

# Android 编译+GIT 安装配置步骤

Ubuntu Server 12.04.x LTS x64

Build 2013-05-13

## 目录

更新日志:	2
一、Ubuntu Server 12.04 LTS	3
1、去 Ubuntu 官方网站下载 Ubuntu Server 12.04 LTS X64	3
2、安装 Ubuntu Server	3
2.1、Ubuntu Server 12.04 LTS X64 安装步骤	3
2.2、安装图形界面	4
2.3、重设网卡管理工具	7
2.4、设置服务器名称	8
3、JDK 安装	9
3.1、安装 sun JDK(Oracle JDK)	9
3.2、安装 OpenJDK7 【Android L 编译，不改变系统默认 JDK】	10
3.3、【Android PDK 编译，设置系统默认 JDK】	13
4、安装编译环境	14
A、Android 4	14
C、Linux[此步骤不对外]	15
D、Android NDK	15
Z、所有 Android 相关 x64 安装放在一起	16
所有 Linux 编译相关安装	16
所有 Android 编译+Linux 编译	17
5、安装开发工具	18
5.1、安装代码比较工具	18
5.2、代码生成文档工具	18
5.3、其他工具	18
5.4、gcc 版本设置	18
5.5、git-email 安装	19
5.6、同步时间	19
6、设置多用户环境	21
6.1、新建用户	21
6.2、新建 Samba 用户	21
6.3、设置 samba 共享目录	21
6.4、配置 sudo 权限组	22
7、性能调优	22
7.1、Swap 优先级调节	22
8、安装性能监控软件	23
99、安装杀毒软件并检查服务器安全配置	23

# 更新日志:

2014/11/10 新增 Android L 编译所需 OpenJDK 安装

2014/10/23 新增 Android PDK 默认 OpenJDK 7 安装

2013/3/1 建议添加一个步骤: 在安装中文支持的前提下, 设置系统默认语言为英文。

2013/5/13 增加 12.04.2 Samba 共享 WinXP 无法使用问题的解决方法, 以及删除 smb.conf 错误选项 publice( 因为 public 但启用 guest ok=no 后无需设置 public 选项)。

2013/5/17 增加 Samba 共享目录权限限制, 避免 Samba 用户只读访问其他用户的共享目录。

2013/7/2 增加 12.04.4 添加 samba 用户的命令改变情况。

# 一、Ubuntu Server 12.04 LTS

## 1、去 Ubuntu 官方网站下载 Ubuntu Server 12.04 LTS X64

<http://releases.ubuntu.com/precise/>

选择最新的 ubuntu-12.04.X-server-amd64.iso 下载，例如目前最新的是 ubuntu-12.04-server-amd64.iso  
注意选择 64 位 Server 版本。

## 2、安装 Ubuntu Server

将下载来的 iso 刻录成光盘，用光盘启动服务器，进行系统的安装。

注意安装系统期间，**不要**连接外网，以免安装时自动连外网更新软件包，安装起来更慢。

### 2.1、Ubuntu Server 12.04 LTS X64 安装步骤

- 1、Language，语言选择，选择“中文（简体）”
- 2、进入安装界面选择第一项：安装 Ubuntu 服务器版
- 3、提示语言不能完全被支持，问 Continue the installation in the selected language? 选择“Yes”
- 4、检测键盘布局？选默认“否”
- 5、键盘布局所属国家，，选默认“汉语”
- 6、键盘布局，选默认“汉语”
- 7、主网络接口，选择“eth0”
- 8、网络自动设置失败时，选“继续”
- 9、网络配置方式，选择“手动进行网络设置”
- 10、进行网络配置，配置 ip、子网掩码、网关、域名服务器
- 11、设置主机名，根据情况设置，例如：Exdroid001
- 12、域名留空不设置
- 13、请输入新用户的全名：Exdroid
- 14、您的帐户的用户名：exdroid
- 15、设置密码，若密码少于 8 位会有提示，建议设置带大写小写数字符号的较复杂密码。
- 16、加密你的主目录，选择“否”
- 17、硬盘分区，选择“手动”，选定磁盘回车，出现要在此设备上创建新的空分区表，选择“是”。  
分区注意  
/boot: 500M  
Swap: 内存的 1-2 倍，如果内存超过 24G，则按内存的 1/2 分配，最大不超过 24G，如果内存大于等于 192G，则按内存的 1/8 计算 Swap 大小。  
/: 200G  
/home: 剩下的都分给 Home  
分区设定结束并将修改写入磁盘
- 18、将改订写入磁盘吗？选择“是”
- 19、HTTP 代理信息，留空不设，选择“继续”

- 20、你想如何管理该系统上的更新，选择“自动安装安全更新”
- 21、请选择要安装的软件，选择全部的服务，但不要勾选最后一行“手动选择 Manual package selection”  
选择时用空格键选定，用上下方向键导航，全部选择完成后，按 Tab 键选中“继续”，按回车进入下一步。
- 22、New password for the MySQL "root" user，设置 MySQL 的 root 用户密码
- 23、General type of mail configuration，选择"Internet Site"
- 24、System mail name,按默认设置名称与主机名相同即可
- 25、将 GRUB 启动引导器安装到主引导记录（MBR）上，选择“是”
- 26、安装完成，把光盘取出，然后选择“继续”
- 27、系统会重启，进入字符界面，登陆 exdroid 用户即可。

