

Zad 4.

środa, 1 marca 2023 13:34

Zadanie 4. Rozważamy słowa kluczowe ze standardu C11 (a nie C++). Jakie jest działanie «volatile» w stosunku do zmiennych? Kiedy programiści muszą go użyć, by program zachowywał się poprawnie? Jaki jest skutek użycia «static» w stosunku do zmiennych globalnych, zmiennych lokalnych i procedur? Kiedy należy go używać? Jaką rolę pełni «restrict» odnośnie typów wskaźnikowych?

Wskazówka: W przypadku «volatile» nie chodzi o wyłączenie optymalizacji!

1) volatile

Modyfikator volatile dodany do zmiennej jest informacją dla kompilatora, że jej zawartość może się zmienić w nieznanych momentach nawet jeśli kod danej funkcji jej nie zmienia. Konsekwencją jest niestosowanie optymalizacji dla zmiennej volatile. Oznacza to, że kompilator przy każdym użyciu odczytuje jej wartość z pamięci zamiast przechowywać ją w rejestrze jeśli wykonuje na niej kilka operacji.

<https://ucqosu.pl/2017/06/uzywac-modyfikatora-volatile/>

Without volatile keyword	With volatile keyword
<pre>#include <stdio.h> int main() { /* These variables will never be created on stack*/ int a = 10, b = 100, c = 0, d = 0; /* "printf" will be called with arguments "%d" and 110 (the compiler computes the sum of a+b), hence no overhead of performing addition at run-time */ printf("%d", a + b); /* This code will be removed via optimization, but the impact of 'c' and 'd' becoming 100 can be seen while calling "printf" */ a = b; c = b; d = b; /* Compiler will generate code where printf is called with arguments "%d" and 200 */ printf("%d", c + d); return 0; }</pre>	<pre>#include <stdio.h> int main() { volatile int a = 10, b = 100, c = 0, d = 0; printf("%d", a + b); a = b; c = b; d = b; printf("%d", c + d); return 0; }</pre>

[https://en.wikipedia.org/wiki/Volatile_\(computer_programming\)#Optimization_comparison_in_C](https://en.wikipedia.org/wiki/Volatile_(computer_programming)#Optimization_comparison_in_C)

Zmienne ulotne pojawiają się przede wszystkim w dostępie do sprzętu, gdzie

korzystanie z pamięci jest wykorzystywane do komunikacji pomiędzy urządzeniami oraz w środowisku wielowątkowym, w którym różne wątki mogą korzystać z tej samej zmiennej.

2) static

Static local variables [edit]

Variables declared as `static` inside a function are statically allocated, thus keep their memory location throughout all program execution, while having the same scope of visibility as `automatic local variables` (`auto` and `register`), meaning they remain local to the function. Hence whatever values the function puts into its `static local variables` during one call will still be present when the function is called again.

[https://en.wikipedia.org/wiki/Static_\(keyword\)](https://en.wikipedia.org/wiki/Static_(keyword))

In the C programming language, static is used with global variables and functions to set their scope to the containing file.

<https://stackoverflow.com/questions/572547/what-does-static-mean-in-c>

3) restrict

In the C programming language, `restrict` is a keyword, introduced by the C99 standard,^[1] that can be used in pointer declarations. By adding this type qualifier, a programmer hints to the compiler that for the lifetime of the pointer, no other pointer will be used to access the object to which it points. This allows the compiler to make optimizations (for example, vectorization) that would not otherwise have been possible.

<https://en.wikipedia.org/wiki/Restrict>