

软件克隆及其优点

吴 民 齐向彤 涂奉生 (南开大学计算机与系统科学系 300071)

摘要:随着 linux 等自由软件的兴起,软件克隆正逐渐引起人们的重视。本文通过一个软件开发实例,介绍了如何在软件开发中运用克隆思想,结合其他例子,说明软件克隆对缩短软件开发周期,提高软件质量和生存力都起着十分重要的作用。

关键词:克隆 Unix Linux

一、引言

现在,生物科学中克隆(clone)技术是尽人皆知的。而在计算机界,尽管没有山羊多利诞生时的轰动反应,越来越多的克隆软件却已经或正在诞生,逐渐改变着软件产业的面貌。其中比较著名的是克隆 Unix 操作系统得到的 Linux 和 FreeBSD 操作系统,克隆图形窗口界面标准 Motif 得到的 Lesstif,等等。这些软件属于自由软件,可以免费获得,而功能接近甚至等于商业软件,因此具有一定的竞争力,在市场上占有一定的份额。

尽管被克隆的软件形形色色,各式各样,软件克隆时却遵循着同一个原则:虽然克隆得到的软件的内部的实现方式不同于原软件,但是遵循的原理是相同的,且保持与原软件外观上的一致性。也就是在一定程度上,从表面看,原软件和克隆得到的软件是相同的。使用运行 Linux 或 FreeBSD 的 PC 机时,如同使用运行 Unix 的工作站一样;在 Motif 下编译的软件,基本上可以在 Lesstif 下编译,生成的执行程序在 Lesstif 库的支持下运行效果与 Motif 支持的运行效果相同。外观的一致性软件克隆的主要特征。

笔者想通过这篇文章,初步阐述在软件开发中,克隆思想发挥的作用,克隆对软件有什么影响,对软件开发过程有什么影响,以及如何进行软件克隆等问题。希望本文能引起大家对软件克隆思路的重视,对广大软件设计者有所启发。

二、软件克隆的优点

如此众多的著名软件都有相应的克隆版本,说明软件克隆思想必定存在着潜在的优点。许多出色的软件的设计思想都是非常巧妙的,是经过反复推敲的,其中存在

必然的道理。克隆别人的软件可以借鉴和吸收别人的优秀思想,从而提高自己的软件的质量,缩短软件的开发周期。这是克隆思想给软件开发带来的直接的好处,而从软件生命力方面考虑,克隆也给软件带来间接的好处。

一个软件的生命力,也就是有多少人在使用它,是软件是否成功的一个重要标志。软件设计者一般都想到以提高自己软件的性能来战胜对手,吸引广大的用户。然而这些年软件市场的激烈竞争告诉人们,技术不是取胜的关键,易用性,特别是保持大量的熟悉自己软件的用户,才是使软件取得成功的首要因素。而采用软件克隆,可以使软件诞生初期,就有大量的熟悉用户存在,这无疑增加了软件的生命力。

下面结合 Linux 操作系统具体地谈谈克隆为软件带来的巨大好处。Linux 近几年来飞速发展,成为自由软件界的主要操作系统。人们一谈及 Linux 首先想到自由软件。诚然, Linux 的成功很大程度上得益于它是自由软件,可以免费获得使用权。但是, Linux 是 Unix 的克隆也是一个不容忽视的主要因素。

Unix 是目前计算机业界的主流操作系统之一,有近三十年的发展史,在多种不同体系结构的计算机中都有实现。Unix 的创始人, Ken Thompson, Dennis Ritchie 和 Brian Kernighan 在最初设计 Unix 时,借鉴了许多早期操作系统(如 Multics)的经验和教训,因此 Unix 的设计思想是比较成熟的。伴随着 Unix 的发展,诞生了程序设计语言 C, 互联网协议 TCP/IP, 图形窗口环境 X-windows 等一系列极大推动计算机工业发展的重要产物。Unix 以其出色的稳定性和高效性,获得普遍的青睞,在工商业有广泛的应用。大量的一般用户熟悉 Unix 的使用,大批计算机专业工作者熟悉 Unix 的管理和 Unix 的开发。

