QEMU

- Stosunkowo szybki emulator procesora, korzystający z tzw. dynamicznej translacji i kilku innych ciekawych technik programistycznych i optymalizacyjnych.

- **działa na procesorach procesorach:** Intel x86, AMD64, IA-64, PowerPC, Alpha, SPARC 32 and 64, ARM, S/390, M68k
- **emuluje procesory:** Intel x86, AMD64, ARM, SPARC 32 and 64, PowerPC, MIPS
- **dostępne dla s.o. hosta:** Windows, Linux, OS X, FreeBSD, BeOS
- gość osiąga zwykle 10-20% prędkości hosta

Dwa tryby pracy

- emulacja pełnego systemu (włączając procesor i urządzenia peryferyjne) Może być używany do uruchamiania różnych systemów operacyjnych bez restartowania systemu bądź np. debugowania kodu systemowego

- emulacja trybu użytkownika (tylko, gdy Linux jest hostem) Potrafi uruchamiać procesy linuxowe skompilowane dla jednego procesora na innym procesorze. Może być użyty dla przeprowadzania cross-kompilacji, cross-debugowania.

Zalety QEMU

- wsparcie dla wielu architektur
- można zadeklarować wielogigabajtowy wirtualny napęd (drive), a obraz dysku będzie dokładnie tak duży, jak wymaga tego aktualne użycie
- można trzymać zapis („snapshot”) systemu gościa, a zmiany zapisywać zmiany do oddzielnego pliku obrazu.
- możliwość emulacji karty sieciowej
- wsparcie dla SMP (architektury wieloprocesorowej)
- narzędzia konsoli pozwalają w pełni kontrolować qemu
- wsparcie dla USB
- pełna obsługa wyjątków i przerwań

Wady QEMU

- niepełne wsparcie dla MS Windows
- nie wspiera mniej popularnych platform sprzętowych
- wymaga X11 i SDL (X11 - system okienkowy, obsługujący też urządzenia wejściowe, SDL - Simple DirectMedia Layer - biblioteka ułatwiająca pisanie gier i programów multimedialnych)
- nie do końca pełna obsługa rozkazów architektury x86

Dynamiczna translacja

- Technika poprawiania wydajności
- Dynamiczny translator, w trakcie wykonywania kodu, dokonuje konwersji instrukcji emulowanego procesora na zbiór instrukcji hosta
- Otrzymany kod binarny jest przechowywany w cache`u, tak, że może być używany wielokrotnie
- Korzyść w porównaniu z interpreterem jest taka, że instrukcje są sprowadzane i odkodowywane tylko raz
- Ciekawym ulepszeniem zastosowanym w QEMU jest łączenie kawałków kodu maszynowego generowanego offline przez kompilator C – technika mikrooperacji

Technika mikrooperacji

- Instrukcje emulowanego procesora rozdzielane są na kilka prostszych instrukcji nazywanych mikrooperacjami.
- Każda z nich jest implementowana przez mały kawałek kodu w C. Ten niewielki kod źródłowy w C jest kompilowany przez GCC.
- Kod maszynowy, powstały po jego skompilowaniu, może juz być wielokrotnie używany do tłumaczenia w miejscu danej instrukcji.
- Osiągamy korzyści w wydajności i przenośności na inne platformy

Samomodyfikujący się kod

- W architekturze x86 aplikacje nie mają mechanizmu sygnalizacji, że kod jest modyfikowany.
- Podstawowy pomysł - ustawienie odpowiednich stron pamięci hosta na "tylko do odczytu" (`mprotect()`).
- Gdy podejmowana jest próba zapisu na tej stronie, podnoszony jest SEGV.
- QEMU rozpoznaje wtedy, że należy unieważnić cały przetłumaczony kod, składowany na tej stronie i znów umożliwić zapisywanie na niej. Wiemy też, który fragment kodu należy przetłumaczyć raz jeszcze.
- Poprawne i efektywne - dzięki utrzymywaniu listy przetłumaczonych bloków.

Zarządzanie pamięcią

QEMU udostępnia dwa tryby:

- do symulacji MMU gościa używa jednostki zarządzania pamięcią gospodarza (używa mmap()). Działa tak długo, jak emulowany system operacyjny nie używa rejonu zarezerwowanego dla hosta. Działa szybko dość szybko, ale brak rozdzielenia między pamięcią hosta i gościa zagraża bezpieczeństwu danych.

- by móc uruchomić jakikolwiek system, QEMU może używać MMU, tworzonego przez oprogramowanie. W tym trybie tłumaczenie adresu wirtualnego na fizyczny dokonuje się przy każdym dostępie do pamięci. Działa wolniej, ale zapewnia separację przestrzeni adresowej hosta i gościa.

Acceleration Module

- Moduł kqemu (QEMU Accelerator) około pięciokrotnie przyspiesza pracę QEMU, zbliżając się tym samym do prędkości natywnej wykonywania.
- Jest to możliwe, dzięki temu że znaczna część kodu jest uruchamiana od razu na procesorze hosta. Emuluje się tylko instrukcje trybu jądra i trybu rzeczywistego. Zasada działania jest podobna jak u przedstawicieli wirtualizacji (Virtual PC, VMware)
- Na razie przeznaczony tylko dla nowszych wersji Linuxa i Windows 2000/XP

Uruchamianie

- tworzenie pustego wirtualnego dysku

```
qemu-img.exe create -f qcow hda.img 3G
```

