Compilers

Chapter 1: Bonus material
Bootstrapping compilers:
"T-diagrams"

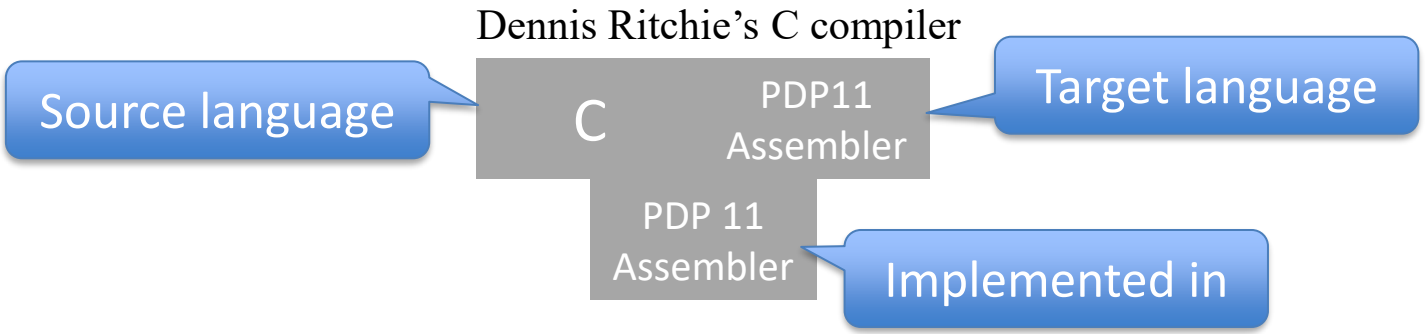
- Lecturer:
  - Paul Kelly (p.kelly@imperial.ac.uk)

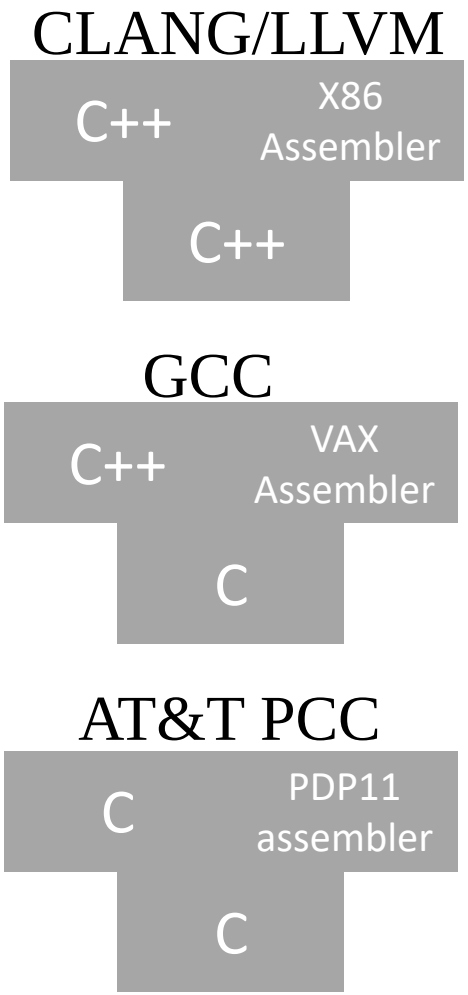
Non-examinable material for fun/interest

- **How was the first ever C compiler written?**

- **In C?**

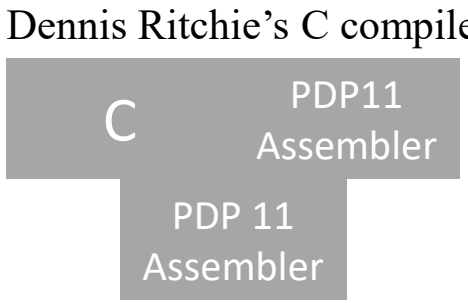
- **How was the first ever C compiler written?**

