

Глава 3: Конкурентно изпълнение

Интерференция: Проблем: Ако две нишки едновременно четат/пишат в споделен обект, могат да възникнат грешки.

Пр.: Една нишка чете стойност, друга я променя, преди първата да запише резултата.

Пр.Орнаментална градина: Две нишки (входове East и West) увеличават броя на посетителите. Проблем: Броячът не отчита правилно, защото двете нишки записват стойности едновременно.

Решение: Взаимно изключване

Синхронизиране в Java: Методите се маркират с ключовата дума **synchronized**.

```
synchronized void increment() {  
    value++;  
}
```

По този начин само една нишка може да достъпи метода.

FSP: Моделът включва заключване:

LOCK = (acquire -> release -> LOCK).

TURNSTILE = (go -> RUN),

RUN = (arrive -> INCREMENT | end -> TURNSTILE),

**INCREMENT = (value.acquire -> value.read[x:T] -> value.write[x+1]
-> value.release -> RUN).**

Конкурентното изпълнение - няколко процеса могат да се изпълняват едновременно или чрез редуване на техните действия.

Интеракция между процесите: Процесите могат да си взаимодействат чрез **споделени действия**, като това взаимодействие трябва да бъде контролирано.

Паралелна композиция на асинхронни процеси: При паралелната композиция действията на различни процеси могат да се **редуват (interleaving)**, като всеки процес запазва своя ред на изпълнение.

Интеракция чрез споделени действия: Процесите могат да синхронизират действията си чрез **споделени събития**.

Етикетиране и преименуване на действия: Действията на процесите могат да бъдат **етикетирани** или **скрити**, за да се моделират различни аспекти на взаимодействието.

Структурни диаграми: Използват се за графично представяне на паралелната композиция и взаимодействията между процесите.

Многонитови програми в Java: В Java конкурентността се реализира чрез **нишки (threads)**, които могат да се изпълняват паралелно или чрез редуване на действията им.

Разлики между конкурентност и паралелност

Конкурентност: Логически едновременно изпълнение. Не изисква множество процесорни елементи. Реализира се чрез **редуване на изпълнението** върху един процесорен елемент.

Паралелност: Физически едновременно изпълнение. Изисква множество процесорни елементи или независими устройства. И двете изискват **контрол на достъпа** до споделени ресурси.

Термините често се използват взаимозаменяемо.

Паралелна композиция и редуване на действия

Ако P и Q са процеси, тогава $(P \parallel Q)$ представлява тяхното конкурентно изпълнение. \parallel е операторът за паралелна композиция

Конкурентност: редуването на действия в споделена среда.

Паралелна композиция: действията на два процеса могат да се редуват по произволен начин.

Нишките в Java-мощен инструмент за конкурентно програмиране.

Композитни процеси - декартово произведение на примитивни процеси. Примитивните процеси се описват чрез действия (action prefix) и избор (choice). Композитните процеси се създават чрез паралелна композиция на примитивни процеси. **Ограничение:** Моделите, описани чрез FSP (Finite State Processes), са крайни.

Алгебрични закони за паралелната композиция

Комуникативност (Commutative): $(P \parallel Q) = (Q \parallel P)$ - Редът на процесите в паралелната композиция няма значение.

Асоциативност (Associative): $(P \parallel (Q \parallel R)) = ((P \parallel Q) \parallel R) = (P \parallel Q \parallel R)$

Начинът, по който групираме процесите, не променя резултата.

процеси - конкурентно изпълнение и интерливинг (преплитане на действия). Взаимодействие между процеси.

Модели: паралелна композиция на асинхронни процеси, интерливинг на действия, споделени действия, етикетиране на процеси и преименуване на действия, структурни диаграми.

Практика: многонитови Java програми.

Конкурентност: Логически едновременно изпълнение, което не изисква множество процесорни елементи (PEs). Изисква интерливинг на изпълнението на един PE.

Паралелизъм: Физически едновременно изпълнение, включващо множество PEs и/или независими операции на устройства.

Моделиране на конкурентност:

Паралелна композиция - интерливинг на действия

Дефиниране на съставни процеси като декартово произведение на примитивни процеси.

Моделиране на взаимодействие - споделени действия: Споделените действия моделират взаимодействието м/у процесите.

Етикетиране на процеси: Префиксиране на етикети на действия за моделиране на споделени ресурси.

Преименуване на действия: Преименуване на действия за синхронизация на съставни процеси.

Скриване на действия: намаляване на сложността на моделите.

Структурни диаграми: улавяне на структурата на моделите, изразени чрез статични комбинатори: паралелна композиция, преименуване и скриване.

Многонитови програми в Java