Einstein & Relativity: The Transition

Newton never dies. It just gets new batteries.

Paul Guyot
Worldwide Newton Conference 2006

Einstein & Relativity: The Transition

Newton never dies. It just gets new hardware.

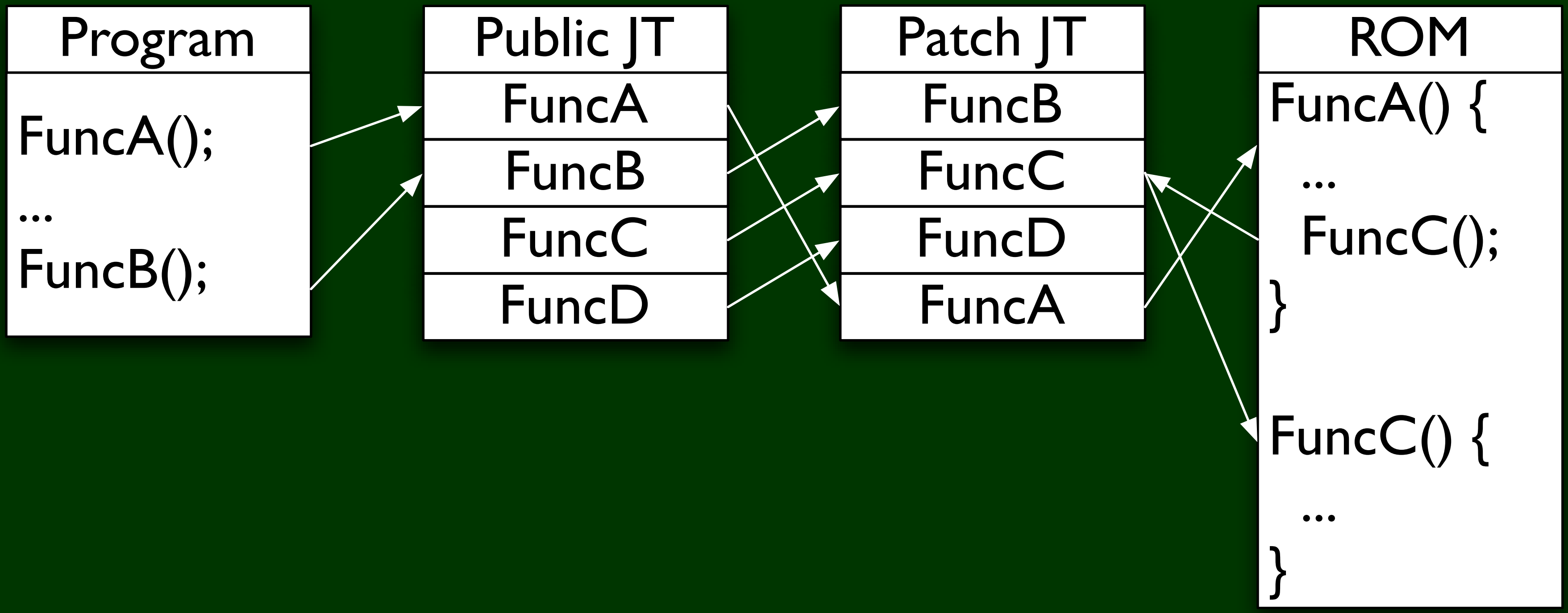
Paul Guyot
Worldwide Newton Conference 2006

