Тема №5. Сборка ядра Linux

Редакция от 09.02.18.

Введение

Ядро рекомендуется пересобирать в следующих случаях:

- если установлено специфичное оборудование или возникает конфликт аппаратного обеспечения со стандартным ядром;
- для задействования свойств, которых нет в поставляемых сборках ядра;
- для оптимизации ядра, удаляя ненужные драйверы для уменьшения времени загрузки;
- создания монолитного ядра, без модулей (бывает необходимо при создании специфичных систем);
- для установки обновлённого ядра, в котором есть необходимые свойства или поддержка вашего оборудования;
- наконец, чтобы больше узнать о ядрах Linux.

Какие бывают ядра?

Прежде всего, надо разобраться, как нумеруются ядра. Пусть имеется ядро версии **a**.**b**.**c**

- **a** - это основной номер версии.
- **b** - это номер ядра в серии, отличия не радикальные, но существенные: появились новые драйвера, устранены ошибки, добавлены новые возможности.
- **c** - означает стабилизацию ядра. Устраняются мелкие ошибки.

Официальные ядра в виде исходных текстов можно скачать с ftp://ftp.kernel.org.

Какие возникают проблемы при сборке ядра?

- вы не включили в ядро то, что очень нужно, и теперь
  - система не загружается
  - пропала возможность работы с некоторым оборудованием

- вы включили в ядро то, что не надо (особенно с надписью EXPERIMENTAL) и теперь
  - система работает нестабильно или очень медленно
  - ядро вываливается в kernel panic

- вы взяли нестабильное ядро и/или компилятор в бета-версии/выставили дикие флаги оптимизации
  - вас предупреждали

Основные проблемы

Как и всё остальное в Linux, тексты ядра прокомментированы и можно в процессе сборки воспользоваться подсказками. В menuconfig это крайняя правая кнопка HELP. Комментарии есть практически к каждой опции.

Не включайте в ядро и не делайте модулем никакие функции, рядом с которыми написано слово EXPERIMENTAL! Это может вывести систему из стабильного состояния.

Сборка ядра, предоставляемого вашим дистрибутивом

1. *Подготавливаем систему*

Установите пакеты, необходимые для сборки ядра в вашем дистрибутиве. Например, kernel-package, fakeroot, libncurses5-dev.

2. *Установка исходного кода ядра*

Установите пакет вашего дистрибутива, предоставляющий исходный код ядра, например, (kernel-source-x.x.xx).

Исходные файлы устанавливаются обычно в /usr/src.

3. *Конфигурируем ядро, используя прежнюю конфигурацию*

cd /usr/src/linux-x.xx

cp /boot/.config .

make menuconfig

4. *Собираем ядро*

Соберите ядро, как это определяет ваш дистрибутив. Например:

fakeroot make-kpkg -jN kernel_image (где N - количество потоков, которое будет задействовано)

5. Устанавливаем новое ядро

После сборки установите собранный пакет типа kernal-x-xx.deb, содержащий ядро. Например:

dpkg -i file.deb

В директории /boot будут созданы следующие файлы:

- vmlinuz-x.xx
- System.map-x.x
- initrd.img-x.xx
- config-x.xx

Загрузчик обновится автоматически при установке пакета ядра.

6. Проверяем правильность установки

reboot

uname -r

Сборка ядра без заплат (ванильное ядро) (не рекомендуется)

1. *Скачиваем последнюю стабильную версию ядра*

cd /usr/src/

wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-3.xx.tar.xz

2. *Распаковываем исходный код ядра*

tar -xvJf linux-3.12.tar.xz

3. *Конфигурируем ядро, используя прежнюю конфигурацию*

cd linux-3.12

cp /boot/.config .

make menuconfig

4. *Собираем ядро*

make -jN (где N - количество потоков, которое будет задействовано)

5. Устанавливаем новое ядро

make install

В директории /boot будут созданы следующие файлы:

- vmlinuz-3.12
- System.map-3.12
- initrd.img-3.12
- config-3.12

6. Проверяем правильность установки

reboot

uname -r

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 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-kpkg перенаправляйте 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>*
7. После согласования с преподавателем

Требования к отчёту

1. Оформление отчёта: взять на кафедре.

2. Название лабораторной работы: «Конфигурация и установка ядра Linux».
3. Целью работы является не процесс (сборка, установка и т. п.), а основной конечный результат работы, например, сконфигурированное и собранное ядро Linux с указанием платформы или время сборки ядра при различном числе потоков сборки.
4. Задачи, необходимые для достижения цели, формулируются, исходя из основных проводимых действий («подготовка системы ...», «установка исходных кодов ядра», «конфигурация ядра...» и т. д.).
5. Отчёт состоит из основных структурных элементов согласно требованиям.
6. Необходимо указать аппаратную и программную платформы, для которых проводится конфигурация и сборка ядра.
7. Основная часть делится на элементы с привязкой к задачам. При описании решаемых задач приводятся вызываемые команды и их вывод. Вывод при сборке ограничить 10 строками. Снимки экрана *не приводятся*.
8. Постройте *график времени сборки* в зависимости от числа потоков сборки для n=0, 2N+2.
9. Приведите возникшие трудности и ошибки, которые были преодолены.
10. В выводах указывается:
    1. достигнутая цель;
    2. решённые задачи;
    3. возникшие трудности;
    4. итоги по необходимому числу потоков для эффективной сборки ядра на вашей системе;
    5. выводы, которые вы сделали для себя.