

# Diplomarbeit

Oliver Götsch

Analyse und Visualisierung von  
Powermanagement-Daten mittels Java-Applets auf  
einem Linux Webserver

Oliver Götsch

Analyse und Visualisierung von Powermanagement-  
Daten mittels Java-Applets auf einem Linux  
Webserver

Diplomarbeit eingereicht im Rahmen der Diplomprüfung  
im Studiengang Informations- und Elektrotechnik  
Studienrichtung Kommunikationstechnik  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Ing Franz Schubert  
Zweitgutachter : Prof. Dr. Thomas Klinker

Abgegeben am 05. Juli 2007

# Inhaltsverzeichnis

I	Bildverzeichnis.....	3
II	Tabellenverzeichnis.....	3
1	Einführung.....	4
1.1	Motivation.....	4
1.2	Ziel der Arbeit.....	6
2	Hardware.....	8
2.1	Komponenten.....	8
2.1.1	Mainboard.....	9
2.1.2	Prozessor.....	9
2.1.3	Arbeitsspeicher.....	10
2.2	Festplatten.....	10
2.3	Gehäuse.....	11
3	Software.....	13
3.1	Betriebssystem.....	13
3.1.1	Linux vs. Windows.....	15
3.1.2	Linux-Distributionen.....	16
3.1.3	Auswahl des Betriebssystems.....	20
3.2	Datenbank.....	21
4	Installation.....	23
4.1	Betriebssystem.....	23
4.1.1	Grundsystem.....	24
4.1.2	Konfiguration Debian.....	30
4.2	Java-Technologie.....	36
4.3	Apache – Webserver.....	39
4.4	Tomcat – Webserver.....	40
4.5	Datenbanksystem Oracle 10g.....	45
4.6	JEVis-Portal.....	55
4.7	Skriptbasierte Installation.....	55
5	Inbetriebnahme des Servers.....	58
5.1	Einspielen einer Datensicherung.....	58
5.2	Parallelbetrieb.....	60
6	Analyse der Daten.....	61

6.1	Auswertung und Visualisierung der Daten.....	61
6.1.1	PControll.....	61
6.1.2	JEVis-Portal.....	62
6.2	Bestimmung des Lastprofils.....	64
6.3	Modellbildung zur Vorhersage eines Lastprofils.....	66
7	Hochverfügbarkeit.....	68
7.1	Verfügbarkeit und Hochverfügbarkeit.....	68
7.1.1	Computercluster.....	69
7.1.2	Failover.....	70
7.1.3	Switchover.....	71
7.1.4	RAID.....	71
7.1.5	Single Point of Failure.....	72
7.1.6	Standby-Datenbank.....	73
7.1.7	Server Load Balancing.....	73
8	Absicherung des Servers.....	75
8.1	Physikalischer Schutz des Servers.....	75
8.2	Sicherheit des Servers und seiner Dienste.....	76
9	Zusammenfassung /Ausblick.....	78
9.1	Zusammenfassung.....	78
9.2	Ausblick.....	79
	Anhang.....	80
A	Konfiguration Debian.....	80
A.1	Spracheinstellung.....	80
A.2	Erweiterte Netzwerkkonfiguration.....	81
A.3	Systemuhr.....	81
A.4	Einstellungen für den apt-Paketmanager.....	81
B	Quelltexte.....	82
B.1	Installationsskript JEVis.....	82
B.2	Installationsskript Java.....	94
B.3	Installationsskript Apache.....	95
B.5	Installationsskript Oracle.....	97
C	Ausdrucke der Konfigurationsdateien.....	99
C.1	etc/apt/sources.conf.....	99
C.2	etc/profile.....	99
C.3	etc/init.d/tomcat.....	100
	Literaturverzeichnis.....	103

# I Bildverzeichnis

Bild 1: Fast vollständige Abstraktion unter Linux [5]	14
Bild 2: Abstraktionsschichten unter Windows NT (vereinfacht) [5]	15
Bild 3: Bestandteile einer Linux-Distribution [6]	17
Bild 4: Bootvorgang am Beispiel von DOS [10]	25
Bild 5: Installationsbildschirm Debian	26
Bild 6: Beispiel einer Partitionstabelle	29
Bild 7: Willkommensbildschirm Debian	31
Bild 8: Softwareauswahl unter Debian	32
Bild 9: Fehlermeldung bei Durchführung eines Updates	34
Bild 10: Default Webseite des Apache - Webservers	40
Bild 11: Default Webseite des Tomcat - Webservers	45
Bild 12: Oracle Installer - Installation Oracle 10g	53
Bild 13: Oracle Enterprise Manager - Performance der Datenbank ora1	54
Bild 14: Screenshot des Programms "Power Control" [2]	62
Bild 15: Darstellung von Lastprofilen in JECart [15]	63
Bild 16: Ranking von Liegenschaften in JEBench [16]	64
Bild 17: Infrastruktur zur Ermittlung der Lastdaten im Projekt „INSEL“ [2]	65
Bild 18: Summenlastprofil der 40 lastintensivsten Liegenschaften Hamburgs [2]	66

# II Tabellenverzeichnis

Tabelle 1: Runlevel unter Linux [11]	44
Tabelle 2: Verfügbarkeitsklassen eines Servers [17]	69

# 1 Einführung

## 1.1 Motivation

Die Situation auf dem Markt der elektrischen Energie wird sich mittelfristig aller Voraussicht nach entscheidend ändern. Während heute die Energieversorger auf die Anforderungen der Verbraucher (Leistungsbedarf) durch Zu- und Abschalten von Kraftwerken reagieren können, da genügend Leistungsreserven vorhanden sind, könnte in der Zukunft eine völlig neue Situation entstehen: Eine Bedarfsregelung auf die herkömmliche Art und Weise wird wegen der reduzierten Kraftwerkskapazität (z.B. Stilllegung von AKW) und des vermehrten Einsatzes regenerativer Energien, deren Einspeisung in das Netz klimatisch bedingt nicht regelbar ist, nicht mehr in bisheriger Weise möglich sein.

Die Energieversorger müssen dann die Möglichkeit haben, auf das Verbraucherverhalten kurzfristig Einfluss zu nehmen, d.h. die Last dem momentanen Energieangebot anzupassen. Eine sinnvolle Nutzung der Energie wird nur dann möglich sein, wenn ein "Globales Energiemanagement" für die Erzeugung und Verteilung der Energie sorgt. Ein wesentlicher Faktor dabei ist die Kommunikation zwischen Energieerzeugern und -verbrauchern. Globales Energiemanagement beinhaltet somit:

- Energieerzeugung
- Energieverbrauch
- Energieverteilung
- Kommunikationssysteme
- Informationssysteme

In Mitteleuropa wird in den nächsten Jahren die sinnvolle Energieverteilung mit Hilfe umfangreicher Informations- und Kommunikationssysteme wachsende Bedeutung gewinnen, da nicht zu erwarten ist, dass größere Kraftwerkseinheiten gebaut werden.

Vielmehr ist es notwendig, kleinere dezentrale Kraftwerke mit unterschiedlichen Arten der Energieerzeugung zu koordinieren. Für den Fachbereich Elektrotechnik und Informatik an der HAW Hamburg ergibt sich hier ein umfangreiches Forschungsgebiet, das neben der klassischen Elektrotechnik mit Energieerzeugung, -verteilung und -umwandlung, auch Informations- und Kommunikationstechnik sowie Softwaretechnik beinhaltet.

Im Rahmen des Forschungsvorhabens (2002 – 2004) "Globales Energiemanagement" an der Hochschule für Angewandte Wissenschaften Hamburg sollen in einem Pilotprojekt die technischen Möglichkeiten für eine derartige Einflussnahme auf den Verbraucher geschaffen und erprobt werden. [1]

Globale Erwärmung und stetig steigende Preise für fossile Energien führen dazu, dass die Einspeisung von Strom aus regenerativen Quellen und aus der gekoppelten Erzeugung von Strom und Wärme in Blockheizkraftwerken (BHKW) in das Stromverteilernetz in einer stetig steigenden Anzahl von Ländern immer mehr zunimmt. 2006 erreichte der Beitrag von Strom aus erneuerbaren Quellen in Deutschland etwa 71,5 Terawattstunden (TWh), was einer neuen Höchstmarke von 11,6% entspricht.

Diese ständig steigende Menge von Strom aus dezentraler Erzeugung stellt neue Anforderungen an die Betreiber der Stromverteilernetze. Laut Dena Netzstudie wird bei einem politisch gewollten Ausbau der regenerativen Energien auf 20% des Strombedarfs bis 2020 der Bedarf an Regenergie bereits bis 2015 um 50% bis 250% (je nach Szenario) zunehmen. Dies liegt zum einen daran, dass insbesondere Wind- und Sonnenenergie sich nicht nach dem Erzeugungsbedarf des Verteilernetzes richten, sondern von sich ggf. kurzfristig ändernden Wetterbedingungen abhängen.

Zusätzlich richtet sich das Erzeugungsmanagement in BHKW normalerweise nach dem Wärmebedarf und nicht nach dem Strombedarf (oder dem Bedarf des Verteilernetzes). Aus diesem Grund ist in den letzten Jahren die Vernetzung von vielen dezentralen Erzeugungseinheiten zu virtuellen Kraftwerken (VK) über schnelle (und sichere?) Kommunikationstechnologien (TCP/IP u. a.) in immer mehr Forschungsprojekten untersucht worden. Ziel all dieser Projekte ist es stets, kleinere Erzeugungseinheiten zu vernetzen, um größere Erzeugungsverbünde zu schaffen, die sich dann leichter in das Managementregime des (Verteil- und/oder Übertragungs-) Netzes einbinden lassen. Dabei steht die Entwicklung geeigneter Kommunikationsmethoden und -standards bei der

Vernetzung kleiner und mittlerer dezentraler Erzeugungseinheiten in vielen Projekten im Focus des Interesses. [2]

Das seit 2006 laufende Forschungsprojekt INSEL untersucht nun, wie die 30 bis 40 lastintensivsten öffentlichen Liegenschaften der Stadt Hamburg zu einem virtuellen Netzwerk zu bündeln sind, damit diese in das virtuelle Kraftwerk der Steag Saarenergie AG eingebunden werden können. Dazu werden die einzelnen Liegenschaften mit Datenloggern ausgerüstet, falls sie es noch nicht sind. Diese Datenlogger sollen die aktuellen Verbrauchswerte der jeweiligen Liegenschaft protokollieren und für eine Server-basierte Abholung bereitstellen.

## 1.2 Ziel der Arbeit

Wie bereits im obigen Kapitel erwähnt, werden die Daten der 30 – 40 lastintensivsten öffentlichen Liegenschaften gesammelt und in einer Datenbank auf einem Server zur Analyse bereitgestellt. Mit diesen Daten wird dann später ein Modell entworfen, mit dem eine Vorhersage der zu erwartenden Abschaltleistung möglich ist.

Im ersten Schritt wird das Prognose-Tool (nicht Teil dieser Diplomarbeit) an gespeicherten Lastgängen getestet und untersucht, in wie weit die vorhergesagten den tatsächlichen Lastgängen entsprechen. Im zweiten Schritt werden aus aktuellen Daten die zukünftigen einzelnen Abschaltleistungen errechnet und aufsummiert. Diese Abschaltleistung wird dann der Steag Saarenergie als Minutenreserve verkauft.

Ziel der Arbeit ist es, einen Server zu installieren, der die oben genannten Aufgaben übernimmt und eine Plattform zur Verfügung stellt, Java-Applets zur Analyse und Modellbildung zu entwickeln und zu betreiben. Des Weiteren wird über den Server eine Möglichkeit geschaffen, einzelne Verbraucher abzuschalten.

Dabei wird zunächst untersucht, welche Anforderungen die zu bewältigen Aufgaben an die Hard- und Software stellen (Kapitel 2 und 3), was später in Kapitel 7 (Hochverfügbarkeit des Systems) vertieft wird. Das Unterkapitel 3.1.2 gibt einen Überblick über die wichtigsten Distributionen und für welche Distribution sich letztendlich entschieden wurde. In Kapitel 4

wird dann zeigt, wie die benötigten Dienste zu installieren sind. Im ersten Schritt wird in den Kapiteln 4.1 – 4.6 beschreiben, wie die Dienste manuell zu installieren sind.

Als zweiter Schritt erfolgt eine Entwicklung von Installationsskripten, die die Erkenntnisse aus den manuellen Installationen aufgreifen und in der Ausführung eines einzelnen Befehls zusammenfassen (vgl. Kapitel 4.7).

Nach der Beschreibung der Installation des Servers, wird im folgenden Kapitel beschrieben, wie der Server in Betrieb genommen wird. In Unterkapitel 5.3 wird aufgezeigt, welche Anpassungen (z.B. an der Datenbank) je nach Szenario gemacht werden müssen, damit der Server in der jeweiligen Umgebung störungsfrei läuft. Während in Kapitel 5 nur die Interoperabilität der Server gewährleistet sein soll, wird im 8. Kapitel beleuchtet, wie die Integrität des Servers verbessert werden kann. Es werden Schritte aufgezeigt, die über eine Standardinstallation hinausgehen. Ziel hierbei ist, den Server sicherer zu machen. In Kapitel 8 wird außerdem der Frage nachgegangen, was zu tun ist, um das System vor inneren und äußeren Beeinträchtigungen zu sichern. Das betrifft sowohl das Betriebssystem an sich als auch die Serverdienste, die darauf ausgeführt werden. Patchmanagement ist hier ein wichtiges Stichwort.

Ist das System nun sicher, kann es seiner eigentlichen Bestimmung zugeführt werden. Mit der Bereitstellung der aus den Datenloggern abgeholten Daten kann eine Analyse dieser stattfinden, in dem die Daten aufbereitet und grafisch dargestellt werden. Außerdem sollen weitere Applets entwickelt werden, die sowohl Steuerungsfunktionen als auch weitere Analysemerkmale bereitstellen. Diese Entwicklung ist aber nicht Teil dieser Diplomarbeit.

Je nach Einsatz des Servers und wie kritisch die Anwendungen sind, die darauf laufen, spielt die Hochverfügbarkeit mehr oder weniger eine Rolle. Da der Server dazu eingesetzt wird, Daten zu sammeln, diese auszuwerten und eine Vorhersage zu treffen, wie viel Minutenreserve verkauft werden kann, ist es erforderlich, dass der Server zu jeder Zeit erreichbar ist. Deshalb beschreibt das Kapitel 7 die Erfordernisse ausführlicher, wie eine Hochverfügbarkeit des Servers zu erreichen ist. Das 9. Kapitel, das eine Zusammenfassung und einen Ausblick enthält, beendet diese Arbeit.

# 2 Hardware

In diesem Kapitel die Komponenten im Allgemeinen behandelt und es wird gezeigt, warum eine solche Behandlung nicht nur sinnvoll, sondern auch erforderlich ist. Danach werden die einzelnen Komponenten, die sich auf dem Mainboard befinden, näher betrachtet. Bei den Festplatten und beim Gehäuse wird auf deren besondere Wichtigkeit eingegangen und untersucht, welche Kriterien zum Ausschluss von Standardkomponenten führen. Im Weiteren wird dann erläutert, aus welchen Gründen spezielle Komponenten so wichtig für die Verwendung des Servers sind. Dabei wird auch darauf eingegangen, weshalb Server einiger Spezialhersteller von vorn herein nicht berücksichtigt werden können.

## 2.1 Komponenten

Allein die unterschiedlichen Aufgaben, die von Servern und Desktop-Computern erfüllt werden, zeigen auf, dass bei Servern ganz andere Kriterien gelten als bei Desktop-Computern. So müssen die Komponenten im Serverumfeld zuverlässig, langlebig und dabei trotzdem noch leistungsfähig sein. Bei den Komponenten aus dem Desktop-Bereich müssen diese Merkmale nicht unbedingt gegeben sein.

Die Betrachtung nach der Verwendung des Servers erübrigt die Frage, warum keine Standardkomponenten eingesetzt werden und verdeutlicht aber auch, dass es notwendig ist, die Art der Verwendung zu untersuchen und welche Komponenten dafür erforderlich sind. Die Anwendung, die später auf dem Server laufen soll, macht es erforderlich, dass der Server so wenig wie möglich ausfällt. Im Anschluss wird für die einzelnen Komponentengruppen erläutert, welche Komponenten ausgewählt werden müssen, damit der Server möglichst immer erreichbar ist bzw. bleibt.

Dabei erfolgt zunächst nur eine grundlegende Betrachtung, nämlich was verfügbar ist und was für den Server ausgewählt wurde. Diese Verfügbarkeit wird dann später im Kapitel 7 „Hochverfügbarkeit des Servers“ vertieft.

Um sich nicht mit der Auswahl der einzelnen Komponenten zu befassen, wurden die Hardware-Lieferanten kontaktiert und mit der Beschaffung des Servers beauftragt, der den Ansprüchen des anvisierten Einsatzgebietes genügt. Bei der Frage, von welchem Hersteller der Server zu bestellen ist, fielen bis auf Dell und Fujitsu Siemens alle anderen heraus, weil die Hochschule für Angewandte Wissenschaften (HAW) Rahmenverträge mit diesen beiden Unternehmen abgeschlossen hat. Die Verträge binden die Hochschule daran, sich in jedem Fall Angebote dieser beiden Firmen einzuholen. Ausschlag gebend für Fujitsu Siemens war schließlich der Preis, da die im Projekt bereit stehenden Geldmittel begrenzt sind.

### **2.1.1 Mainboard**

Das Mainboard ist praktisch das Herzstück des Computers, auf ihm finden sich für den Betrieb alle wichtigen Dinge wie Prozessor, Arbeitsspeicher und Anschlüsse für Peripherie wie Drucker usw. sowie natürlich auch Bildschirm, Tastatur und Maus. Als Mainboard wird ein Serverboard von Fujitsu Siemens verwendet, das Platz für 2 Prozessoren bietet. Dieses Board hat gegenüber Mainboards für Desktops den Vorteil, dass die Komponenten so angeordnet sind, dass sie optimal gekühlt werden können. Dieses ist eine wichtige Voraussetzung, damit der Server im Dauerbetrieb einwandfrei funktioniert. Des Weiteren sind die verwendeten elektronischen Bauteile von höherer Qualität, die eine lange Haltbarkeit garantieren sollen. Außerdem befinden sich 8 Steckplätze für den Arbeitsspeicher auf dem Mainboard.

### **2.1.2 Prozessor**

Als Prozessor wird ein Intel Xeon 3,0 GHz für den Server verwendet. Die Prozessoren der Xeon-Reihe sind Prozessoren, die speziell für den Serverbetrieb ausgelegt sind. Wie oben erwähnt, bietet das Mainboard Platz für zwei Prozessoren. Aus Kostengründen wurde das Board aber nur mit einem Prozessor bestückt. So besteht noch die Möglichkeit, bei Bedarf einen 2. Prozessor nachzurüsten.

Bei Angebotsabgabe der Hardwarelieferanten war ein Single Core Stand der Technik. Single Core besteht aus nur einem Hauptprozessorkern, wobei jetzt Prozessoren mit Dual Core aus zwei und Quad Core aus vier Kernen bestehen. Nach Möglichkeit sollte ein Quad Core

Prozessor Verwendung finden, da die Rechenlast auf die einzelnen Kerne verteilt werden kann. Daraus entsteht ein deutlicher Performancegewinn, z.B. bei komplexen SQL-Abfragen, weil mehrere Befehle gleichzeitig abgearbeitet werden können (Multitasking).

### **2.1.3 Arbeitsspeicher**

Der Arbeitsspeicher oder Hauptspeicher ist in der Informationstechnik der Speicher eines Computers, in dem Datenobjekte, also Programme und die von diesen in Mikroprozessoren zu verarbeitenden Nutzdaten, abgelegt und zu einem späteren Zeitpunkt (unverändert) abgerufen werden können. [3]

Als Arbeitsspeicher waren 2GB vorgesehen. Nach Beratung mit der Firma Envidatec, einem Praxispartner im Projekt INSEL, ergab sich, dass diese Größe grundsätzlich ausreichen würde, wobei nach deren Erfahrung 4GB aber deutlich besser wären. Die Firma Envidatec betreibt derzeit schon eine Datenbank zum Lastmanagement mit realen Daten, so dass deren damit gesammelten Erfahrungen hier einfließen können.

Die Entscheidung, 4GB Arbeitsspeicher zu verwenden, beruht nur auf obiger Empfehlung. Nicht berücksichtigt wurden jedoch die Anzahl der zu erwartenden User/Prozesse und deren Arbeitsverhalten. Aus diesen Daten errechnet sich in der Regel die zu erwartende Last auf das System und daraus wiederum die Dimensionierung des Hauptspeichers.

## **2.2 Festplatten**

Auf den Festplatten werden das Betriebssystem und die Daten abgelegt. Für den Server werden vier Festplatten benötigt, zwei jeweils mit 73 GB Speicherkapazität für das System und zwei mit 146 GB Speicherkapazität für die Daten. Zum Einsatz kommen SAS (Serial Attached SCSI)-Platten, die seriell an den Festplattencontroller angeschlossen werden. SCSI (Small Computer System Interface), das vorwiegend im Serverbereich eingesetzt wird, ist eine standardisierte parallele Schnittstelle für Verbindung von Festplattencontroller und Festplatten sowie Datenübertragung zwischen Peripheriegeräten und dem Computer-Bus[4].

Der bisherige parallele Anschluss wird durch eine serielle Version abgelöst. Die Festplatten werden jeweils zu einem RAID1 (redundant array of independent disks) zusammengeschaltet. Die Zahl nach dem RAID gibt den Modus an, auf welche Art und Weise die Platten zusammengeschaltet werden. Der gewählte Modus 1 bewirkt eine Spiegelung, d.h. die Daten werden gleichzeitig auf zwei Festplatten geschrieben. So lässt sich die Sicherheit der Daten steigern.

Im RAID-Standard sind weitere Modi vorhanden, in welcher Form Festplatten zusammengeschaltet werden können. Weil die Spiegelung für die größtmögliche Sicherheit der Daten sorgt und die Sicherheit favorisiert wird, werden die übrigen Modi nicht näher erläutert.

Nach Erfahrungen des Projektpartners Envidatec, deren Kundendatenbank ca. 40 GB Festplattenkapazität belegt, sind die Datenfestplatten ausreichend dimensioniert. Es ist somit gewährleistet, dass die im Rahmen des Projekts anfallenden Daten gespeichert werden können. Die oben benannte Firma wird an dieser Stelle mitberücksichtigt, da deren Server als Backup-Server weiter verwendet wird. Aus diesem Grund muss deren Datenbank auf den Projektserver kopiert werden. In dieser Konstellation haben sowohl das Projekt als auch die Firma Envidatec einen Primär- und einen Backup-Server. Der Sinn einer redundanten Auslegung des Servers ist in Kapitel 7 näher erläutert.

## **2.3 Gehäuse**

In einem Gehäuse finden die gesamten Komponenten Platz, die zum Betrieb eines Computers notwendig sind. Das sind neben dem oben angesprochenen Mainboard mit seinen Komponenten und den Festplatten noch das Netzteil sowie Lüfter zur Kühlung. Als Gehäuse wird dabei ein Intel Server Chassis in Rack-Bauweise verwendet. Ein Rack ist ein Regal, das für Computer im Serverbereich verwendet wird und eine standardisierte Bauweise von 19“ aufweist.

Eine Besonderheit an einem Servergehäuse ist die Möglichkeit, redundante Netzteile und Gehäuselüfter einzubauen. Das hat gegenüber einem normalen Gehäuse den Vorteil, dass statt eines Netzteils zwei eingebaut werden können, was bewirken soll, dass bei Ausfall eines

Netzteils nicht der gesamte Server ausfällt. Außerdem kann ein evtl. defektes Netzteil im laufenden Betrieb ausgetauscht werden. Dies sind Aspekte, die für eine Hochverfügbarkeit des Servers sorgen. Eine nähere Betrachtung erfolgt dann im Kapitel 7.

# 3 Software

Nachdem im vorherigen Abschnitt die Hardware betrachtet wurde, wendet sich dieses Kapitel dem Gebiet der Software zu. In diesem Kapitel wird genauer untersucht, was es für Serverbetriebssysteme gibt und welches letztendlich eingesetzt wird. Dabei werden die Gründe angegeben, die zu dieser Entscheidung geführt haben. Die gleiche Betrachtungsweise erfolgt dann anschließend im Bereich der Datenbanken.

## 3.1 Betriebssystem

Ein Betriebssystem ist die Software, die die Verwendung (den Betrieb) eines Computers ermöglicht. Es verwaltet Betriebsmittel wie Speicher, Ein- und Ausgabegeräte und steuert die Ausführung von Programmen.

Betriebssystem heißt auf Englisch operating system (OS). Dieser englische Ausdruck kennzeichnet den Sinn und Zweck des Betriebssystems. Die in den Anfängen der Computer stark mit schematischen und fehlerträchtigen Arbeiten beschäftigten Operatoren schrieben Programme, um sich die Arbeit zu erleichtern; diese wurden nach und nach zum operating system zusammengefasst.

