Debian Edu / Skolelinux ITIL 手册

出版日期：2026 年 11 月 01 日

Contents

1 集中化管理的知识分享 1

2 许可证 1

2.1 致谢 . . . 1

2.2 背景 . . . 2

3 服务支持 2

3.1 服务台 . . . 2

3.1.1 任务和角色 . . . 3

3.1.2 预计用时 . . . 4

3.1.3 清单 . . . 5

3.2 事件管理 . . . 6

3.2.1 清单 . . . 6

3.2.2 计划与实施 . . . 7

3.2.3 与运行事件相关的行为 . . . 7

3.2.4 角色 . . . 7

3.2.5 关键点 . . . 7

3.2.6 工具 . . . 8

3.3 问题管理 . . . 8

3.3.1 问题管理的步骤 . . . 8

3.4 配置管理 . . . 9

3.4.1 计划 . . . 9

3.4.2 配置项目（CI）的管理 . . . 9

3.4.3 计划与安装 . . . 9

3.4.4 清单 . . . 10

3.4.5 与其它过程的关系 . . . 10

3.4.6 配置管理的工具 . . . 10

3.5 更改管理 . . . 10

3.5.1 行为 . . . 11

3.6 版本发布管理 . . . 11

3.6.1 基本 . . . 11

3.6.2 可靠软件库（DSL） . . . 12

3.6.3 配置与硬件的数据库 . . . 12

3.6.4 构建管理 . . . 12

3.6.5 测试 . . . 12

3.6.6 后退方案 . . . 13

3.6.7 优点和可能的问题 . . . 13

3.6.8 计划与实施 . . . 13
3.6.9 行为 . . . 14
3.6.10 工具 . . . 14
3.6.11 与其它过程的关系 . . . 14
3.7 用于运行支持的工具 . . . 14
3.7.1 工具的类型 . . . 15
3.7.2 选择工具时的评价标准 . . . 15
3.7.3 产品培训 . . . 16
3.8 在实施服务支持的一开始就进行计划 . . . 16
3.8.1 实施服务支持 . . . 16
3.8.2 可行性研究 . . . 16
3.8.3 确定当前的情况 . . . 16
3.8.4 项目计划的一般指导原则 . . . 16
3.8.5 项目复核与报告 . . . 17

4 服务的交付 17
4.1 服务层次管理 . . . 17
4.1.1 通用清单 . . . 17
4.1.2 计划 . . . 18
4.1.3 实施 . . . 18
4.1.4 运行情况 . . . 18
4.1.5 服务层次协议（SLA）的内容 . . . 18
4.2 财务管理 . . . 19
4.2.1 预算 . . . 20
4.2.2 会计 . . . 20
4.2.3 会计与记账的计划 . . . 20
4.2.4 实施 . . . 20
4.2.5 日常操作 . . . 20
4.3 容量管理 . . . 20
4.3.1 监控 . . . 21
4.3.2 分析 . . . 21
4.3.3 配置 . . . 21
4.3.4 实施 . . . 22
4.3.5 准备容量计划 . . . 22
4.4 可用性的管理 . . . 22
4.4.1 可用性的衡量 . . . 23
4.4.2 基础设施 . . . 23
4.4.3 《单点失效》 . . . 23
4.4.4 风险管理 . . . 23

4.4.5 测试 . . . 24
4.4.6 设计的改进 . . . 24
4.4.7 可用性的计划 . . . 24
4.4.8 恢复的计划 . . . 24
4.5 服务连续性 . . . 24

5 ICT 基础设施管理 25
5.1 设计与计划 . . . 25
5.2 部署 . . . 26
5.2.1 展开过程中的角色 . . . 27
5.3 运行 . . . 27
5.4 配置项目 . . . 28
5.5 技术支持 . . . 28

6 设计和计划的例子 28
6.1 计划的背景 . . . 29
6.2 对 ICT 工具和服务期望什么 . . . 29
6.3 所需的技能 . . . 30
6.4 投资 . . . 31
6.4.1 学生 . . . 31
6.4.2 教师 . . . 33
6.4.3 推荐的技术发展预算 . . . 34
6.4.4 软件、学习平台和服务 . . . 34
6.4.5 清单的集中 . . . 34
6.4.6 软件 . . . 34
6.4.7 学习平台 . . . 35
6.4.8 在线服务 . . . 35
6.5 运行中资源的使用 . . . 36
6.5.1 运行的角色 . . . 36
6.5.2 运行和支持成本 . . . 38
6.6 选项的总结 . . . 39
6.7 推荐 . . . 39
6.8 附件 . . . 40

7 设立基础设施 40
7.1 网络架构 . . . 40
7.1.1 解决方案 . . . 40
7.1.2 例外的处理 . . . 40
7.1.3 确认 . . . 40
7.1.4 更新配置数据库 . . . 40

7.2 服务器的配置 . . . 40
7.2.1 组合服务器，作为组合的解决方案 . . . 40
7.2.2 Skolelinux/Debian-Edu 中配置的描述 . . . 40
7.2.3 解决方案 . . . 41
7.2.4 例外的处理 . . . 41
7.2.5 确认 . . . 41
7.2.6 更新配置数据库 . . . 41
7.3 硬件服务器 . . . 41
7.3.1 解决方案 . . . 42
7.3.2 例外的处理 . . . 42
7.3.3 确认 . . . 42
7.3.4 更新配置数据库 . . . 42
7.4 客户端计算机 . . . 42
7.4.1 客户端类型的表格 . . . 43
7.4.2 解决方案 . . . 44
7.4.3 例外的处理 . . . 44
7.4.4 确认 . . . 44
7.4.5 更新配置数据库 . . . 44
7.5 交换机 . . . 44
7.5.1 解决方案 . . . 44
7.5.2 例外的处理 . . . 44
7.5.3 确认 . . . 44
7.5.4 更新配置数据库 . . . 44
7.6 无线接入点 . . . 44
7.6.1 解决方案 . . . 45
7.6.2 例外的处理 . . . 45
7.6.3 确认 . . . 45
7.6.4 更新配置数据库 . . . 45
7.7 防火墙 . . . 45
7.7.1 解决方案 . . . 45
7.7.2 例外的处理 . . . 45
7.7.3 确认 . . . 45
7.7.4 更新配置数据库 . . . 45
7.8 路由器 . . . 45
7.8.1 解决方案 . . . 45
7.8.2 例外的处理 . . . 45
7.8.3 确认 . . . 45
7.8.4 更新配置数据库 . . . 45
7.9 设立简单的防火墙 . . . 45

7.9.1 解决方案 . . . 45
7.9.2 例外的处理 . . . 45
7.9.3 确认 . . . 45
7.9.4 更新配置数据库 . . . 45
7.10 设置： . . . 45
7.10.1 解决方案 . . . 46
7.10.2 例外的处理 . . . 46
7.10.3 确认 . . . 46
7.10.4 更新配置数据库 . . . 46

8 实用命令 46
8.1 对 4 GB 内存的支持 <-- 已包含在配置管理中 . . . 46
8.1.1 例外的处理 . . . 48
8.1.2 确认 . . . 48
8.1.3 更新配置数据库 . . . 49
8.2 管理软件包（apt-get） . . . 49
8.2.1 例外的处理 . . . 50
8.3 更新软件包档案 . . . 50
8.3.1 例外的处理 . . . 50
8.3.2 确认 . . . 50
8.4 更新到新的软件包 . . . 50
8.4.1 警告 . . . 51
8.4.2 例外的处理 . . . 51
8.4.3 确认 . . . 52
8.5 安装的软件包的概述 . . . 52
8.6 找到特定软件包的名称 . . . 52
8.7 显示软件包的可用信息 . . . 53
8.8 软件包的安装 . . . 53
8.9 将安装的软件包删除 . . . 54
8.10 安装特定的软件包版本 . . . 55
8.11 使用 dpkg 来安装软件包 . . . 56
8.12 搜索软件包中的文件 . . . 57
8.13 找到文件来自哪个软件包 . . . 58
8.14 从软件包中拆包而不安装软件包 . . . 58
8.15 制作您自己的软件包镜像 . . . 58
8.16 安全登录防火墙（ssh） . . . 59
8.16.1 例外的处理 . . . 60
8.16.2 确认 . . . 60
8.16.3 更新配置数据库 . . . 60

8.17 防火墙 (Coyote) 状态的概况 . . . 60
8.18 下一个 . . . 62
8.18.1 例外的处理 . . . 63
8.18.2 确认 . . . 63
8.18.3 更新配置数据库 . . . 63
8.19 最后 . . . 63
8.19.1 例外的处理 . . . 63
8.19.2 确认 . . . 63
8.19.3 更新配置数据库 . . . 63

9 版权与作者 63

10 附件 A ——运行 Debian Edu / Skolelinux 的合同 63
10.1 运行 DEBIAN EDU / SKOLELINUX 的合同 . . . 63
10.1.1 附件 1 ——定义 . . . 64
10.1.2 附件 2 ——客户的责任 . . . 64
10.1.3 附件 3 ——经销商的责任 . . . 65
10.1.4 附件 4 ——价格和支付条件 . . . 66
10.1.5 附件 5 ——总条款 . . . 66
10.1.6 附件 6 －联系人与地址 . . . 69

1 集中化管理的知识分享

```
<b>b'' 翻 b''b'' 译 b'':</b>
2017 Liang Guo
2018-2020 Ma Yong
2020 Tao Wang
```

小的组织在实际中依赖个人，因而如果有人离开会受到伤害。因此，全面且有质量保证的系统管理手册对保证运行实践稳定和持续是必要的。数字技能程序的首要目标开发一套推荐的运行方案，以向学校和教育机构提供稳定性和可预测性为指导原则，从而计算机、网络和基本服务正常工作。

ITIL 手册包含基于“最佳实践”的指导原则，适应那些使用 Skolelinux 那样的自由软件来运行多所学校的集中网络的自治市。这些指导原则适应自治市和地区管理中心。很多自治市在学校里仅有一个运行 ICT 的兼职职位。在挪威这里有超过 300 座小的和中等规模的自治市；在每个自治市他们有 1-4 人以 ICT 为全职工作。因此，在运行组织之间分享专长和经验是完全必要的。

2 许可证

本文档遵照 GNU 通用公共许可证第 3 版编写。这意味着您有：

- 使用本文档用于任何目的的自由（自由之 0）。
- 学习文档和为需要而改编的自由（自由之 1）。
- 转发拷贝使您可以帮助相邻的人的自由（自由之 2）。
- 改进文档并将含有您的改进的文档公开发布，从而使整个社会受益的自由（自由之 3）。

这些自由在维基百科中解释。卑尔根大学法律系的 Torgeir Kielland 分析了 GNU 许可或非盈利版权的条款。他声明 GNU 许可是版权相关的。简单地说，您可以视情况使用本文档中的每一部分。您必须确保您的贡献也接受一般公共许可。

2.1 致谢

很多人对文档做出了贡献。它本质上由 Knut Yrvin 和 Andreas Johansen 撰写，还有很多贡献来自于 Klaus Ade Jahonstad。Skolelinux Drift 有限公司的 Halvor Dhl 在委员会中，对结构、形式和内容做出不少贡献。此外，UNINETT ABC 的 Snorre Løvås、来自 BzzWare AB 的 Finn-Arne Johansen 和来自 LinuxLabs AB 的 Ragnar Wissløff 也做出了贡献，并且参考组也加入撰写文档。后面的人参加了参考组：

- Monica Larssen ——Harstad 自治市
- Aksel Celasun ——Hurum 自治市
- Trond Mæhlum ——Kongsvinger 自治市
- Bjarne Nielsen ——Nittedal 自治市
- Stein Lier ——Akershus 郡

本文档在维基中维护。这确保运行人员能够容易地寻找问题的解决方案，更新配置，以及诸如此类。

本文档的版权状态请参见 版权页。

http://www.gnu.org/copyleft/gpl.html

2.2 背景

数字能力的计划是 2004 年到 2008 年挪威教育部的 ICT 计划。一个目标是开发一组推荐的运行方案和适当的指导原则。它将为学校和教育机构提供稳定性和可预见性，从而计算机、网络和基本服务能够正常工作。运行方案必须适应机构的大小与需求。

本文档包含的指导原则基于为自治市和郡的 ICT 服务部门定制的实践。它也可以用于商业操作者。很多自治市在学校里只有一个计算机网络运行的兼职职位。整体上，挪威仅有 13% 的自治市有超过 2 万的居民；73% 少于 1 万人。自治市的管理部门通常有 1-4 个人为 ICT 全时工作。对于学校，通常只有一个 ICT 兼职职位，可以覆盖大约 5-10 所学校的 500-800 台客户端计算机，大约有 1 700 -3 200 名学生和教师使用系统。

本文档也适用于较大的组织。它基于 ISO 20000 标准而服务于 ICT 的运行，也被称为 ICT 基础设施图书馆。关于其自身标准的更多信息，请参见维基百科：http://en.wikipedia.org/wiki/ITIL

本文档的第一版完成于 2006 年 7 月 19 日。

本文档在维基上维护，并可以在如下的网址上更新 https://wiki.debian.org/!DebianEdu/Documentation/nb/ITIL。此前版本可从如下的网址上获得 http://developer.skolelinux.no/itil/oldindex.html

本文档从 2015 年 3 月开始使用 https://www.transifex.com/projects/p/itil-revitalization/ 翻译成英语。

3 服务支持

如简介中提到的，推荐从一开始就建立集中化运行的办公室，使您能管理标签。这样做的优点就会迅速显示出来，这对客户和用户的满意度是重要的。

一旦办公室建立并使用标签（用户请求与检修）的合理流程来运行，您将行进到管理上最大的挑战。作为规则，这就是对管理的更改或对问题的解决。使用“牛仔”系统管理员的组织经常开始于管理的更改，“牛仔”系统管理员想出聪明的方法但不做很多测试就将其实施。对于遭受客户反复的愤怒的组织，首先是要解决问题。

无论选择从那里开始，某个数量的配置管理都是必要的。对于向用户交付软件和服务，配置的管理是严格的。软件必须正常工作，为了进行有利的更改，必须知道不同程序的配置。

为了对配置的更改进行管理，必须使用数据库（配置管理数据库 CMDB）。很少人对所有配置都使用数据库，您也不必将所有配置都添加到一个单一的数据库中。将配置放入几个较小的并部分独立的仓库中也很好。例如一些人将配置和设置存储在版本控制系统中。但即使您有不同的仓库，如果连接来自不同过程中的信息也会使您获得更大的好处。

对于 Debian Edu 的用户来说，多数服务配置存在于特定的目录内（/etc）。这可能会因为收集并存储在中心版本控制的目录中而获益。如果重新安装的话，这会更容易地恢复丢失的服务并设置机器。这同样应用于服务器和用户笔记本或工作站。作为 Debian Edu 备份系统的一部分，备份从设置目录 /etc 中做出来。但备份系统不过是配置的数据库或版本控制的目录。

3.1 服务台

服务台（Service Desk）是用户询问问题或报告故障的地方。在学校，ICT 联络人经常将操作报告转发给服务台。同样还有像设立新 PC 或安装程序那样的请求。

在学校，ICT 联络人是到服务台的纽带。ICT 联络人还要回应最普通的问题。一些问题非常难于在每所学校被解决，并且必须转发给服务台。学校 ICT 联络人与服务台的操作人员之间的良好合作非常重要。过于昂贵或过分难于在本地解决的任务应该传递给服务台。

用户也会从服务台的操作人员那里得到直接的回复。所有的运行询问都进入服务台。询问会被指定一个案件编号。任何登记案件的人都会收到电子邮件确认请求已经收到。在案件的考虑过程中，在服务台处理案件的那些人会为用户发送状态更新。

这样，用户得到一个联络的点，并且服务台操作人员看到事情的概况。可以期望在组织的所有部分来操作进行检修。团队领导需要定期彻查所有问题和解决方案，将调试排序，并防止错误再次出现，从而为学校提供稳定的运行环境。

事件可以通过电话、传真、电子邮件或 web 表格来报告。更紧急的事件必须优先。需要迅速解决的事件通常通过电话来报告。较不重要的事件通常通过例如电子邮件来报告。应该为事件指定几名支持人员，并需向用户询问问题来调查研究故障。

- 记得要称为积极的倾听者，而不是消极的。

所有的询问应该计入日志，并且应该发出电子邮件确认。重要的是用户感觉安全，并且应该将那些可能是问题的信息与他们交流。当询问到达服务台时，事件的简要描述应该被记入日志。询问会来自学校的 ICT 联络人，或者来自有协议使用服务台的某些人。事件记入日志应该尽可能快地进行，并应该被指定一个案件编号。用户应该通过电子邮件得到确认，知道事件已经收到并且指定了适当的案件编号。

以前，询问写在纸制的工作日志中。今天，软件用于在“请求跟踪程序”中记录询问。将询问记入日志的操作是至关重要的。基本上这就是错误处理、用户请求和各种事件的排序。日志条目对于防止重复发生错误很重要。因为那些运行事件被定期复核，所以可以评价修复并排序。通过对出问题的服务和根据用户认识到有问题的应用进行调试，日志还提供了改进服务基础。

这样，请求的日志对用户和服务台都是基本和必要的工具。有几种可免费获得的将请求记录日志系统，并且它们带有很好的文档 <<注脚（RT Essentials：http://www.oreilly.com/catalog/rtessentials/chapter/index.html ）>>。Skolelinux Drift 使用 RT <<注脚(RT: 请求跟踪程序：http://www.bestpractical.com/)>> 来处理请求。

开始进行支持时的一件重要的事情是，一开始不要太艰难。不要试图立刻实现每一部分；而通过保持用户被通知，以及目标是迅速的回应时间，从而取得“速胜”。如果不能自己处理问题的话，弄清楚服务台要向谁转发事件，这也是很重要的。支持台还必须检查用户是否被中断了。这会迅速并容易地给出反馈。

对用户来说，重要的是事件被处理。对于服务办公室来说，重要的是事件根据服务层次协议被正确地处理，并且超出所协议的工作请求要在学校管理部门与系统管理组织之间进行处理。

3.1.1 任务和角色

我们推荐，就学校的 ICT 联络人所具有的责任与服务台工作的人应该尽到的责任达成协议。相对于自治市的管理部门和私人公司来说，学校通常几乎没有资源。同时，学校经常有非常多的用户，和比自治市其他部分通常使用的更多的客户端机器。

任务根据角色分配。通过建立清晰的角色，能够容易地分配任务，并且弄清将运行问题解决所必须的工作能力。自治市和专业组织的运行经验显示了后面的角色是常见的。

- 在每所学校中的 ICT 联络人。这通常是使用 ICT 教学和/或具有技术背景的教师。
- 在中心 IT 服务部门工作的操作者。这是具有操作技能的人员。
- ICT 协调人，组织 IT 的教育应用，并对开发、运行和教育应用的计划做出贡献。通常这是名教师。
- ICT 负责人。这通常是对 IT 运行负责的校长。

后面是各种日常工作的概况，其中的一部分由自治市承包出去了。

每所学校里的 ICT 联络人的任务：

- 监督学校的服务器室。
- 作为自治市的学校联络人——报告错误和系统中断。
- 执行简单的维护工作诸如更换鼠标和键盘，升级瘦客户端，和简单地打补丁。
- 作为学校的超级用户——能够在各方面建议同事：用户界面，电子邮件，视频投影仪以及相关应用。
- 参加 ICT 会议。
- 创建和管理本地用户。
- 执行简单的打印机维护。
- 创建和管理电子邮件账户。
- 在 ICT 指导教师的指导下执行简单的命令和操作。

- 在教学中促进 ICT 的应用。

操作者的工作：

- 接收事件报告和服务请求。
- 用电话和电子邮件指导 ICT 联络人。
- 通过安排，去访问学校来检修计算机、打印机和服务器上的缺陷与故障。
- 备份。
- 在学校计算机（服务器和客户端）上更新安全软件。

ICT 协调人的责任：

- 辅助学校管理部门和 ICT 联络人开发 ICT 的技术和教学方案。
- 当选择软件及诸如此类时，为服务台和管理部门提供指导。
- 确保学校有恰当的教学 ICT 工具，并且计算机和网络适于学校的课程。
- 为学校里的 ICT 技术和教学法需求提供运行服务的建议和指导。

ICT 负责人的任务（院长，校长，运行服务的领导）：

- 进行计算机设备的共同采购并加入共同协议等。
- 开发能力计划。
- 为学校提供 ICT 教育应用的课程。
- 运行课程。
- 针对运行合同进行谈判。
- 确保 ICT 联络人和 ICT 服务部门拥有必需的资源。

预先解决谁执行这些任务的优点是，对个人的预期是已知的，为计划和管理 ICT 服务提供了良好的基础。这些 ICT 任务通常只由具有教学任务的人员兼职完成。

两名团队成员为 100 名用户全职运行 100 台标准客户端机器的业务可能会很好。在学校里，总体上可能有 30% 的职位在几个人之间分配，来运行由 320 名学生和教师使用的 100 台客户端计算机。

当学校只有这么少的资源来运行时，将这些资源管理好是至关重要的。在谁处理什么任务上达成协议，能够容易地评价是否需要更多的资源，或者因为预算限制而降低学校对 IT 积极性的期望。学校里了解 ICT 任务的良好概况的 IT 管理者，如果必要能够更好地要求增加资源。这可能是需要增加资源来实施基于 ICT 的考试，或者是需要新的设备像电子白板来辅助教学。

3.1.2 预计用时

我们已经建立了表格来显示运行和维护上花费的时间。表格基于集中运行 Debian Edu 的 9-10 所学校的自治市经验，学校有 250-500 台客户端计算机。有几件事没有包括在表格中。这样，对于一些项目中学校开发自己的网络与更多设备的 ICT 方案，还需要额外的时间。

角色	工作责任	每所学校每周用时	所有学校的总共用时
集中的操作人员	500 台机器的监控，调试和运行，例如，有 3 200 名学生和教师的 10 所学校。	2-3 小时（50 台客户端）	1/2 个职位（500 台客户端）
每所学校的 ICT 联络人	照看设备，方便维护，并报告事件与请求	3-4 小时（50 台客户端）	1 个职位（10 所学校/ 500 台客户端）
中心 ICT 协调人	辅助计划并实施学校的教育性和技术性 ICT 工作。	1-2 小时	1/2 个职位
ICT 经理（校长）	进行联合采购，并确保符合服务层次协议。更新日程，或开发解决方案	1 小时	1/4 个职位
学校整体	**50** 台客户端机器（并发用户）	**6** - **10** 小时	
学校整体	**10** 所学校，**500** 台客户端机器（并发用户）	**2 1/4** 个职位	

经验显示，ICT 联络人的工作范围受到并发用户数量的影响。“并发用户”对很多人来讲是新的术语。用个例子来说明：一所有 250 名学生但不超过 50 台计算机的学校。那么最多有 50 名学生同时使用计算机。这远少于在系统拥有账户的总共 250 名用户。是这 50 名登录的用户为 IT 服务部门提供工作。另外没有登录的 200 人几乎不提供额外的工作。

这样，通常使用最大数量的并发用户来计算 IT 成本。其他计算方法也是可能的，专有软件的支付是个例子。但由于 Debian Edu 没有许可成本，那么并发用户的数量就是运行成本最至关重要的因素。使用用户账户来计算成本对学校只有很少或甚至没有意义。

对于 Debian Edu 的用户，管理 100 或 250 个用户账户的成本差异非常小。但有一些例外。有 250 名而不是 100 名学生，一些学生会不断忘记他们的密码。这样，明智的是授权教师来负责班级，为这些学生提供密码。

相比具有 150 台客户端的学校，如果学校有 50 台客户端机器，那么 ICT 联络人在其运行工作上需要的时间更少。客户端更多，花费在运行上的整体时间增加，但每台客户端机器上的运行时间会降低一点。