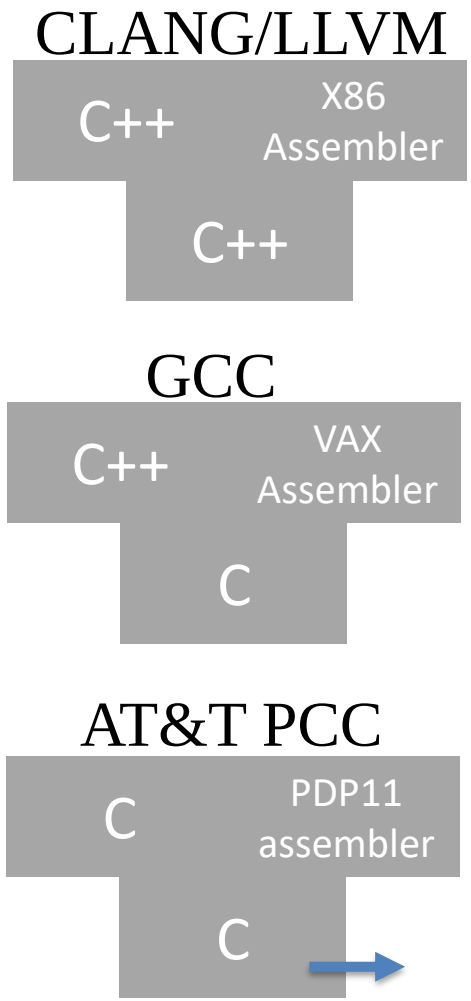




- **In assembly code, by hand!**
- **A "T-Diagram" shows the source language, the target language, and the implementation language**
  - **(GCC and CLANG/LLVM are compiler *frameworks* that support multiple source languages and target architectures)**

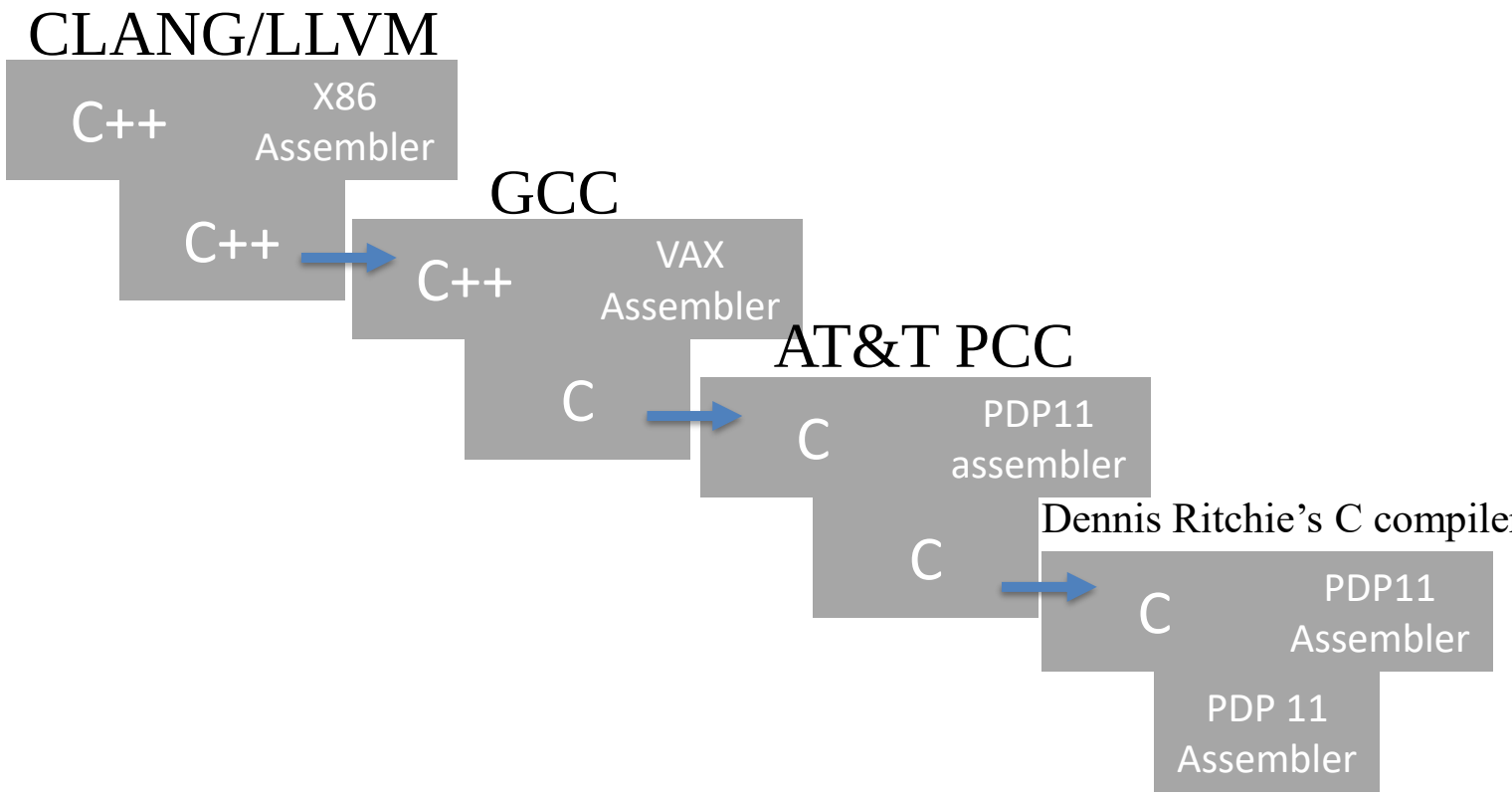
- **How was the *second* C compiler written?**





