Создание дистрибутива Linux для процессоров OMAP3x

В предыдущей статье [1] было рассмотрено, как запустить готовый образ операционной системы Linux на отладочной плате BeagleBoard [2]. Теперь настало время разобраться, как самостоятельно создать такой образ. В статье будет рассмотрен первый шаг на этом пути — построение среды компиляции для создания собственного дистрибутива Linux. Одним из возможных вариантов реализации этого этапа является использование OpenEmbedded (OE). Пакет OE представляет собой инструмент создания Linux-дистрибутивов для встраиваемых систем на базе различных архитектур. Использование OE позволяет построить все с нуля, загрузив исходный код и необходимые кросс-компиляторы.

Игорь ГУК
gii@scanti.ru

В OpenEmbedded используется пакет BitBake, который предназначен для выполнения задач управления процессом создания дистрибутива Linux. BitBake имеет так называемые файлы-рецепты. Они содержат исходный адрес пакета, зависимости, а также опции компиляции и инсталляции. Среда OE использует информацию из этих рецептов для отслеживания зависимостей, кросс-компиляции и создания бинарных пакетов. В процессе компиляции создается полный набор образов, включая загрузчик, ядро и корневую файловую систему. Для получения более подробной информации читайте руководства по OpenEmbedded [3] и BitBake [4].

Прежде всего, на PC должен быть установлен один из последних дистрибутивов Linux. Рекомендуется использовать Ubuntu 9.10. Именно для этого дистрибутива будет показан пример построения среды компиляции и создания образа операционной системы Linux. Если вы используете операционную систему Windows, тогда можно посоветовать установить Ubuntu в виртуальную машину. Автор использует виртуальную машину VirtualBox [5], где устанавливается операционная система Ubuntu 9.10, под которую выделяется не менее 15 Гбайт пространства на жестком диске. Если вам необходимо использовать другие дистрибутивы, тогда обратитесь к документации по OpenEmbedded [6].

Первое, что необходимо выполнить, — это инсталлирование пакета GIT. Данный пакет является программным обеспечением с открытым исходным кодом и был разработан Линусом Торвальдсом для управления версиями ядра Linux. Теперь GIT — это распределенная бесплатная система управления проектами, которая обеспечивает:

- ветвление версий проекта;
- поддержку автономной работы, локальные изменения могут быть синхронизированы позже;
- распространение фиксации изменений на весь проект;
- хранение полной истории проекта в каждом дереве GIT;
- равноправность всех хранилищ GIT.

Инсталляция GIT выполняется командой ***apt-get install git-core***.

Выполнить эту команду необходимо в консоли от имени суперпользователя (пользователя root), то есть с опцией sudo. Вначале запустите в Ubuntu консоль (терминал), как это показано на рис. 1.

А затем в появившемся окне терминала необходимо набрать команду инсталляции (рис. 2, п. 1) и выполнить ее, нажав клавишу «Ввод».

Если команда с приставкой sudo в терминале вводится первый раз или вводилась уже давно, может потребоваться пароль, который был