有几个自治市为具有 30-70 台客户端机器的学校的 ICT 联络人工作每周留出了 3-4 个小时。奥斯陆的教育局每周留出了一天半，或 30% 的职位来支持 150 台客户端机器。其他自治市的经验建议，当学校有 160 台运行 Debian Edu 的瘦客户端或无盘客户端时，20% 的职位用于本地 ICT 联络人的任务是足够的。

此外，还有与集中运行，ICT 管理，以及在学校课程中建立 ICT 工具的教育应用相联系的成本。一个职位可能足以运行 1000 台客户端机器。关于教育支持，在学校里几名校长的 50-100% 的职位用于这个工作。可能有 10-20% 的职位用作 ICT 联络人，而 40-80% 的职位用作教师的教育支持。很多教师将学校里的 IT 工具看作新事物。一些校长希望通过使教师在不同课程中更自信地使用 IT 工具，而更多地给出教育的另一面。

3.1.3 清单

现在我们给出关于设立并运行新的服务办公室所需要什么的清单。

- 将所有学校的人员安排为不同的角色，如 IT 经理、每所学校的 IT 联络人、所有学校的中心运行部门和 IT 协调人。从教学工作中将技术运行和维护工作区分出来是很重要的。
- 建立服务台，使每所学校具有服务协议，来规定什么是标准运行行为，什么是额外的。负责 ICT 的校长是这个过程的一部分，这是迫切的。
- 建立系统将到来的请求进行处理（请求跟踪程序）。通过电子邮件发送的所有询问都需要案件编号。来自学校用户或 IT 联络人的几乎所有询问也都需要案件编号。
- 为了确保学校的计算机设备和网络正常运行，要确保 ICT 预算能够反应出所必须的贡献。今天的要求是，通过使用联网或不联网的 ICT 工具，ICT 系统将会用于全国性和本地性的测试。
- 基本上在所有学校都使用相同版本的标准版 Debian Edu。从这里可以做出您想要的更改。这些更改必须在配置数据库中管理，并且要有与所做的更改相关的文档。版本管理系统可以用于存储更改和文档。

3.2 事件管理

ICT 服务部门的目的是防止干扰，比如像停机或软件问题。如果 ICT 服务部门有用充足的资源来处理运行、设备和发给服务台的询问，那么用户几乎不会经历到有关 ICT 系统的问题。或大或小的问题会中断用户工作，因此对事件的良好处理是必要的。

在跳伞时的准事故被称为“事件（indicents）”。计算机运行中某些部分不工作并不是非常相同的情况。处理事件的目的是尽快恢复服务，从而每一部分都正常运行。如果一些事情出问题，必须使冲击到客户可能性降到最低。在描述服务层次的运行协议中来达成了“正常服务”的内容。

事件的统计很重要，特别是如果组织内有几个人工作的情况。当几个人共同工作时，很容易丢失对工作的跟踪。统计会指出问题的所在，相对于服务台的快速修理，必须更全面地处理。例如，可能有很多请求要更换忘记的密码，所以让教师为班级的学生更改密码会是个明智的方式。

运行干扰定义为：

- 一个事件，不是正常运行的一部分，并导致或能够导致服务质量的中断或降低。

运行干扰的例子会有：

- 程序
  - 办公程序（OpenOffice.org）不启动
  - 浏览器（Firefox）崩溃
  - 硬盘被写满
- 硬件
  - 服务器宕机
  - 不能打印
  - 不能登录
- 请求
  - 请求信息、建议或文档
  - 忘记密码

例子显示了一些最常见的运行问题。这些问题促使用户联络学校或服务台。ICT 服务部门必须排序，哪些问题必须以直接方式处理，哪些问题需要更多的时间解决。为了优先处理哪些问题需要更详细地调试，重要的是将与故障有关的所有询问记入日志。一旦看到了最常见问题的概况，就可以采取适当的行动。

3.2.1 清单

我们已经做出了一个简短的清单，以便使良好地处理事件的步骤和系统被确保就位。

- 进行调试的操作者将状态返回给学校的 ICT 联络人和/或用户。
- 对学校和工作台的那些处理事件的人来说，将事件记入日志的系统必须可用并正常运行（既是技术上也是功能上）。
- 事件日志系统必须用于几乎所有的运行事件。
- 应该定期做出事件日志的统计。统计可以用于识别并消除那些重复出现而烦扰用户的问题。

3.2.2 计划与实施

如果为事件日志设立可工作的系统，需要比安装系统更多的东西。运行部门的每个人必须使用系统。报告错误的人必须通过带有标签编号的电子邮件来接收反馈。这需要显著的努力来配置事件日志系统。此外，对接收请求的那些人必须确保进行基本的用户培训。

实施适当的事件处理，并不需要大而详细的计划。对于服务台工作的人或学校的 ICT 联络人来说，事件处理完全是标准任务。设立计算机工具将事件计入日志，会需要多达几周的时间来进行正确的配置，用户也可以通过电子邮件或电话来报告事件。

日志系统的用户界面相对清晰易于理解，因此不会花费很多时间来开始这项工作。系统的日常使用将使用户熟悉适应记入日志的内容。运行部门的每个人使用日志系统来记录运行消息是至关重要的。

3.2.3 与运行事件相关的行为

为了对报告事件后完成的行为有个概念，我们举个例子。

用户带着问题联络服务办公室，并且报告不能打印。电话挂断后运行部门立刻将事件记入日志。为这个问题打开一个案件，并自动给予案件编号。

服务台的运行部门迅速分析。是后台打印程序又停止了，还是其它什么事？是没有纸或墨粉了？操作者检查后台打印程序，并看到队列排满了。将队列删除并测试是否打印了下一个工作。

这时打印队列再次充满。操作部门联络学校的 ICT 联络人，要求检查打印机的纸盒是否空了。这在事件日志中列出。ICT 联络人回复他们重新装填了纸盒，并且打印正常了。案件关闭，并在系统事件日志中记录。

如果打印没有再次启动，可能是墨粉用完了，或存在其他打印机错误。如果有错误，操作部门会使问题升级。这意味着操作者或 ICT 联络人以外的人员需要解决问题——在这个例子中，是一位修理打印机的技术人员。

这个例子显示了整个工作流程，需要调查研究来使打印机再次工作。如果检查到即使不缺纸和墨粉时打印机也不工作，那么问题问题需要升级。运行部门必须请专家来修复问题——这次是打印机的服务技术人员。

出了什么问题和如何修理记录在事件日志系统中。

3.2.4 角色

当 ICT 服务部门处理报告的问题时，各种角色都会参与进来。在上面的例子中，学校的 ICT 联络人和操作者合作解决打印问题。如果问题更难，他们会请来技术人员。如果打印机不能修好，就需要购买新的打印机。如果学校需要购买新打印机，ICT 经理会需要安排支付。在很多组织中，校长最后拍板。

简单地说，当有些部分不工作时，使更多的人参与进来是很方便的。如果可能，问题会当场解决，而尝试避免包括不必要的人员。将本地能够解决的问题升级会很快变得成本高昂。很多询问是很容易在那里及时解决的，而其它请求会包含更复杂的问题，包括更多的人。如果需要另外的或外部的帮助来解决问题，作为规则，必须要由运行经理来澄清。重要的事情是了解这些处理运行事件的要点，从而适当地使用资源。

3.2.5 关键点

我们设置了一些处理事件的关键点。通过使用可衡量并规定好的要求，来评估事情是否能够正常进行，这些关键点是有帮助的。这些措施的要点是：

- 操作事件的总数。
- 从接收请求到事情解决的平均时间，以编码来分类（组织良好的操作部门为不同类型的事件和错误设置了编码）。
- 在协议的回应时间（如服务层次协议中所规定的）内处理好事件的百分比。
- 每个事件的平均成本
- 通过服务台来解决事件而无需升级的百分比
- 每台客户端机器的事件（工作场所）
- 无需访问学校而通过运行中心来解决事件的数量和百分比

3.2.6 工具

有几个工具可以更容易地处理运行事件。

- 自动日志
- 自动将事件导向正确的人员
- 从配置管理数据库自动取回数据
- 电话和电子邮件与登记请求和事件的工具一起联合使用。

3.3 问题管理

问题管理是一个“调查研究”过程。已知的错误最通常地由服务台直接处理。这是事件处理最通常的形式。调查研究未知的错误需要感觉和本能。好的操作人员使用本能直接找到问题，尽快找到解决方案并恢复服务，从而使每一部分工作正常。

问题管理是

- 问题管理
- 检查错误
- 积极主动的控制，防止问题出现
- 使用信息表格，例如事件管理，来识别错误的模式

问题控制

- 识别问题
- 将问题分类
- 检查/研究问题

错误控制

- 识别并登记已知错误
- 如果可能的话找到临时解决方案
- 联络负责更改管理的人，来永久地删除错误

积极主动的控制

- 在用户报告事件之前识别并解决问题和错误。
- 使用日志和来自事件处理的信息，看看问题会如何出现

3.3.1 问题管理的步骤

SkoleLinux /Debian Edu 手册是解决问题与配置系统的详细方案的集合。每一部分都在 Debian 维基页面上。在学校人员、自治市的 ICT 服务部门、专业人员和志愿者帮助下维护解决方案。请参见英文页面的链接：https://wiki.debian.org/!DebianEdu/Documentation/Manuals。这些页面被翻译为挪威书面语。我们也在工作将页面链接到书面语。

为了在互联网上维护编目的信息，维基技术已经证明是巨大的成功。它易于贡献，并且所有更改被记入日志。还能导入 OpenOffice.org 文档，并将文档导出为 PDF 文件。

3.4 配置管理

在学校里，花费在 IT 系统上的资源必须在财务上谨慎处理，从而控制使用的服务和设备/基础设施。设备、软件和服务具有整个范围的设置——这就是配置，或如何设立基础设施和服务的逻辑模型。

为了管理配置，必须将其识别、存储并维护。还要能够跟踪不同版本的配置。我们将设备的每一部分称为配置项（CI）。配置文件会例如，确保某些用户访问网络上的一些打印机。另一些会确保您在无盘客户端上得到一些缓存。

更新的用于配置管理的数据库，本质上确保快速且可控地处理运行事件，或更改机器的安排、程序或服务。

3.4.1 计划

进行计划来设立用于配置管理的数据库。对于配置的存储与维护，必须决定在哪个领域使用系统、目标、政策和过程。

- 根据 ICT 基础设备的重要部件来识别并选择配置的架构。配置所有者、名称标签（属性）、依赖性、和配置之间的关系都需要考虑。
- 在系统的整个生命周期内，在数据库中只管理允许的配置。对访问配置进行控制可以通过用户组权限来进行，并且可以通过更改管理过程来进行。
- 状态日志——跟踪各种子系统的条件和状态。这应用在服务、软件或硬件的整个生命周期中。在产生、断开或中断时会有配置。
- 检查与修订。每个配置必须被检查，来确定在配置数据库（CMDB）中存储了正确的信息。后面是定期复核，来确保数据库是最新的。

如我们所看到的，需要很多计划使 IT 系统中具有配置管理。计划的目的作为 IT 运行的一部分，在系统宕机时确保能被迅速修理。好的配置管理能够容易地使用新的机器替换出故障的。配置可以迅速转移到新的计算机上，并且 IT 系统能够像以前一样运行良好。

3.4.2 配置项目（**CI**）的管理

配置项目是基础设施的一部分，通常是服务或程序的配置。有时用户想要更改服务如何工作。如果做出了更改，就需要对配置进行跟踪。

为了理解现实，我们可以想象打印服务器的配置。您想要将新的打印机添加到计算机网络，并将它添加到打印系统 CUPS 中。当通过 Web 应用或通过 KDE 中的设置来更改配置时，CUPS 配置文件将被更改，并且必须再次重新启动打印机服务器。这可以在 KDE 工具中或通过 Web 应用来完成。修改的设置文件复制到目录中，文件可以在那里被版本系统处理。

在很多不同的选择中会有一些共同的。也就是服务应该：运行、停止、终止、启动、中断或清除。

应该小心没有合适的计划而更改配置。您很容易忘记已经在服务器或 PC 上做了什么。这样，重要的是将做出的更改记录到更改日志中。

3.4.3 计划与安装

计算机网络的配置与架构相联系。很多计划用 Debian Edu 做出。因为用 Windows server、RedHat 或 GNU/Linux 发布版本将服务器设立到相应的服务层次会花费 3-4 周。Debian Edu 花费 1-2 小时做同样的事。如果您想要网络有个固定的 IP 地址，专家还要在上面额外花费 1/2 个小时。这是因为 Web 服务使用了可再次使用的名称来设立。

然后必须计划的是要使用哪个另外的用户程序，以及用哪个子系统与 Debian Edu 互联。例如学校可能会有电子白板。

3.4.4 清单

我们做了行为与解决方案的列表，这些在良好的配置管理中是很重要的。

- 建立版本控制的区域，为所有的服务器和所选择的工作站与笔记本存储配置。这通常使用 Git 和 SVN。记住将区域每天备份，并且确保在配置中存储所有的更改。
- 使用电子系统来照料那些解释不同型号机器、网络或服务的配置的诀窍。对于其他帮助或接管操作的人，这样的诀窍会用于使其掌握做了什么操作。维基可以适用于这里。
- 在所有机器上使用特定版本的操作系统和软件。这用于避免去维护不同版本的软件。确保软件经过良好测试。这样，在采用最新版的程序前等待 6-12 个月会是明智的做法。

3.4.5 与其它过程的关系

如果系统可用的话，配置管理与处理问题紧密相联系。如果打印经常停止，那么可能更改设置会解决问题。例如可以建立常规来删除打印队列并再次重新启动打印服务。

在配置中做出更改的目的，通常会增加服务或程序的可用性。也可以将某些程序或服务的访问限制在特定次数内。为了实现这个目的，必须配置服务。此外，花费的金钱会超出服务层次或系统容量上达成的数额。

例子显示，对配置的管理用在了很多其他领域。这样，落实在配置中管理更改的良好实践会有很大的收益。如果您想要更大的稳定性，或者在特定的时期访问某些服务的话，同样建议要应用自动化。

3.4.6 配置管理的工具

如清单中阐述的，可以使用

- 将配置文件存储在版本控制系统中，例如 subversion。
- 用于存储设置的文档和向导的维基
- 使用互联网上操作文档的共同目录，它由学校的 Skolelinux/Debian Edu 人员来维护。

3.5 更改管理

在 ICT 系统中处理更改方面，很多 ICT 服务部门不够聪明，导致很多用户不高兴。丹麦公共部门的调查显示，当对更改具有良好的控制时，运行成本会下降。这样，将受过培训并参加更改的用户包括进来会是有益的。

更改消息完全依赖于正常的过程。无论更改是大是小都要应用。这样，重要的是进行更改时使正确的人就位，既提供培训也回答问题。当采用新发布版本的软件和服务时这变得特别重要。这与使用免费软件还是专有软件无关。

更改管理应该确保以标准化且正确的方式进行所有的更改。重要的是将的决策锚定在组织中适当层次上，当进行多次标准的更改时可以预先同意。但主要的更改经常包括学校管理部门与操作者之间较高的决策层次。

要包括管理的原因是，升级通常需要培训用户。会是升级到新的浏览器或新版本的办公软件。这可以迅速导致用半天来培训程序中有什么新内容。这样的更改必须由管理部门同意。更改还必须进行而系统的其他部分不停止工作。

负责同意更改的那些人接收所谓的更改消息或 RFC （更改请求）。当接到 RFC 时可以评估是否要进行更改。多数时间都需要您与管理部门一起弄清是否进行可选的更改，并且如果进行的话什么时间开始。

通过更改，人们必须与学校的 ICT 负责人合作。必须确保当更改与学校的计划匹配时发生。没有更改管理而实施显著的更改，会导致巨大的不满和对服务台更多的询问。这会产生计划外的显著的额外工作。此外，还会导致了更改被迅速回滚。您会很快承担两倍的工作，除了返回到开始而不会在任何地方停下来。有人做出必要的许可以后，更改会以计划的且直接的方式进行。

进行更改管理能避免更多不必要的额外工作。进行更改明显需要做更多的工作，但您会承担计划的更改以外更少的额外工作。还可以避免将更改进行回滚的需求，因为在用户没为实质的更改做好准备的情况下，问题会出现。

例如当您将整个系统更新到新版时，确保通知到每个人。必须要看看更改所影响的那些人是否需要培训。正确的专家必须将一切准备好，从而不会发生意外。

所有的责任必须由负责管理软件版本的人承担，就是版本发布经理。版本发布处理是进行更改的过程，这些更改最好包含很多小的更改。这通常发生在新系统和服务展开，或将整个系统更新到新版本的时候。

3.5.1 行为

- 看到更改消息，或上面说的 RFC（更改请求），并且检查它得到一个唯一的编码。
- 将更改排序并分类
- 删除不可能的更改。通过将它们标记为不可能来进行。
- 向发出更改消息的人反馈
- 确保您有个更改建议委员会，使更改在那里处理、讨论并评估。这个咨询组可以是选择的 ICT 联络人和具有长期经验的操作人员。
- 将更改与版本发布管理相协调，版本发布管理处理不同版本的应用和服务。
- 阅览并完成更改消息（RFC）
- 要记得将修改过的配置存储到配置文件的仓库中。
- 报告

即使那些看起来小的不显著的更改消息，在将更改实施的时候也能具有重大的结果。我们有个学校的例子，他们有稳定的 Debian Edu 网络，所有程序都能工作。一个不断崩溃的测试版本的流行程序被安装了上去，然后 Debian Edu 就遭到指责了。

一个例子是学校安装了程序最终完成前测试版的最新 OpenOffice.org 。一些人认为有趣并去试验。但问题是测试版通常是发布出来用于找到错误与不稳定性的。不是用于生产使用的

在生产上，一般的规则是不要安装测试版的软件。多数操作者推荐使用仅低于最新版的程序用于生产。在 6-12 个月后，通常最糟糕的错误会从新版本的应用中被找出来。

这意味着要更新到新年前发布的程序之前，通常要等到夏天。这个时间与学校的学年匹配得很好。否则会出现不稳定性和不满意的用户。这样，当进行较小或较大更改时，咨询组扮演了关键角色。

3.6 版本发布管理

版本发布处理是为想要的更改而准备的管理和计划行为。更改可大可小，大的更改会包含很多较小的更改。在安装软件和硬件进行生产的实际工作启动前，版本发布管理就在进行了。

首先要执行新发布版本的计划与测试。然后将它展开投入生产。部署是基础设施管理的一部分。步骤是实施那些已经计划、测试并在配置管理系统中准备好的。一旦计划并测试了每一部分，并且存储了配置，就可以在生产中展开解决方案了。

通常会包括很多服务提供者和供应商。他们都用于机器的获取、软件的使用和推荐的配置上。以良好的方式为用户打包并发布新的版本，好的资源计划是至关重要的。在这里图省事会导致设备不工作，或者因为安装不充分而不可用。

版本发布管理通过服务的更改而采取了详细的方法，并确保所有部分的出版物在文本中可以看到。这应用于技术和非技术的因素。

3.6.1 基本

如您能看到的，对于计算机、软件和网络能像计划的那样工作，版本发布管理是至关重要的。版本发布的适当处理会防止混乱。在运行正常继续时，可以引入新的版本发布或更改，而不会中断或降低质量。

实施更改或新的版本发布可以与修建一条新的道路相比。即使您在旧的道路上面修建新的道路，汽车也必须仍然能够通行。好的标志必须放置到位。还要必须拥有必要的资源来重建道路。如果您缺乏资源，那么最好不要进行更改。

如果每次新版本发布时却并不实施这些最新的版本，那么有些人会认为适当的版本发布管理是无聊的。但是，运行部门通常缺乏资源，来处理升级失败时如洪水般袭来的抱怨。长时间的稳定运行需要已经被确认的技术，如 Linux 专家 David Elboth 在《Linux 杂志》(1/2004) 中说的。他写道：

- 你对系统要求的越多，每个部件的要求就越严格。对运行时间的高要求还显示了留给你的选择只有旧的技术。只有相对于时间上的经验数据才可以找出宕机时间的任何问题。我们都注意到我们落在 Red Hat 和 SuSE 的服务器产品后面有多远。

为了得到极少的抱怨，以稳定和可靠的环境，需要坚实的版本发布管理。否则由于安装了没有充分测试的最前沿软件，会导致出现大量的抱怨和不满意的用户。业余爱好者倾向于低估软件升级导致的后果。即使有些部门在您家里的计算机上运行良好，也不意味着它能在具有 500 台客户端计算机和 3 200 名用户的广大网络上工作。

3.6.2 可靠软件库（DSL）

在运行文本中，软件档案是使用中的软件的一组原始副本。如果您使用 Skolelinux 2.0，那么这是个软件包。“软件档案”这个短语在不同文本中的用法不同，特别是在程序员当中。当谈论运行时，我们谈论的是用于安装的特定版本的原始软件包。

使用自由软件时，软件档案会是 Skolelinux 2.0 加上您从各种来源添加的额外的程序。这会是某个版本的 Macromedia Flash、Java 和解码器，能够在浏览器中运行国家测试，或者观看国家电视台的节目。

如果您计划在发布时升级到下一个版本的 Debian Edu，这个新版本会是主要的程序档案。新的档案也应该包括超越 Debian Edu 的适当版本的所有另外的应用。

运行部门定制或在本地新建的设置文件，不包括在主程序档案中。配置另外存储在版本控制系统或数据库中。

3.6.3 配置与硬件的数据库

如在配置管理章节中提到的，您必须新建数据库或版本控制目录，来照料设置文件。应该跟踪所有计算机，使用了那种机器，性能，网卡上唯一的标准地址（MAC 地址）。

需要掌握设备概况的原因很多。一个主要原因是跟踪有多少机器在运行，多少没有使用，以及多少在修理。另一个原因是用于升级的计划。

3.6.4 构建管理

除了浏览器和办公套件以外，学校里安装了大量应用。用于学习的教育程序、浏览器插件和用于多媒体的程序都是需要的。系统还有网络设置和特定程序的更改的设置。当您有很多服务器和可能上千台客户端时，很快就会感觉到需要部署有效的工具。这样的工具在 Debian Edu 中是标准的。

构建管理是关于确保您总是安装所需的软件包、服务以及单独程序与网络的适当设置。很多人听说过所谓的“映像”。人们安装操作系统，并带有所有需要的程序，并配置网络。然后使用映像程序做出硬盘的复本。然后这个“磁盘映像”可以复制到其他计算机上。

没有必要建立这样的磁盘映像。Debian Edu 基于 Debian，它有优秀的软件包管理系统。不需要将应用编译，因为准备好的软件包可以直接从互联网上安装。足以使您对默认设置的 Debian Edu 和使用的主程序档案做出想要的更改。然后您可以做出一个或更多脚本，在每台机器上运行，将每一部分都被安装并设立。

对于多数情况，为了“构建”并展开程序与配置，脚本是容易的方式。但有些情况下建立磁盘映像会是解决方案，例如安装到很多笔记本上面。

如所说明的，处理构建过程是在很多计算机上实现部署。在例外的情况下，这会包括建立剪裁的 Debian 软件包。但在多数情况下，每一部分都准备好打包。然后您必须撰写并部署脚本来安装另外的软件和某些设置。在有很多相似的机器时，例如所有学生的笔记本，还可以建立磁盘映像。

3.6.5 测试

在投入生产前，试新的应用、配置和新的服务是必要的。在安装软件但没有进行必要的调整时，有几所学校体验到了不稳定性。这样，在所有机器上进行更改前，测试配置中的更改或测试新版本的软件是至关重要的。

