

# ОС UNIX, лекция 3: пакетная система и ядро Linux

Пожидаев М. С.

18 февраля 2020 г.

# Понятие пакета в дистрибутиве GNU/Linux

Пакет в дистрибутиве GNU/Linux — сжатый файл (в некоторых случаях серия файлов), содержащий программный код, как бинарный, так и в виде исходных текстов, данные программ, информационные поля и сведения об отношениях с другими пакетами. Подготовленные пакеты публикуются в сети для загрузки пользователями.

1. *Бинарные пакеты*: предназначены для доставки пользователям откомпилированных приложений.
2. *Пакеты с исходными текстами*: предназначены для автоматизированной сборки с последующим получением бинарного пакета.

# Популярные форматы пакетов

- ▶ RPM
- ▶ Deb
- ▶ Pacman
- ▶ Portage

- ▶ APT
- ▶ Zypper
- ▶ DNF и Yum

# Типы отношений

1. *Requires*: пакет *a* требует наличия пакета *b*.
2. *Conflicts*: пакет *a* не может быть установлен вместе с пакетом *b*.
3. *Provides*: пакет *a* предоставляет функциональность пакета *b*.
4. *Obsoletes*: пакет *a* является обновлением пакета *b*.

Существуют другие типы зависимостей, накладывающие более мягкие ограничения или имеющие рекомендательный характер.

- ▶ *Provides*:  $(!p \vee r_1 \vee r_2 \vee r_3)$
- ▶ *Conflicts*:  $(!p \vee !c)$

$$(!p \vee r_1 \vee r_2 \vee r_3) \wedge (!p \vee r_4 \vee r_5) \wedge (!p \vee r_6) \wedge (!p \vee !c_1) \wedge (!p \vee !c_2) = 1$$



1. Исключение альтернатив для provides (сведение к 2-SAT).
2. Использование приближённых методов (minisat).

# Недостатки пакетной системы

1. Низкая бинарная совместимость.
2. Игнорирование таких систем, как PIP, NPM и пр.
3. Слабая пригодность для мобильных систем и систем “умного дома”.

Ответом на указанные недостатки стало появление таких проектов как Flatpack и Snappy.

1. Взаимодействие с оборудованием.
2. Управление процессами.
3. Поддержка файловых систем.
4. Реализация сетевого стека.
5. Разграничение полномочий пользователей.

# Порядок загрузки системы

1. Работа загрузчика GNU GRUB или LILO.
2. Основная загрузка системы.
3. Монтирование корневого раздела в режиме только для чтения.
4. Запуск `/sbin/init`:
  - 4.1 Перемонтирование корневой файловой системы в режиме чтения/записи.
  - 4.2 Запуск демонов.
  - 4.3 Запуск пользовательских сессий.

Системный вызов — обращение прикладной программы к функциям ядра. Обычно выглядит как вызов функции. Обладает следующими особенностями:

1. Часто предполагает повышение привилегий.
2. Подсчёт времени выполнения переходит из пользовательского пространства в пространство ядра.

Модули ядра — способ динамической загрузки кода, исполняемого в пространстве ядра операционной системы. Нельзя путать с динамически подключаемыми библиотеками, код которых исполняется в пространстве пользователя.

Примеры команд для работы с модулями ядра:

- ▶ `modprobe` — загрузка модуля ядра
- ▶ `modinfo` — получение информации о модуле
- ▶ `lsmod` — получение списка загруженных модулей

Операции с модулями ядра может выполнять только суперпользователь `root`.

## Пример блока в выводе lspci

```
02:00.0 Ethernet controller: Realtek Semiconductor
Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit
Ethernet Controller (rev 06)
Subsystem: Micro-Star International Co., Ltd. [MSI]
RTL8111/8168/8411 PCI Express Gigabit Ethernet
Controller
Flags: bus master, fast devsel, latency 0, IRQ 17
...
Kernel driver in use: r8169
Kernel modules: r8169
```

Спасибо за внимание!

Веб-сайт: <http://marigostra.ru/>

E-mail: [mSP@luwrain.org](mailto:mSP@luwrain.org)