SimicsFS Read Me

Table of Contents

1 Introduction

SimicsFS is a Linux kernel module used to mount the host filesystem from within the simulated target system. It provides both read and write access to files located on the host filesystem.

SimicsFS is provided as source code and the user must compile it against the Linux kernel source code for the target system. Pre-compiled modules are available in most images distributed in Simics packages.

1.1 Limitations

- Linux kernel 2.6.12 and later only
- The code is not production quality
- The module receives limited testing

2 Compiling SimicsFS for a running kernel

1. Unpack the simicsfs archive (where x.y is the version number).

   ```
   $ tar xzf simicsfs-x.y.tar.gz
   ```

2. Install the kernel source and development packages.

   See the distribution documentation for more information about how to find and install packages.

3. Build the kernel source code.

   See the distribution documentation for more information on how to build its kernel; typically it is enough to run `make` in the source code root directory.

4. Build the kernel module.

   Change to the simicsfs directory, and then compile:

   ```
   $ make -C /lib/modules/`uname -r`/build M=`pwd`
   ```

5. Install the kernel module.

   The resulting `simicsfs.ko` module file must be inserted into the running kernel. Optionally it may be configured, which is described in section How to override the physical address of the driver:

   ```
   $ insmod simicsfs.ko
   ```

   If a simicsfs module is already inserted in the kernel, it must first be removed by using `rmmod`.

The Linux kernel source code provides more information about building external modules in `Documentation/kbuild/modules.txt`.

3 Adding SimicsFS to a kernel source tree

1. Unpack the simicsfs archive in the Linux `fs` directory.

   You must update the file and path names with the current version numbers:

   ```
   $ cd /path/to/linux-x.y.z/fs/
   $ tar xzf simicsfs-x.y.tar.gz
   ```

2. Edit `fs/Makefile` and `fs/Kconfig`.

   Append the following line to the end of `fs/Makefile`:

   ```
   obj-$(CONFIG_SIMICSFS)             += simicsfs/
   ```

   Add the following line to the end of the section "Miscellaneous filesystems" in the `fs/Kconfig` file:

   ```
   source "fs/simicsfs/Kconfig"
   ```

3. Configure the Linux kernel and select simicsfs as a module.

   ```
   $ make menuconfig
   File systems  --->
   Miscellaneous filesystems  --->
   <M>     Simics hostfs
   ```

   Optionally you may set a default physical address, which is described in the section How to find and set a physical address for the driver.

4. Compile the kernel and its modules.

   Check the distribution documentation for instructions on how to build the Linux kernel; typically it is enough to run `make` in the source code root directory:

   ```
   $ make
   ```

   Or build only the simicsfs module:

   ```
   $ make SUBDIRS=fs/simicsfs modules
   ```

   The resulting module file is located at `fs/simicsfs/simicsfs.ko`.

   See the README file in the Linux source code for more information on how to build the kernel.

5. Install the kernel module.

   The module can be either manually inserted into the target system at run-time or automatically loaded by the kernel.

   Check the distribution documentation for how to add a kernel module to the target file system. Typically it involves placing the module file under `/lib/modules/`uname -r`/extra/fs/` and running `depmod`.

   If necessary, see the section How to override the physical address of the driver.

   To manually load the kernel module at run-time:

```
$ insmod simicsfs.ko
```

See `man insmod` for more information.

4 How to find and set a physical address for the driver

1. Find a 4 kiB (`0x1000`) empty region in I/O memory. This command is executed on the running target system: .. code:: bash

   > $ cat /proc/iomem

   Note that the output is a static snapshot and may not show dynamic usage of I/O memory.

2. Check the kernel configuration.

   ```
   $ grep CONFIG_SIMICSFS_ADDRESS .config
   CONFIG_SIMICSFS_ADDRESS=""
   ```

   The string will not appear if the driver has not been selected and it will be empty unless an address has been explicitly set.

3. Edit the configuration as necessary.

   ```
   $ make menuconfig
   File systems  --->
   Miscellaneous filesystems  --->
   <M>     Simics hostfs
   ()      SimicsFS physical address
   ```

   Once the Simics hostfs module has been selected, an option appears where a default physical address can be entered. A built-in default value is used unless the user specifies a new default address.

   Later, you may want to override the address, see section How to override the physical address of the driver.

   Remember to save any changes before exiting.

5 How to override the physical address of the driver

- If the driver was compiled-in (`<*>` in menuconfig)

  Append this argument to the kernel command-line:

  ```
  simicsfs.phys_addr=0xff00ff00
  ```

- If the driver was compiled as a module (`<M>` in menuconfig)

  Give an argument when loading the module:

  ```
  $ insmod simicsfs.ko phys_addr=0xff00ff00
  ```

  Or by using `modprobe`, which requires `depmod` generated files to be present on the target system:

  ```
  $ modprobe simicsfs phys_addr=0xff00ff00
  ```

Alternatively you can add it as an option in `/etc/modules.conf`, `/etc/modprobe.conf` or similar, depending on what is provided by your target system. Check the distribution documentation.