

操作系统上机实验指南环境搭建

单位:	南开大学机器智能研究所 Institute of Machine Intelligence. Nankai University
日期:	2016/9/27

目录

1. Linux 操作系统简介.....	3
2. Linux 操作系统安装.....	5
2.1. 硬盘安装 Linux.....	6
2.2. Windows 下使用虚拟机安装 Linux.....	9
2.3. Ubuntu 的本地配置	10
3. 常用 Linux 命令.....	10
3 .1. 系统命令	10
3 .2. 文件操作	10
3 .3. 一些常用的文本编辑软件.....	11
4 . JOS 操作系统.....	11
5 . Qemu 模拟器.....	11
6 . 实验所需配置	12

1. Linux 操作系统简介

Linux(你知道这个单词的正确发音吗?)操作系统，是一类 Unix 计算机操作系统的统称。Linux 操作系统的内核的名字也是“Linux”。Linux 操作系统也是自由软件和开放源代码发展中最著名的例子。

严格来讲，Linux 这个词本身只表示 Linux 内核，但在实际上人们已经习惯了用 Linux 来形容整个基于 Linux 内核，并且使用 GNU 工程各种工具和数据库的操作系统（也被称为 GNU/Linux）。基于这些组件的 Linux 软件被称为 Linux 发行版。一般来讲，一个 Linux 发行包包含大量的软件，比如软件开发工具（例如 gcc）、数据库（例如 PostgreSQL、MySQL）、Web 服务器（例如 Apache）、X Window、桌面环境（例如 GNOME 和 KDE）、办公包（例如 OpenOffice.org）、脚本语言（例如 Perl、PHP 和 Python）等等。

Linux 内核最初只是由芬兰人林纳斯·托瓦兹（Linus Torvalds）在赫尔辛基大学上学时出于个人爱好而编写的，当时他并不满意 Minix 这个教学用的操作系统，部分因为只能在有限硬件上运行。最初的设想中，Linux 是一种类似 Minix 这样的一种操作系统。Linux 的第一个版本在 1991 年 9 月被大学 FTP server 管理员 Ari Lemmke 发布在 Internet 上，最初 Torvalds 称这个内核的名称为“Freax”，意思是自由（“free”）和奇异（“freak”）的结合字，并且附上了“X”这个常用的字母，以配合所谓的 Unix-like 的系统。但是 FTP server 管理员嫌原来的命名“Freax”的名称不好听，把内核的称呼改成“Linux”，当时仅有 10000 行代码，仍必须运行于 Minix 操作系统之上，并且必须使用硬盘开机。

Linux 的标志和吉祥物是一只名字叫做 Tux 的企鹅，标志的由来是因为 Linus 在澳大利亚时曾被一只动物园里的企鹅咬了一口，便选择了企鹅作为 Linux 的标志。更容易被接受的说法是：企鹅代表南极，而南极又是全世界所共有的一块陆地。

这也就代表 Linux 是所有人的 Linux。大多数基于 Linux 内核的操作系统使用了大量的 GNU 软件，包括了一个 shell 程序、工具、程序库、编译器及工具，还有许多其他程序，例如 Emacs。正因为如此，GNU 计划的



