Step 2 - Work Log
M2M Lectures
Grenoble University

Stephane BERGER- Jean-Yves Bottraud

February 9, 2016

1 Preface by Pr. Olivier Gruber

This document is your work log for the second step in the M2M course, leading to build your own mini-distribution.

Like for the first step, this work log has two parts. One part is about diverse sections, that you must fill up. Each section has a bunch of questions to help you do that. The questions provide a guideline for your learning. They are about giving your pointers on what to learn about. The other part is really a laboratory log, keeping track of what you do, as you do.

REMEMBER: plaggia is a crime that can get you evicted forever from french universities... The solution is simple, write using your own words or quote, giving the source of the quoted text. Also, remember that you do not learn through cut&paste. You also do not learn much by watching somebody else doing.

2 Outline of Step2

To build your own distribution, almost from scratch, you will need to acquire a number of skills:

- GRUB bootloader
- Linux kernel
- Distribution layout and core files

Many pieces are provided to you, pieces that you are asked to understand. First, you are given a compiled, ready-to-use, boot loader (GRUB 0.97). Second, you are given a minimal distribution, with a linux kernel.

You can build a Qemu disk with mkdist.sh sript, following the on-screen instructions. Once you have a disk, you can boot Qemu with it. When booting,

you will get a GRUB menu, with several boot options, one is called "Hello - RootFS". Select that option and you will get something very similar to what you did in step1, but using a real bootloader and a linux kernel.

After that, the learning process will start and it cannot be done entirely as a linear process, sorry. So you will fill up the various sections incrementally, certainly in an order that will be a bit erratic.

We suggest the following steps.

1. Bootable disk and GRUB

   - Read and start understanding the script: mkdist.sh Explain each step, explaining the related concepts such as the loopback mount, mknod files (character and block).

     loopback : pouvoir travailler sur un disque dur sur notre système alors que le disque n'existe pas physiquement. On va créer un fichier (de type block ou charater)rempli de 0 d'une certaine taille . Après on va lier ce fichier à un loop device libre(8 loop device disponible sous linux). Ensuite on peut le formater (mkfs) et le monter (mount). on effectue une simulation d'un disque dur, d'un cd , d'un matériel.
     Block device : (fichier spécial de type bloc) C'est un device file dans lequel le systme transférera les données sous forme de blocs. On a la possibilité de faire un accs non sequentiel au xdonnées (buffering,accès aléatoire) C'est le cas du disque dur.

     Charactere device : Fichier spécial de type caractre = C'est un device file dans lequel les données seront transmises caractre par caractre (sequntiel). exemple : port série

     pour voir les devices dans Unbuntu : ls -l /dev
     b = block device
     $brw-rw----1rootdisk7,0fevr.809:29loop0$
     $brw-rw----1rootdisk1,0fevr.809:29ram0$
     $brw-rw-----+1rootcdrom11,0fevr.809:29sr0$

     $rm : sr0estdetypeblock, c'estlecddesadditonspourvirtualbox$
     c = charactere device
     $crw--w----1roottty4,0fevr.809:29tty0$
   - Read about and understand the GRUB process and how it is installed on a disk and the GRUB boot stages.
     Sur le disque il faut sur le MBR le stage 1 () ensuite on place le Stage 1.5 pour que certains driver spécifique soit chargés si nécessaire. (dans les 30 kilooctets suivant le MBR). Stage 1 charge Stage 1.5 si il est présent sinon il charge Stage 2.La partie 1.5 peut contenir des pilotes pour pouvoir accéder la partie 2. Stage 1.5 charge Stage 2 qui est placé sur .

- Read about and understand the GRUB menu given to you.

2. Hello - RootFS

   - Understand the first boot option of the GRUB menu (Hello - RootFS). Which kernel boots?
     vmlinufz-2.6.31-22-generic (cf menu.lst dans minidist/boot/grub)
     What is the role of initrd.hello?
     le rôle d'initrd est de permettre au chargeur d'amorçage de monter un disque RAM (system de fichier) à partir duquel il est possible de lancer des exécutables et ensuite charger un autre systeme de fichier

     Look at the hello directory and how hello is compiled, linked, and the initrd.hello is constructed. Explain how it works and why.

     le fichier initrd est le rsultat de la commpression de l'executable hello par cpio.
   - Read about and understand the purpose of initial ramdisks (initrd) for booting linux in general, on any machine.

