

برنامه نویسی سیستم

بخش اول

```
1 if(parameters.contains(\n2   hql += " and p.name = :name\n3 }\n4\n5 if(parameters.contains("age")){\n6   hql += " and p.age = :age\n7 }\n8\n9 TypedQuery<Person> query = em.createQuery(\n10\n11\n12 if(parameters.contains("name")){\n13   query.setParameter("name", values[0]);\n14\n15 if(parameters.contains("age")){\n16   query.setParameter("age", Integer.valueOf(values[1]));\n17 }
```

دکتر اسماعیل صادقی

esmaeilsadeghi.com

فهرست

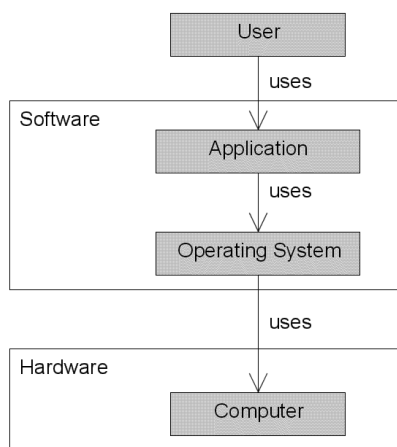
۱. سیستم کامپیوتر.....	۲
۲. معرفی رجیستری ویندوز.....	۲
۱-۲. دسترسی به رجیستری ویندوز.....	۳
۲-۲. ویرایش رجیستری ویندوز.....	۳
۳-۲. جستجو کلید در رجیستری ویندوز.....	۴
۴-۲. ایجاد یک مقدار در رجیستری ویندوز.....	۴
۵-۲. بکاپ گیری و بازگردانی بکاپ رجیستری ویندوز.....	۴
۳. برنامه‌نویسی سیستم.....	۶
۴. اهمیت برنامه نویسی سیستمی.....	۶
۵. تفاوت بین نرم افزار سیستم و نرم افزار کاربردی.....	۶
۱-۵. نرم افزار سیستم.....	۶
۲-۵. نرم افزار کاربردی.....	۷
۳-۵. تفاوت های نرم افزار سیستم در مقابل نرم افزار کاربردی.....	۷
۶. زبان برنامه نویسی سیستم.....	۹
۱-۶. زبان های برنامه نویسی کامپیوتر.....	۱۰
۱-۱-۶. زبان سطح پایین.....	۱۰
۶-۱-۱-۱. زبان اسمبلی.....	۱۲
۱-۲-۶. زبان های سطح بالا.....	۱۲
۷. سوالات.....	۱۴

۱. سیستم کامپیوتر

یک سیستم کامپیوتری شکل یکپارچه ای از اجزای مختلف است که با هم کار می کنند تا نتیجه مطلوبی به دست آورند. دارای اجزای مختلف است و هر کدام برای هدف خاصی کار می کنند. با این حال، آن ها یک نتیجه مشترک به عنوان مورد نیاز کاربر ایجاد می کنند. اجزای سیستم کامپیوتری عبارت است از:

۱. سخت افزار^۱: به اجزاء فیزیکی کامپیوتر گفته می شود که هر بخشی از کامپیوتره که میتونیم لمسشون کنیم. این اجزاء، دستگاه های الکترونیکی اصلی هستن که برای ساخت کامپیوتر استفاده گفته می شوند. پردازنده، دستگاه های حافظه، مانیتور، چاپگر، صفحه کلید، ماوس و واحد پردازش مرکزی نمونه هایی از سخت افزار کامپیوتر هستند.

۲. نرم افزار^۲: مجموعه ای از دستورالعمل ها، شیوه ها و مستنداتیه که وظایف متفاوتی توی سیستم کامپیوتر انجام میده. همچنین میتونیم بگیم که نرم افزار کامپیوتر، کدیه که برنامه ریزی شده و روی پردازنده کامپیوتر اجرا میشه. این کد هم میتونه در سطح ماشین باشه یا برای سیستم عامل بشه. برنامه های ورد، اکسل، پاورپوینت MS Word، Excel، Power Point، Google Chrome، Photoshop و MySQL و آنتی ویروس نمونه هایی از نرم افزارها هستن.



۲. معرفی رجیستری ویندوز

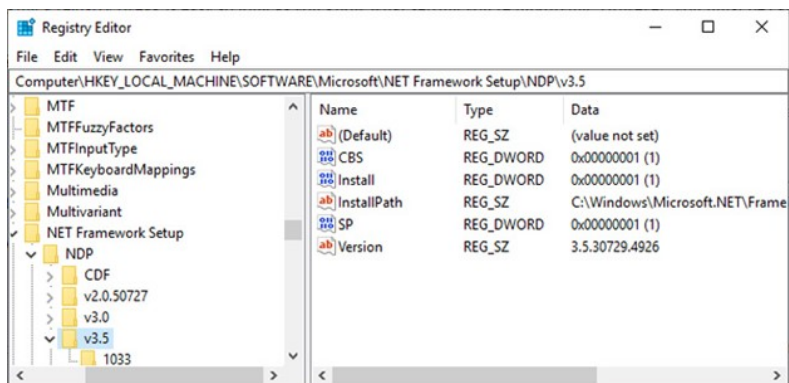
رجیستر به معنی ثبت کردن است و رجیستری محلی ثبت اطلاعات است. ویندوز اکثر تنظیمات مهم خود را درون رجیستری یادداشت می کند. رجیستری ویندوز به مانند اتاق فرمان یک استودیو است، اکثر اتفاقاتی که قرار باشد در ویندوز رخ دهد، ابتدا باید به تائید رجیستری برسد. از نصب یک نرم افزار گرفته تا باز و بسته شدن یک پورت و یا حتی تغییر برنامه پیش فرض برای یک نوع فرمت فایل مثل MP3.

