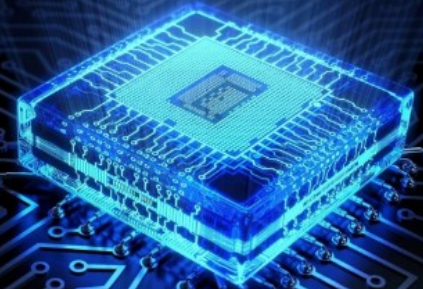
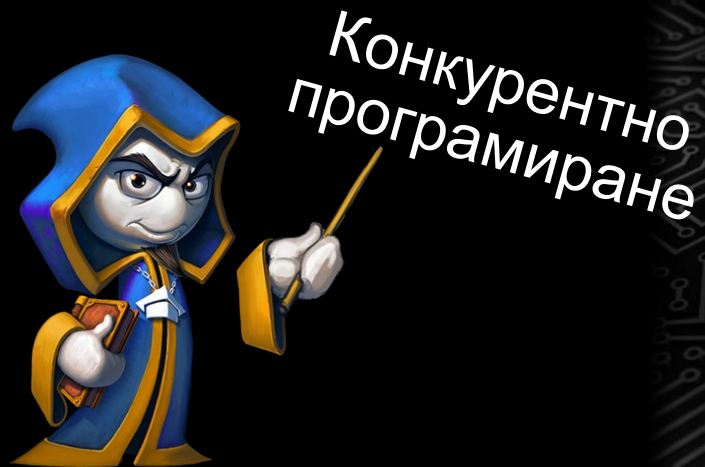


Конкурентно програмиране



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:

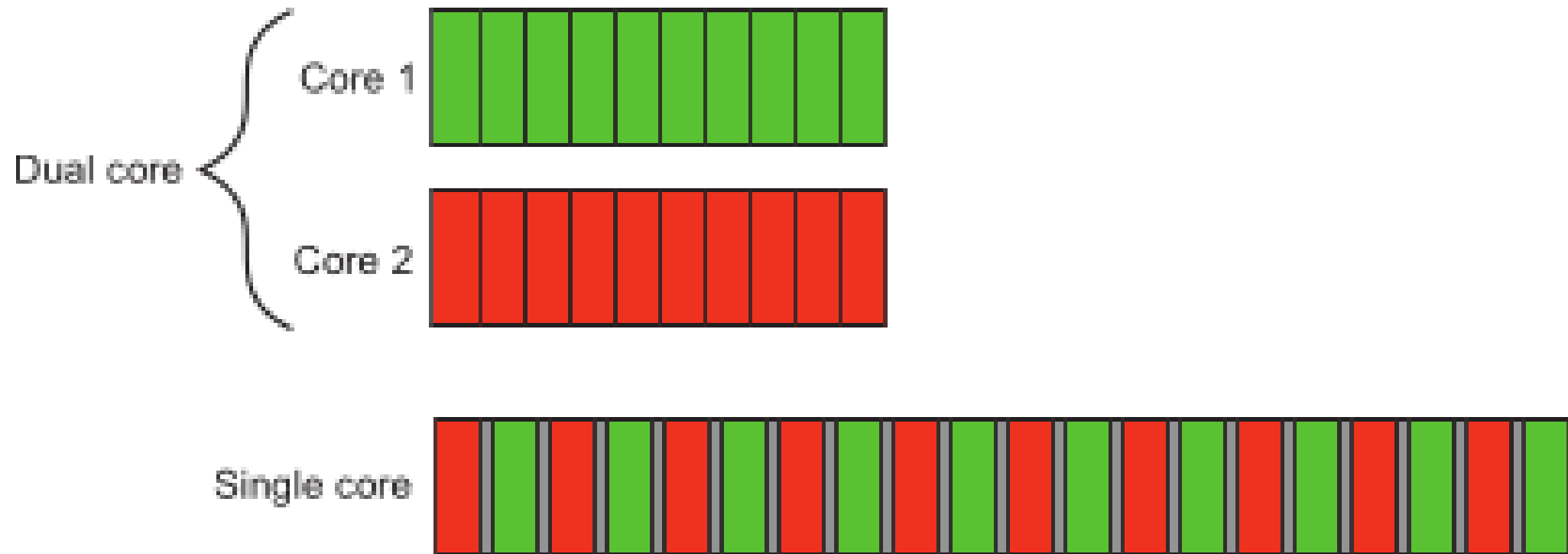
1. Мрежовите услуги трябва да могат да обработват множество заявки от своите клиенти; Google, който може да обработва само една заявка за търсене в даден момент, или от Amazon, чийто клиент може да позволи да си купи само една книга в даден момент, биха били много по-малко полезни.

