

Diese HTML-Datei enthält spezielle CSS2-Anweisungen, die für den [Präsentationsmodus](#) des Web-Browsers [Opera](#) erstellt wurden. Sofern sie diesen Browser benutzen, können sie mit der F11- Taste in diesen Modus (und wieder zurück) schalten. Die Präsentation wurde für eine Auflösung von 1280x1024 Bildpunkten erstellt.



Datenbanken - Was ist das?

- Vortragsreihe "Chaos-Seminar"
- Veranstalter: CCC, Erfa-Kreis Ulm
 - <http://www.ulm.ccc.de/>
 - mail@ulm.ccc.de
 - Montagstreff: ab ca 19:30, Cafe Einstein (Uni)
- Referent: Markus Schaber
 - <http://www.schabi.de/>
- Vortrag ist im HTML-Format online
 - <http://www.ulm.ccc.de/~schabi/datenbanken/>

Inhaltsübersicht

Kurzer, exemplarischer Überblick

- Beispiele für Datenbanken
- Grundlegendes
- Logische Modelle
 - Relational, Objektorientiert, Hierarchisch, Netzwerk, Logisch
- Implementierungsaspekte
 - Datenstrukturen, Transaktionen
- Zugriff auf Datenbanken
 - Kommandozeile, eingebettetes SQL, explizites API
- Datenschutz / Datensicherheit

Beispiele für Datenbanken

- Adressbuch im Handy

- Kundendatenbank
- Einwohnermelderegister
- Kontobewegungsdatenbank
- Genom-Datenbank
- World Wide Web
- Domain Name System

Grundlegendes

- Trennung von
 - Physischem Datenbestand
 - Logische Struktur der Daten
 - Verarbeitungslogik
 - Benutzeroberfläche
- Designhilfsmittel
 - ER-Diagramme
 - Softwaretools

Relationale Datenbanken

- Heute meistverbreitete Datenbankform
- Aufbau aller Daten aus Tabellen
- Mathematische Grundlage: Mengen von Tupeln
- Identifikation über Primärschlüssel
- Verknüpfung über *Fremdschlüssel*
- Standardisierter Zugriff über SQL

Relationale Datenbanken

Studenten:

<u>MatNr</u>	<u>Name</u>	<u>Strasse</u>	<u>Ort</u>	<u>Telefon</u>
0815	Maier	Meyerhofstr. 1	89075 Ulm	555 1234
4711	Müller	Mühlberg 42	89075 Ulm	555 4321

Einschreibungen:

<u>MatNr</u>	<u>StudNr</u>	<u>EinschrDatum</u>	<u>ExMatDatum</u>	<u>Studiengang</u>
0815	1	1. 8. 1999	1. 9. 2000	Mathematik

4711	1	13. 8. 2000	(null)	Informatik-D
0815	2	1. 9. 2000	(null)	Informatik-B

Studiengaenge

Studiengang	Fakultät	Regeldauer	Abschlußgrad
Informatik-D	Informatik	9	Dipl. Inf.
Informatik-B	Informatik	6	Bachelor
Mathematik	Mathe-Wiwi	9	Dipl. Math.

Relationale Datenbanken

- Zugriff mengenbasiert
- Keine explizite Navigation (in der Theorie)
- Verknüpfen von Tabellen
 - Schnitt und Vereinigung
 - Join (Kreuzprodukt)
- Auswahl von Reihen über Selektion
- Auswahl von Spalten über Projektion
- Viele Möglichkeiten zur Anfrageoptimierung
- Konsistenzprüfung durch die Datenbank

Relationale Datenbanken

Probleme:

- Tabellenfelder haben feste Maximalgröße
 - z. B. Foto des Studenten soll elektronisch in der Datenbank archiviert werden
- Sehr aufwendige Modellierung bei komplexen, hierarchischen Objekten
 - z. B. Student hat mehrere Adressen oder Telefonnummern
 - z. B. Studiengang von mehreren Fakultäten gleichzeitig
- Aufwendige Simulation von Polymorphie
 - z. B. Person, davon abgeleitet Student, Mitarbeiter, Professor
- Dies führt zu:
 - Explosion der Tabellenanzahl
 - Komplexe Abfragen
 - Geschwindigkeitsprobleme

- Sonderbehandlung von variablen, großen Objekten (BLOBs)
- Insellösungen und Speziatricks der Datenbankimplementierer