در دنیای کامپیوتر، برای ثبت اطلاعات و تنظیمات باید اطلاعات را درون یک فایل یا دیتابیس ذخیره کنیم. لینوکس و ویندوز دو راهکار مختلف را انتخاب کردند. لینوکس روش ساده تر را برای خودش برگزیده و اطلاعات را درون فایل ها ذخیره می کند. ساده تر برای لینوکس و سخت تر برای کسی که می خواهد کار با آن ها را یاد بگیرد. ایراد اساسی این روش، گستردگی و غیرمتمرکز بودن فایل هاست و کاربر را مجبور می کند تا برای پیدا کردن هر فایل و روش تنظیمات

¹ hardware

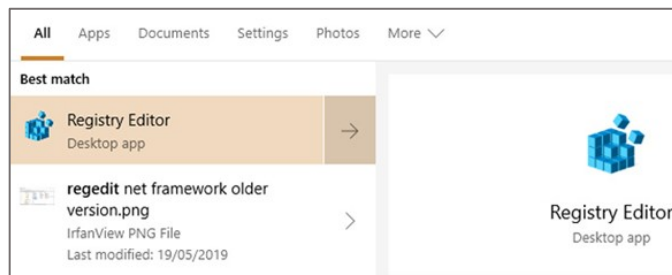
² Software

آن تخصص بالایی نیاز داشته باشد. مایکروسافت اما روش سخت‌تر را برای خودش برگزیده است ولی کار را برای کاربر نهایی ساده‌تر کرده است. مایکروسافت اطلاعات و تنظیمات ویندوز را داخل یک دیتابیس بسیار جالب و پیشرفته ذخیره می‌کند که امکانات بسیار فراوانی نیز دارد و رجیستری نام دارد. مایکروسافت رجیستری را بسیار هوشمندانه طراحی کرده است. شما هر کاری که در ویندوز انجام دهید بر روی رجیستری تأثیرگذار خواهد بود ولی شما اصلاً متوجه آن نخواهید شد چون همه اتفاقات در پشت‌صحنه اتفاق می‌افتد.



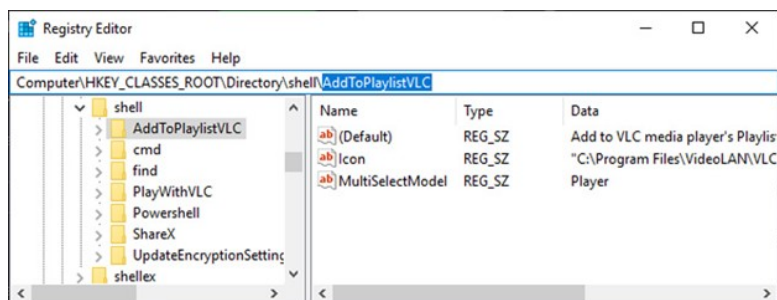
۲-۱. دسترسی به رجیستری ویندوز

برای ویرایش رجیستری ویندوز استارت ویندوز را باز کنید و عبارت regedit را تایپ کنید و پس از آن بر روی گزینه Registry Editor کلیک کنید تا صفحه رجیستری را مشاهده کنید.



۲-۲. ویرایش رجیستری ویندوز

در رجیستری ویندوز ما شاهد یک ساختار درختی هستیم که اگر دقیقاً بدانیم به دنبال تغییر چه تنظیماتی از ویندوز هستیم با استفاده از آن میتوانیم به راحتی به بخش مورد نظر خود دسترسی داشته باشیم. فرض کنید که من میخواهم vlc را به پوشه پلی لیست اضافه کنم به دنبال پوشه Add to VLC Playlist می‌گردم و من میدانم که این گزینه را در این مسیر میتوانم بیابم HKEY_CLASSES_ROOT\Directory\shell\AddToPlaylistVLC پس به راحتی بین نمودار درختی که میبینم حرکت میکنم و این پوشه را پیدا میکنم.

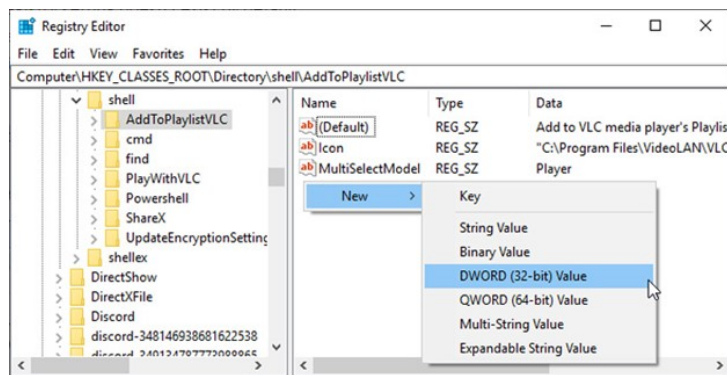


۳-۲. جستجو کلید در رجیستری ویندوز

ما می‌توانیم با استفاده از کلید کنترل و f بخش جستجوی رجیستری را باز کنیم و بین داده‌هایی که در این پایگاه داده کلید یا Key و یا مقدار Value و یا داده‌های مورد نظر خود را جستجو کنیم، برای اینکه بتوانیم یک داده‌ای از رجیستری را ویرایش کنیم بر روی این کلید کلیک راست می‌کنیم و گزینه‌ی Modify را انتخاب می‌کنیم تا تنظیمات مرتبط با کلید و یا داده مورد نظر برای ما باز شود.

