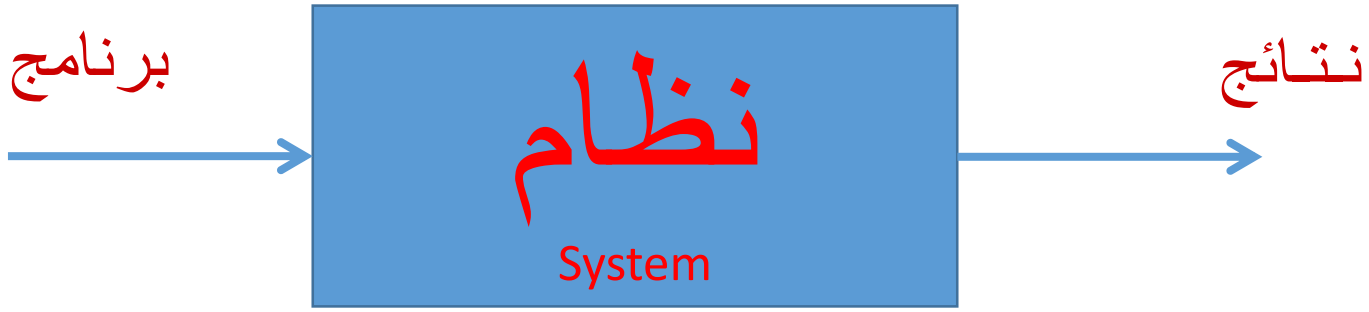
ITSE321
Software Construction
بنـــاء البرمجيات

المحاضر: د. عبدالسلام النويصري
a.nwesri@uot.edu.ly

Implementing a programming Language

تجسيد أو تطبيق أوامر لغة البرمجة

Implementing a programming Language



يوجد نظامين لتجسيد أوامر لغة البرمجة

Implement أسلوبين

المترجم Compiler

المفسر Interpreter

المترجمات Compilers

- **ماهو المترجم Compiler؟**
- نظام يترجم برنامج مكتوب بلغة–مستوى–عالي إلى برنامج مكتوب بلغة أخرى (مستوى أدنى) قابل للتنفيذ executable والتوظيف.
  - ونتوقع أن البرنامج الذي أنتجه المترجم سيكون بمميزات أفضل من البرنامج الأصلي. ما تلك المميزات؟ تفهمه الآلة، تنفذه الآلة مباشرة، مضبوط
- **Compiler** هو أيضاً برنامج (تطبيق) مدخلاته عبارة عن برنامج بلغة ما (اللغة المصدر Source Language) يتم ترجمتها إلى مخرجات عبارة عن برنامج بلغة أخرى (لغة الهدف Target Language).
  - **لغة المصدر** مثل جافا,VB, Python, C, C++, C#, Fortran, Pascal أو غيرها.
  - **لغة الهدف** مثل لغة التجميع أو لغة الآلة.

المترجمات Compilers

- ربما ترجمة الكلمة الإنجليزية Compiler إلى "مترجم" لا تعطي المعنى اللغوي السليم.
- كلمة Compile في الإنجليزية يقصد بها: **تجميع** أشياء كالمعلومات من مصادر مختلفة – كمجلات, وثائق, كتب, تقارير, سجلات وغيرها – ثم **تصنيفها**, ثم **التأليف** بينها لصياغة شئ جديد له معنى.

BASIC REQUIREMENTS FOR COMPILERS

المتطلبات الأساسية للمترجم

- **من مواصفات المترجم**
  - إنتاج تعليمات صحيحة correct code

(byte code from java code, machine code from C code)

  - تنفيذ سريع لعمليات الترجمة
  - البرنامج الناتج يجب أن ينفذ بسرعة run fast
  - وقت الترجمة يتناسب مع حجم البرنامج
  - كشف الأخطاء بدقة lexical and syntax errors
  - دعم تقاطع استدعاء اللغات cross language calls/FFI

مثال: Java Native interface (JNI)

A **foreign function interface** (**FFI**) is a mechanism by which a program written in one programming language can call routines or make use of services written in another