## 2.2、安装图形界面

默认 Ubuntu Server 是命令行界面，没有图形界面

安装图形界面：

启用 root 帐号：

```
$sudo passwd root
```

输入安装时候的用户名 exdroid 和对应的密码；

输入两次 root 密码；

安装 Ubuntu server 12.04 桌面系统

1、连接网络，你一定要确保网络通畅，现在 Liunx 发行版大部分都使用在线安装的方式，当然离线也是可以的，需要修改一些东西，或下载很多相关组件非常的麻烦。还是先想办法将网络连上。

设置 IP：

```
$sudo ifconfig eth0 192.168.x.x netmask 255.255.255.0
```

如果需要手动设置自动获取 IP，则运行命令：

```
$sudo ifconfig eth0 dhcp
```

```
$sudo ifconfig eth0 down
```

```
$sudo ifconfig eth0 up
```

设置网关：

```
$sudo route add default gw 192.168.x.1
```

设置 DNS：

```
$sudo vi /etc/resolv.conf
```

2、安装图形界面

```
$sudo apt-get update
```

```
$sudo apt-get upgrade
```

 (确保安装的是最新的 12.04 LTS 版本，视网络情况，不升级也可以，勿升级到非 LTS 版本)

```
$sudo apt-get install ubuntu-desktop
```

```
$sudo apt-get install language-pack-zh-hans language-pack-zh-hant
```

为防止编译时候出现“初始化月份数字字符串出错”(12.04 LTS)，修改如下文件：

```
$sudo cp /var/lib/locales/supported.d/local /var/lib/locales/supported.d/local.bak
```

```
$sudo vi /var/lib/locales/supported.d/local
```

将文件内容修改如下：

```
en_US.UTF-8 UTF-8
zh_CN.UTF-8 UTF-8
zh_CN.GBK GBK
zh_CN GB2312
```

```
$sudo locale-gen
```

```
$sudo cp /etc/default/locale /etc/default/locale.bak
```

```
$sudo vi /etc/default/locale
```

修改文件内容如下：

```
LANG="zh_CN.UTF-8"
LANGUAGE="zh_CN:zh"
LC_NUMERIC="zh_CN.UTF-8"
LC_TIME="zh_CN.UTF-8"
LC_MONETARY="zh_CN.UTF-8"
LC_PAPER="zh_CN.UTF-8"
LC_NAME="zh_CN.UTF-8"
LC_ADDRESS="zh_CN.UTF-8"
LC_TELEPHONE="zh_CN.UTF-8"
LC_MEASUREMENT="zh_CN.UTF-8"
LC_IDENTIFICATION="zh_CN.UTF-8"
```

```
$sudo apt-get install gnome-session-fallback
```

 (如果不习惯 Unity 界面，登录时选择 GNOME Classic)

安装完，重启，登陆时选择 GNOME Classic。

注意：本文以下所述图形界面的操作都基于 GNOME Classic！

### 3、安装 VNC Server

注意不要启用系统自带的“应用程序”→“系统工具”→“首选项”→“桌面共享”。系统自带的 vino

必须在控制台登录了才能远程连接。按以下方法安装使用更好用的 vnc4server:

打开终端:

```
$sudo apt-get install vnc4server
```

```
$vncserver
```

 (注意不是 sudo vncserver)

(注意按提示设置 vnc 密码, 手动设置 VNC 密码命令: vncpasswd)

```
$vncserver -kill :1
```

```
$ vi ~/.vnc/xstartup
```

在文件最后添加一行 **gnome-session --session=gnome-classic &**, 保存退出。(注意第二个 session 前面有两个英文减号, 不要只输入一个减号或者输入中文字符减号)

#### 4、增强工具

右键打开终端:

```
$sudo apt-get install nautilus-open-terminal
```

系统管理工具:

```
$sudo apt-get install gnome-system-tools
```

右键菜单 “以管理员身份打开”:

[本框内信息知道即可, 可以不用设置] 首先, 在 Ubuntu 11.10 X64 系统用

```
$sudo apt-get install nautilus-gksu
```

安装, 然后拷贝 11.10 上的 `/usr/lib/nautilus/extensions-2.0/libnautilus-gksu.so` 文件 (`\\172.16.1.6\its\IT\ISO\Linux\Ubuntu\Ubuntu 12.04 LTS Resources\nautilus extensions`), 放到 Ubuntu 12.04 LTS x64 系统的 `/usr/lib/nautilus/extensions-2.0/` 目录下, 运行

```
$sudo ln -s /usr/lib/nautilus/extensions-2.0/libnautilus-gksu.so
```

```
/usr/lib/nautilus/extensions-3.0/libnautilus-gksu.so
```

