

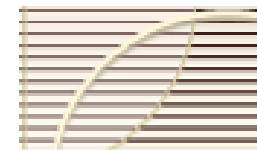
به نام خدا



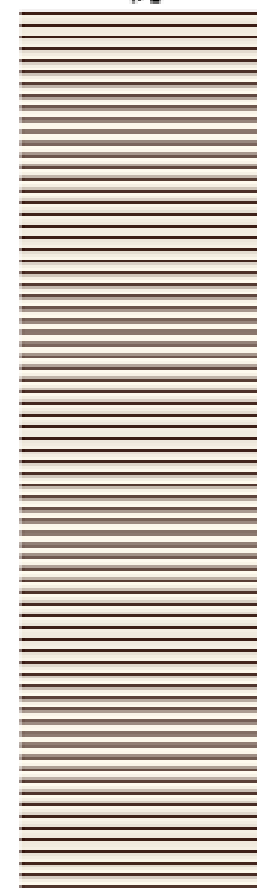
طراحی زبان‌های برنامه‌سازی

مرجان عظیمی
Azimi.marjan@gmail.com

معماری ماشین



۱۳۵۰



عملکرد کامپیوتر



- کامپیوتر مجموعه‌ای از الگوریتم‌ها و ساختمان داده‌ها است که قابلیت ذخیره و اجرای برنامه‌ها را دارد.
- هر کامپیوتر از ۶ جزء تشکیل شده است:
 - ✓ داده‌ها
 - ✓ اعمال اولیه
 - ✓ کنترل ترتیب
 - ✓ دستیابی به داده‌ها
 - ✓ مدیریت حافظه
 - ✓ محیط عملیاتی

حافظه‌ی ذخیره‌سازی داده‌ها



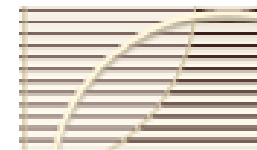
حافظه اصلی: به صورت دنباله‌ای از بیت‌های خطی سازمان‌دهی می‌شود که از کلمات با طول ثابت تشکیل می‌گردد.

ثبات‌های سریع: طول ثبات‌های سریع به اندازه طول کلمات است و طوری تقسیم‌بندی می‌شود که هر قسمت آن قابل دستیابی باشد. محتویات ثبات می‌تواند شامل داده یا آدرس مکانی باشد.

حافظه نهان سریع: معمولاً بین حافظه اصلی و ثبات‌ها قرار می‌گیرد و مکانیزمی برای دسترسی سریع به داده‌های موجود در حافظه است.

فایل‌های خارجی: بر روی دیسک مغناطیسی، نوار مغناطیسی یا CD ذخیره می‌شوند.

اعمال اولیه و کنترل ترتیب

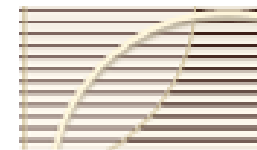


- **اعمال اولیه:** کامپیوتر باید مجموعه‌ای از اعمال اولیه داشته باشد که متناظر با کدهای عملیاتی هستند و به صورت دستورات زبان ماشین می‌باشند.

