

BELEG

Fach: Maschinennahe Programmierung

Thema: Realisierung eines MP3-Tag-Editors

HTWM

Hochschule für Technik
und Wirtschaft Mittweida
- University of applied
sciences

Eingereicht von: Isabel
Drost
Seminargruppe: IF99p1
Datum: 06.04.2001

Inhaltsverzeichnis:

Inhaltsverzeichnis:	2
1. Einleitung	3
1.1. Gründe Für die Wahl des Belegthemas:	3
1.2. Notwendige Kriterien:	3
1.3. Optionale Kriterien:	3
1.4. Abgrenzungskriterien:	4
1.5. Copyright:	4
1. 7. Knappe Definition der Aufgabe des Programms:	4
2. Bedienungsanleitung für Nutzer:	4
2.1. Erster Start	4
2.2. Auflistung der Tasten zur Bedienung des ID3Editors:	6
2.3. Bekannte Probleme	6
3. Quellcodedokumentation:	7
3.1 Allgemeiner Programmaufbau:	7
3.2. Modularisierung des ID3TagEditors:	8
3.3 Das Modul „einlesen.c“:	9
3.5. Das Modul „Edit.c“	11
4. Quellenangabe	12
4.1. Verwendete Literatur	12
4.2. Verwendete Software:	12
5. Erklärung	12

1. Einleitung

Dank dem Fraunhofer-Institut ist es möglich geworden, digitalisierte Musikdateien um ein erhebliches Maß (verlustbehaftet) zu komprimieren. Die Größe der Dateien läßt sich nahezu ohne Qualitätsverlust auf etwa ein Zehntel reduzieren. Dies führt nicht nur dazu, daß der gesamte Hausbestand an Musik in MP3-Form auf wenigen CDs mitführbar ist, sondern veranlaßt auch viele Studenten (und Professoren), die Musik, welche nun auch transportabler geworden ist, mehr oder weniger legal aus dem Internet herunterzuladen.

Zur Spezifizierung dieser Dateien nutzen gängige MP3-Player ein sogenanntes ID3-Tag. Dieses ist aber nicht selten fehlerhaft oder überhaupt nicht vorhanden. Das bedeutet, dass meist ein einfach und schnell zu bedienendes Programm vonnöten ist, um diesen Mißstand zu beheben.

1.1. GRÜNDE FÜR DIE WAHL DES BELEGTHEMAS:

Das Thema für den Beleg im Fach Maschinennahe Programmierung war uns freigestellt. Das heißt, man mußte sich nicht nur mit einer gegebenen Aufgabenstellung auseinandersetzen, sondern vorher überlegen, was man überhaupt realisieren möchte.

Das Projekt sollte, meinen Überlegungen nach, auf der einen Seite in der verhältnismäßig kurzen Zeit realisierbar sein, aber auch einen gewissen Nutzen erfüllen und thematisch interessant sein. Ansprüche, die nicht leicht zu erfüllen sind.

Weiterhin war zu entscheiden, für welches Betriebssystem man ein Programm entwickeln wollte. Meine Entscheidung fiel hier auf Linux. Gründe dafür waren unter anderem, dass ich mich mit Hilfe dieses Projekts – sowie auch mit den Übungsaufgaben aus dem C-Praktikum – näher mit der Entwicklung von Programmen unter Linux auseinandersetzen wollte.

Nun blieb nur noch zu entscheiden, welches Thema ich genau wählen sollte. In dem Projekt wollte ich mich mit der Dateiarbeit, mit der Verwaltung von eigenen Strukturen und der Realisierung einer graphischen Benutzeroberfläche über die ncurses-Bibliothek auseinandersetzen. Da ich schon länger nach einer Möglichkeit suche, id3-Tags einfach und bequem bearbeiten zu können, kam mir die Idee, einen Editor zu erstellen, der genau diese Funktion erfüllt. Er sollte einfach zu handhaben sein und eine Möglichkeit bieten, schnell die mp3-Dateien auszuwählen, die für das Editieren in Frage kommen.

1.2. NOTWENDIGE KRITERIEN:

1. Datei- und Verzeichnisbrowser, mit dem die Dateien ausgewählt und in andere Verzeichnisse gewechselt werden kann
2. Editor in Form eines eigenen Fensters, in dem das aktuelle Tag eingesehen werden und editiert wird
3. Einfache Bedienung die mit relativ wenigen Funktionstasten auskommt, dabei sollte die Verwendung der Tasten etwa dem entsprechen, was der Nutzer aus diversen anderen Programmen (wie z.B. Midnight Commander) schon kennt

1.3 OPTIONALE KRITERIEN:

1. Möglichkeit, die ID3-Tags schon anzuzeigen, wenn man durch die Verzeichnisse geht
2. Variable Größe der Fenster, die sich an die Terminalgröße anpasst, sobald das Programm gestartet wird
3. Auswahl des Genres des jeweiligen Titels nicht nur anhand des üblichen charakteristischen Zeichens, sondern anhand eines Strings, der beim Abspeichern wieder in das entsprechende Zeichen umgewandelt wird
4. Alphabetische Sortierung der einzelnen in einem Verzeichnis enthaltenen Verzeichnisse und Dateien zur besseren Übersicht
5. Sicherheitsabfrage vor Verlassen des Editors und des Programms (In der Form: Wollen Sie wirklich abspeichern?)