1999-2003
From Newton to Einstein

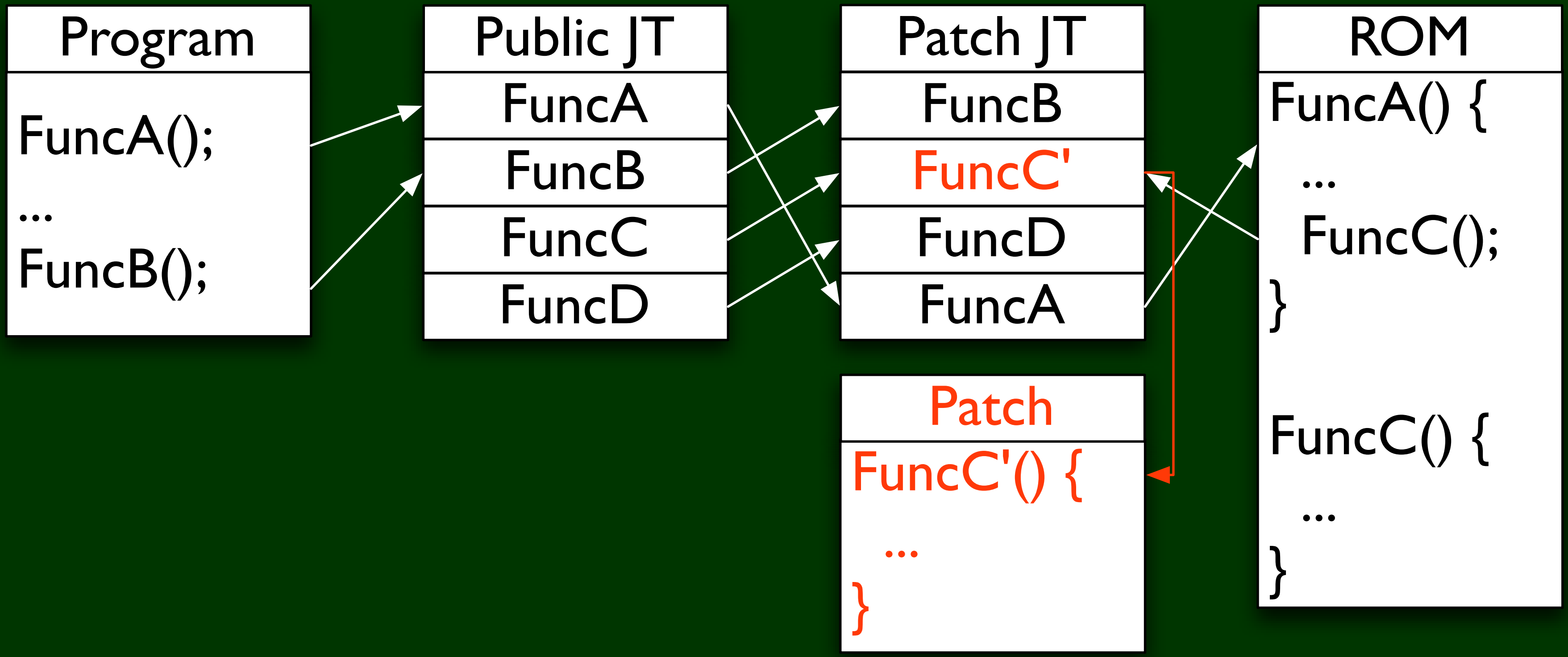
710031 (1999-2000)
The system patch

- First system patch not built by Apple.
- What it does:
  - Fixes a minor bug.
- What I learned:
  - The beautiful complexity of NewtonOS' memory management unit (MMU) and the smartness of the design.

System patches



System patches



Program

FuncA();

...

FuncB();

Public JT

FuncA

FuncB

FuncC

FuncD

Patch JT

FuncB

FuncC'

FuncD

FuncA

Patch

FuncC'() {

...

}

ROM

FuncA() {

...

FuncC();