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.

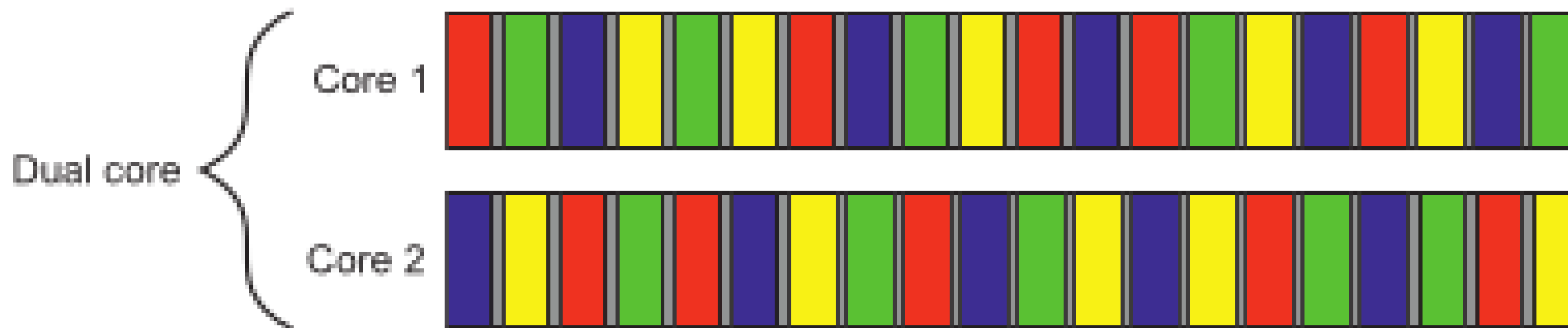
3.Системите за управление на данни се нуждаят от паралелност, за да маскират забавянето на дисковите и мрежовите операции.

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.

Фигура 1.1 показва идеализиран сценарий на компютър с точно две задачи за изпълнение, всяка разделена на 10 еднакви по размер части. На двуядрена машина (която има две процесорни ядра) всяка задача може да се изпълни на собствено ядро. На едноядрена машина, извършваща превключване на задачи, частите от всяка задача се подреждат. Но те също са малко раздалечени (на фигура 1.1 това е показано от сивите ленти, разделящи парчетата, които са по-дебели от разделителните ленти, показани за двуядрената машина); за да извърши преплитането, системата трябва да извършва превключване на контекста всеки път, когато преминава от една задача към друга, а това отнема време.



Фигура 1.1 Два подхода към едновременност: паралелно изпълнение на двуядрена машина срещу превключване на задачи на едноядрена машина