ABSTRACT VIEW OF COMPILERS

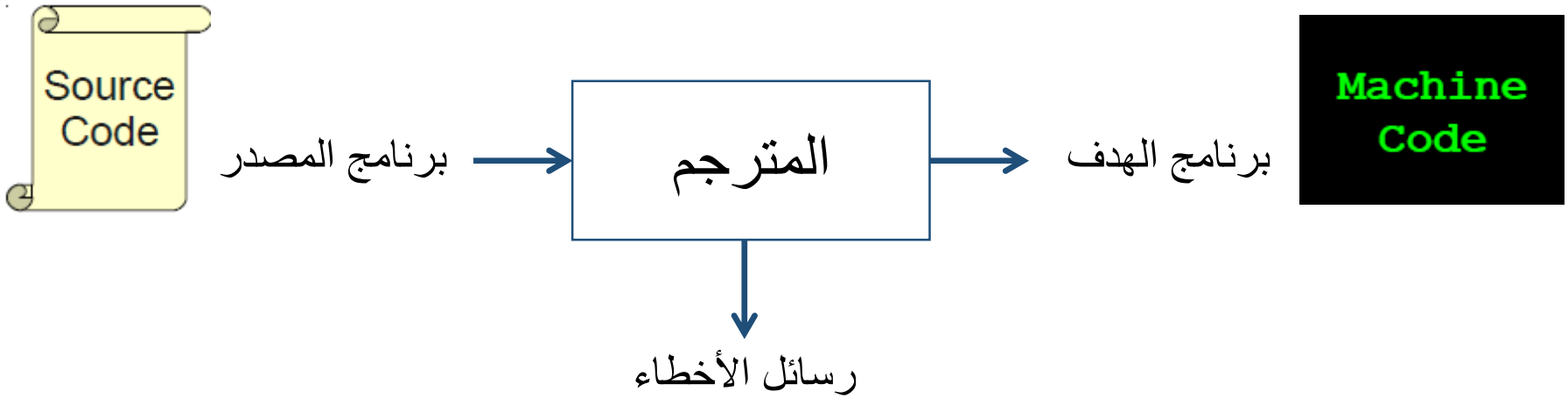
ملخص فكرة المترجمات

- المترجم ينجز عملية الترجمة من خلال عدة خطوات تؤديها عدة مكونات

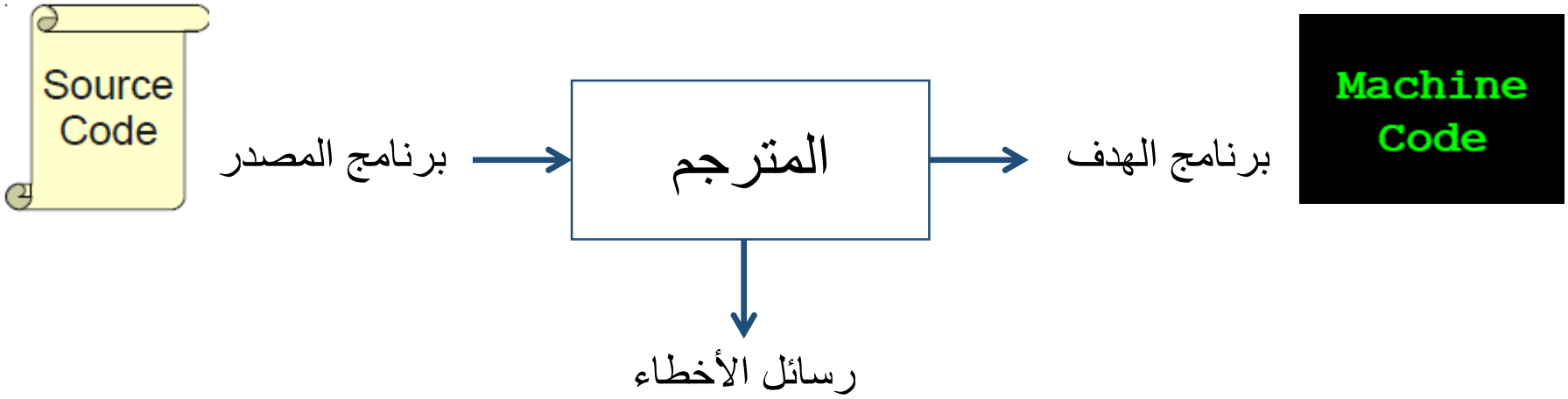
Several steps done by several components

