Programowanie równoległe i rozproszone

Wykład 1

Wprowadzenie
Pojęcia podstawowe
Problemy i zastosowanie obliczeń równoległych

Komputer równoległy to zbiór elementów przetwarzających,
(Zamierzamy użyć wiele procesorów, aby to zrealizować)

które współpracują w celu szybkiego rozwiązywania problemów
(Dbamy o wydajność, Dbamy o efektywność)

Przyśpieszenie

Jedna główna motywacja do korzystania z przetwarzania równoległego

Przyspieszenie = **czas wykonania (przy użyciu 1 procesora)** / **(przy użyciu procesorów P)** **czas wykonania (przy użyciu procesorów P)**

Komputer równoległy to zbiór elementów przetwarzających,
(Zamierzamy użyć wiele procesorów, aby to zrealizować)

które współpracują w celu szybkiego rozwiązywania problemów
(Dbamy o wydajność, Dbamy o efektywność)

Przyśpieszenie

Jedna główna motywacja do korzystania z przetwarzania równoległego

Przyspieszenie = **czas wykonania (przy użyciu 1 procesora)** / **(przy użyciu procesorów P)** **czas wykonania (przy użyciu procesorów P)**

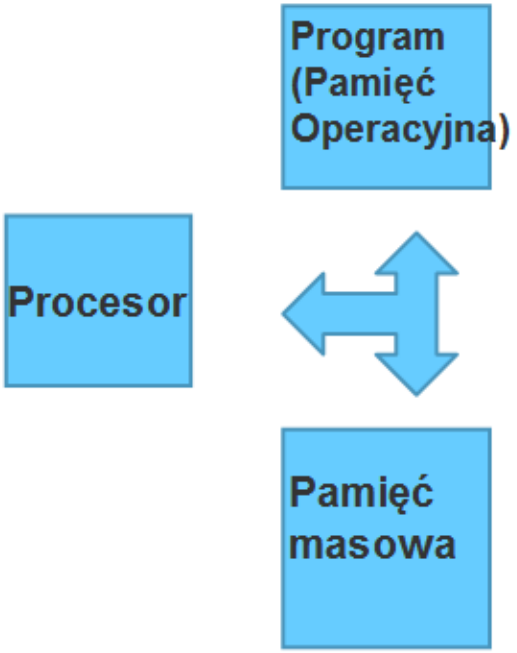
Programowanie - proces projektowania, tworzenia, testowania i utrzymywania kodu źródłowego programów komputerowych

Program - sekwencja symboli opisująca obliczenia zgodnie z pewnymi regułami zwanymi językiem programowania.

Proces - egzemplarz programu będący w trakcie wykonywania przez system komputerowy

Wykonywanie procesu realizowane jest w środowisku, którego niezbędnymi elementami są:

- procesor - dokonujący zmian stanu systemu
- pamięć - w której przechowywany jest kod programu oraz dane



- Komunikacja ograniczała maksymalne osiągnięte przyspieszenie

- Minimalizacja kosztów komunikacji poprawiona przyspieszenie

- Koszty komunikacji mogą dominować równolegle obliczenia, poważnie ograniczające przyspieszenie

- Brak równowagi w przydzielonych zadaniach ograniczone przyspieszenie

Projektowanie i pisanie programów równoległych

- **Myślenie równoległe**

1. Rozkładanie pracy na fragmenty, które można bezpiecznie wykonywać równolegle
2. Przydzielanie pracy procesorom
3. Zarządzanie komunikacją / synchronizacją pomiędzy procesorami tzw
że nie ogranicza przyspieszenia