。并在 12.04 上执行 `$nautilus -q` 退出 nautilus, 重新打开一个主目录即可重新启动 nautilus 会自动安装 nautilus 新扩展。

**注意:** 12.04 上系统默认已经有一个 gksudo, 在打开需要管理员权限的窗口进行更改操作时, 会自动弹出密码输入窗口来应用 sudo 权限。而特别需要注意的是 “用户和组” 有 bug, 如果右键以管理员身份打开, 会出现假死, 必须把 vnc 窗口关闭才能关闭用户和组窗口结束进程 (12.04 版本)。所以 nautilus-gksu (即右键功能: 以管理员身份打开) 更多的适用于右键打开文件时候获得编辑权限, 或者用于其他没有假死 bug 的程序。

图形界面下配置防火墙:

```
$sudo apt-get install gufw
```

图形界面下配置启动服务：

```
$ sudo apt-get install bum
```

图形界面下配置 Samba：

```
$ sudo apt-get install system-config-samba
```

性能监控软件：

```
$ sudo apt-get install nmon
```

网络流量监控：

```
$ sudo apt-get install nethogs iptraf
```

5、重启服务器，待服务器启动到了登陆界面，用 ssh 登录，输入命令开启 VNC 窗口：

```
$vncserver :1
```

6、使用 VNC Viwer 登录进入图形界面，

*注意：由于只允许打开一个 GDM 管理程序，平时如果要使用 VNC 远程登录服务器，**必须**先在控制台注销！否则 VNC 上去的界面没有工具栏。*

*如果出现 VNC 连接后没有工具栏的现象，那么首先将服务器上的用户注销，并按下 Ctrl+Alt+F1 进入字符界面登陆后，使用\$`vncserver -kill :1`的方式来关闭窗口，然后再运行\$`vncserver :1`来重新打开窗口。*

应用软件→系统工具→System Settings→Language Support:

会提示语言支持没有安装完整，按提示安装即可。

应用软件→系统工具→System Settings→Additional Driver:

按提示看是否需要更新驱动。

7、重启服务器

```
#shutdown -r now
```

**注意：**在平时 `vnc4server` 在系统重启后不会自动开启窗口，需要先使用 SSH 登陆服务器，再利用 `$vncserver :1` 来开启窗口。这样才能使用 `vnc viewer` 来远程连接服务器完成图形界面下的相关操作。如果需要开机自动启用 VNC 窗口，则需要利用开机脚本来实现了。

## 2.3、重设网卡管理工具

进入服务器控制台（服务器本地）的图形界面，应用程序→系统工具→首选项→网络连接，看看网络是否正常，如果网络管理器出现问题，如网络管理小工具里面多出了几个 `managed-eth0` 之类的网卡，或者网络连接里面网卡

不见了，则按下面步骤：

1、修改 NetworkManager.conf

```
$sudo gedit /etc/NetworkManager/NetworkManager.conf
```

将"managed=false" 改为"managed=true"

2、修改 interfaces 文件

```
$sudo cp /etc/network/interfaces /etc/network/interfaces.bak
```

```
$sudo gedit /etc/network/interfaces
```

确保**只有**下面内容，其他内容全部删除：

```
auto lo
```

```
iface lo inet loopback
```

3、重启 network-manager 服务

```
$sudo service network-manager restart
```

4、应用程序→系统工具→首选项→网络连接：这时就可以在 Ubuntu 桌面看到“网络连接”窗口，网卡的名字和顺序也都恢复成默认值了。

5、在图形界面“网络连接”窗口设置各网卡的固定 IP 地址等信息（网关不设，需要时候添加路由出外网），然后重启服务器，再次打开“网络连接”，查看 IP 是否正常。

## 2.4、设置服务器名称

如有需要，设置服务器名：

```
$sudo vi /etc/hostname
```

```
$sudo vi /etc/hosts
```

改名之后需要服务器重启后才能生效。

立即生效但重启后失效的方法：

```
$sudo hostname Exdroid001
```



## 3、JDK 安装

### 3.1、安装 sun JDK(Oracle JDK)

**注意：如果需要编译 Android 2.3.4，那么要安装 Oracle JDK 1.6.0\_31，不要装 1.6.0\_33，装 1.6.0\_33 可能出现编译出的固件刷机后无法启动的问题！**

注意，由于复制粘贴时的文档格式自动转换问题，如果直接复制本文档的命令，部分符号可能会被粘贴板自动转换为中文格式，导致命令出错，建议直接在英文输入法的状态下输入本文档中的命令！

Ubuntu 12.04 默认安装了 openJDK, gcj 等 Java 环境。不过如果要构建一个高效可用的 Java 开发环境，需要安装 SUN JRE 也就是 SUN 公司（现在的 Oracle 公司）的 JDK。

