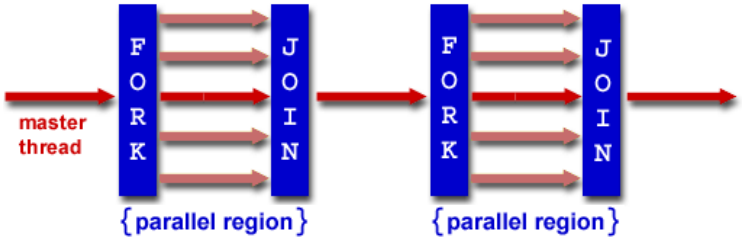
czyli **programowanie równoległe** przy użyciu interfejsu programowania aplikacji (API) **Open Multi-Processing**

Cz. II

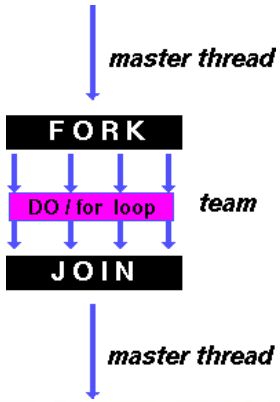
Bogumił Konopka
W-11/I-21
12.04.2010
Seminarium Grupy Bioinformatyki i Biofizyki Nanoporów

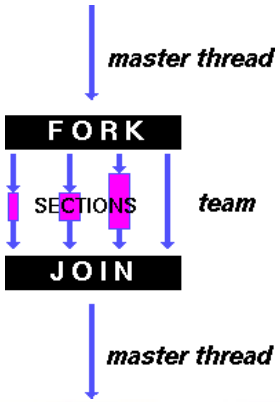
O czym już było?

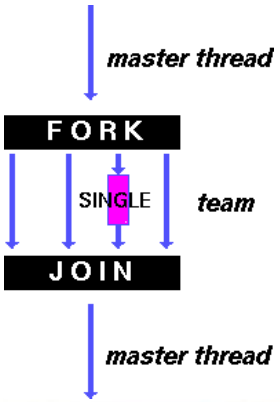
- Co to jest wielowątkowość?



- Jakie są rodzaje segmentów równoległych?

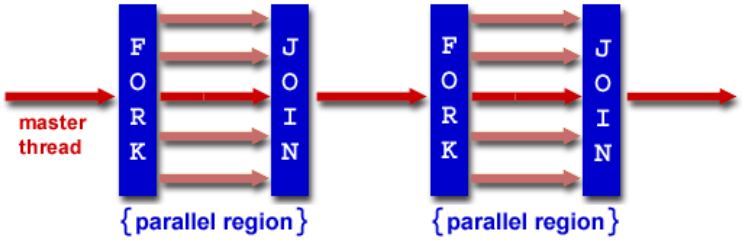






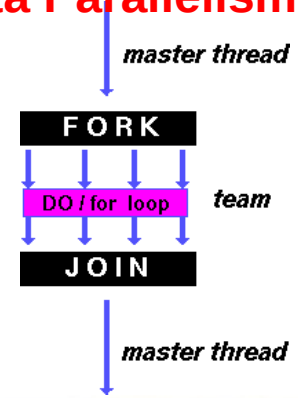
O czym już było?

- Co to jest wielowątkowość?

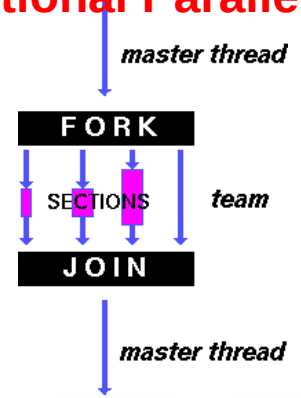


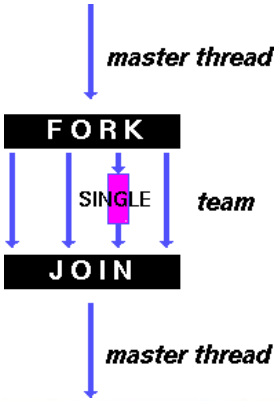
- Jakie są rodzaje segmentów równoległych?

Data Parallelism



Functional Parallelism





O czym już było?

- Jakie rodzaje (ale nie typy!) zmiennych są do dyspozycji
  - shared,private,firstprivate,lastshared
  - threadprivate, copyin, copyprivate
- Jak planowany jest podział pracy w pętlach?
  - Static
  - Dynamic
  - Private
  - Auto
  - Runtime

Ćwiczenia

- Bug1
  - dyrektywa “#pragma omp for” nie poprzedza bezpośrednio pętli w programie
- Bug2
  - Nie zdefiniowano typu zmiennych
  - Total może być typu “shared” lub “private”

Przykłady: bug1.c, bug2.c

Segmenty równoległe - “section”

- Sekcja:
  - oznacza fragmenty kodu, które mają być rozdzielone pomiędzy wątkami
  - w odróżnieniu od pętli nie mają kolejnych iteracji

Przykład omp_workshare2.c

Segmenty równoległe - “single”

- Segment Single
  - Jest wykonywany tylko przez jeden wątek w grupie
  - Wykorzystywane np. Przy operacjach wejścia/ wyjścia

Synchronizacja pracy wątków

- Problem:
  - x=x+1;
- Wątek A pobiera
- Wątek B pobiera
- Wątek A inkrementuje
- Wątek B inkrementuje
- Wynik:
  - Jedna inkrementacja zamiast dwóch

Synchronizacja pracy wątków

- Problem:
  - x=x+1;
- Wątek A pobiera
- Wątek B pobiera
- Wątek A inkrementuje
- Wątek B inkrementuje
- Wynik:
  - Jedna inkrementacja zamiast dwóch

Przykład: bug2_fixed – shared(total)

Synchronizacja pracy wątków

- #pragma omp **master** newline
- #pragma omp **critical** [name] newline
- #pragma omp **barrier** newline
- #pragma omp **taskwait** newline
- #pragma omp **atomic** newline
- #pragma omp **flush** (list) newline
- # pragma omp **for** **ordered**

Dyrektywa MASTER

- Określa obszar realizowany tylko przez wątek główny
- Pozostałe wątki pomijają tę część kodu

Dyrektywa CRITICAL

- Określa region, który w danym momencie może być realizowana tylko przez jeden wątek
- Gdy wątek B dotrze do obszaru, CRITICAL w trakcie jego realizacji przez wątek A, następuje wstrzymanie wątku B.

Przykład omp_critical.c

Materiały

- https://computing.llnl.gov/tutorials/openMP/
- https://computing.llnl.gov/tutorials/openMP/exercis
- Przyklady:
  - omp_bug1.c
  - omp_bug2.c
  - omp_workshare2.c
  - omp_critical.c – wlasn/brak na stronie