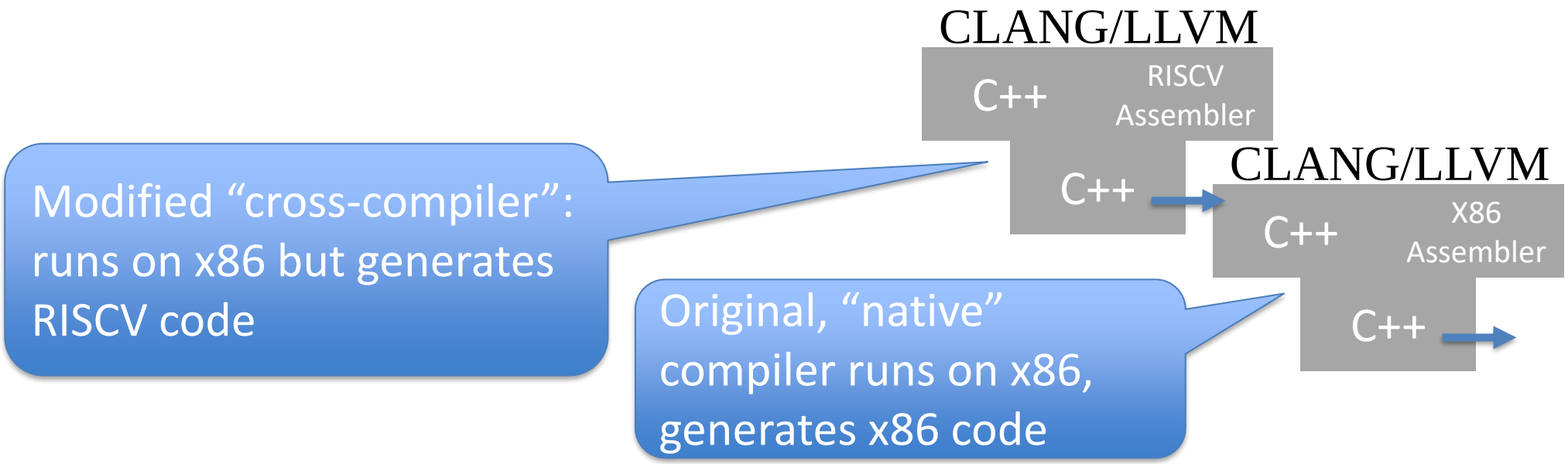
- **In C, compiled using the first C compiler**

