به نام خدا

مبانی برنامه‌نویسی

آرش شفیعی

کتاب‌های مرجع

- زبان برنامه‌نویسی سی، از کرنیگان و ریچی [1]
- چگونه توسط سی برنامه بنویسیم، از دایتل و دایتل [2]
- هنر برنامه نویسی، از دونالد کنوث [3]

[1] The C Programming Language، by Brian Kernighan and Dennis Ritchie
[2] C How To Program، by Paul Deitel and Harvey Deitel
[3] The Art of Computer Programming، by Donald Knuth

معرفی سیستم‌های کامپیوتری

معرفی سیستم‌های کامپیوتری

- کامپیوتر وسیله‌ای است که با دریافت تعدادی ورودی و دریافت دنباله‌ای از عملیات منطقی و حسابی می‌تواند برروی ورودی‌ها محاسباتی انجام داده و نتیجه عملیات را به عنوان خروجی بازگرداند.
- سریع‌ترین کامپیوترها امروزه می‌توانند صدها کوادریلیون [1] یعنی در حدود $۱۰^{۱۷}$ عملیات محاسباتی در ثانیه انجام دهند. به عبارت دیگر اگر جمعیت کرهٔ زمین را ۸ میلیاد نفر در نظر بگیریم، سریع‌ترین کامپیوترهای امروزی قادرند در یک ثانیه ده‌ها میلیون عملیات محاسباتی به ازای هر یک از افراد کرهٔ زمین انجام دهند.

[1] quadrillion

معرفی سیستم‌های کامپیوتری

- دنبالهٔ عملیات منطقی و حسابی که توسط یک کامپیوتر دریافت می‌شود را یک برنامهٔ کامپیوتری [1] یا یک برنامه می‌نامیم. یک برنامه که توسط یک برنامه‌نویس [2] به کامپیوتر داده می‌شود، درواقع تعیین می‌کند چگونه ورودی‌ها پردازش شوند.

[1] computer program
[2] programmer

معرفی سیستم‌های کامپیوتری

- یک کامپیوتر در واقع یک ماشین انتزاعی است. این ماشین انتزاعی را می‌توان توسط قطعات فیزیکی–الکترونیکی پیاده‌سازی کرد.
- موتور تحلیلی چالرز بابیج [1] و ماشین تورینگ جهانی [2] دو نمونهٔ ابتدایی یک کامپیوتر انتزاعی هستند. ماشین تورینگ از یک واحد کنترل کننده و یک نوار ورودی تشکیل شده است.
- برای پیاده‌سازی‌های کامپیوترهای فیزیکی که آن‌ها را سیستم‌های کامپیوتری می‌نامیم از یک واحد محاسبات مرکزی [2] و یک واحد حافظه [3] استفاده می‌شود و برای ارسال ورودی‌ها به کامپیوتر از واحد ورودی [4] شامل موس و کیبورد و غیره و برای دریافت خروجی‌ها از یک واحد خروجی [5] شامل مانیتور استفاده می‌شود.

---

[1] Charles Babbage Analytical Engine
[2] Universal Turing Machine
[2] Central Processing Unit (CPU)
[3] Memory unit
[4] Input unit
[5] output unit

معرفی سیستم‌های کامپیوتری

- سیستم‌های کامپیوتر در ده‌ها سال گذشته به اندازهٔ یک اتاق بودند و میلیون‌ها دلار برای تولید آن‌ها هزینه می‌شد ولی سیستم‌های کامپیوتری امروزی با استفاده از قطعات سیلیکونی تولید می‌شوند و علاوه بر حجم بسیار کمی که دارند، هزینهٔ بسیار پایینی نیز برای تولید آنها صرف می‌شود.
- در مورد فراوانی مادهٔ سیلیکون باید گفت که کرهٔ زمین $۵/۹۷ \times ۱۰^{۲۴}$ کیلوگرم وزن دارد که ۳۲ درصد آن را آهن، ۳۰ درصد را اکسیژن، و ۱۵ درصد آن را سیلیکون تشکیل داده است. به عبارت دیگر، سیلیکون سومین عنصری است که در زمین به وفور یافت می‌شود و بنابراین هزینهٔ بسیار پایینی دارد.

معرفی سیستم‌های کامپیوتری

- واحدهای اصلی سازندهٔ یک سیستم کامپیوتری واحد پردازندهٔ مرکزی و واحد حافظه هستند.

