

Programowanie współbieżne

LABORATORIUM - 6A: LICZBY PIERWSZE...

Andrzej Baran
baran@kft.umcs.lublin.pl



Zliczanie liczb pierwszych

Program sekwencyjny

1. Napisać program zliczania liczb pierwszych w przedziale (k_{min} , k_{max}). Zastosować prosty algorytm sprawdzania, czy liczba jest pierwsza. Zmierzyć czas wykonania programu.
2. Jak sparalelizować problem. Zaproponować rozwiązanie. Opisać jak to zrobić.

Program primes.f90 I

```
program primes
!
! Program zlicza liczby pierwsze w przedziale
! (kmin, kmax); podaje ich ilosc i czas zliczania
!
implicit none
integer, parameter :: kmin=1, kmax=20000000
integer numberOfPrimes, countPrimes
real*4 startingTime, elapsedTime

! pomiar czasu
startingTime=secnds(0.0)

numberOfPrimes = countPrimes(kmin, kmax)
print *, numberOfPrimes

! czas wykonania
elapsedTime=secnds(startingTime) ! elapsed time
print *, "Time = ", elapsedTime

end program primes

function countPrimes(kmin, kmax) result(count)
!
! procedura zlicza liczby pierwsze w przedziale
! (kmin, kmax) i zwraca ich liczbe
!
implicit none
```

Program primes.f90 II

```
integer, intent(in) :: kmin, kmax
integer number, count
logical isPrime
count = 0
do number=kmin, kmax
  if (isPrime(number)) count = count + 1
end do
end function countPrimes

function isPrime(number)
!
! funkcja sprawdza, czy liczba jest pierwsza
! i zwraca .true. jeśli jest lub .false.
!
logical isPrime
integer, intent(in) :: number
integer k, limit

limit = int(sqrt(number/2.))
isPrime = .true.
do k=2, limit
  if (mod(number,k) == 0) then
    isPrime = .false.
    return
  end if
end do
end function isPrime
```

Paralelizacja

1. Urównoleglić program `primes.f90` w środowisku MPI.
2. Wykonać obliczenia i zmierzyć czas wykonania. Porównać czas wykonania z czasem wykonania wersji sekwencyjnej programu.