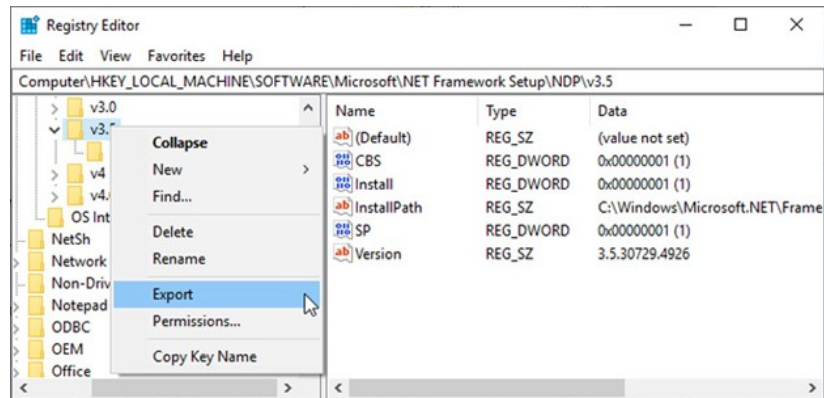
۴-۲. ایجاد یک مقدار در رجیستری ویندوز

برای ساخت یک کلید جدید یا همان Key ابتدا باید به مسیر پوشه مورد نظر برویم و سپس پوشه مورد نظر را باز کنیم سپس در سمت راست صفحه در جای خالی راست کلیک می‌کنیم و یک کلید جدید می‌سازیم پس از اینکه یک کلید در رجیستری ایجاد کردین لازم است مقدار یا Value ای برای آن در نظر بگیریم.

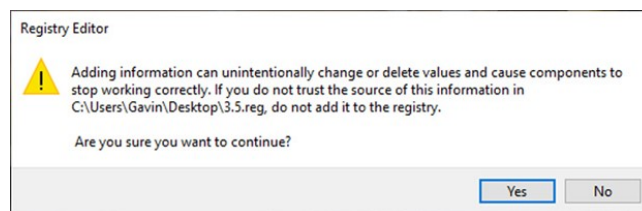


۵-۲. بکاپ‌گیری و بازگردانی بکاپ رجیستری ویندوز

یکی از کارهای مهم قبل از اعمال تغییرات در رجیستری ویندوز این است که شما یک بکاپ از رجیستری تهیه کنید تا اگر به هر دلیلی تغییرات اعمال شده آسیبی به ویندوز وارد کرد، بتوانید فایل بک آپ رجیستری ویندوز را بازگردانی کنید. بنابراین اگر قصد دارید در پوشه‌های رجیستری ویندوز تغییراتی را ایجاد کنید در اولین مرحله یک فایل reg درست کنید، شما می‌توانید بر روی هر کدام از پوشه‌های ویندوز کلیک راست کنید و گزینه Export را انتخاب کنید و این فایل را در مسیر دیگری ذخیره کنید تا در صورت نیاز بتوانید از آن استفاده کنید.

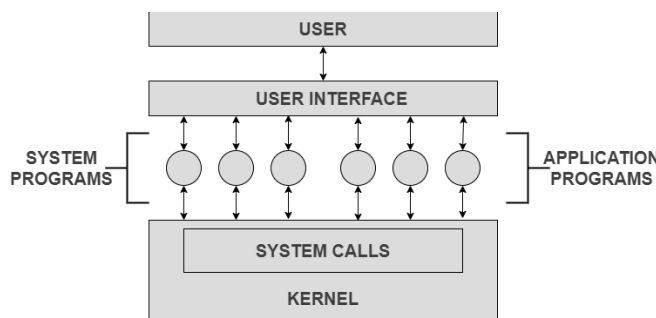


در مرحله قبل ما یک فایل بکاپ از رجیستری ویندوز خود تهیه کردیم در صورتی که بخواهید این بکاپ را بازگردانی کنیم تنها کاری که لازم است انجام دهیم این است که بر روی فایل ایجاد شده دابل کلیک کنیم و این فایل را اجرا کنید، به صورت خودکار مواردی که در این فایل ذخیره کرده بودین در رجیستری اعمال خواهد شد، این تغییرات معمولاً بعد از یک ریستارت کامپیوتر اعمال می شود.



۳. برنامه‌نویسی سیستم

برنامه‌نویسی سیستمی، نوعی زبان پیشرفته از طریق کامپیوتر می‌باشد که هدف آن تولید و تهیه برنامه‌های مرتبط با سخت افزار می‌باشد. تفاوت اصلی آن با دیگر زبان‌ها در این است که برنامه‌های نرم افزاری تولید شده با هدف تولید نرم افزارهای کاربردی تولید و نوشته می‌شوند، اما برنامه‌نویسی سیستمی با هدف تولید برنامه‌های مرتبط با سخت افزارها تولید می‌شود



۴. اهمیت برنامه‌نویسی سیستمی

مسائل ایجاد شده به صورت دسترسی در سطح پایین تعریف می‌شود. این به این معناست که روند اشکال زدایی آن سخت و زمان‌بر می‌باشد. در واقع این زبان برای تولید نرم افزار شما در محیط‌هایی که برنامه‌ها کامپایلر می‌کنند کار می‌کنند. و برای اینکه به اشکالات برنامه پی ببرید از یک برنامه واسط استفاده می‌کنید در صورتی که در برنامه‌نویسی سیستمی این چنین نیست و برنامه مستقیماً با سخت افزار در ارتباط است و فرایند اشکال زدایی آن کاری سخت و زمان‌بر می‌باشد. امروزه برنامه‌نویسی سیستمی برای تولید نرم افزارهای سخت افزارها استفاده می‌شود. یکی از گسترده‌ترین برنامه‌ها نوشته شده با استفاده از این زبان را می‌توان سیستم عامل‌ها نامید. سیستم عامل‌ها در واقع مانند پلی ارتباطی عمل می‌کنند که نرم افزارها بتوانند با سخت افزارها ارتباط برقرار کنند. همچنین سیستم عامل‌ها مانند ویندوز باعث می‌شود که منابع سخت افزاری به صورت ویژه و یا محدود و یا مناسب در اختیار نرم افزارها قرار گیرند و از آن استفاده کنند.

