



公司内部		
第	0.2	版
共	15	页

## Rocky6.0.42 版本构建环境搭建

项目名称: Rocky6.0.42 版本构建方法

项目编号: 项目编号

文档类别: 需求/设计/测试/用户

密 级: 公司内部

版 本 号: 0.2

创 建 人: 韦帅

建立日期: 2015-03-20

批 准 人:

批准日期: XXXX-XX-XX

保 管 人: 高严

存放位置: svn://.....

编辑软件: LibreOffice 4.0

## 文档修订记录

VER	REV	修订人	修订日期	简要说明	评审人员	批准日期
V0.1	C	韦帅	2015-03-20	初次创建		
V0.2	M	韦帅	2015-03-24	添加 250 服务器配置		
V0.3	M	韦帅	2015-03-25	增加服务器环境概括图		
VER (版本编号) : V——版本编号; R——修订编号。						
REV (修订状态) : C——创建; A——增加; M——修改; D——删除。						

# 目 录

<b>1 引言</b> .....	<b>1</b>
1.1 概述.....	1
1.2 适用范围.....	1
<b>2 服务器环境概括</b> .....	<b>1</b>
<b>3 基础服务器环境搭建</b> .....	<b>2</b>
3.1 系统环境介绍.....	2
3.1.1 硬件要求.....	2
3.1.2 操作系统要求.....	2
3.2 系统配置.....	2
3.2.1 修改 grub 默认启动项.....	2
3.2.2 创建用户.....	2
3.2.3 启动服务.....	2
3.3 目录结构介绍.....	3
3.3.1 /home/builder/\${arch}: .....	3
3.3.2 /home/builder/daily-builds/4.2-\${arch}-builds.....	4
3.3.3 /home/builder/kvm_autocreate.....	4
3.3.4 /home/builder/stable-iso/4.2-\${arch}-builds.....	5
3.3.5 /home/builder/rocky6.0.42-test.....	5
<b>4 编译器环境搭建（虚拟机）</b> .....	<b>6</b>
4.1 系统环境准备.....	6
4.1.1 硬件要求.....	6
4.1.2 操作系统要求.....	6
4.2 系统配置.....	6
4.2.1 修改 grub 默认启动项.....	7
4.2.2 挂载公共目录.....	7
4.2.3 配置密钥.....	7
4.3 相关 git 库环境构建.....	8
4.3.1 准备 4.2 自动编译工具.....	8
4.4 测试及配置.....	8