}

FuncC() {

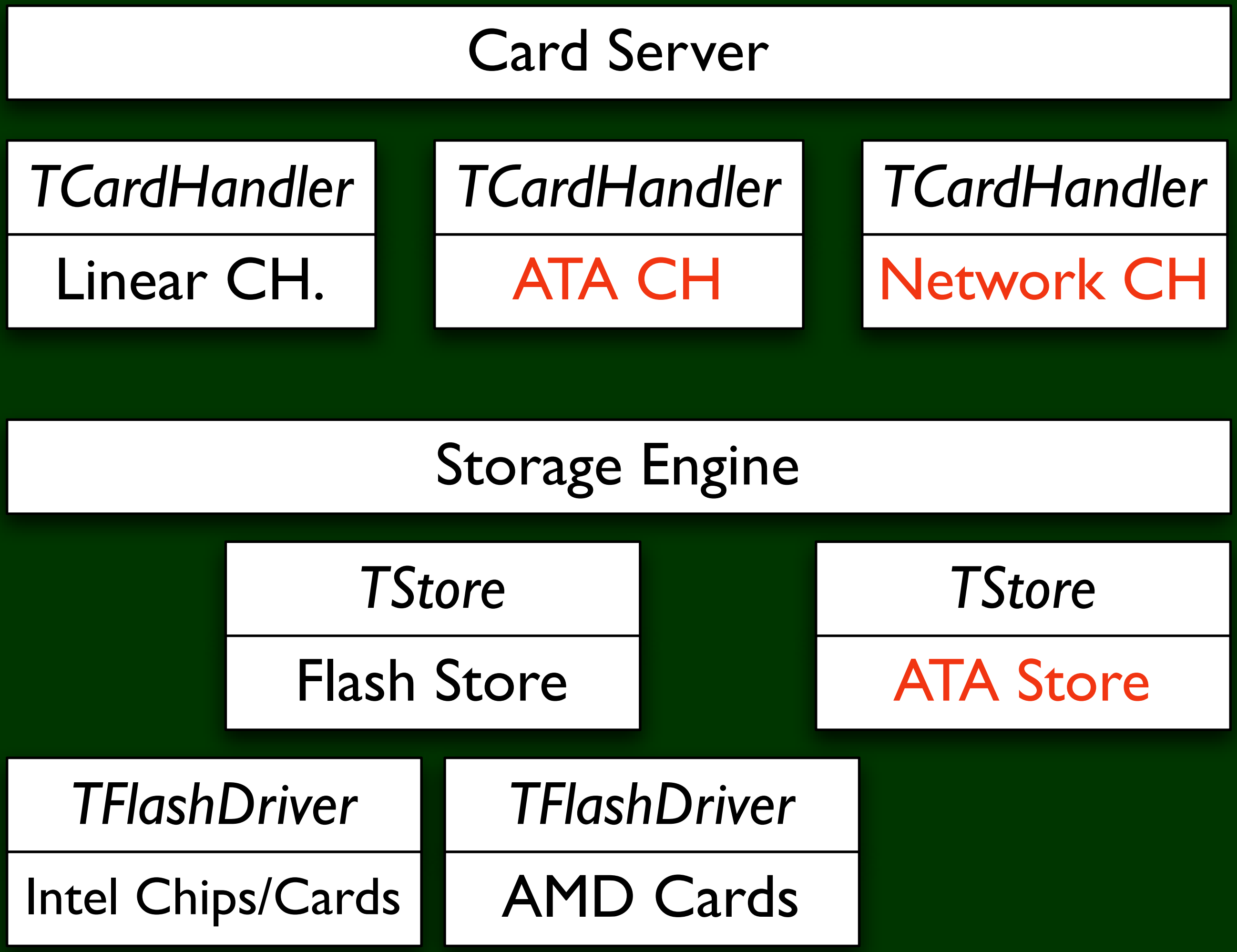
...

}

ATA Support (2000-2003)

- The largest software ever published for NewtonOS.
- What it does:
  - Enables storage on ATA cards “just like” linear cards.
- What I learned:
  - The P-Classes mechanism.

P-Classes



Card Server

TCardHandler

Linear CH.