۵. تفاوت بین نرم افزار سیستم و نرم افزار کاربردی

نرم افزارهای سیستمی و نرم افزارهای کاربردی هر دو برنامه‌هایی هستند که به کاربران کمک می‌کنند تا با رایانه در تعامل باشند و به رایانه دستور دهند تا وظایف خاصی را انجام دهد. با این حال، تفاوت‌های قابل توجهی بین نرم افزار سیستم و نرم افزار کاربردی وجود دارد. هر دو از نظر طراحی و هدف متفاوت هستند.

۵-۱. نرم افزار سیستم

نرم افزار سیستم ترکیبی از برنامه‌هایی است که برای کنترل و مدیریت سخت افزار و سایر منابع سیستم طراحی شده است. چنین نرم افزاری به گسترش عملکردهای پردازشی سیستم کمک می‌کند و عملیات آن را سریع، موثر و ایمن می‌کند. علاوه بر این، نرم افزار سیستم نیز بستری را برای اجرای نرم افزارهای کاربردی فراهم می‌کند. نرم افزار سیستم عمده‌ترین کاربران را قادر می‌سازد تا با استفاده از دستورات مختلف به سیستم دستور دهند. با این حال، کاربران به طور کلی ارتباط مستقیمی با نرم افزار سیستم ندارند. یک رابط کاربری گرافیکی به نرم افزار سیستم

اضافه شده است تا کاربران بتوانند در صورت نیاز به راحتی با نرم افزار سیستم تعامل داشته باشند. به طور معمول، کاربران گزینه ای برای تعامل با نرم افزار سیستم اصلی ندارند. چنین نرم افزارهایی به طور خودکار در پس زمینه اجرا می شوند، عملیات سیستم را مدیریت می کنند و عملکرد کلی سیستم را متعادل می کنند. برخی از رایج ترین نمونه های نرم افزار سیستم عبارتند از: سیستم عامل (OS)، درایورهای دستگاه، BIOS، دیباگرها^۳، اسمبلرها^۴، کامپایلرها^۵، مترجمان^۶. همچنین ویژگی های اصلی نرم افزار سیستم به شرح زیر است:

۱. نرم افزار سیستم سرعت بالایی دارد.
۲. نرم افزار سیستم بیشتر به سیستم یا اجزای سخت افزاری نزدیک است.
۳. نرم افزار سیستم معمولاً در سطح پایین نوشته یا برنامه ریزی می شود زبان دستکاری نرم افزار سیستم بسیار سخت است زیرا چنین است طراحی و درک آن دشوار است.
۴. نرم افزار سیستم تعاملی کمتری دارد و از نظر اندازه کوچکتر است.

۵-۲. نرم افزار کاربردی

نرم افزار کاربردی یک برنامه یا مجموعه ای از برنامه ها است که در درجه اول به کاربران کمک می کند تا وظایف خاصی را با استفاده از یک سیستم کامپیوتری انجام دهند. این نرم افزارها به صورت جداگانه توسط کاربران دانلود و نصب می شوند و هیچ ارتباطی با عملکرد سیستم ندارند. بر خلاف نرم افزار سیستم، نرم افزارهای کاربردی بیشتر در قسمت جلو اجرا می شوند و برای کاربران قابل مشاهده باقی می ماند.

نرم افزار کاربردی فقط می تواند بر روی پلتفرم های ارائه شده توسط نرم افزار سیستم اجرا شود. می توان به نرم افزار کاربردی به عنوان واسطه بین کاربر نهایی و نرم افزار سیستم اشاره کرد. این نرم افزارها در درجه اول با استفاده از زبان های سطح بالا با توجه به نیازهای خاص کاربر نوشته می شوند. ما می توانیم چندین نرم افزار کاربردی را در یک سیستم کامپیوتری نصب کنیم. اگرچه نصب نرم افزارهای کاربردی بر روی سیستم اجباری نیست، اما سیستم کامپیوتری را برای کاربران مفیدتر می کند. برخی از رایج ترین نمونه های نرم افزار کاربردی عبارتند از مرورگرهای وب، میکروسافت آفیس، بازی ها، نرم افزارهای پخش رسانه، فتوشاپ. همچنین ویژگی های اصلی نرم افزار کاربردی به شرح زیر است:

۱. نرم افزار کاربردی به فضای ذخیره سازی بیشتری نیاز دارد زیرا عمدتاً چنین است
۲. در اندازه بزرگتر مثلاً نرم افزار فتوشاپ، بازی و
۳. نرم افزار کاربردی به فضای ذخیره سازی بیشتری نیاز دارد زیرا اندازه آن بیشتر است. مثل فتوشاپ، بازی و
۴. طراحی نرم افزار کاربردی نسبتاً آسان است.
۵. نرم افزار کاربردی برای کاربران تعاملی تر است.

۵-۳. تفاوت های نرم افزار سیستم در مقابل نرم افزار کاربردی

۱. نرم افزارهای سیستم برای مدیریت منابع سیستم مانند مدیریت فرآیند، مدیریت حافظه، امنیت و نگهداری و غیره طراحی شده اند. از سوی دیگر، نرم افزارهای کاربردی در درجه اول به گونه ای طراحی شده اند که به کاربران اجازه می دهند کارهای خاصی مانند فتوشاپ، پخش رسانه، بازی و غیره را انجام دهند.

