WinKVM: Windows Kernel-based Virtual Machine

Kazushi Takahashi, Koichi Sasada
University of Tokyo

About me

- Name:
  - 一志　　高橋
  - Kazushi　Takahashi
- My research area:
  - System software, operating system and virtual machine technology
  - Interested in Linux kernel hacking, distributed system and parallel programming
- Twitter: ddk50
- Blog: http://d.hatena.ne.jp/ddk50/

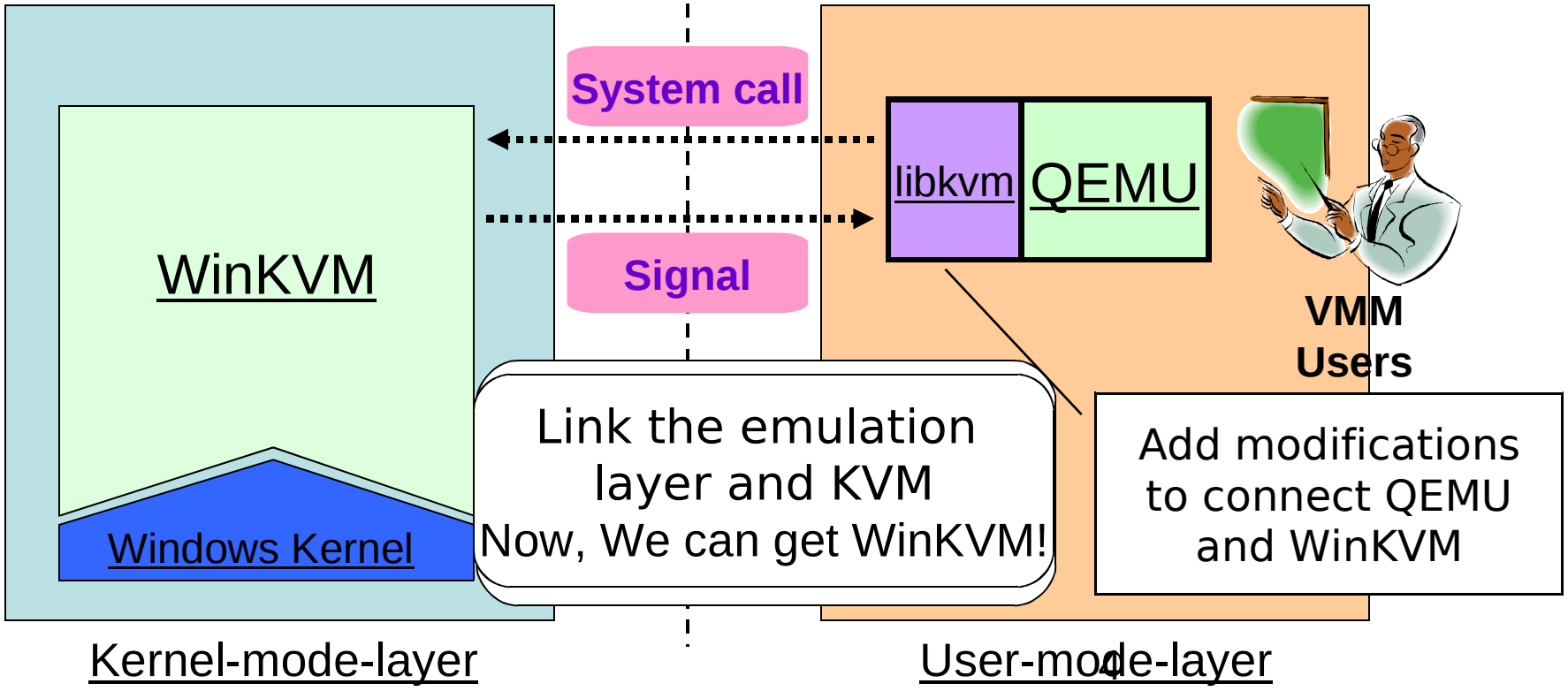
Agenda

- We have implemented WinKVM
  - WinKVM is a port of KVM(-17) to Microsoft Windows.
- Main point of today's talk: "How we developed WinKVM"
  - KVM is implemented as Linux device driver
  - Porting "kvm.ko and intel-kvm.ko" to Windows drivers
  - Developing an emulation layer to run Linux drivers on Windows
  - This emulation layer translates Linux kernel functions into Windows kernel APIs
- Why we develop WinKVM
  - To provide a VMM that supports both Windows and Linux
  - To search for the new way of KVM usage

