

Kiskometer

Angela Schmidt

Copyright © Copyright1995-1996 by Angela Schmidt, Neu-Ulm, Germany

COLLABORATORS

	<i>TITLE :</i> Kiskometer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Angela Schmidt	January 17, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Kiskometer	1
1.1 Stichwortverzeichnis	1
1.2 Start-Hilfe	1
1.3 Kiskometer - Inhaltsverzeichnis	2
1.4 Aufruf	2
1.5 Mögliche Parameter von Kiskometer	3
1.6 Offset-Dateien	6
1.7 Copyright	7
1.8 Haftungsausschluß	7
1.9 Autorin	7
1.10 History	8
1.11 Dankeschön	8

Chapter 1

Kiskometer

1.1 Stichwortverzeichnis

Aufruf	Wie man das Programm aufruft
Autorin	Wie man die Autorin erreicht
Copyright	Nur nicht-gewerbliche Verbreitung erlaubt
Danksagungen	Leute, die ein Dankeschön verdienen
Haftungsausschluß	Es wird für nichts gehaftet
History	Änderungen seit Version 1.0
Inhaltsverzeichnis	Zurück zum Inhaltsverzeichnis
Offset-Dateien	Wie man eigene Offset-Dateien erzeugt
Parameter	Parameter, die von Kiskometer verstanden werden
Start-Hilfe	Start durch einen einfachen Mausklick

1.2 Start-Hilfe

Um Ihnen den Umgang mit Kiskometer etwas zu erleichtern, sind hier ein paar Befehle vorinstalliert. Sie müssen nur draufklicken, und schon werden die entsprechenden Aktionen vorgenommen.

Kiskometer starten Kiskometer mit Auto-Popup-Fenster starten
Kiskometer beenden
Kiskometer-Ausgaben auf serielle Schnittstelle umlenken

1.3 Kiskometer - Inhaltsverzeichnis

Kiskometer ist ein Programm, das überprüft, welche Programme die
Funktion ↔
SetFunction() aufrufen und das anzeigt, welche Funktionen gepatcht
wurden. Dabei kann das Anzeigeformat weitgehend selbst bestimmt werden.
Mit Hilfe eines umfangreichen Filters kann man bestimmte Aufrufe von
SetFunction() von der Anzeige ausschließen. Die Anzeige geschieht
wahlweise auf stdout, in eine Datei oder auf die serielle Schnittstelle.

Copyright
Nur nicht-gewerbliche Verbreitung erlaubt

Haftungsausschluß
Es wird für nichts gehaftet

Autorin
Wie man die Autorin erreicht

Aufruf
Wie man das Programm aufruft

Start-Hilfe
Start durch einen einfachen Mausklick

Offset-Dateien
Wie man eigene Offset-Dateien erzeugt

History
Änderungen seit Version 1.0

Danksagungen
Leute, die ein Dankeschön verdienen

1.4 Aufruf

Das Programm kann nur vom CLI aus gestartet werden. Die ↔
einfachste Art,
es zu starten, ist durch Eingabe des Programmnamens ohne
Übergabeparameter. Man sollte es aber ungedingt mit »Run« starten, sonst
ist die Shell danach bis zum nächsten ^C unbrauchbar.

```
1> Run Kiskometer
```

Es erscheint eine Copyright-Meldung. Danach tut sich scheinbar nichts
mehr, aber wann immer eine Funktion gepatcht wird, wird nun eine

entsprechende Meldung in dem CLI-Fenster, in dem Kiskometer gestartet wurde, ausgegeben.

Möchten Sie das mal ausprobieren? Dann starten Sie beispielsweise Kai 'kiske' Iskes »MagicCX« oder sein »ForceIcon« – und schon sollte sich etwas in jenem CLI-Fenster rühren... Ahnen Sie nun vielleicht, woher der Name »Kiskometer« kommt...? :-)

Wer mehr Steuerungsmöglichkeiten möchte, kann
Übergabeparameter
mit
übergeben.

Das Programm kann durch ^C beendet werden – außer wenn andere Programme, die nach Kiskometer gestartet wurden, ebenfalls SetFunction() gepatcht haben und noch nicht wieder beendet wurden.

Wollen Sie es mal schnell testen? Dann schauen Sie sich
diese
Seite an!

1.5 Mögliche Parameter von Kiskometer

Syntax: Kiskometer [<options>] [<outputfile>] [LIBOFFSDIR <dir>]
oder: Kiskometer QUIT | EXIT
oder: Kiskometer HELP | ? | -?

Man kann Kiskometer stets einen Dateinamen übergeben. In diese Datei werden dann die Ausgaben von Kiskometer (außer Copyright-Meldung, Fehlermeldungen etc.) geschrieben. Wird keine Datei angegeben, erfolgt die Ausgabe auf stdout, falls nicht die Option RAWIO gesetzt ist (siehe unten).