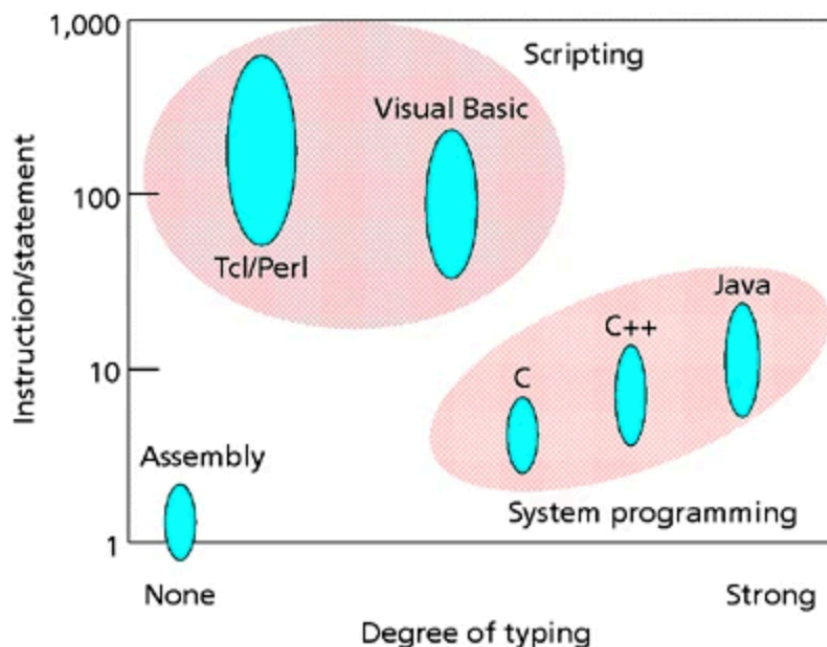
³ debuggers

⁴ assemblers

⁵ compilers

⁶ interpreters

۲. نرم افزارهای سیستمی معمولاً به زبانی سطح پایین نوشته یا برنامه ریزی می شوند، در مقابل، نرم افزارهای کاربردی بیشتر در سطح بالا نوشته می شوند



۳. نرم افزارهای سیستمی معمولاً به عنوان نرم افزارهای همه منظوره شناخته می شوند، در حالی که نرم افزارهای کاربردی معمولاً به عنوان نرم افزارهای هدف خاص شناخته می شوند.

۴. نرم افزار سیستم مستقل از نرم افزار کاربردی است و می تواند به طور مستقل اجرا شود. از سوی دیگر، نرم افزارهای کاربردی نمی توانند به طور مستقل اجرا شوند و برای اجرا به نرم افزار سیستم نیاز دارند.

۵. بدون نرم افزار سیستم، کامپیوتر نمی تواند بوت یا اجرا شود و نرم افزار سیستم را برای عملکرد سیستم اجباری می کند. علاوه بر این، نرم افزارهای کاربردی اجباری نیستند و سیستم می تواند بدون آنها کار کند.

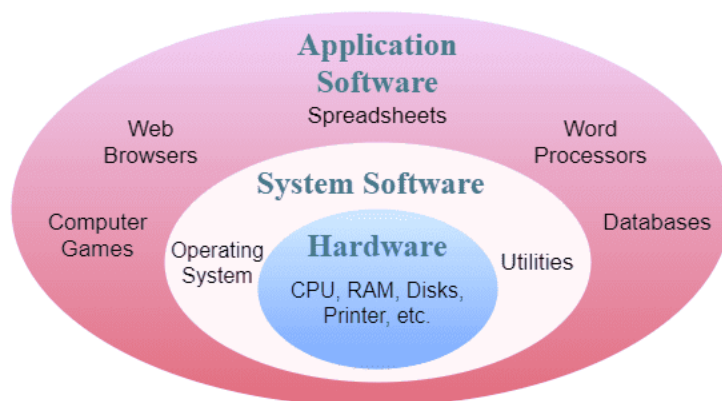
۶. نرم افزارهای سیستمی با نصب سیستم عامل نصب می شوند، در حالی که کاربران می توانند نرم افزارهای کاربردی را در صورتی که فقط برای اهداف خاصی مورد نیاز باشند دانلود و نصب کنند.

۷. وقتی سیستم کامپیوتر را روشن می کنیم نرم افزار سیستم در پس زمینه شروع به کار می کند و تا زمانی که سیستم خاموش شود به کار خود ادامه می دهد. علاوه بر این، نرم افزارهای کاربردی تنها زمانی شروع به اجرا می کنند که ما آنها را به طور خاص راه اندازی کنیم و پس از تکمیل کارها می توان آنها را خاموش کرد. در مورد نرم افزارهای کاربردی، نیازی به خاموش کردن سیستم کامپیوتری نیست.

۸. برنامه نویسی نرم افزار سیستم پیچیده است، در حالی که نرم افزار کاربردی را می توان به راحتی در مقایسه با نرم افزار سیستم برنامه ریزی کرد.

هدف نرم افزار سیستم و نرم افزار کاربردی کمک به کاربران در تعامل با سیستم های کامپیوتری و انجام وظایف مختلف است. با این حال، تفاوت هایی در طراحی و اهداف آنها وجود دارد. در حالی که نرم افزار سیستم برای مدیریت

منابع یا فرآیندهای سیستم طراحی شده است، نرم افزار کاربردی برای رفع نیازهای خاص کاربران برای انجام وظایف خاص طراحی شده است.

