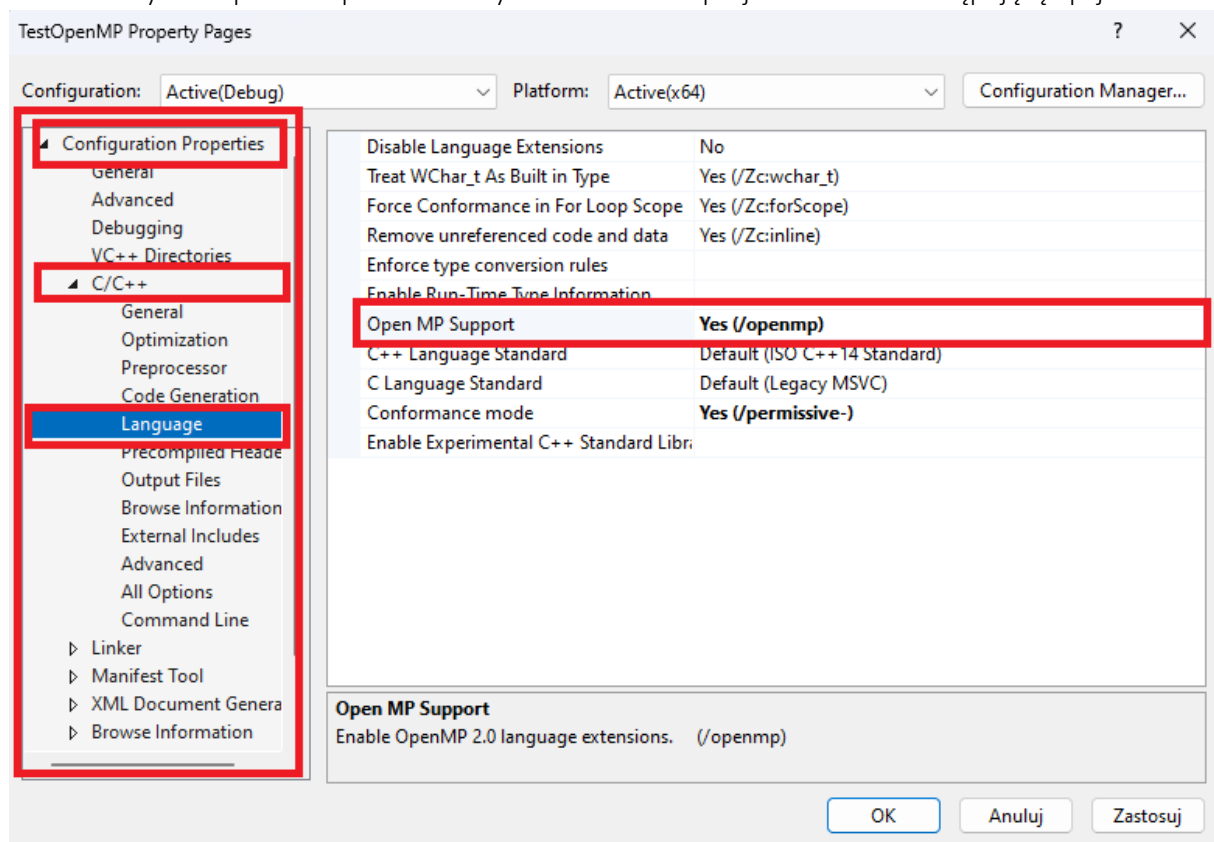


Ustawienie wsparcia dla OpenMP w Visual Studio

Żeby VS wspierało OpenMP należy w ustawieniach projektu zmienić następującą opcję:



Dyrektywy default i schedule

PWIR_09_00.cpp

W przykładzie tym zademonstrowano w jaki sposób działa dyrektywa default.

default(shared) – określa wszystkie zmienne jako wspólne,

default(none) – deklaruje że żadna ze zmiennych nie jest zmienną prywatną ani wspólną, programista musi określić rodzaj każdej ze zmiennych np:

default(none) shared(A, B, C) private(D, E, F)

PWIR_09_01.cpp

Schedule pozwala na określenie sposobu rozdziału zadań. Przedstawienie zastosowania dyrektywy schedule:

schedule(SposóbRozdziałuIteracji, [chunk])

Można ją stosować jedynie z pętlą for, w innym przypadku napotkamy błąd. Parametr **chunk** jest opcjonalny. Określa on licznosc każdego z podzbiorów iteracji.

