

IBM Full-System Simulator: Overview and Installation

Updated: March 2006

The IBM Full-System Simulator, internally referred to as “Mambo,” has been developed and refined by the IBM Austin Research Lab (ARL) in conjunction with several large system design projects built upon the IBM Power architecture. As an execution-driven, full-system simulator, the IBM Full-System Simulator has facilitated the experimentation and evaluation of a wide variety of system components for core IBM initiatives. The IBM Full-System Simulator for PowerPC 970, available from the **IBM alphaWorks Emerging Technologies** web site, enables development teams both within IBM and externally to simulate a PowerPC 970 system in order to develop and enhance application support for this platform.

This document introduces the IBM Full-System Simulator for PowerPC 970 installation environment, summarizes hardware and software prerequisites, describes procedures to install and run a default simulator, and provides troubleshooting information to isolate and fix a potential installation problems.

Simulation Interfaces and Tools

The IBM Full-System Simulator is a complete simulation infrastructure that enables systems and software developers to run a variety of data collection and analysis tools to gather multiple types of system metrics at varying levels of granularity. Users also can launch a number of visualization tools to interactively monitor system behavior and diagnose potential performance bottlenecks. Figure 1 provides an overview of the IBM Full-System Simulator’s application tools and interfaces:

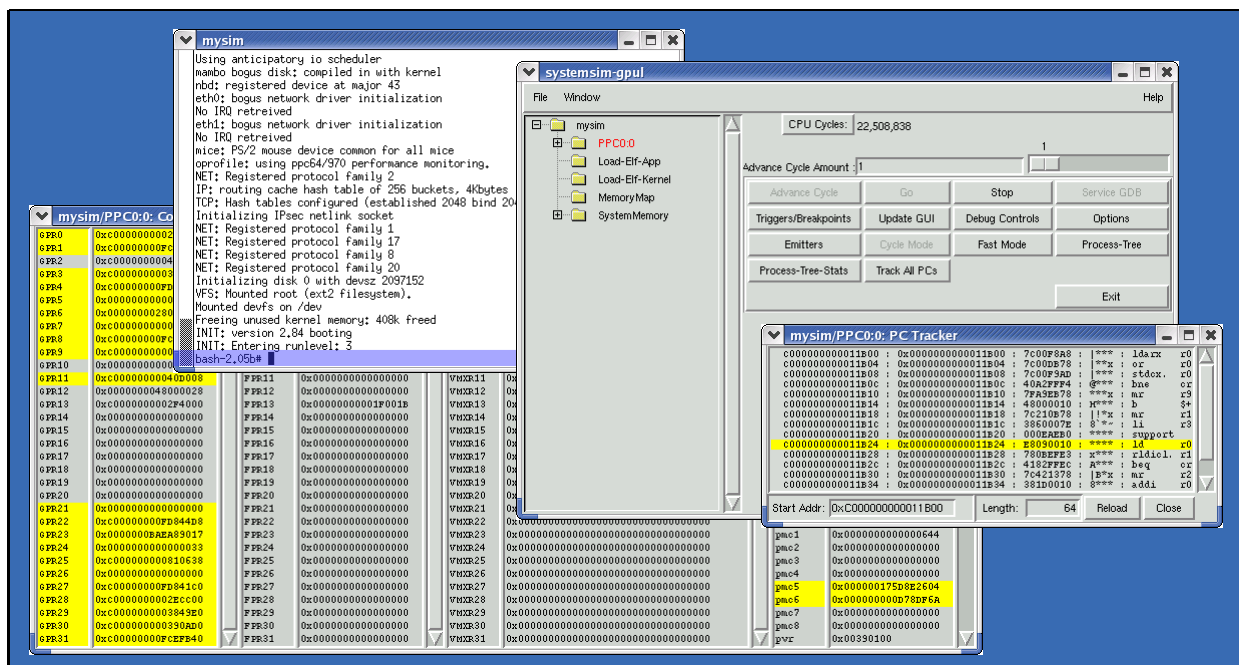


Figure 1. IBM Full-System Simulation Interfaces and Tools

The IBM Full-System Simulator's functional fidelity and runtime performance allow a full operating system, such as Linux, to be run interactively in simulation—in this manner, the simulator provides applications that require inter-process or complex operating system interactions with a complete environment. In addition to this full operating system mode, the IBM Full-System Simulator provides a “standalone” environment for self-contained applications, in which the simulator intercepts and marshals the application’s system calls to the underlying host to optimize execution.