测试通常分三个步骤进行。

- 首先，在测试的网络上安装更改。这是技术上的测试，来检查没有用户时系统中的每一部分都正常工作。要小心，将所有的更改都包含在配置文件中。
- 当在技术方面确定每一部分都工作时，尝试在一所学校里安装这个解决方案。重要的是要与学校的 ICT 联络人预约进行测试。用户也必须充分了解用于测试而进行的更改。要小心将当前的调整保留在设备文件中，设置文件在正常维护的过程制作出来。
- 当确定每一部分能都工作时，您可以将解决方案展开到所有学校。最容易的是新建脚本，使用脚本来简化软件包、服务和配置的升级。

3.6.6 后退方案

在新的安装或升级过程中会有很多出现错误。这样，必须准备好后退方案。这可以迅速回到升级前的系统。技术术语中称之为回滚。

回滚时，将之前版本的软件存档并准备好配置文件是绝对必要的。这意味着可以例如在一小时内安装好 Edu 1.0，并将适当的配置文件放置到位。

但是回滚会花费时间。这样，使一台服务器准备好之前版本的软件、正确的配置和用户家目录的最新副本，这样会是谨慎的。这台服务器可以迅速代替根据计划升级后不工作的任何机器。即使在问题时，保留服务器也可以确保高的可用性。

3.6.7 优点和可能的问题

保留在生产中的软件复本的优点不能被低估。很多人依赖将软件保留在各自的 CD 或 DVD 上。这不是有效的发布方法。为了节省时间并避免麻烦，所有的 Debian Edu 软件可以在线获得。

您的运行部门可以在中心服务器上建立 Debian Edu 档案的副本。所有的软件都可以从这里快速平滑地安装到其他机器上。优点是您的 ICT 服务部门可以持续地了解学校可用软件版本的概况。这还避免了去安装那些没有经过更改管理部门复核的软件。

如果不保留软件档案和配置的话，可能会有相当大的问题。使配置或软件包出错的情况也可能会发生。然后，这会展开到所有机器上。此外，一些学校会在生产中安装没有充分测试的软件，或使用测试版的发布。因此必须有好的过程，并且有人负责维护程序档案和配置。

看起来好像需要很多额外的事情就位，才能安装并维护正在使用的服务和程序。然而，如果跳过那些提供升级管理的工具，您会承担很多额外的工作。ICT 服务部门必须花费很多时间来手动安装每一台机器。出错误的风险上升了。当事情不起作用时，客户会不高兴，并且更多时间要花在修复问题上。

很多管理大型 IT 系统的人对即将到来的更改没有充分的计划。一些人根本就没有计划，只是将软件升级到最新的发布版本。做出的更改会被一些用户察觉到有问题，因为他们熟悉的功能在用户界面上的位置改变了。对于运行来说会是完全错误的。例如在 Arendal 自治市，当他们试图从较旧版本的 Windows 升级为较新版本时，几乎每一部分都停止工作了。IT 部门说他们有几个计算机程序好像被“电缆和胶带”绑在一起。花费了半年时间才理清楚。

3.6.8 计划与实施

在实施更改前制定计划的原因，是避免因为问题导致的成周或成月的延误。用于计划的时间很快会有收益，因为避免了另外的问题。总是有人说系统中临时更改没有问题；但仔细的检查揭示出这样的更改后总是有问题，他们只是没有去沟通了解。

在我们眼里，临时解决方案只是通过更改经由曲线达到目的，并且只是应急措施。临时解决方案就像是使用“电缆和胶带”的临时修理。必须在适当的时候清理这样的解决方案，来确保稳定的运行而不会有持续的意外。跳过计划阶段会导致很多更多的临时解决方案，以及进行更改或升级时的一些运行问题。这样，专家和管理部门有必要理解进行更改时计划良好的过程的价值。

这样，我们推荐为计划来组织会议，为系统的更改做出阶段性计划。阶段性计划会根据更改而自然地变化。OpenOffice.org suite 套件的升级与整个系统的升级非常不同。当升级到新的办公应用时，每所学校的教师花费 2-3 小时浏览办公套件可能就足够了。当更新整个系统时，必须为用户提供培训，并测试出技术细节能够正常工作。

主要的一点是计划并实施是直接的。研究显示，适当计划并确保人们具有正确的技能，会使运行具有较低的操作成本。

3.6.9 行为

计划新的版本发布是至关重要的。应该由管理部门澄清系统的多数修改。后面的行为列表是设计来支持在计划和实施阶段的升级的。

任务	细节
版本发布的排序：	在更改或升级被布设前，检查是否做出了必要的决定。
可靠的软件库	确保那些需要安装的适当软件包在可靠软件库中就位。
配置数据库	确保所有配置文件就位。这既应用于正在使用的软件，也应用于系统中提供的被更改或升级的那些新的软件。
构建管理	用于部署或建立磁盘映像的所有脚本和系统必须就位。
测试	首先，在测试设备上运行试用版。当它工作而没有任何问题时，可以在学校测试。必须将这件事完全通知给学校，使他们能够充分了解新软件的测试。当确定了每一部分都工作时，就可以为所有其它设备升级了。
后退方案	即使进行了大量测试，新的发布版本也会出错。这样，有必要回滚。最简单的解决方案将旧的安装及其数据作为后备，在另外的服务器上准备好。如果更改或升级不工作，那么这样的机器可以再次插入。

3.6.10 工具

从行为列表中可以看到，需要几个工具来跟踪系统中不同发布版本的软件、服务和硬件。这些工具中的一些在之前提到过。但我们在这里重复一下：

- 可靠软件库的 Debian 工具
- 配置和硬件资产数据库（Subversion 设置文件和电子表格提供了所有所有硬件的概况及其各自的物理位置）
- 构建管理（构建 Debian 软件包的系统）
- 用于测试和备份方案的硬件

3.6.11 与其它过程的关系

版本发布管理直接融入 ICT 服务的内核。它是关于实施适当的安全更新、服务的更改或计算机软件的升级的。新发布版本的请求会是由于运行问题，或希望得到新软件。在投入新的发布前，对更改是否有必要做出评价是必要的。

如果更改是直接的，那么应该在配置中进行必要的更改，并准备好应用软件包用于部署。这将被测试，并且应该使备份方案就位。当进行更改时，可能会改变部分运行常规。很容易看到，更改管理会影响到运行支持的所有部分。

3.7 用于运行支持的工具

您应该问自己的第一件事是：“我们真的需要软件工具吗？”如果答案是肯定的，全面检查选项是至关重要的。

携带术语手册，并倾听销售演讲，这就能够完全依赖于这样的工具了。但良好的人员、良好的过程描述、良好的步骤与工作描述是良好的服务管理的基础。需求以及工具的复杂性，依赖于组织对计算机系统的需求以及组织的大小。

在小的组织中，一个单一的自由访问的数据库就足以用于将事件登入日志并管理了（请求跟踪程序）。但在较大的组织中，几乎必然会需要尖端的分布式和集成式的工具，来用于服务管理。这意味着这些过程被连接到事件处理的系统上。

尽管工具会是很重要的，但它们本身并不重要。要执行任务和过程，以及所需要的信息才是重要的。它们会提供必须的信息，来指定哪个工具最适于支持运行。为什么使用软件进行运行和服务管理的一些原因在这里：

- 来自用户的需求增加
- ICT 知识的缺乏

- 预算的限制
- 组织完全依赖于服务的质量
- 来自多个经销商的系统的集成
- ICT 基础设施复杂性的增加
- 国际标准的出现
- ICT 的领域延伸与更改

自动工具允许：

- 关键功能的集中
- 服务交付中功能的自动化
- 数据的分析
- 趋势的识别
- 可能会实施的预防措施

3.7.1 工具的类型

在这一章，我们建议了几种工具来改善运行支持。后面是工具的概括：

- 可靠软件库的 Debian 工具
- 配置和硬件资产数据库（Subversion 设置文件和电子表格提供了所有所有硬件的概况及其各自的物理位置）
- 构建管理（构建 Debian 软件包的系统）
- 用于测试和备份方案的硬件
- 请求跟踪程序
- 用于监控的系统（Munin）

当运行部门获得系统性常规的进一步经验时，可以开发或获得另外的不同类型工具。

3.7.2 选择工具时的评价标准

尽管使用了大量的金钱来建立软件的评价标准，结果仍然只是基于经验的指导原则。对于什么是好的或不那么好的软件，没有最终的答案。更多的是，它部分是围绕品味的。不同的解决方案同样可以做相同的工作，但可能具有非常不同的形式。然而，这里的一些好的规则可能会有用。

主要评价标准是，是否根本需要执行工作。很多 IT 工具绝对优秀且能无错误地工作，但它们可能是解决没必要修复的任务。所以主要的标准是，它是否解决了正确的问题，并且是否根本上需要它做一些事情。

- 所以，要问的第一件事是，工具是否被需要。

如果结果是需要完成任务，那么解决方案会是尽可能简单地手动运行一些命令。最简单的方式才是最好的方式。但当有很多机器需要操作时，自动化是至关重要的。登录 20 台相似的服务器进行安全升级是太多的工作了。那么自动化就是解决方案。

- 所以这里必须要问工具是否对完成任务有用
- 然后，必须要问工具是否可用。

通常有大范围的程序和步骤来解决特定的任务。但当维护 500 台计算机和 11 台服务器，而不是修理您自家的 PC 时，一些问题的解决会完全不同。一个例子恐怕是允许教师在他或她自己的客户端机器上看到每个学生桌面的工具。教师可以为所有学生停止或启动程序，并防止单个学生使用干扰学校工作的例如 IMs。

操作工具的选择，是关于运行任务的自动化和简化的。是关于将手动工作减小到最小的。因此动机只是保持自动化。同样，这里就是能够使事情容易，这会是需要完成的相当量的工作。

如您所看到的，为大型安装选择操作软件而设立好的标准并不容易。这主要是因为软件开发者通常缺乏 IT 系统运行的经验。他们只知道创建新的事情，但为运行创建好的且相关的工具需要很多年的经验。

过去 20 年，一些通用的操作工具还没有被替换掉了。但使用的产品已经被替换掉了。同样一些软件在几年里可能会没有用。这样，在新版本的应用软件用于运行时，在终端用户软件升级更改时，必须准备好得到一些相关的培训。

3.7.3 产品培训

全面的用户培训能够使大量支持问题，通过终端用户之间的直接交流来非正式的解决掉。通常，培训会花费总运行成本的 1% 那么少。很值得在培训上多花费一些。效果非常积极。这同样应用于对学校的 ICT 联络人和操作者的适当培训。培训 ICT 联络人来使用简单的系统进行密码更改，发出错误消息，等等……会对 IT 服务部门提供更高质量的呼叫。

根据挪威的劳动法案 (§ 4-2)，规定了教育与产品培训

- 在计划和实施阶段，雇员及工会代表被通知系统的使用。应该给予他们必要的培训，使他们与这些系统熟悉，并且他们会参加系统的设计。

简短地说，增加培训是有利的，它提高了 ICT 服务质量，并导致显著的成本降低。这是因为用户和 IT 联络人变得更有信心且更好地彼此帮助。应该注意，转向新软件也提供了将一些运行实践简化的机会。简化可以降低对产品培训的需求。

3.8 在实施服务支持的一开始就进行计划

越来越多的组织看到服务控制的必要性。一般的实践通常是基于历史和政治考虑来决策，而不是当前组织的需要。这样，重要的是确保管理部门参加并理解组织中的工作方法，并且彻查现有的过程，并将其与组织的需要和“最佳实践”相比较。

3.8.1 实施服务支持

健康检查

3.8.2 可行性研究

3.8.3 确定当前的情况

健康检查

3.8.4 项目计划的一般指导原则

项目的业务案例

关键的成功因素与可能的问题

项目成本

组织

产品

计划

沟通计划

3.8.5 项目复核与报告

过程

项目的评价

补充工作。

复核来检查与质量参数的兼容性

关键因素的复核

4 服务的交付

服务交付的主要目的是确保积极主动的运行，并确保 ICT 服务部门为用户交付适当的支持。服务交付的目的聚焦在组织的需要上。它是学校需要的，在不同课程中主使用 ICT 工具的主动学习。这章以后面的次序来讲述：

- 服务层次管理
- 经济管理
- 容量管理
- 容量计划
- 访问控制
- 运行连续性

4.1 服务层次管理

服务层次管理通常简称为缩写 SLA。对服务层次的管理是关于运行服务的质量的，联系合同中达成协议的内容来衡量。包括可用性、回应时间、支持、错误纠正等等这些绝对坚实的数字。

目标是对服务层次加以控制，并提高运行服务的质量。通过重复几轮来确定、监控并报告质量水平。目的是改进 ICT 管理者与用户的联系，交付达到协议质量 ICT 服务。

重要的是理解不同类型的 SLAs。可以从很多类型的协议中选择。三种最常用的类型是：

- 对所有客户根据每种服务的协议
- 对所有服务根据每位客户的协议
- 根据每位客户根据每种服务的协议

所有的 SLA 都被管理、报告并维护。它会迅速变得让人困惑，并产生没有任何特别优点的大量工作。目的是得到协议来帮助提高服务质量。这样，当制定协议时仔细考虑是会有用。在确定何时建立服务层次管理的协议时，关于确定什么重要的概况在这里。

4.1.1 通用清单

- 用户与运行部门之间达成的什么被实际衡量的协议。必须从用户的角度而不是 ICT 服务部门的角度来看。
- 衡量并弄清 SLA 中包含的指标
- 为服务层次决定真正的目标（不能遵守的承诺没有意义）
- 连续聚焦在服务的控制上——监控并将实现的结果定期报告

4.1.2 计划

运行中心有技术能力来衡量 SLA 中包含的价值，这是重要的。必须从一开始就考虑进来。

进一步，重要的是确定那些依赖于分包商并由此不同提供服务保障，或依赖于分包商的相似协议的服务。给出依赖的定义是因为应该明确谁去改正问题，并且避免错误被纠正前无休止的谈判。

对于不同的用户组，或者在学年的不同阶段，服务的层次可以不同。例如，教师与学生之间会有差异，进行考试时会有更高的服务质量。与所有相关用户的对话是重要的，来确保能够衡量每个用户组最重要的事情。

4.1.3 实施

必须准备服务目录，所有的服务包含在 SLA 之中。在这个目录中，服务通常是应用（程序）。不同的服务经常要求不同，并且由协议中的不同的目标反映出来。

建立并不断调整用户的期望，这是不能被高估的。用户经常对包含的系统和服务期望过高。ICT 服务部门的责任是，在签署服务层次协议（SLA ）前将期望降低到现实水平。运行管理部门也必须确保所有的用户被真正通知，并且了解通过协议所达成的期望的服务层次。

对于 SLA 的结构，请参见 服务层次协议中的章节。

4.1.4 运行情况

监控真正实现的服务层次，并反馈给客户，这对服务台与用户之间保持良好关系是重要的。报告细节的格式和层次应该在 SLA 中处理。

必须与客户定期举行会议，例如每季度或每学期。这些会议应该导致下个时期的坚实计划，并且导致可能实施新的服务协议。

4.1.5 服务层次协议（SLA）的内容

4.1.5.1 简介

合同方的名称和联络信息，所包含服务的描述，协议的期限，客户与供应商之间的责任。

4.1.5.2 服务时间

在哪个时间段内协议是有效的（像从周一到周五，从上午 8：00 到下午 4：00），在确定的日期和时间内的任何特殊要求（例如考试），和要求延长服务时间限制的途径。

4.1.5.3 可用性

访问服务。最好用一个或多个服务不可用的时间长度来衡量，例如一个日历月。根据例如用户重要性的程度来达成不同服务的不同层次。

重点要强调这是达成协议的服务时间内的可用性，而不是全天、全周和全年（被称为 24/7/365）的整体可用性。例如可以达成这样的协议，系统应该在工作日的 8：00 到 18：00 可用，除了另外达成协议，在这个事件段之后以及在周末，是否使用计算机系统是难以预料的。

可用性还意味着通过电话或电子邮件得到支持。例如，在工作日的 08 点到 16 点之间是否可以联系服务台，或者是否可以全天联系，或者是否可以在下午和晚上，或者甚至在特定的周末。

4.1.5.4 稳定性

通常根据一段时间内宕机时间的量，或者两次宕机时间之间的平均时间来衡量。还可以衡量系统在宕机时间后再次开机所耗费的时间。

4.1.5.5 支持

通常衡量通过电话（例如 1 min）或电子邮件（例如 30 min）对来自用户请求的回应时间。当操作者请求支持时，消息根据严重性分类，并带有保证回复的时间。还有关于如何迅速开始修复错误的协议，这依赖于接收到哪种请求。

支持还是关于在白天或晚上的什么时间能找到人。在 08 点到 16 点的上学时间都应该可以获得支持，或者整个晚上或周末都可以得到支持。还可以在某些假日得到支持。

可获得支持的时间通常在 SLA 中。还要达成协议，关于可以以固定价格获得什么样的支持，在任务的基础上还有什么必须另外解决。协议规定了处理询问的过程，既包括修理什么也包括何时进行。

4.1.5.6 容量

可以用特定应用中某些操作的平均响应时间来衡量。会衡量系统的用户体验。

4.1.5.7 更改管理

对来自用户的更改请求进行管理、同意和实施时间的衡量。

4.1.5.8 安全

可以用一定时间内查明的安全事件的数量来衡量。非常重要的是每个用户的责任要分清，从而确保应用质量保证。

4.1.5.9 记账

价格，记账的时间和结算准备。

4.1.5.10 报告与跟踪

对衡量服务层次的报告规则与周期的描述。推荐规律地举行会议，例如每季度，来彻查报告并预先计划。

4.1.5.11 约束与可能的激励

达成的协议没有做到时进行价格减免的规则。与由于连续违反已保证的服务层次而取消协议的升级步骤与规则。对实现的成就或好于预期的服务的可能的激励。

请参见 SLA 的附录 A。

4.2 财务管理

组织很少能有它们 ICT 花费的概况。挪威自治市 2001 年的调查显示，只有 1/8 的自治市具有 ICT 预算。学校的情况可能不会更好。落实 ICT 预算是重要的。用户经常认为他们为不满意的服务支付得太多了。这经常在用户与 ICT 部门之间产生冲突。

将真实的 ICT 成本存档，这对于运行中心和用户都非常有用。没有的话将难于适当地做出预算。多数时候难于做出现有 ICT 方案的成本/收益评价。校长应该知道 ICT 预算，同样也知道工资预算或教学辅助预算。

有三个主要的关键过程与 ICT 服务的财务管理相关：

1. 预算
2. 会计
3. 记账

4.2.1 预算

预算的目标是对所期望的 ICT 成本做出实际的估计。预算通常包含各种替换的解决方案。它应用于设备和软件，以及您所追求的层次。预算是后续与教育负责人和/或政治家进行预算谈判的起点。

预算必须包括人力和设备成本。一些组织只计算购买设备的成本，省略了用于 ICT 方案运行的多达 60 -70% 的人力成本。获得所有的设备是必须的。

有这样的例子，一些自治市忘记计算学校里的电源线和计算机网络的成本。然后您必须忘掉每台客户端机器的大约 2 000 NOK (10 NOK = 0.85 GBP/1.18 EUR)。对于 70 台新计算机，我们需要大约 140 000 NOK 用于网络和电源。

预算中包括替换的解决方案也很重要。这用于运行与设备。今天有几家经销商以不同价格和质量专门从事学校计算机设备的运行。同时使用的用户数量，以及维护的机器和软件的型号，是重要的。

如果想要所有的教师和学生拥有笔记本，那么相比于每台户端机器机有三名学生使用的台式机方案，很容易得出成本将高出 5-6 倍。

4.2.2 会计

会计将主要包括购买设备、电缆、维修、运行和额外服务的报价。当会计周期结束时，重要的是彻查数字，并将它与预算进行比较。

4.2.3 会计与记账的计划

并不是所有的自治市都有能够显示学校具体 ICT 成本的会计系统。有很多实际原因，如由于自治市的集中而导致的折扣或相似。这样，重要的是做一些计划，当针对预算来评价会计内容时，您能了解运行和采购成本的概况。

一些组织会有复杂的和耗资巨大的会计步骤。例如，如果付账延迟就会迅速得到额外的收费，或者必须有很多人来同意支付。所以重要的是就采购与运行的良好记账实践达成协议，以便能够控制，也是及时处理支付而无需漫长的决策过程。

4.2.4 实施

支付方法由 SLA 规定。当它进入会计系统时，必须与财务部门达成协议，能够以方便的方式得到报告，从而得到 ICT 成本的必要会计概况，而无需耗费漫长的时间来生成。

4.2.5 日常操作

关于合同，通常包括固定额度与可能的附加服务的固定月度记账。根据当前的运行合同，以及进行的额外的服务，由会计办公室进行记账。重要的是根据为客户执行的任务，与会计服务部门保持良好而频繁的接触。

4.3 容量管理

容量计划用于确保 ICT 方案的所有部分具有充分的容量来保证用户的需求。这包括：

- 监控 ICT 服务及其相关基础设施的性能
- 系统的配置，来确保它们被优化地用于用户实际所做的工作
- 理解用户的需求，并为系统可能的更改进行计划，来照顾未来的需求
- 与预算官员合作进行资源计划
- 准备容量计划，来确保根据达成协议的服务层次来交付运行

容量计划是关于平衡的：

- 相对于容量的成本。预算限制了能够实施哪种可能的解决方案
- 供给与需求。系统必须由容量来处理用户设置的需求

容量计划的目标是避免意外。

4.3.1 监控

好的容量计划是必要的，使系统能够被连续监控来的到必要的数据。
监控的典型数据是：

- 处理器的使用
- 内存的使用
- 每个任务的 CPU 用量
- 用户每个任务的响应时间
- 打印机管理——打印机的数量、队列长度、打印的时间
- 存储容量
- 客户端的数量
- 登录的数量
- 同时使用的用户的数量

在 Debian Edu 中，Nagios 用作监控工具。

4.3.2 分析

根据监控常规收集的数据，可以尝试识别系统的任何瓶颈。例如：

- 不良的或变化的硬件应用
- 设计不良的软件
- 内存容量的不良使用
- 数据存储、内存或处理器的瓶颈
- 网络的瓶颈

4.3.3 配置

如果数据分析揭示出可瓶颈，那么需要以更好迎合用户需求的方式来尝试设立系统。
这里的列表是通常遇到的瓶颈，以及如何做来排除它们：

瓶颈	行动
瘦客户端上声音、U 盘支持和 DVD 消失。	安装无盘工作站 (> 800 MHz 处理器，> 256 MB RAM)
60 台瘦客户端连接到服务器上，并想要更多的 PC。	选择无盘客户端，或安装另一个瘦客户端服务器
在扩展了 20 台设备而没有获得新的服务器时，瘦客户端运行缓慢	在服务器上安装 2GB 内存
升级到 Skolelinux 2.0 后，32MB 内存的瘦客户端不启动	打开瘦客户端上的交换系统，或者降级到 LTSP 4.2，它设有交换系统。
当 50 名学生登录相同的服务器时，Flash 动画使客户端变慢	安装无盘客户端

4.3.4 实施

系统配置上实施的可能的更改，必须根据为系统更改而设置的指导方针来进行。在生产系统可以进行更改前，也必须执行计划良好的功能与性能测试。进行测试来避免生产中设置更改而导致的运行干扰。

4.3.5 准备容量计划

容量计划基本上是基于客户当前需求和未来计划的知识，对 ICT 系统的投资计划。

容量计划应该每年更新并被处理，通常与预算过程相结合。计划应该包括后面的项目：

- 简介
- 先决条件
- 概要
- 当前和未来的用户需求
- 服务概要
- 资源概要
- 改善的地方
- 成本模型
- 推荐

4.4 可用性的管理

ICT 服务的良好而稳定的可用性，对用户明显是至关重要的。

从用户角度来看，可用性依赖于后面的假设：

- 技术部分的可用性
- 对失效的容忍
- 维护与支持的质量
- 处理运行服务的步骤和指导原则
- 数据的安全性、整体性和可用性

可以以几种方式衡量可用性。但在展示例子之前，我们将指出什么会是困难的目标数字。如果我们为可用性付出系统新的努力，那么我们必须弄清不同事情意味着什么。例如什么意味着可用性的百分比。

