Report

Rodrigo Siqueira de Melo

March 16, 2014

Contents

1 Informations

1.1 About this report

In the class the professors gave the homework of make some basic changes in the Linux Kernel. In my case, I already use Debian as my main operating system and I don't wanna to make some changes in my main system because I want to avoid to destroy my system. Based on this idea I tried to install Linux in my flash drive, however this task isn't so easy that I expected.

In this report I described each step that I used, however some parts of my steps still unstable. Each means, that sometime works like I expected and another no.

1.2 Basic environment and test

For execute my steps, you need to have Linux for execute each step. Additionally, instead of add my name I add the name "Xenomai" at my kernel because of some special configurations that I made for real time operating system.

2 Prepare the environment

2.1 Basic package

When we want to install Linux in the flash drive, we need some applications for help us in this achievement. Fallow below the list of the main packages that you will need.

```
# apt-get install mbr
# apt-get install syslinux
# apt-get install extlinux
# apt-get install debootstrap
# apt-get install qemu
```

You need to be aware that some extra package maybe will be necessary. This can vary between system.

2.2 Prepare flash drive

1. Plug your flash drive in your computer.
2. Check if your Linux recognize your device, for achieve this objective you can type:

   ```
   dmesg | tail
   ```

   After you execute the command above, you will see some messages log. You need to ready carefully and identify which device was assign to your flash drive. The output of the command below will be similar to:

```
[   12.159263] scsi 6:0:0:0: Direct-Access     SanDisk
    Cruzer Blade     1.26 PQ: 0 ANSI: 6
[   12.160741] sd 6:0:0:0: Attached scsi generic sg2 type 0
[   12.162429] sd 6:0:0:0: [sdc] 15633408 512-byte logical
    blocks: (8.00 GB/7.45 GiB)
[   12.164188] sd 6:0:0:0: [sdc] Write Protect is off
[   12.164194] sd 6:0:0:0: [sdc] Mode Sense: 43 00 00 00
[   12.164503] sd 6:0:0:0: [sdc] Write cache: disabled,
    read cache: enabled, doesn't support DPO or FUA
[   12.189375]  sdc:
[   12.191908] sd 6:0:0:0: [sdc] Attached SCSI disk
[   18.651908] FAT-fs (sdc): utf8 is not a recommended IO
    charset for FAT filesystems, filesystem will be case
    sensitive!
[   19.469279] eth0: no IPv6 routers present
```

As you can see in the output above, my flash drive is referenced by */dev/sdc*.

3. Now we need to format the flash drive, for this task we will put 0 in all the device. This is import for keep the integrity of the device. Execute the command below:

```
# dd if=/dev/zero of=/dev/sdX
```

The command *dd* means ¡¡XYZ¿¿, and we use it for format the flash drive. The attribute *if* after the command *dd* means "input file" and we indicate that we want to use */dev/zero*. Finally the attribute *of* means "output file", in the case the flash drive (change *sdX* to the correspond device for your flash drive). After execute the command, you will see something similar to:

```
writing to '/dev/sdc': No space left on device
15633409+0 records in
15633408+0 records out
8004304896 bytes (8.0 GB) copied, 1967.9 s, 4.1 MB/s
```

4. Now, we need to create the partitions and for this task we will use the *fdisk* (You are free for use any tool that you want). First of type the command:

```
# fdisk /dev/sdX
```

Change *sdX* to the correct name of your device. In the main screen of *fdisk* fallow the steps below:

(a) Create new partition: Type ***n*** and then press ***ENTER***.

(b) Type ***p*** for create a primary partition and followed by the key ***ENTER***.

(c) In the number of partition type ***1*** and then press ***ENTER***.

(d) When you are inquired about the first and last cylinder type ***ENTER*** respectively.

(e) After the last step, you will come back to the main menu of ***fdisk***.

(f) Bootable partition: type ***a***, then ***1*** and finally ***ENTER*** for make the partition bootable.

(g) Before apply the new changes for your device, check if everything is right. For doing this, type ***p*** in the main menu of ***fdisk*** and check the informations that will be displayed. You will see something similar to:

```
Disk /dev/sdc: 8004 MB, 8004304896 bytes
247 heads, 62 sectors/track, 1020 cylinders, total
    15633408 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512
    bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x106394be

Device Boot      Start         End      Blocks   Id
     System
/dev/sdc1   *        2048    15633407     7815680
    83  Linux
```

(h) Finally, type ***w*** for apply all your changes to the device.

(i) If you wanna check, after the command above you can type *fdisk /dev/sdX -l*.

5. Now we need to create one ***Master Boot Record***, as know as ***MBR***. Type:

```
# install-mbr /dev/YYY
```

This command doesn't display any output.

6. At this step, we need to format the partition to ***ext2***[1]. Type:

```
# mkfs.ext2 /dev/YYY1 -L ROOT
```

Notice the use of the flag ***L***, it means that flash drive label will be assign with the name ***ROOT***. This step is important for the correct boot of your system.

7. The last step of preparing the flash drive is prepare the boot system. First of all type the command:

[1]We use ***ext2*** for avoid the journaling, because this isn't good for flash drive

```
# mount /dev/YYY1 /mnt
```

If you want, you can mount your device in any place that you feel comfortable. After that it is necessary to create the boot folder and install the ***extlinux***, type the commands below:

```
# mkdir -p /mnt/boot/extlinux
# extlinux -i /mnt/boot/extlinux
```

3 Install Debian, compile all the source

For install one new kernel, we have many approaches. On this section I will show how to compile all the kernel source. In the next section, I will show another way to install a new kernel in your system.