1.4. ABGRENZUNGSKRITERIEN:

Das Programm dient nicht dem Abspielen der mp3-Dateien. Außerdem soll es lediglich die oben genannten Punkte erfüllen.

1.5. COPYRIGHT:

ID3Edit unterliegt der GNU General Public License. Somit ist die Verwendung sowohl für private als auch für kommerzielle Zwecke absolut kostenlos. Der Quelltext darf nach Belieben ergänzt und erweitert und zu Teilen oder komplett in anderen Projekten weiterverwendet werden. Jede Verwendung der Quellen von ID3Edit verpflichtet den Programmierer mittels der GNU GPL dazu, das Resultat ebenfalls unter diese Lizenz zu stellen.

1.7. KNAPPE DEFINITION DER AUFGABE DES PROGRAMMS:

Das Programm soll es dem Anwender ermöglichen, mithilfe eines Verzeichnisbrowsers, Verzeichnisse nach mp3-Dateien zu durchsuchen, deren ID3-Tag mittels eines Editors zu bearbeiten und nach dem Editieren wieder in der mp3-Datei abzuspeichern. Als graphische Oberfläche wurde eine Konstruktion aus Fenstern gewählt, die auf der Basis der Curses- und der PanelBibliothek erstellt und verwaltet werden.

2. Bedienungsanleitung für Nutzer:

2.1. ERSTER START

Das Programm befindet sich auf der Diskette im Verzeichnis ID3Edit. Sobald das Verzeichnis /floppy unter Linux gemountet wurde, kann darauf mit dem Befehl `cd /floppy` zugegriffen werden. Einfacher ist die dabei die Verwendung eine Tools wie dem MidnightCommander. Wer erst seit kurzer Zeit mit Linux arbeitet und vorher reichlich Erfahrung mit dem NortonCommander unter DOS sammeln konnte, wird hier seinen geliebten Dateimanager wiedergefunden haben.

Wenn Sie nach /floppy/ID3Edit wechseln, werden Sie dort nur eine Datei mit Namen ID3Edit.tar.gz finden (regelmäßige Linuxnutzer wissen natürlich, dass es sich hierbei nur um ein gepacktes Verzeichnis handelt, in das man z.B. mit dem MC mühelos wechseln kann.) In dieser Datei befinden sich sowohl die Dokumentation des Programms, eine ausführbare Version des ID3Editors (wie allgemein üblich im Unterverzeichnis bin) sowie sein Quellcode (im Unterverzeichnis src). Dieser kann mit dem Befehl „make“ neu kompiliert werden.

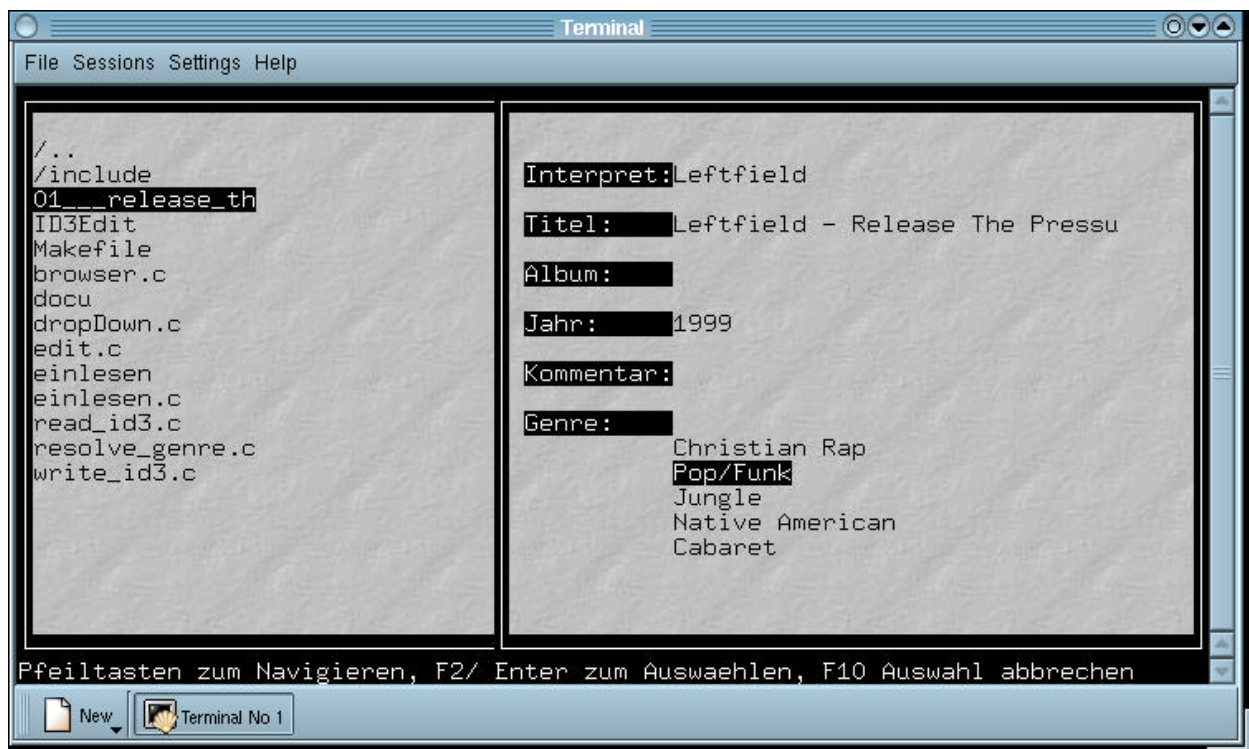
Kopieren Sie sich am besten den gesamten Inhalt des Verzeichnisses in Ihren Homebereich und führen Sie das Programm von dort aus aus. Sie werden schnell feststellen, dass der Filebrowser nach dem ersten Start im aktuellen Verzeichnis steht. Mittels der Cursortasten ist es möglich die Einträge durchzugehen. Wenn die Taste ENTER gedrückt wird, wechselt der Browser entweder das Verzeichnis (kenntlich gemacht durch das Zeichen „/“ vor dem jeweiligen Eintrag) oder aber er öffnet die aktuelle (erkennbar an dem Balken über dem fraglichen Eintrag) mp3-Datei und ermöglicht so ein Editieren des ID3-Tags.