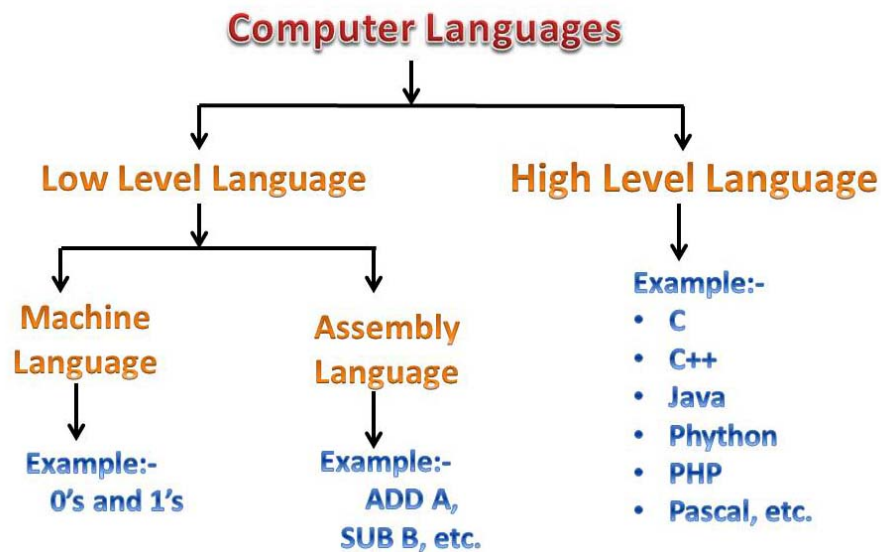


۶. زبان برنامه نویسی سیستم

سیستم ها از قطعات سخت افزاری و نرم افزاری ساخته شده اند. برنامه نویسی سیستم ها در مورد پیاده سازی این مؤلفه ها، رابط های آنها و معماری کلی است. اجزای منفرد وظایف تعیین شده خود را انجام می دهند و در عین حال با هم کار می کنند تا یک سیستم پایدار و کارآمد را تشکیل دهند. زبان اسمبلی و C در طول تاریخ برای برنامه نویسی سیستم ها استفاده می شده است. Go, Rust, Java, Swift و WebAssembly زبان های جدیدتری هستند که برای برنامه نویسی سیستمی مناسب هستند. در کل زبان برنامه نویسی سیستم یک زبان برنامه نویسی است که برای برنامه نویسی سیستم استفاده می شود. چنین زبان هایی برای نوشتن نرم افزارهای سیستمی طراحی شده اند که معمولاً در مقایسه با نرم افزارهای کاربردی به رویکردهای توسعه متفاوتی نیاز دارند. زبان های برنامه نویسی همه منظوره تمایل دارند بر روی ویژگی های عمومی تمرکز کنند تا به برنامه های نوشته شده در این زبان اجازه دهند از همان کد در پلتفرم های مختلف استفاده کنند. نمونه هایی از این زبان ها عبارتند از پاسکال. این کیفیت عمومی معمولاً به قیمت ممانعت از دسترسی مستقیم به عملکرد داخلی دستگاه است و این اغلب تأثیرات منفی بر عملکرد دارد. در مقابل، زبان های سیستم نه برای سازگاری، بلکه برای عملکرد و سهولت دسترسی به سخت افزار زیربنایی طراحی شده اند و در عین حال مفاهیم برنامه نویسی سطح بالا مانند برنامه نویسی ساختاریافته را ارائه می دهند.

۱-۶. زبان های برنامه نویسی کامپیوتر

برنامه های کامپیوتری که به عنوان نرم افزار شناخته می شوند، دستورالعمل هایی هستند که به کامپیوتر می گویند چه کاری انجام دهد.



رایانه ها را برنامه ریزی می کنیم تا وظایف خاصی را انجام دهند و به آن ها آموزش می دهند که هر زمان که ورودی از نوع (های) از پیش تعریف شده دریافت می کنند، طبق مجموعه ای از قوانین (الگوریتم ها) عمل کنند تا خروجی مورد انتظار را دریافت کنند. برای همه این اهداف ما از زبان های برنامه نویسی استفاده می کنیم. همچنین، یک زبان برنامه نویسی مجموعه ای از دستورالعمل ها را تعریف می کند که برای انجام یک کار خاص توسط CPU (واحد پردازش مرکزی) با هم کامپایل می شوند. هر زبان برنامه نویسی شامل مجموعه ای منحصر به فرد از کلمات کلیدی و نحو است که برای ایجاد مجموعه ای از دستورالعمل ها استفاده می شود. هزاران زبان برنامه نویسی تاکنون توسعه یافته اند، اما هر زبانی هدف خاص خود را دارد. این زبان ها در سطح انتزاعی که از سخت افزار ارائه می کنند متفاوت هستند. برخی از زبان های برنامه نویسی انتزاع کمتر یا بدون انتزاع ارائه می دهند در حالی که برخی از آنها انتزاع بالاتری ارائه می دهند. بر اساس سطوح انتزاع می توان آنها را به دو دسته کلی تقسیم کرد:

۱. زبان سطح پایین^۷

۲. زبان سطح بالا^۸

۱-۱-۶. زبان سطح پایین

زبان سطح پایین یک زبان برنامه نویسی است که هیچ انتزاعی از سخت افزار ارائه نمی دهد و به شکل ۰ یا ۱ نشان داده می شود که دستورالعمل های ماشین هستند. زبان های ماشین را به عنوان زبان مادری کامپیوتر، نزدیک ترین زبان به خود سخت افزار تصور کنید. هر کامپیوتر منحصر به فرد دارای یک زبان ماشین منحصر به فرد است. یک برنامه زبان ماشین از یک سری الگوهای

^۷ low level

^۸ high level

باینری (مثلاً ۰۱۰۱۱۱۰۰) تشکیل شده است که عملیات ساده ای را نشان می دهد که می تواند توسط رایانه انجام شود (به عنوان مثال، اضافه کردن دو عملوند، انتقال داده ها به یک مکان حافظه). برنامه های زبان ماشین قابل اجرا هستند، به این معنی که می توانند مستقیماً اجرا شوند. برنامه نویسی به زبان ماشین نیاز به حفظ کدهای باینری دارد و می تواند برای برنامه نویسی انسانی دشوار باشد. مثالی از زبان ماشین در زیر وجود دارد.

مثلاً لامپ به گونه ای سیم کشی شده است که هنگامی که سوئیچ برای اولین بار روشن می شود دستورالعمل در آدرس صفر انجام می شود. سپس دستورات به ترتیب یکی یکی انجام می شود تا زمانی که با دستور "Halt" مواجه شود یا کنترلر از کار بیفتد. اگر کنترل کننده با دستورالعملی روبرو شود که در جدول نیست یا سعی کند دستورالعملی را از آدرسی که وجود ندارد دریافت کند، خراب می شود. دستورالعمل "Skip" روشنایی لامپ را بررسی می کند و احتمالاً از دستورالعمل بعد از آن عبور می کند. دستور "Go back" باعث می شود که پردازنده دوباره با دستورالعمل آدرس ۰ شروع به کار کند.



