

WIRTSCHAFTSUNIVERSITÄT WIEN

BAKKALAUREATSARBEIT

Titel der Bakkalaureatsarbeit:

Einrichten einer Druckumgebung mit Abrechnungssystem für den PC7 der
Wirtschaftsuniversität Wien

Englischer Titel der Bakkalaureatsarbeit:

Installing a printing environment with an accounting system for the PC7 at
the Vienna University of Economics and Business Administration

VerfasserIn:	Daniel-Antonios Zafirakis
Matrikel-Nr.:	0550574
Studienrichtung:	Wirtschaftsinformatik
Kurs:	IT-Praktikum mit Bakkalaureatsarbeit
Textsprache	Deutsch
BetreuerIn:	Dipl.-Ing. Mag. Dr. Albert Weichselbraun
Unternehmen/Betreuer:	Ralf Steuer

Institut für Informationswirtschaft
Wirtschaftsuniversität Wien, UZA II 3.Ebene
Augasse 2-6
A-1090 Wien

Ich versichere:

dass ich die Bakkalaureatsarbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

dass ich die Ausarbeitung zu dem obigen Thema bisher weder im In- noch im Ausland (einer Beurteilerin/ einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

dass diese Arbeit mit der vom Betreuer beurteilten Arbeit übereinstimmt.

_____ Datum

_____ Unterschrift

Zusammenfassung

Um die angebotenen Dienstleistungen des Instituts für Informationswirtschaft der Wirtschaftsuniversität Wien an seinen Studenten zu verbessern, soll eine Druckumgebung mit Abrechnungssystem für den Schulungsraum des Instituts konzeptioniert und umgesetzt werden.

Diese Bakkalaureatsarbeit befasst sich mit der Implementierung der Druckumgebung [Common Unix Printing System \(CUPS\)](#) mit einem dazu passenden und flexiblen Abrechnungssystem, PyKota, auf Debian [GNU/Linux](#).

Auf die spezifischen Rahmenbedingungen des Projektes wie [Open Source Software \(OSS\)](#) wird ebenso eingegangen, wie auf den Softwarehintergrund der verwendeten Programme. Außerdem erfolgt ein Exkurs zu verschiedenen Virtualisierungstechniken unter Linux, da diese verwendet werden, um die Implementierung dieser Arbeit zu testen.

Schlüsselwörter

CUPS, Linux, Debian, Open Source Software, PyKota, Xen, QEMU, KVM

Abstract

To improve the services provided to the students by the Institute for Information Business of the Vienna University of Economics and Business Administration, a printing system with an appropriate print accounting system has to be designed and implemented.

This bachelor thesis deals with the implementation of [CUPS](#), a modular printing system, and [PyKota](#), a very customizable print accounting system, on Debian [GNU/Linux](#).

The general conditions like [Open Source Software \(OSS\)](#) are covered as well as the background information of the software used during this project. Furthermore an excursus to the different virtualization techniques used in Linux takes place, as these will be used to test the implementation of this thesis.

Keywords

CUPS, Linux, Debian, Open Source Software, PyKota, Xen, QEMU, KVM

Inhaltsverzeichnis

1. Anforderungen an das Projekt	6
1.1. Einleitung	6
1.2. Vergleich gegenwärtige und gewünschte Situation	7
1.3. Anforderungen an die Druckumgebung	8
1.4. Projektmanagement	9
2. Informationen zur verwendeten Software	14
2.1. Linux	15
2.2. Virtualisierung unter Linux	18
2.3. Kerberos	25
2.4. Python	26
2.5. CUPS - PyKota	28
3. Implementierung von CUPS und PyKota	32
3.1. Installation und Konfiguration von CUPS	33
3.2. Implementierung von PyKota	37
3.3. Testen der Druckumgebung im virtuellen System	43
3.4. Anpassen der Druckumgebung an die Anforderungen	45
3.5. Überführung auf das Produktivsystem	52
4. Zusammenfassung und Ausblick	54
A. Anhang	55
Glossar	62
Abbildungsverzeichnis	67
Literaturverzeichnis	69

1. Anforderungen an das Projekt

Dieses Kapitel gibt einen Überblick über die Anforderungen an das Projekt. In der Einleitung werden die Beweggründe für dieses Projekt dargestellt, danach erfolgt eine Gegenüberstellung der aktuellen zur gewünschten Situation. Im Anschluss daran werden die Anforderungen an das Projekt aufgelistet, den Schluss macht das Unterkapitel Projektmanagement, welches die Methoden, die bei diesem Projekt zum Einsatz gekommen sind, enthält.

1.1. Einleitung

Im Rahmen des Bakkalaureat-Studiums „Wirtschaftsinformatik“ (Studienkennzahl: J 033 526) an der [Wirtschaftsuniversität Wien \(WU\)](#) müssen für den erfolgreichen Abschluss zwei Bakkalaureatsarbeiten geschrieben werden, eine davon im Rahmen eines IT-Praktikums, welches entweder bei einem Unternehmen oder an der [WU](#) absolviert werden kann. Der Autor dieser Arbeit hat sich dazu entschlossen, das IT-Praktikum, diese Arbeit, am Institut für Informationswirtschaft der [WU](#) durchzuführen. Das Institut gehört zum Department „Informationsverarbeitung und Prozessmanagement“ der [WU](#) und beschäftigt sich mit Forschung und Lehre im Bereich der Informationswirtschaft, Enterprise Resource Planning (ERP) Systems, IT-Controlling, Information Retrieval, Datenbanksystemen und vielem mehr¹. Seinen Studenten bietet das Institut neben einem eigenen Schulungsraum

¹Mehr auf <http://ai.wu-wien.ac.at/>

(PC7) mit 25 PCs und einem institutseigenem Server auch Themen für das IT-Praktikum.

Als Thema für diese Arbeit wurde das Einrichten der Druckumgebung [CUPS](#) mit einem entsprechenden Abrechnungssystem für den PC7 gewählt. Für das Institut bedeutet dies unter anderem eine Verbesserung der Dienstleistung für die Studenten. Die Motivation des Autors liegt im Kern der im PC7 eingesetzten Software und der Anforderungen an das Projekt: [Open Source Software \(OSS\)](#).

Die folgende Arbeit beschäftigt sich mit der Installation und Konfiguration von [CUPS](#) und [PyKota](#), einem sehr flexiblen und unter der [GNU General Public License \(GPL\)](#) stehenden Abrechnungssystem. Eingangs werden die Anforderungen an das Projekt beschrieben, danach werden die theoretischen Grundlagen der verwendeten Software näher betrachtet. Im Anschluss daran wird auf die Implementierung eingegangen.

1.2. Vergleich gegenwärtige und gewünschte Situation

Eine Gegenüberstellung der gegenwärtigen zur gewünschten Situation im herkömmlichen Sinn ist nicht möglich, da diese Arbeit die eigentliche Dienstleistung des PC7 nicht verändert. Diese Dienstleistung, das Anbieten eines Computerraumes mit [OSS](#) für Studenten des Instituts, wird lediglich, um die Option vor Ort zu drucken, erweitert.

Zur Zeit befinden sich 25 PCs im PC7, welche außerhalb von Lehrveranstaltungen mäßig benutzt werden. Er gilt aufgrund seiner Lage, seiner Auslastung und der beschränkten Anwenderzahl, verglichen mit den PC-Räumlichkeiten des [Zentrum für Informatikdienste \(ZID\)](#), als „Geheimtipp“ unter den Wirtschaftsinformatikstudenten (bzw. Studenten die Lehrveranstaltungen im PC7 besuchen). Er ist die perfekte Lernzone, es sind Unterhaltungen möglich, ohne dass man sofort den Raum verlassen muss (natürlich

nur im Rahmen der Kollegialität), man kann nebenbei den PC benutzen, um Aufgabenstellungen zu lösen, und man ist meistens unter gleichgesinnten Studenten, die den PC7 ebenfalls als Lernzone „missbrauchen“.

Um ein Dokument zu drucken während man sich im PC7 befindet, muss man zur Zeit das Dokument z.B. auf einem USB-Stick speichern, in die Schulungsräume des ZID gehen, welche sich im Keller desselben Gebäudes (UZA2) befinden, Jacke und Tasche in den Spind sperren, sich bei einem PC einloggen, den USB-Stick anstecken und den Druckauftrag starten. Eine kleine Weltreise, wenn man lediglich eine Kleinigkeit drucken möchte.

Um dies zu ändern, das Drucken im PC7 zu ermöglichen und das Service an den Studenten weiter zu verbessern, ist eine, in das verwendete Betriebssystem integrierte, Druckumgebung mit dazu passendem Abrechnungssystem notwendig. Primäres Ziel dieses Projektes ist es, eine Druckumgebung einzurichten die für den Benutzer, also nach außen hin, unsichtbar ist und die es dem Institut ermöglicht, Druck- und Benutzerstatistiken anzulegen. Weiters soll eine einfache Kostenkontrolle für den Systemadministrator gewährleistet werden. Ein wichtiges Nebenziel ist, dass sowohl die Drucksoftware als auch das Abrechnungssystem freie Software im Sinne der Open Source Definition zu sein hat.

1.3. Anforderungen an die Druckumgebung

Das Hauptaugenmerk des Projekts liegt bei der Implementierung der Druckumgebung CUPS mit einem dazu passenden Abrechnungssystem. Das Abrechnungssystem soll so flexibel sein, dass es in weiterer Folge einfach zu warten und zu erweitern ist. Weiters soll es Benutzer- und Druckerstatistiken zur Verfügung stellen können, um die Wirtschaftlichkeit im laufenden Betrieb überprüfen zu können.

Darüber hinaus soll das Abrechnungssystem ein frei verfügbares Programm im Sinne der Open Source Definition (OSD) sein und sich benutzerfreundlich

in das jeweilige Desktop Environment integrieren. Auch soll die Möglichkeit der Kommunikation zur Laufzeit geprüft werden, damit der Benutzer, z.B. über etwaige Fehlermeldungen oä, informiert werden kann. Zusätzlich soll eine Applikation für Sekretariat und Studierende implementiert werden, welche

- Kauf und Kontrolle des Druckguthabens,
- Ausdruck eines entsprechenden Belegs und
- Statistiken

ermöglicht.

1.4. Projektmanagement

Die in dieser Arbeit beschriebene Aufgabe kann durchaus als Projekt bezeichnet werden gemäß der in [Abb. 1.1](#) beschriebenen Definition, obwohl einige Ausprägungen eher zu einem Prozess passen würden. Dieser Klassifizierung folgend, handelt es sich bei dieser Arbeit um ein Kleinprojekt, was eine Adaptierung gängiger Projektmanagementregeln erlaubt [zuc07].

Charakteristika von Geschäftsprozessen	Ausprägungen
Häufigkeit	einmalig
Dauer	kurz
Bedeutung	mittel
Leistungsumfang	mittel
Ressourceneinsatz	gering
Kosten	gering
Organisationen	wenige

Abbildung 1.1.: Projektdefinition anhand der Ausprägungen eines Prozesses

Projektauftrag

Der Projektauftrag ist nicht nur der Schlüssel für einen erfolgreichen und korrekten Projektstart, er ist das wichtigste Element für ein Projekt gemäß dem Credo „Kein Projekt ohne Projektauftrag!“. Wie in [Abb. 1.2](#) ersichtlich definiert er die zeitliche Grenzen, spezifiziert die Ziele und Nicht-Ziele des Projektes, dient als Referenzdokument für das restliche Projektmanagement und erleichtert die Kommunikation zwischen Projektteam und Projektauftraggeber, je genauer er ist [[zuc07](#)].

Projektstrukturplan

Der [Projektstrukturplan \(PSP\)](#) stellt die einzelnen Phasen des Projektes übersichtlich dar. Bis auf Phase 1 (Projektmanagement), welche sich über das gesamte Projekt erstreckt, entsprechen die einzelnen [PSP-Einheiten](#) einer zeitlichen Reihenfolge.

Projektmeilensteinplan

Ein Meilenstein ist ein Ereignis besonderer Bedeutung bei einem Projekt, meist ein Zwischenziel, und ist wesentlicher Bestandteil des Projektmanagements. Bei den in [Abb. 1.4](#) angegebenen Meilensteinen wird zusätzlich die Ist-Zeit der jeweiligen Meilensteine eingetragen, um eventuelle Abweichungen im zeitlichen Ablauf so früh wie möglich erkennen und beseitigen zu können

Projektbalkenplan

Der Projektbalkenplan zeigt den Projektstrukturplan im zeitlichen Ablauf, kombiniert mit den Meilensteinen. Der Übersicht halber wurden Teile desselben in [Abb. 1.5](#) ausgeblendet, welche den geplanten Ablauf des Projektes darstellt.

<i>Projektstart:</i> 11.07.2008	<i>Projektende:</i> November 2008
<i>Projektziele:</i> <ul style="list-style-type: none"> • Recherche, Auswahl und Implementierung eines Open Source Abrechnungssystems für CUPS • Benutzerfreundliche Integration in die Desktopumgebung • Dokumentation für Studierende, Sekretariat, Tutoren, Administratoren 	<i>Nicht-Projektziele:</i> <ul style="list-style-type: none"> • Auswahl einer anderen Druckumgebung • Laufender Betrieb und Wartungstätigkeiten <i>Projektkosten:</i> Projektmitglieder: kostenlos Externe: - Sonstiges: € 25,- <i>Projekterträge:</i> keine direkten Erträge
<i>Optionale Ziele:</i> <ul style="list-style-type: none"> • Hinzufügen von weiteren Schulungsräumen / Gruppen • Überführung der Software in das Produktivsystem 	
<i>Hauptaufgaben/Projektphasen:</i> <ul style="list-style-type: none"> • Projektmanagement und Administration • Implementierung • Übergabe 	
<i>Projektpartner:</i> Ralf Steuer	<i>Projektbetreuer:</i> Dipl.-Ing. Mag. Dr. Albert Weichselbraun
<i>Projektleiter:</i> Zafirakis Daniel-Antonios	
<i>Projektmitglieder:</i> Zafirakis Daniel-Antonios	
<i>Zusammenhang zu anderen Projekten:</i> -	
<i>Relevante Projektumgebung:</i> Institut für Informationswirtschaft der WU	

Abbildung 1.2.: Projektauftrag

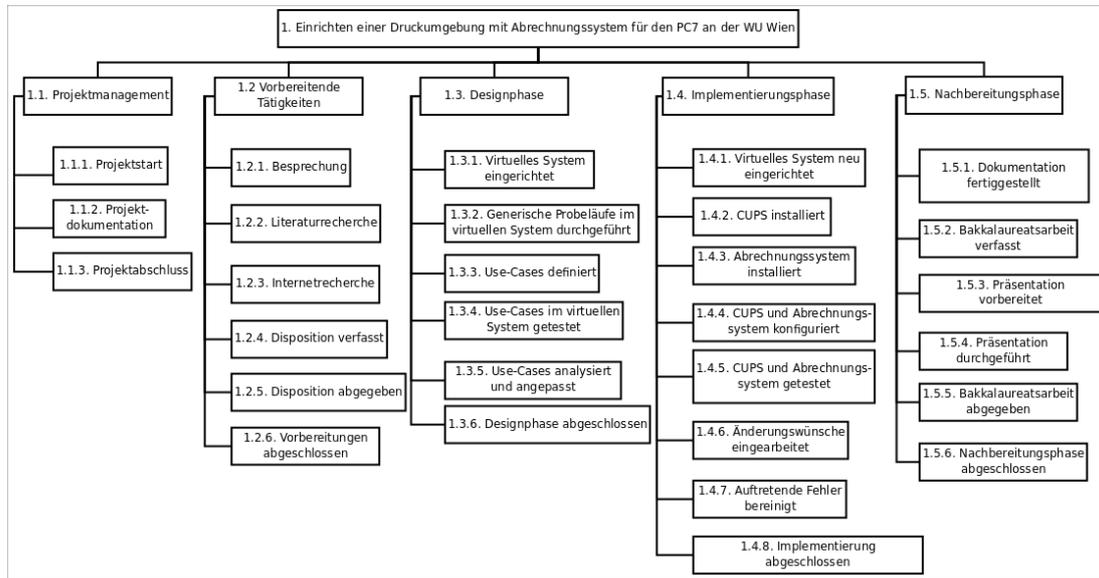


Abbildung 1.3.: Projektstrukturplan

PSP	Beschreibung	Soll	Ist
1.1.1	Projekt gestartet	11.07.08	11.07.08
1.2.4	Disposition abgegeben	05.08.08	12.08.08
1.3.4	Use-Cases analysiert	04.09.08	10.09.08
1.4.4	Druckumgebung und Abrechnungssystem konfiguriert	28.09.08	30.09.08
1.4.7	Aufgetretene Fehler bereinigt	10.10.08	03.11.08
1.6.5	Bakkalaureatsarbeit abgegeben	16.11.08	20.12.08
1.3.1	Projekt abgenommen	16.11.08	03.02.09