معرفی سیستم‌های کامپیوتری

- واحد حافظه داده‌های ورودی را پس از دریافت از واحد ورودی و داده‌های خروجی را قبل از ارسال به واحد خروجی در خود نگهداری می‌کند. همچنین در هنگام پردازش داده‌ها، ممکن است نیاز به تولید داده‌هایی باشد که این داده‌ها نیز بر روی حافظه نگهداری می‌شوند. واحد حافظه معمولاً حافظه دسترسی تصادفی [1] (RAM) نامیده می‌شود زیرا به هر قسمت از داده‌ها می‌توان به طور تصادفی دسترسی پیدا کرد.
- در سیستم‌های کامپیوتری امروزی مقدار حافظه به ده‌ها گیگابایت [2] می‌رسد. یک گیگابایت در حدود یک میلیارد بایت است و یک بایت از هشت بیت تشکیل شده است. یک بیت می‌تواند مقدار صفر یا یک را در خود نگهدار کند.

[1] Random Access Memory
[2] gigabyte

معرفی سیستم‌های کامپیوتری

- واحد پردازندهٔ مرکزی قسمت کنترل کنندهٔ یک سیستم کامپیوتری است که به وسیلهٔ آن اطلاعات از ورودی دریافت می‌شوند، توسط عملیات حسابی (مانند جمع و تفریق و ضرب و تقسیم) و عملیات رابطه‌ای (مانند مقایسه) و عملیات منطقی (مانند عطف و فصل و نقیض) محاسبه می‌شوند و اطلاعات پردازش شده به واحد خروجی ارسال می‌شوند.
- امروزه بسیاری از پردازنده‌ها از چند هسته [1] تشکیل شده‌اند و هسته‌ها می‌توانند به طور مجزا محاسبات را انجام دهند و بدین ترتیب با استفاده از پردازش موازی (پردازش توسط چند هسته) محاسبات می‌توانند سریع‌تر انجام شوند.

[1] multicore

معرفی سیستم‌های کامپیوتری

- کوچکترین واحد اطلاعات در سیستم‌های کامپیوتری یک بیت است. یک بیت می‌تواند مقدار صفر یا یک را نگهداری کند. توسط بیت‌ها می‌توان اعداد را در مبنای دو نمایش داد. همهٔ اطلاعات در سیستم‌های کامپیوتری به صورت دودویی نگهداری می‌شوند.
- یک بایت (B) برابر با $۲^{۳}$ یا ۸ بیت است.
- یک کیلوبایت (KB) برابر با $۲^{۱۰}$ یا ۱۰۲۴ بایت است.
- به همین ترتیب یک مگابایت (MB) برابر با $۲^{۲۰}$ بایت یا ۱۰۲۴ کیلوبایت، یک گیگابایت (GB) برابر با $۲^{۳۰}$ بایت یا ۱۰۲۴ مگابایت، یک ترابایت (TB) برابر با $۲^{۴۰}$ بایت یا ۱۰۲۴ گیگابایت، یک پتابایت (PB) برابر با $۲^{۵۰}$ بایت یا ۱۰۲۴ ترابایت، می‌باشد.

معرفی سیستم‌های کامپیوتری

- برای نمایش حروف از یک سیستم کدگذاری استفاده می‌شود. در استاندارد کدگذاری آمریکایی برای تبادل اطلاعات [1] (ASCII) هر کاراکتر با یک عدد یک بایتی نمایش داده می‌شود، بنابراین در این سیستم کدگذاری می‌توان تنها ۱۲۸ حرف یا کاراکتر را نمایش داد. در کامپیوترهای ابتدایی یک بیت برای بررسی خطاها در نظر گرفته شده بود، بنابراین از ۷ بیت برای کدگذاری استفاده می‌شد.
- سیستم کدگذاری یونیکد [2] برای نمایش حروف زبان‌های غیر انگلیسی استفاده می‌شود. در این سیستم کدگذاری یک حرف می‌تواند با یک عدد ۲، ۳ یا ۴ نمایش داده شود و بنابراین با استفاده از یک نمایش ۴ بایتی می‌توان تا حدود ۴ میلیارد حرف را نمایش داد.