Relationale Datenbanken

Zukunftsaussichten:

- Bessere, umfassendere Standardisierung
- Erweiterung des Modelles
 - Tupel, Listen, Arrays
 - Strukturen
 - Objektrelationale Systeme
 - Vererbung zwischen Tabellen

Objektorientierte Datenbanken

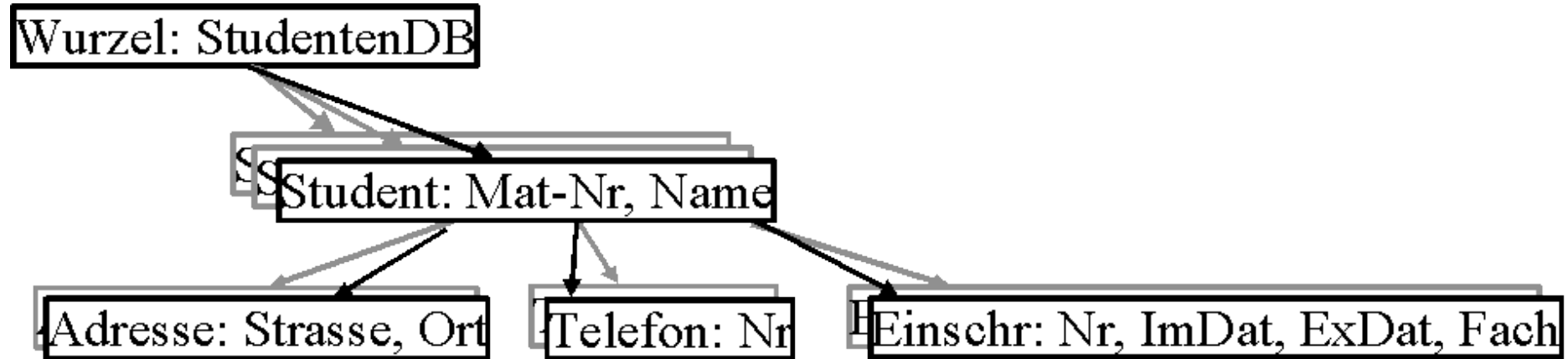
- Basiert auf objektorientierter Programmierung
- Dient als Dauerspeicher in OO-Systemen
- mehr oder weniger Transparent
- Viele verschiedene Systeme
- Oft nur Wrapper für relationale Datenbanken
- Beispiele:
 - ZODB / ZEO
 - EJB-Server
- Oft nur "Storage", keine weitergehende Funktionalität

Hierarchische Datenbanken

- Bekanntestes System: IMS/VS (Seit 1968, von IBM)
 - Inzwischen stark optimiertes, stabiles System
 - Komplexes Datenmodell
 - Kryptischer Zugriff
 - mehrere Tausend Seiten Systemliteratur
- Datenbank hat Baumstruktur
- Navigation geschieht explizit
- Pro logischer Datenbank eine Wurzel
- Alle Records hängen direkt oder indirekt von den Wurzeln ab

Hierarchische Datenbanken

Beispiel



Hierarchische Datenbanken

- Zugriff über PCB-Sicht (virtueller Layer, kann ausblenden)
- Zugriffssprache DL/1 (data language/one)
- Eingebettet in COBOL, PL/1 und 370/Assembler
- Datenein- und Ausgabe über spezielle Variablen
- Cascading Delete von abhängigen Datenblöcken
- Navigation mittels expliziter Kommandos:

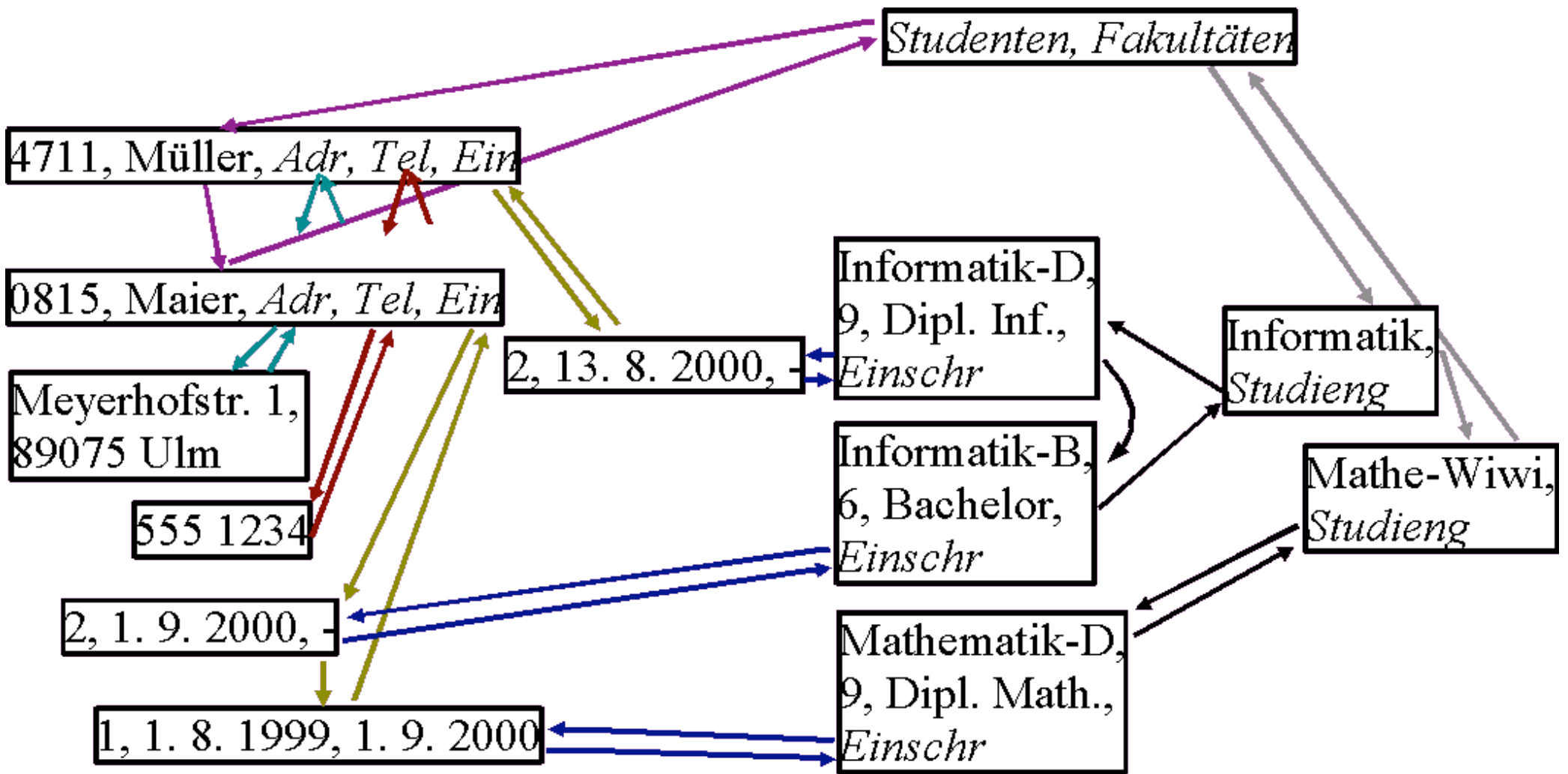
- Gehe zu erstem Kind vom Typ Adresse
- Gehe zu nächstem Kind
- Suche Kind mit Strasse = "Meyerhofstr. 1"

Netzwerk-Datenbanken

- CODASYL-Normierungsbemühungen Ende 60er Jahre
- Aktuelle Implementierungen basieren auf 1978er Vorschlag
- Produkte:
 - DMS/100 (UNIVAC)
 - IDMS (Cullinet Software)
 - DBMS (DEC)
 - UDS (Siemens, in Deutschland sehr erfolgreich)
- Basis-Struktur: Verkettete Listen von Records

Netzwerk-Datenbanken

Beispiel



Netzwerk-Datenbanken

- Zugriff ebenfalls über Kommunikationsvariablen
- Navigation explizit
- für jeden Record-Typen "current"-Pointer
- n:n-Verknüpfungen mittels Kett-Records
 - Einschreibung im Beispiel erfüllt solche Funktion

Logische Datenbanken

- Direkte Verarbeitung logischer Regeln
- Verarbeitung einer Faktenbasis anhand dieser Regeln
- Datenbanksprache DATALOG (ähnlich PROLOG)