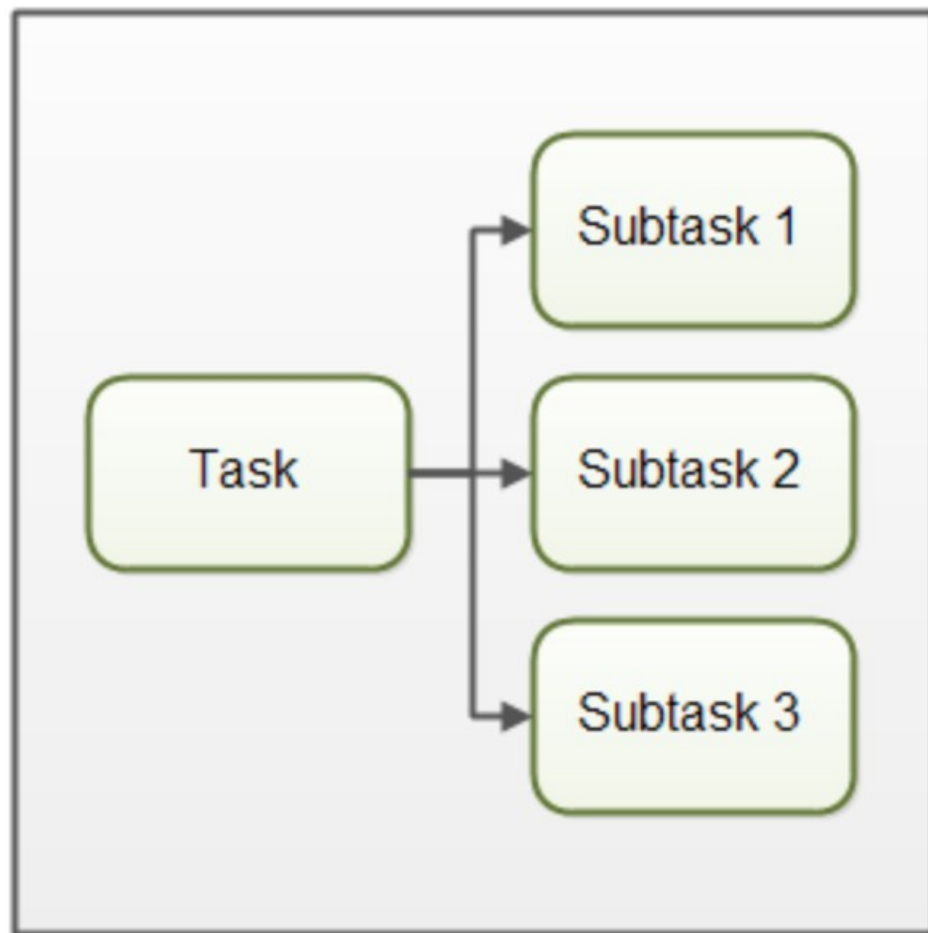


Фигура 1.2 Превключване на задачи на четири задачи на две ядра

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.

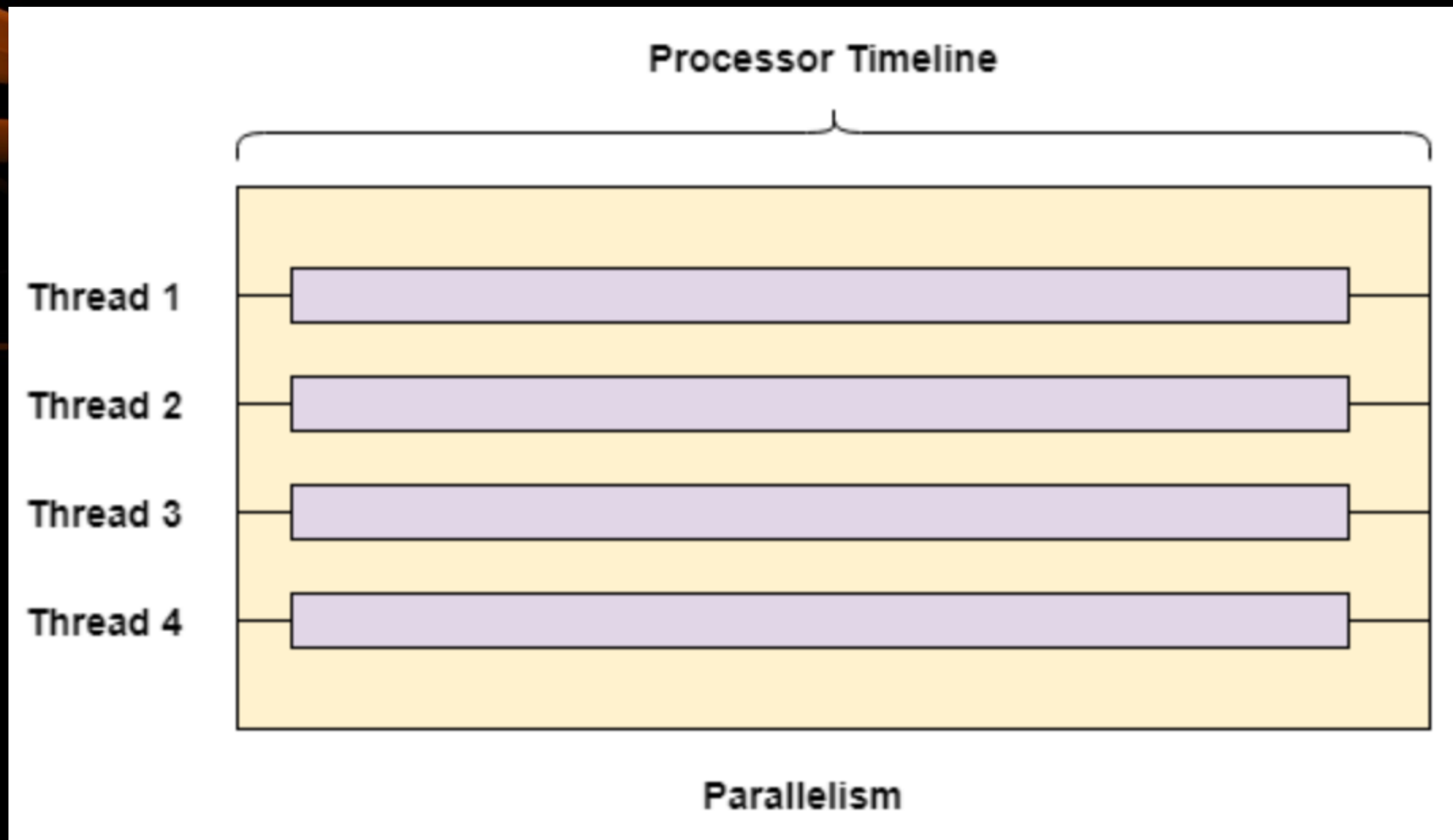
При паралелно изчисление две изчисления напредват едновременно - това е буквално по едно и също време. Това не е възможно с един процесор и вместо това изисква многоядрена настройка. Паралелността означава, че множество процеси или нишки напредват паралелно, т.е. нишките се изпълняват по едно и също 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.



