封面

内封

摘要

目录

前言

自从世界上第一台全自动电子计算机“埃尼阿克”诞生以来，计算机技术的发展极其迅速。在移动互联网及大数据的背景下，计算机已经融入到人类生活的方方面面。从衣食住行到工业生产，计算机已经成为人类社会不可或缺的重要组成部分。随着计算机科学与技术的快速发展而来的各种规模的软件工程，常常需要整合人力以及各种软件开发工具，这时版本控制的概念应运而生。

版本控制作为软件工程的一种技巧，需要让项目中的多个开发人员的开发进度保持同步。在软件开发过程中，参与项目的人数多不代表一定可以加快项目的开发速度，因为软件产品的抽象性，人员之间的沟通与管理极有可能占用很多资源，导致整体开发效率下降，而版本控制系统的作用就是协调开发人员，同步不同开发者的进度，尽可能减少版本管理占用的资源。

现代软件开发已经不是“软件=数据结构+算法”的模式，而是要适应各种变化的需求，使软件架构有足够的灵活性，不至于因为需求的变化而推倒重来。现代软件源于互联网的发展，互联网使得软件进入新的时代。互联网深入生产生活的方方面面，因此需要处理一些难以用算法表达的业务逻辑，如银行的金融业务不仅很复杂，很难用算法表示，而且经常调整，导致需求多变。这就需要版本控制适应变化的需求，可以进行同一系统下的分支管理。

一个算法可以从源代码中识别出来，而业务逻辑则很难从代码中看出来。一些企业级软件即使留下源代码，后来者也很难明白其中的业务逻辑。随着老一代程序员的退休，他们也将业务逻辑带走了。以至于后来者不敢轻易重构遗留代码，怕一个误解造成巨大损失。为了解决这样的问题，版本控制系统需要记录下迭代过程中的版本描述，以简化维护人员分析项目的过程。

源代码是开发人员的产品，因此对版本控制系统的安全性及稳定性有较高的要求。软件及硬件上的错误是无法预料的，同时也很难避免，因此对于版本控制系统的文件系统这方面，分布式架构是一个非常好的选择。分布式系统具有较高容错性，将文件操作等较慢的操作分发给多个服务器也有助于提高文件读写效率，采用RAID技术的底层文件系统对文件安全也有很大保障。

目前主流的版本控制系统有github、svn、rcs和cvs等，