开创者理查德·马修·斯托曼博士提议将 Linux 操作系统改名为 GNU/Linux，但多数人仍然习惯性地使用"Linux"。

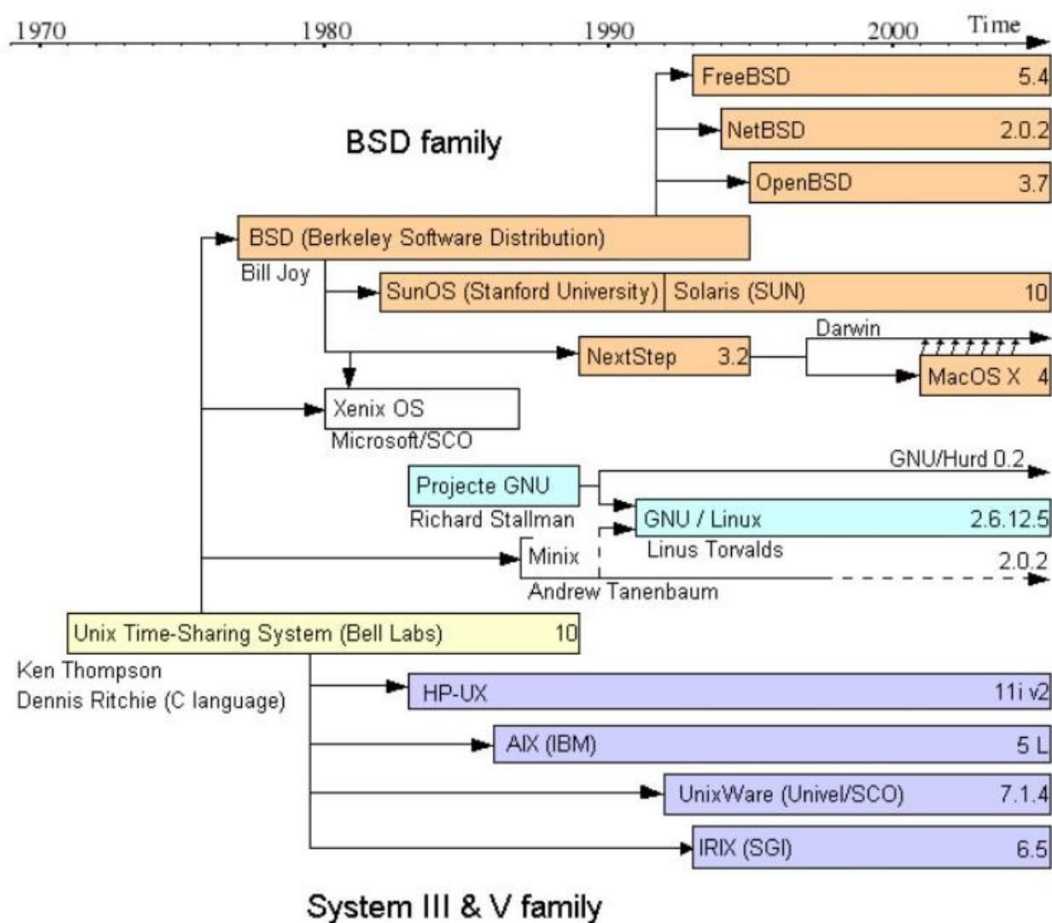
Linux 发行版，指的就是我们通常所说的“Linux 操作系统”，它可能是由一个组织、公司或者个人发行的。Linux 主要作为 Linux 发行版的一部分而使用。通常来讲，一个 Linux 发行版包括 Linux 内核，将整个软件安装到计算机上的一套安装工具，各种 GNU 软件，其他的一些自由软件，在一些特定的 Linux 发行版中也有一些专有软件。发行版为许多不同的目的而制作，包括对不同计算机结构的支持，对一个具体区域或语言的本地化，实时应用，和嵌入式系统。国内的 Linux 发行版如红旗 Linux 等。

一个典型的 Linux 发行版包括：Linux 内核，一些 GNU 程序库和工具，命令行 shell，图形界面的 X Window 系统和相应的桌面环境，如 KDE 或 GNOME，并包含数千种从办公包、编译器、文本编辑器到科学工具的应用软件。

Linux 在根本上是从 Unix 操作系统发展过来的，其内核在 1991 年 9 月发布版本 0.0.1、1994 年发布版本 1.0.0、1996 年发布 2.0.0，最新的发布版本是 2009 年的 2.6.31。

到现在，依然有成千上万的程序设计爱好者在为 Linux 内核的更新和维护做着贡献。

Linux 操作系统与我们经常使用的 Windows 操作系统在设计理念上的不同也产生了许多浅层次的区别，常有用户抱怨在 Linux 的安装和使用过程中遇到了诸多不便，并因此而引发 Linux 爱好者和 Windows 用户之间旷日持久的争执，并进而引发开源软件模式和传统的商业软件模式之间的冲突(这样的事情经常发生，在我爱南开 BBS 上就多次出现过)。简单的说，Windows 的目的是让最广大的用户使用到，因而在易用性方面较为突出；而 Linux 由于是被一群技术爱好者所主导，所以其可定制的特性更为明显。相信在完成本学期的操作系统上机实践之后，惯用 Windows 的你也会对这两种操作系统有个更明确的理解。