Parallelism:

Each task is broken into subtasks which can be processed in parallel.

За да постигне истински паралелизъм, приложението трябва да има повече от една работеща нишка или поне да може да планира задачи за изпълнение в други нишки, процеси, процесори, графични карти и т.н. Паралелизмът е създаден с цел увеличаване на изчислителната скорост чрез използване на множество процесори. Това е техника за едновременно изпълнение на различни задачи в един и същи момент чрез включването на няколко независими изчислителни процесорни единици или изчислителни устройства, които паралелно работят и изпълняват задачи, за да се увеличи изчислителната скорост и да се подобри производителността. Паралелизмът води до припокриване на CPU и I/O дейности в един процес с CPU и I/O дейности на друг процес.



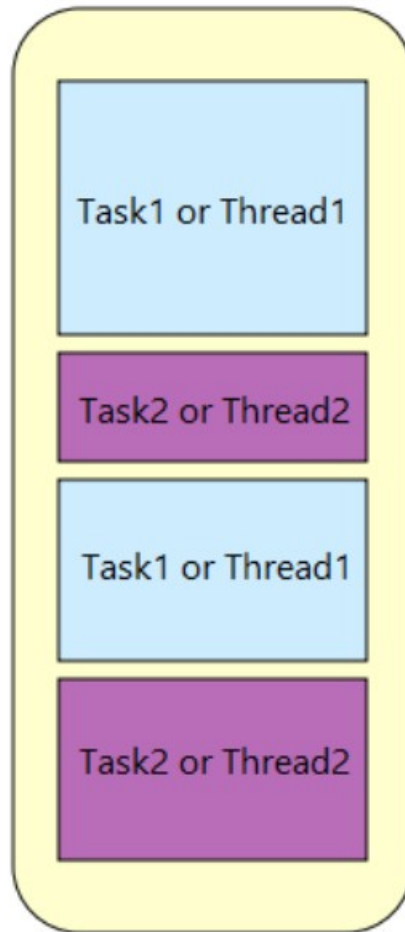
Горната диаграма всичките четири нишки се изпълняват паралелно, т.е. те се изпълняват едновременно.

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.

