



Compiladores e intérpretes **Procesadores de lenguajes**

Primer semestre 2008



Procesadores de lenguajes

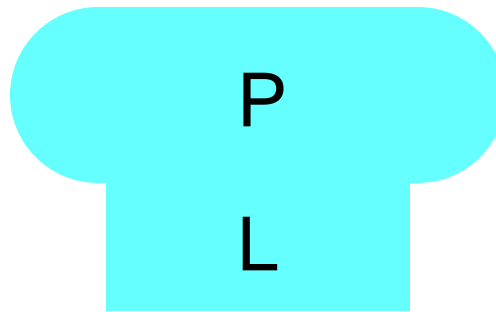
- Son sistemas (hardware o software) que manipulan programas
- Por ejemplo:
 - Editores
 - Traductores / compiladores
 - Intérpretes
 - Computadores

Combinación lenguaje – máquina

- Se representa usando los diagramas de lápida (*tombstone*)

Diagramas de programas

- Programa P expresado en lenguaje L

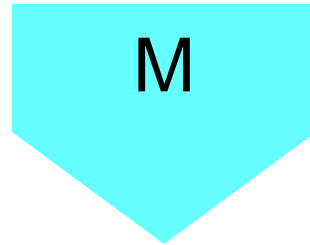


- E.g. programa HelloWorld para procesadores x86

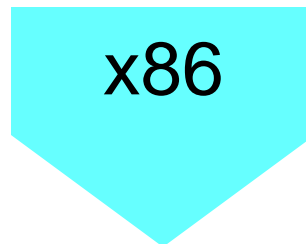


Diagramas de lápida

- Máquina M (i.e. una máquina capaz de correr programas en el lenguaje M)

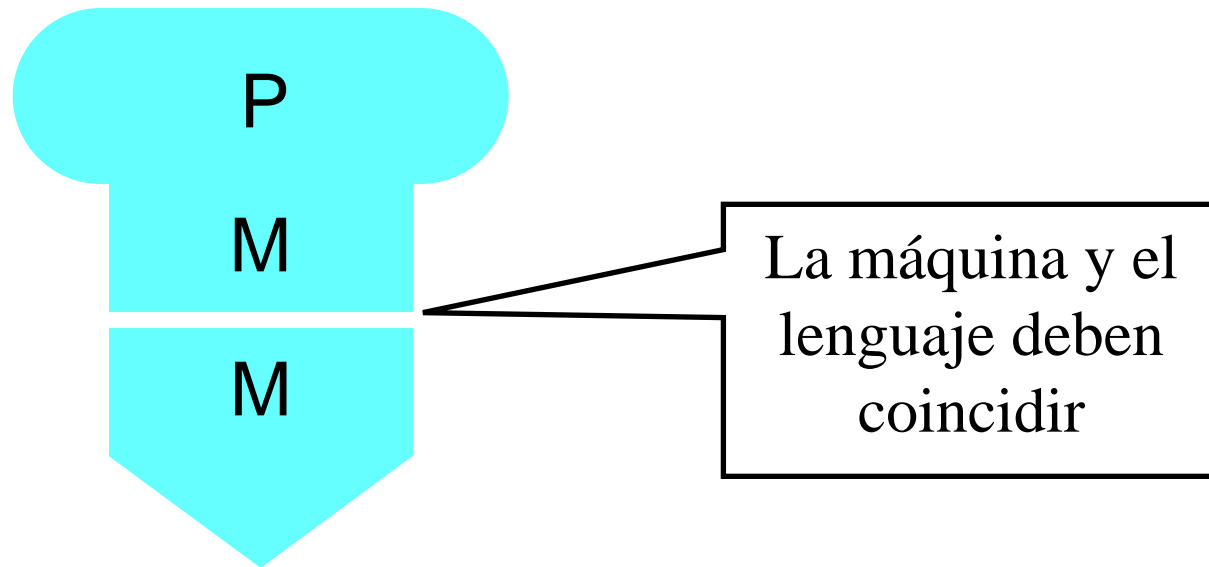


- E.g. Un procesador X86



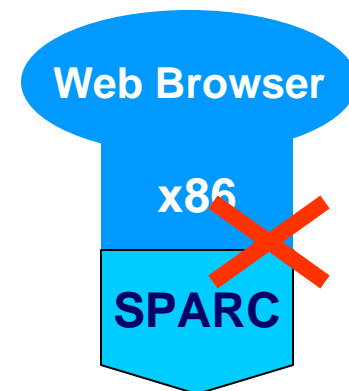
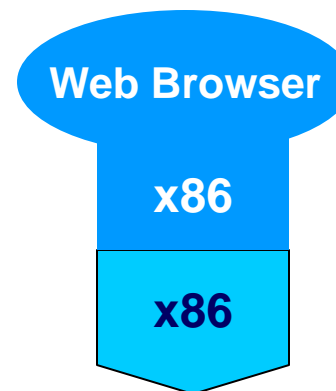
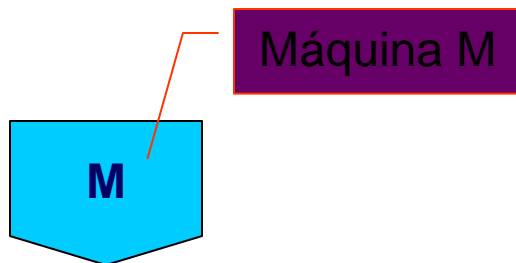
Ejecutar P en una máquina

- Ejecutar un programa escrito en lenguaje M, en la máquina M



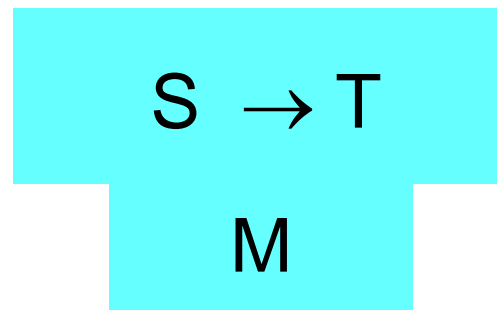
Diagramas de lápida

- Máquina

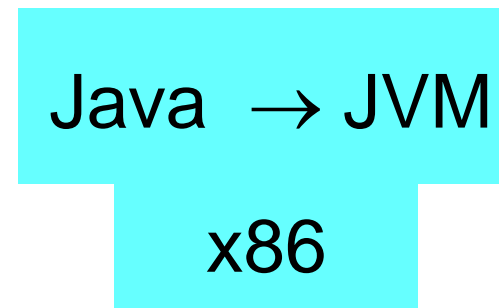


Traductores

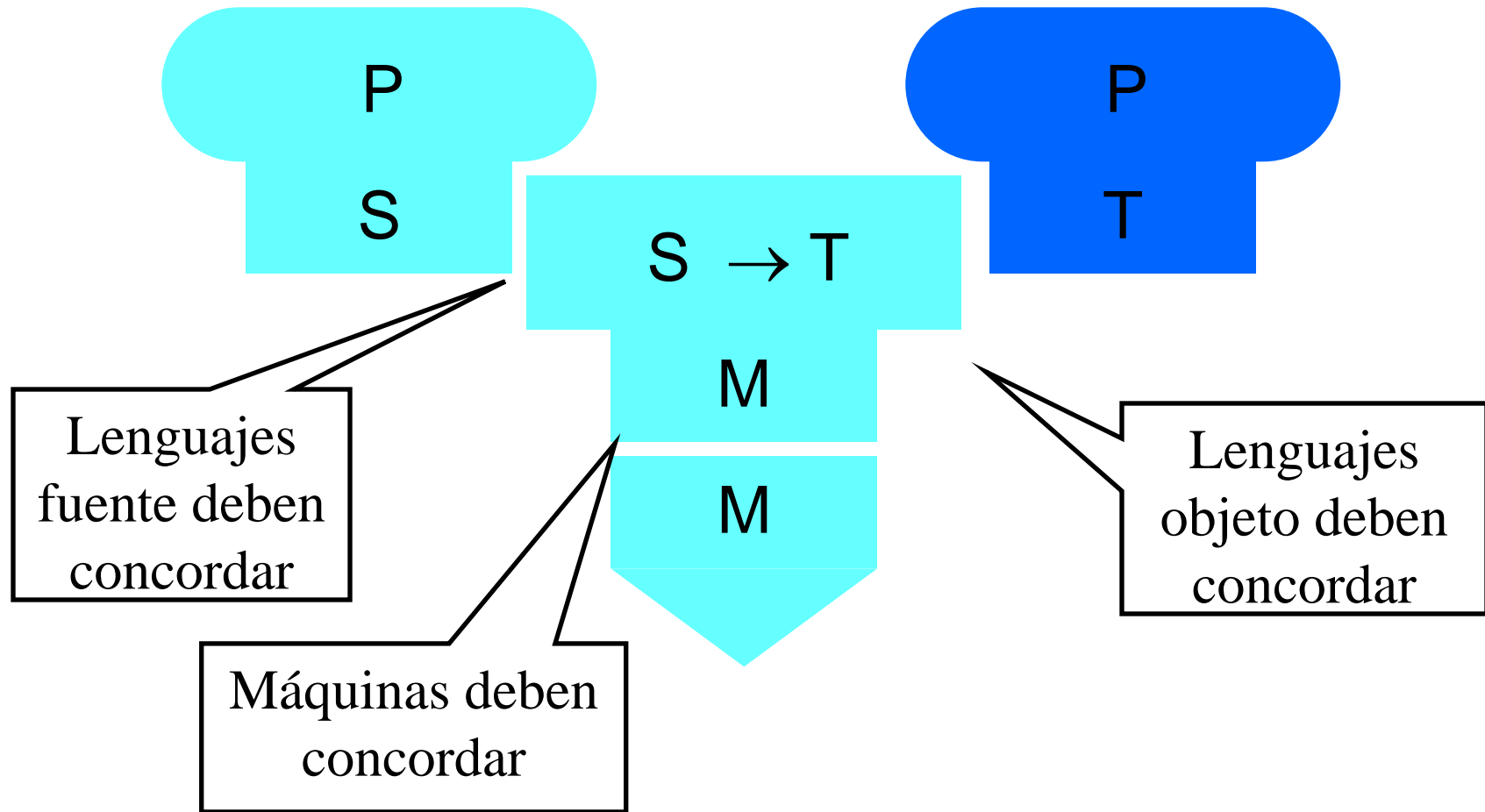
- Un traductor que se ejecuta en la máquina M y convierte el lenguaje S en lenguaje T



