Compiler؟ ما هو المترجم

المترجم عبارة عن برنامج يقوم بتحويل الشيفرة المصدرية المكتوبة بإحدى لغات البرمجة عالية المستوى إلى تمثيلٍ منخفض المستوى مع المحافظة على معنى وهدف الشيفرة المصدرية. يتم استخدام مصطلح "**Source Code الشيفرة المصدرية**" للإشارة إلى النص البرمجيّ المكتوب بإحدى اللغات عالية المستوى، بينما يتم استخدام مصطلح "**Target الشيفرة الهدف**" للإشارة للشيفرة الناتجة عن عمل المترجم، والتي تكون عادةً بإحدى اللغات منخفضة المُستوى. بالإضافة إلى ذلك، يقوم "**Code** المترجم بإجراء عمليات تحسين على الشيفرة المصدرية لجعلها أكثر كفاءة من ناحية سرعة التنفيذ وحجم الذاكرة التي سيشغلها البرنامج.

نواع المترجمات

يمكن تقسيم المترجمات لثلاث أنواع من ناحية كيفية أدائها لمهمتها وتوليدها للشيفرة الهدف:

المترجمات أحادية المسار Single-Pass Compiler: بهذا النمط من المترجمات يتم ترجمة الشيفرة المصدرية عالية المستوى بشكلٍ مباشر للشيفرة منخفضة المستوى بدون أي مراحل وسيطة.

المترجمات ثنائية المسار Two-Pass Compiler: بهذا النمط من المترجمات يتم توليد **شيفرة مصدرية وسيطة Intermediate Code** وظيفة القسم القريب هي تحويل الشيفرة المصدرية للتمثيل الوسيط وإجراء عمليات تحليل النص والمفردات بالإضافة بتحويل الشيفرة المصدرية الوسيطة إلى الشيفرة الهدف مُنخفضة Back-End لعمليات التحسين، ومن ثم يقوم القسم البعيد Back-End والقسم البعيد Front-End وبحيث يُفهم عمل المترجم ككل على أنه قسمين: القسم القريب .End المستوى.

المُترجمات متعددة المسار Mutlipass Compiler: بدلًا من تجزئة عملية الترجمة لقسمين فقط، يوجد مترجمات تقوم بتقسيم عملية الترجمة لعدة أجزاء وينتج عن كلٍ منها شيفرة مصدرية وسيطة، وخرج كل مرحلة يمثل دخل المرحلة التالية. النتيجة النهائية هي الشيفرة المصدرية منخفضة المستوى.

Language Processing System نظام معالجة لغات البرمجة

قبل الدخول بتفاصيل عمل المترجم نفسه والعمليات التي يجريها على الشيفرة المصدرية، يجب توضيح أمر هام: المترجم نفسه هو جزء أكبر من نظام متكامل يتم عبره توليد التعليمات المكتوبة بلغة الآلة والتي يستطيع المعالج فهمها. خرج المترجم غير قابل للتنفيذ مباشرةً من قبل المعالج، كما أن المترجم يعتمد على مراحل أخرى قبله كي يكون قادرًا على أداء مهمته بأفضل شكلٍ ممكن.

للإشارة إلى المراحل المختلفة "**Language Processing System نظام معالجة لغات البرمجة**" يتم استخدام مصطلح التي يتم عبرها تحويل النص البرمجيّ عالي المستوى إلى لغة الآلة، ويتكون هذا النظام من المكونات والمراحل التالية:

الشيفرة المصدرية Source Code: وهي تمثل البرنامج المكتوب بإحدى لغات البرمجة عالية المستوى والمطلوب تنفيذها من قبل المعالج.

مرحلة ما قبل المعالجة Pre-Processing: تعتبر هذه المرحلة جزءًا من عملية الترجمة نفسها، حيث يتم توليد تمثيل جديد للشيفرة المصدرية عالية المستوى يتم استخدامه من قبل المترجم، والهدف الأساسيّ من هذه المرحلة هي تحديد موّجهات ما قبل البرمجة، التعاريف، المكاتب المضمنة.

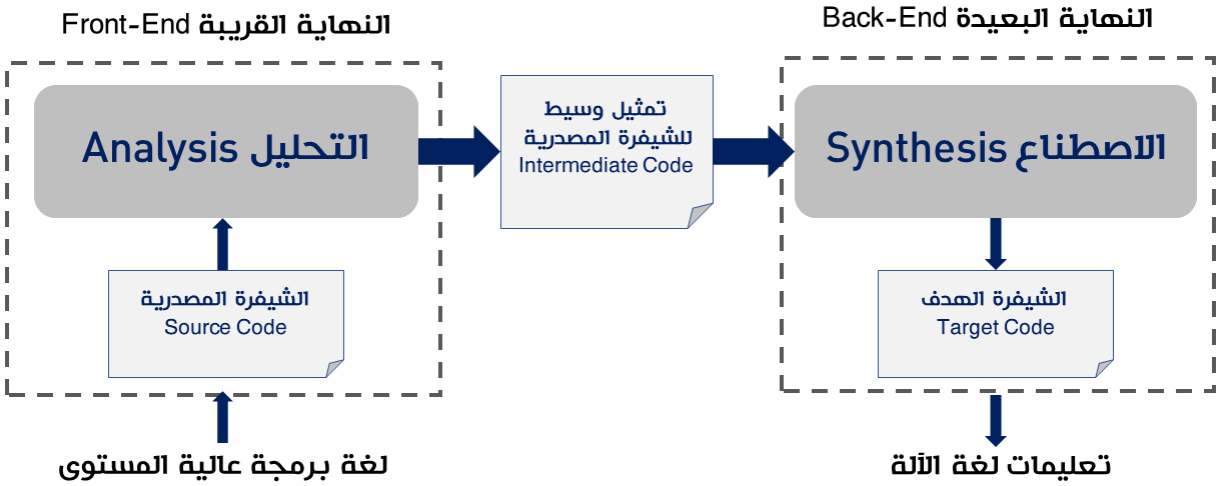
المُفسر Interpreter: المفسر عبارة عن برنامج مشابه للمترجم من ناحية قيامه بتحويل الشيفرة المصدرية لتمثيلٍ منخفض المستوى، إلا أن الفرق الأساسيّ يكمن بأن المفسر يقوم بقراءة الشيفرة المصدرية وتنفيذها تعليمةً تلو الأخرى، بينما يقوم المترجم بتحليل كامل النص البرمجيّ ومن ثم توليد الشيفرة منخفضة المستوى. تعتمد بعض اللغات على المفسر كوسيلةٍ لتنفيذ البرامج بدلًا من المترجم.