3. Hello - Disk

   - Read and understand the second boot option (Hello - Disk). What is different in the boot process? Explain how this process boots the linux kernel from the disk you created with mkdist.sh.
   - Explain the two console options given to the kernel? Why are they useful? How do you exploit them when booting from Qemu?
   - Why do we use a different kernel for this boot process? Instead of using the generic vmlinuz-2.6.31-22?
   - Compare the different kernel configurations given to you: allnoconfig-2.6.32.65, miniconfig-2.6.32.65, m2m-config-2.6.32.65. (hint: compiling a linux kernel, make allnoconfig, make bzImage) (hint2: look at Section 7)

4. MiniDist Layout

   - Dive into the given MiniDist and explain its layout, giving the purpose of the different directories.
   - Explain the purpose of the few files in those directories. In particular the various files under /dev.
   - What is missing for a realistic minidist? (hint: libraries, commands, shell, configuration files and scripts).

Here are a bunch of questions and points to help you progress and focus your attention.

1. Layout of the Qemu disk, in terms of GRUB stages and file system partition.
2. Partition table in the MBR (see parted and fdisk)
3. What kind of file system are we using. Why only ext2? Could we use something other format? What do we have to pay attention to? Think GRUB and Linux kernel.
4. Look in the GRUB manual how to install GRUB manually. Explain the manual steps during the mkdist.sh. Explain why they are necessary.
5. Explain the loop-back setup with the Qemu disk. Why is it necessary?
6. What are those files under /dev? What is their purpose? How are they created? (hint mkdnod)
7. During the mkdist.sh, we boot Qemu twice in a row, with a manual GRUB setup in between. What is going on? Explain why we stop at the GRUB prompt the first time and why we get a GRUB menu the second time around.
8. Look at the GRUB menu and understand its structure.
9. Look at the linux kernel options in the GRUB menu. What are they for? Why use those options? Hint: console history, see the differences between booting Hello-RootFS and Hello-Disk.
10. In your mini distribution, under /boot, you have several kernels. Regarding the 2.6.31.22-generic, what are the different files related to this version (System.map, config, and vmlinuz).
11. Explain how the linux kernel boots with Hello-RootFS. Hint: initial ramdisk (initrd).
12. Explain how the hello program can actually execute? Indeed, the mini distribution has no libraries. Hint: hello makefile.
13. Try replacing the kernel in the Hello-Disk configuration, from vmlinuz to vmlinuz-2.6.31-22-generic, what is happening? Why? Why is it working with vmlinuz.

3 Qemu

This section is here in case you learn some more stuff about Qemu. You will find plenty of information in the README-QEMU document.

4 MiniDist Script

You ask to read and explain the following script (mkdist.sh).

4.1 script mkdist.sh explanation

1. compilation de l'executale hello et intégration dans une archive d'un systeme de fichier racine (root file systeme) = distribution
2. Création du fichier disque (fichier minidist.img) (secteur de 512 octets) 1 secteur : fichier stage 1 secteur 2 à 198 : fichier stage 2 64 Mo de vide
3. Création de la partition EXT2
4. Montage de la partition
5. Copie de la distribution (archive) sur la partition montée
6. synchronisation (sync - flush file system buffers)
7. Démontage de la partition
8. lancement de minidist.img avec Qemu pour installation et configuration de GRUB (Bootloader)
   - **grub : root (hd0,0)**
     commande : ¡device-type¿¡device-number¿,¡partition¿
     definiton : Configure la partition root de GRUB et monte la partition.
   - **grub : setup (hd0)**
     definition : cette commande installe GRUB sur le MBR du premier disque (hd0).
   - **grub : halt**
     definiton : quitter
9. lancement de minidist.img avec Qemu pour lancer le systeme installé et donc le fichier exécutable Hello

5 GRUB

In the grub directory, you will have binary and the sources for grub-0.97, an older and much simpler GRUB version.
sources infos :
$http : //www.linux - france.org/article/sys/chargeurs/ix86/grub/grub - manual - fr.html\#fn - 1$
$http : //lea - linux.org/documentations/Grub$
$https : //www - uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.2/suselinux - adminguide_en/ch07s04.html$
$https : //docs.oracle.com/cd/E37670_01/E41138/html/ch04s02.html$