Preplanning the Simulation Environment

The IBM Full-System Simulator installation sets up the hardware and software infrastructure, development tools, and system services required to start running and using the simulator. The topics in this section describe important installation information to consider before installing your simulation system.

Installation Topology

The IBM Full-System Simulator installation package contains the base installation files and object code for the simulator. The installation also provides scripts and makefiles to build and configure supporting infrastructure components, such as the PowerPC toolchain, 64-bit PowerPC Linux kernel, and 64-bit PowerPC rootdisk. For example, a provided makefile downloads the necessary build tools, binary utilities, and source for GCC, GLIBC, and Unix utilities, and includes steps to build and configure a 64-bit PowerPC Linux kernel and rootdisk. Figure 2 illustrates the standard execution topology for the simulator

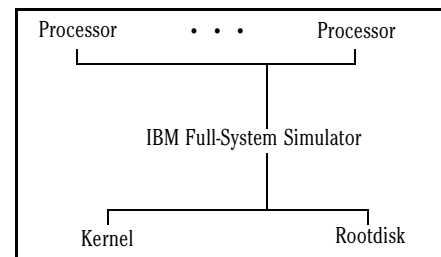


Figure 2. Multiprocessor System Simulation

Once built, the rootdisk image provides a snapshot of a functioning Linux system that is available inside the IBM Full-System Simulator, including all libraries and debuggers that are required to run an actual Linux system—all of which enables the simulator to provide the appropriate run-time support to run applications as they are executed in an actual Linux environment. The IBM Full-System Simulator is designed to optimize the execution of the Linux kernel by reading contents of the rootdisk image as the simulator traverses the root filesystem.

Installation Requirements

Before installing any simulator components, verify that your system meets the following minimum hardware and software requirements.

- ◆ The IBM Full-System Simulator is supported on machines with a minimum of **3 GB** of available hard disk space to install the core simulator files and rootdisk image. The simulator must be created and installed on a non-networked directory.
- ◆ The minimum amount of RAM must equal **twice** the amount of simulated memory—for example, if the simulator is simulating a system with 256 MB of RAM, the host system must have at least 512 MB of RAM.
- ◆ The simulator is supported on RedHat Linux v8.0, RedHat Linux v9.0, RedHat Enterprise Linux v3, Fedora Core 2, Fedora Core 3, and Fedora Core 4.
- ◆ *Root privileges are required to build the rootdisk image; ensure that you are authorized with the correct privileges before building the rootdisk.*

Obtaining Installation Media for Your System

Installation media for the IBM Full-System Simulator for PowerPC 970 is available from the **IBM alphaWorks Emerging Technologies** web site for the following Linux platforms:

Table 1-1. Installation Matrix for Host Systems

<i>Installation Package^a</i>	<i>Host Platform</i>	<i>File Size</i>
SystemSim_x86_rh8_2.0.tar.bz2	Supported on RedHat Linux v8.0 and RedHat Linux v9.0 ^b running on an x86 machine.	7.9 MB
SystemSim_x86_rhel3_2.0.tar.bz2	Supported on RedHat Enterprise Linux v3, Fedora Core 2, and Fedora Core 3 running on an x86 machine.	7.7 MB
SystemSim_x86_fc4_2.0.tar.bz2	Supported on Fedora Core 4 running on an x86 machine.	7.7 MB
SystemSim_ppc_sles9_2.0.tar.bz2	Supported on SUSE LINUX Enterprise Server 9 running on a PowerPC machine.	7.7 MB
SystemSim_ppc_fc4_2.0.tar.bz2	Supported on Fedora Core 4 running on a PowerPC machine.	7.8 MB

a. Package names follow the naming convention: **IBM_SystemSim_alphaworks_sdk_x_y_z.tar.bz2**, where *x* specifies the host platform, *y* specifies the host Linux version, and *z* is the most current version of the binary that is available from the **IBM alphaWorks Emerging Technologies** web site.

b. Installing **IBM_SystemSim_alphaworks_sdk_x86_rh8_y.tar.bz2** on RedHat Linux v9.0 may require you to create symbolic links to point the Linux libraries to the files in the simulator. The “Troubleshooting Your Installation” on page 7 describes steps to resolve this issue.

Creating and Installing the IBM Full-System Simulator Environment

To run the simulator, complete the following series of procedures to build a PowerPC toolchain, a 64-bit PowerPC Linux kernel, and a 64-bit PowerPC rootdisk. Once built, the IBM Full-System Simulator provides a complete simulation environment that includes all available data collection and visualization tools, Tcl commands, and call-thru interfaces. “Troubleshooting Your Installation” on page 7 describes information about installation issues that you may encounter.

