

Linux 课程安排

课程简述

本课程是Linux User Group的学生成员面向全校各院系本科生开设的公共选修课。本课程以应用角度出发，带领学生从零开始了解开源Linux系统并学会初步应用。本课程主要讲授在Linux系统下的基础操作、文件系统、Shell脚本、数据处理、C/C++编程、网络配置与高级操作维护等知识。同时，本课程在讲授过程中将贯穿开源思想，并向学生普及开源世界的相关知识。本课程重视实践，学生通过学习课程内容即可动手使用Linux系统为自己和他人带来便利。

本课程在 2018 年春季学期进行，约 7 次，每次 75~120 分钟左右，地点固定，时间安排在周末，具体时间根据每周情况微调。

预备工作

安装 Linux

在第一次课程之前在东西区各举行一次 Linux Install Party，确保大家参与课程之前能有一个能使用的 Linux 环境。

安排讲师

开学前将每节课的讲师确定，如无特殊情况不做大的改动。
每次课程开始提前两周联系讲师准备课程使用的演示实验和讲稿。
每次的演示资料和讲稿要提前一周上传到 LUG FTP。

安排地点和时间

开学后根据同学们的习题课，课外活动等时间安排，每次课提前一周确定时间和地点并通知。安排一人向学校申请教室并和讲师沟通。

收集参与者联系方式

收集参加课程的同学联系方式，包括邮箱和联系电话，方便后续的通知。

课程内容

课表

课次	主题	讲师	时间
1	初识Linux(1)	李嘉豪	
2	初识Linux(2)	金孜达	
3	文件操作与软件安装	杨肇峰	
4	进程、服务和任务	马凯	
5	分区、备份与批处理(1)	邓胜亮	
6	分区、备份与批处理(2)	崔灏	
7	未定		

详细内容

第一部分 初识Linux

//这一部分的应用场景是为实验室安装远程服务器并且连接上去。

老师让学生了解20世纪系统的发展历程。

在20世纪，除了大名鼎鼎的Windows系统的起源DOS以外，还有开源的UNIX系统，它逐渐演化成了如今的MacOS系统和Linux系列，如今Linux系统的各种发行版(Ubuntu/Debian/CentOS/Arch)被广泛用于各种服务器。

老师要讲解Linux系统与常见的Windows和MacOS系统的区别。

Linux系统的核心在其命令行(或者说Shell)操作。

Linux系统对大小写敏感。

Linux系统大部分都是开源版本，旗下的很多程序也都是开源的，开源思想是Linux哲学的一部分。

老师要告诉学生Linux擅长的环境，告诉学生他们学会Linux系统以后可以做什么。

Linux系统在Shell操作下万能灵活，本身比较安全，可以担当服务器系统。

Linux系统下有很多其他系统没有的专业软件，而且是开源的。可以直接使用软件，或者阅读软件源码了解软件，甚至可以基于其他软件自己开发。

学生在学习Linux基础以后，可以为自己的实验室搭建专属服务器，已经有服务器的可以去维护。

老师要指导大家往自己的电脑上安装最新的某一Linux发行版。

介绍一下科大源Mirrors@USTC，告诉学生科大对Linux的重大贡献和对周边地区的辐射影响。

通过事先准备的U盘或者网路络上的镜像文件，让学生在电脑上安装某一Linux发行版，或者在虚拟机上安装系统。跟随系统安装的每一步，指导安装过程。

老师要演示如何简单使用图形界面，以及开启Shell进行操作。

介绍一下桌面，任务栏，资源管理器，开关机。

打开Shell，试着输入第一个命令（比如ping www.ustc.edu.cn，然后ctrl+c中断），修改命令的参数再试一次（比如ping -c 3 www.ustc.edu.cn）。这样就可以引导学生学会自助学习（比如 man ping）。

老师要带领学生进入开源圈子。

介绍若干网站（比如Ubuntu社区 community.ubuntu.com；Github github.com）。

邀请加入科大Linux QQ群，介绍Linux@USTC社团。

教导学生敢于提问，善于提问。

老师引导学生安装第一个程序。

可以先让学生改成科大源。

如果维护一个服务器，一定会不时要远程连接，因此需要为这个服务器安装ssh服务端。安装第一个程序ssh（比如sudo apt install ssh）并让它成为开机启动的守护进程(sudo systemctl enable ssh)。这里不要深入讲解apt和systemd。

