qemu - система для эмуляции компьютера, поддерживающего разные архитектуры.

qemu можно использовать в 2 режимах:

1) режим виртуализации. Позволяет выполнять код виртуальной ОС на процессоре реальной машины(о нём и пойдёт речь)

2) Режим, позволяющий выполнять код одной архитектуры, на процессоре другой архитектуры. Например на intell процессоре запускать программы, скомпилированные для arm или mips

Синтаксис для вызова qemu.
qemu [options] [disk_image]

Основные опции, которые будут использованы далее:
-net nic создаёт сетевую карту на виртуальной ОС. за -net nic через запятую так же можно передать нужные параметры(к примеру MAC адрес или номер VLAN)
-m ARG - задаёт размер оперативной памяти

Будет рассмотрено 2 способа сетевого взаимодействия qemu:

1) user mode networks.

2) tap interfaces.

USER MODE NETWORKS

Если не указано никаких опций qemu по умолчанию будет эмулировать одну сетевую карту. Причём виртуальная ОС будет видеть виртуальный DHCP сервер, расположеный на адресе 10.0.2.2(сервер будет раздавать адреса начиная с 10.0.2.15). Виртуальный DNS сервер будет доступен на 10.0.2.3 и SAMBA сервер расположится по адресу 10.0.2.4. Изменить конфигурацию можно с помощью опции -net user перечисляя далее все нужные нам параметры через запятую(за подробностями в qemu(1))

Пример:
qemu -m 64 -net nic -net user,net=192.168.1.0/24,dns=192.168.1.1 ./openbsd.vdi

запускаем виртуальную машину расположеную в openbsd.vdi, наделяем её 64 мегабайтами оперативной памяти, помещаем машину в локальную сеть 192.168.1.0, DNS сервер будет распологаться по адресу 192.168.1.1.

Плюсы:

1) Простая конфигурация

Минусы:

1) Нет прямого взаимодействия с остальными виртуальными машинами.

2) Не поддерживает входящие соединения.

3) Поддерживает только протоколы TCP и UDP, поэтому ping и другие ICMP утилиты не будут корректно работать.

Так же с помощью -net user существует возможность редирректить порты с ОС хоста на виртуальную ОС

Пример:
qemu -m 64 -net nic -net user,hostfwd=tcp:188.168.110.235:5555-10.10.10.10:80 ./openbsd.vdi
перенаправляем траффик с IP адреса 188.168.110.235 хост машины и порта 55555 на IP 10.10.10.10 и порт 80 виртуальной

TAP INTERFACES

Для большей гибкости можно построить виртуальную сеть с помощью TAP.

TAP - виртуальный интерфейс канального уровня, симулирующий работу ethernet устройства.
Создать TAP интерфейс можно с помощью tunctl(8). Пример:
tunctl -u someuser

Создаёт tap интерфейс от имени someuser.

Уже существующий TAP интерфейс можно подключить к виртуальной машине с помощью опции -net tap (параметры передаются через запятую).
Пример:

qemu -m 64 -net nic -net tap,ifname=tap0,script=no openbsd.vdi

tap0 - имя созданного с помощью tunctl интерфейса с помощью параметров script/downscript можно указывать скрипты для конфигурации/деконфигурации интерфейсов примеры скриптов есть на http://en.wikibooks.org/wiki/QEMU/Networking
Дополнительно можно объединить несколько tap интерфейсов с помощью bridge(для создания bridge интерфейса используется brctl(8)).

Плюсы:

1) Большая гибкость настройки(по сравнению с user mode networks)
2) Нет ограничений на протоколы

Минусы:

1) чуть более сложная конфигурация

P.S.: для доступа во внешние сети(интернет) с виртуальных ОС, на хосте нужно включить NAT(NAPT) (для linux необходимо ввести iptables -A POSTROUTING -j MASQUERADE)