QEMU

http://www.qemu.com/

QEMU

- Emulator procesora
- Autor: Fabrice Bellard
- Obsługiwane platformy: Windows, Solaris, Linux, FreeBSD, Mac OS X
- Aktualna wersja: 0.9.0
- Większość programu oparta na licencji LGPL, a sama część emulująca na GNU GPL

QEMU

Target CPU	User emulation	System emulation
x86	OK	OK
x86_64	Not supported	OK
ARM	OK	OK
SPARC	OK	OK
SPARC64	Dev only	Dev only
PowerPC	OK	OK
PowerPC64	Not supported	Dev only
MIPS	OK	OK
m68k (Coldfire)	OK	OK
SH-4	Dev only	Dev only
Alpha	Dev only	Dev only

Host CPU	Status
x86	OK
x86_64	OK
PowerPC	OK
Alpha	Testing
Sparc32	Testing
ARM	Testing
S390	Testing
MIPS	Testing
Sparc64	Dev only
ia64	Dev only
m68k	Dev only

QEMU

Tryby pracy:

- Emulacja pełnego systemu (Full System Emulation)
- Emulacja trybu użytkownika (User Mode Emulation)

QEMU

Emulacja pełnego systemu

- emulacja pełnego komputera, przeważnie PC - wraz z procesorem i urządzeniami peryferyjnymi
- używany np. do uruchamiania różnych systemów operacyjnych lub debugowania kodu systemowego (wirtualna maszyna może być łatwo zatrzymana, a jej stan skontrolowany, zapisany lub wznowiony)

QEMU

Emulacja trybu użytkownika:

- pozwala na uruchamianie procesów linuxowych skompilowanych dla jednego procesora na innym procesorze (tylko, gdy Linux jest hostem)
- może być użyty dla sprawdzenia wyników działania cross-kompilatorów

QEMU

Podsystemy:

- emulator CPU
- emulowane urządzenia (karta VGA, mysz, klawiatura, port równoległy, dysk twardy itd.)
- 'generic devices' (block devices, character devices) - służą do podłączenia emulowanych urządzeń do odpowiadających im urządzeń hosta
- debugger
- interfejs użytkownika

QEMU

Dynamiczna translacja:

- Technika poprawiania wydajności
- Instrukcje emulowanego procesora są w trakcie wykonania zamieniane na odpowiadające im instrukcje hosta
- Otrzymany kod binarny jest przechowywany w cache'u (translation cache o rozmiarze 16MB), aby mógł być ponownie użyty
- Korzyść w porównaniu z interpretacją – instrukcje są sprowadzane i odkodowywane tylko raz

QEMU

Technika mikrooperacji:

- Instrukcje emulowanego procesora rozdzielane są na kilka prostszych instrukcji nazywanych mikrooperacjami
- Każda z tych mikrooperacji jest implementowana jako osobny mały kawałek kodu w C. Później ten kod jest kompilowany przez GCC do pliku .o
- Kod maszynowy, powstały po jego skompilowaniu, może juz być wielokrotnie używany do tłumaczenia w miejscu danej instrukcji
- Mikrooperacje są w taki sposób wybrane, aby ich ilość była dużo mniejsza niż ilość wszystkich kombinacji operacji i operandów w emulowanym procesorze
- Wydajność i przenośność na inne platformy

QEMU

Acceleration Module:

- Dostępny jest moduł *kqemu*, zwiększające szybkość emulacji komputera PC na innym komputerze PC z architekturą procesora x86 (około 5-krotnie)
- znaczna część kodu działającej aplikacji uruchamiana od razu na procesorze hosta, emuluje się tylko instrukcje trybu jądra i trybu rzeczywistego
- Od 6 lutego 2007 *kqemu* jest również dostępny na licencji GNU GPL (wcześniej moduł ten był darmowy, ale autor nie chciał udostępnić kodu źródłowego)

QEMU

Acceleration Module c.d.:

- QEMU bez kqemu –> 10-20% prędkości natywnej
  QEMU + kqemu –> 50-100% prędkości natywnej
- *kqemu* przeznaczony na razie wyłącznie dla systemów Linux i Windows
- Równolegle rozwijał się też otwarty odpowiednik tego modułu, *qvm86*, jednak w związku z wydaniem programu VirtualBox na licencji GNU GPL, projekt został zamknięty na początku 2007 roku

QEMU

Zalety:

- wsparcie dla wielu architektur
- szybkość (niektóre aplikacje działają prawie jak na rzeczywistym sprzęcie)
- możliwość zapisywania i wznawiania stanu maszyny
- obsługa snapshotów
- pełna otwartość kodu
- wsparcie dla architektury wieloprocesorowej (SMP)
- możliwość zdalnej pracy na emulowanych maszynach za pomocą zintegrowanego serwera VNC
- wsparcie dla USB (-usb -usbdevice tablet)
- emulacja wirtualnej karty sieciowej

QEMU

Zalety c.d.:

- wiele formatów obrazów dysków twardych np. qcow, vpc, wmdk
- implementacja formatu Copy-On-Write pozwala zadeklarować wielogigabajtowy dysk wirtualny, a obraz dysku będzie na tyle duży, na ile aktualnie jest to wymagane
- wirtualny procesor jest biblioteką (libqemu), która może być wykorzystana w innych projektach
- pełna obsługa wyjątków i przerwań
- brak konieczności modyfikacji/łatek na system operacyjny gościa
- pełna kontrola z linii poleceń

QEMU

Wady:

- niepełne wsparcie dla systemu Windows w roli hosta
- niekompletne wsparcie dla mniej popularnych architektur
- trudniejszy w użytkowaniu niż inne emulatory
- brak specjalnych sterowników dla emulowanych systemów
- wymaga X11 i SDL
- nie do końca pełna obsługa rozkazów architektury x86
- brak obsługi IPC syscall
- nie kompiluje się z nowszymi wersjami GCC (4.x)

QEMU

Qemu-Launcher:

- nakładka graficzna oparta o bibliotekę Gtk+, napisana przez Erik'a Meitner'a oraz Linas'a Žvirblis'a

QEMU

Przykład użycia (pod Linuxem):

- utworzenie wirtualnego dysku
  ```
  qemu-img create -f qcow dysk.img 800M
  ```
- uruchomienie systemu
  ```
  qemu -hda dysk.img -cdrom cdrom.iso -boot d
  -m 256
  ```

QEMU

Przykład użycia (pod Windowsem):

- Qemu Manager
  http://www.davereyn.co.uk/