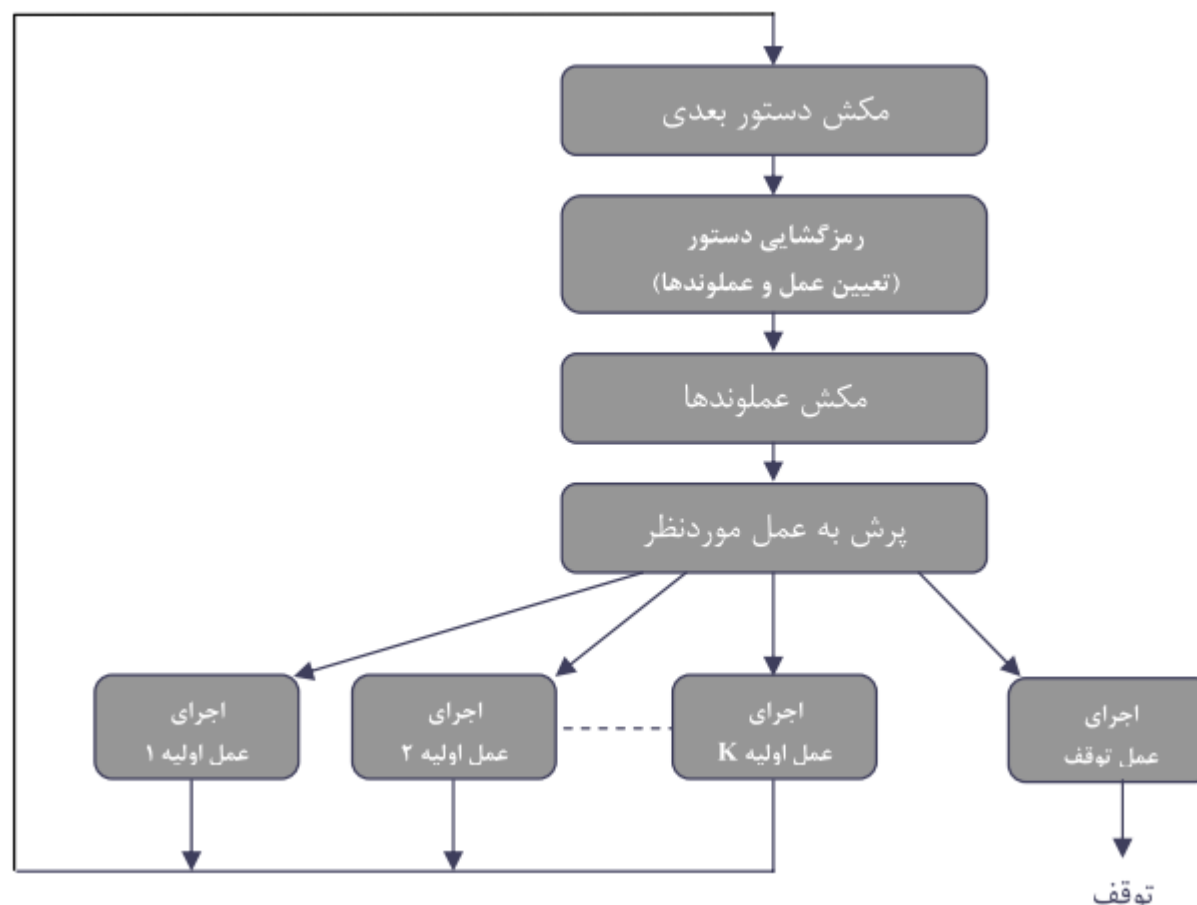
✓ اعمال اولیه برای انجام محاسبات - اعمال اولیه برای تست خواصی از داده های اولیه - اعمال اولیه برای دستکاری قسمتی از عناصر داده‌ها - اعمال اولیه برای کنترل دستگاه‌های جانبی - اعمال اولیه برای کنترل ترتیب اجرا

- **کنترل ترتیب:** در حین اجرای برنامه دستور بعدی که باید اجرا شود توسط محتویات ثبات آدرس برنامه مشخص می‌گردد. این ثبات حاوی آدرس دستور بعدی است.

سلسله مراحل کنترل ترتیب



۱۳۵۰



دسترسی به داده و مدیریت حافظه



- **دسترسی به داده‌ها:** علاوه بر کد عملیاتی هر دستور ماشین باید عملوندهایی را مشخص کند که آن عمل از آن استفاده می کند. عملوند ممکن است در حافظه اصلی یا در ثبات باشد. مکانیزمی که برای تعیین عملوند و بازیابی آن و ذخیره نتایج انجام می گیرد کنترل دستیابی به داده‌ها گویند.

- **مدیریت حافظه:** تمام منابع کامپیوتر (مثل حافظه، پردازنده مرکزی، دستگاه‌های حافظه خارجی) تا آنجایی که ممکن است فعال باشند. عدم توازن سرعت بین پردازنده و حافظه اصلی و داده خارجی وجود دارد.

✓ برای برقرای توازن بین پردازنده و داده خارجی از چند برنامه‌گی و برای چند برنامه‌گی از صفحه بندی استفاده می شود.