Betriebssysteme bestehen in der Regel aus einem Kern (englisch: kernel), der die Hardware des Computers verwaltet, sowie grundlegenden Systemprogrammen, die dem Start des Betriebssystems und dessen Konfiguration dienen. Betriebssysteme finden sich in fast allen Computern. [5]

Generell stellt sich die Frage, welches Betriebssystem zu nehmen ist, wenn der Server oder auch der heimische PC damit ausgestattet werden soll. Dabei besteht nicht nur die Wahl zwischen Windows und Linux, sondern es können auch anderen Betriebssystemen wie AiX, HP-UX, MacOS, Solaris, und SunOS verwendet werden. Dabei handelt es sich bei allen um Unix Derivate. Derivat kommt im Ursprung aus dem Lateinischen vom Wort derivare („ableiten“), was im Bereich der Software bedeutet, dass es eine Ableitung von einem bestehenden Softwareteil ist.

Im Fall von Unix oder aber auch Linux ist damit eine Verbesserung bzw. Erweiterung des Kernels durch proprietäre (an Lizenzkosten oder an bestimmte Bedingungen gebunden) Software und Treiber gemeint.

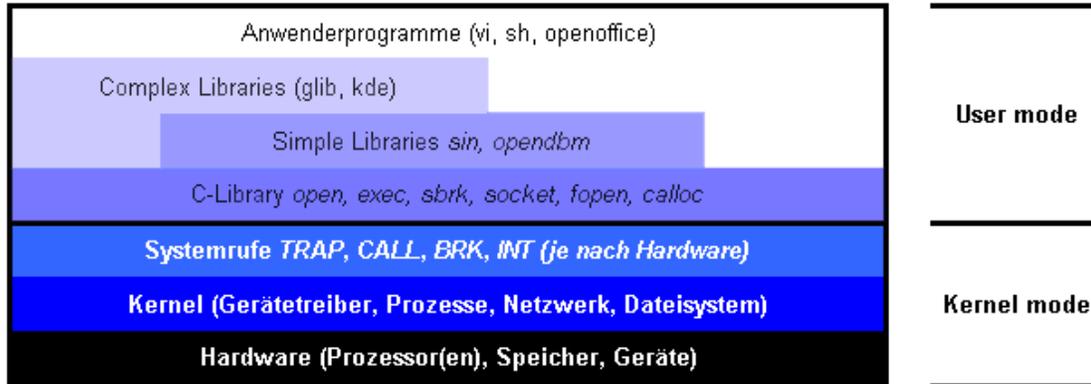


Bild 1: Fast vollständige Abstraktion unter Linux [5]

Bei allen oben genannten Derivaten fallen Lizenzkosten an, d. h. die Software muss gekauft werden, um sie einsetzen zu dürfen. Dies war nicht der entscheidende Grund, diese Betriebssysteme nicht einzusetzen, aber spielte trotzdem eine Rolle. Entscheidend war letztendlich, dass all diese Derivate für spezielle Prozessor-Architekturen entwickelt wurden und somit nicht zu dem oben spezifizierten und erworbenen Server passen.

Windows als weiteres kommerzielles Produkt bietet neben den Clientbetriebssystemen, wie Vista, Windows XP usw., um nur die gängigsten zu nennen, auch ein Serverbetriebssystem mit dem Namen Windows Server an. All diese Versionen basieren auf Windows NT, das damals von Microsoft für das Geschäftsumfeld im Server- und Workstationbereich entwickelt wurde. Ziel dieser Entwicklung war es, eine Zuverlässigkeit zu erreichen.

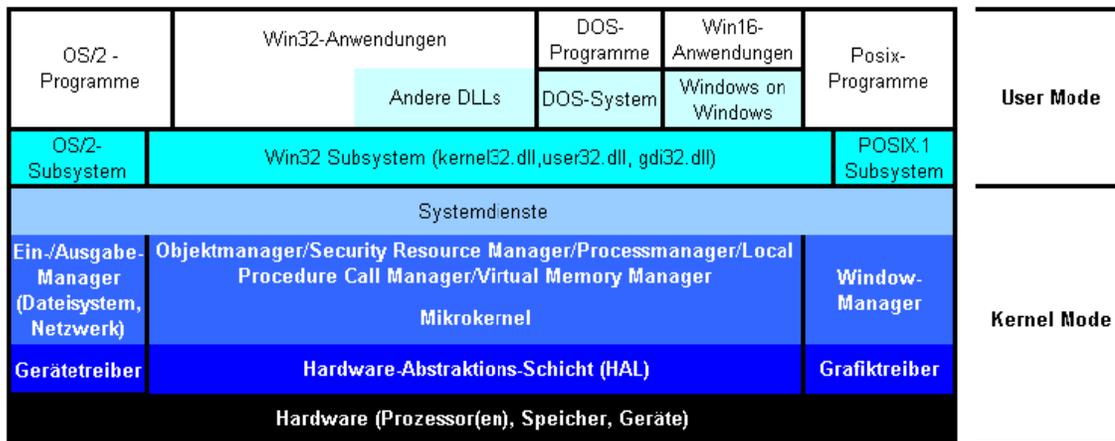


Bild 2: Abstraktionsschichten unter Windows NT (vereinfacht) [5]

Die aktuelle Serverversion ist der Windows Server 2003 R2, wobei die R2 für das 2. Release (Veröffentlichung) steht.

### 3.1.1 Linux vs. Windows

Da Windows - Server und Clients zusammen genommen - das gebräuchlichste Betriebssystem ist, wäre es von Vorteil dies einzusetzen. Denn durch den alltäglichen Umgang ist eine intuitivere Arbeitsweise zu erwarten, als es mit Linux der Fall wäre.

Daraus ließe sich ableiten, dass eine Administration des Systems einfacher ist, zumal sich die meisten Einstellungen über Assistenten vornehmen lassen. Außerdem verfügen alle Programme, die auf dem Server betrieben werden sollen, über eine Setup-Routine. Aber gerade die hohe Verbreitung gereicht Windows zum Nachteil, wenn es darum geht, ein sicheres System zu installieren.

Außerdem sorgt der hohe Verbreitungsgrad dafür, dass eine größere potenzielle Gefahr besteht, sich einen Virus einzufangen oder von außen attackiert zu werden. Das liegt daran, dass die Bedrohungen zum einen zahlreicher sind und zum anderen mit einem Angriff weitaus mehr Schaden angerichtet wird. Der größere Schaden ist unter anderem ein Grund dafür, warum überhaupt ein Angriff ausgeführt wird.

Ein weiterer Nachteil von Windows ist auch, dass für die Nutzung des Servers Lizenzkosten anfallen. Zum einen muss das Serverprodukt selbst lizenziert werden und zum anderen fallen auch Kosten für die Windows-Clients an, die selbst auch lizenziert werden müssen und darüberhinaus eine weitere Lizenz brauchen, um auf den Server zugreifen zu können.

Linux als freies Produkt ist ohne Lizenzkosten erhältlich. Die meisten Distributionen lassen sich aus dem Internet herunterladen, so dass nur die Onlinekosten zum Herunterladen der Software anfallen.

### **3.1.2 Linux-Distributionen**

Eine Distribution ist eine Zusammenstellung von Software, die als Komplettpaket weitergegeben wird. Die Gründe dafür können verschieden sein. Ein typisches Beispiel sind Software-Komponenten, die nur dann sinnvoll genutzt werden können, wenn sie gemeinsam mit anderer Software installiert werden oder weil ein Programm vom Vorhandensein anderer Programme abhängig ist, um gestartet werden zu können. Es kommt auch häufig vor, dass ein Distributor der Zusammenstellung eigene Besonderheiten mitgibt, die es in dieser Form nirgendwo anders gibt.

Vor allen Dingen im Bereich der freien Software wird dieses Konzept häufig genutzt. So gibt es Linux-Distributionen und Distributionen freier DOS-Versionen. Damit entsteht häufig die Situation, dass es für ein und dieselbe Software verschiedene Distributionen gibt. Diese basieren dann auf den gleichen Quelltexten, haben aber spezifische Eigenheiten in der Installation, im Erscheinungsbild (Look-and-Feel), in der konkreten Programmauswahl, im Design, etc. [6]

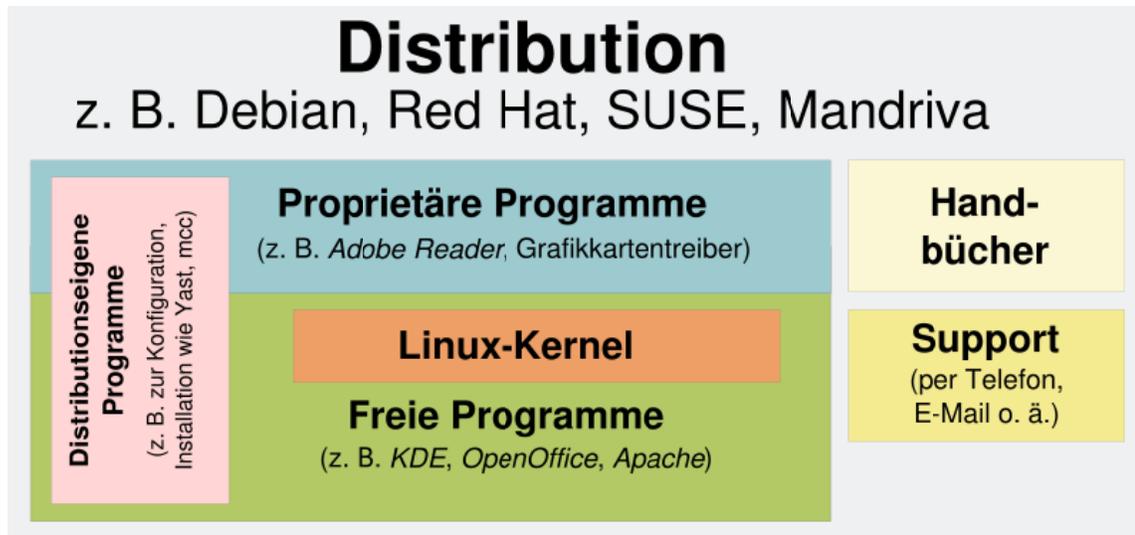


Bild 3: Bestandteile einer Linux-Distribution [6]

### 3.1.2.1 Debian GNU-Linux

Die Debian Distribution ist frei, da sie unter der GNU-Lizenz (GNU is not Unix) steht. Das bedeutet, es darf nur frei erhältliche Software eingesetzt werden. Auf proprietäre Software wird dabei verzichtet.

Bekannt ist Debian vor allem dadurch, dass es eine eigene Paketverwaltungssoftware einsetzt. Außerdem richtet es sich eher an erfahrene Benutzer, da es den Ruf hat, sich nicht allzu leicht installieren zu lassen. Es gilt aber als recht stabil und sicher. Das liegt daran, dass sich Debian nicht den Zwang auferlegte, in regelmäßigen Abständen neue Versionen auf den Markt zu bringen. Das hat den Vorteil, dass die Zeit vorhanden ist, um neue Software ausgiebig zu testen, bevor sie freigegeben wird und in das eigentliche Release einfließt.

Ein weiterer Vorteil Debians ist, dass Software- und sämtliche Sicherheitsprobleme öffentlich behandelt werden. Aspekte der Sicherheit werden öffentlich auf der debian-security Mailing-Liste diskutiert. Debians Sicherheitsgutachten werden über eine öffentliche Mailing-Liste gesendet (sowohl innerhalb als auch außerhalb) und auf einem öffentlichen Server bekannt gegeben. Diese Verfahrensweise verspricht ein schnelleres Auffinden von Sicherheitslücken und damit die Möglichkeit, diese eher beheben zu können. Die

entgegengesetzte Herangehensweise des Security through obscurity wird dagegen, wie oft innerhalb der Open-Source-Community, als nicht praktikabel angesehen.

Dabei passen die Paketbetreuer die Sicherheitsaspekte ihrer jeweiligen Software an die allgemeinen Grundsätze von Debian an. Daher sind Dienste nach der Installation oft „sicher voreingestellt“, was von einem Benutzer als „Einschränkung“ empfunden werden kann. Dennoch versucht Debian, Sicherheitsaspekte und einfache Administration abzuwägen. Zum Beispiel werden Dienste wie ssh und ntp nicht inaktiv installiert, wie es bei anderen Distributionen üblich ist.

Wenn ein Sicherheitsproblem in einem Debian-Paket entdeckt wurde, wird es zusammen mit einer Einschätzung der dadurch entstehenden Gefahr direkt veröffentlicht. Parallel wird so schnell wie möglich ein Sicherheitsupdate dieses Pakets vorbereitet und auf speziellen Servern veröffentlicht. Kritische Sicherheitslücken werden auf diese Weise häufig innerhalb von Stunden geschlossen. [7]

### **3.1.2.2 Red Hat Linux Professional / Fedora**

Red Hat Linux (RHL) war eine der bekanntesten Linux-Distributionen. Diese Distribution wurde von der Firma Red Hat zusammengestellt. Seit September 2003 wird diese nicht mehr weiterentwickelt und ging danach in Fedora und Red Hat Enterprise Linux über.

Red Hat Linux wurde mit einem grafischen Installer installiert, der auch für Einsteiger leicht bedienbar ist. Für die Konfiguration der integrierten Firewall gab es das eingebaute Tool Lokkit. Ab Red Hat Linux 8.0 war Red Hat dazu übergegangen, nur noch Software einzubinden, die vollständig frei ist.

Im September 2003 kündigte Red Hat an, die eigene Consumer-Linux-Distribution zu Gunsten einer Verschmelzung mit dem fedora.us-Projekt aufzugeben und sie als Community-Projekt unter dem Namen Fedora Core weiterzuführen. Im November 2003

veröffentlichte dieses Projekt dann die erste Version der neuen Distribution, die zu dem Zeitpunkt eine direkte Weiterentwicklung des alten Red Hat Linux war.

Das Ziel des Fedora-Projekts ist es, etwa alle 6 Monate eine neue Version zu veröffentlichen. Updates für die einzelnen Versionen werden nur jeweils zwei bis drei Monate nach Erscheinen der nächsten Version angeboten – dies ist verglichen mit anderen Distributionen eine relativ kurze Zeit. [8]

Weiterer Nachteil dieser Distribution ist, dass diese auf einer bestimmten Anzahl DVDs bzw. CDs ausgeliefert wird. Dabei wurde von Red Hat eine Vorauswahl getroffen, welche Software mitgeliefert wird. Dadurch ist auf den CDs / DVDs eine Menge Programme enthalten, die gar nicht gebraucht werden. Außerdem wird bei der Installation eine Menge Software mit installiert, die nicht benötigt wird, was ein Sicherheitsrisiko darstellen kann.

### **3.1.2.3 SUSE Linux Professional / openSUSE**

openSUSE, (ehemals „SuSE Linux Professional“) ist eine Linux-Distribution der Firma Novell, bzw. deren Tochterunternehmen Suse Linux GmbH (Software- und System-Entwicklungsgesellschaft mbH).

SUSE gilt im Allgemeinen als sehr anwenderfreundlich und sorgt vor allen Dingen mit den grafischen Konfigurationswerkzeugen dafür, dass Windows-Anwender einen Wechsel hin zu Linux wagen und sich dort relativ schnell zu Recht finden.

Die größte Besonderheit der Distribution ist das hauseigene Installations- und Konfigurationswerkzeug YaST, das eine zentrale Schnittstelle für das gesamte System darstellt. Ebenso findet sich dort SaX2 („SUSE advanced X11 configuration“), das die Einrichtung der Grafikkarte, Maus und Tastatur vornimmt.

Rückblickend hatte sich SUSE im Vergleich vor allen Dingen durch eine fortgeschrittene Unterstützung und Integration von ISDN-Karten, die gegenüber dem Standard-Kernel deutlich erweiterte Treiberauswahl, das sehr frühzeitige Integrieren des K Desktop Environments und einen großen Paketumfang wesentlich unterschieden.

Eine weitere Besonderheit stellen die SUSE-Support-Datenbank und die SUSE-Component-Database dar. Beide stellen umfangreiche Informationen und Hilfestellungen für die Installation und Konfiguration des Systems und der entsprechenden Hardware zur Verfügung. Dabei ist vor allen Dingen die Hardware-Datenbank auch von Anwendern anderer Distributionen rege genutzt worden. Beide Mechanismen waren ebenfalls auf den Medien der Distribution enthalten und ergänzten so die Produkt-Dokumentation. [9]

Bei SUSE schlagen aber die gleichen Nachteile zu Buche, die schon bei Red Hat erläutert wurden. Darüber hinaus birgt die Verwendung von grafischen Assistenten die Gefahr, dass sie unsicher sind oder leicht manipuliert werden können.

### **3.1.3 Auswahl des Betriebssystems**

Bei der Auswahl des Betriebssystems fiel die Entscheidung auf Debian, das sich frei aus dem Internet herunterladen lässt. Ein Ausschlag gebender Grund für die Verwendung dieses Linux war, dass es sich dabei um die stabilste Distribution handelt. Diese besondere Stabilität lässt sich begründen und erzielen, in dem neuere Komponenten erst nach langen und intensiven Tests in die Distribution aufgenommen werden. Diese Komponenten können im wesentlichen Kernel und andere Anwendungssoftware sein.

Debian setzt sich aber nicht nur im Vergleich zwischen den einzelnen Distributionen durch sondern auch gegenüber Windows. Neben den lizentechnischen Gründen spielen weitere Gründe eine Rolle, warum die Wahl auf Linux als Betriebssystem gefallen ist. So ist in Anbetracht der Sicherheit Linux klar im Vorteil, da im Linux-Umfeld die Häufigkeit, mit der potentielle Gefahren auftreten, längst nicht so hoch ist wie unter Windows. Ein weiterer nicht Ausschlag gebender Grund ist, dass das von der Firma Envidatec entwickelte Energie- und Betriebsdatenportal JEVis bisher nur als Linux-Version existiert.

Die Art und Weise dieser Anwendung erlegt dem Server auf, dass er zuverlässig läuft. Auch nur ein kurzer Ausfall, kann im Hinblick auf die spätere Verwendung des Servers richtig kritisch und wahrscheinlich auch richtig teuer werden. Im Falle eines Ausfalls der wichtigsten Komponente, dem Mainboard, kann unter Umständen bei Windows eine komplette Neuinstallation des Systems notwendig werden. Das hängt im Allgemeinen

davon ab, in wie weit sich das neue Board vom alten unterscheidet. Bei Linux kann aller Wahrscheinlichkeit nach das Mainboard getauscht werden und das System läuft ohne Veränderungen weiter.

## 3.2 Datenbank

Für eine Betrachtung kommen zunächst die Datenbanken MySQL, Microsoft SQL Server, Oracle und IBM DB2 in Frage. Der Microsoft SQL Server kann gleich aus der Betrachtung ausgeschlossen werden, da dieser einen Windows Server voraussetzt und nicht unter dem oben gewählten Linux läuft. Wenn als nächstes Kriterium der Preis für das Produkt herangezogen wird, dann lässt sich die IBM DB2 schnell ausschließen. Somit bleiben für die Betrachtung nur MySQL und Oracle übrig.

Wenn als Kriterium einzig der Preis zugelassen ist, dann kann die Entscheidung nur auf MySQL fallen, da es sich hier um die preisgünstigste Datenbank handelt. Aber die nächste zu stellende Frage ist, ob es sinnvoll ist, nur nach dem Preis zu entscheiden. Im Hinblick auf die spätere Verwendung der Datenbank spielen aber noch andere Dinge eine Rolle, die die Entscheidung möglicherweise beeinflusst.

Für eine an die Anwendung optimal angepasste Datenbank spielen wichtigere Dinge eine Rolle als nur der Preis. Einerseits lässt sich MySQL nur schwer mit Oracle oder den anderen oben genannten Datenbanken vergleichen. Andererseits reicht der Funktionsumfang von MySQL bei weitem nicht an den der anderen Datenbanken heran.

Dafür lassen sich einige Beispiele finden, wie sichere Transaktionen, Cursor-Programmierung, Fremdschlüssel und andere Constraints, Trigger sowie „Gespeicherte Prozeduren“ (Stored Procedures). Viele von denen sind auch schon in der Version 5 von MySQL vorhanden. Da diese aber erst in der jüngeren Vergangenheit implementiert wurden, sind sie noch wenig ausgereift und können das eine oder andere Mal zu Problemen führen.

Alles in allem ist man mit Oracle auf der sicheren Seite, was eine zuverlässige Datenbank anbelangt, da diese speziell für solche Anwendungen entwickelt wurde. In dem

Forschungsprojekt ist außerdem der Umstand gegeben, dass die Lizenzen für Oracle nicht beschafft werden müssen, sondern in ausreichender Zahl in der HAW zur Verfügung stehen. So spielt das Argument, dass Oracle wesentlich teurer ist, keine Rolle. Aus diesen eben genannten Gründen wurde Oracle für das Projekt gewählt.

# 4 Installation

Mit dem Server, der jetzt physikalisch vorhanden ist, erfolgt in diesem Kapitel eine Beschreibung, wie die benötigte Software auf dem Server zu installieren ist. Dabei wird in mehreren Schritten vorgegangen, wobei im ersten Schritt das Betriebssystem installiert wird. Zunächst erfolgt eine Grundinstallation, die im Folgenden dann konfiguriert wird. Nach Abschluss dieses Vorganges ist der Computer bereits lauffähig, der installierte Rechner ist aber noch kein Server. Erst die sich jetzt anschließenden Installationen der beiden Webserver und des Java Development Kits machen aus dem normalen Rechner einen Server.

Entgegen einer Normalinstallation ist es erforderlich, auf diesem Rechner zwei Webserver zu installieren. Das hat den Grund, dass so genannte Java Server Pages (JSP) ausgeführt werden müssen. Diese Funktionalität stellt der Tomcat – Webserver bereit.

Im letzten Schritt erfolgt dann die Installation der Oracle Datenbank, die Teile der vorherigen Installationen benötigt.

Da sich die Konfiguration beim Server und beim Desktop-PC unterscheidet, liegt dem Server eine CD bei. Mit dem Konfigurationsprogramm (Fujitsu Siemens Enterprise Manager) auf dieser CD werden die Komponenten des Servers eingestellt. In erster Linie muss die Konfiguration der Festplatten geändert werden, damit diese paarweise im RAID1-Verbund vorliegen.

## 4.1 Betriebssystem

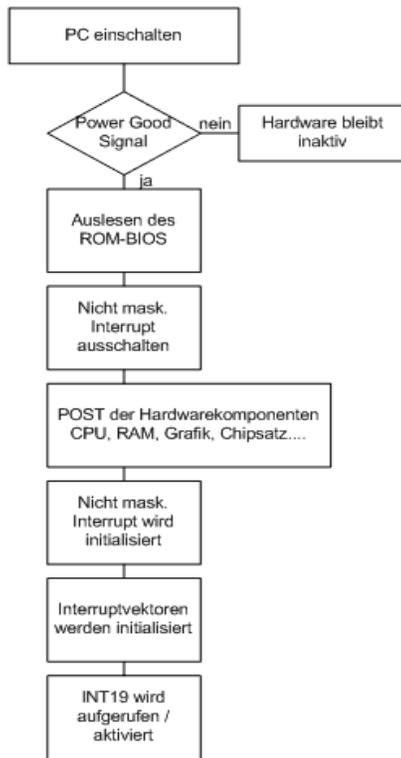
Die Entscheidung fiel am Anfang auf das Linux Betriebssystem Debian. Die Distribution ist auf einer DVD sowie auf mehreren CDs, die sich frei aus dem Internet herunterladen lassen, verfügbar. Da ein Kriterium war, nur so wenig wie möglich an Software zu installieren, wurden nicht die kompletten DVDs bzw. CDs heruntergeladen, sondern nur eine so genannte NetInstall-CD, auf der nur das Grundsystem von Debian vorhanden ist. Neben

dem Vorteil, dass nur ein Grundsystem installiert wird, werden alle weiter benötigten Pakete in der aktuellsten Version direkt aus dem Internet heruntergeladen.

### **4.1.1 Grundsystem**

Zur Installation des Grundsystems wird der Rechner mit der CD gebootet, d. h., es werden die zur Installation benötigten Programme und Einstellungen von der CD geladen. Was dabei genau abläuft, zeigt beispielhaft der unten stehender Programmablaufplan.

