



# Mouse integration for virtualized legacy operating systems

D. Kostiuk, A. Gusev, Yu.Soyko, I. Degtyarik

# Problems of virtualizing old OS

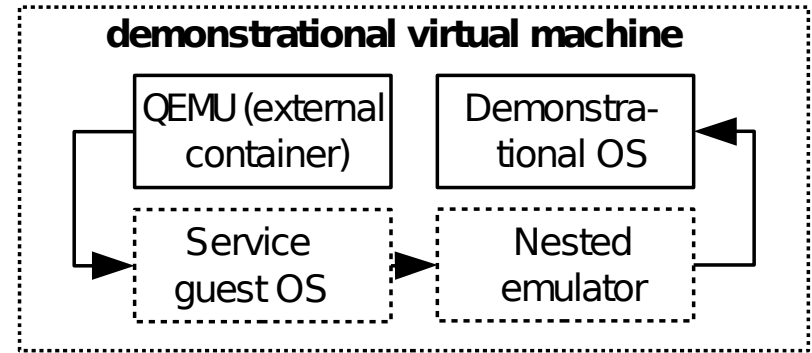
- Find compatible virtualization system / emulator
- Avoid poor user experience with the virtualized system
  - High CPU consumption (no idle CPU cycles in old system)
  - No data exchange with the host system
  - Slow (non-accelerated) graphics
  - Absent drivers for networking, audio, storage devices, etc.
  - Bad UX due to quirky cursor movements and/or mouse lock

# Virtualization for legacy OS

- Mainstream virtualization systems, such as QEMU
  - QEMU supports wide range of architectures SPARC, PowerPC, Motorola 68k, MIPS, ARM...
    - ...but usually not the full ancient computer capable of running ancient OS
    - Absent drivers and mouse integration make user experience not so good
- Dedicated emulators
  - Open source emulators made by fans
  - Official emulators from old SDK
  - Provide mouse integration and do not need any additional drivers
- Unfortunately, dedicated emulators rarely have features typical to mainstream virtualization systems
  - No snapshots
  - No remote access
  - ...

# Nested virtualization

- QEMU (or other mainstream VM) is used as a powerful container with snapshots, etc.
- QEMU runs service guest OS
- This service OS runs some dedicated emulator
- This emulator runs retro OS we want to demonstrate
- Everything works rather well because of very low system resources consumed by the retro system

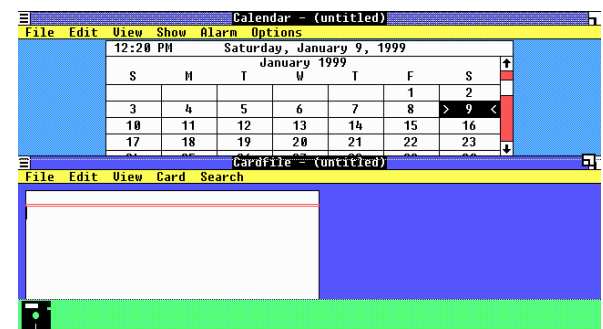
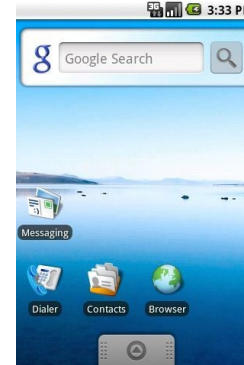
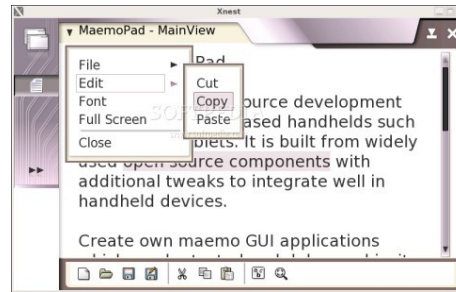
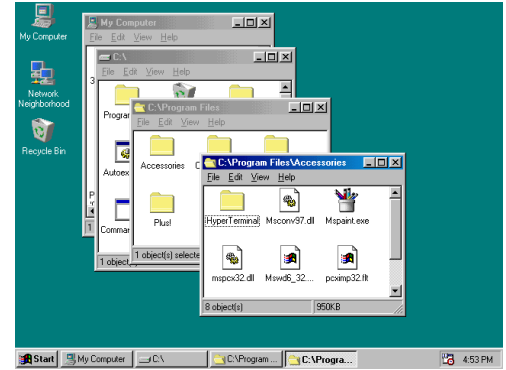
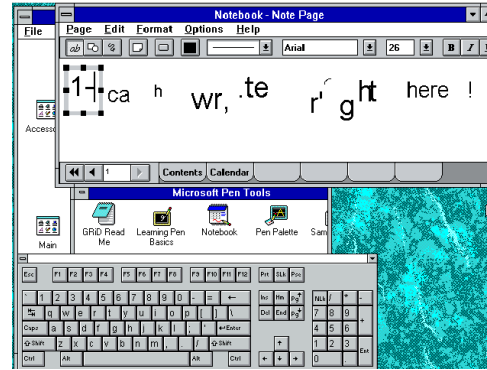