- **Abstrakcje / mechanizmy wykonywania powyższych zadań**

1. Pisanie kodu w popularnych językach programowania równoległego

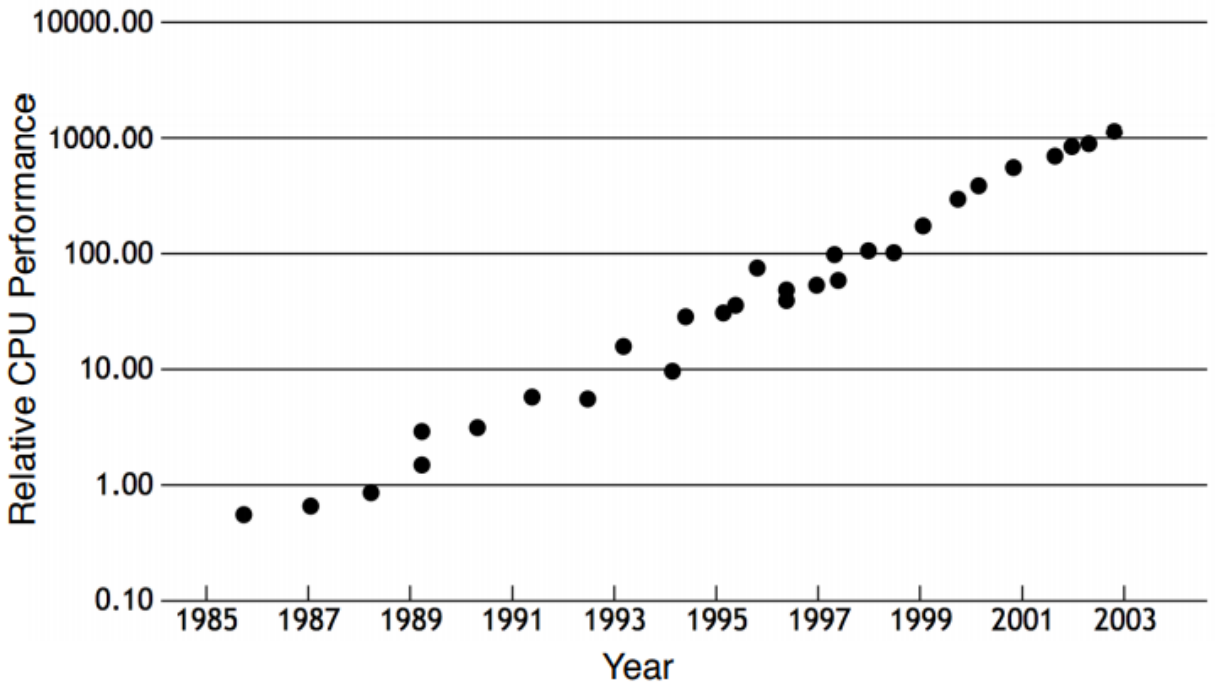
Myślenie o wydajności

SZYBKO! = WYDAJNIE

- Dzieje się tak tylko dlatego, że program działa szybciej na komputerze równoległym nie oznacza, że efektywnie wykorzystuje sprzęt (Czy dwukrotne przyspieszenie na komputerze z 10 procesorami to dobry wynik?)
- Perspektywa programisty: wykorzystaj dostępne możliwości maszyny
- Perspektywa projektanta sprzętu komputerowego: wybór odpowiednich funkcji do wykorzystania system (wydajność / koszt, koszt = powierzchnia krzemu ?, moc? itp.)

Kontekst historyczny:

Dlaczego było brak przetwarzania równoległego w pierwszych dziesięcioleciach ?



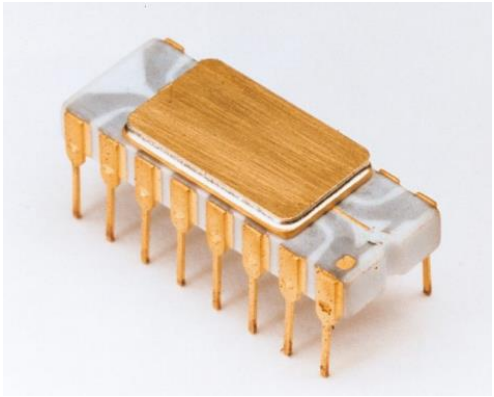
- Podwajanie wydajności jednowątkowych procesorów ~ co 18 miesięcy
- Implikacja: praca nad zrównoleglenie kodu często nie była warta zachodu
- Programista nic nie robi, w przyszłym roku kod sam przyspiesza !!!