- عادة للمترجم مكونات خاصة بالتدقيق في قواعد المفردات وقواعد النصوص

Verifying lexical and syntax rules



جهاز المترجم (Compiler)

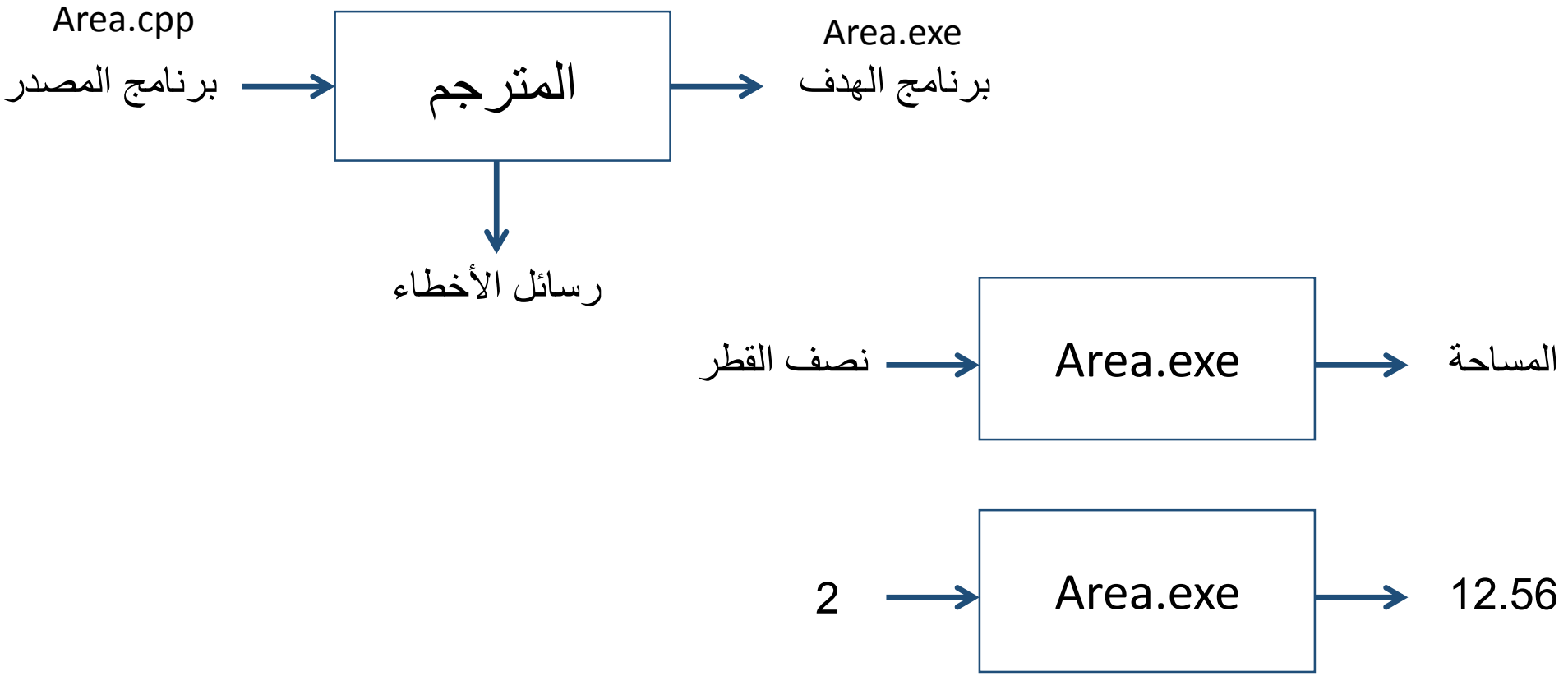


- خلال عملية الترجمة يؤدي المترجم وظائف مساعدة وهامة للمبرمج وهي إحالة تقارير عن الأخطاء والتحذيرات التي تيسر للمبرمج تصحيح أو تعديل برنامج المصدر وذلك لإتمام عملية الترجمة بنجاح.
- **المترجم يسد الفراغات بين المبرمج و الكيان المادي.**

تنفيذ running برنامج الهدف

برنامج الهدف target program

- إذا كان برنامج الهدف قابل للتنفيذ فإنه بإمكان المستخدم استدعاءه لمعالجة بعض المدخلات ومن تم إنتاج مخرجات.