[1] American Standard Code for Information Interchange
[2] unicode

مبنای اعداد

- تبدیل اعداد دهدهی [1] به دودویی [2]: $(x)_{10} = (a_n a_{n-1} \cdots a_1 a_0)_2$ به طوری که $x = \Sigma_{i=0}^{n} a_i \times 2^i$ و $a_i \in \{0, 1\}$
- مثال: $(42)_{10} = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
  $(42)_{10} = 32 + 8 + 2 = (101010)_2$
- اعداد دهدهی می‌توانند علاوه بر قسمت صحیح [3] قسمت اعشاری [4] نیز داشته باشند.
- تبدیل اعداد دهدهی اعشاری به دودویی: $(0.x)_{10} = (0.a_1 a_2 \cdots a_{n-1} a_n)_2$ به طوری که $0.x = \Sigma_{i=1}^{n} a_i \times 2^{-i}$ و $a_i \in \{0, 1\}$
- مثال: $(0.75)_{10} = 1 \times 2^{-1} + 1 \times 2^{-2}$
  $(0.75)_{10} = 0.5 + 0.25 = (0.11)_2$

[1] decimal
[2] binary
[3] integer part
[4] fractional part

مبنای اعداد

- روش تبدیل اعداد دهدهی اعشاری به دودویی: عدد دهدهی را n بار در ۲ ضرب می‌کنیم تا یا عدد به دست آمده قسمت اعشاری نداشته باشد و یا n از تعداد ارقام اعشاری مورد نیاز در عدد دودویی بیشتر شود. سپس عدد دهدهی بدون قسمت اعشاری را به دودویی تبدیل می‌کنیم و در عدد دودویی به دست آمده n رقم از سمت راست جدا می‌کنیم و ممیز اعشار را بعد از n رقم قرار می‌دهیم (در واقع عدد دودویی به دست آمده را n بار بر ۲ تقسیم می‌کنیم).
- مثال: معادل عدد دهدهی $(۴.۷۵)_{۱۰}$ را در مبنای دو را محاسبه کنید.
- $۴.۷۵ \times ۲ \times ۲ = ۱۹$
$(۱۹)_{۱۰} = (۱۰۰۱۱)_{۲}$
$(۱۰۰۱۱)_{۲} \div ۲ \div ۲ = (۱۰۰.۱۱)_{۲}$
$(۴.۷۵)_{۱۰} = (۱۰۰.۱۱)_{۲}$

مبنای اعداد

- مثال: معادل عدد دهدهی $(۰.۳)_{۱۰}$ را در مبنای دو تا ۱۴ رقم اعشار محاسبه کنید.

- $۰.۳ \times ۲^{۱۴} = ۴۹۱۵.۲$
$(۴۹۱۵)_{۱۰} = (۱۰۰۱۱۰۰۱۱۰۰۱۱)_{۲}$
$(۱۰۰۱۱۰۰۱۱۰۰۱۱)_{۲} \div ۲^{۱۴} = (۰.۰۱۰۰۱۱۰۰۱۱۰۰۱۱)_{۲}$
$(۰.۰۱۰۰۱۱۰۰۱۱۰۰۱۱)_{۲} = (۰.۳)_{۱۰}$

مبنای اعداد

- روش تبدیل اعداد دودویی اعشاری به دهدهی: عدد دودویی را n بار در ۲ ضرب می‌کنیم تا عدد به دست آمده قسمت اعشاری نداشته باشد. سپس عدد دودویی بدون قسمت اعشاری را به دهدهی تبدیل می‌کنیم و عدد دهدهی به دست آمده را n بار بر ۲ تقسیم می‌کنیم.
- مثال: عدد دودویی $(۱۰۰.۱۱)_۲$ را به دهدهی تبدیل کنید.
- $(۱۰۰.۱۱)_۲ \times ۲ \times ۲ = (۱۰۰۱۱)_۲$
$(۱۰۰۱۱)_۲ = (۱۹)_{۱۰}$
$۱۹ \div ۲ \div ۲ = ۴.۷۵$
$(۱۰۰.۱۱)_۲ = (۴.۷۵)_{۱۰}$

مبنای اعداد

- یک عدد دودویی را می‌توانیم به صورت یک عدد علامت‌دار [1] یا یک عدد بدون علامت [2] تعبیر کنیم.
- اولین بیت (رقم) از سمت چپ یک عدد را برای نشان دادن علامت آن عدد استفاده می‌کنیم و آن را بیت علامت [3] می‌گوییم.
- اگر بیت علامت برابر با ۱ باشد عدد منفی است و اگر بیت علامت برابر با صفر باشد عدد مثبت است.