Der Bootvorgang eines von Neumann-Rechners bis INT19



Der Bootvorgang eines von Neumann-Rechners ab INT19

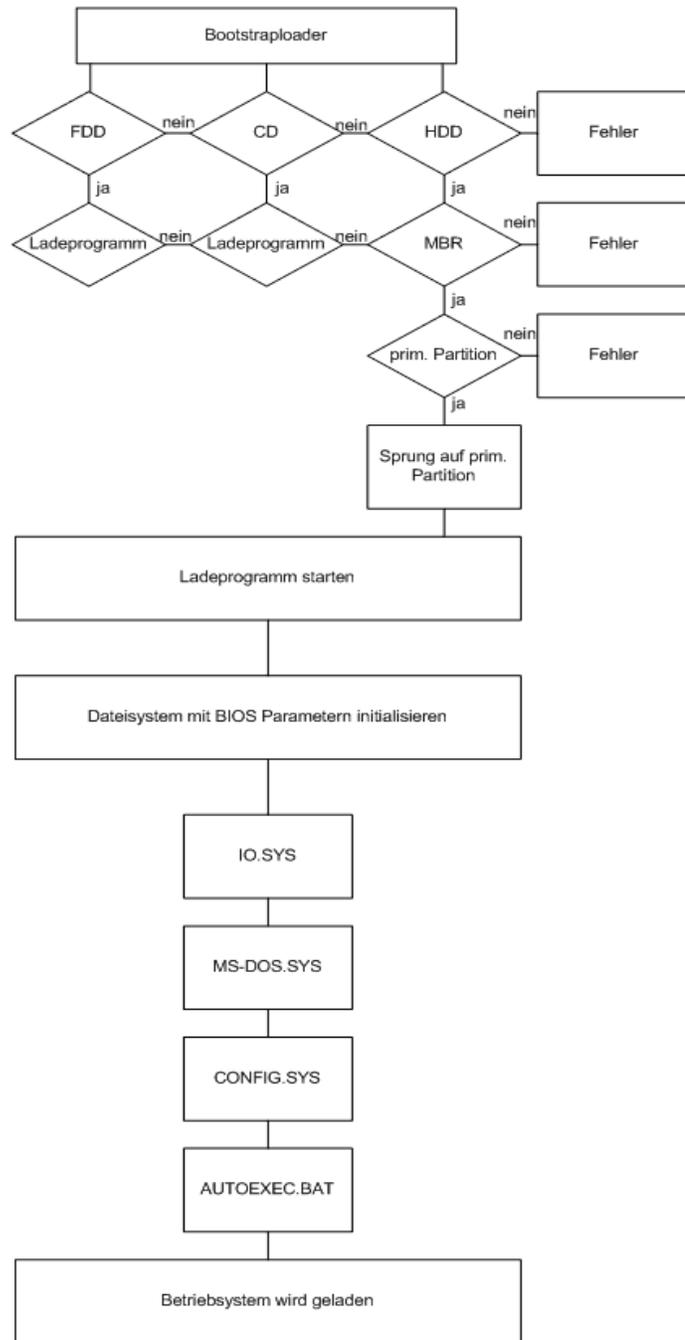


Bild 4: Bootvorgang am Beispiel von DOS[10]

Ist der Bootvorgang abgeschlossen, erscheint folgender Bildschirm, der den Ausgangspunkt der Installation darstellt. Mit der Taste F1 wird eine Hilfeseite zu der Installation angezeigt sowie weitere Informationen zu den einzelnen Optionen.



Bild 5: Installationsbildschirm Debian

- F1 - Übersicht der Hilfefunktionen
- F2 - Voraussetzungen für die Installation von Debian
- F3 - Spezielle Methoden, um das System zu starten
- F4 - Überblick über spezielle Bootparameter
- F5 - Bootparameter für exotische Hardware
- F6 - Bootparameter für ausgewählte Plattencontroller
- F7 - Bootparameter für das Installationssystem
- F8 - Informationen darüber, wie man weitere Hilfe bekommt
- F9 - Informationen über das Debian-Projekt
- F10 - Copyright-Informationen

Mit der Taste F3 können noch spezielle Bootmethoden eingestellt werden. Grundsätzlich kann dort zwischen zwei Installationsmodi in jeweils zwei Kernelvarianten entschieden werden. So steht „linux“ für einen Einsteigermodus und „expert“ für einen Expertenmodus. Die 26 jeweils hinter linux und expert bezieht sich auf die Version des Kernel 2.6. Das entspricht der anderen Auswahlmöglichkeit. Mit dieser Einstellung wird nicht - wie

standardmäßig vorgegeben, der Kernel 2.4.x geladen, sondern der der 2.6.x-Reihe. Für die beabsichtigten Zwecke reicht die Standardeinstellung. Es wird dann mit der Installation begonnen, indem die ENTER-Taste gedrückt wird. Dieser Tastendruck veranlasst den Start des eigentlichen Installationsprogramms. Dazu wird ein Bootimage mit dem Kernel 2.4 geladen, welches textbasiert durch die Installation führt.

Im ersten Schritt erfolgt eine Spracheinstellung. Da diese aber unwichtig für die eigentliche Installation ist, wird sie hier weggelassen und erst im Anhang beschrieben. Das gilt genauso für die jetzt folgende Ländereinstellung und die Wahl des Tastaturlayouts. Mit den Werten, die von der Installationsroutine voreingestellt wurden, kann hier nichts falsch gemacht. Sie lassen sich einfach übernehmen.

Im Anschluss werden noch einige weitere Komponenten zur Installation geladen und versucht, die Netzwerkkarte mittels DHCP (Dynamic Host Configuration Protocol) einzurichten. Dies geschieht, in dem die Netzwerkkarte einen Broadcast-Befehl an den DHCP-Server schickt. Als Antwort erhält die Netzwerkkarte dynamisch eine IP-Adresse zugeteilt, damit der Rechner im Netzwerk erreichbar ist. Außerdem werden der Karte noch die Einstellungen für den DNS-Server und dem Standard-Gateway mitgeteilt. Diese Einstellungen sind extrem wichtig, da sich später die Installation nur mit einem funktionierenden Internetzugang fortsetzen lässt. Die Methode, den Computer mit DHCP einrichten zu lassen, ist die komfortabelste. Alle erforderlichen Einstellungen werden automatisch vorgenommen.

In bestimmten Fällen kann diese Methode aber auch nicht sinnvoll sein. Wenn der Rechner von außen erreicht werden soll, werden diese Anfragen vom Router durch Network Address Translation (NAT) an die jeweilige IP-Adresse des Rechners weitergeleitet. Wird dabei aber DHCP verwendet, kann es vorkommen, dass der Rechner nicht dieselbe Adresse wiederbekommt und somit von außen nicht mehr erreichbar ist. Für einen solchen Fall sollte im Netzwerk kein DHCP-Server vorhanden sein. Die Netzwerkkarte lässt sich auch manuell einrichten, wobei eine statische IP-Adresse vergeben wird.

Die Konfiguration der Netzwerkkarte setzt bestimmte Kenntnisse voraus. So ist die Kenntnis erforderlich, welche IP-Adresse vergeben werden kann und welche die richtigen Einstellungen für die oben schon erwähnten DNS-Server und Standard-Gateway sind. Unabdingbar ist aber, dass im Netzwerk ein Router oder Computer vorhanden sein muss,

der eine permanente Verbindung zum Internet herstellt. Ist dies nicht der Fall, lässt sich Debian per NetInstall-CD nicht installieren. Dadurch ist es erforderlich, sich doch für eine andere Installationsmethode entscheiden. Ursache hierfür ist, dass während der Installation keine Wählverbindung zu einem Provider aufgebaut werden kann und später erfolglos versucht wird, auf das Internet zuzugreifen.

Die beiden nächsten Punkte, nämlich die Vergabe des Rechnernamens und die Einstellung einer Domäne, sind für eine erfolgreiche Installation eher unwichtig, so dass hier nicht näher darauf eingegangen wird. Auch hier können die bei der Installation ermittelten Vorgaben so übernommen werden. Sollten diese Einstellungen nicht der Realität entsprechen, lassen sie sich auch später noch ändern.

Ist der Vorgang abgeschlossen, werden als nächstes die Festplatten partitioniert. Eine Partitionierung unterteilt die Festplatte in mehrere Abschnitte. Diese Partitionen werden später wie echte Festplatten behandelt, sorgen aber gleichzeitig für etwas mehr Sicherheit. Grund für die Partitionierung ist es, dass bei entstanden Fehlern im Home-Verzeichnis nur diese Partition betroffen wäre und nicht das gesamte System.

Beim Partitionierungsvorgang sind jetzt zwei Festplatten zu sehen, eine kleinere für das Betriebssystem und eine größere für die Daten. Zu Beginn der Partitionierung fällt die Wahl auf die kleinere der beiden Festplatten, die aller Wahrscheinlichkeit nach, die Bezeichnung sda trägt. Sda beschreibt unter Linux, um welche Festplatte es sich dabei handelt. So kennzeichnet der Buchstabe s, dass eine SCSI Festplatte installiert ist. Das d steht für Disk und der letzte Buchstabe nummeriert die Festplatten der Reihe nach. So besagt sda, dass es sich dabei um die 1. SCSI Disk handelt.

Im nächsten Fenster lässt sich auswählen, ob eine leere Partitionstabelle angelegt werden soll. Nachdem diese Aktion mit <Ja> bestätigt wurde, gelangt der Bildschirm erneut zur Ansicht, auf dem die im System vorhandenen Festplatten aufgeführt sind. Unter der eben ausgewählten Festplatte taucht nun ein Eintrag für freien Speicher auf, der genauso groß sein sollte wie die Kapazität der Festplatte. Dieser Eintrag lässt sich mit ENTER auswählen und leitet die Erstellung einer neuen Partition ein. Für die Größe der Partition wird 10 GB angegeben, was bewirkt, dass eine Partition genau mit dieser Größe erstellt wird. Da die Partition später primär sein soll, um davon booten zu können, wird mit dem entsprechenden Eintrag der Typ ausgewählt.

Diese Partition wird an den Anfang der Festplatte geschrieben. Die Verwendung im nächsten Fenster kennzeichnet, wie die Partition verwendet werden soll. Sinnvoller Weise wird zuerst eine Partition erstellt, die als Wurzelverzeichnis „/“ verwendet wird. Dieser Vorgang wird einige Male wiederholt, bis alle benötigten Partitionen erstellt wurden. Eine mögliche Partitionstabelle zeigt das nachfolgende Bild.

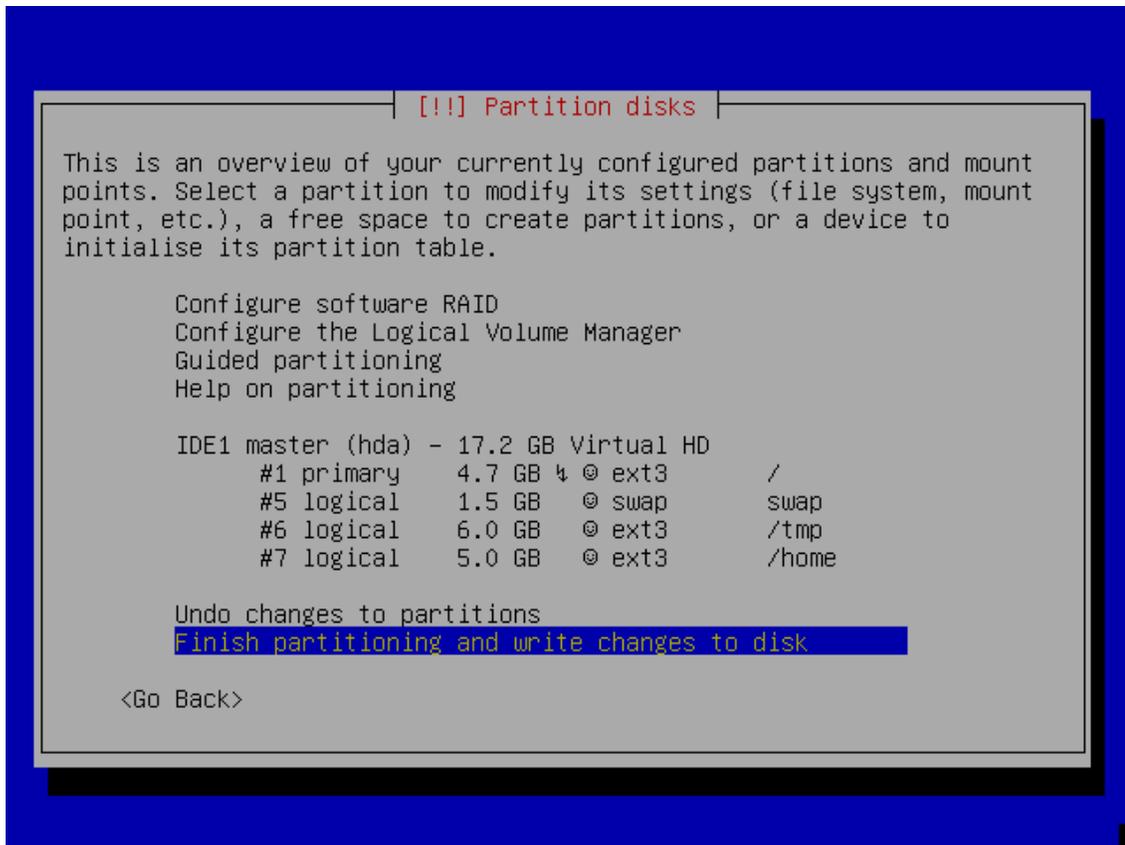


Bild 6: Beispiel einer Partitionstabelle

Entspricht die Partitionstabelle den Vorstellungen, wird der Vorgang mit dem „Beenden der Partitionierung“ abgeschlossen. Für die 2. Festplatte ist der Vorgang genauso anzuwenden, nur, dass für diese Festplatte, auf der später die Daten liegen sollen, lediglich eine Partition benötigt. Nur bei der Art, wie die Festplatte verwendet werden soll, ergibt sich eine Änderung im Vorgehen.

Es wird keine vorgefertigte Verwendungsart ausgewählt, sondern mit dem Menüpunkt „von Hand eingeben“ werden Namen (z.B. Daten) für die Festplatte eingetragen. Der gesamte Partitionierungsvorgang wird dann mit „Partitionierung beenden und Änderungen übernehmen“ abgeschlossen, um die jeweiligen Partitionstabellen auf die dazugehörigen

Festplatten schreiben zu lassen. Im Anschluss darauf werden die eben erstellten Festplatten bzw. die einzelnen Partitionen auf den Festplatten noch formatiert, um sie benutzen zu können.

Sind diese Vorbereitungen getroffen, wird jetzt das Grundsystem von Debian auf dem System installiert. Dazu werden die auf der CD vorhandenen Pakete auf die Festplatte kopiert, entpackt und eingerichtet.

Zum Abschluss dieses Installationsteils wird noch der GRUB (GRand Unified Bootloader) auf die Festplatte geschrieben. Mit ihm kann ausgewählt werden, welches Betriebssystem gebootet werden soll, wenn sich auf dem Rechner mehrere Betriebssysteme befinden. Bevor der Rechner neu gebootet wird, muss noch die CD aus dem Laufwerk entfernt werden, sonst startet der Rechner nicht das Betriebssystem von der Festplatte, die Installation wird erneut ausgeführt.

### **4.1.2 Konfiguration Debian**

Ist die Grundinstallation in Ordnung und wurde der Rechner neu gebootet, sollte der Begrüßungsbildschirm von Debian erscheinen.

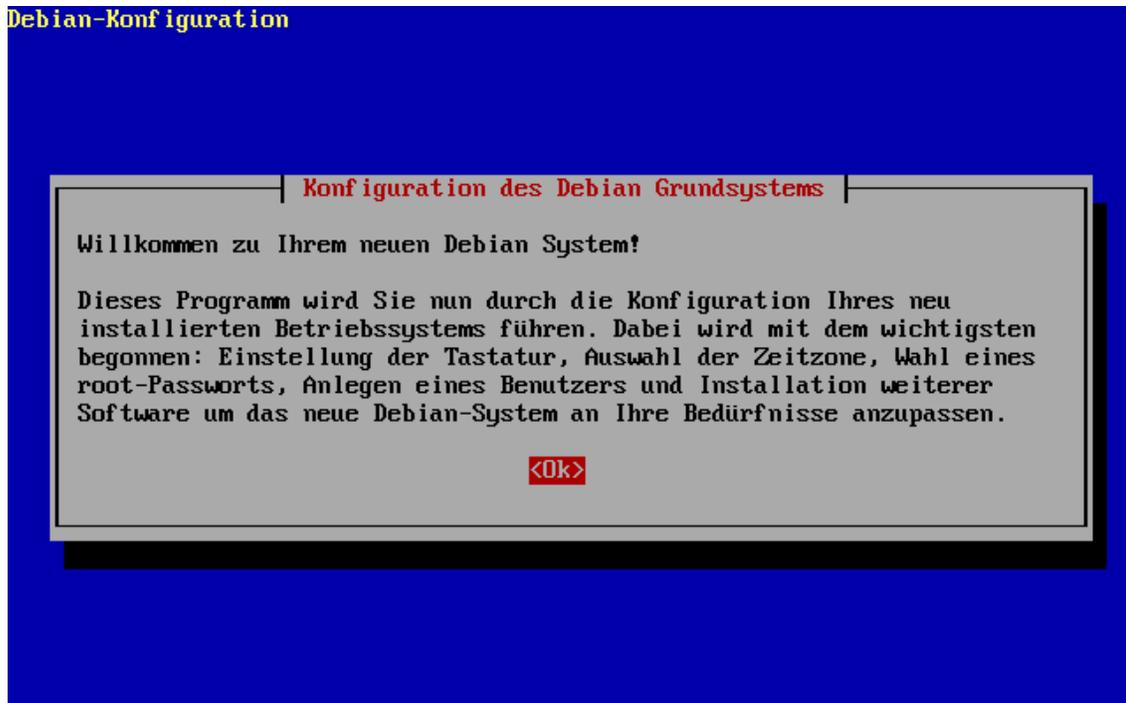


Bild 7: Willkommensbildschirm Debian

Dabei wird in diesem Fenster außerdem bekannt gegeben, was in den nächsten Schritten für Konfigurationen vorgenommen werden. Die zwischenzeitlich durchgeführten Einstellungen von Systemuhr, Zeitzone und die Quelle für den apt-Dienst sind nicht entscheidend, so dass sie ausführlich erst im Anhang behandelt werden.

Im nächsten Schritt erfolgt die Festlegung des Administrator-Passwortes, unter Linux heißt der Administrator-Account root. Wenn dieser Server später produktiv eingesetzt wird, dann sollte dieses Passwort möglichst sicher sein, um ein Eindringen in das System zu verhindern. Dieser Account ist der mächtigste auf einem Linux-System und hat damit Zugriff auf alle Bereiche des Systems. Außerdem darf das Passwort nicht vergessen werden, da später keine Möglichkeit besteht, wieder in das System hinein zu kommen.

Um als Administrator nicht mit dem root-Account arbeiten zu müssen, sollte für diesen noch ein ganz normaler Benutzer-Account eingerichtet werden. Eventuelle Systemmeldungen werden dann an diesen Benutzer weitergeleitet. Die Einstellungen dazu erfolgen während der Konfiguration des Debian-Systems.

Für diesen Benutzer werden zuerst der vollständige Name und danach ein Anmeldenamen abgefragt. Die Standardeinstellung für den Anmeldenamen, den das System aus dem Vornamen des Benutzers bildet, wurde übernommen. Es ist auch möglich, hier individuell einen anderen Namen zu vergeben, der den Vorstellungen des Anwenders entspricht. Auch für diesen Benutzer muss ein Passwort angegeben werden.

Wichtig ist auch in diesem Fall, die Sicherheitsrichtlinien (Große und kleine Buchstaben, Zahlen, Sonderzeichen sowie eine Passwortlänge von min. 8 Zeichen) für Passwörter zu beachten und kein Passwort zu verwenden, was sich leicht erraten lässt, wie z. B. der Geburtstag. Die sicherste Methode ist, ein Passwort aus der Kombination von Klein- und Großbuchstaben, Zahlen sowie Sonderzeichen zu wählen.

Nun erscheint eine Auswahl darüber, welche Software installiert werden soll.



Bild 8: Softwareauswahl unter Debian

Der Server wird nach den eigenen Anforderungen installiert. Es wird kein Punkt ausgewählt, damit nur die Installation der wirklich gebrauchten Pakete vorgenommen werden kann. Dieses Vorgehen dient dazu, auf dem Server so wenig wie möglich Ansatzpunkte für eventuelle Angriffe zu schaffen. Wie die installierten Programme noch weiter abgesichert werden können, ist in Kapitel 8 erläutert.

Nach der Softwareinstallation erfolgt die Konfiguration des exim V4, dem Mailserver unter Linux. Der Mailserver wird nur zum Zustellen der Systemmitteilungen gebraucht und deshalb ist nur eine Standardeinstellung des Mailservers notwendig. Aus diesem Grund und weil auch später keine Änderung an der Konfiguration des Mailservers vorgenommen werden soll, wird gegen eine Aufteilung in kleine Konfigurationsdateien entschieden. Wie bereits erwähnt, sollen nur Systemnachrichten zugestellt werden, somit wird der Mailserver nur für lokale E-Mailzustellung konfiguriert. Diese Einstellung dient auch der Sicherheit, da keine Netzwerkverbindung zu dem Mailserver möglich ist.

Im Folgenden wird die Netzwerkkonfiguration für den Mailserver angegeben, wobei der eingestellte Wert für die IP-Adresse übernommen wird. Die vorgegebene Adresse von 127.0.0.1, trägt auch den Namen „localhost“, der so genannte Loopback-Adapter unter Linux. Der Loopback-Adapter bezweckt, dass der Anwender Anfragen an sich selbst schicken kann.

Diese Einstellung hat zur Folge, dass der Mailserver nur auf dem eigenem System „lauscht“ und sich nicht dafür interessiert, was sonst so im Netzwerk passiert. Zum Abschluss wird noch der Benutzername angegeben, an den die Nachrichten für die Systemkonten weitergeleitet werden. Die Angabe mehrerer Konten ist möglich und sinnvoll, wenn mehrere Benutzer diese Nachrichten erhalten sollen.

Damit ist die Installation des Debian Linux Servers abgeschlossen und dieser kann nun verwendet werden. Die Installation der einzelnen Applikationen erfolgt im nächsten Schritt, da diese für den Betrieb des Servers notwendig sind. Nach Abschluss der Konfiguration des Systems erscheint die Konsole, an deren „login:“-Prompt die Anmeldung als Administrator bzw. root erfolgen kann. Die grafische Oberfläche wird hier vielleicht vermisst, die der Benutzer von Windows gewohnt ist.

Es gibt auch eine Oberfläche für Linux, doch aus Performancegründen wird darauf verzichtet. Die Installationen und Einstellungen können auch ohne grafische Oberfläche auf der Konsole vorgenommen werden. Eine Oberfläche wird dazu nicht gebraucht.

Ist die Anmeldung an der Konsole erfolgt, wird mit dem Befehl

```
# apt-get update
```

ein Update der Paketliste von apt durchgeführt. Bei anschließender Eingabe des Befehls

```
# apt-get upgrade
```

wird für alle installierten Pakete nachgeschaut, ob eine neue Version der Software zur Verfügung steht. Sollte dabei eine Fehlermeldung auftauchen, hat es damit zu tun, dass Abhängigkeiten von einigen Paketen nicht erfüllt sind. Um das zu beheben, wird der Befehl ausgeführt, der auch in der Abbildung zu sehen ist.

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
oral:~# apt-get update
OK  ftp://ftp.uni-erlangen.de stable/main Packages
OK  ftp://ftp.uni-erlangen.de stable/main Release
OK  http://security.debian.org stable/updates/main Packages
OK  http://security.debian.org stable/updates/main Release
OK  ftp://ftp.uni-erlangen.de stable/main Sources
OK  ftp://ftp.uni-erlangen.de stable/main Release
Paketlisten werden gelesen... Fertig
oral:~# apt-get upgrade
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Sie möchten wahrscheinlich »apt-get -f install« aufrufen, um dies zu korrigieren
.
Die folgenden Pakete haben nichterfüllte Abhängigkeiten:
  kernel-image-2.4.27-3-386: Hängt ab: initrd-tools (>= 0.1.48) ist aber nicht i
ninstalliert
E: Nichterfüllte Abhängigkeiten. Versuchen Sie, -f zu benutzen.
oral:~#
```

Bild 9: Fehlermeldung bei Durchführung eines Updates

```
# apt-get -f install
```

Dieser Befehl bewirkt, dass versucht werden soll, die fehlenden Pakete zu installieren, auch wenn eine Integritätsprüfung fehlschlägt. In vorliegendem Fall werden 2 Pakete aktualisiert und 5 weitere Pakete neu installiert. Anschließend wird nochmals eine kurze Konfiguration durchgeführt. Dabei ist es jetzt möglich, mehrere lokale Einstellungen zu verwenden, um möglichen Benutzern ihre individuelle Spracheinstellung zu ermöglichen.

Die Einstellung, die während der Installation des Systems vorgenommen wurde, wird belassen. Da nur diese eine Systemeinstellung verwendet wurde, wird diese auch für die Standardeinstellung der Systems-Umgebung genutzt. Dieses hat zur Folge, dass ein Upgrade für die C-Bibliothek glibc durchgeführt werden muss.

Als Hinweis wird noch gegeben, dass bestimmte Pakete – wie die Bildschirmmanager kdm, xdm, gdm usw. - gestoppt werden müssen. Da nicht alle Pakete installiert wurden, ist dieser Hinweis nicht weiter zu beachten. Da einige grundlegende Systemdienste notwendigerweise installiert sind, müssen diese neu gestartet werden. Bei diesen Diensten handelt es sich um den Dämon für zeitgesteuerte Abläufe atd, den Internet Superdämon inetd, den Taskmanager cron und den Mailserver exim4. Ist das Neustarten erfolgreich abgeschlossen, kann der Befehl

```
# apt-get upgrade
```

nochmals ausgeführt werden.

Diesmal wird der Befehl ohne Beanstandungen ausgeführt, und es werden 85 Pakete aktualisiert. Dazu wird abschließend das Debian Konfigurationsmanagementsystem debconf angepasst, in dem die hier vorgeschlagenen Einstellungen übernommen werden.

Das trifft auch auf die Priorität zu, ab wann die Fragen nach neuen Paketen ignoriert werden. Dies hat den Hintergrund, dass die zur Verfügung gestellten Pakete in die Kategorien kritisch, hoch, mittel und niedrig eingeteilt worden sind. Mit der übernommenen Einstellung werden alle Pakete mit der Priorität niedrig ignoriert.

Eine weitere Konfiguration ist für die man-db notwendig. Hierbei handelt es sich um die Hilfeseiten zu den einzelnen Befehlen, die unter Linux vorhanden sind. Diese Seiten geben Unterstützung, wie die Befehle einzusetzen sind und wie die Syntax der Befehle lautet. Auch hier können ohne Bedenken die vorgegebenen Werte übernommen werden.

Zuletzt wird noch nach der Tastaturbelegung für die Konsole gefragt. Änderungen werden nicht vorgenommen und die Einstellungen werden – wie vorgegeben – übernommen.

Ist dieser Vorgang abgeschlossen, kann mit der eigentlichen Installation der benötigten Anwendungssoftware begonnen werden.

## 4.2 Java-Technologie

Für die Installation der Java-Technologie ist es erforderlich, die apt-Source-Liste `/etc/apt/sources.conf` anzupassen. Siehe dazu die angepasste Liste im Anhang. Um die neueren Versionen von Java nutzen zu können, wird eine Source aus dem Etch-Zweig hinzugefügt. Unter Debian bekommen die einzelnen Versionen Namen. Etch ist die momentan aktuelle Version von Debian und entspricht dem Debian Linux 4.

Die aktuelle Version lag bei der Installation noch nicht vor. Grundsätzlich funktioniert die Installation mit der neueren Version genauso, es müssen jedoch einige Einstellungen nicht mehr vorgenommen werden.

Mit dem Befehl

```
# nano /etc/apt/sources.conf
```

wird folgender Eintrag

```
deb ftp://ftp.de.debian.org/debian/ etch main non free contrib
```

in die oben erwähnte Datei `sources.conf` eingeführt. Mit dem Aufruf

```
# apt-get update
```

werden die Paketlisten um die Einträge aus dem Etch-Bereich erweitert. Dies ist notwendig, um die neueren Versionen von Java nutzen zu können. Da das Java SE Development Kit (JDK), anders als bei Etch, nicht als Paket verfügbar ist, muss man Java manuell installieren. Dazu muss

```
jdk-1_5_0_12-linux-i586.bin
```

als Linux selbst extrahierende Datei von <http://java.sun.com> heruntergeladen werden. Für die Installation ist es unbedingt erforderlich, diese Datei herunterzuladen, da Debian mit den RPM-Dateien nicht zu Recht kommt.

Zuerst wird ein Wechsel in das `/tmp` Verzeichnis vorgenommen und anschließend ausgeführt.