Beispiel:

```
VaterVon(Hans, Otto).  
VaterVon(Otto, Frank).  
VaterVon(Otto, Peter).
```

```
GrossvaterVon(A, B) :-  
  VaterVon(A, C), VaterVon(C, B).
```

Datenstruktur: Hash

- Zugriff in quasi konstanter Zeit
- Datensatznummer errechnet über Hashfunktion
- Lesen, Schreiben und Löschen schnell
- Probleme
 - Gute Hashfunktion finden
 - Kollisionsbehandlung notwendig
 - Reorganisation nötig, wenn Tabelle voll
 - Kein sortiertes Lesen möglich

Beispiel:

Schlüssel = Matrikelnummer
Hashfunktion = mod 7

Hash	Key	Data
0	4711	Müller, Mühlbergstr. 42....
1		
2		
3	0815	Maier, Meyerhofstr. 1,...
4		
5		
6		

Datenstruktur: ISAM

- für große Datensatzmengen
- Optimiert für Suche
- Konstante Zugriffszeit
- Mehrstufiger Index

1	Aubele	02	01	Aalfisch	Bla
	Bebraham	03		Abeler	Additi

mehrstufiger index

- Seitenbasiert
- Reorganisation beizeiten notwendig
- Lücken müssen gelassen werden
- Überlaufbehandlung notwendig
- Sequentielles Lesen gut möglich
- Bereichsauswahl gut möglich

Endres	2
Maier	3
zzzz	4

2. Index-
Stufe

	Dietrich	04
2	Endres	05
	Hans	06
	Kuller	07
3	Maier	08
	Müller	09
	Quackera	10

1. Index-
Stufe

	--	
02	Aubele	Bla
	Annerman	Blubb
	Appelt	4711

(...)

05	Endres	000111
	Ezmail	CCC
	Haarling	nst

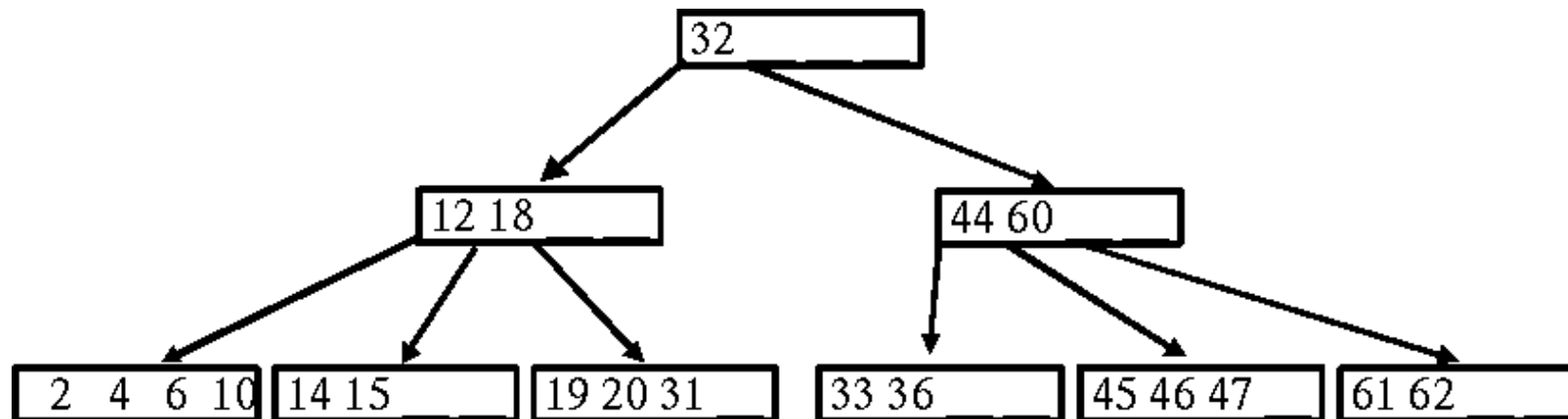
(...)