- Ej.: un compilador de Java a JVM para un microprocesador X86



Uso de traductores



El resultado de la traducción está en tono más oscuro

Compilación

- Compilación, y ejecución del resultado
 - Comprende 2 pasos

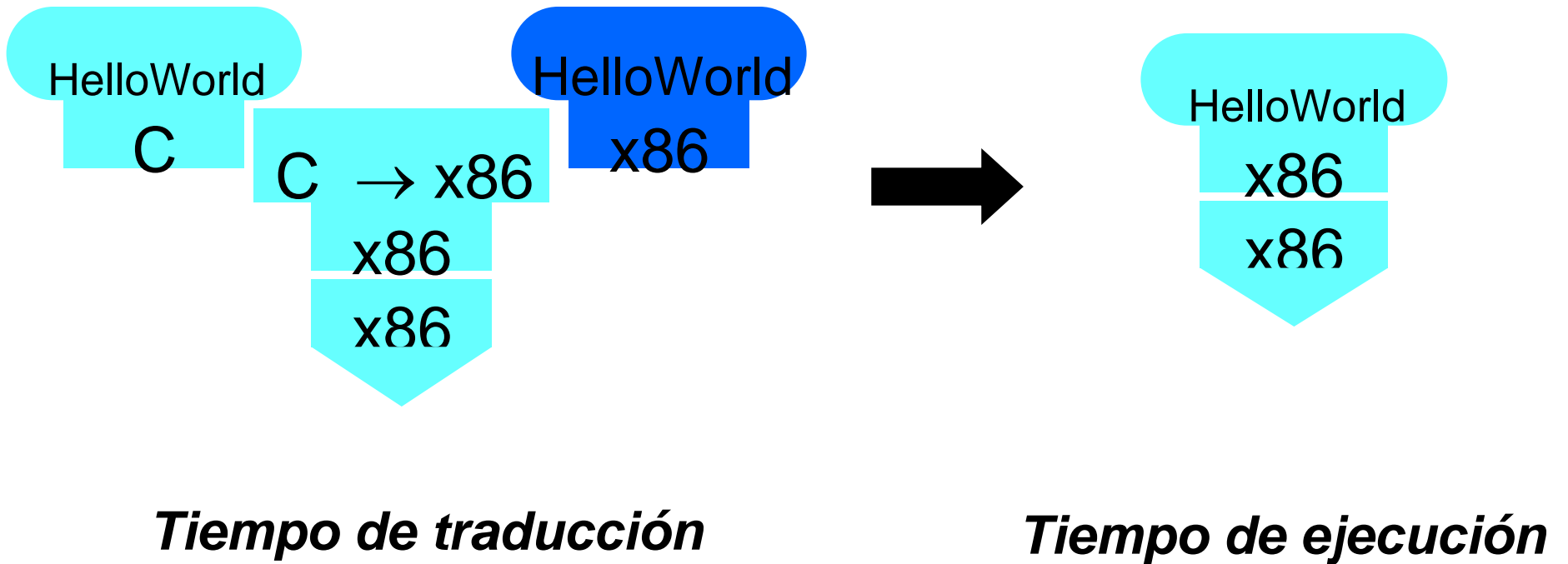


Diagrama de lápida

- Compilación

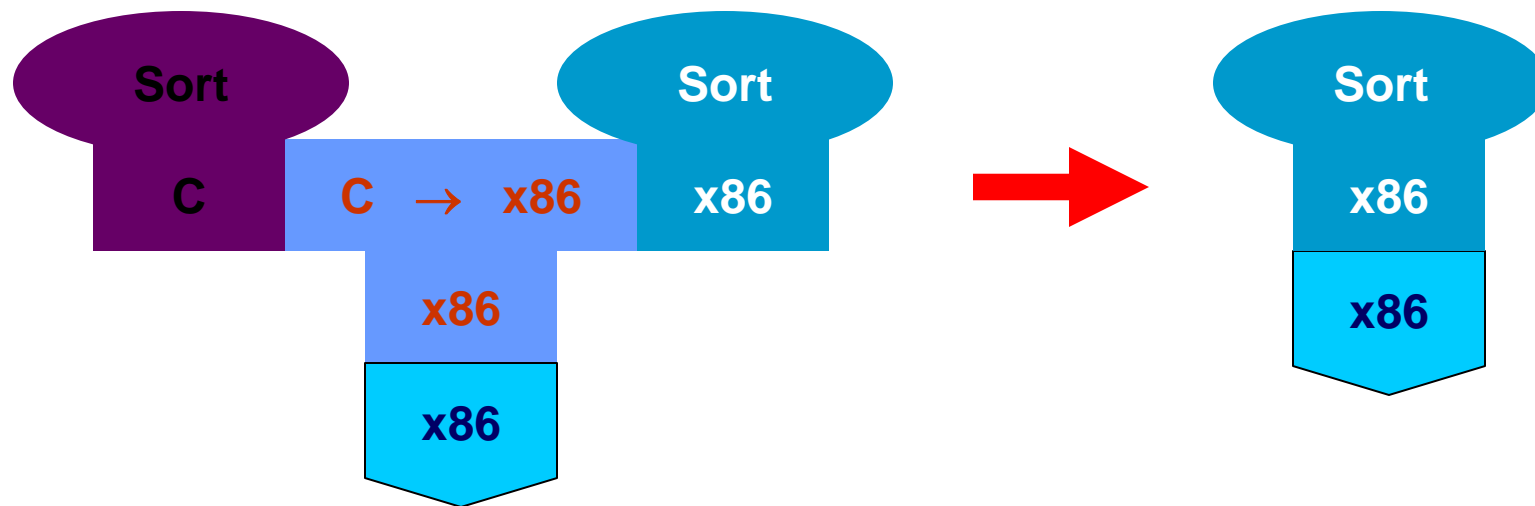
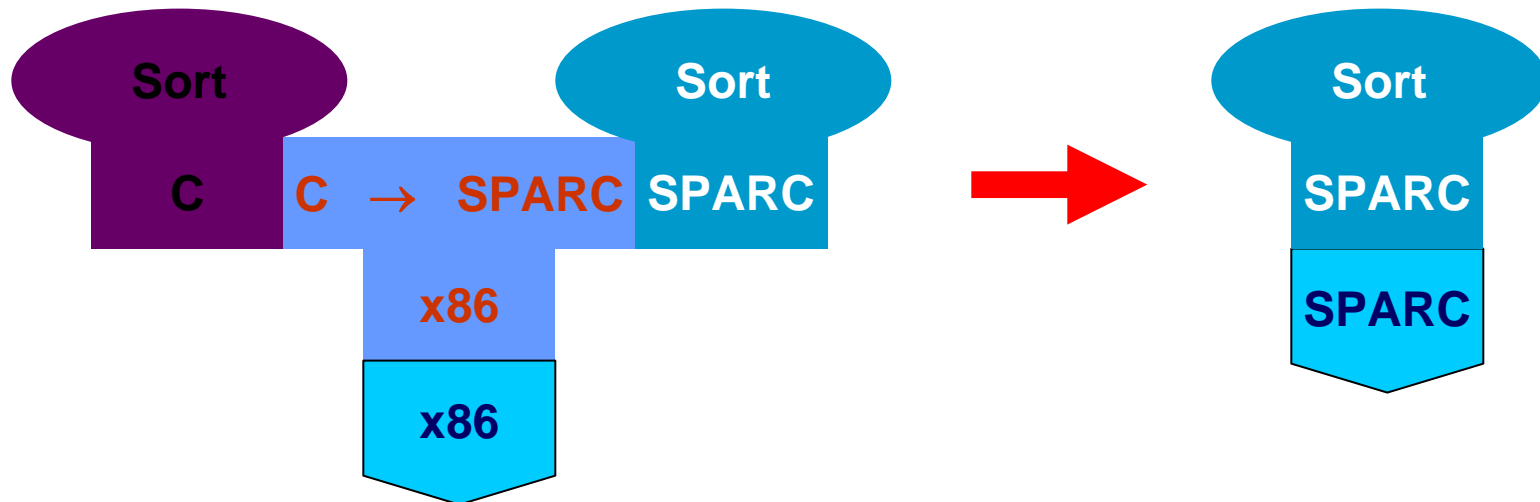


Diagrama de lápida

- “Cross Compilation”

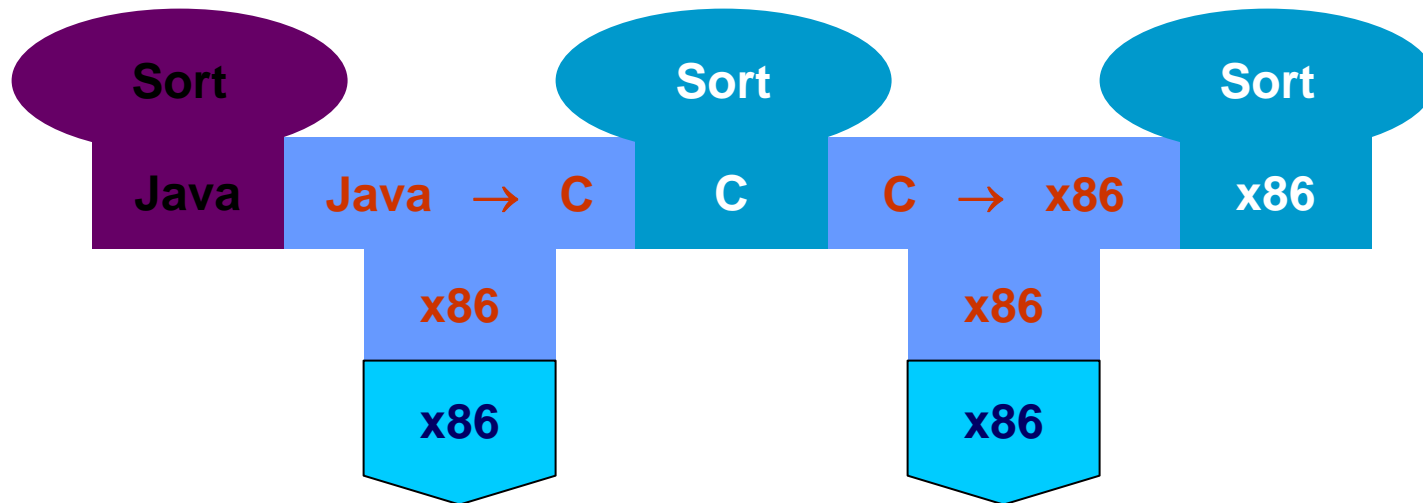


Compilación cruzada

Compilar código en una máquina para ser ejecutado en otra máquina

Diagrama de lápida

- Compilación de *dos* etapas



Compile a un lenguaje, luego a otro.

E.g. Compile Java a C y luego C a x86

Intérpretes

- La interpretación es ejecutar un programa sin ninguna traducción intermedia (al menos visible)
 - Ej. Ejecutar un programa JVM en un X86 sin traducirlo a x86. Diagrama de un intérprete para el lenguaje S, escrito en el lenguaje L

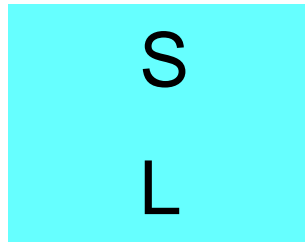
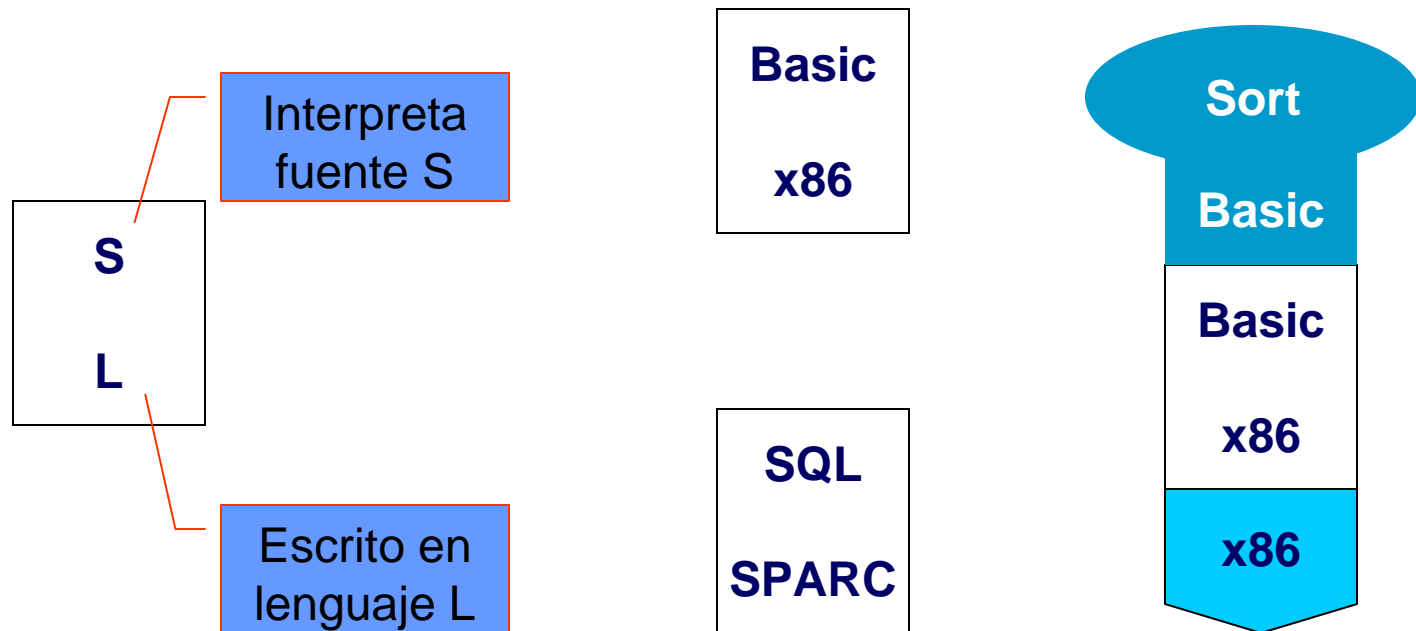