- **How were later compilers written?**

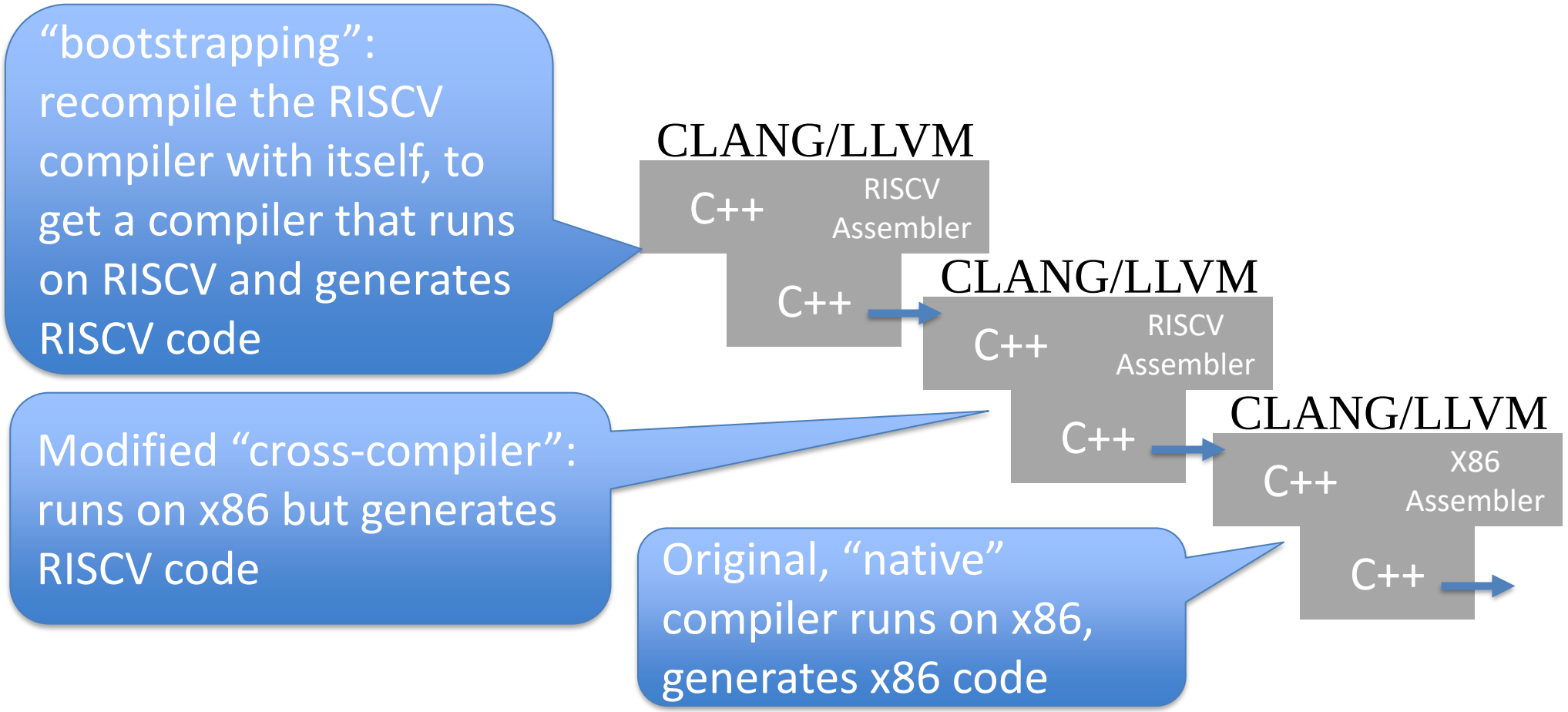


- **In languages compiled by earlier compilers**
- **And ported to new instruction sets**