Overview of Our Method

We implemented a linux emulation layer

- To reduce implementation costs
- To enable any version of KVM to run



Examples of Translated APIs

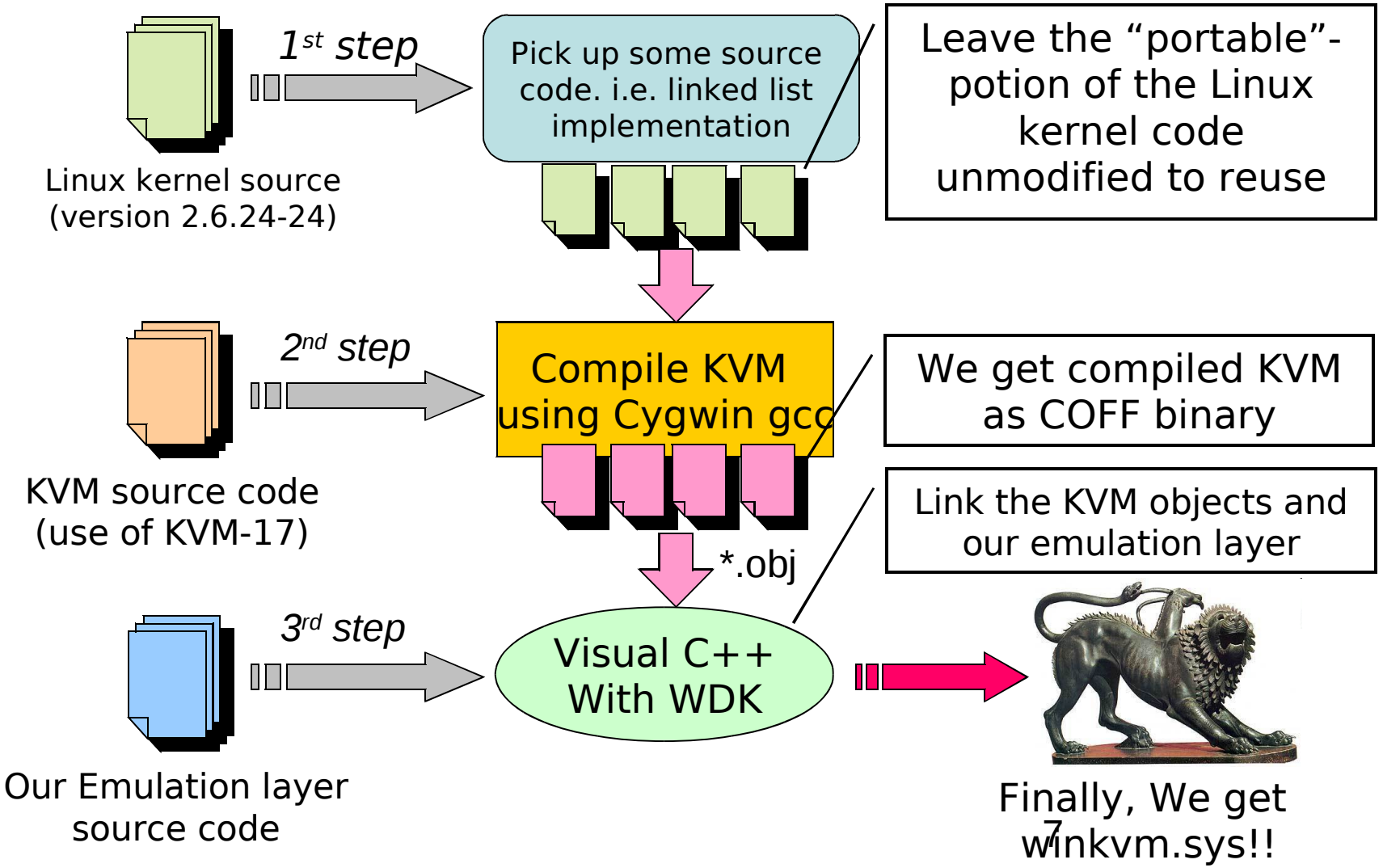
Most of Linux functions have corresponding windows kernel functions