安装步骤如下：

1, 从 Oracle 官网下载最新的 JDK 1.6 Update xx 版本，链接是

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

(\\RFServer\Software\系统工具\升级补丁\运行库\Java\6\JDK, 如果用 SMB 来复制内网服务器文件共享的安装包，那么需要注意，在 Ubuntu12.04 中打开 Windows 文件共享的方法是：打开任意文件夹，按 Ctrl+L，在转到里面输入：

smb://RFServer/Software 后回车，用户名小写，域必须全部大写，密码输入后点击“链接”按钮。)

选择 linux 的 platform, continue 一下开始下载，文件名是 jdk-6u33-linux-x64.bin，暂保存在/home/exdroid/目录下面。

注意：不要贪新去安装 JDK7，因为 JDK7 安装后会无法编译！如果有 JDK6 的更新版本，例如 jdk-6u35，那么可以下载 JDK6 的新 update 版本，下面步骤把所有 6u33 改成 6u35，同理把所有 jdk1.6.0\_33 改为 jdk1.6.0\_35。

**注意：要在 Ubuntu 12.04 上编译 android 2.3.4，推荐安装 JDK1.6.0\_31，安装 1.6.0\_33 版本可能会导致编译出来的固件无法启动的问题。**

2, 新建目录，在 console 终端中，输入：

```
$cd /usr/lib
```

依次输入：

```
$sudo mkdir sunJVM
```

```
$cd sunJVM
```

```
$sudo mkdir JDK
```

这样就新建了/usr/lib/sunJVM/JDK 这样的目录层次。

3, 将安装文件 jdk-6u31-linux-x64.bin 拷贝至新建的目录中，

```
$sudo cp /home/exdroid/jdk-6u31-linux-x64.bin /usr/lib/sunJVM/JDK/ （注意字符间的空格）
```

将可执行权限赋予该安装文件，

```
$cd /usr/lib/sunJVM/JDK
```

```
$sudo chmod a+x jdk-6u31-linux-x64.bin
```

执行安装命令：

```
$sudo ./jdk-6u31-linux-x64.bin
```

按提示安装。

4, 设置环境变量，

在终端执行代码：

```
$sudo gedit /etc/environment
```

添加如下变量：

```
JAVA_HOME=/usr/lib/sunJVM/JDK/jdk1.6.0_31
```

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin"
```

```
CLASSPATH=.:/usr/lib/sunJVM/JDK/jdk1.6.0_31/lib
```

执行下面命令来生效：

```
$source /etc/environment
```

 注意不是 `sudo source /etc/environment`

5, 设置默认的 jdk，因为系统可能将其他已安装的 jdk 作为默认，例如 open-jdk，因此需要在终端执行代码：

```
$sudo update-alternatives --install /usr/bin/java java /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/java 400
```

（400 是权重，可以设置为更大值）

```
$sudo update-alternatives --install /usr/bin/javac javac /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/javac 400
```

```
$sudo update-alternatives --install /usr/bin/jar jar /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jar 400
```

接着输入：

```
$sudo update-alternatives --config java
```

选择对应的 jdk：1.6.0\_xx

```
$sudo update-alternatives --config javac
```

选择 1.6.0\_xx，提示“只有一个候选项”且指向的是 1.6.0\_31 则属于正常情况。

```
$sudo update-alternatives --config jar
```

选择 1.6.0\_xx，提示“只有一个候选项”且指向的是 1.6.0\_31 则属于正常情况。

输入以下命令查看当前 java 版本：

```
$java -version
```

```
$javac -version
```

检查是否为 Sun(Oracle) jdk，至此已成功安装 sun-jdk。

## 3.2、安装 OpenJDK7【Android L 编译，不改变系统默认 JDK】

安装 openjdk-7-jdk：

```
$sudo route add default gw 172.16.0.1
```

```
$sudo apt-get update
```

```
$sudo apt-get install openjdk-7-jdk
```

确认系统已安装 OpenJDK7：

```
$sudo update-alternatives --config java      (看带*的默认 java 的版本是不是 sunJDK, 如果不是则选择 sunJDK)
$sudo update-alternatives --config javac    (看带*的默认 javac 的版本是不是 sunJDK, 如果不是则选择 sunJDK)
$sudo update-alternatives --config jar      (看带*的默认 jar 的版本是不是 sunJDK, 如果不是则选择 sunJDK)
```

javap:

```
$sudo update-alternatives --install /usr/bin/javap javap /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/javap 400
$sudo update-alternatives --install /usr/bin/javah javah /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/javah 400
$sudo update-alternatives --install /usr/bin/javadoc javadoc /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/javadoc 400
$sudo update-alternatives --config javap    (看带*的默认 jar 的版本是不是 sunJDK, 如果不是则选择 sunJDK)
$sudo update-alternatives --config javah    (看带*的默认 jar 的版本是不是 sunJDK, 如果不是则选择 sunJDK)
$sudo update-alternatives --config javadoc  (看带*的默认 jar 的版本是不是 sunJDK, 如果不是则选择 sunJDK)
```

看默认 java 的版本:

```
$java -version      (确认默认 java 是 sunJDK)
$javac -version     (确认默认 java 是 sunJDK)
```

查看 jar 和 javap 版本:

```
$ ll /usr/bin/jar
lrwxrwxrwx 1 root root 21 7月 25 2013 /usr/bin/jar -> /etc/alternatives/jar*
$ ll /etc/alternatives/jar
lrwxrwxrwx 1 root root 39 11月 11 10:31 /etc/alternatives/jar -> /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jar*

$ ll /usr/bin/javap
lrwxrwxrwx 1 root root 21 7月 25 2013 /usr/bin/javap -> /etc/alternatives/javap*
$ ll /etc/alternatives/javap
lrwxrwxrwx 1 root root 39 11月 11 10:31 /etc/alternatives/javap -> /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/javap*
```

删除默认外网路由:

```
$sudo route delete default
```

其他设置优先级:

```
sudo update-alternatives --install /usr/bin/jarsigner jarsigner
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jarsigner 400
sudo update-alternatives --install /usr/bin/jcmd jcmd
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jcmd 400
sudo update-alternatives --install /usr/bin/jconsole jconsole
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jconsole 400
sudo update-alternatives --install /usr/bin/jdb jdb /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jdb
400
sudo update-alternatives --install /usr/bin/jhat jhat
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jhat 400
sudo update-alternatives --install /usr/bin/jinfo jinfo
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jinfo 400
sudo update-alternatives --install /usr/bin/jmap jmap
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jmap 400
sudo update-alternatives --install /usr/bin/jps jps /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jps
```

400

```
sudo update-alternatives --install /usr/bin/jrunscript jrunscript
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jrunscript 400
sudo update-alternatives --install /usr/bin/jsadebugd jsadebugd
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jsadebugd 400
sudo update-alternatives --install /usr/bin/jstack jstack
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jstack 400
sudo update-alternatives --install /usr/bin/jstat jstat
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jstat 400
sudo update-alternatives --install /usr/bin/jstatd jstatd
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jstatd 400
sudo update-alternatives --install /usr/bin/appletviewer appletviewer
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/appletviewer 400
sudo update-alternatives --install /usr/bin/extcheck extcheck
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/extcheck 400
sudo update-alternatives --install /usr/bin/idlj idlj
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/idlj 400
sudo update-alternatives --install /usr/bin/native2ascii native2ascii
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/native2ascii 400
sudo update-alternatives --install /usr/bin/rmic rmic
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/rmic 400
sudo update-alternatives --install /usr/bin/schemagen schemagen
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/schemagen 400
sudo update-alternatives --install /usr/bin/serialver serialver
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/serialver 400
sudo update-alternatives --install /usr/bin/wsgen wsgen
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/wsgen 400
sudo update-alternatives --install /usr/bin/wsimport wsimport
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/wsimport 400
sudo update-alternatives --install /usr/bin/xjc xjc /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/xjc
400
sudo update-alternatives --install /usr/bin/policytool policytool
/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/policytool 400
```

其他配置默认版本，注意选择 sunJDK:

```
sudo update-alternatives --config javap
sudo update-alternatives --config jarsigner
sudo update-alternatives --config javadoc
sudo update-alternatives --config javah
sudo update-alternatives --config jcmd
sudo update-alternatives --config jconsole
sudo update-alternatives --config jdb
sudo update-alternatives --config jhat
sudo update-alternatives --config jinfo
sudo update-alternatives --config jmap
sudo update-alternatives --config jps
sudo update-alternatives --config jrunscript
sudo update-alternatives --config jsadebugd
sudo update-alternatives --config jstack
```

```
sudo update-alternatives --config jstat
sudo update-alternatives --config jstatd
sudo update-alternatives --config appletviewer
sudo update-alternatives --config extcheck
sudo update-alternatives --config idlj
sudo update-alternatives --config native2ascii
sudo update-alternatives --config rmic
sudo update-alternatives --config schemagen
sudo update-alternatives --config serialver
sudo update-alternatives --config wsgen
sudo update-alternatives --config wsimport
sudo update-alternatives --config xjc
sudo update-alternatives --config policytool
```

检查是否有未设置:

```
ll /etc/alternatives/ | grep /usr/lib/jvm/java-7-openjdk-amd64/jre/bin
```

以下为用户自己的配置建议，管理员不要在服务器上设置:

用户如果需要使用 **OpenJDK-7**，请用户设置自己的环境变量:

```
JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64/jre
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/lib/jvm/java-7-openjdk-amd64/jre/bin:/arm/arm-2010.09/bin"
CLASSPATH=./usr/lib/jvm/java-7-openjdk-amd64/jre/lib
```

用户切换回默认环境变量:

```
JAVA_HOME=/usr/lib/sunJVM/JDK/jdk1.6.0_31
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/lib/sunJVM/JDK/jdk1.6.0_31/bin"
CLASSPATH=./usr/lib/sunJVM/JDK/jdk1.6.0_31/lib
```

### 3.3、安装 OpenJDK 7

## 3.3、【Android PDK 编译，设置系统默认 JDK】

安装 openjdk-7-jdk

```
$sudo apt-get install openjdk-7-jdk
```

```
$sudo update-alternatives --config java
```

 (选择 openJDK 7)