[1] signed
[2] unsigned
[3] sign bit

مبنای اعداد

- برای تبدیل یک عدد دودویی علامتدار $(b)_۲$ با بیت علامت ۱ به مبنای دهدهی ابتدا مکمل دو b را محاسبه می‌کنیم. فرض کنیم عدد به دست آمده عدد c است. حال عدد c را به مبنای ده تبدیل می‌کنیم و فرض می‌کنیم عدد به دست آمده برابر با d است: $(c)_۲ = (d)_{۱۰}$
- عدد دودویی b یک عدد منفی است که در مبنای ده برابر است با $-d$ بنابراین: $(b)_۲ = (-d)_{۱۰}$
- برای محاسبهٔ مکمل‌دو [1] یک عدد صفرها را به یک و یک‌ها را به صفر تبدیل کرده، سپس یک واحد به آن عدد می‌افزاییم.
- مثال: عدد بدون علامت $(۱۰۰۱)_۲$ در مبنای دو برابر است با ۹.
- اما عدد علامت‌دار $(۱۰۰۱)_۲$ در مبنای دو برابر است با -۷.

[1] two's complement

مبنای اعداد

- برای تبدیل یک عدد منفی دهدهی به یک عدد منفی دودویی، ابتدا آن عدد را به صورت مثبت در نظر گرفته، آن را به مبنای دو تبدیل کرده، سپس مکمل‌دو آن را محاسبه می‌کنیم.
- مثال: معادل عدد ۴۲− را در مبنای دو محاسبه کنید.
- $(۴۲)_{۱۰} = (۰۱۰۱۰۱۰)_{۲}$
  $(-۴۲)_{۱۰} = (۱۰۱۰۱۱۰)_{۲}$

مبنای اعداد

- اعداد دودویی را می‌توانیم با استفاده از روش زیر به اعداد پایهٔ شانزده (شانزده شانزدهی یا هگزادسیمال [1]) تبدیل کنیم.
- عدد دودویی را از سمت راست چهار بیت چهار بیت جدا می‌کنیم و معادل هگزادسیمال هر چهاربیت را از سمت راست می‌نویسیم. اعداد چهاربیتی می‌توانند بیت ۰ تا ۱۵ باشند. در مبنای شانزده، عدد ۱۰ را با A ، ۱۱ را با B ، ۱۲ را با C ، ۱۳ را با D ، ۱۴ را با E ، و ۱۵ را با F نشان می‌دهیم.
- مثال: معادل عدد ۴۲ را در مبنای شانزده محاسبه کنید.
- $(۴۲)_{۱۰} = (۱۰۱۰۱۰)_{۲} = (۲A)_{۱۶}$

[1] hexadecimal

ماشین و زبان اسمبلی

- هر پردازنده معمولاً یک زبان مختص به خود دارد که به آن زبان ماشین [1] گفته می‌شود. این زبان توسط طراحان پردازنده طراحی می‌شود.
- برای مثال برای جمع دو عدد، یک پردازنده به طور قراردادی یک عدد دودویی را به عنوان عملگر جمع در نظر می‌گیرد و دو عملوند ورودی را به صورت دو عدد دودویی برای عملگر جمع دریافت می‌کند.
- زبان اسمبلی [2] زبانی است بسیار شبیه به زبان ماشین که در آن دستوراتی که توسط پردازنده انجام می‌شوند به نحوی نامگذاری شده‌اند که توسط انسان قابل خواندن و فهمیدن هستند. پردازنده‌ها معمولاً زبان اسمبلی مخصوص خود را دارند. در زبان اسمبلی از کلمات زبان انگلیسی به طور قراردادی برای انجام عملیات توسط پردازنده‌ها استفاده شده است. برای مثال در یک زبان اسمبلی ممکن است از دستور `add` برای جمع دو عدد استفاده شود.
- برنامه مترجمی که زبان اسمبلی را به زبان ماشین تبدیل می‌کند اسمبلر [3] نامیده می‌شود.

