

# Florian Stöhr

---

Kontakt	Lerchenweg 7, 86492 Egling +49-173-3514000 ich@florian-stoehr.de
Profil	Freiberuflicher Softwareentwickler C#, Python (CPython und IronPython), Assembler (16/32/64 bit, DOS/Win32/Win64/Linux/BSD), C++ (Qt, wxWidgets, VCL, MFC, std-C++, STL, Threading, boost); Reversing (native IDA, CIL IDA+ILDASM+Telarik) Auditor für QM-Systeme nach ISO 9001
Berufserfahrung	10/2009–jetzt <b>Freiberufliche Tätigkeit</b> 06/2005–09/2009 Aipermon GmbH & Co. KG München <b>Abteilungsleiter Server-Entwicklung</b> 06/2001–05/2005 LogIn & Solutions AG Gersthofen <b>Entwicklungsleiter, stellv. Abteilungsleiter</b> 01/2001–06/2001 Skybeamer GbR Aulzhausen <b>Gesellschafter, Bereich Softwareentwicklung</b> 02/1997–12/2000 Stöhr Software Egling <b>Nebenberuflich EDV-Dienstleistung und Softwareentwicklung</b>
Ausbildung	10/2009–10/2009 TÜV Süd Akademie GmbH Augsburg <b>Ausbildung zum QM-Auditor ISO 9001</b> 09/1999–06/2001 BOS (Technik) Augsburg <b>Fachhochschulreife</b> 09/1995–02/1999 Deutsche Bahn AG München <b>Ausbildung zum Kommunikationselektroniker</b>
Sprachkenntnisse	<ul style="list-style-type: none"><li>• Muttersprache Deutsch</li><li>• Hervorragende Kenntnisse in Englisch in Wort und Schrift</li><li>• Grundkenntnisse in Hindi in Wort und Schrift (Devanagari)</li></ul>
Sonstiges	<ul style="list-style-type: none"><li>• Medizinprodukteberater (EUROCAT)</li></ul>
Programmiersprachen	Assembler, C/C++, C#, CIL, Delphi/Pascal, Python, lex/yacc, Natural
Debugger	TurboDebugger, gdb, DDD, WinDBG, IDA Pro
Versionskontrolle	cvs, subversion, git, Mercurial, TFS
Datenbanken	Access, BDE, MySQL, ODBC, PostgreSQL, SQLite, SQL-Server
Hardware	Alpha, Amiga, Apple, Ascii/X-Terminals, Drucker, embedded, HP, Mikro- controller, Motorola, PC, Silicon-Graphics, SUN, MFA-8085

---

# Florian Stöhr

---

Betriebssysteme	CP/M, Mac OSX, MS-DOS, Solaris, Unix allgemein, Windows (ab 2.0), Windows CE (Mobile), NetBSD/FreeBSD/OpenBSD, Linux (Debian und Derivate, SuSE, CentOS), iOS, Android
Kommunikation	CORBA, Sockets (TCP/IP, UDP/IP, UDP-Multicast), HTTP, SOAP, RS232, parallel, SMTP, Google Protocol Buffers, JSON via REST, u.v.a. (Eigenbau)
Sonstige Technologien	gcc, Borland C++ Builder, Borland Delphi, Lazarus, FASM, NASM, TASM, Watcom C++, MS-Visual C++, MS-Visual Basic, MS-Visual C#, CIL, Mono, MonoTouch, MonoGame, XNA, IronPython, TurboPascal, vim, (n)curses, TurboVision, VCL, MFC, wxWidgets, Qt, cURL, libUSB, GNU pthreads, Multi-Threading, STL, DirectX, Orbit ORB, MIDAS, make, GNU Autotools (autoconf, automake, libtool, ltdl), bjam, gmake, qmake, boost, Apache, Postfix, Samba, spamd, QtCreator, Django, CherryPy, PySide, C++/Python-Schnittstellen mit boost::python, ImageMagick, InstallShield, NSIS, Windows Services, FinBanks, NUnit, WPF, WCF
Branchen	3D-Rendering, Handelsnetz, eCommerce, Bildverarbeitung, Logistik, Lagersteuerung, Lagerverwaltung, Medizintechnik, Medizingeräte, Medizin, Bildungswesen, Verkehr, Eisenbahn, öffentliche Verwaltung, Computerspiele, Pharmazulassung, Finanzen