Good open source service OS:

- Linux
- FreeDOS
- ReactOS

# Some retro systems can be virtualized natively...

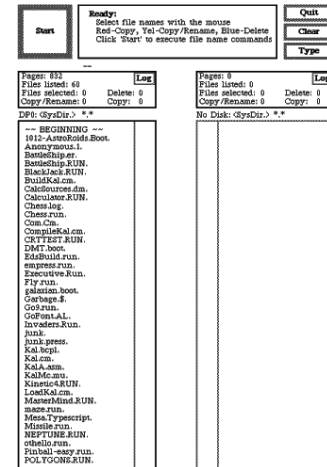
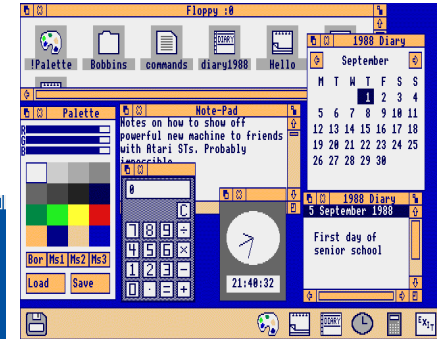
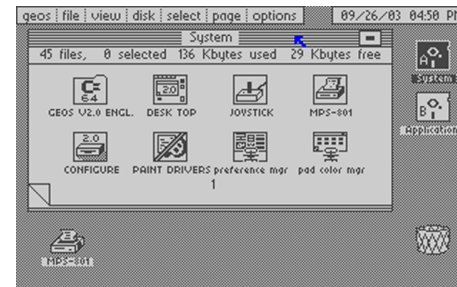
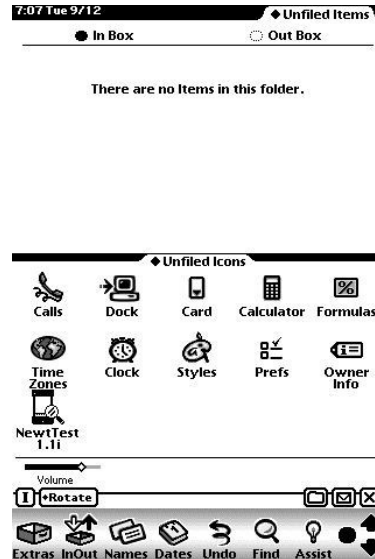
- Examples of systems which don't need nested virtualization:
- Windows 1.x, 2.x, 3.x, 95, ...
- Pen Windows
- GEM from Digital Research
- Maemo
- Android
- WebOS
- ...



# ..some have open source emulators..

Examples of systems which have open source emulators:

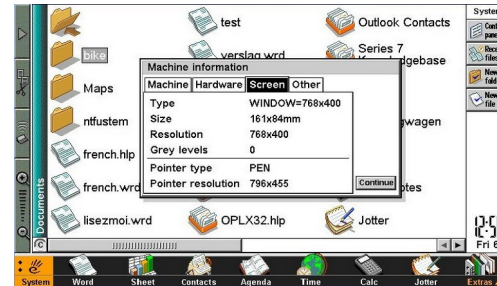
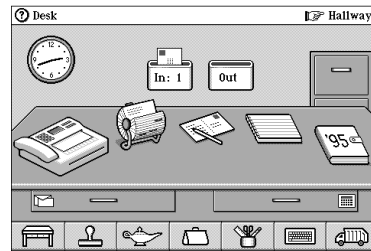
- Xerox Alto (salto)
- GEOS (CC64)
- Amiga (UAE)
- RiscOS (RPCEmu)
- Apple Lisa (lisaem)
- MacOS 1.x (minivmac)
- MacOS 7.x (BasiliskII)
- MacOS X (PearPC)
- Apple Newton (einstein)



