به نام خدا

درس طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

موضوع: **پردازشگرهای زبان‌های برنامه‌سازی**

ارائه: صادق سلیمانی
www.Bioinformation.ir

جلسه‌ی دوم

عناوین جلسه

- تعریف کامپیوتر
- انواع کامپیوتر
- شش جزء اصلی هر کامپیوتر
- پیاده‌سازی اجزای انواع کامپیوتر
- مقایسه **Compilation** و **Interpretation**
- مثالی از یک کامپیوتر نرم‌افزاری
- سلسله‌مراتب کامپیوترها
- **Binding**

تعریف کامپیوتر

- زبان‌های برنامه‌سازی روی چه ابزار و وسایلی اجرا می‌شوند؟

- کامپیوتر (محاسبه‌کننده):
  - مجموعه‌ایی از ساختمان‌داده‌ها و الگوریتم‌ها که توانایی اجرا و ذخیره‌ی برنامه‌ها را دارند

- انواع کامپیوتر
  1. کامپیوترهای سخت‌افزاری
  2. کامپیوترهای نرم‌افزاری
  3. کامپیوترهای Firmware

  } کامپیوتر مجازی

انواع کامپیوتر

1. کامپیوترهای سخت‌افزاری

   تعریف بدیهی: CPU، I/O، Disk، Register و ...

2. کامپیوترهای نرم‌افزاری
   - به وسیله‌ی روتین‌های نرم‌افزاری روی یک کامپیوتر سخت‌افزاری شبیه‌سازی می‌شود
   - شیوه‌های پیاده‌سازی
     - مستقیم
     - ترجمه‌ی زبان‌سطح بالا به زبان ماشین

3. کامپیوترهای Firmware
   - مجموعه دستورات زبان ماشین، ذخیره شده در ROM
     - مثلاً برای کار با مودم و چاپگر

شش جزء اصلی یک کامپیوتر

1. Data:

نوع داده‌هایی (Data Type) که کامپیوتر در اختیار کاربر قرار می‌دهد

- مثال برای کامپیوتر سخت‌افزاری:

عدد صحیح، حقیقی در ثبات‌ها و Cache

2. Primitive Operations:

عملیاتی برای دستکاری داده‌ها

- مثال برای کامپیوتر سخت‌افزاری:

مانند عملیات زبان ماشین از قبیل: جمع، ضرب، تقسیم، تفریق و ...

3. Sequence Control:

مکانیزم ترتیب اجرای عملیات

- مثال برای کامپیوتر سخت‌افزاری:

ثبات PAR یا PC

شش جزء اصلی یک کامپیوتر

4. Data Control:

مکانیزم‌هایی برای قرار دادن داده در اختیار عملیات (دستیابی به داده)

- مثال برای کامپیوتر سخت‌افزاری:

آدرس خانه‌های حافظه، شماره ثبات

5. Storage Management:

روش‌های در اختیار گذاردن حافظه برای برنامه و داده

- مثال برای کامپیوتر سخت‌افزاری:

DAT (Dynamic Address Translation)

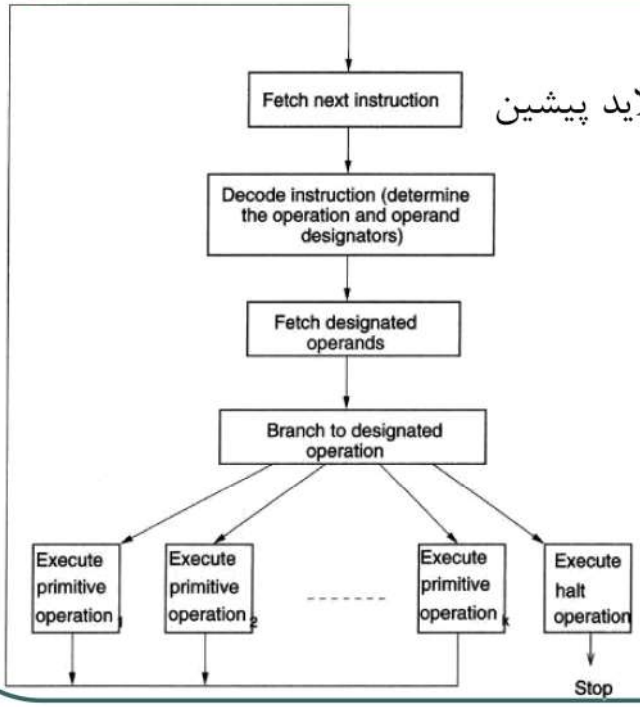
6. Operating Environment:

مکانیزم‌های تبادل اطلاعات با دستگاه‌های جانبی

- مثال برای کامپیوتر سخت‌افزاری: صفحه کلید، صفحه نمایش و ...