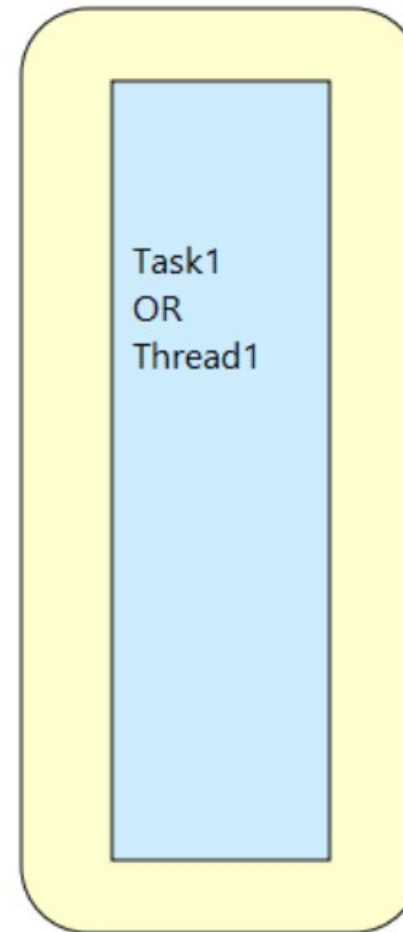
Concurrent

CPU

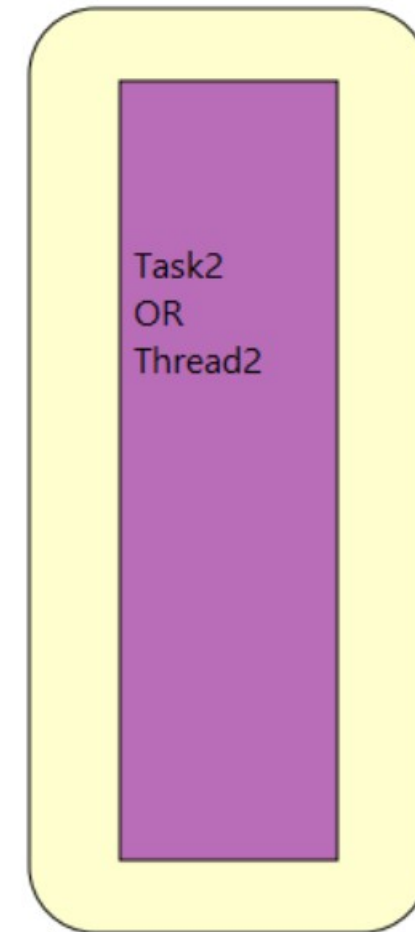


Concurrent [also] Parallel

CPU1



CPU2

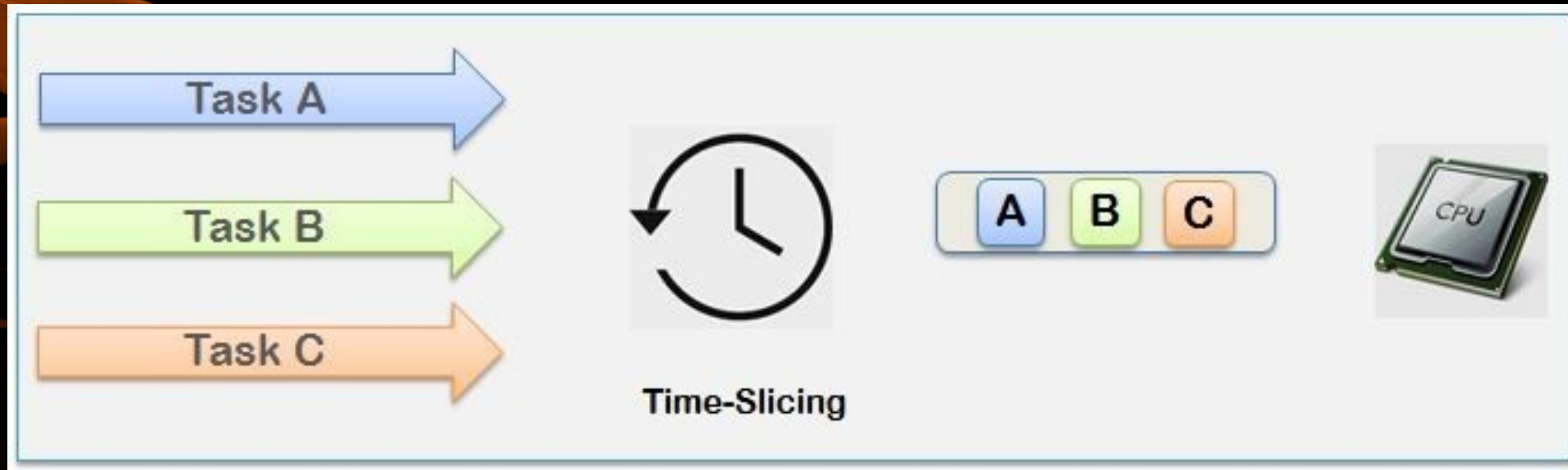


4. Паралелността може да бъде реализирана чрез използване на единична процесорна единица, докато това не е възможно в случай на паралелизъм, който изисква множество процесорни единици. Паралелизмът се получава чрез използване на множество процесори, като многопроцесорна система и управление на различни 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.

