

WarpRace

Sam Jordan

COLLABORATORS

	<i>TITLE :</i> WarpRace		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Sam Jordan	July 19, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	WarpRace	1
1.1	WarpRace	1
1.2	Einleitung	1
1.3	Installation	1
1.4	Benützung	2
1.5	Ausgabeformat	3
1.6	Statistik	3
1.7	Entwickler-Infos	4

Chapter 1

WarpRace

1.1 WarpRace

WarpRace

1997 von Sam Jordan
© HAAGE & PARTNER Computer GmbH

Das modulare Performance-Messungs-Programm

Einleitung
Installation
Benützung
Ausgabeformat
Statistik
Entwickler-Infos

1.2 Einleitung

WarpRace ist ein modulares Programm zur Durchführung von Performance-Messungen. Es unterstützt sowohl 68K- wie auch PPC-Prozessoren und ist somit ideal dazu geeignet, die Performance der verschiedenen Prozessoren zu vergleichen. WarpRace eignet sich auch zum Messen von Speicherdurchsätzen oder von Overhead-Messungen bei Kontextwechseln.

Das modulare Konzept von WarpRace erlaubt es jedem Programmierer, eigene Module zu entwickeln, welche dann von WarpRace ausgeführt werden. In diesem Dokument wird ausführlich erklärt, wie neue Module entwickelt werden können.

WarpRace benötigt mindestens einen 68020-Prozessor und OS2.0. Für das Ausführen von PPC-Modulen wird ein beliebiger PPC-Prozessor sowie eine beliebige Version der powerpc.library benötigt.

1.3 Installation

Die Installation von WarpRace ist sehr einfach. Es muss nur das Verzeichnis 'WarpRace' an einen beliebigen Ort kopiert werden. Auf das Kopieren kann auch verzichtet werden, falls sich WarpRace aber auf einer CD befindet oder auf einem anderen schreibgeschützten Datenträger ist es empfehlenswert, das Verzeichnis zu kopieren, damit neue Module hinzugefügt werden können.

1.4 Benützung

WarpRace ist ein Programm, welches in einem bestimmten Verzeichnis Module sucht und diese ausführt. Module sind gewöhnliche ausführbare Programme, welche aber spezielle Eingabeparameter erwarten und spezielle Ausgabeparameter liefern. Jedes Modul hat die Endung '.worm' (WarpRace-Module). Die Module müssen sich alle im Verzeichnis 'Modules' befinden. Es können allerdings Unterverzeichnisse erstellt werden, um Module zu gruppieren oder um eine Testserie zusammenzustellen.

WarpRace wird von der CLI aus bedient. Mittels 'WarpRace ?' kann man sich eine Liste aller CLI-Parameter ausgeben lassen. Es folgt eine Beschreibung aller CLI-Parameter:

MODULES/M - Hier werden alle auszuführenden Module angegeben. Es darf kein kompletter Pfad angegeben werden, da alle Module im Verzeichnis 'Modules' und in weiteren Unterverzeichnissen gesucht werden. Die Module können auch ohne die Endung '.worm' angegeben werden. Es kann auch ein Verzeichnis angegeben werden (wieder ohne Pfad!), dann werden alle Module, welche sich in diesem Verzeichnis befinden, ausgeführt.

M=M68K/S - Normalerweise führt WarpRace nur die Module aus, welche den PPC-Prozessor unterstützen. Wenn der Schalter M68K angegeben ist, führt WarpRace alle Module aus, welche den 68K-Prozessor unterstützen. Ein Modul kann auch beide Prozessoren unterstützen. Es erhält dann von WarpRace die Information, welcher Test gewünscht ist und kann entsprechend verzweigen.

A=ALL/S - Wenn dieser Schalter angegeben wird, führt WarpRace alle Module aus, welche den eingestellten Prozessor unterstützen und welche sich im Verzeichnis 'Modules' befinden.

F=FULL/S - Wenn diese Option angegeben wird, wird zu jedem Modul eine ausführliche Beschreibung ausgegeben.

S=STATS - Dieser Parameter verlangt zusätzlich einen Dateinamen (kein Pfad). WarpRace erzeugt dann eine speziell formatierte Ausgabe mit den Ergebnissen der Messreihe, welche dann unter dem angegebenen Dateinamen im Verzeichnis 'Stats' abgespeichert wird.