我们说“带有计算机程序的计算机”就是服务。有一天如果计算机程序不工作了，即使其它程序工作良好，那么服务也是不可用的。而如果一间教室里的计算机程序不可用，但是用于学校其他部分的服务是可用的 (因为下层服务)。这就是在实践中弄清并工作的困难事情。

4.4.1 可用性的衡量

可以用几种方法来衡量可用性。这里是一些例子：

数值	意义
可用性百分比	数值可以是 08：00 到 18：00 之间的可用性小时数，如果一天中系统宕机 1 小时，那么在达成协议的时间上，系统可用性为 90%。如果以一个月 20 个工作日来衡量，那么系统的可用性时间为 95%。
不可用性百分比	在达成协议的开机时间，例如每天 10 小时，如果系统宕机一小时，那么系统的不可用性为 10% 的时间。以 20 天来衡量，我们会假设系统有 5% 的时间不可用。
宕机时间	可以在例如一个月（20 天）内，就可接受的系统不可用时间数字达成协议。可以是那段时间内，且在 08：00 到 18：00 之间最多不可用的小时数。
错误频率	甚至每天或每个月的错误频率是可以衡量的。因为系统在 08：00 到 18：00 之间宕机的故障一个月有 3 次，就是一个这样的例子。
故障的结果	对于判断如何回应超出一般错误修正的错误，测量值是个普通的起点。例如客户或学校会要求为当前月份的运行协议更少地支付。

最重要的是您的测量以最好的方式描述了客户的体验。这样，应该衡量那些对用户最重要的参数。

来自学校的反馈是主要是打印机出问题。这包括打印队列停止，缺纸或缺墨的每一部分。也有人经历了浏览器不稳定，OpenOffice.org 套件挂起。这会在您的宽带连接不稳定时，以及您的文档与互联网连接时发生。

4.4.2 基础设施

为了具有稳定的计算环境，人们会依赖于技术质量足够好的网络。因为物理计算机网络是临时的而且量差，几所学校经历了由此导致的不稳定。

今天，很多人投资在无线网络上。如果这样做，还必须知道无线网络具有显著的弱点。无线网络具有有限的容量。当 30 名学生同时观看来自互联网的影片时会非常卡顿。无线网络还有覆盖不到的地方。意味着有的地方会覆盖不到，这导致一些程序在盲区结束。这样网络连接或是比较差，甚至一点也没有。

对维护公司和 ICT 服务供应商的可用性要求，应该规定向学校提供质量良好的网络服务。

4.4.3 《单点失效》

系统的一些部分必须工作。例如防火墙失效，会导致安全性降低，或者（如果幸运的话）关闭整个网络。后者还能够导致分发地址的 DHCP（动态主机配置协议）出问题。

运行部门有责任知道哪部分会使整个系统都停止。重要的是找到这些点，并且以您能承受的程度一个接一个地消除错误。如果不能承受消除错误的源头，那么就必须与整个计算机网络突然不断停止的风险共存。

使每一部分停止的错误源头可能是逻辑的而不是物理的。对于计算机网络和数据库，这特别正确。因此重要的是在遇到这样的错误时要具有广泛的思考方法。

4.4.4 风险管理

必须考虑在网络中承受什么风险。当硬盘失效时用户丢失个人文件和数据是否是可被接受的？能够多迅速地更换故障设备？在病毒攻击后，一些学校花费了几天使服务器启动并再次运行。自治市可能没有资源分配来修复错误。

多数运行工作继续来维护达成协议的服务层次。这是有关避免失去信心和用户满意度的。风险管理是关于使适当的资源就位，来保持整个计算机系统在线，并且在出问题时准备好资源，并满足修复需要。

4.4.5 测试

在一台单一 PC 上安装设备和软件，与在一百台，甚至几千台计算机上安装的区别巨大。负责几百台机器时，PC 上活动的小的故障如果影响到几百名用户的话，会意味着巨大的不稳定性和不满意程度。

为了避免安装过程中产生错误，并为稳定性做贡献，有必要将使用的设备和软件进行测试。这是关于符合所期待的质量的。如果您想要稳定地运行，那么必须经常选择仅低于最新版的设备和软件。

应该避免采用版本号以 0 结尾的软件。例如，应该避免使用 OpenOffice.org 4.0。应该在版本号达到 4.0.2 或更晚时采用这个办公软件。然后程序会被修改掉几个错误。这同样也应用于硬件。

服务器通常具有版本略旧一些的处理器，更坚固的内存，以及硬盘。这是因为很多人同时使用这个硬件。小的故障对一名用户可能不意味着什么，但如果 30 名用户登录计算机就会导致宕机时间。

所以，测试是关于使用可靠的设备和良好地运行半年或一年的软件版本。测试也是关于在较小但实际的文本中试验不同的部分，来确保每一部分正确工作。采用最新的版本或者甚至是 beta 版的软件或完全是最新的硬件，通常会导致维护的很多问题和额外工作。不在真实环境进行小的测试就设置生产系统，通常会导致显著的四处救火和不满意的用户。

当在生产中的设备上进行较小范围的测试时，有必要与受影响的人协调。此外，必须选择何时测试。不应该在例如使用 ICT 工具进行考试的时候进行新的部分的测试。

4.4.6 设计的改进

运行部门通常值得花些时间来增强那些产生很多运行消息的系统。如果用户得到很多垃圾邮件，然么安装垃圾邮件过滤器会是明智的。对于那些持续忘记他们密码的学生，如果教师必须请中心系统管理人员帮助他们的话，会有很多额外的工作。为了避免额外的电子邮件交流和双倍的工作，可以授权教师给学生新密码。

有两个设计改进的例子，它们简化了维护并使用户更满意。运行良好的维护团队有这样改进的排序列表。基于相关的问题有多么经常地显示在服务办公室的消息日志上，以及估计每项改进应该包含多少工作，来将这些排序作为规则。

4.4.7 可用性的计划

这意味着要基于运行成本对 ICT 服务抱有现实的期望。针对期望的接触性是什么而进行计划。例如，当学校需要服务器在崩溃后 1 小时内就启动并运行，那么必须要有原装的机器作为备份，作为故障机器的替换而插入。一小时内所能做的就是将备份文件复制到备份机器上。

对于无盘工作站或瘦客户端失效时，学校应该准备少量机器和显示器。学校 ICT 联络人可以取回并安装替代的机器。这可以容易地进行而不用等待几天去订购设备来填充。

4.4.8 恢复的计划

对于准备替换那些任何产生缺陷的设备，用户还期望能够挽回丢失的文件和数据。这样，规律地备份数据并保持配置文件的副本是至关重要的。还必须拥有架构图和系统的描述，当某些部分故障时使 ICT 人员能够迅速将系统设立起来。

定期备份用户数据和设置是至关重要的。必须预先计划适当的设备和适当的服务。常规途径必须计划出来，在某些错误情况发生并且系统必须恢复时按照它来执行。

4.5 服务连续性

运行连续性或连续性管理通常是工作的最昂贵部分。对运行连续性的需求将需要大量的投资，这必须在制定 SLA 时达成协议。例如可以达成协议，对于某些服务没有灾难计划。如果您有灾难计划，但没有偶尔测试一次的话，那么价值会非常低。通常这是昂贵的。有这样的例子，客户和管理部们封锁机房并关闭电源，来测试 IT 部门的准备情况。

在某些时间中会更加需要运行连续性，像在考试时间中。然后会需要额外的设备，从而在服务器硬盘故障的情况下，能有将备份准备好的设备。但即使这样也使操作人员需要相当多的附加工作。

一位 IT 协调员告诉我们，一天如果计算机系统的某些部分出故障的话，考试可能同样会推迟。这样即使花费很多，也少于每所学校有两倍的服务器。有个学校漏水的例子。通常推迟考试一两天来维修故障。必须以相同的方式考虑学校数据的解决方案。如果为教师和学生建立家目录的备份，您就有时间考虑而不用每所学校的系统都加倍了。在自治市的建筑中备用一两台服务器就足够了，如果有些部分出故障的话，这可以在学校迅速地移动并连接上。

5 ICT 基础设施管理

这部分的运行文档是关于技术的巨大扩展。关于服务支持和服务交付的其它章节，是关于工作过程和步骤的。基础设施管理是关于 ICT 系统的计划、设计、部署和正在进行的管理的。目的是提供适应组织需要的 ICT 方案，它能够以可承受的成本而随着时间来运行。

好的计划、行政和管理是确保发展良好的 ICT 服务的关键，服务就可以适应随着时间而变化的组织需要。这是关于很好地使用资源，并且拥有为了提供良好 ICT 服务而所需要的技能和能力。

即使有发展良好的基础设置，也必须期望 60-70% 的成本用于运行，也就是用于服务支持和服务交付。相似的，基础设施组成了总成本的 20-30%，这部分必须与运行同样严肃。选择的基础设施还主要冲击运行所耗费成本，以及系统所能够交付的。

多数人将基础设施与道路、上下水和供电相联系。如果我们要安排满足某个住房建筑标准的话，建造房子需要基础设施就位。在计算机世界中，基础设施通常与数据网络相联系。这是在 1980 年代。经过其后的 20 年，基础设施扩展到网络、计算机、软件和维护。因此在文档的这部分都是基础设施的网络、硬件和软件的显著部分。

我们甚至在这里聚焦在实际的计划与实施上。从那些通过预算和采购而具有良好的 ICT 计划的自治市，我们已经收集了坚实的计划数据。我们彻查了设计和计划、部署过程、运行过程和支持。重要的是记住在运行方式上的差异，服务台用例如支持所做的，与支持运行部门用例如到学校的网络电缆来做的之间的差异。在基础设施管理上基本有四个过程反复出现：

<ol style="list-style-type: decimal;"> 设计与计划过程 ICT 战略的发展与维护，和组织的 ICT 基础设施的适当解决方案的部署与实施的过程。   部署过程   以对组织的影响最小，来考虑行为的实施并部署，和/或设计与计划的 ICT 方案   运行过程   交付与/或维护想要使用的 ICT 基础设施的所有行为和积极性。   技术支持过程   所有当前与未来基础设施解决方案的评估、支持和质量保证的知识发展。

5.1 设计与计划

设计与计划是关于为机构的需要而提供 ICT 基础设施的发展与安装的无所不在的战略指导原则。它不只应用于基础设施网络、计算机和应用。基本过程同样必须就位，使技术发挥作用。它应用于服务台和服务管理过程。

避免制定日程或图省事是非常冒风险的。这样，通常明智的是花费多一点时间和精力在计划上，这在实施过程中会降低风险并提供显著的优点。多数搁浅的项目缺少计划而这样做。使 ITIL 过程在组织中就位，完全依赖于准备和计划，结合有效地使用人力、过程和产品（工具与技术）。

非常重要的是在计划 ITIL 时，与组织的每一部分沟通并对话。在挪威，这由劳动法案 §4-2 来规定：

- 雇员及其代表应该被通知在工作的计划和执行中使用的系统。应该给予他们必要的培训以熟悉这些系统，并使他们参加系统的设计。

目标是为您的组织交付正确的 ICT 方案。这些必须易于维护，并适应学校的需要。在长时间，以及当系统扩展时，解决方案必须要合理。在设计和计划过程中，通常应该与掌舵组和参考组关联。好的项目确保有技能的人在掌舵组，而做贡献的人在参考组。好的计划者聪明地使用这些组和其他雇员，来提出好的解决方案。

我们新建了一个基础设施项目中行为和交付的清单。

反馈

- 为学校计划，课程和活动的计划
- 现有的 ICT 战略
- 期望什么样的运行服务
- 当前的 ICT 系统与运行管理

过程

- 彻查所有建议和文档
- 看看执行的设计与计划行为的其他人
- 做出并维护 ICT 计划和决定
- 做出并维护 ICT 架构
- 做出并维护 ICT 战略

交付

- ICT 战略
- ICT 决定（具有正当理由）
- IT 计划
- 整个 IT 架构
- 过程和步骤的设计与计划
- 组织结构与框架
- 设计、计划标准与决定
- SWOT 分析（优势、劣势、机会、威胁）
- 用户案例与使用性研究
- 要求列表与投标文档
- 项目计划
- 技术图纸、计划与路线图
- 评论与反馈

如您所看到的，为基础设施项目执行了广泛的计划。自治市中的学校的 ICT 项目如果交付 500-1 000 台计算机及供电、计算机网络和软件，那么可以迅速达到几百万挪威克朗（几十万英镑）。对于这样的量，重要的是具有好的可行的计划与现实的预算。

有几个自治市低估了学校中对 ICT 资助需求的例子。他们安装了很多好的设备但一直没有使用。可能是软件缺失了，网络的质量差或者缺乏电源线。如果 10 所学校的 800 台机器要有网络和电源线的话，自治市会迅速看到另外的 2 百万挪威克朗的花费（大约 165 000 英镑）。

好的发展计划是制定出来避免意外的。计划也要制定出来确保现实预算下适当水平的志向。

5.2 部署

定义：

- 是关于对组织的干扰最小来设计和计划的行为和/或 ICT 方案的实施与部署。

项目的计划应该评价学校具有什么设备，并且有多少可用。由此，在每所学校并在每个中心运行服务部门新建计划，来展开新的设备或者更换设备。

这是关于将设备放置在被使用的地方的。设置桌子上的每台 PC 并用网线将它连接到计算机网络上。将电源线插入电源插座。连接屏幕，并将网线放在正确的交换机中。

术语展开（或部署）用于放置设备，并用于在很多计算机上安装并配置软件。展开软件也可以被称为“版本发布管理”。但单词展开又短又好，尽管需要弄清谈论的是硬件还是软件，它需要完全不同的步骤。

展开管理是关于实施那些发布过程中所计划并设计的。从应该的地方找出设备通常比我们想象的难，并且要花费相当的时间。这是因为要包含很多部门来交付设备，或者在很多部门中来接收它。以某种方式可以说，车轮螺丝将车轮固定在轮轴上，这与展开与是一样的。

为了使每一部分就位，这完全依赖于很多协调。必须制定好的战术计划，这包括更改管理和项目管理。必须确认展开与设计和计划过程相联系。

低估部署对现有系统影响会经常发生危险。采用新的解决方案或升级，会影响或更改组织。工作路径被更改，并且要采取新的方式来完成任务。

在学校里，更改意味着在学校课程中引入 ICT 工具。这对于教师是新的和不同的。很多教师并不熟悉如何在教学中使用设备。同时，运行和维护服务应该就位，来为学校提供安全稳定的 ICT 方案。这导致了组织的更改，它必须要计划并需要资源。这样，重要的是在计划与展开之下进行考虑……

5.2.1 展开过程中的角色

建立 IT 基础设施可以比作建造房子。当建造房子时，真正需要的是建筑学家、建造者、所有者、泥瓦匠、木匠、水暖工、电工以及一个或多个监督（工长）。这同样应用于基础设施的部署。我们将推荐的角色概括为运行标准 ITIL 的一部分。

- 部署过程的所有者——负责部署的过程，并负责要以好的和有效的方式来进行。
- 负责展开的项目经理——负责开发适当的计划来部署 ICT 系统，并每天指导展开。
- 负责展开的协调人——负责协调所有展开行为。协调人应该确保项目获得系统的目标和验收要求，并确保有序交接。
- 展开分析员——负责确保设备以适当的设置来安装。应该确认设备和前提适于达成协议的标准、测试和部署……
- 展开团队的雇员——负责 ICT 方案和网络环境，并且支持验收和测试过程。

正如我们看到的，展开触及了组织的很多部分。它在技术上影响软件和设备的配置与版本。它还影响更改的过程以及如何进行服务工作。

必须仔细考虑每个角色指定给谁。甚至在花费几十万英镑的全规模展开项目中，一个人也会具有几个角色。但给予一个人几个角色并不总是谨慎的，因为这非常难以满足对设备的供应商和客户的跟进。

对于最小的升级和调整，可能很快就会有太多的角色。例如，会不需要项目经理来放置新的服务器，或更换开关。这是基础设施的一部分，但非常贴近运行和维护。这里重要的事情是区别基础设施的展开与运行服务。运行部门在根据达成的协议运行之前并不接管设备。换句话说，您拥有转移文档，知道根据达成的协议设备已经交付。

5.3 运行

定义：

- 用于评价、支持和质量保证所有当前和未来基础设施解决方案的知识的开发。

设备的运行是关于使工具和机器就位，作为根据达成协议的 ICT 服务的交付基础。设备的运行强烈地聚焦在技术上。这支持了 ICT 系统进行的所有其它行为。运行部门通常被看作楼道尽头的办公室中隐藏的支持。“作为卫生服务”。当有些部分出问题时，首先联络运行人员。好的运行服务对 ICT 工具正常工作仍然是至关重要的。没有好的运行部门，那么就得接受时间的损失和不能解决的问题。例如学校在使用 ICT 工具进行测试时会遇到问题。

人们会问是否需要运行部门。在今天的高科技世界需要人来运行设备吗？还没有人发现一种方式来自动地解决运行任务吗？为什么您让人员来运行？答案通常是自动进行的工作与人们必须跟进的工作的平衡。重要的实现方式是，当问题发生时多数人希望有人能够对话。他们想要错误被修复，并且他们想要得到每一部分都平滑运行的反馈。这种类型的错误只通过更换机器来修复并不特别容易。

好的运行部门在任何可能的地方选择自动化。同时也需要人员来监控，并保持对自动化解决方案的控制。自动化必须进一步开发。但仍有地方自动化并不充分。设备会中断，以及应用会崩溃。您需要有人来方便地修复错误，或者对不能修复的进行替换。

组织很差的运行服务浪费了大量时间来救火，并通过手动过程来工作，手动过程本可以通过自动化过程来避免。花费在自动化上的时间可以迅速通过腾出的时间来回报自身。这个时间可以用于为用户改进支持，提供更多服务，并提升质量。为了永久的修复错误，一些人可能必须延迟升级，或删除用于临时修理的服务。当手动监控系统会另外占据所有时间时，这会腾出时间来适当地修复问题。

运行多数是关于防止错误，或将报告出错的设备进行纠正。人们一般不会完全熟悉错误的原因。必须检修来找到故障。好的运行部门的雇员是有天分的。他们使用过去的经验来揭示出错误。然后他们几乎直接找到问题的解决方案并纠正错误。

5.4 配置项目

配置项目（CI）是基础设施的一部分。它通常描述了对更改或问题的希望。也许要使新的服务就位，或者对已经投入生产的服务做出调整。它通常是升级一些设备，或者是得到一些新的部分的问题。

当涉及设备和基础设施时，配置项目在配置管理中是重要的。通常配置项目是关于系统是否应该：

- 运行
- 关闭
- 关机
- 启动
- 中断
- 去除

5.5 技术支持

技术支持确保具有正确技能的员工来支持计算机网络提供的服务，与服务台巩工作的员工一样。作为技术支持的一部分，人们应该拥有带技术建议的深入的文档。建议应该提供展开行为的信息、指导原则和例子，并描述 ICT 服务所有部分的支持与维护。为了实现这点，员工必须知道或者能够得到，关于使用中的技术、过程和文档的信息。如下面所列出的，技术支持包括几种行为：

- 与新技术相关的研究与开发。
- 回应来自服务台的事件报告并处理一般问题的三线服务。
- 交付管理——对深入的知识或使用中的技术的理解比较缺乏的技术支持，并且需要来自其他人的技术支持。
- 设计与计划的连贯性。特别是支持和文档。例如，当准备投标文档时。
- 部署新系统版本的连贯性，和在运行环境下的验收。
- 对来自报告和日志的信息的分析、解释和发布。
- 提高交付的 ICT 服务的质量的战术组成。

6 设计和计划的例子

作为如何制造基础设施的例子，我们采用了 Nittedal 2005-2008 中的为学校所作的 ICT 计划的显著部分。我们已经做出了一些调整，并把它做得更通用，并使其他人更容易复制。

- 计划的背景
- 对 ICT 工具和服务期望什么

- 所需的技能
- 投资
- 目标
- 学生与教师
- 状态与目标
- 成本
- 其它购买选项
- 软件、学习平台和服务
- 软件和学习平台
- 在线服务
- 资源使用
- 集中的运行和角色
- 运行和支持成本
- 推荐
- 附件

6.1 计划的背景

在它的“数字能力项目 2004-2008”中，教育与研究部为在挪威的学校中使用数据技术设定了目标。“到 2008 年，我们将具有基础设施、组织和文化，在教学与学习中 ICT 的研发和教育应用方面，使我们的学校成为世界上的领导者之一。”

能够使用数字工具，被定义为所有 13 学年中的基本技能。学生基本技能的发展在所有课程中被给予优先性。新的科目将意味着学生学习时必须大量使用数据工具。学生在学习时应能够使用与工作中相同的技术，从而形成最终评价的基础。当使用数字工具进行考试时，这会提供在学习过程与最终评价之间更好的连贯性。

挪威的国家调查（Skolenes digitale tilstand 2003, ITU, 2004 年 2 月）显示，计算机极少包括在小学的课程中，并且在学校里计算机几乎不被学生使用。

这个计划建立在“Nittedal 的学校能力计划（2005-2008）”之上，并实施了在 Nittedal 的学校中关于数字知识的能力计划目标。此外，这是与我们的 Linux 网络运行相关的投资与资源需求计划。

6.2 对 ICT 工具和服务期望什么

对于不同学校的不同组和 ICT 承诺的不同方面，我们有不同的目标。简要地说，我们的目标是：

- 通过增加对 ICT 设备的物理访问，在学生和教师中增加 ICT 的使用。
- 以工具为导向，由此，重点是学校课程中 ICT 工具的使用。
- 充分访问对于音乐作曲的每一部分的教育软件，并使用互联网来学习写作、模拟和游戏。
- 节约，并以最佳的方式使用我们所具有的金融资源。

通过这些主要目标，我们将实现：

- 教师在工作中得到良好的工作与交流工具。
- 学生有机会成为 ICT 的个人用户，并且在日常学校生活中使用 ICT 作为自然而然的工具。
- 学校能够物理地满足与 ICT 相关课程的各个方面。
- 运行和维护成本不大于学校预算所能够承担的。

6.3 所需的技能

为了建立并维护基础设施，您需要很多不同专家之间的协调。作为例子，我们将显示您在哪些设备领域需要专长。这些设备领域在普通学校中形成基础设施的一部分。

- 具有局域网（LAN）和广域网（WAN）的网络基础设施。多数情况下容易得到交换机和其他网络设备。这是货架套件。但是，必须为集中运行而设计的计划架构而设立。这是专家的工作。自治市建立部门必须同意所作的更改。
- 支持客户端计算机、服务器和网络设备的电源（230V/110V）。很多学校没有为教室、计算机实验室或图书馆中放置的所有计算机安装电源插座。这是专家的工作，来计划具有充足插座并合规的电网。自治市建立部门必须同意所作的更改。
- 服务器和客户端支持更大量的网络服务和终端用户应用。得到正确的设备是个相当大的工作。它是关于找到具有合适的能力，好的质量，合宜的保修方案和低的价格的设备。
- 用于监控硬件的机器设置与系统。为了确保所有的设备运行，通常会伴随有远程监控系统。您能够以那种方式在集中运行中心了解设备健康状态的概况。
- 设计合适的环境或房间，来放置需要冷却的设备。计算机和网络电子设备放出相当大量的热量。首先，最近设备的生产商强调不断增加的能源消耗。这样，有时必须确保将过量的热传走。这样的冷却系统需要专家来安装。
- 对于软件不同性能要求的知识。用于视频编辑的程序必须运行在具有 >1.5 Ghz 处理器和充足内存的工作站上。其他程序可能在瘦客户端上容易地使用。对于不同类型的客户端机器能期待什么性能，必须具有相对好的知识，来选择设备进行正确的混合。这需要洞察在不同的科目和学校的不同房间中，计算机要被如何使用。
- 可选设备如打印机、视频投影仪、计算机主板以及类似的安装与设置。设立附件会迅速占据相当多的时间。例如视频投影仪需要用螺栓固定在天花板上，并且必须安排屏蔽电缆和电源来连接它们。打印机必须有网络节点，并连接到网络上。这种类型的安装通常需要专家来安装并设置。