TCardHandler

ATA CH

TCardHandler

Network CH

Storage Engine

TStore

Flash Store

TStore

ATA Store

TFlashDriver

Intel Chips/Cards

TFlashDriver

AMD Cards

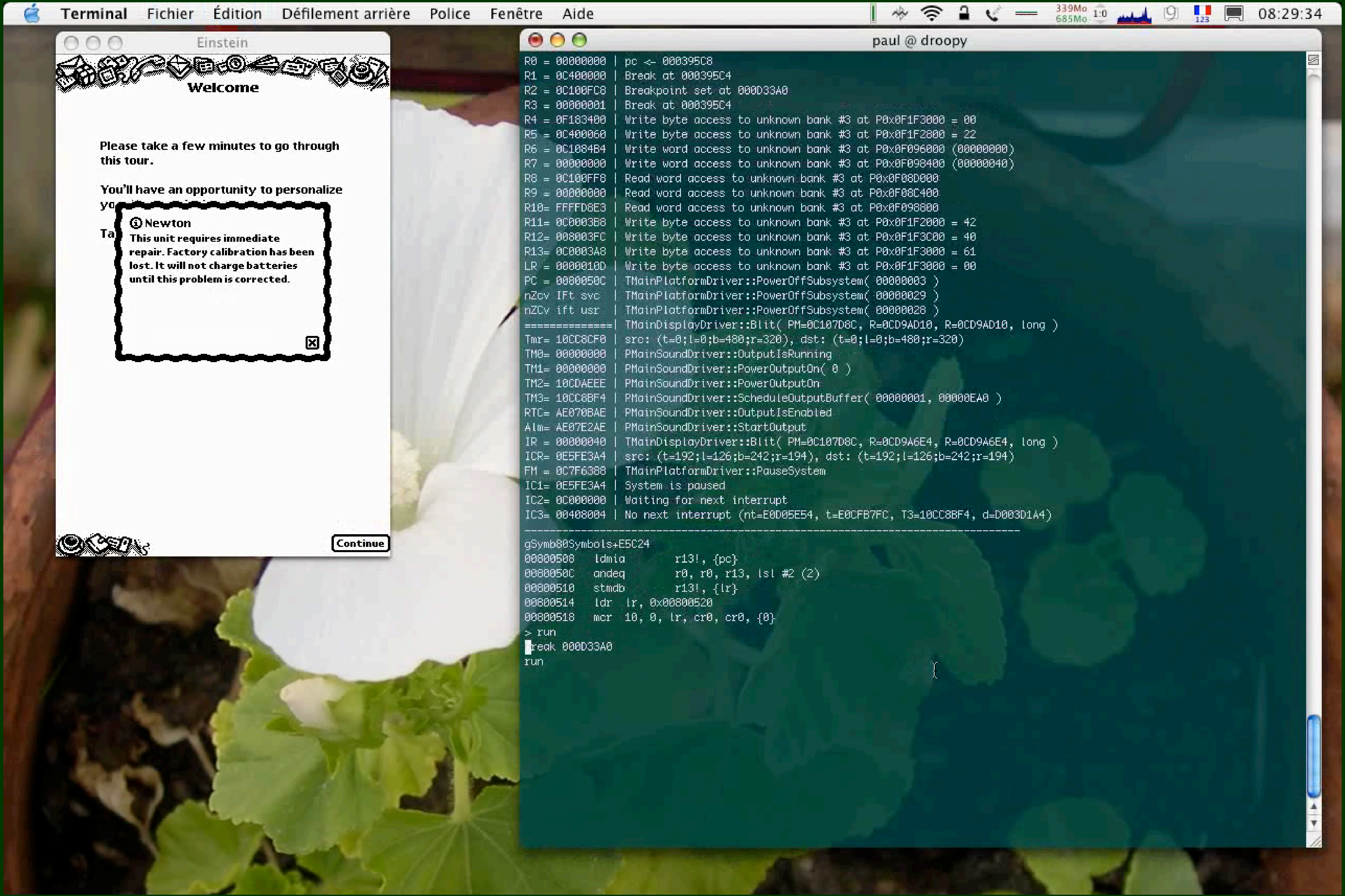
2004-2005
Einstein Emulator

August 2004

- The hardware will eventually die.
- We cannot write an emulator because we don’t know how the Voyager, an essential chipset of the Newton, works.
- The OS (most of the time) accesses the Voyager via a P-Class. How different from the case of ATA Support?
- The Worldwide Newton Conference is near.