►► DOWNLOAD AND EXTRACT SIMULATOR BINARY FILES

1. Download the **IBM_SystemSim_alphaworks_sdk_x_y.tar.bz2** package from the **IBM alphaWorks Emerging Technologies** web site, as outlined in the previous “Obtaining Installation Media for Your System” section. To determine the version of Linux that is installed on your system, type the following **rpm** command at the command line:

```
rpm -qa | grep release
```

2. Extract files from the package; the following sample **tar** command extracts the base installation files:

```
tar xjf {installation_directory}/IBM_SystemSim_alphaworks_sdk_x.tar.bz2
```

where *{installation_directory}* is the directory that contains the installation tar file, and *x* is the current version of the binary. The **tar** command extracts files in the package into the **ibmsim** directory.

▶▶ CREATE A POWERPC TOOLCHAIN

1. Build the PowerPC toolchain via the makefile provided in the **ibmsim/toolchain** directory. This step requires an Internet connection in order to access the required tools, libraries, and files from third-party web sites.

NOTE

The toolchain build operation is a fairly lengthy process—it may be useful to concurrently develop the PowerPC *rootdisk* image. See the “Create a PowerPC Root Environment” procedure for instructions on building a rootdisk.

Change to the **toolchain** directory:

```
cd ibmsim/toolchain
```

2. Use the **make** command to compile the toolchain:

```
make toolchain_all
```

▶▶ CREATE A POWERPC LINUX KERNEL

1. Once the toolchain is constructed, build a 64-bit PowerPC Linux kernel with the makefile provided in the **ibmsim/toolchain** directory. Change to the **toolchain** directory:

```
cd ibmsim/toolchain
```

2. Use the **make** command to compile the kernel:

```
make kernel_all
```

▶▶ CREATE A POWERPC ROOT ENVIRONMENT

Complete the following procedures to build a 64-bit PowerPC rootdisk:

NOTE

- ◆ **Root privileges are required to build the rootdisk image; ensure that you are authorized with the correct privileges before completing the steps in this section.**
- ◆ **No system files are modified while running the build process as root. IBM recommends viewing the `build_rootdisk` target in the Makefile to learn about the sequence of steps in this automated process.**
- ◆ ARL has chosen Gentoo Linux for the simulator root environment. Other operating systems may work as well, with the provision that the contents of the **inittab** file may differ.

1. Build a 64-bit PowerPC root environment with the makefile provided in the **ibmsim/toolchain** directory. Change to the **toolchain** directory:

```
cd ibmsim/toolchain
```

2. Use the **make** command to compile the kernel:

```
make build_rootdisk
```

Running the IBM Full-System Simulator

The installation process described in “Creating and Installing the IBM Full-System Simulator Environment” installs and configures a local instance of the simulator, which contains a default **.systemsim.tcl** file that can be used to configure

machine definitions and environment settings that are loaded when the simulator starts. At startup, the simulator loads instructions in **.systemsim.tcl** to set up default simulation behavior. Alternatively, a custom Tcl file may be created from the default **.systemsim.tcl** file to start up and configure a simulation environment with system-specific settings.

To view and modify simulation settings in your simulation environment, open the **.systemsim.tcl** file in a text editor, for example such as emacs:

```
emacs ibmsim/simulators/systemsim-gpul-release/run/gpul/linux/.systemsim.tcl
```

Once the simulation environment is installed and configured, either the IBM Full-System Simulator graphical user interface or the command line can be used to configure components of the microprocessor model, generate performance metrics with new or revised configurations, and run workloads on the modeled architecture. The command line interface also can be used to perform a number of operations on the simulator itself, such as adding commands to control a simulation, or starting data collection and visualization tools. The following procedures describe how to start a simulation from the command line and graphical user interfaces.

►► TO START A SIMULATION FROM THE COMMAND LINE INTERFACE

1. The IBM Full-System Simulator is launched from the **ibmsim/simulators/systemsim-gpul-release/run/gpul/linux** directory. Change to the **linux** directory and start the simulator command line interface:

```
cd ibmsim/simulators/systemsim-gpul-release/run/gpul/linux
../run_cmdline
```

The default **run_cmdline** behavior is to read Tcl commands defined in **.systemsim.tcl**. The following command line options are available to modify the simulator startup:

- f {file}** Overrides the **.systemsim.tcl** file to start the simulator with the specified Tcl file.
- g** Starts the graphical user interface. ARL also provides the **../run_gui** command to start the graphical user interface. See page 6 for a procedure to launch the simulator graphical user interface.
- n** Does not open an XTerm for the console.
- q** Runs the simulator in *quiet mode*. Quiet mode suppresses the printing of the IBM legal notice on startup and the periodic printing of the number of instructions being executed. Quiet mode is typically used for running regression tests, in which the varying speed or load of the host processor is expected to change simulator output in uncontrollable ways.