برای شروع یا توقف برنامه روی لامپ کلیک کنید

عملکرد ماشین	دستورالعمل ماشین
توقف برنامه	00000000
لامپ را کاملاً روشن کنید	00000100
لامپ کم نور با ۱۰٪	00001000
اگر لامپ کاملاً روشن است، از دستورالعمل بعدی رد شوید	00010000
اگر لامپ کاملاً خاموش است، از دستورالعمل بعدی رد شوید	00100000
برو به شروع برنامه (آدرس ۰)	01000000

در اینجا حافظه اصلی برای کنترلر است:

دستورالعمل ماشین	آدرس
00000001	0
00000010	1
00000001	2
00000010	3
00000001	4
00000100	5
00000100	6
00000100	7
00000100	8
00000000	9
00000000	10

00000000	11
00000000	12

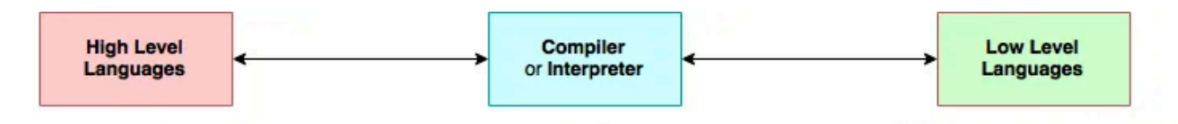
۶-۱-۱. زبان اسمبلی

زبان اسمبلی نشان دهنده تلاشی برای آسان کردن برنامه نویسی برای انسان هستند. دستورالعمل های زبان ماشین با اختصارات پنومونی^۹ ساده (مانند ADD، MOV) جایگزین می شوند. بنابراین زبان های اسمبلی برای یک کامپیوتر خاص (ماشین) منحصر به فرد هستند. قبل از اجرا، یک برنامه زبان اسمبلی نیاز به ترجمه به زبان ماشین دارد. این ترجمه توسط یک برنامه کامپیوتری معروف به اسمبلر انجام می شود. اسمبلرها برای هر زبان ماشین منحصر به فرد نوشته شده اند. به عنوان مثال عدد ۴۲ را از عدد ۵۳ کم کنید:

Number 53	00110101
Number 42	00101010
Reverse the bits of 42	11010101
Add 1	00000001
Number -42	11010110
$53 - 42 = 11$	00001011

۶-۱-۲. زبان های سطح بالا

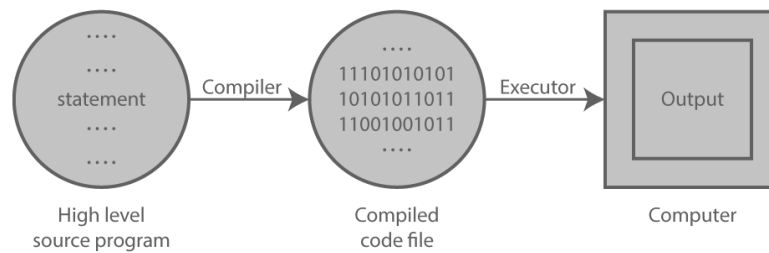
زبان های سطح بالا، مانند C، C++، JAVA و غیره، بیشتر شبیه انگلیسی هستند و بنابراین، فکر کردن در زبان برنامه نویسی را برای برنامه نویسان آسان تر می کنند. زبان های سطح بالا نیز قبل از اجرا نیاز به ترجمه به زبان ماشین دارند. این ترجمه توسط یک کامپایلر یا یک مترجم انجام می شود. کامپایلرها کل برنامه کد منبع را قبل از اجرا ترجمه می کنند. همچنین مفسرها تعامل بیشتری نسبت به کامپایلرها دارند. از مزایای زبان های سطح بالا می توان سهولت در برنامه نویسی و قابلیت حمل بودن را نام ب



ابزارهای اجرای برنامه های نوشته شده در زبان های برنامه سازی سطح بالا: کامپایلر و مفسر

^۹ mnemonic

کامپایلر^{۱۰} نرم افزاری برای تبدیل کد منبع^{۱۱} به کد شی^{۱۲} است. به عبارت دیگر می توان گفت که کامپایلر کدهای نوشته شده به زبان سطح بالا (نزدیک به زبان انسان) توسط برنامه نویسان را به زبان دودویی ماشین تبدیل می کند.



کامپایلر^{۱۳} نرم افزاری برای تبدیل کد منبع^{۱۴} به کد شی^{۱۵} است. به عبارت دیگر می توان گفت که کامپایلر کدهای نوشته شده به زبان سطح بالا (نزدیک به زبان انسان) توسط برنامه نویسان را به زبان دودویی ماشین تبدیل می کند.

¹⁰ compiler
¹¹ source Code
¹² object Code
¹³ compiler
¹⁴ source code
¹⁵ object code

۷. سوالات

۱. سیستم کامپیوتری را شرح دهید.

جواب: یک سیستم کامپیوتری می تواند پردازش انجام دهد، از اجزای الکترونیکی تشکیل شده است، دارای حافظه و قابل برنامه ریزی می باشد

۲. بخش های سیستم کامپیوتری را نام ببرید.

جواب: نرم افزار و سخت افزار

۴. تفاوت های نرم افزار سیستم در مقابل نرم افزار کاربردی چیست.

جواب: مراجعه به متن.

۳. زبان را تعریف کنید.

جواب: مجموعه ای از علائم و نشانه ها که در کنار هم قرار می گیرند و برای ارتباط ارتباط استفاده می شود. بسیاری از زبانها تعدادی قالب از ویژگی های نوشته شده دستوری یا نحو (syntax) و معناشناسی (semantics) دارند، چرا که رایانه ها دستورهای دقیقاً مشخص نیاز دارند.

۴. زبان های برنامه نویسی را تعریف کنید.