Einige Beispiele zur Benützung von WarpRace:

```

warprace      ALL                ;führt alle PPC-Module aus
warprace      ALL M68K           ;führt alle 68K-Module aus
warprace      LongRead LongWrite ;führt die beiden Module 'LongRead'
;und 'LongWrite' auf dem PPC aus.
warprace      Memory M68K       ;führt alle Module im Unterverzeichnis
;'Memory' auf dem 68K aus.
warprace      Copy STATS test.txt ;führt alle Module im Unterverzeichnis
;'Copy' auf dem 68K aus und schreibt
;eine statistische Ausgabe nach
;'Stats/test.txt'
```

1.5 Ausgabeformat

WarpRace gibt direkt nach dem Start eine Titelzeile aus und danach zwei weitere Informationen:

```

CPU:                - Der Prozessor auf welchem die Module laufen.
Version of powerpc.library - Die Version der powerpc.library.
```

Die Ausgabe, welche ein Modul erzeugt, ist standardisiert. Im folgenden sind die verschiedenen Ausgabeinformationen erklärt.

```

Module:            - Der Namen des Moduls und seine Version
Author:            - Der Autor des Moduls
Short:             - Eine Kurzbeschreibung der Aufgabe des Moduls
Description:       - Eine ausführliche Beschreibung des Moduls. Diese Ausgabe
                    erscheint nur, wenn der CLI-Parameter FULL angegeben
                    wurde.
Result:            - Das Resultat der Messung. Es steht dem Modul frei, wie
                    die Ausgabe erfolgen soll. Es kann z.B. eine Zeitangabe
                    oder eine Angabe des Speicherdurchsatzes ausgegeben
                    werden.
```

1.6 Statistik

WarpRace erzeugt auf Wunsch eine speziell formatierte Ausgabe, welche die Messresultate beinhaltet. Dazu muss der CLI-Parameter STATS mit einem Dateinamen angegeben werden.

Der Sinn dieser Ausgabe ist der, dass damit eine Präsentation der Resultate gemacht werden kann (z.B. in Form von Balkendiagrammen). Die spezielle Formatierung soll die Auswertung der Resultate vereinfachen.

WarpRace beinhaltet zur Zeit kein solches Auswertungsprogramm. Es kann jetzt aber jedermann ein solches Programm schreiben, welches die Resultate wesentlich lesbarer machen soll.

Im folgenden ist die Formatierung der Ausgabedatei erklärt. Jede Information hat folgenden Aufbau:

```
KEYWORD=Information
```

Es gibt Keywords, welche globalen Charakter haben und es gibt solche, welche Modulspezifisch sind.

Globale Keywords (können überall in der Datei vorkommen):

CPU=<CPUString> - Der Prozessor-Typ als String (z.B. PPC604E)
LIBVER=<Versionnummer> - Die Version der powerpc.library (z.B. 12.0)

Modulspezifische Keywords:

NAME=<Modulname> - Der Name des Moduls (z.B. TurboCopy). Dieses Keyword ist IMMER der Beginn eines neuen Moduls.
VERSION=<Versionsnummer> - Die Version des Moduls (z.B. 1.0)
RESTYPE=<Resultatstyp> - Ein numerischer Wert, welcher die Art des Resultats angibt. Die möglichen Werte sind in der Include-Datei 'warprace.i' angegeben.

Mögliche Werte für den Resultatstyp:

- 0 - Der Typ des Resultats ist nicht bekannt, somit können auch keine Vergleiche vorgenommen werden.
- 1 - Der Typ des Resultats ist nicht bekannt, jedoch verhält sich das Resultat proportional zur Leistung (wie z.B. Speicherdurchsatzraten), somit können Vergleiche vorgenommen werden.
- 2 - Der Typ des Resultats ist nicht bekannt, jedoch verhält sich das Resultat umgekehrt proportional zur Leistung (wie z.B. Zeitangaben), somit können Vergleiche vorgenommen werden.
- 3 - Das Resultat ist vom Typ 'Anzahl Mikrosekunden' (somit umgekehrt proportional zur Leistung). Das Resultat ist numerisch.
- 4 - Das Resultat ist vom Typ 'Bytes pro Sekunde' (Speicherdurchsatz) und ist somit proportional zur Leistung. Das Resultat ist numerisch.

