تفاوت کامپایلر (compiler) و مفسر (interpreter)

ما، معمولا ترجیح میدیم که برنامه های کامپیوتری را با یک زبان که برایمان قابل درک باشد ، بنویسیم (مثلا یک زبان نزدیک به محاوره انگلیسی و دستورالعمل های ریاضی)، ولی کامپیوتر وCPU فقط زبان مختص به خود یعنی زبان ماشین (machine code) را میفهمند ، پس باید به نحوی ، دستورالعمل هایی را که نوشته ایم (source code) ، به زبان ماشین تبدیل کنیم ، برای این منظور، ما باید از کامپایلر (compiler) یا مفسر(interpreter) استفاده کنیم...

1. یک مفسر یک دستورالعمل (instruction) یا یک خط (line) را در واحد زمان میخواند ، وبعد از تبدیل کردن آن به کد ماشین (machine code) ، آن را اجرا میکند . در هنگامی که دارد این کد توسط ماشین اجرا میشود ، مفسر خط بعدی را خوانده و آن را تبدیل به کد ماشین میکند. مزیت این روش این است که ، ساده است و ما میتوانیم در زمانی که یک مفسر در حال اجرا میباشد ، کدهای مورد نظر خود را تغییر دهیم یا مفسر را متوقف کنیم و یا آن را دوباره اجرا کنیم. اشکال این روش آن است که همه خطوط ، در هر بار اجرا شدن برنامه ، دوباره از نو ترجمه میشوند . با توجه به این مسئله ، مفسرها کند هستند. بعنوان نمونه ، زبان Basic درکامپیوترهای اولیه از نوع مفسری بود، یا بعنوان نمونه از یک زبان اسکریپتی، میتوانJava Script را نام برد و زبان های برنامه نویسی Lisp و Forth هم از نوع مفسری بودند.

2. یک کامپایلر (compiler) ، همه ی کد برنامه (source code) را بصورت یک جا خوانده و آن را به کد ماشین تبدیل میکند ، و سپس این کد تولید شده را درون یک فایل جدید ذخیره میکند ، این فایل اجرایی جدید، کاملا مستقل از فایل اصلی برنامه (source code) میباشد. بزرگترین مزیت این روش این است که ، این ترجمه ی کد ، فقط یکبار صورت میگیرد و یک پروسس (process) مجزا و مستقل ایجاد میکند. از آنجا که ، کل برنامه ما بصورت یکجا به کد ماشین تبدیل شده ، اجرای آن خیلی سریع صورت میگیرد. عیب این روش این است که ، ما قادر به تغییر کد برنامه در زمان اجرا نیستیم و برای این منظور باید دوباره به کد اصلی (source code) رجوع کنیم و پس از اعمال تغییرات مورد نظرمان ، برنامه را دوباره کامپایل کنیم.(ولی این

روش مورد علاقه ی برنامه نویسان حرفه ای و شرکت های تجاری نرم افزار میباشد ، زیرا از کپی شدن کد اصلی (source code) برنامه های آنها جلو گیری میکند). بعنوان نمونه ، زبان های Visual Basic و C و C++ و Fortran و Cobol و Pascal و. Net Language VB.Net) و C#) و ... همگی از نوع کامپایلری میباشند.

3. بعضی از اوقات ممکن است که به نوع سومی از مترجم ها برخورد کنید که به اسمبلر (assembler)معروف هستند. اسمبلرها مانند کامپایلرها عمل میکنند اما در سطح پایین تر. کد اصلی (source code)، در هر خط ، بصورت مستقیم به کد ماشین تبدیل میشود. اسمبلر ها معمولا توسط افرادی مورد استفاده قرار میگیرند که ، نیاز دارند یک برنامه با حجم کم و سرعت بالا و بصورت کد خالص ماشین ، تولید کنند.

کامپایلر (compiler):

یک کامپایلر ، برنامه ای هست که ، وظیفه ترجمه ی برنامه ی نوشته شده توسط یک زبان برنامه نویسی (مانند C) ، به زبان ماشین (machine code) را ، به عهده دارد. (Translated source code into machine code).

کامپایلرها معمولا ، با زبان های سطح بالا یا GL3 (زبان های نسل سوم - rd-generation 3 languages) کارمیکنند. (مانند زبان C).

مفسر (interpreter):

یک مفسر، در واحد زمان ، فقط یک خط از کد اصلی را به کد ماشین ترجمه میکند و سپس آن را

اجرا میکند ، و در زمانی که یک خط در حال اجرا میباشد خط بعد در حال ترجمه است . یک مثال خارجی برای یک مفسر (interpreter) این است که ، یک کتاب انگلیسی را خط به خط به فارسی ترجمه کنیم . یک مفسر هرگز یه برنامه را به یک واحد مستقل و مجزا (مانند یک فایل اجرایی) ترجمه نمیکند.

از آنجا که یک مفسر بصورت خط به خط وظیفه ترجمه را انجام میدهد ، برای یک برنامه نویسی مبتدی ، مراحل ترجمه یک برنامه قابل فهم تر میشود. و با توجه به این موضوع ، اگر خطایی در کد برنامه وجود داشته باشد ، میتوان به راحتی آن را پیدا کرده و رفع نمود.

برنامه های نوشته شده توسط یک زبان مفسری ، بدون وجود مفسر خود ، قابل اجرا نیستند.
برنامه های نوشته شده توسط یک زبان کامپایلری ، با سرعت بالایی اجرا میگردند ولی برنامه های مفسری کند تر اجرا میگردند.
دسترسی به کد اصلی برنامه های کامپایلری ، با توجه به اینکه در دسترس نیستند، امکان پذیر نمیباشد یا حداقل خیلی سخت میشود.

مرتضی امیـــــرپناه
برگرفته ازسایتهای اینترنتی