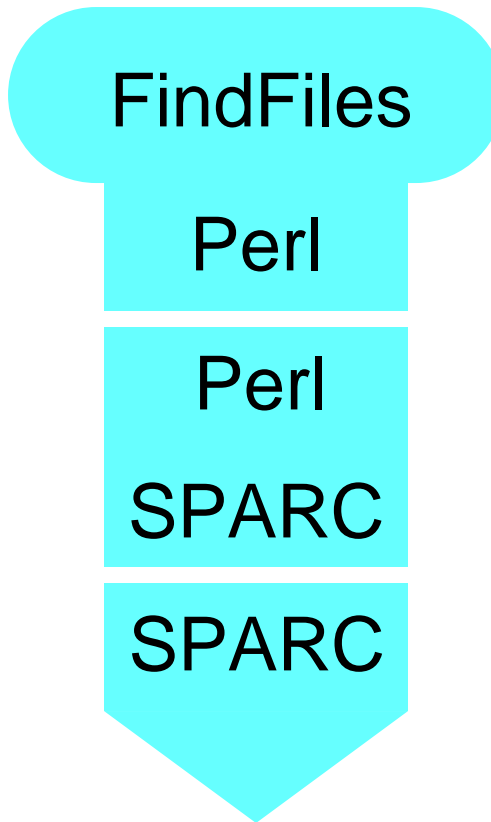
Diagrama de lápida

- Intérprete



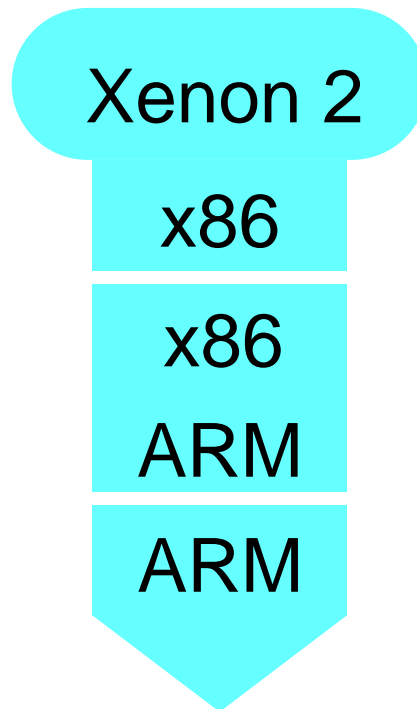
Uso de un intérprete

- Algunas personas usan Perl interpretado en computadores SPARC que corren Unix



Una MV para emular hardware

- Con un intérprete se puede ejecutar código de una máquina real en otra (real o virtual).



Compilar intérpretes

- ¿De dónde viene un emulador de x86?
- Posiblemente fue escrito en C y se compiló en ARM

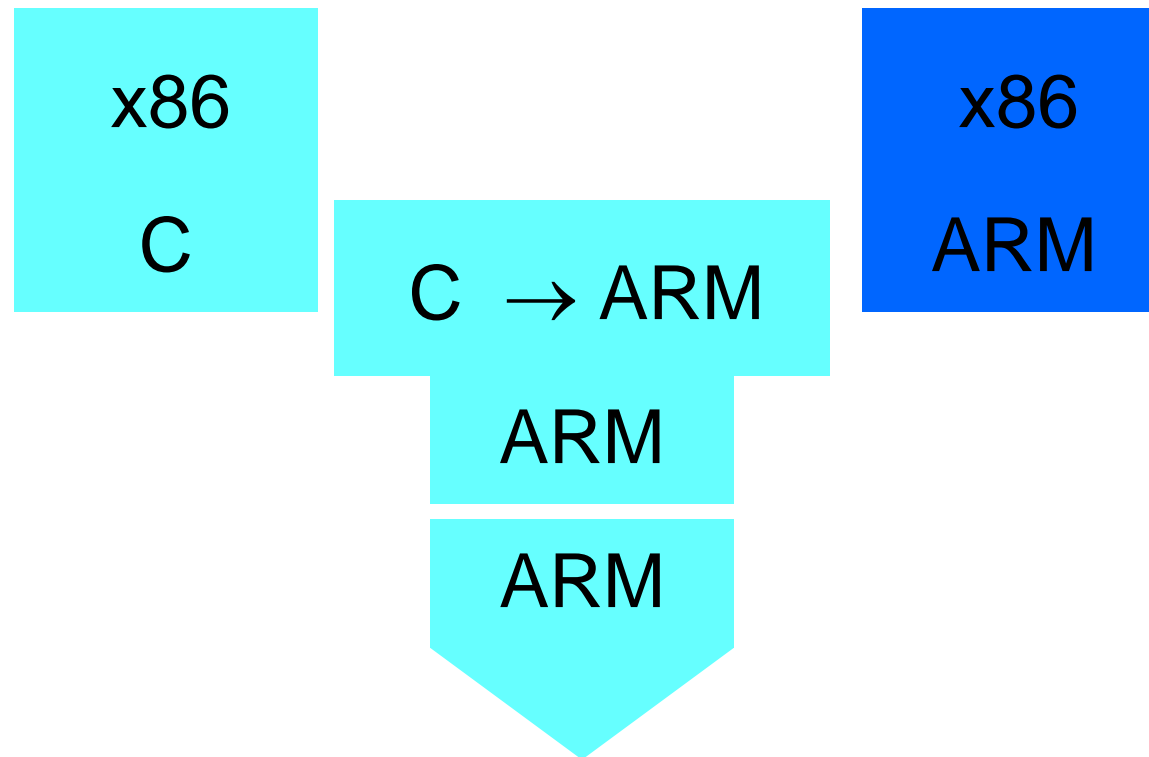


Diagrama de lápida

- Máquina virtual o abstracta = Emulador de hardware
 - Intérprete para lenguaje de bajo nivel.

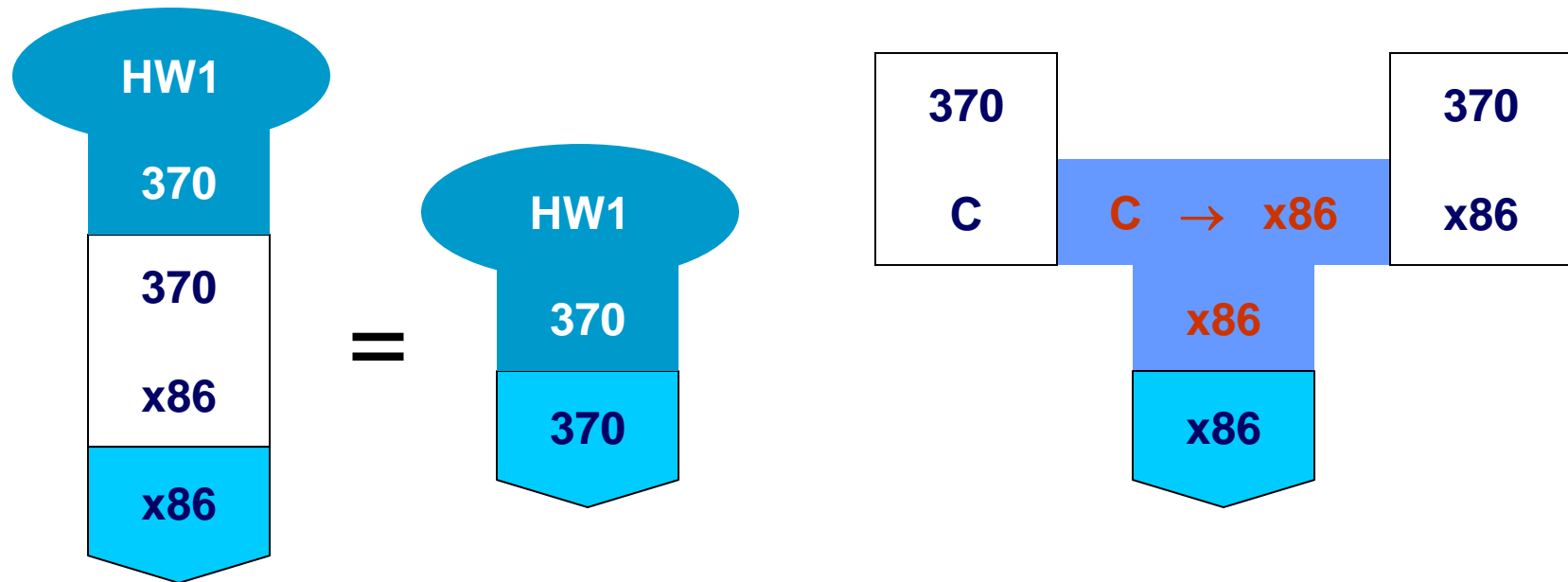


Diagrama de lápida

- Java
 - Ambiente portátil: “write-once-run-anywhere”
 - Compilador interpretativo