除了需要不同专家来构建基础设施，您会另外需要：

- 部署过程的所有者——负责部署过程，并且负责以好的和有效率的方式进行。这会是掌舵组。
- 负责展开的项目经理——负责开发适当的计划来部署 ICT 系统，并每天指导展开。
- 展开的协调人——负责协调部署行为。协调人应确保项目满足应用于解决方案的目标和验收要求，并确保有序交接。这会是项目领导的助手。
- 部署分析员——负责确保设备所在位置的合适环境。应该确认设备和前提适合达成协议的标准、测试和部署。这会是项目领导的助手，他的任务还包括将计划的偏差报告给掌舵组。
- 展开团队中的雇员——负责 ICT 方案和工作环境，并且支持验收和测试过程。这是参加一个或多个子项目的雇员。

在组织上看起来会像这样

组织的部分	任务
参考组	应该代表系统的用户。他们提出措施的建议，来促进学校中好的日常 ICT 方案。
掌舵组	他们的任务是确保项目具有足够的资源，并且确保项目管理根据计划来展开。这个组会包括具有技能的专家，他们熟悉学校中项目的实施，系统解决的方案和 ICT 工具的使用。
项目	具有建立解决方案的任务。项目通常包括很多子项目，这些子项目交付解决方案中自己的部分。

6.4 投资

为了满足新的课程，学校必须为他们的学生和员工提供足够的计算机。这个投资计划包括将学校中的计算机库存提高到满足国家目标的真实成本。这些目标要求至少每四名学生一台客户端机器或 PC。设备的要求可能会在几年内增加，所以我们将它扩展到每三名学生一台 PC 工作站。所有教师应该能够在学校的日常工作中访问计算机。

在今天的学校里，学校网络包括服务器和瘦客户端，以及自治市中备份的共享服务器。由于我们可以在网络中使用二手计算机作为客户端机器，用户的计算机并不是最贵的（我们购买二手设备，并接收工业界捐赠的设备）。主要成本存在于教室中增加的电源插座，以及学校中可能增加的用电量账单。

数量增加的并发用户也会增加支持和运行成本。新的 PC 工作空间总是需要桌子和椅子。此外，所有学校以固定的价格接收宽带连接。后面我们会说明设备库存加倍的总成本。

2005 年 06 月 01 日的 PC 覆盖状态是：

1. 在小学，每台计算机 8.9 名学生
2. 在中学，每台计算机 4.4 名学生。

学生的目标：	每个学生组（以前被称为班级）应该能够访问至少 5 台计算机，加上学校应该有包括至少 15 台计算机的机房。此外，学校需要一些特别的设备用于视频编辑、特殊教育和读/写课程。
教师的目标：	所有教师应该能够在学校的日常工作中访问计算机。

机器的总数：

根据 2005 年 06 月 01 日的状态		2008 年的需求				
服务器的状态		客户端	笔记本	文件服务器 + 瘦客户端服务器：	客户端	笔记本
Holumskogen 市	1	25	5	2	68	15
Ulverud 市	1	35	5	2	111	15
Slattum 市	1	44	8	2	87	15
Rotnes 市	1	35	5	2	80	15
Sørli 市	1	31	5	2	60	15
Kirkeby 市	1	31	5	2	94	15
Hagen 市	1	7	5	1	46	15
Li 市	2	70	5	2	130	30
Nittedal 市	1	55	20	2	110	30
Hakadal 市	1	45	5	2	52	30
总计	11	378	68	29	838	195

我们设想了瘦客户端、无盘客户端和笔记本的组合。学校的基础设施应该能够在每间教室展开瘦客户端。学生可以在这里书写，计算，使用互联网和做展示。此外，学校还有机会将笔记本借给不同的组。以这种方式，在某些工作环境中能得到 PC 对学生接近完全覆盖。笔记本连接到无线网络中的服务器上。教学以这种方式变得更灵活。

6.4.1 学生

我们推荐至少每三个学生一台客户端计算机的投资，政府已经声明将其作为学校里 ICT 工具的目标。为了实现这个目标，我们需要将客户端机器的数量几乎加倍。

6.4.1.1 状态与目标

为了实现目标，我们需要将机器总量从 506 台机器增加到 1 033 台。这只是小于 600 台机器的增加。(瘦客户端、无盘客户端和笔记本)。

6.4.1.2 成本

我们已经以这些价格来计算，这些价格可能会被更改：

- 瘦客户端：700 NOK，每台
- 服务器：大约 50 000，每台
- 监视器：500，每台
- 便携机：8 000，每台
- 电源插头 750，每套
- 桌子/椅子：700，-
- 这里增加的资源意味着增加学校里 ICT 联络人的小时数。这里我们考虑每名教师的小时工资为每小时 NOK 270 (£22.5)，或者每年 NOK 467 100 (£38 925)。我们还考虑为自治市的集中运行部门增加一点的资源。我们期望为超过 1 000 台客户端机器的集中运行，新雇佣略少于 1 人。此外还有每所学校的 ICT 联络人，培训和 ICT 协调人。
- 许可成本。今天，我们可以使用学校现有的网络在笔记本上安装 Linux。这样，我们避免了租用微软产品，如 Windows 和 Office。学校租用微软程序的费用，多到相当于在 5-6 年里所有计算机的成本。
- 宽带协议，所有学校具有宽带连接。价格依赖于各个学校的协议。

最近二手设备比 3-4 年前得到的机器的性能更高。如果用户的机器具有 256 MB 的内存和 800 MHz 的处理器，那么这些会适合用作无盘客户端。这简化了对使用 CD / DVD 播放器、音频、U 盘及相似的支持。

2006 年	**2007** 年	**2008** 年	总计	
瘦客户端和无盘工作站	130 000	130 000	130 000	322 000
服务器	500 000		500 000	1 000 000
监视器	80 000	80 000	80 000	230 000
笔记本	340 000	340 000	340 000	1 020 000
其他：开关、电缆	150 000	150 000	150 000	450 000
电源插头/电缆	290 000	290 000	290 000	870 000
桌子/椅子	190 000	190 000	190 000	570 000
增加的资源、运行	700 000			
便携式机器的许可成本	40 000	40 000	40 000	120 000
宽带协议	100 000	100 000	100 000	300 000
总计	5 582 00			

6.4.1.3 其它购买选项

政治家、家长和教师已经增加了在中学中转向笔记本计算机的兴趣。笔记本和无线网络会给予学校在教室布置和教学上完全不同的灵活性。

只聚焦在笔记本上导致的问题是：

- 除了机器，我们只能购买微软的许可。
- 机器的寿命周期大约为 3 年。这样，为自治市带来每年的花费来覆盖中学新班级。

- 增加的保险成本
- 当所有计算机必须用电时，对电源插座的需求更大。
- 对 ICT 联络人资源的需求增加
- 为笔记本准备磁盘映像等等诸如此类的集中运行成本加倍，并且在 266 台另外的笔记本上维护本地安装的系统也都加倍。

这个选项的粗略价格大概是 1 200 万 NOK，刚超过 1 百万英镑。(这不包括保险费用的任何增加。)

6.4.2 教师

每名教师应该在学校能访问客户端机器。

6.4.2.1 状态与目标

状态：今天，学校有大约 65 台 PC 在大约 266 名雇员中分配。这达到了每 4 名教师一台 PC 的覆盖率。

我们想赞扬 Nitteldal 市的教师。新的课程对教师的 ICT 技能设置了高的要求。有必要确保 Nittedal 市的所有教师能访问计算机。今天的教师使用并依赖数据来进行计划并实施教学。他们制作文档并报告，撰写周计划、工作计划、年度计划和 IEPs。越来越多的教师在家庭与学校使用电子邮件来联络。

学校已经安排为他们的员工购买计算机。结果，计算机的数量随着学校而不同。我们瞄准在工作中使每名教师能够访问计算机。

这里我们概述两个选项，在 Nittedal 市实现 PC 对教师的全面覆盖。

6.4.2.2 成本

选项 1：瘦客户端与便携机的组合。这使每名教师访问瘦客户端 + 3.3 名教师分享访问一台笔记本。

成本	总成本		
替换选项 1	瘦客户端	140 000	
监视器	100 000		
笔记本	640 000		
其他：开关、电缆	80 000		
桌子，椅子	140 000		
电源插头/电缆	400 000		
许可成本	40 000		
总计	1 540 000		
另外的平板显示屏	200 000		
加上 LCD 总计	1 740 000		

瘦客户端对教师的优点是低的采购成本。与笔记本相比，我们还可以期望瘦客户端较长的寿命。

但二手设备通常没有平板显示屏。客户端机器的机箱可能很大，使教师的工作空间更少。如果我们为所有教师提供平板显示屏，就必须花费 3 倍的成本，从 500 NOK (£ 43) 达到 1 500 NOK。总成本会增加到 200 000 NOK。教师的整个设备花费 174 万 NOK （ £ 150 000)。

6.4.2.3 其它购买选项

选项 2：所有教师配备便携机

成本	总成本		
替换选项 2	笔记本	2 128 000	
无线交换机	80 000		
电源插头	75 000		
许可成本	117 000		
2 400 000			

我们看到的问题是瘦客户端的实际位置。教师通常在大的公共房间只有小的工作桌。每名教师的带有老式（CRT）显示器的瘦客户端会增加学校的空间问题。使用现代的平板显示屏幕可以极大地减少这样的问题。

便携机的优点是需要很小的空间。教师可以容易地将他们的工作带回家。缺点是便携机的寿命，大约是固定设备的一半。合理的假设是笔记本是维护台式机的 2 倍那么贵，比运行瘦客户端或无盘客户端贵三到四倍。

6.4.3 推荐的技术发展预算

在 2005 到 2008 年间，我们为学校的 IT 基础设施建立了后面的推荐内容。

数量	物品	成本
600	包括所有基础设置的瘦客户端和无盘工作站	5 582 00
200	带有平板显示屏和所有基础设施的瘦客户端和无盘工作站	1 740 000
总共 **800** 台客户端机器	总计	**7 322 000**

6.4.4 软件、学习平台和服务

软件在哪里运行依赖于基础设施和网络容量。在学校运行所有来自集中位置的安装是很好的，例如来自自治市的 ICT 服务部门或集中放置的运行部门。

必须考虑学校的网络对能够同时下载多少的限制，或者服务器最好放在哪里而得到设备的全部功能。与大约 30 名学生从例如 NRX 同时下载影片相比，一名教师做同样的事情会是非常的不同。如果学校具有 1.5 Mbit/s 的宽带容量，对于 30 名用户，不能同时从 NRK 下载影片。那么必须在学校使缓存代理就位。

6.4.5 清单的集中

UNINETT ABC 做了一个与 ICT 运行集中相关的带有推荐的文档 <<注脚(UNINETT ABC 的推荐：http://www.uninettabc.no/?p=veiledning&sub=annet)>> 。它给出了学校中关于服务器放置的建议，以及从学校可获得的带宽容量中可以集中哪些运行任务。

提高客户端和服务器运行的一般措施	相对于本地服务器的瘦客户端或无盘客户端胖客户端的锁定本地服务器	远程操作一些功能的集中本地服务器	地区性/国家性的服务器所有操作的集中
学校网络的容量	低带宽（ISDN）	中带宽（ADSL 或相似）	高带宽（光纤及相似）

6.4.6 软件

新的课程（L2006）强调了使用数字工具作为人的“基本技能”。我们希望 ICT 的使用应该超越教学，在提供了接触知识的方便途径的同时，提供了更多教育与管理工具，来支持学习活动和新形式的学习。给予数字化学习平台的经验，是能力计划的目标之一。目的是在 2006 年有一所或更多学校来进行试验。

研究显示，在学校中，在有限的程度上使用计算机设备来教学。研究还显示（ITU 监控报告 2005），计算机的使用停滞了，并且在一些科目中降低了。学校中 ICT 的应用经常个别化，并且学生学习成为消费者。学校里教学方法阻碍了知识的分享。极少有教师日常使用 ICT。互联网和文本相关的服务是学校里使用计算机的最重要的形式。

简单地说，教师太多地聚焦于使用工具进行办公室管理工作，如 MS Office 或 OpenOffice.org。他们应该聚焦在使用模拟，编辑图片，互联网上的音频和视频交流，以及游戏。

家庭使用非常不同。学生在家里是制作者，最多的是共同使用 ICT，并用于交流。他们聚在一起，彼此发送图片，交换内容，使用主要的机会来记录、编辑并分享影片，这些在今天使用宽带的计算机上是可能的。孩子们和年轻人在家比在学校还玩更多的视频游戏（ITU 监控报告 2005）。

研究者说，视频游戏是孩子们和年轻人的主要休闲活动。四分之一的孩子每天玩（青年机构 2006）。视频游戏是社交活动。游戏觉醒时，虚拟和物理交流出现了，从在控制台上一起玩，到参加年轻人可以玩的聚会。

重要的任务是，学校的发展做出贡献，来更新课程通用部分的通用教育观点。允许与阶段和学习战略相联系而发展的数字化判断和数字化教育，如国家 ICT 教育研究与资源中心所宣称的。

使设备得到更多的应用会需要教师相当大的努力。他们必须持续接受教育，以新的形式来学习将新的 ICT 工具用于教学。必须更多地关注年轻人对媒体的实际使用和交流的形成。提供教学平台和电子邮件是不够的。工具应该充分支持使用媒体的新方式。

为了实现这点，设备必须适应教师和学生在学校工作中使用的软件和在线服务。浏览器无疑是学生学习中使用的最重要的程序。很多人还会惊讶于 OpenOffice.org 或 MS Office 那样的办公程序与较低的年级无关。在那些年级，用于练习书写、绘画、交流、模拟和音乐制作的简单程序会有用。所以在软件的选择中，重要的是提供对互联网的良好访问，以及使用与学校课程相关的 ICT 工具来支持主动学习。

使用无盘工作站，您可以获得对多媒体、影片、U 盘以及更多的完全支持。瘦客户端的优点是能够再次利用 1995 年那么老的设备。那时机器还没有视频能力。USB 标准还没有充分发展。2000 年及以后的计算机通常具有更高的容量。这样的机器可以容易地显示来自 NRK、DVD 的视频片段，并且可以玩游戏。

无盘客户端的优点是他们提供了与所谓的胖客户端或本地安装了多数软件的计算机相同的性能。同时可以得到无盘客户端或瘦客户端那样低的运行成本。这是因为所有的软件在中心服务器上管理。

今天，Skolelinux 带有超过 50 个与学校相关的程序。此外，它还有浏览器、电子邮件客户端和带有 8 个办公应用的 OpenOffice.org。这远多于微软所带有的多数只提供了浏览器、电子邮件和 5 种当前的办公工具。

使用 Debian Edu，还以相对容易地为不同阶段的教育来定制菜单，从而可以减少教育程序的数量。特别是因为第 4 - 第 5 年级引用了一些应用。当程序在教育的第一和第二阶段流行时，当学生成长并学习更多时，教育会非常容易。此外，互联网上教育程序的数量在增长。这是工作在任何平台上的软件。所以您可以在家，在苹果、Windows 和学校的 Debian Edu 上使用程序。学生能很好地处理这个情况。

6.4.7 学习平台

市场上能够找到各种数字学习平台。有些耗费金钱，其它的免费。它们都为教师和学生提供了分享并存储文档，以及发送并接收信息的地方。

产品		价格的例子：
数字化学习平台	Itslearning	3 300 NOK，每所学校每年
School network		免费

6.4.8 在线服务

无论带宽如何，后面的功能都可以集中：

- 配置管理，也就是了解概况，并控制机器、网络、应用和服务的配置
- 程序管理，也就是了解概况，并控制对应用和服务的访问、使用和性能
- 更新与补丁
- 用户管理，最好有 FEIDE 兼容的用户管理系统（BAS）
- 许可管理
- 监控和衡量

可以集中或复制的服务的清单。例如备份可以集中。同样也用于带有中心目录服务器（LDAP）的用户数据库，中心目录服务器可以复制到每所学校。

服务	描述	本地	中心
Apache	Web 服务器允许所有用户建立网站		
CUPS	打印服务器。它的目的也是管理打印配额		
DHCP	计算机自动连接到网络上		
DNS	域名服务器		
LDAP	目录服务器，包含用于登录、文件分享和用户组信息的用户数据		
LTSP	瘦客户端服务器		
NFS	网络文件系统		
NTP	时钟服务器，使所有的机器具有正确的时间		
通过 SSL 的 SMTP/IMAP	学校本地每个人的电子邮件		
SSH	加密连接的远程控制		
Squid	网站的缓存（节省带宽）		
Webmin	通过 Web 浏览器进行系统管理		
用户管理	简化的用户管理		
备份	备份（应该在不同的机器上进行）		
SMB	用于连接 Windows 计算机的 Samba		
cfengine	系统设置的自动控制		
托管与服务监控	监控服务器机器的健康		
Appletalk	连接 Macs		
SQL-server	已提供（未设立）		

6.5 运行中资源的使用

对了计算机的每日应用，每所学校都有 ICT 联络人。ICT 联络人每周为他的工作分配 2-4 个小时。此外，自治市还有 50% 的职位是 ICT 指导教师，除了其他事物还从事能力与运行。在 2005-2006 学年，Linux 网络的运行会逐渐转移给自治市的 ICT 服务。

Kapp næringshage（业务园区）代表 Debian Edu 做了一个计算程序，估计了学校中用于 ICT 资源的使用情况。今天我们为超过 3 000 名用户以 2.1 FTEs 运行学校网络。（学校的 ICT 联络人为他们的工作分配了总共 1.6 FTEs，为自治市分配了 0.5 FTEs。）当 Kapp næringshage 计算我们运行的资源需求时，他们估计当前的需求是 4.6 FTEs。这显示了学校以较少的资源来管理了很多工作。

增加的资源必须主要放置在学校中，因为当 PCs 的数量增加时，学校的 ICT 联络人会更忙。PCs 数量的增加意味着使用的增加，以及在 ICT 工具的教育应用中对指导需求的增加。

与同时使用用户的数量相比，维护增加得更快，但机器自身的维护相对于机器的数量几乎是线性地增加。我们想要更多地聚焦在设备的教育使用上，并且想要多数增加用于学校课程中 ICT 工具的应用。

对于增加自治市 ICT 服务的资源还有些更大的需求，但因为经济规模的原因，这里的增加会很小。

今天学校难于对 ICT 联络人的时间排序。既因为必须从已经紧张的学校财务中拨出金钱用在这里，也因为对 ICT 联络人在学校执行什么和如何执行，学校缺少指导。

6.5.1 运行的角色

每所学校 ICT 联络人的责任：

- 监督学校的服务器室。
- 作为自治市的学校联络人——报告错误和系统中断。
- 执行简单的维护工作诸如更换鼠标和键盘，升级瘦客户端，和简单地打补丁。
- 作为学校的超级用户——能够在各方面建议同事：用户界面，电子邮件，视频投影仪以及相关应用。
- 参加 ICT 会议。
- 创建和管理本地用户。
- 执行简单的打印机维护。
- 创建和管理电子邮件账户。
- 在教学中促进 ICT 的应用。
- 在 ICT 指导教师的指导下执行简单的命令和操作。

对于具有 50 台瘦客户端或无盘客户端的学校，我们从经验中预期，这些任务每周将花费至少 4 小时。如果需要的机器较少，那么会减少一点小时数。而机器更多时，例如 150 台机器，那么本地 ICT 联络人需要每所学校大约 30% 的职位，来容易地处理技术维护工作。

如果学校不能为 ICT 联络人留出充足的小时数，那么上面列出的工作责任必须去除，并且如果学校能够使用更多小时的话，那么情况则相反。

超出于此的任务，例如更新网站，或作为指导者（超出一般的社团指导），必须对补偿/占据的时间来各自达成协议。

ICT 主管为后面的 ICT 和 ICT 服务主管，推荐了后面的任务。

运行：

- 用电话和电子邮件指导 ICT 联络人。
- 访问学校在计算机、打印机和服务器上检修缺陷和错误。
- 进行计算机设备的共同采购并加入共同协议等。
- 备份。
- 持续更新学校服务器上的软件。
- 通过招标在市场上采购设备和软件。

技能：

- 开发能力计划。
- 为学校提供 ICT 教育应用的课程。
- 运行课程。
- 培训学校的 ICT 联络人。
- 为教师介绍用户界面和标准程序。

需要多少集中运行资源？这依赖于您选择什么类型的客户端。工作站的运行几乎是无盘客户端运行的两倍那么贵。

6.5.2 运行和支持成本

运行成本的定义：

- 交付和/或维护所想要的使用 ICT 基础设施的所有行为和积极性。

我们已经描述了什么是真正的运行环境，考虑了积极运行的中等层次服务。挪威的“数字化竞争项目”是我们评价的基础。

主动运行就是在错误影响用户之前将它们发现并改正。主动运行的例子是，每周用新的磁盘映像更新笔记本。当教师在第二天早晨登录时，所有的机器已经根据学校的偏好而重置了。

在用户那里出故障前，运行部门会得到系统缺陷的消息。在用户注意到之前更正缺陷并修理故障。提供消息用于积极操作的系统例子是 diskstore。当磁盘出现缺陷时，或者磁盘缓存充满时，它们可以发出通知。如果计算机网络可用的话，运行部门还会得到信息，或者当用户登出时进程是否必须终止。

- 优点：假设访问正确的工具和正确的技能，可以实现高稳定性的系统。维护多种型号的计算机变得更容易，因为他们知道设备的工作还是不工作，以及是否可以更换故障设备。缺点 1：需要较高的技术专长。建立和管理运行的更高成本。缺点 2：如果不计算设备故障导致的工时损失的话，主动运行比响应运行成本更高。您所聚焦的部分依赖于系统宕机时的结果是。当 ICT 工具不工作时，会难于计算教学的损失。如果要求学生和教师的宕机时间很少，那么必须在高的运行时间上投资。

当我们在学校中扩展队伍时，这一定会冲击 ICT 联络人的工作资源，以及自治市为提供学校的 ICT 服务。

为了将需求量化，在我们的一些投资选项中估计了增加的资源需求：

投资	服务器	客户端/便携机的数量	用户：	在 2008 年规定的资源需求	今天的真实资源：
2005 年的今天的需求：	11	506	超过 3 000	4.6 FTEs（人年）	2.1 FTEs
2008 年的学生数量	29	1 033	超过 3 000	6.9 FTEs	

6.5.2.1 2008 年的教师

替换选项 1	280	266	4.3 FTEs	
替换选项 2（笔记本）	266	266	5.9 FTEs*	

- ）维护 266 台笔记本的另外的 FTEs

为学生和教师管理所有计算级的成本。关于学生和教师的瘦客户端和一些笔记本，我们从替换选项 1 出发。

年	***PCs*** 的数量	中心操作者	用于整个自治市的 ***ICT*** 指南	每所学校的 ***ICT*** 联络人（平均）	总计
2005	506	1/2 FTE	1/2 FTE	8.5 % FTE（3:30 小时每周）	2.1 FTEs
2005	用于运行的人力资源成本 *			**NOK 980 910**	
2008	1 333	1 个职位	1/2 FTE	100 % FTE（26 小时每周）	11.5 FTE
2008 年	用于运行的人力资源成本 *			**NOK 5 400 00.00**	

