

# Programowanie współbieżne

## LABORATORIUM - 9: MPI, BŁĄDZENIE

Andrzej Baran  
baran@kft.umcs.lublin.pl

# Błądzenie przypadkowe, MPI, test I

## Fortran 95

- Napisz program błądzenia przypadkowego cząstki w 2 wymiarach. Rozważ 100000 lub więcej cząstek, wykonujących 10 kroków. Znajdź rozkład odległości cząstek od ich miejsca startu (punkt (0,0)). Napisz wersję sekwencyjną i wersję równoległą programu w MPI i Open MP.

### PROGRAM SEKWENCYJNY:

---

! RS6000 SP: Practical MPI programming

PROGRAM randomwalk

PARAMETER (n = 10000)

INTEGER itotal(0:9)

REAL seed

pi = 3.1415926

DO i = 0, 9

    itotal(i) = 0

END DO

! Uwaga: zależy od kompilatora

! tutaj: gfortran

iseed = 5

CALL srand(iseed)

DO i = 1, n



# Błądzenie przypadkowe, MPI, test II

Fortran 95

```
x = 0.0
y = 0.0
DO istep = 1, 10
  angle = 2.0 * pi * rand()
  x = x + cos(angle)
  y = y + sin(angle)
END DO
itemp = sqrt(x*x + y*y)
itotal(itemp) = itotal(itemp) + 1
END DO
PRINT *, "total =", itotal
END PROGRAM randomwalk
```

---

# Błądzenie przypadkowe, MPI, test I

Fortran 95

## PROGRAM RÓWNOLEGŁY, MPI:

---

```
! RS6000 SP: Practical MPI programming
PROGRAM randomwalk_p
  INCLUDE "mpif.h"

  PARAMETER (n = 100000)
  INTEGER itotal(0:9), iitotal(0:9)
  REAL seed

  CALL MPI_INIT(ierr)
  CALL MPI_COMM_SIZE(MPI_COMM_WORLD, nprocs, ierr)
  CALL MPI_COMM_RANK(MPI_COMM_WORLD, myrank, ierr)

  CALL para_range(1, n, nprocs, myrank, ista, iend)
  pi = 3.1415926

  DO i = 0, 9
    itotal(i) = 0
  END DO
```

! Uwaga: zależy od kompilatora



# Błądzenie przypadkowe, MPI, test II

## Fortran 95

```
! tutaj: gfortran
seed = 5 + myrank
CALL srand(seed)
DO i = ista, iend
  x = 0.0
  y = 0.0
  DO istep = 1, 10
    angle = 2.0 * pi * rand()
    x = x + cos(angle)
    y = y + sin(angle)
  END DO
  itemp = sqrt(x**2 + y**2)
  itotal(itemp) = itotal(itemp) + 1
END DO

CALL MPI_REDUCE(itotal, iitotal, 10, MPI_INTEGER, MPI_SUM, 0, &
  MPI_COMM_WORLD, ierr)

PRINT *, "total =", iitotal

CALL MPI_FINALIZE(ierr)
```



# Błądzenie przypadkowe, MPI, test III

Fortran 95

END

```
SUBROUTINE para_range(n1, n2, nprocs, irank, ista, iend)
  iwork1 = (n2 - n1 + 1) / nprocs
  iwork2 = MOD(n2 - n1 + 1, nprocs)
  ista = irank * iwork1 + n1 + MIN(irank, iwork2)
  iend = ista + iwork1 - 1
  IF (iwork2 > irank) iend = iend + 1
END SUBROUTINE para_range
```

---