在Windows/MacOS系统上安装SSH客户端。

测试一下远程连接。

老师开始初步介绍用户、用户组、文件和文件权限。

介绍用户和用户组，UID和GID的概念，介绍root用户。

查看自己的用户信息（id）。为系统添加一个组（groupadd）；为系统添加一个用户（useradd）；设置初始密码（passwd）；把新用户加进新组（usermod -g）；删除用户组（groupdel）；删除用户（userdel -r）。

Linux文件系统目录从根目录（/）开始，启动Shell的时候从家目录开始（root用户：/root/；一般用户：/home/xxx/）。查看自己所在的目录（pwd），进入某一目录（比如家目录cd ~），查看目录下的文件（ls -al）。

介绍文件和文件夹的读、写、执行权限，每个文件的所属用户和用户组。为文件修改所属用户（chown）或者用户组（chgrp），修改文件的权限设置（chmod）。

介绍执行用户切换（sudo），用户切换（su）。

老师介绍系统的关机和重启命令。

关机（shutdown），重启（reboot）。

第一周内容提炼

UNIX和Linux的历史(20)

Linux系列的特性(10)

Linux目前的主要用途(15)

安装Linux发行版(20)

操作图形界面(10)

操作Shell：ping(10)

介绍开源文化(5)

合计90分钟

第二周内容提炼

基础软件安装：apt, systemctl(5)

远程连接：ssh(5)

介绍用户和用户组(20)

目录操作：pwd, cd, ls(5)

介绍文件和文件夹的权限(20)

文件权限操作：chown, chgrp, chmod(15)

用户切换：sudo, su(15)

合计90分钟

第二部分 文件操作与软件安装

//这一部分的应用场景是处理实验数据，以及安装一些必须的工作软件或者系统软件。

老师讲解如何获取并阅读数据。

连接到远程主机下载数据（scp），或者从网站上下载数据(wget)。

创建目录（mkdir），并把下载的数据放到那里（mv, cp）。

讲解打包（tar）和常见压缩类型（.bz2/.gz），解压文件（tar -jxvf）。

查看数据（cat, less, head, tail）。

老师讲解如何操作数据。

对比一下两组实验数据（行比较用diff，字节比较用cmp）。

介绍文本编辑器nano/vim的用法，修改数据文件。

把修改好的文件打包压缩（tar -jcvf），上传(scp)，删除压缩包(rm)。

老师给学生介绍软件的下载、安装、更新和删除。

apt是一个很好用的软件管理器，它可以很方便地操作软件安装而无需担心复杂的问题，例如依赖。可以用安装mysql服务端举例。找到想要的软件的名称（apt search），安装起来（apt install）。更新软件信息（apt update），升级软件（apt upgrade）。删除不要的软件（apt autoremove）。

很多软件的安装需要其他软件作为前提，这就是依赖。删除这个软件的时候，会删除一些不再需要的依赖。介绍自动安装标记，可以将某个软件标记成手动安装以避免被自动除去（apt-mark）。

老师给学生介绍开源软件的安装特性。

一般情况可以直接用apt来安装已经编译好的二进制文件，但是很多时候有些软件可能没有专门为你的发行版准备二进制文件。因此为了让程序能在不同平台运作，很多作者会开放程序源代码，这时候就可以进行开源编译。开源编译是Linux系统的特色。

为了能够进行编译安装，需要先安装build-essentials。

老师演示编译安装一个开源软件（mpv）。

作为实验室的服务器，很可能需要一个网页服务，可以用apache2作为例子。

从官网上下载安装包，解压到/usr/local下，进入解压目录。浏览README（如果是apache2，会告诉你到INSTALL下阅读说明）。然后执行三步命令（./configure, make, make install），执行过程中稍微讲解一下每一步是干什么的。

把程序的帮助文件链接到man下（/etc/manpath.config）。

老师说一下执行目录的环境变量。

源码目录下的configure也是一个命令，但是它只能在源码目录下用./configure执行（或者说需要指定命令地址）；命令ls在/bin下，但随时随地可以直接用ls执行。因此引出\$PATH，让学生试着把加入\$PATH，在源码目录下就可以直接configure。随后让其移除，因为这样不安全。

老师要指出编译安装需要自己解决依赖问题。

apt一般会自动解决依赖问题，但是编译安装则只能自己解决。一般阅读README来找到依赖，或者通过./configure提示丢失的头文件找到依赖（apt-find）。给出一个例子。

老师让学生找一下安装好的软件放在哪里。

安装好的软件和命令一定有个存放地点，找一下他们的位置（which, locate, find），介绍这些命令的区别。

第三周内容提炼

远程下载/上传数据：scp, wget(10)

操作文件/目录：mkdir, mv, cp, rm(15)