Abbildung 1.4.: Meileinsteinplan

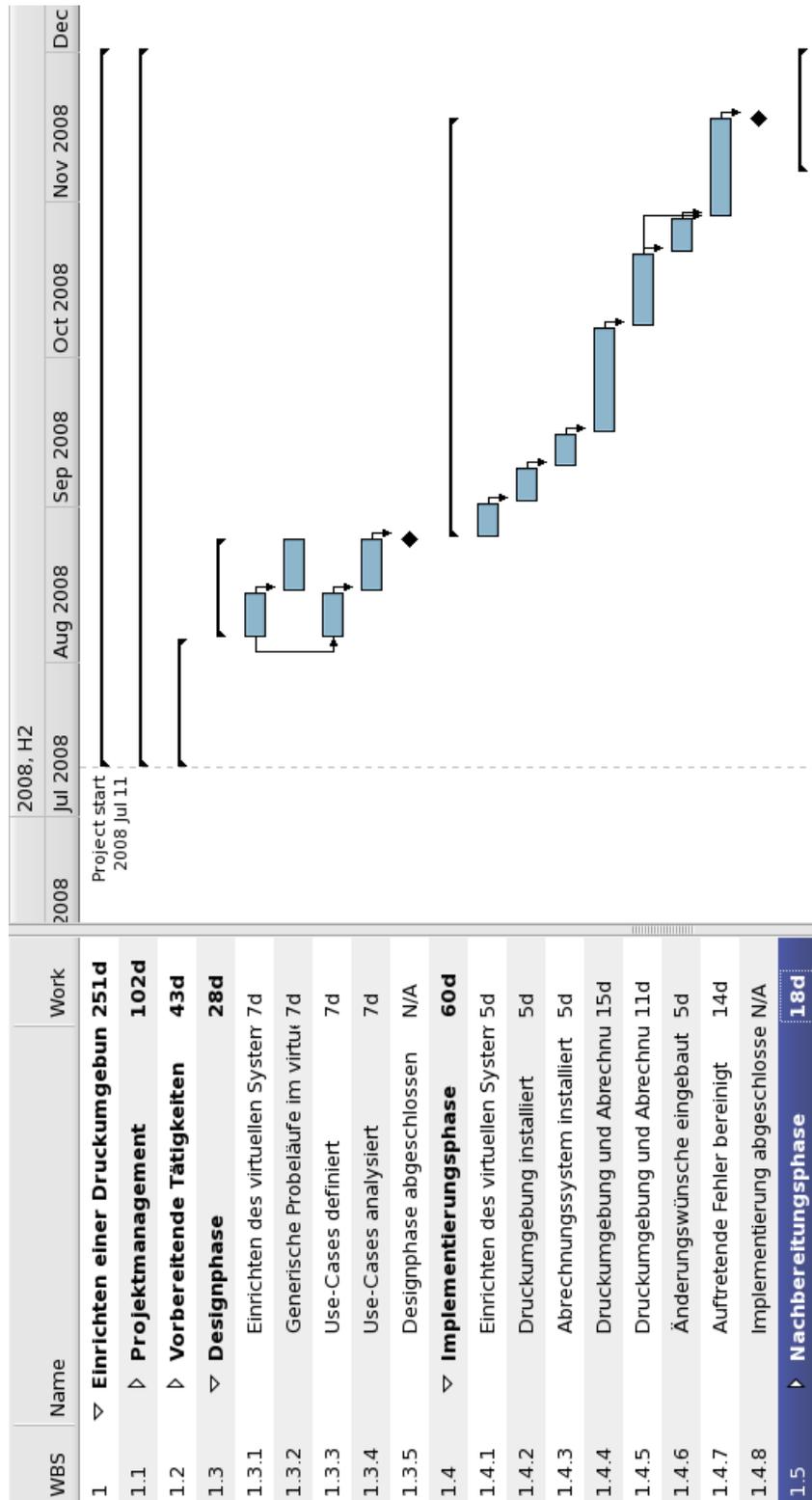


Abbildung 1.5.: Projektbalkenplan

2. Informationen zur verwendeten Software

Dieses Kapitel erläutert theoretisch die verwendeten Werkzeuge und versucht einen Überblick über die selbigen zu geben. Den Anfang macht Linux. Es wird kurz auf die Geschichte des Betriebssystems eingegangen, die Unterscheidung Linux (Kernel) und Linux (Distribution) erklärt und Debian GNU/Linux vorgestellt, da dies die verwendete Distribution ist.

Anschließend werden die verschiedenen Arten von Virtualisierung generell, und im speziellen unter Linux, näher betrachtet. Installation von neuer Software auf einem Produktivsystem birgt gewisse Gefahren in sich, welche durch den Einsatz von Virtualisierung verhindert werden können. So können z.B. auf einem Host-System verschiedene Gast-Systeme laufen, welche andere Betriebssysteme verwenden oder, wie in dem Fall dieser Projektarbeit, als Spielwiese für Testinstallationen des Produktivsystems dienen.

Weiters wird ein kurzer Überblick zu [Kerberos](#) und seiner Funktionsweise gegeben. Den Abschluss des Kapitels machen die Programmiersprache Python, das Linux-Drucksystem [CUPS](#) und Pykota, ein flexibles Abrechnungssystem für [CUPS](#).

Auf die Datenbanksprache SQL wird nicht näher eingegangen obwohl sie von PyKota verwendet wird, da es im Rahmen der Installation zu keiner Veränderung der Abfragen gekommen ist.

2.1. Linux

Linux ist ein Unix-ähnliches Betriebssystem. Genau genommen verstand man unter Linux lediglich den Kernel des Betriebssystems, so wie er am 26. August 1991 von *Linus Torvalds* in der Usenet-Gruppe *comp.os.minix* vorgestellt worden ist und am 17. September 1991 als Download auf einem öffentlichem FTP-server verfügbar war¹ [lto91]. Ein Kernel ist der zentrale Bestandteil eines Betriebssystems, er kümmert sich um z.B. elementare Funktionen wie Speicherverwaltung, Lastenverteilung, Prozessverwaltung und Steuerung der Hardware. Der Kernel bildet die hardwareabstrahierende Schicht und stellt, der auf dieser Basis aufsetzenden Software, eine einheitliche Schnittstelle (API) zur Verfügung, die unabhängig von der Rechnerarchitektur ist. Die Software kann so immer auf die Schnittstelle zugreifen, ohne genauere Kenntnisse der zugrundeliegenden Hardware zu benötigen. Linux ist ein modularer monolithischer Betriebssystemkern. Die Versionsnummer des Linuxkernel besteht grundsätzlich aus drei Ziffern in der Form *x.y.z*, die aktuelle stabile Version ist 2.6.27 [mk05, wlk08].

Die erste Ziffer wird nur bei grundlegenden Änderungen in der Systemarchitektur geändert, allerdings wurde dies beim Wechsel der stabilen Version 2.4 auf Version 2.6 nicht durchgeführt, da Linus Torvalds aus verschiedenen Gründen dagegen war. Ob es jemals eine Version 3.0 geben wird, wurde noch nicht direkt ausgeschlossen, allerdings erscheint ein solcher Release, dem eine starke Änderung der Codebase vorangehen müsste, in absehbarer Zeit nicht notwendig [lto02, jbl07].

Die zweite Ziffer gibt das jeweilige „Majorrelease“ an. Bisher wurde zwischen stabilen Kernelversionen mit gerader Ziffer (z.B. 2.2.n, 2.4.n, 2.6.n) und Entwicklerkernel mit ungerader Ziffer (z.B. 2.3.n, 2.5.n) unterschieden. Allerdings gibt es zur Zeit keine Version 2.7, stattdessen werden Änderungen oder neue Features² laufend in die aktuelle 2.6er-Serie eingearbeitet.

¹Damals noch unter einer eigenen, proprietären Lizenz. 0.99 war der erste Kernel unter der [GPL](#).

²So wird z.B. ein neues Dateisystem (ext4) mit 2.6.28 eingeführt [ext04].

Weiters bezeichnet die dritte Ziffer das „Minorrelease“, welches die eigentliche Version kennzeichnet. Werden neue Funktionen hinzugefügt, so steigt die dritte Ziffer. Der Kernel wird so z.B. mit der Nummer 2.6.25 bestimmt.

Um die Korrektur eines schwerwiegenden Fehlers schneller verbreiten zu können, wurde mit 2.6.8.1 erstmals eine vierte Ziffer eingeführt. Seit Kernel 2.6.11 wird diese Nummerierung offiziell verwendet. So wurde es möglich, die Stabilität des Kernels trotz der schnelleren Veröffentlichungszyklen zu gewährleisten und zu verbessern, da Korrekturen kritischer Fehler innerhalb weniger Stunden in den offiziellen Kernel übernommen werden konnten, wobei sich die vierte Ziffer (z.B. von 2.6.17.4 auf 2.6.17.5) erhöht [jku05, wlk08].

Wenn man heutzutage von Linux spricht, dann spricht man normalerweise nicht vom Linux-Kernel, sondern von einer Linux-Distribution, wie z.B. Gentoo, Ubuntu, Debian, Archlinux, CentOS, ... Es gibt eine große Anzahl an verschiedensten Linuxdistributions für verschiedenste Anwendungsbereiche; die meisten sind bei <http://www.distrowatch.com>³ gelistet. Der größte Anwendungsbereich von Linux liegt im Serverbereich, der genaue Marktanteil kann allerdings nicht genau ermittelt werden⁴, da die meisten Untersuchungen in diesem Bereich auf den verkauften Lizenzen basieren und nicht das tatsächliche Betriebssystem des funktionalen Web-Servers ermitteln [cma07, rml08].

Mittlerweile nimmt Linux seit dem Beginn der Linux-Distribution Ubuntu im Oktober 2004 auch im Desktopbereich eine immer ernster zu nehmende Rolle ein, spätestens jedoch seit dem bei DELL die Möglichkeit besteht, Desktop-PCs bzw Laptops mit vorinstalliertem Ubuntu-Linux zu kaufen. Ubuntu ist eine auf Debian GNU/Linux basierende Linux-Distribution und eignet sich durch die sehr einfache Installation und Handhabung („Windows-like“, mit Mac OS X Touch), dem ansprechendem Design, den sehr

³Distrowatch.com listet nicht ausschließlich Linux-Distributionen.

⁴Der Marktanteil gemäß verkaufter Lizenzen wird auf, je nach Studie, zwischen 12% und 30% geschätzt. Der Markt für Linux-basierende Server wächst laut einer Studie der International Data Corporation (IDC) bis zum Jahr 2012 um durchschnittlich 8% [lml07, svn07, woh08].

guten Software-Repositories und der großen Community vor allem als Einsteigerdistribution in die Welt von Linux und [OSS](#). Auch der aktuelle Netbook-Trend erhöht die Bekanntheit von Linux im Desktop-Bereich, da sehr viele Netbooks mit vorinstalliertem Linux verkauft werden [[dll07](#)].

Bei der für den PC7 verwendeten Linux-Distribution handelt es sich um Debian [GNU/Linux](#), in den Versionen 3.1 (aka *sarge*) auf dem Studentenserver balrog bzw 4.0 (aka *etch*) auf den Clients im PC7.

Debian [GNU/Linux](#) wurde am 16. August 1993 durch Ian Murdock ins Leben gerufen. Der Name „Debian“ leitet sich aus den Vorname des Gründers („Ian“) und seiner Frau („Debra“) ab. Es ist eine der wenigen Distributionen die sich *GNU/Linux* nennen. Damit folgt das Debian-Projekt der Auffassung der [Free Software Foundation \(FSF\)](#), das Linux eine Variante des [GNU-Systems](#) ist⁵. Bekanntheit erlangte das Projekt vor allem durch sein Paket-Managementsystem APT. Mit APT ist es möglich Software zu suchen, zu de-/installieren oder das komplette System auf den neuesten Stand zu bringen. Das besondere (und damals einzigartige) an APT war die Art der Auflösung der benötigten Abhängigkeiten eines gewünschten Programmes.

Das Debian-Projekt konstituiert sich durch die Debian-Verfassung, der Posten des Leiters wird einmal pro Jahr durch Wahlen neu vergeben. Mit dem Gesellschaftsvertrag „Debian Social Contract“ verpflichtet das Debian-Projekt sich und seine Entwickler zu freien Werken im Hauptzweig *main*. Ein besonderer Teil des Vertrages, die [Debian-Richtlinien für Freie Software \(DFSG\)](#), welche bei Debian festlegen, ob eine Software frei ist, wurde als Grundlage für die Definition des Begriffes *Open Source* in der Öffentlichkeit verwendet. Diese [OSD](#) wird seitdem von der [Open Source Initiative \(OSI\)](#) verwendet, mittlerweile haben sich allerdings einige Unterschiede zur [DFSG](#) ergeben. Die [DFSG](#) wird vom Projekt sehr ernst genommen, zentrale Diskussionen im Linux-Umfeld werden maßgeblich vom Projekt bestimmt, wie die konsequent freie Dokumentation der Programme (Diskussion über die

⁵Mehr zum [GNU/Linux-Namenstreit](#) gibt es unter <http://de.wikipedia.org/wiki/GNU/Linux-Namensstreit>.

GNU Lizenz für freie Dokumentation (GFDL)) oder die Vermeidung von Markennamen, weil z.B. ein Hersteller darüber das Projekt beeinflussen könnte (Iceweasel statt Firefox) [deb04, deb07, osi06].

2.2. Virtualisierung unter Linux

Dieses Kapitel wird auf die bekannteren Virtualisierungslösungen unter Linux eingehen. Virtualisierungstechniken gibt es bereits länger als man vermuten würde. Schon in den 1960ern legte IBM den Grundstein für diese Technik. Damals noch aus der Not heraus, denn die damaligen Rechner waren sehr groß, selten und dementsprechend teuer. Für den Begriff Virtualisierung gibt es viele verschiedene Definitionen, eine passende Definition für dieses Projekt wäre die folgende:

Virtualisierung bezeichnet im übergreifenden Sinne Software- oder Hardware-Techniken, welche eine Abstraktionsschicht zwischen dem Benutzer (oder Applikationen oder Schnittstellen) einerseits und physischen Ressourcen wie z.B. Hardwarekomponenten eines Rechners andererseits, implementieren [itw08].

Es gibt verschiedene Arten von Virtualisierung. Man kann eine grobe Unterscheidung zwischen Software- und Hardware-Virtualisierung treffen, tatsächlich gibt es viel mehr und vor allem viel unterschiedlichere Arten der Virtualisierung. Da im Rahmen des Projektes nur Software-Virtualisierung in Frage kommt und verwendet wird, wird hier auf die Möglichkeiten der Hardware-Virtualisierung (bis auf die Prozessorerweiterungen von Intel bzw AMD) nicht näher eingegangen. Die folgende Grafik (Abb. 2.1) zeigt die verschiedenen Arten der Virtualisierung. Links durch eine Virtualisierungssoftware (z.B. VMware), rechts durch Virtualisierung auf Hardwareebene (z.B. durch Intel VT bzw AMD-V).

Die Software-Virtualisierung kann vielseitig eingesetzt werden, z.B. zum Simulieren/Emulieren einer Hardwareumgebung für ein Betriebssystem oder

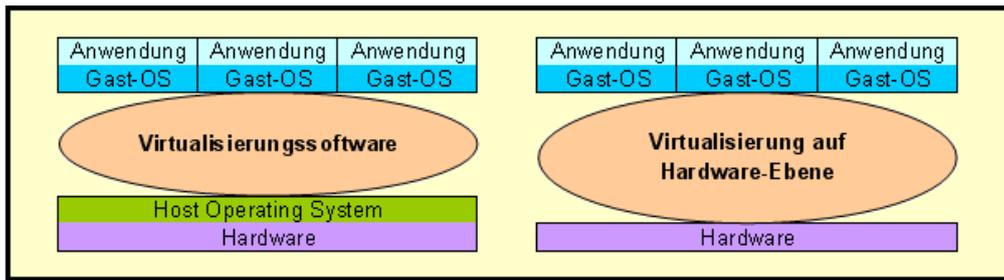


Abbildung 2.1.: Zwei verschiedene Arten von Virtualisierungsansätzen.

einer Betriebssystemumgebung für eine Anwendung. Weitere Anwendungsbereiche sind

- die Hardware-Emulation,
- die Hardware-Virtualisierung und
- die Paravirtualisierung.

