حافظه کنترل

یک واحد کنترل ریزبرنامه‌نویسی شده در کامپیوتر معمولا دارای دو حافظه مجزا است که به عنوان حافظه اصلی و حافظه کنترل شناخته می‌شوند. حافظه کنترل نوعی از حافظه دسترسی تصادفی RAM است.

حافظه کنترل شامل ثبات‌های ذخیره‌سازی قابل آدرس‌دهی می باشد. داده‌ها در حافظه کنترل به صورت موقت ذخیره می‌شوند. سرعت دسترسی به حافظه کنترل نسبت به حافظه اصلی بیشتر است که این مسئله باعث افزایش سرعت پردازنده مرکزی می‌شود.

توالی آدرس

ریزدستورالعمل‌ها در حافظه کنترل به صورت گروهی ذخیره می‌شوند. هر گروه، مشخص کننده یک روال ((است. هر دستورالعمل کامپیوتر، روال ریز برنامه مربوط به خودش را برای تولید ریزعمل‌ها دارد. سخت‌افزاری که توالی آدرس (| Address Sequencingترتیب‌دهی آدرس | دنبال کردن آدرس) را در حافظه کنترل مدیریت می‌کند، باید قابلیت توالی ریز عملیات را در یک روال داشته باشد و بتواند از یک روال به روال دیگر انشعاب انجام دهد. بنابراین، قابلیت انشعاب از یک روال به روال دیگر در حافظه را ترتیب‌دهی یا توالی آدرس می‌نامند. به طور خلاصه، قابلیت‌های مورد نیاز برای توالی آدرس در یک حافظه کنترل به شرح زیر است:

افزایش ثبات آدرس کنترل

انشعاب غیرشرطی یا انشعاب شرطی بسته به شروط بیت‌های وضعیت یک پروسه نگاشت (Mapping) از بیت‌های دستورالعمل به یک آدرس برای حافظه کنترل امکانی برای فراخوانی و بازگرداندن زیر روال از جمله موضوعاتی که در حیطه توالی آدرس در درس معماری کامپیوتر و مبحث کنترل ریز برنامه‌نویسی شده مطرح می‌شود، می‌توان به انشعاب شرطی، نگاشت دستورالعمل و زیر روال‌ها اشاره کرد.

برنامه نویسی کامپیوتر پایه

یک رایانه کامل هم سخت‌افزار دارد و هم دارای نرم‌افزار است. سخت‌افزار از قطعات فیزیکی و تمامی تجهیزات مربوط به آن تشکیل شده است. نرم‌افزار به برنامه‌هایی مربوط می‌شود که برای کامپیوتر نوشته شده‌اند. آن دسته از افرادی که با معماری کامپیوتر در ارتباط هستند، باید هم در مورد سخت‌افزار و هم نرم‌افزار دانش و آگاهی کافی را داشته باشند.

دستورالعمل‌های داخل کامپیوتر یک الگوی باینری تشکیل می‌دهند که درک آن برای افراد اگر غیرممکن نباشد، می‌توان گفت بسیار دشوار است. بنابراین، بهتر است نمادهای برنامه‌ها شباهت بیش‌تری به مجموعه علائم حروف و اعداد داشته باشند. در نتیجه، نیاز به ترجمه برنامه‌های نوشته شده با نمادهای کاربر محور به برنامه‌های باینری است که برای سخت‌افزار قابل تفسیر باشد. یک برنامه نوشته شده توسط کاربر ممکن است به کامپیوتر فیزیکی وابسته و یا مستقل از آن باشد.

زبان ماشین

یک برنامه، لیستی است از دستورالعمل‌ها یا عبارات که برای هدایت کامپیوتر جهت اجرای عملیات پردازش روی داده‌ها نوشته می‌شود. هر برنامه‌ای، با هر زبانی که نوشته شده باشد، قبل از این‌که توسط کامپیوتر قابل اجرا باشد باید به نمایش دودویی دستورالعمل‌ها ترجمه شود.

برنامه‌های نوشته شده برای یک کامپیوتر در یکی از دسته‌بندی‌های زیر قرار می‌گیرند:

کد باینری: نمایش عینی دستورالعمل‌ها وقتی در حافظه کامپیوتر ظاهر می‌شوند.

کد مبنای هشت یا مبنای ۱۶: در واقع معادل ترجمه شده کد باینری به مبنای هشت یا ۱۶ هستند.

کد نمادی (Symbolic Code) کاربر نمادهایی (حروف، اعداد یا کاراکترهای خاص) را برای بخش عملیات، بخش آدرس و سایر بخش‌ها به کار می‌گیرد. ترجمه این نمادها به باینری به وسیله اسمبلر انجام می‌شود. بنابراین این نوع برنامه نمادگذاری شده را یک برنامه زبان اسمبلی می‌نامند.

زبان برنامه نویسی سطح بالا: این سطح شامل زبان‌های مخصوصی است که به جای درگیری با رفتار سخت‌افزار، برای بازتاب دستورالعمل‌هایی به کار می‌روند که در راه حل یک مسئله استفاده می‌شوند.