The IBM Full-System Simulator launches the simulation console window, which displays output of the simulated machine and allows users to configure and interact with the simulation. Figure 3 displays a sample console at the Linux prompt after the simulator has booted Linux.

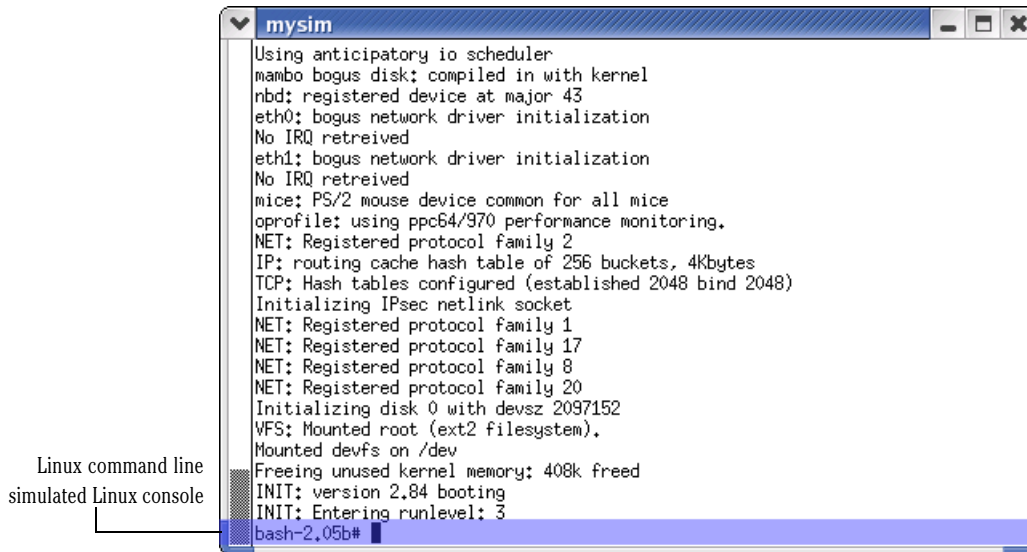


Figure 3. Sample Simulator Console Window During Linux Boot

2. To interrupt a simulation, type **CTRL+C** at the simulator command line. The simulator stops the simulation and returns to the **systemsim %** command line prompt.
3. To resume the simulation, type the **mysim go** command at the **systemsim %** command line prompt, as follows:

```
mysim go
```
4. To end the simulation session, type **quit** at the **systemsim %** command line prompt.

▶▶ TO START THE SIMULATOR GRAPHICAL USER INTERFACE

As with the simulator command line interface, the graphical interface is launched from the **ibmsim/simulators/systemsim-gpul-release/run/gpul/linux** directory. To start the IBM Full-System Simulator in the graphical user interface mode, change to the **linux** directory and run the **run_gui** command:

```
cd ibmsim/simulators/systemsim-gpul-release/run/gpul/linux
../run_gui
```

Alternatively, you can start the graphical user interface with the **-g** option to the **run_cmdline** command, as follows:

```
../run_cmdline -g
```

The IBM Full-System Simulator launches the **SystemSim GPUL** graphical user interface (GUI) that provides a set of tools to monitor and debug a simulation.