جواب: یک زبان مصنوعی انگلیسی است که برای برقراری ارتباط با یک ماشین (مخصوصاً رایانه) استفاده می شود.

۵. تفاوت کامپایلر و مفسر چیست.

جواب:

- در زبان های کامپایلری ابتدا یک بار کدها به صورت کامل ترجمه می گردند. سپس برنامه برای اجرا از روی فایل کامپایل شده اجرا می گردد. اما در زبان های مفسری کدها به صورت خط به خط تفسیر و اجرا می شوند. سپس هر دفعه که بخواهیم برنامه را دوباره اجرا کنیم به همین صورت کد تفسیر و اجرا می شود.
- زبان مفسری برای اجرا نیاز به یک نرم افزار مفسر ۱۶ دارد اما زبان کامپایلری تنها یک بار کامپایل شده و کدهای قابل اجرا در ماشین را ایجاد می کند. از آن پس بر روی هر سیستمی بدون نیاز به کامپایلر قابل اجرا می باشد.
- زبان های مفسری کدها را به صورت خط به خط تفسیر و اجرا می کنند، این کار باعث می شود تا مقدار زیادی از CPU و RAM کامپیوتر اشغال شود و مشخص است که سرعت پایین تری نیز خواهند داشت. اما کامپایل شدن یکباره ی زبان های کامپایلر این است که در اجراهای متعدد سبک تر بوده و سرعت بهتری خواهند داشت.
- همین مزیت به صورت یک جا کامپایل شدن زبان های کامپایلر در بخش خطایابی و رفع مشکل، نقص به حساب می آید. زیرا در زبان های مفسر هنگامی که برنامه خط به خط اجرا می شود، در هر بخشی از برنامه که مشکلی وجود داشته باشد، قابل پیدا کردن و رفع می باشد. اما در زبان های کامپایلری برای رفع یک مشکل در کدها می بایست سورس خود برنامه را مجدد بازبینی کرده، سپس خطا را پیدا و تصحیح کرد که این کار بسیار دشوارتر از کار با زبان های مفسری است.

¹⁶ interpreter

۶. برنامه نویسی سیستم را تعریف کنید.

جواب: برنامه نویسی سیستمی نوعی زبان پیشرفته از طریق رایانه است که با هدف تولید برنامه های مرتبط با سخت افزار تولید و تهیه می شوند. تفاوت اصلی آن با دیگر زبان ها در این است که زبان های عادی برای تولید نرم افزار به کار می روند ولی برنامه نویسی سیستم برای تولید سیستم عامل به کار می رود.

۷. انواع سطوح زبان های برنامه نویسی را نام ببرید.

جواب: زبان های برنامه نویسی سطح پایین و سطح بالا

۸. انواع زبان های برنامه نویسی سطح پایین را نام ببرید.

جواب: زبان های برنامه نویسی ماشین و اسمبلی

۹. زبان برنامه نویسی اسمبلی را توضیح دهید.

جواب: زبان اسمبلی یک زبان برنامه نویسی سطح پایین است و دستورهای این زبان به کد ماشین و در نتیجه، به سخت افزار بسیار نزدیک هستند. هر زبان اسمبلی مختص به یک معماری رایانه خاص است، برخلاف بیشتر زبان های برنامه نویسی سطح بالا که با معماری های مختلف سازگار هستند ولی برای اجرا شدن به مفسر و مترجم نیاز دارند. زبان اسمبلی با استفاده از یک برنامه جانبی به نام اسمبلر یا همگذار، به کد ماشین قابل اجرا تبدیل می شود، به این فرایند تبدیل، اسمبل کردن گفته می شود. مثال: مراجعه به متن.

۱۰. دستور جمع را برای دو عدد ۱۲ و ۵ در زبان برنامه نویسی اسمبلی بنویسید.

جواب: دستورالعمل زیر حاصل جمع صحیح دو عملوند خود را محاسبه و نتیجه را در عملوند اول قرار می دهد.

add dest, src

دستورالعمل add محتوای عملوند src را با عملوند dest جمع می کند و نتیجه را در dest ذخیره می کند

dest := dest + src

پس کرکتر ۱۲ و ۵ را جایگزین محتوی قبلی ثبات eax و ebx می شود و برای جمع دو عدد به شکل زیر عمل می کنیم:

mov eax, 12

mov ebx, 5

add eax, ebx

۱۱. رجیستری در ویندوز چیست؟

جواب: ویندوز رجیستری یک پایگاه داده است که طیف گسترده ای از تنظیمات پیکربندی را ذخیره می کند. تقریباً همه تنظیمات پیکربندی در ویندوز در آن ذخیره می شوند.

۱۲. ویرایش رجیستری ویندوز چگونه است؟

جواب: ویندوز برای ویرایش رجیستری می توانید از طریق ساختار درختی پوشه های رجیستری ویندوز به آن دسترسی پیدا کنید. برای باز کردن شاخه های زیرمجموعه هر پوشه کافی است روی پیکان مقابل هر کدام از آنها کلیک کنید.

بعنوان مثال، اگر می خواهید گزینه Add to VLC Playlist را به منوی مربوط به کلیک راست برنامه اضافه کنید، باید مسیر

HKEY_CLASSES_ROOT\Directory\shell\AddToPlaylistVLC را دنبال کنید. در صورت نیاز به راهنمایی بیشتر

ویرایش رجیستر می توانید از خدمات مشاوره کامپیوتری بهره مند باشید.

۱۳. وارد شدن به بخش رجیستری ویندوز چگونه است؟

جواب: برای وارد شدن به رجیستری ویندوز کافی است عبارت regedit را در run ویندوز وارد کنید (با فشردن کلیدهای پنجره + R محیط Run ویندوز اجرا می‌شود) همچنین می‌توانید عبارت registry editor را در قسمت جستجوی ویندوز تایپ کنید تا وارد این بخش شوید.