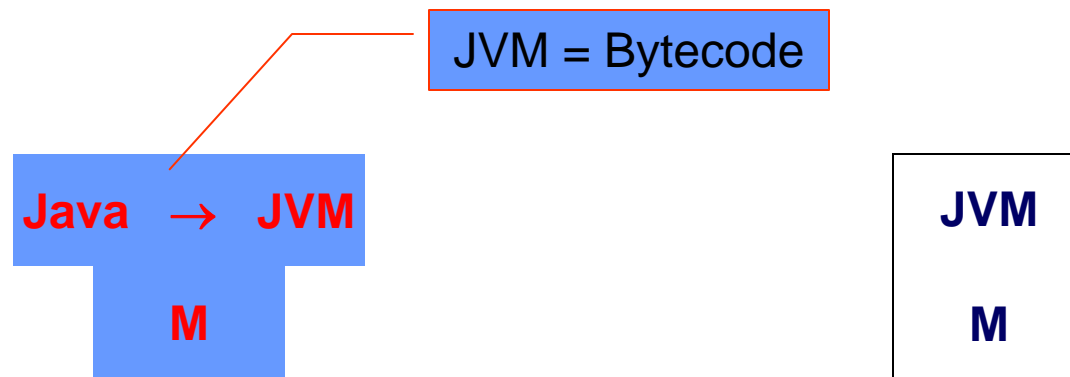
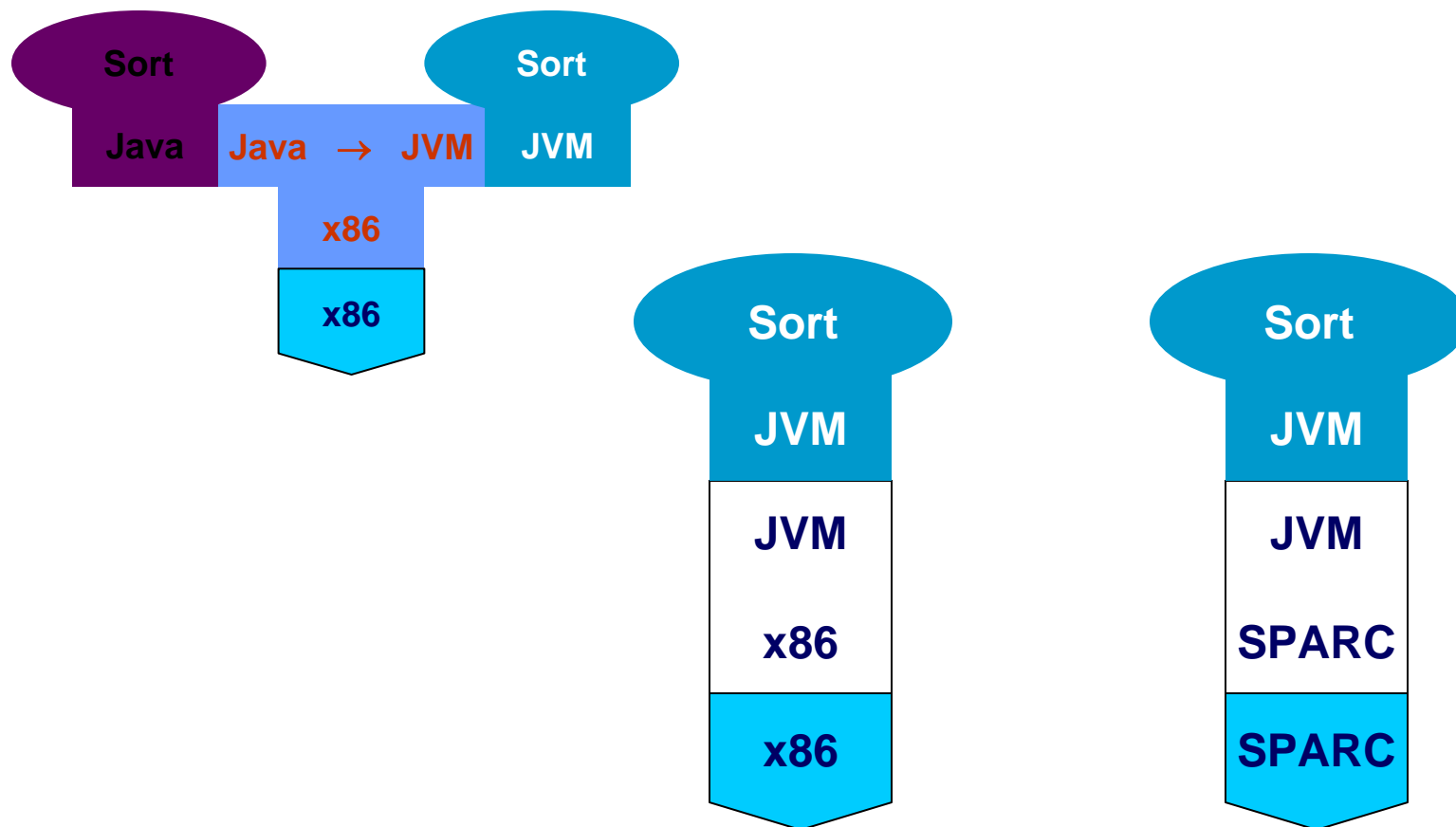


Diagrama de lápida



El compilador de Java

- Está realmente escrito en Java
- La JVM está escrita en Java
- La JVM se puede escribir en otro lenguaje
- ¿Cómo se hizo eso?

Bootstrapping

- *Bootstrapping* es compilar un compilador consigo mismo...
- Algunas personas lo consideran un hito o prueba de calidad.
 - ¿Es razonable?

Diagramas de lápida

- *Bootstrapping*
 - El compilador L está escrito en L
- *Full bootstrap*
 - Se inicia desde 0
- *Half bootstrap*
 - Se inicia desde otra máquina

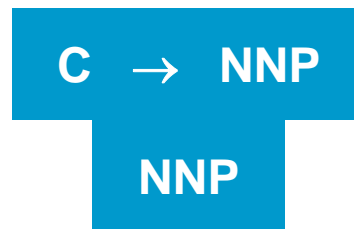


Diagrama de lápida

- Full Bootstrap

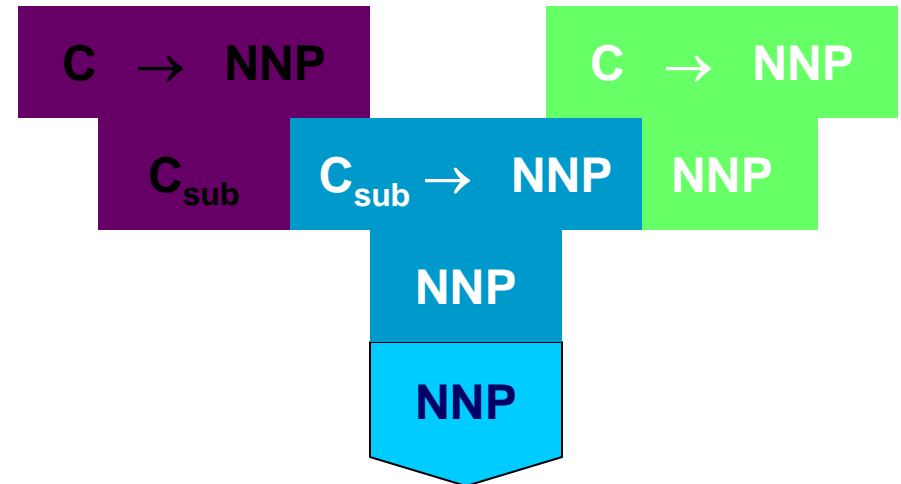
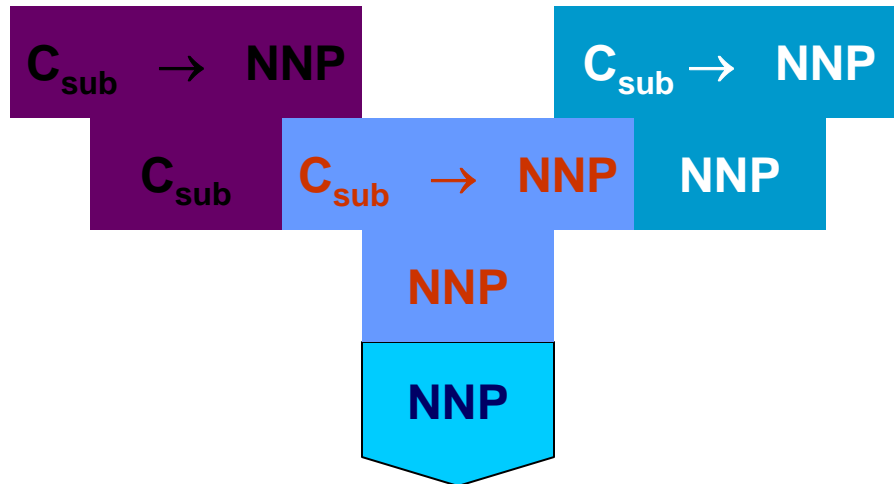
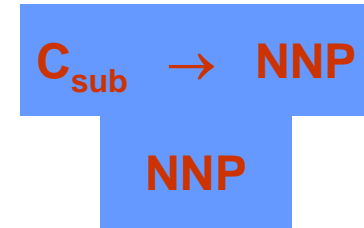