```
wget -O jdk-1_5_0_06-linux-i586.bin
http://192.18.108.149/ECom/EComTicketServlet/BEGIN5C5902DBA428CEF9DA9F4B38862
185DF/-
2147483648/2207627667/1/832394/832190/2207627667/2ts+/westCoastFSEND/jdk-
1.5.0_12-oth-JPR/jdk-1.5.0_12-oth-JPR:5/jdk-1_5_0_12-linux-i586.bin
```

Der Parameter `-O` muss mit angegeben werden, weil der von Sun generierte Link zu lang ist und `wget` den Download nicht ausführen kann.

Ist die Java-Datei heruntergeladen, wird mit

```
# apt-get install java-package
```

ein Java spezifisches Paket installiert. In diesem Paket ist ein Script `make-jpkg` enthalten, das die heruntergeladene Datei in ein Debian-Paket übersetzt.

Da einige Java-Skripte das System direkt verändern und diese Veränderung nicht erwünscht ist, wird die weitere Erstellung des Debian-Pakets nicht weiter als `root` ausgeführt, sondern es wird eine Generierung als normaler Benutzer vorgenommen.

Dazu wird mit Hilfe des Befehls

```
# su oliver
```

der `root` zu einem Benutzer und der Vorgang gestartet.

```
# fakeroot make-jpkg jdk-1_5_0_12-linux-i586.bin
```

Danach wird automatisch das Archiv entpackt und in ein temporäres Verzeichnis geschrieben. Nach Bestätigung der Lizenz von Sun wird das Paket für Debian generiert und in das Verzeichnis /tmp geschrieben.

Im Hinblick auf die Sicherheit sollte dieser gesamte Vorgang nicht auf dem Produktivserver ausgeführt werden. Ein besserer Weg wäre, die Erstellung des deb-Pakets auf einem separaten Computer durchzuführen und dieses Paket dann anschließend auf dem Server bereitzustellen und es anschließend zu installieren. Die Installation wird eingeleitet, in dem der Befehl

```
# dpkg -i sun-sdk1.5_1.5.0+update12_i386.deb
```

ausgeführt wird. Dazu ist es erforderlich, sich mit

```
# su
```

auf der Konsole als root anzumelden oder mit Hilfe von sudo den Befehl auszuführen.

```
# sudo dpkg -i sun-sdk1.5_1.5.0+update12_i386.deb
```

Da es für Java in regelmäßigen Abständen neue Versionen gibt, wird ein symbolischer Link auf die gerade installierte Version angelegt, um mit diesem Link immer Zugriff auf die aktuellste installierte Version zu haben. Der Link wird immer angepasst, so dass die Änderungen in den Konfigurationsdateien wegfallen, was den Vorteil hat, dass keine Datei vergessen werden kann.

Um den Link anzulegen, wird auf der Konsole

```
# ln -s /usr/lib/j2sdk1.5-sun /usr/lib/java
```

einggegeben.

Damit vom ganzen System aus auf Java zugegriffen werden kann, wird die Profildatei unter /etc/profile angepasst.

```
# nano /etc/profile
```

Direkt in die erste Zeile wird

```
JAVA_HOME="/usr/lib/java"
```

geschrieben und die beiden PATH-Zeilen

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11"
```

```
PATH="/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games"
```

werden um den Eintrag ergänzt

```
:$JAVA_HOME/bin
```

und am Ende der export PATH -Anweisung wird der Befehl JAVA\_HOME PS1 angefügt.

Die komplette Zeile lautet dann

```
export PATH JAVA_HOME PS1
```

Ist Java erfolgreich installiert worden, lässt sich dies mit

```
# java -version
```

überprüfen und als Ausgabe erscheint

```
java version "1.5.0_12"
```

```
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_12-b04)
```

```
Java HotSpot(TM) Client VM (build 1.5.0_12-b04, mixed mode, sharing)
```

genau die Version, die installiert wurde.

## 4.3 Apache – Webserver

Die Installation des Apache –Webservers der Version 2 ist ganz einfach, da dafür ein Debian-Paket vorhanden ist. Dazu muss auf der Konsole nur

```
# apt-get install apache2
```

eingegeben werden, denn die Pakete für den Apache – Webserver sind in den Paketlisten von Debian vorhanden. Alle benötigten Komponenten werden heruntergeladen und installiert. Außerdem wird der Webserver auch vollständig ohne Zutun des Anwenders konfiguriert. Ist dieser Vorgang abgeschlossen, sollte im Browser folgendes zu sehen sein.

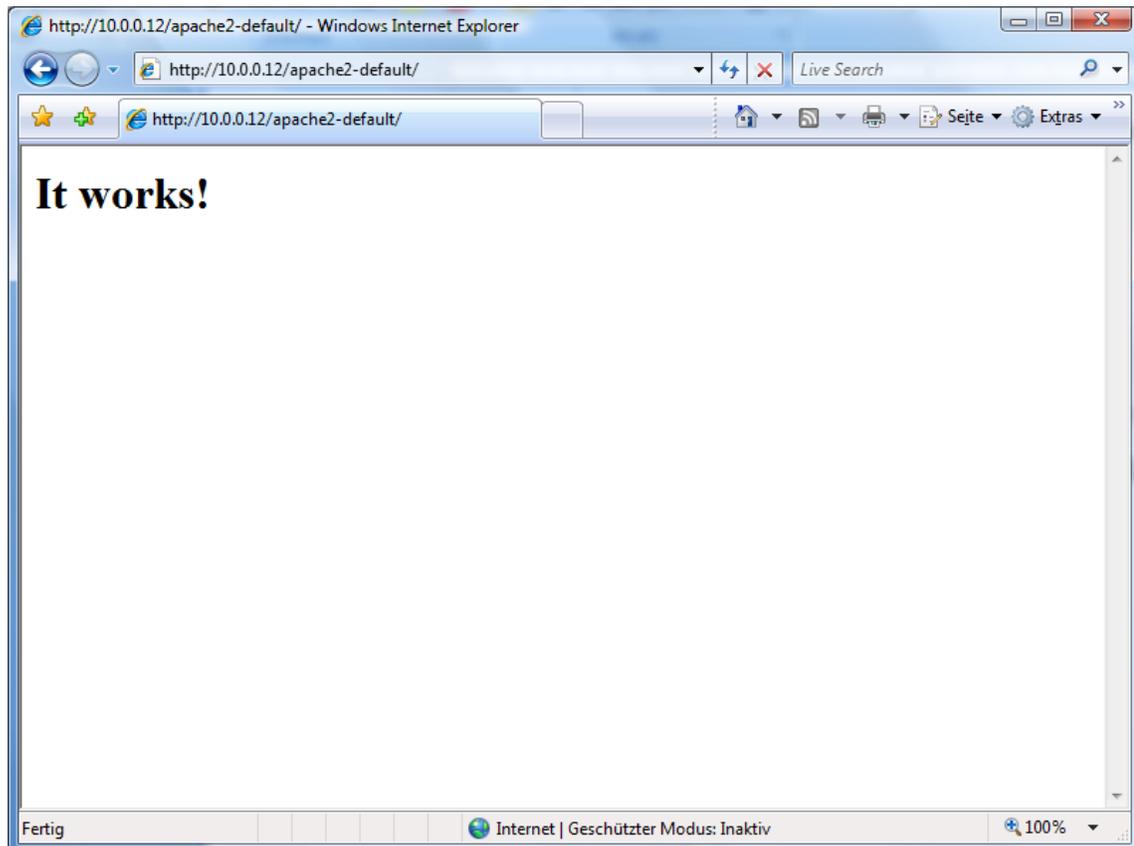


Bild 10: Default Webseite des Apache - Webservers

Damit ist die Installation des Apache –Webservers bereits abgeschlossen.

## 4.4 Tomcat – Webserver

Zunächst müssen wieder die Installationsdateien aus dem Internet heruntergeladen werden. Diese Dateien sind unter

*<http://tomcat.apache.org/download-55.cgi>*

zu finden und können abgerufen werden.

```
# wget http://ftp.hosting-studio.de/pub/linux/apache/tomcat/tomcat-5/v5.5.23/bin/apache-tomcat-5.5.23.tar.gz
```

Die gerade heruntergeladene Datei wird nun entpackt und installiert.

```
# tar -xzf apache-tomcat-5.5.23.tar.gz
# mv apache-tomcat-5.5.23 /opt
# chmod -R 775 /opt/apache-tomcat-5.5.23
# ln -s /opt/apache-tomcat-5.5.23 /opt/tomcat
```

Mit dem letzten Befehl wird ein Link auf diese Version angelegt, um später über `/opt/tomcat` die aktuelle Version im Zugriff zu haben, wie schon am Beispiel der Installation von Java erläutert.

Im Hinblick auf die Sicherheit sollten Programme wie Tomcat niemals mit Administratorrechten ausgeführt werden. Sinnvoller ist es hier, für den Tomcat einen eigenen Benutzer sowie eine eigene Gruppe anzulegen, mit dem der Webserver später ausgeführt wird. Darüber hinaus soll der Benutzer auch Zugriff auf die Verzeichnisse vom Apache-Webserver haben. Dazu werden die unten stehenden Befehle ausgeführt.

```
# groupadd -g 65 tomcat
# useradd -u 65 -g 65 -d /opt/tomcat -s /bin/false -c "Tomcat User" tomcat
# chown -R tomcat:tomcat /opt/tomcat
# usermod -G www-data tomcat
```

Um den Tomcat - Webserver nicht jedes Mal wieder manuell starten zu müssen, wäre es vorteilhaft, dass dieser Dienst beim Systemstart automatisch gestartet wird. Zum einen kann das Starten vergessen werden und zum anderen wird nach einem automatischen Reboot der Tomcat solange nicht gestartet, bis der Anwender sich explizit auf dem System einloggt und das Starten vornimmt.

Das bedeutet, dass auch die Anwendung, die auf diesem System installiert ist, im gleichen Zeitraum nicht genutzt werden kann. Dies sollte möglichst vermieden und alles Mögliche

getan werden, die Ausfallzeit des Servers so gering wie möglich zu halten. Aus diesem Grund wird wie beim Apache – Webserver ein Dämon eingerichtet, der über den Internet Superdämon initd beim Systemstart ausgeführt wird.

Dieser Dienst ist in dem bereits heruntergeladenen und installierten Teil von Tomcat enthalten. Der Anwender wechselt dazu in das Binary-Verzeichnis des Tomcat-Verzeichnisses

```
# cd /opt/tomcat/bin
```

und die Datei jsvc wird entpackt.

```
# tar -xzf jsvc.tar.gz
```

Um in das Verzeichnis jsvc-src schreiben und die Konfiguration ausführen zu können, muss die Zugriffsberechtigung für dieses Verzeichnis geändert werden. Anschließend wird in das Verzeichnis gewechselt

```
# chmod -R 755 jsvc-src  
# cd jsvc-src
```

und dort das portierbare Shellskript configure ausgeführt.

```
# ./configure
```

Dieses Skript läuft völlig automatisch ab und erfordert vom Anwender keinerlei Eingaben. Aus vorgefertigten Templates, die vom Entwickler bereitgestellt wurden, erzeugt das Skript an die Systemumgebung angepasste Header- und Makefiles. Diese Makefiles werden dann von

```
# make
```

verarbeitet und dabei wird eine Quelldatei in eine Objektdatei umgewandelt. Das bedeutet zum Beispiel, dass eine C-Datei in eine ausführbare Datei übersetzt wird. In vorliegendem Fall wird der Dämon jsvc erzeugt und dann in das Binary-Verzeichnis kopiert.

```
# cp jsvc /opt/tomcat/bin/
```

Um den Tomcat später als Dämon starten zu können, wird noch ein Startskript für den Internet Superdämon benötigt. Eine Vorlage dafür befindet sich im Unterverzeichnis `native`. Diese Vorlage wird in das Verzeichnis des Internet Superdämons kopiert.

```
# cd native
# cp Tomcat5.sh /etc/init.d/tomcat
```

Damit das Skript lauffähig ist, muss es an die jeweiligen Gegebenheiten angepasst werden und wissen, in welchen Verzeichnissen Java und der Tomcat – Webserver zu finden sind. Das Skript ist mit einem Editor aufzurufen und folgende Einträge sind zu ändern.

```
# nano /etc/init.d/tomcat
```

Das komplette Skript befindet sich im Anhang.

```
JAVA_HOME=/usr/lib/java
CATALINA_HOME=/opt/tomcat
DAEMON_HOME=/opt/tomcat
TOMCAT_USER=tomcat
${DAEMON_HOME}/bin/jsvc \
```

Nach Durchführung der Änderungen muss noch die Berechtigung für das Skript angepasst werden, damit der Internet Superdämon auch das Skript ausführen kann.

```
# chmod 755 /etc/init.d/tomcat
```

Systemdienste arbeiten mit so genannten Runleveln. Das sind mehrere abgestufte Systemzustände, die beim Booten durchlaufen werden. Bei Initialisierung eines Runlevels (Wechsel zu einem höheren Runlevel) werden die benannten Dienste automatisch gestartet. Umgekehrt werden bei Beendigung eines Runlevels die Prozesse automatisch gestoppt und Systemressourcen (Hauptspeicher, Ports etc.) freigegeben.

Mögliche Level finden sich in folgender Tabelle, können sich aber von Distribution zu Distribution unterscheiden.

Runlevel	Beschreibung
0	Shutdown. Alle Netzverbindungen werden geschlossen, Dateipuffer werden geschrieben, Partitionen umounted..
S	Single-User-Runlevel; niedrigster Systemzustand für Wartungsarbeiten, in dem ausschließlich Systemressourcen wie Festplatten oder Dateisysteme aktiv sind.
1	Einzelnutzerbetrieb ohne Netzwerk mit ausschließlich lokalen Ressourcen. In vielen Implementationen identisch mit 'S'.
2	Lokaler Mehrnutzerbetrieb ohne Netzwerk mit ausschließlich lokalen Ressourcen. Unter einigen Linuxdistributionen (z. B. Debian) wird in Runlevel 2 auch das Netzwerk konfiguriert.
3	Netzwerkbetrieb, über das Netzwerk erreichbare Ressourcen sind nutzbar, eine grafische Oberfläche steht nicht zur Verfügung. Firewalls sollten aktiviert werden.
4	Ist normalerweise nicht definiert. Kann aber für diverse Dienste genutzt werden.
5	wie 3, zusätzlich wird die grafische Oberfläche bereitgestellt.
6	Reboot. Alle Netzverbindungen werden geschlossen, Dateipuffer werden geschrieben, die Mounts auf die Partitionen werden entfernt.

**Tabelle 1: Runlevel unter Linux [11]**

Um die Runlevel für den Tomcat festzulegen, muss nachstehendes Kommando ausgeführt werden.

```
# update-rc.d tomcat start 91 2 3 4 5 . stop 20 0 1 6 .
```

Somit ist die Installation abgeschlossen und alle erforderlichen Einstellungen wurden vorgenommen. Abschließend kann der Tomcat – Webserver jetzt gestartet werden.

```
# /etc/init.d/tomcat start
```

Es ist sicher anwenderfreundlicher, dafür nur einen einzelnen Befehl zu haben. Aus diesem Grund wird für obiges Kommando ein Alias angelegt.

```
# alias tomcat=/etc/init.d/tomcat
```

War die Installation erfolgreich, ist nachfolgender Inhalt im Browser zu sehen.

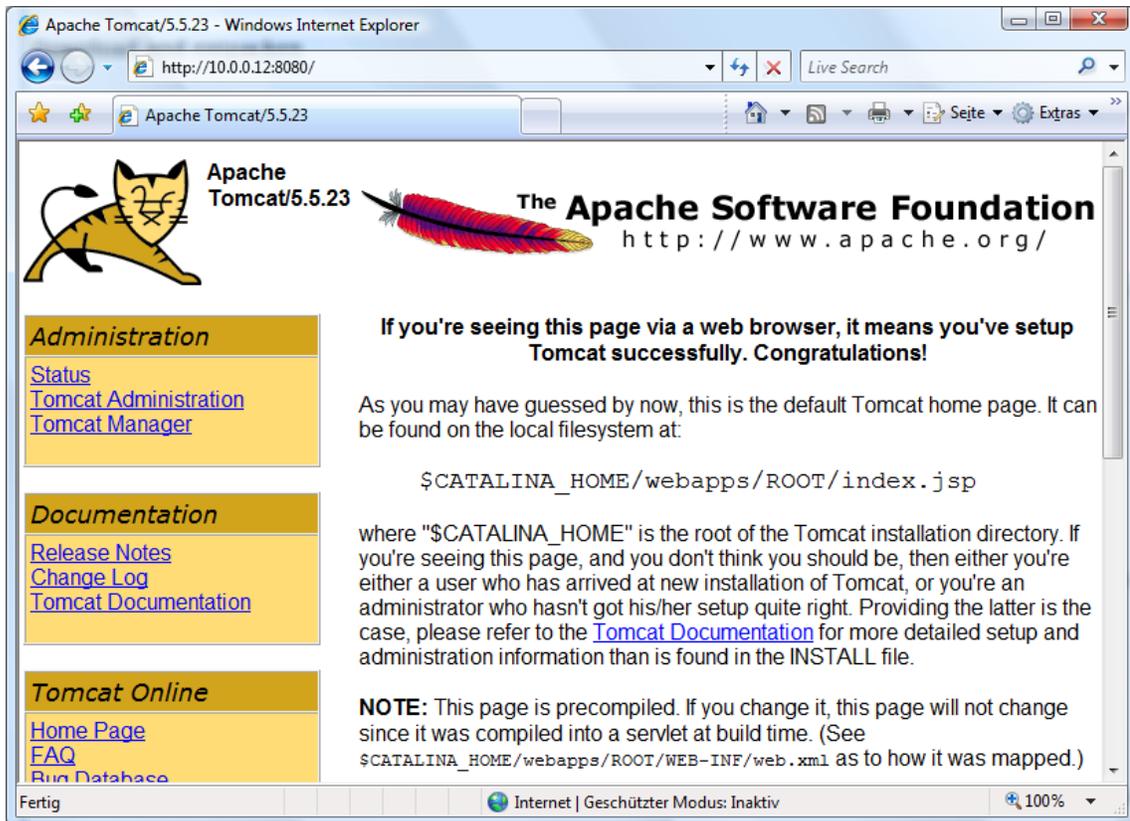


Bild 11: Default Webseite des Tomcat - Webservers

## 4.5 Datenbanksystem Oracle 10g

Die Installation des Datenbanksystems gestaltet sich etwas schwieriger, da die Installation nicht textbasiert über die Konsole erfolgt, sondern Oracle 10g einen grafischen Installer mitliefert. Um diesen darstellen zu können, muss ein X Window System vorhanden sein.

Das X Window System ist ein System zum Anzeigen einer grafischen Benutzeroberfläche unter Unix- und Linux-Systemen.

Es handelt sich um eine Sammlung von Protokollen, Computerprogrammen und Standards zur Ansteuerung grafischer Bildschirme im Allgemeinen und wird vorwiegend von Linux- und Unix-Systemen, deren eigene Benutzerschnittstellen nicht grafisch sind, verwendet. Das X Window System arbeitet auf Basis der Client-Server-Technik. [12]

Das Ziel war, auf dem Server selbst keine grafische Oberfläche wegen der Performance zur Verfügung zu stellen, und deshalb wurde ein anderer Weg beschritten. Der erste Ansatz war dabei, die Ausgabe auf einen anderen Rechner mit installiertem X Window System umzuleiten.

Bevor mit der eigentlichen Installation begonnen werden kann, müssen noch einige Vorbereitungen getroffen werden. Dazu muss zunächst das Datenbanksystem mit dem Installer von der Oracle Webseite heruntergeladen werden. Um Software von der Webseite bekommen zu können, muss sich auf dieser angemeldet werden. Nach erfolgreicher Anmeldung wird das Oracle Paket auf einen anderen Computer, auf dem sich ein Webbrowser befindet, heruntergeladen und danach auf den Server in ein temporäres Verzeichnis kopiert. Anschließend wird das Archiv mit

```
# unzip 10201_database_linux32.zip
```

entpackt, so dass sich dort ein Verzeichnis database befindet. Eine andere Möglichkeit wäre, das Archiv direkt auf dem Rechner zu entpacken, auf dem die Datei heruntergeladen worden ist, und die Daten direkt auf eine CD zu brennen. Das hat den Vorteil, dass bei einem Ausfall des Servers die Datei nicht erneut heruntergeladen werden müsste.

Außerdem muss sichergestellt sein, dass die folgenden Debian-Pakete auf dem Rechner installiert sind.

- gcc
- make
- binutils
- libmotif3
- lesstif2
- rpm
- libaio
- unzip

Dazu wird im Prompt der Befehl `apt-get` ausgeführt.

```
# apt-get install gcc make binutils libmotif3 lesstif2 rpm libaio unzip
```

Der Oracle Installer geht davon aus, dass die Datenbank auf einem Red-Hat-System installiert werden soll. Da es sich bei dieser Installation aber um ein Debian-System handelt, müssen einige Verzeichnisse angepasst werden, nach denen der Oracle Installer sucht. Dazu werden für die unten stehenden Verzeichnisse Links angelegt.

```
# ln -s /usr/bin/awk /bin/awk
# ln -s /usr/bin/rpm /bin/rpm
# ln -s /usr/bin/basename /bin/basename
# ln -s /etc /etc/rc.d
```

In dem oben entpackten Verzeichnis befindet sich im Unterverzeichnis Disk1/install die Datei oraparam.ini. In dieser Datei muss noch eine Zeile verändert werden. Dazu wird mit

```
# nano /tmp/database/Disk1/install/oraparam.ini
```

die Datei in einem Editor geöffnet und die Zeile

```
Linux=redhat-2.1,UnitedLinux-1.0,redhat-3
```

in

```
Linux=
```

geändert.