- **How were compilers ported to new hardware?**

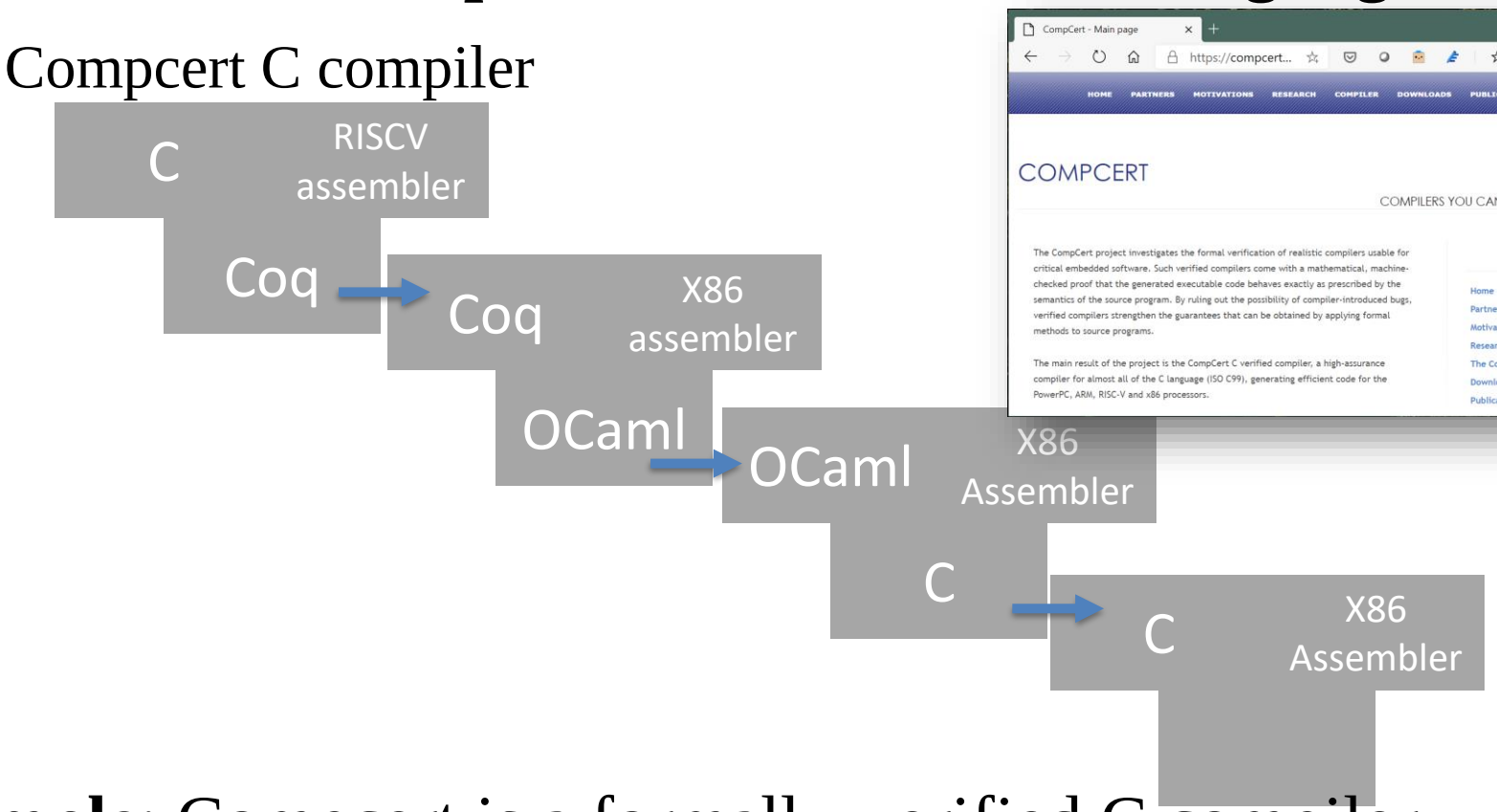


- **How were later compilers ported to new hardware?**



- **By "bootstrapping":**
  - **First, develop a new back-end that generates code for the new target**
  - **Then use it to recompile the compiler itself**

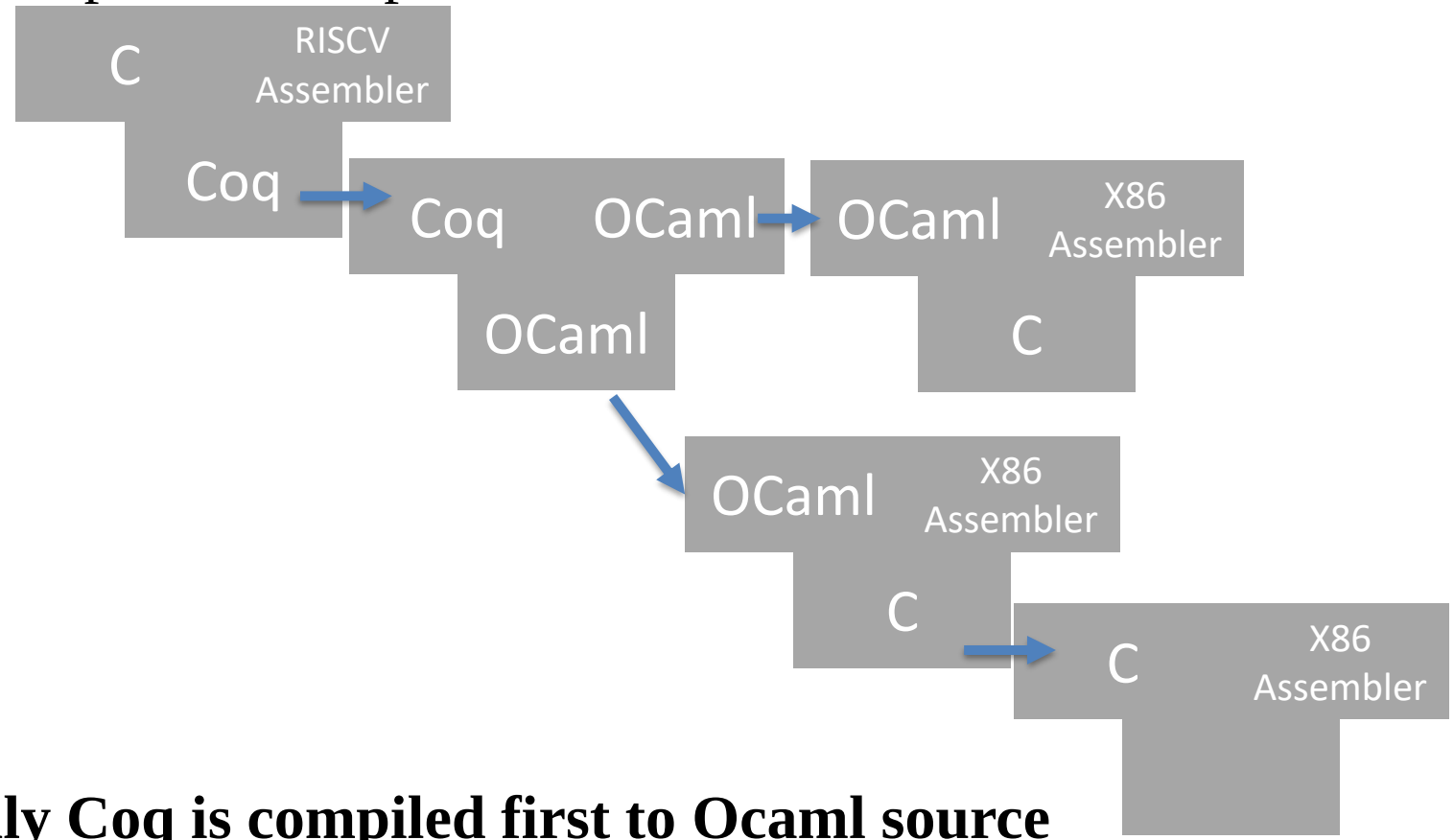
- **How were later compilers written in other languages?**