表格

插图

# 1 引言

## 1.1 概述

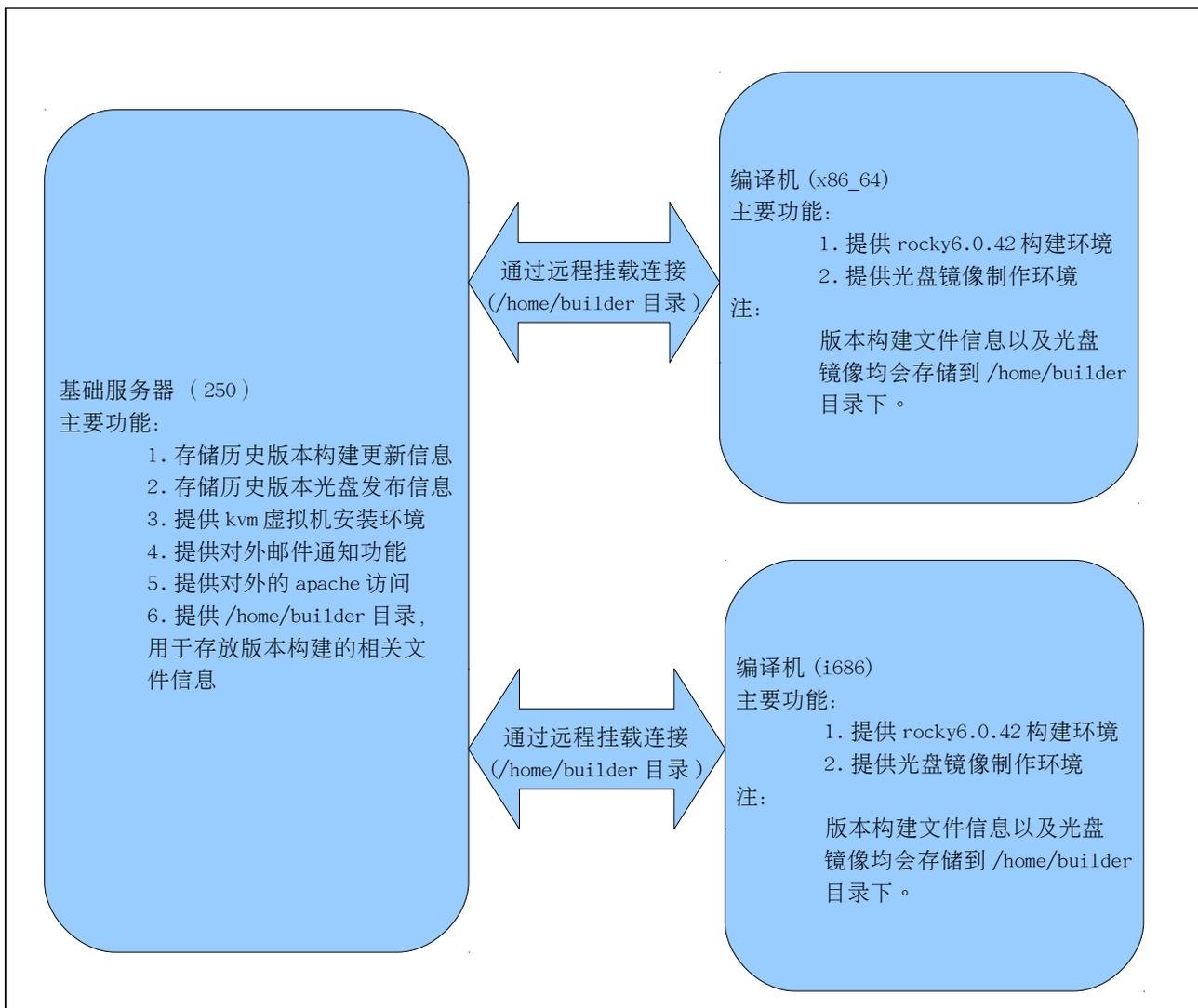
本文档，完整描述了 Rocky6.0.42 版本构建环境的搭建流程，用以今后 Rocky6.0.42 版本构建环境的搭建基准。

## 1.2 适用范围

本文档仅适用于 Rocky6.0.42.41 以及以后的版本

# 2 服务器环境概括

服务器环境及功能介绍如下图所示：



## 3 基础服务器环境搭建

### 3.1 系统环境介绍

#### 3.1.1 硬件要求

内存：4G 以上

硬盘：1TB 以上

#### 3.1.2 操作系统要求

操作系统：linux6.0.60.4

（本次使用系统版本为 linux6.0.60.4-x86\_64-2014-09-02）

操作系统的功能要求：

1. kvm 虚拟机支持

网络情况：

对网络进行配置，保证网络畅通（建议进行双网卡绑定配置）

（本次网络配置地址为：172.16.0.250）

## 3.2 系统配置

本节,讲述了对操作系统环境的部分必要的配置内容

### 3.2.1 修改 grub 默认启动项

设置默认 grub 启动为有 root 模式

```
1. #vim /etc/default/grub
```

```
将 GRUB_DEFAULT=1 修改为 GRUB_DEFAULT=0
```

```
2. #update-grub
```