Diagrama de lápida

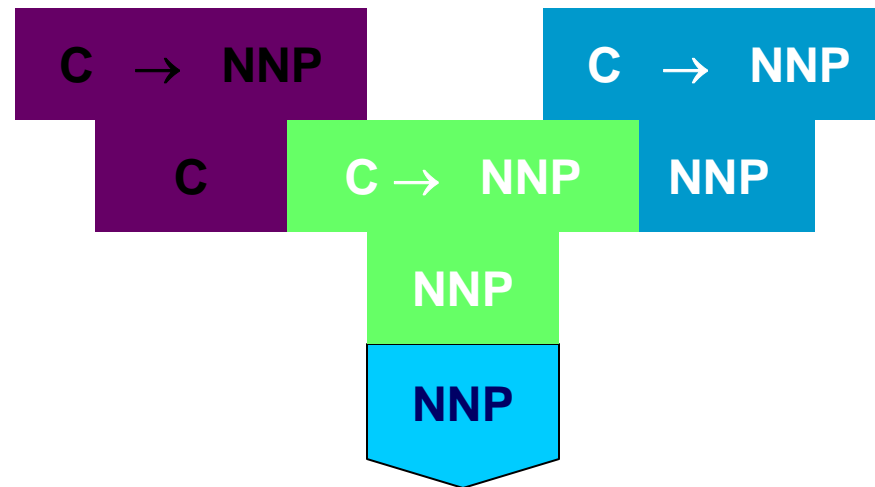


Diagrama de lápida

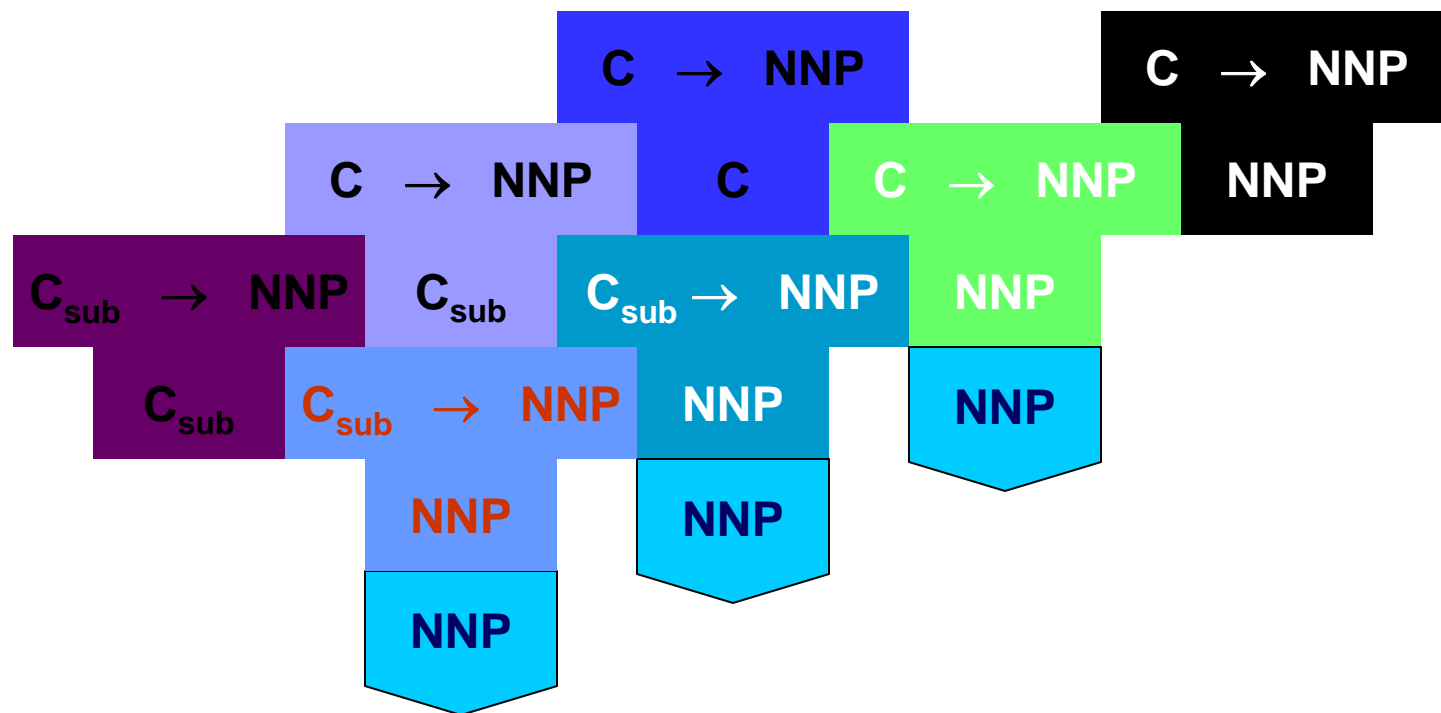
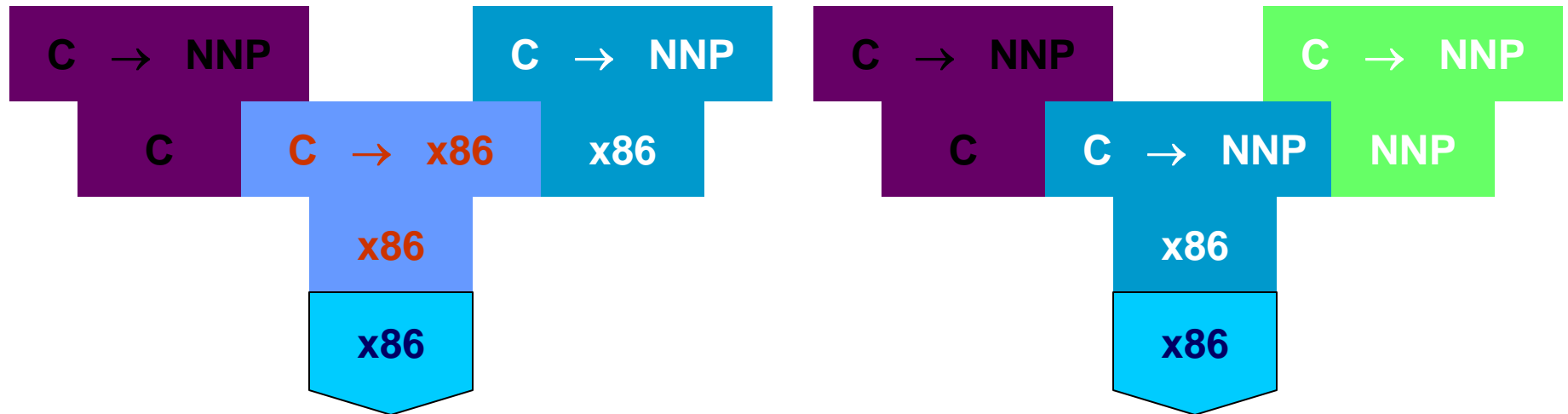
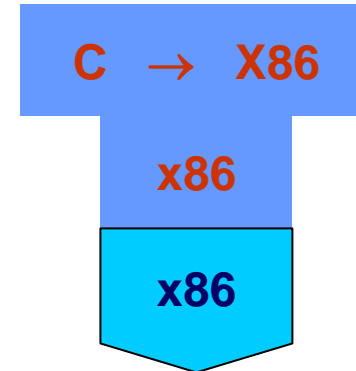