- **Example**: Compcert is a formally-verified C compiler, generating code for x86, ARM, PowerPC and RISCV
- The compiler is implemented in the theorem prover **Coq**
- Which is implemented in the functional language **OCaml**

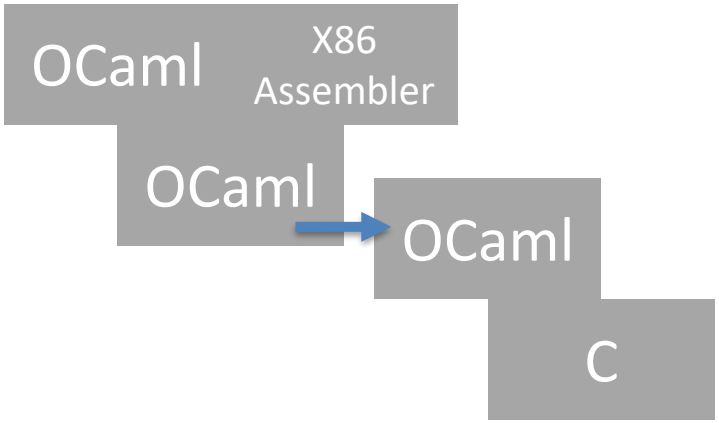
- **How about compilers that generate high-level code?**