## Projekte

–Auszug–

### **09/2015–jetzt: Embedded-Medizingerät**

Branche:

Medizinbranche (Pulsion GmbH, München)

Rolle:

Entwicklung, Planung, Requirements Engineering

Projekt:

Ein Blutdruckmeßgerät für den Einsatz in der Intensivmedizin sollte in einen Arbeitsteil und einen Monitorteil getrennt werden, bisher in einem Produkt untergebracht. Der Kunde wollte den Arbeitsteil als eigenes Modul vermarkten, so daß Mitbewerber dieses in ihre eigenen Monitore integrieren können. Die ursprüngliche Implementierung lag in Delphi, Assembler und C++ vor, das neue Modul sollte vollständig in C/C++ auf einem ARM Cortex M4-Controller implementiert werden. Zusätzlich wurde ein C++-SDK auf PC-Seite und ein Python-Wrapper dafür benötigt. Das Testsystem wurde komplett in Python implementiert (unittest-Modul), außerdem wurde ein Webfrontend (Twitter-Bootstrap+jQuery) mit Python/CherryPy als Backend bereitgestellt

Technologien:

Delphi, Assembler, GCC/G++-ARM, Makefiles, Subversion, Python, CherryPy

Plattformen:

ARM32, Win64

### **09/2014–09/2015: C# Logistik/Lagersteuerung/Kampagnenplanung**

Branche:

Modebranche (Best Secret GmbH, München)

Rolle:

Entwicklung

Projekt:

Mitarbeit an der Lagersteuerungssoftware für einen großen Modeversender. Die Software steuert den kompletten Lagerablauf, also Auftragsaufbereitung, Kommissionierung, Steuerung der Packstationen, Druck der Labels + Interface zu DHL/Zoll usw. Außerdem Mitarbeit am Programm für die Kampagnenplanung und Artikelbestellung der Verwaltung. Außerdem diverse Python-basierte Hilfsprogramme zur Datenbereinigung und ein (mit Tkinter) implementiertes Frontend zur Batchkompilierung/Distribution der .NET-Services

Technologien:

C#/.NET, WCF, WPF, Mercurial, PDFFileWriter, NHibernate, Autofac, VB.NET, Python

Plattformen:

Win64

---

## 06/2014–07/2014: Magletics-Frontend für Gehirnstimulator

Branche:

Medizintechnik (Mag&More GmbH, München)

Rolle:

Entwicklung

Projekt:

Für eine Sonderanwendung der Ansteuerungssoftware für einen magnetischen Gehirnstimulator mußte ein neues Frontend auf WPF-Basis erstellt werden. Die Steuerung des Stimulators erfolgt über einen direkt angesteuerten FTDI-Chip via USB-Schnittstelle.

Technologien:

C#, WPF, .NET, FTDI

Plattformen:

Win64

## 10/2013–06/2014: Webapplikation für Pharmazulassung

Branche:

Pharma (EXTEDO GmbH, München)

Rolle:

Entwicklung (Backend)

Projekt:

Aufbau einer C#-basierten Webapplikation zur gesetzeskonformen Einreichung und Verwaltung der Zulassungsdokumente, Stofflisten und Nebenwirkungstabellen für Medikamente. Die Software unterstützt die Medikamentenzulassung über den gesamten Lebenszyklus des Produkts. Bedingt durch den enormen Zeitdruck seitens des Gesetzgebers wurde zusätzlich – bis das JavaScript-basierte Frontend fertig ist – als vorübergehende Lösung noch eine Import/Export-Schnittstelle via Excel-Dateien realisiert. Desweiteren wurde ein PDF-Generator implementiert. Ein Teil des Codes wurde über einen in Python geschriebenen Generator erzeugt.