Linux 是 Unix 的克隆,它的源代码没有一行与 Unix 源代码相同,但继承了 Unix 的设计思想,内核小而效率高,其管理进程,内存,以及外部设备等的方式也与 Unix 基本相同。Linux 为应用程序提供的开发环境也与 Unix 相同, Linux 内核的系统调用几乎包括所有的 Unix 系统调用。这使一个熟悉 Unix 开发的程序员很容易掌握如何在 Linux 下开发程序,而且使向 Linux 移植软件比较容易。一般的 Linux 发行版本中,除了 Linux 系统内核外,都配有基本的系统软件,如编辑器 vi,编译器 gcc 等。在大多数情况下,使用和管理 Linux 与使用和管理 Unix 没有任何区别。

上面所说的种种情况,都导致自 Linux 诞生之日起,就有大批能够使用 Linux 的人存在,有大批能够开发 Linux 软件的程序员存在。这些人的存在,是 Linux 能够在操作系统的激烈竞争中生存和迅速发展的主要基础。可以说,克隆使 Linux 获得较强的生命力。Linux 自诞生之日至今仅经历了短短的八年时间。这期间, Linux 基本上是在没有大软件公司的支持下发展的(这一状况现在发生变化,许多著名软硬件公司已开始支持 Linux),却在技术上,达到目前支持多种物理设备,支持 Windwos 网络, Novell 网络,支持多处理机,内核的稳定性不逊于任何商业操作系统。这一成果得来殊为不易。克隆 Unix,借鉴 Unix 设计的经验,应是它能如此优异的重要原因。

三、一个软件克隆实例

在本节中,笔者介绍自己进行软件克隆的实例。虽然该软件规模不大,但是克隆的思想,克隆的优点都表现得比较充分。

1. NMS 简介

NMS 是在一些银行单位中使用的网络通信系统,一般用它配合 OLTP(储蓄后台), OLTC(储蓄前台), GLTP(会计后台), GLTC(会计前台)等上层业务程序完成银行中的储蓄和会计业务,并具备通存通兑的能力。

由于 NMS 的开发时间比较早,因此 NMS 存在一些不足。首先, NMS 仅支持串行通信方式,必须用串行线将主机连接起来,才能使用 NMS 系统。而其他网络设备如以太网, X.25 网, NMS 都是不支持的。其次, NMS 使用的协议是特别设计的,不是 Unix 下的标准网络协议 TCP/IP。

在网络技术日臻成熟的今天, NMS 显得有些落后,有必要对它进行改造,使它建立在 TCP/IP 协议基础之上。这样, NMS 就可以通过各种网络设备通信,甚至在将来更好的网络环境下,也可以不加改动地运行 NMS。

2. 剖析 NMS

笔者最初接触 NMS 时,没有拿到 NMS 的源程序,加上 NMS 的文档中技术细节比较少,因此需要自己动手剖析 NMS。笔者采用的方法主要是观察系统运行时的信息交换。

经过反复的摸索,笔者对 NMS 的结构和运行机制有了一定的了解,简述如下。NMS 系统由多个进程组成,其中主要有 msx, msx_tx, msx_rx 等进程,它的核心进程是 msx, msx 通过共享内存与其他进程(OLTP, OLTC, GLTP, GLTC, msx_tx, msx_rx 等)交换数据。每个 OLTP(GLTP)进程或 OLTC(GLTC)进程对应一块共享内存, msx_tx, msx_rx 也分别对应一块共享内存。这些进程仅读写自己对应的共享内存的数据。msx_tx 和 msx_rx 进程主要负责在串口上读和写数据。所有进程组成的系统可以用下面的示意图描述。

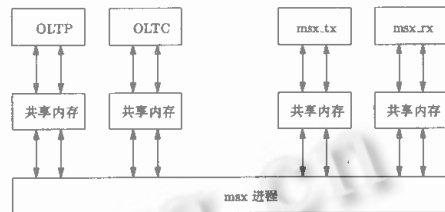


图 1 原 NMS 示意图

完整的系统(包括 NMS, OLTP, OLTC, GLTP, GLTC等)运行时,一个交易是以柜员操作 OLTC 或 GLTC,发出交易请求开始的。下面仅以储蓄为例,介绍交易全过程。OLTC 将交易请求通过共享内存传送给 OLTP, OLTP 在查询和更新数据库后,将结果通过共享内存交给 msx, msx 根据结果中的源地址将它通过共享内存交给相应的 OLTC;如果请求中的目的地址是其他主机, msx 根据自己保存的路由表将它交给相应的 msx_tx, msx_rx 通过串行线传送请求至另一主机,由该机的 msx_rx 进程接收并交给该机的 msx, msx 进程仍要根据请求中的目的地址进行判断。如果该机不是目的主机,则 msx 要再次根据路由表将请求交给适当的 msx_tx,