Compcert C compiler



- **Actually Coq is compiled first to Ocaml source code *then* to machine code**

- **How about interpreters?**



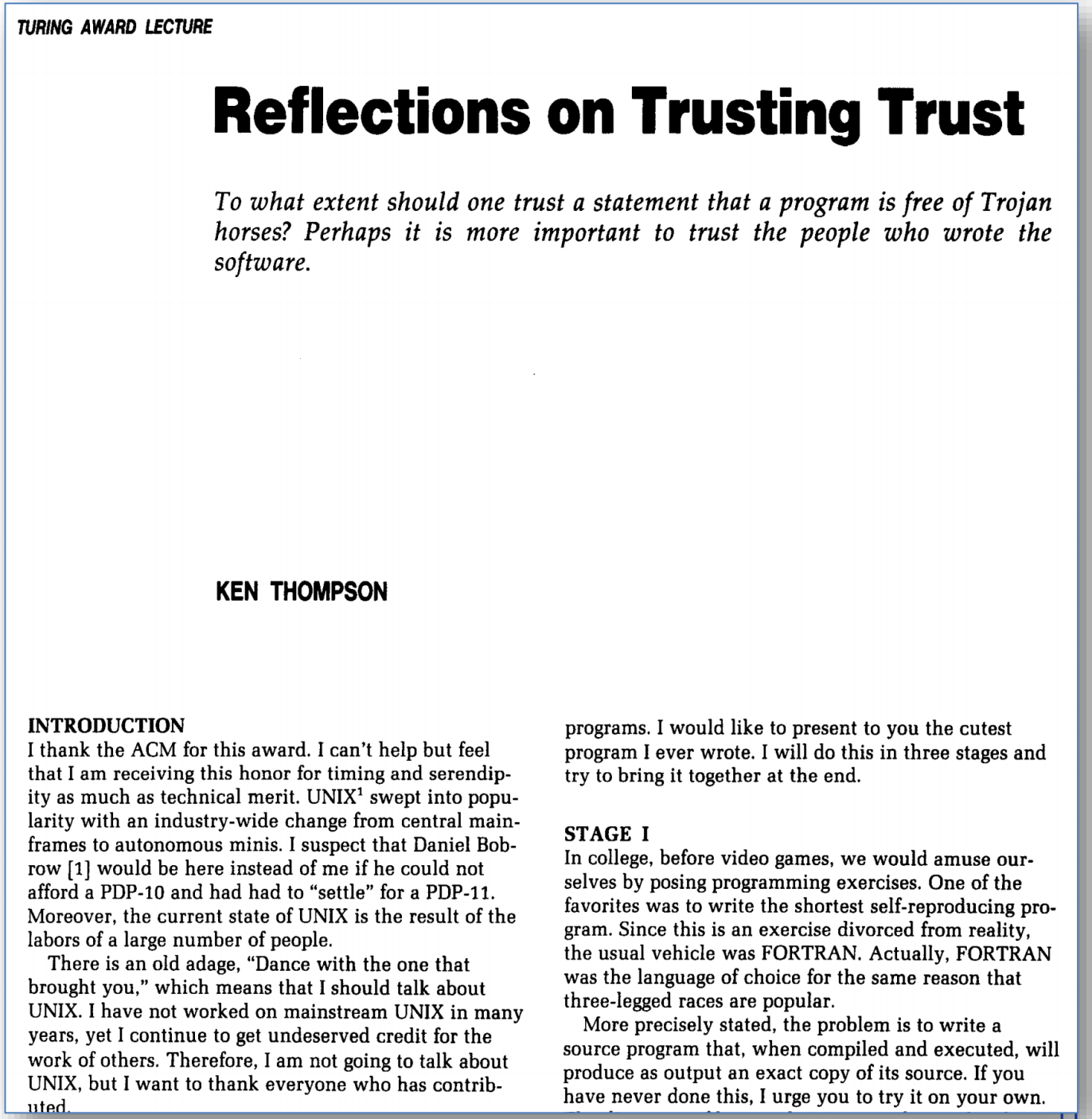
- **Actually the OCaml compiler is written in OCaml**
- **An Ocaml *interpreter* is used to compile the OCaml compiler**

- **See also "J-diagrams" (https://johnwickerson.wordpress.com/2020/05/21/diagrams-for-composing-compilers/)**

If you like this kind of thing, check out this:

Ken Thompson. 1984. Reflections on trusting trust. Commun. ACM 27, 8 (Aug 1984), 761–763. DOI: https://doi.org/10.1145/358198.358210

Then figure out if you can find a way to ever trust a compiler again…



TURING AWARD LECTURE

Reflections on Trusting Trust

To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.

KEN THOMPSON

INTRODUCTION

I thank the ACM for this award. I can't help but feel that I am receiving this honor for timing and serendipity as much as technical merit. UNIX[1] swept into popularity with an industry-wide change from central mainframes to autonomous minis. I suspect that Daniel Bobrow [1] would be here instead of me if he could not afford a PDP-10 and had had to "settle" for a PDP-11. Moreover, the current state of UNIX is the result of the labors of a large number of people.