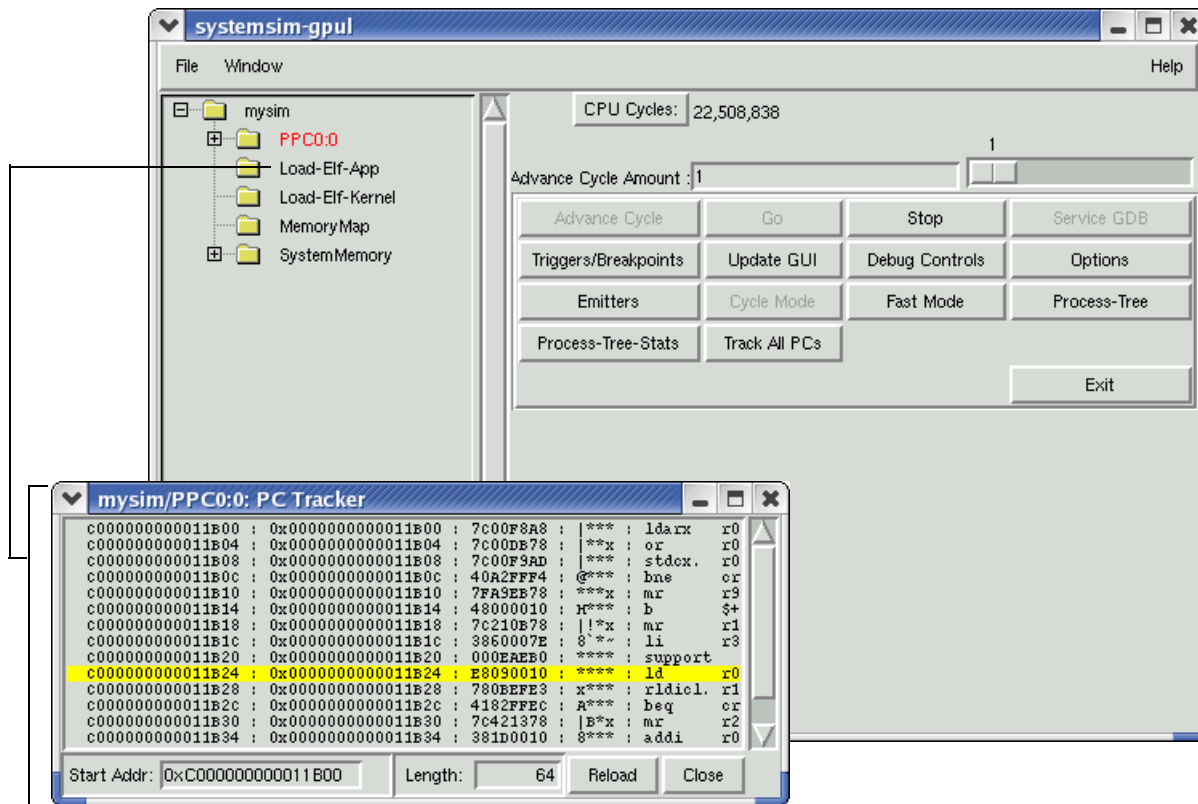


Figure 4. Simulator Graphical User Interface with PC Tracker Tool

The IBM Full-System Simulator GUI is most commonly used to perform the following operations:

- ◆ **Debugging.** The simulator GUI provides a number of tools and features to accomplish low-level debugging tasks, such as disassembling instructions at the assembly level or monitoring registers during a simulation. For example, the PC Tracker tool shown in Figure 4 displays the logical sequence of instructions.
- ◆ **Performance tracking.** The simulator GUI provides tools to monitor performance of system components in a simulation.

Troubleshooting Your Installation

Topics in this section describe potential problems and provide solutions to issues that you may encounter during installation or while using the simulator.

QUESTION: I am encountering problems when I try to install the IBM Full-System Simulator on a networked drive.

SOLUTION: Installing on a networked machine is *not* recommended. ARL recommends that you install the IBM Full-System simulator on a local hard-drive to avoid installation problems associated with machine permissions or download issues.

QUESTION: I receive the following error related to shared Tcl/Tk libraries when trying to launch the IBM Full-System Simulator:

```
../../../../bin/systemsim-gpul: error while loading shared libraries:  
libtcl.so.0: cannot open shared object file: No such file or directory
```

SOLUTION: The names of Tcl and Tk libraries is most often dependent on the host operating system—some versions name the Tcl and Tk libraries as **libtcl.so** and **libtk.so**, respectively. The IBM Full-System Simulator, however, references the Tcl and Tk libraries as **libtcl.so.0** and **libtk.so.0**. To resolve this issue, create new symbolic links to point the Linux libraries to the files in the simulator. With **root** permissions, type the following **ln** commands in the **/usr/lib** directory:

```
ln -s libtcl.so libtcl.so.0  
ln -s libtk.so libtk.so.0
```

QUESTION: I receive the following error related to shared C++ libraries when trying to launch IBM Full-System Simulator:

```
../../../../bin/systemsim-gpul: error while loading shared libraries:  
libstdc++.so.5: cannot open shared object file: No such file or directory
```

SOLUTION: This error occurs if you have installed a version of the IBM Full-System Simulator that is not compatible with your system environment, or if you have installed the simulator on an unsupported version of Linux. See “Preplanning the Simulation Environment” on page 2 for information about IBM Full-System Simulator installation requirements on supported platforms.