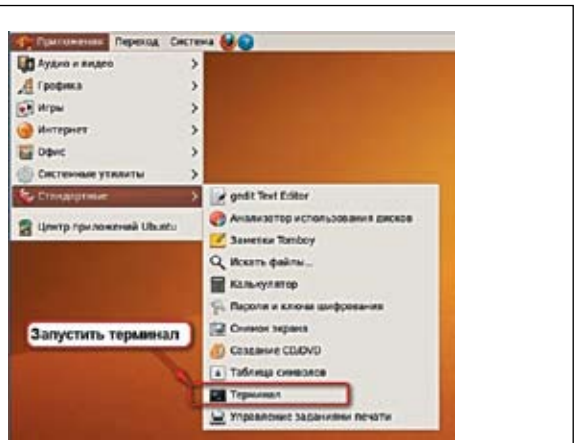

Рис. 1. Запуск консоли в Ubuntu

Рис. 2. Инсталляция системы управления GIT

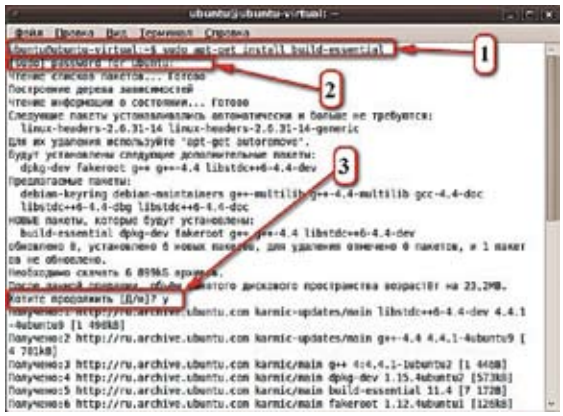

Рис. 3. Установка пакета build-essential

определен при инсталляции системы Ubuntu (пример такого запроса можно видеть на рис. 3, п. 2). При введении пароля необходимо быть очень внимательным, так как он не отображается, даже звездочками. После его набора нажмите «Ввод».

Вначале система будет проверена на наличие необходимых компонентов. Появится сообщение с результатом сканирования и предложением продолжить установку (рис. 2, п. 2). Необходимо ответить положительно: ввести латинскую букву «y» (Yes) и нажать «Ввод». Процесс установки проходит достаточно быстро. По завершению процесса появится консольное приглашение для введения новой команды (последняя строка на рис. 2).

Обратите внимание, что при инсталляции этого и всех последующих дополнительных пакетов необходимо подключение к Интернету. При использовании виртуальной машины подключение к Интернету должно быть у базовой (host) машины. Подключение к Интернету гостевых виртуальных машин, созданных в VirtualBox, происходит автоматически, при помощи механизма NAT (эта настройка для сети стоит в VirtualBox по умолчанию).

Следующий шаг — инсталлирование ряда так называемых дополнений, зависимостей и «заплаток». Этот шаг характерен для операционных систем на базе Linux, где при установке одной программы приходится устанавливать целый ряд дополнительных, возможности которых устанавливаемая программа использует. В данном случае дополнительные компоненты необходимы для корректной работы кросс-компилятора, позволяющего создавать программный код для архитектуры ARM.

Первое, что необходимо установить, — это build-essential. Это дополнение содержит информацию о том, какие пакеты необходимы при создании собственного дистрибутива Linux для встраиваемых систем. Установка выполняется командой ***apt-get install build-essential***, запускаемой от имени суперпользователя так, как это показано на рис. 3.

Инсталляция проходит по уже знакомому сценарию: запуск команды (рис. 3, п. 1), при необходимости ввод пароля (рис. 3, п. 2), сканирование системы и подтверждение дальнейшей установки (рис. 3, п. 3). Процесс завершается появлением консольного приглашение для введения новой команды (на рис. 3 окончание процесса инсталляции не показано).

Затем устанавливаются следующие пакеты:

- help2man — автоматический генератор простых страниц руководств.
- diffstat — программа-фильтр, позволяющая оценить различия, которые происходят, например, при наложении так называемых «патчей» («заплаток»).
- texi2html — Perl-скрипт, который преобразует texinfo-файлы в HTML-файлы.
- cvs — система управления версиями, позволяющая хранить старые копии файлов, например, исходные тексты программ, регистрировать изменения этих файлов, а также кто, когда и почему их выполнил.
- gawk — GNU-реализация языка программирования AWK.
- texinfo — система документирования, использующая единый исходный файл для создания встроенной программной справки, а также подготовки документации для печати.
- subversion — еще одна система управления версиями, иногда называется “svn”, она позволяет множеству разработчиков (даже распределенных географически) совместно работать над одним проектом (чаще всего это файлы с исходными текстами программ).

Более подробную информацию об устанавливаемых дополнениях можно получить в [7]. Этот сайт содержит достаточно полную информацию о пакетах Debian, на которых базируется система Ubuntu.