- ）每教师每个小时 NOK 270，一年 1 730 小时。每所学校的 ICT 联络人使用 75% 的时间用于教学支持。

替换选项 2 为每名教师配备笔记本：

年	***PCs*** 的数量	中心操作者	用于整个自治市的 ***ICT*** 指南	每所学校的 ***ICT*** 联络人（平均）	总计
2008	1 333	1 + 4/5 FTE*	1/2 FTE	100 % FTE（26 小时每周）	12.8 FTEs
2008 年	用于运行的人力资源成本 *			**NOK 6 000 000**	

- ）维护笔记本的另外的职位。

根据挪威和英国学校里进行的调查，学生和教师使用 Windows 与 Linux 的培训成本大致相同。这是因为培训与学校日常生活中终端用户程序的使用相关。

- 通常，在具有 300 名学生和教师的学校里，只有一两个人需要培训来操作计算机系统。这里提到的是学校的 ICT 联络人和自治市的操作者。

我们已经包括了用于 Linux 的另外的培训成本。当所有教师有一天体验一下 Linux 桌面选项时，对于那些认为能够用好 Windows 的人，转向新系统会更容易。像 LærerIKT 以及相似的显示成本没有包括进来，就像在这个调查中我们所做的自治市的成本概况一样。

6.6 选项的总结

为了实现每三名学生一台计算机并且每名教师一台 PC 的目标，推荐了选项 1。与当前 506 台客户端机器所覆盖的相比，这个选项需要超过 800 台另外的计算机。整体上，我们然后将得到超过 1 300 台客户端计算机，重点在瘦客户端和无盘客户端。一些机器还是便携的，为学校的日常生活提供另外的灵活性。

替换选项	超过 **3** 年的成本
选项 1：发展 800 台客户端	NOK 7 322 000
选项 2：为所有教师配备便携机 + 选项 1。	NOK 12 000 000

运行成本：

替换选项	年成本
替换选项 0. 管理 506 台客户端机器	NOK 1 000 000
选项 1. 增加 800 到 1 300 台 PCs	NOK 5 400 000
选项 2. 为所有教师配备便携机 *	NOK 6 000 000
选项 3. 便携机房布置 *	NOK 8 000 000

从当前选项 0 到选项 1 的运行花费的巨大增加是由于对学校 ICT 联络人的投资。这从 10% 增加到 100% FTE。今天 ICT 联络人进行的维护大约为 10% FTE。当客户端机器加倍时，这大概增加到 20%。在增加为的全职职位时，80% 用于支持学校科目中 ICT 工具的教育应用。这意味着学校的校长必须为此分配资源，从而能从 2006 年开始符合国家课程。

6.7 推荐

替换选项 1：

成本的类型	量
这件事是管理 1 300 台客户端计算机：	NOK 2 000 000
三年中的每年投资：	NOK 2 440 667
支持 ICT 工具的教育应用：	NOK 3 400 000
选项 1 的投资与运行的年度成本：	NOK 7 841 000

6.8 附件

很多学校已经为学校中的 ICT 应用发展了活动计划。这应该作为附件包括进来。

7 设立基础设施

7.1 网络架构

用户案例：应该设立这样规模的计算机网络，可以在本地运行系统，或者连接到集中运行解决方案上

7.1.1 解决方案

7.1.2 例外的处理

7.1.3 确认

7.1.4 更新配置数据库

7.2 服务器的配置

使用案例：如何为学校或自治市的很多学校，安装整个计算机网络的机器。

  {{attachment:bilder50.png}} 不同服务器上的不同配置。

7.2.1 组合服务器，作为组合的解决方案

主服务器和瘦客户端服务器组合在一起的两个配置被称为组合服务器

- {{attachment:bilder51.png}}

这是非常小的步骤，能够在主干网上容易地使用适当的交换机，并使用交叉电缆将防火墙连接到组合服务器上

注意：要了解，在瘦客户端网络如果主机名是 **printer00** 的话，将打印机的地址设置为 192.168.0.0/24 会不工作。确认编辑 KDE 打印管理器，在 192.168.0.0/24 -net 上查找打印机。而不是标准的 10.0.2.0/23-net。

7.2.2 Skolelinux/Debian-Edu 中配置的描述

在发源于 debian-edu-install 软件包的文件 `src/debian-edu-install/debian/debian-edu-install.templates` 的安装中显示的配置。

图形桌面

会越来越多地看到对图形桌面的参考。简单地说，这意味着当代的桌面，具有指示和点击、窗口、图标和文件夹。图形用户界面在 1973 年由施乐 PARC 首先做出来，早于在市场上可以买到计算机的 10 年以前。这是图形用户界面非常短暂的出现。

Skolelinux/Debian-edu 中的不同的设置的简要总结，以及它们如何组合

<ol style="list-style-type: decimal;"> 主服务器   警告：所有的 Skolelinux/Debian-edu-networks 必须具有且只有一个主服务器，并且只有一台机器具有那个配置。最普通的是，配置可以与瘦客户端服务器组合，或者只是个工作站。 每个 Skolelinux 网络需要一台，且只有一台机器运行“主服务器”。这台机器提供网络服务，例如在目录服务器（LDAP）等等的帮助下来登录。没有这台机器网络会不工作。因为这台机器存储所有的数据文件，它需要很多磁盘空间。通过安装这个配置您将没有图形用户界面。如果您想要图像用户界面，必须还要安装工作站配置，或瘦客户端服务器.   工作站

运行“工作站”配置的机器是我们所知道的一般 PCs。用户登录工作站，并且在主服务器上获得存储空间。文档、用户个人设置和很多网络服务在主服务器上。用户程序运行在工作站上。

为了访问 CD/DVD 播放机/刻录机、数字相机、扫描仪，这是要安装的配置。

1. 瘦客户端服务器

运行瘦客户端服务器的机器支持瘦客户端。这个配置还包括工作站配置。为了防止网络饱和，需要两台 NICs。配置：主服务器、工作站和瘦客户端服务器可以安装在相同机器上。

这个配置还包括工作站配置

1. 无盘工作站

运行瘦客户端服务器的机器如果允许的话支持无盘客户端。在 Skolelinux 2.0 中，这必须在后面允许。这个配置还包括工作站配置。配置主服务器、工作站和瘦客户端服务器可以安装在相同机器上。

- 这个配置还包括工作站配置
- 主服务器 + 瘦客户端服务器（包括工作站）

这个组合的配置，也被称为组合配置，提供了只在一台服务器上设置完整 Skolelinux / Debian-edu 网络 工作站 和 瘦客户端 的能力。在带有 10-15 台瘦客户端和一些工作站的小型 Skolelinux/Debian-edu 网络中，这是可接受的解决方案。对于更大的安装，通常必须选择 更大的服务器。

1. 主服务器 + 工作站

这个组合的配置主要为您提供带有 GUI 的主服务器。如果您不喜欢从命令行管理您的服务器的理念，这是个好的组合。

独立的配置并不是 Skolelinux/Debian-edu 网络的一部分。这个配置的目的是支持家用 PC 或便携机。

1. 单独的机器

单独机器的配置不能与主服务器、工作站或瘦客户端一起安装。

单独机器的配置最好不连接到 Skolelinux / Debian-edu 网络上使用。

Skolelinux / Debian-edu 中的所有程序包括在单独机器的配置中

7.2.3 解决方案

7.2.4 例外的处理

7.2.5 确认

7.2.6 更新配置数据库

7.3 硬件服务器

用户案例：应该配置什么

7.3.1 解决方案

7.3.2 例外的处理

7.3.3 确认

7.3.4 更新配置数据库

7.4 客户端计算机

用户案例：客户端机器的选择。您应该选择无声计算机还是用于多媒体的计算机。为所有人配备笔记本还是台式机。

几种类型的技术可以提供 PC 上的应用。最普通的是每台计算机上本地运行的胖客户端。但在台式机上还有其他类型的技术应用。很多人可能已经听说过图形终端。例子包括 Citrix、！FreeNX 和 Windows Terminal Server。还有其他的选项，像低脂客户端和真正瘦客户端。这篇文章描述了选项，并提供了各种终端技术在哪里使用最好的概况。写这篇文章的原因是分享在带有低、中或高容量网络的很多不同建筑中，集中运行计算机的企业解决方案的经验。

客户端技术以后面的顺序描述。图形终端 Citrix 和！freeNX，带有 X window 的瘦客户端，带有 Linux 和 Windows 的胖客户端，介于 Linux 之间的客户端，和笔记本。后面是各种基于业务的安装中最常使用的服务器系统。计算成本的关键因素是并发用户的数量和服务器的数量。在几所学校集中管理计算机设备，在实践中可以与较大公司中如何运行 ICT 系统相比较。通常学校比社会活动的其它部分具有更多的计算机。如果没有充分考虑在学校选择什么客户端解决方案，可以迅速导致自治市中 IT 服务部门的雇员数量加倍。

Citrix 是知道最多的 图形客户端产品。制造这个产品的公司成立于 1989 年。第一个图形客户端是为 OS/2 操作系统制造的。第一个 Windows 产品于 1995 年与 NT 3.51 一起发布。还有几个与 Citrix 竞争的产品。最成功的一个是 NX 技术公司的。简单地说，您可以从带有 Citrix 或 NX 的服务器上运行应用。屏幕经过网络，从服务器导出到胖客户端的图形终端上。

图形客户端具有很强的能力，无论客户端上运行什么操作系统，它都是无缝的。也可以在服务器上使用应用。可以在 64 kbps 的 ISDN 线路上运行标准办公程序和客户端电子邮件。也就是说，在图像软件中限制了是否使用多媒体或交互图像。如果自治市在 2-8 Mbps 的宽带上，分布着 5-6 所学校的 30 或 50 台图形终端，解决方案会迅速地变得不实际。无法以这个容量运行交互图形应用。网络会被通信堵塞，Citrix 客户端会从服务器机器上断开。

使用图形客户端，运行部门必须为软件的维护运行两条平行的路径。维护发生在所有客户端计算机上，以及本地和中心服务器上。为了使例如 Citrix 合理地运行良好，除了中心应用服务器，必须在每座建筑中部署另外的两台服务器。此外，为了使用多媒体，通常还需要一些胖客户端。例如在奥斯陆的学校中，1/3 的机器是胖客户端，来为多媒体提供支持。

瘦客户端于 1984 年在 MIT 引入。这大概是苹果公司发布 Macintosh GUI 的相同时期。微软在下一年将第一版的 MS-Windows 推向市场。实际上，瘦客户端被命名为 X window 系统，并且可以用在所有可能的平台上，如 Linux、Mac 或 Windows。X window 将事情翻转过来了。实际上，应用运行在服务器上，而 GUI 通过网络发送给客户端计算机。客户端计算机运行服务器程序来显示图像窗口。X server 可以运行来自很多不同服务器上不同程序的您自己的应用窗口。胖客户端还在 PC 上使用虚拟本地网络来运行 X window 系统。所有带有图形用户界面的 Unix 系统都运行 X servers。

瘦客户端的主要优点是再次使用较旧的设备，而不会增加运行的复杂性。很多人使用 233 MHz 和 32 MB 内存的 PC 作为瘦客户端。不需要本地硬盘驱动器。用户可以处理较重的图像、声音与简单视频。几所学校已经在 瘦客户端上开放使用 CD / DVD-Rom 和 U 盘。运行人员不必跟踪每台 PC 上的各自的操作系统。每一部分由服务器处理。在一般使用时，每台瘦客户端使用大约 2 Mbps 的网络容量。瘦客户端的显著地好于图形终端。与使用例如 Citrtix 的图形客户端相比，瘦客户端平均需要更少的服务器，如奥斯陆的教育部的研究所显示的。

胖客户端或标准 PCs 是今天最常使用的。个人计算机这个术语在 1962 年 11 月 3 日第一次使用。带有网络和图形用户界面的第一台 PC 在 1973 年由 Xerox PARK 创造。今天，PC 的概念由 IBM 在 1981 年发布，并被熟知并广泛接受。整个操作系统和所有软件应用都在当地的数据商店安装到每台客户端计算机上。最知名的操作系统是微软的 Windows 和 Linux。但还有很多人使用其它操作系统，包括 BSD 的版本。

胖客户端的优点是所有程序在本地运行，这可以为用户提供很大的灵活性与性能。由于多数用户程序在本地运行，那么只需要较少的中心服务器。使用胖客户端的解决方案如果标准化的话，可以相对不那么昂贵地运行。在 Windows 上，拥有最相似的机器是极大的优点，但这随着时间而变得困难。例如，学校有 4、5 种型号的 PC 是非常普通的。这会影响运行成本。Linux 更灵活，因为即使不同型号的 PC，系统也能更容易地管理。Linux 还需要更少的内存，并允许更老的计算机使用更长久，而不损失性能，如在英国教育通信与技术局（BECTA）的报告中所讲到的。

无盘客户端是另一种令人激动的技术。今天通过 Lessdisks 或新的 LTSP 在 Linux 上支持。15 年前，Novell 有个无盘客户端的虚拟垄断服务。简单地解释，整个操作系统和应用安装在服务器上。操作系统通过网络从服务器上传到客户端。文件、打印和 Web 服务由为网络设计的操作系统来处理。通过引入 Windows 95，Novell 遇到了技术障碍。微软更来了通过注册表而不是文本文件来控制 Windows 的方式。现在，只有 Linux 和其他 Unix 的变体提供无盘工作站。

无盘客户端的优点是，您以瘦客户端的操作优点来得到胖客户端的性能。这意味着组织可以将很多客户端机器连接到服务器上，而不用在每台客户端上本地安装操作系统。每一部分由服务器处理。系统支持音频、视频、CD/DVD-Rom 和 U 盘。今天已经不容易找到小于 800 MHz 处理器和 256 MB 内存的机器了，它们很适于用作半胖客户端。推荐使用本地硬盘缓存。

便携机本质上是胖客户端。笔记本在原理上可以用作瘦客户端、半胖客户端或图形终端。但有几个原因导致非常不实际。便携机应该用作胖客户端。为了将笔记本连接到固定的计算机网络上，必须选择使用哪种服务。

在有很多用户的无线网络中，使用便携机面临很大的挑战。无线网络具有有限的容量。便携机还受到粗暴对待，并且与固定设备的通常情况相比需要更频繁地更换。不应该在无线网络的笔记本上运行图形终端。当有很多用户时会迅速变得不稳定。带有 Linux 或 Windows 的胖客户端运行良好。相对于网络它们可以相对容易地进行身份验证。用户可以以安全可靠的方式访问文件目录、打印和其它网络服务。由几家供应商为学校提供笔记本，连接到运行 Debian Edu 的计算机网络上。

7.4.1 客户端类型的表格

主要解决方案	对多媒体的支持	特点
胖客户端 (Windows、Linux 或 Mac)	在客户端机器上带有足够强大的处理器和内存，为声音、图像和视频提供良好的支持。	所有用户应用安装在客户端机器上。用户程序运行在客户端机器上。客户端机器会是固定的或便携的。在网络上运行如电子邮件、文件存储、立案系统等等那样的多种服务。优点：需要极少的集中服务器。对多媒体的支持良好 缺点：需要在每台客户端机器上安装并维护所有软件
无盘工作站 (Linux。早期这是来自 Novell 的使用 Windows 3.X 的解决方案)	在客户端机器上给予足够强力的处理器和内存，对声音、图像和视频提供良好的支持。	所有用户应用安装在服务器上。用户程序运行在客户端机器上。客户端计算机通常是固定的。在网络上运行如电子邮件、文件存储、立案系统等等那样的多种服务。优点：与胖客户端相同的功能。需要较少的服务器。客户端机器上不需要安装软件。
瘦客户端 (X Window 系统)	在客户端机器上给予足够强力的处理器和内存，对声音、图像和视频提供合宜的支持。	所有用户程序和服务安装在服务器上。用户程序运行在服务器上。客户端计算机通常是固定的。在网络上运行如电子邮件、文件存储、立案系统等等那样的多种服务。优点：给予二手计算机以新生。客户端机器上不需要安装软件。缺点：与胖客户端和无盘客户端相比，需要更多的服务器。

图形终端（FreeNX、Citrix、RDP）	在客户端机器上给予了足够强力的处理器和内存，以及大容量网络，对图像合宜的支持。在中等容量网络上对交互图像的支持很弱或没有。	所有用户程序和服务安装在服务器上。带有图形界面的完全操作系统通常安装在客户端机器上。用户程序运行在服务器上。客户端计算机通常是固定的。提供如电子邮件、文件存储、立案系统等等那样的几种服务。优点：给予二手计算机以新生。缺点：必须在每台客户端机器上安装并维护操作系统。与真正瘦客户端相比，需要更多的服务器。与胖客户端和无盘客户端相比，需要更显著多的服务器。多媒体的性能差或没有支持。网络过载时终端断开。这会一小时发生几次。
笔记本	在客户端机器上带有足够强大的处理器和内存，为声音、图像和视频提供良好的支持。	优点：可以将 PC 带到任何合适的地方 缺点：必须在每台客户端机器上安装与维护操作系统。必须设立并维护服务，使它容易地连接或断开网络上的机器。便携式设备有相当大的损坏，并且寿命平均为 3 年；比台式机少 2-5 年。便携设备的管理很昂贵。

7.4.2 解决方案

7.4.3 例外的处理

7.4.4 确认

7.4.5 更新配置数据库

7.5 交换机

用户案例：应该配置什么

7.5.1 解决方案

7.5.2 例外的处理

7.5.3 确认

7.5.4 更新配置数据库

7.6 无线接入点

用户案例：应该配置什么

7.6.1 解决方案

7.6.2 例外的处理

7.6.3 确认

7.6.4 更新配置数据库

7.7 防火墙

用户案例：应该配置什么

7.7.1 解决方案

7.7.2 例外的处理

7.7.3 确认

7.7.4 更新配置数据库

7.8 路由器

用户案例：应该配置什么

7.8.1 解决方案

7.8.2 例外的处理

7.8.3 确认

7.8.4 更新配置数据库

7.9 设立简单的防火墙

用户案例：应该配置什么

7.9.1 解决方案

7.9.2 例外的处理

7.9.3 确认

7.9.4 更新配置数据库

7.10 设置：

用户案例：应该配置什么

7.10.1 解决方案

7.10.2 例外的处理

7.10.3 确认

7.10.4 更新配置数据库

8 实用命令

8.1 对 4 GB 内存的支持 <-- 已包含在配置管理中

使用案例：因为 Skolelinux/Debian-Edu CD 上有限的空间，只包括了一个 Linux 内核，也就是最低的共同点。这意味着包括其中内核 工作在尽可能多的不同类型硬件上。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

您可以用命令 **uname -a** 来发现运行的内核的类型。这个命令可以在后面使用来确保您已经升级到所需要的内核。然后，看起来会像这样：

```
tjener:~# uname  -a
Linux tjener.intern 2.6.8-2-386 #1 Thu May 19 17:40:50 JST 2005 i686 GNU/Linux
```

这里运行 386 内核，它应该运行在几乎所有 PCs 上。但对于双核处理器或多于 940 MB 内存，这不是最优的。

如果您想要新的内核用于带有大量内存和多个处理器的服务器，可以以后下载与安装。Debian 的软件包系统使您很容易做到。

关于 **apt-get** 和 **dpkg** 的更多具体描述，请查看 第 8.9 节。

smp 是查找支持多于 940 MB 内存和双处理器的 Linux 内核时的关键词。它是 *Symmetric Multi-Processors* (对称多处理器) 的缩写。从 shell 来运行命令，它列出准备好安装的处理器核心的数量：

apt-cache search kernel-image | grep smp

. 在撰写本文时，列出如下：

```
kernel-image-2.4-686-smp - Linux kernel image for version 2.4 on PPro/Celeron/PII/PIII/P4 ↩
    SMP
kernel-image-2.4-k7-smp - Linux kernel image for version 2.4 on AMD K7 SMP
kernel-image-2.4.27-2-686-smp - Linux kernel image for version 2.4.27 on PPro/Celeron/PII/ ↩
    PIII/P4 SMP
kernel-image-2.4.27-2-k7-smp - Linux kernel image for version 2.4.27 on AMD K7 SMP
kernel-image-2.6-686-smp - Linux kernel image for version 2.6 on PPro/Celeron/PII/PIII/P4 ↩
    SMP.
kernel-image-2.6-amd64-k8-smp - Linux kernel image for version 2.6 on AMD64 SMP systems
kernel-image-2.6-em64t-p4-smp - Linux kernel image for version 2.6 on Intel EM64T SMP ↩
    systems
kernel-image-2.6-k7-smp - Linux kernel image for version 2.6 on AMD K7 SMP.
kernel-image-2.6.8-11-amd64-k8-smp - Linux kernel image for version 2.6.8 on AMD64 SMP ↩
    systems
kernel-image-2.6.8-11-em64t-p4-smp - Linux kernel image for version 2.6.8 on Intel EM64T ↩
    SMP systems
kernel-image-2.6.8-2-686-smp - Linux kernel image for version 2.6.8 on PPro/Celeron/PII/ ↩
    PIII/P4 SMP.
kernel-image-2.6.8-2-k7-smp - Linux kernel image for version 2.6.8 on AMD K7 SMP.
```

不需要指定特定的内核版本，像 2.4.27 或 2.6.8。只需要使用 2.4 或 2.6。这归结为

```
kernel-image-2.4-686-smp - Linux kernel image for version 2.4 on PPro/Celeron/PII/PIII/P4 ↩
    SMP
kernel-image-2.4-k7-smp - Linux kernel image for version 2.4 on AMD K7 SMP
kernel-image-2.6-686-smp - Linux kernel image for version 2.6 on PPro/Celeron/PII/PIII/P4 ↩
    SMP.
kernel-image-2.6-amd64-k8-smp - Linux kernel image for version 2.6 on AMD64 SMP systems
kernel-image-2.6-em64t-p4-smp - Linux kernel image for version 2.6 on Intel EM64T SMP ↩
    systems
kernel-image-2.6-k7-smp - Linux kernel image for version 2.6 on AMD K7 SMP.
```

现在您只需要知道您有哪种处理器即可，例如 686 (Intel)、k7 (AMD) AMD64 或 EM64T

一旦内核与机器的匹配最佳，就可以使用命令来安装了

apt-get install kernel-image-2.6-< 您的处理器型号 >-smp

如果机器包含 Intel Xeon，那么可以使用

apt-get install kernel-image-2.6-686-smp

如果使用 2.4 内核

apt-get install kernel-image-2.4-< 您的处理器型号 >-smp

如果有 AMD Athlon(TM) MP 2000，可以使用

apt-get install kernel-image-2.6-k7-smp

当安装新的内核时，您会看到像这样的一些东西：

```
tjener:~# apt-get update
tjener:~# apt-get install kernel-image-2.6-686-smp
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  kernel-image-2.6.8-2-686-smp
Suggested packages:
  lilo kernel-doc-2.6.8 kernel-source-2.6.8
Recommended packages:
  irqbalance
The following NEW packages will be installed:
  kernel-image-2.6-686-smp kernel-image-2.6.8-2-686-smp
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 15.3MB of archives.
After unpacking 44.9MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://ftp.debian.org sarge/main kernel-image-2.6.8-2-686-smp 2.6.8-16 [15.3MB]
Get:2 http://ftp.debian.org sarge/main kernel-image-2.6-686-smp 101 [2154B]
Fetched 15.3MB in 1m13s (208kB/s)
Selecting previously deselected package kernel-image-2.6.8-2-686-smp.
(Reading database ... 80762 files and directories currently installed.)
Unpacking kernel-image-2.6.8-2-686-smp (from .../kernel-image-2.6.8-2-686-smp_2.6.8-16_i386 ↩
    .deb) ...
Selecting previously deselected package kernel-image-2.6-686-smp.
Unpacking kernel-image-2.6-686-smp (from .../kernel-image-2.6-686-smp_101_i386.deb) ...
Setting up kernel-image-2.6.8-2-686-smp (2.6.8-16) ...
File descriptor 3 left open
File descriptor 4 left open
File descriptor 5 left open
File descriptor 6 left open
File descriptor 7 left open
    Finding all volume groups
    Finding volume group "vg_data"
    Finding volume group "vg_system"
Searching for GRUB installation directory ... found: /boot/grub .
```

```
Testing for an existing GRUB menu.list file... found: /boot/grub/menu.lst .
Searching for splash image... none found, skipping...
Found kernel: /boot/vmlinuz-2.6.8-2-686-smp
Found kernel: /boot/vmlinuz-2.6.8-2-386
Updating /boot/grub/menu.lst ... done
Setting up kernel-image-2.6-686-smp (101) ...
```