Für den Installer wird außerdem noch ein eigener Benutzer bzw. eigene Gruppe benötigt. Da dieses System gerade neu installiert worden ist, sind diese noch nicht vorhanden. Bei einem System, auf dem schon einmal Oracle installiert worden ist, muss das Vorhandensein dieses Benutzers / dieser Gruppe vorher abgeklärt werden.

```
# groupadd oinstall
# groupadd dba
# groupadd nobody
# useradd -g oinstall -G dba -p passwd -d /home/oracle oracle
# useradd -g nobody nobody
```

Für die Installation werden noch einige Verzeichnisse benötigt. Außerdem müssen die Oracle-Benutzer Rechte haben, in diesen Verzeichnissen schreiben zu dürfen. Des Weiteren wird der Besitzer dieser Verzeichnisse von root in oracle geändert.

```
# mkdir -p /home/oracle
# mkdir -p /opt/oracle/product/oracle
# mkdir -p /opt/oracle/product/oradata
# chown -R oracle:oinstall /opt/oracle
# chmod -R 775 /opt/oracle
# chown -R oracle:oinstall /home/oracle
# chmod -R 775 /home/oracle
```

Im nächsten Schritt müssen einige Kernel-Parameter überprüft werden, ob diese so eingestellt sind, wie Sie vom Installer benötigt werden. Dazu werden die nachstehenden Befehle (beginnend mit #) der Reihe nach ausgeführt. Darunter befindet sich jeweils die zu erwartende Ausgabe:

```
# /sbin/sysctl -a | grep sem

kernel.sem = 250 32000 100 128

# /sbin/sysctl -a | grep shm

kernel.shmmin = 4096
kernel.shmall = 2097152

# /sbin/sysctl -a | grep file-max

kernel.shmmax = 2147483648
fs.file-max = 65536

# /sbin/sysctl -a | grep ip_local_port_range

net.ipv4.ip_local_port_range = 1024 65000
```

Ist dies nicht der Fall, muss die Datei /etc/sysctl.conf geändert werden.

```
# nano /etc/sysctl.conf
```

Obiger Befehl öffnet die Datei in einen Editor. Alle nicht übereinstimmenden Werte werden entsprechend den unten stehenden Vorgaben geändert.

```
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
```

Nachdem alle Parameter geändert wurden, muss diese Konfigurationsdatei neu eingelesen werden.

```
# /sbin/sysctl -p
```

Um noch einige Sicherheitseinstellungen vorzunehmen, wird die Datei `limits.conf` angepasst und unten stehende Zeilen eingefügt, um `nproc` und `nofile` zu begrenzen.

```
# nano /etc/security/limits.conf

*soft nproc 2047
*hard nproc 16384
*soft nofile 1024
*hard nofile 65536
```

In die Dateien `/etc/pam.d/login` und `/etc/pam.d/su` wird die `session`-Zeile jeweils eingefügt.

```
# nano /etc/pam.d/login

session required /lib/security/pam_limits.so

# nano /etc/pam.d/su

session required /lib/security/pam_limits.so
```

Abschließend wird die Datei `/etc/profile` erweitert, damit jeder Benutzer die unten stehenden Beschränkungen zugewiesen bekommt.

```
# nano /etc/profile
```

In der profile-Datei wird folgendes dazugetragen:

```
if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16384
        ulimit -n 65536
    else
        ulimit -u 16384 -n 65536
    fi
fi
```

Das Vorgehen, die Ausgabe auf einen Rechner mit installiertem X Window System umzuleiten, hat aber bei der Installation nicht funktioniert. Aus diesem Grund wurde ein anderer Weg gesucht und der Server mit einer Linux Live-CD gebootet. Im Internet sind mehrere dieser Live-CDs vorhanden, so dass die Entscheidung für die Live-CD von Knoppix fiel. Knoppix anzuwenden, hatte die Gründe, dass Knoppix über eine sehr gute Hardwareerkennung verfügt und eine Vielzahl von Treibern mitliefert.

Dadurch, dass der Server mit dieser Live-CD gebootet wird, steht jetzt eine grafische Oberfläche bereit, auf die die Ausgabe des Oracle-Installers umgeleitet werden kann. Dazu wird der mit Tastenkombination

```
#Strg-Alt-F1
```

auf eine Konsole gewechselt, um Vorbereitungen zu treffen, damit der Installer läuft. Anschließend werden nachfolgende Befehle eingegeben.

```
# su
# X:1 &
# export DISPLAY=localhost:1.0
```

Mit su wechselt der Benutzer zum Administrator. Nachdem dies geschehen ist, muss für den Installer ein neues X-Server-Fenster geöffnet werden. Dies wird mit dem zweiten Befehl erreicht. Nach Durchführung dieses Befehls ist im Hintergrund ein neues Fenster verfügbar.

Mit der `export`-Anweisung wird die X-Ausgabe schließlich in das neue Fenster geleitet. Anschließend wird mit

```
# xhost +
```

anderen Rechnern erlaubt, auf diesen X-Server zuzugreifen.

Da aber der Server mit der Live-CD gebootet wurde, ist das Dateisystem des Servers im Moment nicht im direkten Zugriff. Um auf die Verzeichnisse des Servers zugreifen zu können, müssen die Festplatten gemountet (ins System eingebunden) werden. Dazu wird auf der Konsole der Befehl

```
# mount /dev/sda1 /mnt
```

einggegeben. Mit Eingabe des Befehls

```
# chroot /mnt
```

wird die eben gemountete Partition zum Stammverzeichnis.

Erst jetzt sind alle auf dem Server angelegten Benutzer und Gruppen bekannt. Der Oracle-Installer läuft nicht mit root-Rechten. Wäre das möglich, würde es zudem ein Sicherheitsrisiko darstellen. In diesem Fall würden alle Oracle-Dienste auch mit root-Rechten laufen. Aus diesem Grund ist anfangs der Benutzer `oracle` angelegt worden. Mit

```
# su oracle
```

wird der Nutzer zum Benutzer `oracle`. Anschließend wird die eigentliche Installation mit

```
# ./runInstaller
```

oder mit

```
# /opt/oracle/runInstaller
```

vorgenommen.

Wenn beim Aufruf des Installers die Fehlermeldung

```
...i386/libawt.so: libXp.so.6: cannot open shared object file: No such file
```

ausgegeben wird, benötigt der Installer noch einige weitere X Komponenten. Diese werden mit

```
# apt-get install konsole
```

nachinstalliert. Anschließend sollte der Installer ohne weitere Probleme ausgeführt werden können.

Um den Installer sehen zu können, muss mit der Tastenkombination

```
# Strg-Alt-F6
```

zu dem Fenster des X-Servers gewechselt werden. Dort erscheint jetzt folgendes Bild:

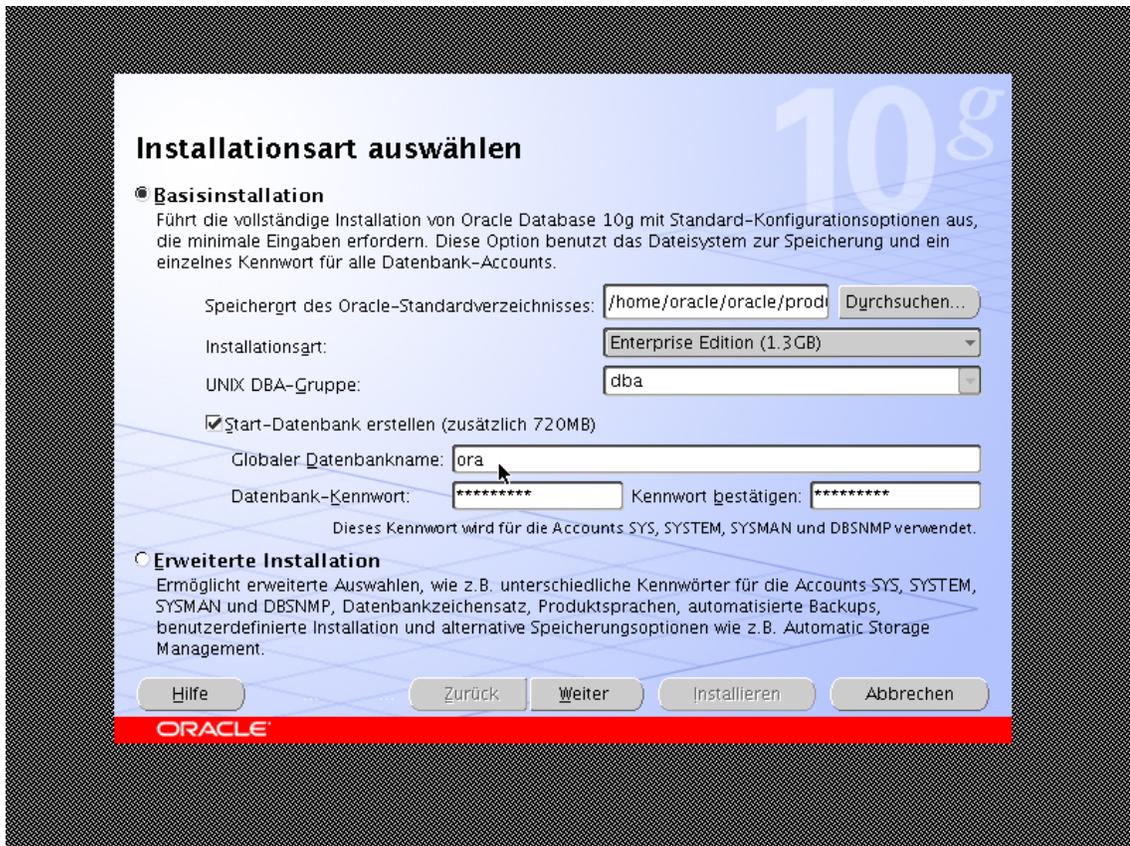


Bild 12: Oracle Installer - Installation Oracle 10g

Auf dem Bildschirm wird der Speicherort der Datenbank angegeben, was in diesem Fall

*/opt/oracle/oracle/product*

ist. Als Globaler Datenbankname wird ora festgelegt und ein Passwort für die Datenbank ausgewählt. Die in den nachfolgenden Bildschirmen erscheinenden Eingaben, werden so übernommen wie sie vorgeschlagen sind. Wenn alle Angaben gemacht wurden, erscheint eine Übersicht, in der nochmal die gemachten Einstellungen zusammengefasst werden. Mit einem Klick auf „Weiter“ wird die eigentliche Installation gestartet.

Ist die Installation der Software abgeschlossen, wird wie gewünscht eine Beispieldatenbank angelegt. Abschließend werden die Systemdienste von Oracle für die Datenbank konfiguriert.

Eigentlich ist die Installation beendet, aber unter bestimmten Bedingungen, die nicht ermittelt werden konnten, kann es nötig sein, die im Hinweisschirm aufgeführten Konfigurationsskripts auszuführen. Dazu wird mit

```
# Strg-Alt-F3
```

eine weitere Konsole aufgerufen. Zum Ausführen dieser Konfigurationsskripte ist es notwendig, sich als root anzumelden. In dem Oracle-Verzeichnis wird dann nacheinander

```
# orainstRoot.sh
```

```
# root.sh
```

ausgeführt. Nach Durchlaufen dieser Skripte wird wieder das X-Fenster des Installers geöffnet und die Installation beendet.

Damit ist die Oracle Datenbank installiert und mit der Eingabe von

```
# localhost:1158/em
```

erscheint der Enterprise Manager von Oracle mit der Datenbank-Instanz „Ora1“.

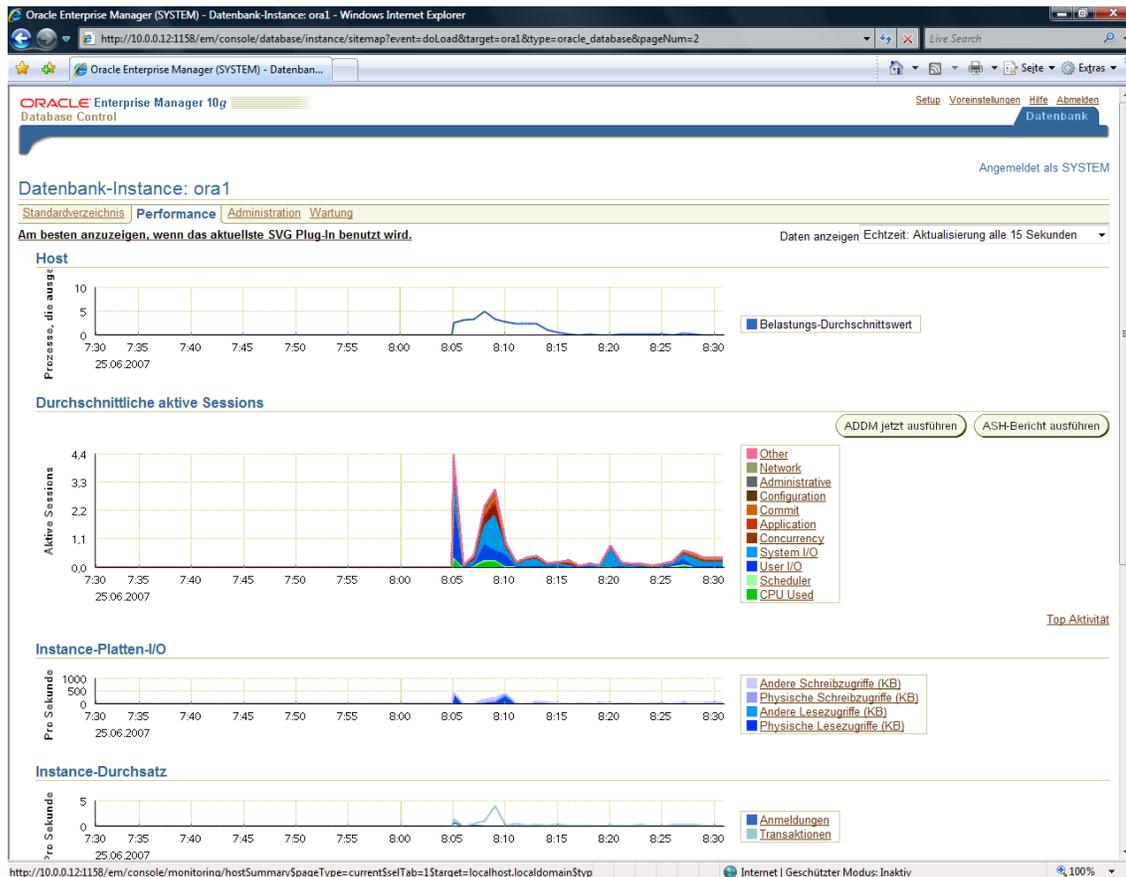


Bild 13: Oracle Enterprise Manager - Performance der Datenbank ora1

## 4.6 JEVIS-Portal

Das JEVIS-Portal wird per Installationsskript installiert. Dieses Skript ist von der Firma Envidatec entwickelt und freundlicher Weise zur Verfügung gestellt worden. Dazu wird in der Konsole

```
# sh install.sh -type server
```

der Aufruf zum Start der Installation eingegeben. Von dem Skript werden bei Ablauf einige Voraussetzungen geprüft, ob Java, Apache, Tomcat und Oracle vorhanden sind. Ist das der Fall, läuft Installation völlig automatisch ab und legt in der Datenbank alle erforderlichen Tabellen für das JEVIS-Portal an.

## 4.7 Skriptbasierte Installation

Nachdem in den vorherigen Kapiteln 4.3 bis 4.6 die Installationen manuell durchgeführt und die einzelnen Befehle per Hand eingegeben wurden, soll hier gezeigt werden, wie die Installationen der einzelnen Programme auch skriptbasiert erledigt werden können. Dazu werden aus den oben gezeigten Installationen diese Skripte entwickelt.

Für die Installation des Java Development Kits (JDK) bedeutet dieses, dass erst die Zeile mit der Paketliste aus dem „Etch“-Zweig in die Konfigurationsdatei des apt-Paketmanagers eingetragen wird. Dazu lautet die Zeile des Shell-Skripts

```
# echo="deb ftp://ftp.de.debian.org/debian/ etch main non free contrib" >> /etc/apt/sources.conf
```

Das Skript besteht im Weiteren aus den Befehlen, die bei der manuellen Installation von Java verwendet wurden. Das komplette Skript zur Installation von Java (JDK) findet sich im Anhang. Für eine Installation muss dann lediglich der Befehl

```
# java-inst.sh
```

ausgeführt werden. Für die Durchführung der weiteren Installationen ist es notwendig, sich wieder mit

```
# su
```

als root anzumelden.

Da aber während der Installation noch die `/etc/profile`-Datei angepasst werden muss, wird dabei so vorgegangen, dass diese durch eine bereitgestellte Datei ersetzt wird. Dabei ist aber Vorsicht geboten, denn das funktioniert nur solange, bis keine anderen Programme vorher installiert worden sind, die Einträge in diese Datei vorgenommen haben. Mit dem Kopieren der bereitgestellten `profile`-Datei wird die vorhandene ersetzt und vorher gemachte Einträge gehen verloren. Das hat den Effekt, dass diese Programme nicht mehr funktionieren.

Für die Installation des Apache – Webservers macht es weniger Sinn, ein Installationskript zu schreiben, da nur ein Befehl ausgeführt werden muss. Um die Vorgehensweise zu vereinheitlichen, besteht das Skript aus folgender Zeile.

```
apt-get install apache2
```

Danach wird auf der Konsole für den Apache- Webserver

```
# apache-inst.sh
```

aufgerufen.

Für den Tomcat – Webserver müssen von dem Installationskript wieder mehrere Befehle abgearbeitet werden, damit der Webserver komplett eingerichtet wird. Dazu werden alle Befehle, die vorher manuell eingegeben wurden, in das Skript übertragen.

Ausnahme hierbei sind die Befehle, die aus den Source-Dateien des Tomcat ein Startskript generieren. Das am Ende benötigte Startskript für den Tomcat wird nicht aus den Source-Dateien erstellt, sondern hierfür wird ein vorgefertigtes Startskript an die entsprechende Stelle kopiert.

Mit dem Befehl

```
# tomcat-inst.sh
```

wird die komplette Installation des Tomcat – Webservers durchgeführt.

Da zur Installation der Oracle-Datenbank eine grafische Oberfläche verwendet wird, lässt sich nicht die komplette Einrichtung skriptbasiert steuern. Die gesamten Vorbereitungen lassen sich aber per Skript durchführen. Nach Aufruf des Skripts mit

```
# oracle-req-inst.sh
```

wird mit dem Umleiten X Ausgabe, wie in Kapitel 4.5 beschrieben, fortgefahren. Ist die Ausgabe umgeleitet worden, wird der Oracle-Installer gestartet.

# 5 Inbetriebnahme des Servers

Wie im vorherigen Kapitel beschrieben, wurde die benötigte Software installiert und ist jetzt einsatzfähig. Aufgabe ist es nun, die Software entsprechend ihrer zugeordneten Aufgaben zu konfigurieren. Dabei werden die Unterschiede einer Konfiguration untersucht, für welchen Zweck der Server neu installiert bzw. wiederhergestellt wurde. So wird gezeigt, wie eine Datensicherung zurückgespielt wird. Des Weiteren wird in diesem Kapitel beschrieben, warum ein neu installierter Server zuerst im Parallelbetrieb laufen sollte.

## 5.1 Einspielen einer Datensicherung

Der für das Projekt installierte Server soll den bisherigen Server der Firma Envidatec ablösen, so dass es notwendig ist, eine Datensicherung zurückzusichern. Da im Allgemeinen neben der Datenbank auch das System gesichert wird, kann im Normalfall die Installation wegfallen, und es wird nur eine Datensicherung von System und Datenbank zurückgespielt.

Da es sich aber bei dem neuen Server, um eine komplett andere Hardware handelte, ist sicherheitshalber der Weg einer Neuinstallation gegangen worden.

Für den Fall, dass der gesamte Server ausfällt, also alle Festplatten nicht weiter benutzt werden können, müssen diese zuerst gegen neue getauscht werden. Sind die Festplatten getauscht, wird der Server mit einer bootfähigen CD neu gestartet. Diese CD muss einen Linux-Kernel enthalten, der die Komponenten SCSI-Bandlaufwerk und den SCSI-RAID-Controller unterstützt. Nachdem die Festplatten dann partitioniert worden sind, wird das gesamte Backup in die jeweiligen Verzeichnisse zurückgesichert. Danach muss das System wieder bootfähig gemacht werden, in dem die Systemfestplatte im GRUB-Bootloader als „bootable“ gesetzt wird.

Um das gesamte System wieder herzustellen, wird in ein temporäres Verzeichnis gewechselt und führt folgenden Befehl aus.

```
# cd /tmp
# tar -x -v -M -f /dev/tapeA -f /dev/tapeB
```

Wenn bei dem tar-Befehl keine Datei angegeben wird, dann wird das gesamte Archiv zurückgesichert.

Um bei einer Rücksicherung nicht ungewollt irgendetwas zu überschreiben, so wird empfohlen, die Sicherung in ein temporäres Verzeichnis wie /tmp oder /var/tmp schreiben zu lassen, das genügend Speicherplatz für die Sicherung bietet. Aus diesem Verzeichnis werden die Dateien dann manuell an ihren Bestimmungsort kopiert.

Da nach einem Totalausfall auf den neuen Festplatten noch keine Verzeichnisse vorhanden sind, wird mit

```
# mkdir /tmp
# chmod 777 /tmp
```

ein temporäres Verzeichnis erstellt und mit dem chmod-Befehl für alle Benutzer auf dem System zugänglich gemacht.

```
V----- 0/0          0 2004-11-09 03:03:01 L0 daily backup of SERVER:/daten -- 09.11.2004 03:03--
Volume Header--
drwxr-xr-x 0/0          0 2004-09-28 10:07:42 ./
drwx----- 0/0          0 2003-08-04 18:45:09 ./lost+found/
drwxrwsr-x 0/100       0 2004-11-05 22:38:12 ./env_share/
drwxrwxr-x 504/100     0 2004-01-05 17:09:30 ./env_share/admin/
drwxr-sr-x 0/100       0 2004-09-27 18:56:47 ./env_share/admin/pgp/
-rw-r--r-- 0/0          59735 2004-09-27 18:56:57 ./env_share/admin/pgp/pubring.gpg
-rw-r--r-- 0/0          63386 2004-05-21 17:48:53 ./env_share/admin/pgp/pubring.gpg.pre-20040521
-rw-r--r-- 0/0          55685 2004-06-02 14:41:39 ./env_share/admin/pgp/pubring.gpg.pre-20040602
-rw-r--r-- 0/0          57740 2004-08-19 13:17:39 ./env_share/admin/pgp/pubring.gpg.pre-20040819
-rw-r--r-- 0/0          57485 2004-09-21 16:30:33 ./env_share/admin/pgp/pubring.gpg.pre-20040921
-rw-r--r-- 0/0          58745 2004-09-27 18:56:47 ./env_share/admin/pgp/pubring.gpg.pre-20040927
-rw----- 501/100      28 2003-12-04 16:08:47 ./env_share/admin/pgp/.directory
-rw-r----- 0/0          431 2004-01-05 18:33:43 ./env_share/admin/pgp/import-key-HOW-TO.txt
drwxrws--- 504/100     0 2004-01-05 17:09:30 ./env_share/admin/odauter/
-rw-rw---- 504/100    264018 2002-06-05 20:05:15 ./env_share/admin/odauter/orginal-cups-admin.pdf
-rw-rw---- 504/100    97874 2002-06-05 20:05:15 ./env_share/admin/odauter/orginal-cups-user.pdf
< ... >
```

Beispiel: Zurückgesicherte Dateien mit ihren Verzeichnissen und Berechtigungen [13]

Wie das obige Beispiel zeigt, hat jedes gesicherte Archiv einen Bezeichner, der den Namen des gesicherten Verzeichnisses, den Typ und das Datum enthält. Im obigen Beispiel ist es das Archiv für das Verzeichnis /daten.

Wenn die Rücksicherung abgeschlossen ist, können die Dateien aus obigem Verzeichnis unter

```
# cd /tmp/daten
```

gefunden werden und müssen mit dem Kopierbefehl

```
# cp -R /tmp/daten /daten
```

an seinen Ursprungsort zurückkopiert werden. [13]

## 5.2 Parallelbetrieb

Nach der Installation und Kopieren der Datenbank auf diesen Server ist dieser soweit lauffähig. Da die Firma Envidatec den installierten Server verwendet, um ihr tägliches Geschäft damit abzuwickeln, muss sichergestellt sein, dass der Server korrekt eingerichtet wurde.

Da man im Testbetrieb nicht alle Eventualitäten ausschließen kann, wird der Server im realen Betrieb eingesetzt. Um aber trotzdem die Sicherheit zu gewährleisten, dass bei einem Fehler, keine Informationen verloren gehen, wird der bisherige Server im Parallelbetrieb mitgeführt. Das bedeutet, dass die Daten auf beide Server geschrieben werden.

Läuft der neuere Server über einen Zeitraum von zwei Monaten stabil und treten im Betrieb keine Fehler auf, kann der ältere Server abgeschaltet werden. Da der Server aber nicht ausgedient hat, wird dieser umkonfiguriert. Dabei wird der Server so eingerichtet, dass er als Backup-Server dient. Er hat somit den gleichen Datenbestand und kann bei Ausfall des 1. Servers die Aufgaben übernehmen.

# 6 Analyse der Daten

In diesem Kapitel soll betrachtet werden, wie die Datenbank mit Werten gefüllt wird. Wenn diese Daten dann in der Datenbank stehen, können sie analysiert werden. Dazu existiert das Programm PControll, das im Rahmen einer Diplomarbeit entwickelt wurde. Außerdem steht mit dem JEVIS-Portal ein weiteres Programm zur Verfügung.

## 6.1 Auswertung und Visualisierung der Daten

Für eine Auswertung der Lastgänge müssen die Daten, die in den Loggern der einzelnen Liegenschaften gesammelt wurden, abgeholt werden. Dazu ruft der Server jeden einzelnen Datenlogger an und liest die Werte aus dem Speicher aus. Die Daten werden mit einer Datensatznummer, einem Zeitstempel und den Leistungswerten der 3 Liegenschaften in die Datenbank geschrieben.

### 6.1.1 PControll

Die Datenbank ist so eingestellt, dass sie momentan die Daten der letzten 10 Tage behält. Somit befinden sich maximal 15000 Werte in der Datenbank. Für eine Auswertung kann das Programm PControll genommen werden, das die Werte der letzten 24 Stunden aus der Datenbank ausliest und grafisch darstellt.