# ...but proprietary old emulators are also ok :)

Examples of systems which have proprietary emulators:

- Xerox GlobalView
- Psion EPOC16
- Psion EPOC32
- PalmOS
- Magic Cap
- Windows CE



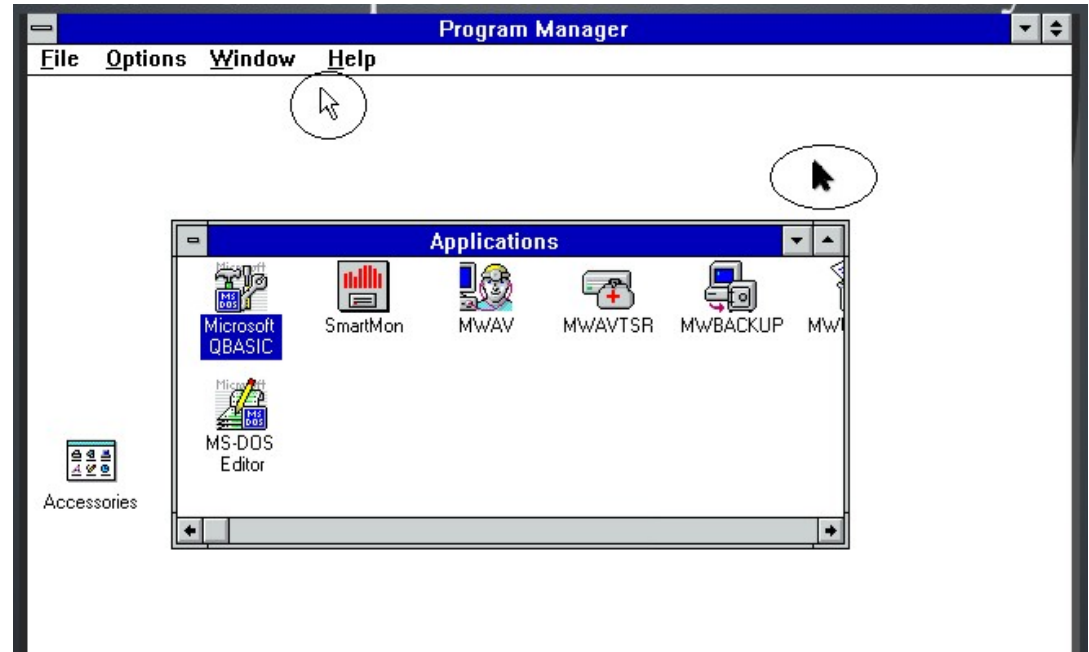
# Native vs. nested virtualization

- Natively virtualized OS
  - can use (awful) mouse lock mode
  - can use mouse integration
- Nested virtualization with dedicated emulator inside
  - needs mouse integration in service OS only...
    - ...which is specially chosen for being compatible with both external VM and emulator...
    - ...so mouse integration just works there



# Guest and host cursor without mouse integration

- Cursor has different speed and acceleration in guest and host systems
- Mouse movements produce different deltas in guest and host systems
- Virtualization system hides host cursor until special key press
- It feels like mouse is locked down in guest system



# Guest and host cursor with mouse integration

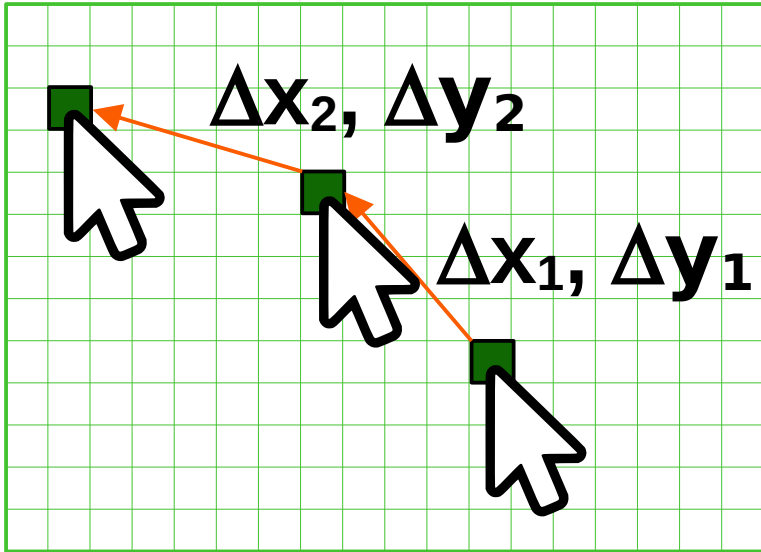
Guest system uses special driver (two variants):