Linux Kernel Functions	Windows Kernel Functions used to emulate the function
kmalloc() kfree()	ExAllocatePoolWithTag() ExFreePoolWithTag()
mutex_init() mutex_lock() mutex_unlock() mutex_trylock()	ExInitializeFastMutex() ExAcquireFastMutex() ExReleaseFastMutex() ExTryToAcquireFastMutex()
alloc_page() free_page()	ExAllocatePoolWithTag() ExFreePoolWithTag()
__va() __pa()	MmGetVirtualForPhysical() MmGetPhysicalAddress()

Two Technical Problems

1. Difference in binary formats (compilers):
   - How to link our emulation layer and KVM drivers
   - KVM source code (inline-asm) depends on GCC
   - Windows driver developers have to use Visual C++
   - Visual C++ CANNOT compile KVM source code
1. Difference in memory architectures:
   - KVM driver and QEMU share guest OS memory region
   - Both OSs support memory sharing between kernel and user memory space
   - Difficulties in emulating Linux memory interfaces by Windows kernel
   - The fault (nopage) handler, which is used by KVM is not supported by Windows.

How to handle the difference in binary formats



How to handle the differences in memory architectures (1/2)

- Problem:
  - The fault handler in Linux is difficult to emulate by Windows kernel functions
- Solution:
  - Modify KVM source code to avoid use of the fault handler
  - Only 1-line modification

How to handle the differences in memory architectures (2/2)

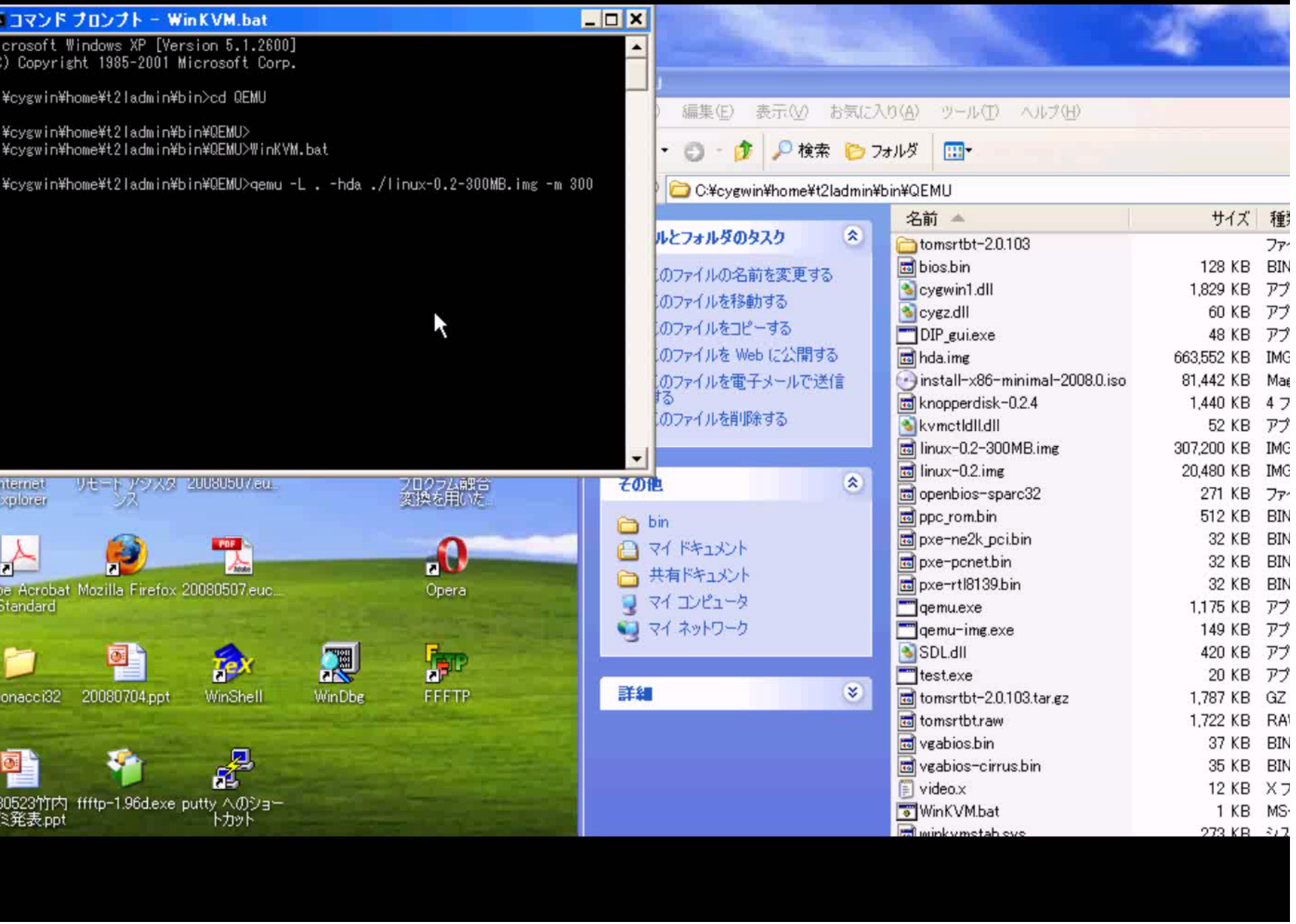
- The mechanism of this modification
  1. Before starting KVM emulation, our emulation layer construct memory mapping regions between kernel and user-space. The layer has already mapped GPA to HVA
  2. When KVM itself also tries to map GPA to HVA, our patch overwrites the mapping with our emulation layer mapping
  3. Never call the fault (nopage) handler in KVM