پیاده‌سازی اجزای کامپیوتر

- برای کامپیوتر سخت‌افزاری
  - موارد ذکر شده در تشریح اسلاید پیشین



پیاده‌سازی اجزای کامپیوتر

- برای کامپیوتر سخت‌افزاری
  - طراحی Primitive Operationها:

1. CISC
   - Complex Instructions Set Computer
   - دستورات زبان ماشین پیچیده‌تر است
2. RISC
   - Reduced Instructions Set Computer
   - دستورات زبان ماشین ساده‌تر است ولی با روش‌هایی سرعت اجرا را بالا می‌برند
     - افزایش تعداد رجیسترهای کنترلی
     - استفاده از Pipelining

پیاده‌سازی اجزای کامپیوتر

- برای کامپیوترهای Firmware
  - کامپیوتری است که اجزای آن با Micro Operationها روی یک کامپیوتر سخت‌افزاری شبیه‌سازی می‌شود
    - دستورات زبان ماشین آن، Micro Instruction است
    - طبعا تعداد دستورات و امکانات در آن محدود است
    - انتقال داده کماکان بین ثبات‌ها و حافظه‌ی اصلی صورت می‌گیرد
  - به زبان ساده
    - یک ریزبرنامه نوشته می‌شود و در ROM گذارده می‌شود که چرخه‌ی اجرا و تفسیر و ... را در رابطه با دستورالعمل‌ها شبیه‌سازی می‌کند.

پیاده‌سازی اجزای کامپیوتر

- برای کامپیوترهای نرم‌افزاری
  - High Level Language Computers
  - کامپیوتری که زبان ماشین آن یک زبان سطح بالاست
    - در پیاده‌سازی مستقیم کارایی ندارد
      - پیچیده‌بودن مدارها
      - گران بودن آن‌ها
      - انعطاف‌ناپذیری
    - به‌صرفه است که زبان ماشین یک زبان سطح پایین باشد و نرم‌افزارهایی واسطه شوند و زبان سطح بالا را ترجمه کنند

زبان سطح بالا ← (ترجمه) زبان ماشین

پیاده‌سازی اجزای کامپیوتر

- اولین شیوه‌ی اجرای زبان‌های سطح بالا
  - استفاده از Translatorها (واسطه‌های یا مترجم‌ها)

1. Compiler
   - ورودی:
     - برنامه به زبان سطح بالا
   - خروجی:
     - برنامه به زبان اسمبلی
2. Link Editor
   - ورودی:
     - چندین برنامه قابل اجرا
   - خروجی:
     - یک برنامه قابل اجرا: آدرس‌ها و ارتباطات مشخص شده‌اند

پیاده‌سازی اجزای کامپیوتر

- Translatorها (واسطه‌های یا مترجم‌ها)

3. Loader
   - ورودی:
     - برنامه قابل اجرا با آدرس نسبی
   - خروجی:
     - برنامه قابل اجرا با آدرس حقیقی
4. Assembler
   - ورودی:
     - برنامه به زبان اسمبلی
   - خروجی:
     - برنامه به زبان ماشین

پیاده‌سازی اجزای کامپیوتر

- Translatorها (واسطه‌های یا مترجم‌ها)

5. Preprocessor
   - ورودی:
     - فرم گسترده‌ی یا توسعه یافته‌ی زبان
   - خروجی:
     - فرم استاندارد زبان

6. Interpreter
   - ورودی:
     - برنامه به زبان سطح بالا
   - خروجی:
     - برنامه به‌زبان ساده‌تر (لزوماً کد ماشین نیست، ولی اجرای ساده‌تر دارد)

