



ETNA  
Projet de Fin d'Étude 2005-2007  
RimElse  
Cahier des charges technique

© Copyleft 2006, ELSE Team

26 avril 2006

# Table des matières

<b>1</b>	<b>Norme de codage</b>	<b>3</b>
1.1	Généralités . . . . .	3
1.1.1	Nommage . . . . .	3
1.1.2	Présentation . . . . .	3
1.2	Commentaires . . . . .	3
1.2.1	Entête des fichiers . . . . .	3
1.2.2	Fichier Ruby . . . . .	4
1.2.3	Autres langages . . . . .	4
<b>2</b>	<b>Séquence de démarrage</b>	<b>5</b>
2.1	Chargeur d'amorçage . . . . .	5
2.2	Initramfs . . . . .	5
2.3	Reconnaissance matérielle . . . . .	6
2.4	Init script . . . . .	6
2.5	Bootsplash . . . . .	6
2.6	Démarrage du serveur X . . . . .	6
2.6.1	Utilisation du framebuffer . . . . .	6
2.6.2	Sélection des drivers (optionnel) . . . . .	6
<b>3</b>	<b>Gestionnaire de paquets (rebgen)</b>	<b>7</b>
3.1	Structure du binaire . . . . .	7
3.2	Identification des patches . . . . .	7
3.3	Gestion des dépendances . . . . .	7
3.4	Application des patches . . . . .	7
<b>4</b>	<b>Gestionnaire de configurations</b>	<b>8</b>
4.1	Langage . . . . .	8
4.2	Gestionnaire de paquets . . . . .	8
4.3	Éditeur de configurations . . . . .	8

# Introduction

Ce document fait suite au cahier des charges fonctionnel de la distribution RimElse, rédigé par la Else Team.

Après avoir décrit dans le précédent document les fonctionnalités générales de RimElse, celui-ci expose les différents paramètres intervenant dans son élaboration.

Pour ce faire, ce document décrit dans un premier temps la norme de codage utilisée dans les différentes productions. Puis dans un second, il définit les étapes de la séquence de démarrage. La troisième partie, analyse la gestion des paquets et les développements nécessaires à sa réalisation. Pour finir, ce document expose les caractéristiques de l'implémentation du gestionnaire de configuration.

# Chapitre 1

## Norme de codage

### 1.1 Généralités

#### 1.1.1 Nommage

#### 1.1.2 Présentation

Indentation

### 1.2 Commentaires

#### 1.2.1 Entête des fichiers

Identification du fichier

```
File : rim/else.rb
Description rapide du fichier
Projet : RimElse
```

**Inclusion de la GNU GENERAL PUBLIC LICENSE (GPL)**

Copyright (C) 2006 Else Team

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License

along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

### 1.2.2 Fichier Ruby

Pour les fichiers correspondant à du code Ruby, nous utiliserons le générateur de documentation “rdoc”<sup>1</sup>.

Rdoc générera à partir du code source une documentation sous forme de fichiers HTML. Rdoc extraira automatiquement les définitions des classes, modules, méthodes et attributs. Il générera aussi l’arbre des différentes classes et de leurs dépendances.

Les commentaires suivants devront donc être ajoutés, pour les :

- Classes
  - # Author:: Else Team
  - # Copyright:: Copyright (c) 2006 Else Team
  - # License:: GNU General Public License version 2 (GPLv2)
  
  - # Description de la classe
- Fonctions

### 1.2.3 Autres langages

---

<sup>1</sup><http://rdoc.sourceforge.net/>

## Chapitre 2

# Séquence de démarrage

### 2.1 Chargeur d'amorçage

Le démarrage du système requiert un chargeur d'amorçage (bootloader) qui s'exécute hors système d'exploitation. Il s'agit d'un programme appelé par le BIOS qui charge l'image du noyau d'un système d'exploitation dans la RAM.

Dans le cadre de la RimElse, le chargeur d'amorçage doit être présent dans les premiers secteur de notre support (clé USB, CD-ROM, ...). Celui-ci doit supporter les systèmes de fichier FAT et iso9660.

isolinux, syslinux, grub, lilo

### 2.2 Initramfs

Le noyau Linux version 2.6 que nous allons utiliser dispose de trois solutions pour monter son "root".

La première, basique, s'effectue sans Initial Ramdisk (initrd). Pour monter sa partition root et exécuter /sbin/init, le noyau va chercher dans les paramètres qui lui auront été fournis par le bootloader.

Ce type d'initialisation se restreint à une architecture matérielle précise. Pour pallier à ce problème, l'initrd, est chargé au démarrage par le noyau. Dans ce système de fichier en mémoire vive, le noyau va exécuter /linuxrc. Celui-ci va s'occuper du montage du root, en fonction de l'architecture physique et lancer /sbin/init.

Le noyau 2.6 apporte une nouvelle solution. L'initramfs ajoute à l'initrd les avantages de la compression dans une archive cpio.

## **2.3 Reconnaissance matérielle**

## **2.4 Init script**

## **2.5 Bootsplash**

## **2.6 Démarrage du serveur X**

### **2.6.1 Utilisation du framebuffer**

### **2.6.2 Sélection des drivers (optionnel)**

## Chapitre 3

# Gestionnaire de paquets (rebgen)

### 3.1 Structure du binaire

Quelques petits bench pour comparer les formats de compression. Banc d'essais réalisés sur le même répertoire compressé avec les 3 méthodes, ce répertoire fait 21 Mo.

Compression	Gzip	
Sans affichage	tar czf PFE.tar.gz PFE 1,54s user 0,08s system 98% cpu 1,650 total	tar c
Avec affichage	tar czvf PFE.tar.gz PFE 1,66s user 0,10s system 31% cpu 5,651 total	tar cj
Taille du fichier	12 Mo	
Décompression	Gzip	
Sans affichage	tar xzf PFE.tar.gz 0,26s user 0,09s system 48% cpu 0,723 total	tar xjf PF
Avec affichage	tar xzvf PFE.tar.gz 0,27s user 0,12s system 14% cpu 2,799 total	tar xjvf PF

### 3.2 Identification des patches

### 3.3 Gestion des dépendances

### 3.4 Application des patches



## Chapitre 4

# Gestionnaire de configurations

4.1 Langage

4.2 Gestionnaire de paquets

4.3 Éditeur de configurations

# Conclusion

# Bibliographie

# Table des figures