Technologien:

C#, EntityFramework, SQL-Server, TFS, Python, OpenXML/ClosedXML, "PDFFileWriter" library

Plattformen:

Win64

---

## 10/2012–06/2013: Integration des Python-Interpreters in 3D-Visualisierungssoftware

Branche:

3D-Rendering (Realtime Technology (RTT) AG, München)

Rolle:

Architektur, Entwicklung

Projekt:

Einbau des Python-Interpreters in eine 3D-Visualisierungssoftware für die Automobil- und Luftfahrtbranche. Der Interpreter-Host wird als PlugIn geladen und kann intern beliebig viele Subinterpreter abspalten. Die Ausführung der Python-Skripte erfolgt üblicherweise (wobei die Skripte dafür zuständig sind) in Hintergrundthreads. Zusätzlich stellt das PlugIn ein internes Python-Modul (in C) zur Verfügung, über das Python-Callables auf dem Qt-Hauptthread der Anwendung ausgeführt werden können, da ggf. einige API- Aufrufe zwingend den Qt-Hauptthread voraussetzen.

Technologien:

Python, Python C API, Qt, C++

Plattformen:

Win64

## 10/2012–10/2012: Updateservice für iOS-Banking-Lösung

Branche:

Finanzen

Rolle:

Architektur, Entwicklung

Projekt:

Für eine neuartige iOS-Banking-Lösung, die durch Vorausladen von Umsätzen einen Geschwindigkeitsvorteil gegenüber anderen Lösungen erzielen soll, musste eine Serverkomponente erstellt werden. Diese ist als Windows-Service ausgeführt. Aufgabe war es, regelmässig Bankumsätze via FinBanks-Bibliothek von den offiziellen Gegenstellen der Banken zu laden und in einer Datenbank abzulegen. Ferner sind automatische Joberzeugung und -verteilung sowie Lastausgleich in der Komponente integriert.

Technologien:

C#, FinBanks, PostgreSQL, NUniT

Plattformen:

.NET 4.0

---

## 08/2012–09/2012: Spiel "Pancake" für iOS

Branche:

Computerspiele (selbständig)

Rolle:

Alle

Projekt:

Nachbau des LCD-Hand-Konsolenspiels "Pancake", ursprünglich von VTECH 1981 auf den Markt gebracht, vorerst für iOS (iPhone/iPad). Komplette mit MonoTouch (C# unter iOS) und dem MonoGame-Framework (Nachbau von Microsoft XNA) implementiert. Für den Verkauf im AppStore.

Technologien:

C#, MonoTouch, MonoGame

Plattformen:

OS 5.1 (iPhone, iPad)

## 03/2012–06/2012: EEG-Viewer für iOS

Branche:

Medizin (sh computing UG)

Rolle:

Design, Entwicklung

Projekt:

Als Demonstrator für den Messeinsatz wird eine Software benötigt, die Gehirnstromkurven aus dem branchenüblichen EDF-Format performant und mit Touch-Unterstützung auf einem iPad anzeigen kann. Implementierung erfolgt über das MonoTouch-Framework (C#-Entwicklung auf iOS).

Technologien:

C#, MonoTouch

Plattformen:

iOS 5.1 (iPad)

---

## **12/2011–08/2012: Parser für NATURAL-Sourcecode und IronPython-Tool**

**Branche:**

Zulieferer öffentliche Verwaltung (it&more GmbH, München)