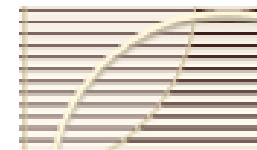
✓ برای برقرای توازن بین پردازنده و حافظه اصلی از حافظه نهان استفاده می شود.

محیط عملیاتی



محیط عملیاتی: متشکل از مجموعه‌ای از حافظه جانبی و دستگاه‌های ورودی و خروجی است. این دستگاه‌ها محیط خارج از کامپیوتر را نشان می‌دهند و هر ارتباطی با کامپیوتر از طریق محیط عملیاتی صورت می‌گیرد. مثل حافظه‌های سریع، حافظه‌هایی با سرعت متوسط، حافظه‌های کند و دستگاه‌های ورودی و خروجی.

انواع کامپیوترها

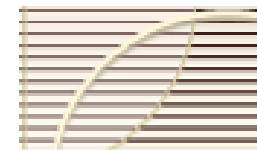


کامپیوتر سخت افزار (hardware): کامپیوتر سخت افزار کامپیوتری است که کاملاً از اجزا سخت افزاری و مدارات الکتریکی ساخته شده است. وضعیت ابتدایی برنامه مطابق با initial state کامپیوتر و نتیجه نهایی برنامه دقیقاً final state کامپیوتر است.

کامپیوترهای میان افزار (firmware): توسط ریز برنامه‌ای شبیه سازی می‌شود که بر روی کامپیوتر سخت افزار قابل ریز برنامه نویسی اجرا می‌گردد. زبان ماشین آن، مجموعه بسیار سطح پایین از ریز دستورات است که انتقال داده‌ها را بین حافظه اصلی و ثباتها، بین خود ثباتها و از ثباتها از طریق پردازنده‌ها انجام می‌دهد.

کامپیوتری را که از طریق شبیه‌سازی ریز برنامه‌ای بوجود می‌آید کامپیوتر مجازی گویند چون از طریق ریز برنامه‌ی شبیه‌سازی شده به وجود آمده است و بدون این ریز برنامه، ماشین وجود نخواهد داشت.