Die folgenden Programme sind sehr bekannte und viel verwendete Virtualisierungsprogramme unter Linux, die meisten sind dem Bereich der Hardware-Virtualisierung zuzuordnen [wvi08, mfu08].

VMware

VMware ist ein 1998 gegründetes Unternehmen mit dem Ziel, Software im Bereich der Virtualisierung herzustellen und virtuelle Maschinen auf Standard-Computern zur Anwendung zu bringen. Aufgrund der Vielfalt der Produktpalette, dem Hintergrund der proprietären Lizenz und der Kosten dieser Software⁶, wird im Rahmen dieser Arbeit auf kein spezielles Produkt näher eingegangen. Der Vollständigkeit halber wird VMware aufgeführt, da das Unternehmen zum Teil Pionierarbeit in diesem Bereich geleistet hat.

Die VMware Desktop Software läuft auf Microsoft Windows, Linux und Mac OS X, benötigen also ein Betriebssystem und bestimmte Treiber, die

⁶Mittlerweile gibt es auch kostenlose Software wie z.B. VMware Server oder VMware Player, allerdings erhält man eine Lizenz erst nach erfolgreicher Registrierung auf der Homepage des Unternehmens.

der Installation beiliegen. Die Server Applikationen hingegen laufen direkt auf der Hardware. Die Produkte von VMware versuchen CPU Code der Gast-Systeme, sofern möglich, direkt an die CPU des Host-System weiter zu geben. Sollte dies nicht möglich sein, schreibt VMware den jeweiligen Code dynamisch um. Dieses Verfahren wird *dynamic translation* genannt und findet auch Anwendung bei z.B. QEMU/KVM. Dadurch erreicht VMware 80%+ der Geschwindigkeit der Host-CPU, der Overhead soll zwischen 3% und 6% liegen, selbst bei rechenintensiven Aufgaben [vmw08, wvm08].

Xen

Xen ist ein *virtual machine monitor* für IA-32, IA-64, PowerPC 970, x86 und x86_64 Architekturen. Die Hardware wird von Xen für die laufenden Systeme (Domänen) paravirtualisiert (Abb. 2.2), dh es werden Software-schnittstellen bereitgestellt, die ähnlich, aber nicht identisch zur tatsächlichen Hardware sind. Dadurch wird eine sehr hohe Performance erzielt, da die Hardware nicht emuliert wird, sondern lediglich mit einem kleinen Overhead den Gastsystemen zur Verfügung gestellt wird.

Um diese Schnittstellen in der virtuellen Umgebung zu nutzen, müssen die Gastsysteme auf diese virtuelle Umgebung, in diesem Fall Xen, portiert werden. Dies erforderte bis Version 3.0 einen quelloffenen Kernel des Gast-systemes, also z.B. Linux, NetBSD oder OpenBSD. Unter Linux geschieht dies, je nach Distribution, durch Patchen des Kernels mit den Xen-Patches oder durch Aktivieren der entsprechenden Kernelparameter bei bereits vorgepatchten Kernel. Seit Version 3.0 können auch Betriebssysteme wie z.B. Microsoft Windows, dank dafür entwickelter Prozessorerweiterungen (Intel VT oder AMD-V)⁷, ohne Modifikation mit Xen genutzt werden.

Xen unterscheidet privilegierte (sog. Domäne-0) und unprivilegierte Domänen, dh virtuelle Systeme. Domäne-0 hat die volle Kontrolle über das System und den anderen Gast-Domänen. Aus der Domäne-0 werden mittels

⁷Intel VT bzw AMD-V bezeichnen die jeweiligen Implementierungen einer *Secure Virtual Machine* für deren Prozessoren.

PARAVIRTUALIZATION

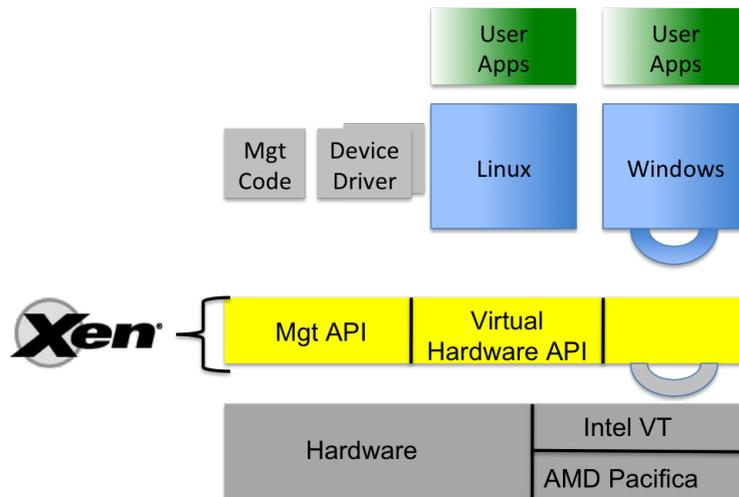


Abbildung 2.2.: Paravirtualisierung unter Xen [xen08]

der Xen-Tools andere Domänen gestartet. Die Anzahl an parallel laufenden Gastsystemen wird nur durch die zur Verfügung stehenden Ressourcen des Rechners beschränkt. Konfiguration und Administration von Xen erfolgt mittels Konfigurationsdateien, grafische Tools stehen ebenfalls zur Verfügung.

Xen empfiehlt sich vor allem zur Server-Virtualisierung. Dies führt zu einer höheren Auslastung der Hardware bei gleichzeitiger Senkung der Total Cost of Ownership (TCO) und Reduzierung des ökologischen Fußabdruckes. Hinter Xen, das an der Universität Cambridge entstand, stehen Unternehmen wie AMD, Cisco, Dell, HP, IBM, Intel, Novell, Red Hat, Sun uvm. Red Hat und SuSe haben Xen relativ früh in ihr System integriert, sowohl bei ihren Community-Distributionen Fedora Core (Red Hat) als auch bei OpenSuse (SuSe). Red Hat hat mit dem Kauf von Qumranet im September 2008, dem Unternehmen das hinter der Entwicklung von KVM steht, einen Schwenk von Xen hin zu KVM vollzogen. Hintergrund dürfte vor allem die Integration von KVM im Vanilla-Kernel ab 2.6.20 sein. Im Gegensatz dazu bietet

Xen lediglich Patches für den zwei Jahre alten Kernel 2.6.18⁸. Durch diesen Wechsel spart sich Red Hat eine Menge Arbeit, erst Recht wenn man die Feature-Liste der Kernels seit 2.6.18 kumuliert betrachtet. Wenn man hingegen als Benutzer Xen auf einem aktuellen Kernel wie z.B. 2.6.26 verwenden will, muss man entweder die Patches auf diesen selber portieren oder eine Distribution benutzen, die dies für einen vornimmt [rh108, wix08, xen08].

Xen ist zur Zeit noch die Virtualisierungslösung, die auf dem Server des Instituts eingesetzt wird. Eine Überprüfung bestehender Alternativen wie z.B. Qemu/KVM wurde bereits durchgeführt. Ob ein Wechsel stattfinden wird, ist noch ungewiss.

VirtualBox

Sun xVM VirtualBox, in Folge der Einfachheit halber nur als „VirtualBox“ bezeichnet, ist eine virtuelle Maschine der Firma Sun Microsystems für x86 und x86_64 Architekturen. Als Host-System kann Linux (>2.4), Windows (>NT), Mac OS X/Intel oder Solaris eingesetzt werden, welche auch als Gastsystem⁹ unterstützt werden.

Es gibt zwei unterschiedliche Ausgaben in unterschiedlichen Lizenzen von VirtualBox:

- *Sun xVM VirtualBox* mit allen Funktionen unter proprietärer Lizenz (PUEL¹⁰), welche für persönliche Zwecke sowie zu jeglicher Verwendung in Bildungseinrichtungen kostenfrei genutzt werden darf.
- *VirtualBox Open Source Edition* (OSE) mit fehlenden Funktionen (z.B. Serial ATA-Controller im Gastsystem) unter der GPL. Die OSE ist ausschließlich als Source-Tarball verfügbar.

⁸Siehe: <http://www.xen.org/download/>

⁹Eine Liste aller unterstützten Gastsysteme ist auf http://www.virtualbox.org/wiki/Guest_OSes verfügbar.

¹⁰http://www.virtualbox.org/wiki/VirtualBox_PUEL

Im Gegensatz zu Xen muss man VirtualBox unter Linux einfach nur installieren, fertige Binaries gibt es für die meisten Distributionen auf der Projektseite bzw befindet sich VirtualBox auch in den Repositories der meisten Distributionen. Der Kernel muss nicht gepatcht werden, ein Kernelmodul wird automatisch mit installiert. Die Konfiguration erfolgt ähnlich wie bei VMWare über ein Frontend, von denen verschiedene zur Auswahl stehen. Die Funktionen von VirtualBox können durch sog. Gasterweiterungen erweitert werden, z.B.

- Der Mauszeiger kann das Fenster der VM verlassen, ohne die „Host-Taste“ benutzen zu müssen.
- Die Grafikauflösung wird sofort an die Fenstergröße im Host angepasst.
- „seamless windows“ – die Programmfenster des Gastsystems können frei auf der Arbeitsfläche des Host-Systems platziert werden.
- Gemeinsame Ordner zwischen Host- und Gastsystem.

Herauszuheben wäre noch die Public API (für Java, Python, [Simple Object Access Protocol \(SOAP\)](#), [Cross Platform Component Object Model \(XP-COM\)](#)) von VirtualBox, über die das Programm konfiguriert und ausgeführt werden kann, und die Möglichkeit auch Festplattendateien von VMWare und Microsoft Virtual PC zu verwenden. Auch werden die Prozessorerweiterungen von Intel und AMD, Intel VT bzw AMD-V, unterstützt, allerdings sind diese standardmäßig nicht aktiviert [[svb08](#), [wvb08](#)].

QEMU - KVM

QEMU ist ein Emulator, der x86-, x86_64-, PowerPC- und Sparc32/64-Hardware emuliert. Anhand von Kommandoparametern kann der Benutzer die Tiefe der Emulation zum Teil selber entscheiden (z.B. Soundsystem) oder sie an eigene Ansprüche anpassen. Mit Hilfe eines Kernel-Moduls,

dem QEMU Accelerator (KQEMU) oder der [Kernel-based Virtual Machine \(KVM\)](#), wird die Hardware nicht mehr emuliert, sondern paravirtualisiert, womit QEMU sich genauso performant verhält wie Xen oder VMware. Komplette virtualisiert können zur Zeit nur einige Teile der Hardware werden und das auch nur, wenn das Gast-System einen Linux-Kernel >2.6.25 verwendet und die entsprechenden Module kompiliert worden sind. Mit Veröffentlichung des Kernel 2.6.20 wurde [KVM](#) in den offiziellen Linux-Kernel aufgenommen, was es zum Einsatz unter Linux auf x86-Hardware empfiehlt. KVM benötigt allerdings die Virtualisierungserweiterung des jeweiligen Prozessors, Intel VT bzw AMD-V.

Um [KVM](#) benutzen zu können, benötigt man zur Zeit noch eine modifizierte QEMU-Version, QEMU/KVM. Bemühungen zum Rückführen des modifizierten QEMU-Codes in den QEMU-Hauptzweig werden zur Zeit unternommen, sodass man in naher Zukunft zwischen den Modulen KQEMU oder [KVM](#) zur Virtualisierung von QEMU wählen kann. QEMU/KVM stehen unter verschiedenen Varianten der [GPL](#) und seit September 2008 steht mit Red Hat einer der größten Linux-Distributoren hinter der Entwicklung von QEMU/KVM. Ähnlich wie VirtualBox kann QEMU/KVM Festplattendateien, die mit VMware erstellt worden sind, verwenden.

Die Konfiguration von QEMU/KVM erfolgt ausschließlich durch Übergabe von Kommando-parameter beim Start des Programmes. Mittlerweile existieren auch diverse grafische Werkzeuge hierfür.

Um die Druckumgebung und das Abrechnungssystem nicht im produktiven System zu testen, wird in diesem Projekt, neben Xen, auch KVM benutzt, um die Software in einer virtuellen Umgebung zu installieren, zu konfigurieren und zu testen. Zur Zeit wird kvm-77 mit dem kvm-intel Modul des Kernels 2.6.27 auf einem Gentoo Linux Host verwendet, als Gäste fungieren Ubuntu 8.10, Debian [GNU/Linux 4.0r5 \(etch\)](#) und Debian [GNU/Linux 5.0 \(lenny\)](#) [[kvm08](#), [qem08](#), [wkv08](#), [wqe08](#)].

2.3. Kerberos

Da die Authentifizierung im PC7 über [Kerberos](#) erfolgt und im Rahmen des Projektes der Vorteil des Single-Sign-On zu tragen kommt, wird an dieser Stelle ein kleiner Überblick über [Kerberos](#) gegeben.

[Kerberos](#) ist ein verteilter Authentifizierungsdienst, der für offene und unsichere Computernetze entwickelt worden ist. [Kerberos](#) ist auch der Name der freien Softwarelösung des [Massachusetts Institute of Technology \(MIT\)](#), welches den Authentifizierungsdienst implementiert. Diese Implementierung dient vielfach als Referenzimplementierung, vor allem bei Linux-Distributionen. Der Name leitet sich vom Höllenhund [Kerberos](#) (Cerberus) aus der griechischen Mythologie ab, der den Eingang zur Unterwelt bewacht.

[Kerberos](#) bietet sichere und einheitliche Authentifizierung in einem ungesicherten TCP/IP-Netzwerk auf sicheren Hostrechnern. Die Authentifizierung übernimmt eine vertrauenswürdige dritte Partei, in diesem Fall ein besonders geschützter [Kerberos-5](#)-Netzwerkdienst auf einem Server, dem [Kerberos-Server](#). Der große Vorteil von [Kerberos](#) ist Single-Sign-On. Das bedeutet, dass sich ein Benutzer nur einmal anmelden muss, danach kann er alle verfügbaren Netzwerkdienste benutzen, ohne erneut ein Passwort eingeben zu müssen.

Bei der Authentifizierung sind drei Parteien beteiligt: der [Kerberos-Server](#), der Client und der Ziel-Server, also der Server, auf den der Client zugreifen will. Der Kerberos-Dienst verifiziert nicht nur den Client gegenüber dem Server (und vice-versa), sondern auch sich selbst gegenüber dem Client und dem Server, um einerseits z.B. Man-In-The-Middle-Angriffe zu unterbinden und andererseits die Identität des Clients und des Servers zu überprüfen.

Nach erfolgreicher Anmeldung des Clients auf dem Server erhält dieser ein sogenanntes *Ticket-Granting-Ticket (TGT)*. Mit dem *TGT* ist der Client nun berechtigt, weitere Tickets für die jeweiligen Dienste (z.B. telnet) anzufordern, ohne erneut ein Passwort eingeben zu müssen. Diese Dienste

müssen natürlich über **Kerberos**-Support verfügen und müssen dementsprechend konfiguriert worden sein.

Ein großer Nachteil von **Kerberos** ist die Voraussetzung der Verfügbarkeit des **Kerberos**-Servers. Ohne diesen kann sich niemand einloggen. Außerdem müssen alle Uhren im System synchronisiert sein, standardmäßig darf die Uhr eines Clients um max. fünf Minuten von der Zeit des **Kerberos**-Servers abweichen. Erfüllt ein Client diese Voraussetzung nicht, kann sich der Benutzer nicht einloggen. Problematisch hierbei ist, dass der Benutzer keine Rückmeldung erhält, weshalb der Anmeldevorgang gescheitert ist [krb03, wkb09].

2.4. Python

Python ist eine Skriptsprache, die mehrere Programmierparadigmen ermöglicht. Dies bedeutet, dass es dem Programmierer überlassen bleibt, das für die jeweilige Aufgabe am besten geeignete Paradigma zu wählen. Objektorientierte und strukturierte Programmierung werden vollständig unterstützt, zusätzlich gibt es Spracheigenschaften für funktionale und aspektorientierte Programmierung. Python wurde mit dem Ziel entworfen, möglichst einfach und übersichtlich zu sein. Das konsequente Verfolgen dieses Designziels führte dazu, dass Python eine Sprache ist, in der man schnell, einfach und leicht programmieren kann.