- Guest driver can use absolute cursor coordinates
  - Special absolute positioning device with existing driver can be emulated
- Guest driver can get additional information from VM about the desired cursor movements
  - *Special driver* from the virtualization system vendor is needed *for all guest operating systems*

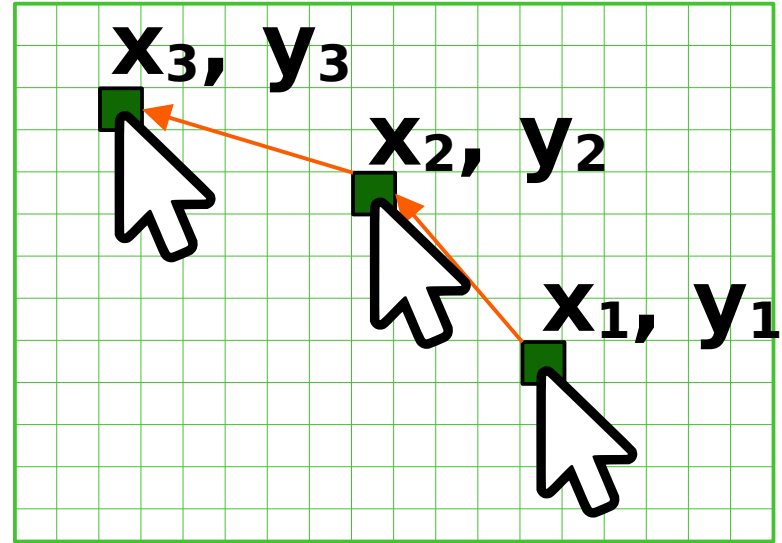
In both cases there's no mouse lock effect as host and guest cursors are in identical positions

# Cursor positioning systems

- Relative positioning (mouse)



- Absolute positioning (tablet)



# USB tablet is typically emulated as an absolute positioning device

- Wacom PenPartner
- Uses USB interface
- Uses Wacom IV protocol
  - same as some earlier models
- Is emulated by QEMU
  - `-device usb-tablet` option
- Is supported by modern operating systems without additional drivers
  - But can do nothing with pre-USB OS :(



# Serial tablet

## Wacom PenPartner CTS 0405-R

- Produced in 1997
- Uses Wacom IV protocol (same as earlier models) via serial port
- Has same name as PenPartner USB tablet emulated by QEMU :)
- Still available on e-bay

Implemented since QEMU 1.9:

`-chardev wctablet,id=somename -serial chardev:somename`



# wctablet compatibility with old OS

Guest OS still needs drivers for such tablet

- Such drivers do exist for DOS, OS/2, Windows 3.x and Windows 95, BeOS, Mac OS 9.x and Mac OS X, AmigaOS
- x86 and x86\_64 are more or less in good shape
  - Windows 3.x and Windows 95 drivers even work :)
  - DOS driver works with a minimal patch
  - BeOS drivers partially work
  - OS/2 driver doesn't recognize the tablet
- qemu-system-ppc just has no appropriate port :)