10	Quackera	sadfax
	Zirler	
	--	

Daten

Datenstruktur: B-Baum

- Knoten fester Größe
- Exaktes Regelwerk gewährleistet Balance
 - Split, Zusammenfassen, Rotationen
 - Füllgrad der Knoten mind. 50% (Ausnahme: Wurzel)



Transaktionen

- Alles-oder-Nichts-Prinzip
- Koppelt zusammengehörende Zugriffe
- Dient der Konsistenzwahrung
- Problematik bei konkurrierenden Zugriffen
 - Dirty Read
 - Non-Repeatable Read
 - Phantom Read
- ANSI/ISO SQL definiert 4 "Isolation Levels"

Zugriff: SQL

- Zugang zu SQL-Datenbanken über Kommandozeileninterface

Codebeispiele:

```
SELECT Name, Strasse, Ort
FROM Studenten
WHERE MatNr = '4711'
```

```
SELECT Name, MatNr
FROM Studenten NATURAL JOIN Einschreibungen
WHERE ExMatDatum = NULL
```

Zugriff: Embedded SQL

- SQL-Abfragen in "Mutterprogramm" eingebettet
- Verarbeitung mittels Präprozessor

Codebeispiel:

```
EXEC SQL INCLUDE SQLCA;
```

```
main() {
    EXEC SQL BEGIN DECLARE SECTION;
    int MatNr;
    char Name[32];
    long Anzahl;
```

```
EXEC SQL END DECLARE SECTION;

EXEC SQL CONNECT TO dbs98;

EXEC SQL SELECT count(*) INTO anzahl
FROM Studenten NATURAL JOIN Einschreibungen
WHERE ExMatDatum = NULL;
}
```

Zugriff: JDBC

- Standard-Schnittstelle für SQL-Zugriff unter Java
- Plugin-Architektur

Codebeispiel:

```
import java.sql.*;

public class Test {
    public static void main( String args[] ) {
        Class.forName("org.postgresql.Driver");
        Connection db = DriverManager.getConnection("jdbc:postgresql:database",
            "user", "password");

        Statement st = db.createStatement();
        ResultSet rs = st.executeQuery("SELECT * FROM Studenten");

        while (rs.next()) {
            System.out.println(rs.getString(1));
        }
        db.close();
    }
}
```

Zugriff: ZODB

- Objektorientierte Datenbank für Python
- Ursprünglich für Zope entwickelt
- Verschiedene Backends (File, ODBC, Verteiltes DBS)

Codebeispiel

```
import ZODB
from ZODB import FileStorage, DB
from Persistence import Persistent

storage = FileStorage.FileStorage("/tmp/test-filestorage.fs")
db = DB(storage); conn = db.open(); dbroot = conn.root()
userdb = dbroot['userdb']

class User(Persistent): pass

X = User(); X.id = "schabi"
X.first_name = "Markus"; X.last_name = "Schaber"

userdb[X.id] = X

get_transaction().commit()
```

Datenschutz / Datensicherheit

- Schutz vor unbefugtem Zugriff notwendig
 - Sicherheitslücken
- Beachtung der Datenschutzgesetze
- Beste Sicherheitsstrategie: Daten vermeiden
- "*Gewährleistung und Begrenzung des Informationsflusses* "

Weiterführendes

Ausgelassene Themen - z. B.:

- Datenbankentwurf
- Verteilte Datenbanken
- Indizierung mehrdimensionaler Daten
- Multimedia-Datenbanken
- Wissensbasierte Systeme und Knowledge Management
- Workflow-Management-Systeme
- Recovery nach Crash
- Und vieles andere mehr...

Literatur

- Prof. Peter Dadam: Datenbanksysteme, Skript zur Vorlesung, Fassung WS 2000/2001, Universität Ulm, Fakultät für Informatik
- Gottfried Vossen, Kurt-Ulrich Witte (Hrsg): Entwicklungstendenzen bei Datenbanksystemen, Oldenburg Verlag, München, 1991
- PostgreSQL 7.1 User's Guide und PostgreSQL 7.1 Programmer's Guide <http://www.postgresql.org/>
- Klaus Meyer-Wegener: Multimedia-Datenbanken: Einsatz von Datenbanktechnik in Multimedia-Systemen, Teubner-Verlag, Stuttgart, 1991
- Andrew M. Kuchling, ZODB & ZEO Introduction, <http://www.amk.ca/zodb/zodb-zeo.html>

Ende

- <http://www.ulm.ccc.de/>
- mail@ulm.ccc.de
- Montagstreff: ab ca 19:30, Cafe Einstein (Uni)
- Verwendete Software:
 - Opera 5.12/Win und 5.05TP1/Linux
 - Netscape 6.1
 - StarOffice 5.2
 - GIMP 1.2.1
 - Debian Woody Linux
 - MS Windows 2000