Все перечисленные выше пакеты также устанавливаются от имени суперпользователя. Для этого необходимо выполнить следующую команду: ***apt-get install help2man diffstat texi2html cvs gawk texinfo subversion***.

Процесс инсталляции показан на рис. 4. Он стандартен: запуск команды (рис. 4, п. 1), ввод пароля (рис. 4, п. 2), сканирование системы, подтверждение установки (рис. 4, п. 3) и, наконец, установка, завершающаяся приглашением ввести новую команду (рис. 4, п. 4).

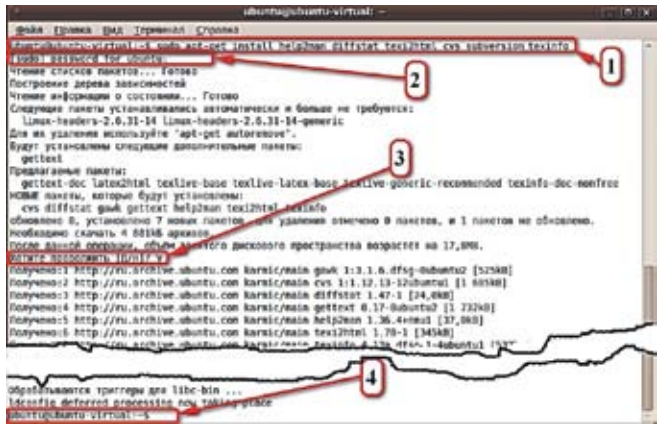

Рис. 4. Установка дополнительных пакетов

Можно, но необязательно, инсталлировать пакет Psyco — компилятор, для повышения скорости работы BitBake. Для его установки запустите от имени суперпользователя, как это показано на рис. 5, команду ***apt-get install python-psyco***.

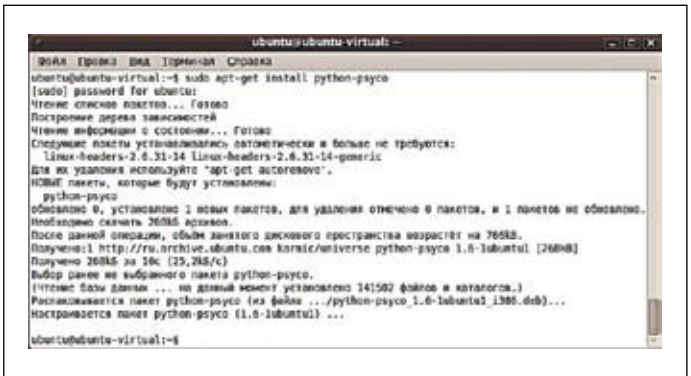

Рис. 5. Установка компилятора Psyco

При использовании дистрибутива Ubuntu вероятнее всего, что ссылка “/bin/sh” связана с “/bin/dash”. Необходимо связать ссылку “/bin/sh” с “/bin/bash”. Игнорирование этого приводит к искажению файла инициализации создаваемой системы, и полученный в результате компилирования образ не будет загружаться! Выполните от имени суперпользователя команду ***dpkg-reconfigure dash*** (рис. 6).

Рис. 6. Реконфигурация символьных ссылок

В процессе выполнения команды появится окно, вид которого приведен на рис. 7, где необходимо выбрать «Нет» при ответе на вопрос, не хотите ли вы установить dash как “/bin/sh”.

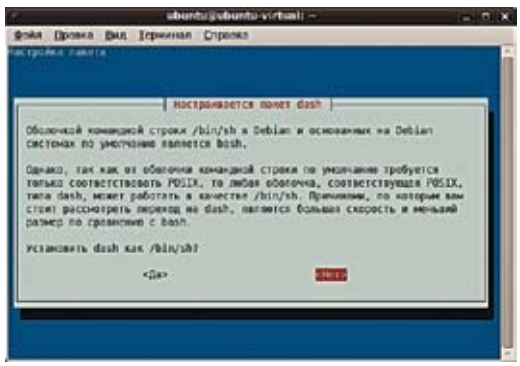

Рис. 7. Настройка пакета dash

Следующий шаг — создание в домашней директории папки, где будут размещаться все данные, необходимые для создания дистрибутива, а также сам результат работы: образы ядра, загрузчика и файловой системы.

