软件工程工具与调试

“工欲善其事，必先利其器”

- 从一些机械、繁琐的工作中解脱出来，集中精力解决软件开发中更加本质的问题。
- 让工具去做计算机可以做的事情，让人做必须人做的事情。

工具

- 软件需求工具，用于描述系统的需求、需求建模和需求跟踪。
- 软件设计工具，用于描述软件设计。
- 软件构造工具，用于程序设计、编码和编译，主要是集成开发环境（Integrated Development Environment, IDE），提供程序语言的代码编辑器、代码生成器、预编译、编译、链接、集成、运行环境和调试器。
- 软件测试工具，用于对系统、子系统、模块或单元进行测试的工具。
- 软件开发支撑工具，包括软件配置管理工具、项目管理工具、软件工程工具等等。

集成开发环境（Integrated Development Environment, IDE）

- 不同工具完成不同任务：使用文本编辑器编辑代码；使用编译器进行编译；使用链接器进行链接；使用其它独立工具进行调试、版本控制、部署、安装程序制作等工作。
- 复杂、学习难度大、信息难以共享。

IDE

- 将各种软件编程工具集成到一个应用程序中，提供一致的界面，并有效协同各个工具的使用，使得工具可以使用另外工具提供的信息，提高了软件开发人员的效率，该应用程序称之为集成开发环境。

常见IDE功能

- 编辑源代码
- 输入代码时进行代码语法检查，提供即时反馈给程序员
- 根据语法，高亮代码
- 编译、链接
- 调试
- 浏览代码结构图

其它IDE功能

- 输入代码时进行代码自动完成
- 自动创建类、方法和属性
- 可视化编程支持
- 自动化重复任务
- 集成源代码控制工具
- 集成Web Server
- 集成测试工具
- 重构
- UML支持
- 多语言支持
- .....

IDE使用习惯

- Unix平台仍然有很多程序员使用独立命令行编程工具，也有很多程序员使用Emacs和Vim。
- 微软Windows平台上很少有程序员使用命令行工具，大部分使用具有图形用户界面的IDE，比如Microsoft Visual Studio等。
- 苹果公司Mac OS平台上的常见IDE为Xcode。
- 另外，开源IDE，例如Eclipse等，在各个平台上都可以使用。

Eclipse

- Eclipse（http://www.eclipse.org/）是一个跨平台的支持多种编程语言的开源IDE。
- Eclipse源于IBM公司的商业项目VisualAge IDE，于2001年11月开源。Eclipse的设计目标除了满足用户对IDE的需求以外，还设计了灵活的插件（Plugin）结构，程序员可以根据自己的需要来扩展功能。
- Eclipse是当前使用广泛的Java IDE之一。

代码目录管理

- 超过10000个源代码文件；超过500个资源文件；超过5000个测试脚本文件；超过100个库文件（不同人员用不同版本的库文件）？
- 创建工具约定目录结构Maven.
- 有常见目录使用习惯：Ant

常见Ant文件目录机构

目录名	目的
项目目录	包括配置文件 build.xml 和本表中其它所有子目录
src	包括项目源代码文件
test	包括项目测试代码
lib	包括项目中需要的库文件
dist	部署文件，可能为 jar 或 war 文件等

版本控制

- 软件版本控制（或源代码控制，英文常见为：revision control，或version control，或source control）
- 软件配置管理，简称SCM（Software Configuration Management），用于跟踪和控制软件变更。它应用于整个软件工程过程，通常由相应的工具、过程和方法学组成。软件版本控制是软件配置管理工作的一部分。
- 软件配置管理包括：版本控制、变更管理和过程支持。

Why

- 程序的备份?
- 不同程序版本比较，回退?
- 团队成员协作?
  - 例如一个团队成员基于另外成员的代码进行了大量的开发，但是另外的成员一个月前就删除或改写了相应代码，这样他的工作将全部浪费。