Други примери

Паралелността е, когато две задачи могат да стартират, изпълняват и завършват в припокриващи се периоди от време. Това не означава непременно, че някога и двете ще работят едновременно. Ако имаме само едноядрен компютър, тогава никога не можем да стартираме паралелна програма на него. Това е така, защото имаме един процесор и той може да прави само едно нещо наведнъж. Можем обаче да изпълняваме паралелни програми на едноядрен компютър. Компютърът разпределя времето между различните задачи, поддържайки илюзията, че те се изпълняват паралелно. Представете си, че имате две задачи: Задача А, чието изпълнение отнема десет секунди, и Задача Б, която отнема петнадесет секунди. На един процесор извършването на А и Б ще отнеме 25 секунди. На компютър с два процесора, които работят независимо, извършването на А и Б ще отнеме само петнадесет секунди.

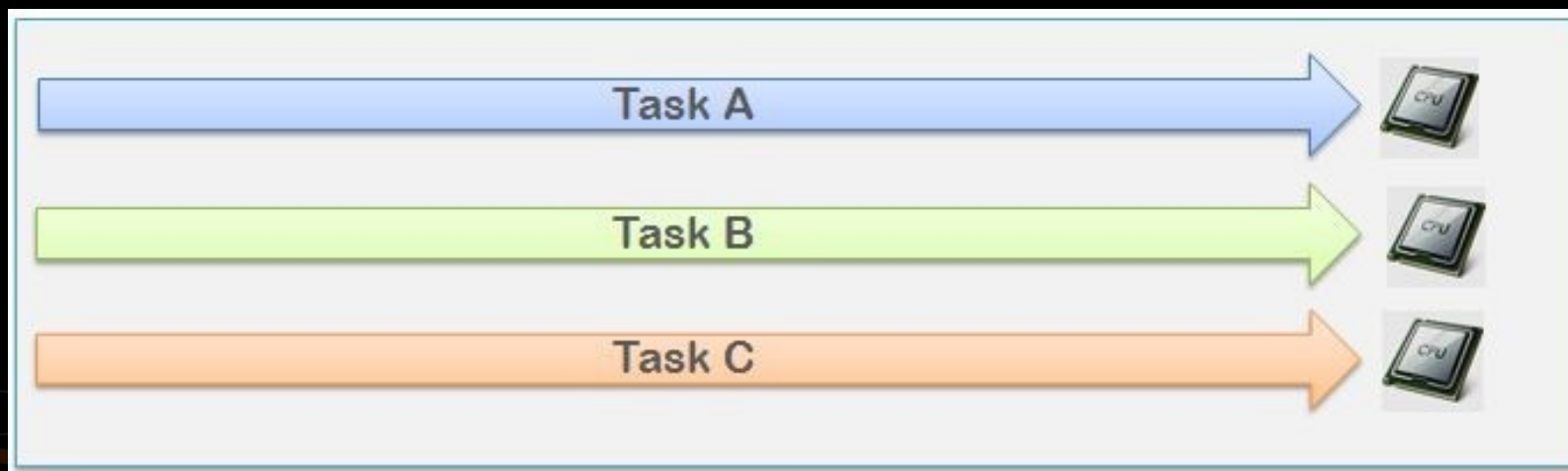


Операционна система, която поддържа превантивна многозадачност, създава ефекта на едновременно изпълнение на множество нишки от множество процеси. Тя прави това, като разделя наличното процесорно време между нишките, които се нуждаят от него, разпределяйки процесорното време за всяка нишка една след друга.

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.

Паралелизъм:/Parallelism/

Паралелизъмът е един от начините за прилагане на паралелност. Това е, когато задачите буквално се изпълняват по едно и също време, напр. на многоядрен процесор. Терминът паралелизъм се отнася до техники за ускоряване на програмите чрез извършване на няколко изчисления паралелно. Това изисква хардуер с множество процесори. Едновременно програма може да работи по-бързо в компютър, който има повече от едно ядро (или CPU), като изпълнява всяка задача на отделен CPU.



Паралелност срещу паралелизъм

Паралелността и паралелизмът имат до голяма степен припокриващи се значения по отношение на многонишковия код. Всъщност за мнозина те означават едно и също нещо. И двата термина се отнасят за изпълнение на множество задачи едновременно, като се използва наличният хардуер, но паралелизмът е много по-ориентиран към производителността. Когато се говори за паралелизъм, основната грижа на програмистите е да се възползват от наличния хардуер, за да се увеличи производителността на обработката на масивни данни, а когато се говори за паралелност, основната им грижа е разделянето на проблемите и задачите.



Въпроси?