پیاده‌سازی اجزای کامپیوتر

- ارتباط انواع مترجم‌ها

Preprocessor -> Compiler -> Assembler ->
Link Editor -> Loader

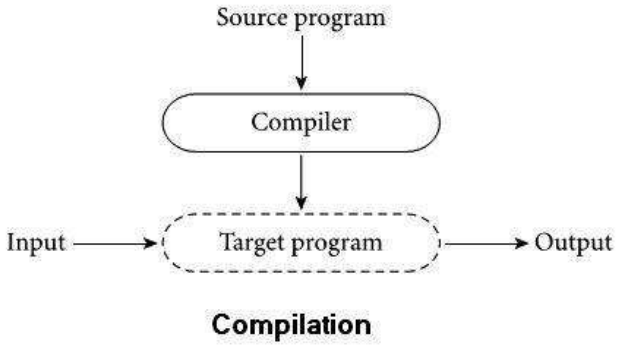
پیاده‌سازی اجزای کامپیوتر

- دومین شیوه‌ی اجرای زبان‌های سطح بالا
  - Software Simulation
    - اجرای مستقیم برنامه‌ی ورودی
    - شبیه‌سازی دستورالعمل‌ها با روتین‌های نرم‌افزاری
- نتیجه
  - انواع زبان‌ها
    1. Compiled Languages
    2. Interpreted Languages

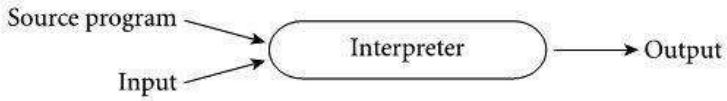
مقایسه Compilation و Interpretation

- زبان‌های کامپایلی
  - ترتیب فیزیکی دستورات دنبال می‌شود
  - حافظه‌ی بیشتری اشغال می‌شود
  - سرعت اجرا بالاتر است
  - تمام دستورات ترجمه می‌شوند
- زبان‌های مفسری
  - ترتیب منطقی دستورات دنبال می‌شود
  - سرعت اجرا کمتر است
  - اجرای تکراری برخی دستورات
  - عدم اجرای برخی دستورات

مقایسه Compile و Interpretation



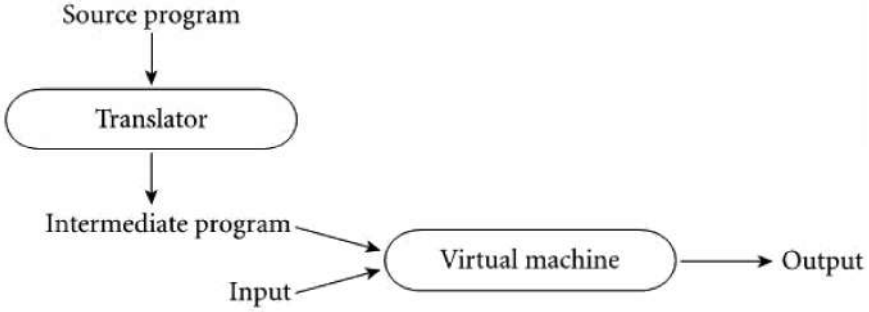
Compilation



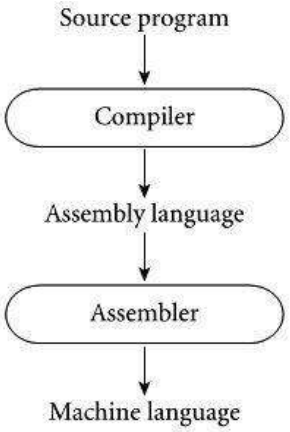
Interpretation

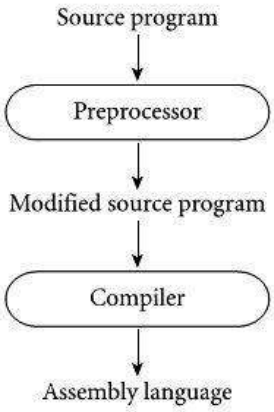
مقایسه Compile و Interpretation

- Pure Compilation و Pure Interpretation به ندرت انجام می‌شود
- زبان‌ها معمولاً ترکیبی از این دو را به کار می‌گیرند
  - کامپایل محض: زبان اسمبلی
  - ترجمه‌ی محض: نسخه‌های اولیه‌ی Basic، Shell Script



