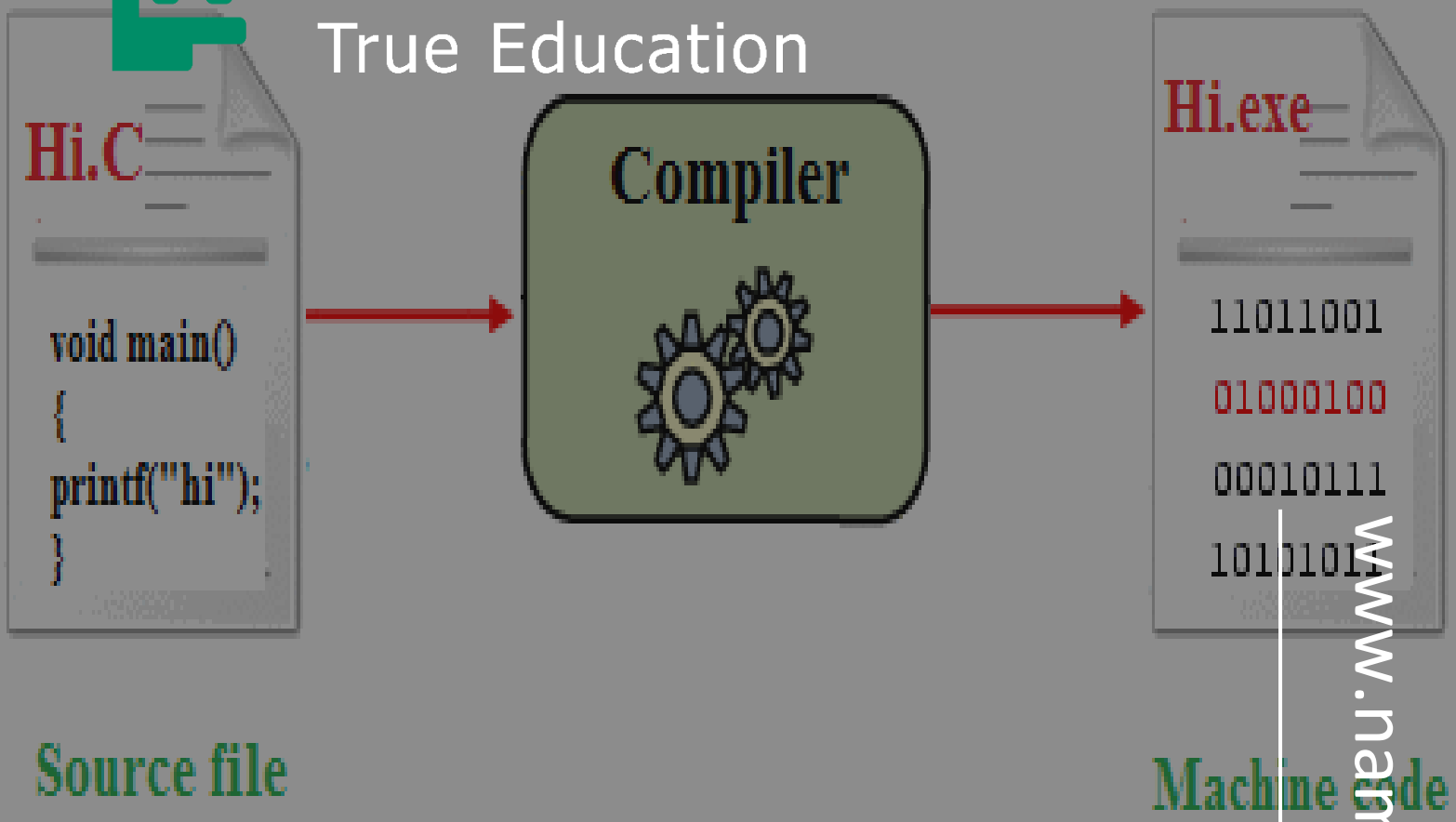
Namatek
True Education



Compiler

www.namatek.com

کامپایلر

www.namatek.com

فهرست مطالب

به عنوان یک برنامه نویس و یا علاقه مند به این حوزه شما هم حتما تا به حال با نرم افزارهای کامپایلر رو به رو شده اید. اما سوال این جاست که نقش این نرم افزارها در برنامه نویسی چیست و چرا باید از آن ها استفاده کرد. در این مقاله سعی داریم به زبان ساده طرز کار این نرم افزارها رو توضیح دهیم. با ما همراه باشید.

