

Flib, das Fontlibrary-Programm für den AMIGA

Version 1.03

**vom
25. August 1991**

**von
Georg Heßmann**

Abstract

Flib ist ein Programm zum Erstellen und Verwalten von Fontlibraries. Dies sind Libraries, in denen viele PK-Fonts zusammengefaßt werden. Der Vorteil von Libraries gegenüber einzelnen Files ist die Platzeinsparung auf Diskette oder Festplatte, die bessere Übersicht und der schnellere Zugriff auf einzelne Elemente. Diese Libraries werden von den T_EX-Treibern ShowDVI und DVIPrint verwendet.

Inhaltsverzeichnis:

1. Copyright und ähnliches	2
2. Allgemeines über Flib	2
3. Der Aufruf	3
4. Der Aufbau der Libraries	3
5. Portierbarkeit	4

1. Copyright und ähnliches

Das Programm Flib ist “copyrighted Public Domain”. Das heißt jeder kann das Programm verbreiten und verändern unter Beachtung einiger Einschränkungen. Zum einen behält das Programm mein (Georg Heßmann) Copyright. Das Copyright darf nicht aus dem Programmtext gelöscht werden, und die Copyright-Meldung muß auch weiterhin wenigstens per Option angezeigt werden können. Alle Änderungen, die am Programm gemacht werden, müssen mir mitgeteilt werden, damit ich die Versionen nach Möglichkeit angleichen kann. Diese Copyright-Bestimmungen steht auch am Anfang des Files “flib.c” und dürfen auch nicht von dort entfernt werden. Auch müssen die Files flib.c, flib.tex sowie das dazugehörige README-File immer zusammen vertrieben werden. Dafür kann jeder das Programm verwenden wie er möchte und kommerziell oder nicht-kommerziell vertreiben¹.

Für weitere Erklärungen siehe das README-File.

2. Allgemeines über Flib

Flib ist ein Hilfsprogramm für die T_EX-Treiber ShowDVI und DVIPrint. Es ist dazu da, die PK-Fonts einer Auflösungsstufe in einem File zusammenzufassen, da dies einige überzeugende Vorteile hat.

Als erstes wäre da die Speicherersparnis zu nennen. Da jedes einzelne File immer aus einem Vielfachen der Blockgröße des jeweiligen FileSystems besteht, wird auch bei jedem File durchschnittlich ein halber Block verschwendet. Bei manchen Betriebssystemen wird auch noch für jeden Directory-Eintrag ein ganzer Block benötigt. Das heißt, pro File ist grob ein Verschnitt von eineinhalb Blöcken zu erwarten. Bei etwa 100 Modulen pro Library werden so knapp 150 Blöcke gespart.

Aber es gibt noch andere Vorteile. Da wären noch die bessere Übersicht und auch der schnellere Zugriff zu nennen. Da man meistens mehrere Fonts einer Auflösungsstufe benötigt, braucht man nur ein File zu öffnen und dort das Directory einzulesen. Danach ist jeder Zugriff auf einen Font nur noch ein “fseek”-Kommando. Da so nur für jede Auflösung ein File zu öffnen ist, kann man diese den ganzen Programmablauf offen halten. So vermeidet man auch ein ständiges Öffnen und Schließen von Files.

¹ Dies gilt nur für das Flib Programm und *nicht* für die anderen Programme

Das Flib-Programm ist aber nicht nur auf PK-Fonts spezialisiert. Im Prinzip kann man damit jeden beliebigen Typ von Files in einer Library archivieren. Da der Programmtext zum Flib-Programm verfügbar ist, könnte man auch andere Programme, die viele Hilfs-Files benötigen, daran anpassen.

3. Der Aufruf

Der Syntax des Aufrufes ist folgendermaßen:

```
usage: flib c|a|x|d|z|p|l|t|v] library [-|[modules]*]
```

- c,a hängt ein Modul an eine Library an. Existiert die Library noch nicht, so wird sie neu erzeugt. Falls das Modul schon in der Library ist, so wird es ersetzt.
- x kopiert das Modul aus der Library in ein eigenes File mit dem Namen des Moduls. Wird kein Modul angegeben, so werden alle Module aus der Library kopiert.
- d löscht ein Modul aus einer Library. Falls das Modul das letzte der Library war, so wird die Library gelöscht.
- t gibt die Liste aller Module der Library aus.
- r benennt alle angegebenen Module um. Dazu wird der neue Name von der Console (stdin) erwartet.
- z transformiert Fontlibraries des alten Typs in die aktuelle Version. Alle neu angelegten Libraries sind automatisch vom neuen Typ.
- p testet alle Module, ob sie noch korrekt sind. Dazu wird beim Einfügen in die Library eine Testsumme erzeugt und im Directory der Library gespeichert.
- l erzeugt einen Library-Link. Der Aufruf sieht hier folgendermaßen aus:

```
flib l link-name dest-name
```