```
$sudo update-alternatives --config javac
```

 (选择 openJDK 7)

```
$sudo update-alternatives --config jar
```

 (选择 openJDK 7)

```
$sudo update-alternatives --config javap
```

 (选择 openJDK 7)

如果报错只有一个选择，那么：

```
$sudo update-alternatives --install /usr/bin/javap javap /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/javap 400
```

```
$sudo update-alternatives --config javap      （看带*的默认 jar 的版本是不是选择 openJDK 7，如果不是则选择选择 openJDK 7）
```

设置 java PATH:

```
$sudo vi /etc/environment
```

```
JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64/jre
```

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/lib/jvm/java-7-openjdk-amd64/jre  
/bin:/arm/arm-2010.09/bin"
```

```
CLASSPATH=./usr/lib/jvm/java-7-openjdk-amd64/jre/lib
```

看默认 java 的版本:

```
$java -version      (确认默认 java 是 OpenJDK)
```

```
$javac -version     (确认默认 java 是 OpenJDK)
```

查看 jar 和 javap 版本:

```
$ ll /usr/bin/jar
```

```
lrwxrwxrwx 1 root root 21  7 月 25  2013 /usr/bin/jar -> /etc/alternatives/jar*
```

```
$ ll /etc/alternatives/jar
```

```
lrwxrwxrwx 1 root root 39 11 月 11 10:31 /etc/alternatives/jar -> /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/jar*
```

```
$ ll /usr/bin/javap
```

```
lrwxrwxrwx 1 root root 21  7 月 25  2013 /usr/bin/javap -> /etc/alternatives/javap*
```

```
$ ll /etc/alternatives/javap
```

```
lrwxrwxrwx 1 root root 39 11 月 11 10:31 /etc/alternatives/javap -> /usr/lib/sunJVM/JDK/jdk1.6.0_31/bin/javap*
```

## 4、安装编译环境

### A、Android 4

以下是 Android 4 编译服务所需软件包:

ubuntu64:

```
$sudo apt-get install git-core git-email gnupg flex bison gperf build-essential zip curl libc6-dev
```

```
libncurses5-dev x11proto-core-dev libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386
```

```
libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc
```

```
zlib1g-dev:i386 zlib1g-dev lib32ncurses5-dev ia32-libs libx11-dev lib32readline-gplv2-dev
```

```
lib32z1-dev libreadline6-dev libgl1-mesa-glx gcc-multilib uboot-mkimage gcc-4.4 g++-4.4
```

```
g++-4.4-multilib      gcc-4.4-multilib      gcc-4.4-multilib      gcc-4.5-arm-linux-gnueabihf  
gcc-arm-linux-gnueabihf g++-arm-linux-gnueabihf c++-arm-linux-gnueabihf  
  
$sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so
```

## C、Linux[此步骤不对外]

以下是 Linux 编译服务所需软件包：

```
$sudo apt-get install exuberant-ctags emacs cscope mutt-patched tofrodo  
$sudo apt-get install libglib2.0-dev libatk1.0-dev libdirectfb-dev libgtk2.0-dev libghc6-zlib-dev  
$sudo apt-get install uboot-mkimage libtool libtool-doc autoconf automake  
$sudo apt-get install gcc-4.4 g++-4.4 g++-4.4-multilib gcc-4.4-multilib
```

安装 ARM 交叉编译器

```
$sudo mkdir /arm
```

下载 dl\_arm-2010.09-50-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 到桌面。

（12.04 的 nautilus 开始区别大小写，但是有 bug，不能像 10.04 那样使用“位置→连接到服务器”的方法来连接共享目录。首先打开任意文件夹，例如：主文件夹，按 Ctrl+L 键，地址栏就会出现，在地址栏输入 smb://172.16.1.6/its 回车，用户名输入小写，域输入大写的 ALLWINNERTECH，密码输入后，点击“连接”按钮即可登录共享文件夹：\\172.16.1.6\its\IT\Git\安装软件\Package）

右键点击“dl\_arm-2010.09-50-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2”选择“以管理员身份打开”，将里面的 arm-2010.09 文件夹解压缩到/arm/下面；如果没有“以管理员身份打开”，那么输入命令行：

```
$sudo tar xjvf dl_arm-2010.09-50-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 -C /arm
```

然后执行下面的命令修改系统变量：

```
$sudo gedit /etc/environment
```

在 PATH 最后添加：

```
:/arm/arm-2010.09/bin
```

保存 environment 文件后，输入如下命令应用生效：

```
$ source /etc/environment      （注意，不是 sudo source /etc/environment）
```

## D、Android NDK

Android NDK(Native Development Kit)，对于个别用户需要安装 NDK 的，按下面步骤给用户安装：

## 1、下载 NDK

官方网址: <https://developer.android.com/sdk/index.html>

内网地址: ..\Git\安装软件\Package\android-ndk-\*-linux-x86\_64.tar.bz2

## 2、安装 NDK

解压缩 android-ndk-\*-linux-x86\_64.tar.bz2 里面的 android-ndk-\*到用户根目录下面:

```
$cp android-ndk-*-linux-x86_64.tar.bz2 /home/username
```

```
$cd /home/username
```

```
$tar jxvf android-ndk-*-linux-x86_64.tar.bz2
```