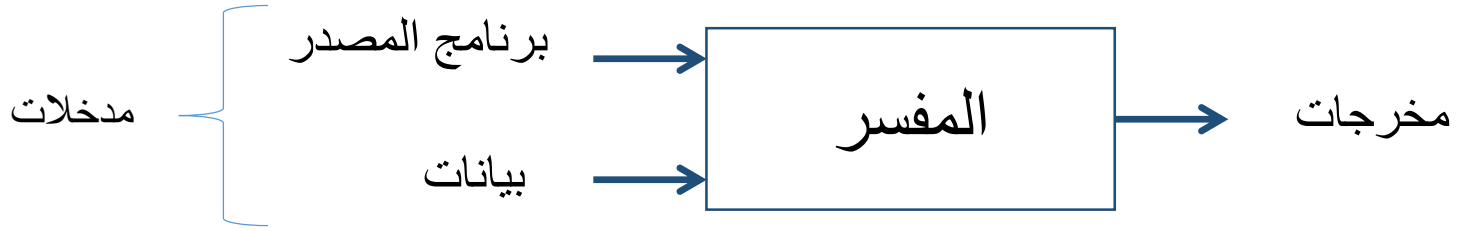
المفسر Interpreter

- **ما هو المفسر Interpreter؟**
  - نظام يقرأ برنامج ويخرج لنا نتائج results مباشرة

- **في هذه المادة سنهتم بالمترجم وإن كان هنالك مفاهيم كثيرة مشتركة بينهما**

المفسر (Interpreter)

- المفسر (مترجم فوري/آني) من الأنماط المعروفة في وسط معالجة اللغات فهو يقوم بتفسير وتنفيذ التعليمات كل على حده. أي لا ينتقل إلى الأمر التالي حتى ينفذ الأمر الحالي. مثال على ذلك لغة الاستفسار/الاستعلام SQL التابعة لقواعد البيانات.
- المفسر لا ينتج برنامج الهدف لذلك فعملية تنفيذ التلعيمات دائما تعتمد على برنامج المصدر الخالي من الأخطاء والبيانات التي تعطى تباعا حسب طلب كل تعليمة/أمر.