Dabei wird eine Library *link-name* erzeugt, die lediglich ein Verweis auf eine Library mit dem Namen *dest-name* ist. Dies verwendet man zum Beispiel in dem Fall, wo man Fontlibraries der Auflösungen 109 dpi und 110 dpi benötigt, aus Speicherplatzgründen aber nur eine von ihnen auf Platte halten möchte. Da der Unterschied der beiden Fontlibraries nicht so groß ist, kann man sich dafür entscheiden, lediglich eine von beiden auf Platte zu halten und die andere nur als Link zu erzeugen.

- v schaltet den Verbose-Modus ein. Dieses Flag kann zu jedem der obigen Optionen eingeschaltet werden. Normalerweise werden nur Fehlermeldungen ausgegeben. Mit der v Option wird jede Aktion mitprotokolliert. Dies ist vor allem bei komplizierteren Aufgaben sinnvoll.

Bei allen Kommandos können hinter dem Library-Namen beliebig viele Module angegeben werden. Es gibt aber auch die Möglichkeit nur ein Minus (–) anzugeben. Dann werden die Modulnamen von der Console (stdin) eingelesen (ein Modul pro Zeile), bis ein File-Endezeichen kommt.

4. Der Aufbau der Libraries

Bei den Libraries müssen zwei verschiedene Typen unterschieden werden: Einmal

die Library, in der tatsächlich Module gespeichert sind, und zum anderen die Libraries, die lediglich ein Link auf eine andere Library sind.

4.1. Richtige Libraries

Die Fontlibraries bestehen aus drei Teilen. Am Anfang der Library steht der Library-Kopf, dann kommt das Library-Directory und am Schluß die einzelnen Module.

Der Library-Kopf besteht aus zwei 32-Bit-Worten (Langworten). Das erste ist die "magic number", die aus den vier Buchstaben "FLIB" besteht. Danach kommt die Directory-Größe. Sie muß nicht unbedingt gleich der Anzahl der Module der Library sein. Es können auch Directory-Einträge als unbelegt gekennzeichnet werden.

Das Directory der Library besteht aus den einzelnen Directory-Einträgen. Die Anzahl wurde im zweiten Langwort festgelegt. Ein Directory-Eintrag sieht folgendermaßen aus:

```
struct new_flib_dirent {
    char  mname[22];          /* module name */
    unsigned short checksum; /* checksum */
    long  size;              /* size in bytes */
    long  where;            /* position */
};
```

Wobei `mname` der Modulname, `checksum` die Prüfsumme, `size` die Größe des Moduls und `where` die Position des Moduls innerhalb der Library ist. Wenn `where` gleich Null ist, so ist der Directory-Eintrag unbelegt.

Bei all diese Zahlangaben ist die Byte-Reihenfolge zu beachten. Verwendet wird das Format der 680x0 Prozessoren. Also zuerst immer das "low significant byte" und dann das "high significant byte". Die Ausgaberroutinen des Flib-Programms sind so geschrieben, daß sie auch auf anderen Prozessoren das geforderte Format erzeugen und lesen.

Der dritte und letzte (größte) Teil der Library sind die einzelnen Module, die einfach hintereinandergelegt sind.

4.2. Library-Links

Library-Links haben einen sehr einfachen Aufbau, der sich im Prinzip auch mit einem Texteditor erzeugen läßt. Zum Beispiel:

```
"FLNK<PK0110>"
```

Dies ist ein Library-Link auf eine Library mit dem Namen *PK0110*.

Das Magic-Word ist hier also FLNK. Danach kommt lediglich der Name der Library, auf die der Link zeigen soll, eingerahmt in ein Kleiner- und ein Größerzeichen.

5. Portierbarkeit

Das Programm ist in ANSI-C geschrieben, läßt sich aber auch durch Änderung eines

Defines mit einem normalen C-Compiler übersetzen. Übersetzt und getestet wurde das Programm auf AMIGA, ATARI, Sun 3, Sun SPARC und einer DECstation unter den Betriebssystemen AMIGA DOS, ATARI TOS und UNIX (SunOS/ULTRIX).