如要了解更多，请访问 <http://www.kernel.org/> <http://www.linux.org/>

2. Linux 操作系统安装

传统的 Linux 发行版因较为注重可定制的特性，在进行安装时需要进行大量的选择，显得十分繁琐。但是较新的发行版中已经对这些进行了改善，其安装过程十分简便和快捷。除安装之外，由于平常使用 Windows 操作系统，所以为不影响原有的使用环境，将依次介绍几种不同的安装方式。

实验中对 Linux 的发行版本和内核版本均不做限制，我们给出的版本为最新的 ubuntu 14.04.1 作为实验平台。

常用的安装方式主要是硬盘安装和虚拟机安装两种，当然还有别的方式，可自行学习。你们需要选择一种方案进行 Linux 的安装，本文档给出了硬盘安装的方法。

2

.1.硬盘安装 Linux

硬盘安装 Linux 是比较危险的一种安装方法(说危险是指相比虚拟机安装的), 如果安装不慎或操作失误, 就有可能造成原有系统引导失败、分区错误甚至数据丢失。但是这种方法也有一定的好处, 就是能充分应用计算机的硬件资源, 特别是在进行显示输出的硬件渲染方面有着先天的优势

✧ 硬盘安装 Linux 的几个较大的问题是:

- 1) 划分分区。由于 Linux 使用的分区格式和 Windows 默认的 Fat32、NTFS(这两个单词的意思, 可自行查阅相关资料)不同, 所以需要划分出不用的硬盘空间给 Linux 进行安装。
- 2) 多系统引导。如果主机上已经装有其他系统, 那么需要进行多系统的引导设置, 以保证每个系统都可以被使用。
- 3) 硬件驱动。较新的硬件的驱动程序都已经在 Linux 发行版中自带, 具体情况, 请查阅你所选择的发行版的支持信息。

✧ 硬盘安装 Linux 的几个优势是:

- 1) 硬件性能。可以完整的发挥计算机的意见性能, 获得更好的显示效果;
- 2) 访问硬盘的其余分区。通过良好的设置(在 Ubuntu 下甚至可以自动配置), 可以直接访问原先的 Windows 系统中的分区, 并提供读写操作。

✧

ubuntu 14.04.1 的安装文件 ([ubuntu-14.04.1-desktop-i386.iso](#) , 附件中提供 [lab1/resource/ubuntu-14.04.1-desktop-i386.iso](#)) 或者可以通过以下多种方式取得:

- 1) 通过其官方网站(<http://www.ubuntu.org.cn/>)下载
- 2) 通过其官方网站或第三方站点进行下载, 南开校内软件之家提供下载 (<http://ftp.nankai.edu.cn/#Ubuntu>), 不过校内的下载目前适合的最新的应该只有 13 年的。

- Ubuntu-13.10 32位: [ubuntu-13.10-desktop-i386.iso](#)
- Ubuntu-13.10 64位: [ubuntu-13.10-desktop-amd64.iso](#)

✧ 获取以上文件后, 正式进入安装过程:

- 1) 下载分区助手, 对磁盘进行分区, 分出一个 20G~30G 大小的区, 文件系统选择 Ext3, (如下图)。此分区用于安装存放 Linux 系统。

分区	文件系统	容量	已使用	未使用	类型	状态	4KB对齐
磁盘 1							
G: 系统保留	NTFS	100.00MB	30.27MB	69.73MB	主	系统	是
C:	NTFS	99.90GB	85.71GB	14.19GB	主	引导	是
D:	NTFS	199.99GB	159.82GB	40.17GB	主	无	是
E:	NTFS	135.93GB	48.11GB	87.83GB	逻辑	无	
*:	Ext3	29.83GB	109.29MB	29.72GB	逻辑	无	是

- 2) 将 **ubuntu-14.04.1-desktop-i386.iso** 复制到 C 盘根目录，打开 **ubuntu-14.04.1-desktop-i386.iso** 并复制其中的 **.disk** 文件夹和 **casper** 文件夹下的 **initrd.lz** 和 **vmlinuz** 至 C 盘根目录，并给 **vmlinuz** 添加后缀名 **“.efi”**

filesystem.manifest	2014/7/23 6:38	MANIFEST 文件	58 KB
filesystem.manifest-remove	2014/7/23 6:38	MANIFEST-REM...	2 KB
filesystem.size	2014/7/23 6:38	SIZE 文件	1 KB
filesystem.squashfs	2014/7/23 6:38	SQUASHFS 文件	974,820 KB
initrd.lz	2014/7/23 6:38	LZ 文件	19,832 KB
vmlinuz	2014/7/23 6:38	文件	5,684 KB