**Rolle:**

Entwicklung in C#, Design und Implementierung IronPython-Werkzeug, Beratung

**Projekt:**

Im Auftrag der öffentlichen Hand musste aufgrund einer neuen gesetzlichen Vorgabe die NATURAL-Applikation des Meldewesens auf Unicode-Fähigkeit umgestellt werden. Diese Applikation besteht aus ca. 3 Millionen Zeilen NATURAL-Quellcode, historisch über mehr als 20 Jahre gewachsen. Im Projekt musste eine C#-Lösung erstellt werden, die automatisiert den NATURAL-Code auf Unicodefähigkeit umstellt. Hierbei müssen zum Teil Konstrukte im Code automatisch durch völlig andere Konstrukte ersetzt werden, so dass simples Ersetzen nicht funktioniert - der Code muss tatsächlich geparsed und zum Teil auch "verstanden" werden (Wanderung von Daten zwischen Modulen, Werteweitergabe zwischen Variablen, Generieren von neuen Steuerelementen in Dialogen usw.).

Zusätzlich wurde ein leistungsfähiges Debug-Werkzeug auf Basis von IronPython (Python für .NET = skriptbasierter Zugriff auf das komplette Datenmodell der Applikation) erstellt.

**Technologien:**

C#, .NET, Python, IronPython

**Plattformen:**

Windows 7 64, Windows Server 2008 R2

## **08/2011–11/2011: Steuerungssoftware für Gehirnstimulator**

**Branche:**

Medizin (Mag&More GmbH, München)

**Rolle:**

Entwicklung (Neuentwicklung), Beratung

**Projekt:**

Der Kunde konstruiert und vertreibt Medizingeräte, die durch Magnetfelder Impulse direkt in Gehirnregionen einleiten (nicht-invasiv). Hierzu musste eine Software neu erstellt werden, die Patientendaten verwaltet sowie Ansteuerkurven erstellt, verarbeitet und in ein Format umsetzt, welches der Controller der Geräteschnittstelle versteht. Die Verbindung zur C#-Software wird über einen FTDI-USB-Chip hergestellt, welcher im Direktverfahren angesteuert wird. Zusätzlich muss die Software eine Echtzeitvisualisierung des "Programmablaufs" in der Kurve anzeigen (der Controller meldet -gepollt- Spulentemperatur und Scriptfortschritt zurück).

**Technologien:**

C#, .NET, FTDI

**Plattformen:**

Windows XP/Vista/7

---

## **05/2010–10/2011 (Teilauslastung): Systemadministration Linux + VCS**

Branche:

eCommerce (euroblaze Wapsol GmbH, Stuttgart)

Rolle:

Administration, Integration, teilweise Entwicklung

Projekt:

euroblaze erstellt nach Kunden/Designvorgaben Onlineshops auf Basis von OXID. Die Entwicklung selbst findet in Indien statt, aktuell werden über 100 Projekte von euroblaze betreut. Ich bin verantwortlich für die Systemadministration, Installation auf produktiven Maschinen und die Verwaltung des Versionskontrollsystems (subversion, bootstrap, Berechtigungen, Branchzuweisung, ...).

Technologien:

subversion, Linux, OXID, Apache, PHP, ZendGuard, ionCube, MySQL, Python

Plattformen:

Linux, FreeBSD

## **07/2011–07/2011: Virtuelle Maschine für OXID-Tests**

Branche:

eCommerce (euroblaze Wapsol GmbH, Stuttgart)

Rolle:

sämtliche Rollen

Projekt:

Virtuelle Maschine zum Testen von Installationen der OXID eCommerce-Lösung (PHP-basiert). Die VM automatisiert die die Administration des Linux-Hosts sowie die Installation von OXID-Versionen. Sie bootet automatisch in eine Python/cdialog- Applikation, die Installationen (+Systemaccounts +Datenbank +Konfiguration +Apache- Alias), Datenbanktausch, Filemanagement, Systemupdate und Update der Installationstemplates bzw. der VM-Anwendung selbst menügesteuert auch unerfahrenen Anwendern zur Verfügung stellt. Die VM erfordert keinerlei manuelle Eingriffe mehr und konfiguriert im Fall einer IP-Änderung der VM Anpassungen der OXID- Konfigurationen automatisch durch.