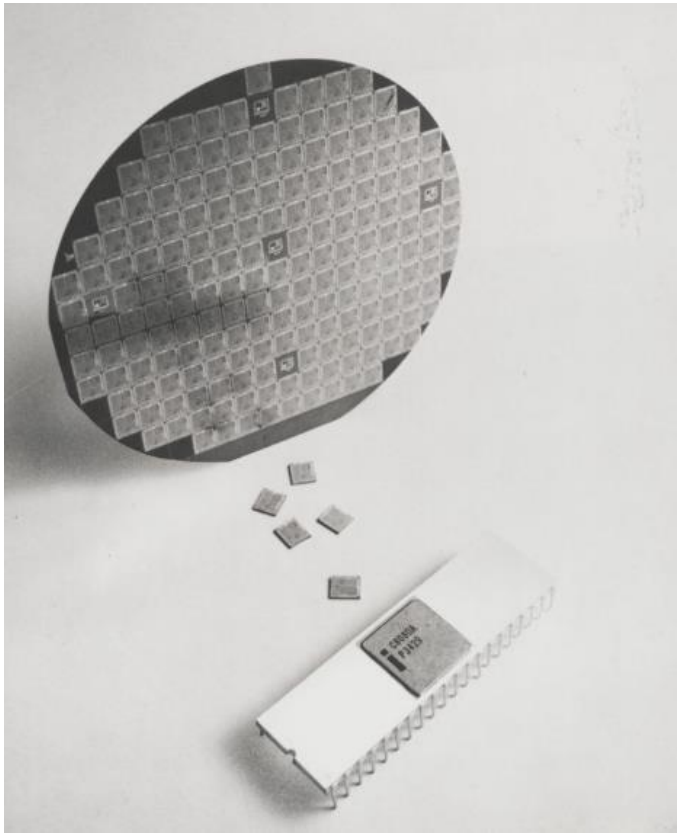
Kontekst historyczny

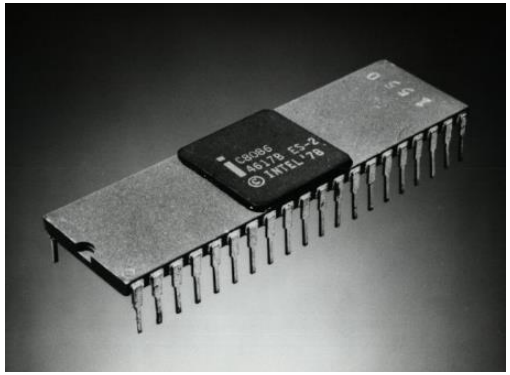
Dlaczego było brak przetwarzania równoległego w pierwszych dziesięcioleciach ?

Założyciele Intela Andy Grove, Robert Noyce, Gordon Moore

Intel







Klon 8080 – MCY7880 był jedynym produkowanym w Polsce procesorem

4004 (1971)
4 bitowy
740 kHz

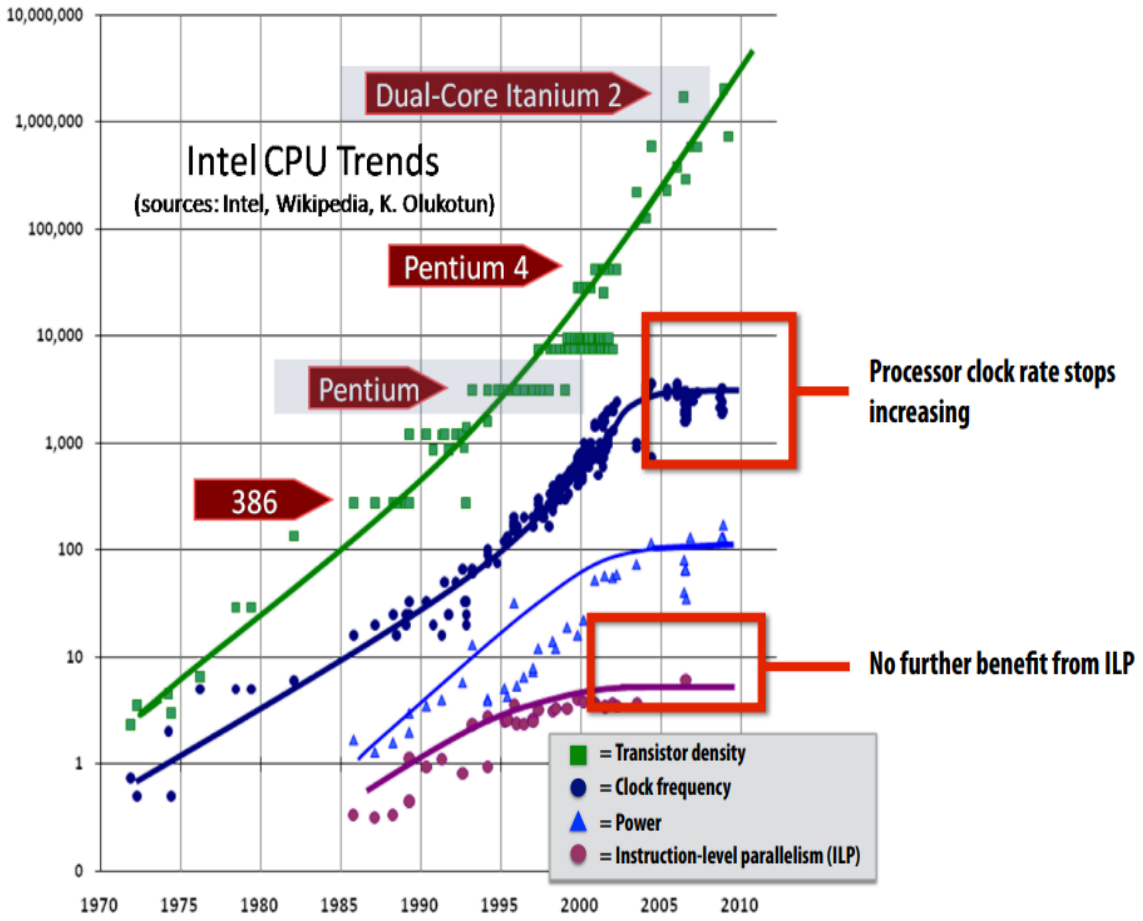
8080 (1974)
8 bitowy
3.125 MHz

8086 (1978)
16 bitowy
10 MHz

„Ściana mocy”