Während des Browsens durch die Verzeichnisse ist es möglich, sich die Tags anzusehen: Jedesmal, wenn der aktuelle Eintrag einer mp3-Datei entspricht, wird deren Tag angezeigt.



Hilfreich bei der Bedienung sollte die Statuszeile sein: Sie gibt Tips, zur Bedienung des Programms und Meldungen, wenn ein Fehler auftrat.

Im Editor selbst kann man wiederum mit Hilfe der Cursortasten durch die einzelnen Bestandteile gehen. Auch ein Bewegen im aktuellen Tag zu einem bestimmten Buchstaben ist möglich - meist liegt ja der Fehler im Detail. Mithilfe der Taste F8 ist ein Löschen des Buchstaben möglich, über dem aktuell der Cursor steht (entspricht der Wirkung der Delete-Taste). F10 bricht das Editieren ab – speichert also KEINE Änderungen des Tags. F2 verlässt



den Editor und speichert die Änderungen.

Mittels der Taste F10 kann auch das Programm selbst verlassen werden.

2.2. AUFLISTUNG DER TASTEN ZUR BEDIENUNG DES ID3EDITORS:

1. Der Browser:

Pfeil nach oben/ unten	Navigieren in den Einträgen
Enter	Ermöglicht das Editieren des aktuellen ID3Tags
F10	Verlassen des Programms

3. Der Editor

Pfeil nach oben/ unten	Wechseln zum jeweils unteren oder oberen Eintrag des Tags
Pfeil nach rechts/ links	Navigieren im Eintrag selbst
Backspace	Löscht das Zeichen, das vor dem aktuellen Cursor steht
F8	Entspricht in ihrem Verhalten Delete und ist als Alternative zu Backspace gedacht
F2	Abspeichern des Eintrags und Verlassen des Editors
F10	Verlassen des Editors ohne Abspeichern des Tags

2. Die Genreauswahlliste

Pfeil nach oben/ unten	Navigieren in den Einträgen
F2/ Enter	Genre übernehmen
F10	Ausgewähltes Genre nicht übernehmen

2.3. BEKANNTE PROBLEME

Meine BackspaceTaste geht nicht.

In diesem Editor wird angenommen, das Backspace mit Ctrl-? oder Ctrl-h eingestellt ist. Sollte dies bei Ihrem Terminal nicht der Fall sein, können Sie alternativ F8 verwenden oder das Terminal mit stty erase ^h einfach umstellen.

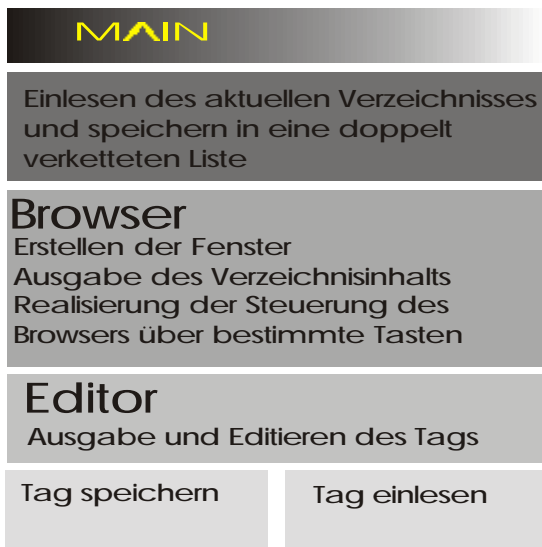
Der Browser flackert, wenn ich durch die Verzeichnisse scrolle

Derzeit wird diese Liste noch aktualisiert, indem das gesamte Fenster neu aufgebaut wird. Es kann also bei langsamen Terminals leicht zum Flackern kommen (z.B. wurde dies beim Eterm bemerkt). Getestet wurde der Editor intensiv auf einem xTerm. Dort sollten kaum Probleme auftreten. Sollte es dennoch zu einem leichten Flackern kommen, versuchen Sie einfach das Terminalfenster zu verkleinern – meist hilft das schon.

3. Quellcodedokumentation:

In dieser Dokumentation soll nicht jede einzelne Anweisung innerhalb des Programms genauer beschrieben werden – dies wurde schon im Quelltext mittels Kommentaren versucht, soweit die jeweilige Anweisung nicht selbsterklärend war. Es soll lediglich eine Übersicht über den Gesamtaufbau des Projekts gegeben werden und auf bestimmte Besonderheiten eingegangen werden.

