

高性能计算应用实践

实验一：Linux操作系统与Shell命令

郑海刚



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

本讲概述

- 课程介绍
- 实验一内容
 - Linux环境安装
 - Shell命令
 - man手册
 - vi编辑器
 - 版本管理Git

个人简介

- 实验课程
 - 数字逻辑设计、计算机组成、计算机设计与实践
 - 人工智能、高性能计算应用实践
- 指导的竞赛：ASC超算竞赛、并行计算竞赛
- 高性能计算云服务中心：[GPU集群](#)管理员

课程介绍

- 课程立项背景：服务于高性能相关竞赛， 2021首次开课
- 教学方式：先讲原理，再动手实验，部分需要看书预习
- 课程目标:
 - 了解CPU、计算机系统、超算系统的发展和基本结构
 - 以参考资料阅读为主
 - 熟悉Linux下多进程、多线程、openMP、MPI并行编程技术。
 - 以理论讲解、实验、编码调试为主

课程安排

| 序号 | 实验项目 | 学时 |
|----|-------------------|----|
| 1 | Linux操作系统与shell命令 | 2 |
| 2 | Linux环境下C编程与矩阵乘法 | 2 |
| 3 | Linux环境下调试与矩阵乘法优化 | 2 |
| 4 | Linux环境下多进程编程 | 2 |
| 5 | Linux环境下多线程编程 | 4 |
| 6 | OpenMP并行编程 | 2 |
| 7 | MPI并行编程 | 4 |
| 8 | HPL性能测试 | 2 |
| 9 | 综合实验：应用优化 | 6 |

课程资料

- 仓库地址: <https://gitee.com/hackenzhg/hpc>
 - 课件
 - 指导书
 - 代码

什么是高性能计算

- HPC (High-performance computing)
 - [Wikipedia](#): uses supercomputers and computer clusters to solve advanced computation problems.
- 主要两点
 - 硬件上：具有高计算性能的单机或集群系统，如DGX、神威太湖之光超算集群
 - 软件上：尽可能地利用系统的算力，如blas、MKL、cuda等
- 区别：互联网的高性能服务（高并发）、高性能处理器

参考资料

- 《深入理解计算机系统》 CSAPP, 兰德尔 E.布莱恩特著
 - 《计算机系统基础》 第2版, 袁春风 (南京大学)
- 《操作系统导论》 OSTEP, 王海鹏译
- 《并行计算导论》 第2版, Ananth Grama

实验1：Linux操作系统与shell命令

Linux介绍

- Linux很重要
 - 互联网服务器装的操作系统, [80%是unix/unix like](#)
 - 超算Top500的操作系统全部为Linux
- [Unix/Linux 发展历史图](#)

Unix like 介绍

- [Unix like](#) :类Unix, 行为上像unix操作系统, 不一定经过认证
 - 认证标准: [Single UNIX Specification](#)
 - 核心是[POSIX](#) (可移植操作系统接口) 标准
 - 系统级、用户级API, shell命令等
 - Linux 属于 [Mostly POSIX-compliant](#)
- [Linux distribution](#): Linux发行版
 - which includes the [kernel](#) and supporting system software and libraries
 - [推荐用Debian或Ubuntu](#)

Linux环境安装

- 安装方式：虚拟机、双系统、WSL
- WSL([Windows Subsystem for Linux](#))
 - WSL1： 2016年发布
 - WSL2： 2019年发布，推荐用该版本
 - 可以装不同的发行版
 - [Install Linux on Windows with WSL](#)
 - 先安装WSL
 - 再用WSL安装具体的发行版

shell及shell命令

- shell 是一个命令行解释器(man sh看一下)
- 常见的命令
 - ls 查看目录内容
 - pwd 输出当前目录
 - cd /usr 切换到其他目录
 - ...

```
$ ~ »cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
/usr/bin/tmux
/usr/bin/screen
/bin/zsh
/usr/bin/zsh
```

shell命令分类

- 功能上：文件编辑、应用软件等
- 实现上分为5类：可执行文件、内置命令、别名、保留字、shell函数
 - which ls
 - which cd
 - which top
 - type top # type查看命令的类型

GNU: GNU's Not Unix

- [coreutils](#): 常用的基本工具, ls、cat等
- [binutils](#): 二进制文件相关的工具
 - vi /usr/bin/top 乱码
 - objdump -d /usr/bin/top 查看代码段反汇编
- [GNU Software](#): 列出了GNU所有的软件, 包括coreutils、binutils

Shell小结

- unix, 一个命令只做一件事, 做到极致
- 要完成复杂的事情, 需要多个命令组合, 通过管道进行数据的传递
 - `ls | wc -l`
- 命令组合也是编程: shell脚本
- 熟悉shell后推荐使用zsh/fish

shell命令怎么学

- 不熟悉的命令，man 一下， 比如man ls
- man手册要怎么看
 - 主要小节：SYNOPSIS (使用方法大纲)、DESCRIPTION (功能说明)
 - 怎么上下移动或翻页？怎么查找关键词？怎么退出？
 - 选项、参数，长选项，短选项，选项的组合

man的用法

- man man 看一下
 - man 3 printf

The table below shows the section numbers of the manual followed by the types of pages they contain.

| | |
|---|---|
| 1 | Executable programs or shell commands |
| 2 | System calls (functions provided by the kernel) |
| 3 | Library calls (functions within program libraries) |
| 4 | Special files (usually found in /dev) |
| 5 | File formats and conventions, e.g. /etc/passwd |
| 6 | Games |
| 7 | Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7) |
| 8 | System administration commands (usually only for root) |
| 9 | Kernel routines [Non standard] |