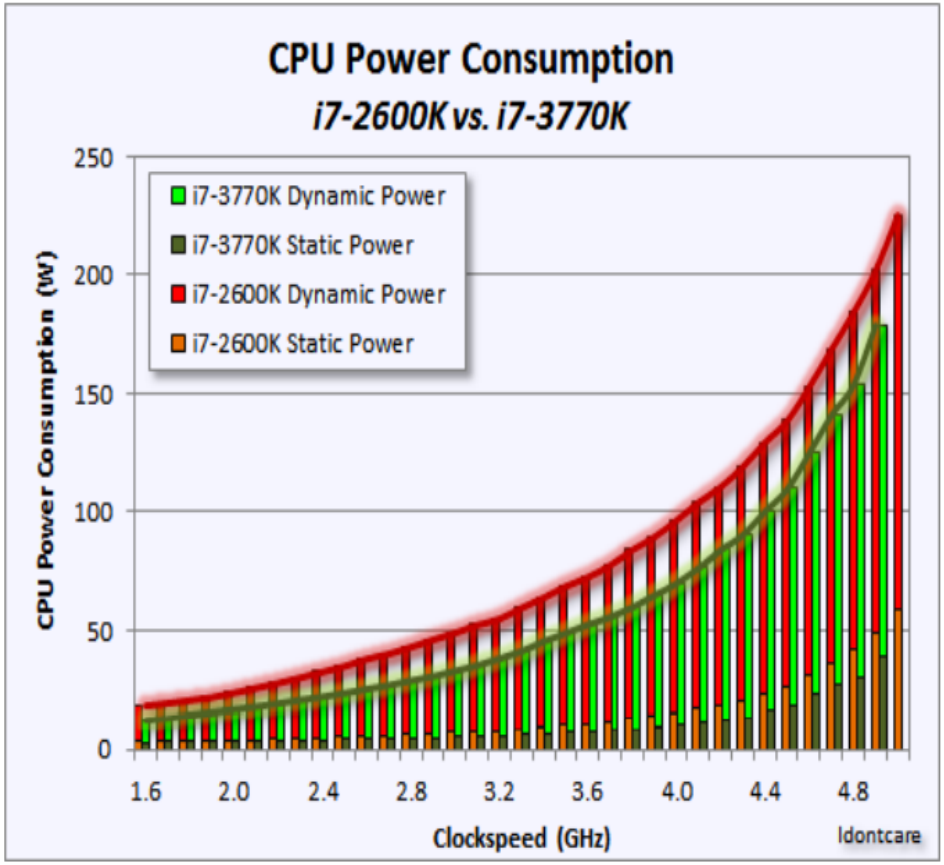
Moc dynamiczna = obciążenie pojemnościowe × napięcie^2 × częstotliwość

Wysoka moc = duże ciepło



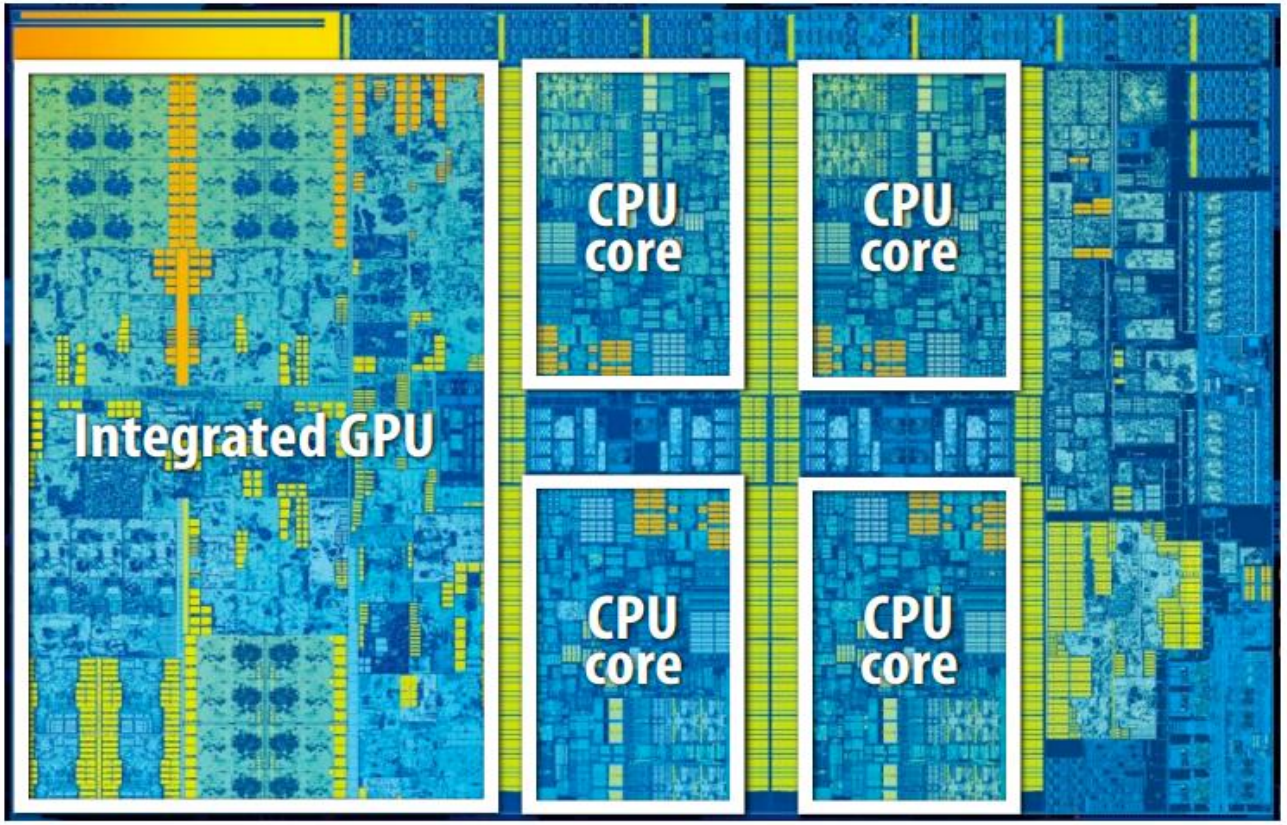
Moc jest krytycznym ograniczeniem projektowym w nowoczesnych procesorach