В рассматриваемом примере создание дистрибутива проходит в домашней директории пользователя с именем “ubuntu”. Папка, где будут располагаться все рабочие материалы, называется “oe”. Для ее создания необходимо запустить команду ***mkdir -p oe***.

Теперь необходимо перейти в созданную папку. Это выполняется командой ***cd oe***. На рис. 8 показано, как это сделать.

Рис. 8. Создание рабочей директории и переход в нее

Следующее, что требуется, — получить исходные файлы для сборки собственной системы Linux (около 250 Мбайт). Если сервер не очень загружен и есть широкополосный доступ в сеть, этот шаг может занять не менее 15 минут. Необходимо как минимум 10 Гбайт свободного пространства на жестком диске для того, чтобы сделать полную сборку ядра и образа корневой файловой системы (именно по этой причине необходимо было выделить не менее 15 Гбайт под виртуальную машину).

В рамках пакета OE существует несколько проектов создания дистрибутивов для встроенных систем. В статье рассматривается построение системы Arago [8]. Получить специфические файлы системы Arago можно запустив команду ***git clone http://arago-project.org/git/arago.git*** (рис. 9).

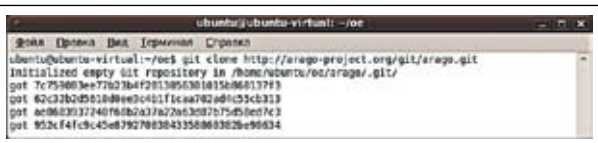

Рис. 9. Получение специфических файлов системы Arago

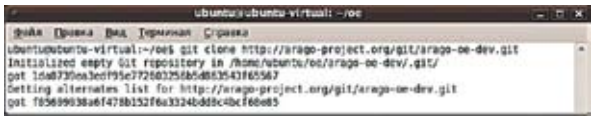

Рис. 10. Получение файлов OE для Arago

Процесс занимает 5–10 мин и заканчивается приглашением ввести новую команду.

Теперь необходимо получить файлы OE для Arago (рис. 10): ***git clone http://arago-project.org/git/arago-oe-dev.git***.

После загрузки всех компонентов (а это займет более 10 мин) вновь появится приглашение ввести новую команду.

И, наконец, необходимо получить файлы BitBake для Arago. Процесс протекает достаточно быстро — 2–3 мин. Для его запуска нужно выполнить следующую команду (рис. 11): ***git clone http://arago-project.org/git/arago-bitbake.git***.

Рис. 11. Получение файлов BitBake для Arago

На этом процесс получения необходимых исходных файлов закончен. В результате в директории “oe” должна быть сформирована структура папок, показанная на рис. 12.

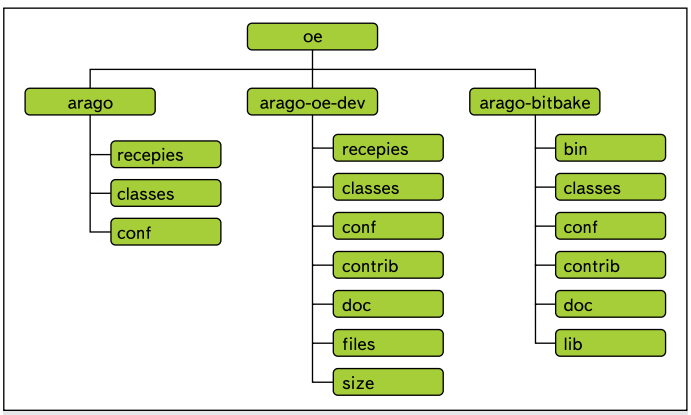

Рис. 12. Структура папок с исходными файлами системы Arago

Следующий шаг — получение и установка кросс-компилятора для непосредственного преобразования исходных файлов в требуемые образы ядра, файловой системы и загрузчика формируемого дистрибутива Linux.

