB

***TYPE***: DDR3

***FREQUENCY***:1600MHz

***DETAILS***: PC3-12800 204-Pin SoDIMM non-ECC Unbuffered 1.35V Dual Rank Notebook Memory KN2M64-ETB

***COUNT***: **x2**

**HDD**: Seagate ST1000LM014-1EJ1 931GiB (1TB)

Программная платформа

**PRETTY NAME**: "Debian GNU/Linux 9 (stretch)"

**NAME**: "Debian GNU/Linux"

**VERSION ID**: "9"

**VERSION**: "9 (stretch)"

**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 -j N со значениями от 0 до 2N+1, где N– число ядер в системе, включая виртуальные. Число ядер можно узнать по # cat /proc/cpuinfo. Сценарий возвращает только время работы сборки на процессоре (используйте time, а все сообщения make-kpkgперенаправляйте в /dev/null). На каждой итерации очищайте дерево исходного кода (например, make-kpkg clean).
5. Предоставить отчет о проделанной работе. Дополнительно необходимо предоставить файл конфигурации ядра.
6. Отчет и файл конфигурации необходимо представить в виде архива, названного в соответствии со следующим шаблоном: <первая буква имени студента><фамилия студента><номер группы студента>

Цели

1. Сконфигурированное и собранное ядро Linux
2. Время сборки ядра при различном числе потоков сборки
3. Оптимальное число потоков для сборки ядра

Ход работы

Подготовка к компиляции ядра

#### Последовательность команд, использованных при подготовке:

Авторизация в качестве root:

$ su

Переход в папку, куда будет загружен архив с ядром:

# cd /home/usr/as/Downloads/tmp

Получение архива с ядром:

# wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.1.tar.xz

Распаковка архива с ядром в текущую папку:

# tar xvJf linux-4.19.1.tar.xz

Удаление архива:

# rm linux-4.19.1.tar.xz

Переход в папку с ядром:

# cd ./linux-4.19.1

Копирование файла конфигурации текущего ядра:

# cp /boot/config-$( uname -r ) ./.config

Установка пакетов, необходимых для сборки:

# apt-get install kernel-package

# apt-get install fakeroot libncurses5-dev

# apt-get install build-essential pkg-config bison flex libssl-dev

Конфигурирование:

# make menuconfig

Компиляция ядра

Запуск компиляции ядра с использованием 2 ядер:

# fakeroot make-kpkg --initrd kernel\_image -j 2

Установка ядра

Был получен следующий пакет: **linux-image-4.19.1\_4.19.1 10.00.Custom\_amd64.deb**

Установка данного пакета:

# dpkg -i linux-image-4.19.1\_4.19.1 10.00.Custom\_amd64.deb

Перезагрузка для применения изменений:

# reboot

Проверка корректности установки:

# uname -a

Linux debian 4.19.1 #1 SMP Thu Nov 8 03:00:50 MSK 2018 x86\_64 GNU/Linux

### Проверка скорости сборки

Скрипт сборки ядра с использованием разного количества потоков:

#!/bin/sh

for n in $(seq 1 5)

do

make-kpkg clean

/usr/bin/time -o stats -a fakeroot make-kpkg --initrd -j $n kernel\_image > /dev/null

done

Запуска скрипта:

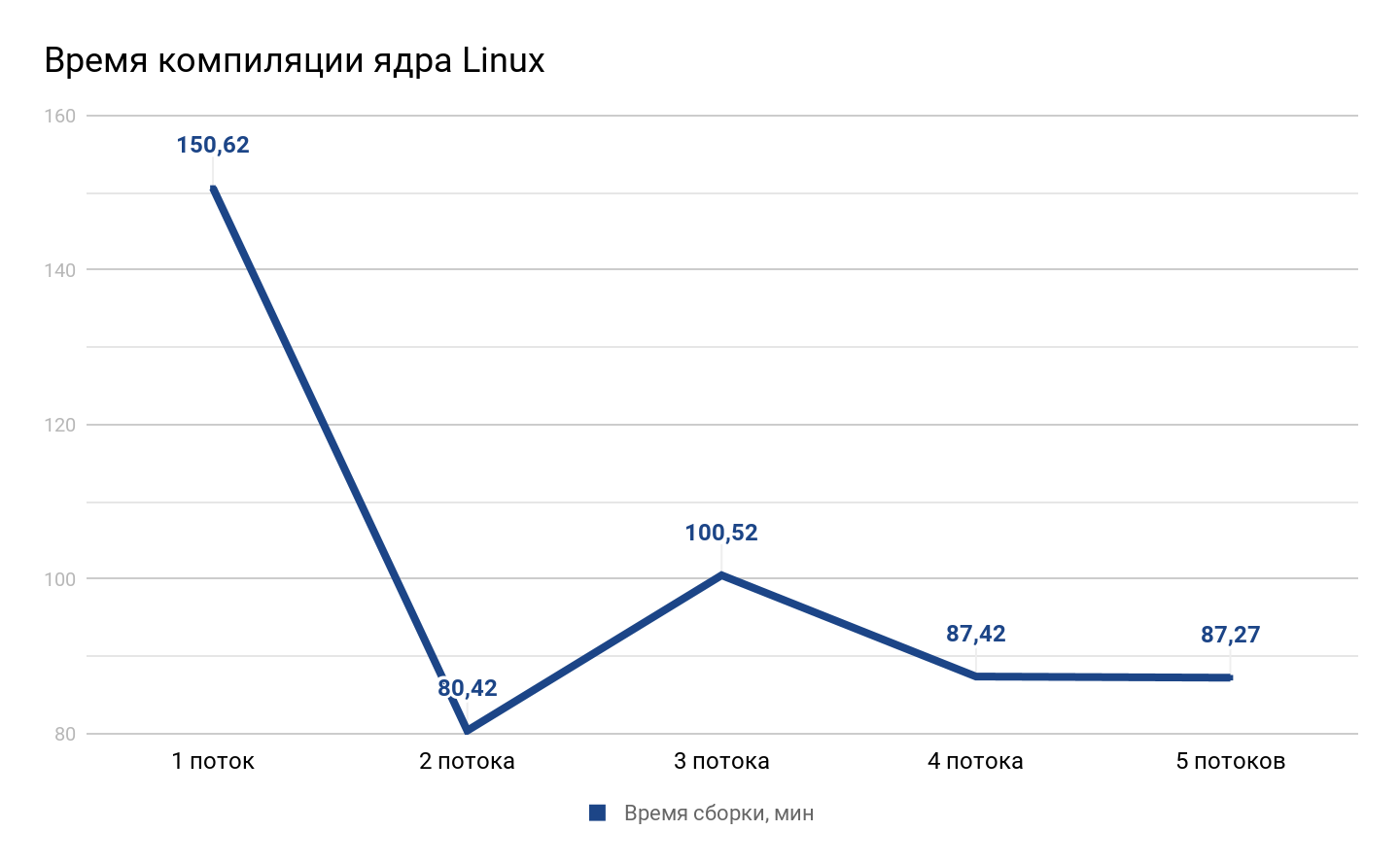
# chmod +x ./build-kernel.sh

# ./build-kernel.sh

### Результаты:

|  |  |
| --- | --- |
| **Потоков** | **Время, мин** |
| 1 | 150,62 |
| 2 | 80,42 |
| 3 | 100,52 |
| 4 | 87,42 |
| 5 | 87,27 |

График времени сборки ядра



Вывод

1. В результате проведённой нами работы мы получили собранное и сконфигурированное ядро Linux.

2. Нам удалось установить это ядро, проверить его работоспособность.

3. Было выявлено оптимальные число потоков при сборке ядра:

N = 2 (Равное количеству ядер процессора).

4. Были решены трудности, возникшие на этапе компиляции ядра: установка недостающих пакетов (build-essential pkg-config bison flex libssl-dev).

5. В результате выполнения этой работы мы научились собирать и конфигурировать кастомные ядра Linux.