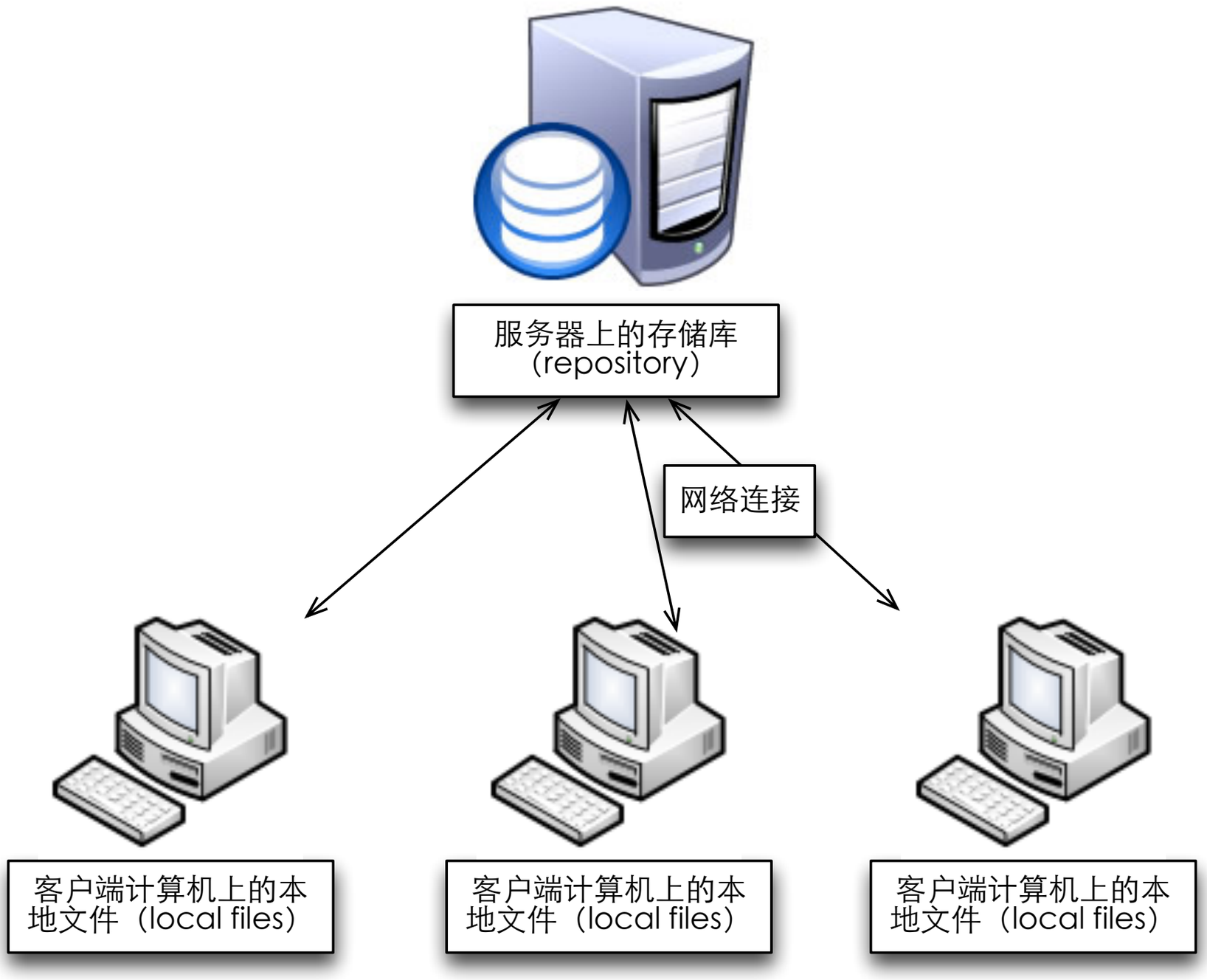
概念

- 版本控制是软件配置管理的核心和基础，也是很多其它软件工程活动（比如集成）的必需条件。
- 版本控制的对象是软件开发过程中涉及的计算机文件，包括最为重要的源代码，以及库文件、图形资源文件、计划文档、需求文档、设计文档、测试文档等可以存储为计算机文件的有关资源。

组成

- 常见版本控制系统由服务器和客户端组成。
- 服务器使用项目“资源库（Repository）”存储受版本控制的所有文件以及文件的完整修订历史。
  - 通常一个项目会在服务器上建立一个资源库，项目所需文件全部存储在该资源库中。
  - 该资源库具有独立的用户访问控制，确保具有授权的开发人员才可以进行相关操作。
  - “增量存储”

- 程序员使用客户端来使用版本控制系统。（图形工具、命令行工具）
  - 客户端最基本的功能包括文件或目录的检出（Check Out或更新Update）和检入（Check In或提交Commit）。
  - 程序员在本地建立一个和服务器端资源库相对应的本地工作目录（或称为工作空间、工作拷贝），该目录通常只保存资源库中文件的最新版本（有时也称为快照，Snapshot，或Mainline）。
  - 检出操作从项目资源库中将文件的最新修订版本拷贝到本地工作目录。版本控制软件会检查本地工作目录中文件有无更新，如果有更新，检入操作将本地工作文件作为新的版本复制回资源库。


服务器上的存储库
(repository)
网络连接
客户端计算机上的本地文件（local files）
客户端计算机上的本地文件（local files）
客户端计算机上的本地文件（local files）