设置用户权限:

```
$cd /home/username
```

```
$sudo chown -R username:username android-ndk-*
```

对于 patch 和 NDK\_ROOT 变量, 由于不是全局变量, 由用户自己 export 即可。

## 2、所有 Android 相关 x64 安装放在一起

```
$sudo apt-get install git-core git-email gnupg flex bison gperf build-essential zip curl zlib1g-dev  
gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev ia32-libs x11proto-core-dev libx11-dev  
lib32readline-gplv2-dev lib32z1-dev gitk gettext texinfo uboot-mkimage libc6-dev libgl1-mesa-dev  
mingw32 tofrodos python-markdown libxml2-utils meld docbook-utils xmlto exuberant-ctags  
libncurses5-dev libx11-dev:i386 libreadline6-dev:i386 libreadline6-dev libgl1-mesa-glx:i386  
libgl1-mesa-glx zlib1g-dev:i386 gcc-4.4 g++-4.4 g++-4.4-multilib gcc-4.4-multilib gcc-4.4-multilib  
gcc-4.5-arm-linux-gnueabihf gcc-arm-linux-gnueabihf g++-arm-linux-gnueabihf  
c++-arm-linux-gnueabihf uuid-dev tree  
  
$sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so
```

## 所有 Linux 编译相关安装

```
$sudo apt-get install git-core git-email gnupg flex bison gperf build-essential zip curl zlib1g-dev  
gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev ia32-libs x11proto-core-dev libx11-dev  
lib32readline-gplv2-dev lib32z1-dev gitk gettext texinfo uboot-mkimage libc6-dev libgl1-mesa-dev
```



```
mingw32 tofrodos python-markdown libxml2-utils meld docbook-utils xmlto libx11-dev:i386
libreadline6-dev:i386 libreadline6-dev libgl1-mesa-glx:i386 libgl1-mesa-glx zlib1g-dev:i386
exuberant-ctags emacs cscope mutt-patched libglib2.0-dev libatk1.0-dev libdirectfb-dev libgtk2.0-dev
libghc6-zlib-dev libtool libtool-doc autoconf automake gcc-4.4 g++-4.4 g++-4.4-multilib
gcc-4.4-multilib gcc-4.4-multilib gcc-4.5-arm-linux-gnueabihf gcc-arm-linux-gnueabihf
g++-arm-linux-gnueabihf c++-arm-linux-gnueabihf tree
```

## 所有 Android 编译+Linux 编译

```
$sudo apt-get install git-core git-email gnupg flex bison gperf libssl1.2-dev libbsd0-dev libwxgtk2.6-dev
build-essential zip curl libncurses5-dev zlib1g-dev valgrind gitk gettext texinfo uboot-mkimage
gcc-multilib g++-multilib libc6-dev libc6-dev-i386 lib32ncurses5-dev ia32-libs x11proto-core-dev
libx11-dev lib32readline-gplv2-dev lib32z1-dev libc6-dev libgl1-mesa-dev mingw32 tofrodos
python-markdown libxml2-utils meld docbook-utils xmlto libx11-dev:i386 libreadline6-dev:i386
libreadline6-dev libgl1-mesa-glx:i386 libgl1-mesa-glx zlib1g-dev:i386 exuberant-ctags emacs cscope
mutt-patched libglib2.0-dev libatk1.0-dev libdirectfb-dev libgtk2.0-dev libghc6-zlib-dev libtool
libtool-doc autoconf automake xsltproc gcc-4.4 g++-4.4 g++-4.4-multilib gcc-4.4-multilib
gcc-4.5-arm-linux-gnueabihf gcc-arm-linux-gnueabihf g++-arm-linux-gnueabihf
c++-arm-linux-gnueabihf uuid-dev tree

$sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so
```

## 5、安装开发工具

### 5.1、安装代码比较工具

```
#sudo apt-get install meld
```

安装 meld 后用 `git config` 命令设置 meld 为默认的比较工具，并且把 `prompt` 设为 `false`（不显示提示对话框），这样以后就可以使用 meld 作为比较工具了。

```
$sudo git config --system diff.tool meld
```

```
$sudo git config --system difftool.prompt false
```

或者直接修改配置文件/etc/gitconfig

```
[diff]
    tool = meld
[difftool]
    prompt = false
```

### 5.2、代码生成文档工具

```
$sudo apt-get install docbook-utils xmlto
```

### 5.3、其他工具

```
$sudo apt-get install exuberant-ctags uuid-dev tree
```

### 5.4、gcc 版本设置

```
$sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.4 100
```

```
$sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.6 50
```

```
$sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.4 100
```

```
$sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.6 50
```

```
$sudo update-alternatives --install /usr/bin/cpp cpp-bin /usr/bin/cpp-4.4 100
```

```
$sudo update-alternatives --install /usr/bin/cpp cpp-bin /usr/bin/cpp-4.6 50
```