5.1 Qu'est-ce que GRUB ?

5.1.1 Présentation

GRUB (ou GNU)est un chargeur de démarrage à l'instar de LILO. Il est extrêmement flexible et peut charger un système sans enregistrer la position physique du noyau sur le disque. Vous pouvez lancer un noyau simplement en précisant son nom de fichier et le disque (et la partition) sur lequel il réside. Pour ce faire, vous pouvez utiliser l'interface ligne de commande ou l'interface menu. Il est inutile de relancer GRUB après une modification de sa configuration. Toute modification du fichier de configuration /boot/grub/menu.lst est immédiatement prise en compte. Rappelons enfin que Linux peut tre installé sur n'importe quelle partition. GRUB utilise la technique standard pour charger un OS ou du chargement chané (chain-loading), pourcharger des systèmes d'exploitation non-supportés en activant un autre chargeur.

5.1.2 Principe

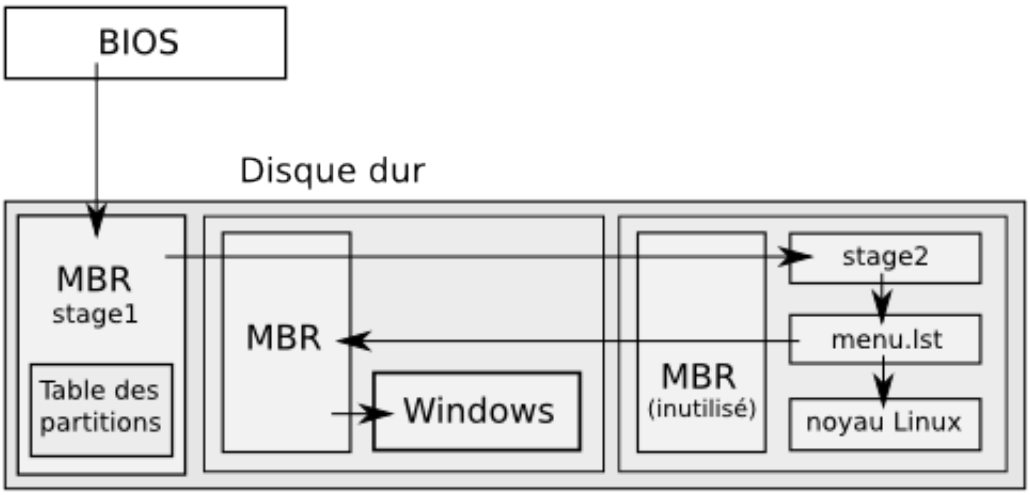

Schéma simplifié d'un dual-boot

5.1.3 rappels

(hd0,1)
Ici, hd signifie qu'il s'agit d'un disque dur. Le premier nombre 0 indique le numéro du disque, qui est ici le premier disque, alors que le second entier 1 indique le numéro de la partition (ou le numéro de slice dans la terminologie BSD). Encore une fois, notez que les numéros de partitions sont déterminés à partir de zéro, et non depuis un. Cette expression désigne la deuxième partition du premier disque dur. Dans ce cas, GRUB n'utilise qu'une partition du disque à la place du disque entier.

root device [hdbias]

Définit le device comme étant la partition racine, puis essaie de la monter pour obtenir la taille de la partition (pour passer le descripteur de la partition en ES:ESI, utilisé par certains chargeurs chanés), le type de disque BSD (pour démarrer des noyaux BSD en utilisant leur format de démarrage natif), et déterminer correctement la partition PC ou se trouve la sous-partition BSD. Le paramètre facultatif hdbias est un nombre pour informer un noyau BSD du nombre de disque BIOS qui se trouvent sur un contrleur avant le disque actuel. Par exemple, s'il y a un disque IDE et un disque SCSI, et que votre partition racine FreeBSD se trouve sur le disque SCSI, utilisez 1 pour hdbias.

5.2 GRUB principles and stages

1. Look at eltorito stage for CD-ROMs.

   $source : https : //fr.wikipedia.org/wiki/El\_Torito$

   (a) Rôle
   El Torito (de son nom anglais complet El Torito Bootable CD Specification) est une extension la norme ISO 9660 sur les CD-ROM décrivant comment un ordinateur peut démarrer - ou "booter" - à partir d'un CD-ROM

   (b) Fonctionnement
   Selon la norme El Torito, le BIOS d'un PC récent peut rechercher un programme de boot sur un CD compatible ISO 9660. Si c'est le cas, le BIOS va alors assigner un numéro de périphérique au lecteur de CD. Ce numéro sera soit 80 (émulation du disque dur), 00 (émulation du lecteur de disquettes) ou un numéro au hasard si aucune émulation n'est nécessaire.