Der Name Python bezieht sich auf die britischen Komiker von *Monty Python* (z.B. Das Leben des Brian, Die Ritter der Kokosnuss) und nicht, wie man fälschlicherweise annehmen könnte, auf die Schlange. Python wird unter einer eigenen Lizenz, der *Python License*, vertrieben, die allerdings OSI zertifiziert¹¹ und in einigen Teilen sogar weniger restriktiv als die **GPL** ist. Aktuell ist die Version 2.6, erschienen am 1. Oktober 2008. Diese Version soll den Grundstein für einen sanften Übergang zu Python 3.0 legen, welches eine größere Überarbeitung der Sprache sein wird [kpy08, pyt08].

¹¹Siehe <http://www.opensource.org/licenses/pythonpl.php>

Python kommt mit relativ wenigen Schlüsselwörtern aus und die Syntax ist reduziert und auf Übersichtlichkeit optimiert. Dies führt dazu, dass Python eine Sprache ist, in der man schnell, einfach und leicht programmieren kann. Sie ist daher besonders dort geeignet, wo Übersichtlichkeit und Lesbarkeit des Codes eine herausragende Rolle spielen, z.B. bei Teamarbeiten, bei Beschäftigung mit dem Quelltext nach längeren Pausen oder bei Programmieranfängern. Besonders durch die Möglichkeit auch Programme anderer Sprachen als Modul einzubetten, werden viele Nischen in der Programmierung abgedeckt. So wird Python oft als Skriptsprache für ein anderes Programm eingesetzt, so auch z.B. bei OpenOffice.org, Gimp oder SPSS. Hinzu kommt, dass Python für alle größeren Betriebssysteme verfügbar ist, und von Haus aus mit einer großen Anzahl an Modulen installiert wird.

```
if a == 1:
    if b == 1:
        a = 42
else:
    b = 42
```

Abbildung 2.3.: Einrückungen sind das Strukturierungselement in Python.

Python benutzt Einrückungen als Strukturierungselement ([Abb. 2.3](#)). Damit unterscheidet es sich von den meisten anderen Programmiersprachen, bei denen Blöcke durch Klammern oder Schlüsselworte markiert werden und im Gegenzug Leerräume keine Semantik tragen. Bei diesen Sprachen ist die Einrückung zur optischen Hervorhebung eines Blockes zwar erlaubt und in der Regel auch erwünscht, aber nicht vorgeschrieben.

Da wxPython im Rahmen dieses Projektes Anwendung findet, soll an dieser Stelle kurz auf dieses Modul und die dahinter liegende Bibliothek eingegangen werden. wxPython ist ein Modul für die Integration von wxWindows in Python, einer GUI-Bibliothek, welche Windows, Mac OS und Unixes unterstützt. Ein großer Vorteil von wxWindows ist, dass es das darunter liegende Toolkit des Betriebssystems (oder Fenstermanager) respektiert und somit nicht wie eine „externe“ Applikation aussieht, sondern sich äußerlich anpasst

[pyt08, wpy08].

Da viele Teile von PyKota in Python geschrieben sind, kann es aus unterschiedlichsten Gründen vorkommen, dass Code gelesen oder adaptiert werden muss. Sollten Codeteile angepasst/optimiert werden, so werden diese gegebenenfalls Upstream geschickt (als Patches oä), um dem OSS Gedanken gerecht zu werden.

2.5. CUPS - PyKota

CUPS ist ein freies Drucksystem, ein **Daemon**, der das Drucken unter den verschiedenen unixoiden Betriebssystemen ermöglicht. CUPS wurde von Easy Software Products entwickelt und befindet sich jetzt im Besitz von Apple. Als Lizenz kann sowohl die **GPL** als auch eine proprietäre Lizenz verwendet werden.

CUPS besteht aus einer Client-Server-Architektur und lässt sich über CUPS-Backends beliebig anpassen. Ein CUPS-Server hat standardmäßig einen integrierten HTTP-Server. Über diesen Server, der standardmäßig auf Port 631 lauscht, erhält man Zugriff auf das Webinterface, welches die Administration des CUPS-Systems erlaubt. Neben dem Webinterface existieren eine Vielzahl an grafischen und konsolenbasierten Dienstprogrammen, mit welchen sich CUPS-Systeme auch über das Netzwerk steuern lassen. Als Basis für das Verwalten der Druckaufträge und Warteschlangen benutzt CUPS das **Internet Printing Protocol (IPP)**. Als Druckertreiber werden **PostScript Printer Description (PPD)** Dateien verwendet, welche die jeweiligen Druckeroptionen enthalten.

Der Hauptvorteil von CUPS gegenüber anderen Drucksystemen ist, dass es ein standardisiertes und modularisiertes System ist, welches eine Vielzahl von unterschiedlichen Daten auf dem Druckserver versteht (z.B. PDF-Dateien, PNG-Bilder, LaTeX-Texte). So wird plattformübergreifendes Arbeiten in heterogenen Netzwerken ohne großen Aufwand ermöglicht. Da IPP

auf HTTP 1.1 basiert, können alle Erweiterungen, die für HTTP vorgesehen sind, auch für IPP verwendet werden (z.B. [SSL/TLS](#)).

Wie eingangs erwähnt, lässt sich [CUPS](#) mit eigenen Backends anpassen. Backends sitzen am Ende der Verarbeitungskette, [CUPS](#) selbst benutzt Backends (z.B. [IPP-Backend](#), [USB-Backend](#)) um verschiedene Aufgaben erledigen zu können.

In der [CUPS](#) Dokumentation auf der [CUPS](#) Homepage befinden sich alle notwendigen Informationen, um mit Hilfe der verfügbaren [Application Programming Interface \(API\)](#) ein eigenes Backend oder einen eigenen Filter zu schreiben. Dieses kann entweder ein Shell-Skript, ein Perl-Programm oder ein ausführbares Programm sein. Damit [CUPS](#) dieses verwenden kann, muss es zunächst in den Ordner `/usr/lib/cups/backend` kopiert werden, eine symbolische Verknüpfung genügt hierfür auch. Danach muss [CUPS](#) noch mitgeteilt werden, welcher Drucker das Backend benutzen soll, indem in der Konfiguration dem Drucker das jeweilige Backend übergeben (eigentlich vorangestellt) wird, z.B. `cupsykota://<Drucker>` [[cup08](#), [wcu08](#)].

PyKota ist ein Softwarepaket für [CUPS](#), welches eine anteilmäßige Bilanzierung von Druckjobs ermöglicht. Das Programm ist komplett in Python geschrieben, wodurch der Code an individuelle Bedürfnisse schnell und einfach angepasst werden kann. Als [Datenbankmanagementsystem \(DBMS\)](#) können PostgreSQL, LDAP oder MySQL verwendet werden. Um Zugang zu den offiziellen Repositories bzw fertig gepackten Paketen und Dokumentation zu erhalten, verlangt der Autor für einen unlimitierten Zugang €25. Diese Unterstützung ist allerdings freiwillig, über *subversion* hat man kompletten Zugriff auf die aktuellste Version der Software. Die Installation muss im Normalfall, sofern kein Paket für die jeweilige Distribution vorliegt, händisch erfolgen. Für Debian [GNU/Linux 5.0 \(lenny\)](#) bzw Ubuntu gibt es mittlerweile automatisierte Installationsroutinen. PyKota steht unter der [GPL](#).

Es wird eine breite Palette von Druckern und Druckformaten unterstützt. Die Bilanzierung und Abrechnung der Druckjobs kann auf Hardware- un-

d/oder Softwareebene durchgeführt werden. Jeder Drucker, der seinen internen Seitenzähler und seinen aktuellen Status (printing/idle/error) über das [Simple Network Management Protocol \(SNMP\)](#) oder die [Printer Job Language \(PJI\)](#) nach außen hin kommunizieren kann, wird nativ von PyKota für die Abrechnung auf Hardwareebene unterstützt. Bei der Berechnung auf Softwareebene wird die Seitenanzahl des Druckauftrages von PyKota selbst berechnet, unabhängig vom Drucker. Voraussetzung für eine genaue Berechnung ist, dass der Drucker eines der folgenden Formate akzeptiert¹²:

- PostScript
- PDF
- DVI
- OpenDocument (ISO/IEC DIS 26300)
- Microsoft Word
- Plain text

PyKota kann den eigenen Bedürfnissen sehr stark angepasst werden. So können z.B. verschiedene Druckkontingente an Gruppen oder einzelne Benutzer vergeben werden. Die Abrechnung eines Druckauftrages kann per Seite, per Drucker, per Druckergruppe oder per Credit erfolgen und auf verschiedenste Art und Weise eingestellt werden. Credits sind individuell wählbar, dh man kann sich an der realen Währung orientieren, muss dies aber nicht zwangsläufig. Weiters kann eine maximale Größe (Seitenanzahl) des Druckjobs per Drucker eingestellt werden. Eine Authentifizierung zur Laufzeit, also bei der Abgabe des Druckauftrages, kann erfolgen, ebenso kann mit dem Benutzer kommuniziert werden, z.B. über den Status des Druckauftrages, mittels eines Programmes am Client. Das komplette System kann auch noch durch eigene Skripte erweitert werden, sofern dies notwendig erscheint.

Die Konfiguration von PyKota erfolgt durch Konfigurationsdateien. Mittels Werkzeuge, die über die Kommandozeile agieren, können Benutzer oder

¹²Vollständige Liste verfügbar auf: <http://www.pykota.com/software/pykota/features/>

Drucker hinzugefügt/gelöscht, Druckerlimits zurückgesetzt, oder Drucker-einstellungen geändert werden. Zusätzlich existiert ein Webinterface, welches, bei Vorhandensein eines Webservers auf dem Druckserver, diverse Statistiken ausgeben kann.

3. Implementierung von CUPS und PyKota

Das folgende Kapitel beschäftigt sich mit der Implementierung der zu verwendenden Software, **CUPS** in der Version 1.2.7-4**etch6** und **PyKota** in der Version 1.26+**hotfixes**. Als Testsystem wird primär eine virtuelle Debian **GNU/Linux** 4.0r5 (*etch*) Installation verwendet. Als Host-System fungiert am Anfang Gentoo Linux mit **kvm-78** auf dem zur Zeit aktuellsten Kernel (2.6.27), danach findet eine Überführung, und parallele Konfiguration, auf dem vom Institut zur Verfügung gestellten, virtuellen Server, einer Xen 3.0 Installation auf *etch*. Weiters erfolgen Installationen auf Ubuntu 8.10 und Debian **GNU/Linux** 5.0 (*lenny*), da die Pakete von *etch* zum Teil veraltet sind. Ubuntu hingegen hat aktuelle Pakete, weshalb man die fehlenden Pakete in den *etch*-backports¹ gezielter suchen kann. Das gesamte Kapitel wird sich allerdings ausschließlich auf die Installation unter *etch* beziehen, aufgrund des möglichen Wechsels der PC7-Infrastruktur auf *lenny* werden Abweichungen gesondert markiert.

Bei dem Drucker, der bei dieser Arbeit verwendet wird, handelt es sich um einen Lexmark Optra T614 s/w Laserdrucker mit Ethernetkarte. Dieser ist bereits im Netzwerk registriert als *pc7print.ai.wu-wien.ac.at*. Zusätzlich wird ein virtueller **PDF**-Drucker eingerichtet, um beim Testen des Abrechnungssystems nicht unnötig Papier zu verschwenden.

Um diese Arbeit nicht mit Screenshots von grafischen Paketverwaltungsprogrammen uä. zu durchsäen, und da auf der Xen-Installation kein **X**-

¹Ein Backport ist die Modifikation einer Software (Patch), die in eine ältere Version der gleichen Software eingepflegt wird.

Server installiert ist, werden die entsprechenden Kommandozeilenbefehle in schriftlicher Form dargestellt. Alle Befehle können als normaler Benutzer durchgeführt werden, mittels `Sudo` werden benötigte root-Rechte erlangt. Dies wird allerdings nicht gesondert angegeben werden, um den Lesefluss der Kommandos nicht zu stören.

Für die konkrete Installation von Debian GNU/Linux oder Ubuntu sei auf die Live-CDs im Netz verwiesen², die meistens über ein automatisches und grafisches Installationsprogramm verfügen. Dieses führt den Benutzer Schritt für Schritt durch den Installationsprozess. Die nachfolgenden Kapitel gehen von einer erfolgreichen Debian GNU/Linux Installation und verfügbaren Internetanschluss aus.

Nach der Installation wird auch auf die notwendigen Anpassungen der Software eingegangen sowie auf die allgemeinen Tests die durchgeführt worden sind. Den Abschluss des Kapitels macht die Überführung der Installation auf die Produktivumgebung mit allen dazugehörigen Tätigkeiten.

3.1. Installation und Konfiguration von CUPS

Als erstes Softwarepaket muss der CUPS-Server mit seinen Abhängigkeiten installiert werden (Abb. 3.1). Zusätzlich wird jetzt auch noch der, eingangs erwähnte, PDF-Drucker und der CUPS-Client installiert.

```
1 ~ $ apt-get update
2 ~ $ apt-get install cupsys cups-pdf
```

Abbildung 3.1.: Installation von CUPS und einem virtuellen pdf-Drucker

Optional können auch noch Zusatzpakete (Abb. 3.2) installiert werden wie z.B. `foomatic-filters`, welche die Einrichtung bekannter Drucker unter Linux vereinfachen.

²Debian - <http://www.debian.org/distrib/> | Ubuntu - <http://www.ubuntu.com/getubuntu>

```
3 ~ $ apt-get install cupsys-driver-gutenprint foomatic-filters foomatic-  
db-engine foomatic-db-gutenprint
```

Abbildung 3.2.: Optionale Treiber für CUPS

Die Werkseinstellungen der CUPS-Installation sind in Bezug auf Sicherheit sehr gut. Um die Sicherheit zu erhöhen, wird die Verschlüsselung für die Authentifizierung aktiviert (Abb. 3.3). Mit einem Neustart werden die neuen Einstellungen übernommen und geladen (`/etc/init.d/cupsys restart`).

```
--- /etc/cups/cupsd.conf.old 2008-11-08 18:06:22.000000000 +0100  
+++ /etc/cups/cupsd.conf 2008-11-08 18:06:33.000000000 +0100  
@@ -23,7 +23,6 @@  
  
# Default authentication type, when authentication is required...  
DefaultAuthType Basic  
+DefaultEncryption IfRequested  
  
# Restrict access to the server...  
<Location />
```

Abbildung 3.3.: Optionale Verschlüsselung für die Authentifizierung

Mittels eines Browsers kann nun das Webinterface von CUPS auf <http://localhost:631> aufgerufen werden (Abb. 3.4). Aufgrund der Standardeinstellungen ist dies nur vom localhost aus möglich. Dies kann allerdings in der Konfigurationsdatei angepasst werden, sodass ein Zugriff auch von bestimmten anderen Rechnern (z.B. Administrator) aus möglich ist. Über dieses Webinterface können in Zukunft alle Einstellungen vorgenommen, Drucker eingerichtet und Druckjobs kontrolliert werden.