مترجم‌ها و معماری‌های مجازی

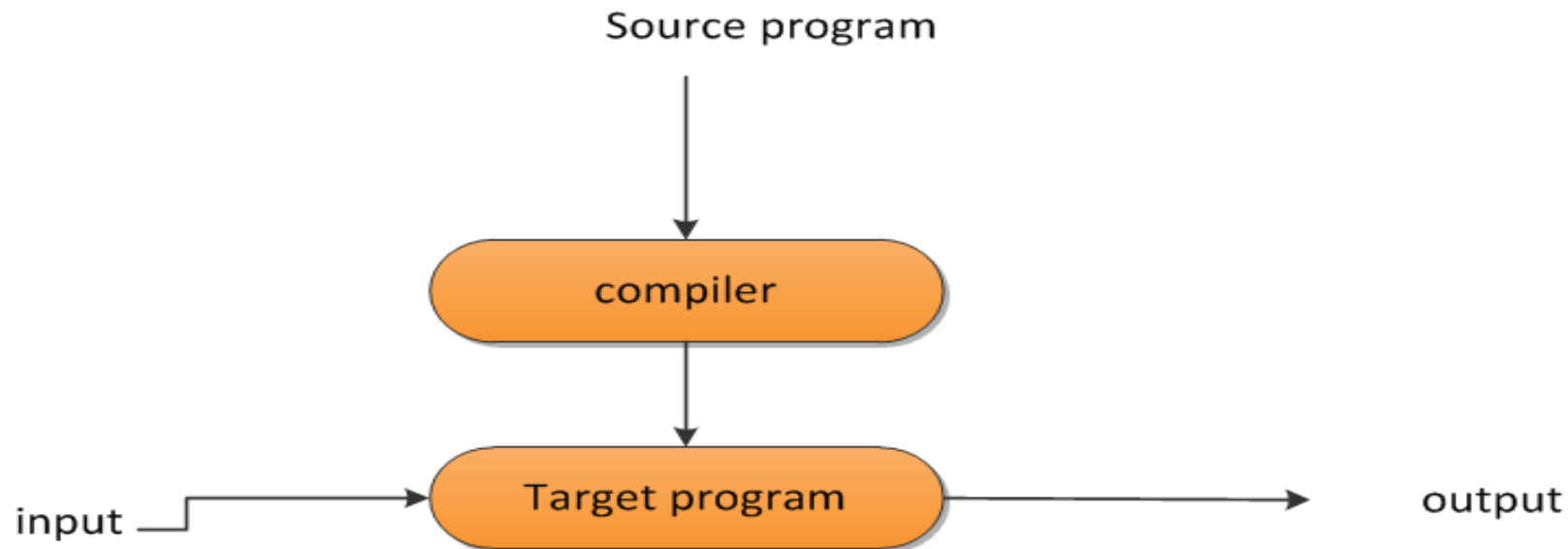


دو روش برای اجرای برنامه سطح بالا در کامپیوتر مجازی:

- ترجمه (کامپایل کردن): در این روش برنامه به زبان سطح بالا طی فرآیندهایی تبدیل به زبان ماشین می‌شود که قابل اجرا روی سخت افزار است.

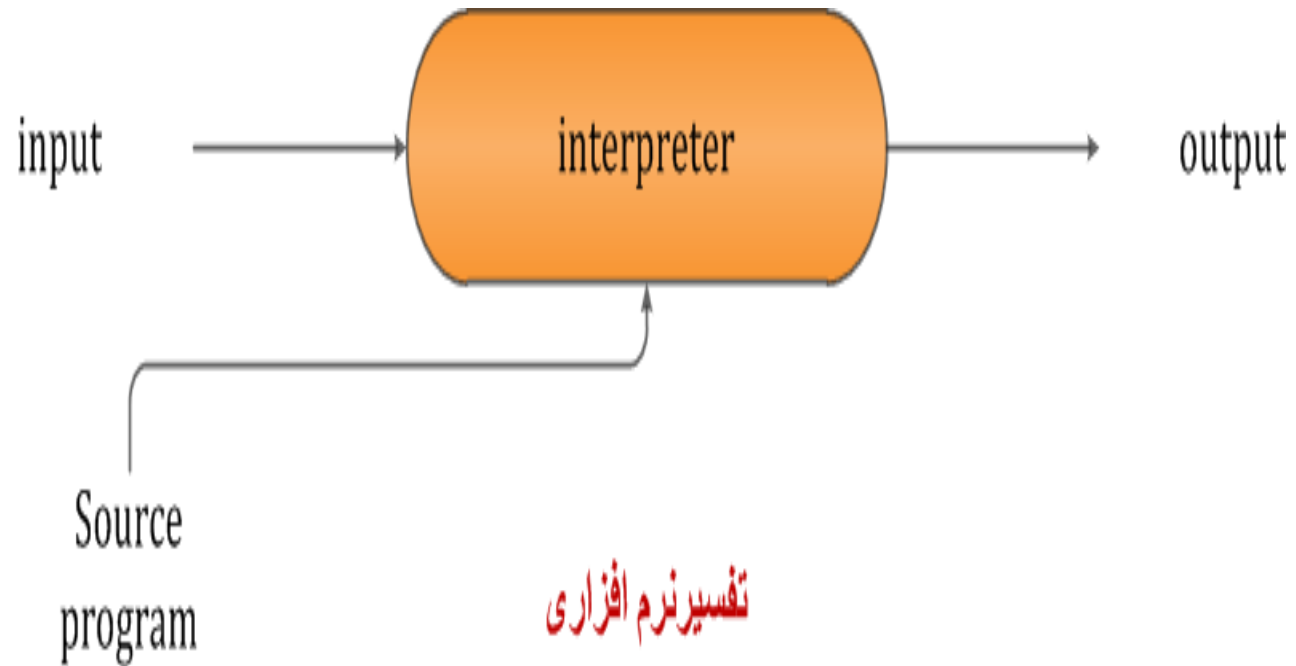
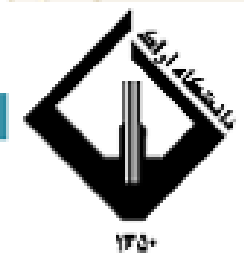
- شبیه‌سازی نرم‌افزاری (تفسیر نرم‌افزاری): در این روش کد برنامه منبع مستقیماً به شبیه‌ساز نرم‌افزاری یا مفسر داده می‌شود و مفسر دستورات زبان سطح بالا را تفسیر و بلافاصله اجرا می‌کند.