مقایسه Compile و Interpretation

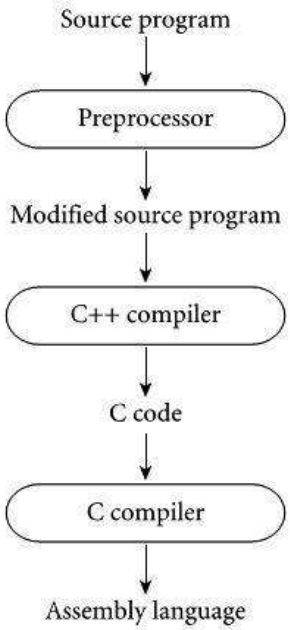




رفتار بسیاری زبان‌ها بدین شکل است

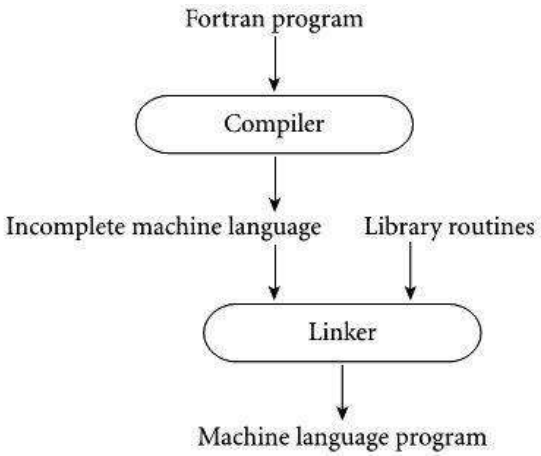
مقایسه Compile و Interpretation

- نحوه‌ی اجرای زبان C++



مقایسه Compile و Interpretation

- نحوه‌ی اجرای زبان فرترن



مثال از یک کامپیوتر نرم‌افزاری

- شش جزء اصلی زبان C

1. Data :
2. Primitive Operations:
3. Sequence Control:
4. Data Control

مثال از یک کامپیوتر نرم‌افزاری

- شش جزء اصلی زبان C
  5. مدیریت حافظه
     -
  6. محیط عملیاتی
     -

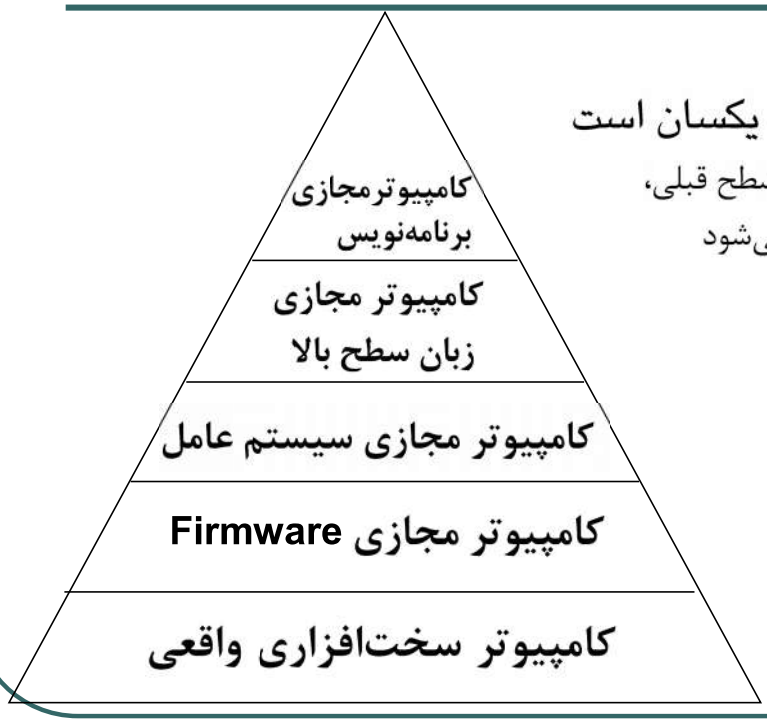
کل درس بر این 6 مورد بنا شده است

سلسله مراتب کامپیوترها

- کامپیوترمجازی برنامه‌نویس
  - برنامه‌ایی که کاربر می‌نویسد و به‌وسیله‌ی زبان، اجرا می‌شود
- کامپیوتر مجازی زبان سطح بالا
  - به وسیله‌ی برنامه‌هایی (کامپایلر و روتین‌های کمکی) که به وسیله‌ی سیستم عامل پیاده‌سازی می‌شود، پیاده می‌شود
- کامپیوتر مجازی سیستم عامل
  - با برنامه‌هایی که بر روی زبان‌ماشین یا Firmware اجرا می‌شوند، پیاده‌سازی می‌شود
- کامپیوتر مجازی Firmware
  - با Micro program که کامپیوتر سخت‌افزاری اجرا می‌کند، پیاده‌سازی می‌شود
- کامپیوتر سخت‌افزاری واقعی: پیاده‌سازی به وسیله‌ی مدارها و اجزای فیزیکی