通过串行线传送至另一主机,直至请求抵达目的主机,由目的主机的 `msx_rx` 接收并将其交给 `msx`。`msx` 把请求交给 `OLTP`,`OLTP` 在查询和更新数据库后,将结果交给 `msx`,`msx` 根据结果中的源地址通过查路由表将结果交给相应的 `msx_tx`,由 `msx_tx` 通过串行线传送至另一主机。结果以与传送请求相同的方式在主机之间传送,直至抵达源主机,由 `msx_rx` 接收并交给 `msx`,`msx` 再交给相应的 `OLTC`。`OLTC` 显示结果于屏幕上,交易结束。

实际上在剖析 `NMS` 时,笔者在一些关键环节的分析中就运用了克隆思想,以清楚地掌握每个进程的工作内容。例如,为了了解 `msx` 进程的功能,笔者设计了一个假 `msx` 进程,它虽然完成一些初始化工作,并在系统运行时标记自己为 `msx`,但是在关键的数据交换中却什么也不做。将包含假 `msx` 进程的系统的运行情况与原系统的运行情况相比较,就可以获得 `msx` 进程的行为细节和功能。

3. 克隆 `NMS`

改造 `NMS` 的目标是使之建立在 `TCP/IP` 协议基础之上。为了实现这一目标,有多种改造方案可供选择。人们一般会想到通过修改 `OLTP`,`OLTC`,`GLTP`,`GLTC` 等源程序,以使整个系统的信息交换建立在 `TCP/IP` 基础之上。然而在清楚地认识 `NMS` 系统的基础上,我们可以大胆的提出克隆 `NMS` 的方案:即编写新程序替代 `msx`,`msx_tx` 和 `msx_rx`,但保留 `NMS` 系统的进程间共享内存通信机制。这一方案的优点是明显的。

(1)`OLTP`,`OLTC`,`GLTP`,`GLTC` 等程序均保持不变,减少了编程的工作量。而且在安装时也只是用新系统替代 `NMS` 系统,工作比较小。

(2)从上面的介绍可以看出,`NMS` 系统和 `OLTP`,`OLTC` 等应用之间的通信协议是严格的和完整的。克隆

方案遵守这些协议,因此出现错误的可能性比较小。

(3)`NMS` 的进程间通信机制是比较可靠的,笔者沿用它,而不再考虑其他通信机制,就可以集中精力以提高软件的质量。

笔者采用克隆方案,在很短的时间内就完成了 `NMS` 的初步改造工作,速度之快是采用其他方案无法比拟的。改造后的系统完全可以与原 `OLTP`,`OLTC`,`GLTP`,`GLTC` 等应用程序配合,完成银行业务,如同原 `NMS` 系统一样。新系统由于建立在 `TCP/IP` 协议上,因此可以支持局域网,广域网等,而且调试也比较方便。在试用中发现,有时 `OLTP` 需要和 `OLTP` 交换数据(跨行储蓄或支取现金时,银行间的帐目要平衡),这是笔者为新系统写程序时未想到的。虽然如此,新系统却能正确地处理这种数据传输,这也说明克隆方案是可取的。

四、结论

有时,软件克隆被简单地认为是重复前人的工作,然而它实际上是继承和发展前人的成果。科学技术的每次进步都是建立在前人工作的基础上的。计算机软件是需要高投入,并且对市场依赖性很强的产业,采用克隆方式进行软件开发,有时是需要的,并能给开发者带来比较好的收益。软件克隆是软件发展进步的表现。

参考文献

- [1] 莫里斯,贝奇,UNIX 操作系统设计,北京,北京大学出版社,1989。
- [2] 孙玉方,实用 UNIX 系统—使用与管理,北京,清华大学出版社,广西科学技术出版社,1993。

(来稿时间:1999年3月)