

Модульная анатомия

Цель

Разобраться в структуре загружаемых модулей Linux, больше узнать об устройстве ядра.

Ресурсы

Ноутбук Acer Aspire 3680 (старенький), дистрибутив Debian, голова и интернет.

Действия

Первый пункт

Нам нужно скачать, скомпилировать исходники ядра, с которым мы будем работать.

Пользуемся возможностями Debian и получаем исходники ядра из репозитория Debian (всегда можно воспользоваться и <http://kernel.org>)

- sudo apt-get update
- sudo apt-get install liux-doc-2.6.32 linux-manual-2.6.32 linux-source-2.6.32
- cd /usr/src/
- tar jxf /usr/src/linux-source-2.6.32.tar.bz2
- sudo apt-get install build-essential fakeroot kernel-package
- make menuconfig
- sudo make-kpkg clean
- sudo fakeroot make-kpkg -initrd -append-to-version=-mine kernel_image kernel_headers
- sudo dpkg -i linux-image-2.6.32-mine-10.00.Custom_i386.deb
- sudo dpkg -i linux-headers-2.6.32-mine-10.00Custom_i386.deb
- sudo update-initramfs -c -k 2.6.32-mine
- sudo shutdown -r now

Дальше, если ядро сконфигурировано нормально, то загружаемся и вроде все, кстати update-initramfs нужен только, если initrd.img-2.6.32-mine не создался на лету, когда устанавливался пакет

*1 советую обновить gcc (если пользуетесь stable, то обновлять с testing, в противном случае ядро не соберется, так как не будет нужных заголовочных файлов), вообще стоит использовать новые версии всех требуемых пакетов

*2 возможно придется доставить некоторые другие пакеты (см /usr/share/doc/kernel-package/Kernel.htm)

*3 ядро можно собрать и не “в стиле Debian”, а обычным образом - нет никакой разницы

*4 в данной версии ядра пришлось поправить файл /usr/src/linux-source-2.6.32/Documentation/lguest/lguest.c, в нем нужно было убрать строку #include <sys/eventfd.h> (21 строка), в противном случае оно просто отказывалось компилироваться, хотя я не понял почему (потому что такое же ядро на другом компьютере собралось без проблем), но погуглив нашел, что такая проблема не только у меня, и что такое решение используют и другие, после такого решения мы получаем при компиляции implicit декларацию функции, короче если в этом месте будет ошибка, то когда она вылезет непонятно

Второй пункт

Проверим, что все работает, для этого напишем какой-нибудь бесполезный модуль, скомпилируем его и посмотрим, что получится

```
# //hello-1.c
# #include <linux/module.h>
# #include <linux/kernel.h>
#
# int init_module(void) {
#     printk(KERN_INFO "Hello world\n");
#     return 0;
# }
#
# void cleanup_module(void) {
#     printk(KERN_INFO "Goodbye world\n");
# }
```

From:

<http://wiki.osll.ru/> - Open Source & Linux Lab



Permanent link:

<http://wiki.osll.ru/doku.php/etc:users:kernel?rev=1264708015>

Last update: **2010/01/28 22:46**