Neue Werte werden in Zukunft eventuell hinzugefügt werden, wenn dies gewünscht wird.

RESULT=<Resultat> - Das Resultat welches mit Hilfe des Resultatstyp interpretiert werden kann.

1.7 Entwickler-Infos

Im folgenden wird nun ausführlich erklärt, wie man ein WarpRace-Modul erstellt. Ein WarpRace-Modul ist ein gewöhnliches ausführbares Programm, welches allerdings andere Parameter erhält und zurückgibt.

Wenn ein Modul in einer Hochsprache geschrieben wird, muss nur der

Prototyp der 'main'-Funktion entsprechend abgeändert werden. Zudem muss ohne Startup-Code gelinkt werden. Ein Modul sollte auch nicht direkte Ausgaben in das CLI machen. Fehlermeldungen sollten über die unten erklärten Parameter ausgegeben werden.

Im Verzeichnis 'ModSrc' befinden sich einige Beispiele von Modulen, welche in Assembler geschrieben wurden.

WICHTIG: Die 'main'-Funktion muss in 68K-Code vorliegen, selbst bei Modulen, welche nur den PPC unterstützen. Die 'main'-Funktion muss in diesem Falle den PPC-Teil des Moduls aufrufen.

Die nötigen Strukturen und Definitionen befinden sich in den Dateien 'warprace.h' (C) und 'warprace.i' (Assembler).

Der Prototyp der 'main'-Funktion des Moduls sieht wie folgt aus:

```
struct WR_Output* main(WR_ID, struct WR_Input*)
d0                d0        a0
```

Die Eingabe-Parameter werden sowohl in den angegebenen Registern, wie auch über den Stack übergeben.

Die Eingabeparameter:

WR_ID : Dies ist eine Konstante (definiert in der Include-Datei), welche dazu dient zu unterscheiden, ob das Modul tatsächlich von WarpRace gestartet wurde oder nicht. Wenn ein solches Modul versehentlich direkt in der Shell gestartet wurde, sollte es sich sauber beenden.

WR_Input : Diese Struktur enthält einige wichtige Angaben für das Modul. Sie hat folgenden Aufbau:

```
struct WR_Input {
APTR    WRI_PowerPCBase;
ULONG   WRI_Version;
void    (*WRI_StartTimer_68K)(void);
ULONG   (*WRI_StopTimer_68K)(void);
void    (*WRI_StartTimer_PPC)(APTR);
ULONG   (*WRI_StopTimer_PPC)(APTR);
APTR    WRI_LinkDB;
BOOL    WRI_68K;
ULONG   WRI_Flags;
APTR    WRI_Ext;
};
```

Die einzelnen Felder haben folgende Bedeutung:

WRI_PowerPCBase - Basisadresse der powerpc.library
WRI_Version - Version der powerpc.library
WRI_StartTimer_68K - Zeiger auf eine 68K-Funktion, welche den internen Timer startet. Diese Funktion wird immer zu Beginn einer Zeitmessung aufgerufen.

- WRI_StopTimer_68K - Zeiger auf eine 68K-Funktion, welche den internen Timer wieder anhält. Diese Funktion gibt die Zeit in Mikrosekunden zurück, welche zwischen den Aufrufen von WRI_StartTimer_68K und WRI_StopTimer_68K verstrichen ist.
- WRI_StartTimer_PPC - Zeiger auf eine PPC-Funktion, welche den internen Timer startet. Diese Funktion erhält als Eingangswert das Element WRI_LinkerdB.
- WRI_StopTimer_PPC - Zeiger auf eine PPC-Funktion, welche den internen Timer wieder anhält. Diese Funktion erhält als Eingangswert das Element WRI_LinkerdB und gibt die Zeit in Mikrosekunden zurück, welche zwischen den Aufrufen von WRI_StartTimer_PPC und WRI_StopTimer_PPC verstrichen ist.
- WRI_LinkerdB - Dieses Element muss als Eingabeparameter für die PPC-Timer-Funktionen verwendet werden, falls diese aufgerufen werden.
- WRI_68K - Ist dieser Schalter gesetzt, soll das Modul auf dem 68K laufen, ansonsten auf dem PPC. Wenn das Modul die gewünschte CPU nicht unterstützt, sollte sie sich mit NULL als Ausgangswert beenden.
- WRI_Flags - Zur Zeit unbenützt.
- WRI_Ext - Zur Zeit unbenützt.

