

اصول و مبانی برنامه‌نویسی



مجید شبیری

کارشناسی ارشد آ، گرایش شبکه
از دانشگاه صنعتی امیرکبیر



فرآیند توسعه نرم افزار

(۹) آشنایی با اسمبلر، مفسر و کامپایلر

(۱۰) محیط‌های برنامه‌نویسی

(۱) نرم افزار چیست؟

(۲) گام‌های توسعه نرم افزار

(۳) برنامه‌نویسی چیست؟

(۴) برنامه‌نویس کیست؟

(۵) زبان برنامه‌نویسی چیست؟

(۶) دسته‌بندی زبان‌های برنامه‌نویسی

(۷) دستور چاپ "Hello, World"

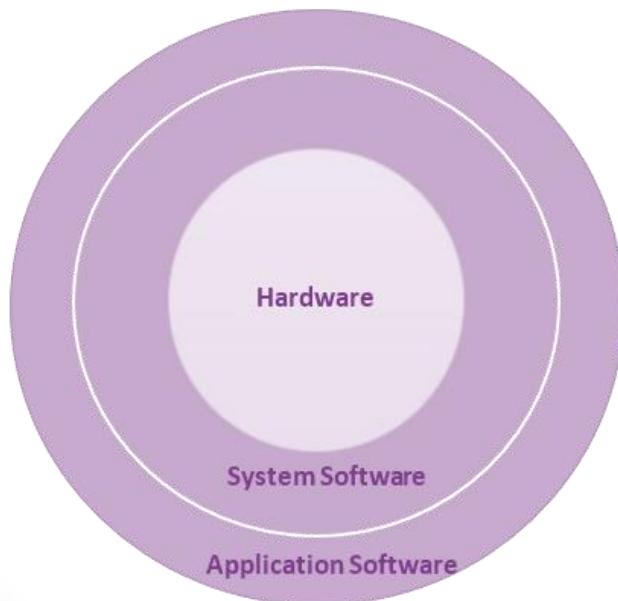
(۸) فرآیند تبدیل برنامه به زبان ماشین

- نرم افزار (software) از کنار هم قرار گرفتن چندین برنامه (program) تشکیل می شود.
- برنامه یا رویه (procedure) به مجموعه ای از دستورالعمل ها گفته می شود که منجر به یک نتیجه مشخصی می شوند.
- دستورالعمل (instruction) مشخص کننده یک task برای یک وسیله سخت افزاری در کامپیوتر می باشد.

■ انواع نرم افزار

○ نرم افزار کاربردی

○ نرم افزار سیستمی



مانند: فتوشاپ - میکروسافت آفیس - آنتی ویروس - مرورگر وب - نرم افزارهای مالی - اتوماسیون اداری و ...

■ نرم افزارهای کاربردی روی سیستم عامل اجرا می شوند.

■ نرم افزارهای کاربردی به کاربر کمک می کنند تا استفاده بیشتر و متنوع از کامپیوتر داشته باشند.

■ نرم افزارهای کاربردی به کاربر کمک می کنند تا راحت تر کارهای مورد نظرشان را انجام دهند.

مانند: سیستم عامل - درایور - اسمبلر - مفسر - کامپایلر

■ ماشین (سخت افزار) **زبان باینری** (۰ و ۱) را می شناسد ولی انسان از طریق **برنامه نویسی سطح بالا** (غیر از زبان ماشین) و

برنامه های کاربردی با سخت افزار کامپیوتر ارتباط برقرار می کند.

■ از آنجایی که زبان **ماشین و انسان** متفاوت است، بنابراین برای اینکه منظور همدیگر را متوجه شوند نیاز به **واسط** دارند.

■ نرم افزارهای سیستمی به عنوان **واسط** بین **سخت افزار و برنامه های کاربردی** قرار می گیرند و وظیفه تبدیل

دستورالعمل های کاربر به زبان ماشین را دارند.