سلسله مراتب کامپیوترها

- مفهوم داده و برنامه در این سلسله مراتب یکسان است
  - زیرا هر برنامه برای ماشین سطح قبلی، داده محسوب می‌شود



Binding (انتساب)

- یکی از موارد که بسیاری از تفاوت‌های زبان‌ها را رقم می‌زند
- تعریف
  - اختصاص یک صفت یا مشخصه، از بین مجموعه صفات و مشخصات، به یک جزء برنامه
- مثال
  - انتساب یک اسم به یک متغییر

```
int counter;
```

Binding

- انواع Binding (انتساب)
  - بر حسب زمان انجام انتساب

1. در زمان اجرا (Run or Execution Time)
2. در زمان ترجمه (Translation or Compile Time)
3. در زمان پیاده‌سازی (Implementation Time)
4. در زمان تعریف (Definition Time)

1. Binding زمان اجرا

- به متغییر، مقدار منتسب می‌شود
- به متغییر، حافظه‌ی واقعی منتسب می‌شود
- در زمان ورود به زیر برنامه به پارامترهای فرمال مقدار حقیقی منتسب می‌شود
- در یک نقطه ی اختیاری از برنامه (زمانی که دستور انتساب اجرا می‌شود)
  - مثال:

```
cnt=98;
```

2. Binding زمان کامپایل

- به متغییر نام، Type، نحوه‌ی ذخیره و آدرس نسبی منتسب می‌شود
- نحوه‌ی ذخیره، به شرطی در این زمان منتسب می‌شود که در اعلان بیاید
- مثال

DCL x Fixed DEC 99.999

- توسط برنامه نویس انتخاب می‌شود: نوع متغیرها و انتساب‌ها
- توسط کامپایلر انتخاب می‌شود: آدرس‌های نسبی متغیرها و آرایه‌ها
- توسط لودر انتخاب می شود: آدرس های مطلق

3. Binding زمان پیاده‌سازی

- به متغییر، نحوه‌ی نمایش و ذخیره‌ی مقادیر منتسب می‌شود
- به operationها الگوریتم منتسب می‌شود
- مثال:

نمایش عملیات با سخت‌افزار باشد یا نرم‌افزار

محدوده‌ی مقادیر چه باشد مثلا برای عدد صحیح

4. Binding زمان تعریف

- اکثر مفاهیم زبان در زمان تعریف منتسب می‌شود
- نوع داده های قابل دسترسی و ساختارهای زبان در این زمان منتسب می‌شود

- مثال:

در c++ دستور انتساب بصورت = در حالی در پاسکال بصورت := باشد

و به همین ترتیب

For، ; ، + و ...

مثال از Binding

- x:=x+10;
  - به متغییر x، در حین کامپایل منتسب شده
  - در زمان تعریف زبان منتسب شده
  - به متغییر x در زمان Compile منتسب شده
  - در زمان پیاده‌سازی منتسب می‌شود
  - به علایم =: ، + و ; مفهوم منتسب شده
  - ، در هر نقطه از برنامه ممکن است مقداری به x، منتسب شود
  - به علامت 10، مفهوم در زمان ، منتسب شده
  - به‌علامت10،نحوه‌ی ذخیره در حافظه، در زمان ، منتسب شده

انواع زبان‌ها برحسب نوع **Binding**

- انتساب زودرس (Early Binding)
  - اکثر انتساب‌ها در زمان کامپایل
  - سرعت اجرا بالاست
    - مثال

      C، پاسکال، فرترن، کوبول
- انتساب دیرهنگام (Late Binding)
  - تعداد زیادی از انتساب‌ها در زمان اجرا صورت می‌گیرد
  - انعطاف‌پذیری بیشتر است
    - مثال:

      SNOBOL ،Basic ،Lisp

انواع زبان‌ها برحسب نوع **Binding**

- حتی پیاده‌سازی‌های مختلف یک زبان خاص ممکن است به دو نوع مختلف Early Binding و Late Binding باشند
- زمان انتساب روی **توانایی‌های زبان، مؤثر است**
  - Fortran و SNOBOL
    - هر دو آرایه و عملیات ریاضی دارند
    - اما فرترن برای عملیات ریاضی و SNOBOL برای پردازش رشته مناسب است
    - چون فرترن Early Binding و SNOBOL، Late Binding است
      - یعنی عملیات ریاضی بر اعداد سریعتر بشود

        و عملیات بر روی رشته‌ها پویاتر

پرسش؟