- 3) 下载 Easy BCD。它能够极好地支持多种操作系统与 Windows 7、Vista 结合的多重启动，包括 Linux、Mac OS X、BSD 等。

选择“添加条目”→“NeoGrub”→“安装”→“配置”→在弹出的 txt 文档中输入以下内容：

```
title Install Ubuntu root
```

```
(hd0, 0)
```

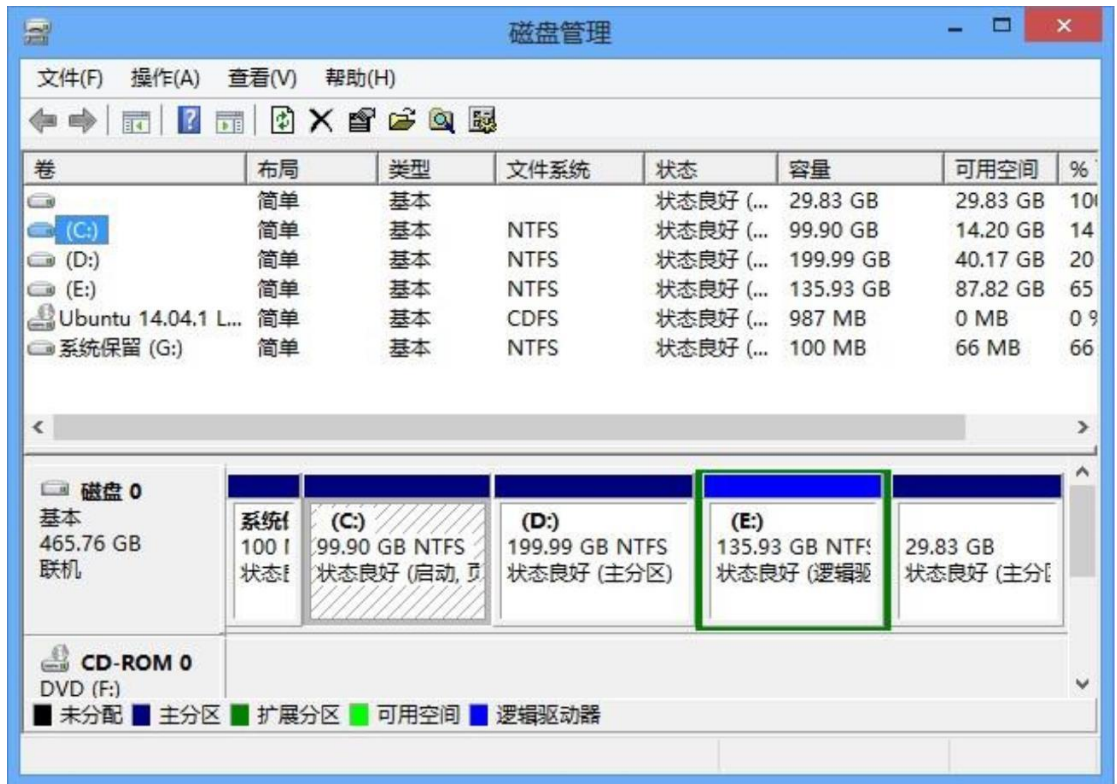
```
kernel (hd0, 0)/vmlinuz.efi boot=casper iso-scan/filename=/u
```

```
buntu-14.04.1-desktop-i386.iso locale=zh_CN.UTF-
```

```
8 initrd (hd0, 0)/initrd.lz title reboot reboot title halt
```

```
halt
```