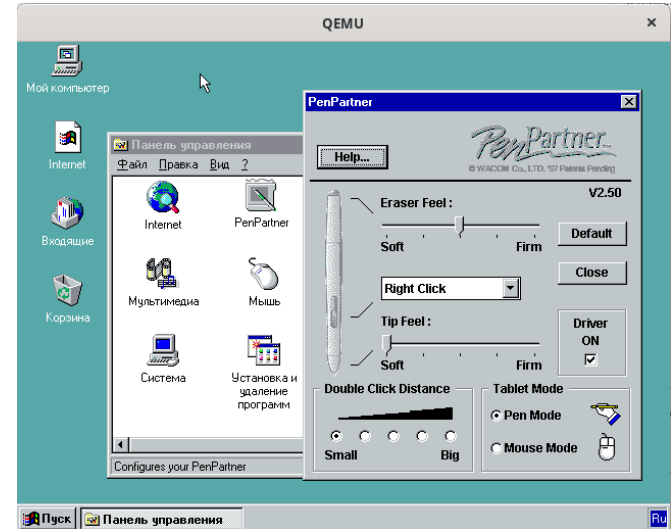
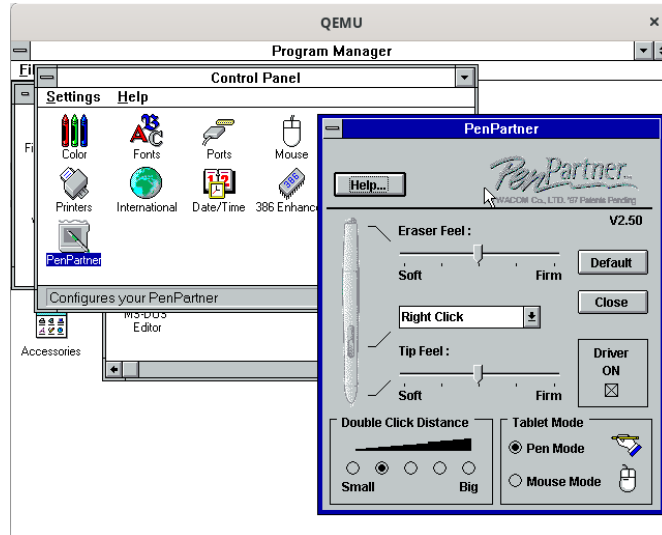
# QEMU versions compatibility

QEMU is known for some compatibility problems across versions:

- Some combinations of guest OS and QEMU version are “unlucky”
  - Guest OS may be partially or totally broken (legacy OS are not so much tested)
  - The reasons are QEMU optimizations, code complexity, etc.
- Specifying VM architecture in command line (like `-M pc-i440fx-1.4`) may help, but not 100%
- Even VM snapshots are not always compatible across versions
  - For example, snapshots compatibility was broken after the version 1.4
  - So it's good idea to have written instruction how to reproduce your snapshot

# Windows compatibility with wctablet in different QEMU versions...

... is not always perfect :)





# wctablet needs the following patch to work with DOS

There is a Wacom mouse driver for DOS and DOS-based GUI (GEM/OpenGEM, etc.)