- [man入门教程](#)
- man手册有中文版: <https://github.com/man-pages-zh/manpages-zh>
- 命令自带的--help选项提供简要的说明: ls --help

Linux下文件编辑：VI/Vim (1)

- VI的设计哲学
 - 大量时间花在阅读而不是在写代码上。所以，Vi 是一个多模态编辑器
 - normal模式、命令行模式、可视化模式
 - Vi 避免了使用鼠标，因为手指移动太多太慢
 - Vi 甚至避免用上下左右键因为需要太多的手指移动
 - 这样的设计哲学使得 Vi 成为了一个能跟上你思维速度的编辑器

Linux下文件编辑：VI/Vim (2)

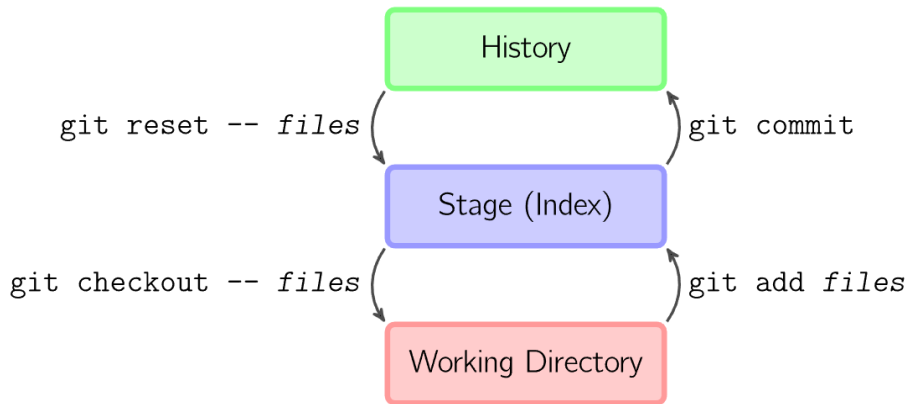
- 但对初学者不友好
 - [Stack Overflow: 帮助百万开发者退出Vim](#)
- vimtutor
 - 终端中直接运行vimtutor
 - 实际是个脚本，使用vim打开tutor.vim文件，跟着内容练习对应的操作

版本管理：git

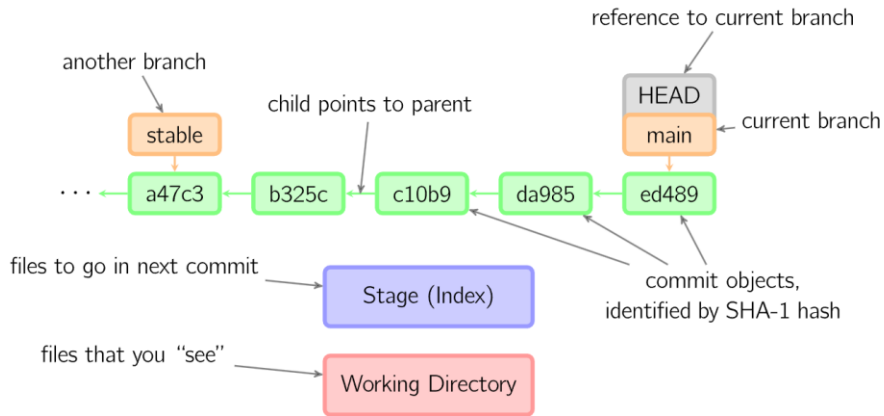
- 版本管理的需求一直有
 - 最原始的版本管理复制一份，命名xx-v1，全量备份
 - 早期的版本管理SVN
 - [Git](#)：分布式的版本管理，2005年由Linus开发用于Linux 内核的开发

图解Git

- git本地：工作目录、索引和仓库



- 版本提交，修改是增量备份



Git 远程仓库

- 没有远程仓库git仍可以正常在本地进行版本管理
- 有远程仓库方便备份、同步，实现分布式管理
- 远程仓库是指托管在因特网或内网中的项目的版本库。
- gitee、github、gitlab等远程仓库，或者私有的gitlab等服务器

资料推荐

- 工具类的细节多，很少有学校开课专门讲工具的使用
- 多阅读多实践，参考资料和阅读材料需要很多课后时间消化
 - [中科大Linux101](#)
 - MIT: [The missing semester of your CS education](#)
 - 南京大学: [jyy wiki](#), 计算机系统基础 ([ICS PA](#))

专业的学习方法

- STFW - Search The Friendly Web

- 英文关键词搜索

- 搜索渠道

| | 搜索引擎 | 百科 | 问答网站 |
|---------------|---|---|--|
| 推荐 使用 | 这里有google搜索镜像 | http://en.wikipedia.org | http://stackoverflow.com |
| 不推 荐使 用 | http://www.baidu.com | http://baike.baidu.com | http://zhidao.baidu.com http://bbs.csdn.net |

- RTFM - Read The Friendly Manual

- RTFSC - Read The Friendly Source Code

实验内容&要求

- 根据实验指导书完成实验内容
- 如果已经有Linux环境的，可以选择不做
- 后续实验所要求的代码、报告都要求推送到远端仓库，不同实验归档在不同子目录。
- **课后阅读：**《深入理解计算机系统》第7章 “链接”