- **تعریف مسئله** (تحلیل نیازهای کاربر، چالش‌های مسئله، امکانات مورد نیاز از لحاظ فنی و ...)
- **ارائه راه‌حل برای مسئله** (ارائه مناسب‌ترین راه حل از بین گزینه‌های موجود)
- **توسعه الگوریتم** (ارائه تعریف دقیق از راهکار و تبیین راه حل ارائه شده به صورت گام به گام و سیستماتیک)
- **تایید الگوریتم** (تست دقت و رفع اشکالات احتمالی جهت صرفه‌جویی زمان، هزینه و انرژی در آینده)
- **کدنویسی** (وقتی که ارائه راه‌حل برای مسئله و طراحی الگوریتم به اتمام رسید، کدنویسی واقعی برای راه حل ارائه شده در یک زبان برنامه‌نویسی انجام می‌شود)
- **تست** (برنامه‌نویس حین کدنویسی، کدهای نوشته شده را تست می‌کند تا خطاهای احتمالی را با هزینه کمتری یافته و برطرف نماید)
- **مستندسازی** (وقتی کدنویسی به اتمام رسید، برنامه‌نویس باید تمام مراحل و ویژگی‌های برنامه را مستند کند تا هر کاربری به راحتی بتواند با برنامه کار کند)
- **نصب و استقرار** (وقتی تمام مراحل بالا به درستی انجام شد، کد توسعه یافته به همراه راهنمای کاربری روی کامپیوتر نصب می‌شود)

- **برنامه‌نویسی (Programming)** یا **کدنویسی** یکی از مراحل تولید نرم افزار می باشد.
- برنامه‌نویسی یعنی نوشتن دنباله‌ای از دستورات، جهت انجام یک کار مشخص توسط کامپیوتر.
- برای انجام برنامه‌نویسی باید یکی از زبان‌های برنامه‌نویسی را انتخاب کنیم و نحوه کدنویسی در آن زبان را یاد بگیریم.

○ **برنامه‌نویس (programmer) یا توسعه دهنده (Developer)** کسی است که از طریق وارد کردن دستورات

برنامه‌نویسی، برای انجام کارها توسط کامپیوتر، نرم افزار خاصی را تولید می‌کند.

○ برنامه‌نویس سعی دارد با توسعه یک نرم افزار، وقایع دنیای بیرون را در کامپیوتر شبیه‌سازی نماید به طوری که کارهای مورد

نیاز انسان توسط کامپیوتر انجام شود.

- زبان برنامه نویسی (Programming Language) یک سینتکس (Syntax) یا قانون گذاری مشخص، برای انتقال درخواستها به کامپیوتر می باشد.
- همانطور که زبان های طبیعی (فارسی، انگلیسی و ...) برای ارتباط بین انسان ها طراحی شده اند، زبان برنامه نویسی نیز برای ارتباط بین انسان و کامپیوتر طراحی شده است.
- از آنجایی که یک زبان برنامه نویسی برای انجام تمام خواسته های انسان مناسب نیست، زبان های برنامه نویسی مختلفی برای اهداف و کاربردهای مختلف طراحی شده است.

زبان ماشین

■ زبانی که برای سخت افزار طراحی شده و توسط CPU مستقیماً قابل درک می باشد ولی برای انسان قابل درک نیست.

زبان اسمبلی

■ این زبان یک کدگذاری از زبان ماشین است که درک آن را برای انسان ساده تر کرده و خوانایی بیشتری دارد.

زبان های سطح بالا

■ این زبان ها از اسمبلی ساده تر و قابل درک تر بوده و حتی بعضاً شبیه زبان انگلیسی هستند.

■ Assembly

section .data

Msg: db "hello world", 10 ; message to print

MsgLen: equ \$ - Msg ; length of message

section .text

global start

start:

push dword MsgLen ; length of message to print

push dword Msg ; message to print

push dword 1 ; FD of 1 for standard output

sub esp, 4 ; OS/X requires extra 4 bytes after arguments

mov eax, 4 ; 4 - write() system call

int 80H ; perform system call

add esp, 16 ; restore stack (16 bytes pushed: 3 * dword + 4)

push dword 0 ; exit value of 0 returned to the OS

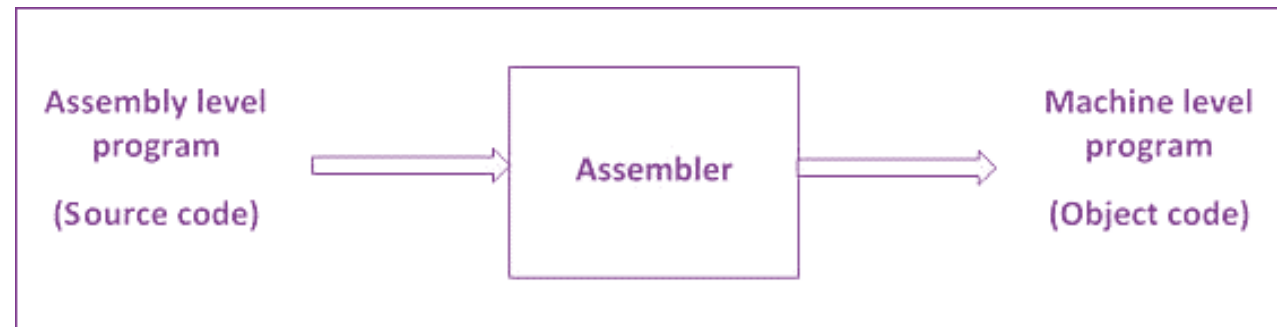
sub esp, 4 ; OS/X requires extra 4 bytes after arguments