- It treats eraser as a left mouse button
- It wants high pressure value for a click

```
chardev/wctablet.c | 5 +++-
1 file changed, 4 insertions(+), 1 deletion(-)

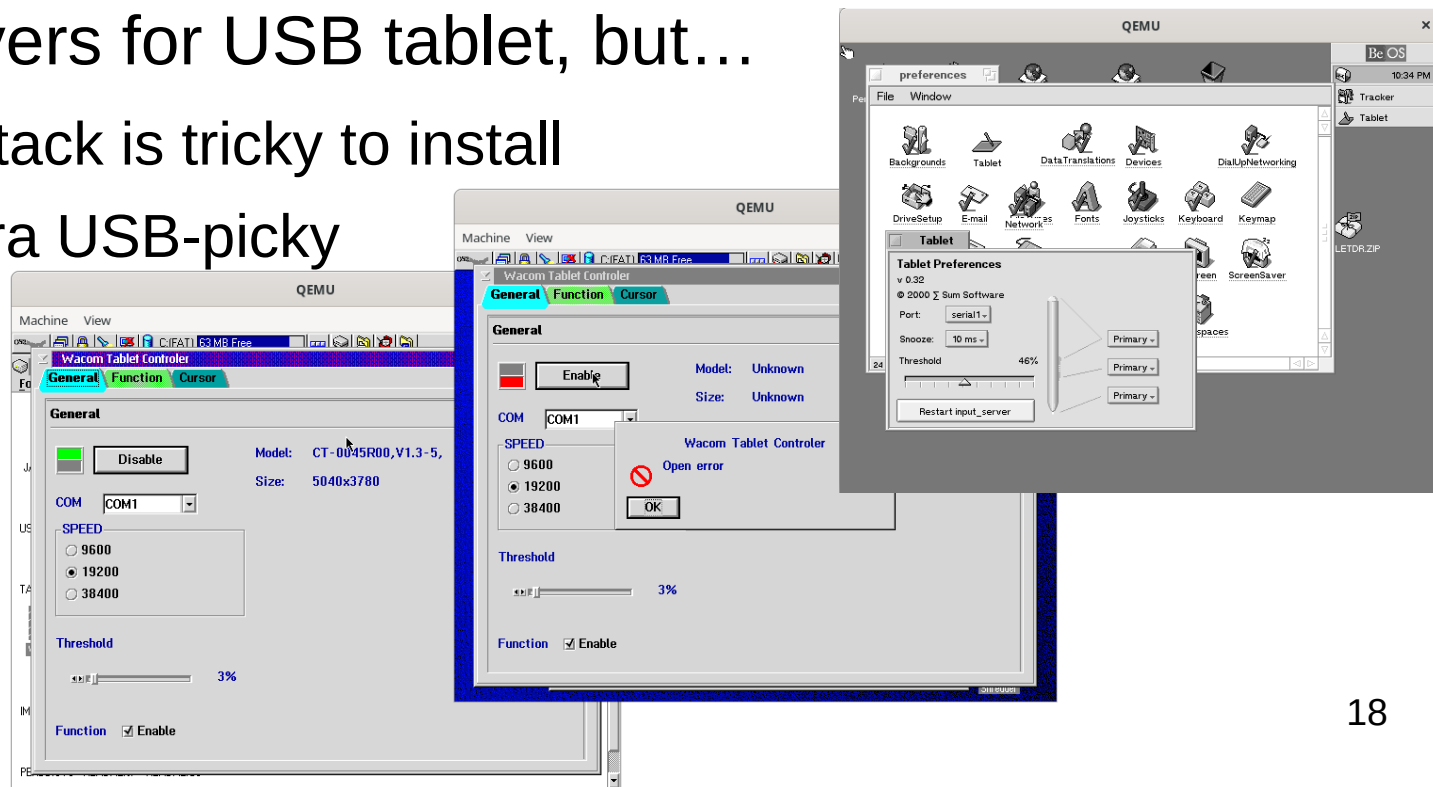
diff --git a/chardev/wctablet.c b/chardev/wctablet.c
index 95e005f5a5..0e53b194d8 100644
--- a/chardev/wctablet.c
+++ b/chardev/wctablet.c
@@ -139,7 +139,10 @@ static void wctablet_queue_event(TabletChardev *tablet)
     codes[5] = codes[5] | WC_L7(nexY);

     if (tablet->btns[INPUT_BUTTON_LEFT]) {
-        codes[0] = 0xa0;
+        //codes[0] = 0xa0;
+        codes[0] |= 0x0c; /* 0x08 (button pressed) + pressure p0 bit */
+        codes[3] |= 0x0c; /* 8 (button number is 1) + pressure p1 bit */
+        codes[6] = 0x3f; /* high positive pressure is applied */
     }

     wctablet_queue_output(tablet, codes, 7);
--
2.30.0
```

# wctablet with OS/2 and BeOS

- Both OS need better tablet response for detection
- There are drivers for USB tablet, but...
  - OS/2 USB stack is tricky to install
  - BeOS is extra USB-picky
  - Haiku already works with usb-tablet



# Instructions and news on wctablet

ostimeline

News About Downloads Projects

## QEMU and mouse integration in Windows 95 and Windows 95 OSR2

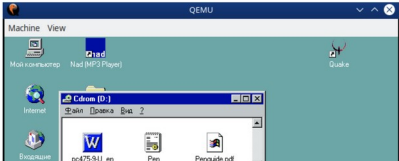
Special wctablet serial device supported by QEMU is able to bring mouse integration in the same way as “-usb-device tablet” (old way) and “-usb -device usb-tablet” (new way) do.

**Note:** It was noticed, that under QEMU 2.11 Windows 95 with serial tablet hangs and thus doesn't work. In earlier and later versions everything is working. Windows 95 OSR2 works with mouse integration in all tested versions.

**The following steps are needed to get mouse integration with Windows 95 and Windows 95 OSR2 running in QEMU:**

1. You will need to install a driver for the Wacom PenPartner serial tabled. You can download the driver installation CD [here](#)
2. Run your virtual machine with the downloaded ISO image in addition to your HDD image with Windows, and add two additional command line parameters. The first one will create a special character device with some arbitrary name you (e.g. “mywctablet1”). The second one will attach this character device to your serial (COM) port:  

```
qemu-system-i386 -m 32 -hda image.qcow -cdrom Wacom\ Pen\ Partner\ \ (Version\ 2.50\)\ (WACOM\)\ (1 997\)\.iso -chardev wctablet,id=somename -serial chardev:somename
```
3. QEMU will use the new wctablet device to control your pointer, but Windows knows nothing about it yet. So use your keyboard (Tab, arrow keys and Enter) for navigaton. Open the needed folder in a file manager, run Setup.exe and proceed with the installation as follows:



ostimeline.org/wctablet