Als nächstes wird der virtuelle PDF-Drucker über das Webinterface eingerichtet. Im Administrationsteil (Abb. 3.5) werden von CUPS neu erkannte Drucker angezeigt, weshalb ein Klick auf den Knopf neben „PDF Printer (Virtual Printer)“ genügt, um den Installationsprozess für den PDF-Drucker zu starten. Als *make* wird *Postscript* ausgewählt, als *Model/Driver* der *Ge-*

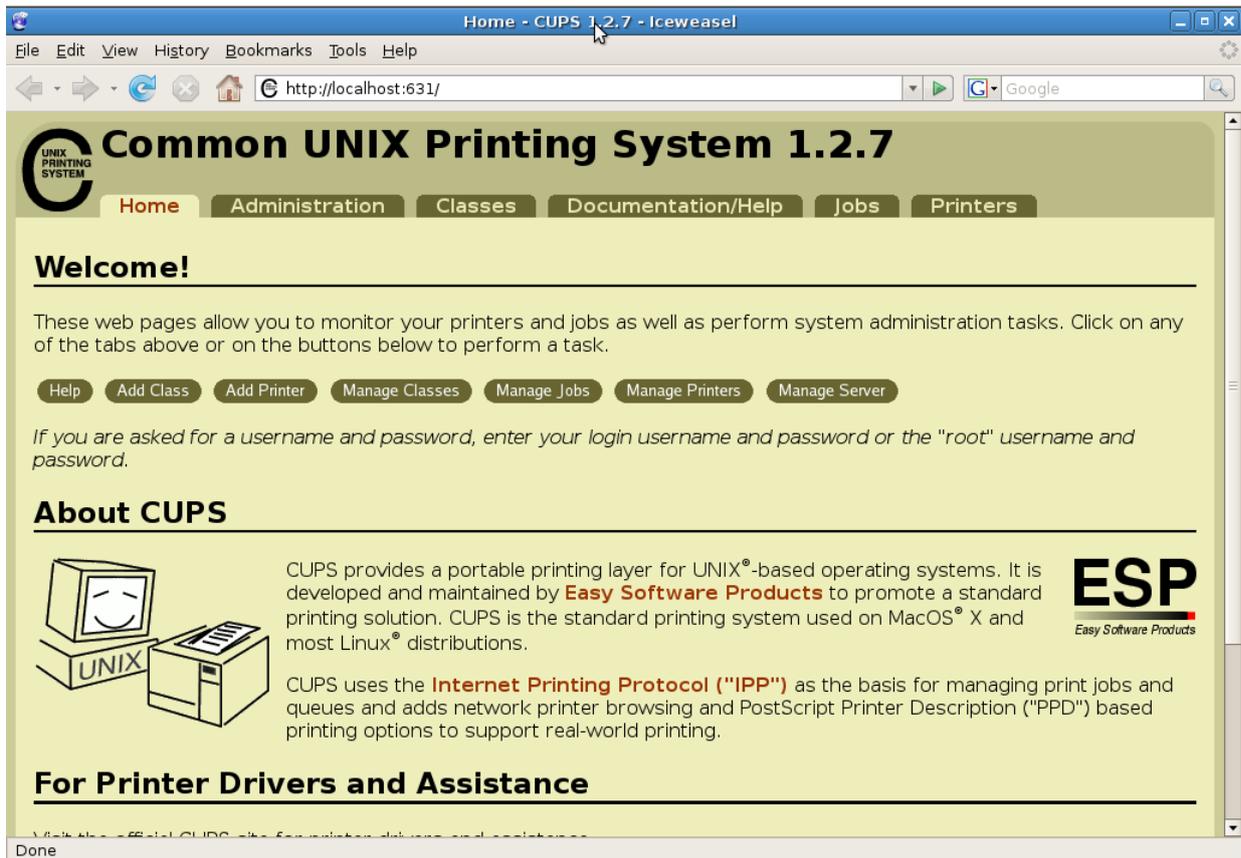


Abbildung 3.4.: Startbildschirm des CUPS-Webinterfaces

neric postscript color printer rev4 (en). Da nur CUPS-Administratoren Einstellungen verändern können, ist es erforderlich, zumindest einen Druckeradministrator einzurichten und diesen der Gruppe *lpadmin*, welche von CUPS bei der Installation automatisch erstellt worden ist, hinzuzufügen (`adduser <BENUTZER> lpadmin`). *root* ist standardmäßig der einzige Druckeradministrator. Nach einer erforderlichen Authentifizierung ist der PDF-Drucker eingerichtet und einsatzbereit. Mit derselben Prozedur wird auch der Netzwerkdrucker hinzugefügt. Als Device Uniform Resource Identifier (URI) wird `ipp://pc7print.ai.wu-wien.ac.at:631/printers/pc7print` eingegeben, sowohl bei *make* als auch bei *Model/Driver* kann, wie beim PDF-Drucker, *Postscript* ausgewählt werden. Allerdings gibt es auch zwei verschiedene Treiberdateien in Form von PPDs für den Drucker, zu finden auf <http://>

www.lexmark.com im Downloadbereich von Lexmark und auf <http://www.openprinting.org>, einer Webseite der Linux Foundation. Beide Treiber bieten ähnliche Funktionalitäten. Der Autor dieser Arbeit hat sich für den Offiziellen entschieden, letzten Endes bleibt es dem Administrator frei für welchen Treiber er sich entscheidet. Für die Testphase sollte der Einfachheit halber der **PDF-Drucker** als Standarddrucker ausgewählt werden.



Abbildung 3.5.: CUPS Administrationsseite

Da der **CUPS-Client** bereits am Anfang installiert worden ist, können nun Dokumente zum Testen lokal ausgedruckt werden. Bei grafischen Programmen in **X** erfolgt dies durch den Menüunterpunkt *Drucken* in den Programmmenüs. Um über die Konsole zu drucken kann der Befehl **lp** verwendet werden. Das Ergebnis wird in beiden Fällen im Ordner *PDF* im Home-Verzeichnis des Benutzers gespeichert.

3.2. Implementierung von PyKota

Nachdem CUPS installiert und eingerichtet worden ist, wird PyKota installiert. Um nicht auf die instabile Version über Subversion (SVN) zugreifen zu müssen, die zwar auf einer Testinstallation einwandfrei funktioniert hat, war der Autor so frei, die erforderlichen €25 zu bezahlen, um Zugriff auf die stabile Version und die Dokumentation zu erhalten. Die erforderlichen Logindaten gehen bei Übergabe des Projektes in den Besitz des Instituts über.

Zur aktuellen Version von Pykota, v1.26, gibt es offizielle Debian Pakete für *etch*. Da aber eine Installation über den Paketmanager nicht sehr aufschlussreich ist, sich nicht sehr gut dokumentieren lässt, und es bereits Hotfixes für v1.26 gibt, wird der offizielle Tarball inkl. Hotfixes als Installationsdatei gewählt. Dieser ist im privaten Downloadbereich auf <http://www.pykota.com/> zu finden.

Direkt nach dem Entpacken empfiehlt es sich, die Abhängigkeiten zu überprüfen und das Ergebnis für eine spätere Verarbeitung zu speichern (Abb. 3.6). Je nach verwendeter Software und gewünschter Konfiguration müssen einige Abhängigkeiten zwingend aufgelöst werden, z.B. das Python Modul, welches das Interface für das jeweilige DBMS bereitstellt. Weiters sollten alle Abhängigkeiten die als *mandatory* gekennzeichnet sind, vor dem Beginn der Installation aufgelöst werden. Unter *lenny* sind alle erforderlichen Pakete über die normalen Debian-Repositories zu beziehen, unter *etch* muss das Paket *pkpgcounter* manuell installiert werden³. Eine Auflistung der installierten Abhängigkeiten auf dem Testsystem befindet sich im Anhang.

Nach der Überprüfung kann mit der Installation begonnen werden. Hierfür stehen zwei Möglichkeiten zur Verfügung. Einerseits eine halbautomatisierte Installation (Abb. 3.7) für Debian GNU/Linux und Ubuntu, andererseits gibt es auch die Möglichkeit einer manuellen Installation. Die halbautomatisierte Installation ist bei der ersten Installation zu bevorzugen, da sie einen

³Erhältlich auf <http://www.pykota.com>.

```

1 ~ $ tar -xf pykota-1.26_fixes_official.tar.gz
2 ~ $ mv pykota-1.26_fixes_official pykota && cd pykota
3 pykota $ python checkdeps.py > ~/deps_start.txt
4 pykota $

```

Abbildung 3.6.: Entpacken von PyKota v1.26 und Überprüfung der Abhängigkeiten

erheblichen Zeitvorteil gegenüber der Manuellen hat. Außerdem löst sie die benötigten Abhängigkeiten, allerdings nur in *lenny*, automatisch auf. Als Nachteil erweist sich die zur Zeit starre Art und Weise der Installationsroutine. So besteht z.B. keine Wahlmöglichkeit zwischen den unterstützten **DBMS**. PostgreSQL wird automatisch konfiguriert, ebenfalls wird ein nicht mehr verfügbares Softwarepaket angeboten, welches allerdings nicht benötigt wird.

```

pykota $ bin/pksetup debian
The following command will be launched to install PyKota files on your
system :
python ~/pykota/setup.py install
Do you agree ? y

running install
...
What is your DNS domain name ? ai.wu-wien.ac.at
What is the hostname or IP address of your SMTP server ?
...
The following command will be launched to import the Virtual_Printer
print queue into PyKota's database and reroute it through PyKota :
pkprinters --add --cups --description "Printer created with pksetup" "
Virtual_Printer"
Do you agree ? y
Creation...
Done. Average speed : 4.62 entries per second.

```

Abbildung 3.7.: Installationsassistent für Debian - stark verkürzt

In weiterer Folge wird detailliert auf die manuelle Installation eingegangen. Als **DBMS**-Lösung hat sich der Autor für PostgreSQL entschieden, da diese

eine einfachere Möglichkeit zur Sicherung der Anbindung anbietet und die installierte Version von MySQL noch keine Subqueries unterstützt. Unter der Annahme, das sich Datenbank- und Druckserver auf demselben Host befinden, wird nun die PostgreSQL-Datenbank für Pykota eingerichtet. Den Anfang macht die Beschränkung der Zugriffsberechtigungen auf die Pykota-Datenbank und die Verschlüsselung der Passwörter bei der Übermittlung mittels MD5 (Abb. 3.8)⁴. Nach der Änderung muss der PostgreSQL-Server neu gestartet werden, damit die Änderungen wirksam werden.

```
5 pykota $ echo "local pykota pykotaadmin,pykotauser___md5" >> /etc/  
    postgresql/*/pg_hba.conf  
6 pykota $ /etc/init.d/postgresql restart
```

Abbildung 3.8.: Beschränkung der Zugriffsberechtigungen auf die Pykota-Datenbank

Als nächstes wird die Datenbank mit Hilfe eines Templates für PostgreSQL erstellt (Abb. 3.9). Dies kann nur von einem Benutzer mit ausreichenden Rechten für die Erstellung von PostgreSQL-Datenbanken erfolgen (Standardeinstellung: postgres). Da bei diesem Vorgang auch die Benutzernamen und Passwörter festgelegt werden, sollten diese vorher angepasst werden (Abb. 3.10). Eine spätere Änderung der Benutzernamen und Passwörter ist mit der bei PostgreSQL üblichen Möglichkeiten durchführbar. Je nach verwendeter DBMS existieren auch Templates für MySQL, SQLite und LDAP, welche ähnlich angepasst werden können.

Da die Abhängigkeiten bereits am Anfang des Installationsprozesses aufgelöst worden sind, kann nun mit der eigentlichen Installation von Pykota begonnen werden (Abb. 3.11). Danach müssen der Systembenutzer und die Gruppe *pykota* mit dem Home-Verzeichnis `/etc/pykota` erstellt und die Beispiel-Konfigurationsdateien nach `/etc/pykota` kopiert werden. Im Anschluss daran kann man mit der Konfiguration von Pykota beginnen. Die Konfigurationsdateien sind sehr stark kommentiert, weshalb man die Konfiguration zügig vornehmen kann. Die angepassten Konfigurationsdateien

⁴Die unterstrichenen Abstände sind Tabulatoren, keine Leerzeichen.

```

7 pykota $ cd initscripts/postgresql
8 postgresql $ psql -h localhost -U postgres template1
9 Welcome to psql, the PostgreSQL interactive terminal.
10
11 Type: \copyright for distribution terms
12       \h for help with SQL commands
13       \? for help on internal slash commands
14       \g or terminate with semicolon to execute query
15       \q to quit
16
17 template1=# \i pykota-postgresql.sql
18 ...
19 pykota=# \q

```

Abbildung 3.9.: Erstellung der PostgreSQL PyKota-Datenbank

befinden sich im [Anhang](#). Zu beachten ist weiters, dass der Benutzername und das Passwort für den [DBMS-Zugriff](#) in `pykotadmin.conf`, an die bei der Erstellung der Datenbank verwendeten, angepasst werden **müssen**.

Um die Sicherheit des Abrechnungssystems zu erhöhen, werden die Zugriffsberechtigungen der Konfigurationsdateien angepasst ([Abb. 3.12](#)), sodass nur noch Administratoren, [CUPS](#) und PyKota im Lesemodus, auf diese zugreifen kann. Für den Fall dass [CUPS](#) nicht als root läuft, sondern z.B. als Benutzer *lp*, muss sichergestellt werden, dass dieser Benutzer ebenfalls Leseberechtigungen erhält. Der einfachste Weg ist es, den Benutzer in die *pykota*-Systemgruppe aufzunehmen (`adduser lp pykota`). Drei Punkte sollte man bei der Rechtevergabe beachten:

- Der Benutzer, unter dessen Namen [CUPS](#) läuft, muss die PyKota-Konfigurationsdateien lesen können.
- Jeder Benutzer der `pykotadmin.conf` lesen kann, wird als PyKota-Administrator betrachtet, dh er hat uneingeschränkten Zugriff auf die Datenbank und kann alles tun.
- Sollte `cupsd.conf` `RunAsUser` enthalten, ist keine Authentifizierung mittels *pknotify* und *pykoton* möglich. Sollte dies erforderlich sein, müs-

```

--- a/pykota-postgresql.sql 2007-01-17 23:19:42.000000000 +0100
+++ b/pykota-postgresql.sql 2008-11-09 01:53:49.000000000 +0100
@@ -36,8 +36,8 @@
-- Create the print quota database users
-- NOTE: Change the password values to the passwords you would like.
--
-CREATE USER pykotauser WITH UNENCRYPTED PASSWORD 'readonlypw'
  NOCREATEDB NOCREATEUSER;
-CREATE USER pykotaadmin WITH UNENCRYPTED PASSWORD 'readwritepw'
  NOCREATEDB NOCREATEUSER;
+CREATE USER benutzer WITH UNENCRYPTED PASSWORD '123456abcd' NOCREATEDB
  NOCREATEUSER;
+CREATE USER pc7printadmin WITH UNENCRYPTED PASSWORD 'GanzGeheimesPwd'
  NOCREATEDB NOCREATEUSER;

--
-- Now connect to the new database

```

Abbildung 3.10.: Anpassen der Benutzernamen/Passwörter für die PyKota-Datenbank unter PostgreSQL

sen die Besitzer der PyKota-Konfigurationsdateien und des [CUPS](#)-Prozesses dieselben sein.

Die Passwörter sind im Klartext in `pykotadmin.conf` gespeichert, was ein potentielles Sicherheitsrisiko darstellt, sollte jemand aufgrund eines Lecks Rechte erlangen, diese Datei lesen zu können. Um dem vorzubeugen, könnte das Passwort als [MD5](#)-Hash gespeichert werden. Dies erfordert, dass bei der Erstellung der Datenbank ([Abb. 3.9](#)) die Passwörter ebenfalls als [MD5](#)-

```

20 postgresql $ cd ~/pykota
21 pykota $ python setup.py install
22 pykota $ adduser --system --group --home /etc/pykota --gecos PyKota
  pykota
23 pykota $ cp conf/pykota.conf* ~/pykota/pykota.conf
24 pykota $ cp conf/pykotadmin* ~/pykota/pykotadmin.conf

```

Abbildung 3.11.: PyKota Installation und Erstellung des Systembenutzers pykota

```
25 pykota $ chown -R pykota.pykota ~pykota/  
26 pykota $ chmod 750 ~pykota/  
27 pykota $ chmod 640 ~pykota/pykota.conf  
28 pykota $ chmod 640 ~pykota/pykotadmin.conf
```

Abbildung 3.12.: Anpassen der Berechtigungen der PyKota Konfigurationsdateien

Hash gespeichert werden und die Übermittlung im Klartext erfolgt (damit nicht ein Hash des Hashes erzeugt und übermittelt wird). Dies wird allerdings nicht vorgenommen, da ein korrektes Funktionieren der Berechtigungen Voraussetzung für ein funktionierendes Linuxsystem ist.