کامپایلر چیست؟

همان گونه که برای دوبله کردن یک فیلم خارجی، ابتدا باید کل فیلم به زبان فارسی ترجمه شود و سپس در اختیار بینندگان قرار بگیرد، کامپایلر نیز به همین ترتیب عمل می کند. به گونه ای که کامپایلر همانند یک مترجم حرفه ای زبان برنامه نویسی مبدأ را به زبان ماشین تبدیل کرده و سپس این برنامه را روی کامپیوتر اجرا می کند.

در این مقاله قصد داریم تا برایتان از مفهوم کامپایلر که یکی از مفاهیم اصلی در علوم کامپیوتر و فنی مهندسی می باشد، به صورت کامل و منطبق بر منابع علمی به روز بگوییم.

مبحث کامپایلر یکی از دروس اصلی در کنکور کارشناسی ارشد نرم افزار نیز می باشد، به همین دلیل این مقاله آموزشی در برطرف ساختن بخشی از دغدغه های داوطلبان کارشناسی ارشد نیز ممکن است مفید باشد.

تعریف کامپایلر (Compiler)

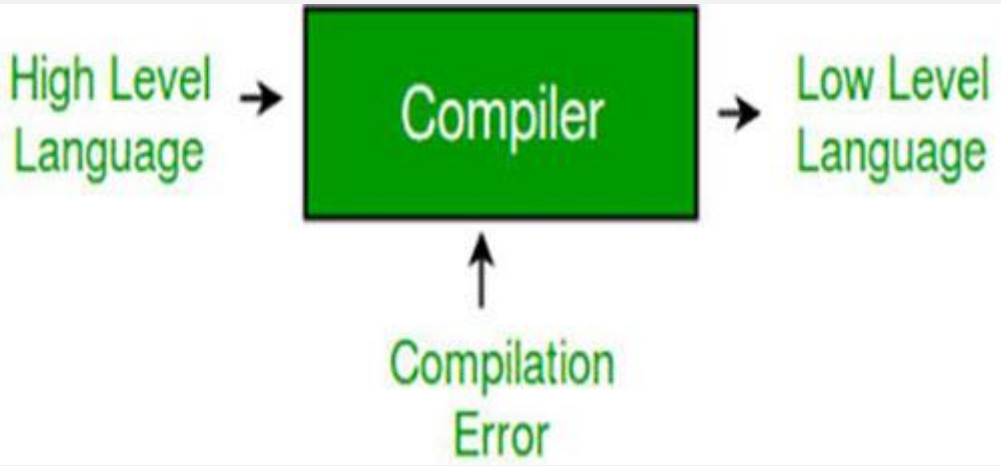
کامپایلر در واقع یک نوع نرم افزار یا یک برنامه تخصصی است که عبارت هایی را که به یک زبان برنامه نویسی خاص نوشته شده، پردازش می کند و آن ها را به زبان ماشین یا "کد" تبدیل می کند؛ به طوری که پردازنده کامپیوتر بتواند از آن ها استفاده کند. به طور معمول، یک برنامه نویس با استفاده از یک ویرایشگر، جملات زبان را به زبانی مانند Pascal یا C در یک خط می نویسد. پرونده ای از تمام آن چه **گفته های منبع** نامیده می شود ایجاد می گردد. برنامه نویس کامپایلر، زبان مناسب را اجرا می کند، سپس نام پرونده ای را که حاوی گفته های منبع باشد، مشخص می کند که این نام همان کد منبع برنامه می باشد.

وظیفه کامپایلر این است که کد منبع را به کد شی ترجمه کند. بنابراین، یک کامپایلر با یک مفسر فرق می کند، منبع را پشت سر هم آنالیز و اجرا می کند، بدون این که به کل برنامه نگاه کند. با این حال، برنامه هایی که توسط کامپایلرها تولید می شوند بسیار سریع تر از همان برنامه هایی که توسط یک مفسر اجرا شده است.

