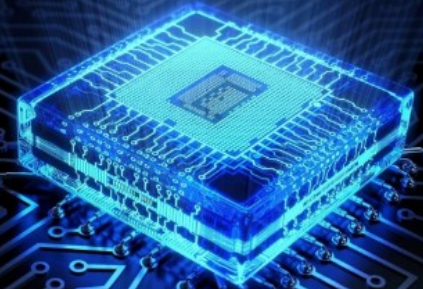
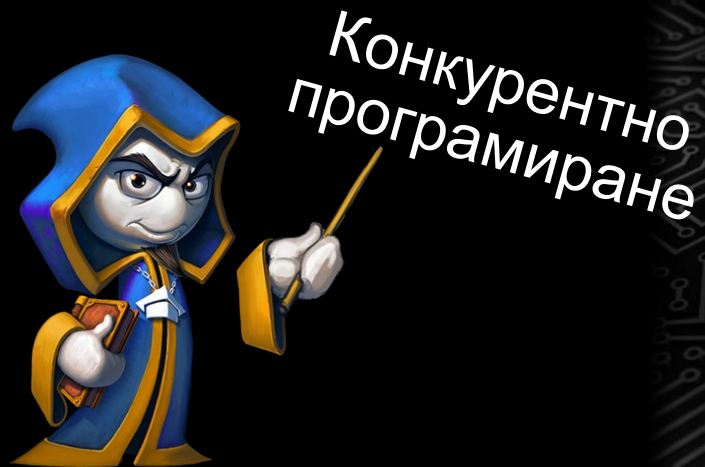


Конкурентно програмиране



Инж. Венцеслав Кочанов

Въведение в конкурентното /паралелно/ програмиране

Паралелността е концепция, свързана с многозадачност и асинхронен вход-изход (I/O). Обикновено се отнася до съществуването на множество нишки за изпълнение, всяка от които може да получи част от времето за изпълнение преди да бъде изпреварена от друга нишка, която също получава част от време. Паралелността е необходима, за да може програмата да реагира външни на фактори, като въвеждане на данни от потребителя чрез клавиатурата, устройства и сензори.

Паралелността също е проблем за много разработчици на приложения. Въпреки че абстракциите, които обсъждаме, първоначално са разработени, за да улеснят писането на правилен код на операционната система, те са широко използвани в следните приложения:

1. Мрежовите услуги трябва да могат да обработват множество заявки от своите клиенти; Google, който може да обработва само една заявка за търсене в даден момент, или от Amazon, чийто клиент може да позволи да си купи само една книга в даден момент, биха били много по-малко полезни.

2.Повечето приложения днес имат потребителски интерфейс; осигуряването на добра комуникация на потребителите при едновременно изпълнение на логиката на приложението е много по-лесно със структуриран подход с паралелността. Паралелните програми трябва да могат да насочват работата към множество процесори, за да получат предимствата на производителността на многоядрените архитектури.

3.Системите за управление на данни се нуждаят от паралелност, за да маскират забавянето на дисковите и мрежовите операции.

Паралелността означава да правите няколко неща едновременно. Например, ако трябва да изпълним милион задачи, тогава вместо да ги изпълняваме последователно една по една, можем да ги правим едновременно, като по този начин намаляваме продължителността на изпълнение на програмата. Или казано най - просто паралелността е извършване на две или повече отделни дейности, по едно и също време.

Паралелизъм

Приложението може да обработва задачата последователно от началото до края или да я раздели на подзадачи, които могат да бъдат изпълнени паралелно. Паралелизмът е, когато множество задачи или няколко части от уникална задача буквално се изпълняват по едно и също време, напр. на многоядрен процесор.

При паралелно изчисление две изчисления напредват едновременно - това е буквално по едно и също време. Това не е възможно с един процесор и вместо това изисква многоядрена настройка. Паралелността означава, че множество процеси или нишки напредват паралелно, т.е. нишките се изпълняват по едно и също време, което се случва, когато всички нишки са планирани на паралелни процесори. С паралелизма дадено приложение разделя задачите си на по-малки подзадачи, които могат да се обработват паралелно, например на множество процесори по едно и също време.

Разлики

1. Паралелността е свойство на програмата, докато паралелното изпълнение е свойство на машината.
2. Паралелността е актът на изпълнение и управление на множество задачи едновременно.
3. Паралелността може да включва различни задачи, които се изпълняват и имат припокриващо се време. Паралелизмът включва различни задачи, които се изпълняват едновременно и обикновено имат едно и също начално и крайно време.

4. Паралелността може да бъде реализирана чрез използване на единична процесорна единица, докато това не е възможно в случай на паралелизъм, който изисква множество процесорни единици. Паралелизмът се получава чрез използване на множество процесори, като многопроцесорна система и управление на различни процеси на тези процесори. Обратно, паралелността се постига чрез преплитане на операциите на процесите в процесора и по-специално превключване на контекста.

5. Паралелно изпълняваните процеси трябва да са едновременно, освен ако не се управляват в един и същи момент, но едновременно изпълняваните процеси никога не могат да бъдат паралелни, защото не се обработват в един и същи момент.

Текущо изпълняваната нишка се преустановява, когато нейният отрязък от време изтече и друга нишка възобновява изпълнението. Когато системата превключва от една нишка към друга, тя запазва контекста на изтеглената нишка и презарежда запазения и контекст в опашката на нишката. Дължината на отрязъка от време зависи от операционната система и процесора. Тъй като всеки времеви отрязък е малък, изглежда, че множество нишки се изпълняват едновременно, дори ако има само един процесор.



Въпроси?