Вначале создадим директорию, где будет располагаться кросс-компилятор. Процедура такая же, как и при создании директории “oe”. Необходимо выполнить следующую команду от имени суперпользователя (рис. 13): ***mkdir /opt***.

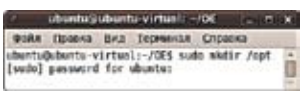

Рис. 13. Создание директории для установки кросс-компилятора

Рис. 14. Получение установочного файла кросс-компилятора

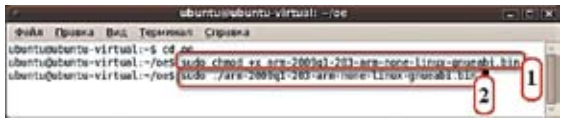

Рис. 15. Установка кросс-компилятора

Обратите внимание, что директория “opt” может существовать, так как является достаточно типичной для различных дистрибутивов Linux и автоматически создается при их инсталляции. В этом случае повторно создавать ее не надо.

Теперь нужно получить инсталляционный бинарный файл кросс-компилятора для процессоров на базе архитектуры ARM. Это можно сделать командой (рис. 14): ***wget http://www.codesourcery.com/sgpp/lite/arm/portal/package4573/public/arm-none-linux-gnueabi/arm-2009q1-203-arm-none-linux-gnueabi.bin***.

Объем файла большой, поэтому его загрузка может занять более получаса. После получения нужно установить кросс-компилятор в директорию “opt”. Но вначале требуется сделать загруженный файл исполняемым. Эта процедура выполняется командой (рис. 15, п. 1): ***chmod +x arm-2009q1–203-arm-none-linux-gnueabi.bin***.

Теперь нужно запустить инсталляцию кросс-компилятора командой (рис. 15, п. 2): ***./chmod +x arm-2009q1–203-arm-none-linux-gnueabi.bin***.

Обратите внимание, что команды выполняются от имени суперпользователя.

Последняя команда запустит инсталлятор кросс-компилятора с графическим интерфейсом (рис. 16).

Мастер установки прост и понятен, необходимо только внимательно читать сопровождающие процесс установки инструкции. Вначале подтвердите свое согласие с лицензионным соглашением, установив галочку в соответствующем пункте и нажав кнопку Next (рис. 17).

Затем выберите тип инсталляции Custom и снова нажмите кнопку Next (рис. 18).

Выбрать указанный тип установки необходимо для того, чтобы иметь возможность определить, где будет инсталлирован кросс-компилятор. В тот момент, когда процесс дойдет до выбора директории установки компилятора (рис. 19, п. 1), необходимо нажать кнопку “Choose…” (рис. 19, п. 2).

В появившемся окне нужно выбрать директорию, где должна размещаться папка с компилятором (рис. 20, п. 1), затем нажать кнопку с изображением домика (рис. 20, п. 2) для создания папки и в появившемся пустом прямоугольнике набрать требуемое имя (рис. 20, п. 3). В рассматриваемом примере необходимо набрать имя “arm-2009q1” и нажать «Ввод». Окончательно подтвердить выбор папки нужно нажатием клавиши Select (рис. 20, п. 4).