نحوه عملکرد مترجم

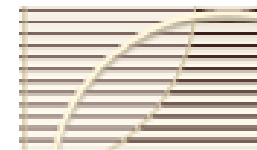


کامپایلر

نحوه عملکرد مفسر

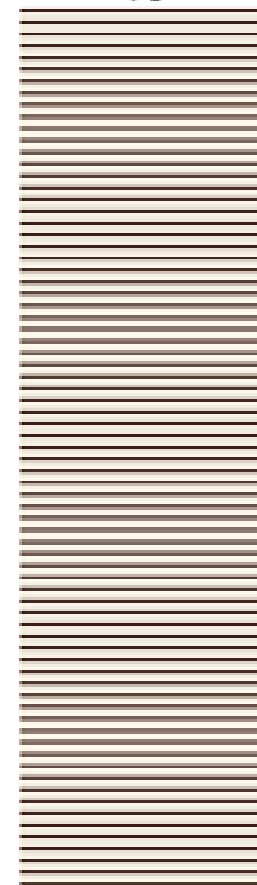
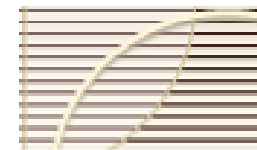
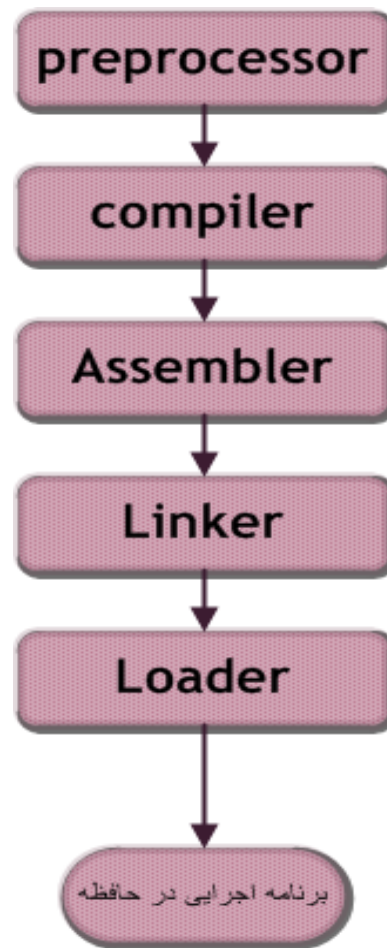


انواع مترجم‌ها

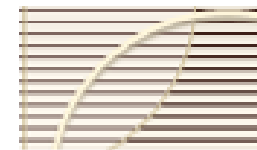


- **کامپایلر:** زبان منبع آن سطح بالا و زبان مقصد آن نزدیک به زبان ماشین است.
- **اسمبلر:** زبان منبع آن اسمبلی و زبان مقصد آن زبان ماشین است.
- **بارکننده یا ویراستار پیوند:** زبان مقصد آن کد ماشین و زبان منبع آن مشابه زبان ماشین است. برنامه را در RAM قرار می‌دهد.
- **پیش پردازنده یا پردازنده ماکرو:** زبان منبع آن شکل توسعه یافته‌ای از سطح بالا و زبان مقصد آن شکل استاندارد آن زبان سطح بالا می‌باشد.

ترتیب اجرای مترجم‌ها



مقایسه روش ترجمه و تفسیری



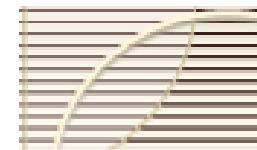
• در ترجمه برنامه ابتدا به زبان ماشین تبدیل شده سپس اجرا می شود ولی در روش تفسیری تک تک دستورات ابتدا تفسیر و مجموعه دستورات برای شبیه سازی آن دستور اجرا می شود.