September 2004

- Einstein Emulator is unveiled at the first Worldwide Newton Conference in Paris.
- It is awfully slow.
- There are some bugs left.
- But overall it works.



Please take a few minutes to go through this tour.

You'll have an opportunity to personalize yo

Ta

ⓘ Newton

This unit requires immediate repair. Factory calibration has been lost. It will not charge batteries until this problem is corrected.

☒

```
R0 = 00000000 | pc <- 000395C8
R1 = 0C400000 | Break at 000395C4
R2 = 0C100FC8 | Breakpoint set at 000D33A0
R3 = 00000001 | Break at 000395C4
R4 = 0F183400 | Write byte access to unknown bank #3 at P0x0F1F3000 = 00
R5 = 0C400060 | Write byte access to unknown bank #3 at P0x0F1F2800 = 22
R6 = 0C1084B4 | Write word access to unknown bank #3 at P0x0F096000 (00000000)
R7 = 00000000 | Write word access to unknown bank #3 at P0x0F098400 (00000040)
R8 = 0C100FF8 | Read word access to unknown bank #3 at P0x0F08D000
R9 = 00000000 | Read word access to unknown bank #3 at P0x0F08C400
R10= FFFFD8E3 | Read word access to unknown bank #3 at P0x0F098800
R11= 0C0003B8 | Write byte access to unknown bank #3 at P0x0F1F2000 = 42
R12= 008003FC | Write byte access to unknown bank #3 at P0x0F1F3C00 = 40
R13= 0C0003A8 | Write byte access to unknown bank #3 at P0x0F1F3000 = 61
LR = 0000010D | Write byte access to unknown bank #3 at P0x0F1F3000 = 00
PC = 0080050C | TMainPlatformDriver::PowerOffSubsystem( 00000003 )
nZcv IFt svc  | TMainPlatformDriver::PowerOffSubsystem( 00000029 )
nZCv ift usr  | TMainPlatformDriver::PowerOffSubsystem( 00000028 )
==============| TMainDisplayDriver::Blit( PM=0C107D8C, R=0CD9AD10, R=0CD9AD10, long )
Tmr= 10CC8CF0 | src: (t=0;l=0;b=480;r=320), dst: (t=0;l=0;b=480;r=320)
TM0= 00000000 | PMainSoundDriver::OutputIsRunning
TM1= 00000000 | PMainSoundDriver::PowerOutputOn( 0 )
TM2= 10CDAEEE | PMainSoundDriver::PowerOutputOn
TM3= 10CC8BF4 | PMainSoundDriver::ScheduleOutputBuffer( 00000001, 00000EA0 )
RTC= AE070BAE | PMainSoundDriver::OutputIsEnabled
Alm= AE07E2AE | PMainSoundDriver::StartOutput
IR = 00000040 | TMainDisplayDriver::Blit( PM=0C107D8C, R=0CD9A6E4, R=0CD9A6E4, long )
ICR= 0E5FE3A4 | src: (t=192;l=126;b=242;r=194), dst: (t=192;l=126;b=242;r=194)
FM = 0C7F6388 | TMainPlatformDriver::PauseSystem
IC1= 0E5FE3A4 | System is paused
IC2= 0C000000 | Waiting for next interrupt
IC3= 00408004 | No next interrupt (nt=E0D05E54, t=E0CFB7FC, T3=10CC8BF4, d=D003D1A4)
--------------------------------------------------------------------------------
gSymb80Symbols+E5C24
00800508   ldmia         r13!, {pc}
0080050C   andeq         r0, r0, r13, lsl #2 (2)
00800510   stmdb         r13!, {lr}
00800514   ldr  lr, 0x00800520
00800518   mcr  10, 0, lr, cr0, cr0, {0}
> run
break 000D33A0
run
```

Optimizing Einstein (2004-2005)

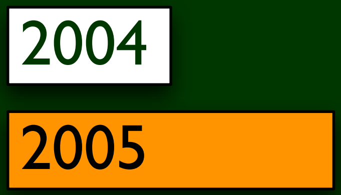
- Einstein 2004 was a simulator: it translated all instructions one by one. It also emulated some hardware with high (useless) precision.
- Einstein 2005 was entirely rewritten. It uses dynamic translation, cache everywhere and highly optimized MMU calls.
- I lost a lot of hair.