المُجمّع Assembler: يقوم المجمع بأخذ ناتج عمل المترجم وتحويله لتعليماتٍ مكتوبة بلغة الآلة. ناتج عمل المفسر هو ما الذي يتضمن التعليمات التنفيذية بالإضافة للمعطيات اللازمة لتخزين هذه "Object-File يطلق عليه اسم "الملف الكائني **Relocatable تعليمات الآلة القابلة للنقل**" التعليمات ضمن الذاكرة. يتم أيضًا تعريف ناتج عمل المجمع على أنه "**Machine Code** ، وهذا يعني أن تشغيل البرنامج نفسه (وهنا نعني التعليمات التنفيذية الخاصة به المكتوبة بلغة الآلة) لا من الأمثلة الشهيرة .exe ، والملفات التنفيذية بصيغةRAM يتطلب وضعه ضمن عناوين محددة في ذاكرة الوصول العشوائيّ على تعليمات الآلة القابلة للنقل.

كما وجدنا، فإن المترجم نفسه يمثل خطوةً ضمن خطوات المعالجة اللازمة لتحويل النص البرمجيّ من شكله عالي المستوي إلى تعليمات لغة الآلة القابلة للتنفيذ من قبل المعالج، ولكن هذا لا يعني أن المترجم نفسه هو مرحلة مستقلة بحد ذاتها، بل يتكون هو أيضًا من مجموعة من الخطوات، وبشكلٍ عام يُمكن تقسيم عمل المترجم إلى قسمين أساسيين:

- **Analysis Phase / Front-End مرحلة التحليل – النهاية القريبة**
- **Synthesis Phase / Back-End مرحلة الاصطناع – النهاية البعيدة**

المراحل الأساسية لعمل المترجم



كوسيلةٍ للإشارة إلى طبيعة الكود "**Back-End النهاية البعيدة**" و "**Front-End النهاية القريبة**" يتم استخدام مصطلحات الذي يتم التعامل معه، ففي مرحلة التحليل (أي النهاية القريبة) يتعرض النص البرمجيّ من اللغة عالية المستوى لمجموعةٍ من الخطوات التي تعالجه وفقًا لشكله الأصليّ، أي أنه لا يزال "قريبًا" من الشكل الذي يفهمه المبرمج، في حين أن مرحلة الاصطناع (أي النهاية البعيدة) تتعامل مع النص البرمجيّ بعد تحويله لشكلٍ أكثر قربًا من الآلة وأكثر "بعدًا" عن الشكل الذي يفهمه المبرمج.

Analysis Phase المترجم : مرحلة التحليل

يقوم المترجم ضمن هذه المرحلة بإجراء مسح للشيفرة المصدرية والتحقق من صحته القواعدية والدلالية وذلك بعد تقسيمه لأجزاءٍ مختلفة. في حال نجحت مرحلة التحليل ولم يتم اكتشاف أي خطأ قواعدي ضمن الشيفرة المصدرية سيتم توليد **تمثيل وسيط للشيفرة المصدرية** Intermediate Code Representation والذي يمثل مرحلةً بين اللغات عالية المستوى ولغة الآلة. بشيءٍ من التفصيل، فإن مرحلة التحليل تتضمن الخطوات التالية:

Lexical Analysis: تحليل المفردات يقوم المترجم ضمن هذه المرحلة بمسح النص البرمجيّ عبر التعرف على الرموز Tokens (أخذ التعليقات بعين الاعتبار. الهدف من مسح النص البرمجيّ هو تحديد العلامات(الرموز المتواجدة ضمن النص بعين الاعتبار، وكذلك مع عدم White-spaces والأحرف التي يتضمنها واحدًا تلو الآخر، مع عدم أخذ المسافات الفارغة تعتبر Identifiers ، رموز العمليات، مُحددات المتحولاتKeywords نفسه، وكمثال، فإن الكلمات المفتاحية المحجوزة جميعها رموز صالحة يتم التعرف عليها ضمن مرحلة تحليل المفردات. لو أردنا أخذ مثال بسيط على كيفية تحليل الرموز مع تخزين قيمة 10 ضمنه Int من نوع myNumber وفهمها من قبل المترجم، لنأخذ التصريح البسيط التالي لمتحولٍ اسمه C: وذلك بلغة

.