

- [Log in](#)

#### Hosts/Linux - QEMU

- [Page](#)
- [Discussion](#)
- [View source](#)
- [History](#)

Search

Go

Search

- [About](#)
  - [Home](#)
- [Get](#)
  - [Download](#)
- [Contribute](#)
  - [Start Here](#)
  - [Report a Bug](#)
  - [Report a security issue](#)
  - [Submit a Patch](#)
  - [Mailing Lists](#)
  - [Testing QEMU](#)
- [Virtualize](#)
  - [KVM](#)
- [Learn](#)
  - [Documentation](#)
  - [Links](#)
  - [License](#)
- [Toolbox](#)
  - [What links here](#)
  - [Related changes](#)
  - [Special pages](#)

# Hosts/Linux

## From QEMU

### Contents

- [1 QEMU on Linux hosts](#)
  - [1.1 Building QEMU for Linux](#)

- 1.1.1 Fedora Linux / Debian GNU Linux / Ubuntu Linux / Linux Mint
  - 1.1.1.1 Required additional packages
  - 1.1.1.2 Recommended additional packages
- 1.1.2 Getting the source code
- 1.1.3 Simple build and test
- 1.1.4 Simple build and test with KVM
- 1.1.5 Cross builds
- 1.1.6 Native builds
- 1.2 Running QEMU on Linux
  - 1.2.1 System emulation
  - 1.2.2 User mode emulation
- 1.3 Links

## QEMU on Linux hosts

This documentation is work in progress - more information needs to be added for different Linux distributions.

Linux is QEMU's main host platform. Therefore it is the platform which gets most support. Both 32 and 64 bit Linux hosts are supported. Most of the following instructions are valid for both variants.

### Building QEMU for Linux

Most Linux distributions already provide binary packages for QEMU (or KVM).

Usually they also include all packages which are needed to compile QEMU for Linux. The default installation of most distributions will not include everything, so you have to install some additional packages before you can build QEMU.

### Fedora Linux / Debian GNU Linux / Ubuntu Linux / Linux Mint

Fedora, Debian and Debian based or similar distributions normally include compiler and compilation tools (gcc, make, ...) in their default installation.

#### Required additional packages

- git (30 MiB), version manager
- glib2.0-dev (9 MiB), this automatically includes zlib1g-dev
- libfdt-devel

#### Recommended additional packages

- git-email, used for sending patches
- libSDL1.2-dev (23 MiB), needed for the SDL based graphical user interface
- gtk2-devel, for a simple UI instead of VNC
- vte-devel, for access to QEMU monitor and serial/console devices via the GTK interface

## Getting the source code

If you want the latest code, follow the development of the code, work with several versions or maybe even contribute to the code, you will need a local copy of the QEMU code repository which is managed using git.

Get the code like this:

```
git clone git://git.qemu-project.org/qemu.git
```

The resulting directory qemu is your QEMU root directory.

## Simple build and test

QEMU supports builds in this directory (not recommended) or in an extra directory (out-of-tree builds, recommended). There can be any number of out-of-tree builds, so if you plan to make cross builds, debug and release builds, out-of-tree builds are what you need.

### Here is my typical build scenario:

```
# Switch to the QEMU root directory.
cd qemu
# Prepare a native debug build.
mkdir -p bin/debug/native
cd bin/debug/native
# Configure QEMU and start the build.
../../configure --enable-debug
make
# Return to the QEMU root directory.
cd ../../..
```

### Now let's start a simple test:

```
bin/debug/native/x86_64-softmmu/qemu-system-x86_64 -L pc-bios
```

This test runs the QEMU system emulation which boots a PC BIOS.

## Simple build and test with KVM

This example will show an in-tree build.

```
# Switch to the QEMU root directory
cd qemu
# Configure QEMU for x86_64 only - faster build
./configure --target-list=x86_64-softmmu --enable-debug
# Build in parallel - my system has 4 CPUs
make -j4
```

## Getting ready to install a guest OS in a VM:

```
# Create a disk for the VM
./qemu-img create -f qcow2 test.qcow2 16G
# Download an install ISO - I have Fedora 20
ls -la Fedora-Live-Desktop-x86_64-20-1.iso
-rwxr-xr-x. 1 xxxxx xxxxx 999292928 May  4 16:32
```

## Run QEMU with KVM enabled (w/o VNC):

If you have gtk2-devel installed, this will launch a simple UI and you can install your OS.

```
x86_64-softmmu/qemu-system-x86_64 -m 1024 -enable-kvm \
-drive if=virtio,file=test.qcow2,cache=none \
-cdrom Fedora-Live-Desktop-x86_64-20-1.iso
```

## Run QEMU with KVM enabled (with VNC):

If you prefer VNC, try this:

```
x86_64-softmmu/qemu-system-x86_64 -m 1024 -enable-kvm \
-drive if=virtio,file=test.qcow2,cache=none \
-cdrom Fedora-Live-Desktop-x86_64-20-1.iso \
-vnc :1
```

Connect using your favorite VNC viewer to localhost:1 and install your OS.

## Cross builds

Cross building for non-native architectures is quite common. TODO: add description.

## Native builds

# Running QEMU on Linux

## System emulation

All QEMU system emulation should be working.

## User mode emulation

User mode emulation is also supported.

## Links

Retrieved from "<http://wiki.qemu.org/Hosts/Linux>"

---

- Content is available under GNU Free Documentation License 1.2.  
QEMU is a trademark of Fabrice Bellard.  
Hosting generously provided by OSUOSL.