Falls bereits ein Kiskometer aktiv ist, werden beim erneuten Starten von Kiskometer nur die neuen Parameter an das alte Kiskometer übergeben. Danach kehrt das neue Kiskometer sofort zurück. Lediglich LIBOFFSDIR kann nicht nachträglich geändert werden.

Kiskometer läßt sich entweder durch ^C beenden, oder durch erneutes Starten mit den Übergabeparametern QUIT oder EXIT.

Der Übergabeparameter »HELP« erzeugt eine ausführliche Hilfe-Seite für Kiskometer.

Folgende Optionen können Kiskometer übergeben werden. Wenn eine Zahl erwartet wird, so kann diese dezimal, hexadezimal oder oktal übergeben werden. Der Anfang der Zahl entscheidet, wie diese interpretiert wird:

1-9	=>	dezimal
0x oder 0X	=>	hexadezimal
0	=>	oktal

LIBOFFSDIR <dir>

Hier kann man – falls nötig – das Verzeichnis angeben, in dem sich die

Library-Offset-Dateien befinden. Default ist »PROGDIR:LibOffsets«. Diese Option kann nur direkt beim Programmstart angegeben werden.

[IN]ACTIVE

»ACTIVE« aktiviert Kiskometer wieder, nachdem es zuvor mit »INACTIVE« in den nicht-aktiven Zustand versetzt wurde. Im nicht-aktiven Zustand bleibt Kiskometer zwar komplett im Speicher (es können also weiterhin Anfragen mit »STATUS« gestellt werden, die Library-Offsets bleiben im Speicher etc.), aber der Library-Vektor »SetFunction« wird wieder komplett freigegeben. Man hat danach also wieder ein »sauberes« System. Falls der Patch nicht entfernt werden kann, weil inzwischen schon ein anderes Programm SetFunction() gepatcht hat, wird eine Fehlermeldung ausgegeben.

QUIET

Gibt die Copyright-Notiz etc. nicht aus.

VERYQUIET

Unterdrückt nicht nur die Copyright-Notiz, sondern sogar alle Fehlermeldungen. Das bedeutet, daß nur noch die reinen Patch-Meldungen ausgegeben werden.

[NO]RAWIO

RAWIOQUICK

STDIO

Die Option RAWIO bewirkt, daß die Patch-Meldungen mit KPrintf() auf der seriellen Schnittstelle ausgegeben werden. Mit NORAWIO kann man die Meldungen wieder auf stdout ausgeben. Fehlermeldungen werden nicht umgeleitet - nur die Patch-Meldungen!

STDIO bewirkt exakt dasselbe wie NORAWIO.

Wenn man die Option RAWIOQUICK statt RAWIO angibt, so wird die Patchmeldung schon während des SetFunction()-Aufrufs ausgegeben. Aber Vorsicht - seit OS 2.1 benötigt das sehr viel Stack und kann so das patchende Programm zum Absturz bringen!

[NO]SHAREOUTFILE

Wenn die Ausgabe in eine Datei gehen soll, so wird diese Datei normalerweise exklusiv geöffnet und bleibt auch bis zum Schließen der Datei in diesem Zustand. Wenn SHAREOUTFILE gesetzt ist und mindestens OS 2.04 vorhanden ist, wird das Filehandle nur während des Schreibvorgangs exklusiv geöffnet. Das heißt, daß die Datei fast immer zu Lesezwecken geöffnet werden kann. Allerdings kann es dann passieren, daß der eine oder andere Patch verlorenght, weil der Filemode im entscheidenden Moment nicht auf den Exklusiv-Modus umgeschaltet werden kann.

BEMERKUNG: einige Dateisysteme sind buggy (z. B. die Ram-Disk von OS 3.1), so daß es mit dieser Option Probleme geben kann. Die OS 3.1 Ram-Disk läßt beispielsweise auch nach NOSHAREOUTFILE noch Lesezugriffe anderer Prozesse zu, wenn zuvor SHAREOUTFILE gesetzt war, oder wenn die angegebene Datei bereits existierte.

[NO]SLEEP

Diese Option setzt Kiskometer in einen schlafenden Zustand bzw. holt es wieder aus dem schlafenden Zustand zurück. Während Kiskometer schläft,

geschehen keine Ausgaben. Sowie Kiskometer jedoch wieder erwacht, werden all diese Ausgaben nachgeholt.

