فصل ۱

مقدمه ای بر کامپایلرها

۱.۱ انواع زبانهای برنامه نویسی

۱.۱.۱ زبان ماشین

زبانی که در آن داده ها و دستورالعمل ها به صورت کدهای باینری (صفر و یک) نمایش داده می شوند و تنها زبانی است که کامپیوتر درک می کند.

۲.۱.۱ زبان اسمبلی

زبان اسمبلی به جای کد های باینری از کلمات اختصاری استفاده می کنند. برای تبدیل برنامه اسمبلی به زبان ماشین از نرم افزار مترجمی به نام اسمبلر استفاده می شود.

۳.۱.۱ زبان های سطح بالا

زبان های سطح بالا، به زبان محاوره ای نزدیکترند و دارای ساختارها و دستورات بیشتر و قدرتمندتر نسبت به زبان اسمبلی هستند. برنامه به زبان های سطح بالا توسط کامپایلرها به زبان ماشین ترجمه می شوند تا قابل اجرا بر روی کامپیوترها شوند

۲.۱ تفاوت اسمبلر و کامپایلر

اسمبلر هر دستور اسمبلی را فقط به یک دستور زبان ماشین ترجمه می کند در حالیکه کامپایلرها، هر دستور سطح بالا را ممکن است به چندین دستور زبان ماشین ترجمه کنند.

۳.۱ روش های ترجمه و اجرای برنامه های سطح بالا

- استفاده از مفسر
- استفاده از کامپایلر

۱.۳.۱ استفاده از مفسر

در این روش دستورالعملهای برنامه یک به یک توسط نرم افزاری به نام مفسر خوانده شده و اجرا می گردد .

مزایای استفاده از مفسر

- سهولت اشکال زدایی
- قابلیت انعطاف بالا
- پیاده سازی آسان
- قابلیت حمل بالا

معایب استفاده از مفسر

- تکرار تفسیر
- سرعت اجرای پایین : زیرا برنامه مستقیماً توسط سخت افزار اجرا نمی شود
- نیاز به مفسر : هرجا که برنامه باید اجرا شود، مفسر باید وجود داشته باشد
- دسترسی به کد منبع

۴.۱ استفاده از کامپایلر

۱.۴.۱ کامپایلر چیست؟

کامپایلر نرم افزاری است که برنامه نوشته شده به زبان مبدا را به برنامه معادلی در زبان مقصد ترجمه می نماید.

۲.۴.۱ مزایای استفاده از کامپایلر

- سرعت اجرای بالا
- اجرای مستقل برنامه از کامپیوتر
- حفاظت از کد منبع برنامه
- عدم تکرار کامپایلر

۳.۴.۱ معایب استفاده از کامپایلر

- زمانبر بودن اشکالزدایی
- قابلیت حمل پایین
- سهولت پیاده سازی

۵.۱ جلوبندی و عقب بندی کامپایلر

اگر n تعداد زبانهای برنامه سازی مانند

- C
- C++
- Java
- Python
- . . .

و k تعداد انواع مختلف کامپیوترها باشد، در این صورت به $n * k$ کامپایلر نیاز است .
ایجاد این تعداد کامپایلر بسیار زمانبر و پرهزینه است، برای حا این مشکل از تقسیم کامپایلر به جلوبندی و عقب بندی استفاده می کنیم . برای این کار یک زبان میانی در نظر میگیریم، ابتدا برنامه مبدا را به زبان میانی ترجمه میکنیم و سپس از زبان میانی به زبان مقصد ترجمه می کنیم .
بخشی از کامپایلر که وظیفه ترجمه برنامه ی مبدا به زبان میانی را بر عهده دارد، جلوبندی و بخشی از کامپایلر که وظیفه ترجمه برنامه به زبان میانی را به زبان مقصد بر عهده دارد را عقب بندی کامپایلر می نامیم .
با استفاده از این روش برای n زبان مبدا و k کامپیوتر مختلف به n جلوبندی و k عقب بندی نیاز است که در مجموع n+k برنامه احتیاج است .
جلویندی کامپایلر به زبان مبدا و عقب بندی کامپایلر به زبان مقصد وابسته است .

۶.۱ تحلیلگر لغوی

تحلیلگر لغوی بخشی از کامپایلر است که مستقیماً به برنامه مبدا دسترسی دارد، تحلیلگر لغوی برنامه مبدا را به صورت جریانی از کاراکترها دریافت کرده، کلمات تشکیل دهنده ی برنامه و نوع آنها را تشخیص داده و برای تحلیلگر نحوی ارسال می کند .
کشف خطاهای مربوط به ساختار تک تک لغات نیز وظیفه ی تحلیلگر لغوی است .

۷.۱ تحلیلگر نحوی

وظیفه ی تحلیلگر نحوی بررسی صحت و درستی ترتیب لغات برنامه مبدا است .

۸.۱ تحلیلگر معنایی

تحلیلگر معنایی معنی دار بودن عباراتی که از نظر نحوی درست بوده اند را مورد بررسی قرار می دهد .
برخی از مواردی که توسط تحلیلگر معنایی انجام می شود عبارتند از :

- بررسی هماهنگی پارامترهای تابع با پارامترهای فراخوانی

- بررسی و کنترل نوع داده
- تبدیل نوع ضمنی
- بررسی تعریف دوباره متغیر : تحلیلگر معنایی بررسی می کند تا هیچ متغیری دو بار تعریف نشده باشد .

۹.۱ تولید کننده کد میانی

پس از بررسی و تبدیلات انجام شده در مراحل تحلیل لغوی، نحوی، معنایی و اطمینان از صحت برنامه مبدا، تولید کد میانی آغاز می گردد .

۱۰.۱ بهینه کننده کد میانی

بهینه کننده کد میانی وظیفه ی بررسی کد میانی را بر عهده دارد تا در صورت امکان آن را بهینه سازی کند . بهینه سازی یعنی اعمال تغییراتی در برنامه که بدون تغییر در عملکرد برنامه، مصرف حافظه را کاهش یا سرعت اجرای برنامه را افزایش دهد .

۱۱.۱ تولیدکننده کد

بخش تولیدکننده کد، کد میانی بهینه سازی شده را به زبان اسمبلی ترجمه می کند .

۱۲.۱ مدل تجزیه و ترکیب

فرآیند کامپایل شامل دو مرحله است : در مرحله ی اول برنامه به اجزای تشکیل دهنده آن تجزیه می شود . در طی این مرحله خطاهای برنامه مبدا مشخص می گردد . فازهای تحلیل لغوی، تحلیل نحوی، تحلیل معنایی و تولید کد میانی عملیات تجزیه را انجام می دهند، این فازها، جلوبندی کامپایلر را تشکیل می دهند . پس از اطمینان از صحت برنامه مبدا برنامه تجزیه شده برای ایجاد برنامه مقصد ترکیب می گردد .