[1] machine langague
[2] assembly language
[3] assembler

زبان سطح بالا

- اگر کامپیوتر و زبان مختص دستورات کامپیوتری را در پایین‌ترین سطح در نظر بگیریم و انسان و زبان‌های طبیعی را در بالاترین سطح، آنگاه زبان ماشین یک زبان سطح پایین محسوب می‌شود که در پایین‌ترین سطح این تقسیم‌بندی قرار دارد. زبان اسمبلی در سطحی بالاتر از زبان ماشین قرار می‌گیرد ولی همچنان به آن یک زبان سطح پایین گفته می‌شود چون به زبان کامپیوتر نزدیک‌تر است.
- با افزایش پردازنده‌ها و تنوع زبان‌های اسمبلی نیاز به زبان یا زبان‌هایی بود که به صورت یکپارچه برنامه‌نویسان بتوانند با استفاده از این زبان‌ها برای همهٔ پردازنده‌ها و معماری‌های متفاوت برنامه بنویسند. همچنین زبان اسمبلی به زبان ماشین بسیار نزدیک بود و بهتر بود زبان یا زبان‌هایی به وجود می‌آمدند که به زبان برنامه‌نویسان نزدیک‌تر باشند.
- چنین زبان‌هایی، که در دهه ۱۹۵۰ به وجود آمدند، زبان‌های سطح بالا [1] نامیده می‌شوند.

[1] high-level languages

زبان سطح بالا

- برای تبدیل یک برنامه در زبان سطح بالا به یک برنامه به زبان اسمبلی از برنامه‌ای به نام مترجم یا کامپایلر [1] استفاده می‌شود. با استفاده از یک زبان سطح بالا می‌توان دستوراتی نوشت که به زبان انگلیسی و زبان ریاضی شباهت بیشتری دارند تا زبان ماشین.
- اولین زبان برنامه‌نویسی، در یک رسالهٔ دکتری در سال ۱۹۵۱ توسط کورادو بوهم [2] در دانشگاه ای‌تی‌اچ زوریخ توصیف شد و به همراه یک کامپایلر عرضه شد.
- دو زبان مهم تجاری که در این دهه به وجود آمدند، عبارتند از فورترن [3] و کوبول [4] .
- نوآوری جدید فورترن این بود که به برنامه‌نویس کمک می‌کرد تا بتواند فرمول‌های ریاضی را به همان صورتی که برروی کاغذ نوشته می‌شوند بنویسید. در واقع کلمهٔ فورترن مخفف کلمهٔ ترجمهٔ فرمول [5] بود.

[1] compiler
[2] Corrado Bohm
[3] Fortran
[4] Cobol
[5] Formula Translation

زبان سطح بالا

- قبل از به وجود آمدن زبان فورترن برنامه‌نویسان برای نوشتن فرمول $i + ۲ * j$ نیاز بود مقدار i و j را در قسمتی از حافظه ذخیره کنند . سپس مقدار j را دو برابر کنند و سپس مقدار i را با دو برابر j جمع کنند، اما با استفاده از فورترن برنامه‌نویسان می‌توانستند فرمول را به شکلی که روی کاغذ می‌نوشتند در برنامه خود وارد کنند.

سیستم عامل

- سیستم عامل [1] برنامه‌ای است که برای مدیریت قطعات جانبی یک سیستم کامپیوتری مانند ورودی، خروجی‌ها و همچنین مدیریت منابع مانند حافظه و پردازنده استفاده می‌شود. با استفاده از یک سیستم عامل می‌توان برنامه‌ها را به طور همزمان اجرا کرد و برای اجرای همزمان برنامه‌ها نیاز به تخصیص حافظه و زمان پردازنده به برنامه‌ها به طور بهینه است.

[1] operating system

سیستم عامل

- سیستم عامل یونیکس [1] که یکی از مهم‌ترین سیستم عامل‌های ابتدایی است در سال ۱۹۶۹ بر روی یک کامپیوتر PDP۷ با استفاده از زبان اسمبلی توسط دنیس ریچی [2] و کن تامسون [3] در آزمایشگاه‌های بل طراحی و پیاده‌سازی شد.

[1] Unix
[2] Dennis Richie
[3] Ken Thompson

زبان سی

- یونیکس در نسخهٔ بعدی برای یک کامپیوتر PDP۱۱ پیاده‌سازی شد و از آنجایی که برای کامپیوتر جدید به تعدادی ابزار نیاز بود، طراحان تصمیم گرفتند کامپایلری برای یک زبان سطح بالا طراحی کنند تا ابزارها را بتوان با استفاده از آن زبان سطح بالا راحت‌تر پیاده‌سازی کرد. در آن زمان زبان BCPL طراحی شده بود. طراحان یونیکس با استفاده از ایده‌های این زبان، و همچنین زبان الگول [4] کامپایلری برای یک زبان جدید طراحی و پیاده‌سازی کردند و زبان جدید را B نامیدند.
- بین سال‌های ۱۹۷۱ و ۱۹۷۲ به تدریج امکاناتی به زبان B افزوده شد و در نتیجه زبان جدیدی به وجود آمد که بعدها زبان C نامیده شد.
- در سال ۱۹۷۸ اولین نسخه از کتاب زبان برنامه نویسی سی [1] منتشر کرد.