版本

- 在版本控制系统中，对一个文件的修改通常用数字或字母表示。
  - 例如，初始的文件被标识为“版本1”，然后第一次的修改被标识为“版本2”，后继的修改也会按照同样的方法进行标识。
- 每个版本都会和时间戳（Timestamp）以及修改的用户相关联。程序员可以对同一个文件的每个版本进行比较，回退到任何一个已经标识的版本，或者对于某些文件类型进行合并。

冲突

- 不同程序员同时编辑一个文件，冲突。
- 文件锁（File Locking）和版本合并（Version Merging）

其它常见操作

- 标签（Labels，Baselines，Tags）
  - 标签用于标识项目快照，即为某一时刻项目所有文件的指定一个标签，是用于标识文件集的编号方案。标签常用来标识项目的发行版本，分支或里程碑。用户也可以方便的将整个项目按照标签进行恢复或检出。

- 冲突（Conflict）
- 分支（Branch）
  - 在某个时刻，版本控制系统下的文件集可以进行分支操作，以后两份文件集可以进行完全独立的开发。通常创建分支来尝试新功能，或特定软件产品版本，同时不影响开发的主分支。
- 导出（Export）
  - 导出指从资源库中取得文件，它和检出操作类似，但导出会创建一个干净的目录结构包含所有文件，但不包含通常工作目录中保存的用于版本控制的元文件（Meta File）。

使用规范

- 在开始一个新的软件开发活动之前应进行一次检出，确保新的工作基于团队最新的软件版本，避免在旧的软件版本上工作造成混乱。
- 在完成了一部分工作并确认正确后进行检入，将自己的工作同步给团队其他成员。
- 开发人员应当提高使用版本控制系统进行检入、检出的频率，现代软件开发过程通常建议每天至少进行一次检入。

版本控制工具

- CVS
- SVN（推荐使用）
- GIT
- Rational ClearCase

调试

- 什么是软件bug？软件bug有时也被称为错误、缺陷、问题、故障、失效等（有些文献严格区分这些词的用法，在本书暂不做严格区分），用于描述软件系统产生了非预期的行为。
- 调试（Debugging）就是理解系统的行为，发现隐错（Bug）的根源并去除它的过程，是软件开发中的重要活动之一。
- 调试与测试不同，测试的主要工作在于发现bug，而调试的目的是去除bug。

Bug

- 没有人愿意使用充满bug的软件。
- 有bug的软件有可能造成生命损失！
- Therac-25

调试基本过程

- ①重现bug。重现测试或其它方式发现的问题。有些bug很难重现，比如和多线程同步有关的一些bug。
- ②定位bug。对bug提出尽可能多的假设，并通过各种调试工具和方法确认bug可能出现的区域。

- 在进行调试时，程序员要分析软件的内部构造，使用各种调试工具和方法来逐步缩小可疑代码区域，寻找错误的根源。

- ③改正bug。在改正bug之前需要理解问题的本质，并理解整个程序。Bug是系统非预期的行为，那么在寻找bug的时候必须正确理解系统的行为，这样才能改正bug。

- 调试程序时，要避免引入新的bug。

使用输出语句调试

- 使用类似System.out.println()的语句来输出所需要观察的值，但这种方法对于大规模的程序而言不太现实的：
  - 一是因为调试过程中可能的输出值太多，难以在屏幕上找到需要的值；
  - 另外，这会对源程序进行修改，程序员必须保证在调试完成后恢复所有的代码，非常容易引入新的bug。

调试器常见调试操作

- 断点（Breakpoint）
- 调试中语句运行控制
  - Step Into：跳入，进入到某个方法中。
  - Step Over：单步跳过突出显示的代码行并执行同一个方法中的下一行（或者它在调用当前方法的方法中继续）。
  - Step Return：单步跳出正在执行的方法。
  - Resume：继续执行程序。

准备测试

- 理想状况：程序没有bug，但这不可能！
- 为了保证软件行为能够符合人们的预期，软件开发人员需要对软件进行验证（Verification）和确认（Validation）工作。
  - 验证：指确定某个工作阶段是否正确完成的过程，在每个工作阶段结束时进行。我们是否正确地构建了软件（软件是否符合软件规格说明书）？
  - 确认：确认是在产品交付用户之前进行的深入细致的评估，它的目的是确定整个产品是否满足用户期望。我们是否构建了正确的软件（软件是否是用户需要的）？

测试定义

- “使用人工和自动手段来运行或测试某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别。”

测试方法

- 测试是为了发现程序中的错误而执行程序的过程。
- 基本的测试方法模拟软件的运行给出一个输入，观察它的输出，如果和预期一致则认为测试通过。
- 测试用例（Test Case）是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果，以便测试某个程序是否正确工作。
- 手工、自动化。