Рис. 17. Подтверждение лицензионного соглашения

Рис. 18. Выбор типа инсталляции

Рис. 19. Выбор произвольной директории установки

Рис. 16. Запуск графической оболочки инсталлятора кросс-компилятора

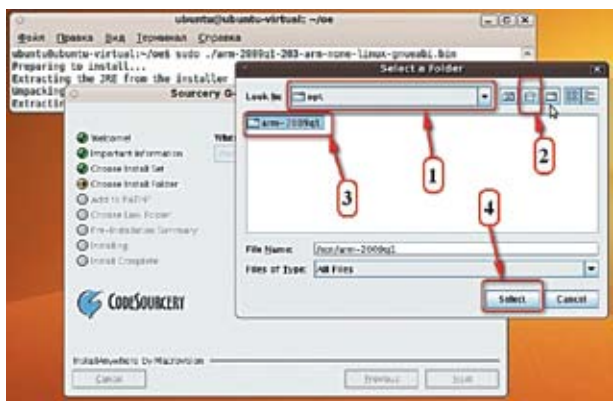

Рис. 20. Создание директории установки

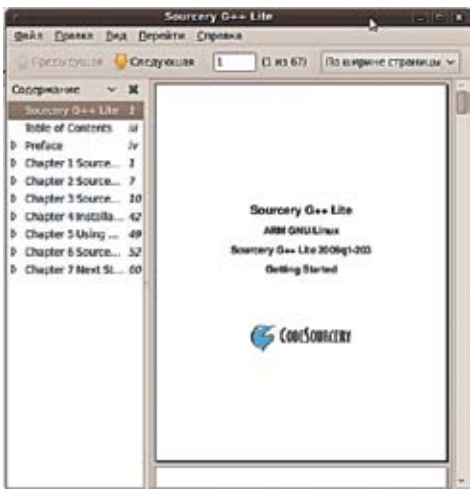

Рис. 21. Загрузка «Руководства пользователя»

Дальнейшая инсталляция не должна вызвать затруднений, так как можно согласиться с предложенными по умолчанию настройками. Завершается процесс установки кросс-компилятора подключением к Интернету и загрузкой «Руководства пользователя» (рис. 21).

Рис. 22. Загрузка дополнительного инструментария для Arago

Следующий шаг — получение дополнительного инструментария для создания дистрибутива Arago, что выполняется командой (рис. 22) ***wget http://arago-project.org/files/short-term/extras/arago-csl-sdk.tar.bz2***.

Рис. 23. Распаковка дополнительного инструментария

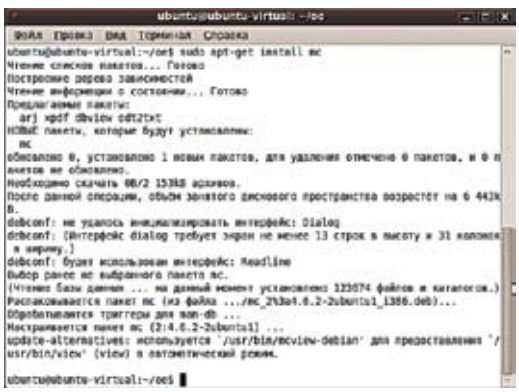

Рис. 24. Инсталлирование файлового менеджера

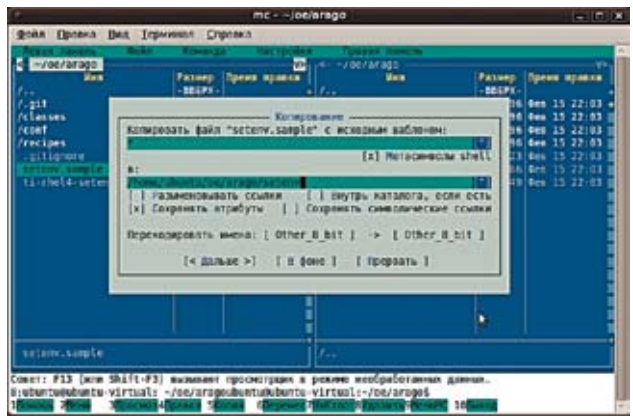

Рис. 25. Инсталлирование файлового менеджера

После закачки архива его необходимо распаковать в директорию установки кросс-компилятора следующей командой с правами суперпользователя (рис. 23): ***tar xjf arago-csl-sdk.tar.bz2 –C/opt/arm-2009q1/***.