	TDP
Intel Core i7 (in this laptop):	45W
Intel Core i7 2700K (fast desktop CPU):	95W
NVIDIA GTX 780 GPU	250W
Mobile phone processor	1/2 - 2W
World's fastest supercomputer	megawatts
Standard microwave oven	700W

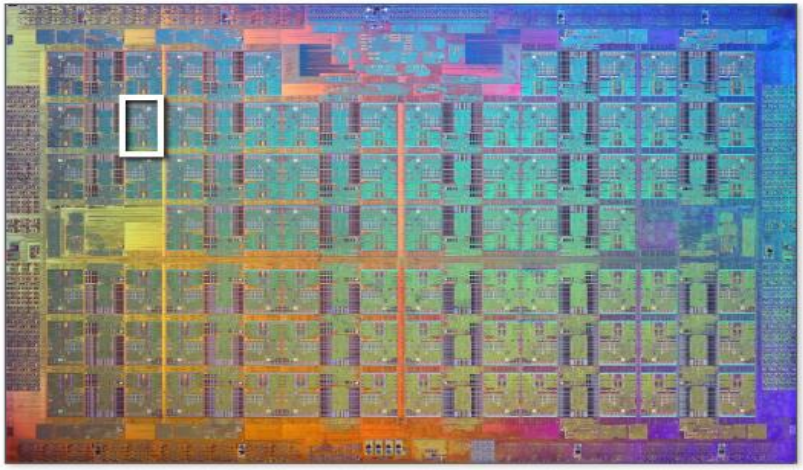


Poprawa wydajności obliczeń aktualnie jest możliwa głównie poprzez wprowadzenie równoległości obliczeń

Intel Skylake (2015) na przykładzie i7 6 generacji
Quad-core CPU + multi-core GPU w jednym układzie scalonym



Intel Xeon Phi 7290 “coprocessor” (2016) 72 rdzeni (1.5 Ghz)



NVIDIA Maxwell GTX 1080 GPU (2016)

Mobilne przetwarzanie równoległe

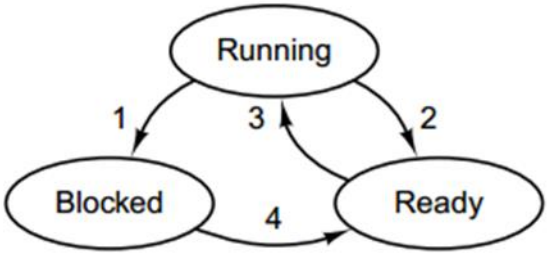
Ograniczenia mocy mają duży wpływ na projektowanie systemów mobilnych (mniejsza częstotliwość pracy rzędu 1Ghz)

Apple A9: (in iPhone 6s)
Dual-core CPU + GPU + image processor and more on one chip

NVIDIA Tegra X1:
4 ARM A57 CPU cores +
4 ARM A53 CPU cores +
NVIDIA GPU + image processor...

Superkomputery

klastry wielordzeniowych procesorów + GPU
np.
Oak Ridge National Laboratory: Titan (superkomputer nr 3 na świecie)
- 18688 x 16 rdzeniowych procesorów AMD + 18688 procesorów graficznych NVIDIA K20X



1. Process blocks for input
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available

Podstawowe stany procesów:
- uruchomiony (running) - w danej chwili wykonuje się na procesorze,
- gotowy (ready) - gotowy do wykonania, ale wstrzymany w oczekiwaniu na przydział czasu procesora,
- wstrzymany (blocked) - nie może kontynuować pracy do momentu wystąpienia pewnego zewnętrznego zdarzenia,

Cele programowania równoległego i rozproszonego

1. Wzrost szybkości realizowanych obliczeń
2. Optymalizacja dostępnych zasobów sprzętowych
3. Praca wielozadaniowa i dla wielu użytkowników
4. Usystematyzowanie metodologii oprogramowania