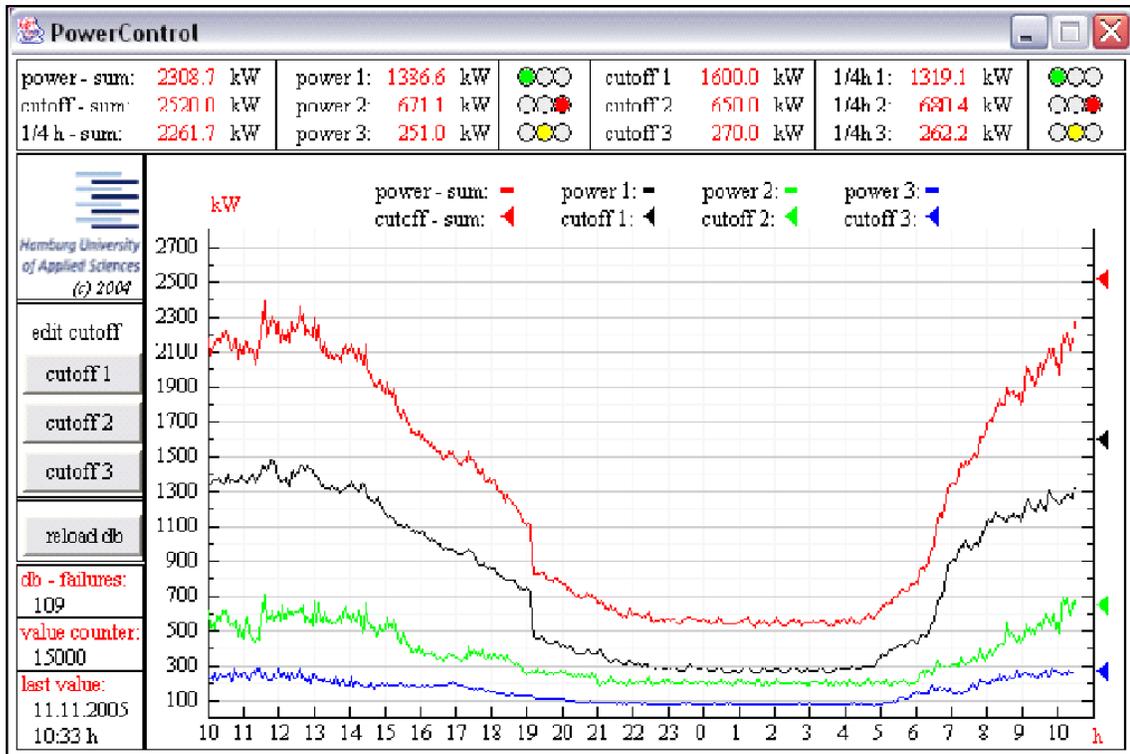


Bild 14: Screenshot des Programms "Power Control" [2]

## 6.1.2 JEVIS-Portal

Das JEVIS-Portal dient der Visualisierung und Bearbeitung von Betriebsdaten aller Art. JEVIS besteht aus einem Internet-Portal, redundanten Datenbanken mit einer Vielzahl von Auswertemodulen und verschiedenen Visualisierungstools.

Sämtliche Datenpunkte einer Liegenschaft, die sich für eine Erfassung und Übertragung eignen, lassen sich mit den verschiedenen JEVIS-Modulen darstellen. Die Daten werden verschlüsselt auf die JEVIS-Datenbank übertragen, dort mit entsprechenden Auswerteprogrammen bearbeitet und im Portal online für eine Analyse zur Verfügung gestellt.

Wenn beispielsweise der Stromverbrauch in Abhängigkeit der Laufzeiten von Lüftungs- oder Kälteanlagen oder die absoluten Energieverbräuche mehrerer Liegenschaften miteinander verglichen werden sollen, bietet JEVIS die dafür passenden Module. [14]

### 6.1.2.1 JEChart

Das Modul JEChart ist ein Universaltool, um die Betriebsdaten zu visualisieren und zu vergleichen. JEChart ermöglicht die grafische Darstellung der erfassten Daten. Die Anzahl der visualisierbaren Datenpunkte ist unbegrenzt. Die Datenreihen von 25 Liegenschaften können gleichzeitig in einer Grafik dargestellt werden. So lassen sich für die aus den Daten generierten Kurvenverläufe Zoom, Farbe und der sichtbare Ausschnitt festlegen. Ebenfalls können in dieser Grafik Darstellungen von Vergleichs- und Referenzkurven hinzugefügt werden. [14]

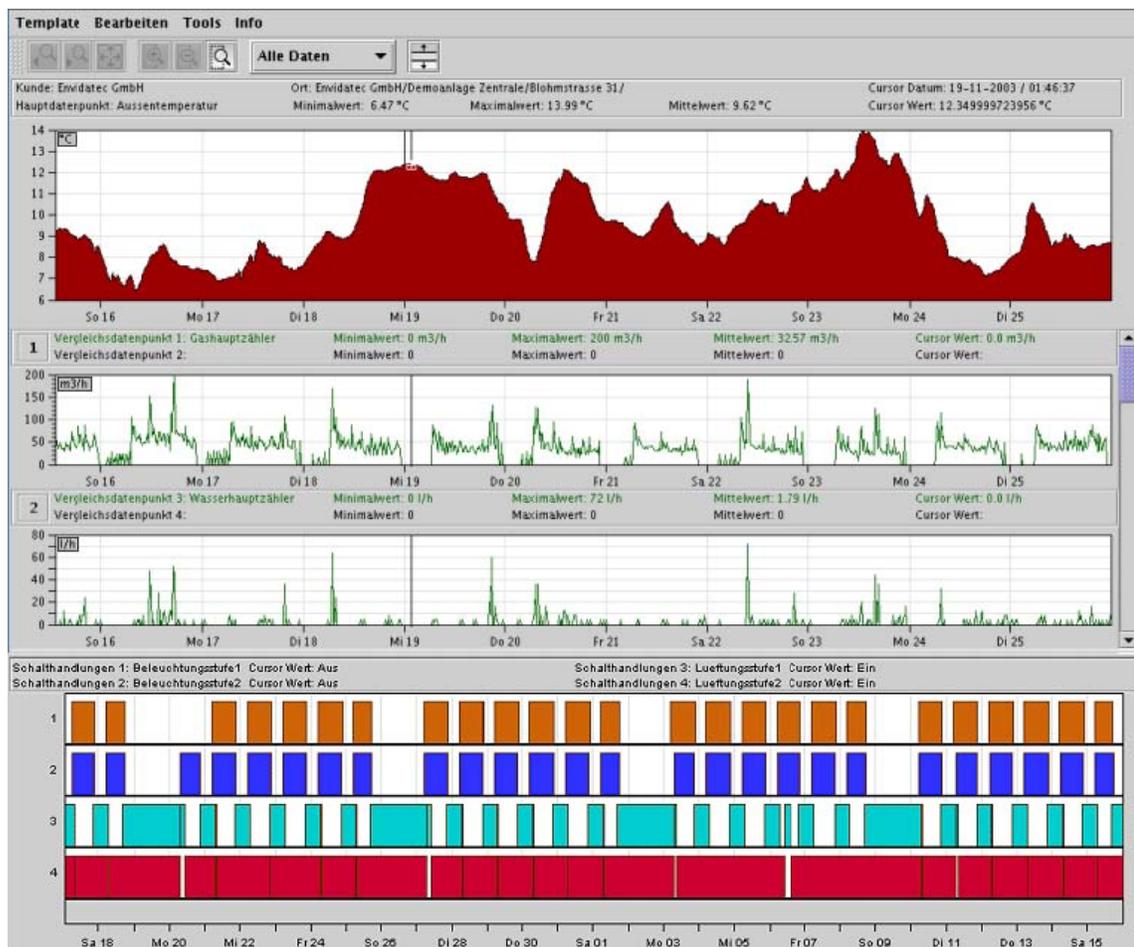


Bild 15: Darstellung von Lastprofilen in JEChart [15]

## 6.1.2.2 JEBench

JEBench ist ein Benchmarking-Tool zum Vergleichen von Verbräuchen, Kosten, Kennzahlen und anderen Größen. Damit lassen sich die Liegenschaften optimal vergleichen. Auf der Basis statischer und/oder dynamischer Betriebs- und Umgebungsdaten werden Kennzahlen für jede Liegenschaft ermittelt und in einer Liste dargestellt. Der Energiebedarf verschiedener Liegenschaften kann somit unter Berücksichtigung ortspezifischer Umgebungsdaten miteinander verglichen werden. Besonders energieintensive Liegenschaften werden sofort erkannt. [14]

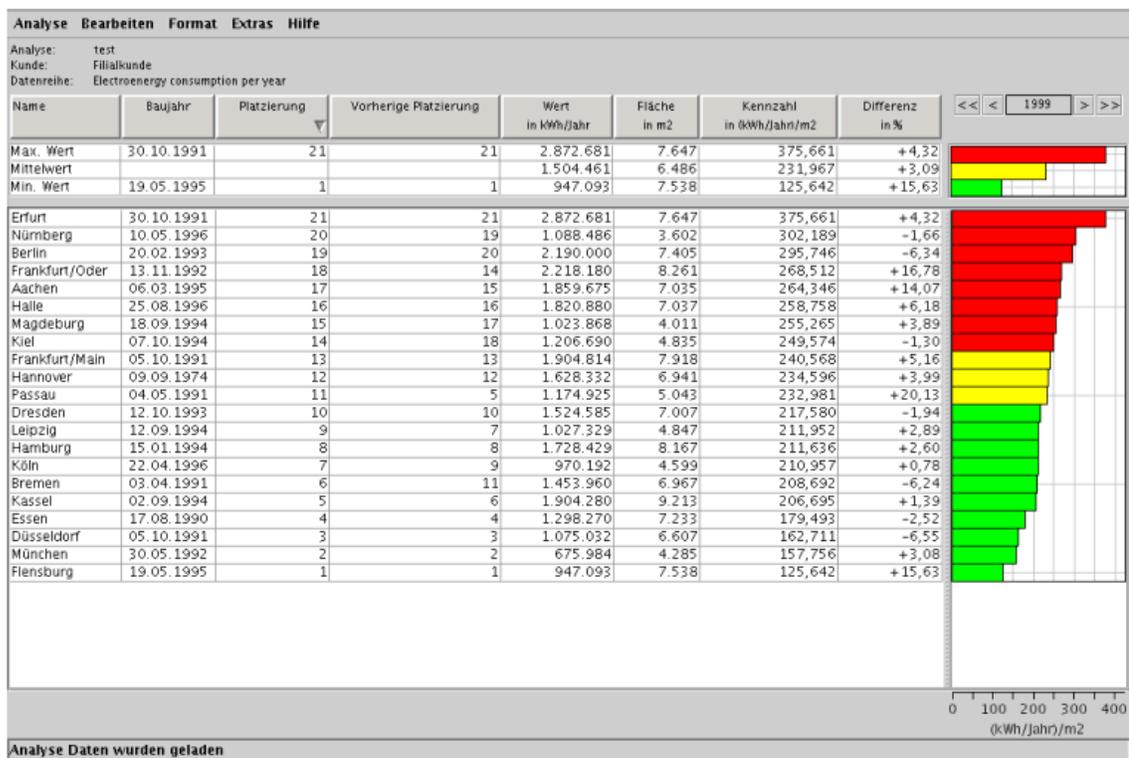


Bild 16: Ranking von Liegenschaften in JEBench [16]

## 6.2 Bestimmung des Lastprofils

Zur Bildung eines Lastprofils werden über 24 Stunden alle 15 Minuten der momentane Verbrauch ermittelt und in dem Datenlogger gespeichert. Regelmäßig einmal am Tag (alle

24 Stunden) werden diese Werte aus den Datenloggern zum Server übertragen und in die Datenbank geschrieben.

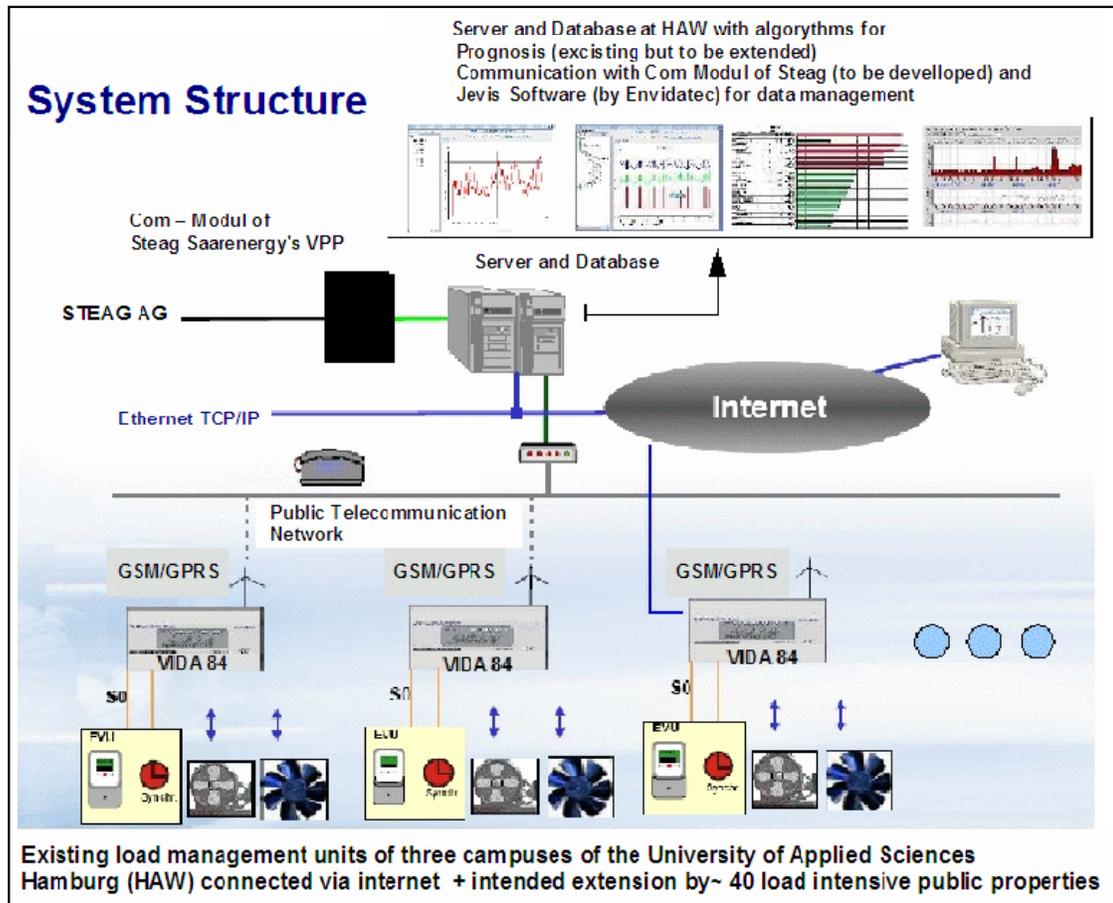


Bild 17: Infrastruktur zur Ermittlung der Lastdaten im Projekt „INSEL“ [2]

Danach werden die Daten der einzelnen Liegenschaften grafisch dargestellt. Um ein Gesamtprofil aller Liegenschaften zu bekommen, werden die jeweiligen Kurven kumuliert.

$$P_{ges} = \sum P_{Ls}$$

$P_{ges}$  : Gesamtleistung aller Liegenschaften

$P_{Ls}$  : Leistung eine Liegenschaft

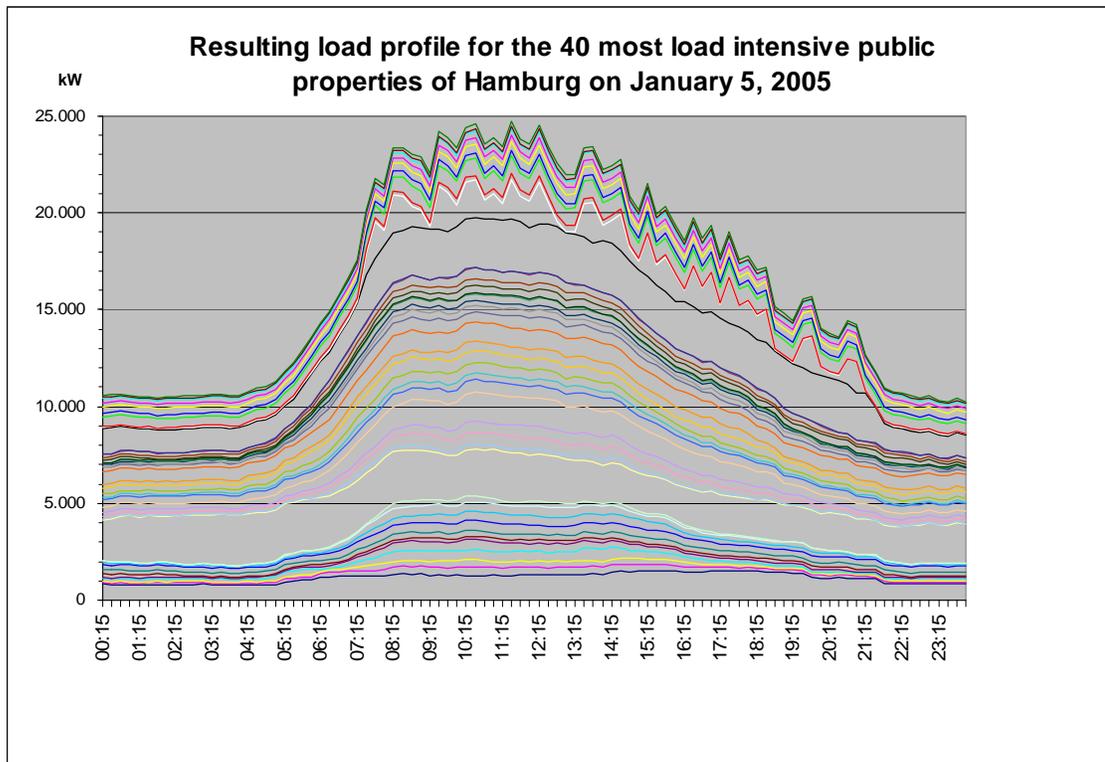


Bild 18: Summenlastprofil der 40 lastintensivsten Liegenschaften Hamburgs [2]

Am Beispiel der Daten (05. 01. 2005) der 40 lastintensivsten Liegenschaften in Hamburg wird ein kumuliertes Lastprofil gezeigt.

## 6.3 Modellbildung zur Vorhersage eines Lastprofils

Wie beschrieben, werden zurzeit die Verbrauchsdaten der einzelnen Liegenschaften alle 15 Minuten ermittelt und in den Datenloggern gespeichert. Ziel ist es aber, die Daten im Intervall von einer Minute aufzunehmen und diese in der Vida 84 für die tägliche Abholung durch den Server zu speichern. Bei einigen „Schlüsselliegenschaften“ soll die Aufzeichnung der Daten in Realtime erfolgen, so dass für diese Liegenschaften Sekundenwerte aufgenommen und in der Datenbank abgelegt werden.

Auf Grundlage dieser Daten und den Wetterdaten der HAW (Standort Bergedorf) wird ein Prognose-Tool trainiert, das aber noch entwickelt werden muss (nicht Teil dieser

Diplomarbeit). Mit den vorhergesagten Lastprofilen soll überschüssige Leistung entweder zur Glättung des Gesamtprofils aller Liegenschaften eingesetzt oder, wenn diese dafür nicht benötigt wird, als Minutenreserve an die Steag Saarenergie verkauft werden. [2]

Um das Prognose-Tool auf Richtigkeit zu überprüfen, werden Datenreihen gespeicherter Lastprofile genommen, ob die vorhergesagten Lastprofile mit den tatsächlichen übereinstimmen.

# 7 Hochverfügbarkeit

In diesem Kapitel soll näher untersucht werden, wie nach den vorgegebenen Anforderungen, beschrieben in Kapitel 2 und 3, der Server so abgesichert werden kann, dass er permanent für Zugriffe bereitsteht. Dabei wird von einer Hochverfügbarkeit des Servers gesprochen. Der Begriff Hochverfügbarkeit bezeichnet die Fähigkeit eines Systems, um bei Ausfall einer seiner Komponenten weiterhin einen uneingeschränkten Betrieb zu gewährleisten. Der Grund für diese uneingeschränkte Präsenz ist einfach zu erklären: Bei einem Ausfall des Servers können die Daten aus den Liegenschaften nicht mehr aufgenommen werden. Somit findet bei einem Ausfall eine Auswertung auf falschen oder nicht vorhandenen Daten statt. Diese Auswertung führt in der Folge zu falschen Vorhersagen, was in bezug auf eine zugesicherte Minutenreserve fatal wäre. Deshalb wird in diesem Kapitel vertieft, welche besonderen Eigenschaften Hard- und Software erfüllen, um die erwartete Verfügbarkeit des Servers (Kapitel 2) auf der einen Seite und der Software (vor allem die Datenbank, Kapitel 4) auf der anderen Seite zu gewährleisten.

## 7.1 Verfügbarkeit und Hochverfügbarkeit

Die Verfügbarkeit eines technischen Systems ist die Wahrscheinlichkeit oder das Maß, dass das System bestimmte Anforderungen und innerhalb eines vereinbarten Zeitrahmens zu erfüllen hat und ist somit eine Eigenschaft und Qualitätskriterium des Systems.

$$\text{Verfügbarkeit} = \frac{\text{Gesamtzeit} - \text{Gesamtausfallzeit}}{\text{Gesamtzeit}}$$

Ein System gilt als hochverfügbar, wenn eine Anwendung auch im Fehlerfall weiterhin verfügbar ist und ohne unmittelbaren menschlichen Eingriff weiter genutzt werden kann. In der Konsequenz heißt dies, dass der Anwender keine oder nur eine kurze Unterbrechung wahrnimmt.

Hochverfügbarkeit (abgekürzt auch HA, abgeleitet von engl. High Availability) bezeichnet also die Fähigkeit eines Systems, bei Ausfall einer oder mehrere seiner Komponenten, einen uneingeschränkten Betrieb zu gewährleisten. Dazu wird die Hochverfügbarkeit in verschiedene Klassen eingeteilt. [17]

HRG Klasse	Verfügbarkeitsklasse	Bezeichnung	Verfügbarkeit in %	Ausfallzeit pro Jahr	Erklärung
-	1	-	90	36,5 Tage	Funktion kann unterbrochen werden, Datenintegrität ist nicht essentiell.
AEC-0	2	Conventional	99	3,7 Tage	Funktion kann unterbrochen werden, Datenintegrität muss jedoch gewährleistet sein.
AEC-1	3	Highly Reliable	99,9	8,8 Stunden	Funktion kann unterbrochen werden, Datenintegrität muss jedoch gewährleistet sein.
AEC-2	4	High Availability	99,99	52,2 Minuten	Funktion darf nur innerhalb festgelegter Zeiten bzw. zur Hauptbetriebszeit minimal unterbrochen werden.
AEC-3	5	Fault Resistant	99,999	5,3 Minuten	Funktion muss innerhalb festgelegter Zeiten bzw. während der Hauptbetriebszeit ununterbrochen aufrechterhalten werden.
AEC-4	6	Fault Tolerant	99,9999	32 Sekunden	Funktion muss ununterbrochen aufrechterhalten werden, 24*7 Betrieb (24 Stunden, 7 Tage die Woche) muss gewährleistet sein.
AEC-5	7	Disaster Tolerant	99,99999	3 Sekunden	Funktion muss unter allen Umständen verfügbar sein.

Tabelle 2: Verfügbarkeitsklassen eines Servers [17]

## 7.1.1 Computercluster

Ein Computercluster oder einfach nur Cluster (von engl. cluster – Schwarm, Gruppe, Haufen) bezeichnet eine Anzahl von vernetzten Computern, die von außen als ein Computer gesehen werden können. Dabei sind die einzelnen Elemente eines Clusters untereinander über ein schnelles Netzwerk verbunden.

Ziel des "Clustering" besteht meistens in der Erhöhung der Rechenkapazität oder der Verfügbarkeit gegenüber einem einzelnen Computer. Die in einem Cluster befindlichen Computer oder Server (Knoten) werden auch oft als Serverfarm bezeichnet.

Der Begriff Cluster an sich beschreibt eher die Architektur der einzelnen Bausteine und ihr Zusammenwirken. Dabei wird zwischen so genannten homogenen und heterogenen Clustern unterschieden. Computer homogener Cluster laufen unter dem gleichen Betriebssystem und gleicher Hardware. Bei einem heterogenen Cluster können unterschiedliche Betriebssysteme oder Hardware eingesetzt werden.

Cluster werden dabei nach ihrer Aufgabe unterschieden. Beispiele dafür sind Hochverfügbarkeitscluster und Server Load Balancing (SLB) Cluster (Kapitel 7.8). Ein weiteres Einsatzgebiet für einen Cluster ist das High Performance Computing (HPC), das aber nicht näher betrachtet wird.

Hochverfügbarkeitscluster (engl. High-Availability-Cluster – HA-Cluster) werden zur Steigerung der Verfügbarkeit bzw. zur besseren Ausfallsicherheit eingesetzt. Tritt auf einem Knoten des Clusters ein Fehler auf, werden die auf diesem Server laufenden Dienste auf einen anderen Server migriert.

Die meisten HA-Cluster besitzen zwei Knoten. Sowohl die Hardware als auch die Software eines HA-Clusters muss frei von Single-Point-of-Failures (Kapitel 7.6) sein. Anwendung finden solche HA-Cluster in kritischen Umgebungen, in denen maximale Ausfallzeiten von nur wenigen Minuten im Jahr erlaubt sind. [18]

## **7.1.2 Failover**

Failover bezeichnet eine Technologie aus der Informationstechnik, mit deren Hilfe Daten und Dienste hochverfügbar gehalten werden können. Unter einem Failover ist der ungeplante Wechsel von einem Primärserver zu einem zweiten Standby-System zu verstehen. Kritische Anwendungen - wie Datenbanken oder Webserver - können auf diese Weise auch bei einem Ausfall eines Rechners weiterhin zur Verfügung gestellt werden, da der Zweitserver im Fehlerfall die Aufgaben des ausgefallenen Systems übernimmt. Zusammen bilden beide Systeme einen Failover-Cluster.