[4] Algol

[1] The C Programming Language

زبان سی

- زبان سی به عنوان یکی از زبان‌های بسیار کارامد برای سال‌ها در کنار اسمبلی تنها زبان سطح بالایی بود که در توسعه سیستم عامل لینوکس استفاده می‌شد. در سال ۲۰۲۲ برای اولین بار از زبان سطح بالای راست [1] برای توسعه سیستم عامل لینوکس استفاده شد.

[1] Rust

زبان سی

- علاوه بر سیستم عامل‌ها، زبان سی در بسیاری از سیستم‌های نهفته [1] شامل سیستم‌های تعبیه شده در لوازم خانگی، ربات‌ها، سیستم‌های هواپیمایی و سیستم‌های کنترلی استفاده می‌شود.
- علت عمده اهمیت زبان سی در کارامدی بالا و سادگی استفاده از این زبان است.
- برای مثال در یک سامانهٔ کنترل ترافیکی کوچکترین تأخیری در سیستم می‌تواند خطرآفرین باشد و بنابراین نیاز به زبانی است که تا حد امکان با سرعت بالایی برنامه‌ها را اجرا کند.
- به عنوان مثال دیگر در سامانه‌های ارتباطی نیاز به ارسال و دریافت سریع اطلاعات صوتی و تصویری و کدگذاری و کدگشایی آنهاست و تأخیر در این سامانه‌ها باعث اختلال در ارتباط می‌شود و بنابراین به زبانی با سرعت اجرای بالا نیاز است.

[1] embedded systems

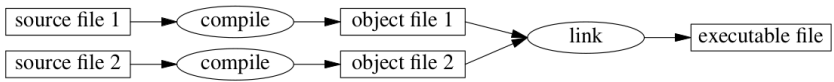
زبان سی

- زبان سی در زمان ابداع به شیوه‌های متفاوتی تعمیم داده شده و پیاده‌سازی شد و در سال ۱۹۸۹ توسط مؤسسهٔ استاندارد ملی آمریکا [1] به صورت استاندارد درآمد.

[1] Amercan National Standards Institute (ANSI)

زبان سی

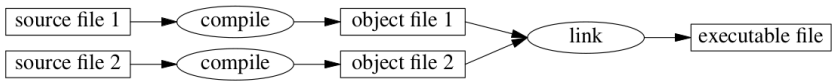
- کامپایلر سی متن برنامه را که در یک فایل منبع یا فایل سورس [1] نگهداری می‌شود به زبان ماشین ترجمه می‌کند و فایل‌هایی به نام فایل آبجکت [2] می‌سازد که حاوی برنامه به زبان ماشین مقصد برای اجرا است.
- سپس فایل‌های آبجکت باید توسط پیوند دهنده یا لینکر [3] به یکدیگر پیوند داده شوند و یک فایل اجرایی برای اجرا تهیه شود. معمولاً یک برنامه از تعداد زیادی فایل سورس تشکیل شده است.



[1] source file
[2] object file
[3] linker

زبان سی

- گاهی برای برنامه‌ای که توسط برنامه‌نویسان دیگر نوشته شده است فایل سورس در اختیار برنامه‌نویس قرار نمی‌گیرد بلکه فایل‌های آبجکت ارائه می‌شوند.
- در چنین مواردی لینکر وظیفه دارد فایل‌های آبجکت یک برنامه و فایل‌های آبجکت برنامه‌های مورد نیاز دیگر را به صورت یک فایل یکپارچه اجرای درآورد.



- در نهایت یک برنامه اجرایی [1] در قالب یک فایل اجرایی [2] برای یک پردازندهٔ مقصد تهیه می‌شود. این فایل قابل انتقال [3] نیست، بدین معنی که نمی‌توان آن را از یک ماشین یا سیستم عامل به یک ماشین یا سیستم عامل متفاوت دیگر انتقال داد و اجرا کرد.

[1] executable program
[2] executable file
[3] portable