زبان اسمبلی

یک زبان برنامه‌نویسی به وسیله مجموعه‌ای از قوانین تعریف می‌شود. کاربران باید تمام قالب قوانین آن زبان را رعایت کنند تا برنامه‌های آن‌ها به درستی ترجمه شوند. همان‌طور که در بخش زبان ماشین توضیح داده شد، زبان اسمبلی سومین سطح در برنامه‌نویسی زبان‌های ماشین است.

واحد بنیادی یک برنامه به زبان اسمبلی، یک خط کد است. یک زبان خاص به وسیله مجموعه قوانینی مشخص می‌شود که نمادهای مورد استفاده و نحوه ترکیب آن‌ها برای تشکیل یک خط کد را تعیین می‌کنند. از جمله مباحثی که در این بخش مطرح می‌شود، می‌توان به قوانین زبان اسمبلی، آدرس نمادین، شبه‌دستورالعمل، ترجمه به دودویی و جدول نماد آدرس اشاره کرد.

اسمبلر

اسمبلر برنامه‌ای است که یک زبان برنامه‌نویسی نمادین را دریافت و زبان ماشین متناظر دودویی آن را تولید می‌کند. زبان نمادین ورودی «برنامه منبع» نامیده می‌شود و به برنامه باینری حاصل شده نیز «برنامه هدف» گفته می‌شود. از جمله مطالبی که در خصوص اسمبلر اهمیت دارد نمایش برنامه نمادین در حافظه، گذر اول، شمارنده موقعیت ، گذر دوم و تشخیص خطا است.

برنامه‌نویسی عملیات ریاضی و منطقی

تعداد دستورالعمل‌های قابل دسترسی در یک کامپیوتر ممکن است در یک سیستم بزرگ به چندصدتا و در یک سیستم کوچک به چند ده‌تا برسد. برخی از کامپیوترها، یک عملیات داده شده را فقط با یک دستورالعمل ماشین اجرا می‌کنند؛ سایر کامپیوترها ممکن است به تعداد دستورالعمل‌های ماشین بیش‌تری برای اجرای همان عملیات نیاز داشته باشند. عملیاتی که در مجموعه دستورالعمل‌های ماشین وجود ندارند باید به وسیله یک برنامه پیاده‌سازی شوند.

زیر روال

زیر روال(رویه) یک مجموعه از دستورالعمل‌ها است که به دفعات در یک برنامه مورد استفاده قرار می‌گیرد. تنها یک نسخه از دستورات زیر روال در حافظه ذخیره می‌شود. یک زیرروال را می‌توان به تعداد مورد نیاز در طول اجرای یک برنامه خاص فراخوانی کرد. باید در زمان بازگرداندن یک زیر روال احتیاط شود؛ چرا که یک زیر روال می‌تواند از محل دیگری در حافظه فراخوانی شود. محتویات کامپیوتر باید به وسیله دستورالعمل فراخوانی زیر روال ذخیره شود تا بازگشت به برنامه‌ای که زیر روال را فراخوانی کرده به درستی انجام شود.

روش پیوند زیر روال شیوه‌ای است که در آن کامپیوتر زیر روال را فراخوانی و بازگردانی می‌کند. ساده‌ترین راه برای پیوند زیر روال، ذخیره آدرس بازگشت در محل مشخصی مثل یک ثبات است که می‌تواند به عنوان زیر روال فراخوانی ثبات پیوند فراخوانده شود.

برنامه نویسی ورودی-خروجی

بخش مهمی از زبان برنامه‌نویسی اسمبلی با ورود و خروج اطلاعات به/از دستگاه‌های جانبی سر و کار دارد. روش‌های مختلف برنامه‌نویسی معرفی شده، در اصل با جنبه‌های محاسباتی سر و کار دارند، اما این ابعاد بدون امکان ورود و خروج داده‌ها غیر قابل استفاده هستند.

کنترل ریز برنامه نویسی شده

واحد کنترل، سیگنال‌های زمان‌بندی و کنترلی را برای عملیات در کامپیوتر تولید می‌کند. واحد کنترل با ALU و حافظه اصلی ارتباط برقرار می‌کند. همچنین، واحد کنترل نقل و انتقال میان پردازنده، حافظه و دستگاه‌های ورودی-خروجی مختلف را مدیریت می‌کند. علاوه بر این، صدور دستوراتی به ALU برای تعیین عملیاتی که باید روی داده‌ها انجام شود نیز از جمله وظایف واحد کنترل است.

برای اجرای یک دستورالعمل، واحد کنترل پردازنده مرکزی باید سیگنال کنترلی لازم را در ترتیب صحیح تولید کند. دو رویکرد برای تولید سیگنال‌های کنترلی وجود دارد. کنترل ریز برنامه نویسی شده در نقطه مقابل واحد کنترل سخت‌افزاری قرار می‌گیرد.

سخت‌افزار کنترل را می‌توان به عنوان یک ماشین وضعیت تصور کرد که در هر چرخه ساعت، از یک وضعیت به وضعیت دیگری تغییر حالت می‌دهد. این تغییر وضعیت به محتویات ثبات دستورالعمل، کدهای وضعیت و ورودی‌ها بستگی دارد. از جمله موضوعات مهم در مورد کنترل ریز برنامه‌نویسی شده می‌توان به حافظه کنترل و ترتیب‌دهی آدرس اشاره کرد.