Die Installation von PyKota ist beinahe abgeschlossen. Der letzte Schritt ist die Integration von PyKota in **CUPS** über das *cupspykota*-Backend. Dies geschieht durch Setzen eines symbolischen Links in das Backendverzeichnis von **CUPS** und Anpassen der Berechtigungen des Backends ([Abb. 3.13](#)).

```
29 pykota $ cd /usr/lib/cups/backend  
30 backend $ ln -s /usr/share/pykota/cupspykota cupspykota  
31 backend $ chmod 700 /usr/share/pykota/cupspykota
```

Abbildung 3.13.: Anpassen der Berechtigungen

Damit die Einstellungen übernommen werden, muss **CUPS** neugestartet werden (`/etc/init.d/cupsys restart`). Die Option beim Hinzufügen neuer Drucker im Administrationsteil von **CUPS** wurde durch alle von PyKota erkannten Drucker erweitert. Fügt man nun dem System neue Drucker hinzu, so kann man direkt die durch PyKota gemanagten Geräte benutzen. Damit bestehende Drucker ihre Druckaufträge über das *cupspykota*-Backend erhalten, müssen die jeweiligen Backend-URIs angepasst werden. Dies kann über das **CUPS**-Webinterface erfolgen, aber auch direkt in der **CUPS** Konfigurationsdatei ([Abb. 3.14](#)), welche die Informationen der Systemdrucker enthält, oder mittels des PyKota-Kommandos `pkprinters -C <Drucker1> <Drucker2> <DruckerN>`. Nach einem letzten Neustart von **CUPS** (`/etc/init.d/cupsys restart`) ist das System komplett instal-

liert. Einsatzbereit ist es noch nicht, da weder Benutzer- noch Drucker in den entsprechende PyKota-Datenbanken angelegt worden sind. Dies erfolgt im nächsten Kapitel.

```
32 pykota $ grep DeviceURI /etc/cups/printers.conf
33 DeviceURI ipp://pc7print.ai.wu-wien.ac.at:631/printers
34 DeviceURI cups-pdf:/
35 pykota $ sed 's/DeviceURI /DeviceURI cupspykota:\\\\//g' /etc/cups/
    printers.conf > /tmp/printers-conf && mv /tmp/printers-conf /etc/cups/
    /printers.conf
36 pykota $ grep DeviceURI /etc/cups/printers.conf
37 DeviceURI cupspykota://ipp://pc7print.ai.wu-wien.ac.at:631/printers
38 DeviceURI cupspykota://cups-pdf:/
```

Abbildung 3.14.: Alle Drucker über cupspykota umleiten

3.3. Testen der Druckumgebung im virtuellen System

Um zu überprüfen, ob die Installation von **CUPS** und PyKota erfolgreich war und ob das Zusammenspiel der Programme einwandfrei funktioniert, werden nun einige Testdrucke durchgeführt. Hauptsächlich wird hierbei der **PDF-Drucker** verwendet, allerdings wird auch der Lexmark angesprochen werden, um dessen Konfiguration und Funktionalität zu überprüfen.

Da beide Drucker bereits über PyKota umgeleitet werden (**Abb. 3.14**), ist es erforderlich, die Drucker der PyKota-Datenbank hinzuzufügen, einen Testbenutzer anzulegen und diesem Benutzer das Drucken auf den Zieldruckern zu erlauben. Dies erfolgt über die Kommandos **pkusers**, **pkprinters** und **edpykota**. Gleichzeitig werden die Kosten pro Seite pro Drucker, das Guthaben des Benutzers und die Quoteneinstellungen des Benutzers auf den Druckern über Parametereingaben eingestellt (**Abb. 3.15**). Nachfolgend erfolgt eine Erklärung der übergebenen Parameter, genauere Informationen zu

den Kommandos sind in den Manpages oder über den Parameter `--help` zu finden.

- `pkprinters --charge 0.1 pdf pc7print`
Setzt den Preis pro Seite auf 0.1 auf den Druckern `pdf` und `pc7print`.
- `pkusers --add --limitby balance --balance 50.0 j0550574`
Erstellt den Benutzer `j0550574` mit einem Startguthaben von 50.0 und setzt als Limit für Druckaufträge das Guthaben. Dh wenn die Kosten des Druckauftrages das bestehende Guthaben überschreiten, wird der Druckauftrag nicht durchgeführt.
- `edpykota --add --noquota j0550574`
Setzt das erlaubte Druckkontingent für den Benutzer `j0550574` auf allen Druckern auf *unbegrenzt*.

```
1 ~ $ pkprinters --charge 0.1 pdf pc7print
2 ~ $ pkusers --add --limitby balance --balance 50.0 j0550574
3 ~ $ edpykota --add --noquota j0550574
```

Abbildung 3.15.: Hinzufügen der Drucker und Benutzer zur PyKota-Datenbank

Sollten alle Kommandos erfolgreich durchgeführt worden sein, so ist nun das Drucken möglich. In einer Shell kann, mit dem Kommando `lp`, jedes beliebige Dokument über den Standarddrucker gedruckt werden. Da dies der PDF-Drucker ist, befinden sich die nun gedruckten Dokumente als PDF Dateien im Ordner `PDF` im Home-Verzeichnis des Benutzers `j0550574` (Abb. 3.16).

```
4 ~ $ lp testdok.txt
5 Auftrags ID ist pdf-58 (1 Datei(en))
6 ~ $ ls -lhG PDF/
7 -rw----- 1 j0550574 47K 2008-11-10 11:34 PDF/_testdok_txt_.pdf
```

Abbildung 3.16.: Drucken eines Testdokumentes auf den pdf-Drucker

Auch der Lexmark Drucker kann auf diese Weise getestet werden. Hierzu muss der Drucker als Parameter an `lp` übergeben werden (`lp -d pc7print`).

Da das Dokument 21 Seiten lang war und zweimal gedruckt worden ist, müsste sich das Guthaben des Benutzers um 4.2 verringert haben. Um dies zu überprüfen, kann der Benutzer mit `pkusers --list` sein aktuelles Guthaben abrufen (Abb. 3.17).

```
8 ~ $ pkusers --list
9 j0550574 - <j0550574@wu-wien.ac.at>
10   Limited by : balance
11   Account balance : 45.80
12   Total paid so far : 50.0
13   Overcharging factor : 1.00
```

Abbildung 3.17.: Überprüfen des Guthabens

3.4. Anpassen der Druckumgebung an die Anforderungen

Dieses Kapitel geht auf die konkreten Tätigkeiten ein, welche vorgenommen worden sind, um die installierte Software an die zu Anfang definierten Anforderungen anzupassen.

Kommunikation PyKota - Endbenutzer

Für die Kommunikation zwischen PyKota und den Studenten wurde ein Client-Server-Programm gewählt, `pknotify` und `pykoticon`. Während `pknotify` mit PyKota installiert worden ist, muss `pykoticon` extra von der PyKota Homepage heruntergeladen und installiert werden⁵. `PyKotIcon` ist ein plattformunabhängiger, generischer, netzwerkbasierender Dialogbox Manager, auf dessen konkrete Funktionen nicht näher eingegangen wird. Es sollte allerdings nicht unerwähnt bleiben, dass `PyKotIcon` auch von anderen Anwendungen genutzt werden kann.

⁵Erhältlich auf <http://www.pykota.com/software/pykoticon/>.

PyKotIcon lauscht standardmäßig auf dem TCP Port 7654 und sitzt in der Taskbar, sofern der Student in X, dh KDE oder GNOME, einloggt. `pknotify` wird von `cupspykota` aufgerufen und schickt die Nachrichten, durch von PyKota generierte temporäre Prozessparameter, an `pykoticon`, welches die Nachricht als Pop-up anzeigt. In der Konfigurationsdatei von PyKota (siehe [Anhang](#)) befindet sich eine Auflistung aller Parameter sowie der verwendeten Aufrufe.



Abbildung 3.18.: Das neue TaskbarIcon von PyKotIcon

Vor der Installation müssen noch die Abhängigkeiten von `pykoticon` aufgelöst werden. Die meisten sind bereits auf dem System enthalten, lediglich das Python Modul für wxWindows (`python-wxgtk2.6`) und wxWindows müssen nachinstalliert werden. Danach kann mit der Installation von `pykoticon` begonnen werden. Während der Tests ist der Autor auf einen Anzeigefehler gestoßen, welcher mit einem kleinen Patch⁶ behoben werden kann. Außerdem wurde das TaskbarIcon verschönert ([Abb. 3.18](#)). Es empfiehlt sich, den Patch und das neue Icon vor der Installation in den entpackten Ordner zu kopieren, PyKotIcon zu patchen und danach zu installieren ([Abb. 3.19](#)).

```
1 ~ $ apt-get install python-wxgtk2.6
2 ~ $ tar -xf pykoticon-1.02.tar.gz
3 ~ $ patch pykoticon-1.02/bin/pykoticon pykoticon.patch
4 ~ $ cp new_pykoticon-green.ico pykoticon-1.02/icons/pykoticon-green.ico
5 ~ $ python pykoticon-1.02/setup.py install
```

Abbildung 3.19.: Installation von PyKotIcon

Damit PyKotIcon automatisch beim Einloggen in die Desktopumgebung gestartet wird, muss eine Autostartverknüpfung ([Abb. 3.20](#)) erstellt werden. Da GNOME und KDE dieselbe Syntax, aber zur Zeit noch unterschiedliche Speicherorte verwenden (GNOME: `~/.config/autostart` / KDE:

⁶Genauerer zum Patch im [Anhang](#).

~/kde/Autostart), muss eine symbolische Verknüpfung der Datei vom **GNO-** **ME-** zum **KDE-**Verzeichnis erstellt werden. Damit PyKotIcon die Nachrichten des Druckservers empfängt, muss der Druckserver beim Start von PyKotIcon angegeben werden (`pykoticon xmprint`).

```
1 ~ $ less pykoticon.desktop
2 [Desktop Entry]
3 Type=Application
4 Encoding=UTF-8
5 Version=1.0
6 Name=PyKotIcon
7 Exec=pykoticon
8 ~ $ cp pykoticon.desktop ~/.config/autostart
9 ~ $ cd .kde/Autostart
10 Autostart $ ln -s ../../.config/autostart/pykoticon.desktop .
```

Abbildung 3.20.: Autostartverknüpfungen für PyKotIcon

Anlegen neuer Benutzer - Kauf von Guthaben - Ausdruck eines Beleges

Das Hinzufügen neuer Benutzer in PyKota kann auf verschiedene Arten erfolgen:

- Kommandoeingabe auf dem Druckserver (Zugriff via [SSH](#))
- Ausführen eines Skriptes (Zugriff via [SSH](#))
- Aufrufen einer Webseite

Mit [SSH](#) kann direkt, sowohl auf die Benutzer-, als auch auf die Druckerdatenbank, zugegriffen werden. Mit `pkusers`, `pkprinters` und `edpykota` ist jede Veränderung möglich. Zusätzlich zur direkten Manipulation wurde vom Autor ein kleines Skript geschrieben, welches mit `mykota` ausgeführt wird und interaktiv nach den erforderlichen Parametern fragt. Es kann

neue Benutzer hinzufügen und die Guthaben bestehender erhöhen. Entweder bleibt es im `home`-Verzeichnis des ausführenden Benutzers und wird mit `sudo ./mykota` ausgeführt. Oder aber es wird nach `/usr/bin` kopiert, mit `chmod u+x,g+x mykota` ausführbar gemacht für den Besitzer und die Gruppe, sodass der ausführende Benutzer das Skript direkt aufrufen kann, so wie jeden anderen Befehl auch (Abb. 3.21).

```
~ $ mykota
Dieses Skript soll durch die Erstellung
der Benutzer für das Drucksystem führen!

!!!Ausschliesslich mit sudo ausführen!!!

Bitte die entsprechenden Daten eingeben
und mit "Enter" bestätigen (STRG+C zum Abbrechen)!

#####
Benutzername (PC7-Login): j0550574
eingezahltes Guthaben: 20
#####
Wollen Sie "j0550574" mit 20 Guthaben anlegen? (j/n)j
Creation...
Done. Average speed : 3.63 entries per second.
Benutzer "j0550574" wurde mit 20 Guthaben erfolgreich angelegt.
```

Abbildung 3.21.: mykota - Interaktives Skript für PyKota

Neben den soeben genannten Optionen ist es ebenfalls möglich, Benutzer über eine Webseite dem System hinzuzufügen. Dies erfordert allerdings besondere Sicherheitsmaßnahmen. Dies geschieht einerseits durch Authentifizierung auf der Webseite durch integrierte Apache-Module und andererseits durch Beschränken des Zugriffs auf bestimmte Rechner, in diesem speziellen Fall auf die IP des Rechners des Sekretariats des Instituts. Damit die Seite allerdings die erforderlichen Befehle `pkusers` und `edpykota` ausführt, bedarf es einiger kleiner Änderungen. Der Webserver Apache wird standardmäßig aus Sicherheitsgründen nicht als `root` sondern als Benutzer/Gruppe `www-data` ausgeführt. Dadurch werden Befehle standardmäßig als dieser Be-

nutzer ausgeführt, welcher allerdings keine PyKota Befehle ausführen kann. Um dies zu ermöglichen können verschiedene Ansätze in Betracht gezogen werden wie z.B.

- Ändern der Gruppe/des Besitzers der PyKota Befehle auf `www-data`
- Anpassen der Zugriffsberechtigungen von `pkusers` und `edpykota`
- `www-data` der `pykota` Gruppe hinzufügen
- Ändern des ausführenden Apache-Benutzers in `pykota` oder `root`
- `www-data` erlaube mittels `Sudo` die PyKota Befehle auszuführen
- Erstellen eines virtuellen Hosts welcher cgi-Skripte mittels `Suexec` als ein anderer Benutzer ausführen kann

Von diesen sechs Ansätzen ist der letzte der sicherste und effektivste Ansatz, sowohl für das System, als auch für PyKota. Die ersten zwei sollten überhaupt nicht in Betracht gezogen werden aufgrund ihrer augenscheinlichen Schwächen für die Integrität von PyKota. Von der Möglichkeit, direkt in die Datenbank zu schreiben wurde abgesehen, da hier einerseits der Benutzername und das Passwort erneut in einem Dokument vorhanden wären, andererseits könnte nicht gewährleistet werden, das die Tabellen- und Spaltenbezeichnungen mit zukünftigen Versionen übereinstimmen.

Um `Suexec` verwenden zu können, sollte ein eigener Virtueller Host erstellt werden und die ausführenden Dateien müssen dieselbe Benutzer/Gruppe-ID haben, wie `Suexec`.

Nach Rücksprache mit dem Systemadministrator ist der dritte Ansatz implementiert worden. Mit einer in Python geschriebenen CGI-Seite können Benutzer hinzugefügt und das Guthaben bestehende Benutzer erhöht werden (`addthem.cgi`). Zugleich ist sie die einzige Möglichkeit, eine Rechnung anzuzeigen und auszudrucken ([Abb. 3.22](#)).

Der Kauf von Guthaben erfolgt ähnlich wie das Erstellen neuer Benutzer. Es können sowohl das Skript, als auch die Webseite verwendet werden. Darüber



Institut für Informationswirtschaft
Wirtschaftsuniversität Wien, UZA II 3. Ebene
Augasse 2-6, A-1190 Wien



Rechnung für eingezahltes Druckguthaben

Benutzername: j1234567
eing. Guthaben: € 10.0 (0% USt)

Danke!
Wien, am 03.12.2008

Abbildung 3.22.: Aussehen der Rechnung

hinaus kann mit beiden Methoden den Benutzern Guthaben auch wieder abgezogen werden.

Abrufen des Guthabens

Die Benutzer können ihr Guthaben auf zwei verschiedene Arten überprüfen. Einerseits werden sie bei jedem Druckauftrag um eine Bestätigung desselben gebeten, bei der, neben den Kosten, auch das aktuelle Guthaben angezeigt wird (Abb. 3.23). Andererseits können sie dieses auch über die, von PyKota ausgelieferten, CGI-Seite, welche ebenfalls in Python geschrieben ist, aufrufen. Voraussetzung hierfür ist, dass auf dem Druckserver ebenfalls ein Webserver (z.B. Apache) läuft. Um die Sicherheit der Druckumgebung zu erhöhen, sollte der Zugriff auf diese Seite beschränkt werden (z.B. auf den Adressbereich des PC7).

Die Seite kann über `<URL>/printquota.cgi` aufgerufen werden, das Aussehen der Seite (z.B. Logo, Link) wird über Parameter in `pykota.conf` gesteuert.

Damit die Studenten, wenn sie ihr Guthaben bzw. ihre Statistiken auf `http://xmprint/cgi-bin/printquota.cgi` abrufen, sich nicht erneut authentisieren müssen, wird der Apache HTTP Server (Apache) so eingerich-

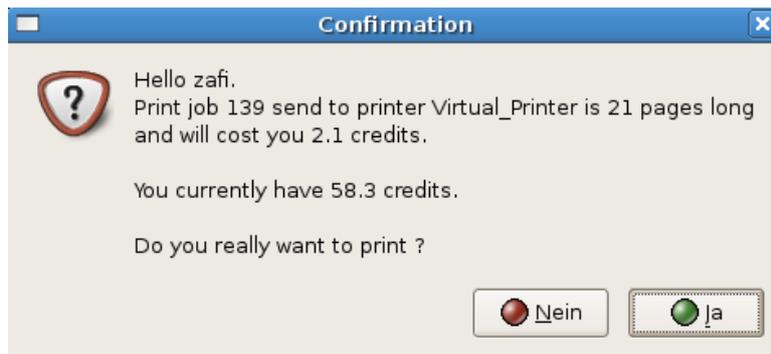


Abbildung 3.23.: Aktuelles Guthaben bei Bestätigung des Druckauftrages (PyKotIcon)

tet, das er mit [Kerberos](#) arbeiten kann. Hierfür muss das [Apache](#)-Modul `mod_auth_kerb` installiert und der Webserver eingerichtet werden.