3.1 Get Debian with bootstrap

1. We will use ***bootstrap*** for get the basic data about the Debian, and use it for install our distribution in the flash drive. At this point you need a Internet connection and be aware that you will need to wait. You can install many different version of the Debian, look below some options and choose one:

   (a) You can install Wheezy (used in this tutorial):

   ```
   # debootstrap wheezy /mnt http://http.debian.net/
       debian/
   ```

   (b) If you want to install Squezze:

   ```
   # debootstrap --arch i386 squeeze /mnt http://ftp.us
       .debian.org/debian
   ```

   (c) If you want to install Lenny:

   ```
   # debootstrap --arch i386 lenny /mnt http://archive.
       debian.org/debian/
   ```

   (d) If you want to install Etch:

   ```
   # debootstrap --arch i386 etch /mnt http://archive.
       debian.org/debian/
   ```

(e) **ATTENTION:** For this tutorial I choose Wheezy, keep in mind that if you choose another distribution maybe you need to adapt some steps.

2. At this point of the tutorial we will use the command ***chroot***. This software is really powerful, because it change the current root folder of the process to another one. In our case, this is really useful because we can "entry" in the flash drive and install some software, make some configurations, and so forth. Type:

```
chroot /mnt /bin/bash
export PS1="(chroot) $PS1"
```

ATTENTION 1: After the command above, all the commands that you execute in the terminal will occur in the flash drive.
ATTENTION 2: If the command ***chroot*** show some message error, you can overcome the problem with the command:

```
echo 1 > vdso_enable
```

3. Mount the ***proc*** folder, with the command below:

```
# mount -t proc proc /proc
```

4. Set the list file of source, you can find it in ***/etc/apt/source.list***. Keep in mind that this configuration change a lot during the year. I recommend you copy the line below to the file, however be aware that maybe those lines can fail (is your responsibility to keep this line updated). (Wheezy)

```
deb http://ftp.us.debian.org/debian/ wheezy main
deb-src http://ftp.us.debian.org/debian/ wheezy main

deb http://security.debian.org/ wheezy/updates main
    contrib
deb-src http://security.debian.org/ wheezy/updates main
    contrib

# wheezy-updates, previously known as 'volatile'
deb http://ftp.us.debian.org/debian/ wheezy-updates main
     contrib
deb-src http://ftp.us.debian.org/debian/ wheezy-updates
    main contrib

deb http://http.debian.net/debian/ wheezy main contrib
    non-free
deb http://ppa.launchpad.net/webupd8team/java/ubuntu
    precise main
deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu
     precise main
```

(Squeeze)

```
deb http://ftp.us.debian.org/debian/ squeeze main contrib
    non-free
deb http://ftp.br.debian.org/debian/ squeeze-updates main
deb-src http://ftp.br.debian.org/debian/ squeeze-updates
    main
```

After that, type:

```
# apt-get update
```

5. For define the local language type:

```
# apt-get install locales
# dpkg-reconfigure locales
```

Choose the language that fit well for your case.

3.2 Prepare kernel for compile

In this part of the tutorial I will make some basic changes in the kernel, however fell free for use the basic configuration.

1. First of all, is necessary to install some basic package in the flash drive. Type:

```
# apt-get install kernel-package libncurses-dev fakeroot
    zlib1g-dev
# apt-get install zlibc zlib-bin firmware-linux-nonfree
# apt-cache search zlib.h
# apt-get install zlib-bin zlib1g zlib1g-dev zlib1g-dbg
    libio-zlib-perl zlibc zlib-gst
```

2. Change to the folder:

```
cd /usr/src
```

3. Download the kernel source with the command below:

```
# wget https://www.kernel.org/pub/linux/kernel/v3.x/linux
    -3.13.6.tar.xz
```

After that, unzip the file:

```
# tar xvfJ linux-3.13.6.tar.xz
```

4. Create a symbolic link for the directory:

```
# ln -s /usr/src/linux-3.13.6 linux
```

5. Type:

```
#cd linux
```

6. Copy the default file of configuration with the command below:

```
# make defconfig
# make menuconfig
```

7. At this point we have all the pre requirement for compile the kernel.

3.3 Configure kernel

The kernel configuration is something particular for each computer, I will just show some basic options. Fell free for apply or not my suggestions.

- General Setup −− >
  – Local Version - append to kernel release −− >
  Type: You can add extra name for your kernel. Ex.: -Rodrigo

- Processor type and features −− >
  – Choose the specific processor. Avoid to use generic processor, like "generic i586".
  – Enable -fstack-protector buffer overflow detector
  Disable this option.
  – HPET Timer Support
  Disable this option
  – Preemption mode
  Choose preemptive kernel (Low-Latency Desktop).

- Device drives
  – Input device support
  — Miscellaneous device
  PC speaker support - Disable

3.4 Compile the kernel

- After configure your kernel, you must compile it. For it execute the command below:

```
make -j9
```

This command will compile the kernel. The flag *-j?* means how many threads do you want to use for compile. It is recommended to use the double amount of cores that you have.

- Afterward compile the modules:

```
make modules
```

- Finally we will install the modules and the new kernel.

```
make modules_install
make install
```

4 Install Debian, second approach

ATTENTION: If you execute the last section, jump for the next session.

- Check all the kernels available:

```
apt-cache search ^linux-image
```

Choose the option that best fit for your hardware.

- After choose one kernel, execute the command below:

```
apt-get install firmware-linux-nonfree
apt-get install linux-image-686
```

- *ATTENTION:* If the question below be displayed to you: "*Abort initrd kernel image installation?*". Choose No.If you reply Yes, you will need to type the command:

```
#update-initramfs -c -k [Kernel version]
```

5 Result

After follow all the steps described in the last section, I had Debian system installed in my flash drive. As I said in the begin of this report, I change the kernel name to "Xenomai" because of the extra API that I installed in my system.

(a) New kernel name