1x Faster

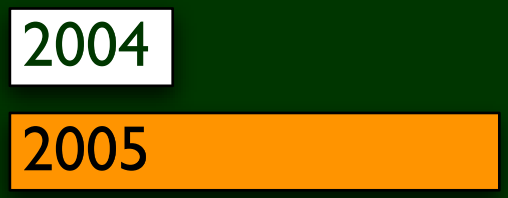
2004

2005

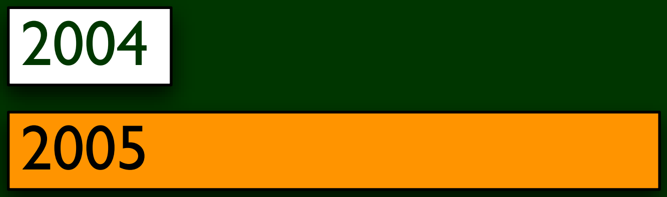
2x Faster



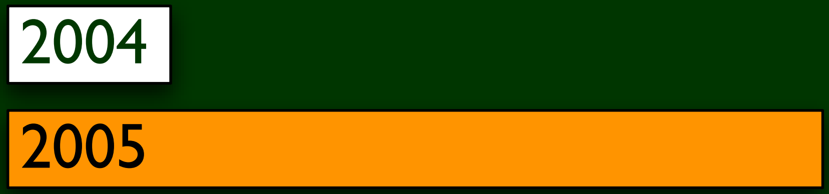
3x Faster



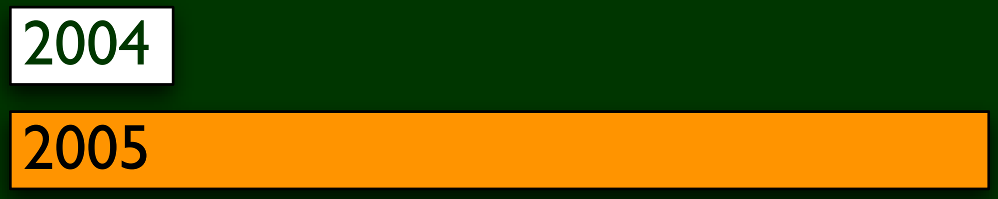
4x Faster



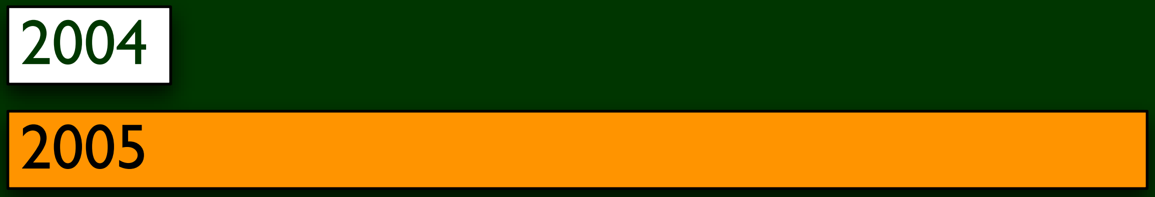
5x Faster



6x Faster

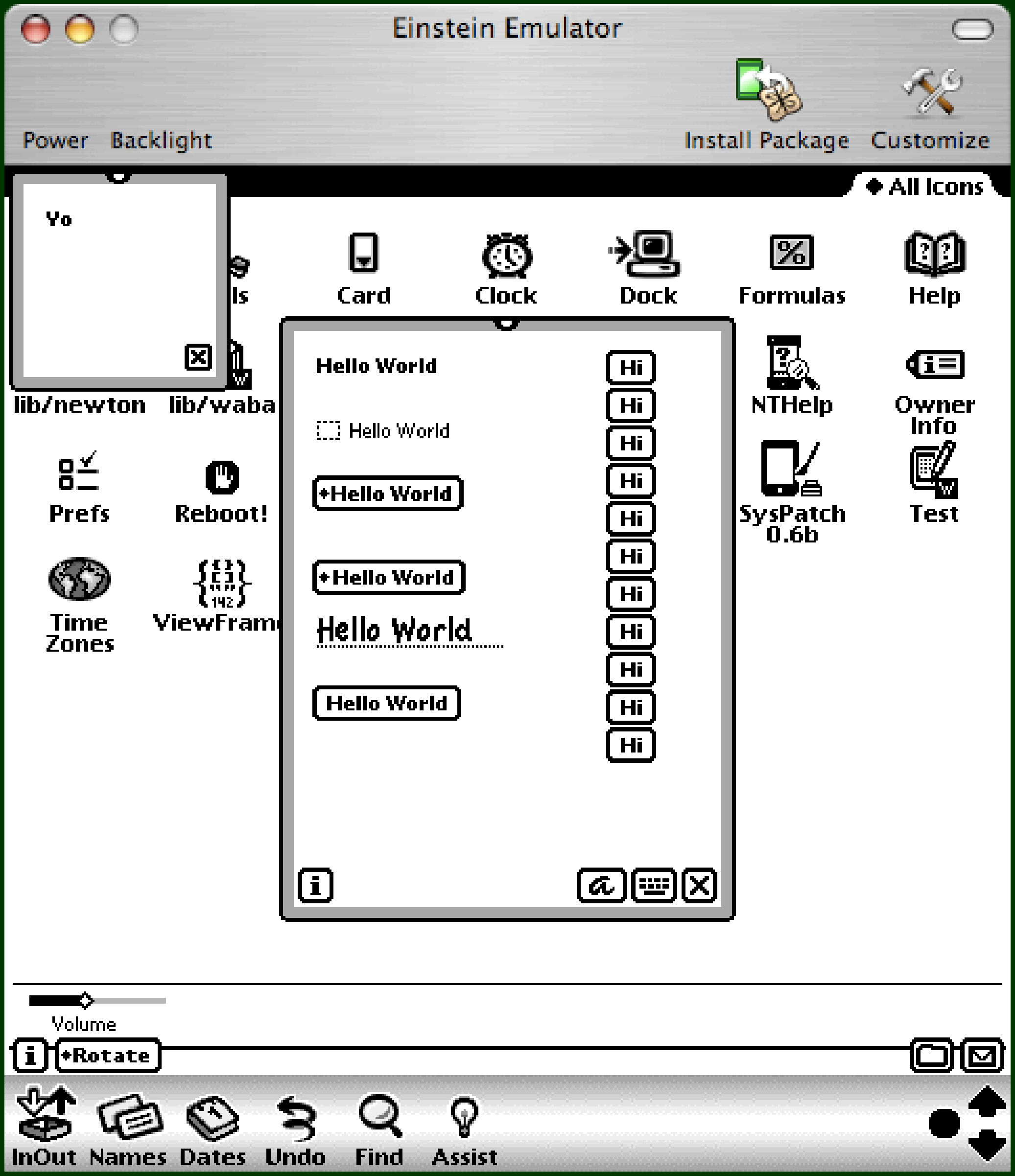


7x Faster



Einstein 2005

- Is a great development tool for MacOS X.
- Sean Luke uses it for new Waba developments.



In Summary

Einstein 2004:

- was awfully slow
- had bugs left
- but worked

Einstein 2005:

- is useful to developers
- is 7x times faster

Einstein 2006…

…is not an emulator…

...it is...

01/12/2006
Zaurus
SHARP
OK
Cancel
On/Off

Einstein 2006

- It runs on the Zaurus SL5500 with a bootstrap OpenZaurus ROM.
- It is 40 MB on a CompactFlash card.
- It should run on any embedded system with X11 and enough memory. You name it.