Diagrama de lápida

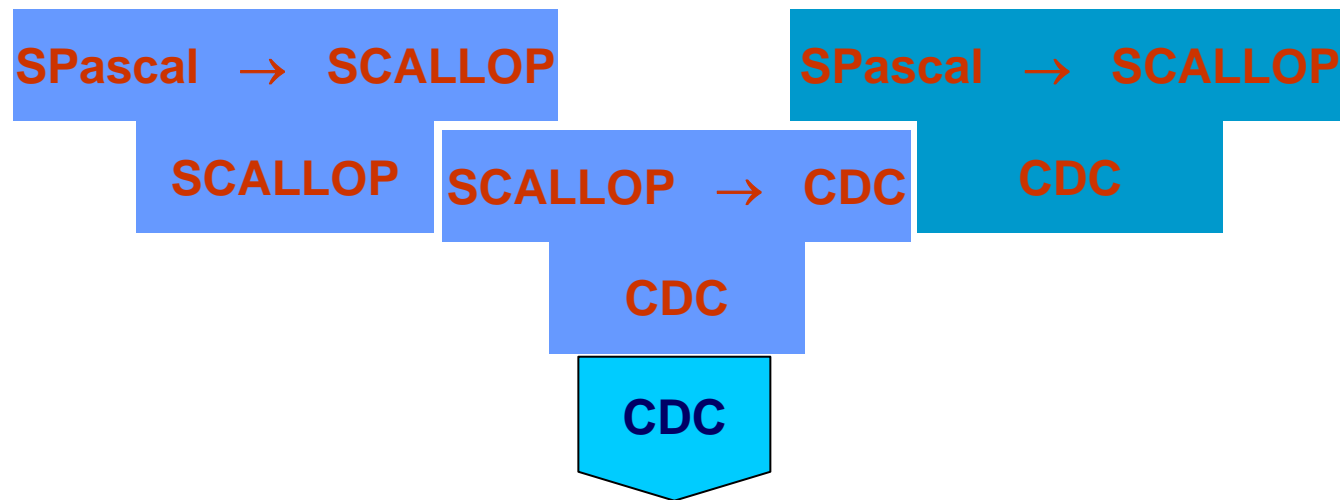
- Half Bootstrap



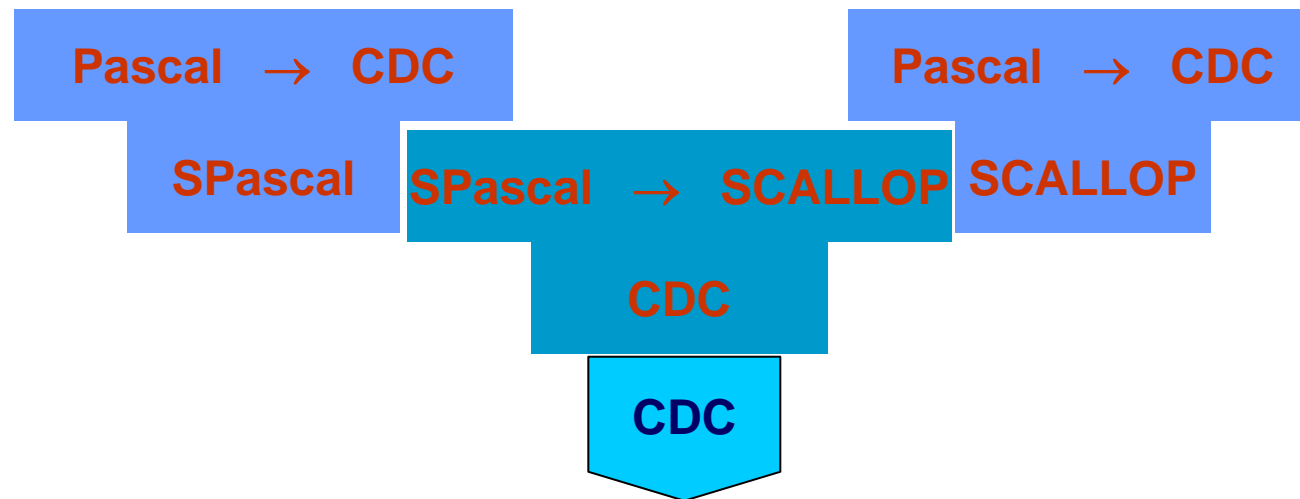
Antes de Pascal-P



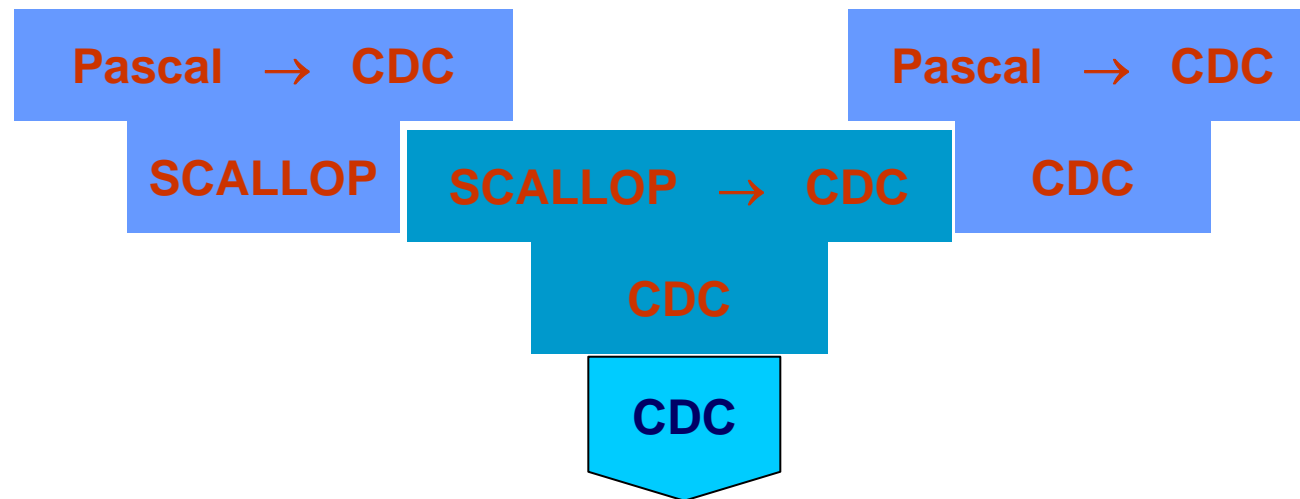
Antes de Pascal-P



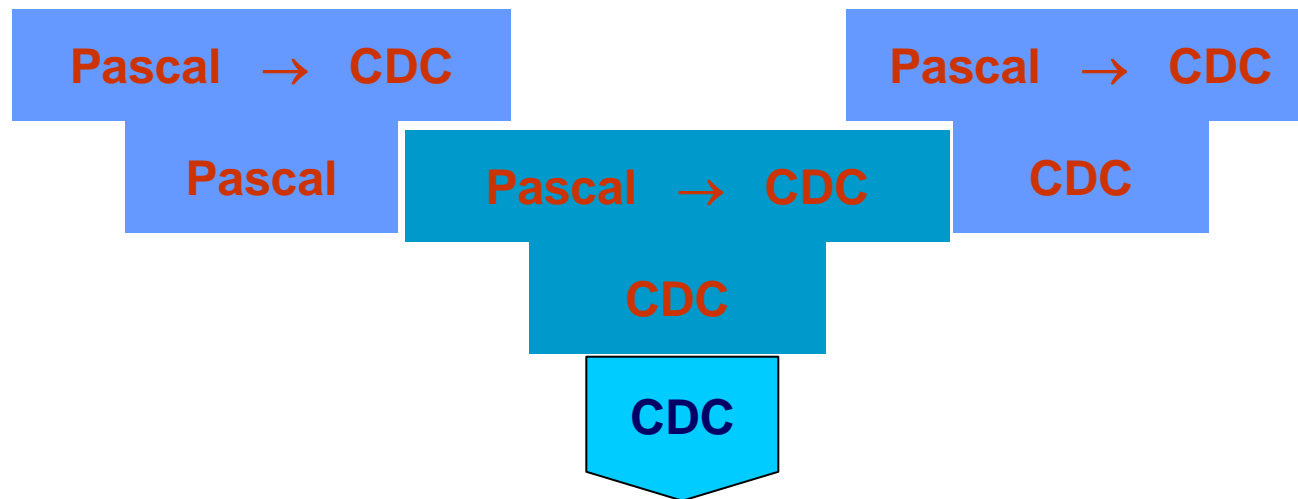
Antes de Pascal-P



Antes de Pascal-P

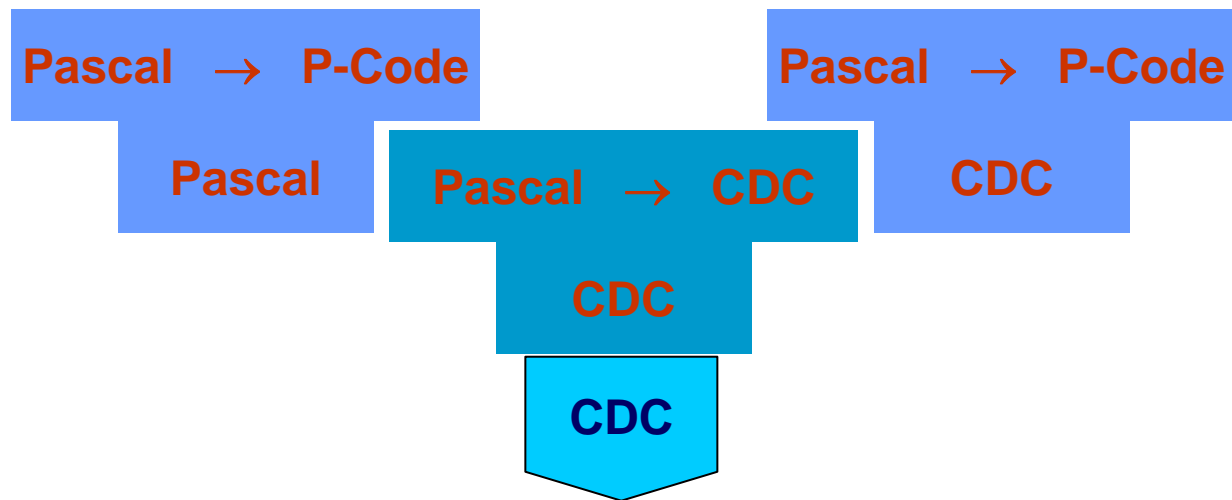


Antes de Pascal-P

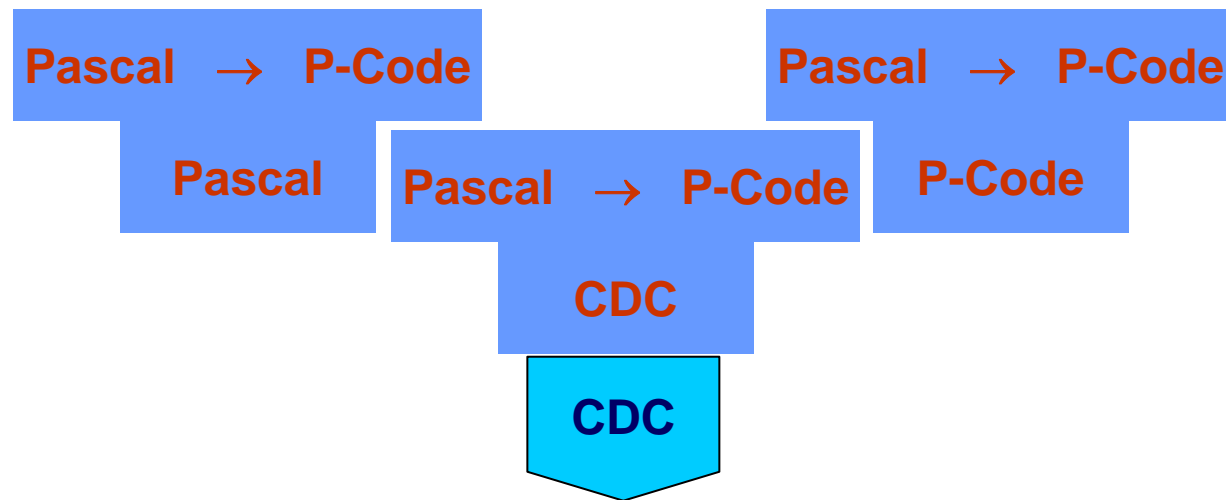


Buscamos una mejor generación de código

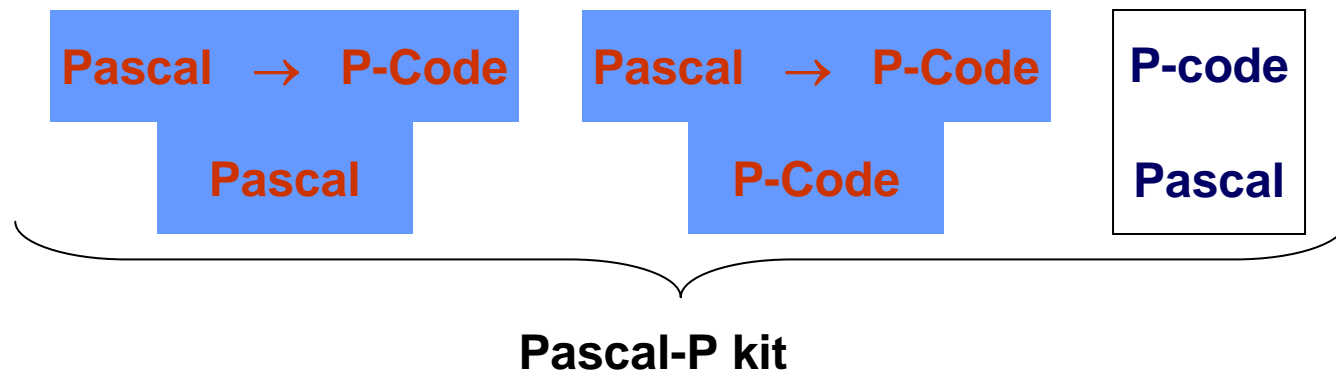
Evolución de Pascal-P



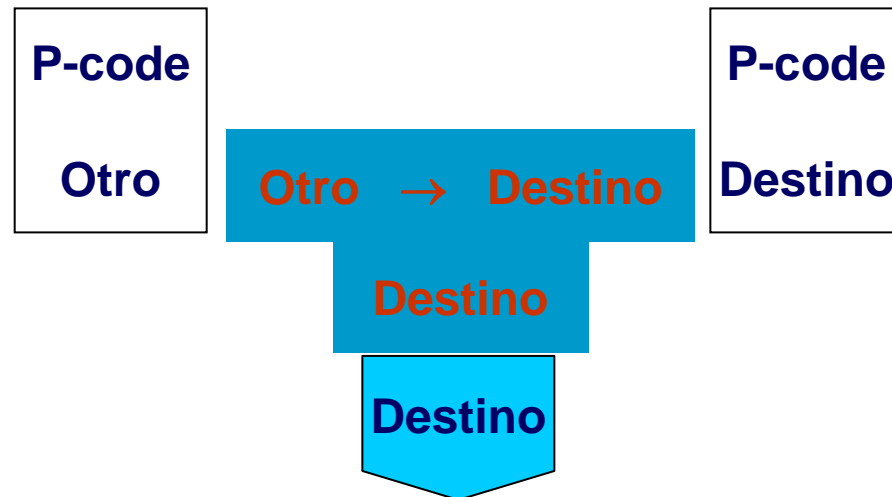
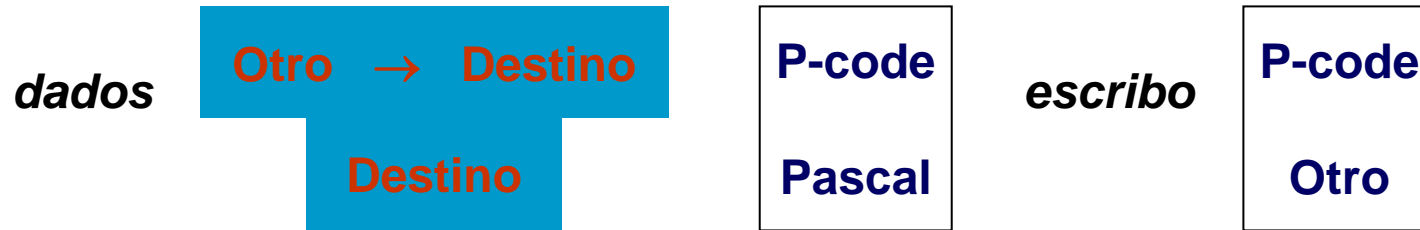
Evolución de Pascal-P