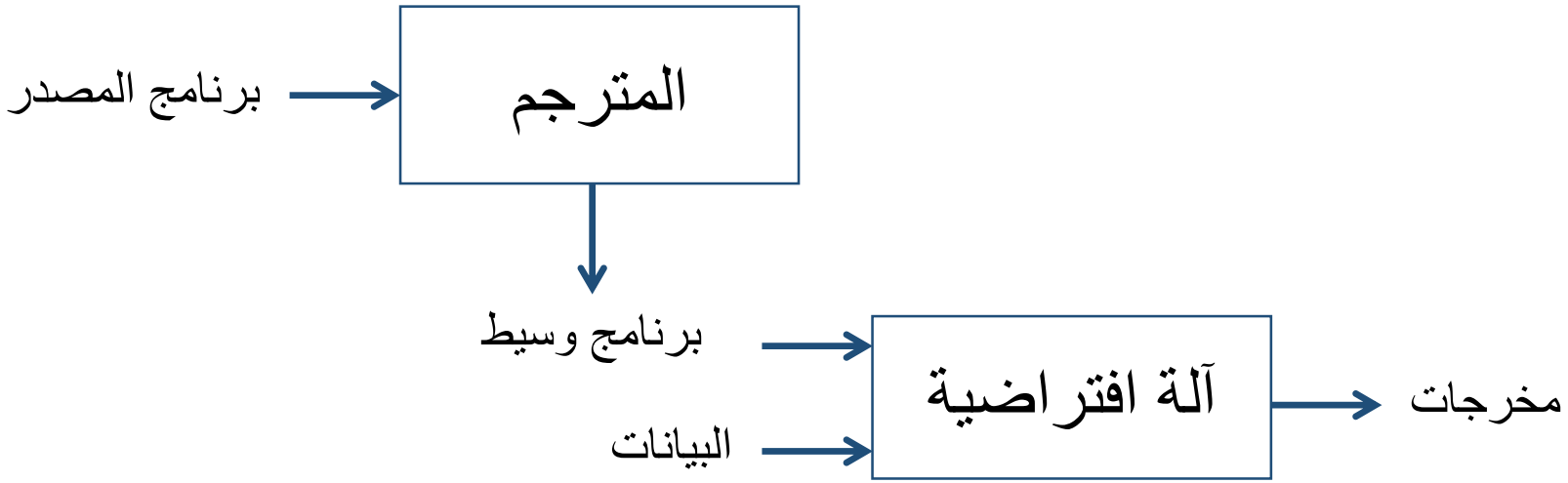
مفارقات المترجم مع المفسر

- تحويل البيانات إلى مخرجات بواسطة برنامج لغة الآلة (برنامج الهدف) المنتج من المترجم يكون أسرع من المفسر. **علل**! لايوجد مرحلة ترجمة
- يمكن للمفسر أن يعطي تشخيص أفضل للأخطاء. **علل**! لا تتراكم الأخطاء
- المفسر لا ينتج ملف جديد قابل للتنفيذ بذلك يوفر مساحة تخزين في الحاسوب.
- يحتاج المفسر إلى معاودة تفسير الأوامر لتنفيذها, مما يترتب عليه بطؤ إجراءات التنفيذ.
  - تكرار عملية التفسير للبرنامج كاملاً كلما نحتاج لتنفيذه
  - تكرار تفسر الأوامر داخل الحلقات التكرارية عند كل لفة

الطرق المهجنة Hybrid Methods

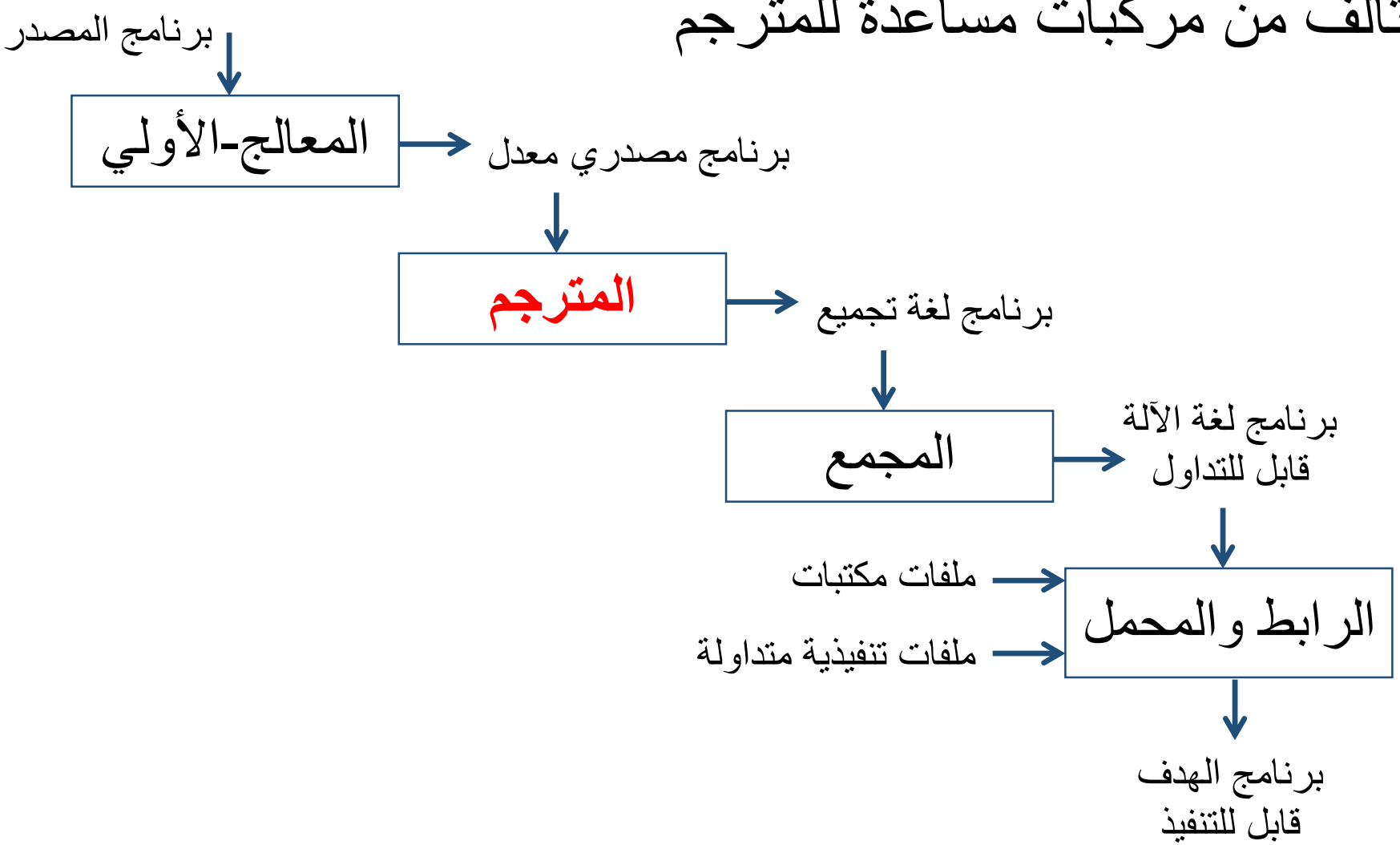
- توجد طرق لمعالجة البرامج تجمع بين الترجمة والتفسير وتعرف بالمهجنة.
- يترجم البرنامج المصدري في نظام حاسوب مستقل ويتم التنفيذ عبر آلة افتراضية Virtual Machine.
- مثال على هذا, ملف الرمز المصدري لبرنامج بلغة جافا (Java source code) يتم في البداية ترجمته إلى برنامج/ملف وسيط يسمى ترميز_بايت (Bytecodes) ثم تُفسر عن طريق آلة افتراضية.
- الفائدة هي أن البرنامج يمكن أن يترجم في نظام حاسوب وينفذ في نظام حاسوب مختلف, مثلا عبر الشبكات المحلية أو الدولية, بين PC و Mac.