Modele programowania równoległego
Miary efektywności obliczeń równoległych
Poprawność programów równoległych

Wzajemnie oddziaływanie procesów:

- współpraca
- współzawodnictwo

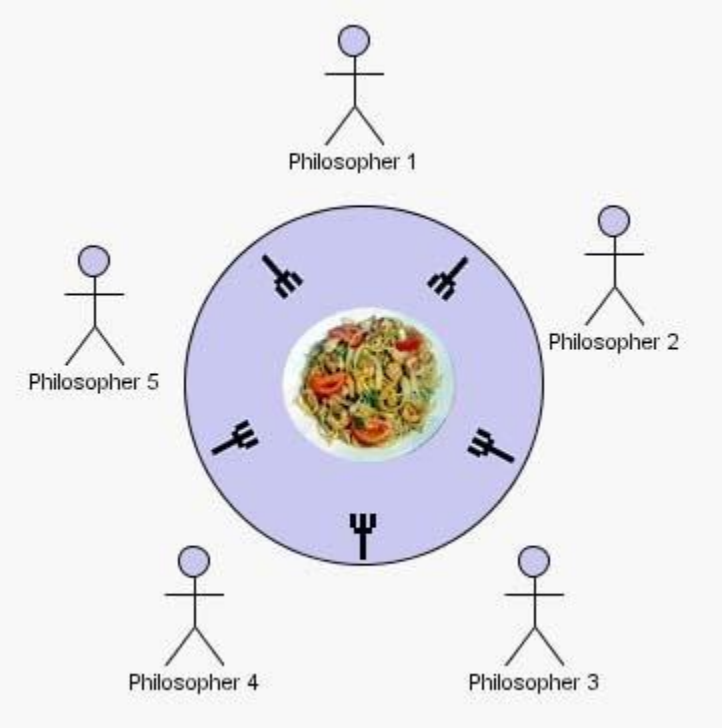
Współpraca procesów – realizowane zadanie jest podzielone na kilka podzadań wykonywanych przez oddzielne procesy

Współzawodnictwo procesów – niezależne zadania wykonywane przez odrębne procesy współzawodniczące o ograniczone zasoby.

Zasób – każdy obiekt, który jest konieczny do realizacji kolejnych instrukcji procesu. Jego brak powoduje przejście procesu w stan wstrzymania wykonywania.

Zdarzenie – każdy fakt zachodzący poza procesem, który ma wpływ na wykonywanie procesu np. zajęcie/zwolnienie zasobu przez inny proces wykonywany równolegle.

Przy okrągłym stole siedzi pięciu filozofów. Przed każdym filozofem stoi talerz, a pomiędzy kolejnymi dwoma talerzami leży jeden widelec. Na środku stołu stoi półmisek ze spaghetti.



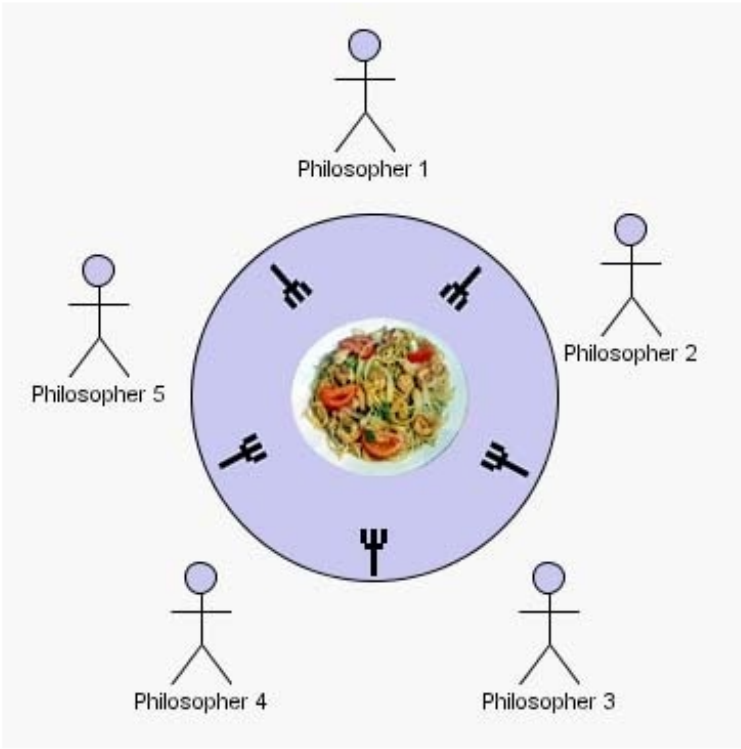
Każdy filozof myśli, a po pewnym czasie (gdy zgłodnieje) stara się jeść. Aby rozpocząć jedzenie, filozof musi podnieść oba widelce leżące obok jego talerza. Tym samym uniemożliwia jedzenie swoim sąsiadom. Po zaspokojeniu głodu filozof odkłada oba widelce na stół.