- instalowanie systemu

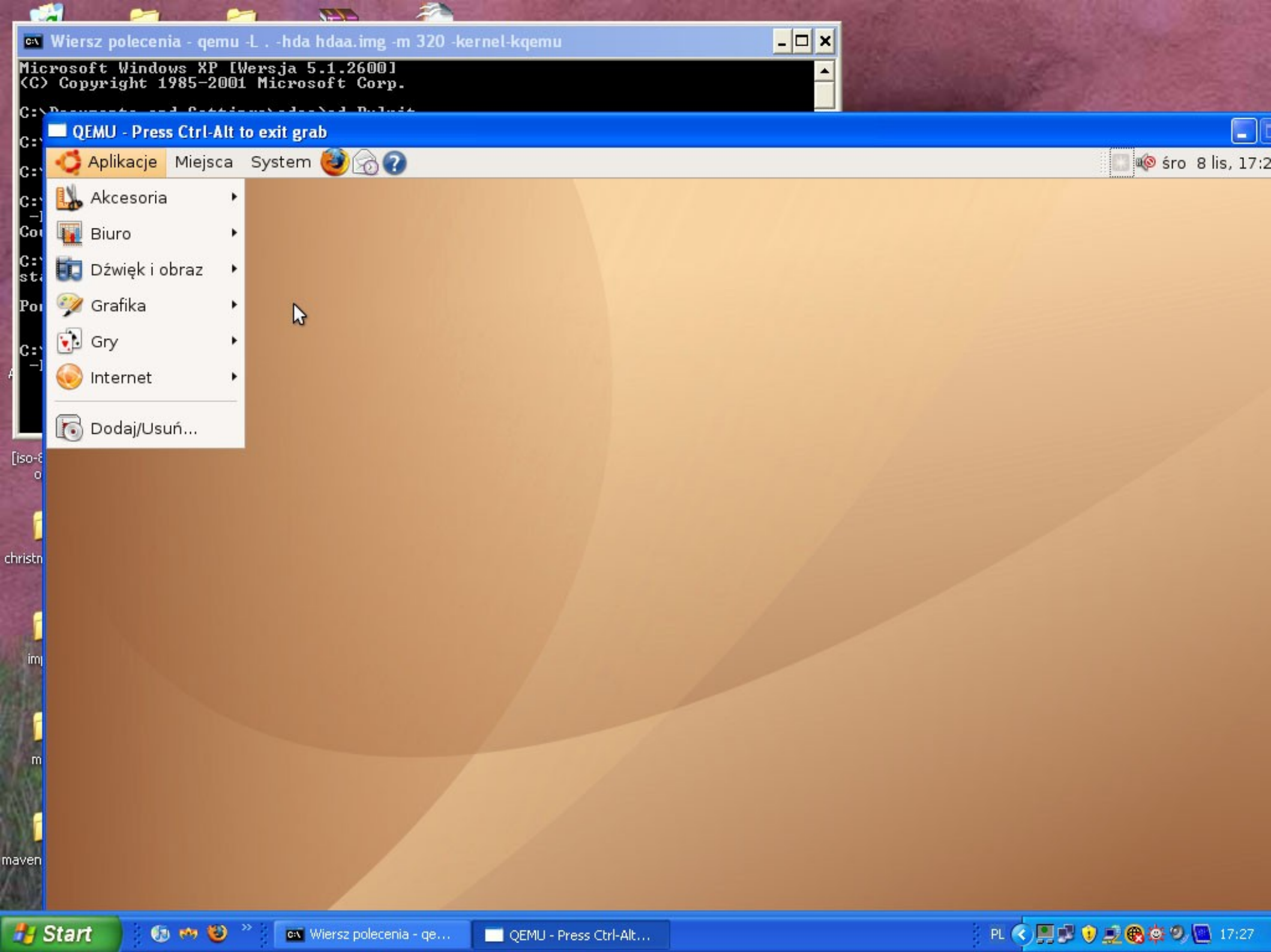
```
qemu.exe -L . -cdrom "\\.\D:" -hda hda.img -m 256 -boot d
qemu.exe -L . -cdrom my_os_install.iso -hda hda.img -m 256
  -boot d
```

- uruchamianie akceleratora

```
net start kqemu
```

- uruchamianie

```
qemu.exe -L . -hda hda.img -m 256 -kernel-kqemu
```


Wiersz polecenia - qemu -L . -hda hdaa.img -m 320 -kernel-kqemu
Microsoft Windows XP [Wersja 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
QEMU - Press Ctrl-Alt to exit grab
Aplikacje
Miejsca
System
śro 8 lis, 17:2
Akcesoria
Biuro
Dźwięk i obraz
Grafika
Gry
Internet
Dodaj/Usuń...
Start
Wiersz polecenia - qe...
QEMU - Press Ctrl-Alt...
PL
17:27

Wiersz polecenia - qemu -l ... -hda hdaa.img -m 320 -kernel-kqemu

QEMU

Aplikacje Miejsca System śro 8 lis, 17:51

Uniwersytet Warszawski - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

http://www.uw.edu.pl/

Google

Getting Started Latest BBC Headlines

190 lat Uniwersytetu Warszawskiego

program obchodów

Strona Uniwersytetu Warszawskiego








wejście / enter

Partnerzy strategiczni: ATENEUM Mercure TEATR POLSKI COLLEGIUM NOBILIUM TEATR Orbis Transport

Partnerzy: fundacja optimi elaturi FUNDACJA UNIWERSYTETU LIBER

Start Wiersz polecenia - qe... zrzut1.bmp - Paint QEMU PL 17:51

Bibliografia

- **QEMU, a Fast and Portable Dynamic Translator** - http://www.usenix.org/publications/library/proceedings/usenix05/tech/freenix/full_papers/bellard/bellard_html/index.html
- **QEMU Documentation** - http://fabrice.bellard.free.fr/qemu/user-doc.html
- **Wikipedia** - http://en.wikipedia.org/wiki/QEMU