如所显示的，您被要求安装 kernel-image-2.6-686-smp，并且自动翻译为安装 kernel-image-2.6.8-2-686-smp。还建议安装一些其他可能有用的软件包。

用命令：shutdown -r now 来重新启动机器

8.1.1 例外的处理

为了激活内核需要重新启动机器。

为 Skolelinux / Debian-Edu 机器构建内核，是唯一需要重新启动的时候。当安装其它程序时不需要重新启动。

8.1.2 确认

当安装后运行命令 **uname -a** 时，显示后面的内容

```
tjener:~# uname  -a
Linux tjener.intern 2.6.8-2-686-smp #1 SMP Thu May 19 17:27:55 JST 2005 i686 GNU/Linux
```

安装 smp 内核并且重新启动后，您可以运行命令 **free** 和 **cat /proc/cpuinfo**。然后可以看到新的内核是否使用了所有内存和两个处理器。

```
ltspserver00:~# free
             total       used       free     shared    buffers     cached
Mem:       4074752    4045556      29196          0     339248    2327780
-/+ buffers/cache:    1378528    2696224
Swap:      1835000       5852    1829148
```

这里是缩短的打印内容，删除了没必要的打印内容。

```
ltspserver00:~# cat /proc/cpuinfo
processor       : 0
vendor_id       : !GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 2.66GHz

processor       : 1
vendor_id       : !GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 2.66GHz

processor       : 2
vendor_id       : !GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 2.66GHz

processor       : 3
vendor_id       : !GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 2.66GHz
```

8.1.3 更新配置数据库

8.2 管理软件包（apt-get）

使用案例：安装新程序或更新程序

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

为了安装软件包，需要告知从哪里取回软件包。换句话说，使用哪个软件包档案。

可以在文件 `/etc/apt/sources.list` 中指定软件包档案

可以用命令行进行软件包管理。也有更多的图形应用像例如 KPackage 7 或 Webmin 12

这一节提供使用命令行来管理软件包的快速介绍。

这是在互联网上或来源于 CD ROM 的关于软件包仓库的文件内容：

```
#deb file:///cdrom/ sarge main local

deb cdrom:[Debian GNU/Linux edu _Sarge_ - Unofficial i386 Binary-1 (20050808)]/ unstable ↩
    contrib local main non-free

 1. deb http://security.debian.org/ stable/updates main contrib non-free
 1.deb http://security.debian.org/ sarge/updates main contrib non-free
   1. Use (by uncommenting) either http or ftp, NOT both
   1. http based apt source: ----------------
 1. deb http://ftp.debian.org/debian/ sarge main contrib non-free
 1. deb http://non-us.debian.org/debian-non-US/ sarge/non-US main contrib non-free
 1. deb http://ftp.skolelinux.no/skolelinux/ sarge local
   1. ftp based apt source: -----------------
 1. deb ftp://ftp.debian.org/debian/ sarge main contrib non-free
 1. deb ftp://non-us.debian.org/debian-non-US/ sarge/non-US main contrib non-free
 1. deb ftp://ftp.skolelinux.no/skolelinux/ sarge local
```

注意，不带有井号（#）的行可以用作软件包档案的参考。例子显示了在安装时从 CD ROM 得到软件包。其它档案没有激活。当进行时应该打开用于安全升级。您可以尝试更多软件包的其它档案。

作为开始，看起来应该像这样：

```
#deb file:///cdrom/ sarge main local

 1.deb cdrom:[Debian GNU/Linux edu _Sarge_ - Unofficial i386 Binary-1 (20050808)]/ unstable ↩
      contrib local main non-free

 1.deb http://security.debian.org/ stable/updates main contrib non-free
deb http://security.debian.org/ sarge/updates main contrib non-free
   1. Use (by uncommenting) either http or ftp, NOT both
   1. http based apt source: ----------------
deb http://ftp.debian.org/debian/ sarge main contrib non-free
deb http://non-us.debian.org/debian-non-US/ sarge/non-US main contrib non-free
deb http://ftp.skolelinux.no/skolelinux/ sarge local
   1. ftp based apt source: -----------------
 1. deb ftp://ftp.debian.org/debian/ sarge main contrib non-free
 1. deb ftp://non-us.debian.org/debian-non-US/ sarge/non-US main contrib non-free
 1. deb ftp://ftp.skolelinux.no/skolelinux/ sarge local
```

注意，在包含“deb: cdrom”的行前面有个 # 号。能够从互联网得到所有部分时，没必要从 CD-ROM 装入软件包。

如果将新的行添加到这个文件中，必须使用什么软件包是可用的信息，来更新数据库。

关于添加作为软件包源的其它行，请参见 第 13 章。

8.2.1 例外的处理

软件包档案的连接具有特定的形式。如果没有按照这个形式，那么会在升级时给出错误消息，要求改正错误。

注释符号 (#) 也在文件中的一些行的前面。“注释掉”的技术典型用于多数 Linux 中的配置文件中。可以使用的其他符号有分号 (;) 和双斜线 (//)。但这里强制使用井号，并且当删除时，行内所写的内容就可运行。

8.3 更新软件包档案

使用案例：用升级的程序概要来升级软件包仓库。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

可用软件包的选择被持续更新。最普通的是新的安全更新。新版本的软件也会发表出来。这样，必须更新软件包档案。这使用后面的命令来完成

```
tjener:~# apt-get update
Get:1 http://ftp.skolelinux.no sarge/local Packages [17.4kB]
Ign http://ftp.skolelinux.no sarge/local Release
Get:2 http://non-us.debian.org sarge/non-US/main Packages [20B]
Get:3 http://non-us.debian.org sarge/non-US/main Release [102B]
Get:4 http://non-us.debian.org sarge/non-US/contrib Packages [20B]
Get:5 http://non-us.debian.org sarge/non-US/contrib Release [105B]
Get:6 http://non-us.debian.org sarge/non-US/non-free Packages [20B]
Get:7 http://non-us.debian.org sarge/non-US/non-free Release [106B]
Get:8 http://ftp.debian.org sarge/main Packages [3347kB]
Get:9 http://security.debian.org sarge/updates/main Packages [155kB]
Get:10 http://security.debian.org sarge/updates/main Release [110B]
Get:11 http://security.debian.org sarge/updates/contrib Packages [538B]
Get:12 http://security.debian.org sarge/updates/contrib Release [113B]
Get:13 http://security.debian.org sarge/updates/non-free Packages [20B]
Get:14 http://security.debian.org sarge/updates/non-free Release [114B]
Get:15 http://ftp.debian.org sarge/main Release [95B]
Get:16 http://ftp.debian.org sarge/contrib Packages [56.2kB]
Get:17 http://ftp.debian.org sarge/contrib Release [98B]
Get:18 http://ftp.debian.org sarge/non-free Packages [58.4kB]
Get:19 http://ftp.debian.org sarge/non-free Release [99B]
Fetched 3635kB in 23s (157kB/s)
Reading Package Lists... Done
```

这个命令必须在升级 之前执行，或者在添加新软件包之前执行。

8.3.1 例外的处理

8.3.2 确认

8.4 更新到新的软件包

使用案例：将安装的软件包更新到较新的版本，如果可用的话

作者：Klaus Ade Johnstad

共同作者：Knut Yrvin

将所有安装的软件包更新到较新的版本，可以使用命令

apt-get upgrade

```
tjener:~# apt-get upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be upgraded:
  apache apache-common apache2-utils bsdutils cfengine cfengine-doc courier-authdaemon  ↩
      courier-base courier-imap courier-imap-ssl courier-ldap
  courier-ssl cpio debian-edu-config debian-edu-install education-common education-main- ↩
      server education-networked education-tasks libapr0 libice6
  libmysqlclient12 libpam-ldap libpcre3 libsensors3 libsm6 libsnmp-base libsnmp5 libssl0 ↩
      .9.7 libungif4g libx11-6 libxext6 libxft1 libxi6 libxmu6 libxmuu1
  libxp6 libxpm4 libxrandr2 libxt6 libxtrap6 libxtst6 localization-config lynx mount mysql- ↩
      common ntp ntp-refclock ntp-server ntpdate openssl python2.3
  slbackup snmp squid squid-common tcpdump util-linux xdebconfigurator xfree86-common xlibs ↩
       xlibs-data
62 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 23.7MB of archives.
After unpacking 225kB disk space will be freed.
Do you want to continue? [Y/n]
```

只需要按 **Enter** 键，或者'Y' 和 **Enter** 键。软件包会自动下载并安装。当升级开始时会得到更改日志。

一旦完成升级，您就可以删除 /var/cache/apt/archives/ 目录中下载的软件包了。使用命令

apt-get clean

来删除档案。这个命令应该不时地执行。否则 /var 会充满。

8.4.1 警告

有时在升级 之前看看将发生什么是可以的。判断是否有必要下载几个巨大的软件包，因为可能您需要等待直到有更大的带宽可用的时候。如果运行

apt-get upgrade --simulate

如果模拟什么将发生，而不是真实发生。屏幕上是否有太多信息，可以运行

apt-get upgrade --simulate | more

如果看起来不错，可以再次运行不带有 **--simulate** 参数的命令

还能够使用 **aptitude dist-upgrade**，与 **apt-get upgrade** 结合。

8.4.2 例外的处理

有时您会得到消息，更改会影响升级或安装的软件包，如这里

```
kdeaddons (4:3.1.0-4) unstable; urgency=low

  * Rebuilt against libvorbis0a (closes: #184713).
  * Removed alpha compile flags.
  * Fresh admin/ sync.

 -- Ben Burton <bab@debian.org>  Sun, 16 Mar 2003 16:00:19 +1100

kdeaddons (4:3.1.0-2) unstable; urgency=low

  * First KDE3 upload to debian!
  * Applied Ewald Snel's patch for xine support.
  * Rolled the epoch to aid upgrades from the unofficial repository on
    ftp.kde.org.. *sigh*
```

使用键盘上的 空格键来浏览消息。然后您会看到

```
quanta (1:3.0pr1-1) unstable; urgency=low

  * New upstream release.
  * Built for KDE3.

 -- Ben Burton <benb@acm.org>  Wed,  4 Sep 2002 10:36:12 +1000

(END)
```

按 **q** 键来退出，并且您得到

```
Fetched 60.2MB in 11m24s (87.9kB/s)
Reading changelogs... Done
apt-listchanges: Do you want to continue? [Y/n]?
```

要继续的话您需要按 **Y** 键来同意。

8.4.3 确认

8.5 安装的软件包的概述

使用案例：想要安装的软件包的概述

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

为了得到安装的软件包的概述，运行这个命令

dpkg --list | more

注意当列表中的第一组字母为“ii”时，意味着软件包被完全安装。

为了得到一个特定的软件包的状态，可以使用 **grep** 来搜索它：

```
tjener:~# dpkg --list | grep apache
ii  apache          1.3.33-6       versatile, high-performance HTTP server
ii  apache-common   1.3.33-6       support files for all Apache webservers
ii  apache2-utils   2.0.54-4       utility programs for webservers
```

8.6 找到特定软件包的名称

使用案例：经常很难记住软件包的名称。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

为了找到特定软件包，可以使用这个命令来搜索词语：

apt-cache search < 软件包名称 >

如果屏幕上的文本太多的话，尝试这个

apt-cache search < 软件包名称 >**|more**

这两个符号 < 和 > 必须 不使用。它们只用在这个例子中。

```
tjener:~# apt-cache search apache
apache - versatile, high-performance HTTP server
apache-common - support files for all Apache webservers
apache-dbg - debug versions of the Apache webservers
apache-dev - development kit for the Apache webserver
```

```
apache-doc - documentation for the Apache webserver
apache-perl - versatile, high-performance HTTP server with Perl support
apache-ssl - versatile, high-performance HTTP server with SSL support
apache-utils - utility programs for webservers (transitional package)
```

如下载的屏幕内容显示的，相对于已经安装的软件包，有很多与 Apache 相关的东西。

8.7 显示软件包的可用信息

用户案例：想要得到关于软件包的信息。可能有对其它软件包的依赖性等等……

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

命令

apt-cache showpkg < 软件包名称 >

和

<**apt-cache policy** < 软件包名称 >

给出关于软件包的细节。

```
tjener:~# apt-cache showpkg kdissert
Package: kdissert
Versions:
0.3.8-1(/var/lib/apt/lists/ftp.debian.org_debian_dists_sarge_main_binary-i386_Packages)

Reverse Depends:
Dependencies:
0.3.8-1 - kdelibs4 (2 4:3.3.2-4.0.2) libc6 (2 2.3.2.ds1-4) libgcc1 (2 1:3.4.1-3) libqt3c102 ↩
    -mt (2 3:3.3.3) libstdc++5 (2 1:3.3.4-1)
Provides:
0.3.8-1 -
Reverse Provides:
tjener:~# apt-cache policy  kdissert
kdissert:
  Installed: (none)
  Candidate: 0.3.8-1
  Version Table:
     0.3.8-1 0
        500 http://ftp.debian.org sarge/main Packages
```

所以，注意到软件包 kdissert 没有安装，但可以从 `http://ftp.debian.org sarge/main` 得到来安装 0.3.8-1 版本

8.8 软件包的安装

使用案例：想要安装程序或程序软件包。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

当找到要安装的软件包时，运行命令

apt-get install < 软件包名称 >

如果想要在安装过程中看到发生了什么，您可以首先使用命令来运行模拟

apt-get install < 软件包名称 > **--simulate**

```
tjener:~# apt-get install  aterm --simulate
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  aterm
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Inst aterm (0.4.2-11 Debian:3.1r0/stable)
Conf aterm (0.4.2-11 Debian:3.1r0/stable)
tjener:~# apt-get install  aterm
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  aterm
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 91.6kB of archives.
After unpacking 287kB of additional disk space will be used.
Get:1 http://ftp.debian.org sarge/main aterm 0.4.2-11 [91.6kB]
Fetched 91.6kB in 1s (71.0kB/s)
Selecting previously deselected package aterm.
(Reading database ... 32924 files and directories currently installed.)
Unpacking aterm (from .../aterm_0.4.2-11_i386.deb) ...
Setting up aterm (0.4.2-11) ...
```

8.9 将安装的软件包删除

使用案例：想要删除不使用的某些软件包。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

为了找到需要删除的特定软件包，使用上面列出的命令。

当您找到软件包的名称，运行命令

apt-get remove < 软件包名称 >

当删除软件包时如果想要看到发生什么，您可以用命令来模拟

apt-get remove < 软件包名称 > --**simulate**

```
tjener:~# apt-get remove  aterm --simulate
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  aterm
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
Remv aterm (0.4.2-11 Debian:3.1r0/stable)
tjener:~# apt-get remove  aterm
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  aterm
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
Need to get 0B of archives.
After unpacking 287kB disk space will be freed.
Do you want to continue? [Y/n]
(Reading database ... 32936 files and directories currently installed.)
Removing aterm ...
```

8.10 安装特定的软件包版本

用户案例：想要特定版本的软件包。可以是例如之前发行版本的程序。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

当用命令安装软件包时

apt-get install < 软件包名称 >

那么安装最新的软件包。有时想要较旧的版本而不是最新的版本。

apt-get install < 软件包名称 >= 旧版本号

为了得到较旧版本的 Webmin 备份模块，可以运行

apt-cache showpkg webmin-slbackup

来得到可用版本的概况

```
tjener:~#  apt-cache policy webmin-slbackup
webmin-slbackup:
  Installed: 0.0.10-1
  Candidate: 0.0.10-1
  Version Table:
 *** 0.0.10-1 0
        500 http://ftp.skolelinux.no sarge/local Packages
        100 /var/lib/dpkg/status
     0.0.9-1 0
        500 http://ftp.debian.org sarge/main Packages
```

这里可以看到，有两个版本可用。0.0.9-1 和 0.0.10-1

如果想要 0.0.9-1 版本的程序，可以使用后面的命令安装

apt-get install webmin-slbackup=0.0.9-1

```
tjener:~# apt-get install webmin-slbackup=0.0.9-1 --simulate
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be DOWNGRADED:
  webmin-slbackup
0 upgraded, 0 newly installed, 1 downgraded, 0 to remove and 0 not upgraded.
Inst webmin-slbackup [0.0.10-1] (0.0.9-1 Debian:3.1r0/stable)
Conf webmin-slbackup (0.0.9-1 Debian:3.1r0/stable)
tjener:~# apt-get install webmin-slbackup=0.0.9-1
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be DOWNGRADED:
  webmin-slbackup
0 upgraded, 0 newly installed, 1 downgraded, 0 to remove and 0 not upgraded.
Need to get 22.0kB of archives.
After unpacking 131kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://ftp.debian.org sarge/main webmin-slbackup 0.0.9-1 [22.0kB]
Fetched 22.0kB in 0s (23.6kB/s)
dpkg - warning: downgrading webmin-slbackup from 0.0.10-1 to 0.0.9-1.
(Reading database ... 32924 files and directories currently installed.)
Preparing to replace webmin-slbackup 0.0.10-1 (using .../webmin-slbackup_0.0.9-1_all.deb)  ↩
    ...
Unpacking replacement webmin-slbackup ...
Setting up webmin-slbackup (0.0.9-1) ...
```

8.11 使用 dpkg 来安装软件包

用户案例：有时需要从其他地方下载软件包，而不是位于 Debian web 档案中。Opera 浏览器就是这样的软件包。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

从程序创立者的主页来下载软件包。这可以是例如 Opera。使用后面的命令安装程序：

dpkg -i < 软件包的全路径 >

. 如果您想要模拟这个，尝试

dpkg --no-act -i < 软件包的完全路径 >

```
tjener:~# dpkg --install --no-act opera_8.51-20051114.5-sharedqt_en_sarge_i386.deb
Selecting previously deselected package opera.
(Reading database ... 32924 files and directories currently installed.)
Unpacking opera (from opera_8.51-20051114.5-shared-qt_en_sarge_i386.deb) ...
tjener:~# dpkg --install  opera_8.51-20051114.5-shared-qt_en_sarge_i386.deb
Selecting previously deselected package opera.
(Reading database ... 32924 files and directories currently installed.)
Unpacking opera (from opera_8.51-20051114.5-shared-qt_en_sarge_i386.deb) ...
dpkg: dependency problems prevent configuration of opera:
 opera depends on libqt3c102-mt; however:
  Package libqt3c102-mt is not installed.
dpkg: error processing opera (--install):
 dependency problems - leaving unconfigured
Errors were encountered while processing:
 opera
```

dpkg 比 apt-get 需要更多的操作，因为它不处理软件包的依赖性。这意味着您需要在后面立即运行带有更多参数的 **apt-get** 命令。例如，运行 **apt-get --fix-broken** 来帮助收拾

```
tjener:~# apt-get install --fix-broken --simulate
Reading Package Lists... Done
Building Dependency Tree... Done
Correcting dependencies... Done
The following extra packages will be installed:
  libaudio2 liblcms1 libmng1 libqt3c102-mt libxcursor1 libxft2
Suggested packages:
  nas liblcms-utils libqt3c102-mt-psql libqt3c102-mt-mysql libqt3c102-mt-odbc
The following NEW packages will be installed:
  libaudio2 liblcms1 libmng1 libqt3c102-mt libxcursor1 libxft2
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
1 not fully installed or removed.
Inst libaudio2 (1.7-2 Debian:3.1r0/stable) [opera ]
Inst liblcms1 (1.13-1 Debian:3.1r0/stable) [opera ]
Inst libmng1 (1.0.8-1 Debian:3.1r0/stable) [opera ]
Inst libxcursor1 (1.1.3-1 Debian:3.1r0/stable) [opera ]
Inst libxft2 (2.1.7-1 Debian:3.1r0/stable) [opera ]
Inst libqt3c102-mt (3:3.3.4-3 Debian:3.1r0/stable)
Conf libaudio2 (1.7-2 Debian:3.1r0/stable)
Conf liblcms1 (1.13-1 Debian:3.1r0/stable)
Conf libmng1 (1.0.8-1 Debian:3.1r0/stable)
Conf libxcursor1 (1.1.3-1 Debian:3.1r0/stable)
Conf libxft2 (2.1.7-1 Debian:3.1r0/stable)
Conf libqt3c102-mt (3:3.3.4-3 Debian:3.1r0/stable)
Conf opera (8.51-20051114.5 )
tjener:~# apt-get install --fix-broken
Reading Package Lists... Done
Building Dependency Tree... Done
Correcting dependencies... Done
```

```
The following extra packages will be installed:
  libaudio2 liblcms1 libmng1 libqt3c102-mt libxcursor1 libxft2
Suggested packages:
  nas liblcms-utils libqt3c102-mt-psql libqt3c102-mt-mysql libqt3c102-mt-odbc
The following NEW packages will be installed:
  libaudio2 liblcms1 libmng1 libqt3c102-mt libxcursor1 libxft2
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
1 not fully installed or removed.
Need to get 3489kB of archives.
After unpacking 8753kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://ftp.debian.org sarge/main libaudio2 1.7-2 [71.5kB]
Get:2 http://ftp.debian.org sarge/main liblcms1 1.13-1 [123kB]
Get:3 http://ftp.debian.org sarge/main libmng1 1.0.8-1 [171kB]
Get:4 http://ftp.debian.org sarge/main libxcursor1 1.1.3-1 [23.7kB]
Get:5 http://ftp.debian.org sarge/main libxft2 2.1.7-1 [54.4kB]
Get:6 http://ftp.debian.org sarge/main libqt3c102-mt 3:3.3.4-3 [3045kB]
Fetched 3489kB in 16s (212kB/s)
Selecting previously deselected package libaudio2.
(Reading database ... 33027 files and directories currently installed.)
Unpacking libaudio2 (from .../libaudio2_1.7-2_i386.deb) ...
Selecting previously deselected package liblcms1.
Unpacking liblcms1 (from .../liblcms1_1.13-1_i386.deb) ...
Selecting previously deselected package libmng1.
Unpacking libmng1 (from .../libmng1_1.0.8-1_i386.deb) ...
Selecting previously deselected package libxcursor1.
Unpacking libxcursor1 (from .../libxcursor1_1.1.3-1_i386.deb) ...
Selecting previously deselected package libxft2.
Unpacking libxft2 (from .../libxft2_2.1.7-1_i386.deb) ...
Selecting previously deselected package libqt3c102-mt.
Unpacking libqt3c102-mt (from .../libqt3c102-mt_3%3a3.3.4-3_i386.deb) ...
Setting up libaudio2 (1.7-2) ...

Setting up liblcms1 (1.13-1) ...

Setting up libmng1 (1.0.8-1) ...

Setting up libxcursor1 (1.1.3-1) ...

Setting up libxft2 (2.1.7-1) ...

Setting up libqt3c102-mt (3.3.4-3) ...

Setting up opera (8.51-20051114.5) ...
```