3.1 ALLGEMEINER PROGRAMMAUFBAU:

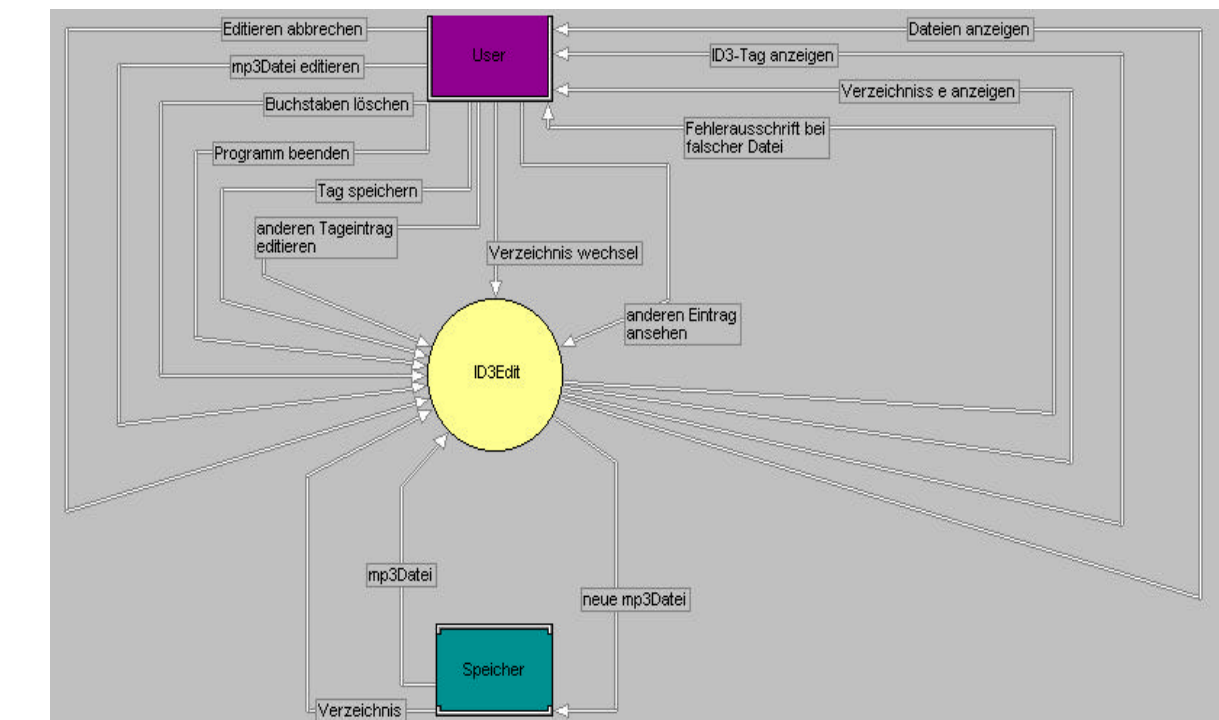


Es wurde versucht, das Programm soweit wie möglich zu modularisieren, so dass ein Verständnis für Außenstehende erleichtert wird und Teilkomponenten weiterverwendet werden können.

Z.B. wurde das Auslesen und Abspeichern der ID3-Tags so konzipiert, dass man beide Funktionen auch komplett getrennt von ID3Edit in anderen Programmen einsetzen kann. Somit wäre ein ID3Tag Editor auf Kommandobasis ohne jegliche graphische Oberfläche der von diesen beiden Modulen Gebrauch macht, denkbar einfach zu realisieren.

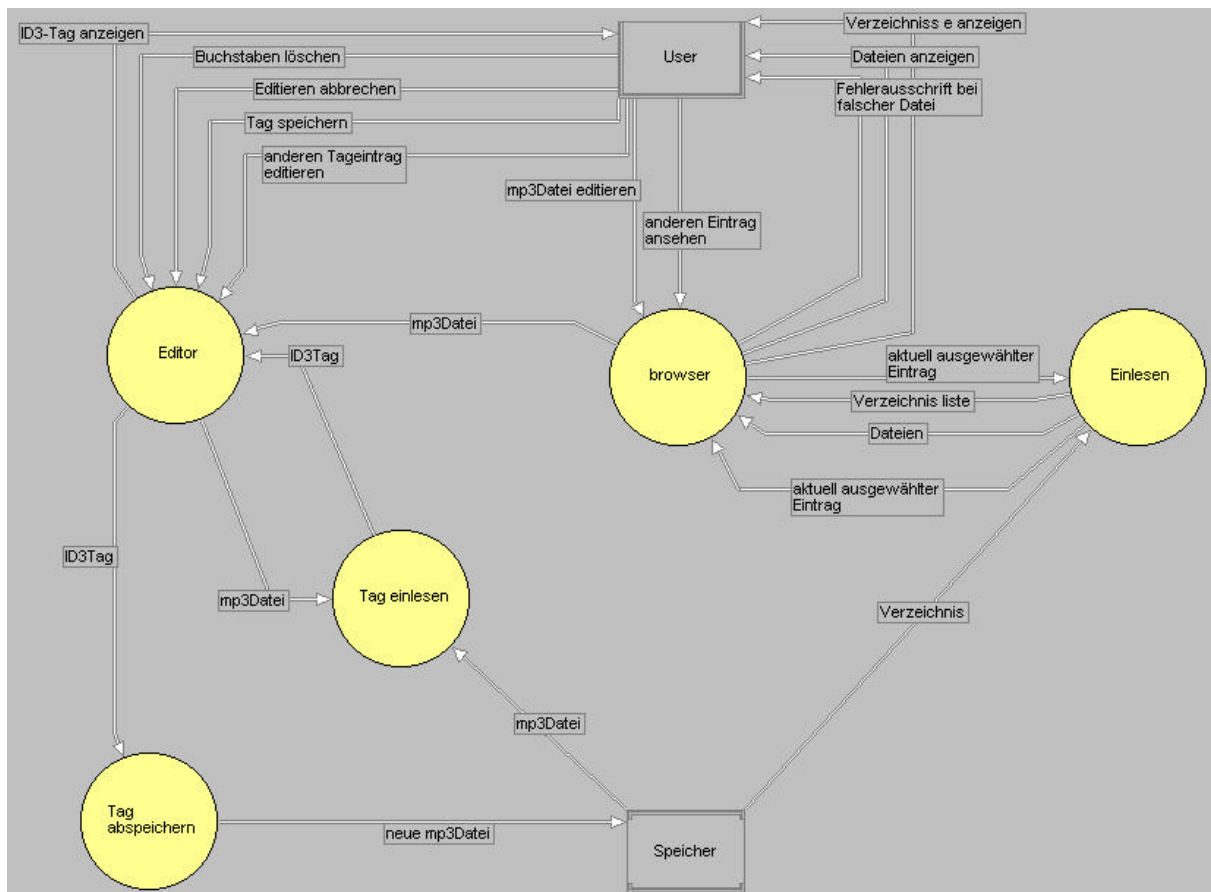
Angestoßen wird der gesamte Programmablauf wie üblich über die Funktion main.