الطرق المهجنة Hybrid Methods



نظام الترجمة Compilation System

- يتألف من مركبات مساعدة للمترجم



أدوات مساعدة للمترجم

- بالإضافة إلى المترجم, هنالك عدد من البرامج المساعدة التي نحتاجها للحصول على برنامج الهدف التنفيذي.
- برنامج المصدر يمكن أن يكون مقسما إلى أجزاء (Modules) مستقلة يخزن كل منها في ملف منفرد, ومهمة تجميع أجزاءه يتولاها برنامج يعرف (Preprocessor) أو المعالج-الأولي.
- إذاً البرنامج المصدري سيتم تحويره من قبل المعالج-الأولي.

أدوات مساعدة للمترجم

- بعد ذلك, برنامج المصدر المعدل/المحَور يمرر إلى المترجم الذي بدوره يمكن أن ينتج برنامج بلغة التجميع (Assembly Language) ذلك لأن برنامج لغة التجميع سهل الاستخلاص من برنامج المصدر وسهل التصحيح (Debugging).
- بعدها, تتم معالجة ملف برنامج لغة التجميع بواسطة المجمع (Assembler) ليخرج منه برنامج بلغة الآلة قابل للتداول والتنفيذ.
- عادة, البرامج الضخمة تترجم مقطعة على هيئة جزئيات لكي يمكن تداول الملفات التنفيذية الجزئية ويمكن ربطها ببعض وبمكتبات النظام أو مكتبات برامج تطبيقية أخرى لتكون مع بعض برنامج تنفيذي مطلوب.

أدوات مساعدة للمترجم

- يقوم برنامج الرابط (Linker) بمسألة ربط العناوين بين الملفات المنفصلة عندما يستدعي/يشير ملف تنفيذي ما لوظيفة عنوانها موجود بملف تنفيذي آخر أومكتبة ما.
- وأخيراً يضع المحمل (Loader) الملفات التنفيذية مع بعض في الذاكرة ليتم تنفيذها.