## 5.5、git-email 安装

```
$sudo apt-get install git-email
```

配置 git-sendmail:

查看 /var/log/mail 的日志确定使用 sendmail 或是 postfix:

如果是 sendmail 代理:

```
#vi /etc/mail/sendmail.cf
```

找到 DS 在后面加上:中间不需要空格。

```
DS:repeater.allwinnertech.com
```

如果是 postfix 代理:

```
#grep relayhost /etc/postfix/main.cf
```

然后设置 relayhost = repeater.allwinnertech.com

然后重启 sendmail 或 postfix,

查看 maillog 已经使用 repeater.allwinnertech.com

## 5.6、同步时间

内网服务器设置 NTP 时间同步源: time.allwinnertech.com, 即 BCDE 网段的 38 服务器。[禁止添加到客户发布文档]

外网服务器设置 NTP 时间同步源: time.asia.apple.com。[可添加到客户发布文档]

执行一次时间同步:

```
$sudo ntpdate time.allwinnertech.com
```

如果提示找不到主机 time.allwinnertech.com, 则根据服务器所在网段设置 hosts 文件后再手动同步一次:

```
$sudo vi /etc/hosts
```

```
172.16.1.38  time.allwinnertech.com
```

添加同步时间任务:

```
$sudo vi /etc/crontab
```

```
# auto sync time
30 * * * * root ntpdate time.allwinnertech.com
```

**注意:** 如果 crontab 里面的示例里面没有 user-name, 那么上面的任务语句里面删除执行用户 (root)。



## 6、设置多用户环境

注意：Ubuntu Server x64 **12.04.2** 版本自带的 Samba 版本有问题，导致 Win7 可以访问 Samba 共享而 WinXP 无法进入 Samba 共享文件夹。必须升级。

执行：

```
$sudo apt-get install samba
```

按提示升级即可！

### 6.1、新建用户

为各开发人员在 Ubuntu Server 中新增用户，应用程序→系统工具→系统管理→用户和组，或者通过命令行：

```
$sudo useradd -m -d /home/user001 -s /bin/bash user001
```

### 6.2、新建 Samba 用户

在终端下输入：

```
$smbpasswd -a username
```

 (注意用户名和系统用户名一致)

或者

```
$sudo smbpasswd -a username
```

 (注意用户名和系统用户名一致，12.04.4 开始要加 sudo)

### 6.3、设置 samba 共享目录

方法 1：使用图形工具

应用程序→系统工具→系统管理→Samba，

点击“添加”按钮，在“基本”选项页填写目录、共享名、描述后，选择“可擦写”和“显示”，在“访问”选项页选择“只允许指定用户的访问”，只让用户自己能访问自己的 home 目录。

方法 2：修改配置文件

修改/etc/samba/smb.conf 文件：

```
$sudo gedit /etc/samba/smb.conf
```

在文件的最后一行添加：

```
[username]
    comment = username
    path = /home/username
    guest ok = no
    writable = yes
    valid users = username
```

## 6.4、配置 sudo 权限组

允许 sudo-users 用户组使用 sudo 权限：

1、新建用户组 sudo-users，并把相关用户添加进这个组

```
$sudo groupadd sudo-users
```

应用程序→系统工具→系统管理→用户和组，点击“管理组”，找到并双击组：sudo-users，将各 sudo 用户勾选后，点击“确定”按钮，再次双击 sudo-users 组，看是否成功添加成员。“用户和组”管理工具有 Bug 见本文 2.2 中“右键以管理员身份打开”部分。如果添加不了使用如下命令：

```
$sudo gpasswd -a user001 sudo-users      //添加 user001 到 sudo-users 组
```

```
$sudo gpasswd -d user002 sudo-users      //从 sudo-users 组删除成员 user002
```

2、在终端下，输入 visudo，修改/etc/sudoer 配置文件：

```
$sudo visudo
```

在# User privilege specification 下面增加%sudo-users 的描述：

```
# User privilege specification
```

```
root    ALL=(ALL) ALL
```

```
%sudo-users ALL=(ALL) NOPASSWD: /bin/mknod, /bin/mount, /bin/umount
```

Ctrl+O 保存写入，提示保存路径时直接按回车确认；

Ctrl+X 退出。

## 7、性能调优

### 7.1、Swap 优先级调节

```
$cat /proc/sys/vm/swappiness
```

可以看到显示的是 60，即 60%机会分配给 Swap

设置临时修改，立即生效：

```
$sudo sysctl vm.swappiness=10
```

设置重启后可生效，修改 sysctl.conf 文件，最后一行增加一条：

```
$sudo vi /etc/sysctl.conf
```

```
vm.swappiness = 10
```

## 8、安装性能监控软件

安装 Nagios。

## 99、安装杀毒软件并检查服务器安全配置

按《服务器安全配置》标准来安装杀毒软件；

按《服务器安全配置》标准来检查服务器的安全性；