Для удобства дальнейшей работы рекомендуется установить файловый менеджер «mc». Он устанавливается от имени суперпользователя следующей командой (рис. 24): ***apt-get install mc***.

Теперь запустите его командой “mc”, зайдите вначале в папку “/home/ubuntu/oe/arago” и сделайте в этой же папке копию файла “setenv.sample” с новым именем “setenv” (рис. 25), для чего выделите его и нажмите мышкой кнопку «5» внизу файлового менеджера. Появится окно копирования файлов, где указывается, куда и с каким именем необходимо поместить файл. Укажите требуемое имя и нажмите кнопку «Дальше» (рис. 25).

Точно такую же операцию проделайте для файла “local.conf.sampele”, расположенного в папке “/home/ubuntu/oe/arago”, сделав копию с именем “local.conf”.

Как нетрудно догадаться, эти файлы являются примерами конфигурационных файлов для процесса создания собственного дистрибутива Linux для встраиваемых систем. После выполнения операций копирования закройте файловый менеджер, нажав кнопку «10» внизу.

А теперь настроим переменные окружения. Первое, что необходимо сделать, — это отредактировать файл “/etc/sysctl.conf”, установив значение “vm.mmap_min_addr = 0”. Это удобно выполнить в текстовом редакторе файлового менеджера, поэтому запустите его с правами суперпользователя, как это показано на рис. 26.

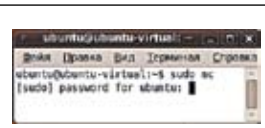

Рис. 26. Запуск файлового менеджера с правами суперпользователя

Затем перейдите в папку “/etc/”, выделите файл “sysctl.conf” и откройте его в текстовом редакторе, нажав мышкой кнопку «4». В конце файла допишите строку “vm.mmap_min_addr = 0” (рис. 27), сохраните изменения, нажав сочетание клавиш “Ctrl+O”, и подтвердите изменение файла, нажав клавишу «Ввод». Затем выйдите из текстового редактора, нажав одновременно “Ctrl+X”.

Теперь закройте файловый менеджер, терминал и перезагрузите систему, чтобы изменения вступили в силу.

После перегрузки необходимо снова запустить файловый менеджер, перейти в папку “/home/ubuntu/oe” и выполнить следующие команды настройки оставшихся переменных окружения: ***export OEBASE=/home/ubuntu/oe; export PATH=/opt/arm-2009q1/bin:$PATH; source arago/setenv***.

Рис. 27. Редактирование файла "sysctl.conf"

Обратите внимание, что три последние команды необходимо вводить каждый раз, после перегрузки системы.

После этого введите и запустите команду тестирования ***bitbake***.

Все эти команды приведены на рис. 28.

Рис. 28. Настройка переменных окружения и запуск теста

Если процесс установки был выполнен корректно, тогда, спустя некоторое время, должна появиться информация, показанная на рис. 29.

Рис. 29. Успешное прохождение теста

На этом процесс формирования среды компиляции дистрибутива Linux для встраиваемых систем завершен. Планируется в следующих статьях данного цикла рассмотреть, как изменить и настроить файлы конфигурации для получения дистрибутива под конкретную платформу с требуемыми свойствами.

В заключение хотелось бы выразить огромную благодарность Илье Чепурину, инженеру технической поддержки московского представительства компании Texas Instruments, который консультировал автора при написании данного материала.

Литература

1. Операционная система Linux на отладочной плате BeagleBoard // Компоненты и технологии. 2010. № 1.
2. www.beagleboard.org
3. http://docs.openembedded.org/usermanual/usermanual.html
4. http://bitbake.berlios.de/manual/
5. http://www.virtualbox.org/
6. http://wiki.openembedded.net/index.php/OEandYourDistro
7. http://packages.debian.org/ru/
8. http://arago-project.org/git/people/vaibhav/ti-psp-omap-video.git?p=people/vaibhav/ti-psp-omap-video.git;a=summary