### 3.2.2 创建用户

创建普通用户 Auto-builder 用于，邮件发送

### 3.2.3 启动服务

执行命令

```
#chkconfig postfix on
```

启动 postfix 服务以支持邮件的发送

### 3.3 目录结构介绍

本节，对服务器上 rocky6.0.42 版本构建过程中涉及到的目录进行了简单的介绍

#### 3.3.1 /home/builder/\${arch}:

Rocky6.0.42 版本更新记录目录：

目录内存放了对应体系结构的，每次更新对应的以日期和时间命名的目录，以 2015-03-24-005001 更新为例，其内目录结构如下：

```
root@linux-info: /home/builder/x86_64/2015-03-24-005001# tree
.
├── cfg
│   ├── iso_env
│   │   ├── additional_pkgs
│   │   ├── auto_build
│   │   ├── git_manage
│   │   └── manipulating_build
│   ├── sys_env
│   │   ├── additional_pkgs
│   │   ├── auto_build
│   │   ├── git_manage
│   │   ├── libI686Pkglist
│   │   └── manipulating_build
│   └── xmlDir
│       ├── dependency.xml
│       ├── install_cfg.xml
│       ├── install-d5000.xml
│       └── install.xml
├── commitRecord
│   ├── applications
│   ├── kernel
│   ├── liblinxsn
│   ├── linux-firmware
│   ├── linx-app
│   ├── linx-serialnumber
│   ├── manipulating_build_pkgs
│   ├── new_install
│   └── StateGrid
├── pkgRecord
│   ├── base
│   └── security
│       └── update-drivers#20150319-x86_64-linx2-Rocky4.2.pkg.tar.gz
8 directories, 23 files
root@linux-info: /home/builder/x86_64/2015-03-24-005001#
```

- `cfg/iso_env`：存放了光盘环境的软件包列表信息
- `cfg/sys_env`：存放了系统环境的软件包列表信息
- `xmlDir`：系统安装过程中使用到的 xml 配置文件
- `pkgRecord/ {base/security}`：本次更新的软件包 {base 版/security 版}

### 3.3.2 /home/builder/daily-builds/4.2- $\{\text{arch}\}$ -builds

Daily-build 版本发布目录：

目录内存放了对应体系结构的以发布日期命名的目录，其内存放了对应发盘日期的光盘镜像以及相关文档信息，以 20150312 为例，其目录结构如下：

```
root@linx-info: /home/builder/daily-builds/4.2-x86_64-builds/20150312# tree
.
├── build_cfg
│   └── relate_build.list
├── buildenv_info
│   ├── libarchive.buildenv
│   ├── pkgutils.buildenv
│   └── sed.buildenv
├── build_log
│   ├── applications_build.log
│   ├── get_config_formdep.log
│   ├── get_install_xml.log
│   ├── get_mkiso_info.log
│   ├── mk_iso.log
│   └── pkg
│       ├── libarchive.log
│       ├── pkgutils.log
│       └── sed.log
├── changelog
├── git_log
├── MD5SUMS
├── pkgs
│   ├── base
│   └── security
│       └── pkgutils#5.31.0-x86_64-linux11-Rocky4.2.pkg.tar.gz
└── rocky6.0.42.41-x86_64-security-20150312.iso

7 directories, 17 files
root@linx-info: /home/builder/daily-builds/4.2-x86_64-builds/20150312#
```

- build\_cfg/relate\_build.list:
- buildenv\_info/ $\{\text{pkg}\}$ .buildenv: 编译软件包 $\{\text{pkg}\}$ 时的编译环境需求
- build\_log: 目录下存放了构建过程中的所有的 log 信息文件
- pkgs/ $\{\text{base/security}\}$ : 其内存放了，本次更新对应更新的软件包二进制

### 3.3.3 /home/builder/kvm\_autocreate

目录内存放了各体系结构最近几次更新的对应的虚拟机硬盘镜像，其目录结构如下：

```
root@linux-info: /home/builder/kvm_autocreate# tree
.
├── kvmimage
│   ├── 20141011.img
│   ├── kvm_install.img
│   ├── rocky6.0.42.41-i686-security-20150320.img
│   ├── rocky6.0.42.41-i686-security-20150321.img
│   ├── rocky6.0.42.41-i686-security-20150324.img
│   ├── rocky6.0.42.41-x86_64-security-20150321.img
│   ├── rocky6.0.42.41-x86_64-security-20150323.img
│   └── rocky6.0.42.41-x86_64-security-20150324.img
└── mnt
    ├── 3367
    └── 3367_iso

4 directories, 8 files
root@linux-info: /home/builder/kvm_autocreate#
```

