

به نام کسی که آفرید از عدم

به انسان عطا کرد فکر و قلم

موضوع مقاله: مزایا و معایب زبان C

تاریخ: 1389/5/28

ارایه دهنده: سید محمد میری

نام استاد: مهندس سلیمی

WWW.SALIMITEACH.COM

از هر زبان کامپیوتری که استفاده کنید ظاهر سورس کد خاص خود را دارد . APL ، ظاهر هیروگلیفی خاص خود را دارد . زبان اسمبلی ، ستون های ممتد حافظه خود را دارد و پاسکال ، دستور زبان خوانای خود را دارد . بعد نوبت به C می رسد . بسیاری از برنامه نویسانی که برای اولین بار با C مواجه می شوند ، دستور زبان آن را رمزنگارانه و شاید ترس آورمی بینند . C شامل تعداد کمی از ساختارهای دستوری شبه انگلیسی که در زبان های برنامه نویسی دیگر یافت می شوند ، می باشد . در عوض ، زبان C عملگرهایی دارد که غیر معمول به نظر می رسند و اشاره گرهای بسیار زیاد و بیش از حدی را در اختیار مهندس نرم افزار قرار می دهد . برنامه نویسان جدید C به زودی تعداد زیادی خصوصیات مختلف زبانی را پیدا می کنند که ریشه آنها به جد اولیه نرم افزاری / سخت افزاری C برمی گردد . قسمت های زیر نقاط قوت زبان C را نشان می دهند .

اندازه کد بهینه

قواعد دستوری کمتری در زبان C نسبت به بسیاری از زبان ها وجود دارد و می توان یک کامپایلر C با کیفیت بالا نوشت که در فقط K 256 حافظه کارکند . در واقع در زبان C بیشتر از آنکه کلمه کلیدی وجود داشته باشد ، عملگرها و ترکیب عملگرها وجود دارد .

مجموعه کوچکی از کلمات کلیدی

زبان C اولیه که توسط Dennis Ritchie ایجاد شد ، فقط شامل 27 کلمه کلیدی بود . استاندارد ANSI C (که بعدا مورد بحث قرار می گیرد) چندین کلمه رزرو شده را اضافه کرده است . ++Microsoft C/C مجموعه دستورالعمل ها را بیشتر گسترش داده است و تعداد کلمات کلیدی ++Microsoft C/C را به بیش از 70 رساند . بسیاری از توابعی که به طور معمول به عنوان بخشی از زبان های برنامه نویسی دیگر تعریف می شوند ، در زبان C وجود ندارند . برای مثال ، C هیچ قابلیت نهادین برای ورودی و خروجی ندارد و نیز هیچ عملیات ریاضی (به غیر از جمع و تفریق مقدماتی) یا توابع کار با رشته ندارد . از آنجایی که هر زبانی که این قابلیت ها را ندارد استفاده زیادی ندارد ، زبان C یک مجموعه غنی از توابع کتابخانه ای را برای ورودی و خروجی ، عملیات ریاضی و کار با رشته ها ارائه کرده است . این مجموعه کتابخانه های توافق شده ، آن چنان به طور رایج مورد استفاده قرار می گیرند که تقریباً می توان آن را به عنوان بخشی از خود زبان دانست . با این حال ، یکی از نقاط قوت C ، ساختار آزاد آن است که شما را قادر می کند تا این توابع را به آسانی دوباره بنویسید .

فایل هایی که به سرعت برق اجرا می شوند

کد C که به وسیله اکثر کامپایلرها ایجاد می شود ، بسیار مؤثر و سریع است . ترکیب یک زبان کوچک ، یک سیستم زمان اجرای کوچک و این واقعیت که زبان به سخت افزار نزدیک است ، باعث می شود که بسیاری از برنامه های C با سرعتی نزدیک به معادل های اسمبلی خود اجرا شوند .

بررسی نوع محدود (Checking Limited Type)

برخلاف پاسکال که یک زبان با بررسی نوع شدید است ، C با انواع داده ها تا حدودی راحت تر رفتار می کند . این یک ویژگی بازمانده از زبان B است که آن هم از نظر بررسی نوع راحت تر بود . این راحتی و عدم سخت گیری به شما امکان می دهد تا داده ها را به روش های مختلفی مشاهده کنید . برای مثال ، در یک نقطه از برنامه ، برنامه ممکن است یک متغیر را به صورت کاراکتر ببیند و برای تبدیل به حروف بزرگ (با کم کردن 32 واحد از آن ، می تواند همین سلول حافظه را به صورت معادل اسکی کاراکتر ببیند .