Die automatische Authentisierung funktioniert allerdings nur mit auf Gecko basierenden Browsern wie z.B. Firefox/Iceweasel, da diese eine voll funktionsfähige [Generic Security Services Application Program Interface \(GSSAPI\)](#) Implementierung besitzen. Bei anderen Browsern (z.B. Konqueror, Opera, Safari, ...) sind in den meisten Fällen Addons bzw. erweiterte Einstellungen für ein reibungsloses Funktionieren mit der [GSSAPI](#) verantwortlich, welche standardmäßig nicht verfügbar oder aktiviert sind. In solchen Fällen erfolgt eine normale Abfrage zur Authentisierung.

Die eingegebenen Daten werden über das [Kerberos](#)-Modul des [Apache](#) an den [Kerberos](#)-Server weitergereicht, welcher den Benutzer authentifiziert und den Zugriff auf die Seite gestattet. Ein authentifizierte Benutzer kann ausschließlich sein eigenes Konto abfragen, lediglich der Benutzer *root* und PyKota-Administratoren können alle relevanten Informationen abfragen.

Sollte [SSH](#) Zugriff auf den Druckserver möglich sein, so können eingeloggte Benutzer mit `pkusers --list` immer ihr eigenes Guthaben abrufen.

Statistiken

Informationen zum Druckverhalten können einerseits über die bereits bekannten Befehle als PyKota-Administrator abgerufen werden, andererseits ist dies ebenfalls über die von PyKota mitgelieferten CGI-Seiten möglich (Abb. 3.24).

User	LimitBy	overcharge	used	soft	hard	balance	grace	total	paid	warn
administrateur	NQ	1.0	1	None	None	-0.70		1	0.00	0
ambre	NQ	1.0	6	None	None	-0.60		6	0.00	0
fleur	NQ	1.0	0	None	None	0.00		0	0.00	0
guest	NQ	1.0	0	None	None	0.00		0	0.00	0
jerome	-B	1.0	148	None	None	990.20		148	1006.50	0
rach	NQ	1.0	16	None	None	-1.60		16	0.00	0
rachel	NQ	1.0	528	None	None	-53.40		528	0.00	0
remroot	NQ	1.0	0	None	None	0.00		0	0.00	0
root	NQ	0.5	0	None	None	0.00		0	0.00	0
								Total : 699	1006.50	
								Real : 42545		

Abbildung 3.24.: Benutzerstatistiken auf verschiedenen Drucker

3.5. Überführung auf das Produktivsystem

Nachdem die notwendigen Anpassungen vorgenommen worden sind, kann die Überführung auf das Produktivsystem erfolgen. Als Druckserver wird der bestehende virtuelle Server fungieren. Damit er von den Clients des PC7 angesprochen werden kann, erfolgt eine Umstecken des Servers beim Rechenzentrum des ZID, da dieser zuvor in einem anderen Adressbereich war und auch eine andere Netzmaske hatte. Dadurch wird der CUPS-Server sichtbar für die CUPS-Clients des PC7.

Danach erfolgt die Installation des gepatchten PyKotIcon mit allen notwendigen Abhängigkeiten auf dem Hauptserver des PC7, sowie eine Anpassung der Autostart-Routinen auf die jeweiligen Desktopumgebungen.

Weiters wurde das Sekretariat in das Webinterface eingewiesen und ein Informationsblatt für die Tutoren des PC7 angebracht. Ein weiteres Informationsblatt als Bekanntmachung für die Studenten wurde ebenfalls an verschiedenen Orten des Institutes angebracht.

Zu guter Letzt wurde der Preis pro Seite an die Institutsvorgaben von €0.05 angepasst, weiters erhält jeder Student bei der Aktivierung des Druckkontos fünf Gratisseiten (im Gegenwert von €0.25). Schlussendlich wurde ebenfalls die Tonerkassette des Druckers ausgetauscht, da die ursprüngliche hässliche Schlieren auf dem Papier gezogen hat.

4. Zusammenfassung und Ausblick

Im Rahmen dieses Projektes wurde auf dem Schulungsraum (PC7) des Institutes für Informationswirtschaft die Druckumgebung **CUPS** mit PyKota, einem flexiblen Abrechnungssystem für **CUPS**, installiert. Ziel war es, den Benutzern des PC7 das Drucken vor Ort zu ermöglichen.

OSS war das Hauptkriterium für diese Arbeit, die verwendeten Programme und die Software wurden dem Leser nähergebracht. Weiters wurden auch kleine Exkurse in die theoretischen Hintergründe der verschiedenen Softwareprodukte vorgenommen. Auf diesem Wissen aufbauend, wurde die Installation von **CUPS** mit PyKota beschrieben mit anschließender Konfiguration und Anpassung an die gewünschten Anforderungen.

Die Überführung auf das Produktivsystem wurde gemeinsam mit dem Systemadministrator, Ralf Steuer, koordiniert und durchgeführt. Darüber hinaus wurde das Sekretariat des Institutes eingeschult, um der Druckumgebung Benutzer hinzuzufügen, sowie Dokumente mit entsprechenden Informationen für die Tutoren und Studenten verfasst.

Leider war es nicht erwünscht, die Guthaben des installierten Systems mit denen des Druck-System des **ZID** zu verbinden. Dies wäre eine sehr interessante Verbesserung für die Studenten, welche allerdings nur durch ein gemeinsames Handeln der betroffenen Stellen umsetzbar wäre. Auch wäre eine Überarbeitung bzw. Anpassung des Webinterfaces denkbar oder eine eigene Adaptierung von PyKotIcon, vielleicht im Rahmen eines Vertiefenden Übungsprojekts. Weiters könnte die Admin-Seite von **CUPS** so eingestellt werden, das es den Tutoren möglich wäre, z.B. Druckaufträge zu löschen oder den Drucker neu zu starten.

A. Anhang

Installierte Abhängigkeiten

Folgende Abhängigkeiten waren auf dem Testsystem installiert:

Checking PyKota dependencies...

Checking for Python-PygreSQL availability : OK

Checking for Python-SQLite availability : NO.

Python-SQLite is mandatory if you want to use SQLite as the quota database backend. See <http://www.pysqlite.org>

Checking for MySQL-Python availability : OK

Checking for Python-egenix-mxDateTime availability : OK

Checking for Python-LDAP availability : NO.

Python-LDAP is mandatory if you plan to use an LDAP

directory as the quota database backend. See <http://python-ldap.sf.net>

Checking for Python-OSD availability : OK

Checking for Python-SNMP availability : OK

Checking for Python-JAXML availability : OK

Checking for Python-ReportLab availability : OK

Checking for Python-Imaging availability : OK

Checking for Python-Psyco¹ availability : NO.

Python-Psyco speeds up parsing of print files, you should use it.

See <http://psyco.sourceforge.net/>

Checking for Python-pkpgcounter availability : OK

Checking for Python-PAM availability : OK

¹Nur für x86 verfügbar.

Checking for Python-pkipplib availability : OK
Checking for Python-chardet availability : OK
Checking for GhostScript availability : OK
Checking for SNMP Tools availability : NO.
SNMP Tools are needed if you want to use SNMP enabled printers.
Checking for Netatalk availability : NO.
Netatalk is needed if you want to use AppleTalk enabled printers.

pykota.conf

Um die Lesbarkeit der Konfigurationsdatei in dieser Arbeit zu gewährleisten und die Arbeit nicht unnötig aufzublähen, wurden Kommentare und Leerzeilen mit `sed -e 's/#.*//'` `-e 's/[^I]*$//'` `-e '/^$/ d'` entfernt.

```
[global]
config_charset : UTF-8
storagebackend: pgstorage
storageserver:
storagename: pykota
storageuser : pykotauser
storageuserpw : readonlypw
storagecaching: No
disablehistory: No
logger: system
debug : no
logourl : http://wwwai.wu-wien.ac.at/images/ai-logo_big.gif
logolink : http://wwwai.wu-wien.ac.at
smtpserver: localhost
maildomain: wu-wien.ac.at
usernamecase: native
```

privacy : no
onbackenderror : nocharge
askconfirmation : /usr/bin/pknotify -N continue --destination
\$PYKOTAJOBORIGINATINGHOSTNAME:7654 --timeout 120 --confirm "Hello
\$PYKOTAUSERNAME.\nPrint job \$PYKOTAJOBID send to printer
\$PYKOTAPRINTERNAME is \$PYKOTAPRECOMPUTEDJOBSIZE pages long\nand
will cost you \$PYKOTAPRECOMPUTEDJOBPRICE credits.\n\nYou currently
have \$PYKOTABALANCE credits.\n\nDo you really want to print ?"
unknown_billingcode : deny(/usr/bin/pknotify --destination
\$PYKOTAJOBORIGINATINGHOSTNAME:7654 --timeout 60 --notify "The billing
code specified is not allowed")
keepfiles : no
accounter: software()
skipinitialwait : no
preaccounter: software()
onaccountererror: stop
admin: Ralf Steuer
adminmail: steuer@ai.wu-wien.ac.at
mailto : DevNull
balancezero : 0.0
gracedelay : 7
poorman : 3.5
poorwarn : Your Print Quota account balance is low.
Soon you'll not be allowed to print anymore.
softwarn : Your Print Quota Soft Limit is reached.
This means that you may still be allowed to print for some
time, but you must contact your administrator to purchase
more print quota.
hardwarn : Your Print Quota Hard Limit is reached.
This means that you are not allowed to print anymore.
Please contact your administrator at root@localhost
as soon as possible to solve the problem.
policy: deny

```
maxdenybanners: 0
printcancelledbanners: no
enforcement : strict
trustjobsize : yes
denyduplicates : no
duplicatesdelay : 0
noprintingmaxdelay : 60
statusstabilizationloops : 5
statusstabilizationdelay : 4.0
snmperrormask : 0x4FCC
```

pykotadmin.conf

Der Vollständigkeit halber ist hier ebenfalls die PyKota-Konfigurationsdatei, welche das Passwort für den Datenbankzugriff enthält.

```
[global]
storageadmin: pykotaadmin
storageadminpw: readwritepw
```

pykoticon.patch

Dieser Patch behebt einen kleinen Anzeigefehler von `pykoticon`, welcher mit `patch pykoticon-1.02/bin/pykoticon pykoticon.patch` angewendet wird. Der ursprüngliche Code enthält einen Aufruf, der lediglich unter Windows funktioniert und dafür sorgt, dass sich das Hauptfenster stets im Hintergrund bzw. minimiert aufhält und neue Fenster automatisch in den Vordergrund treten. Unter [GNOME](#) bzw. [KDE](#) war das nicht der Fall und das Hauptfenster blieb als Punkt am Desktop hängen nach der ersten Nachricht.

Der Patch behebt diese strukturellen Schwäche des Programmes nicht direkt, er führt lediglich dazu, das Nachrichten nicht mehr über das Hauptfenster erscheinen, sondern ihre eigenen Unterfenster erhalten.

```
-- /usr/bin/pykoticon 2008-11-11 23:03:24.000000000 +0100
+++ /usr/bin/pynew.py 2008-11-11 23:01:41.000000000 +0100
@@ -315,13 +315,13 @@
     message = _("Sorry, this was forbidden by your system administrator.")
     caption = _("Information")
     style = wx.OK | wx.ICON_INFORMATION | wx.STAY_ON_TOP
- dialog = wx.MessageDialog(self, message, caption, style)
+ dialog = wx.MessageDialog(wx.Frame.__init__(self,
        parent=None, id=wx.ID_ANY), message, caption, style)

def OnAbout(self, event) :
    """Displays the about box."""
- dialog = wx.MessageDialog(self, aboutbox % globals(),
+ dialog = wx.MessageDialog(wx.Frame.__init__(self,
        parent=None, id=wx.ID_ANY), aboutbox % globals(),
        _("About"), wx.OK | wx.ICON_INFORMATION)
    dialog.Raise()
@@ -338,7 +338,7 @@
    caption = _("Information")
    style = wx.OK | wx.ICON_INFORMATION
    style |= wx.STAY_ON_TOP
- dialog = wx.MessageDialog(self, message, caption, style)
+ dialog = wx.MessageDialog(wx.Frame.__init__(self, parent=None,
        id=wx.ID_ANY), message, caption, style)
    dialog.Raise()
    self.dialogAnswer = ((dialog.ShowModal() == wx.ID_NO) and
        "CANCEL") or "OK"
    dialog.Destroy()
```



Drucken im PC7 – Leitfaden für Tutoren

Voraussetzungen:

- Aktiver PC7-Account
- Guthaben auf dem Druckkonto
 - Das Druckkonto wird bei der ersten Einzahlung aktiviert.
 - Einzahlungen erfolgen im Sekretariat des Institutes für Informationswirtschaft (UZA2/Ebene3), Öffnungszeiten sind der Homepage des Institutes zu entnehmen entnehmen.
<http://wwwai.wu-wien.ac.at/offen.html>

Kosten:

- 0,05 €/Seite

Allgemeine Informationen:

- Das Druckkonto ist unabhängig von dem ZID-Konto (dh keine Gratisseiten).
- Eingezahltes Guthaben verfällt **nicht**.
- S/W-Laserdrucker → dh keine Farbausdrucke!
- Übersteigen die Kosten eines Druckauftrages das verfügbare Guthaben, so wird der Druckauftrag nicht durchgeführt.
- Beim **erstmaligen** Aktivieren des Druckkontos gibt es 5 Gratisseiten.
- **Benutzerstatistiken sind abrufbar unter:**
<https://xmprint/cgi-bin/printquota.cgi> (Firefox/Iceweasel only)
- **Die Seitenanzahl eines Druckauftrages kann online auf**
<http://xmprint/cgi-bin/pykotme.cgi> **berechnet werden.**

Mögliche Fehler & Lösungen:

- Das Dokument wird zum Drucker geschickt, allerdings wird nichts ausgedruckt.
 - Manchmal wacht der Drucker aus dem Stromsparmmodus nicht mehr auf, sollte dies der Fall sein den Drucker neu starten. Druckaufträge müssen nicht neu in Auftrag gegeben werden, sie sollten nach dem Neustart automatisch gedruckt werden.
- Das Pop-Up zur Bestätigung des Druckauftrages erscheint nicht.
 - Läuft das Programm? (🟢 Symbol rechts unten)
Falls nicht, ausloggen und erneut einloggen. Oder manuell neu starten.
 - Einige Nachrichten werden im Hintergrund geöffnet. „**Fenster minimieren**“ oder „**Alt-Tab**“ sollte den Dialog zum Vorschein bringen.
- Beim Aufrufen von <https://xmprint/cgi-bin/printquota.cgi> erscheint eine Passwortabfrage. Welchen Benutzernamen/Passwort muss ich angeben?
 - Der Benutzername/Passwort stimmt mit den Logindaten des PC7 überein.

Ansprechperson:

- Ralf Steuer (sys-pc7@aic.ai.wu-wien.ac.at)

Voraussetzungen:

- Aktiver PC7-Account
- Guthaben auf dem Druckkonto
 - Das Druckkonto wird bei der ersten Einzahlung aktiviert.
 - Einzahlungen erfolgen im Sekretariat des Institutes für Informationswirtschaft (UZA2/Ebene3), Öffnungszeiten sind der Homepage des Institutes zu entnehmen.
<http://wwwai.wu-wien.ac.at/offen.html>

Kosten:

- 0,05 €/Seite

Allgemeine Informationen:

- Das Druckkonto ist unabhängig von dem ZID-Konto (dh keine Gratisseiten).
- Eingezahltes Guthaben verfällt **nicht**.
- S/W-Laserdrucker → dh keine Farbausdrucke!
- Übersteigen die Kosten eines Druckauftrages das verfügbare Guthaben, so wird der Druckauftrag nicht durchgeführt.
- Beim **erstmaligen** Aktivieren des Druckkontos gibt es 5 Gratisseiten.
- **Benutzerstatistiken sind abrufbar unter:**
<https://xmprint/cgi-bin/printquota.cgi> (Firefox/Iceweasel only)
- **Die Seitenanzahl eines Druckauftrages kann online auf**
<http://xmprint/cgi-bin/pykotme.cgi> **berechnet werden.**

Anleitung zum Drucken

- Datei -> Drucken (oder ähnliches im Menü) anklicken
- Als Drucker „**pc7print**“ auswählen
- **OK/Drucken** anklicken
- Beim darauffolgenden Pop-Up den Druckauftrag bestätigen. (Das Pop-Up erscheint nicht, wenn nicht genügend Guthaben verfügbar ist.)