Dostępne sposoby rozdziału iteracji:

static, wówczas iteracje rozdzielane są na zbiory o rozmiarze chunk. Następnie są one kolejno przydzielane dostępnym wątkom. Jeśli np.: program ma do wykonania 10 iteracji, a mamy dwa wątki, wówczas najlepiej przyjąć za chunk wartość równą 5. Oznacza to, że mamy dwa zbiory po 5 iteracji, które zostaną przypisane oddzielnym wątkom. Jeśli programista nie określi wielkości zmiennej chunk, wówczas zbiór iteracji jest dzielony na podzbiory, których wielkość wynosi w przybliżeniu $\text{liczba_iteracji} / \text{liczba_wątków}$. Każdemu wątkowi zostanie przypisany co najwyżej jeden taki podzbiór.

dynamic, wtedy każdemu wątkowi przypisywana jest liczba iteracji określona przez parametr chunk. Po wykonaniu obliczeń wątki otrzymują kolejną porcję iteracji do wykonania. Ostatni przydzielony podzbiór iteracji może mieć wielkość mniejszą niż ta, która jest określona przez chunk. Jeśli programista nie określił wielkości chunk, to podzbiory są jedno-elementowe.

guided, wówczas iteracje rozdzielane są pomiędzy wątki podobnie jak przy opcji dynamic. Różnica polega na tym, że w tym przypadku rozmiar przypisywanych podzbiorów iteracji zmniejsza się w czasie. Jeśli za chunk przyjmujemy w programie wartość 1, to rozmiar każdego przypisywanego zbioru iteracji jest proporcjonalny do wielkości $\text{liczba_nie_przydzielonych_iteracji} / \text{liczba_wątków}$. W trakcie wykonywania pętli wielkość ta zmierza do 1. Jeśli przyjmujemy $\text{chunk}=m$ ($m>1$), to rozmiar każdej porcji iteracji jest ustalany jak wyżej, z tą różnicą, że rozmiar ten nie może być mniejszy od m . Wyjątek stanowi ostatni przydzielony zbiór iteracji, którego wielkość może być mniejsza niż m . Gdy parametr chunk nie jest jawnie określony, to jego domyślna wartość wynosi 1.

runtime, wtedy jedna z powyższych opcji oraz wartość chunk jest ustalana w czasie działania programu, na podstawie zmiennej środowiskowej OMP_SCHEDULE.

Zadanie:

Przetestuj każdą z opcji schedule. Wykonaj między 5 a 10 pomiarów (w zależności od wydajności) dla każdej z opcji a następnie wyciągnij z nich średnią. Wyniki zapisać i umieścić wraz z kodem w repozytorium.

Wyniki oraz program prześlij do swojego repozytorium. Umieść je w folderze o tej samej nazwie co ten PDF.

Dyrektywa reduction

PWIR_9_02.cpp

reduction (operator : zmienna)

Operator informuje jaką operacja będzie wykonywana na zmiennej. Lista dostępnych operacji:

Operator	Nazwa działania
+	Dodawanie
-	Odejmowanie
*	Mnożenie
&	Koniunkcja bitowa
	Alternatywa bitowa
^	Operacja bitowa EXOR
&&	Koniunkcja logiczna
	Alternatywa logiczna

Pisząc `reduction(+ : a)`, informujemy kompilator, że do zmiennej `a` będziemy dodawać (operator `+`) kolejno wyliczone wartości. Kompilator utworzy odpowiednią liczbę prywatnych kopii zmiennej `a` dla każdego wątku i rozdzieli iteracje pomiędzy dostępne wątki. Każdy wątek będzie operował tylko na swojej kopii zmiennej `a`. Po wykonaniu wszystkich iteracji, wartości wyliczone przez wszystkie wątki są do siebie dodawane.

Zadanie:

- 1.Usuń dyrektywę `reduction` i porównaj wyniki. Dlaczego wyniki się różnią?
2. Napisz funkcję tworzącą wektor jednowymiarowy o wielkości 10.000 elementów. Uzupełnij go `RAND` w zakresie 0-10, a następnie oblicz długość wektora. Porównaj i zapisz wyniki z wykorzystaniem zrównoleglenia oraz bez.

Wyniki oraz program prześlij do swojego repozytorium. Umieść je w folderze o tej samej nazwie co ten PDF.