Technologien:

Linux, OXID, Apache, PHP, ZendGuard, ionCube, MySQL, Python, cdialog, Shell

Plattformen:

Linux (Debian6-i386)

---



## **10/2010–07/2011: Webbasierte Kontaktplattform/Backend/PySide (QT4)-Frontend**

Branche:

Vermittlung von Handelspartnern (sh computing UG)

Rolle:

sämtliche Rollen

Projekt:

Webplattform, die Vertriebsleute und Hersteller weltweit zusammenführt. 100% Python-basiertes Projekt. Frontend via Django, Backend CherryPy-Applikationsserver, einzelne Python-Minicons und ein PySide (=Qt4 via Python)-Frontend zum Freischalten von Annoncen.

Technologien:

Linux, Python, Django, CherryPy, PySide, QT4, PostgreSQL, git

Plattformen:

Linux (Ubuntu Server), Multi-Plattform (QT4-Frontend)

## **09/2010–06/2011: Schnittstelle zu 3D-Render-Suite und Prozeßautomatisierung**

Branche:

3D-Rendering (Realtime Technology (RTT) AG, München)

Rolle:

Entwicklung

Projekt:

Für eine führende 3D-Render-Suite (QT4) im Automobil/Luftfahrt/Bekleidungs Umfeld sollte eine C++-Schnittstelle mit Python-Anbindung entwickelt werden. Das System arbeitet selbstkonfigurierend im Cluster (selbständiges Wählen eines Masters) und beherrscht Lastverteilung. Die komplette Schnittstelle kann auch – via boost::python – von Python aus angesteuert werden. In der zweiten Phase musste eine komplett scriptbasierte, automatisierte Renderlösung zum Generieren von Katalogbildern für den Bekleidungskonzern Adidas auf Basis von 3D-Daten erstellt werden. Auf C++-Seite zusätzliche Bildverarbeitung (u.a. combine, color profile, clipping path, PNG chunk reordering) sowie Erstellen und Warten des Installationsprogramms (NSIS)

Technologien:

C++; boost; boost::python; STL; TCP/IP; Multicast; Verschlüsselung; Eigene Kommunikationsprotokolle; Python; CherryPy; Curses; ImageMagick; WinDBG; git; subversion

Plattformen:

Win64, Linux

---

## **09/2009–07/2010: Generischer Sprachtrainer**

Branche:

Privatanwender

Rolle:

sämtliche Rollen

Projekt:

Die Software ermöglicht Vokabular/Übersetzungen/Lückentexte für beliebige Sprachen zu trainieren. Sie unterstützt auch exotische Sprachen wie z.B. Hindi oder Sprachen, die von rechts nach links geschrieben werden.

Technologien:

C++; Qt4; STL; XML; Verschlüsselung; git

Plattformen:

Win32, Mac OS X, Linux, BSD

## **06/2005–09/2009: Systeme für medizinische Datenverarbeitung**

Branche:

Medizintechnik (Konsortialprojekt; Gefördert durch Bundeswirtschaftsministerium)

Rolle:

Systemdesign; Projektmanagement; Planung; Implementierung; Teamleitung; Serveradministration; Mitarbeit an der QM-Dokumentation für ISO 9001 und ISO 13485. Wechselnd 2-10 Mitarbeiter im Team.

Projekt:

Das System besteht aus mehreren Komponenten. Zum einen ein ausfallsicherer und hochverfügbarer Server (C++ unter FreeBSD), der die von den Endgeräten roh zur Verfügung gestellten Messdaten verarbeitet und an eine elektronische Patientenakte zur weiteren Aufbereitung weiterleitet. Zum anderen eine auf Windows-Mobile-PDAs installierte Lösung, die via Bluetooth Messdaten von (hardwareseitig manipulierten = mit Bluetooth-Modulen nachgerüsteten) Messgeräten in Empfang nimmt und Rückmeldung an den Endbenutzer liefert. Ausserdem diverse Wartungs- und Hilfsprogramme.