Glossar

Notation	Description
Apache	Apache HTTP Server. Der Apache Webserver ist ein modularer Webserver und einer der meistgenutzten im Internet. 49 , 50
API	Application Programming Interface. Programmierschnittstelle eines Softwaresystems, welches anderen Programmen eine Anbindung ermöglicht. 28
CGI	Common Gateway Interface. Standard für den Datenaustausch zwischen einem Webserver und dritter Software; erlaubt interaktive Webseiten. 48 , 49 , 51
CUPS	Common Unix Printing System. Drucksystem für unixoide Betriebssysteme. http://cups.org/ . 3 , 6 , 7 , 13 , 27 , 28 , 31–36 , 39–42 , 51 , 53
Daemon	Disk And Execution MONitor. Unter UNIX ein Programm, welches im Hintergrund läuft und bestimmte Dienste zur Verfügung stellt. 27
DBMS	Datenbankmanagementsystem. Verwaltungssoftware für Datenbanken - organisiert intern die strukturierte Speicherung der Daten und kontrolliert alle lesenden und schreibenden Zugriffe auf die Datenbank. 28 , 36 , 37 , 39
DFSG	Debian-Richtlinien für Freie Software. Richtlinien, welche bei Debian festlegen, ob eine Software „frei“ ist. 16

Notation	Description
FSF	Free Software Foundation. Gemeinnützige Organisation zur Förderung und Produktion Freier Software. 16
GFDL	GNU Lizenz für freie Dokumentation. Für freie Software-Dokumentationen gedachte Lizenz, welche mittlerweile auch bei anderen freien Inhalten verwendet wird. 17
GNOME	GNOME. Frei verfügbare Desktopumgebung für unixoide Betriebssysteme. http://www.gnome.org/ . 45, 46, 57
GNU	GNU is not Unix. Im Rahmen des GNU-Projekts in Entwicklung befindliche, freie Betriebssystem. Oft ist auch das GNU-Projekt gemeint. 3, 13, 15, 16, 23, 28, 31, 32, 36, 62
GPL	GNU General Public License. Beliebte Lizenz für freie Software. http://www.gnu.org/copyleft/gpl.html . 6, 14, 21, 23, 25, 27, 28
GSSAPI	Generic Security Services Application Program Interface. Programmierschnittstelle für Anwendungen, die auf Security Device zugreifen. 50
IPP	Internet Printing Protocol. Stellt Druckerdienste über ein Netzwerk zur Verfügung. Basiert auf HTTP1.1. 27, 28
KDE	K Desktop Environment. Frei verfügbare Desktopumgebung für unixoide Betriebssysteme. http://www.kde.org/ . 45, 46, 57

Notation	Description
Kerberos	Kerberos ist ein verteilter Authentifizierungsdienst, der für offene und unsichere Computernetze entwickelt wurde. 13 , 24 , 25 , 50
KVM	Kernel-based Virtual Machine. Offizielle Virtualisierungslösung des Linux-Kernel. 23
MD5	Message-Digest Algorithm 5. Weit verbreitete kryptographische Hash-Funktion, die einen 128-Bit-Hashwert erzeugt. 38 , 40
MIT	Massachusetts Institute of Technology. Eine der weltweit führenden Universitäten im Bereich von technologischer Forschung und Lehre. 24
OSD	Open Source Definition. Richtlinie zur Bewertung von Software-Lizenzen. 7 , 16
OSI	Open Source Initiative. Organisation, die sich der Förderung von OSS widmet. 16 , 25
OSS	Open Source Software. Wird gemeinhin auch als quelloffene Software bezeichnet. Genaue Definition auf http://www.opensource.org/docs/osd . 3 , 6 , 16 , 27 , 53 , 63 , 64
PDF	Portable Document Format. Plattformübergreifendes Dateiformat für Dokumente. 31–35 , 42 , 43

Notation	Description
PJL	Printer Job Language. Von Hewlett-Packard entwickelte Methode um die Druckersprache durch den Druckauftrag zu ändern und den Status des Druckers zurückzugeben. 29
PPD	PostScript Printer Description. Textdatei, in der die speziellen Eigenschaften eines PostScript-Druckers beschrieben werden. 27
PSP	Projektstrukturplan. Gliederung des Projekts in planbare und kontrollierbare Teilaufgaben. 9
SNMP	Simple Network Management Protocol. Protokoll zur zentralen Überwachung und Steuerung von Netzwerkelementen. 29
SOAP	Simple Object Access Protocol. Netzwerkprotokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können. 22
SSH	Secure Shell. Netzwerkprotokoll und gleichnamiges Programm, welches eine verschlüsselte Verbindung mit einem entfernten Computer herstellen kann. 46 , 50
SSL/TLS	Secure Sockets Layer/Transport Layer Security. Ein hybrides Verschlüsselungsprotokoll zur Datenübertragung im Internet. TLS ist der Nachfolger von SSL. 28
Sudo	Sudo. Erlaubt Benutzer oder Gruppen Kommandos als jemand anderes auszuführen. http://www.sudo.ws/ . 32 , 48
SVN	Subversion. OSS zur Versionsverwaltung von Dateien und Verzeichnissen. 36

Notation Description

URI	Uniform Resource Identifier. Zeichenfolge, die zur Identifizierung einer physischen oder abstrakten Ressource dient - starke Verwendung im Internet. 34, 41
WU	Wirtschaftsuniversität Wien. Homepage - http://www.wu-wien.ac.at . 5, 10, 65
X	X11. Protokoll für den Bau einer grafischen Benutzeroberfläche bei Unix-artigen Betriebssystemen. 31, 35, 45
XPCOM	Cross Platform Component Object Model. Ist ein plattformunabhängiges Komponentenmodell von Mozilla und stellt vor allem Schnittstellen zur Verfügung. 22
ZID	Zentrum für Informatikdienste. IT-Abteilung der WU. http://www.wu-wien.ac.at/zid/ . 6, 7, 53

Abbildungsverzeichnis

1.1.	Projektdefinition anhand der Ausprägungen eines Prozesses	9
1.2.	Projektauftrag	11
1.3.	Projektstrukturplan	12
1.4.	Meileinsteinplan	12
1.5.	Projektbalkenplan	13
2.1.	Zwei verschiedene Arten von Virtualisierungsansätzen. . . .	19
2.2.	Paravirtualisierung unter Xen [xen08]	21
2.3.	Einrückungen sind das Strukturierungselement in Python. .	27
3.1.	Installation von CUPS und einem virtuellen pdf-Drucker . .	33
3.2.	Optionale Treiber für CUPS	34
3.3.	Optionale Verschlüsselung für die Authentifizierung	34
3.4.	Startbildschirm des CUPS-Webinterfaces	35
3.5.	CUPS Administrationsseite	36
3.6.	Entpacken von PyKota v1.26 und Überprüfung der Abhän- gigkeiten	38
3.7.	Installationsassistent für Debian - stark verkürzt	38
3.8.	Beschränkung der Zugriffsberechtigungen auf die PyKota- Datenbank	39
3.9.	Erstellung der PostgreSQL PyKota-Datenbank	40
3.10.	Anpassen der Benutzernamen/Passwörter für die PyKota- Datenbank unter PostgreSQL	41
3.11.	PyKota Installation und Erstellung des Systembenutzers py- kota	41

3.12. Anpassen der Berechtigungen der PyKota Konfigurationsdateien	42
3.13. Anpassen der Berechtigungen	42
3.14. Alle Drucker über cupspykota umleiten	43
3.15. Hinzufügen der Drucker und Benutzer zur PyKota-Datenbank	44
3.16. Drucken eines Testdokumentes auf den pdf-Drucker	44
3.17. Überprüfen des Guthabens	45
3.18. Das neue TaskbarIcon von PyKotIcon	46
3.19. Installation von PyKotIcon	46
3.20. Autostartverknüpfungen für PyKotIcon	47
3.21. mykota - Interaktives Skript für PyKota	48
3.22. Aussehen der Rechnung	50
3.23. Aktuelles Guthaben bei Bestätigung des Druckauftrages - (PyKotIcon)	51
3.24. Benutzerstatistiken auf verschiedenen Drucker	52
A.1. Infoblatt - Tutoren	60
A.2. Infoblatt - Studenten	61

Literaturverzeichnis

- [cma07] Caitlyn Martin. *How Can Linux Market Share Be Accurately Measured?* Oktober 2007.
<http://www.oreillynet.com/linux/blog/2007/10/how_can_linux_market_share_be.html>.
- [cup08] Apple Inc. *Common UNIX Printing System*. August 2008.
<<http://www.cups.org/>>.
- [deb04] Software in the Public Interest Inc. *Eine kurze Geschichte von Debian – Was ist das Debian-Projekt?* April 2007.
<<http://www.debian.org/doc/manuals/project-history/ch-intro.de.html>>.
- [deb07] Debian-Dokumentationsteam. *Debian – Debian-Gesellschaftsvertrag*. April 2004.
<http://www.debian.org/social_contract>.
- [deb08] Software in the Public Interest Inc. *Debian – Das universelle Betriebssystem*. August 2008.
<<http://www.debian.org/>>.
- [dll07] Lionel Menchaca. *Dell to Offer Ubuntu 7.04*. May 2007.
<<http://direct2dell.com/one2one/archive/2007/05/01/13147.aspx>>.
- [ext04] Theodore Ts'o. *ext4: Rename ext4dev to ext4*. Oktober 2008.
<<http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=commit;h=03010a3350301baac2154fa66de925ae2981b7e3>>.

- [ftu06] Frank Tuzi. Tracking and charging for printing with pykota. *linux.com*, September 2006.
<<http://www.linux.com/articles/56831>>.
- [itw08] ITWissen Das große Online-Lexikon für Informationstechnologie. *Virtualisierung*. Oktober 2008.
<<http://www.itwissen.info/definition/lexikon/Virtualisierung-VT-virtualization-technology.html>>.
- [jbl07] James Buchanan. *Linus Torvalds talks future of Linux*. August 2007.
<http://apcmag.com/linus_torvalds_talks_future_of_linux.htm>.
- [jku05] Jürgen Kuri. *Neue Kernel-Serie mit Linux 2.6.11.1 gestarte*. März 2005.
<<http://www.heise.de/newsticker/Neue-Kernel-Serie-mit-Linux-2-6-11-1-gestartet--/meldung/57122>>.
- [ker08] Inc. Linux Kernel Organization. *The Linux Kernel Archives*. August 2008.
<<http://kernel.org/>>.
- [kpy08] A.M. Kuchling. *What's New in Python 2.6 — Python v2.6 documentation*. Oktober 2008.
<<http://docs.python.org/whatsnew/2.6.html>>.
- [krb03] Jason Garman. *Kerberos: The Definitive Guide*. 2003. verfügbar auf Google Book Search:
<http://books.google.at/books?id=mVIA-CQHDjEC&printsec=frontcover&dq=kerberos+books&source=gbs_summary_r&cad=0>.
- [kvm08] Qumranet. *Front Page Kernel Based Virtual Machine*. August 2008.
<<http://kvm.qumranet.com/kvmwiki>>.

- [lml07] Lionel Menchaca. *Linux Losing Market Share to Windows Server*. Oktober 2007.
<<http://www.eweek.com/c/a/Linux-and-Open-Source/Linux-Losing-Market-Share-to-Windows-Server/>>.
- [lto02] Linus Torvalds. *Linux: 2.6 vs. 3.0; What's In A Name?* September 2002.
<<http://kerneltrap.org/node/436>>.
- [lto91] Linus Torvalds. *What would you like to see most in minix?* August 1991.
<<http://groups.google.de/group/comp.os.minix/msg/b813d52cbc5a044b>>.
- [mfu08] Marcus Fischer. *Ubuntu GNU/Linux*. Juli 2008.
<<http://openbook.galileocomputing.de/ubuntu/>>.
- [mk05] Michael Kofler. *Linux - Installation, Konfiguration, Anwendung*. Addison-Wesley, 7th edition, 2005.
- [osi06] Open Source Initiative. *The Open Source Definition - Open Source Initiative*. Juli 2006.
<<http://www.opensource.org/docs/osd>>.
- [pyk08] Jerome Alet. *PyKota's Home*. 2008.
<<http://www.pykota.com/>>.
- [pyt08] Python Software Foundation. *Python Programming Language - Official Website*. Oktober 2008.
<<http://www.python.org/>>.
- [qem08] Fabrice Bellard. *QEMU - open source emulator*. August 2008.
<<http://bellard.org/qemu/>>.
- [rhl08] Red Hat Inc. *redhat.com | The World's Open Source Leader*. September 2008.
<<http://www.redhat.com/about/news/prarchive/2008/qumranet.html>>.

- [rml08] Richard Morris. *Linus Torvalds, Geek of the Week*. July 2008.
<<http://www.simple-talk.com/opinion/geek-of-the-week/linus-torvalds,-geek-of-the-week/>>.
- [sem08] Albert Weichselbraun. *PC7 Druck-Accounting*. Juni 2008.
<http://semanticlab.net/index.php/PC7_Druck-Accounting>.
- [svb08] Sun Microsystems Inc. *VirtualBox*. Oktober 2008.
<<http://www.virtualbox.org/>>.
- [svn07] Steven J. Vaughan-Nichols. *Linux server market share keeps growing*. Mai 2007.
<<http://www.linux-watch.com/news/NS5369154346.html>>.
- [ubu08] Canonical Ltd. *Ubuntu Home Page*. August 2008.
<<http://www.ubuntu.com>>.
- [vmw08] VMware Inc. *VMware: Virtualization via Hypervisor, Virtual Machine + Server Consolidation - VMware*. 2008.
<<http://www.vmware.com/>>.
- [wcu08] Wikipedia. *Common Unix Printing System*. August 2008.
<<http://en.wikipedia.org/wiki/CUPS>>.
- [wix08] Wikipedia. *Xen*. Oktober 2008.
<<http://de.wikipedia.org/wiki/Xen>>.
- [wkb09] Wikipedia. *Kerberos*. Januar 2009.
<http://de.wikipedia.org/wiki/Kerberos_%28Informatik%29>.
- [wkv08] Wikipedia. *Kernel-based Virtual Machine*. Oktober 2008.
<http://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine>.
- [wlk08] Wikipedia. *Linux (Kernel)*. Oktober 2008.
<[http://de.wikipedia.org/wiki/Linux_\(Kernel\)](http://de.wikipedia.org/wiki/Linux_(Kernel))>.

- [woh08] Wolfgang Herrmann. *Linux-Server werden zum Mainstream*. August 2008.
<http://www.computerwoche.de/knowledge_center/open_source/1872498/>.
- [wpy08] Wikipedia. *Python (Programmiersprache)*. Oktober 2008.
<[http://de.wikipedia.org/wiki/Python_\(Programmiersprache\)](http://de.wikipedia.org/wiki/Python_(Programmiersprache))>.
- [wqe08] Wikipedia. *QEMU*. Oktober 2008.
<<http://en.wikipedia.org/wiki/QEMU>>.
- [wvb08] Wikipedia. *VirtualBox*. Oktober 2008.
<<http://en.wikipedia.org/wiki/VirtualBox>>.
- [wvi08] Wikipedia. *Virtualisierung (Informatik)*. Oktober 2008.
<[http://de.wikipedia.org/wiki/Virtualisierung_\(Informatik\)](http://de.wikipedia.org/wiki/Virtualisierung_(Informatik))>.
- [wvm08] Wikipedia. *VMware*. Oktober 2008.
<<http://de.wikipedia.org/wiki/VMware>>.
- [xen08] Inc. Citrix Systems. *Welcome to Xen.org, home of the Xen hypervisor*. 2008.
<<http://www.xen.org/>>.
- [zuc07] Dagmar Zuchi. *Is-projektmanagement und teamarbeit*. Folien-skriptum basiert auf: Roland Gareis. *Happy projects!* Manz Verlag 2006.