FORMAT <string>

Hiermit kann man die Ausgabe formatieren. Es stehen folgende Formatier-Kürzel zur Verfügung:

- %o - Library-Offset als negative Dezimalzahl
- %O - Library-Offset als Funktionsname
- %l - Librarybase als hexadezimale Adresse
- %L - Libraryname
- %t - Taskadresse als hexadezimale Adresse
- %T - Taskname oder Name des momentan geladenen CLI-Programmes
- %c - Anzahl aller SetFunction-Aufrufe, die angezeigt wurden
- %C - Anzahl aller SetFunction-Aufrufe, also auch die Funktionen, die mit der Option FILTER unterdrückt wurden.
- %a - Adresse der in Zukunft anzuspringenden Funktion
- %A - Adresse der ursprünglich angesprungenen Funktion
- \n - Zeilenvorschub
- \t - Tabulator-Zeichen

Außerdem gibt es noch ein sehr spezielles Kürzel:

%(*<library-pattern>*)(*<library-offset>*) - Dieses Kürzel sucht nach allen Libraries, die auf *<library-pattern>* passen. Von der ersten Library, die paßt, wird der zu *<library-offset>* gehörende Funktionsname ermittelt und dann ausgegeben. Dieses Format-Kürzel wird erst bei der unten beschriebenen Funktion »STATUS« richtig sinnvoll.

Man kann durch Einschleichen von wohldefinierten Steuerzeichen direkt hinter das Prozent-Zeichen die Ausgabe auch schön formatieren. Hierzu werden die auch in C üblichen Formatieranweisungen verwendet. Zusätzlich dazu kann man »,<z>« anhängen, das die Ausgabe auf <z> Zeichen beschränkt und ggf. drei Punkte an den Schluß setzt um anzudeuten, daß etwas weggeschnitten werden mußte.

Voreingestellt ist folgende Format-Zeile:

```
"(%c) Offset %4o=%-20,200 of %-20,20L by %T\n"
```

Wer ein breites Ausgabefenster hat, dem sei folgende Zeile empfohlen:

```
"(%c) Offset %4o=%-20,200 of %-20,20L (%A->%a) by %T\n"
```

Hierdurch wird zusätzlich angezeigt, was die alte Funktionsadresse war und wodurch sie ersetzt wurde.

STATUS <formatstring>

Diese Option gibt <formatstring> direkt auf stdout aus. Es stehen dieselben Formatier-Kürzel wie unter »Format« beschrieben zur Verfügung, bis auf diejenigen, die nur Sinn ergeben, wenn ein Programm gerade SetFunction aufruft, also bis auf %o, %O, %l, %L, %t, %T und %a. Wer ein geschicktes Alias baut, erhält hierdurch eine nette, kleine, schnelle Funktion, die den zu einem Offset gehörenden Funktionsnamen ermittelt.

FILTER <filter>

Oft möchte man vielleicht nicht alle Patches ausgeben lassen sondern nur ganz bestimmte. Dazu sind die folgenden umfangreichen Filter-Funktionen nützlich:

- (AND arg1 arg2 ...) - Und-Verknüpfung aller Übergabe-Parameter.
- (OR arg1 arg2 ...) - Oder-Verknüpfung aller Übergabe-Parameters.
- (NOT arg) - Negierung des Übergabe-Parameters.
- (LIBNAME <pattern>) - Dieser Filter gibt TRUE zurück, wenn der Name der zu patchenden Library auf <pattern> paßt, sonst FALSE.
- (LIBBASE <address>) - Dieser Filter gibt TRUE zurück, wenn die Library-Base der zu patchenden Library die Adresse <address> hat, sonst FALSE.
- (TASKNAME <pattern>) - Dieser Filter gibt TRUE zurück, wenn der Name des patchenden Tasks auf <pattern> paßt, sonst FALSE. Im Falle eines CLI-Prozesses wird der Name des momentan geladenen Programmes überprüft.
- (TASKADDR <address>) - Dieser Filter gibt TRUE zurück, wenn die Task-Adresse des patchenden Tasks die Adresse <address> hat, sonst FALSE.
- (FUNCNAME <pattern>) - Dieser Filter gibt TRUE zurück, wenn der Name der zu patchenden Funktion auf <pattern> paßt, sonst FALSE.
- (OFFSET <offset>) - Dieser Filter gibt TRUE zurück, wenn der zu patchende Offset den Wert <offset> hat, sonst FALSE.
- (<string>) - Nützlich, um Leerzeichen in <string> zu schützen.
Beispiel: "(TASKNAME (Services Manager))"

Beispiel:

```
1> Kiskometer FILTER "(AND (LIBNAME dos.library) (FUNCNAME Open))"
```

Es ist beliebige Verschachtelung möglich.