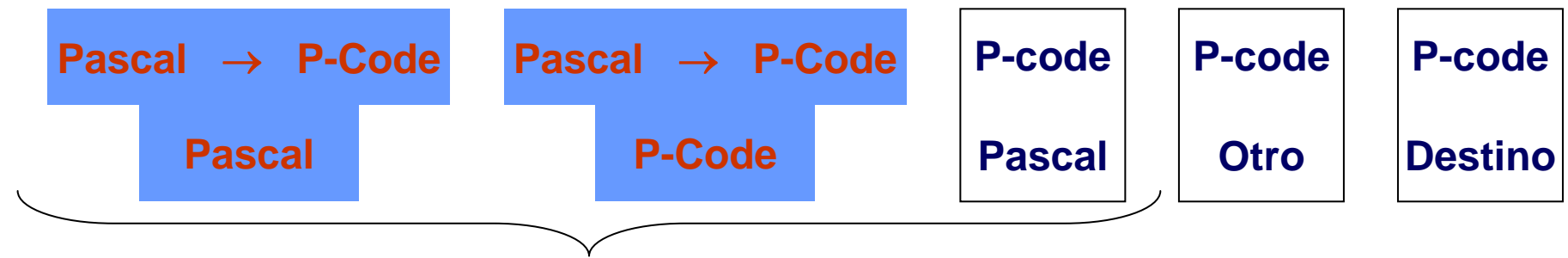
Kit Pascal-P



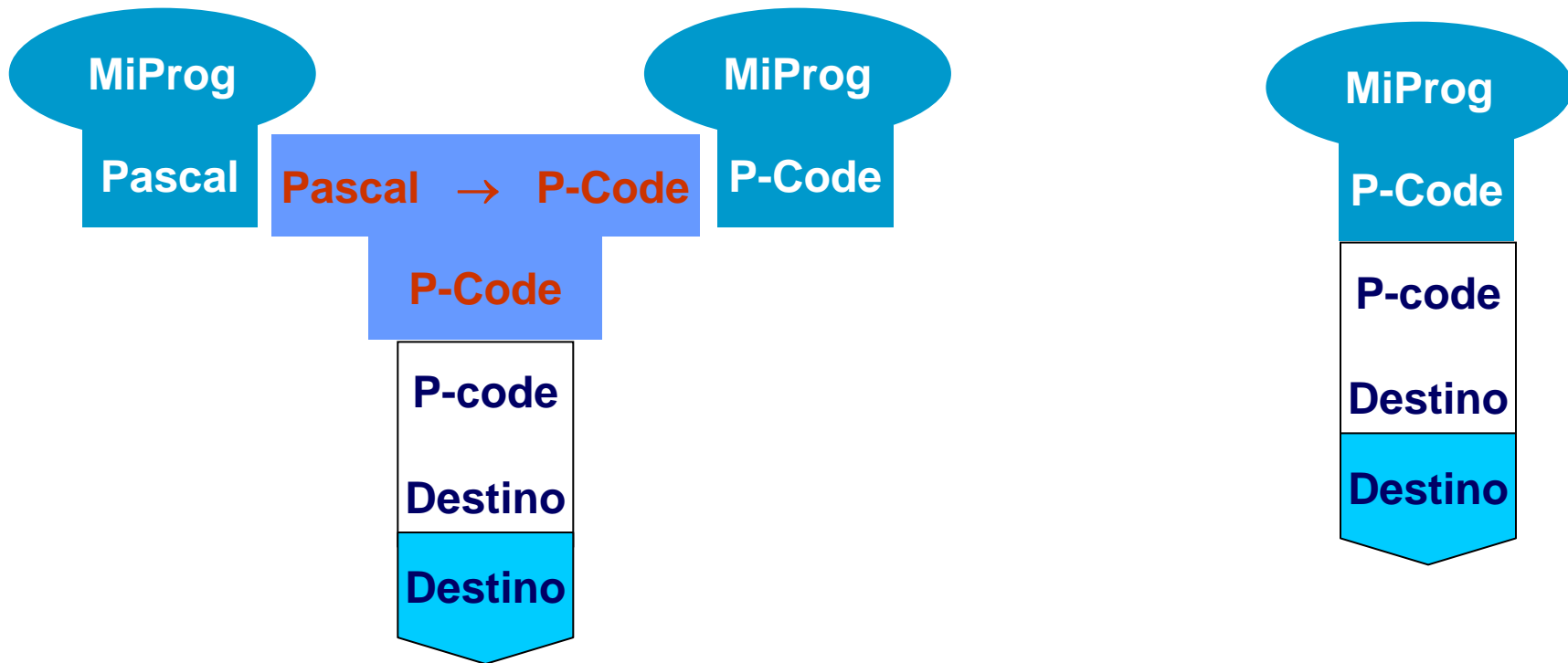
Portar Pascal-P



Portar Pascal-P

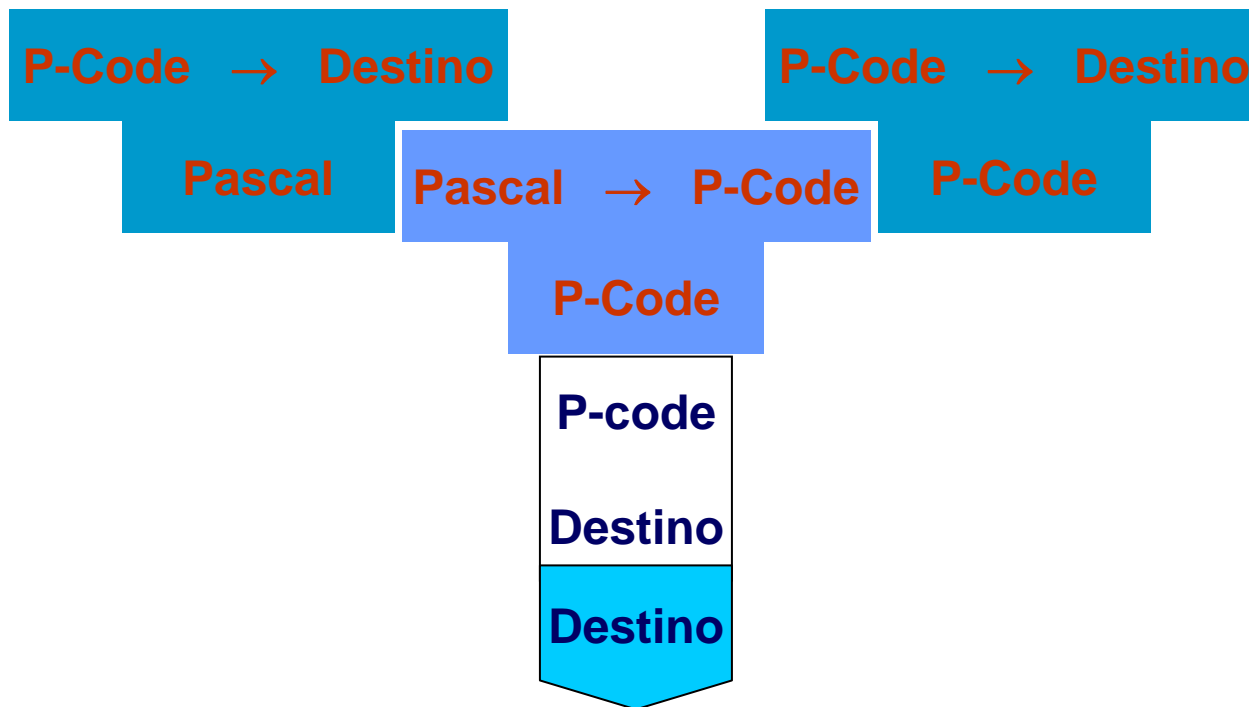


Pascal-P kit



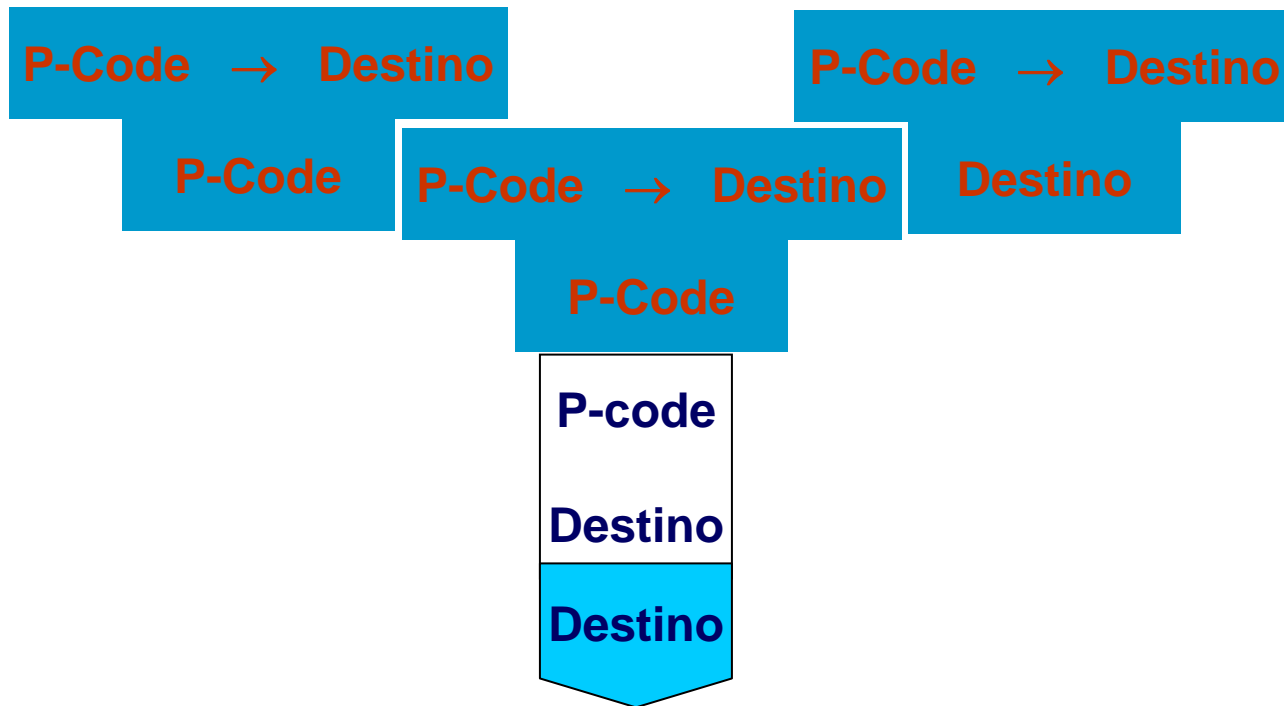
Mejorar eficiencia de Pascal-P

Escribo un generador de código nativo a partir de P-Code



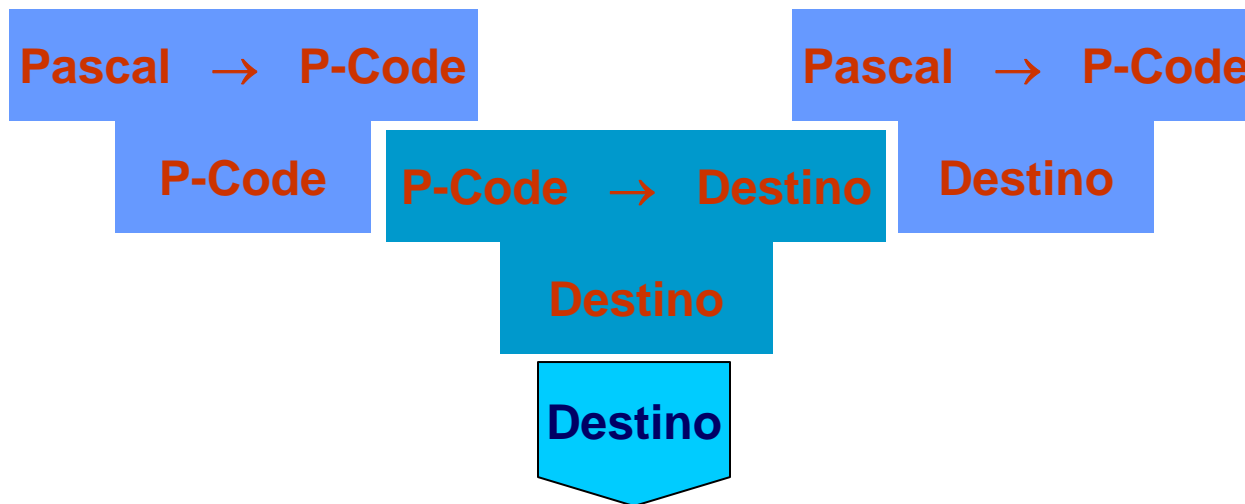
Mejorar eficiencia de Pascal-P

Hago *bootstrap*



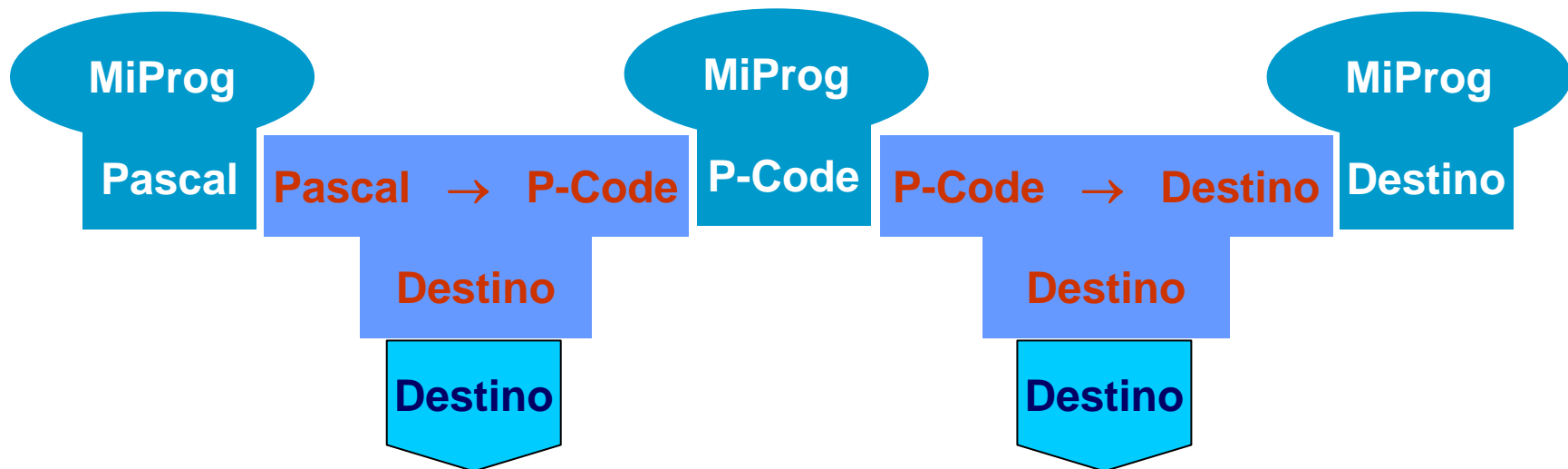
Compilador nativo de Pascal en dos etapas