- dwóch filozofów nie korzystało jednocześnie z tego samego widelca (wzajemne wykluczanie),
- każdy filozof kończył jedzenie sam (niewywłaszczalność),
- nie nastąpiła sytuacja, w której każdy filozof trzyma jeden widelec i czeka na drugi (blokada),
- nie nastąpiło zagłodzenie filozofa czyli sytuacja, w której sąsiedzi filozofa zawsze będą korzystać z widelca.

Działanie filozofa przedstawia następujący program:

Proces Filozof(i)
powtarzaj cyklicznie:

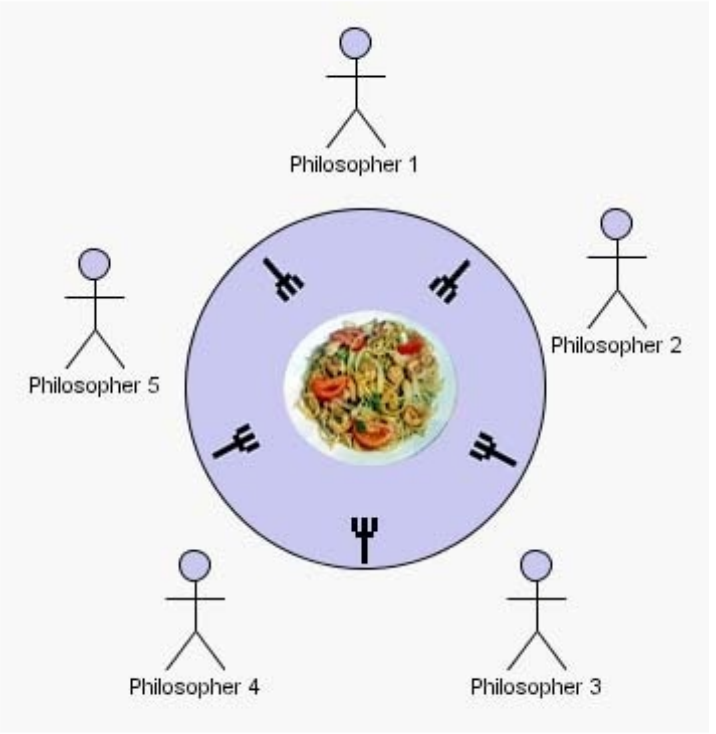
1) myśli
2) siada przy stole na swoim miejscu
3) podnosi lewy widelec
4) podnosi prawy widelec
5) je
6) odkłada oba widelce
7) odchodzi od stołu



Program nie spełnia warunku bezpieczeństwa, gdyż zasoby dzielone (widelce) mogą być wykorzystywane jednocześnie przez dwóch filozofów.
Aby program spełniał warunek bezpieczeństwa, należy go uzupełnić o dodatkowe instrukcje oczekiwania.

Proces Filozof(i)
powtarzaj cyklicznie:

1) myśli
2) siada przy stole na swoim miejscu
3) czeka, aż lewy widelec będzie wolny
4) podnosi lewy widelec
5) czeka, aż prawy widelec będzie wolny
6) podnosi prawy widelec
7) je
8) odkłada oba widelce
9) odchodzi od stołu



Program nie spełnia warunku bezpieczeństwa, gdyż zasoby dzielone (widelce) modą być wykorzystywane jednocześnie przez dwóch filozofów.

Aby program spełniał warunek bezpieczeństwa, należy go uzupełnić o dodatkowe instrukcje oczekiwania.

Warunek bezpieczeństwa spełnia poniższy program :

Proces Filozof(i)
powtarzaj cyklicznie:

1) myśli
2) siada przy stole na swoim miejscu
3) czeka, aż lewy widelec będzie wolny
4) podnosi lewy widelec
5) czeka, aż prawy widelec będzie wolny
6) podnosi prawy widelec
7) je
8) odkłada oba widelce
9) odchodzi od stołu