mov eax, 1 ; 1 - sys_exit() system call

int 80H ; perform system call

- کدنویسی به زبان اسمبلی، تعداد خطاها و مدت زمان خطایابی (دیباگ) را کاهش می دهد.
- در زبان اسمبلی برنامه نویس می تواند مصرف حافظه را بهینه ساخته و بهره وری سخت افزار را بالاتر ببرد.
- برنامه نویس دسترسی مستقیم به سخت افزار دارد و در نوشتن برنامه برای نوع خاصی از کامپیوترها، انعطاف بیشتری دارد.

◉ اسمبلر یک نرم افزار سیستمی است که وظیفه آن تبدیل **کدهای اسمبلی** به **کد ماشین** است.



- هر چند کدنویسی به زبان اسمبلی، بهره‌وری حافظه و سخت‌افزار را بالا می‌برد، ولی با پیشرفت تکنولوژی، از محدودیت‌های سخت‌افزاری کاسته شده و امروزه برنامه‌نویس دیگر به اندازه گذشته نگران موضوعات بهینه‌سازی نیست.
- معیار دنیای امروزه، سادگی هر چه بیشتر کد نوشتن برای برنامه‌نویس است.
- همچنین برای اینکه اقشار مختلف جامعه، بدون نیاز به دانش فنی بالا و آشنایی کامل با سخت‌افزار، بتوانند برنامه‌نویسی کنند، نیاز به زبان‌های ساده‌تری بود که شبیه زبان طبیعی (محاوره ای) انسان باشد.

- General purpose : C , C++ , C# , Java
- Scriptings : Perl , Php , Python
- Web : JavaScript
- Frameworks and libraries : JQuery , React , Angular

- C `printf ("Hello, World");`
- Python `print "Hello, World"`
- Java `System.out.println("Hello, World");`
- JavaScript `alert("Hello, World")`

- همانطور که کدهای نوشته شده به **زبان اسمبلی** می بایست توسط **اسمبلر** به **کد ماشین** تبدیل شوند، کدهای نوشته شده به **زبانهای سطح بالا** نیز باید توسط یک نرم افزار سیستمی به **کد ماشین** تبدیل شوند.
- سورس کد برنامه کامپیوتری در قالب یک فایل به نام فایل برنامه (program file) روی کامپیوتر ذخیره می شود.
- برای اینکه کامپیوتر بتواند برنامه نوشته شده را اجرا کند، برنامه باید به **کد ماشین** تبدیل شود و سپس فایل با **فرمت باینری** قابل اجرا تولید شود.
- برای انجام این کار، دو فرآیند وجود دارد:

■ **تفسیر (Interprete)**

■ **کامپایل (Compile)**

❶ برخی از زبان‌های برنامه‌نویسی سطح بالا (مانند پایتون و جاوا اسکریپت) طی فرآیند **تفسیر** قابلیت اجرا پیدا می‌کنند.

❷ به این زبان‌ها، **زبان‌های تفسیری** گفته می‌شود.

❸ زبان‌های تفسیری، توسط یک برنامه سیستمی به نام **مفسر (interpreter)** اجرا می‌شوند.

❹ مفسر، کدهای نوشته شده را به صورت خط به خط خوانده و به کد ماشین تبدیل می‌کند.

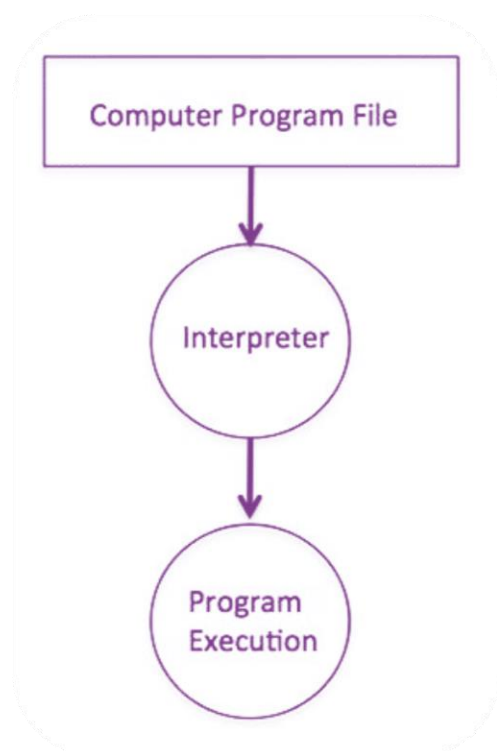
❺ مفسر برای هر خط، کد ماشین تولید شده را به فایل **Object** برنامه اضافه می‌کند.