Obtengo compilador nativo

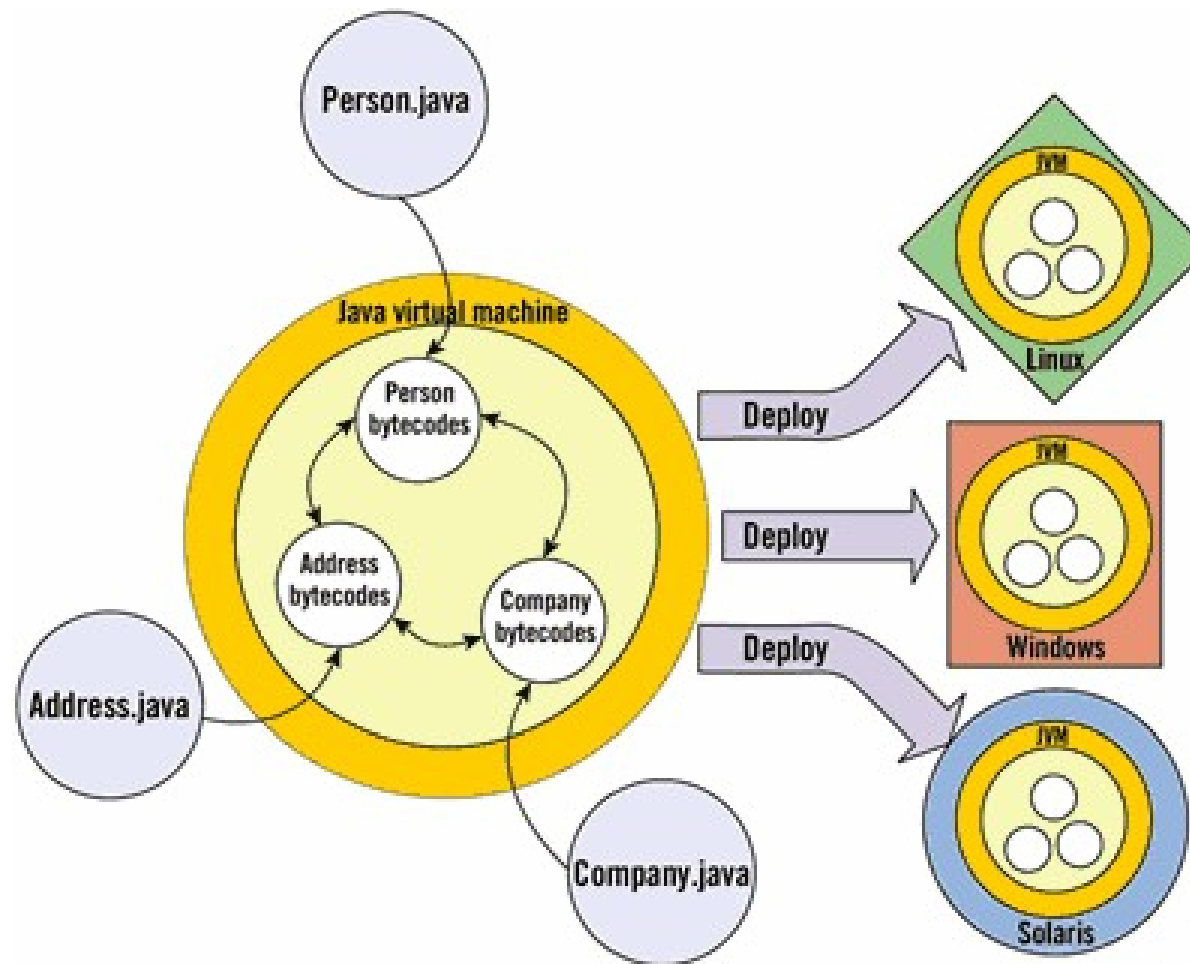


Compilador nativo de Pascal en dos etapas

Compilador en dos etapas



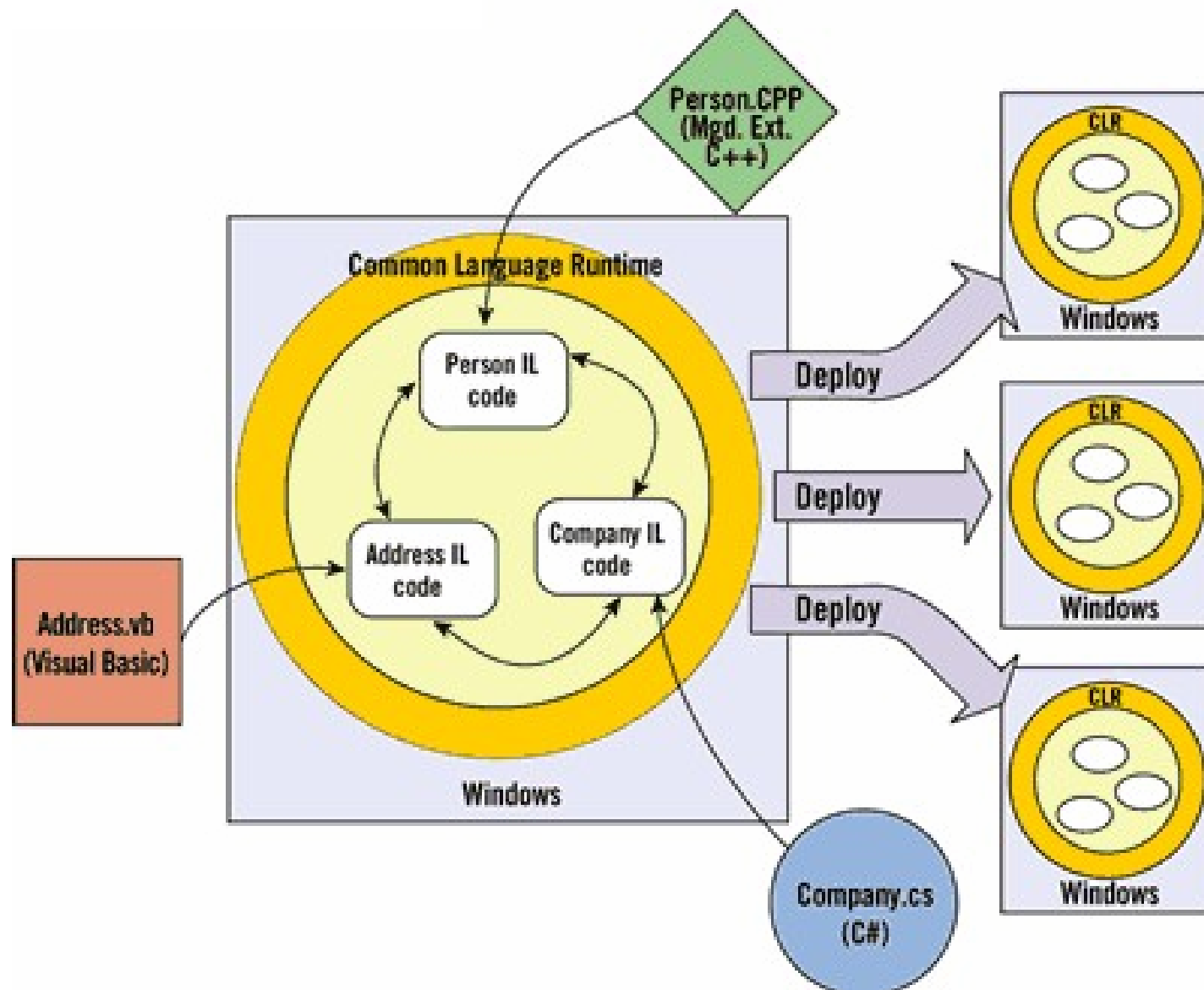
Ejecución de Java (JVM)



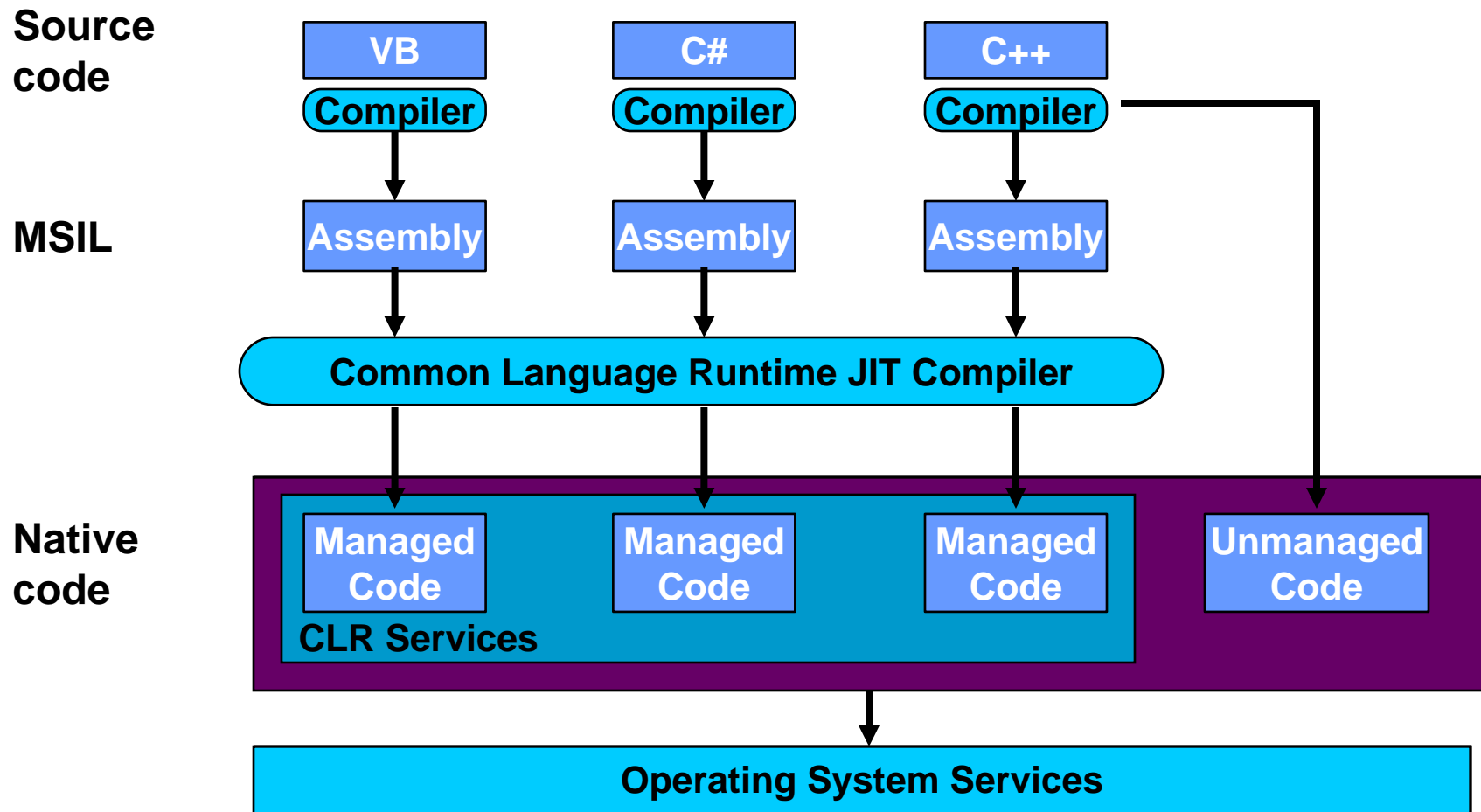
Compilación en .NET



Ejecución en .NET (IL, CLR)



Modelo de ejecución del CLR



Modelo de ejecución