注：由于考虑到占用空间的问题，因此只保留最近的三个版本的硬盘镜像

### 3.3.4 /home/builder/stable-iso/4.2- $\{\text{arch}\}$ -builds

Rocky6.0.42 正式版本发布目录：

其内存放了，对应体系结构的对应正式发布的光盘镜像信息及相关文档，其目录结构如下：

```
root@linux-info: /home/builder/stable-iso# tree
.
├── 4.2-i686-builds
│   ├── 20140829-2 -> /home/builder/daily-builds/4.2-i686-builds/20140829-2/
│   ├── 20141106 -> /home/builder/daily-builds/4.2-i686-builds/20141106/
│   ├── 20141113 -> /home/builder/daily-builds/4.2-i686-builds/20141113/
│   ├── 20150123 -> /home/builder/daily-builds/4.2-i686-builds/20150123/
│   └── isoUpdates
└── 4.2-x86_64-builds
    ├── 20140829-2 -> /home/builder/daily-builds/4.2-x86_64-builds/20140829-2/
    ├── 20141106 -> /home/builder/daily-builds/4.2-x86_64-builds/20141106/
    ├── 20141113 -> /home/builder/daily-builds/4.2-x86_64-builds/20141113/
    ├── 20150123 -> /home/builder/daily-builds/4.2-x86_64-builds/20150123/
    └── isoUpdates

12 directories, 0 files
```

其内只是存放了对应正式发布版本的软链接，此软链接为在正式发布时手动生成。

### 3.3.5 /home/builder/rocky6.0.42-test

测试光盘发布目录：

其内存放了各测试版本的发布镜像以及相关的文档信息，其目录结构如下：

```
root@linx-info: /home/builder/stable-iso# tree
.
├── 4.2-i686-builds
│   ├── 20140829-2 -> /home/builder/daily-builds/4.2-i686-builds/20140829-2/
│   ├── 20141106 -> /home/builder/daily-builds/4.2-i686-builds/20141106/
│   ├── 20141113 -> /home/builder/daily-builds/4.2-i686-builds/20141113/
│   ├── 20150123 -> /home/builder/daily-builds/4.2-i686-builds/20150123/
│   └── isoUpdates
├── 4.2-x86_64-builds
│   ├── 20140829-2 -> /home/builder/daily-builds/4.2-x86_64-builds/20140829-2/
│   ├── 20141106 -> /home/builder/daily-builds/4.2-x86_64-builds/20141106/
│   ├── 20141113 -> /home/builder/daily-builds/4.2-x86_64-builds/20141113/
│   ├── 20150123 -> /home/builder/daily-builds/4.2-x86_64-builds/20150123/
│   └── isoUpdates
└── 12 directories, 0 files
```

## 4 编译机环境搭建（虚拟机）

### 4.1 系统环境准备

本节，讲述了系统环境的硬件需求以及操作系统要求。

#### 4.1.1 硬件要求

内存：1GB 以上

硬盘：100GB 以上

#### 4.1.2 操作系统要求

操作系统：rocky6.0.42.41 以上版本

（本次使用系统版本为 rocky6.0.42.41-x86\_64-2015-03-20）

软件包需求：

1. 光盘镜像制作工具：

xosview#1.8.3-x86\_64-linx-Rocky4.2.pkg.tar.gz

网络配置：

对网络进行配置，保证网络畅通

（本次网络配置地址为：172.16.250.122）

### 4.2 系统配置

本节，讲述了对操作系统环境的部分必要的配置内容

### 4.2.1 修改 grub 默认启动项

设置默认 grub 启动为有 root 模式

1. #vim /etc/default/grub  
将 GRUB\_DEFAULT=1 修改为 GRUB\_DEFAULT=0
2. #update-grub

### 4.2.2 挂载公共目录

目前，172.16.0.250:/home/builder/目录为各个体系结构的公共目录，用于存放各个体系结构的历史软件包以及历史光盘镜像。

设置开启自动挂载

1. 修改 172.16.0.250 配置

```
#ssh root@172.16.0.250
#vim /etc/exports
添加配置行
/home/builder 172.16.250.122(rw,anonuid=1001,anongid=1001,sync,
no_subtree_check,no_root_squash)
```