Um für den Leser diesen Sachverhalt zu verdeutlichen, wurden mit dem Tool case4.0 verschiedene Informationsflußdiagramme erstellt und hier eingefügt. Dabei gehe ich hier zuerst von einer Sichtweise „von oben“ auf das Programm aus: Das heißt, es wird lediglich dargestellt, was zwischen Programm, Nutzer und dem Datenspeicher, auf dem die mp3-Dateien abgelegt sind, abläuft.



Nachdem das vorhergehende Bild die allgemeine Funktionalität des Programms verdeutlichen sollte, wird in der folgenden Darstellung eher auf die Modularisierung des ID3TagEditors eingegangen. Dabei wurden auch die Beziehungen zwischen den einzelnen Modulen berücksichtigt.

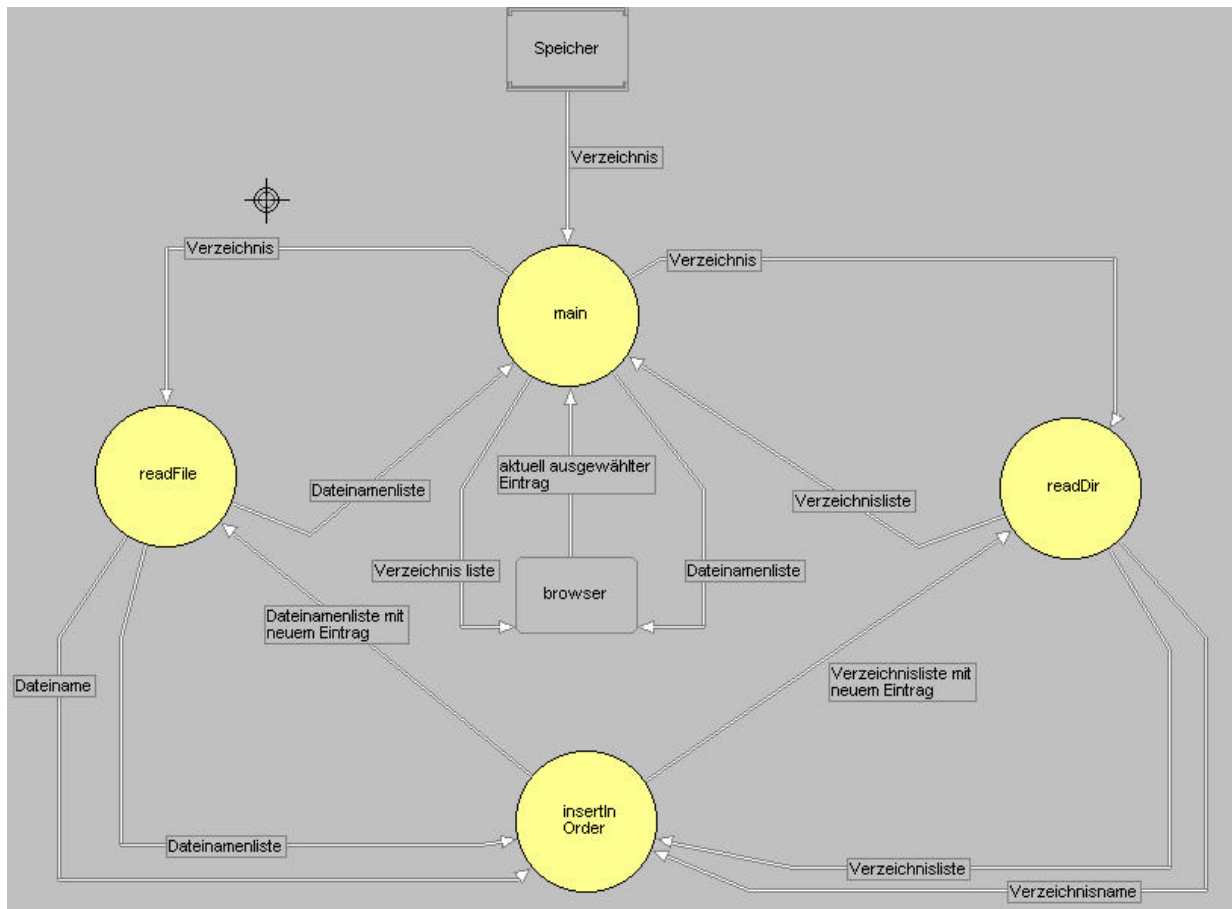
3.2. MODULARISIERUNG DES ID3TAGEDITORS:



Um auch die Arbeitsweise der einzelnen Module leichter verstehen zu können, wurden auch sie mithilfe von Case verfeinert dargestellt. Dabei wurden einige Funktionen, die für das Verständnis nicht notwendig sind, weggelassen. Zum Beispiel wurden Funktionen, die lediglich aus Effektivitätsgründen aus den dargestellten ausgelagert wurden, nicht dargestellt, um das Diagramm nicht unnötig komplex zu gestalten.

Zuerst soll hier auf die Funktion eingegangen werden, die die Verzeichnis- und Dateilisten einliest und sie an den Browser zur graphischen Darstellung weitergibt. Es werden dabei doppelt verkettete Listen verwendet, um so relativ schnell von einem aktuell gewählten Eintrag (Datei oder Verzeichnis im aktuellen Verzeichnis) weiter nach oben oder unten in der Liste wechseln zu können.

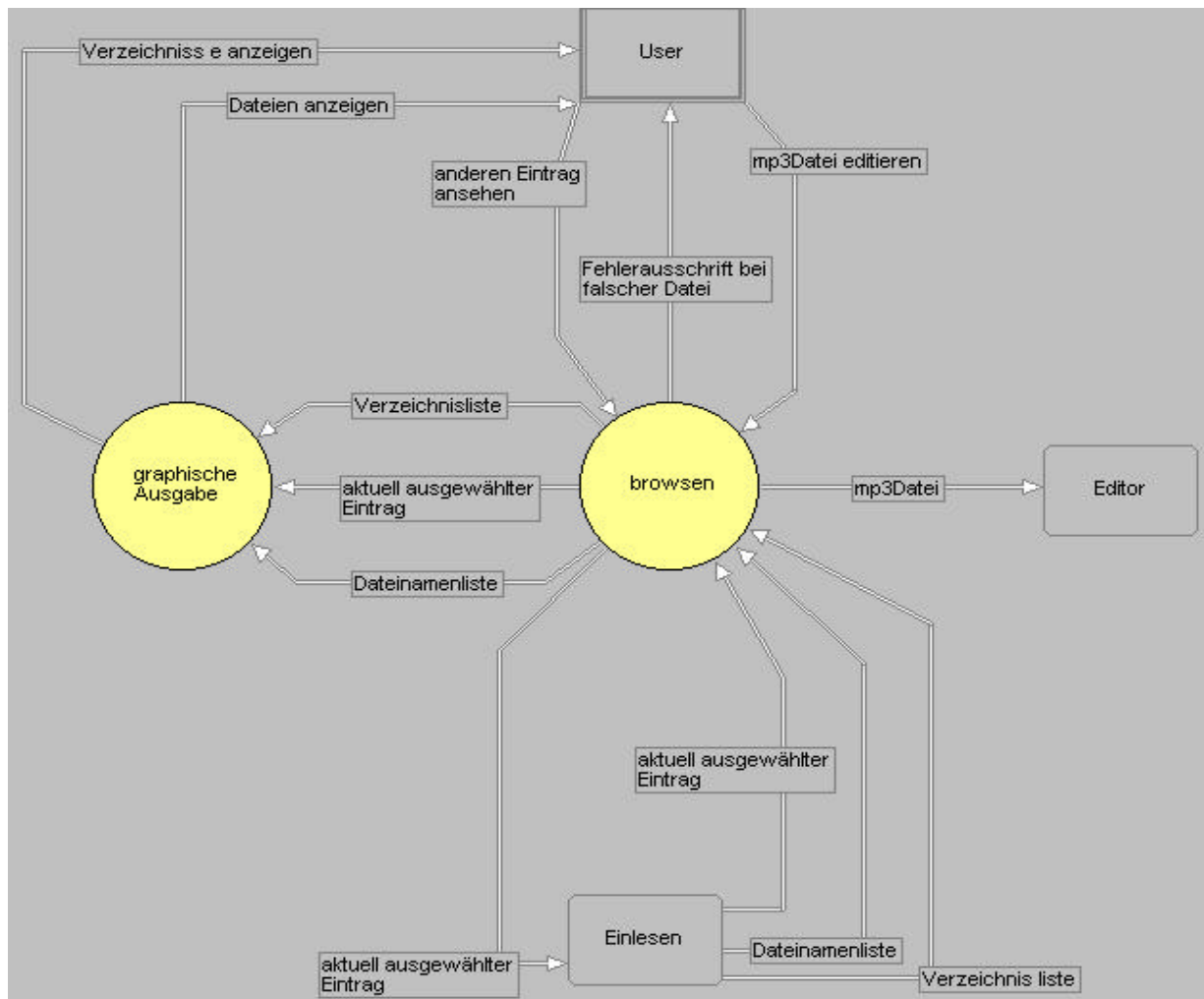
3.3 DAS MODUL „EINLESEN.C“:



Die main-Funktion übernimmt hier die Schlüsselrolle. Sie veranlasst zu Beginn des Programmablaufs, dass das aktuelle Verzeichnis eingelesen wird. Dabei werden, wie schon erwähnt, zwei verkettete Listen erstellt, die jeweils die Verzeichnisse und die Dateien im aktuellen Verzeichnis enthalten. Diese werden mithilfe des Browsermoduls auf dem Terminal dargestellt. Bewegt sich der Nutzer des Programms in der Liste der Verzeichnis- oder Dateieinträge, werden die beiden Listen neu dargestellt – dies wird noch in browser.c veranlasst – und der aktuell gewählte Eintrag wird an die main-Funktion zurückgegeben, die nun entscheiden kann, ob einfach nur auf eine erneute Bewegung des Nutzers im Verzeichnisbrowsers gewartet oder aber das Verzeichnis gewechselt werden muß. Letzteres wird realisiert, indem eine zusätzliche Information mit zurückgegeben wird. Da aber in C nur ein return-Wert erlaubt ist, wird diese Information in Form einer Variablen zurückgeliefert, die wie ein Flag arbeitet: Will der Nutzer das Verzeichnis wechseln, wird es in browser.c auf eins gesetzt, dient der main-Funktion als Indikator dafür, dass das aktuelle Verzeichnis gewechselt und neu eingelesen werden muss und wird dort auch wieder auf Null gesetzt. InsertInOrder ist lediglich eine Funktion, die das Einfügen der einzelnen Namen in alphabetischer Reihenfolge in die Listen realisiert.

Wie vielleicht schon deutlich geworden ist: Das Modul browser.c übernimmt die graphische Darstellung des aktuellen Verzeichnisses. Dazu wurde die Bibliothek ncurses genutzt. Einen groben Überblick über den Ablauf soll wieder das folgende Bild vermitteln.

3.4 DAS MODUL „BROWSER.C“



Was hier in dieser Darstellung fehlt, ist der direkte Aufruf der graphischen Ausgabe des Verzeichnisinhalts von Einlesen aus. Dies geschieht genau zu Beginn des Programms und nach einem Verzeichniswechsel.

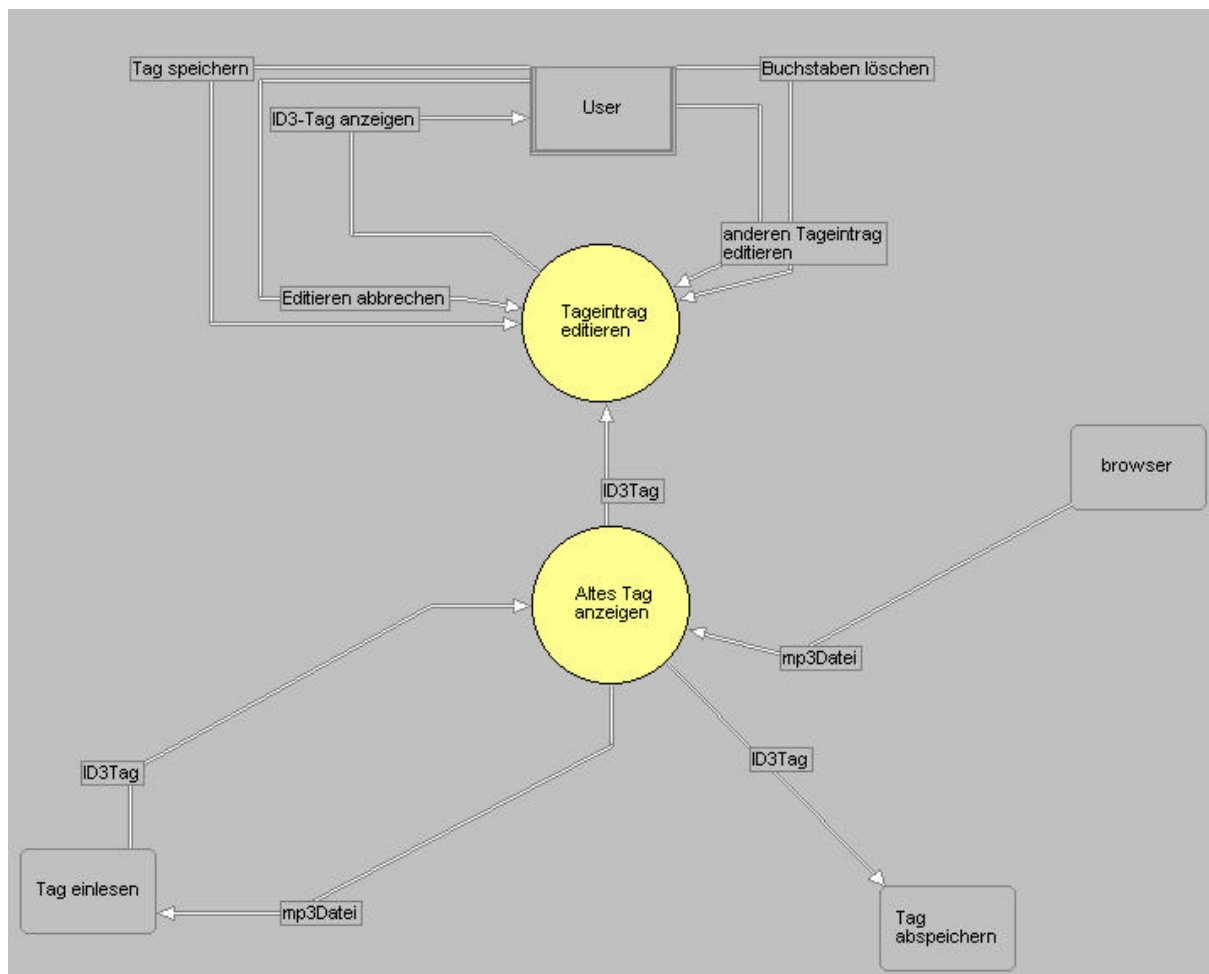
Wie arbeitet dieses Modul nun? In browsen wird auf eine Nutzereingabe gewartet, also z.B. darauf, dass dieser in der Liste einen Eintrag weiter oben auswählen will. Geschieht das, so wird die Funktion „graphische Ausgabe“ aufgerufen. Als Parameter erhält sie die Verzeichnis-, die Dateiliste und das ausgewählte Element. Anschließend kann das jetzt aktuell ausgewählte Element an einlesen.c zurückgegeben werden.

