Конкурентно програмиране(MultiThreading) доклад

1 определение

Конкурентно програмиране e програмна парадигма за създаване на компютърни програми, в които многобройни изчисления могат да се изпълняват в застъпващи се периоди от време (конкурентно), вместо последователно (където едно изчисление трябва да завърши преди друго да започне). Конкурентните изчисления могат да се изпълняват на един или повече процесори на едно компютърно устройство, или на процесори разпределени в мрежа от устройства.

2 конкурентност

Конкурентните изчисления са свързани с паралелните изчисления, но двете са различни, въпреки че концепциите споделят дефиницията „множество процеси, които се изпълняват по едно и също време“.

Паралелните изчисления се изпълняват буквално по едно и също време – примерно, на различни процесорни ядра на устройство с многоядрен процесор – с цел ускоряване на изпълнението. Те не са възможни на едноядрен процесор, защото на такъв може да се извършва само едно изчисление в даден момент. Контрастно на това, конкурентните процеси могат да се изпълняват на едно ядро, като се раздели изпълнението на всеки процес на стъпки отнемащи определено време. По този начин, на определен интервал от време се преминава от изпълнението на стъпка от един процес към стъпка от изпълнението на друг процес. Така, във всеки един момент се изпълнява само един процес, но множество процеси са частично изпълнени.

3 изпълнение на програма

Има различни методи за осъществяване на конкурентно програмиране. Например изпълнение на всяко изчислително изпълнение като процес на операционната система или осъществяване на изчислителен процес като набор от нишки в рамките на един процес на операционната система.

Конкурентно взаимодействие и комуникация

Споделена памет: конкурентните компоненти си комуникират чрез промени по съдържанието на обща памет (примерно Java и C#). Този вид конкурентно програмиране обикновено изисква някаква форма на заключване (например mutexes, semaphores или monitors), за да може да се координират отделните нишки. Програма, която изпълнява всички от тях, се нарича thread-safe.

4 процес

5 видове блокиращи операции

Операция,която блокира продължението на изпълнение на нишката до настъпване на дадено условие

6 нишка

Нишката е прост път на изпълнение в рамките на даден процес. Нишката е толкова мощна, колкото процес, защото нишката може да направи всичко, което процесът може. Нишката е лек процес и се нуждае от само по-малко ресурси. Нишките могат да четат и пишат към същите променливи и променливи структури от данни. Нишката може лесно да комуникира между нишките.

7 връзка между процес и нишка

1. Всички нишки на програмата са логически включени в процеса.
2. Процесът е тежък, но нишката е претеглена.
3. Програмата е изолирано изпълнително устройство, докато нишката не е изолирана и споделя паметта.
4. Нишката не може да има индивидуално съществуване; тя е свързана с процес. От друга страна, процесът може да съществува индивидуално.
5. В момента на изтичане на нишката, свързаният с него стак може да бъде възстановен, тъй като всяка нишка има свой собствен стек. Обратно, ако процесът умира, всички нишки умират, включително процеса.