• سرعت اجرا در ترجمه بیشتر از روش تفسیری است.

• مترجم دستورات را به ترتیب فیزیکی ورودی پردازش می کند ولی شبیه ساز، جریان منطقی برنامه را دنبال می کند.

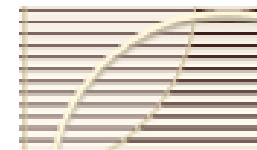
• مترجم، هر دستور را فقط یک بار پردازش می کند ولی مفسر ممکن است دستوری را چندین بار پردازش کند (مانند دستور حلقه).

مقایسه روش ترجمه و تفسیری



- در روش کامپایلری برای n بار اجرا یک ترجمه لازم است ولی در روش تفسیری برای n بار اجرا n ترجمه لازم است.
- ترجمه محض یا شبیه سازی محض در عمل به ندرت استفاده می شوند.
- ایراد مهم ترجمه از دست رفتن اطلاعاتی در رابطه با برنامه است.
- برنامه مقصد در روش ترجمه بزرگ تر از برنامه منبع است ولی در روش تفسیری معمولاً کوچکتر می باشد.
- در روش تفسیری چون دستورات تا زمان اجرا شکل اولیه خود را دارند لذا چند کپی از آنها نگهداری نمی شود. در نتیجه در حافظه صرفه جویی می شود. در مقابل در روش ترجمه چندین فایل داریم که نتیجه ترجمه در آنها ذخیره می شود.

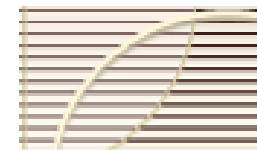
انواع زبان‌ها



۱۳۵۰

- **زبان‌های کامپایلری: C, C++** ، فرترن ، پاسکال و ادا . برنامه‌های آن قبل از شروع اجرای برنامه به زبان ماشین کامپیوتر واقعی ترجمه می‌شوند. به طوریکه شبیه‌سازی به مجموعه‌ای از روال‌های پشتیبانی زمان اجرا محدود می‌شود که اعمال اولیه موجود در زبان منبع را شبیه‌سازی می‌کند که شباهت زیادی به زبان ماشین ندارد.
- **زبان‌های مفسری: لیسپ ، ام ال ، پرل ، پست اسکریپت ، پرولوگ و اسمالتاک** معمولاً با مفسر نرم‌افزاری پیاده‌سازی می‌شود. مترجم کد ماشین را برای کامپیوتر تولید نمی‌کند مفسر شکل میانی از برنامه را تولید می‌کند.

روش‌های ساخت کامپیوتر



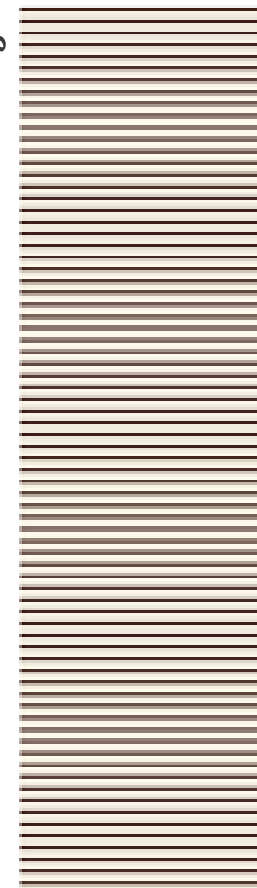
کامپیوتر مجازی: کامپیوتری که با زبانی غیر از زبان ماشین کار می‌کند.

- از طریق سخت افزار: ساختمان داده‌ها و الگوریتم‌ها مستقیماً با دستگاه‌های سخت‌افزاری نمایش داده می‌شوند.
- از طریق میان افزار: ساختمان داده‌ها و الگوریتم‌ها از طریق ریز برنامه‌نویسی نمایش داده می‌شوند.
- از طریق شبیه‌سازی نرم‌افزاری: ساختمان داده‌ها و الگوریتم‌ها از طریق برنامه‌نویسی در زبان‌های دیگر نمایش داده می‌شوند.
- از طریق ترکیبی از این تکنیک‌ها: بخش‌های مختلف کامپیوتر مستقیماً در سخت‌افزار یا به وسیله شبیه‌سازی نرم‌افزاری نمایش داده می‌شوند.