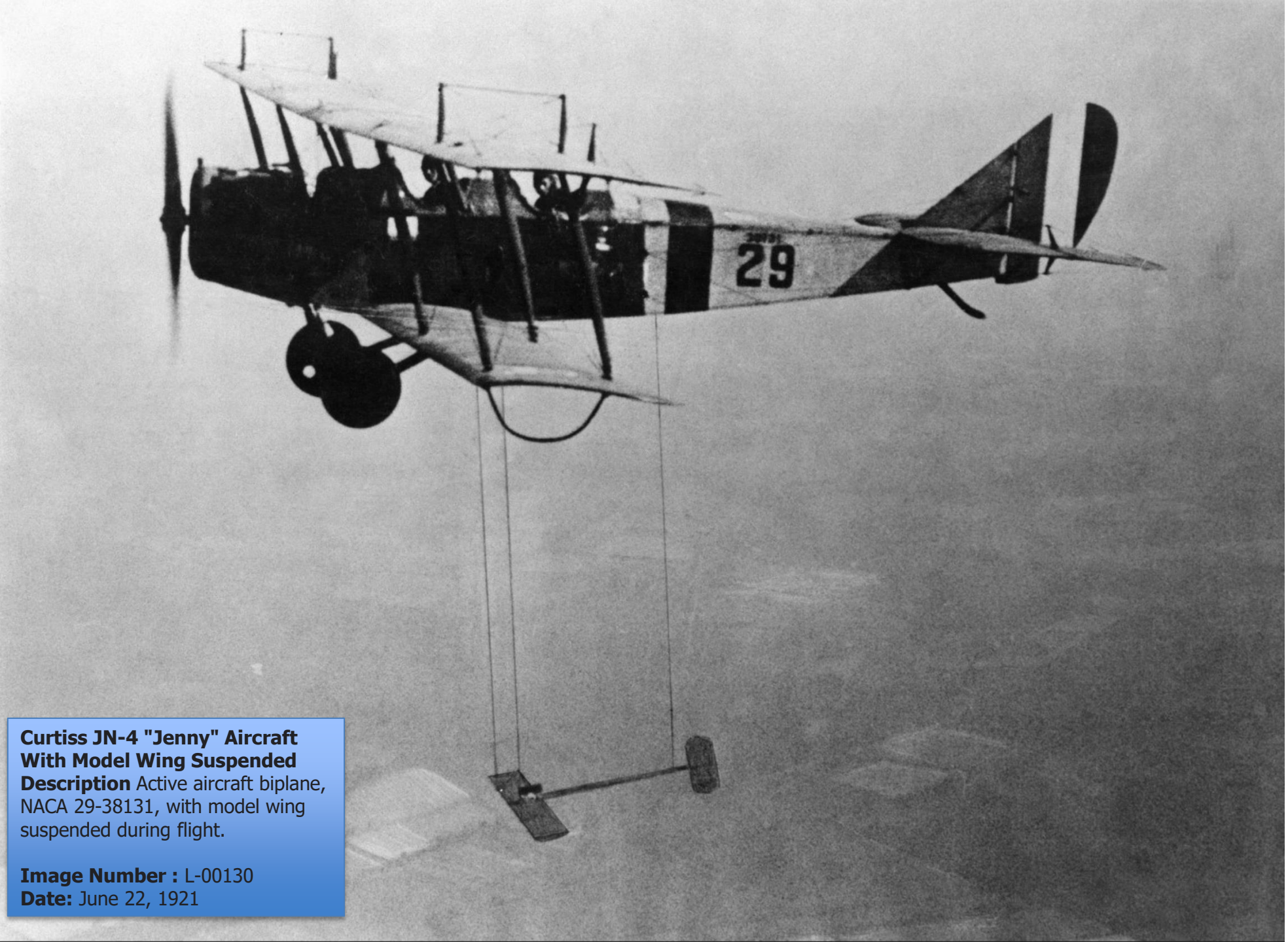
There is an old adage, "Dance with the one that brought you," which means that I should talk about UNIX. I have not worked on mainstream UNIX in many years, yet I continue to get undeserved credit for the work of others. Therefore, I am not going to talk about UNIX, but I want to thank everyone who has contributed.

programs. I would like to present to you the cutest program I ever wrote. I will do this in three stages and try to bring it together at the end.

STAGE I

In college, before video games, we would amuse ourselves by posing programming exercises. One of the favorites was to write the shortest self-reproducing program. Since this is an exercise divorced from reality, the usual vehicle was FORTRAN. Actually, FORTRAN was the language of choice for the same reason that three-legged races are popular.

More precisely stated, the problem is to write a source program that, when compiled and executed, will produce as output an exact copy of its source. If you have never done this, I urge you to try it on your own.



Curtiss JN-4 "Jenny" Aircraft With Model Wing Suspended
Description Active aircraft biplane, NACA 29-38131, with model wing suspended during flight.

Image Number : L-00130
Date: June 22, 1921