❻ **مزیت استفاده از زبان‌های تفسیری**

■ نوشتن کد در آنها راحت‌تر بوده

■ نیاز به فضای حافظه بالایی ندارند

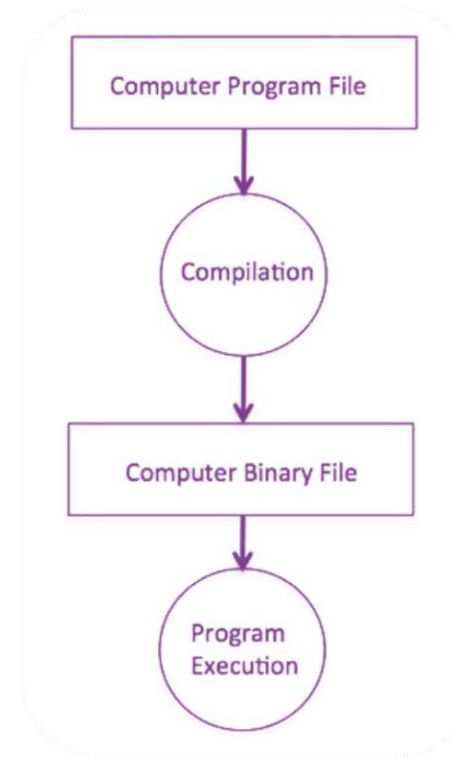
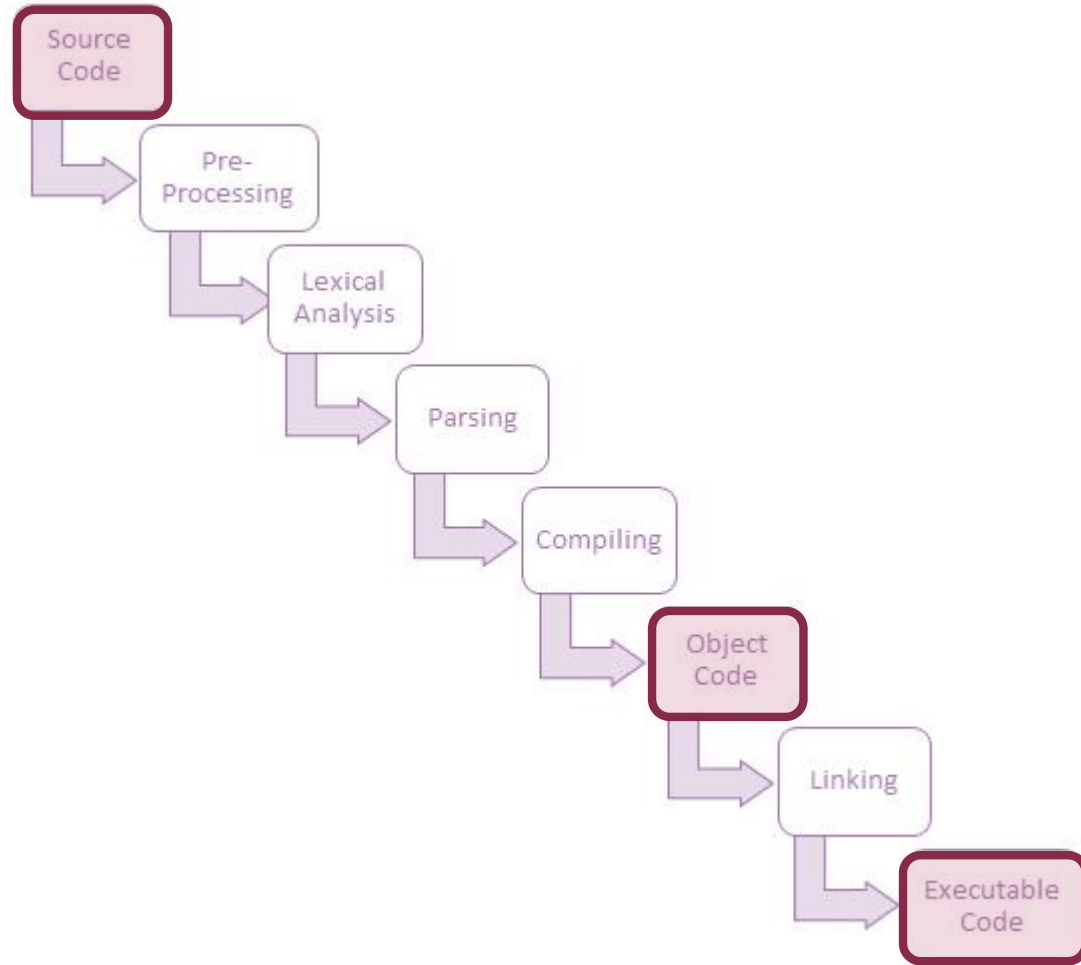
■ برای نوشتن دستورات **bash** جهت انجام تنظیمات و پیکربندی سیستمی مناسب هستند