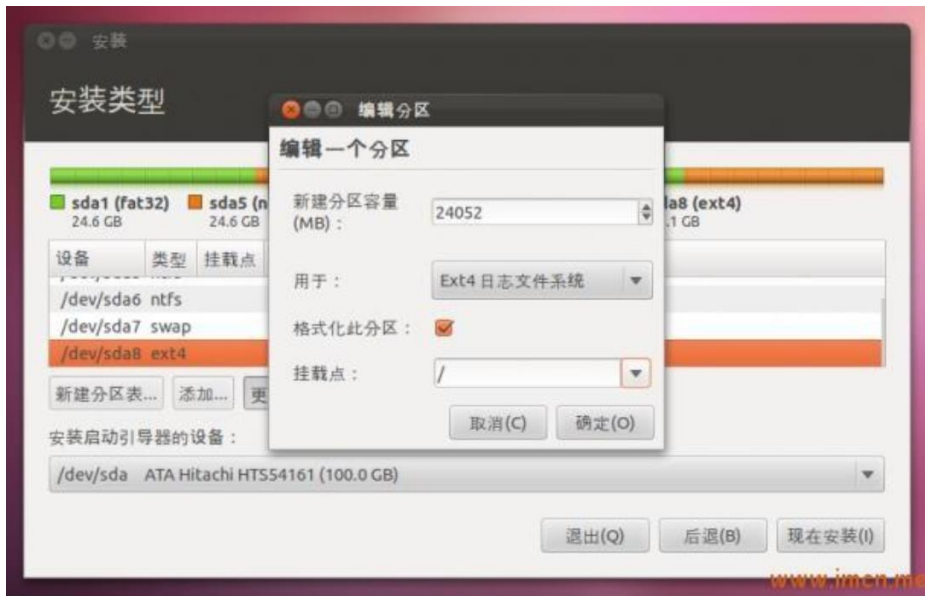
注意以上的 **(hd0, 0)** 中的第一个 0 表示第一块磁盘，第二个 0 表示 C 盘所在的分区索引，此处可以进入计算机的磁盘管理，如图所示，C 盘的位置即为 C 盘的索引。下图中应为 **(hd0, 1)**



- 4) 重启计算机，你就会看到有 2 个 启动菜单给你选择 我们选择第 2 个 NeoGrub。进入后，按 **Ctrl+Alt+T** 打开终端，输入代码:`sudo umount -l /isodevice` 这一命令取消掉对光盘所在 驱动器 的挂载，否则分区界面找不到分区。
- 5) 开始安装。选择自定义安装



- 6) 挂载分区：选择之前我们分配的区



- 7) 选择安装启动引导器的设备时，可以选择我们分好的 / 区，也可以新建一个/boot 区。
- 8) 设置引导。安装好后重启进入 Windows，打开 EasyBCD，删除 NeoGrub (add new entry ——neoGrub——remove); 之后，创建 ubuntu 引导项：
 - ✧ 点击 Linux/BSD
 - ✧ Type 选 GRUB (Ubuntu 使用的是 Grub 启动管理器)
 - ✧ Name 是显示在启动界面上的文字，自己填
 - ✧ Drive 就是你刚才安装 Grub 的分区， “/” 或 “/boot” 的那个分区
 - ✧ 点击 Add 然后你也可以点击 View Settings 修改默认启动哪个系统、点击 Edit Boot Menu 修改默认等待多少秒之类
- 9) 最后，C 盘根目录下 vmlinuz 和 initrd.lz 文件可以删除了，如果你不需要 EasyBCD 了也可以将它卸载掉。重启电脑就可以进入 Windows、Ubuntu 双启动界面了。

2

.2.Windows 下使用虚拟机安装 Linux

参考网上一些配置方法，如：<http://www.doc88.com/p-786388872439.html>

2

.3.Ubuntu 的本地配置

✧ gmake 与 make make 和 gmake 是 Linux 下进行编译常用的工具，但是 Ubuntu 没有自带 gmake，而我们的实验中则大量使用了 gmake，解决方案是到/usr/bin 下创建一个符号连接即可，具体是在 Terminal 中输入如下命令(如果需要，请输入密码)：`cd /usr/bin && sudo ln -s make gmake` 有关 gmake 和 make 的详细关系和纠葛，请仔细查阅资料。

3. 常用 Linux 命令

由于大多数的 Linux 操作都是在终端 (Terminal ， 可以在屏幕最上方的菜单中的 Applications Accessories Terminal 找到，或者通过 `ctrl+alt+t` 快捷键打开)中进行的，所