کامپایلر با چه زبان هایی کار می کند؟

همان طور که می دانید چندین زبان مبدا تخصصی برنامه نویسی مانند Fortran، C++، C، Pascal و... وجود دارند و آن چه تا این جا از مفهوم کامپایلر برداشت کرده ایم این است که وظیفه اصلی آن تبدیل کد نوشته شده با یک زبان سطح بالا (مثلاً C++) به مجموعه ای از دستورالعمل های زبان ماشین است؛ به طوری که برای CPU یک کامپیوتر دیجیتال قابل درک باشد.

حالا یک سوال اساسی مطرح است که آیا کامپایلر باید تمام زبان ها را بداند و با چه زبان هایی سر و کار دارد؟ با وجود پیچیدگی ظاهری، کارهای اساسی که هر کامپایلر باید انجام دهد در اصل یکسان هستند. در بالاترین سطح، یک کامپایلر دارای یک قسمت ورودی و یک قسمت خروجی است. مطلوب است که قسمت ابتدایی با جنبه های زبان ورودی سروکار داشته باشد، اما تا حد امکان آن را از دستگاه مستقل می کند. برخی از کامپایلرها زبان سطح بالا را به یک زبان مونتاژ واسطه ای ترجمه می کنند که توسط برنامه مونتاژ یا اسمبلر به کد دستگاه ترجمه می شود. سایر کامپایلرها مستقیماً زبان دستگاه تولید می کنند.



نحوه کار کامپایلر

از نظر مفهومی، یک کامپایلر در فازهای متفاوتی کار می کند که با تغییر هر یک از آن ها برنامه منبع از یک مرحله به مرحله دیگر فرستاده می شود.

می توان گفت که مراحل به دو بخش طبقه بندی می شوند:

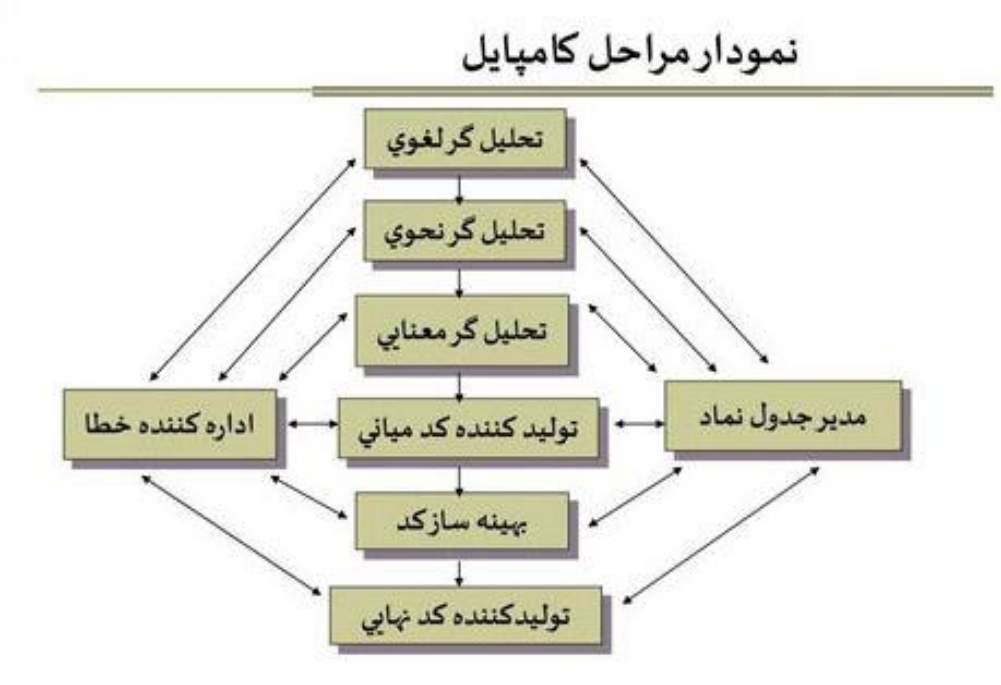
تجزیه و تحلیل

برنامه مبدل را به قطعات تشکیل دهنده می شکند و آن را ایجاد می کند و با گرفتن نمایندگی واسطه از برنامه منبع مراحل زیر را پیاده سازی می کند.

- تحلیل واژگانی
- تجزیه و تحلیل نحوی
- تحلیل معنایی

سنتز (ترکیب)

برنامه هدف مورد نظر را از نمایندگی واسطه ایجاد می کند و در قسمت برگشت با مراحل تولید کد آن را بهینه سازی می کند. به زبان ساده تر در این مرحله تبدیل کد میانی به برنامه مقصد در زبان دیگر با بیشترین روش های خاص انجام می شود.