通过本章较早给出的不同命令，现在我们可以确认 Opera 已经安装了

```
tjener:~# apt-cache policy opera
opera:
  Installed: 8.51-20051114.5
  Candidate: 8.51-20051114.5
  Version Table:
 *** 8.51-20051114.5 0
        100 /var/lib/dpkg/status
tjener:~# dpkg --list|grep opera
```

ii opera 8.51-20051114. The Opera Web Browser

8.12 搜索软件包中的文件

使用案例：想要在软件包中找到程序名称或文件

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

您可以用这个命令得到概况

dpkg --listfiles < 软件包名称 >

```
tjener:~# dpkg --listfiles opera
/usr/bin
/usr/bin/opera
.
.
.
/etc
/etc/opera6rc
/etc/opera6rc.fixed
```

8.13 找到文件来自哪个软件包

用户案例：想要找到软件来自哪个软件包。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

dpkg --search < 文件名 >

可以看起来像这样

```
tjener:~# dpkg --search /etc/opera6rc.fixed
opera: /etc/opera6rc.fixed
```

8.14 从软件包中拆包而不安装软件包

使用案例：可能重要的系统文件被意外删除，但没有备份。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

当使用命令

dpkg --search < 文件名 >

警告：永远不要在根目录将软件包拆开

如果找到文件来自哪个软件包。可以提取软件包来找回系统文件，像后面显示的。

首先您需要取回那个 deb 软件包。可以通过将其放置在 /tmp 中来实现。使用这个命令将文件展开在这个目录中

dpkg --vextract < 软件包名称 > **/tmp**

. 然后所需的目录会新建在 /tmp 中，并且文件也放置在哪里。

dpkg --vextract < 软件包名称 > **/tmp**

8.15 制作您自己的软件包镜像

使用案例：一些软件包被经常安装。对于其它的，避免从互联网下载它们是有用的。

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

—**apt-get** 命令可以容易地从互联网上安装软件包。但当程序从互联网的 Debian 档案下载时，会使用相当大的网络容量。因为这个原因，可以告诉 **apt-get** 来使用来自本地软件包仓库的软件包。这种方式能够通过使用 **apt-get** 来简单地安装已经下载的软件包。这提供了 快速安装。

mkdir /var/www/dpkg

cp /var/cache/apt/archives/*.deb /var/www/dpkg

cd /var/www/

dpkg-scanpackages dpkg /dev/null | gzip -9c > dpkg/Packages.gz

此后，将新的行添加到文件 `/etc/apt/sources.list` 中：

```
deb file:///var/www dpkg/
```

然后必须像通常一样执行 **apt-get update** 命令来更新数据库中的软件包。

8.16 安全登录防火墙（ssh）

使用案例：有时当 web 浏览器不可用时，需要登录 Coyote Linux。可能最好选用命令行？然后可以使用 ssh 来连接到 Coyote Linux 上。

如果登录到 Skolelinux / Debian Edu 网络上的机器中，您可以使用

ssh -l root 10.0.2.1

来登录到 Coyote Linux 上

如果您在 Skolelinux / Debian Edu 网络外，值 10.0.2.1 可以用网卡在 WAN 中的适当值来代替。在这种情况下，可能会是 **ssh -l root 192.168.1.10**

这里您会遇到登录 Coyote Linux web 管理时出现的相同选项。这用基于菜单的文本来显示。

```
               Coyote Linux Gateway -- Configuration Menu


  1) Edit main configuration file          2) Change system password
  3) Edit rc.local script file             4) Custom firewall rules file
  5) Edit firewall configuration           6) Edit port forward configuration

  c) Show running configuration            f) Reload firewall
  r) Reboot system                         w) Write configuration to disk

  q) quit                                  e) Exit
  ----------------------------------------------------------------------------
  Selection:
```

选项与登录 Coyote Linux 进行 web 管理时提供的选项大致相同。菜单选择的快速描述请参见第 3.7 节。

当选择 **q) quit** 时，您将结束在 Coyote Linux 的命令行，如果您需要回到 Coyote Linux 的主菜单，请输入 **menu** 并按 **Enter** 键。

当试图登录 Coyote Linux 时您看到这个

```
klaus@tjener:~$ ssh 10.0.2.1 -l root
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
34:b7:a3:9b:06:4c:e2:30:1b:0d:03:45:7b:22:b7:dd.
Please contact your system administrator.
```

```
Add correct host key in /skole/tjener/home0/klaus/.ssh/known_hosts to get rid of this  ↩
    message.
Offending key in /skole/tjener/home0/klaus/.ssh/known_hosts:27
RSA host key for 10.0.2.1 has changed and you have requested strict checking.
Host key verification failed.
```

这很可能是因为之前用 IP 地址 10.0.2.1 登录到其它机器上，或者因为 Coyote Linux 的网卡已经被更换。还可能是来自中间未知的人的攻击。解决方案是删除密钥，在这个情况下是 `/skole/tjener/home0/klaus/.ssh/known_hosts` 文件中的第 27 行。

8.16.1 例外的处理

8.16.2 确认

8.16.3 更新配置数据库

8.17 防火墙（Coyote）状态的概况

使用案例：哪个命令可以用于得到菜单或得到防火墙状态的概况？

主要作者：Klaus Ade Johnstad

Coyote Linux 中有用的命令

- ping

用于找出网络是否工作。命令检查是否连接到 Skolelinux / Debian Edu 主服务器。

```
coyote# ping -c5 10.0.2.2
PING 10.0.2.2 (10.0.2.2): 56 data bytes
64 bytes from 10.0.2.2: icmp_seq=0 ttl=64 time=0.9 ms
64 bytes from 10.0.2.2: icmp_seq=1 ttl=64 time=0.5 ms
```

- uptime

这个命令给出自从 Coyote Linux 重新启动后的时间。

```
   coyote# uptime\n  2:37pm  up 80 days,  7:55, load average: 0.00, 0.00, 0.00
```

- dmesg

这个命令显示机器上运行的 Linux 内核的信息。它列出了像内存、处理器和网卡这样的内容。如果 **dmesg** 有太多输出，您可以将输出通过所谓的分页程序，像“more”来发送，并使用 **Space** 键来读取每一部分，**dmesg|more**

- ifconfig

像是网卡的额外信息。

```
coyote# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:FC:F8:D2:44
          inet addr:10.0.2.1  Bcast:10.0.3.255  Mask:255.255.254.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:314723 errors:0 dropped:0 overruns:0 frame:0
          TX packets:312105 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:53700845 (51.2 !MiB)  TX bytes:277496136 (264.6 !MiB)
          Interrupt:11 Base address:0x7000
```

```
eth1      Link encap:Ethernet  HWaddr 00:E0:18:A8:B1:BA
          inet addr:192.168.100.133  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:307395 errors:0 dropped:0 overruns:0 frame:0
          TX packets:281202 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:272404311 (259.7 !MiB)  TX bytes:47880640 (45.6 !MiB)
          Interrupt:10 Base address:0xb800 Memory:e3000000-e3000038

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:14565 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14565 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1290756 (1.2 !MiB)  TX bytes:1290756 (1.2 !MiB)
```

- lsmod

这个命令列出驱动模块。这用于查看网卡使用了哪个模块。

```
coyote# lsmod
Module                  Size  Used by
eepro100               17516   1
3c59x                  24408   1
mii                     1852   0 [eepro100]
ip_nat_quake3           1608   0 (unused)
ip_nat_mms              2448   0 (unused)
ip_nat_h323             2044   0 (unused)
ip_nat_amanda           1020   0 (unused)
```

这个列表列出了装入的网卡驱动模块。对于 Intel，pro100 模块被命名为 eepro100，而对于 3Com，模块被命名为 3c59x (对于型号名称 3c590, 3c595, 3c900, 3c905 都是合法的)。请参见 第 3.12 节

- route
- traceroute

用于指出互联网数据包移动去哪里。如果有问题，查看互联网数据包使用的路径是有用的。

- showcfg

给出网卡状态信息的另一个命令。

```
Coyote running configuration display utility.

Internet    (eth1): UP
LAN network (eth0): UP

-------------Internet configuration--------------
IP Address    192.168.100.133 (Static)
Netmask       255.255.255.0
Gateway       192.168.100.2
----------------LAN configuration----------------
IP Address    10.0.2.1
Netmask       255.255.254.0
Broadcast     10.0.3.255
----------------DNS configuration----------------
```

```
domain localdomain
nameserver 213.184.200.1
nameserver 213.184.200.2
-------------------------------------------------
10:51am up 7 days, 20:53, load average: 0.00, 0.00, 0.00

Press enter to return to system menu.
```

- free

这个命令用于查看有多少内存可用，并且使用了多少。这台机器有 32 MB 内存。

```
coyote# free
             total         used         free       shared      buffers
  Mem:        30860         6004        24856            0            0
 Swap:            0            0            0
Total:        30860         6004        24856
```

- menu

这个命令启动 Coyote Linux 菜单

```
               Coyote Linux Gateway -- Configuration Menu


  1) Edit main configuration file         2) Change system password
  3) Edit rc.local script file            4) Custom firewall rules file
  5) Edit firewall configuration          6) Edit port forward configuration

  c) Show running configuration           f) Reload firewall
  r) Reboot system                        w) Write configuration to disk
```

- reboot

coyote#reboot

这个命令使 Coyote Linux 重新启动

- shutdown

coyote#halt

这里关闭 Coyote Linux

8.18 下一个

使用案例：

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

8.18.1 例外的处理

8.18.2 确认

8.18.3 更新配置数据库

8.19 最后

使用案例：

作者：Klaus Ade Johnstad。

共同作者：Knut Yrvin

8.19.1 例外的处理

8.19.2 确认

8.19.3 更新配置数据库

9 版权与作者

这个文档由 Knut Yrvin (2006)、Andreas Johansen (2006)、Klaus Ade Johnstad (2006)、Halvor Dahl (2006)、Snorre Løvås (2006)、Finn-Arne Johansen (2006)、Ragnar Wissløff (2006)、Petter Reinholdtsen (2015)、Ole-Erik Yrvin (2015) 和 Ingrid Yrvin (2015) 撰写并被版权保护，并且文档在 GPL3 任何更新的版本下发布。源文档最初用挪威书面语来撰写，并在 2015 年翻译为英语。

如果您将内容添加上面，请只在您是作者的情况下去做。您需要以相同的条件来发布！然后将您的姓名添加到这里，并在“GPL v3 或任何更新的版本”许可下发布。

10 附件 A ——运行 Debian Edu / Skolelinux 的合同

合同号：.....

客户编号：.....

10.1 运行 DEBIAN EDU / SKOLELINUX 的合同

合同双方为

Driftselskapet AS, Maskinrommet 1, 0313 奥斯陆

组织编码：989 313 313

(此后被称为经销商)

和

NN

组织编码：

(此后被称为客户)

合同方按照后续合同条款，就运行服务的交付达成协议 (此后被称为“协议”)。后面的附件是协议的组成部分：

- 附件 1 ——定义

- 附件 2 ——客户的责任
- 附件 3 ——经销商的责任
- 附件 4 ——价格和支付条件
- 附件 5 ——总条款
- 附件 6 ——代理人

协议自签订之日和交付日期的至少 12 个月内有效。除非合同的一方没有在合同到期前的三个月以书面形式通告废除，协议将自动延期 12 个月。

合同一式 - 2 - 份，每方持有 - 1 - 份。

地址：.....

日期: 2006

经销商：.....

客户：.....

10.1.1 附件 1 ——定义

术语	描述
运行周期	从交付日到协议终止生效之日，而不论原因是什么。
提供的服务	在运行周期来自经销商的服务。提供的服务在附件 3 中进一步描述。
ICT 经理	在客户那里作为供应商联络人的胜任人员。
交付日期	客户可以使用所提供的服务的日期。
Skolelinux	在 Debian Linux 上构建并为在挪威学校里使用而调整的 Linux 发行版。

10.1.2 附件 2 ——客户的责任

10.1.2.1 1. ICT 技能的需求

ICT 管理员（1-3 名在客户处的指定人员）处理来自用户的有关使用 Skolelinux/Debian Edu 中包括的应用的请求。ICT 管理员应该具有充分的专长，能够胜任评价问题是否与系统的使用或运行相关。

ICT 管理员应该通过电话或电子邮件联络供应商的用户支持中心。客户的用户应该不直接联系供应商。

10.1.2.2 2. 机器需求

客户已经安装并测试出设备在交付日之前满意地运行。

10.1.2.3 3. 程序需求

客户应该在交付之前，已经安装 Skolelinux / Debian Edu，而获得确认的、功能满意的安装。

10.1.2.4 4. 通讯需求

客户应该在交付日前，已经安装并配置了到互联网的通信，并测试其工作满意。为了能够提供服务，客户必须为合同方安排能够通过互联网访问客户的 ICT 设施。

10.1.2.5 5. 来自经销商的通知

当满足上述所有要求时，客户应该以书面或通过电子邮件的方式通知合同方，ICT 系统已经准备好由合同方来提供服务了。

系统所有用户的列表包括全名、用户名、想要的密码，列表最晚应该与本消息一起通过电子方式发送给经销商。

10.1.3 附件 3 ——经销商的责任

10.1.3.1 1. 交付日期要求

供应商在收到来自客户与附件 2 第 5 段一致的通知后，应该尽快对客户安排接收提供的而服务。交付日期应该不晚于供应商收到通知的 4 周时间。

10.1.3.2 2. 提供给客户的信息

当所有上面的要求满足时，合同方应该以书面或通过电子邮件通知客户，ICT 系统已经为客户准备好来接收提供的服务了。

10.1.3.3 3. 服务要求

后面的表格显示了与运行 Skolelinux/!DebianEdu 所有的相关服务。表格中的打叉符号显示了供应商与客户之间对不同服务的责任：

交付的（包括的）由供应商执行，并且包括在协议价格中，交付的（运行的）由供应商执行，根据第 7 章的价格记录在客户的账户中。客户，由供应商使用客户的费用来执行。

服务	交付的（包括的）	交付的（运行的）	客户
检修和通过电话或电子邮件的用户支持	x		
参加用户论坛	x		
更换硬件[1]	x		
添加、更改、删除用户[2]		(x)	x
当忘记密码时更改密码		(x)	x
Skolelinux 上的安全更新	x		
Skolelinux 上的版本更新	x		
更改用户权限		(x)	x
监控磁盘的填充	x		
监控相关部件的寿命	x		
扩展磁盘分区	x		
运行并监控防火墙	x		
运行并监控网络	x		
应 ICT 管理员的请求删除队列中堆积的打印工作	x		
监控并确保备份了复件	x		
在 ICT 管理员的请求下删除数据		x	
更换备份媒体并存储备份复件		x	
在 ICT 管理员的请求下恢复安全备份。		x	
设立新的打印机和打印机队列		(x)	x
在 ICT 管理员的请求下停止并重新启动打印机队列	x		

将服务器上由于应用错误导致的挂起的进程停止	×	

10.1.3.4 4. 回应时间要求

供应商应该不过度延误，来启动检修并解决问题。ICT 管理员应该不断地将错误更改的状态与进度进行更新。

10.1.3.5 5. 技能要求

供应商在任何时候都应该有充足的资源和相关的专长，以专业的方式提供服务

10.1.4 附件 4 ——价格和支付条件

10.1.4.1 1. 提供服务的补偿

提供服务的补偿根据网络上工作站的数量计算。协议包括至少 60 台工作站。客户应该每年为每台工作站支付供应商 £78，不含 VAT，以补偿提供的服务，也就是 60 台工作站每月 £390，不含 VAT。

如果工作站的数量更改，客户应该提供书面通知，包括更改的相应日期。以可能的重新计算对记账基础的调整，将包括在下一份发票中

10.1.4.2 2. 咨询支持

咨询的小时费用为 NOK 800 （65 £）不含 Moms （VAT）。进行记账的所有工作在工作开始前应该由用户同意。记录的差旅费向客户收取。差旅时间的补偿用经历的时间以 NOK 400 不含 Moms 的小时费率来计算。

10.1.4.3 3. 支付条款

提供服务的补偿在每个季度前记账。对于第一季度，记账从交付日期开始，并且运行直到当前季度结束。

咨询补偿根据达成的与进行的工作作为后支付来记账。

所有的发票在 30 天的期限内完成。

10.1.4.4 4. 价格规定

价格会随着国家消费者价格指数（SSB CPI）的增加而每年调整。这会在签订协议后一年首次发生。

10.1.5 附件 5 ——总条款

10.1.5.1 1. 合同方的合作与责任

一般

合同方应该合作使协议最有效地实施。合同双方应该，以书面形式，使用五个工作日的通知彼此召集会议，来讨论与协议实施相联系而出现的事宜。合同方有义务无延误地通知对方，有关他们理解或应该理解的事情会影响协议的实施。这样的通知不使合同方解除协议所导致的责任。

供应商的义务

供应商负责按照协议的条款提供合同业务业绩。供应商负责分配必要的资源来实施协议中的承诺。

客户的义务

客户应该支付达成协议的补偿。客户必须辅助供应商，使供应商不会延误或另外避免履行义务。客户负责分配必要的资源，确保来自达成协议的第三方的必要辅助。

[1]供应商的责任限制为管理硬件的更改。供应商不负责硬件以及保修、定价、运输成本等等，这些必须另外与机器的供应商达成协议。

[2]客户在 Debian Edu 中使用另外的应用来进行。供应商可以以每人 NOK 50 不含税的价格做这项服务。

10.1.5.2 2. 保密

合同方有义务相互强制保密，并且不传播他们所知到的与执行协议相联系的信息，使这些信息不被公开。这同样应用于所有标记为保密的材料。个人事件，会伤害到合同方或被业务以外人利用的信息。这项保密义务应用于合同方及其雇员，以及代表合同方与执行协议相联系而执行的其他方。相关联的保密责任在合同结束后依然有效。

10.1.5.3 3. 不可抗力

在双方可控之外的极端情况下，这种情况在一开始不可预见并且极大地妨碍了双方义务的履行，另一方应该被通知而不过度延误。只要极端情况出现，受影响一方的义务中断到相关的程度。另一方相应中断相同的时间。如果不可抗力情况使其维持协议的负担特别沉重，任何一方应该给予另一方一个月的书面通知，从而可以终止协议。

10.1.5.4 4. 协议的转让

合同方只能在另一方书面同意的情况下重新分配其权利和义务。同意不能被不合理地拒绝。如果一方与一家或多家其他公司合并，或者分配给子公司，那么不认为是转移。在此协议下的补偿权力应自由分配，但这样的转移不解除合同方的义务和责任。

10.1.5.5 5. 不履行

10.1.5.5.1 5.1 交付延误

10.1.5.5.2 a. 已清偿损失额

如果在合同方达成协议的日期没有进行交付，并且不是由于第 3 款提到的状况或客户负责的状况，那么从达成协议的日期开始实施每日罚金。罚金的费用是延误提供服务部分中达成协议的年补偿额的 0.1%，以延误的日历日来计算，直到最长 60 天。只要实施了每日罚金，客户既不可以终止协议，也不能对延误要求折扣或其他补偿。

10.1.5.5.3 b. 取消

如果在清偿损失时间终止时依然没有交付，您可以终止合同并立即生效。

10.1.5.5.4 c. 客户导致的延误

在延误由客户导致的情况下，供应商可以通过书面通知，取消他们的工作直到客户采取纠正行为。供应商有权追回由于客户违约导致的另外的成本，以及从新分配资源的合理时间。

10.1.5.5.5 5.2 运行周期的违约

10.1.5.5.6 5.2.1 供应商未履约

10.1.5.5.7 a. 缺陷

如果供应商提供的服务不满足协议中规定的要求和规格，并且这导致供应商负责的状况，那么是供应商的缺陷。如果在运行中有任何缺陷，供应商应不过分延误来弥补缺陷。当缺陷可以在合理时间修复时，客户应该有权要求一定比例的折扣，参考相面的 b. 部分。

10.1.5.5.8 b. 缺陷的价格折扣

如果客户不能全部或部分地使用提供的服务，作为缺陷的结果，客户有权，从错误或缺陷被书面告知时到缺陷修复后，接收一定比例的折扣。任何由于相同状况而缺少可用性的退款，在计算折扣时扣除。

10.1.5.5.9 c. 终止

在任何其他缺陷，显著影响客户使用提供的服务，并且在客户将缺陷书面通知供应商后 30 个工作日还没有纠正的情况下，客户可以以书面形式通知供应商终止协议。如果供应商在通知后没有在 14 个工作日内改正问题，客户有权终止协议并立即生效。

10.1.5.5.10 5.2.2 客户未履约

如果客户没有按时支付，供应商有权收取逾期数量的利息。(在挪威，这来源于 1976 年 12 月 9 日第 100 号关于延迟支付的法律，§ 3，第一节。) 在到期 14 天内没有支付款项与利息的情况下，供应商可以发出书面通知，提供的服务将中断，或者协议将终止。除非客户在接到通知的 7 日内安排好未支付的账单。在由于客户的过错而终止协议时，供应商应该被客户赔偿与协议相联系的执行成本与债务。

10.1.5.6 6. 替代

客户可以针对由于缺陷而导致的合理损失要求补偿，除非供应商可以说明，违约或违约的原因不是由于供应商自己。对于相同的违约，任何根据条款 5a 的由于延误导致的清偿损失，可以通过计算补偿来扣减。如果客户违反本协议下的义务，供应商有权要求补偿由于客户违约导致的合理的另外成本，除非客户可以证明违约或违约的原因不在于客户。

合同方不负责另一方的非直接损失，包括预期的节省或收益。非直接损失包括在其它之中

- 由于减少或丢失的产品或销售导致的损失 (运行中断);
- 因为提供的服务不能如预期而使用发生的损失 (作为结果的损失);
- 落下或没有适当满足的与第三方合同而导致损失的收益。

合同方对彼此的责任限制在达成协议的年度补偿内，或者最大 NOK 1 百万，而无论损坏案件的数量。合同方责任的限制不适用，如果他负责的合同方或任何人，显示出严重的过失或故意的失职的话。

10.1.5.7 7. 法律缺陷

如果第三方断言，客户或经销商具有许可责任的软件的使用违反了第三方的权利，合同方应确保适当的权利被保持或被获取，或者确保得到相同的软件功能而不向另一方收费。如果所有权来自第三方针对客户或经销商根据与另一方关系中的固有缺陷，那么侵权方自己承担费用，来辅助并且最终为双方引领案件。从一方接管案件的时间开始，另一方有义务协助特殊补偿。

10.1.5.8 8. 分包商的责任

合同方完全负责由分包商执行的达成协议的服务。

10.1.5.9 9. 规定协议的终止

终止时，双方应该做出清算客户关系和双方义务的联合计划，在此清算的实际工作中帮助彼此。经销商有义务在此协议终止时返还客户软件和以达成协议的格式的当前数据。客户选择运输方式，并负责从经销商假定的运输。在协议终止后，客户立即承担义务将属于经销商的所有设备返还。经销商选择运输方式，并负责由经销商假定的运输。

10.1.5.10 10. 合法性与争议的解决

本协议中的权力和义务应该完全符合挪威法律。与本协议相联系的争议应该通过双方协商解决。如果合同方不能在两周内通过协商解决分歧，那么任何一方可以要求，在 1915 年 8 月 13 日的第 6 号法律第 32 章 (民事诉讼程序) 的条文下，通过仲裁解决。每一方应该指定一名仲裁人，双方的仲裁人共同指定仲裁法庭。如果在一方要求仲裁并指定其代表后，另一方没有在两周内指定其代表，那么将由奥斯陆地区法庭的首席法官来指定。如果双方仲裁人没有在被指定后 14 天内选择主席，那么相同规则应用于主席的选举。

10.1.6 附件 6 －联系人与地址

1. 关于协议的回应请求应当为书面形式，并且地址如下：

对于经销商	对于客户
运营公司 LtdBy 授权的人，机房 10313，奥斯陆	NN 由授权的人

2. 授权的人

后面的人有权根据协议在他们的部分签字。

姓名	职位/职责	电话	传真	电子邮箱
经销商				
Petter Smart	CEO	+47 22 31 31 31		ps@driftselskapet.no
客户				