پیاده سازی و طراحی به صورت بالا به پایین

زبان C حاوی تمام ساختارهای کنترلی است که از یک زبان امروزی انتظار دارید . این موضوع وقتی قابل توجه است که دوره شکل گیری C در سال 1971 را در نظر بگیرید که قبل از برنامه نویسی ساخت یافته رسمی بود . حلقه های for ، ساختارهای if و if-else ، دستور case/switch) و حلقه while ، همه در زبان گنجانده شدند . C همچنین با مدیریت محدوده کد و داده ها ، امکان تقسیم کردن کد و داده ها به چند قسمت را فراهم می کند . برای مثال ، زبان C متغیرهای محلی را برای این هدف و فراخوانی با مقدار را برای خصوصی سازی داده های زیر روال ارائه کرده است .

ساختار ماژولار

C از برنامه نویسی ماژولار پشتیبانی می کند که مفهوم آن کامپایل و Link کردن جداگانه است . این ویژگی به شما امکان می دهد که فقط بخش هایی از یک برنامه را کامپایل مجدد کنید که در طول توسعه تغییر کرده اند . وقتی که برنامه های بزرگ یا حتی برنامه های متوسط را روی سیستم های کند ایجاد می کنید ، این قابلیت می تواند فوق العاده مهم باشد . بدون پشتیبانی از برنامه نویسی ماژولار ، زمان مورد نیاز برای کامپایل یک برنامه کامل می تواند تغییر ، کامپایل و تست چرخه را بی نهایت کند نماید .

واسط پنهان برای زبان اسمبلی

یک روش خوب برای فراخوانی روتین های زبان اسمبلی در اکثر کامپایلرهای C وجود دارد . این روش به همراه جداسازی کامپایل و Link کردن ، زبان C را در برنامه هایی که نیاز به ترکیبی از روتین های سطح بالا و اسمبلی دارند رقیب بسیار قدرتمندی نموده است . روتین های زبان C در اکثر سیستم ها می توانند در زبان اسمبلی گنجانده شوند .

کار با بیت ها

اغلب در برنامه نویسی سیستمی ، لازم است که آبجکت ها را در سطح بیتی دستکاری کرد . به طور طبیعی ، چون اصل و ریشه C به طور نزدیکی به سیستم عامل یونیکس مرتبط است ، این زبان یک مجموعه غنی از عملگرها را برای کار با بیت ها فراهم کرده است .

نوع داده ای اشاره گر

یکی از قابلیت هایی که یک سیستم عامل از یک زبان می خواهد ، قابلیت آدرس دهی ناحیه های خاصی از حافظه است . این قابلیت همچنین سرعت اجرای برنامه را بالا می برد . زبان C ، با استفاده از اشاره گرها به این نیاز طراحی پاسخ گفته است . با اینکه زبان های دیگر هم اشاره گرها را پیاده سازی کرده اند ، زبان C به خاطر قابلیتش برای انجام عملیات ریاضی روی اشاره گرها معروف است . برای مثال ، اگر متغیر `student_record_ptr` به اولین عنصر از آرایه `student_records` اشاره کند ، `student_record_ptr + 1` آدرس عنصر دوم `student_records` خواهد بود .

ساختار های قابل گسترش

تمام آرایه ها در C یک بعدی هستند . آرایه های چند بعدی از ترکیب این آرایه های یک بعدی ساخته می شوند . آرایه ها و ساختارها (رکوردها) را میتوان به هر شکلی با هم ترکیب کرد و سازمان های پایگاه داده ای را ایجاد کرد که فقط توسط توانایی برنامه نویس محدودند .

صرفه جویی در حافظه

به خاطر بسیاری از دلایلی که برنامه های C سریع هستند ، از نظر مصرف حافظه هم خوبند . عدم وجود توابع نهادهین (built-in) می شود که لازم نباشد که برنامه توابعی را با خود داشته باشد که به وسیله آن برنامه مورد نیاز نباشد .

قابلیت حمل از یک سکو به سکوی دیگر

قابلیت حمل یعنی راحتی تبدیل یک برنامه که روی یک کامپیوتر یا سیستم عامل اجرا می شود به کامپیوتر یا سیستم دیگر . برنامه هایی که به زبان C نوشته می شوند یکی از قابل حمل ترین برنامه های دنیای پیشرفته کامپیوتر می باشند . این ویژگی به ویژه در دنیای مینی کامپیوتر و میکرو کامپیوتر صدق می کند .

روتین های کتابخانه ای قدرتمند

کتابخانه های توابع تجاری برای تمام کامپایلرهای محبوب C موجود است . کتابخانه هایی برای گرافیک ، مدیریت فایل ها ، پشتیبانی از پایگاه داده ها ، ایجاد پنجره ها در صفحه نمایش ، ورود داده ها ، ارتباطات و توابع پشتیبانی کلی وجود دارد . با استفاده از این کتابخانه ها می توانید به میزان زیادی در زمان ایجاد برنامه صرفه جویی کنید .

