15-410
“...1337C0DE...”

Project 4 - “PebPeb”
Nov. 20, 2017

Dave Eckhardt
Dave O'Hallaron
Synchronization

**Deadline reminders**

- P3extra deadline != P4 deadline
  - This is for real, not some oversight
- Don't forget about the book report...
  - Hand-in directories have been created
Acknowledgments

Leadership
- Joshua Wise, Wes Filardo, Ben Blum, Reid Long

Reference host, reference guests
- Elly Jones, Michael Sullivan, Reid Long

Payloads
- Josiah Boning, Matt Schnall, Ryan Pearl, Reid Long

Inspirational artwork
- Chris Lu

Inspirational quote
- “The most fun I've had doing systems programming since I took 410.” – Jacob Potter
Outline

P4
  - Write a hypervisor!

Impossible!?!?
  - “Some restrictions apply”

“A picture is worth 1000 words”
  - I'll still need to talk fast

Component inventory

Suggestions
How can 2 students re-invent VMware in 10 days??

- Page-table tracing
- Page-table *shadowing*
  - Shadow page-table *caching*
- Binary translation of POPF
- Decoding/simulation of key instructions
  - MOVL %EAX, %CR3
  - MOVL %EBX, %CR3
  - ...etc.
- Virtual interrupts and exceptions
- Decoding I/O port operations into device-I/O requests
How can 2 students re-invent VMware in 10 days??

- Page-table tracing
- Page-table *shadowing*
  - Shadow page-table *caching*
- Binary translation of POPF
- Decoding/simulation of key instructions
  - MOVL %EAX, %CR3
  - MOVL %EBX, %CR3
  - ...etc.
- Virtual interrupts and exceptions
- Decoding I/O port operations into device-I/O requests

You probably can't
Scope

How can 2 students re-invent VMware in 10 days??

- Page-table tracing
- Page-table **shadowing**
  - Shadow page-table **caching**
- Binary translation of POPF
- Decoding/simulation of key instructions
  - MOVL %EAX, %CR3
  - MOVL %EBX, %CR3
  - ...etc.
- Virtual interrupts and exceptions
- Decoding I/O port operations into device-I/O requests

You probably can't

- Hey, remind me what are the **kinds** of virtualization?
Paravirtualization is smaller!

Virtualization Paravirtualization

- Page-table conversion
- Page-table shadowing
  - Shadow page-table caching
- Binary translation of POPF
- Decoding/simulation of key instructions
  - MOV $EAX, %CR3
  - MOV $EBX, %CR3
  - ...etc.
- Virtual interrupts and exceptions
- Decoding I/O port operations into device I/O requests
  I/O hypercalls
Paravirtualization

- Page-table tracing
- Virtual interrupts and exceptions
- I/O hypercalls
- Ok, “miscellaneous” hypercalls too
Paravirtualization

Paravirtualization - “PebPeb” Hypercalls

- Page-table tracing
  - \texttt{hv\_setpd()}, \texttt{hv\_adjustpg()}
- Virtual interrupts and exceptions
  - \texttt{hv\_setidt()}, \texttt{hv\_\{disable,enable\}_interrupts()}
  - \texttt{hv\_iret()}
- I/O hypercalls
  - \texttt{hv\_print()}, \texttt{hv\_cons\_set\_cursor\_pos()}, ...
- Ok, “miscellaneous” hypercalls too
  - \texttt{hv\_magic()}
  - \texttt{hv\_exit()}
Picture Time!

**Logical (“marketing”) pictures**
- Lots of colored boxes, everybody's happy

**Illogical (“physical”) pictures**
- Maybe too many boxes
15-410 Virtual Memory Layout

Kernel Data

Kernel Program

Program

Stack

Stack

Stack

Stack

Program

Program

Program

Program

4080 MB

16 MB

[Logical]
15-410 Physical Memory Layout

User Memory: 240 MB

Kernel Memory: 16 MB

[Physical]
Host Kernel, Host User Programs

[Logical]

Red: supervisor-only

HK

HU

HU
Guest Kernel “Sees” Guest User
Guest User Stands Alone
\{\text{Host}, \text{Guest}\} \times \{\text{Kernel}, \text{User}\}

Both kernels live under \text{USER\_MEM\_START}
The “Real” (not Virtual) Picture

[Physical]

Guest User

Guest Kernel

Host Kernel

16 MB
The “Real” (not Virtual) Picture

Other Stuff

Guest User

Guest Kernel

Host Kernel

16 MB

[Physical]
PebPeb Address Spaces

**Host kernel**
- All of kernel is mapped (0..16M)
- One user program is mapped (17..4096M)

**Host user**
- One user program is mapped
- Kernel is present but invisible until a surprise
PebPeb Address Spaces

**Host kernel**
- All of kernel is mapped (0..16M)
- One user program is mapped (17..4096M)

**Host user**
- One user program is mapped
- Kernel is present but invisible until a surprise

**Guest kernel**
- All of guest kernel is mapped (0..16M)
- One guest user program is mapped (17..4080M)

**Guest user**
- One guest user program is mapped
- Guest kernel is [somewhere] until a surprise
- Host kernel is present but invisible until a surprise
PebPeb Address Spaces

**Host kernel**
- [You know how to do this]

**Host user**
- [You know how to do this]

**Guest kernel**
- How can the guest kernel occupy 0..16M?
  - That range is used by the host kernel!

**Guest user**
- When a guest user program is running, and a surprise happens, the hardware will enter host kernel mode.
  - How do we get into guest kernel mode?
Once Upon a Time...

“Hey, let's launch a guest kernel!”
“Hey, let's launch a guest kernel!”
exec("pathos", argv);
Guest Kernel in “Boot VM” Mode

Both kernels live under USER_MEM_START??
Segmentation to the Rescue!

Guest kernel runs in custom segments!
- B: 16M
- L: 4080M
  - (or a bit less)

Guest sees memory starting at 0

Surprises switch %CS and %SS so host kernel works
Component Inventory

What are the pieces?

- Virtual consoles – independent, really not tricky
- Creating shifted segments – not very hard
- “Boot VM” page-table generator
- Tweaked guest-kernel ELF loader
- HV console-output – straightforward
- Basic hv_iret()
- Delivery of virtual interrupts to guests
  - Timer, then keyboard – this part is significant
  - Result: “P1 guests” run: cool!
- Page-table compiler
  - Conceptually non-trivial
  - Not huge amounts of code
- Advanced hv_iret(), hv_exit(), “mop up”
  - Result: “P3 guests” run: wow!
Grading

All or nothing???
- P4 is actually two different projects
  - Virtual consoles + hosting P1 guests
  - Virtual consoles + hosting P1 guests + hosting P3 guests

The first version is actually fairly educational!
- We expect to award grades around 80% for solid hosting of paravirtualized P1 guests
Summary

P4 – “PebPeb” paravirtualization
Segmentation to the rescue!
Impossible!?!?
- Ok to aim for “P1 guests + virtual consoles”