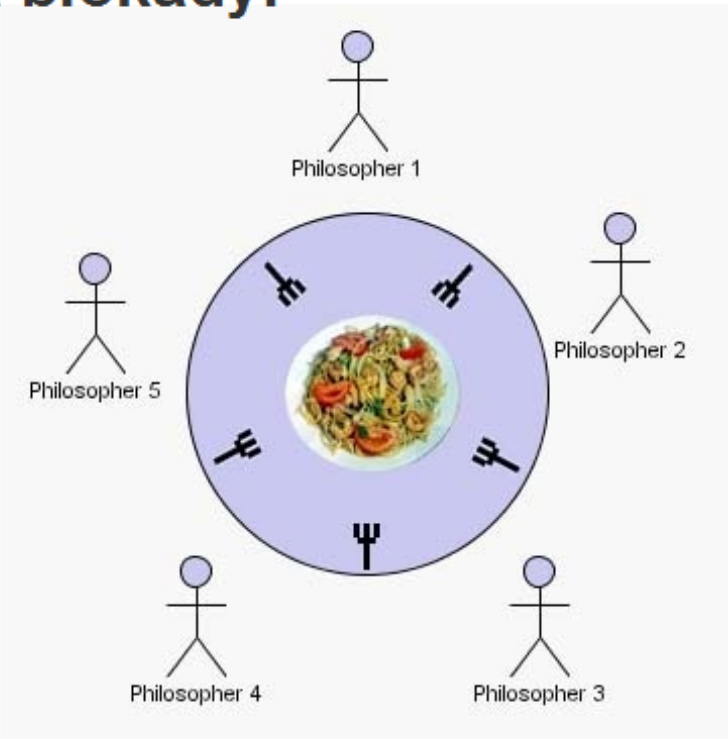
W programie występuje:
- wzajemne wykluczanie – widelec może być wykorzystywany w danej chwili tylko przez jednego filozofa,
- niewywłaszczalność – żaden filozof nie może odebrać widelca sąsiadowi,
- częściowy przydział – filozof podnosi najpierw jeden widelec, a następnie drugi widelec,
- jeśli każdy filozof podniesie lewy widelec w tym samym momencie, to wystąpi zamknięty łańcuch, w którym każdy filozof będzie oczekiwać na oddanie widelca przez poprzednika.

Program nie spełnia warunku żywotności, gdyż może wystąpić blokada.
Aby usunąć możliwość powstania blokady można wyeliminować częściowy przydział zasobów tzn. zapewnić, że filozof podnosi oba widelce jednocześnie.

Poniższy program spełnia warunek bezpieczeństwa oraz nie dopuszcza do powstania blokady:

Poniższy program spełnia warunek bezpieczeństwa oraz nie dopuszcza do powstania blokady:

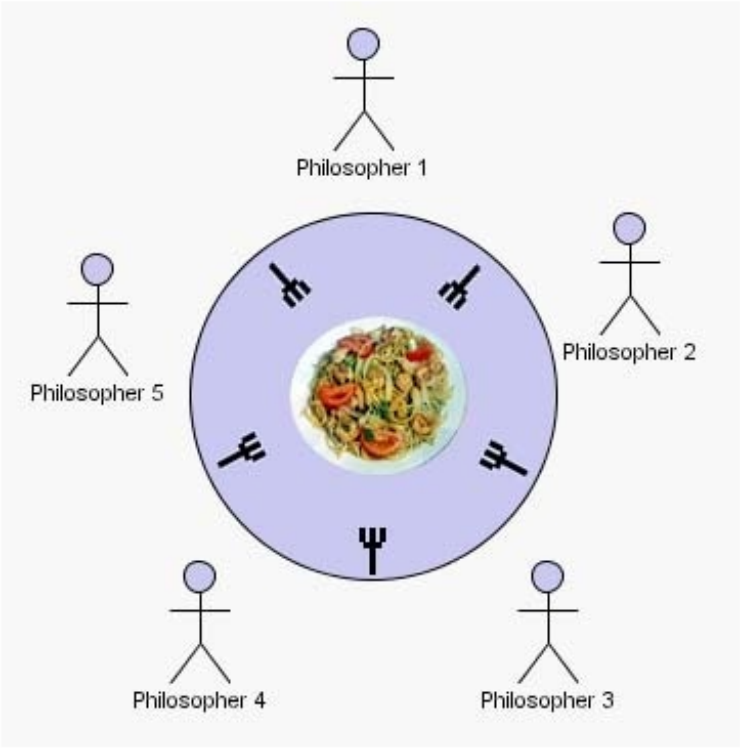
Proces Filozof(i)
powtarzaj cyklicznie:
1) myśli
2) siada przy stole na swoim miejscu
3) czeka, aż oba widelce będą wolne
4) podnosi oba widelce
5) je
6) odkłada oba widelce
7) odchodzi od stołu



Jeśli dwóch sąsiadów filozofa będzie na przemian trzymać widelce, to nastąpi zagłodzenie.
Program nie spełnia warunku żywotności.
Innym sposobem eliminacji możliwości powstania blokady jest niedopuszczenie do powstania zamkniętego łańcucha wzajemnego oczekiwania.
Można to osiągnąć poprzez ograniczenie liczby filozofów, którzy jednocześnie zasiadają przy stole.

W jadalni jest lokaj, który dba o to, by przy stole zawsze zasiadało nie więcej niż 4 filozofów.

Proces Filozof(i)
powtarzaj cyklicznie:
1) myśli
2) czeka, aż lokaj pozwoli dojść do stołu
3) siada przy stole na swoim miejscu
4) czeka, aż lewy widelec będzie wolny
5) podnosi lewy widelec
6) czeka, aż prawy widelec będzie wolny
7) podnosi prawy widelec
8) je
9) odkłada oba widelce
10) odchodzi od stołu



Program spełnia warunki bezpieczeństwa żywotności