هیچ زبان برنامه نویسی بدون نقصی وجود ندارد. مسائل مختلف برنامه نویسی نیاز به راه حل های متفاوتی دارند و وظیفه مهندس نرم افزار این است که بهترین زبان را برای یک پروژه انتخاب کند. در هر پروژه ای، این موضوع یکی از اولین تصمیماتی است که باید بگیرید و وقتی که شروع به برنامه نویسی کردید این تقریباً چاره ناپذیر است. انتخاب زبان برنامه نویسی همچنین می تواند تفاوت بین موفقیت و عدم موفقیت پروژه را باعث شود. در زیر برخی از نقاط ضعف زبان C مطرح می شود تا به شما ایده بهتری دهد که برای یک برنامه چه وقت از زبان C استفاده کنید و چه وقت از زبان C استفاده نکنید.

بررسی نوع محدود!

این واقعیت که زبان C در مورد نوع داده زیاد سخت گیر نیست یکی از نقاط قوت آن است. ولی در عین حال یکی از نقاط ضعف آن نیز می باشد. از نظر تکنیکی، بررسی نوع یعنی اینکه یک زبان تا چه اندازه استفاده از انواع متغیرها را اجباری می کند (برای مثال، عدد صحیح و عدد با ممیز شناور دو نوع مختلف عددی هستند). در برخی زبان ها، غیر مجاز است که یک نوع داده ای را بدون فراخوانی یک تابع تبدیل به نوع داده ای دیگر منتسب کنید. این امر باعث می شود که داده ها توسط گرد شدن ناخواسته از دست نروند.

همان طور که قبلاً بحث شد، زبان C به یک عدد صحیح اجازه می دهد که به یک متغیر کاراکتری داده شود و برعکس. این برای شما به این معنی است که باید به درستی متغیرهای خود را مدیریت کنید. برای برنامه نویسان با تجربه، این موضوع مشکلی را ایجاد نمی کند. ولی، برنامه نویسان تازه کار باید به یاد داشته باشند که این امر می تواند باعث اثرات جانبی شود. اثر جانبی (effect side) در یک زبان، یک تغییر غیرمنتظره در یک متغیر یا آیتم دیگر است. از آنجایی که زبان C در مورد نوع داده زیاد سخت گیری نمی کند، برای کار با داده ها، انعطاف زیادی را در اختیار شما قرار می دهد. برای مثال، عملگر انتساب (=) می تواند بیش از یک بار در یک عبارت ظاهر شود. این انعطاف پذیری که می تواند از آن به نفع خود استفاده کنید، به این معنی است که می توان عبارت هایی نوشت که مقدار مشخص و روشنی نداشته باشند. محدود کردن استفاده از عملگر انتساب و عملگرهای مشابه یا حذف تمام اثرات جانبی و نتایج غیر قابل پیش بینی، مقدار زیادی از قدرت و محبوبیت زبان C به عنوان یک زبان اسمبلی سطح بالا را کم می کند.

نظارت محدود در زمان اجرا

عدم بررسی در سیستم زمان اجرای C می تواند باعث شود که بسیاری از مسائل پیچیده و گذرا تشخیص داده نشوند. برای مثال، اگر برنامه شما از محدوده های یک آرایه تجاوز کند سیستم زمان اجرا اخطار نمی دهد. این یکی از هزینه های کارآمد کردن یک کامپایلر به دلیل سرعت و کارایی است.

زبان C برای بچه ها نیست !

دامنه فوق العاده قابلیت های C ازکار با بیت ها تا I/O شکل گرفته به صورت سطح بالا و سازگاری نسبی آن از یک ماشین به ماشین دیگر به پذیرش آن در برنامه های علمی ، مهندسی و تجاری انجامیده است و این مستقیما به در دسترس بودن گسترده سیستم عامل یونیکس برروی تمام کامپیوترها از هر نوع و اندازه ای ، کمک کرده است .

با این حال ، زبان C مثل هر ابزار قدرتمند دیگری مسئولیت های سنگینی را برعهده استفاده کنندگان خود قرار می دهد . برنامه نویسان C باید به سرعت یک نظم خاص را به خود بگیرند و باید قواعد و قرارداد های مختلفی را اتخاذ کنند تا برنامه هایشان هم مدت ها بعد از اینکه نوشته شدند برای خودشان قابل فهم باشند و هم برای کسانی که سعی می کنند تا کد را برای اولین بار تجزیه و تحلیل کنند . در زبان C ، نظم برنامه نویسی ضروری است . نکته خوشحال کننده این است که این نظم تقریبا به صورت خودکار توسط تمرین به دست می آید .