Meist überwachen sich beide Server des Clusters gegenseitig. Dabei wird ein Signal (Heartbeat), das über eine Netzwerkverbindung, beispielsweise über ein Nullmodem-Kabel oder Ethernet, zwischen den Servern zyklisch ausgetauscht. Antwortet ein Server oder eine Anwendung nicht, weil er/sie ausgefallen ist, so initiiert der andere Server automatisch die Übernahme der Dienste. Binnen weniger Sekunden bis Minuten stehen diese dann wieder den Nutzern zur Verfügung. [19]

### **7.1.3 Switchover**

Unter einem Switchover ist in der Informationstechnik ein Wechsel von einem Primärsystem zu einem Standby-System zu verstehen. Anders als bei einem Failover, mit dem ein ungeplanter Wechsel zum Standby-System bezeichnet wird, ist unter einem Switchover ein geplanter Wechsel zu verstehen, der gezielt von einem Systemadministrator eingeleitet wird. Ein Switchover wird beispielsweise ausgeführt, um die Funktionsfähigkeit des Standby-Systems zu prüfen oder aber, um bei Wartungsarbeiten am Primärsystem die Dienste vorübergehend auf dem Standby-System zur Verfügung zu stellen. [20]

### **7.1.4 RAID**

Ein RAID-System (ursprünglich redundant array of inexpensive disks, heute redundant array of independent disks) dient zur Organisation mehrerer physischer Festplatten eines Computers zu einem logischen Laufwerk, das eine größere Speicherkapazität, eine höhere Datensicherheit bei Ausfall einzelner Festplatten und/oder einen größeren Datendurchsatz erlaubt als eine physische Platte.

Während die meisten in Computern verwendeten Techniken und Anwendungen darauf abzielen, Redundanzen (das Vorkommen doppelter Daten) zu vermeiden, werden bei RAID-Systemen redundante Informationen gezielt erzeugt, damit beim Ausfall einzelner Komponenten das RAID als Ganzes seine Funktionalität behält.

Die Entwicklung des RAID-Konzepts führte zunehmend zum Einsatz in Serveranwendungen, die den erhöhten Datendurchsatz und die Ausfallsicherheit nutzen. Die Möglichkeit, in einem solchen System einzelne Festplatten im laufenden Betrieb zu wechseln, entspricht der heute gebräuchlichen Übersetzung: Redundant Array of Independent Disks (Redundante Anordnung unabhängiger Festplatten).

Der Betrieb eines RAID-Systems setzt mindestens zwei identische Festplatten voraus. Die Festplatten werden somit gemeinsam betrieben und bilden einen Verbund, der unter mindestens einem Aspekt betrachtet, leistungsfähiger ist als eine einzelne Festplatte. Mit RAID-Systemen können folgende Vorteile erreicht werden:

- Erhöhung der Ausfallsicherheit (Redundanz oder Spiegellung, siehe auch Kapitel 4)
- Steigerung der Transferraten (Performance)
- Aufbau großer logischer Laufwerke
- Austausch von Festplatten während des Systembetriebes
- Erhöhung der Speicherkapazität
- hohe Steigerung der Systemleistungsfähigkeit

Die genaue Art des Zusammenwirkens der Festplatten wird durch den RAID-Level spezifiziert. Die gebräuchlichsten sind RAID0, RAID1 und RAID5. RAID1 wurde in Kapitel 4 näher beschrieben. Eine Betrachtung von RAID0 und RAID5 erfolgt hier aber nicht, da sie nicht zur Sicherung des Systems beitragen. [4]

### **7.1.5 Single Point of Failure**

Unter einem Single Point of Failure (SPOF, zu Deutsch: „einzelne Fehlerstelle“) sind diejenigen Komponenten eines Systems zu verstehen, die bei einem Fehler den

Komplettausfall eines Systems nach sich ziehen. Bei hochverfügbaren Systemen muss darauf geachtet werden, dass alle Komponenten eines Systems redundant ausgelegt sind.

So werden zum Beispiel die Gehäuselüfter, das Netzteil und die Festplatten (vgl. dazu auch RAID im Kapitel vorher) redundant ausgelegt. Prozessorkühler, die in einem Computer nur einfach vorhanden sind und nicht redundant gemacht werden können, lassen sich dafür aber bei einem Ausfall im laufenden Betrieb austauschen. [22]

### **7.1.6 Standby-Datenbank**

Standby-Datenbanken werden dazu verwendet, um die Ausfallsicherheit zu erhöhen. Eine Standby-Datenbank ist eine transaktionskonsistente (Die Datenbank enthält alle durchgeführten und nicht abgeschlossenen Änderungen) Kopie einer produktiven Primärdatenbank auf einem zweiten Rechnersystem.

Eine Standby-Datenbank wird durch sogenanntes Log Shipping auf dem aktuellen Stand gehalten. Dabei wird das Transaktionslog der Primärdatenbank auf das Standby-System übertragen und dort nachvollzogen. Die Standby-Datenbank ist leicht zeitverzögert auf demselben Stand wie die Primärdatenbank und enthält somit den gleichen Datenbestand.

Fällt die Primärdatenbank aus, können Anwender mit der Standby-Datenbank weiterarbeiten. Für Transaktionen, die auf der Standby-Datenbank ausgeführt werden, ist es notwendig, diese später wieder auf die Primärdatenbank zu übertragen. Optionen für Standby-Datenbanken bieten unter anderem die Hersteller von Oracle und IBM DB2 an. [23]

### **7.1.7 Server Load Balancing**

Der Begriff Serverlastverteilung oder englisch Server Load Balancing (SLB) beschreibt in der Netzwerktechnik Methoden zur Lastverteilung auf mehrere getrennte Server im Netzwerk. Server Load Balancing kommt überall dort zum Einsatz, wo sehr viele Clients oder Prozesse eine hohe Anfragedichte erzeugen und damit einen einzelnen Server

überlasten würden. Typische Kriterien zur Ermittlung der Notwendigkeit von SLB sind die Datenrate, die Anzahl der Clients / Prozesse und die Anfragerate.

Ein weiterer Aspekt ist die Erhöhung der Datensicherheit durch SLB. Der Einsatz mehrerer Systeme ermöglicht redundante Datenhaltung. Die Aufgabe des SLB ist hier die Vermittlung der Clients an die einzelnen Server. [24]

# 8 Absicherung des Servers

Nachdem in Kapitel 7 darauf eingegangen wurde, wie ein Server hochverfügbar gemacht werden kann, lässt sich der Server an sich und die darauf laufenden Dienste auch noch weiter absichern. In Kapitel 4.2.2. wurde am Beispiel des Mailservers `exim4` erläutert, nur die wirklich benötigten Dienste zu installieren. Darüber hinaus können an den Einstellungen von Konfiguration, Diensten usw. Veränderungen vorgenommen werden, die den Server sicherer und damit weniger angreifbar machen.

In diesem Kapitel werden an einigen weiteren Beispielen gezeigt, wie der Server und die laufenden Dienste abgesichert werden können. Leider kann mit diesem Kapitel nur erläutert werden, warum diese Einstellungen und Veränderungen für die Sicherheit des Servers wichtig sind. Aber ein komplett abgesicherter Server lässt sich in Rahmen dieser Diplomarbeit nicht herstellen. Das Thema der Sicherung eines Servers ist so umfangreich, dass das eine weitere Diplomarbeit füllen könnte, um alle Aspekte ausführlich zu untersuchen.

## 8.1 Physikalischer Schutz des Servers

Mit dem physikalischen Schutz des Servers ist gemeint, was unternommen werden kann, um den Ort und die dortigen Gegebenheiten sicherer zu machen. Dabei ist drauf zu achten, dass der Raum, in dem der Server steht, nicht für jedermann zugänglich und immer abgeschlossen ist. Für diesen Raum sollten dann nur ausgewählte Personen einen Schlüssel bekommen.

Bei der Stromversorgung ist dafür zu sorgen, dass diese immer gewährleistet ist. So ist bei der Verkabelung des Servers darauf zu achten, dass die Kabel nicht offen herumliegen. Diese Kabel könnten entweder aus der Steckdose oder aus dem Server herausgerissen werden.

Für den Fall, dass der Strom ausfällt, ist mit einer USV (Unterbrechungsfreie Stromversorgung), die eine Batterie enthält, zu sorgen, so dass der Strom für den Server mindestens kurzfristig weiterhin vorhanden ist. Einen längeren Ausfall der

Stromversorgung lässt sich nur mit dem in Kapitel 7.2 beschriebenen Server-Cluster mit Servern an verschiedenen Standorten überbrücken.

Neben der Kühlung der einzelnen Komponenten im Server sollte auch für die Kühlung des gesamten Servers gesorgt werden. Dabei ist für eine ausreichende Belüftung des Raumes zu sorgen, in dem der Server steht. Darüber hinaus lässt sich für eine Kühlung des Servers am besten durchführen, in dem der Server in einem klimatisierten Rack steht.

## 8.2 Sicherheit des Servers und seiner Dienste

Unter der Sicherheit des Servers ist zu verstehen, dass der Zugang zum Server erschwert wird. So kann mit Setzen eines BIOS-Passwortes verhindert werden, dass die Bootreihenfolge der sich im Server befindlichen Komponenten geändert wird. Außerdem ist gewährleistet, dass der Server immer von der Festplatte gebootet wird. Ein Umgehen der auf dem Server installierten Sicherheitsvorkehrungen mit einer Boot-CD ist dann nicht mehr möglich.

Für eine gewisse Sicherheit kann schon gesorgt werden, indem nur so wenig wie möglich Software installiert wird. Dieser Ansatz wurde bereits bei der Installation von Debian (Kapitel 4.2) verfolgt. Aber selbst bei dieser Grundinstallation werden mehr Dinge mit installiert, die zum beabsichtigten Betrieb dieses Servers gar nicht benötigt werden.

Wie am Beispiel des `exim4` Mailserver gezeigt, sollte immer nur die wirklich benötigte Konfiguration vorgenommen sein. Der Mailserver wurde dabei nur für das lokale Zustellen der Systemnachrichten konfiguriert. Um den Server in diesem Punkt noch sicherer zu bekommen, lassen sich die Systemnachrichten auch über so genannte Cron-Jobs zustellen. Bei einer solchen Konfiguration kann der `exim4` Mailserver komplett abgeschaltet werden.

Während der Installation wurde auch schon darauf hingewiesen, dass Pakete wie der C-Compiler `gcc` nichts auf einem Server zu suchen haben. Mit diesem könnte eventuell Schadcode (z.B. ein Virus) kompiliert werden. Wenn, wie im Beispiel des Erstellens eines Java-Pakets in Kapitel 4.3, ein Compiler benötigt wird, so ist die Erstellung des Pakets auf

einem separaten Rechner durchzuführen. Anschließend wird das Paket auf den Server kopiert.

Beim Betrieb eines Servers, der auch Webinhalte bereitstellt, erfolgt ein Angriff im ersten Schritt meist auf diesen Webserver. So ist bei Verwendung des Apache - Webservers darauf zu achten, dass nur die wirklich benötigten Module geladen werden.

Bei dieser Installation des Servers werden die beiden Webserver (vergl. dazu auch die Kapitel 4.4 und 4.5) über den Internet Superdämon mit geladen. Eine sichere Variante wäre es, diesen Dämon komplett abzuschalten und das Starten der benötigten Dienste direkt über das System vornehmen zu lassen, indem auch hier Cron-Jobs verwendet werden.

Außerdem sollten die neuen Sicherheitsupdates immer eingespielt werden. Dieser Aspekt wurde aber schon bei der Installation des Grundsystems von Debian besprochen und gezeigt, wie das Einspielen der Updates vorgenommen wird (Kapitel 4.2.1).

Wenn im Netzwerk schon ein Router vorhanden ist, ist bereits automatisch ein Paketfilter im Netz installiert. Darüber hinaus kann dem Netz aber auch noch mindestens eine Firewall hinzugefügt werden, um nicht autorisierte Zugriffe auf das Netz zu verhindern. Für die Kommunikation mit dem Server könnte noch SSL (Secure Sockets Layer, eine gesicherte Internetverbindung) eingesetzt werden.

# 9 Zusammenfassung /Ausblick

## 9.1 Zusammenfassung

Diese Arbeit gibt einen Überblick, was bei einer Installation eines Servers wichtig ist. Dabei wurden sowohl die Auswahl und die Beschaffung als auch die Installation eines Servers betrachtet. Den Anfang (Kapitel 2) bildete die Auswahl der Hardware. Wichtige Eigenschaften, die die Hardware haben muss, sind ausführlich beschrieben worden. Außerdem wurde in diesem Kapitel aufgezeigt, was unternommen werden muss, um die in dem Server verwendeten Komponenten (Redundanz von Festplatten, Lüftern und Netzteil) ständig verfügbar zu machen. Dabei wurden auch Vergleiche zu Komponenten aus dem Desktop-Bereich gezogen.

Das darauf folgende Kapitel behandelte die Vor- und Nachteile einzelner Betriebssysteme. Dabei wurde auch darauf eingegangen, was bei der Verwendung der Hardware aus Kapitel 2 zu beachten ist. Darüberhinaus wurde gezeigt, was zur Entscheidung geführt hat, Debian GNU-Linux zu verwenden.

Um zu erläutern, wie die ausgewählte Hard- und Software genutzt werden kann, wurde in Kapitel 4 ausführlich gezeigt, wie das Betriebssystem Debian zu installieren ist. Des Weiteren wurde beschrieben, wie die benötigten Server-Dienste einzurichten sind. Die daraus gewonnenen Erkenntnisse, zeigen die Notwendigkeit auf, den Server vor eventuellen Gefahren abzusichern. Nach Inbetriebnahme des Servers (Kapitel 5) wird im 8. Kapitel gezeigt, wie die installierte Software, wie an einigen Beispielen demonstriert, abgesichert werden kann, um das Risiko einer Kompromittierung des Systems zu minimieren.

Das 6. Kapitel zeigt, wie die in den Liegenschaften gesammelten Daten zunächst auf den Server transferiert und in eine Datenbank geschrieben werden. Anschließend wird in den Unterkapiteln 6.1.1 und 6.1.2 gezeigt, wie die Daten mit Hilfe der beiden Programme PControl und My-JEVis visualisiert werden können. Die beiden weiteren Unterkapitel (6.2 und 6.3) demonstrieren, wozu die gesammelten Daten verwendet werden können.

In Kapitel 7 wurde beschrieben, wie der installierte Server die Eigenschaft „Hochverfügbarkeit“ erlangen kann. Dazu wurden die einzelnen Punkte, mit der die Hochverfügbarkeit erreicht werden kann, aufgezeigt. Abschließend wurde in Kapitel 7.9 untersucht, welchen Nutzen die Hochverfügbarkeit bringt. Dazu wurde betrachtet, wofür der Server eingesetzt werden soll.

## 9.2 Ausblick

Die Erkenntnisse, die während der Installation gewonnen wurden, lassen die Notwendigkeit erkennen, dass auch eine Grundinstallation abzusichern ist. Selbst bei einer solchen Installation, wird noch zu viel Software zur Verfügung gestellt, die gar nicht benötigt wird. Diese Software stellt aber ein Angriffspotential dar, so dass diese wieder entfernt werden müsste.

Aufgrund der Menge der Absicherungen, die allein für Debian vorgenommen werden könnten, war es nicht möglich, alle zu beschreiben, was aber in Zusammenhang eines sicheren Servers interessant gewesen wäre.

Besonders in Bezug auf die Hochverfügbarkeit wurde herausgestellt, dass der Server aufgrund seines Einsatzzwecks, die Anforderungen einer Hochverfügbarkeit erfüllt. Aus diesem Grund ist es erforderlich, die aufgeführten Maßnahmen umzusetzen. Dabei wurde aber nur darauf eingegangen, welche Maßnahmen unternommen werden könnten.

Ein hochverfügbarer Server ist für die geplante Anwendung, Vorhersagen über die Lastprofile zu machen, sicher unabdingbar, da fehlende Informationen in der Datenbank zu falschen Ergebnissen führen.

Außerdem könnte in einem Katastrophenszenario untersucht werden, was unternommen werden muss, um einen Totalausfall eines Servers zu beheben (Disaster Recovery). Ferner könnte noch eine Betrachtung vorgenommen werden, wie und mit welchem Aufwand eine transaktionssichere Datenbank herzustellen ist.

# Anhang

## A Konfiguration Debian

### A.1 Spracheinstellung

Nach Erhalt einer Auswahl von Sprachen wird entschieden, welche Sprache auf dem System installiert werden soll. Die ausgewählte Sprache ist im Moment nicht von Bedeutung, da diese lediglich die Installation erleichtert. Die folgenden Seiten des Installers sind in der Sprache gehalten, die ausgewählt wurde.

Aus diesem Grund fiel die Entscheidung an dieser Stelle für Deutsch. So lässt sich besser nachvollziehen, was auf dem Bildschirm passiert. Wird dieser Rechner später als Server in eine Produktivumgebung eingesetzt, ist die Auswahl der englischen Sprache sinnvoll. Das hat den Vorteil, dass sich der Server universeller einsetzen lässt.

Ein weiterer wichtiger Punkt, der zu beachten ist, ist die Sicherheit des Systems. Gegebenenfalls erforderliche Patches sind früher verfügbar, weil diese in aller Regel zuerst in Englisch bereitgestellt werden. Lokalisierte Patches folgen meistens mit 2-4-monatiger Verspätung. Sollte der Server bzw. die Anwendung, die darauf läuft, sicherheitskritisch sein, bringt das spätere Erscheinen der Patches einen erheblichen Nachteil.

Die ausgewählte Sprache wird mit ENTER bestätigt. Die beiden folgenden Seiten für die Ländereinstellung und das Tastaturlayout werden mit ENTER übernommen. Ein besonderes Augenmerk sollte auf das Tastaturlayout gelegt werden, da ein deutscher Tastaturreiber die Arbeit bei der Installation wesentlich erleichtert. Grund der Erleichterung ist, dass die Installation weitestgehend über eine Konsole erfolgt. So müssten später Befehle eingegeben werden, die einige Zeichen enthalten, wobei nicht unbedingt bekannt sein könnte, wo diese auf der englischen Tastatur liegen.

## **A.2 Erweiterte Netzwerkkonfiguration**

Ist die Konfiguration der Netzwerkkarte erfolgreich abgeschlossen, muss für den Computer ein Name vergeben werden. Letztendlich ist dieser völlig egal. Er muss aber unbedingt aussagekräftig sein. Zu beachten ist ebenfalls, dass für den Namen keine deutschen Sonderzeichen verwendet werden, da diese zum einen nicht zulässig sind und zum anderen zu Problemen führen können.

Nach der Eingabe des Namens wird nach einer Domäne gefragt. Ist eine im Netzwerk vorhanden und wurde die Netzwerkkarte mit DHCP konfiguriert, wird diese jetzt hier automatisch eingetragen. Bei einer manuellen Konfiguration der Netzwerkkarte müsste die Domäne per Hand eingegeben werden. Ist im Netzwerk keine Domäne vorhanden, bleibt der Eintrag leer, und es wird mit ENTER fortgefahren.

## **A.3 Systemuhr**

Zunächst kann die Systemuhr umgestellt werden. Von einer Umstellung ist abzuraten, wenn ein Rechner benutzt wird, auf dem Windows läuft. Da der genutzte Rechner ein Linux-System ist, kann die Uhr ohne Probleme umgestellt werden. Für einen Produktivserver empfiehlt sich die Umstellung der Systemuhr auf GMT. Dazu wird noch die Zeitzone angegeben, die eigentlich automatisch erkannt wird, und mit ENTER bestätigt.

## **A.4 Einstellungen für den apt-Paketmanager**

Für apt, dem Packet Management Tool von Debian, muss jetzt eine Zugriffsmethode festgelegt werden, wie apt sich die neuen Pakete holen soll. Zunächst fiel die Entscheidung, nur das Grundsystem auf CD zu haben, wenn zwischen den Punkten http oder ftp gewählt wird. Mit dieser Einstellung lädt Debian die Pakete von einem Spiegelserver aus dem Internet herunter. Dazu wird im Folgenden angegeben, in welchem Land sich dieser Server befindet. Zu empfehlen ist, einen Server aus Deutschland zu wählen, um genügend Bandbreite für den Download zur Verfügung zu haben. Ansonsten kann womöglich das Herunterladen der Pakete länger dauern als unbedingt nötig. Die Entscheidung fiel auf einen Zugang über FTP und den Spiegelserver ftp.de.debian.org. Anschließend werden die verschiedenen Quellen geprüft und die Paketlisten gelesen.

# B Quelltexte

## B.1 Installationskript JEVIS

```
#!/bin/sh
#
# JEVIS Project installation script
#
# Author: Alec Kojaev
# Date: 2003-05-15
# ChangeLog:
# 2003-05-15 AK Initial version;
#     QUIRKS:
#     - doesn't check for bash(1) version;
#     - relies on dirty heuristics to find version number;
#     - non-numeric subversions are ignored;
#     - Oracle installation is insufficiently tested;
# 2003-07-28 AK Bugs corrected:
#     - error in version comparision (1.3.2 < 1.2.4 ???);
# 2003-07-30 AK Changes:
#     - if exists file ".default", default destination is
#       read from it;
#     - now uninstall and downgrade are done by the same
#       script (specified by options or different name);
# 2003-08-13 AK Changes:
#     - now allows make variables on command line;
# 2003-11-26 AK Changes:
#     - Oracle detection completely rewritten to parse XML
#       inventory;
#     - new options "--initial", "--type";
# 2003-12-01 AK Changes:
#     - Oracle detection is done by components;
#     - Installation type is not guessed;
# 2003-12-02 AK Changes:
#     - Oracle Java support is not required (yet);
#     - Oracle OCCI support is not required (OCI only yet);
# 2003-12-08 AK Changes:
#     - Information about all options in log and before
```

```

#           confirmation request;
# 2004-02-04 AK Changes:
#           - Added information about distribution date (if available);
# 2004-09-29 AK Bugs corrected:
#           - Action 'information' does not require installation type;
# 2005-03-23 AK Changes:
#           - Change in distribution version format;
#
if [ -z "$BASH_VERSION" ]
then
    echo "*** Sorry, installation script requires bash(1) as a shell"
    exit 1
fi

CHK_TOOLS=yes
CHK_ORACLE=yes
DO_MAKE=yes
DO_OPERATION=install
DO_FORCE_CONFIG=no
DEST_PATH=/jevis
INSTALL_TYPE=""
APP_PARAM=""

MAKE_EXEC=make

[ -r ./default ] && DEST_PATH=`head -n 1 ./default`

[ "$(basename $0)" = "uninstall.sh" ] && DO_OPERATION=uninstall
[ "$(basename $0)" = "downgrade.sh" ] && DO_OPERATION=downgrade
[ "$(basename $0)" = "information.sh" ] && DO_OPERATION=information

save_param=""
for param
do
    if [ -n "$save_param" ]
    then
        eval $save_param="\$param"
        save_param=""
        continue
    fi
    case "$param" in

```

```

--help | -h)
    echo "Usage: $0 [options] [name=value...]"
    echo "  where options are:"
    echo "    -d <path>, --destination <path>"
    echo "        Specify destination directory;"
    echo "    -f, --force"
    echo "        Force configuration;"
    echo "    -t, --no-tools"
    echo "        Do not check for tools;"
    echo "    -o, --no-oracle"
    echo "        Do not check for Oracle installation;"
    echo "    --type {client|server}"
    echo "        Force selection of installation type;"
    echo "    -n, --no-make"
    echo "        Do not execute operation;"
    echo "    -i, --install"
    echo "        Install operation (default);"
    echo "    -u, --uninstall"
    echo "        Uninstall operation (default if called as uninstall.sh);"
    echo "    -g, --downgrade"
    echo "        Downgrade operation (default if called as downgrade.sh);"
    echo "    -x, --execute <operation>"
    echo "        Execute specified operation (allows non-standard operations);"
    echo "    -h, --help"
    echo "        This help message;"
    exit 1
;;
--no-tools | -t)
    CHK_TOOLS=no
;;
--no-oracle | -o)
    CHK_ORACLE=no
;;
--no-make | -n)
    DO_MAKE=no
;;
--type)
    save_param="INSTALL_TYPE"
;;
--install | -i)
    DO_OPERATION=install

```

```

;;
--uninstall | -u)
    DO_OPERATION=uninstall
;;
--downgrade | -g)
    DO_OPERATION=downgrade
;;
--execute | -x)
    save_param="DO_OPERATION"
;;
--force | -f)
    DO_FORCE_CONFIG=yes
;;
--destination | -d)
    save_param="DEST_PATH"
;;
-*)
    echo "*** Unrecognized option '$param'"
    exit 1
;;
*=*)
    APP_PARAM="$APP_PARAM $param"
;;
*)
    echo "*** Unrecognized parameter '$param'"
    exit 1
;;
esac
done

INSTALL_TYPE=`echo $INSTALL_TYPE | tr 'A-Z' 'a-z' | tr -d -c 'a-z0-9_'`
[ -z "$INSTALL_TYPE" -a -r "$DEST_PATH/.install_type" ] && \
INSTALL_TYPE=`head -n 1 "$DEST_PATH/.install_type"`
if [ "$DO_OPERATION" != 'information' ]
then
    [ -z "$INSTALL_TYPE" ] && \
    { echo "*** Installation type required"; exit 1; }
    [ "$INSTALL_TYPE" != "client" -a "$INSTALL_TYPE" != "server" ] && \
    { echo "*** Unknown installation type $INSTALL_TYPE"; exit 1; }
fi