Technologien:

C; C++; C#; STL; POSIX; Threading; FreeBSD; Windows Mobile; MFC; wxWidgets; AES (Verschlüsselung); Bluetooth; TCP/IP; Eigene Übertragungsprotokolle; embedded; PostgreSQL; SQLite; FTDI-Devices (Direktansteuerung); git; subversion; cvs

Plattformen:

FreeBSD; Linux; Win32; Windows Mobile (5, 6)

---

## **07/2003–06/2005: Softwarebasierte Personalressourcenplanung**

Branche:

Logistik - Andreas Schmid Logistik AG; Siemens SBS

Rolle:

Systemdesign; Projektmanagement; Implementierung; Teamleitung; Serveradministration;

Projekt:

Es handelt sich um eine Anwendung in Client/Server-Anordnung, die dazu dient, Arbeitsabläufe elektronisch abzubilden und zu verfolgen. Hierbei können Abläufe verzweigt und Subabläufe wiederverwendet werden. Die Dauer einzelner Vorgänge wird durch Mitarbeiterinformationen oder sensorische Meldungen erfasst. Auch können so Abläufe beendet bzw. in Gang gesetzt werden. Das Programm verwaltet ferner die verwendeten Ressourcen, die entstehenden Statistiken können zur Resourceoptimierung oder für Vorhersagen verwendet werden. Benötigte Ressourcenaufwände lassen sich auch für zukünftige Projekte vorhersagen.

Technologien:

C++; STL; VCL; PostgreSQL; ODBC; TCP/IP; Eigene Übertragungsprotokolle; cvs

Plattformen:

Win32; FreeBSD

## **05/2001–06/2005: Lagersteuerungssystem Gefahrgutlager + Rechnerverteilung**

Branche:

Logistik, IT - Andreas Schmid Logistik AG; Siemens SBS

Rolle:

Systemdesign; Projektmanagement; Implementierung; Teamleitung; Serveradministration;

Projekt:

Das Projekt musste aus Kostengründen so aufgesetzt werden, dass es sowohl ein Gefahrgutlager als auch die Verteilung von Rechnern bei Siemens-SBS (also ebenfalls Lager, kein Gefahrgut, aber diverse Zusatzfunktionen) abbilden konnte, dementsprechend generisch und erweiterbar konzipiert. Das System besteht aus einer Serverkomponente, die die eigentliche Lagerverwaltung bzw. Warenflußsteuerung/Belegsteuerung abwickelt sowie aus diversen Clientmodulen, etwa für die Arbeitsplätze im Leitstand, die Anbindung der Scanner (Funkscanner, Echtzeit) oder sonstigen Pick-Endgeräte (Pick-by-voice, Pick-by-light). Das System beherrscht Gefahrgutlagerung, Chargen, Barcodescanner, Mehrbenutzerfähigkeit, Verteilte Standorte, Drucksysteme gesetzter Formulare, One-Touch-Logistik (=automatische Prozesskette), diverse Automaten, Mandantenfähigkeit und beliebige Import/Exportschnittstellen.

Technologien:

C++; Python; VCL; STL; Threading; PostgreSQL; ODBC; VT100; Delphi; TCP/IP; XML; Eigene Übertragungsprotokolle; cvs

Plattformen:

FreeBSD; Linux; Win32

---

## **01/1999–05/2001: Steuerung für internetbasiertes Shopsystem**

Branche:

Shopsysteme - Skybeamer GbR

Rolle:

Systemdesign; Projektmanagement; Planung; Implementierung; 3 Mitarbeiter.

Projekt:

Die Software war das Steuerungsmodul für ein internetbasiertes Shopsystem und wurde auch als (kleine) Warenwirtschaftslösung mit minimaler Lagerverwaltung eingesetzt.

Technologien:

C++; STL; VCL; PostgreSQL; ODBC; TCP/IP;

Plattformen:

Win32

## **09/1997–02/1998: Echtzeit-Überwachung im Zugverkehr (S-Bahn München)**

Branche:

(Eisenbahn-)Verkehr - Deutsche Bahn AG

Rolle:

Eigenverantwortliche Planung und Implementierung (Prototyp).