تفاوت مفسر و کامپایلر

هر کامپایلر نسبت به مفسر چهار تفاوت اساسی دارد که به شرح زیر است:

1. نحوه اجرا شدن
2. عدم وابستگی به سیستم عامل
3. سرعت و میزان استفاده از حافظه و پردازنده
4. یافتن خطا

انواع کامپایلر

کامپایلرها را می توان بر اساس نوع زبان و داده ورودی یا خروجی، ساختار داخلی و چگونگی رفتار در حین اجرای برنامه تقسیم بندی کرد. معمولا کامپایلرها را به دو دسته Native و Cross تقسیم می کنند.

Native

به این نوع از کامپایلرها که کدهای باینری برای اجرای برنامه ها تولید می کنند، کامپایلرهایی با کد محلی یا Native گفته می شود. به این دلیل به این نام خوانده می شوند چون فقط در کامپیوترهای یک نوع با سیستم عامل های یکسان به کار گرفته می شوند.

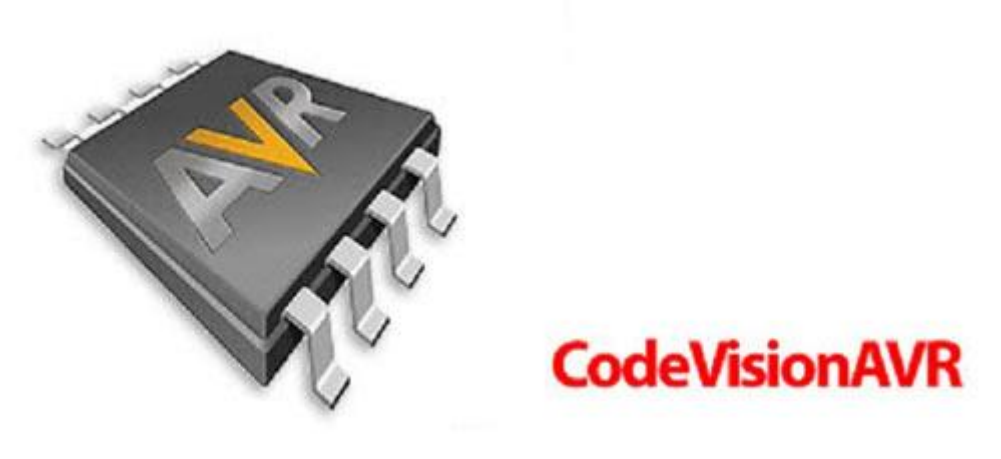
Cross

گاهی نیز نیاز است، کامپایلرها کدهای باینری را تولید کنند که بتوان از آن ها در سیستم های مختلف استفاده کرد. به این نوع از کامپایلرها که به سخت افزار خاصی وابستگی ندارند، کامپایلرهای عبوری یا Cross گفته می شود. در این نوع از کامپایلرها فقط باید یک بار سخت افزار را به آن معرفی نمود، پس می توان به این نتیجه رسید که کامپایلرهای عبوری مفیدتر می باشند.

در ادامه قصد داریم شما را با دو برنامه کامپایلر معروف و پر کاربرد که در رشته های فنی مهندسی از جمله تمام گرایش های رشته برق بسیار کاربرد دارد آشنا سازیم.

کدویژن Code vision

یکی از نرم افزارهای مهم و تخصصی برای رشته های برق و کامپیوتر (گرایش سخت افزار) Code Vision AVR می باشد. در حقیقت این نرم افزار یک کامپایلر برای زبان برنامه نویسی C می باشد که از آن برای برنامه نویسی میکروکنترلر های AVR استفاده می شود. این برنامه محیط مناسب برای برنامه نویسی و کامپایل کردن برنامه های نوشته شده میکروکنترلر را برای شما فراهم می کند.



بسکام Bascom-AVR

کامپایلری است که برای برنامه نویسی و طراحی مدارات الکترونیکی بر مبنای میکروکنترلرها با بهره گیری از زبان برنامه نویسی Basic نوشته شده است ایجاد شده است. به وسیله بسکام می توان با استفاده از آی سی های

خانواده AVR و MCS-8051 مدارات پیشرفته ای را طراحی و راه اندازی کرد.