Einstein 2006

- Einstein Platform is unveiled today, at the WWNC 2006. You saw it first.
- It is available for download today.
- It is awfully slow.
- There are some bugs left.
- But overall it works.

The Future

1. Speed

- Einstein has been optimized to be 7 times faster.
- It can be done again.
- It can. I have some hair left.

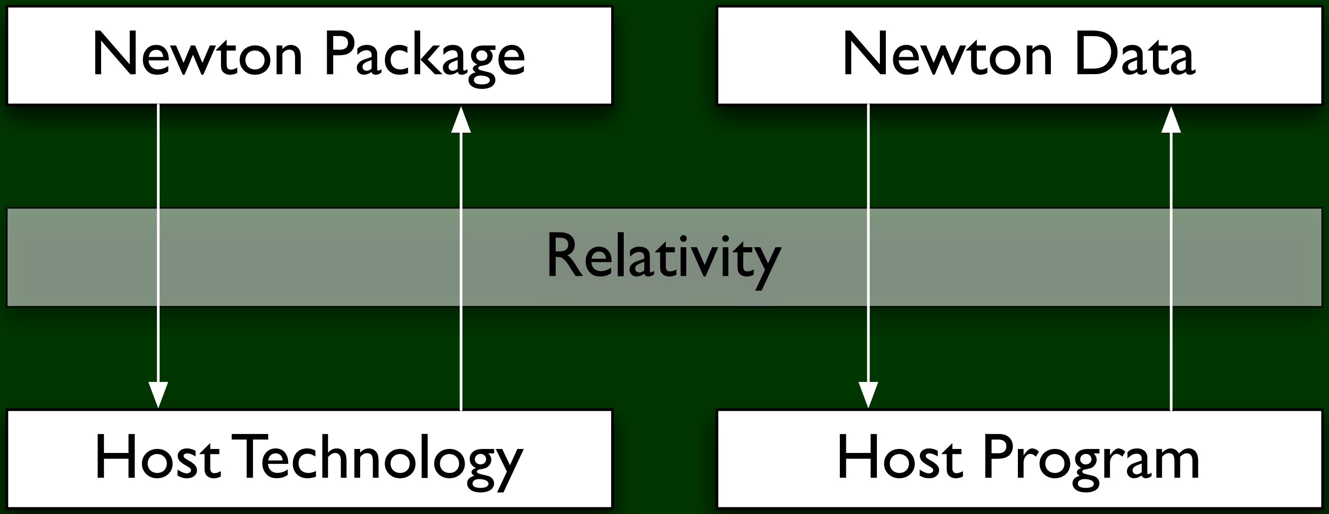
1. Speed

- Einstein has been optimized to be 7 times faster.
- It can be done again.
- It can. I have some ideas.

2. Relativity

- Relativity is Einstein Program’s next technology.
- It will allow the merge of host and newton applications, technologies and data.
- As an example, Einstein units will better synchronize than good ol’ Newtons.

2. Relativity



3. Color

3. Colors

- Millions. Millions of colors.
- Are made possible by a technology used in Einstein 2006.

The Future

- Speed: can be done again.
- Relativity: synchronization, PDF viewing, whatever modern PDAs do plus what you love in your Newton. And more.
- Color: get prepared to update your good old Newton programs.

In Summary

Einstein 2006:

- is awfully slow
- has bugs left
- but works

Einstein 2???:

In Summary

Einstein 2006:

- is awfully slow
- has bugs left
- but works

Einstein 20??:

In Summary

Einstein 2006:	Einstein 200?:
- is awfully slow	- will be much faster
- has bugs left	- will feature Relativity
- but works	- will feature Colors

When?

- Next conference, maybe?
- In 2006, I will also:
  - Get my Ph.D.
  - Get a new job.

Questions?

Paul Guyot
pguyot@kallisys.net
http://www.kallisys.com/newton/einstein/