1.6 Offset-Dateien

Im Lieferumfang dieses Programmpaketes ist ein ARexx-Script enthalten, das dazu dient, FD-Dateien in ein Format zu konvertieren, welches von Kiskometer erkannt wird. Mit Hilfe dieser Dateien können die Funktionsnamen, die gerade gepatcht werden, angezeigt werden.

Aufruf:

makelibnames.rexx <source> <dest> <libname>

<source>: Name der FD-Datei, die konvertiert werden soll.

<dest>: Name der Ziel-Datei, die für Kiskometer erstellt werden soll.

<libname>: Name der Library, z.B. »exec.library«. Es ist unbedingt darauf zu achten, daß die Groß-/Kleinschreibung exakt eingehalten wird!

1.7 Copyright

Kiskometer unterliegt dem Copyright © 1995-1996 von Angela Schmidt. ←

Dieses Softwarepaket darf zu nicht-gewerblichen Zwecken frei benutzt werden. Das Vervielfältigen und Verbreiten dieses Softwarepakets ist ebenfalls erlaubt, solange es nicht-gewerblich geschieht und die Software unverändert bleibt. Das Verlangen einer sogenannten »Kopiergebühr« ist nicht gestattet, die Weitergabe dieses Softwarepakets hat also völlig kostenlos zu geschehen. Ausnahmen bedürfen einer schriftlichen Genehmigung der

Autorin

.

Hiermit wird die Genehmigung erteilt, dieses Softwarepaket auf der »Meeting Pearls« CD-ROM-Serie und auf der »Amiga Developer CD V1.0 April 1996« zu verbreiten.

1.8 Haftungsausschluß

Weder die

Autorin

noch Dritte übernehmen die Haftung für Schäden, die direkt oder indirekt durch Benutzung dieser Software entstehen.

1.9 Autorin

Meine Adressen sind:

Angela Schmidt
Finkenweg 26
89233 Neu-Ulm

Alles, was hier hingeschickt wird, sollte mich irgendwann mal erreichen. Es kann zwar ein wenig dauern, aber solange die Post nichts verbaselt, wird es irgendwann bei mir ankommen.

Angela Schmidt
Klosterweg 28/I501
76131 Karlsruhe

Diese Adresse wird wahrscheinlich irgendwann 1997 ungültig werden. Ob sich dann jemand erbarmt, mir die Post hinterherzuschicken, ist sehr fragwürdig. Da dies meine Studienadresse ist, erreicht mich Post an diese Adresse momentan jedoch deutlich schneller! Momentan wird diese Adresse bevorzugt!

Telefon: 0721/695307 Dies ist keine Hotline! Es wird keinerlei Support geleistet! Wer mir jedoch etwas Wichtiges mitteilen möchte, der darf diese Nummer zu menschlichen Zeiten wählen.

E-Mail: Angela.Schmidt@stud.uni-karlsruhe.de

Bitte beachten Sie, daß ich aus Zeitgründen weder alle Briefe noch alle E-Mails beantworten kann!

Danke für Ihr Verständnis!

1.10 History

Version 1.0 (10-Jun-95)

- Erste Version

Version 1.1 (3-Oct-95)

- Option RAWIO ruft nun nicht mehr innerhalb von SetFunction() KPrintf() auf, da RawDoFmt() ab OS 2.1 maßlos Stack frißt und es bei einigen Programmen zu Systemabstürzen kam.
- Die Option RAWIOQUICK wurde eingeführt. Für Debugging-Zwecke kann die Ausgabe auf die serielle Schnittstelle weiterhin direkt innerhalb von SetFunction() getätigt werden.
- Die Option NORAWIO leitet die Ausgabe nun immer auf stdio um.
- Die Option STDIO kam hinzu. Sie arbeitet wie NORAWIO.
- Die Format-Zeile hat einen neuen Parameter kennengelernt: %A zeigt die alte Funktionsadresse an.
- Ein Enforcerhit wurde entfernt.

Version 1.2 (5-Apr-96)

- Die Optionen SHAREOUTFILE und NOSHAREOUTFILE wurden hinzugefügt.

Version 1.3 (14-Okt-96)

- Wenn SetFunction() als Offset nur ein WORD übergeben wurde, hatte Kiskometer Probleme. Dieser Fehler sollte nun behoben sein. Danke an Ben Hutchings für den Bugreport - leider fand ich kein Programm, um das selber testen zu können. ;-)

1.11 Dankeschön

Ein Dankeschön geht an:

Bernhard Möllemann

Für sehr gutes Betatesting und sehr gute Bugreports!

Klaus Melchior

Für Bugreports.

Christoph Dietz

Für Bugreports und Erweiterungs-Vorschläge.

Ralph Babel

Für den Hinweis, daß `RawDoFmt()` ab OS 2.1 jede Menge Stack frißt.

Kai Iske

Für seinen Namen. :-)