文件打包压缩：tar(10)

文件查看：cat, less, head, tail(10)

数据差异比较：diff, cmp(5)

文本编辑器：nano, vim(15)

软件管理：apt, apt-mark(15)

开源软件和编译安装(20)

执行目录环境变量：\$PATH(5)

解决依赖问题(10)

命令和文件查找：which, locate, find(10)

合计125分钟

第三部分 进程、服务和任务

//这一部分的应用场景是日常监测执行学术任务的实验室机器的情况，并且自己写一些能后台执行/自动执行的任务。

老师告诉学生什么是进程。

进程是程序的实例，每个运行中的程序都是一个进程，每个进程有唯一PID。

因为命令也是程序，因此运行一条命令也会启动一个进程。

进程也可以产生进程，分别叫父进程和子进程，产生的过程是一个fork&exec。

老师讲解前台和后台的概念和使用。

对于一个Shell，它有前台和后台之分。运行在前台的进程接受标准输入流，运行在后台的进程无法接受标准输入流，但是仍然使用标准输出流和标准错误流。

运行一个前台进程（比如ping www.ustc.edu.cn），使用ctrl+z暂停到后台，使用jobs命令查看后台命令，发现是停止的。使用bg %1让它跑起来（屏幕上会出现ping 输出）。关闭这个后台任务（kill -9 %1）。讲解一下jobs, fg, bg, kill的详细用法。

运行一个后台进程（比如ping www.ustc.edu.cn &），观察其仍可输出但是无法用ctrl+c终止，使用fg %1移动到前台，ctrl+c关闭。

老师讲解一些与前后台进程有关的程序。

用nohup运行一个后台进程（比如nohup ping -c 10 www.ustc.edu.cn &），观察其已经没有输出，依据提示进入到nohup.out下查看结果。

安装tmux，新建若干会话，并行执行不同命令，再关闭它们。关闭当前Shell再打开一个新的Shell，列出tmux窗口，发现它们没有被清除。

老师演示监控进程和资源。

运行一个命令（比如nohup ping www.ustc.edu.cn &），查看一下当前运行的进程（ps）。

安装htop，用htop查看所有进程，搜索特定进程，查看资源占用率，把htop和Windows的任务管理器进程板块类比。

使用资源命令free（free -h）查看当前内存和交换空间的使用情况。

老师演示结束进程。

从ps或者htop中获得PID，关闭该进程（kill）。

老师告诉学生什么是服务。

那些常驻在内存中的进程，而且它们可以提供一些系统或者网络功能的，这个进程就是服务。

指出一些常见的服务（比如systemd, httpd, mysqld, crond）。

老师告诉学生怎么管理服务。

使用systemctl工具管理服务。用一个常见的服务作为例子（比如sshd），开启此服务（systemctl start），关闭此服务（systemctl stop），查看服务状态（systemctl status），系统自启动（systemctl enable），关闭自启动（systemctl disable）。检查一下服务开启和关闭时的不同（比如ssh 127.0.0.1在sshd关闭的时候是不会成功的），确认一下自启动的程序在重启以后确实是开启状态。

老师告诉学生什么是任务。

系统会执行各种任务，有些是要延期执行的一次性任务，有些是反复执行的例行性任务。更新locate数据库，删除临时文件等都是系统的自带任务。

老师解释如何安排任务。

利用atd服务和at命令安排延迟一次性任务，比如6小时后重启机器，配置/etc/at.allow和/etc/at.deny来给予at权限。

利用crond服务和crontab命令安排例行性任务，比如每天从网站上拉取数据，配置/etc/cron.allow和/etc/cron.deny来给予cron权限。

可以使用crontab的@reboot来安排开机启动的任务，比如给一个欢迎信息或者实验室规范提示。

crontab假设24小时不断电，但是这可能是不成立的。anacron在开机启动一次，判断有没有未被crontab执行的命令，如果有，就会延迟一定时间后执行（规则写在/etc/anacrontab里）。

第四周内容提炼

进程的概念(10)

前台进程和后台进程：jobs fg bg kill(20)

前后台进程工具：nohup, tmux(20)

进程监控工具：ps, free, htop(20)

服务的概念(10)

服务管理：systemctl(15)

一次性任务：at(10)

例行性任务：crontab, anacron(15)