Projekt:

Der sogenannte "Streckenspiegel", der den Fahrdienstleitern für ganz Süddeutschland in Echtzeit Einblick in den Zugverkehr gibt, hat informativen Charakter; Die Steuerung des Zugverkehrs (S-Bahn München) selbst erfolgt losgelöst davon über Prozeßrechner, die allerdings mit identischen Eingangsdaten arbeiten. Die alte Version des Streckenspiegels war mit speziellen Maschinen mit 8085-Prozessoren implementiert. Ich habe einen Prototyp der Software für den Einsatz unter MS-Windows (damals Win16 und Win32) implementiert während meines Aufenthalts im damaligen Rechenzentrum der DB in München im Rahmen meiner Ausbildung. Die Software umfasste sowohl den eigentlichen Streckenspiegel (Anzeigeprogramm) als auch einen Gleisplaneditor und einen speziellen Fonteditor für die Symbole. Mittlerweile (2012) ist der Streckenspiegel in Form der Smartphone-App "S-Bahn-Navi" auch für die Öffentlichkeit zugänglich.

Technologien:

C++; MFC; Watcom

Plattformen:

Win16; Win32

---

## Veröffentlichungen

–Auszug–

- **ISBN 978-3936546-48-4 (C&L-Verlag, 2007)**  
Buch "GNU Autotools - UNIX-Buildsysteme mit Autoconf, Automake und Libtool"
  - **Toolbox - Entwicklermagazin, 02/2008**  
Titelthema ".NET-PlugIns mit C#"
  - **Toolbox - Entwicklermagazin, 03/2007**  
Beitrag "Make my file - so funktioniert GNU Make"
  - **iX - Magazin für professionelle Informationstechnik, 09/2005**  
Beitrag "Verschlüsselte Block-Devices mit NetBSDs CGD"
  - **freeX - Magazin für freie UNIX-Systeme, 06/2005**  
Beitrag "VND gut komprimiert"
  - **freeX - Magazin für freie UNIX-Systeme, 04/2005**  
Beitrag "(Net)BSD auf SSH-Rootservern"
  - **Postfix**  
Artikel "Postfix@NetBSD - Setting up a secure real-life mailserver with SMTP AUTH and anti-spam capabilities"
  - **NetBSD-Handbuch ("The NetBSD Guide")**  
Abschnitt "The cryptographic device driver (CGD) - Creating an encrypted CD/DVD"
  - **OpenBSD zyd(4) Gerätetreiber**  
Erste Version des Gerätetreibers für ZyDAS ZD-1211-basierte USB-WLAN-Adapter.
  - **neb-cd512**  
Programm zur Anpassung der Kernel-in-core-Labels einer CD/DVD an eine andere Sektorengroße (für Krypto-CDs/DVDs)
  - **neb-wipe**  
Programm zur sicheren und endgültigen Vernichtung von Datenträgerinhalten. Basierend auf Forschungsergebnissen zur Wiederherstellung von magnetischen Datenträgern unter Laborbedingungen. Implementierung des von Peter Gutmann vorgeschlagenen 35-Pass-Musterverfahrens. Kann gesamte Datenträger oder einzelne Partitionen sicher zerstören. Entwickelt für NetBSD.
  - **neb-hdtoolbox**  
Programm um NetBSD für die amiga-Plattform crossinstallieren zu können. Kann das spezielle RDSK/PART-Format des Amiga (vergleichbar mit der Partitionstabelle auf der Intel-Plattform) selbständig erzeugen und verändern. NetBSD/amiga verfügte vorher nicht über entsprechende Software und war auf Fremdprogramme angewiesen. Kann auch eingetragene Partitionen ändern.
  - **cloop**  
Komprimiertes Dateisystem für Live-CD-Images (Kompressor/Steuerung)
-