Latest KVM may not need this modification

DEMO

Does our method work well?

In this demo, we execute on Linux kernel 2.6.20 attached to QEMU, and execute some applications


コマンド プロンプト - WinKVM.bat
crosoft Windows XP [Version 5.1.2600]
) Copyright 1985-2001 Microsoft Corp.
¥cygwin¥home¥t2ladmin¥bin>cd QEMU
¥cygwin¥home¥t2ladmin¥bin¥QEMU>
¥cygwin¥home¥t2ladmin¥bin¥QEMU>WinKVM.bat
¥cygwin¥home¥t2ladmin¥bin¥QEMU>qemu -L . -hda ./linux-0.2-300MB.img -m 300
編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
検索 フォルダ
C:¥cygwin¥home¥t2ladmin¥bin¥QEMU
名前 サイズ
tomsrtbt-2.0.103
bios.bin 128 KB
cygwin1.dll 1,829 KB
cygz.dll 60 KB
DIP_gui.exe 48 KB
hda.img 663,552 KB
install-x86-minimal-2008.0.iso 81,442 KB
knopperdisk-0.2.4 1,440 KB
kvmctldll.dll 52 KB
linux-0.2-300MB.img 307,200 KB
linux-0.2.img 20,480 KB
openbios-sparc32 271 KB
ppc_rom.bin 512 KB
pxe-ne2k_pci.bin 32 KB
pxe-pcnet.bin 32 KB
pxe-rtl8139.bin 32 KB
qemu.exe 1,175 KB
qemu-img.exe 149 KB
SDL.dll 420 KB
test.exe 20 KB
tomsrtbt-2.0.103.tar.gz 1,787 KB
tomsrtbt.raw 1,722 KB
vgabios.bin 37 KB
vgabios-cirrus.bin 35 KB
video.x 12 KB
WinKVM.bat 1 KB
ルとフォルダのタスク
このファイルの名前を変更する
このファイルを移動する
このファイルをコピーする
このファイルを Web に公開する
このファイルを電子メールで送信する
このファイルを削除する
その他
bin
マイ ドキュメント
共有ドキュメント
マイ コンピュータ
マイ ネットワーク
詳細
Mozilla Firefox
20080507.euc...
Opera
20080704.ppt
WinShell
WinDbg
FFFTP
ffftp-1.96d.exe
putty へのショートカット

Future Work

- Overcome guest memory limitation
  - Max 300MB guest memory from 2Gbyte physical memory
  - We are able to solve this problem
    - Modify QEMU to gather scattered memory chunks as a single guest memory space
- Add new functions to the emulation layer
  - Implement SMP functions such as smp_call_function()
  - Catch up the latest version of KVM
    - Nested paged KVM
    - PCI Pass-through
- Debug :-(

Summary

- We have implemented WinKVM
  - A port of KVM(-17) to Microsoft Windows
- Main point of today's talk: "How we developed WinKVM"
  - We implemented an emulation layer to run Linux drivers on Windows
  - We developed WinKVM using this emulation layer

Thank you for your attention!

Have a look at WinKVM repository in GitHub

http://github.com/ddk50/