

WHITE PAPER

TurboHercules Overview

A Quick Overview of the TurboHercules System

TurboHercules Corporation

August 2009

Abstract

This paper is an overview of the structure and operation of the TurboHercules package.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of TurboHercules Inc.

TurboHercules may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from TurboHercules, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2009 TurboHercules Inc. All rights reserved.

TurboHercules is a trademark of TurboHercules SAS in France and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

1	TurboHercules Emulator Overview	1
2	Emulator Structure	1
3	TurboHercules Operation	2
4	Summary	3

1 TurboHercules Emulator Overview

The TurboHercules Emulator is a Windows or Linux based program that emulates the instruction set and many peripherals of an IBM mainframe in software. The models and peripherals emulated are listed in the <http://www.hercules-390.org/HerculesUserReference.pdf>. When the emulator is running it effectively creates a virtual machine that can execute software that would run on a real mainframe of the same configuration. At the moment this covers all of the IBM supplied operating systems from **OS/360 to z/OS** as well as the mainframe Linux operating systems from Novell (SUSE Enterprise Linux for s390x), Red Hat and CentOS. ***Please note that it is the responsibility of the user to insure that the appropriate licensing requirements are met for any software being executed on the emulator.***

2 Emulator Structure

Like most of the emulator/virtualization packages, Hercules is comprised of three major parts. The first is the binary code of the emulator itself. All of the binaries are stored in a single directory. Its location depends on the host operating system.

For the Windows operating system the binaries of the emulator are stored in the Program Files (x86)\TurboHercules directory. When the system is installed, shortcuts to the appropriate binaries are installed in the Start menu.

The second part of the emulator is the **hercules.cnf** configuration file. This is read on Hercules startup to determine the configuration of the emulated machine. Among other items it contains the CPU model number, the number of processors, the amount of main memory and a list of the peripherals (type and address), etc that the emulated machine will have. For each configured storage device the configuration file maps the device address/type to the supporting host's data file. The exact format of the configuration file can be found at <http://www.hercules-390.org/hercconf.html>.

The exact location of this file (and possibly the name depends on the host operating system and the guest or target operating system. For a Hercules emulator running SUSE Enterprise Linux on a Windows host operating system this file is stored in **C:\TurboHercules\data\SLES\conf\hercules.cnf**. The beta version of the Hercules emulator running MVS on a Windows host operating system has this file stored in **C:\TurboHercules\data\mvs\conf\turnkey_mvs.conf**. (In future revisions it may be named hercules.cnf, but this is not a requirement.)

Finally you have the data files that contain the images of the emulated storage peripherals. The information stored on an emulated DASD device is stored in a file or series of files in the host operating system. The printer, punch, etc. are handled in a similar fashion. Connections to emulated 3270 terminals are supported by a 3270

emulator through a network connection. All of the data files for the emulator running on a Windows host are stored in **C:\TurboHercules\data\mvs** (MVS guest operating system) or **C:\TurboHercules\data\SLES** (SUSE Enterprise Linux operating system).

This structure allows the user to have a separate set of configuration and data files for each mainframe configuration they want to be able to emulate. The demo package contains two configuration/data file sets for starters.

The first set is configured to support the installation and execution of SUSE Enterprise Linux 10 SP2. It lists a single processor, 1GB of main memory, a single 9GB DASD device and a single network interface. The DASD device is pre-initialized for a Linux installation, but the disk is otherwise empty. See the Linux installation section for instructions on how to obtain and install the operating system.

The second set is configured to support the execution of MVS 3.8J. It lists a single processor, an appropriate amount of main memory, multiple DASD devices and a selection of 3270/3215 terminals. The DASD devices are pre-loaded with the MVS 3.8J image so all that is necessary is to IPL from the appropriate device. Instructions for running the MVS image are described in Volker Bandke's site:

(<http://www.bsp-gmbh.com/turnkey/index.html>). (Note: if you are not a mainframe programmer/operator, MVS may seem a bit strange to you.)

For the user who must be able to support multiple mainframe hardware/software configurations, you would only be limited by the total amount of disk space necessary on the host operating system. With a modern PC based server, it is not unusual to have access to several terabytes of local storage within the server itself so dozens of configurations with all of the included data could easily be accommodated

3 TurboHercules Operation

To start the emulator you simply execute:

hercules -f <hercules configuration file>

on the command line of the host operating system. This can be scripted as it is in both the Linux and Windows based installations. The emulator first reads and checks the configuration file. It then checks for the existence and format of the data files listed in the configuration file. If everything is correct it creates its own command prompt and waits for user input. At this point the emulator is running, but it is not executing any mainframe instructions yet.

You begin emulation by an initial program load (IPL) from a device that contains executable code. For instance, if you have installed SUSE Enterprise Linux in the default

configuration, you would IPL from device oA80. This would start a fairly standard Linux boot procedure. Once the operating system is fully booted, you would log on using standard SSH sessions as you would with any Linux based system.

Since the emulator itself is a multi-threaded application, it is possible to enter commands on the emulator command line while the emulator is running. See the command guide on the main Hercules website (<http://www.hercules-390.org/>) for a list of the emulator commands and their effects.

4 Summary

TurboHercules is a general purpose emulator for IBM mainframes from the System 370 to the present (z9/z10 family). It can run the IBM mainframe operating systems like z/OS and z/VM as well as z/Linux. Since it is configured completely in software, it can be a flexible platform for mainframe development, testing, education, disaster recovery and business continuity applications.

For more information and to arrange a proof of concept demonstration please contact us at info@turbohercules.com.