انقیاد(binding)



اختصاص یک ویژگی به مولفه‌ای از برنامه از بین مجموعه‌ای از ویژگی‌ها تقید محسوب می‌شود و زمان این اختصاص نیز زمان تقید نامیده می‌شود.



انواع binding



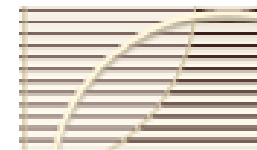
• زمان اجرا

- ✓ در ورود به زیر برنامه یا بلوک: انقیاد پارامترهای مجازی به واقعی در C
- ✓ در هر جای دلخواه از برنامه: اختصاص مقدار به متغیرها

• زمان ترجمه (زمان کامپایل)

- ✓ توسط برنامه نویس: اسامی متغیرها و نوع متغیرها
- ✓ توسط مترجم: آدرس نسبی متغیرهای برنامه در حافظه
- ✓ توسط بارکننده: آدرس واقعی متغیرهای برنامه

انواع binding



- **زمان پیاده‌سازی زبان:** جزئیات مربوط به نمایش داده‌ها و اعمال محاسباتی

- **زمان تعریف زبان:** اغلب ساختارهای زبان برنامه نویسی در زمان تعریف زبان انجام می‌گیرد مثل شکلهای مختلف دستورات، انواع ساختمان داده، ساختارهای برنامه

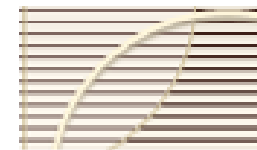
مثال از انقیاد



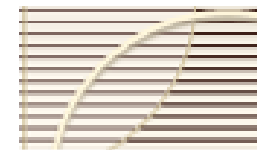
فرض کنید برنامه $x := x + 10$ در زبان L نوشته شده است.

- مجموعه‌ای از انواع ممکن برای متغیر X: انقیاد در زمان تعریف زبان
- نوع متغیر: انقیاد در زمان ترجمه
- مجموعه‌ای از مقادیر ممکن برای X: انقیاد در زمان پیاده‌سازی
- مقدار متغیر X: انقیاد در زمان اجرا
- نمایش مقدار ثابت ۱۰: انقیاد در زمان تعریف زبان
- خواص عملگر +: انقیاد در زمان تعریف زبان

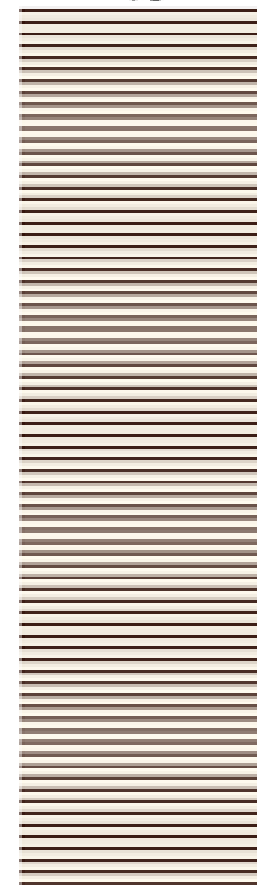
اهمیت زمان‌های انقیاد



- **انقیاد زودرس:** زبان‌هایی که در آن‌ها انقیاد در زمان ترجمه انجام می‌شود.
 - ✓ مثال: فرترن و پاسکال و C
 - ✓ مزایا: کارایی بالاتر (برای مثال کار با آرایه‌های بزرگ و محاسبات راحت‌تر)
- **انقیاد دیررس:** زبان‌هایی که در آنها انقیاد در زمان اجرا انجام می‌شود.
 - ✓ مثال: ام‌ال و لیسپ
 - ✓ مزایا: انعطاف بیشتر (برای مثال دستکاری رشته راحت‌تر)



۱۳۵۰



۱۳۵۰