```

```

###
### Some helper functions
###
function version_split() {
    local head="$1" tail part="" digit list=""

    #
    # QUIRK: Non-numeric subversion is ignored
    #
    while [ -n "$head" ]
    do
        tail="${head###[0-9]}"
        if [ "$tail" != "$head" ]
        then
            part="$part${head%%$tail}"
            head="$tail"
        else
            [ -z "$part" ] && break
            list="$list $part"
            head="${head###\."}"
            part=""
        fi
    done
    [ -n "$part" ] && list="$list $part"
    echo $list
}

function version_compare() {
    local vl="$1" v1o="$2"
    local el
    for el in $vl
    do
        local elo=${v1o%% *}
        [ -z "$elo" ] && break
        [ -n "$el" -a "$el" -ge "$elo" ] || return 1;
        [ "$el" -gt "$elo" ] && return 0;
        v1o="${v1o#$elo}"
        v1o="${v1o## }"
    done
    return 0;
}

```

```

###
### Check for required tools
###
function gnu_version_hack() {
    #
    # Dirty heuristics here:
    #
    # 1. String 'GNU' or package name should be somewhere
    # 2. Version is the first word matching a pattern that will break
    #    when major version number reaches 10;
    #
    local pkg="$1"
    shift
    local ver test gnu=no vers=""
    while [ -n "$1" ]
    do
        ver="$1"
        # Remove punctuation
        test=".$ver"
        while [ "$test" != "$ver" ]
        do
            ver="$test"
            test="${test##[^0-9A-Za-z]}"
            test="${test%[^0-9A-Za-z]}"
        done
        { [ "$ver" = "GNU" ] ||
          [ "$ver" = "$pkg" ]; } && gnu=yes
        [ -z "$vers" -a "$${ver##[0-9].}" != "$ver" ] && vers="$ver"
        shift
    done
    [ $gnu = yes -a -n "$vers" ] && echo $vers
}

function gnu_version_compare() {
    local cmd="$1" package="$2" vlist="$3"
    local cmdver=`command $cmd --version 2>/dev/null || command $cmd -V 2>/dev/null`
    local ver=`gnu_version_hack "$package" $cmdver`
    local vl=`version_split "$ver"`
    version_compare "$vl" "$vlist" ||
        { echo "'$cmd' not found or older ($ver)"; exit 2; }
}

```

```

}

function check_gnu_package() {
    local package="$1" version="$2" cmd
    echo -n "Checking GNU package $package version $version or higher... "
    shift 2
    local vlist=`version_split "$version"`
    for cmd
    do
        gnu_version_compare "$cmd" "$package" "$vlist"
    done
    echo "OK"
}

```

```

function check_gnu_app() {
    local command="$1" version="$2"
    echo -n "Checking GNU $command version $version or higher... "
    local vlist=`version_split "$version"`
    gnu_version_compare "$command" "" "$vlist"
    echo "OK"
}

```

```

function check_gnu_lib() {
    local library="$1" version="$2"
    echo -n "Checking library $library version $version or higher... "
    local vlist=`version_split "$version"`
    local command=`gcc -print-file-name=$library 2>/dev/null`
    [ -z "$command" ] && { echo "NOT FOUND"; exit 2; }
    gnu_version_compare "$command" "library" "$vlist"
    echo "OK"
}

```

```

if [ $CHK_TOOLS = yes ]
then
    check_gnu_package sh-utils 2.0 \
        echo printf yes false true test expr tee dirname basename pathchk \
        pwd stty printenv tty id logname whoami groups users who date \
        uname hostname chroot env nice nohup su sleep factor seq
    check_gnu_package fileutils 4.0p \
        ls dir vdir dircolors cp dd install mv rm shred ln mkdir rmdir \
        mkfifo mknod chgrp chmod choown touch df du sync

```

```

check_gnu_package textutils 2.0a \
    cat tac nl od fmt pr fold head tail split csplit wc sum cksum md5sum \
    sort uniq comm ptx tsort cut paste join tr expand unexpand
check_gnu_package binutils 2.9.5 \
    ar nm objcopy objdump ranlib readelf size strings strip addr2line
check_gnu_app tar 1.13.17
check_gnu_app gzip 1.2.4
check_gnu_app gcc 2.95.2
check_gnu_app ld 2.9.5
check_gnu_app make 3.79
check_gnu_lib libc.so.6 2.2
fi

###
### Check for Oracle 9i installation
###
function check_oracle_comp() {
    local component="$1" listfile="$2" expected="$3" varname="$4"
    local version vlist vl
    version=`egrep "^"$component"[[:space:]]" "$listfile" | cut -f 2`
    if [ -n "$version" ]
    then
        vlist=`version_split "$expected"`
        vl=`version_split "$version"`
        version_compare "$vl" "$vlist" ||
            { echo "version too old"; rm -f "$listfile"; exit 2; }
    else
        version="NONE"
    fi
    eval V_ORACLE_${varname}="\ "$version"
}

function check_oracle() {
    [ -z "$ORACLE_HOME" ] &&& { echo "NOT FOUND"; exit 2; }
    local expected="$1"
    local listfile="/tmp/oracle-components.$$"
    local compfile="$ORACLE_HOME/inventory/ContentsXML/comps.xml"
    [ ! -f "$compfile" -o ! -r "$compfile" ] &&& { echo "not found or incompatible"; exit 2; }
    awk -f - "$compfile" >"$listfile" <<'EOF'
/<COMP / {
    if (match($0, "NAME=\ "[^"]*" ))

```

```

        name = substr($0, RSTART + 6, RLENGTH - 7);
    else
        next;
    if (match($0, "VER=\["^"\]*\""))
        vers = substr($0, RSTART + 5, RLENGTH - 6);
    else
        next;
    printf "%s\t%s\n", name, vers;
}
EOF

check_oracle_comp "oracle\java\jdbc" "$listfile" "$expected" "JDBC"
check_oracle_comp "oracle\rdbms\oci" "$listfile" "$expected" "OCI"
check_oracle_comp "oracle\rdbms\sqlplus" "$listfile" "$expected" "SQLPLUS"
rm -f "$listfile"
}

if [ $CHK_ORACLE = yes ]
then
    echo -n "Checking Oracle 9i installation... "
    check_oracle 9.2.0.0.0
    echo "JDBC $V_ORACLE_JDBC, OCI $V_ORACLE_OCI, SQL*Plus $V_ORACLE_SQLPLUS"

    if [ "$V_ORACLE_JDBC" = "NONE" -o \
        "$V_ORACLE_OCI" = "NONE" -o \
        "$V_ORACLE_SQLPLUS" = "NONE" ]
    then
        echo "*** Required Oracle components not found"
        exit 2
    fi
fi

MAKE_EXEC="$MAKE_EXEC $APP_PARAM INSTALLMODE=$INSTALL_TYPE"

if [ "$DO_OPERATION" = "information" ]
then
    echo ""
    cd src/ && \
        $MAKE_EXEC roster
    exit
fi

```

```

###
### Execute make
###
START_DATE=`date +%Y-%m-%d-%H-%M-%S`
DIST_VARIANT=""
DIST_VERSION=""
DIST_DATE=""
if [ -f ".version" -a -r ".version" ]
then
    DIST_VARIANT=`head -n 1 ".version" | cut -f 1`
    DIST_VERSION=`head -n 1 ".version" | cut -f 2`
    DIST_DATE=`head -n 1 ".version" | cut -f 3`
fi

function more() {
    #
    # Hack for Sun Java SDK installator
    #
    less -M "$@" 2>/dev/tty >/dev/tty
}

function result_report() {
    local curtime=`date +%Y-%m-%d-%H-%M-%S`
    echo ""
    echo "@@@ $curtime *"
}

function log_header() {
    echo "###"
    echo "### Distribution type = $DIST_VARIANT"
    echo "### Distribution build = $DIST_VERSION"
    echo "### Distribution date = $DIST_DATE"
    echo "### Destination path = $DEST_PATH"
    echo "### Operation = $DO_OPERATION"
    echo "### Installation type = $INSTALL_TYPE"
    echo "###"
    echo "### Check tools = $CHK_TOOLS"
    echo "### Check Oracle = $CHK_ORACLE"
    echo "### Execute operation = $DO_MAKE"
    echo "### Force configuration = $DO_FORCE_CONFIG"
    echo "### Additional param. = $APP_PARAM"
}

```

```

    echo "###"
}

function do_install() {
    {
        echo "@@@ $START_DATE Starting $INSTALL_TYPE installation to $DEST_PATH"
        log_header
        echo ""
        {
            { [ $DO_FORCE_CONFIG != yes ] || $MAKE_EXEC config; } &&
            $MAKE_EXEC install-opt &&
            $MAKE_EXEC install &&
            $MAKE_EXEC clean-build &&
            result_report "Installation completed successfully";
        } || result_report "Installation completed with errors";
    } 2>&1
}

function do_execute() {
    local oper="$1" textlow="$2" textcap="$3"
    {
        echo "@@@ $START_DATE Starting $textlow in $DEST_PATH"
        log_header
        echo ""
        {
            $MAKE_EXEC $oper &&
            result_report "$textcap completed successfully";
        } || result_report "$textcap completed with errors";
    } 2>&1
}

if [ $DO_MAKE = yes ]
then
    if [ "$DO_OPERATION" != "install" ]
    then
        if [ ! -d "$DEST_PATH" -o \
            ! -w "$DEST_PATH" -o \
            ! -f "$DEST_PATH/inst/config" ]
        then
            echo "*** Installation not found at path $DEST_PATH"
            exit 2
        fi
    fi
fi

```

```

fi
fi
if [ ! -d /tmp -o ! -w /tmp ]
then
    echo "*** /tmp should exist and be writeable"
    exit 2
fi

inst_log="/tmp/install.$$"
real_log="$DEST_PATH/var/instlog/$START_DATE"

cd src/
export PROJ_ROOT="$DEST_PATH"
export -f more

log_header
retval=""
while [ -z "$retval" ]
do
    echo -n "Are you ready to continue [y/n]? "
    read retval
    [ "$retval" = "y" -o "$retval" = "Y" ]      && retval="yes"
    [ "$retval" = "n" -o "$retval" = "N" ]      && retval="no"
    [ "$retval" != "yes" -a "$retval" != "no" ] && retval=""
done
[ "$retval" != "yes" ] && echo "Aborted, exiting." && exit 0

case $DO_OPERATION in
install)
    do_install
    ;;
uninstall)
    do_execute uninstall "uninstallation" "Uninstallation"
    ;;
downgrade)
    do_execute downgrade "downgrade" "Downgrade"
    ;;
*)
    do_execute $DO_OPERATION "operation '$DO_OPERATION'" "Operation '$DO_OPERATION'"
    ;;
esac | tee "$inst_log"

```

```
mv -f "$inst_log" "$real_log"
fi
```

## B.2 Installationskript Java

```
#!/bin/bash
#
#-----
# ----- Anpassung der Apt Sources -----
#-----
echo="deb ftp://ftp.de.debian.org/debian/ etch main non free contrib" >> /etc/apt/sources.conf
#-----
# ----- Installation java-package -----
#-----
apt-get update
apt-get install java-package
#
#-----
# ----- Herunterladen des aktuellen Packets -----
#-----
cd /tmp/
wget -O jdk-1_5_0_06-linux-i586.bin
http://192.18.108.149/ECom/EComTicketServlet/BEGIN5C5902DBA428CEF9DA9F4B38862185DF/-
2147483648/2207627667/1/832394/832190/2207627667/2ts+/westCoastFSEND/jdk-1.5.0_12-oth-JPR/jdk-
1.5.0_12-oth-JPR:5/jdk-1_5_0_12-linux-i586.bin
#
#-----
# ----- Erzeugen des deb-Packets -----
#-----
su oliver
fakeroot make-jpkg jdk-1_5_0_12-linux-i586.bin
dpkg -i sun-sdk1.5_1.5.0+update12_i386.deb
sudo dpkg -i sun-sdk1.5_1.5.0+update12_i386.deb
#
#-----
# ----- Anlegen der Links für Java -----
#-----
ln -s /usr/lib/j2sdk1.5-sun /usr/lib/java
```

```

cp /tmp/profile /etc/profile
#
#-----
# ----- Ueberprüfen der Java-Version -----
#-----
java -version

```

### B.3 Installationskript Apache

```

#!/bin/bash

apt-get update
apt-get install apache2
/etc/init.d/apache2 start

```

### B.4 Installationskript Tomcat

```

#!/bin/bash
#
#-----
# ----- Verzeichnis, wo die Installationspakete zu finden sind: -----
#-----
SW_REPOSITORY="/tmp/"
#-----
# ----- Installationsverzeichnis: -----
#-----
INSTALL_BASE="/opt"
#
#-----
# ----- Herunterladen des aktuellen Packets -----
#-----
cd $SW_REPOSITORY/
wget http://ftp.hosting-studio.de/pub/linux/apache/tomcat/tomcat-5/v5.5.23/bin/apache-tomcat-5.5.23.tar.gz
#
#-----
# ----- Installation der Tomcat Binarys -----
#-----

```

```

cd ${INSTALL_BASE}
tar -xzf ${SW_REPOSITORY}/apache-tomcat-5.5.23.tar.gz
chmod -R 775 ${INSTALL_BASE}/apache-tomcat-5.5.23
ln -s ${INSTALL_BASE}/apache-tomcat-5.5.23 /opt/tomcat
#
#-----
# ----- Anlegen eines eigenen Benutzers für den Tomcat -----
#-----
groupadd -g 65 tomcat
useradd -u 65 -g 65 -d ${INSTALL_BASE}/tomcat -s /bin/false -c "Tomcat User" tomcat
chown -R tomcat:tomcat ${INSTALL_BASE}/tomcat
usermod -G www-data tomcat
#
#-----
# ----- Instalations des Tomcat Daeons -----
#-----
cd ${INSTALL_BASE}/tomcat/bin
tar -xzf jsvc.tar.gz
chmod -R 755 jsvc-src
cd jsvc-src
autoconf
./configure
make
cp jsvc ..
cd ..
#-----
# ----- Startscript für den Tomcat erstellen -----
#-----
cp ${SW_REPOSITORY}tomcat /etc/init.d/tomcat
chmod 755 /etc/init.d/tomcat
#-----
# ----- Runlevel für den Tomcat festlegen -----
#-----
update-rc.d tomcat start 91 2 3 4 5 . stop 20 0 1 6 .
#
#-----
# ----- Start des Tomcats -----
#-----
alias tomcat=/etc/init.d/tomcat
tomcat

```

## B.5 Installationsskript Oracle

```
#!/bin/bash
#
#-----
# ----- Installation weiterer Pakete -----
#-----
apt-get update
apt-get install gcc make binutils libmotif3 lesstif2 rpm libaio unzip
#
#-----
# ----- Anlegen der Links -----
#-----
ln -s /usr/bin/awk /bin/awk
ln -s /usr/bin/rpm /bin/rpm
ln -s /usr/bin/basename /bin/basename
ln -s /etc /etc/rc.d
cp /tmp/oraparam.ini /tmp/database/Disk1/install/
#
#-----
# ----- Anlegen eines eingenen Benutzers oracle -----
#-----
groupadd oinstall
groupadd dba
groupadd nobody
useradd -g oinstall -G dba -p passwd -d /home/oracle oracle
useradd -g nobody nobody
mkdir -p /home/oracle
mkdir -p /opt/oracle/product/oracle
mkdir -p /opt/oracle/product/oradata
chown -R oracle:oinstall /opt/oracle
chmod -R 775 /opt/oracle
chown -R oracle:oinstall /home/oracle
chmod -R 775 /home/oracle
#
#-----
# ----- Kopieren der Konfigurationsdateien -----
#-----
cp /tmp/sysctl.conf /etc/sysctl.conf
/sbin/sysctl -p
```

```
cp /tmp/limits.conf /etc/security/limits.conf
cp /tmp/login /etc/pam.d/login
cp /tmp/su /etc/pam.d/su
cp /tmp/profile /etc/profile
```

# C Ausdrücke der Konfigurationsdateien

## C.1 /etc/apt/sources.conf

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

deb ftp://ftp.de.debian.org/debian/ stable main
deb-src ftp://ftp.de.debian.org/debian/ stable main

deb http://security.debian.org/ stable/updates main

deb ftp://ftp.de.debian.org/debian/ etch main non-free contrib
```

## C.2 /etc/profile

```
JAVA_HOME="/usr/lib/java"
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "`id -u`" -eq 0 ]; then

    PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11:$JAVA_HOME/bin:$ORACLE_HOME/lib"
else
    PATH="/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games:$JAVA_HOME/bin:$ORACLE_HOME/lib"
fi

if [ "$PS1" ]; then
    if [ "$BASH" ]; then
        PS1='\u@\h:\w\$ '
    else
        if [ "`id -u`" -eq 0 ]; then
            PS1='# '
        else
            PS1='$ '
        fi
    fi
fi
```

```

fi
fi

if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16834
        ulimit -n 65536
    else
        ulimit -u 16834 -n 65536
    fi
fi

export PATH JAVA_HOME ORACLE_HOME PS1

umask 022

```

## C.3 /etc/init.d/tomcat

```

#!/bin/sh
#####
#####
#
# Copyright 2004 The Apache Software Foundation.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#####
#####
#
# Small shell script to show how to start/stop Tomcat using jsvc

```

```

# If you want to have Tomcat running on port 80 please modify the server.xml
# file:
#
# <!-- Define a non-SSL HTTP/1.1 Connector on port 80 -->
# <Connector className="org.apache.catalina.connector.http.HttpConnector"
#     port="80" minProcessors="5" maxProcessors="75"
#     enableLookups="true" redirectPort="8443"
#     acceptCount="10" debug="0" connectionTimeout="60000"/>
#
# That is for Tomcat-5.0.x (Apache Tomcat/5.0)
#
# Adapt the following lines to your configuration
JAVA_HOME=/usr/lib/java
CATALINA_HOME=/opt/tomcat
DAEMON_HOME=/opt/tomcat
TOMCAT_USER=tomcat

# for multi instances adapt those lines.
TMP_DIR=/tmp
PID_FILE=/var/run/jsvc.pid
CATALINA_BASE=/opt/tomcat

CATALINA_OPTS="-Djava.library.path=/home/jfclere/jakarta-tomcat-connectors/jni/native/.libs"
CLASSPATH=\
$JAVA_HOME/lib/tools.jar:\
$CATALINA_HOME/bin/commons-daemon.jar:\
$CATALINA_HOME/bin/bootstrap.jar

case "$1" in
start)
#
# Start Tomcat
#
echo "Starting web server (tomcat)..."
$DAEMON_HOME/bin/jsvc \
-user $TOMCAT_USER \
-home $JAVA_HOME \
-Dcatalina.home=$CATALINA_HOME \
-Dcatalina.base=$CATALINA_BASE \
-Djava.io.tmpdir=$TMP_DIR \
-wait 10 \

```

```

-ioutfile $CATALINA_HOME/logs/catalina.out \
-errfile '1' \
$CATALINA_OPTS \
-cp $CLASSPATH \
org.apache.catalina.startup.Bootstrap
#
# To get a verbose JVM
#-verbose \
# To get a debug of jsvc.
#-debug \
exit $?
;;

stop)
#
# Stop Tomcat
#
echo "Stopping web server (tomcat)..."
$DAEMON_HOME/bin/jsvc \
-stop \
-pidfile $PID_FILE \
org.apache.catalina.startup.Bootstrap
exit $?
;;

*)
echo "Usage tomcat.sh start/stop"
exit 1;;
esac

```

# Literaturverzeichnis

1. **Schubert, Prof. Dr. Ing Franz.** Forschung. [Online] 2007. [Zitat vom: 3. Mai 2007.] <http://users.etech.haw-hamburg.de/users/schubert/forschung.html>.
2. **Schäfers, Hans.** *Zwischenbericht Projekt INSEL*. [Doc-Datei] Hamburg : s.n., 2007.
3. Arbeitsspeicher. [Online] Wikimedia Foundation Inc, 22. Juni 2007. [Zitat vom: 28. 06 2007.]
4. Serial Attached SCSI. [Online] 04. Juni 2007. [Zitat vom: 15. Juni 2007.] [http://de.wikipedia.org/wiki/Serial\\_Attached\\_SCSI](http://de.wikipedia.org/wiki/Serial_Attached_SCSI).
5. Betriebssystem. [Online] 30. 05 2007. [Zitat vom: 05. Juni 2007.] <http://de.wikipedia.org/wiki/Betriebssystem>.
6. Distribution. [Online] Wikimedia Foundation Inc, 06. April 2007. [http://de.wikipedia.org/wiki/Distribution\\_%28Software%29](http://de.wikipedia.org/wiki/Distribution_%28Software%29).
7. Debian. [Online] Wikimedia Foundation Inc, 10. Juni 2007. [Zitat vom: 11. Juni 2007.] <http://de.wikipedia.org/wiki/Debian>.
8. Red Hat. [Online] Wikimedia Foundation Inc, 10. Juni 2007. [Zitat vom: 12. Juni 2007.] [http://de.wikipedia.org/wiki/Red\\_Hat](http://de.wikipedia.org/wiki/Red_Hat).
9. openSUSE. [Online] Wikimedia Foundation Inc, 10. Juni 2007. [Zitat vom: 12. Juni 2007.] <http://de.wikipedia.org/wiki/Suse>.
10. Booten. [Online] 30. April 2007. [Zitat vom: 03. Mai 2007.] <http://de.wikipedia.org/wiki/Booten>.
11. Runlevel. [Online] Wikimedia Foundation Inc, 09. Mai 2007. [Zitat vom: 15. Juni 2007.] <http://de.wikipedia.org/wiki/Runlevel>.
12. X Windows System. [Online] Wikimedia Foundation Inc, 27. Mai 2007. [Zitat vom: 05. Juni 2007.] [http://de.wikipedia.org/wiki/X\\_Window](http://de.wikipedia.org/wiki/X_Window).
13. **Envidatec.** *Serverbackup*. [odt-Datei] Hamburg : Envidatec GmbH, 2004.
14. My-JEVis Portal. [Online] Envidatec GmbH, 2007. [Zitat vom: 15. Juni 2007.] <http://www.envidatec.com/templates/standard.php?naviid=31&&sprache=1&&charset=ISO-8859-1&&proj=1>.
15. JEChart. [Online] Envidatec, 2007. [Zitat vom: 09. Juni 2007.] [http://www.my-jervis.com/html\\_help/jechart/index\\_help\\_jechart.html](http://www.my-jervis.com/html_help/jechart/index_help_jechart.html).
16. JeBench. [Online] Envidatec, 2007. [Zitat vom: 09. Juni 2007.] [http://www.my-jervis.com/html\\_help/jebench/index\\_help\\_jebench.html](http://www.my-jervis.com/html_help/jebench/index_help_jebench.html).
17. Hochverfügbarkeit. [Online] Wikimedia Foundation Inc, 07. Mai 2007. [Zitat vom: 15. Mai 2007.] <http://de.wikipedia.org/wiki/Hochverf%C3%BCgbarkeit>.
18. Computercluster. [Online] 12. Mai 2007. [Zitat vom: 19. Mai 2007.] <http://de.wikipedia.org/wiki/Computercluster>.

19. Failover. [Online] 04. April 2007. [Zitat vom: 15. Mai 2007.] <http://de.wikipedia.org/wiki/Failover>.
20. Switchover. [Online] Wikimedia Foundation Inc, 08. Februar 2007. [Zitat vom: 15. Mai 2007.] <http://de.wikipedia.org/wiki/Switchover>.
21. RAID. [Online] Wikimedia Foundation Inc, 30. April 2007. [Zitat vom: 15. Mai 2007.] <http://de.wikipedia.org/wiki/RAID>.
22. Single Point of Failure. [Online] Wikimedia Foundation Inc, 05. Mai 2007. [Zitat vom: 15. Mai 2007.] [http://de.wikipedia.org/wiki/Single\\_Point\\_of\\_Failure](http://de.wikipedia.org/wiki/Single_Point_of_Failure).
23. Standby-Datenbank. [Online] Wikimedia Foundation Inc, 09. März 2007. [Zitat vom: 15. April 2007.] <http://de.wikipedia.org/wiki/Standby-Datenbank>.
24. Server Load Balancing. [Online] Wikimedia Foundation Inc, 13. Mai 2007. [Zitat vom: 15. Mai 2007.]
25. **Gulbins, Jürgen, Obermayer, Karl und Snoopy.** *Linux*. Berlin : Springer, 2003. 3-540-00815-2.
26. **Alex, Wulf.** *Debian GNU/Linux in der Praxis*. Berlin : Springer, 2006. 3-540-23786-0.
27. **Schroder, Carla.** *Linux Kochbuch*. Köln : O'reilly, 2005. 3-89721-405-9.
28. **Kersken, Sascha.** *Apache 2*. Bonn : Galileo Press, 2006. 3-89842-787-0.

# Erklärung

Hiermit erkläre ich, die vorliegende Diplomarbeit im Sinne der Prüfungsordnung nach §25(4) selbständig und ausschließlich unter Verwendung der angegebenen Literatur angefertigt zu haben.

Hamburg, den 5. Juli 2007

Oliver Göttsch