- ❶ مشکل فرآیند **تفسیر** این بود که **سرعت** اجرای برنامه‌های تفسیر شده بخصوص برای برنامه‌های بزرگ **پایین** بود.
- ❷ برای رفع این مشکل، فرآیند کامپایل معرفی شد.
- ❸ کامپایلر ابزاری است که **سورس کد برنامه** را یک بار به طور کامل **اسکن** کرده و سپس کل برنامه را یکجا به **object code** تبدیل می‌کند.
- ❹ در نهایت آجکت کد به **کد قابل اجرا** توسط کامپیوتر تبدیل می‌شود.

❶ تفاوت‌های کامپایلر با مفسر

- کامپایلرها پیچیدگی بیشتری نسبت به مفسرها دارند
- کامپایلرها به فضای حافظه بیشتری نیاز دارند
- کامپایلرها زمان بیشتری برای کامپایل سورس کد نیاز دارند (برنامه‌نویس برای دیدن نتیجه اجرای یک خط کد، باید منتظر کامپایل کل برنامه بماند)
- برنامه‌های کامپایل شده با سرعت بیشتری روی کامپیوتر اجرا می‌شوند.



پیش پردازش (pre-processing): در این مرحله دستورات پیش پردازشی که اغلب در زبان های C و C++ دیده می شود به زبان اسمبلی تبدیل می شوند.

تحلیل واژگان (lexical analysis): تمام دستورالعمل ها به واحدهای واژگانی مانند متغیرها، علائم محاسباتی و ... تبدیل می شوند.

تجزیه (parsing): همه دستورالعمل ها از لحاظ مطابقت با قواعد گرامری زبان، بررسی می شوند و در صورت وجود خطا کامپایلر اطلاع می دهد.

ترجمه (compiling): در این مرحله، سورس کد به object code تبدیل می شود.

پیوند (Linking): اگر سورس کد از کتابخانه ها یا فایل های خارجی استفاده کرده باشد، آدرس فایل اجرایی آنها به برنامه اضافه می شود. همچنین اگر کد

برای اجرای نهایی، نیاز به بازآرایی داشته باشد، مرتب سازی کد نیز انجام می شود و نهایتاً **فایل اجرایی** تولید می شود.

◉ هر زبان برنامه‌نویسی، یک یا چند محیط برنامه‌نویسی دارد که برنامه‌نویس می‌تواند دستورات خود را در آن محیط نوشته و یک برنامه کامپیوتری بسازد.

◉ انواع محیط برنامه‌نویسی

■ ویرایشگر (Editor): فقط اجازه نوشتن و ویرایش کد را می‌دهند

◉ مانند Notepad

■ محیط توسعه مجتمع (IDE): تمام نیازهای برنامه‌نویسی و توسعه نرم افزار اعم از کدنویسی، اشکال زدایی، اسمبل، کامپایل و ... را پوشش می‌دهد

◉ مانند Visual Studio و VSCode



- ◉ https://www.tutorialspoint.com/basics_of_computer_science/index.htm
- ◉ https://www.tutorialspoint.com/basics_of_computers/index.htm
- ◉ https://www.tutorialspoint.com/computer_fundamentals/index.htm
- ◉ https://www.tutorialspoint.com/computer_programming/index.htm

اصول و مبانی برنامه نویسی



مجید شبیری

کارشناسی ارشد IT، گرایش شبکه
از دانشگاه صنعتی امیرکبیر