合计120分钟

第四部分 分区、备份与批处理

//这一部分的应用场景是需要对实验室的重要数据进行备份和还原。

老师讲解一下文件和目录的特殊权限和特殊属性。

文件和目录有三个特殊权限：SUID/SGID/SBIT。通过chmod可以设置这些权限。

文件还可以通过chattr增加其他属性（比如+a为仅附加，+i为写保护），用lsattr查看。

老师说明一下Linux的文件系统的文件夹命名规范。

介绍Linux常用的标准FHS，讲解几个主要的根目录下的一级目录都是储存什么的。引导学生回忆，管理员的主文件夹就是/root，之前的源码和安装好的程序都放在/usr下，配置文件都在/etc下，基础的命令都放在/bin下等。

强调一下Linux系统视一切信息为文件，所以/dev下面的是表示磁盘的文件。

老师演示如何查看磁盘和文件信息。

使用df查看各个设备的信息，使用du查看一个目录和它的下属目录与文件的大小信息。

老师演示关于备份的整个过程。

介绍一下磁盘的主分区、扩展分区和逻辑分区。插入一块新磁盘，使用fdisk工具为一个新磁盘分区，或者为旧磁盘调整分区，用partprobe热更新分区表。分区以后，用mkfs命令为分区格式化。

解释一下Linux系统的挂载概念，与Windows的分区做类比。利用mount命令，在/mnt下新建一个目录，再把设备挂载到该目录下。

使用dump工具为一个目录进行完整备份或者增量备份，或者用rsync备份。

利用umount命令卸载备份磁盘。

老师演示关于还原的整个过程。

一般要做还原可能是数据删除或者硬件损坏，遇到莫名其妙的错误时，可以用fsck检查磁盘有没有坏道。有坏道的话还原可能也没用，就需要挂载新磁盘。

挂载备份盘，用restore命令还原整个备份或者抽取出一部分备份。卸载备份磁盘。

老师演示自动挂载。

有时候备份盘可以一直插在主机上，但是所有设备的挂载关机以后就丢失了，因此需要自动挂载。在/etc/fstab下修改文件来自动挂载备份盘。

老师演示批量处理数据文件。

例如现在已经有了100个完整备份，要清除一些太老的备份腾出空间，可以选择前90个备份，或者选择2个月以前的备份。进一步讲解find的用法，找到它们，用xargs来移动到另外的磁盘（配合mv）或者直接删除（配合rm）。

老师演示使用Shell脚本一次性处理多个事件。

先讲解一下脚本的顺序、判断和循环操作，以及参数读入（\$0, \$1, \$2...）。

写一个脚本。例如，接受三个参数，如果是autodump /path/to/file /path/to/backup，那就自动把/path/to/file目录备份起来自动命名，存到/path/to/backup，然后在/path/to/backup下的某一个仅附加(chattr +a)的日志文件里加一笔记录；如果是autorestore /path/to/backup /path/to/file那就读取/path/to/backup的一个日志文件，把其下最新的备份还原到/path/to/file下。更健壮更多功能的Shell可以留作作业。

第五周内容提炼

文件和目录的特殊权限：SUID/SGID/SBIT(15)

文件的隐藏属性：chattr, lsattr(10)

Linux目录标准配置FHS(10)

磁盘和目录信息：df, du(15)

磁盘分区：fdisk, partprobe(20)

磁盘格式化：mkfs(5)

挂载：mount, umount(15)

合计90分钟

第六周内容提炼

备份与还原：dump, rsync, restore(10)

自动挂载：fstab(10)

文件的查找：find(20)

参数代换：xargs(10)

Shell Script应用(40)

合计90分钟

第五部分 未定：从以下主题备选并细化

备选主题1：日志处理

//这一部分的应用场景是通过阅读系统日志来排查服务器问题。

//内容有讲解数据流，正则表达式，数据处理、阅读日志和处理日志。

备选主题2：网络配置

//这一部分的应用场景是为实验室局域网做配置。

//内容有讲解必要的网络知识，网络配置，网络工具（mtr, iproute2等），防火墙和端口映射。

备选主题3：Linux上编程

//这一部分的应用场景是在实验室中，一些源码必须在Linux上编译运行的情况。

//内容有gcc/g++编译参数，makefile，简单的gdb调试和一些python的使用方法。

老师向同学介绍gcc/g++的一些参数

- E:预处理文件
- S:编译到汇编代码
- c:编译到目标文件
- o:指定输出文件名字
- L:链接指定的目标文件
- I:将指定文件夹加入编译时环境变量
- O:指定优化级别
- g:保留调试信息

高级：

-fsanitize=

thread:多线程竞争冲突检测

address:内存泄露或越界访问检测

undefined:未定义行为检测

-std=Cxx:指定标准版本

-mxx:指定支持某些架构或者指令集