以将主要介绍一下 Terminal 中的命令(注意，Linux 的命令是十分庞大的，以下所列的只是凤毛麟角九牛一毛沧海一粟而已，如果需要，Linux 系统完全可以在没有 GUI 的条件下运行，所以，熟悉 Linux 系统就必须掌握常用命令。下面提到的只是很少的一部分，还有无穷无尽的命令等待你的发现，如果你感兴趣，也可以编写完全属于自己的命令)。

3

.1.系统命令

✧ sudo 是指 super user do 的意思，所以有时候会要求输入密码，该命令用于一些需要高级权限进行执行的时候；

✧ help 获得命令的帮助，如 `help cd`；`sudo apt-get install vim` 通过源服务器获得 vim 并自动安装 `sudo apt-get install emacs` `sudo apt-get install wine`

你也可以试试 `sudo apt-get install emacs` 的作用(自动查找源服务器并下载 emacs 的安装文件进行安装)。

3

.2.文件操作

✧ cd 进入目录

- ✧ `ls` 列出当前目录下的文件和文件夹
- ✧ `mv` 移动文件，也可以用来重命名文件
- ✧ `rm` 删除文件或文件夹
- ✧ `tar` 解压缩和压缩的命令，专用于打开或创建`.tar.gz`的文件
- ✧ `sudo su` 获取管理员权限你可以通过在命令后面加上 `--help` 来获得该命令的帮助信息。

3

3. 一些常用的文本编辑软件

以下是一些文本编辑器：`vi`

`vim` 相当于 `vi` 的升级版本，基本命令相同 `emacs`

`gedit` 可视化的文本编辑工具

注意，以上的前三个都是在终端中使用的(`vim` 和 `emacs` 也是一对冤家，如果你有兴趣，可以在网络上搜索相关内容)，而且使用中鼠标是可以没有的，所以，这也是 `Linux` 操作系统的一个特点，“可以完全脱离鼠标”，请熟悉前三个工具中的至少一个再开始进行之后的内容。

4. JOS 操作系统

JOS 是一个开源的操作系统，本课程的实验部分主要修改完善 JOS 源码，掌握操作系统的引导启动、内存管理、中断、调度、文件管理的方面的内容。

5. Qemu 模拟器

QEMU 是一套由 Fabrice Bellard 所编写的以 GPL 许可证分发源码的模拟处理器，在 GNU/Linux 平台上使用广泛。我们将使用该模拟器模拟硬件，然后在该模拟器上引导启动 JOS 代码，模拟操作系统中的各项任务。

本次课程的实验中，我们将使用 `qemu-6.828-1.7.0`（附件中提供：[lab/resource/qemu-6.828-1.7.0.zip](#)）

6. 实验所需配置

配置部分主要配置一些实验中可能需要的库和软件，以及实验中所需要的模拟器 `qemu`，另外，在实验过程中，如果出现问题可以通过网络搜索解决，很多问题是缺少库文件造成，届时可以再进行安装。

如果需要更新软件或者库文件，可以通过以下命令更新：`sudo`

`apt-get update`

`sudo apt-get upgrade`

以下为本次试验的配置：

- `sudo apt-get install build-essential`

Ubuntu 缺省情况下，并没有提供 C/C++ 的编译环境，因此还需要手动安装。如果单独安装 `gcc` 以及 `g` 比较麻烦，幸运的是，为了能够编译 Ubuntu 的内核，Ubuntu 提供了一个 `build-essential` 软件包。

- `sudo apt-get install binutils`

- `sudo apt-get install libgtk2.0-dev`

安装此部分时，可能会有包依赖的错误，此时可以通过以下两种方案解决：

a) 安装 `aptitude`，然后通过命令 `sudo aptitude install libgtk2.0-dev` 进行安装。

b) 安装 `gnome-core-devel`：`sudo apt-get install gnome-core-devel`，然后执行 `sudo apt-get install libgtk2.0-dev`。

- `sudo apt-get install git`

- 解压 `qemu`（在目录 `lab/resource/qemu-6.828-1.7.0.zip`）进入 `qemu` 目录

`cd qemu-6.828-1.7.0`

- 配置 `qemu`

`./configure --disable-kvm --prefix=/usr/local/qemu --target-list="i386-softmmu x86_64-softmmu"`

- `make`

- `sudo make install` 至此，配置结束。