   Les émulations sont principalement utilisées pour permettre à d'anciennes version d'OS de démarrer à partir d'un CD en leur faisant croire qu'il démarrent à partir d'un disque dur ou d'un lecteur de disquette.

2. Look at the various stages for a disk (stage1, stage1.5, stage2).

   (a) stage 1
   Le premier niveau de GRUB se situe dans le MBR. Comme ce premier niveau ne peut être que très petit puisqu'il n'y a que peu de place à disposition dans l'enregistrement d'amorçage matre ou dans les secteurs d'amorçage, la seule tche de ce niveau est de charger le deuxième niveau de GRUB.soit le stage 1.5 ou directement le 2.
   pour le code : /boot/grub/stage1

   (b) stage 1.5
   La partie 1.5 peut contenir des pilotes pour pouvoir accéder la partie 2

   (c) stage 2
   Le deuxième niveau de GRUB fournit les fonctions réelles du chargeur

d'amorçage : entre autres le menu de choix (ou cache l'interpréteur de commandes) et le code du programme avec lequel le contrôle de l'ordinateur est passé au système d'exploitation.. Lors de l'installation du chargeur d'amorçage, l'emplacement physique du disque dur dans lequel se trouve le second niveau est signalé dans le fichier du premier niveau.
pour le code : /boot/grub/stage2
You can find some relevant documentations in the docs directory, in particular the GRUB manual and a great explaination of QEMU disk images.

3. Look in the GRUB manual how to install GRUB manually.
4. Look at GRUB menu.

5.2.1 Building GRUB

— **THIS SECTION IS FOR YOUR INFORMATION ONLY** — ,

Do not rebuild GRUB, unless you have to and you are damn sure. You have been given a working for x86_32 (IA-32).

For your information, be warned that to build this version of GRUB, you will need GCC-3.4. So you can install it on your system, via apt-get install. Let's assume you have downloaded your sources under /M2M/grub/grub-0.97 in your dev guest. You can compile and install like this:

```
        # export CC=gcc-3.4
# cd ~/M2M/grub/grub-0.97
# ./configure prefix=~/M2M/grub/grub
# make
# make install
```

Notice the important setup in the configure step where you specify a path where to install GRUB. Notice this is a local path to your home directory, **DO NOT INSTALL IT ON YOUR MACHINE.** By default, it installs on /boot/grub and this is a really bad idea.

6 GRUB and Linux Boot Process

Look at what is going on in the /boot directory. The idea here is to be able to explain the boot process, starting with GRUB and ending with the linux kernel booting.

You must understand the three options in the GRUB menu.

1. Hello - RootFS
2. Hello - Disk
3. Your Mini Distribution

Each represents a different boot strategy.

7 Linux Kernel Configurations

You are given three configurations:

1. allnoconfig-2.6.32.65
2. miniconfig-2.6.32.65
3. m2m-config-2.6.32.65.

All three for the version 2.6.32.65 of the linux kernel.

The all-no-config is generated with the "make allnoconfig" from a kernel source directory. It corresponds to the absolutely smallest linux kernel, with all optional code turned off, but it is no longer functional.

The mini-config is a functional kernel, corresponding to the kernel given in the MiniDist, called vmlinuz. You are asked to compare and explain the differences between the all-no-config and the mini-config.
(hint: a usefull program for this is called meld.)
(hint: the differences are related to the minimal hardware support, like adding a pci bus support).

Then you are asked to do the same between the mini-config and the m2m-config. The m2m-config is intended for a kernel specifically compiled for our target distribution: small, booting directly from disk, and with network connectivity.

You are asked the Qemu configuration that would correspond to booting this kernel. (hint: look at README-QEMU).

8 Mini-Distribution

You are given this layout for your mini-distribution.

```
MiniDist$ ls
bin  boot  dev  etc  hello  lib  root  sbin
MiniDist$
```

Explain each directory and each directory's contents.

If you add anything, you will document what you add and why.

9 Laboratory Log