معرفی کلمه کلیدی **Voaltile**

گاهی در برنامه هایی که برای سیستم های نهان توسعه داده می شود، یکی از حالات زیر رخ می دهد:

1. برنامه یا کد نوشته شده با فعال کردن گزینه ی بهینه سازی کامپایلر ، دیگر پاسخ درستی نمی دهد .
2. برنامه یا کد نوشته شده با فعال شدن وقفه ، دیگر پاسخ درستی نمی دهد .
3. سیستم عامل در حالت ایزوله به درستی کار می کند ، اما با فعال شدن وظیفه ای به خصوص پاسخ مناسبی مشاهده نمی شود.

اگر یکی از حالات فوق رخ داده باشد، دلیل آن عدم استفاده از کلمه کلیدی volatile است.

متغیرهایی از نوع  volatileمتغیرهایی هستند که ممکن است مقدار آنها توسط یک پردازش خارجی تغییر یابد. این پردازش می‌تواند وقوع یک وقفه، یا تغییر از طریق یک پردازش موازی باشد. کامپایلرهای زبان برنامه‌نویسیC++  اغلب از روش‌های بهینه‌سازی برای دست‌یابی به متغیرهای تعریف شده در طول برنامه استفاده می‌کنند. به عنوان مثال، آخرین تغییر هر متغیر در طول پردازش را به عنوان مقدار آن متغیر در نظر گرفته و اگر دستوری مقدار متغیر را تغییر ندهد، همان مقدار نهایی را در پردازش مدنظر قرار می‌دهد. اما متغیرهایی از نوع volatile متغیرهایی هستند که لزوما از طریق پردازش جاری تغییر نمی‌کنند. بنابراین ممکن است مقدار آنها توسط پردازش دیگر یا یک وقفه عوض شود، بدون آنکه کامپایلر متوجه باشد. برای جلوگیری از چنین خطایی، کلمه‌ی کلیدی volatile در ابتدای تعریف متغیر استفاده می‌شود، تا کامپایلر عملیات بهینه‌سازی روی آن انجام نداده و در هر بار فراخوانی اسم متغیر در طول برنامه، مقدار آن را از محل حافظه بخواند.

با یک متغیر معمولی، گام‌های برنامه ما تنها چیزهایی هستند که آن را تغییر می‌دهیم، بنابراین برای مثال اگر ۵ را از یک متغیر بخوانیم و آن را تغییر ندهیم، هنوز مقدار متغیر ما 5 است. از آنجا که کامپایلر ما هوشمند است ، زمانی که می‌خواهیم از متغیر مجددا استفاده کنیم ، لازم نیست که برنامه دوباره برگردد بلکه همان 5 را به خاطر می سپارد و به یاد می آورد. اما ممکن است سیستم اطلاعات را از دیسک به حافظه منتقل ‌کند و اشتباها و صرفا جهت بهینه سازی یا بعضی موارد آتی ، متغیر ما را به ۵۰۰ تغییر ‌دهد. بنابراین ما باید به کامپایلر بگوییم که هر بار ، مقدار متغیر را دوباره چک کند .این کاری است که کلمه‌ی کلیدی volatile انجام میدهد.

به نوعی میتوانیم بگوییم که استفاده از volatile مثل یکpragma برای غیر فعال کردن بهینه‌سازی کامپایلر است ولی اگر بخواهیم خیلی واضح تر بگوییم ، volatile به کامپایلر ما اعلام می‌کند که :

"مقدار متغیر ما ممکنه خارج از روال عادی برنامه -که تو ازش خبر داری- تغییر کنه، پس حواستو جمع کن و هر وقت هم قرار شد ازش استفاده کنی یک بار مقدارش رو بخون - فرض رو بر این بگیر که مقدارش تغییر کرده.-"

در این صورت کامپایلر هرجا این متغیر را می‌بیند بر رویش بهینه سازی‌ انجام نمی‌دهد و اینطور در نظر می‌گیرد که، از جایی دیگر ممکن است مقدارش را تغییر داده باشند.

به طور کلی در کد ها و برنامه هایی که برای سیستم ها نهان توسعه داده می شوند ، بایستی متغیر ها مرتبط با موارد زیر از نوع volatile تعریف شود:

1. استفاده از رجیستر های جانبی در نگاشت حافظه .
2. استفاده از متغیر های عمومی که توسط زیر برنامه های وقفه تغییر می کنند.
3. استفاده از متغیر های عمومی که در دسترس سیستم های چند وظیفه ای با چند هسته ی پردازنده هستند.

نکته قابل توجه اینکه ، دستور volatile هیچ فضایی از حافضه اشغال نمی کند، بلکه نحوه ترجمه برنامه نوشته شده توسط کمپایلر را اصلاح و تغییر می دهد.

امیرمهدی مختاری(9831143)

Resources:

1. <https://barrgroup.com/embedded-systems/how-to/c-volatile-keyword>
2. <https://www.embedded.com/introduction-to-the-volatile-keyword/>
3. <https://www.geeksforgeeks.org/understanding-volatile-qualifier-in-c/>
4. <https://www.geeksforgeeks.org/understanding-volatile-qualifier-c-set-1-introduction/>
5. <https://www.drdobbs.com/cpp/volatile-the-multithreaded-programmers-b/184403766>
6. <https://www.tutorialspoint.com/What-does-the-volatile-keyword-mean-in-Cplusplus>
7. <https://stackoverflow.com/questions/30472561/volatile-keyword-in-microcontrollers>
8. <https://en.cppreference.com/w/cpp/language/cv>
9. <https://softwareengineering.stackexchange.com/questions/307128/what-does-it-mean-to-declare-a-volatile-variable>