2. 重启 nfs-kernel-server 服务

```
/etc/init.d/nfs-kernel-server restart
```

3. 配置 172.16.250.122 的 fstab

```
#ssh root@172.16.250.122
#vim /etc/fstab
添加配置行 172.16.0.250:/home/builder /home/builder nfs nolock 0 0
```

4. 创建挂载目录

```
#mkdir -p /home/builder
```

5. 挂载测试

```
#mount -a
```

注：在 rocky6.0.42 系统上可能，无法自动挂载上，需要在/etc/rc.d/rc.local 文件增加配置行 mount -a 以确保开机自动挂载。

### 4.2.3 配置密钥

登录 172.16.0.108，对当前系统用户的权限进行配置，使得其对构建过程中涉及到的 Git 库有克隆权限，涉及道德 Git 库列表如下：

序号	Git 库	Git 库地址
----	-------	---------

1	autobuild-tools	git@172.16.0.108:linux6.0.42/autobuild-tools.git
2	install	git@172.16.0.108:linux6.0.42/install.git
3	StateGrid	git@172.16.0.108:linux6.0.42/StateGrid.git
4	new_install	git@172.16.0.108:linux6.0.42/new_install.git
5	linx-app	git@172.16.0.4:sources/linx-app.git
6	linux-firmware	gitweb@172.17.0.252:~/linux-firmware
7	linx-serialnumber.	git@172.16.0.4:sources/linx-serialnumber.git
8	chroot_git	i686:git@172.16.0.108:chroot_git/chroot_i686_git.git ia64:git@172.16.0.108:chroot_git/chroot_ia64_git.git x86_64:git@172.16.0.108:chroot_git/chroot_x86_64_git.git
9	manipulating_build_pkgs	i686:git@172.16.0.108:manipulating_build_pkgs/i686.git x86_64:git@172.16.0.108:manipulating_build_pkgs/x86_64.git it ia64:git@172.16.0.108:manipulating_build_pkgs/ia64.git
10	applications	git@172.16.0.108:linux6.0.42/applications.git
11	liblinxsn	git@172.16.0.4:secret/liblinxsn.git

## 4.3 相关 git 库环境构建

### 4.3.1 准备 4.2 自动编译工具

1. 创建工作目录

```
#mkdir -p /home/x86_64-workdir
```

2. 克隆 autobuild-tools.git

```
#git clone git@172.16.0.108:linux6.0.42/autobuild-tools.git
```

3. 使用工具克隆相关 git 库

```
#!/autobuild-tools/build-pkg/pre.sh
```

4. 等待执行结束即可

## 4.4 测试及配置

1. 至此，rocky6.0.42 构建环境结束，执行

```
#!/home/x86_64-workdir/autobuild-tools/build-pkg/auto_gitcheck.sh -m
```

进行构建测试

2. 配置默认开机启动

```
#vim /etc/rc.d/rc.local
```

添加配置行

```
nohup /home/x86_64-workdir/autobuild-tools/build-pkg/auto_gitcheck.sh >>  
/home/x86_64-workdir/auto_gitcheck.log &
```

```
root@linx-info:/home/builder/stable-iso# tree  
.  
├── 4.2-i686-builds  
│   ├── 20140829-2 -> /home/builder/daily-builds/4.2-i686-builds/20140829-2/  
│   ├── 20141106 -> /home/builder/daily-builds/4.2-i686-builds/20141106/  
│   ├── 20141113 -> /home/builder/daily-builds/4.2-i686-builds/20141113/  
│   └── 20150123 -> /home/builder/daily-builds/4.2-i686-builds/20150123/  
│   └── isoUpdates  
├── 4.2-x86_64-builds  
│   ├── 20140829-2 -> /home/builder/daily-builds/4.2-x86_64-builds/20140829-2/  
│   ├── 20141106 -> /home/builder/daily-builds/4.2-x86_64-builds/20141106/  
│   ├── 20141113 -> /home/builder/daily-builds/4.2-x86_64-builds/20141113/  
│   └── 20150123 -> /home/builder/daily-builds/4.2-x86_64-builds/20150123/  
│   └── isoUpdates  
└──  
12 directories, 0 files
```

5