Die graphische Ausgabe wird so realisiert, dass beide Fenster gelöscht werden, und dann die Listen beginnend beim ersten Element ausgegeben werden. Stößt man dabei auf ein Element, das dem ausgewählten entspricht, so werden Vorder- und Hintergrundfarbe vertauscht.

Stellt man nach der letzten ausgebbaren Zeile fest, dass das ausgewählte Element noch nicht dargestellt wurde, so werden beide Fenster gelöscht und mit den folgenden Einträge gefüllt. Diese Vorgehensweise ist vor allem bei Verzeichnissen mit vielen Unterverzeichnissen bzw. Dateien unvorteilhaft und könnte in einer späteren Version so ersetzt werden, dass die Darstellung erst ab einem Element erfolgt, ab dem sicher gestellt werden kann, dass auch das ausgewählte Element noch in eines der beiden Fenster „passt“.

In obiger Darstellung wird auch deutlich, dass das eigentliche Editieren des ID3Tags im Modul editor.c erfolgt. Dessen genauere Aufbau soll im folgenden dargestellt werden

3.5. DAS MODUL „EDIT.C“



Das Editieren erfolgt in einem eigenen Fenster. Dieses wird auch genutzt, um das ID3-Tag anzuzeigen, wenn der Nutzer über eine mp3-Datei „scrollt“. Das heißt, es besteht die Möglichkeit, das Tag anzuzeigen und erst dann zu editieren, wenn „Enter“ gedrückt wurde. Der grobe Ablauf besteht aus dem Initialisieren der einzelnen Fenster und Panels, dem anschließenden Ausgeben der Zeilenbeschriftung („Interpret“, „Titel“ ...), dem Einlesen des ID3Tags in ein eigenes Strukt und dessen anschließende Ausgabe. Wurde Enter gedrückt, kann es auch noch editiert werden. Es wird auffallen, dass eine lange case-Anweisung ausgewählt, in welcher Zeile sich der Nutzer gerade bewegt. Hier wird zunächst ausgewählt, welcher Tageintrag editiert werden soll, dieser wird in einem Extrastring zwischengespeichert der dann unabhängig davon, welchen Tageintrag man editieren möchte, geändert wird. Am Ende der Funktion wird er schließlich in den der aktuellen Zeile entsprechenden Tageintrag zurückkopiert.

Die Auswahl des Genres wurde in ein extra Modul ausgelagert: Sobald der Nutzer die Zeile mit dem Genreeintrag erreicht, erscheint eine Auswahlliste in Form eines kleinen, in die Zeile eingepassten, Extrafensters. Somit wird der Eindruck erweckt, es handele sich um eine echte DropDownliste. Ihr Inhalt wird anhand eines in der Headerdatei `resolve_genre_list.h` definierten Arrays generiert. Dies ist besonders deshalb von Vorteil, als die Länge des Arrays aufgrund der allgemein üblichen Genre begrenzt ist und der Index jeden Eintrags gleichzeitig genutzt werden kann, um das für den Genreeintrag entsprechend charakteristische Byte zu speichern. Ansonsten erfolgt der Ablauf der Auswahl in etwa ähnlich dem Ablauf der Auswahl eines Eintrags in der Verzeichnis-/Dateiliste.

Ich gehe auch an dieser Stelle lediglich grob auf den genauen Ablauf ein und verweise den interessierten Leser nur auf den Quellcode. Dieser sollte ausreichend kommentiert sein, um ihn auch ohne größere Erklärungen zu verstehen.

4. Quellenangabe

4.1. VERWENDETE LITERATUR

- Die man-pages
- www-info.mpi.htwm.de/~geiler/masch-c

4.2. VERWENDETE SOFTWARE:

mc – zum Erstellen des Quelltextes und zur hexadezimalen Darstellung der mp3Dateien, um mehr über den Aufbau des ID3Tags zu erfahren

gcc Version egcs-2.91.66 19990314/Linux (egcs-1.1.2 release) – zum Compilieren und Linken des Quellcodes

Word97/ Adobe Acrobat 4.0 – zur Erstellung der Dokumentation

CorelDraw8, CorelPhoto-Paint8, Corel Capture, Case4.0 – zum Erstellen der graphischen Darstellung des Programmablaufs

5. Erklärung

Hiermit versichere ich, den Beleg selbstständig und nur unter Zuhilfenahme der oben genannten Quellen erstellt zu haben.