مخرجات المترجم

- مترجمات تنتج مخرجات هي عبارة عن برامج تنفيذية بلغة الآلة

Machine code

- مترجمات تنتج مخرجات هي عبارة عن برامج وسطية تنفذ بآلة افتراضية

Byte Code runs on Virtual Machine

- مترجمات تنتج مخرجات هي عبارة عن برامج برامج بلغة أخرى عالية المستوى

Convert C++ code to Java code

لمحة من تاريخ المترجمات

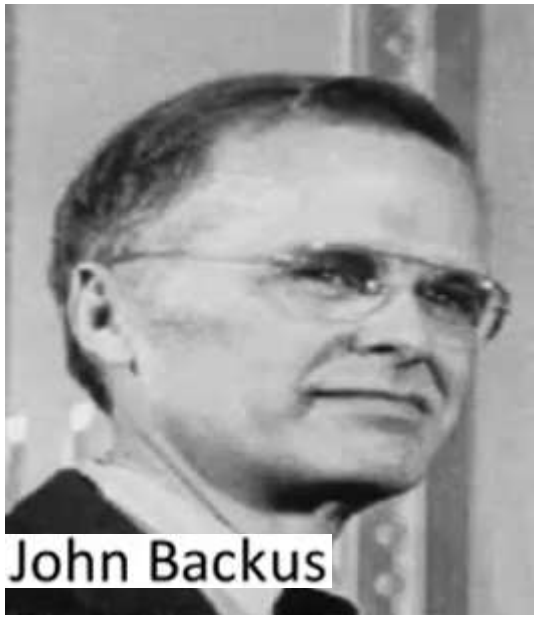
IBM 704

- بدأ تطوير المترجم في الخمسينات
- 704 هو أول آلة ترجمة تنجح تجارياً

صنع شركة IBM في سنة 1954

- تكلفة Software كانت أكثر بكثير من تكلفة Hardware
- مع إن الكيات المادي Hardware كان غالي الثمن في تلك الفترة وربما أكثر من الآن
- وهذا جعل المطورين يفكرون كيف تكون البرمجيات أفضل إنتاجية وقابلية للتسويق

لمحة من تاريخ المترجمات



- **Speedcoding** كانت من أول الأعمال لتحسين البرمجة في 1953 على يد **جون باكس**
- **Speedcoding** كانت مثل المفسر
  - ✓ اسرع في تطوير البرامج (أفضل إنتاجية)
- ولكن سرعة تنفيذ البرامج على **Speedcoding** كانت 10 إلى 30 مرة أقل من البرامج التي كانت تدخل للآلة مباشرة باليد
  - وهذا مازال من سمات المفسر مقارنة ببرامج لغة الآلة
- مفسر **Speedcoding** أخذ 300 بايت وهذا كان 30% من الذاكرة

لمحة من تاريخ المترجمات

FORTRAN I

- **Speedcoding** لم تحض بشعبية كبيرة
- **جون باكس** اعتبارها واعدة ففكر في مشروع جديد
- اكثر التطبيقات كانت علمية Scientific، كيف نحول الصيغ الرياضية لأوامر برمجية في **Speedcoding** كانت المعادلات تفسر لكي يتم تنفيذها وهذا يستهلك وقت، فجاءت فكرة صياغة المعادلات من المستوى الأعلى للغة لتكون جاهزة للتنفيذ **For**mula **Tran**slation (1954 - 1957).
- استغرق 3 سنوات بناء مترجم فورتران بدلاً سنة واحدة كما كانوا يظنون
- مع سنة 1958 حوالي 50% من البرمجيات كانت **بالفورتران**

لمحة من تاريخ المترجمات

- أول مترجم ناجح زاد من إنتاجية البرمجيات FORTRAN I
- ساهم في الإنتاج العلمي
- ساهم بشكل كبير في ربط النظريات بالتطبيق (الممارسة): Computer Science

Theory and Practice

مهارت هندسية فهم عميق لنظريات

- المبادئ والعمليات التي بني عليها مترجم FORTRAN I مازالت تؤثر في المترجمات الحديثة إلى اليوم

نهاية المحاضرة,
موضوعنا التالي:

عملية الترجمة

Compilation Process