Die Timer-Funktionen dürfen nicht gemischt benützt werden.

Der Ausgabeparameter:

WR_Output: Zeiger auf eine WR_Output-Struktur oder NULL, falls das Modul nicht ausgeführt werden kann (z.B. weil die gewünschte CPU nicht unterstützt ist. Wenn eine NULL zurückgegeben wird, findet keine Ausgabe in der CLI statt. Falls das Modul einen Fehler erkennt, kann es auch eine WR_Output-Struktur zurückgeben und die Elemente zur Fehlerbehandlung entsprechend setzen. In diesem Fall erscheint die Fehlermeldung in der CLI.

Die WR_Output-Struktur hat folgenden Aufbau:

```
struct WR_Output {
STRPTR  WRO_Modname;
STRPTR  WRO_Short;
STRPTR  WRO_Description;
STRPTR  WRO_Author;
ULONG   WRO_Version;
ULONG   WRO_Revision;
ULONG   WRO_RevSize;
ULONG   WRO_Flags;
STRPTR  WRO_Result;
ULONG   WRO_ResultType;
STRPTR  WRO_ResultString;
APTR    WRO_ResultParams;
ULONG   WRO_Status;
STRPTR  WRO_ErrorString;
APTR    WRO_ErrorParams;
```

```
APTR    WRO_Ext;  
};
```

Die einzelnen Felder haben folgende Bedeutung:

WRO_Modname - Der Name des Moduls
WRO_Short - Eine Kurzbeschreibung des Moduls (sollte nicht grösser als eine Zeile sein).
WRO_Description - Eine ausführliche Beschreibung des Moduls. Kann mehrere Zeilen umfassen (die letzte Zeile sollte aber kein LineFeed mehr enthalten).
WRO_Author - Der Name des Autors
WRO_Version - Die Version des Moduls
WRO_Revision - Die Revision des Moduls (Nachkommastellen)
WRO_RevSize - Anzahl Stellen der Modul-Revision. Damit werden Probleme vermieden bei Versionen von z.B. 1.01. In diesem Fall müsste WRO_Revision gleich 1 und WRO_RevSize gleich 2 sein.
WRO_Flags - Zur Zeit unbenutzt.
WRO_Result - Das Resultat der Messung. Dieses Resultat wird nur verwendet, wenn eine statistische Ausgabe gemacht wird. Es ist zu beachten, dass das Resultat mit dem Resultatstyp zu vereinbaren ist.
WRO_ResultType - Der Typ des Resultats (siehe Kapitel Statistik). Wird nur bei statistischen Ausgaben verwendet.
WRO_ResultString - Ein Formatstring, welcher von WarpRace beim Punkt 'Result' mittels VPrintf ausgegeben wird.
WRO_ResultParams - Ein Zeiger auf die Parameter für WRO_ResultString.
WRO_Status - Kann folgende Werte beinhalten:
STATUS_SUCCESS : Das Modul wurde erfolgreich ausgeführt.
STATUS_ERROR : Das Modul hat einen Fehler erkannt.
WRO_ErrorString - Ein Formatstring, welcher im Fehlerfall von WarpRace mittels VPrintf ausgegeben wird.
WRO_ErrorParams - Ein Zeiger auf die Parameter für WRO_ErrorString.
WRO_Ext - Zur Zeit unbenutzt.

Es ist darauf zu achten, dass die Struktur vollständig und korrekt ausgefüllt wird und den Fall beachtet, dass der Benutzer die statistische Ausgabe einschaltet.

Es sei nochmals darauf hingewiesen, dass ein Modul nicht eine Fehlermeldung zurückgeben soll, falls die gewünschte CPU nicht unterstützt ist. Hingegen sollte eine NULL als Ausgabeparameter zurückgegeben werden. Damit wird vermieden, dass viele unnötige Ausgaben gemacht werden, die den Benutzer nur stören.